APLICACIÓN DE REDES NEURONALES ARTIFICIALES AL TRATAMIENTO DE DATOS INCOMPLETOS

José Blas Navarro Pastor

Tesis doctoral dirigida por los Drs. Josep Maria Losilla Vidal
Lourdes Ezpeleta Ascaso

Departament de Psicobiologia i Metodologia de les Ciències de la Salut
Facultat de Psicologia
Universitat Autònoma de Barcelona

1998

Aplicación de redes neuronales artificiales al tratamiento de datos incompletos

José Blas Navarro Pastor

Tesis doctoral dirigida por los Doctores:
Josep Maria Losilla Vidal
Lourdes Ezpeleta Ascaso

Departament de Psicobiologia i Metodologia
de les Ciències de la Salut

Facultat de Psicologia

Universitat Autònoma de Barcelona

1998

A Laura

Aplicación de redes neuronales artificiales al tratamiento de datos incompletos

José Blas Navarro Pastor

Tesis doctoral dirigida por los Doctores:
Josep Maria Losilla Vidal
Lourdes Ezpeleta Ascaso

Departament de Psicobiologia i Metodologia
de les Ciències de la Salut

Facultat de Psicologia

Universitat Autònoma de Barcelona

1998

Este trabajo ha sido posible gracias a la ayuda DGICYT PM95-0126
del Ministerio de Educación y Cultura

INDICE

1
Introducción

Los avances tecnológicos de la última década han contribuido notablemente a popularizar el análisis estadístico de datos: en pocos segundos se pueden invertir matrices enormes, o estimar complejos modelos a partir de sofisticados procesos iterativos o, en general, realizar cálculos que un humano tardaría días en completar, probablemente con más de un error. Como consecuencia de ello, no son pocos los investigadores que se han decidido a realizar ellos mismos el análisis estadístico de sus datos. Pero no todo son ventajas, ya que los problemas empiezan a aparecer cuando se olvida, como sugiere Cobos (1995), que un ordenador no es más que un "tonto veloz". La potencia de la microinformática no sirve de nada si, por ejemplo, tras ajustar un modelo de regresión, no se sabe interpretar adecuadamente sus parámetros, o todavía peor, si éstos se interpretan sin comprobar previamente las suposiciones sobre la distribución de los datos en que se basan las estimaciones de dicho modelo estadístico. Afortunadamente, en los últimos años se ha ido alcanzando una conciencia general de que el análisis estadístico debe ser realizado por un especialista en la materia, y a raíz de ello cada día son más los investigadores que invierten parte de su formación en la adquisición de conocimientos de tipo metodológico y estadístico. Sin embargo, no se puede decir lo mismo de un aspecto previo al análisis, y en nuestra opinión tan o más importante que éste porque condiciona los resultados, como es el de la calidad de los datos que se analizan. Actualmente, aunque se acepta que las conclusiones de un estudio dependen en gran medida de la calidad de sus datos, en el contexto de la investigación en psicología la verificación de la calidad de los datos sigue ocupando un nivel de prioridad secundario.

Parece incomprensible que se dedique tanto esfuerzo al análisis de datos y tan poco a garantizar la calidad de los mismos, pero aún así, es una costumbre tan extendida que incluso se ha incorporado en nuestra jerga un nuevo término que identifica el problema: GIGO, iniciales de *Garbage in, garbage out*, que se puede traducir como "entra basura, sale basura", en referencia a los resultados que se obtienen a partir del análisis de una matriz de datos de baja calidad.

La calidad de los datos es un concepto amplio que engloba diferentes aspectos. Redman (1992) enumera cuatro dimensiones que hacen referencia al valor que pueden tomar los datos:

1. *Precisión.* Se define como la diferencia entre el valor registrado y el verdadero valor. Cuantificar la precisión de los datos registrados suele ser una tarea difícil, ya que para ello es necesario conocer los valores verdaderos, que por lo

general son desconocidos. Asimismo, puede ser que exista más de un valor verdadero o, en el peor de los casos, que éste no haya sido operacionalmente definido.

2. *Plenitud.* Se refiere al grado en que los datos están presentes en la matriz. En las siguientes páginas se profundiza en el análisis de esta dimensión de calidad.
3. *Actualidad.* Los datos provenientes del estudio de fenómenos dinámicos cambian con el paso del tiempo, provocando un decremento de la precisión. Para minimizar el impacto del cambio se debe reducir tanto como sea posible, y siempre teniendo en cuenta la posible periodicidad del atributo registrado, el intervalo temporal entre la recogida de información y la elaboración de conclusiones.
4. *Consistencia.* En sentido popular consistencia significa que dos o más cosas no entren en conflicto entre sí. Este uso es extensible al contexto de la calidad de datos, en el que dos datos observados son inconsistentes si ambos no son correctos cuando se consideran simultáneamente.

En lo referente a la plenitud de los datos, el problema de los datos incompletos es tan antiguo como la estadística. Hasta hace relativamente poco tiempo los investigadores recurrían a soluciones específicas en la materia que estaban estudiando; así, por ejemplo, en el campo de la psicopatología, la revisión que Vach y Blettner (1991) llevaron a cabo sobre la literatura publicada entre 1976 y 1991 pone de manifiesto que los autores no comentan cómo afrontan el problema de los datos faltantes. La estrategia de análisis de datos incompletos más utilizada, implementada como opción por defecto en los paquetes estadísticos de uso más habitual, consistía, y lamentablemente todavía consiste, en eliminar del análisis estadístico los registros que presentan algún valor faltante[1] (método *listwise*). Como señalan Graham, Hofer y Piccinin (1994), Huberty y Julian (1995), Orme y Reis (1991), Smith (1991) y Whitehead (1994), entre otros, este método implica una sustancial mengua del tamaño muestral, que conlleva a su vez una reducción en la precisión de las estimaciones de los parámetros poblacionales y, además, una disminución de la potencia de las pruebas estadísticas de significación, por lo que en general es desaconsejable. Por otra parte, existen determinados tipos de diseños, como los de caso único o ciertos diseños observacionales en los que, obviamente, es inviable la aplicación del método *listwise*.

---

[1] En este trabajo se utilizarán de forma indistinta los términos valor faltante, valor perdido, valor desconocido, valor ausente y el término inglés valor missing.

No obstante, en las dos últimas décadas se ha producido un importante cambio. Actualmente se dispone de soluciones estadísticas para prácticamente cualquier problema de datos incompletos, y algunas de ellas se hallan implementadas en los paquetes estadísticos de uso más habitual. Ya no hay ninguna excusa legítima para continuar empleando métodos como la comentada eliminación de registros.

1.1. DELIMITACIÓN DEL CONCEPTO DE DATOS INCOMPLETOS

En el presente trabajo, el término "datos incompletos" hace referencia a la no disponibilidad de determinada información en los sujetos que componen la muestra. A este respecto, seguimos la clasificación propuesta por Azorín y Sánchez-Crespo (1986) y por Groves (1989), quienes, en función del origen de la ausencia de información, diferencian entre:

- *Errores de cobertura.* Se conceptúan generalmente como un sesgo en la estimación de los parámetros poblacionales, entendiendo por población la "población objetivo", compuesta por los sujetos a quienes se desearía inferir las conclusiones. Las discrepancias entre la población objetivo y la población marco, formada por los sujetos a quienes realmente se infiere, dan lugar a los errores de cobertura, que suelen ser debidos a una incorrecta estrategia de muestreo. Los errores de cobertura pueden ser por defecto (omisiones) o por exceso (duplicidades). Para corregirlos, se debe disponer de información sobre las características de los sujetos omitidos y duplicados en el proceso de selección durante la fase de muestreo. Wachter y Trussell (1982) presentan un sencillo pero ilustrativo ejemplo de detección y corrección del error de cobertura debido a un sesgo del muestreo. El objetivo del estudio es estimar la media de altura de la población masculina de Estados Unidos, para lo que se cuenta con la altura de los reclutas que se presentan a las Fuerzas Armadas de dicha nación. Puesto que la legislación exige una altura mínima de admisión, el muestreo realizado no cubre la población objetivo. En el gráfico de la Fig. 1 la zona no sombreada representa la altura de los reclutas, mientras que la zona sombreada representa la altura de los hombres que no se presentan a reclutas. Esta distribución se ha obtenido bajo el supuesto, altamente probable, de que la variable altura en la población norteamericana se distribuye normalmente. La asimetría observada en los datos muestrales y el valor de corte de la altura de admisión son la base para establecer la forma de la distribución completa. En este ejemplo, gracias a que se conoce la causa del error de cobertura, se puede evitar el sesgo que provocaría un análisis convencional.

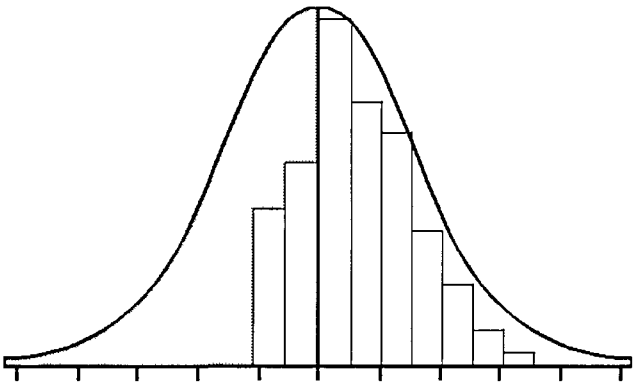


Fig. 1. Distribución observada y poblacional de la altura de la población masculina estadounidense

- *Falta de respuesta total o parcial.* Al igual que los errores de cobertura, se trata de un error de no observación, pero a diferencia de aquellos, se refiere a la imposibilidad de obtener información (parcial o total) de un determinado registro. Para reducir el impacto de la falta de respuesta total se han desarrollado estrategias metodológicas, como por ejemplo la sustitución del sujeto no disponible por otro de similares características, que son adecuadas en determinados tipos de estudios. Por contra, el problema de la información parcial en un sujeto es tratado habitualmente durante la fase de análisis de datos.

  Las causas de la falta de respuesta pueden ser muy diversas, como ilustra Vach (1994) mediante una serie de ejemplos:

  - En estudios retrospectivos, la información se obtiene frecuentemente a partir de documentos, algunos de los cuales se pueden haber traspapelado o ser incompletos.
  - En estudios prospectivos el reclutamiento de sujetos puede durar varios años. Durante este período los avances científicos pueden revelar nuevos factores relevantes que se decida registrar. En esta situación, el valor de las nuevas variables será desconocido en los casos anteriores a dicha decisión, y, por contra, conocido en los nuevos sujetos incluidos en el estudio desde ese momento.
  - Si la medida de una variable es muy costosa, se puede restringir su obtención a un subconjunto de sujetos de la muestra.
  - El informante se puede negar a responder una determinada pregunta por diversos motivos.
  - Pueden ocurrir incidencias durante la fase de recogida y/o registro de los datos que impliquen la irremisible pérdida de información: el extravío de

un formulario de datos, un error del entrevistador, una inconsistencia del instrumento detectada demasiado tarde, etc.

Desde una óptica metodológica se distingue también la falta de respuesta de la inexistencia de respuesta. Un dato faltante es aquel que existe pero que no ha sido recogido por algún motivo concreto (no respuesta del sujeto, reactivo inadecuado, metodología incorrecta, etc.). Por contra, un valor inexistente es aquel que no se ha registrado porque no es susceptible de existir o responde a un reactivo no aplicable (edad de inicio o duración de los síntomas de un trastorno ausente, curso que realiza un niño no escolarizado, etc.).

A nuestro juicio, y desde la perspectiva de la calidad de los datos, la solución al problema de los datos incompletos por falta de respuesta no se debe dar en la fase de análisis, sino que, en la medida de lo posible, se debe planificar durante el diseño de la investigación. En esta línea, Gassman, Owen, Kuntz, Martin y Amoroso (1995) sugieren que, en los ensayos clínicos en particular y en los estudios prospectivos en general, el control de la falta de respuesta se realice principalmente durante la etapa de "Gestión de datos" (*Data management*). Somos conscientes de las limitaciones asociadas a los diseños empleados habitualmente en psicología frente al exhaustivo control disponible en un ensayo clínico, pero creemos que las ideas básicas referentes al control de los datos que se derivan de este tipo de diseños (uso de cuestionarios y entrevistas ampliamente contrastados, auditorias regulares de la base de datos, informes periódicos sobre la calidad de los datos, inclusión de personal especializado, etc.) han de convertirse en un referente para la investigación psicológica. Mientras esto no sea así, seguiremos encontrando multitud de psicólogos que, tras haber empleado semanas aplicando cuestionarios y entrevistas, sólo dispondrán de algunas horas para intentar remediar, mediante procedimientos de carácter estadístico, las limitaciones derivadas de la presencia de valores faltantes en sus datos.

En el presente trabajo nos hemos centrado en el estudio de las estrategias de análisis de la falta de respuesta parcial. Damos por supuesto que en el diseño de la investigación se han tenido presentes los errores de cobertura y de falta de respuesta total. Así, en lo sucesivo, al utilizar el término "datos incompletos" hacemos referencia exclusivamente a matrices con falta de respuesta parcial en uno o en varios registros.

La mayor parte de las técnicas estadísticas para el análisis de datos incompletos asumen que los valores missing tienen un origen aleatorio, atributo que se define a partir de la relación de los valores perdidos con los valores observados de la propia variable que los contiene (variable z) y los de otras variables del estudio (variables x). La clasificación más habitual que divide los valores faltantes en

función de su aleatoriedad fue presentada por Little y Rubin en 1987. Estos autores distinguen entre:

- Datos missing aleatorios (*missing at random-MAR).* La probabilidad de que un valor sea desconocido es independiente de los valores de la propia variable z que los contiene.
- Datos missing completamente aleatorios (*missing completely at random-MCAR).* Cuando los datos perdidos en z no dependen de los valores de otras variables x observadas, se dice que la variable z contiene observaciones aleatorias (*observations at random-OAR*). Rubin (1976) define una variable con datos missing completamente aleatorios (MCAR) cuando es MAR y OAR. Así, con MCAR la probabilidad de que un valor sea missing no depende ni de los valores de la propia variable z que los contiene ni de los valores de otras variables x.
- Datos missing no aleatorios (*missing not at random-MNAR).* La probabilidad de que un valor sea desconocido se relaciona con los valores de la propia variable z que los contiene, y puede o no guardar relación con los valores de otras variables x. Por ejemplo, si z es una medida de educación, la variable tendría datos missing no aleatorios si los individuos con un bajo nivel de educación no informaran sobre ella.

La mayoría de técnicas estadísticas de análisis de datos incompletos asumen, más que comprueban, que los missing son aleatorios, y lo asumen porque para confirmar en la práctica si los missing son o no *MAR* en el sentido definido anteriormente, sería necesario conocer el valor de esos datos desconocidos, y entonces, obviamente, dejarían de serlo (Von eye, 1990). Alguna propuesta en este sentido es errónea y da lugar a confusión. Por ejemplo, Allison (1982) y Rubin (1976) sugieren evaluar la aleatoriedad de los datos ausentes mediante la generación de un nuevo conjunto de variables dicotómicas que tomen el valor cero si el dato es observado y el valor uno si el dato es desconocido. Según esta propuesta, los datos missing tendrían un origen aleatorio si no existiesen diferencias, en otras variables, entre los grupos definidos por dichas variables dicotómicas. Este tipo de pruebas no son en realidad un test que identifique si un patrón de datos missing es aleatorio (*MAR*), sino únicamente si las observaciones son aleatorias (*OAR*).

Todas las técnicas de análisis estadístico que se presentan en los siguientes capítulos suponen que los missing son aleatorios (*MAR*). A pesar de que se han desarrollado procedimientos para el análisis de datos con valores missing no aleatorios (*MNAR*), no son incluidos en esta revisión, ya que todos ellos necesitan información precisa sobre el mecanismo generador de la distribución de los datos ausentes que es, por lo general, desconocida.

1.2. ¿SON LOS DATOS INCOMPLETOS UN PROBLEMA?

Excepto en determinados tipos de muestreo que generan un patrón de datos missing predecible, como, por ejemplo, el de formulario triple (Graham, Hofer y Mackinnon, 1996), en general es difícil determinar la magnitud del problema de los datos ausentes. Kim y Curry (1977) sugieren que el tamaño de la muestra, la proporción de datos ausentes, el mecanismo generador de los datos ausentes y el tipo de análisis estadístico a realizar, determinan el grado en que la información incompleta es problemática. Este último aspecto pasa a menudo desapercibido a pesar de la relevancia que tiene. Efectivamente, si la proporción de datos faltantes en cada variable es baja y se asume que los valores missing son aleatorios, éstos no representarán un problema importante en un análisis estadístico de tipo univariado. Sin embargo, en un análisis multivariado la situación anterior puede ser problemática. Así, si por ejemplo se registran diez variables en cien casos, con un 5% de valores faltantes en cada una de ellas, con un patrón de datos missing aleatorio, y con un pequeño solapamiento de éstos entre los casos (es decir, un caso no contiene muchos valores ausentes, sino que éstos se reparten en diferentes casos), es muy probable que, si se aplica el tradicional método de eliminación de registros, casi la mitad de la muestra se pierda en un análisis estadístico multivariado que incluya las diez variables, mientras que en cada análisis univariado sólo se perdería un 5% de casos.

El principal objetivo del presente trabajo se centra en la evaluación del uso de redes neuronales artificiales en el tratamiento de matrices de datos con información faltante en el sentido definido anteriormente, y, más específicamente, en la estimación de un modelo de clasificación de una variable binaria mediante este tipo de modelos de representación distribuida.

Siguiendo una estructura clásica, la tesis se divide en dos grandes bloques: en la primera parte, que incluye los capítulos 2 a 5, se revisan desde un punto de vista teórico las diferentes materias relacionadas con el tratamiento de los valores faltantes. Así, en el capítulo 2 se presentan los principales procedimientos estadísticos para el análisis de datos incompletos; en el capítulo 3 se comentan las principales características de las redes neuronales artificiales; en el capítulo 4 se analizan detalladamente los dos tipos de redes neuronales más empleados en el contexto de la clasificación: las redes perceptrón multicapa y las redes de función de base radial; por último el capítulo 5 comienza estableciendo las diferencias y similitudes entre los modelos estadísticos clásicos y los modelos de red neuronal para, posteriormente, centrarse en la revisión de las estrategias de clasificación de datos incompletos mediante redes neuronales.

La segunda parte de la tesis, que incluye los capítulos 6 y 7, es de carácter empírico. En el capítulo 6 se exponen los resultados y las conclusiones obtenidos a partir de un experimento de simulación diseñado para comparar diversos

métodos de imputación y análisis de datos incompletos, enfatizando la contribución específica de las redes neuronales artificiales. En el capítulo 7 se aplican las conclusiones extraídas del experimento realizado al tratamiento de la información faltante en una matriz de datos reales procedente de una investigación en el ámbito de la psicopatología infantil.

Finalmente, en el capítulo 8 se exponen conclusiones y reflexiones globales sobre las aportaciones de nuestro trabajo desde una perspectiva múltiple.

2

Análisis estadístico de datos incompletos

Al igual que en muchas otras disciplinas científicas con un fuerte componente matemático, la literatura sobre análisis estadístico de datos incompletos se ha visto notablemente incrementada desde inicios de la década de los 70, gracias sobre todo a los avances de la informática, que han convertido en anecdóticos los complejos cálculos que años atrás ocupaban la mayor parte del tiempo dedicado a investigación.

Las aproximaciones estadísticas clásicas al análisis de datos incompletos son recientes si se comparan con su equivalente para datos completos, las primeras revisiones sobre análisis estadísticos con datos incompletos corresponden a Afifi y Elashoff (1966), Hartley y Hocking (1971) y Orchard y Woodbury (1972). Tomando una perspectiva histórica, se ha evolucionado desde soluciones particulares para problemas concretos, hacia modelos y algoritmos de carácter general que permiten tratar la información faltante como información recuperable o imputable.

El tipo de diseño bajo el cual se han obtenido los datos juega un importante papel en la elección de la estrategia de análisis. En diseños experimentales los valores ausentes se dan con mayor frecuencia en la variable resultado que en los factores del experimento, puesto que estos últimos suelen ser fijados por el experimentador con el objetivo de establecer un diseño balanceado, en el que sea sencillo realizar la estimación de los parámetros y errores estándar del modelo mediante el método de mínimos cuadrados en un análisis de la variancia. Es por ello que el análisis de datos incompletos procedentes de un diseño experimental representa un caso particular que es objeto de un apartado específico en el presente capítulo.

Por contra, en diseños de tipo no experimental, los valores missing se acostumbran a hallar tanto en las variables independientes como en las dependientes. Los registros con algún valor faltante en las variables dependientes suelen ser eliminados del estudio, ya que una imputación del valor desconocido a partir de otras variables implicaría un importante sesgo al aumentar ficticiamente la relación entre variables independientes y dependientes. Para tratar los datos faltantes en las variables independientes de un diseño no experimental se han desarrollado un amplio conjunto de técnicas que, a grandes rasgos, se pueden

agrupar en cinco grandes categorías, no excluyentes entre sí en el sentido de que algunas de ellas se pueden aplicar conjuntamente:

- *Procedimientos basados en registros completos.* Consisten simplemente en eliminar del análisis estadístico los registros que presenten valores missing. Esta opción sólo da resultados aceptables cuando el número de datos faltantes es pequeño y tienen un origen aleatorio. Cuando no se cumplen dichas condiciones estos procedimientos pueden conducir a importantes sesgos, motivo por el cual no suelen ser la opción recomendada.

- *Procedimientos basados en la imputación del valor faltante.* Consisten en imputar un valor a los datos missing para, posteriormente, aplicar un análisis estadístico convencional de datos completos. Los procedimientos habituales son la imputación media, la imputación por regresión, la imputación "hot deck" y la imputación múltiple.

- *Procedimientos basados en la función de verosimilitud.* Bajo el supuesto de que el origen de los valores missing es aleatorio, se han desarrollado un conjunto de métodos de estimación en matrices de datos incompletos que se basan en imputar un valor obtenido a partir de la función de verosimilitud calculada con los datos observados.

- *Procedimientos basados en la estimación Bayesiana.* Se caracterizan por incorporar información previa en la estimación de la probabilidad de cada valor de la variable que contiene datos desconocidos.

- *Procedimientos basados en ponderaciones.* Son métodos que incluyen en su cálculo ponderaciones obtenidas a partir de la probabilidad que cada valor de la variable (incluyendo el valor missing) tiene de ser incluido en la muestra. Los procedimientos de estimación por ponderación son muy poco utilizados en la práctica.

Los procedimientos basados en registros completos, en imputación de valor y en estimación máximo verosímil son el principal objeto de estudio de este capítulo.

En determinados tipos de análisis de datos incompletos también conviene distinguir entre diseños transversales y diseños longitudinales. Ello se debe a que en los datos procedentes de un estudio longitudinal suele darse un patrón de valores ausentes particular, caracterizado por un aumento monótono de la cantidad de valores missing con el paso del tiempo, y al hecho de que la disponibilidad de medidas repetidas proporciona información adicional para estimar los valores desconocidos (Von eye, 1990).

2.1. DATOS INCOMPLETOS EN DISEÑOS EXPERIMENTALES

A continuación se presenta el método de estimación de mínimos cuadrados para datos completos, así como algunas soluciones para aplicar este método cuando la variable dependiente contiene valores missing.

2.1.1. Mínimos cuadrados con datos completos

El análisis tradicional de mínimos cuadrados en un diseño experimental permite estimar los parámetros que satisfacen el modelo lineal $Y=Xb+e$, donde Y es el vector de resultados (rango $n \times 1$), X es la matriz de datos (rango $n \times p$), b es el vector de parámetros (rango $p \times 1$), y e es el vector de error (rango $n \times 1$).

Bajo el supuesto de que los errores son independientes y se distribuyen normalmente con media 0 y variancia σ^2, la estimación mínimo cuadrática del vector de parámetros b es la estimación insesgada de menor variancia, y se obtiene a partir de:

$$\hat{\beta} = (X^t X)^{-1}(X^t Y)$$

Mientras que la mejor estimación insesgada de la variancia del error es:

$$\hat{\sigma}_e^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n-p}$$

2.1.2. Mínimos cuadrados con datos incompletos

Ante una matriz de datos procedente de un diseño experimental con valores desconocidos en la variable dependiente, el análisis tradicional de mínimos cuadrados sólo puede ser realizado bajo el supuesto de que los missing son aleatorios (MAR). Si la matriz de datos contiene n registros, de los cuales m tienen valor observado en la variable resultado y $n-m$ tienen valor missing, la aproximación clásica, propuesta por Yates (1933), consiste en:

1. Imputar todos los valores missing con las predicciones resultantes del modelo de regresión lineal $\hat{y}_i = x_i\hat{\beta}_*$, donde el vector de parámetros $\hat{\beta}_*$ se obtiene a partir de la estimación mínimo cuadrática calculada con la matriz de datos con los m valores observados en la variable resultado.
2. Usar el método de análisis de mínimos cuadrados para datos completos a partir de la matriz completa con los m valores originales y los $\hat{y}_i$ valores imputados.

Este método tradicional tiene el inconveniente de generar sumas de cuadrados atribuibles al modelo demasiado grandes, así como estimaciones de la matriz de variancias-covariancias demasiado pequeñas, especialmente cuando la cantidad de datos faltantes es importante.

Healy y Westmacott (1956) describieron un procedimiento de carácter iterativo que unas veces es atribuido a Yates y otras a Fisher. Este método evita los problemas anteriores y proporciona estimaciones insesgadas de los parámetros del modelo lineal de análisis de la variancia. En síntesis, dicho procedimiento consiste en:

1. Sustituir cada valor missing con un valor aleatorio que haya sido realmente observado en algún otro registro.
2. Realizar el análisis de mínimos cuadrados para datos completos, obteniendo una primera estimación del vector de parámetros $\hat{\beta}$.
3. Sustituir los valores inicialmente desconocidos con el resultado de la predicción a partir del vector de parámetros $\hat{\beta}$ obtenido en el punto anterior.
4. Iterar el proceso desde el punto 2 hasta que el vector de parámetros no cambie sustancialmente entre dos iteraciones sucesivas, o lo que es lo mismo, hasta que la suma de cuadrados residual deje de disminuir de forma sensible.

En algunos casos la convergencia puede ser muy lenta, por lo que diferentes autores han sugerido técnicas de aceleración (Pearce, 1965; Preece, 1971), que, aunque mejoran la velocidad de convergencia en determinados ejemplos, en otras situaciones pueden alterar el decremento monótono de la suma de cuadrados residual (Jarrett, 1978).

2.2. DATOS INCOMPLETOS EN DISEÑOS NO EXPERIMENTALES

La cantidad de información recogida en estudios con diseños de tipo no experimental, sumado a otras características de este tipo de estudios como, por ejemplo, los procedimientos de obtención de datos (habitualmente observacionales o de encuesta), hacen que sea habitual partir de matrices de datos con información ausente en diferentes variables independientes. En los siguientes apartados se presentan soluciones al problema de los datos faltantes en las variables independientes de un diseño no experimental.

Ejemplo

Para ilustrar con un ejemplo las técnicas que se describen posteriormente, se utilizará una pequeña matriz de datos con 40 registros y 6 variables. Los datos se obtuvieron a partir de un diseño de casos y controles, con el objetivo de obtener un modelo de clasificación de la presencia/ausencia de una alteración del estado de ánimo (TA) a partir de la puntuación en cinco variables independientes en un conjunto de niños y adolescentes. Las variables independientes registradas son:

- Antecendentes familiares de psicopatología (AF) (0:No, 1:Sí)
- Número de problemas evolutivos tempranos (PET).
- Puntuación en una escala de humor social (EH).
- Puntuación de la madre en un test de rechazo familiar (ERM).
- Puntuación del padre en un test de rechazo familiar (ERP).

Los estadísticos descriptivos básicos de las variables PET, EH, ERM y ERP se hallan en la Tabla 1:

Tabla 1. Estadísticos descriptivos de los datos de ejemplo

	Media		Mediana	Moda	Variancia
	Estadístico	Error Est	Estadístico	Estadístico	Estadístico
Problemas evol. tempr.	1.03	.19	1.00	0	1.21
Escala Humor	19.11	.80	19.00	18	22.57
Rechazo madre	27.99	2.15	27.00	11	156.55
Rechazo padre	29.12	2.18	29.00	18	147.31

	Asimetría		Apuntamiento		Mínimo	Máximo
	Estadístico	Error Est	Estadístico	Error Est	Estadístico	Estadístico
Problemas evol. tempr.	.648	.398	-.924	.778	0	3
Escala Humor	-.156	.398	-.614	.778	9	28
Rechazo madre	.410	.403	-.672	.788	11	55
Rechazo padre	.158	.421	-.843	.821	11	53

El coeficiente de correlación de Spearman entre cada par de variables independientes se encuentra en la Tabla 2:

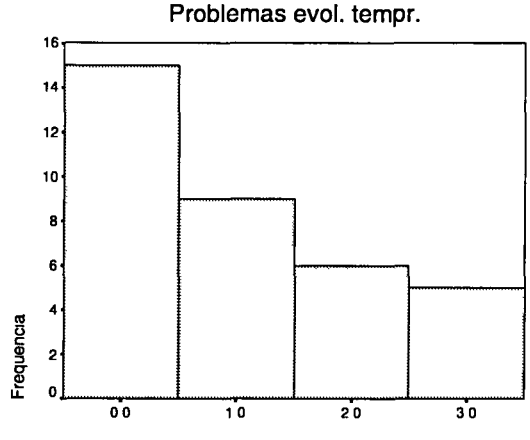
Tabla 2. Coeficientes de correlación en los datos del ejemplo

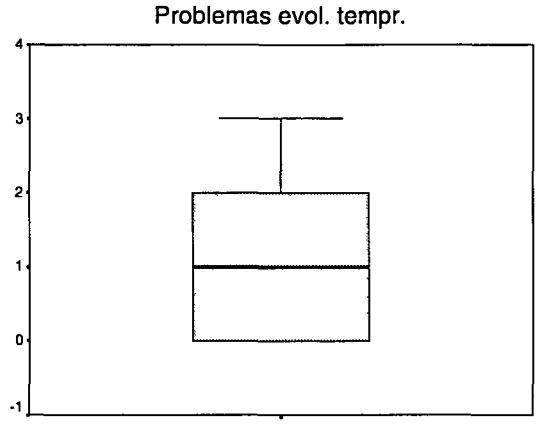
Correlación de Spearman

	Antecedentes familiares	Escala Humor	Rechazo madre	Rechazo padre	Problemas evol. tempr.
Antecedentes familiares	1.000	-.311	.235	.314	.280
Escala Humor	-.311	1.000	-.509	-.404	-.159
Rechazo madre	.235	-.509	1.000	.800	.011
Rechazo padre	.314	-.404	.800	1.000	.038
Problemas evol. tempr.	.280	-.159	.011	.038	1.000

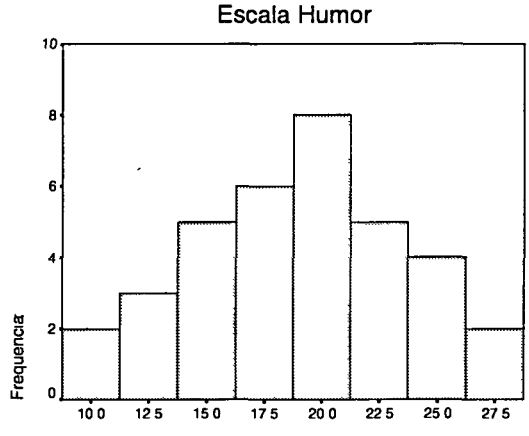
Los histogramas y diagramas de caja de las variables PET, EH, ERM y ERP, y la distribución de frecuencias de las variables TA y AF aparecen en la Ilustración 1:

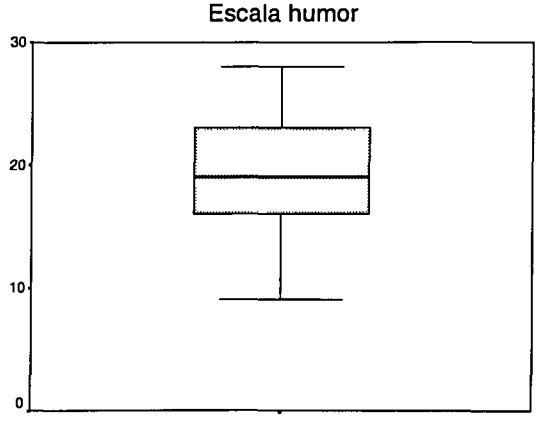
Problemas evolutivos tempranos



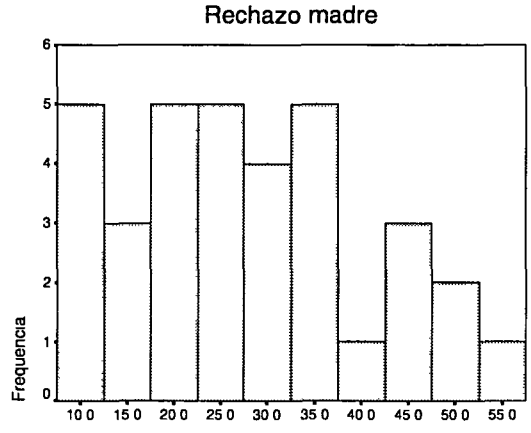


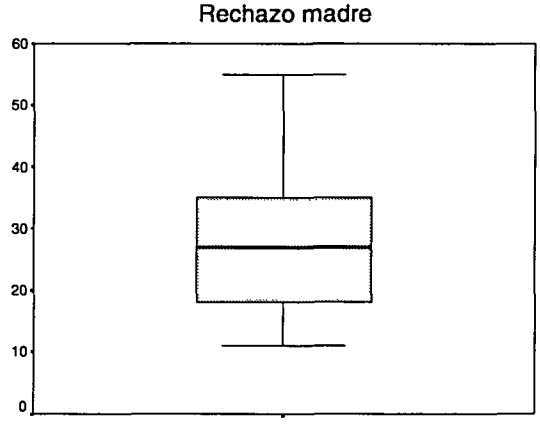
Escala Humor



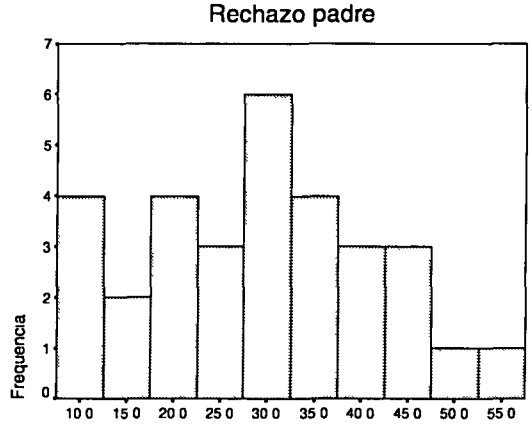


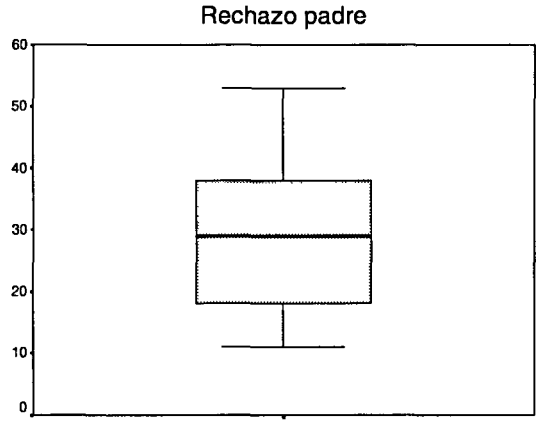
Escala de rechazo de la madre



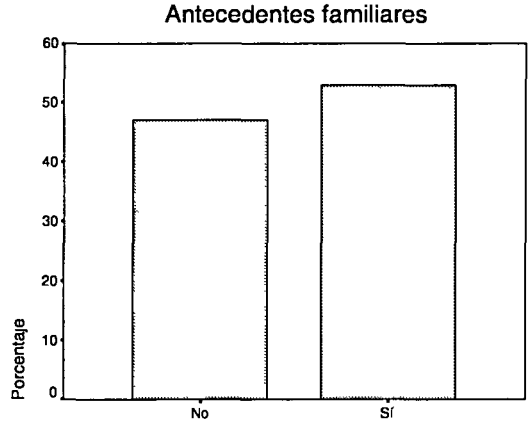


Escala de rechazo del padre





Antecedentes familiares de psicopatología



Trastorno del estado del ánimo

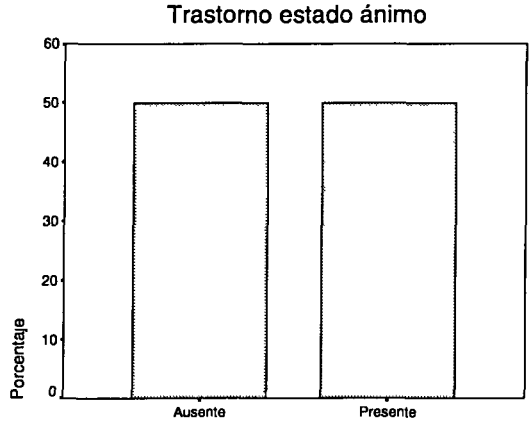


Ilustración 1. Distribuciones de frecuencia y diagramas de caja de los datos del ejemplo

En la diagonal de la Tabla 3 se halla el porcentaje de valores desconocidos de cada variable, mientras que fuera de la diagonal se halla el porcentaje de casos en que una de las variables es desconocida y la otra no. Excepto en la variable ERP, se trata de porcentajes de datos faltantes que se pueden considerar moderados. Los valores desconocidos no se concentran en unos determinados registros, sino que se encuentran dispersos entre los sujetos del estudio, como se deduce del incremento en el porcentaje de valores missing cuando se consideran dos variables respecto a cuando se considera una sola.

Tabla 3. Porcentaje de valores missing en los datos del ejemplo

Porcentaje de missings[a,b]

	EH	PET	AF	ERM	ERP
EH	12.50				
PET	20.00	12.50			
AF	27.50	22.50	15.00		
ERM	22.50	27.50	25.00	15.00	
ERP	25.00	35.00	27.50	12.50	22.50

a. En la diagonal se halla el porcentaje de missings de la variable

b. Fuera de la diagonal se halla el porcentaje de casos en que una de las variables es missing y la otra no.

Las diferentes combinaciones de datos perdidos se presentan en la Tabla 4. Obsérvese que sólo 21 registros (52.5%) no tienen ningún valor missing, por lo que el tradicional método de eliminación de los registros incompletos, que se presenta detalladamente en el próximo apartado, implica la pérdida de casi la mitad de la muestra. Respecto a las combinaciones específicas de datos ausentes halladas en los datos, no se observa ninguna con una frecuencia especialmente elevada.

Tabla 4. Patrones de valores missing en los datos del ejemplo.

Patrones de missings

Numero de casos	Patrones de missings[a]						Completo si ...[b]
	TA	EH	PET	AF	ERM	ERP	
21							21
3			X				24
1		X	X				27
2		X					23
1		X				X	25
1						X	22
2				X		X	26
2				X			23
1			X	X			27
1				X	X		24
4					X	X	26
1		X			X	X	30

a. Las variables están ordenadas *según los patrones de missings*

b. Número de casos completos si las variables missing en ese patrón (marcadas con X) no son usadas

Se ha comprobado que los datos son OAR y se asume que son MAR, por lo que los missing serán considerados completamente aleatorios (MCAR).

2.2.1. Análisis de datos completos

El análisis de datos completos consiste en la eliminación de los registros que presentan algún valor faltante y en realizar el análisis estadístico con los registros que tienen todas las variables observadas (*listwise*). Contra la simplicidad de ejecución de esta opción cabe objetar la importante pérdida de información que suele conllevar, especialmente si el número de variables es elevado. Sin embargo, no se debe olvidar que, bajo la asunción de que los missing son completamente aleatorios, la pérdida de información no implica un sesgo en las estimaciones de los parámetros, de manera que cuando se cumple el supuesto *MCAR* y la cantidad de valores faltantes es relativamente pequeña, el procedimiento de análisis de los datos completos se puede aplicar como el método de elección más simple.

Los registros con datos incompletos que hayan sido excluidos del análisis estadístico pueden ser utilizados para comprobar que las observaciones son aleatorias. Para ello, un sencillo procedimiento consiste en comparar la distribución de una determinada variable *Y* entre los registros con datos completos y los registros con datos incompletos.

Con los datos del ejemplo, el método de análisis de datos completos ofrece las estimaciones del vector de medias, matriz de variancias-covariancias y matriz de correlaciones que se presentan en la Tabla 5. Para condensar los listados de resultados, la variable AF es tratada como cuantitativa, ya que, al estar codificada con los valores 0 y 1, la media se corresponde con la proporción de casos con valor 1.

Tabla 5. Medias, variancias-covariancias y correlaciones obtenidas con el método de análisis de datos completos (*listwise*)

Medias (listwise)

Casos	AF	EH	ERM	ERP	PET
21	.57	19.57	25.89	28.38	1.14

Covariancias (listwise)

	AF	EH	ERM	ERP	PET
AF	.257				
EH	-.493	20.857			
ERM	1.514	-36.598	157.541		
ERP	2.381	-28.313	120.527	150.756	
PET	.114	-.686	-2.296	-1.525	1.329

Correlaciones (listwise)

	AF	EH	ERM	ERP	PET
AF	1.000				
EH	-.213	1.000			
ERM	.238	-.638	1.000		
ERP	.382	-.505	.782	1.000	
PET	.196	-.130	-.159	-.108	1.000

2.2.2. Análisis de datos disponibles

Una alternativa natural al análisis de datos completos consiste en utilizar en el análisis de cada variable todos los datos de que se disponga (denominados casos "*pairwise*") (Matthai, 1951). Así, el calculo de la media de la variable x_1 se realiza con los registros que tienen valor en dicha variable, mientras que en la media de x_2 intervienen los registros con valor en x_2, aunque no tengan valor en x_1.

Las estimaciones de índices bivariantes se realizan con los registros que tienen valor en ambas variables. Por ejemplo, la estimación de la covariancia entre Y_j e Y_k se obtiene a partir de:

$$s_{jk}^{(jk)} = \sum_{(jk)} (y_{ij} - \bar{y}_j^{(jk)})(y_{ik} - \bar{y}_k^{(jk)}) \Big/ (n^{(jk)} - 1)$$

donde los superíndices entre paréntesis indican los registros que se incluyen en el cálculo, por ejemplo $n^{(jk)}$ simboliza los registros con valor observado tanto en Y_j como en Y_k.

Dixon (1983, 1990) propone una alternativa que permite utilizar toda la información disponible en cada variable, consistente en estimar la covariancia incluyendo las medias calculadas con todos los registros con valor en cada variable:

$$\tilde{s}_{jk}^{(jk)} = \sum_{(jk)} (y_{ij} - \bar{y}_j^{(j)})(y_{ik} - \bar{y}_k^{(k)}) \Big/ (n^{(jk)} - 1)$$

Combinando la primera estimación de la covariancia presentada con las estimaciones de la variancia (s_{jj}^{j} y s_{kk}^{k}), obtenidas a partir de los datos disponibles en cada una de las variables implicadas, se obtiene la siguiente estimación de la correlación:

$$r_{jk}^{*} = s_{jk}^{(jk)} \Big/ \sqrt{s_{jj}^{(j)} s_{kk}^{(k)}}$$

que presenta el inconveniente de poder tomar valores fuera del intervalo (-1,1), ya que los registros empleados en la estimación de la covariación pueden ser diferentes a los empleados en las estimaciones de la variancia. Dicho inconveniente puede ser fácilmente solucionado empleando los mismos sujetos en la estimación de la covariancia y de las variancias:

$$r_{jk}^{(jk)} = s_{jk}^{(jk)} \Big/ \sqrt{s_{jj}^{(jk)} s_{kk}^{(jk)}}$$

Bajo el supuesto de que los missing son *MAR*, a pesar de que los métodos de análisis de datos disponibles conducen a estimaciones consistentes de la covariancia y de la correlación, cuando estos índices se consideran

colectivamente se detectan importantes incongruencias. Los datos del siguiente ejemplo, que incluye tres variables registradas en 12 sujetos, con un tercio de datos perdidos en cada una, así lo demuestran:

Y_1	1	2	3	4	1	2	3	4	-	-	-	-
Y_2	1	2	3	4	-	-	-	-	1	2	3	4
Y_3	-	-	-	-	1	2	3	4	4	3	2	1

Los índices de correlación lineal calculados con los casos disponibles dan como resultado $r_{12}=1$, $r_{13}=1$, $r_{23}=\mathbf{-1}$, valores incongruentes puesto que si $r_{12}=1$ y $r_{13}=1$, ello implica que $r_{23}=\mathbf{1}$.

La elección entre los métodos *listwise* y *pairwise* ha sido tradicionalmente objeto de controversia. Haitovsky (1968) concluye que el método *listwise* es superior en la estimación de coeficientes de regresión cuando los missing son *MAR* y relativamente poco frecuentes. Kim y Curry (1977) muestran, sin embargo, que para correlaciones menores a .70 el método *pairwise* proporciona mejores estimaciones cuando el porcentaje de valores missing se halla entre el 1% y el 10%.

Con los datos del ejemplo, el método de análisis de datos disponibles arroja los resultados de la Tabla 6:

Tabla 6. Frecuencias, medias, variancias-covariancias y correlaciones obtenidas con el método de análisis de datos disponibles (*pairwise*)

Frecuencias (pairwise)

	AF	EH	ERM	ERP	PET	TA
AF	34					
EH	29	35				
ERM	29	30	34			
ERP	27	28	30	31		
PET	30	31	29	26	35	
TA	34	35	34	31	35	40

Medias (pairwise)

	AF	EH	ERM	ERP	PET
AF	.53	19.21	26.65	28.74	1.00
EH	.59	19.11	28.09	29.21	1.10
ERM	.52	19.57	27.99	29.49	1.07
ERP	.48	19.71	27.29	29.12	1.12
PET	.57	18.90	26.96	27.80	1.03
TA	.53	19.11	27.99	29.12	1.03

← *Media de la variable en columnas cuando la variable en filas tiene valor.*

Covariancias (pairwise)

	AF	EH	ERM	ERP	PET
AF	.257				
EH	-.768	22.575			
ERM	1.439	-31.750	156.555		
ERP	1.829	-24.842	130.596	147.313	
PET	.138	-.857	-.257	-.316	1.205

Correlaciones (pairwise)

	AF	EH	ERM	ERP	PET
AF	1.000				
EH	-.322	1.000			
ERM	.249	-.546	1.000		
ERP	.325	-.441	.828	1.000	
PET	.253	-.155	-.019	-.022	1.000

Las medias obtenidas con el método *pairwise* son bastante similares a las obtenidas con la técnica *listwise*, resultado esperado debido a que los datos desconocidos, a pesar de ser bastante frecuentes, son aleatorios. Respecto a la estimación de variancias-covariancias y de correlaciones, en general las diferencias son pequeñas, sin observarse un patrón sistemático de cambio.

2.2.3. Imputación de los valores faltantes

Una solución intuitiva al problema de datos incompletos consiste en asignar a cada missing un valor y realizar, a continuación, los análisis estadísticos como si se tratara de datos completos.

Desde la sencilla imputación incondicional de la media hasta la sofisticada imputación múltiple, se han desarrollado diferentes métodos que se presentan con detalle seguidamente.

La imputación de valor a los datos faltantes es sin duda la opción más utilizada en la práctica, puesto que presenta las siguientes ventajas respecto a otras alternativas: (1) en el caso de diseños balanceados es más sencillo especificar el modelo cuando se dispone de toda la información, (2) es más fácil calcular índices estadísticos, y (3) es más fácil interpretar los resultados.

A pesar de ser un método bastante empleado, la imputación no está exenta de defectos. En palabras de Dempster y Rubin (1983, pág. 45):

> *La idea de la imputación es seductiva y peligrosa. Es seductiva porque conduce al usuario al placentero estado de creer que los datos son completos tras la imputación, y es peligrosa porque reúne situaciones en que el problema es lo bastante pequeño como para legitimar su uso, con situaciones en que las estimaciones aplicadas a datos reales y a datos imputados presentan substanciales sesgos.*

A un primer nivel, las técnicas de imputación de valor se pueden agrupar en dos categorías: (1) las que utilizan exclusivamente información contenida en la propia variable que contiene los valores missing (valor aleatorio, media), y (2) las que utilizan información registrada en otras variables de la matriz de datos (regresión). La elección de uno u otro método depende de diferentes factores como el número de casos, la magnitud de las correlaciones, el número de datos ausentes, el propósito de la estimación, y los patrones de datos incompletos (Dixon, 1990). Así, si por ejemplo algunas variables están correlacionadas entre sí, será preferible la imputación por regresión. Sin embargo, si los valores imputados tienen una elevada variabilidad será más conveniente hacer uso de la imputación múltiple (Schafer, 1997).

De los diferentes métodos tradicionales de imputación que se presentan a continuación, cabe destacar los dos de imputación de la media, por ser los más difundidos e implementados en los paquetes estadísticos convencionales, así como la imputación múltiple, por representar un importante cambio cualitativo en la estrategia de asignación de valores (Little y Rubin, 1987; Von Eye, 1990).

2.2.3.1. Imputación de un valor aleatorio

La imputación de un valor aleatorio a los datos faltantes consiste en reemplazar cada valor ausente por un valor aleatorio. En este procedimiento se pueden distinguir dos formas de generar el valor aleatorio: (1) generación de un valor aleatorio dentro del rango de valores de la variable, asignando la misma probabilidad a todos los valores (V.A.D.U., valor aleatorio de una distribución de probabilidad uniforme), y (2) generación de un valor aleatorio a partir de la función de probabilidad que caracteriza la variable (binomial, multinomial,

normal, etc.) (V.A.D.E., valor aleatorio de una distribución de probabilidad estimada para la variable).

Tanto la imputación VADU como la imputación VADE son opciones poco utilizadas en la práctica, ya que acostumbran a provocar un importante sesgo, especialmente en el caso de variables con una elevada dispersión.

2.2.3.2. Imputación incondicional de la media

La forma más simple de imputación no aleatoria de un valor desconocido consiste en asignar el valor promedio de la variable que lo contiene, calculado en los casos que tienen valor. Si se trata de una variable categórica se imputa la moda de la distribución. Obviamente, la media de los valores observados e imputados coincide con la media observada, mientras que la variancia es:

$$s_{jj}^{(ij)} = \left[\left(n^{(j)} - 1\right)(n-1)\right] s_{jj}^{(j)}$$

que, aunque bajo la asunción de que los missing son MCAR es una estimación consistente con la verdadera variancia, es sesgada porque la reduce en $\left[\left(n^{(j)} - 1\right)(n-1)\right]$. Las correcciones de la variancia (y de la covariancia) por el factor de reducción dan como resultado las estimaciones obtenidas con el cálculo directo mediante métodos para datos completos que, como se ha comentado anteriormente, presentan serias limitaciones.

Con los datos del ejemplo, el método de imputación incondicional de la media ofrece los resultados de la Tabla 7:

Tabla 7. Medias, variancias-covariancias y correlaciones obtenidas con el método de imputación incondicional de la media

Medias (Imputación incondicional media)

Casos	AF	EH	ERM	ERP	PET
40	.60	19.11	27.99	29.12	1.03

Covariancias (Imputación incondicional media)

	AF	EH	ERM	ERP	PET
AF	.246				
EH	-.580	19.681			
ERM	1.516	-23.574	132.470		
ERP	1.357	-17.161	96.910	113.318	
PET	.112	-.670	-.215	-.278	1.051

Correlaciones (Imputación incondicional media)

	AF	EH	ERM	ERP	PET
AF	1.000				
EH	-.264	1.000			
ERM	.266	-.462	1.000		
ERP	.257	-.363	.791	1.000	
PET	.220	-.147	-.018	-.025	1.000

Las medias coinciden con las obtenidas mediante el método *pairwise*, excepto en la variable TA, ya que al tratarse de una variable categórica el valor imputado es la moda. Las variancias-covariancias y las correlaciones son sustancialmente inferiores a las conseguidas con los métodos de datos completos.

2.2.3.3. Imputación estocástica de la media

Una solución al problema de la estimación de la variancia en la imputación de la media consiste en añadir un término estocástico al valor imputado. De esta manera, el valor a imputar es:

$$x_{i_{mis}} = \bar{x}_i + cr$$

donde *c* es un valor constante, que suele ser proporcional a la cantidad en que la variancia es reducida, y *r* es un número aleatorio normalmente comprendido en el intevalo (-1, 1).

Aplicando este método a los datos del ejemplo se obtienen los resultados de la Tabla 8:

Tabla 8. Medias, variancias-covariancias y correlaciones obtenidas con el método de imputación estocástica de la media

Medias (Imputación estocástica de la media)

EH	ERM	ERP	PET
19.14	28.77	29.79	1.02

Covariancias (Imputación estocástica de la media)

	EH	ERM	ERP	PET
EH	19.787			
ERM	-28.578	169.518		
ERP	-13.330	102.191	188.664	
PET	-.738	-1.913	-1.389	1.052

Correlaciones (Imputación estocástica de la media)

	EH	ERM	ERP	PET
EH	1.000			
ERM	-.493	1.000		
ERP	-.218	.571	1.000	
PET	-.162	-.143	-.099	1.000

Las estimaciones de la media presentan cierto sesgo respecto a las obtenidas con el método *pairwise*, debido a que con pocos sujetos con valor ausente, como ocurre en el ejemplo, el azar no equilibra los valores positivos y negativos del término estocástico que se suma a la media. Las variancias y las correlaciones se ven afectadas por la misma cuestión, motivo por el que aumentan y disminuyen sin seguir un patrón determinado.

2.2.3.4. Imputación por regresión: el método de Buck

Buck (1960) propuso un eficiente método de imputación de valores missing que durante años ha sido utilizado con resultados generalmente satisfactorios.

En un conjunto de datos que siguen una distribución normal multivariante con media μ y matriz de variancias-covariancias V, los valores missing de cada patrón de datos ausentes pueden ser imputados con el valor obtenido a partir de un modelo de regresión lineal cuyos coeficientes, que son una función de μ y V, se calculan a partir de otras variables que tienen valor en ese patrón de datos faltantes.

Las variables a utilizar como independientes en el modelo son las más correlacionadas con la variable que contiene los datos faltantes, y pueden ser obtenidas mediante un método de entrada secuencial de variables al modelo de regresión (método *stepwise*). También es habitual realizar la predicción a partir de un modelo que incluya todas las variables de la matriz de datos.

La media y la variancia estimadas sobre el conjunto de valores observados e imputados son consistentes, y aunque la variancia es ligeramente sesgada porque este método la infravalora, la magnitud del sesgo es menor que la obtenida con el método de imputación incondicional de la media.

El método de Buck presenta algunas limitaciones. Por ejemplo, cuando en el conjunto de datos hay variables categóricas con datos faltantes que se han categorizado mediante variables ficticias (*dummy variables*), al predecir mediante regresión lineal el valor de estas variables se pueden obtener valores diferentes a 0 y 1. En estos casos, cuando la variable que contiene los datos

desconocidos es categórica, la solución pasa por realizar la imputación mediante el modelo de regresión logística.

Con los datos del ejemplo, mediante el método de imputación por regresión se obtienen los resultados presentes en la Tabla 9:

Tabla 9. Medias, variancias-covariancias y correlaciones obtenidas con el método de imputación por regresión

Medias (imputación por regresión)

AF	EH	ERM	ERP	PET
.53	19.17	28.11	29.76	1.03

Covariancias (imputación por regresión)

	AF	EH	ERM	ERP	PET
AF	.22				
EH	-.72	19.88			
ERM	1.38	-29.12	141.31		
ERP	1.58	-21.49	110.62	122.73	
PET	.12	-.80	-1.06	-.86	1.05

Correlaciones (imputación por regresion)

	AF	EH	ERM	ERP	PET
AF	1.000				
EH	-.342	1.000			
ERM	.248	-.549	1.000		
ERP	.304	-.435	.840	1.000	
PET	.253	-.174	-.087	-.075	1.000

Las medias obtenidas en las cinco variables son muy próximas a las calculadas con la técnica *pairwise*. Sin embargo, como es previsible, las variancias-covariancias son inferiores, si bien no tanto como las correspondientes a la imputación incondicional de la media.

2.2.3.5. Imputación por regresión estocástica

De forma similar a la imputación estocástica de la media, presentada anteriormente, este procedimiento consiste en reemplazar el valor desconocido con el valor predicho por el modelo de regresión más un valor aleatorio, que es incluido para reflejar la incertidumbre sobre el valor imputado. Es un

procedimiento poco utilizado en la práctica. Una aplicación de la imputación por regresión estocástica se halla en el trabajo de Herzog y Rubin (1983).

Con los datos del ejemplo, el método de imputación por regresión estocástica, añadiendo a la predicción del modelo de regresión un valor aleatorio de una distribución normal, arroja los resultados de la Tabla 10:

Tabla 10. Medias, variancias-covariancias y correlaciones obtenidas con el método de imputación por regresión estocástica

Medias (Imput. regresión estocástica)

AF	EH	ERM	ERP	PET
.58	19.09	28.02	30.32	.92

Covariancias (Imput. regresión estocástica)

	AF	EH	ERM	ERP	PET
AF	.28				
EH	-.60	20.91			
ERM	1.82	-24.59	149.23		
ERP	2.06	-20.42	115.13	127.63	
PET	.12	-.42	-3.93	-3.32	1.37

Correlaciones (Imput. regresión estocástica)

	AF	EH	ERM	ERP	PET
AF	1.000				
EH	-.248	1.000			
ERM	.282	-.440	1.000		
ERP	.345	-.395	.834	1.000	
PET	.191	-.079	-.275	-.252	1.000

Nuevamente se obtienen estimaciones de la media muy semejantes a las obtenidas con los métodos anteriores. En lo referente a las variancias-covariancias y las correlaciones, los resultados muestran un significativo incremento de sus estimaciones, posiblemente debido a que el término estocástico que se añade al valor predicho por regresión es demasiado elevado.

2.2.3.6. Imputación Hot Deck y Col Deck

Bajo el nombre de imputación *Hot Deck* se engloba un amplio conjunto de métodos que se basan en imputar a cada missing un valor seleccionado de una distribución estimada para ese missing. Para implementar esta estrategia, en

primer lugar se genera una distribución de frecuencias basada en los sujetos con valor en la variable, en la que cada valor es ponderado por su frecuencia. Los valores ausentes son reemplazados mediante la selección aleatoria de un valor de dicha distribución.

La imputación *Hot Deck* en ocasiones implica complicados esquemas de selección de las unidades que proporcionarán la distribución de frecuencias. Las principales variantes de este procedimiento de imputación son el *Sequential Hot Deck* (Colledge, Johnson, Paré y Sande, 1978), el *Nearest Neighbor Hot Deck* (Sande, 1983) y el *Hot Deck within adjustment cells* (Little y Rubin, 1987).

La imputación *Hot Deck* tiende a proporcionar estimaciones sesgadas de la media. En cuanto a las estimaciones de la variancia, tienden a ser mayores que la variancia poblacional, aunque pueden ser corregidas a partir de la ponderación previa de cada valor presente en los datos.

La imputación *Col Deck*, por su parte, consiste en imputar a cada valor faltante un valor obtenido de una fuente externa a la matriz de datos, como puede ser una realización previa de la misma investigación.

2.2.3.7. Imputación múltiple

A diferencia de los métodos anteriores, que imputan un valor único a cada dato desconocido, la imputación múltiple se basa en la imputación de más de un valor para cada valor ausente, creando así M conjuntos completos de datos que pueden ser analizados con técnicas para datos completos. Ripley (1995) y Schafer (1997) definen la imputación múltiple como una aproximación al análisis estadístico de datos incompletos basada en procesos de simulación, enfatizando que se trata de un procedimiento que simula varios conjuntos de datos mediante la asignación de diferentes valores a cada dato desconocido y que, posteriormente, los analiza de forma conjunta.

La imputación múltiple fue esbozada por Rubin (1978), aunque el desarrollo de la técnica se produjo a inicios de la década de los 80 (véase p.ej. Herzog y Rubin, 1983; Li, 1985; Rubin, 1986; Rubin y Schenker, 1986). Un ejemplo paradigmático del uso de la imputación múltiple con datos del censo norteamericano se halla en Freedman y Wolf (1995).

El principal defecto de la imputación simple, que al imputar un único valor no refleja la variabilidad muestral del modelo de distribución de los valores faltantes, es solucionado en la imputación múltiple. Las M imputaciones se suelen realizar bajo un único modelo de distribución de los valores faltantes, en cuyo caso los resultantes M análisis de datos completos pueden ser fácilmente combinados para generar estimaciones que reflejen la variabilidad muestral debida a los valores ausentes. Sin embargo, en ocasiones las M imputaciones

provienen de más de un modelo de distribución de los valores faltantes, permitiendo que las estimaciones bajo los diferentes modelos puedan ser contrastadas para mostrar la sensibilidad de la inferencia a cada modelo, cuestión especialmente relevante cuando el mecanismo generador de missing no es ignorable, o dicho con otras palabras, cuando los missing son *MNAR*.

Como se acaba de comentar, teóricamente el vector *M* de valores imputados debe ser creado a partir de uno o varios modelos de distribución de los datos ausentes, aunque en la práctica se suele combinar las predicciones resultantes de dichos modelos con las obtenidas a partir de modelos explícitos, como un modelo de regresión o una imputación incondicional de la media.

El principal inconveniente de la imputación múltiple respecto a la imputación simple reside en el hecho de que requiere un mayor esfuerzo para crear el vector de valores imputados y para analizar los datos, si bien los avances computacionales hacen que este trabajo adicional cada vez tenga un menor coste para el investigador, ya que, básicamente, implica realizar el mismo análisis *M* veces en lugar de una.

El análisis de los *M* conjuntos de datos completos es sencillo. Si denominamos $\hat{\theta}_i$ a las estimaciones realizadas en cada conjunto de datos y $\hat{W}_i$ a sus respectivas variancias, obtenidas mediante métodos para datos completos, la estimación combinada es simplemente el promedio de las *M* estimaciones:

$$\bar{\theta}_M = \sum_{i=1}^{M} \frac{\hat{\theta}_i}{M}$$

Mientras que la variabilidad total asociada a dicha estimación es:

$$T_M = \overline{W}_M + \frac{M+1}{M} B_M$$

Que incluye los componentes intra ($\overline{W}_M$) y entre (B_M):

$$\overline{W}_M = \sum_{i=1}^{M} \frac{\hat{W}_i}{M} \qquad B_M = \frac{\sum_{i=1}^{M} (\hat{\theta}_i - \bar{\theta}_M)^2}{M-1}$$

y el factor de corrección $\frac{M+1}{M}$ por ser M un número finito.

Si el parámetro θ estimado es un escalar, las estimaciones por intervalo y las pruebas de significación se realizan fácilmente teniendo presente que el estimador $\overline{\theta}_M$ es insesgado y sigue una distribución t con grados de libertad:

$$\nu = (M-1)\left(1+\frac{\overline{W}_M}{B_M(M+1)}\right)^2$$

En caso contrario, cuando θ tiene varios componentes, las pruebas de significación para contrastar la hipótesis de nulidad del parámetro estimado deben ser realizadas a partir de las M estimaciones realizadas, y no a partir de la estimación combinada.

2.2.4. Métodos basados en la función de verosimilitud

La inferencia basada en la función de verosimilitud, a diferencia de la inferencia aleatoria, no considera fijos los parámetros que definen la distribución poblacional, sino que se basa en calcular la verosimilitud de diferentes parámetros dados los datos observados en la muestra (Cox y Hinkley, 1974).

El origen de la inferencia basada en la función de verosimilitud se halla en los trabajos de Fisher sobre el concepto de verosimilitud. Si denominamos θ al parámetro o parámetros que definen la distribución poblacional con función de densidad $f(Y|\theta)$, e Y_o al conjunto de datos observados, la función de verosimilitud $L(\theta|Y_o)$ es una función de θ proporcional a $f(Y|\theta)$ que determina la verosimilitud de los posibles valores de θ.

La Fig. 2 ilustra el concepto de verosimilitud. Esta función aparece al invertir el papel de la función de densidad (o función de probabilidad si la variable es discreta) consecuencia del cambio de óptica: en lugar de suponer que conocemos θ y queremos calcular las probabilidades de diferentes Y_o, suponemos que hemos observado una muestra Y_o y evaluamos la verosimilitud de diferentes valores de θ (Peña, 1991).

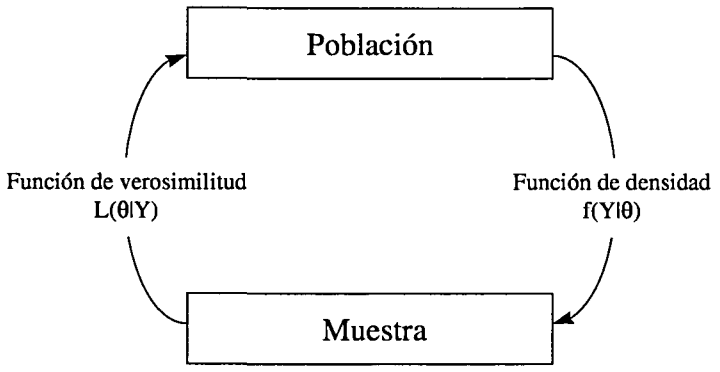


Fig. 2. La función de verosimilitud

La función de verosimilitud se define partiendo de la función de probabilidad de la distribución conjunta de la muestra, que se obtiene como el producto de las probabilidades individuales de cada valor muestral. Así, a partir de la función de distribución de una variable X se puede obtener fácilmente la probabilidad de cada muestra y su producto.

Puesto que la función de verosimilitud refleja la probabilidad que cada parámetro tiene de ser el generador de los datos muestrales observados, el mejor estimador será el resultado de maximizar dicha función de verosimilitud.

De cara a simplificar el cálculo matemático, los estimadores máximo-verosímiles (EMV) se suelen obtener maximizando el logaritmo natural de la función de verosimilitud en lugar de la propia función de verosimilitud.

En principio, el método de estimación máximo-verosímil puede ser empleado sin ninguna modificación cuando la probabilidad de un valor faltante no depende de los propios valores de la variable que lo contiene ni de los de otras variables que intervienen en el análisis, es decir, los missing son *MCAR*.

2.2.4.1. El algoritmo EM

En determinados patrones de datos la función de verosimilitud puede ser compleja, sin un máximo evidente, y con una complicada matriz de varianzas-covariancias. En estos casos, incluso asumiendo que los missing son *MCAR*, los procedimientos para datos completos no permiten la estimación máximo-verosímil de θ. Una solución, propuesta por Anderson (1957), consiste en factorizar la función de verosimilitud aplicando una función monótona que la descomponga en j términos independientes diferenciables y unimodales. Así, la maximización por separado de cada uno de estos términos permite la estimación máximo-verosímil de θ. Sin embargo, este método sólo puede ser aplicado a patrones de datos determinados que permitan la descomposición indicada, como, por ejemplo, patrones de datos monótonos en los que la ausencia de información

es jerárquica entre las variables (ver Fig. 3). Estos patrones son habituales en diseños longitudinales en los que hay un progresivo abandono de los participantes en el estudio.

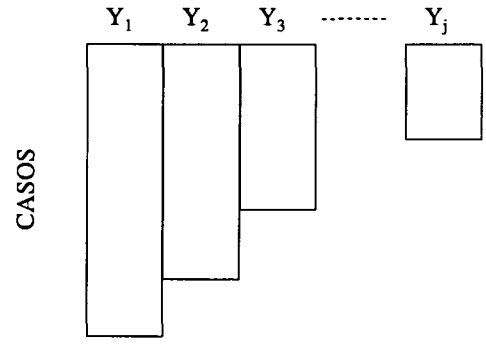


Fig. 3. Representación esquemática de un patrón de datos monótono en que Y_1 es más observado que Y_2, Y_2 es más observado que Y_3, y en general Y_j es más observado que Y_{j+1}

Un procedimiento más general lo constituyen los algoritmos de carácter iterativo. Un ejemplo típico de éstos es el algoritmo de *Newton-Raphson*, en el que el valor estimado en la iteración *i* se obtiene a partir de la estimación *i-1* y la derivada segunda de la función de verosimilitud.

Un algoritmo iterativo con la ventaja de no requerir el cálculo de las segundas derivadas de la función de verosimilitud es el algoritmo *"Expectation-maximization"* (EM).

El algoritmo EM es un algoritmo iterativo general para estimación máximo-verosímil en problemas con datos incompletos en variables cuantitativas. Cada iteración del algoritmo EM consiste en un paso E (expectativa) y un paso M (maximización). Ambos pasos son conceptualmente sencillos y fácilmente implementables en programas informáticos.

La convergencia de la estimación con el algoritmo EM en la verdadera EMV ha sido demostrada (Little y Rubin, 1987). Una desventaja del algoritmo EM es que la convergencia se lentifica proporcionalmente a la cantidad de datos ausentes (Dempster, Lair y Rubin, 1977).

Puesto que el algoritmo EM se basa en la intuitiva idea de imputar los valores faltantes e iterar, no es de extrañar que a lo largo de los años haya sido propuesto en diferentes contextos. La primera referencia parece ser de McKendrick (1926), quien lo considera en el ámbito de una aplicación médica. Hartley (1958) desarrolló extensamente la teoría del algoritmo EM y la aplicó al caso general de datos procedentes de recuentos. Baum, Petrie, Soules y Weiss (1970) usaron el

algoritmo en un modelo de Markov y demostraron matemáticamente sus propiedades en este caso. Sundberg (1974) y Beale y Littel (1975) completaron el desarrollo teórico centrándose en el caso de modelos de distribución normal. El término *"Expectation-maximization"* fue introducido por Dempster, Lair y Rubin (1977), quienes demostraron la generalidad del algoritmo al comprobar en un amplio rango de ejemplos que cada iteración incrementa la verosimilitud del modelo estimado. Estos mismos autores mostraron que el número de iteraciones hasta la estabilización de la estimación aumenta en proporción lineal con el porcentaje de valores faltantes, afirmación contrastada por Wu (1983) con diferentes criterios de convergencia.

En el paso E (expectativa) del algoritmo EM se calculan los valores esperados en la información ausente a partir de los valores observados y las estimaciones actuales de θ, para posteriormente reemplazar la información ausente con dichos valores esperados. Un aspecto crucial es que por información ausente no se entiende cada uno de los valores desconocidos Y_{mis}, sino las funciones de éstos que intervienen en la función de verosimilitud para datos completos, genéricamente denominadas estadísticos suficientes de la función de verosimilitud. Específicamente, si $\theta^{(t)}$ es la estimación actual de θ, el paso E calcula el valor esperado de la función soporte (o de sus estadísticos suficientes) si θ fuera $\theta^{(t)}$, mientras que el paso M (maximización) determina $\theta^{(t+1)}$ maximizando la función soporte obtenida en el paso E.

Las estimaciones iniciales de θ (en el momento t=0) pueden ser realizadas mediante cuatro procedimientos alternativos, cuya operativa ha sido expuesta en el apartado correspondiente a procedimientos clásicos para el análisis de datos multivariantes con datos incompletos: (1) análisis de datos completos, (2) análisis de datos disponibles, (3) imputación de los valores faltantes, y (4) cálculo de las medias y variancias con los valores observados fijando las covariancias a 0. La opción (1) proporciona estimaciones consistentes si los missing son MCAR y hay un número suficiente de registros con datos completos; la opción (2) tiene la ventaja de usar toda la información disponible, pero puede llevar a estimaciones de la matriz de variancias-covariancias no definida positivamente que conduzca a problemas en la primera iteración; las opciones (3) y (4) generalmente conducen a estimaciones inconsistentes de la matriz de variancias-covariancias (Little y Rubin, 1987).

Con los datos del ejemplo, el método de estimación con el algoritmo EM, con la estimación inicial de θ a partir de los registros con datos completos, arroja los resultados de la Tabla 11:

Tabla 11. Medias, variancias-covariancias y correlaciones obtenidas con el algoritmo EM

Medias (algoritmo EM)

AF	EH	ERM	ERP	PET
.54	19.28	28.38	30.21	.97

Covariancias (algoritmo EM)

	AF	EH	ERM	ERP	PET
AF	.26				
EH	-.86	22.37			
ERM	2.12	-33.01	166.77		
ERP	2.39	-25.17	130.62	149.42	
PET	.16	-1.02	-1.96	-1.64	1.22

Correlaciones (algoritmo EM)

	AF	EH	ERM	ERP	PET
AF	1.000				
EH	-.354	1.000			
ERM	.321	-.540	1.000		
ERP	.382	-.435	.827	1.000	
PET	.282	-.195	-.137	-.121	1.000

Las medias estimadas mediante el algoritmo EM no varían respecto a las calculadas con los procedimientos anteriormente revisados. Nuevamente las diferencias se hallan en las estimaciones de la matriz de variancias-covariancias y de correlaciones, ya que el algoritmo EM sobrevalora dichos índices.

2.2.5. Estimación bayesiana

En los métodos de estimación Bayesiana los parámetros son tratados como variables aleatorias caracterizadas por una determinada distribución de probabilidad. La estimación Bayesiana se distingue de los restantes métodos de estimación por utilizar información previa sobre la distribución de los parámetros a estimar. En la práctica, dicha información puede provenir de diferentes fuentes: conocimiento a partir estudios previos, sesgos iniciales del investigador, restricciones sobre los valores que pueden tomar los parámetros, etc. La estimación se realiza a través del cálculo de la probabilidad posterior, aplicando

el teorema de Bayes, que implica multiplicar la probabilidad previa del parámetro por su verosimilitud.

En el contexto de datos incompletos, la información previa que caracteriza los métodos Bayesianos puede usarse como en el caso de datos completos, convirtiéndose en información posterior a través de la verosimilitud. Para ello es necesario que los datos missing sean *MAR*. Sin embargo, mientras que con datos completos se dan soluciones analíticas para el cálculo de la distribución posterior (Box y Tiao, 1973), en problemas con datos incompletos es indispensable el uso de procedimientos iterativos de simulación y de técnicas de muestreo (Schafer, 1994).

3
Redes Neuronales Artificiales

3.1. DEFINICIÓN

No existe una definición universalmente aceptada de red neuronal artificial, aunque la mayoría de personas que trabajan con ellas estarían de acuerdo en definir una red neuronal artificial como un sistema de procesamiento de información formado por un conjunto de procesadores simples organizados en paralelo, cada uno de los cuales tiene una pequeña cantidad de memoria. Las unidades operan sólo con la información disponible localmente, que reciben a través de las conexiones con otras unidades mediante canales de comunicación por los que fluye información de tipo numérico (frente a simbólico). Muchas redes neuronales artificiales tienen una regla de aprendizaje que les permite aprender a partir de los datos que reciben del exterior. Por aprender se entiende el proceso de modulación de las conexiones entre neuronas, ya que es en dichas conexiones, y no en las unidades de procesamiento, donde reside el conocimiento de una red neuronal artificial.

A continuación se presentan algunas definiciones de red neuronal artificial que recogen los aspectos esenciales de éstas:

> *"Los sistemas neuronales artificiales, o redes neuronales, son sistemas celulares físicos que pueden adquirir, almacenar y utilizar conocimiento empírico." (Zurada, 1992, pág. XV).*

> *"Una red neuronal es un circuito inspirado neuralmente que está compuesto por un gran número de elementos de procesamiento simples. Cada elemento opera sólo con información local. Además cada elemento opera asincrónicamente, es decir no hay un temporizador global del sistema." (Nigrin, 1993, pág. 11).*

> *"Una red neuronal es un procesador masivamente distribuido en paralelo que tiene una habilidad natural para almacenar conocimiento a partir de la experiencia para poder usarlo posteriormente." (Haykin, 1994, pág. 2).*

> *"En general, las redes neuronales son (los modelos matemáticos representados por) un conjunto de unidades computacionales simples interconectadas por un sistema de conexiones. El número de unidades puede ser muy grande y las conexiones intrincadas." (Cheng y Titterington, 1994, pág. 2).*

"Las redes neuronales son el resultado de un nuevo enfoque en computación que implica el desarrollo de estructuras matemáticas con la capacidad de aprender. Son el resultado de investigaciones académicas para modelar el aprendizaje del sistema nervioso." (Z Solutions, 1997, pág. 1).

"Las redes neuronales están compuestas por capas de conexiones que relacionan inputs con outputs. Las redes neuronales en la toma de decisiones funcionan como el cerebro, ya que imitan las características de adaptación e inferencia de los sistemas biológicos." (Stern, 1997, pág. 3).

"Una red neuronal es un conjunto interconectado de elementos de proceso simples, unidades o nodos, cuya funcionalidad está basada en las neuronas animales. La habilidad de procesamiento de la red se almacena en las fuerzas de conexión entre unidades, o pesos, obtenidos mediante un proceso de adaptación, o aprendidos de un conjunto de datos de entrenamiento." (Gurney, 1997, pág.1-1).

3.2. FUNDAMENTOS BIOLÓGICOS

Muchas definiciones de red neuronal hacen referencia a su similitud con las redes neuronales que forman el sistema nervioso humano, cuestión que no sorprende si se tiene presente que los orígenes de las redes neuronales artificiales se hallan en los trabajos de los psicofisiólogos y neurofisiólogos de los siglos XIX y XX (Jackson, Ramón y Cajal, Golgi, Luria, etc.), y en las aportaciones de Turing (1937, 1950), quien formuló los fundamentos matemáticos de la computación moderna al idear una máquina simple capaz de realizar cualquier tipo de cálculo siempre que tuviera memoria ilimitada.

El desarrollo temprano de redes neuronales artificiales (o modelos conexionistas o de procesamiento distribuido en paralelo) estuvo muy influido por los imparables avances científicos en el conocimiento del mecanismo director del procesamiento del cerebro humano. Diversas características del cerebro se tuvieron en cuenta al formular los modelos conexionistas, algunas de las cuales, revisadas en los trabajos de Crick y Asanuma (1986), Mira (1993), Nelson y Illingworth (1991), Rumelhart y McClelland (1986) y Sejnowski y Rosenberg (1986), entre otros, son revisadas a continuación.

a. Las neuronas son lentas

Las neuronas son mucho más lentas que los componentes computacionales convencionales (del orden de 10^6 veces más lentas). Puesto que el hombre es capaz de realizar tareas de gran complejidad (procesamiento perceptivo, procesamiento del lenguaje, razonamiento intuitivo, etc.) en unos pocos cientos

de milisegundos, con un procesamiento de tipo serial tales tareas deberían realizarse en no más de unos 100 pasos. Esto es lo que Feldman (1985) denominó la restricción de los 100 pasos del programa. Puesto que es conocido que las neuronas no computan funciones muy complejas, parece improbable que el procesamiento cerebral se realice de modo serial. El cerebro alcanza el éxito mediante un paralelismo masivo, que permite la actividad cooperativa de muchas unidades de procesamiento relativamente simples operando en paralelo.

b. Hay un enorme número de neuronas y de conexiones

Hay un gran número de neuronas implicadas en el procesamiento cerebral (del orden de 10^{10} a 10^{11}) que trabajando en paralelo conceden al cerebro su pasmosa capacidad. Un modelo de computación paralela de un problema de mediana complejidad puede requerir miles de unidades.

De forma complementaria, un importante rasgo del procesamiento cerebral son los grandes abanicos de entrada y salida de información de cada neurona. Las estimaciones para una neurona cortical fluctúan entre 1.000 y 100.000 sinapsis, de donde se deduce que ninguna neurona se encuentra separada por muchas sinapsis de otra neurona. Así si, por ejemplo, cada neurona cortical estuviera conectada con otras 1000 neuronas, formando todas ellas una especie de celosía, todas las neuronas del cerebro se encontrarían, como máximo, a una distancia de cuatro sinapsis una de otra.

c. El conocimiento se almacena y representa de forma distribuida

Frente a las representaciones del conocimiento de tipo localizacionista, en las que cada elemento computacional representa una entidad, existen otros tipos de representaciones más complejas en las que no hay una correspondencia uno a uno entre conceptos y unidades del sistema.

En la representación distribuida cada entidad se representa mediante un patrón de actividad distribuido sobre una gran cantidad de unidades computacionales, y cada elemento computacional participa en la representación de muchas entidades.

Una fuente común de discusión es la idea de que las representaciones distribuidas entran en conflicto con la hipótesis de la localización de las funciones en el cerebro (Luria, 1973). Un sistema que use representaciones distribuidas requerirá también muchos módulos distintos para poder representar simultáneamente conceptos completamente diferentes. Los modelos conexionistas sostienen que las representaciones distribuidas tienen lugar dentro de estos módulos. En una frase, las representaciones son locales a una escala global, pero distribuidas a una escala local.

d. Aprender implica modificar conexiones

Una importante característica del cerebro es que el conocimiento se encuentra en las conexiones en lugar de en las propias neuronas. Los modelos conexionistas suponen que en los estados de las unidades sólo puede ocurrir almacenamiento a corto plazo, el almacenamiento a largo plazo tiene lugar en las conexiones.

e. Las neuronas se comunican por señales de excitación-inhibición

La comunicación entre las neuronas se realiza mediante sencillos mensajes excitatorios e inhibitorios. Esto significa que, a diferencia de otros sistemas paralelos en lo que se transmiten mensajes simbólicos, en los modelos conexionistas se transmite información de tipo numérico. Los símbolos emergen cuando son necesarios a partir de este nivel subsimbólico de representación del conocimiento (Smolensky, 1988).

f. Degradación progresiva

El estudio de las lesiones cerebrales ha demostrado que no existe ninguna neurona cuyo funcionamiento sea esencial para la actividad de ningún proceso cognitivo. El funcionamiento del cerebro se caracteriza por una especie de degradación progresiva, en la que el rendimiento se deteriora gradualmente a medida que más y más neuronas son destruidas, pero no existe un punto crítico en el que el rendimiento se interrumpa. Esta virtud de los modelos conexionistas es una consecuencia directa de la representación distribuida de la información.

g. La información se halla disponible continuamente

Las neuronas proporcionan una salida de información disponible continuamente, es decir, no hay una fase de decisión apreciable durante la que una unidad no proporcione salida de información. Esta característica de los modelos conexionistas contrasta con los modelos por etapas del procesamiento de la información (Sternberg, 1969), que han sido habituales en la historia psicológica.

3.3. EVOLUCIÓN HISTÓRICA

A pesar de que siempre es difícil establecer el origen cronológico de una teoría de gran alcance, y el conexionismo no es una excepción, no son pocos los autores (Catalina, 1996; Cheng y Titterington, 1994; McClelland, Rumelhart y Hinton, 1986; Murtagh, 1996; Smith, 1993) que sitúan los inicios escritos del conexionismo como tal en el artículo de McCulloch y Pitts (1943). En este escrito, de carácter lógico y neurofisiológico, se presentaron la estructura y funcionamiento de la unidad elemental de procesamiento de una red conexionista, y se analizaron sus capacidades computacionales. La neurona de

McCulloch-Pitts (ver Fig. 4), como actualmente se conoce, tiene un funcionamiento muy sencillo: si la suma de entradas excitatorias supera el umbral de activación de la unidad, y además no hay una entrada inhibitoria, la neurona se activa y emite respuesta (representada por el valor 1); en caso contrario, la neurona no se activa (valor 0 que indica la ausencia de respuesta).

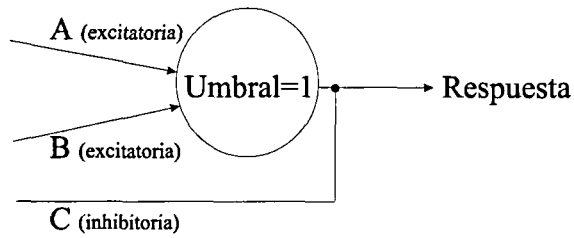
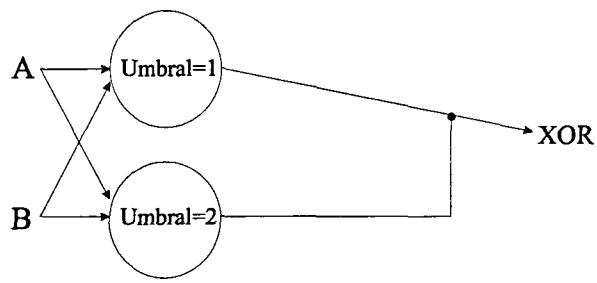


Fig. 4. Neurona de McCulloch-Pitts

Combinando varias neuronas de este tipo con los adecuados umbrales de respuesta, se puede construir una red que compute cualquier función lógica. La siguiente red (ver Fig. 5), por ejemplo, computa la función lógica "O exclusivo" (XOR) a partir de dos unidades elementales interconectadas.



Los 4 posibles patrones de entrada y el patrón de salida XOR de cada uno de ellos son:

A	B	XOR
0	0	0
1	0	1
0	1	1
1	1	0

Fig. 5. Solución de la función XOR con neuronas de McCulloch-Pitts

La idea del umbral de respuesta de la neurona de McCulloch-Pitts ha sido adaptada en la ingeniería de circuitos lógicos (Anderson y Rosenfeld, 1988).

En una línea similar a la de McCulloch y Pitts, y muy influenciado todavía por los descubrimientos neurológicos de la época, Kenneth Craik (1943) defendió

que la capacidad de la mente para representar la realidad se basa en la combinación de patrones coherentes de activación entre las células cerebrales.

Cuando parte de la comunidad científica aceptó que el poder computacional de la mente se basa en su capacidad de conectar neuronas, se planteó el interrogante de cómo se podían establecer y modular las conexiones entre neuronas que dan lugar a patrones de activación coherentes que representan información. En 1949, Donald Hebb postuló un sencillo pero potente mecanismo de regulación de las conexiones neuronales, que constituyó la base de las reglas de aprendizaje que más tarde se desarrollaron. La regla de Hebb, en su versión más elemental, se expresa como sigue:

> *"Cuando la unidad A y la unidad B se encuentren excitadas al mismo tiempo, se aumentará la fuerza de la conexión entre ellas." (en McClelland, Rumelhart y Hinton (1986), pág. 72)*

La propuesta de Hebb es de especial relevancia porque indica que la información necesaria para modificar el valor de una conexión se encuentra localmente disponible a ambos lados de la conexión. La regla de Hebb tiene estrechas similitudes con los algoritmos de aprendizaje desarrollados años después por Kohonen (Kohonen, 1972) y Anderson (Anderson, 1972).

Un año después de la publicación de la regla de Hebb, Karl Lashley (1950) presentó un breve artículo que complementaba las hipótesis de la década anterior sobre el funcionamiento de la mente. Con afirmaciones como *"no hay células concretas que estén reservadas para almacenar memorias específicas"* (Lashley, 1950, pág. 500), manifestaba su creencia de que el conocimiento se halla almacenado de forma distribuida y no localizada.

Los modelos lógico-simbólicos predominantes fueron duramente atacados durante la década de los 50 por ser biológicamente improbables. Pero la crítica fue constructiva, y alrededor del año 1960 Frank Rosenblatt (1958, 1960, 1962), un psicólogo de los laboratorios aeronáuticos Cornell, comenzó a investigar y diseñar el Perceptrón, una red neuronal estructural y funcionalmente más compleja que sus antecesores y con una regla de aprendizaje modificada, que permitía establecer asociaciones entre varios inputs binarios y un output también binario.

El Perceptrón de Rosenblatt está formado por tres capas de unidades (ver Fig. 6). La capa de entrada o retina consiste en un conjunto de unidades de entrada binarias conectadas por conexiones con valor fijo con las unidades de la capa de asociación o de predicados. La última capa es la de respuesta o decisión, cuya única unidad, con salida binaria, tiene conexiones modificables con los predicados de la capa anterior.

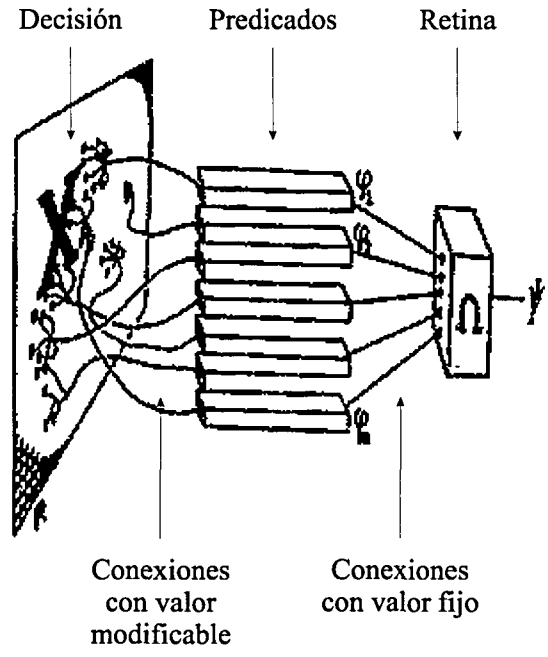


Fig. 6. Perceptrón de Rosenblatt (1962)

Rosenblatt desarrolló varios modelos de aprendizaje que gobernaban la modulación de la intensidad de las conexiones. A grandes rasgos, distinguió entre aprendizaje competitivo (también llamado autoorganizado) y aprendizaje por reforzamiento. En el aprendizaje competitivo, las unidades de asociación que están activas cuando se emite la respuesta incrementan el valor de su conexión con la unidad de respuesta. Por contra, en el aprendizaje por reforzamiento la red recibe información externa sobre cual es la respuesta correcta y, en función del error cometido, se actualizan las conexiones.

La aparición del Perceptrón causó un gran impacto porque por primera vez era posible conceptuar en detalle un mecanismo biológico realista con importantes funciones cognitivas. Aunque finalmente modelos como el Perceptrón no satisficieran las expectativas de Rosenblatt, es innegable que su enfoque del sistema humano de procesamiento como un sistema dinámico, interactivo y autoorganizado se encuentra en el núcleo del procesamiento distribuido en paralelo (PDP) (McClelland, Rumelhart y Hinton, 1986).

En 1960 se propuso un nuevo tipo de unidad de procesamiento, con estructura similar a la del Perceptrón pero con un mecanismo de aprendizaje diferente que permitía también la entrada de información de tipo continuo: la neurona Adaline *(Adaptive linear neuron)* (Widrow y Hoff, 1960). La innovación de esta tipología de neurona se halla en su mecanismo de aprendizaje, denominado

aprendizaje supervisado, regla delta o regla de Widrow-Hoff, que introduce el concepto de reducción del gradiente del error. La regla de aprendizaje de la neurona Adaline es muy similar a la regla de aprendizaje del Perceptrón, pero convenientemente modificada para conseguir que en la modelización de los pesos se incluya el error cometido medido cuantitativamente. La regla de Widrow-Hoff consiste en que la neurona modifica la intensidad de sus conexiones en la dirección y sentido de reducir el gradiente de error, medido como la diferencia entre la respuesta de la unidad y la respuesta correcta. Una red Madaline *(Multilayer Adaline)*, formada por varias capas de neuronas Adaline interconectadas, fue la primera red neuronal artificial aplicada para resolver un problema práctico: eliminar ecos en las líneas telefónicas (Hilera y Martínez, 1995).

Durante el resto del decenio se incrementaron de forma notable las investigaciones sobre el paradigma conexionista y su implementación en redes neuronales. El espectacular auge que desde la aparición del Perceptrón había experimentado el conexionismo fue drásticamente interrumpido en 1969 con la publicación del libro "Perceptrons" (Minsky y Papert, 1969). Marvin Minsky y Seymour Papert fueron los líderes de una escuela de pensamiento rival denominada inteligencia artificial, cuyo objeto de estudio eran los procesos cognitivos superiores como los procesos lógicos, el pensamiento racional y la resolución de problemas. En este sentido, la inteligencia artificial suponía el retorno a los modelos de procesamiento serial de símbolos que Rosenblatt había criticado por ser biológicamente improbables.

En su obra "Perceptrons", Minsky y Papert realizaron un cuidadoso análisis matemático del Perceptrón de Rosenblatt y mostraron qué funciones podía y cuales no podía computar este tipo de unidades. En concreto demostraron que el Perceptrón de Rosenblatt o Perceptrón de una capa, como también se ha llamado para distinguirlo de los posteriores Perceptrones con más de una capa de unidades de asociación, era incapaz de computar funciones matemáticas como la paridad (si un número par o impar de puntos se hallan activos en la retina), la función topológica de la conectividad (si todos los puntos activos se encuentran conectados entre sí directamente o mediante otros puntos también activos) y en general funciones no lineales como la paradigmática función "O exclusivo" (XOR).

La crítica de Minsky y Papert fue tan contundente que, en cierto tono irónico, Hecht-Nielsen (1991) atribuye un móvil conspiratorio a los motivos que llevaron a Minsky a desprestigiar el Perceptrón, basándose en la circunstancia de que Minsky y Rosenblatt fueron compañeros de clase durante la enseñanza superior.

Sin embargo el conexionismo, no murió, porque su esencia, la idea de representación distribuida, no fue criticada. Anderson desarrolló un asociador

lineal de patrones que posteriormente perfeccionó en el modelo BSB *(Brain-State-in-a-Box)* (Anderson, 1977; Anderson, Silverstein, Ritz y Jones, 1977). Casi simultáneamente, en Finlandia, Teuvo Kohonen desarrollaba un modelo similar al de Anderson (Kohonen, 1977), y unos años después, basándose en los avances en las reglas de organización del aprendizaje competitivo conseguidas por Amari (1977), el mismo Kohonen creó un modelo topográfico con aprendizaje autoorganizado en el que las unidades se distribuían según el tipo de entrada al que respondían (Kohonen, 1984). En Japón, Fukushima culminó su trabajo con la invención del Neocognitrón (Fukushima, 1980, 1988; Fukushima, Miyake e Ito, 1983), un modelo de red neuronal autoorganizada para el reconocimiento de patrones visuales que superaba los problemas iniciales con la orientación de la imagen que había sufrido el primitivo Cognitrón (Fukushima, 1975). En Alemania, von der Malsburg (1973) desarrolló un detallado modelo de la emergencia en la corteza visual primaria de columnas de neuronas que responden a la orientación de los objetos.

El trabajo de Stephen Grossberg merece una atención más detallada, ya que supone la culminación de varios años de trabajo sobre modelos con aprendizaje autoorganizado. La producción literaria de Grossberg fue muy prolífica (ver p.ej. Grossberg, 1976, 1978, 1980, 1982, 1987, 1987a), tanto que Klimasauskas (1989) lista 146 publicaciones en las que interviene Grossberg entre 1967 y 1988. Grossberg puso de manifiesto la importancia de los mecanismos inspirados neuralmente en la organización de diversas áreas de la percepción y la memoria. En su Teoría de Resonancia Adaptativa (ART), desarrollada en colaboración con Carpenter (Carpenter y Grossberg, 1987a, 1987b, 1990), explica como la percepción y la memoria se basan en un amplio conjunto de códigos cognitivos que se construyen y modifican a partir de la experiencia. La principal característica de la ART es que las reglas de funcionamiento que rigen la creación y alteración de los códigos cognitivos son muy similares a las de los sistemas biológicos. Efectivamente, los modelos de red recurrentes, inspirados en la Teoría de Resonancia Adaptativa, tienen un flujo de información bidireccional que permite explicar diferentes características del procesamiento humano como su velocidad, su plasticidad, o la influencia que pueden tener las expectativas.

En 1982 coincidieron numerosos eventos que hicieron resurgir de forma definitiva el interés por las redes neuronales. Por una parte, el físico John Hopfield presentó su trabajo sobre memorias asociativas autoorganizadas, fruto del cual se diseñó un nuevo tipo de redes neuronales recurrentes denominadas redes de Hopfield (Hopfield, 1982; Hopfield y Tank, 1985, 1986; Tank y Hopfield, 1987). Según McClelland, Rumelhart y Hinton (1986, pág. 79), el concepto clave de una red de Hopfield es que considera la fase de ajuste de las conexiones como una *"búsqueda de valores mínimos en unos paisajes de energía"*. Según esta idea, cada combinación de pesos de las conexiones de la

red tiene asociada una energía, que resulta de evaluar las restricciones determinadas por los datos de entrada y el resultado producido por la red. El intercambio de información entre unidades se mantiene hasta que el input y el output de cada unidad son el mismo; es decir, en términos de Hopfield se ha llegado a un estado de equilibrio energético. La regla de aprendizaje empleada se basa en la regla de Hebb, convenientemente modificada para que cuando dos unidades se encuentren simultáneamente desactivadas se disminuya la fuerza de conexión entre ellas.

También en 1982, Feldman y Ballard (1982) sentaron las bases de muchos de los principios computaciones del PDP, e introdujeron el nombre de conexionismo en lugar de procesamiento distribuido en paralelo. Además, ese mismo año se celebró la *"U.S.-Japan Joint Conference on Cooperative and Competitive Neural Networks"*, que puede considerarse la primera conferencia relevante sobre redes neuronales (Hilera y Martínez, 1995).

Como consecuencia de estos trabajos, un considerable número de investigadores comenzó a reexaminar los principios del conexionismo. En 1985, Hinton, Sejnowsky y Ackley tuvieron la brillante idea de introducir una capa de unidades ocultas, sin conexión directa con el exterior y con conexiones con las capas anterior y posterior modificables, en una red del tipo perceptrón (más concretamente en una red Madaline), dando lugar a lo que se conoce como redes perceptrón multicapa. La capa de unidades ocultas es generalmente usada para crear un "cuello de botella", obligando a la red a simplificar el modelo generador de los datos y, por tanto, mejorando su capacidad de generalización (Michie, Spiegelhalter y Taylor, 1994).

La aparición de las unidades ocultas con todas sus conexiones modificables hizo necesaria una nueva regla de aprendizaje para modificar los pesos de la capa intermedia. Un grupo de investigación, encabezado por David Rumelhart y James McClelland, examinaron una amplia variedad de diseños de redes. En 1986, tres miembros de este grupo (Rumelhart, Hinton y Williams, 1986, 1986a) anunciaron y publicaron el descubrimiento de una regla de aprendizaje supervisado que permitía discriminar entre clases de patrones que fueran no linealmente separables: la regla de propagación del error hacia atrás o retropropagación del error, o, simplemente, *backpropagation*. Se trata de un refinamiento de la regla delta de Widrow-Hoff que define como modificar los pesos que ligan las unidades ocultas con las unidades de entrada (Nelson y Illingworth, 1991).

El algoritmo de retropropagación del error es un claro ejemplo de invención múltiple (Fausset, 1994; Martín, 1993; Nelson y Illingworth, 1991), ya que, aunque su invención es habitualmente atribuida al grupo de investigación PDP, con Rumelhart a la cabeza, ya en 1974 Paul Werbos lo esbozó en su tesis

doctoral (Werbos, 1974), aunque no gozó de una amplia publicidad, e independientemente fue presentado por Parker (1982, 1985) y LeCun (1986), aunque no con el carácter integrador y el nivel de formalización del grupo PDP, que encuentra su mayor expresión en el trabajo recopilatorio de McClelland y Rumelhart (1988).

En 1987 se adaptó el algoritmo de retropropagación del error a redes recurrentes. El algoritmo se llamó retropropagación recurrente (*recurrent backpropagation*) (Almeida, 1987; Pineda, 1987), y permitió aplicar las redes recurrentes a problemas de reconocimiento de patrones estáticos, que hasta entonces eran tratados únicamente con redes perceptrón multicapa.

En los últimos años, distintos investigadores han desarrollado nuevos métodos a partir de la retropropagación del error. Casi todos tienen como objetivo la disminución del tiempo necesario para entrenar una red. Los principales algoritmos se presentan en el capítulo siguiente.

En la actualidad, los avances tecnológicos han hecho disminuir el interés por la velocidad del entrenamiento. Una breve revisión a la literatura actual pone de manifiesto el interés por aplicar los principios de la estadística Bayesiana a las redes neuronales artificiales, bajo el nombre genérico de *entrenamiento Bayesiano* (ver p.ej. Mueller y Isua, 1995; Neal, 1996; Thodberg, 1996). Se trata de un nuevo enfoque de lo que significa aprender de los datos, desde el que la probabilidad de la muestra observada es usada para representar la incertidumbre sobre la función que la red aprende. La aplicación del entrenamiento Bayesiano se ha concentrado en un tipo concreto de redes: las perceptrón multicapa, aunque puede ser empleado en cualquier tipo de red con tal de que tenga una interpretación estadística. Los principales aspectos del entrenamiento Bayesiano se presentan en un capítulo posterior.

3.4. ELEMENTOS DE LAS REDES NEURONALES ARTIFICIALES

En el diseño y uso de una red neuronal hay implicados un conjunto de elementos comunes a todas ellas y unas reglas que determinan como la información es transformada y el aprendizaje almacenado. La literatura sobre redes neuronales presenta varios manuales que caracterizan este tipo de modelos (Amari, 1977; Feldman y Ballars, 1982; Kohonen, 1977, 1984), aunque ninguno de ellos tiene la integridad y formalización expuestas por Rumelhart, Hinton y McClelland (1986). Dichos autores reconocen ocho aspectos principales en un modelo de procesamiento en paralelo.

a. Unidades de procesamiento

Las unidades de procesamiento de una red neuronal son las encargadas de transformar la información. Su trabajo es simple y consiste en recibir la entrada de sus vecinas, calcular el nuevo valor de activación y computar un valor de salida, que envían a las unidades adyacentes.

En un sistema de representación distribuida las unidades representan pequeñas entidades cuyo análisis conceptual no es significativo. En este caso, el nivel de análisis representativo es el patrón de activación del conjunto de unidades.

b. Estado de activación

Un rasgo, entidad o patrón en la información de entrada está simbólicamente representado en la red mediante una matriz de datos con los estados de activación de las unidades en un determinado momento. En el cálculo de los estados de activación intervienen tanto los inputs que recibe la unidad como su valor de activación previo.

Los estados de activación pueden ser continuos o discretos. Así, en algunos modelos las unidades son continuas, pudiendo tomar cualquier valor real como valor de activación; en otros casos, pueden tomar cualquier valor real comprendido en un intervalo, como por ejemplo el intervalo [0,1]. Cuando los valores de activación están limitados a valores discretos, a menudo son binarios, habitualmente {0,1} o {-1,1}. Cada uno de estos supuestos conduce a un modelo con características diferentes.

c. Salida de las unidades

El valor de salida de las unidades depende de su valor de activación. En ocasiones se aplica una función de tipo umbral, de manera que una unidad no afecta a otra a menos que su valor de activación exceda un cierto umbral. Con mucha frecuencia el nivel de salida es exactamente igual a la activación (función identidad). En determinados modelos se asume que la función a aplicar es estocástica, con un modelo de probabilidad que vincula ambos valores.

d. Patrón de conexiones

Las unidades están vinculadas unas a otras mediante unos pesos de conexión que determinan como responderá la red a un patrón de datos de entrada. El patrón de conexión suele representarse especificando los valores de cada conexión en una matriz de pesos. Las conexiones de signo positivo son excitatorias, en el sentido de que incrementan el estado de actividad de las unidades en que inciden, las de signo negativo son inhibitorias y las de valor cero son nulas.

e. Regla de propagación

La regla de propagación es una función que determina como se combinan las entradas a una unidad con las conexiones para obtener el valor de entrada neto. Habitualmente la regla es simple, y el valor neto de entrada es la suma ponderada por las conexiones de las entradas excitatorias e inihibitorias.

f. Regla de activación

De especial importancia es la función que establece como se fusionan la entrada neta a una unidad y su estado actual para obtener un nuevo estado de activación, que normalmente será además la salida de la unidad. Se trata, a menudo, de una función determinista, aunque algunas redes muy específicas utilizan funciones de activación probabilísticas.

La regla o función de activación suele ser la misma en todas las unidades de una misma capa. En muchas ocasiones se trata de una función de activación no lineal, y específicamente, siempre que el estado de activación tome valores continuos, la función será de tipo sigmoidal.

Las funciones de activación más comunes son las siguientes (Cheng y Titterington, 1994; Fausett, 1994; Fernández, 1993; Nelson y Illingworth, 1991):

- *Función identidad.* Las unidades de la capa de entrada, cuya única misión es pasar la información a la red, no transforman el valor de entrada que reciben al hacer uso de la función identidad.

$$f(u) = u$$

- *Función umbral.* Utilizada sobre todo en redes no supervisadas para convertir la entrada a la unidad de tipo contínuo en una salida binaria (0,1) o bipolar (-1,1).

$$f(u) = \begin{cases} 0 & \text{si } u < \theta \\ 1 & \text{si } u \geq \theta \end{cases}$$

- *Función rampa.* Es la unión de las funciones umbral e identidad. Actualmente son poco utilizadas puesto que han sido reemplazadas por las funciones sigmoidales.

$$f(u) = \begin{cases} 0 & \text{si } u < 0 \\ u & \text{si } 0 \leq u \leq 1 \\ 1 & \text{si } u > 1 \end{cases}$$

- *Función gausiana.* En redes neuronales de función de base radial (RBF), las unidades ocultas utilizan una función gausiana para delimitar el rango de

valores de entrada al que son receptivas. La función gausiana también es usada en algunas redes de tipo probabilístico como las redes de Hopfield.

$$f(u) = \frac{1}{e^{x^2}}$$

- *Función sigmoidal.* Son especialmente útiles en redes perceptrón multicapa. Sus características se presentan detalladamente en un capítulo posterior.

g. Regla de aprendizaje

Uno de los aspectos más investigados en el mundo de redes neuronales se refiere al establecimiento de reglas de aprendizaje que optimicen el funcionamiento de la red mediante la modificación de sus conexiones. Varios modelos de red, y la conocida retropropagación del error es un ejemplo, toman su nombre y taxonomía a partir de su regla de aprendizaje. Es más, una de los principales criterios de clasificación de redes se define en función de esta regla.

h. Ambiente

En la mayor parte de modelos de procesamiento distribuido en paralelo, el ambiente en que se desarrolla la red se caracteriza por una distribución estable de probabilidad del conjunto de posibles patrones de entrada, independientes de las entradas y salidas anteriores del sistema.

3.5. CLASIFICACIÓN DE LAS REDES NEURONALES ARTIFICIALES

Existen numerosos criterios de clasificación de las redes neuronales artificiales (Hilera y Martínez, 1995): según el número de capas, según el mecanismo de aprendizaje, según la dirección en que fluye la información, según el tipo de asociación entre entradas y salidas, etc., aunque posiblemente, el criterio más utilizado es el que hace referencia al mecanismo de aprendizaje. Pero el aprendizaje es un concepto muy amplio que incluye varios aspectos que de nuevo permiten varias taxonomizaciones: redes con aprendizaje determinista o estocástico, competitivo o cooperativo, hebbiano o por corrección de errores, sincrónico o asincrónico, etc. En este trabajo nos centraremos en distinguir los tipos de aprendizaje en función de su relación con la presencia o ausencia de un mecanismo que proporcione a la red el verdadero output que debería dar ante un input determinado. Bajo esta perspectiva, se distingue entre aprendizaje supervisado y no supervisado.

En el aprendizaje supervisado se trata de construir un modelo neuronal que relacione un conjunto de datos de entrada con un conjunto de datos de salida, a partir de un conjunto de datos que contienen ejemplos de la función a aprender. Cada entrada es presentada a la red, que computa un valor de salida que por lo

general no coincidirá con el correcto. Se calcula entonces el error de predicción, que sirve para ajustar los parámetros de la red en el sentido de disminuir dicho error. El ejemplo paradigmático de aprendizaje supervisado es la retropropagación del error, que se presenta detalladamente en el siguiente capítulo.

En el aprendizaje no supervisado, por contra, se pretende que la red descubra las regularidades presentes en los datos de entrada sin que éstos tengan asociados a priori una determinada salida. Dentro de este grupo, las redes con aprendizaje competitivo, como el *Mapa autoorganizado* de Kohonen (Kohonen, 1995), representan un elevado porcentaje de los modelos aplicados.

Sarle (1998) presenta una exhaustiva clasificación de los modelos de red neuronal artificial más conocidos en función de si reciben o no información sobre el output correcto, y de la dirección en que la información fluye por la red: sólo adelante (*feedforward*), o adelante y atrás (*feedback*). Para evitar incorrecciones en la traducción al castellano se presentan los nombres originales en inglés (ver Tabla 12).

Tabla 12. Clasificación de las redes neuronales artificiales

Supervisado		**No supervisado**	
Adelante	**Adelante-atrás**	**Adelante**	**Adelante-atrás**
• *Perceptron* • *Adaline / Madaline* • *Backpropagation* • *Radial Basis Function* • *Cauchy machine* • *Adaptive heuristic critic* • *Time delay* • *Associative reward penalty* • *Avalanche* • *Backpercolation* • *Cascade correlation* • *Learning vector quantization* • *Probabilistic* • *General regression*	• *Brain-State-in-a-Box* • *Fuzzy cognitive map* • *Boltzmann machine* • *Mean field annealing* • *Recurrent cascade correlation* • *Backpropagation through time* • *Real-time recurrent learning* • *Recurrent extendend Kalman filter*	• *Learning matrix* • *Driver-reinforcement learning* • *Linear associative memory* • *Optimal Linear associative memory* • *Sparse distributed associate memory* • *Fuzzy associative memory* • *Counterpropagation*	• *Additive Grossberg* • *Shunting Grossberg* • *Binary adaptive resonance theory* • *Analog adaptive resonance theory* • *Discrete Hopfield* • *Continuous Hopfield* • *Temporal associative memory* • *Adaptive bidirectional associative memory* • *Kohonen self-organizing map*

4
Redes Perceptrón Multicapa y de Función Base Radial

4.1. REDES PERCEPTRÓN MULTICAPA (MLP)

A pesar de la relevancia que la invención del algoritmo de aprendizaje por retropropagación del error (o propagación del error hacia atrás - *backpropagation*) tuvo en el resurgimiento de las redes neuronales, se trata en realidad de una técnica con un componente estadístico fundamental. Más concretamente, la retropropagación del error es un método de modelado estadístico no paramétrico, en el que la función que relaciona entradas con salidas es determinada por los datos en lugar de por el propio modelo estadístico (Werbos, 1991).

El uso del algoritmo de aprendizaje de retropropagación del error define un tipo de redes neuronales específicas denominadas redes perceptrón multicapa, o más simplemente redes MLP (*multilayer perceptron*). La mayoría de redes MLP están formadas por tres capas de unidades (entrada-oculta-salida), si bien se pueden encontrar topologías que incluyan dos o más capas ocultas. Cada unidad de la capa de entrada incorpora a la red el valor de una variable independiente, las unidades de la capa oculta hacen la mayor parte del trabajo de modelado (se revisarán detalladamente más adelante), y cada unidad de la capa de salida calcula el valor de una variable dependiente cuantitativa o de una categoría de una variable dependiente categórica.

Las unidades de cada capa están conectadas con todas las unidades de las capas adyacentes y cada conexión tiene una fuerza denominada peso de la conexión, que es determinada durante la fase de entrenamiento mediante el algoritmo de retropropagación del error. Asimismo, cada unidad oculta y de salida tiene un peso de tipo umbral, que con funciones de activación de tipo sigmoidal debe ser fijado a un valor diferente de cero para garantizar que las zonas de decisión no se solapen innecesariamente (Hornik, 1993). El peso umbral interviene en el cálculo del valor de activación como un término aditivo.

La información se transmite en el sentido de la capa de entrada hacia la de salida, por lo que se trata de redes con conexiones hacia adelante. En este sentido, es importante no confundir el sentido en que fluye la información de los datos con el sentido en que fluye la información del error, ya que es el primero y no el segundo el que determina la clasificación de la red en las mencionadas redes con

conexiones hacia adelante (*feedforward*) frente a las redes con conexiones hacia adelante y hacia atrás (*feedforward/feedback*) (Ripley, 1996).

Como todas las redes neuronales, las redes MLP tienen dos modos de funcionamiento: entrenamiento y predicción. En el modo de predicción la red recibe información de las variables independientes, que es propagada a través de la red proporcionando un valor de la variable dependiente. En el modo de entrenamiento se realiza el procedimiento anterior y a continuación se compara el valor calculado por la red con el valor real, el error de predicción cometido se propaga hacia atrás (desde la capa de salida a la de entrada) para modificar los pesos de las conexiones con el objetivo de minimizar dicho error. La Fig. 7 esquematiza los pasos realizados en cada uno de estos modos de funcionamiento.

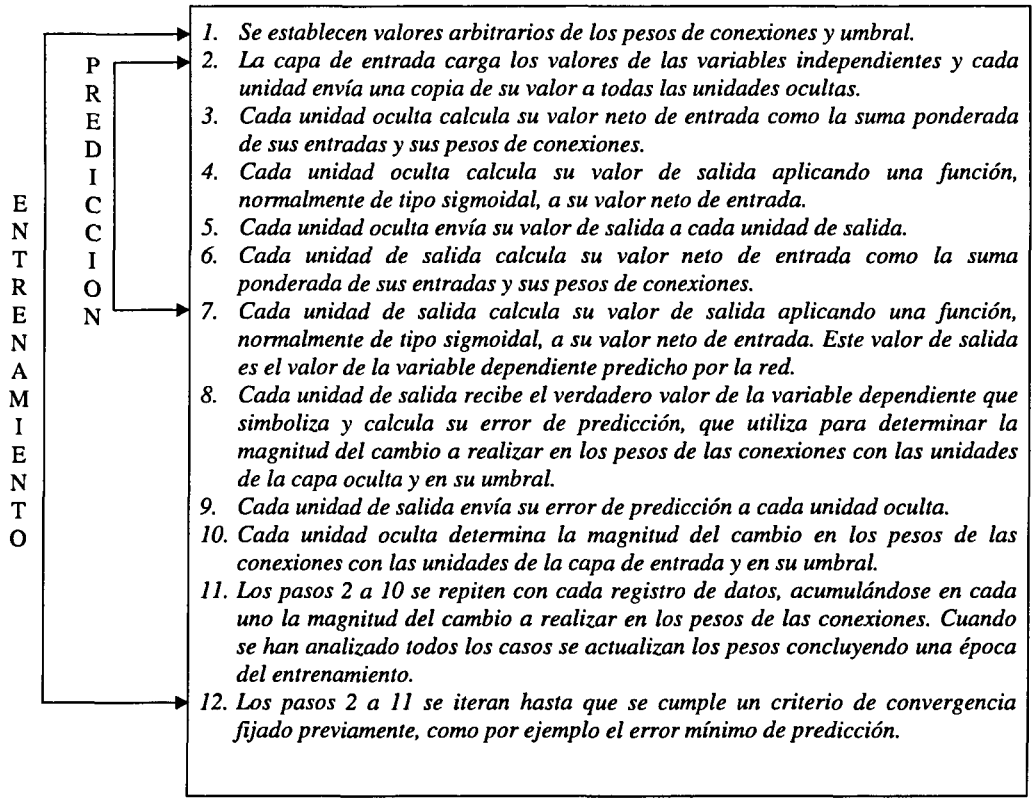


Fig. 7. Operaciones realizadas en los modos de predicción y entrenamiento de una red MLP

En posteriores apartados de este capítulo se detallan cada una de las operaciones que las redes MLP con retropropagación del error realizan para establecer, a partir de un conjunto de datos, la función que relaciona entradas con salidas.

4.1.1. Redes MLP como aproximadores universales de funciones

De entre las virtudes de las redes MLP cabe destacar, por la trascendencia que tiene en su uso aplicado, que se trata de aproximadores universales de funciones. La base matemática de esta afirmación se debe a Kolmogorov (1957), quien constató que una función continua de diferentes variables puede ser representada por la concatenación de varias funciones continuas de una misma variable. Este principio fundamenta la representación distribuida de la información de las redes neuronales artificiales, convirtiendo las redes MLP en particular en herramientas de propósito general, flexibles y no lineales que, con suficientes datos y unidades ocultas, pueden aproximar cualquier función con el nivel de precisión que se desee (Funahashi, 1989; Hecht-Nielsen, 1989; Sarle, 1994; White, 1992). Aquí por aproximar se entiende que la función producida por la red puede ser diseñada para ser arbitrariamente tan próxima como se quiera a cualquier función. Por ejemplo, la función producida por la red no puede ser una recta perfecta, pero sí puede aproximarla tanto como se desee. Demostraciones matemáticas de que las redes MLP son aproximadores universales de funciones se pueden hallar en Cybenko (1989), Hornik, Stinchcombe y White (1989, 1990) o en Irie y Miyake (1988).

La capacidad de las redes MLP para aproximar cualquier función se fundamenta en la incorporación de funciones de tipo sigmoidal como funciones de activación de las unidades de la capa oculta. Las funciones de tipo sigmoidal tienen determinadas características que las hacen especialmente adecuadas. En primer lugar, una función sigmoidal es acotada, es decir su valor nunca excede un determinado límite superior y nunca está por debajo de un determinado límite inferior. En segundo lugar, una función sigmoidal es monótona creciente, es decir su valor $s(u)$ siempre aumenta (o siempre disminuye) según aumenta u. En tercer lugar, una función sigmoidal es continua y, a diferencia de las funciones de tipo escalón, suavizada. Estas tres características hacen que las funciones de tipo sigmoidal sean derivables en cualquier punto, cuestión fundamental para métodos de aprendizaje que, como la retropropagación del error, se basan en el cálculo de derivadas para optimizar el cambio de los pesos de las conexiones.

Existen diferentes funciones sigmoidales, si bien la más usada en redes MLP y en la que se basa el presente capítulo es la función logística, $g(u)$:

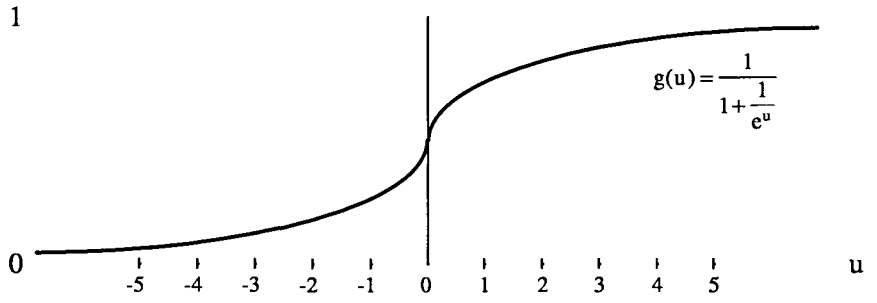


Fig. 8. Función logística

Otras funciones sigmoidales cuyo límite inferior es -1 en lugar de 0, que en ocasiones son usadas como funciones de activación en lugar de la función logística son las siguientes (Cheng y Titterington, 1994; Fausett, 1994):

$$h(u)=\frac{1-\frac{1}{e^u}}{1+\frac{1}{e^u}} \quad ; \quad \tanh(u)=\frac{e^u-\frac{1}{e^u}}{e^u+\frac{1}{e^u}}$$

$$g_2(u)=\frac{2}{1+\frac{1}{e^u}}-1 \quad ; \quad \operatorname{atanh}(u)=\frac{2}{\pi}\arctan(u)$$

Para los propósitos de las redes MLP todas las funciones sigmoidales son válidas. La elección de una u otra se realiza en ocasiones en función de sus límites, ya que el valor de salida de la red debe estar comprendido entre dichos límites.

4.1.2. Predicción con una red MLP

La información contenida en una matriz de datos sufre constantes transformaciones cuando pasa a través de las unidades de procesamiento de una red MLP.

La Fig. 9 esquematiza una sencilla red MLP con una unidad de entrada, dos unidades ocultas y una unidad de salida (red MLP 1-2-1), y muestra como a partir de la información contenida en la variable independiente del modelo (x_l) se genera el valor de salida de la red (z), que representa el valor predicho para la variable dependiente. En la Fig. 9 se presenta también la notación que utilizaremos para representar los pesos de tipo umbral y de tipo conexión, los valores de entrada y los de salida de las unidades ocultas y de salida.

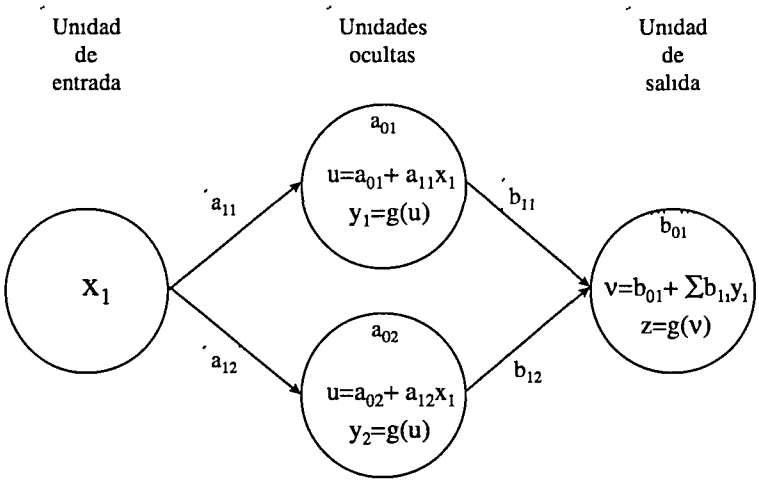


Fig. 9. Predicción con una red MLP 1-2-1

Con una sola variable independiente, cada unidad oculta calcula su valor de activación (u) como la suma de su peso umbral (a_{0i}) más el valor de la variable independiente (x_1) ponderado por el peso de la conexión (a_{1i}). Así, el valor de activación de cada unidad oculta se obtiene como una combinación lineal del input y de los pesos asociados a dicha unidad. La no linealidad se introduce al aplicar al valor de activación la función logística, cuyo resultado es el valor de salida de la unidad.

La modificación de los valores de los pesos umbral y de conexión a_{0i} y a_{1i} permiten ajustar la forma de la función logística y, por tanto, el valor de salida de la unidad, a cualquier necesidad, de acuerdo a los siguientes criterios (Smith, 1993):

- El signo del peso de la conexión (a_{1i}) determina que la función logística sea creciente (signo positivo) o decreciente (signo negativo).
- El valor absoluto del peso de la conexión (a_{1i}) determina que la función logística tenga mayor (valor alto) o menor (valor bajo) pendiente.
- Los valores de los pesos umbral (a_{0i}) y de conexión (a_{0i}) determinan en qué valor de x_i se produce la inflexión de la función logística. La fijación correcta del valor de inflexión de la función logística tiene una relevancia especial durante el proceso de entrenamiento, ya que este valor establece un punto que divide el rango de valores del input en dos zonas: una en que la unidad oculta da un valor de salida alto y otra en que lo da bajo.

La unidad de salida de la red de la Fig. 9 tiene un comportamiento idéntico al de las unidades ocultas, excepto en el hecho de que en el cálculo de su valor de activación recibe información de dos unidades. En este caso, la función logística

se representa en tres dimensiones y su zona de inflexión es una recta, que geométricamente se obtiene como la proyección en el plano x_1-x_2 de la recta que hace intersección con el plano que divide en dos mitades idénticas la función logística (ver Fig. 10).

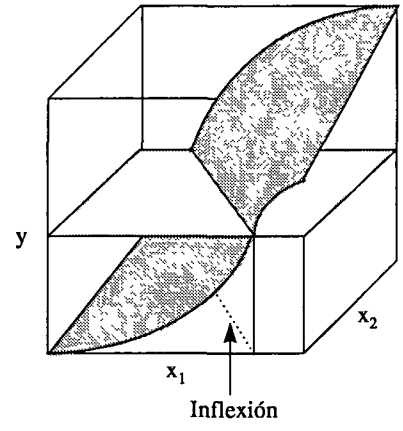


Fig. 10. Recta de inflexión de la función logística con dos variables.

Ejemplo

Para ejemplificar el funcionamiento de una red MLP en modo de predicción presentamos una red MLP 1-2-1 capaz de computar la función ilustrada en la Fig. 11.

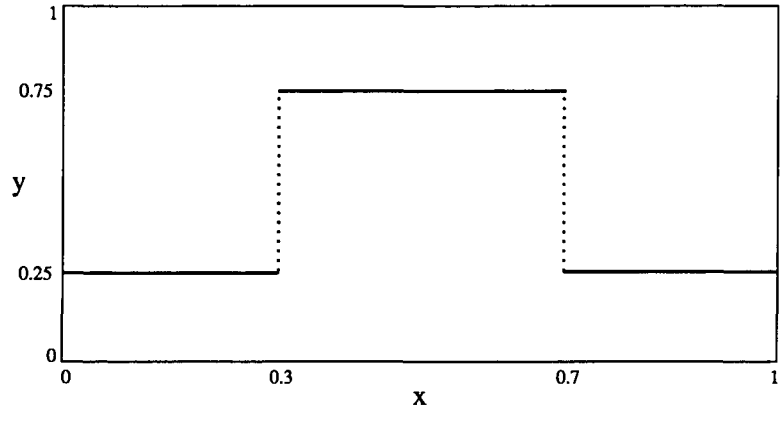


Fig. 11. Función a predecir. Las líneas punteadas representan discontinuidades de la función.

Una red con los pesos indicados en la Fig. 12 aproxima la función con un error medio cuadrático de sólo 0.002, que incluso podría ser reducido aumentando el valor de los pesos a_{1i}, ya que así se incrementaría la pendiente de la función logística en los puntos de discontinuidad de la función.

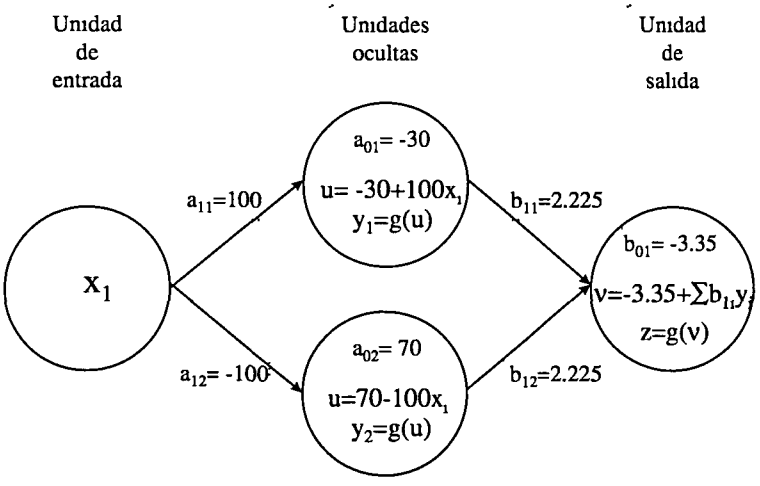


Fig. 12. Red MLP para predecir la función de la Fig. 11

El procesamiento de la información en redes MLP con más de una unidad de entrada y de salida sigue las mismas reglas presentadas para la red MLP de la Fig. 9.

4.1.3. Entrenamiento de una red MLP

El entrenamiento de una red MLP es, en la mayoría de casos, un ejercicio de optimización de una función no lineal. Los métodos de optimización no lineales han sido estudiados durante años, y como consecuencia de ello existe una amplia literatura al respecto (ver p.ej. Bertsekas, 1995; Gill, Murray y Wright, 1981; Masters, 1995). A pesar de ello, no se ha hallado un método que en general pueda ser considerado mejor a los demás (Sarle, 1998a).

El objetivo del entrenamiento de una red neuronal es hallar los pesos que proporcionan la mejor aproximación a la función que relaciona los inputs y los outputs del conjunto de datos de entrenamiento. La entrada y la salida son vectores o pares de entrenamiento con los datos de entrada y de salida. La medida de ajuste que se emplea suele ser el error medio cuadrático (*EMC* o simplemente *E*), calculado sobre todos los registros (*N*) y todos los outputs (*K*):

$$EMC = \frac{\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}(z_{kn} - y_{kn})^2}{NK}$$

Donde Z_{kn} es el valor predicho por la red en el output k y el registro n e y_{kn} es el verdadero valor en el mismo output y el mismo registro. El término $\frac{1}{2}$ se incluye para simplificar el cálculo con las derivadas que posteriormente se han de calcular.

Aunque en problemas de clasificación, en los que el output es categórico, parece más natural usar el porcentaje de clasificaciones incorrectas como medida de ajuste, se utiliza el EMC porque tiene interesantes propiedades como su suavidad y diferenciabilidad.

Los usuarios de redes neuronales acostumbran a hablar de la superficie de error al referirse a cuestiones implicadas con el error de predicción de sus modelos. La superficie de error es el resultado de representar gráficamente el error medio cuadrático en función de los pesos de la red (ver Fig. 13). Obviamente, el mejor conjunto de pesos es aquel que tiene asociado un valor menor en la superficie de error, o en otras palabras, al que corresponde el mínimo global de dicha superficie (Bishop, 1995).

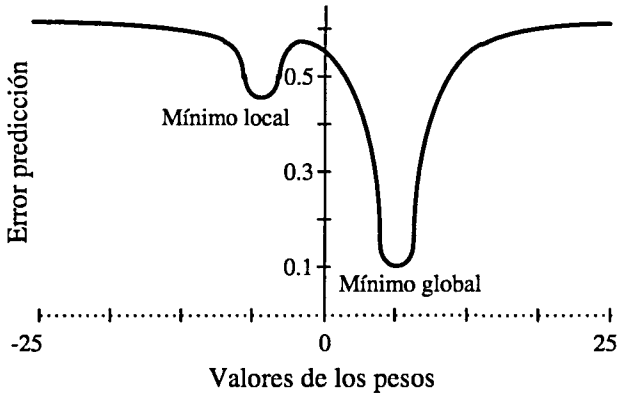


Fig. 13. Superficie de error

La presencia de más de un valle en la superficie de error representa un potencial problema, ya que existe el riesgo de que el entrenamiento conduzca a un mínimo local en lugar de al mínimo global (Hecht-Nielsen, 1989). A pesar de la importancia que se ha dado al problema de los mínimos locales de error, que ha dado lugar a mecanismos muy complejos para evitarlos, como por ejemplo el desarrollado por Wasserman (1988), se trata en realidad de un problema más teórico que aplicado (Gurney, 1997; Smith, 1993). Efectivamente, cabe tener

presente, en primer lugar, que la probabilidad de que un mínimo local exista decrece según aumenta el número de unidades ocultas y de salida de la red, y las redes aplicadas pueden llegar a tener cientos de unidades. Además, puede ser que el error de predicción en un mínimo local sea lo suficientemente pequeño como para satisfacer las expectativas del investigador. Finalmente, en el peor de los casos los mínimos locales pueden ser evitados entrenando varias veces la red partiendo cada vez de un conjunto de pesos iniciales diferente, lo que se traduce en un punto distinto de la superficie de error.

Un método intuitivo para hallar el mínimo absoluto de la superficie de error consiste en calcular el error para cada combinación de valores de pesos dentro de un rango determinado. A pesar de ser lógica, esta posibilidad es computacionalmente inviable en la práctica incluso con la velocidad de procesamiento de los ordenadores actuales. Basta con saber que, en una pequeña red MLP 5-3-1 en la que los pesos pudieran tomar los valores enteros en el intervalo [-2,+2], se tendrían que realizar aproximadamente 10^{21} mediciones del error.

Las técnicas basadas en el descenso del gradiente de error, como la propagación del error hacia atrás y en general todos los algoritmos de entrenamiento de redes MLP, proporcionan una solución a este problema. Este conjunto de métodos de aprendizaje buscan un valor mínimo en la superficie de error mediante la aplicación de sucesivos pasos descendentes.

A grandes rasgos, los procedimientos que utilizan el descenso del gradiente del error tienen un carácter iterativo basado en los siguientes pasos, que se describen detalladamente a lo largo de este capítulo:

1. Inicializar la matriz de pesos de la red.
2. Calcular la derivada del error con la matriz de pesos actuales.
3. Presentar los datos a la red y cambiar los pesos según una determinada regla de aprendizaje.
4. Iterar los pasos 2 a 3 hasta que se produce la convergencia, que suele estar definida como el error mínimo de predicción asumible por el investigador, o como el cambio mínimo del valor de los pesos entre dos iteraciones (denominadas también "épocas" en terminología de redes) sucesivas.

4.1.3.1. Inicialización de la matriz de pesos

Cuando una red neuronal es diseñada por primera vez, se deben asignar valores a los pesos a partir de los cuales comenzar la fase de entrenamiento. Los pesos de umbral y de conexión se pueden inicializar de forma totalmente aleatoria, si bien es conveniente seguir algunas sencillas reglas que permitirán minimizar la duración del entrenamiento.

Es conveniente que la entrada neta a cada unidad sea 0, independientemente del valor que tomen los datos de entrada. En esta situación, el valor devuelto por la función logística es un valor intermedio, que proporciona el menor error medio cuadrático si los valores a predecir se distribuyen simétricamente alrededor de este valor intermedio (como habitualmente sucede). Además, al evitar los valores de salida extremos se escapa de las zonas saturadas de la función logística en que la pendiente es prácticamente nula y, por tanto, el aprendizaje casi inexistente (Fausett, 1994).

En la práctica, el objetivo anterior se puede alcanzar de diferentes maneras. La más sencilla y utilizada es asignar a las unidades ocultas pesos iniciales pequeños generados de forma aleatoria, en un rango de valores entre -0.5 y 0.5 o entre -0.1 y 0.1 (Martín, 1993; SPSS Inc., 1997a). Otra posibilidad consiste en inicializar los pesos de las unidades ocultas con pequeños valores distribuidos de forma aleatoria alrededor del valor 0, e inicializar la mitad de pesos de las unidades de salida al valor de 1 y la otra mitad a -1 (Smith, 1993).

Una opción algo más sofisticada, propuesta por Nguyen y Widrow (1990), introduce en la inicialización de los pesos que unen las unidades de entrada con las ocultas información sobre el rango de valores de entrada con el que cada unidad oculta aprenderá más rápidamente.

4.1.3.2. Derivada del error respecto a los pesos

El aspecto central de la retropropagación del error, y en general de las técnicas de descenso del gradiente del error, es la sensibilidad del error de predicción a cambios en los pesos, que en términos geométricos corresponde a la necesidad de conocer la pendiente de la superficie de error para cualquier matriz de pesos, y en términos matemáticos implica la necesidad de conocer las derivadas parciales del error con respecto a los pesos, que pueden ser halladas aplicando la regla de la cadena (Hecht-Nielsen, 1990; Widrow y Lehr, 1990).

Si seccionáramos la superficie de error haciendo un corte paralelo a uno de sus ejes, que representa un peso de la red, muy probablemente obtendríamos un gráfico similar al de la Fig. 14.

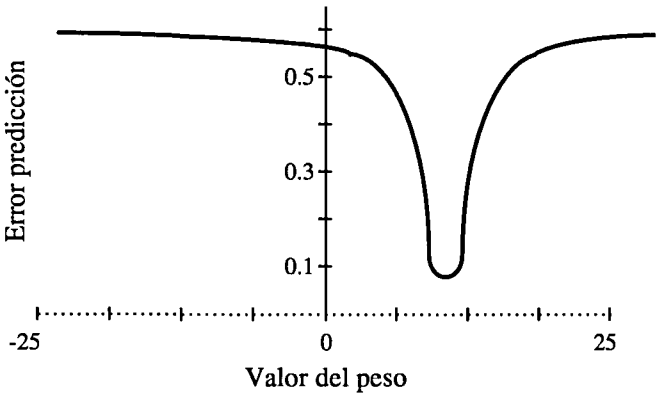


Fig. 14. Sección de la superficie de error

El objetivo del entrenamiento es hallar el valor del peso que hace mínimo el error de predicción; es decir el peso que corresponde al fondo del cuenco del mínimo global. En este punto, la recta tangente a la curva tiene una pendiente igual a cero, en puntos a su izquierda tiene una pendiente negativa, y en puntos a su derecha la pendiente de la recta tangente es positiva (ver Fig. 15). En general, si la pendiente es negativa será necesario incrementar el peso, mientras que si es positiva el valor del peso se deberá reducir. La magnitud del cambio del peso en valor absoluto depende de la magnitud de la pendiente: a mayor pendiente mayor cambio.

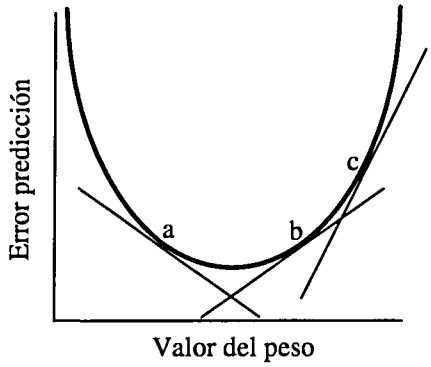


Fig. 15. Pendiente de las rectas tangentes según el valor del peso en tres puntos de la superficie de error

La pendiente de la recta tangente a la curva de error para un valor concreto de un peso es la derivada parcial del error medio cuadrático con respecto a dicho peso. La derivada parcial del error respecto a un peso se obtiene multiplicando un conjunto de términos que se corresponden a los pasos por los que el peso

contribuye al error (regla de la cadena). Por ejemplo, en el caso de una conexión de unidad oculta con una de entrada, un cambio del peso afecta en primer lugar a la entrada neta a la unidad oculta, la cual a su vez influye en la salida de dicha unidad. El valor de salida de la unidad oculta es enviado a todas las unidades de salida, modificando sus entradas netas, que a su vez modifican sus salidas. Así, un cambio en el peso de una unidad oculta causa cambios en toda la cadena y, por tanto, en el error cometido (Fausett, 1994; Martín, 1993).

a. Derivada del error en una unidad de salida

La regla de la cadena para una unidad de salida tiene tres términos, que representan los pasos por los que un cambio en su peso (b) influye en el error:

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial z}\frac{\partial z}{\partial v}\frac{\partial v}{\partial b}$$

- El primer término $\partial E/\partial z$ es la derivada del error respecto al valor de salida de la unidad, o con otras palabras, el efecto que tiene sobre el error el valor de salida de la unidad. Dicho efecto es la diferencia entre el valor de salida de la unidad (z) y el verdadero valor (t).
- El segundo término $\partial z/\partial v$ es la derivada de la salida de la unidad respecto a su entrada neta, o lo que es lo mismo, el efecto que tiene sobre la salida de la unidad (z) el valor neto de entrada (v). Puesto que z es la función logística de v, la sensibilidad de z respecto a un cambio en v es la pendiente de la función logística, que puede ser obtenida multiplicando el valor predicho por su complementario $z(1-z)$.
- El tercer término $\partial v/\partial b$ es la derivada de la entrada neta a la unidad respecto al valor del peso que se está analizando, es decir, el efecto que tiene sobre la entrada neta un cambio en un peso. En este término se debe distinguir entre peso umbral y peso de conexión, puesto que cada uno interviene de una forma en el cálculo del valor de entrada.
- El valor del peso umbral (b_o) es un término aditivo independiente en la fórmula de cálculo de la entrada neta, de donde se deduce que $\partial v/\partial b_0 = 1$, ya que la entrada neta cambia en la misma magnitud que el peso umbral.
- El valor del peso de una conexión con una unidad oculta (b_j) es un término que multiplica la salida de la unidad oculta (y_j) en el cálculo de la entrada neta a la unidad de salida (v), de donde se deduce que $\partial v/\partial b_j = y_j$.

Aunque el tercer término depende del tipo de peso, los dos primeros son idénticos para todos los pesos de una unidad de salida. Al producto de ambos le denominaremos p para simplificar la posterior notación:

$$p = \frac{\partial E}{\partial z}\frac{\partial z}{\partial v} = (z - t)\, z\, (1 - z)$$

Si se consideran conjuntamente los tres términos analizados, se puede escribir completa la ecuación que calcula la derivada parcial del error respecto a un peso de una unidad de salida:

$$\frac{\partial E}{\partial b_j} = \begin{cases} p & \text{para un peso de tipo umbral} \\ py_j & \text{para un peso de tipo conexión} \end{cases}$$

b. Derivada del error en una unidad oculta

La regla de la cadena para una unidad de oculta tiene también tres términos, que representan los pasos por los que un cambio en su peso (*a*) influye en el error:

$$\frac{\partial E}{\partial a} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial a}$$

- El primer término $\partial E/\partial y$ es la derivada del error global respecto al valor de salida de la unidad oculta. Como el término error se refiere al error global de la red, la influencia de la salida de una unidad oculta sobre dicho error global se produce a través de las unidades de salida, por lo que este primer término incluye a su vez tres elementos dentro de un sumatorio para las *k* unidades de salida: los dos primeros son conocidos del apartado anterior, mientras que el tercero es el efecto de un cambio en la salida de la unidad oculta sobre la entrada neta a la unidad de salida:

$$\frac{\partial E}{\partial y} = \sum_{k=1}^{K} \frac{\partial E}{\partial z_k}\frac{\partial z_k}{\partial v_k}\frac{\partial v_k}{\partial y}$$

  Como a los dos primeros elementos les llamamos anteriormente *p*, y puesto que el tercero es el valor del peso b_k, podemos reescribir el primer término:

$$\frac{\partial E}{\partial y} = \sum_{k=1}^{K} p_k b_k$$

- El segundo término $\partial y/\partial u$ es la derivada de la salida de la unidad oculta respecto a su entrada neta, que de forma equivalente a una unidad de salida es $y(1-y)$.
- El tercer término $\partial u/\partial a$ es la derivada de la entrada neta a la unidad respecto al valor del peso que se está analizando. Como en el caso de una unidad de salida, en este término se debe distinguir entre peso umbral y peso de

conexión. Así, para un peso de tipo umbral (a_o) $\partial u/\partial a_0 = 1$, y para un peso de tipo conexión con una unidad de entrada (a_i) $\partial u/\partial a_i = x_i$.

Si denominamos q al producto de los dos primeros términos:

$$q = \frac{\partial E}{\partial y}\frac{\partial y}{\partial u} = \left(\sum_{k=1}^{K} p_k b_k\right) y\,(1-y)$$

Podemos reescribir completamente la ecuación que calcula la derivada parcial del error respecto a un peso de una unidad oculta:

$$\frac{\partial E}{\partial a_i} = \begin{cases} q & \text{para un peso de tipo umbral} \\ qx_i & \text{para un peso de tipo conexión} \end{cases}$$

4.1.3.3. Regla de aprendizaje

Para completar una época (iteración) de la fase de entrenamiento se ha de modificar el valor de los pesos. La regla de aprendizaje determina la cantidad de cambio en cada peso de la red a partir de las derivadas parciales del error calculadas en el apartado anterior.

Desde que en 1986 se presentara la regla de retropropagación del error (o regla delta generalizada) se han desarrollado diferentes algoritmos de aprendizaje. A continuación se presentan los más relevantes.

a. Descenso más abrupto

El método de descenso más abrupto del error (*steepest descent*) es la regla de aprendizaje inicialmente propuesta por Rumelhart, Hinton y Williams (1986) que, como su nombre indica, consiste en el cambio de los pesos en la dirección en que el error decrece más rápidamente.

El proceso de aprendizaje comienza con el cálculo de la derivada del error para cada registro de datos en el primer peso de la red (w_1). Casi siempre, unos registros sugerirán un cambio positivo (incrementar el valor del peso) y otros un cambio negativo (decrementarlo). El promedio de la derivada del error en todos los registros informa de la dirección y de la magnitud en que el peso w_1 se ha de modificar para minimizar el error atribuible a él. Es útil representar las aportaciones de cada registro y su promedio con vectores de diferente dirección y longitud (ver Fig. 16).

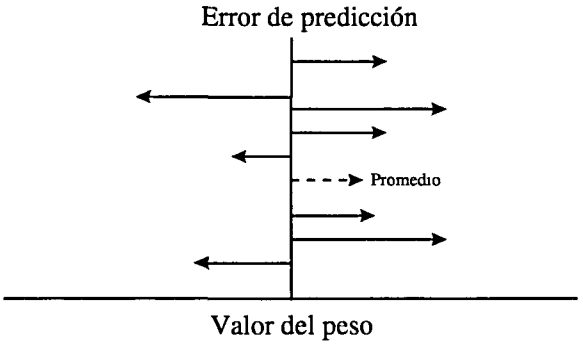


Fig. 16. Derivadas del error respecto a un peso en varios registros y en promedio

Pero puesto que en la red hay más de un peso, al repetir el cálculo para un segundo peso w_2 se dispone de dos promedios, representados en la Fig. 17 por dos líneas discontinuas. Si sólo hubieran dos pesos en la red, la dirección y la magnitud en que el error global de predicción aumenta más abruptamente viene determinado por la bisectriz del rectángulo formado por los dos promedios. La flecha en la dirección opuesta representa la dirección y la magnitud en que el error disminuye más rápidamente. Aunque no sea posible su representación gráfica, en redes con más de dos pesos se aplica el mismo principio.

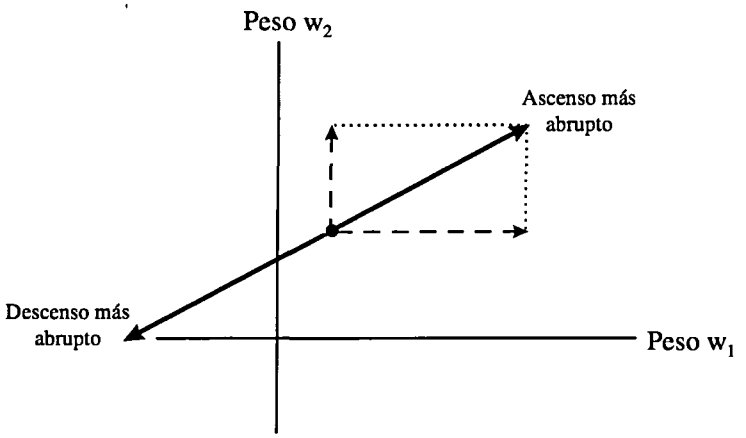


Fig. 17. Derivada del error respecto a dos pesos

La regla de descenso más abrupto incluye un coeficiente de aprendizaje multiplicativo de la magnitud de descenso obtenida con las derivadas del error. Así, el valor de un peso *w* en la época *m* viene determinado por:

$$w_m = w_{m-1} + c_m$$

donde c_m es el cambio a realizar en el peso *w* en la época *m*, e incluye el mencionado coeficiente de aprendizaje ε:

$$c_m = -\varepsilon d_m$$

Con d_m el sumatorio para todos los registros de la derivada del error para el peso *w* en la época *m*:

$$d_m = \sum_{n=1}^{N} \left(\frac{\partial E}{\partial w_m} \right)_n$$

Si $\varepsilon=1$, el coeficiente no tiene efecto; si $\varepsilon=0.5$ el cambio real del peso es la mitad del obtenido mediante la derivada del error; si $\varepsilon=2$ el cambio real del peso es el doble. El coeficiente de aprendizaje ε (*learning rate*) es fijado por el investigador en cada conjunto de datos. Ante el desconocimiento de la forma completa de la superficie de error, la única manera de descubrir el mejor valor de ε es por ensayo y error, lo cual representa uno de los inconvenientes de esta regla de aprendizaje. Es de vital importancia asignar un valor adecuado al coeficiente de aprendizaje, ya que un valor demasiado pequeño provocaría cambios muy pequeños en cada época, y la convergencia del procedimiento se lentificaría sensiblemente, mientras que un valor demasiado grande provocaría cambios muy grandes que podrían imposibilitar encontrar el mínimo de la superficie de error, ya que la red cruzaría de un extremo a otro de la superficie sobrepasando su mínimo global (Werbos, 1994).

b. Aprendizaje ejemplo a ejemplo

Una variante del método de descenso más abrupto consiste en cambiar los pesos después de analizar cada registro, y no a partir del promedio de todos los registros, o con otras palabras, en considerar que una iteración del entrenamiento se basa en los datos proporcionados por un único registro.

Con este método, el error de predicción desciende rápidamente al inicio del entrenamiento, ya que no es necesario esperar a analizar todos los casos para actualizar los pesos. Sin embargo, a pesar de la velocidad inicial en el descenso del error, no parece que la duración total del entrenamiento se vea sensiblemente reducida, ya que el aprendizaje ejemplo a ejemplo es computacionalmente más costoso porque el cambio en los pesos se ha de calcular para cada registro, en lugar de una sola vez con todos los registros.

El principal inconveniente del aprendizaje ejemplo a ejemplo radica en el hecho de que los casos presentados al final del entrenamiento influyen más que los presentados al inicio. Si los últimos casos provocan cambios importantes y en el mismo sentido, su efecto no podrá ser contrarrestado por sujetos que provoquen cambios en sentido contrario. Por ello, con este método se ha de poner especial consideración en el orden de los registros de entrenamiento, siendo la mejor opción aleatorizar el orden de presentación en cada época del entrenamiento.

c. Descenso más abrupto con momento

Con el aprendizaje ejemplo a ejemplo la red invierte mucho tiempo alternando cambios en un sentido y en el contrario: lo que aprende con un registro lo desaprende con el siguiente. El método de descenso más abrupto soluciona este problema basando el cambio de los pesos en el promedio de todos los registros, pero es un procedimiento muy lento. Una alternativa que acelera la velocidad de convergencia del método de descenso más abrupto consiste en introducir, en la ecuación que determina la magnitud del cambio de un peso, un nuevo término denominado "momento" (Bishop, 1995; Gurney, 1997; Smith, 1993).

El aprendizaje con momento se basa en promediar el cambio real en los pesos más que en promediar las derivadas del error. En la práctica, se calcula el cambio que correspondería al peso *w* con el método de descenso más abrupto, pero el cambio que realmente se realiza es un promedio entre este cambio y el último cambio efectuado. Puesto que el último cambio era también un promedio, se trata de un proceso recursivo.

En términos de la superficie de error, el uso del momento se traduce en la incorporación de una especie de inercia, que se va incrementando según sucesivas épocas sugieren cambios en la misma dirección. Ello permite evitar las típicas oscilaciones sobre la superficie de error que se producen con el método de descenso más abrupto.

Formalmente, el aprendizaje por descenso más abrupto con momento consiste en cambiar el peso *w* en:

$$c_m = \mu c_{m-1} - (1-\mu)\varepsilon d_m$$

donde μ es el coeficiente de momento, cuyo valor debe estar comprendido en el intervalo (0 a 1]. Si μ vale 0 el cambio actual del peso *w* no está influido por el cambio anterior. Según aumenta μ, el cambio anterior tiene mayor impacto sobre el cambio actual. Obviamente, μ no puede valer 1, ya que entonces la derivada del error en la época actual no intervendría en el cálculo del cambio. El valor del coeficiente de momento no parece afectar demasiado el resultado final del entrenamiento, sino únicamente su velocidad, por lo que un valor $\mu=0.9$ es comúnmente usado.

d. Coeficiente de aprendizaje adaptativo

En las reglas de aprendizaje basadas en el descenso más abrupto del gradiente del error (con o sin momento) todos los pesos de la red tienen el mismo valor de coeficiente de aprendizaje ε. Puesto que el cambio producido en una época se obtiene promediando los cambios individuales de todos los pesos, es evidente que en la época m algunos pesos se verán más favorecidos que otros, o dicho de otra manera, contribuirán más a reducir el error global de la red. Sería de gran utilidad, especialmente para acelerar la fase de entrenamiento, que cada peso tuviera un coeficiente de aprendizaje propio, y que éste se pudiera ir modificando durante el entrenamiento.

En este sentido, se han desarrollado diferentes algoritmos que corrigen el coeficiente de aprendizaje durante el entrenamiento. Algunos tienen usos muy específicos; así, por ejemplo, en un problema de clasificación en el que determinadas categorías están poco presentes, se puede incrementar el coeficiente de aprendizaje cuando se presentan a la red los registros pertenecientes a las categorías subrepresentadas (DeRouin, Brown, Beck, Fausett y Schneider, 1991).

El algoritmo de aprendizaje conocido como "máximo salto seguro" (Weir, 1991) se basa en determinar el mayor cambio a realizar en cada peso sin que ello contribuya a cambiar de signo la pendiente de la recta tangente a la superficie de error. Para ello es necesario elaborar complejos cálculos adicionales a los realizados en la retropropagación estándar, por lo que es un método poco implementado en el *software* de redes neuronales artificiales.

Posiblemente, los métodos de adaptación del coeficiente de aprendizaje más difundidos son los que se basan en algoritmos conocidos con el nombre genérico de "asunción de riesgo". Algoritmos de este tipo han sido desarrollados por diferentes investigadores, entre ellos Cater (1987), Fahlman (1988), Riedmiller y Braun (1993) y Silva y Almeida (1990), aunque el más conocido es la regla *delta-bar-delta*, presentado por Jacobs (1988) basándose en los trabajos de Barto y Sutton (1981) y, sobre todo, de Sutton (1986).

El concepto subyacente a la regla delta-bar-delta es bastante intuitivo: si el cambio en un peso en la época actual es en la misma dirección (aumento o decremento) que en las épocas anteriores, el coeficiente de aprendizaje para este peso debe incrementarse. Por contra, si la dirección del cambio del peso es alterna el coeficiente de aprendizaje debe disminuirse.

Para saber si el cambio en un peso tiene la misma dirección durante varias épocas, se compara el signo de la derivada del error en la época actual (d_m) con el signo del promedio de la época actual más las épocas anteriores (f_m). Para definir qué se entiende por épocas anteriores se introduce un nuevo parámetro θ. La

dirección en que el error ha decrecido recientemente (promedio de época actual y anteriores) se obtiene como:

$$f_m = \theta f_{m-1} + (1-\theta) d_m$$

El parámetro θ puede tomar cualquier valor en el intervalo (0 a 1]. Si θ vale 0 no se tienen en cuenta las épocas anteriores, según θ se aproxima a 1 el valor del cambio actual d_m influye menos y los cambios anteriores más. Puesto que en el cálculo del promedio f_m interviene f_{m-1}, que es a su vez un promedio, se trata de un proceso recursivo que incluye todas las épocas del entrenamiento.

El coeficiente de aprendizaje ε_m para el peso w en la época m se ha de incrementar o disminuir según el signo del producto $d_m f_m$. En concreto, si el signo de $d_m f_m$ es positivo, es decir, se sigue en la misma dirección, el coeficiente de aprendizaje es incrementado por la constante κ. Si el signo de $d_m f_m$ es negativo, o sea la dirección ha cambiado, el coeficiente de aprendizaje se multiplica por la constante ϕ, cuyo valor debe estar en el intervalo [0 a 1)

$$\varepsilon_m = \begin{cases} \varepsilon_{m-1} + \kappa & \text{si} \quad d_m f_m > 0 \\ \varepsilon_{m-1} \times \phi & \text{si} \quad d_m f_m \leq 0 \end{cases}$$

donde κ y ϕ son constantes fijadas por el investigador.

Aunque la regla delta-bar-delta permite no tener que decidir qué coeficiente de aprendizaje se utilizará, sí obliga a fijar tres parámetros (θ, κ, ϕ), aunque en la práctica el rendimiento final del sistema no se ve muy afectado por los valores que se asignen. Valores que en general dan buenos resultados son θ=0.7, κ=0.1, ϕ=0.5 (Smith, 1993).

Jacobs (1988) presenta un experimento en el que compara las tres reglas de aprendizaje presentadas hasta el momento (retropropagación simple, retropropagación con momento y regla delta-bar-delta). Se realizaron veinticinco simulaciones del problema "O exclusivo" (XOR) con diferentes pesos iniciales, usando una red MLP con dos unidades de entrada, dos unidades ocultas y una unidad de salida. Las entradas eran binarias (0,1) y las salidas se codificaron con los valores 0.1 y 0.9. Se consideró como criterio de éxito de convergencia que el error de predicción fuera inferior a 0.4 durante cincuenta épocas. Los valores de los diferentes parámetros se hallan en la Tabla 13.

Tabla 13. Valores de los parámetros de entrenamiento del experimento de Jacobs (1988). (*): Valor inicial

	ε	μ	θ	κ	ϕ
Retropropagación	0.1				
Retropropagación con momento	0.75	0.9			
Delta-bar-delta	0.8*		0.035	0.333	0.7

Los resultados (ver Tabla 14) reflejan que, aunque en tres simulaciones no converge, la regla delta-bar-delta es mucho más rápida que las otras dos reglas de aprendizaje evaluadas.

Tabla 14. Resultados del experimento de Jacobs (1988)

	Nº de éxitos en convergencia	Promedio de épocas
Retropropagación	24	16859.8
Retropropagación con momento	25	2056.3
Delta-bar-delta	22	447.3

e. Quickprop

El algoritmo *Quickprop* es una modificación del método de descenso más abrupto que acelera la velocidad del entrenamiento. *Quickprop* fue desarrollado por Fahlman (1989) en una época en que la velocidad de los procesadores hacía necesario aumentar la eficiencia del entrenamiento.

La idea subyacente es sencilla: si, por ejemplo, un cambio de -2 en un peso disminuye la derivada del error de 4 a 3, para reducir de 3 a 0 en un solo paso el peso debe reducirse en -6. En forma de ecuación, el cambio a realizar en el peso w en el momento m es:

$$c_m = \frac{d_m}{d_{m-1} - d_m} c_{m-1}$$

f. Reglas de aprendizaje de segundo orden

Las reglas de aprendizaje de segundo orden se basan en el cálculo de la segunda derivada del error con respecto a cada peso, y en obtener el cambio a realizar a partir de este valor y el de la derivada primera. Presentan la ventaja de acelerar sustancialmente la velocidad del entrenamiento, aunque su uso exige experiencia y un análisis de la forma de la superficie de error respecto a cada peso que se modifique.

Los primeros trabajos sobre el uso de métodos de segundo orden en retropropagación fueron realizados por Watrous (1987), Parker (1987) y Becker y LeCun (1988), aunque el algoritmo más utilizado y divulgado es el del "gradiente conjugado" (Battiti, 1992).

Mientras que la primera derivada del error informa de la pendiente de la superficie de error, la segunda derivada informa de la curvatura de dicha superficie, o lo que es lo mismo, de la razón de cambio de la pendiente. Haciendo un símil físico, se puede afirmar que la primera derivada representa la velocidad y la segunda la aceleración del error.

La cantidad en que se debe modificar el peso *w* para obtener una derivada igual a cero puede ser estimada dividiendo la primera derivada por la segunda derivada:

$$c = \frac{\partial E / \partial w}{\partial^2 E / \partial w^2}$$

El cálculo exacto de la segunda derivada del error es por lo general inviable, aunque sí existen aproximaciones como la propuesta por Prina, Ricotti, Ragazzini y Martinelli (1988), que para la función logística como función de activación y para un peso *b* que conecte una unidad oculta con una de salida, da como resultado:

$$\frac{\partial^2 E}{\partial w^2} \approx z(1-z)\left[(1-2z)(z-t)+z(1-z)\right]y^2$$

donde recordemos que *y* es la salida de la unidad oculta, *z* es la salida de la unidad de salida y *t* es el verdadero valor de la salida. Para un peso de tipo umbral de una unidad de salida se elimina el término y^2 final.

Para un peso *a* que conecte una unidad de entrada con una oculta, la segunda derivada del error es aproximadamente:

$$\frac{\partial^2 E}{\partial a^2} \approx x^2 y(1-3y+2y^2)\sum_{k=1}^{K} z_k(1-z_k)(z_k-t_k)b_k$$

$$+\left[y^2(1-y^2)\sum_{k=1}^{K} b_k\left(z_k(1-3z_k+2z_k^2)(z_k-t_k)+z_k^2(1-z_k)^2\right)\right]$$

Para un peso de tipo umbral de una unidad oculta se elimina el término x^2 inicial.

4.1.4. Generalización y sobreentrenamiento en redes MLP

En la fase de entrenamiento de una red neuronal MLP se reduce el error de predicción en el conjunto de datos utilizados en el entrenamiento. Sin embargo, esta medida del error no es la de mayor interés, ya que lo que se desea en último término es utilizar la red neuronal para predecir nuevos casos, y no los de entrenamiento. La capacidad de una red entrenada para producir la respuesta correcta en registros de datos similares (provenientes de la misma población), pero no idénticos a los de entrenamiento, es conocida como "generalización" (Bishop, 1995; Gurney, 1997; Nelson, 1991; Ripley, 1996).

El responsable de que al minimizar el error en los datos de entrenamiento no se minimice necesariamente también el error medido en registros no usados en el entrenamiento es el "ruido" de los datos. Por ruido se entiende tanto la información ausente como la inadecuada que puede causar variaciones en la relación real entre entradas y salidas de una red en los datos de entrenamiento (Geman, Bienenstock y Doursat, 1992).

Sarle (1998b) menciona dos condiciones necesarias, aunque no suficientes, para conseguir una buena generalización. En primer lugar, la función que la red aprende debe ser suavizada, es decir, un cambio muy pequeño en los valores de entrada debe provocar un cambio muy pequeño en los valores de salida. Por otra parte, la muestra de datos de entrenamiento ha de ser lo suficientemente grande y representativa de la población. Efectivamente, si el número de grados de libertad de una red, definido a partir del número de casos de entrenamiento, es mayor que el número de parámetros, la activación de las unidades ocultas tiende a formar un conjunto ortogonal de variables, como combinaciones lineales o no lineales de las variables de entrada, que abarcan el mayor subespacio posible del problema. Cuando los datos de entrenamiento tienen ruido, las unidades ocultas actúan como detectores de las características abstractas de los datos, modelando sólo su estructura subyacente, y permitiendo así la generalización a patrones de datos nuevos.

El principal problema relacionado con la capacidad de generalización de una red neuronal artificial es el del sobreentrenamiento, que se produce cuando la red aprende, no sólo la función subyacente que relaciona entradas y salidas en los datos de entrenamiento, sino también el ruido. En este sentido, se puede afirmar que la capacidad de las redes neuronales artificiales para modelar cualquier tipo de función es al mismo tiempo su cualidad y su defecto.

Una red sobreentrenada puede incluso llegar a predecir valores imposibles, como demuestra el siguiente ejemplo: se desea predecir el número de votos afirmativos en un referéndum a partir del número de personas que votan, y se utilizan como datos de entrenamiento los de la Fig. 18. Una red sobreentrenada podría dar

como resultado un número de votos afirmativos mayor que el número de votantes.

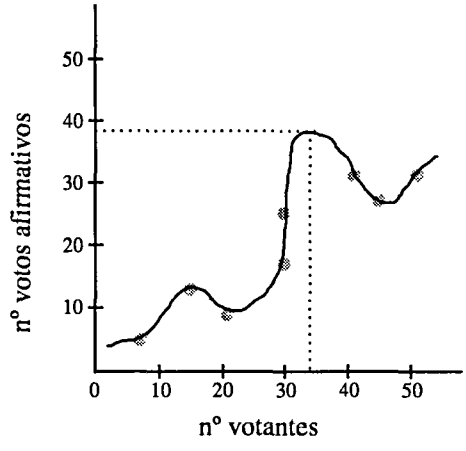


Fig. 18. Resultado de una red sobreentrenada. Para 33 votantes el número de votos afirmativos predicho es de 38.

Por contra, con los mismos datos, una red no sobreentrenada daría como resultado la función de la Fig. 19.

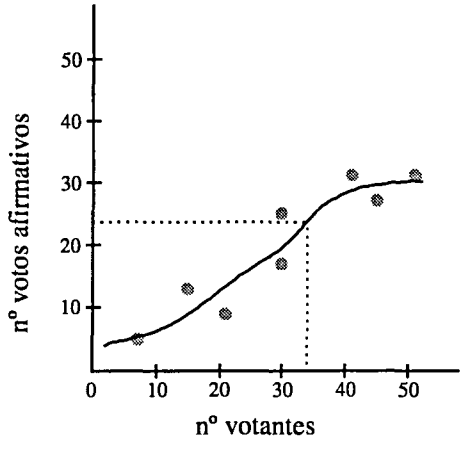


Fig. 19. Resultado de una red No sobreentrenada. Para 33 votantes el número de votos afirmativos predicho es de 24.

Uno de los ámbitos de investigación actualmente más prolífico es la medición de la capacidad de generalización (o, mejor dicho, de no generalización, ya que lo que se mide es el error y no el acierto) de una red neuronal artificial. El método más usado consiste en reservar una parte de los datos disponibles como datos de validación, los cuales no deben ser tenidos en cuenta durante el entrenamiento en modo alguno. Al finalizar el entrenamiento se predice mediante la red la salida en los datos de validación, y el error de predicción obtenido proporciona una estimación insesgada del verdadero error de generalización (Hecht-Nielsen, 1990; Olmeda, 1993).

En ocasiones, es necesario el uso de una muestra de validación para llevar a cabo alguno de los métodos para evitar el sobreentrenamiento que se presentan posteriormente, por lo que esta muestra no puede ser utilizada para medir la generalización ya que, aunque indirectamente, ha intervenido en el entrenamiento. En estos casos, una alternativa es reservar otra submuestra diferente de los datos de entrenamiento (muestra de test) con los que exclusivamente se mide el error de generalización (Sarle, 1998b; SPSS Inc., 1997a).

A pesar de su amplia divulgación, el uso de una muestra de validación es indeseable por dos motivos principales (Weiss y Kulikowski, 1991). En primer lugar, reduce el número de ejemplos disponibles para entrenar la red y, en consecuencia, la precisión del modelo. Además, y principalmente, es un método que no ofrece garantías, ya que el error de generalización obtenido depende de las características específicas de los registros de validación seleccionados. En esta línea, Weiss y Kulikowski (1991) exponen, mediante la selección aleatoria de diferentes muestras de validación dentro de un mismo conjunto de datos de entrenamiento, que el error de generalización depende en gran parte de la muestra seleccionada.

Las líneas recientes de investigación se centran en la estimación del error de generalización sin la necesidad de una muestra de validación. Moody (1992) propuso un método fundamentado en la relación entre el número de parámetros de la red, la variancia del error y el tamaño de la muestra de entrenamiento. Mackay (1992, 1992a, 1992b) propuso una aproximación Bayesiana a la cuestión. Vladimir Vapnik (1992) presentó el modelo "minimización estructural del riesgo", basado en la generación de un intervalo de confianza del error de generalización de los datos de entrenamiento.

Otras aproximaciones son la validación cruzada y los métodos *bootstrap*, detalladamente presentadas en Hjorth (1994) y Masters (1995). En su versión más elemental, la validación cruzada consiste en dividir los datos disponibles en m submuestras, frecuentemente con sólo un caso en cada una, por lo que m es igual al tamaño muestral (Lachenbruch y Mickey, 1968). Cada submuestra es

predicha a partir del modelo establecido con las *m-1* submuestras restantes, y el error de generalización es el error promedio en las *m* submuestras. El principal defecto de la validación cruzada es que produce estimaciones del error muy dispersas, con intervalos de confianza del verdadero error muy amplios. Los métodos *bootstrap* proporcionan intervalos más estrechos (Efron, 1983; Crawford, 1989), si bien en muestras de pequeño tamaño, que es precisamente donde más interesan para reducir la variabilidad, dan estimaciones del error sesgadas. La metodología *bootstrap* se basa en remuestrear con reemplazamiento a partir de la muestra original para obtener nuevas muestras, en cada una de las cuales se obtiene un modelo predictivo. Al remuestrear con reemplazamiento, algunos datos no aparecerán en la muestra *bootstrap* (en promedio 1/e=37% de los datos originales), pudiendo ser utilizados como datos de test en los que medir el error. Finalmente, una combinación de la medida de error en todas las muestras proporciona la estimación global.

A pesar de los numerosos avances, actualmente no existe consenso en cual de los métodos enumerados es más adecuado, y el *software* de redes sigue implementando de forma generalizada el uso de una muestra de validación y otra de test en la medición del error de generalización.

4.1.4.1. Métodos para evitar el sobreentrenamiento

La mejor manera de evitar el sobreentrenamiento es disponer de un gran número de registros como datos de entrenamiento. En este sentido, Sarle (1998b) afirma que si el número de casos de entrenamiento es como mínimo treinta veces el número de parámetros de la red[2], el sobreentrenamiento es muy improbable, mientras que Baum y Haussler (1989) presentan un sencillo método de cálculo del número de casos necesarios para conseguir una buena generalización en función del número de parámetros de la red.

Sin embargo, el número de registros de entrenamiento no suele ser fijado por el investigador, y no se puede reducir arbitrariamente el número de parámetros de la red para evitar el sobreentrenamiento, ya que ello podría conducir a un problema más importante, el subentrenamiento, o incapacidad de la red para modelizar la función deseada por falta de unidades ocultas.

Supuesto un número fijo de casos de entrenamiento, la solución al problema del sobreentrenamiento consiste en limitar cuidadosamente la complejidad de la red, definida a partir del número de parámetros que contiene y de sus valores. Barlett

---

[2] El número de parámetros de una red neuronal artificial MLP es el número de pesos de conexión y de umbral que contiene. Si se considera que el número de unidades de entrada y de salida está fijado por el diseño de la investigación, la única manera de modificar el número de parámetros es cambiando el número de unidades ocultas.

(1997) considera que para mejorar la generalización limitando la complejidad de la red, es más importante controlar el valor que puedan tomar los pesos que el propio número de pesos. En esta línea, en los últimos años se han propuesto diferentes procedimientos de reducción de la complejidad de una red neuronal, de entre los que cabe destacar los cinco que se presentan en los siguientes subapartados. Los dos primeros se basan en reducir el número de parámetros, y los tres restantes, englobados bajo el nombre de métodos de regularización (Barron, 1991), en limitar el valor que pueden tomar dichos parámetros.

a. Limitar el número de unidades ocultas

Algunos libros y artículos ofrecen reglas de decisión del número de unidades ocultas que ha de tener una red neuronal en función del número de inputs, de outputs y de registros de entrenamiento. Desafortunadamente, tales reglas no son útiles en la práctica, ya que la topología óptima depende de factores como la cantidad de "ruido" presente en los datos y la complejidad de la función que se está modelizando. Hay problemas con una variable independiente y una variable dependiente que necesitan cientos de unidades ocultas, y problemas con cientos de variables independientes y cientos de variables dependientes que necesitan una sola unidad oculta (Sarle, 1998b).

El problema de seleccionar una arquitectura óptima para resolver un problema debe ser visto desde la perspectiva más amplia de la "selección de modelos" (Rissanen, 1986). Desde esta óptica, la topología de red neuronal idónea es aquella que proporciona el menor error de generalización y, en caso de que dos o más topologías alternativas den el mismo resultado, de acuerdo con el principio de parsimonia, la seleccionada debe ser la menos parametrizada (Weigend, Huberman y Rumelhart, 1990, 1992).

En la práctica, para determinar el número óptimo de unidades ocultas se deben seguir un conjunto de pasos que se resumen en: (1) dividir la muestra en datos de entrenamiento y datos de validación, (2) entrenar diferentes topologías con diferente número de unidades ocultas hasta cumplir el criterio de convergencia, (3) medir el error de predicción de cada una de estas topologías en los datos de validación y (4) elegir la topología con menor error en la submuestra de validación (Moody, 1992).

Este procedimiento tiene tal coste computacional que en problemas reales puede incluso llegar a ser inviable, ya que requiere entrenar diferentes topologías y, además, cada una de ellas se debe entrenar hasta que se cumpla el criterio de convergencia.

b. Arquitecturas constructivas y destructivas

Ante la dificultad de establecer a priori la topología óptima de la red en un problema determinado, parece lógico diseñar procedimientos que permitan automatizar esta costosa decisión. Un enfoque consiste en partir de una topología sobreparametrizada y dotarla de un mecanismo que permita eliminar las unidades ocultas sobrantes; las redes de este tipo se denominan "arquitecturas destructivas" (Kung y Hwang, 1988; LeCun, Denker y Solla, 1990; Sietsma y Dow, 1991; Wynne-Jones, 1991, 1992). También se puede adoptar el enfoque contrario, es decir, partir de una topología mínima y de un procedimiento que incremente el número de unidades ocultas cuando se requiera, en este caso hablamos de redes de "arquitecturas constructivas" (Ash, 1989; Frean, 1990; Hirose, Yamashita y Hijiya, 1991).

Entre las arquitecturas de este tipo destaca el modelo de "correlación en cascada" (Fahlman y Lebiere, 1990), que añade no unidades sino capas (de una sola unidad oculta en su versión elemental) a la red original. Esta arquitectura es una de las favoritas en el entorno aplicado gracias a su superior capacidad de generalización.

c. Añadir ruido a los datos

Cuando el tamaño muestral no es lo suficientemente grande, algunos autores (Fausett, 1994; Sarle, 1998b; Stricker, 1998) sugieren introducir ruido deliberadamente en los datos de entrenamiento para crear nuevos registros y así mejorar la capacidad de generalización de la red. Por añadir ruido se entiende sumar a las entradas de algunos casos de entrenamiento, elegidos aleatoriamente, pequeñas cantidades obtenidas a partir de una distribución normal con media cero y variancia s, fijada por el investigador como una cantidad proporcional al rango de la variable de entrada. El valor de salida de un nuevo registro es el mismo que el de su registro de partida. Se asume que el sesgo cometido es despreciable siempre y cuando el ruido añadido sea lo suficientemente pequeño. Koistinen y Holmstrom (1992) discuten la cantidad y tipo de ruido que se puede agregar según las características de los datos de entrenamiento.

Esta técnica funciona porque al aumentar el número de casos de entrenamiento se suaviza la función que la red ha de aprender, y supuesto un número fijo de parámetros, ello implica necesariamente reducir la magnitud de los pesos.

d. Detención prematura del entrenamiento

El método de regularización más popular e implementado en el *software* de redes neuronales es la detención prematura del entrenamiento. Sarle (1995) resume este procedimiento en siete puntos:

1. Dividir la muestra en datos de entrenamiento y datos de validación.
2. Usar un número elevado de unidades ocultas.
3. Inicializar los pesos con valores muy pequeños.
4. Usar un coeficiente de aprendizaje pequeño.
5. Calcular periódicamente el error de generalización en la submuestra de validación.
6. Detener el aprendizaje cuando el error de generalización en los datos de validación comience a aumentar sistemáticamente.
7. Seleccionar el conjunto de pesos con menor error en los datos de validación.

La Fig. 20 ilustra la evolución típica del error de predicción a medida que el entrenamiento avanza, y el punto adecuado de detención del entrenamiento, cuando el error en los datos de validación comienza a aumentar.

En función del número de unidades ocultas que contenga la red, puede suceder que el error de validación no comience a aumentar durante el entrenamiento. Ello indica que la red no tiene suficientes unidades ocultas y, en consecuencia, se han de añadir nuevas unidades y volver a comenzar el proceso.

El método de detención prematura del entrenamiento funciona porque el aprendizaje de una red neuronal es progresivo, es decir, la complejidad de la función que se aprende aumenta progresivamente. Sin embargo, este incremento de complejidad no es estable en el tiempo, sino que ocurre de forma súbita en momentos muy concretos. En general, cada incremento corresponde a una importante alteración de los pesos de sólo una unidad oculta, y dichos incrementos están separados por largos períodos de tiempo de "ajuste fino" en los que no ocurre ningún cambio relevante. Así, durante la fase de entrenamiento la red pasa por sucesivas etapas en las que el número de unidades ocultas que realmente contribuyen a la predicción (unidades activas) aumenta una a una. Según la red es entrenada y la función que aprende aumenta en complejidad, pasa por un momento, con un determinado número de unidades ocultas activas y un conjunto de pesos asociados, que proporcionan la mejor generalización. A partir de este punto lo que la red aprende es el *ruido* de los datos de entrenamiento, que al no ser el mismo que el de los datos de validación, hace que el error de predicción de los datos de entrenamiento siempre disminuya, pero no el error medido en los datos de validación (Geman, Bienenstock y Doursat, 1992).

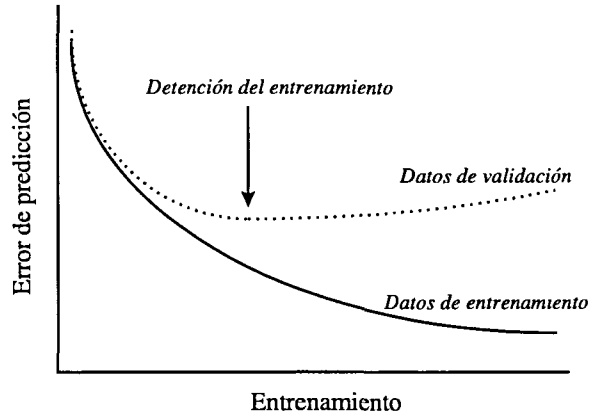


Fig. 20. Evolución típica del error en las submuestras de datos de entrenamiento y validación

La detención prematura del entrenamiento tiene diversas ventajas (Sarle, 1995):

- Es un método rápido y computacionalmente poco costoso.
- Puede ser aplicado con éxito a redes en las que el número de parámetros excede el tamaño muestral.
- Sólo requiere una decisión importante del investigador: que proporción de casos de validación usar.

Aunque también tiene algún inconveniente. El principal hace referencia al momento en que considera que el error en los datos de validación "comienza a aumentar sistemáticamente", ya que éste puede subir y bajar numerosas veces durante el entrenamiento. Prechelt (1994) sugiere que la opción más segura, aunque no más eficiente, es entrenar hasta que se cumpla el criterio de convergencia, y entonces mirar atrás y seleccionar el conjunto de pesos con el menor error de validación.

e. Penalización por pesos grandes

Con el método de "penalización por pesos grandes", el cambio a realizar en un determinado peso se basa en la derivada de una función de error (*O*) que incluye un nuevo término (*C*) que refleja la complejidad de la función modelizada (Bishop, 1995). La importancia de este término respecto al error viene determinado por una constante, fijada por el investigador, denominada constante de penalización (λ). Así, la función que se deriva para obtener la dirección y magnitud del cambio óptimo en un peso es:

$$O = E + \lambda C$$

siendo *E* el error medio cuadrático (*EMC*).

La complejidad de la función modelizada suele medirse a partir de la suma de pesos al cuadrado (el término $\frac{1}{2}$ se incluye para simplificar el cálculo con las derivadas que posteriormente se han de calcular):

$$C = \frac{1}{2}\sum w^2$$

Al minimizar esta medida combinada de error y complejidad, cada incremento en complejidad debe ser compensado con una reducción del error, por lo que no se aumentará significativamente un peso a no ser que ello reduzca también de forma significativa el error de predicción.

La cuestión más delicada al utilizar el método de penalización es decidir el valor de la constante de penalización λ, ya que con un valor muy bajo la penalización por complejidad sería demasiado pequeña y no se evitaría el sobreentrenamiento, mientras que un valor muy grande impediría que la función aprendida tuviera la mínima complejidad necesaria (subentrenamiento). Una posibilidad consiste en entrenar varias topologías con diferentes constantes de penalización y seleccionar aquella con un menor error de generalización (Smith, 1993). Una opción más sofisticada consiste en actualizar iterativamente el valor de la constante durante el entrenamiento (Weigend, Rumelhart y Huberman, 1991).

4.2. REDES DE FUNCIÓN BASE RADIAL (RBF)

Del conjunto de redes con aprendizaje supervisado y propagación de la información hacia adelante, las redes de función base radial (*Radial basis function* - RBF) son, tras las perceptrón multicapa, las más usadas en problemas de predicción y clasificación.

Las redes neuronales llamadas de función base radial son una adaptación de las ideas tradicionalmente conocidas bajo el nombre de estadísticos convencionales (Powell, 1987). Los trabajos iniciales en la formulación de este tipo de redes fueron realizados por Broomhead y Lowe (1988), Lee y Kil (1988), y Moody y Darken (1988, 1989).

4.2.1. Predicción con una red RBF

Las redes del tipo perceptrón multicapa (MLP) revisadas anteriormente, construyen aproximaciones globales a la función no lineal que relaciona inputs con outputs en un determinado problema de predicción. Por consiguiente, son capaces de generalizar a rangos de valores donde pocos o ningún dato son disponibles durante el entrenamiento. En cambio, las redes RBF construyen aproximaciones locales a la función no lineal objetivo. Como apunta Orr (1996), una red RBF con una capa de unidades ocultas, construye una aproximación a la función de regresión verdadera mediante la combinación lineal de un conjunto de

funciones no lineales. Como consecuencia de ello, su capacidad de generalización se puede ver reducida, si bien su entrenamiento es mucho más rápido, ya que se realiza en una etapa en lugar de utilizar un proceso iterativo.

La principal desventaja de las redes RBF es que el número de unidades ocultas necesario se incrementa geométricamente con el número de variables independientes, por lo que su uso se torna inviable en problemas con muchos inputs.

La principal diferencia entre las redes RBF y las redes MLP radica en el funcionamiento de las unidades ocultas. Como en las redes MLP, una unidad oculta de una red RBF tiene un parámetro por cada unidad de entrada; sin embargo, estos parámetros no son pesos en el sentido tradicional, sino que representan las coordenadas -en el espacio de valores de entrada- de un determinado punto, que se constituye en el centro de la unidad oculta.

La función de salida de la unidad oculta es una función radial con una altura de valor uno en el centro y un ancho que es controlable por un parámetro adicional. Las funciones radiales más habituales son la función gausiana, la función multicuadrática, la función multicuadrática inversa y la función Cauchy, que aparecen en la Fig. 21:

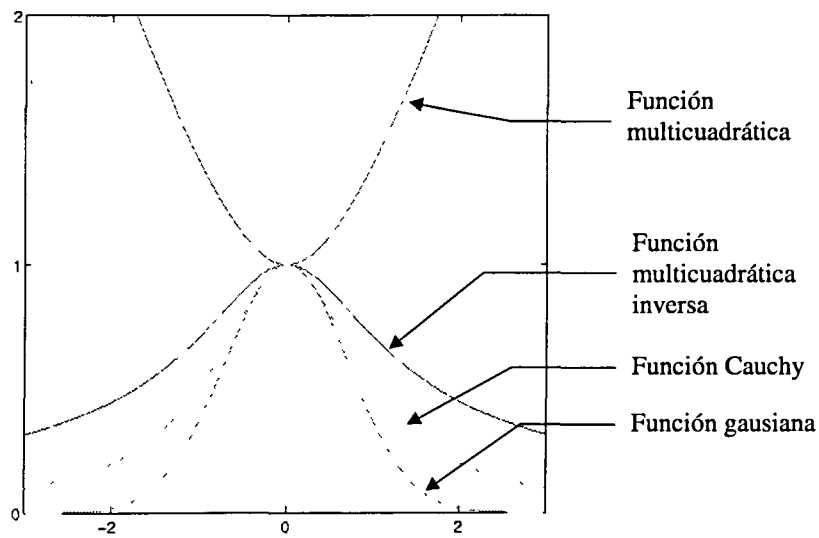


Fig. 21. Funciones radiales habitualmente empleadas en redes RBF

Así, la salida de una unidad oculta para un ejemplo determinado es una función de la distancia de este ejemplo al valor que es el centro de la unidad oculta. La salida es máxima (con valor 1) cuando el ejemplo está localizado exactamente en dicho centro, y decrece según el ejemplo se aleja de este valor central,

rápidamente al principio y después de forma asintótica hacia cero. La Fig. 22 representa el valor de salida de tres unidades ocultas de una red RBF ante un mismo valor de entrada.

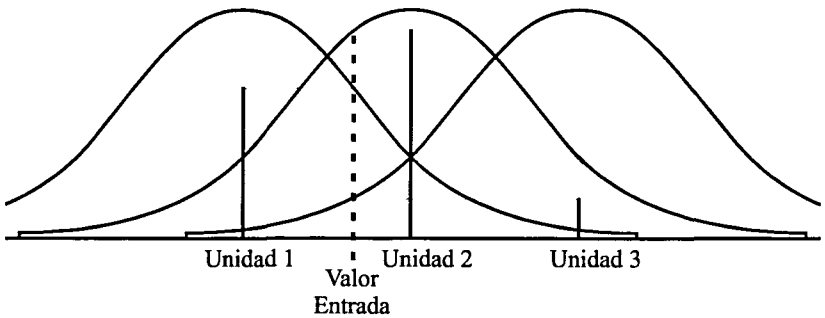


Fig. 22. Salida de 3 unidades ocultas de una red RBF ante un mismo valor de entrada. La unidad 1 tiene un valor de salida moderado, la unidad 2 un valor alto y la unidad 3 un valor bajo.

Si sólo hay un valor de entrada, la distancia de este valor al centro de la unidad es, simplemente, la diferencia entre el valor de la entrada y el parámetro que determina la localización de la unidad. Si hay varias entradas, la distancia del ejemplo al centro de la unidad se obtiene habitualmente a partir de la distancia euclidea:

$$\delta_j = \sqrt{\sum_{i=1}^{I}(x_i - w_{ij})}$$

donde x_i son los valores de entrada y w_{ij} son los parámetros.

La salida de la unidad oculta es una función radial de esta distancia, habitualmente la función gausiana:

$$y_j = \exp(-\delta^2/\sigma_j^2)$$

donde σ_j^2 es el parámetro que controla la anchura de la función de salida de la unidad oculta (lo rápido que la salida de la unidad decrece según aumenta la distancia del ejemplo al centro de la unidad). Así pues, cada unidad oculta envía a las unidades de salida un valor que indica lo alejado que un ejemplo está del valor que dicha unidad oculta representa.

Por su parte, cada unidad de salida tiene un parámetro por cada unidad oculta. La unidad de salida multiplica cada valor que recibe por el parámetro correspondiente, suma estos productos, y divide la suma por el total de entradas

que recibe de todas las unidades ocultas. El resultado de la unidad de salida es, por tanto:

$$z_k = \frac{\sum_{j=1}^{J} b_{jk} y_j}{\sum_{j=1}^{J} y_j}$$

donde b_{jk} es el valor de la conexión entre la unidad oculta j y la unidad de salida k.

El valor de salida de una unidad de salida no es la suma ponderada de sus entradas en el sentido en que estamos acostumbrados. Cada parámetro b_{jk} representa la salida deseada para un ejemplo localizado exactamente en el centro de la correspondiente unidad oculta. La salida de la unidad oculta indica lo cerca que cada ejemplo particular está de su centro, y la unidad de salida usa este valor como el peso del correspondiente parámetro. Por tanto, la unidad de salida realmente calcula una suma ponderada, pero en dicha suma los pesos son las entradas recibidas de las unidades ocultas y las cantidades ponderadas son los valores de los correspondientes parámetros.

4.2.2. Entrenamiento de una red RBF

El entrenamiento de una red RBF tiene como objetivo hallar los valores de los parámetros de la red que minimizan el error de predicción. Se distinguen tres tipos de parámetros: (1) pesos de las unidades de entrada a las unidades ocultas (determinan la ubicación del centro de la unidad oculta), (2) anchura de las funciones radiales y (3) pesos de las unidades ocultas a las unidades de salida. Además, se debe determinar, bien a priori bien durante el entrenamiento, el número óptimo de unidades ocultas para conseguir la mejor generalización.

Previa separación de los datos disponibles en datos de entrenamiento y de test, el algoritmo clásico realiza el entrenamiento de una red RBF en dos fases, lo cual permite que el entrenamiento sea muy rápido (Smith, 1993):

1. Cálculo de los pesos de las unidades de entrada a las unidades ocultas y de la anchura de las funciones radiales: la solución más sencilla consiste en hacer corresponder la ubicación de cada una de las unidades ocultas con la de un determinado registro, seleccionado de forma aleatoria del conjunto de datos de entrenamiento. Una solución más refinada consiste en seleccionar los centros de las unidades ocultas de manera que se minimice la suma de cuadrados de la distancia de cada registro a su centro más próximo (Sarle, 1994). Respecto a la anchura de la función radial, la aproximación más simple consiste en fijar todos los anchos al mismo valor, que es el promedio de las distancias de cada

centro al centro más próximo (Moody y Darken, 1988, 1989). Con este método se pretende cubrir, no todo el espacio de valores de entrada, sino la porción de este espacio donde los datos se hallan realmente. Hay que destacar que, bajo este enfoque, la capa oculta es autoorganizada, ya que sus parámetros dependen exclusivamente de la distribución de los valores de entrada, y no de la verdadera función de regresión que relaciona entradas y salidas.

2. Cálculo de los pesos de las unidades ocultas a las unidades de salida: se obtienen mediante aprendizaje supervisado. Los valores óptimos son los que producen el menor error de predicción, y se calculan mediante descenso del gradiente de error (sin propagación hacia atrás porque sólo hay una capa de pesos a estimar) o por regresión lineal.

Respecto al número de unidades ocultas, la solución clásica consiste en comparar varios modelos con diferente número de unidades cada uno, y seleccionar aquel que produce el menor error de predicción. En la práctica, sin embargo, se acostumbra a emplear alguna técnica de inclusión o exclusión secuencial de unidades (Orr, 1996). Por ejemplo, en la inclusión secuencial se comienza con una unidad oculta, y en cada paso se va añadiendo la unidad que más reduce el error. El entrenamiento finaliza cuando el error de predicción no ha disminuido sensiblemente en un número determinado de pasos. Los métodos de selección secuenciales requieren la división de los datos disponibles en tres grupos: datos de entrenamiento, de validación y de test, tal como se ha expuesto en apartados anteriores.

5
Clasificación de datos incompletos con redes neuronales artificiales

5.1. MÉTODOS DE CLASIFICACIÓN

Una breve revisión de las últimas publicaciones que incluyen un apartado de métodos estadísticos es suficiente para poner de manifiesto que, en los últimos años, el uso más habitual de las técnicas estadísticas consiste en establecer modelos predictivos de atributos de naturaleza continua, o modelos de clasificación de atributos categóricos, siendo estos últimos los más frecuentes.

El problema de la clasificación, también denominado reconocimiento de patrones o aprendizaje supervisado, aparece en un amplio rango de actividades humanas, y ya a principios de siglo fue objeto de los estudios de Fisher, quien formuló el conocido modelo de discriminación lineal. Desde entonces se han desarrollado decenas de algoritmos diferentes, que desde una perspectiva histórica pueden ser clasificados en tres grandes grupos (Michie, Spiegelhalter y Taylor, 1994):

- *Modelos estadísticos.* Se incluyen en este grupo modelos clásicos como el modelo de discriminación lineal de Fisher, la discriminación cuadrática y la discriminación logística, junto a un conjunto de modelos más actuales y flexibles, cuyo propósito es proporcionar una estimación de la función de distribución conjunta de los inputs dentro de cada categoría del output. Entre estas técnicas estadísticas "modernas" destacan la estimación no paramétrica de la función de densidad mediante funciones de regresión del tipo *kernel* (Fix y Hodges, 1951), el método de los *k* vecinos más cercanos (*k-nearest neighbour*), la clasificación mediante *Projection pursuit regression* (Friedman y Stuetzle, 1981), y los más recientes *Alternating conditional expectation (ACE)* (Breiman y Friedman, 1985) y *Multivariate adaptive regression spline (MARS)* (Friedman, 1991).
- *Algoritmos de inducción de reglas.* Consisten en reglas de tipo lógico, que frecuentemente se estructuran en árboles jerárquicos de decisión para generar un sistema de clasificación. Las reglas de clasificación son lo bastante simples como para poder ser fácilmente comprendidas, característica que distingue este tipo de algoritmos de las redes neuronales artificiales. El mecanismo de generación de reglas de este tipo de algoritmos fue establecido inicialmente por Quinlan (1983, 1986) en el algoritmo ID3. La mayoría de algoritmos

presentados posteriormente son variaciones encaminadas a mejorar la capacidad de generalización de estos modelos. Así, entre otros, se han desarrollado el CART (Breiman, Friedman, Olshen y Stone, 1984), el CN2 (Clark y Niblett, 1988), el Itrule (Goodman y Smyth, 1989), y el C4.5 (Quinlan, 1993).

- *Redes neuronales artificiales*. Engloban un amplio conjunto de modelos que, como se ha comentado en el capítulo 3, tienen la característica común de estar basados en un complejo sistema de unidades simples de procesamiento interconectadas, que incorporan relaciones no lineales entre los inputs. Las redes del tipo perceptrón multicapa con aprendizaje por retropropagación del error son las más utilizadas en la práctica, estimándose que sobre un 80% de proyectos de investigación se basan en este tipo de redes (Ruiz, Pitarque y Gómez, 1997).

5.1.1. ¿Redes neuronales o estadística?

Actualmente existe un amplio consenso en considerar los algoritmos de inducción de reglas como un conjunto de procedimientos con unas características distintivas y exclusivas, claramente diferenciables de los modelos estadísticos y de las redes neuronales artificiales. Sin embargo, la diferencia entre estos dos últimos no parece estar tan clara. La habitual virtud atribuida a las redes neuronales de no realizar restricciones sobre el modelo de distribución y de covariación de los datos (Garson, 1991; Rohwer, Wynne-Jones y Wysotzky, 1994; Chatterjee y Laudato, 1995) no justifica su identificación como un sistema de clasificación revolucionario, ya que actualmente existen varias técnicas estadísticas que comparten esta particularidad. Quizás como consecuencia de ello, hoy día hay un sentimiento generalizado entre los estadísticos de que las redes neuronales en general, y las del tipo perceptrón multicapa en particular, no son más que una forma altamente parametrizada de regresión no lineal (Flexer, 1995). En el extremo opuesto, los usuarios de redes mantienen que éstas representan, sino una alternativa, sí un complemento a los modelos estadísticos, con un conjunto de características propias y específicas, que resultan de especial utilidad en la modelización de problemas de alta complejidad, como por ejemplo el reconocimiento de códigos postales manuscritos (Le Cun et al., 1990).

Para poner luz en la polémica, Sarle (1994) sugiere la importancia de distinguir entre modelos y algoritmos de aprendizaje. Muchos modelos de redes neuronales son similares o idénticos a populares técnicas estadísticas como la regresión no lineal, el modelo lineal generalizado, la regresión no paramétrica, el análisis discriminante, o el análisis de *clusters*. También hay unos pocos modelos, como el *counterpropagation*, el *Learning vector quantization* (LVQ), y los mapas autoorganizados de Kohonen, que no tienen un equivalente estadístico preciso.

Pero, la diferencia más importante parece estar en el algoritmo de aprendizaje, es decir, en como son empleados los datos para calcular los parámetros del modelo: mientras que en una red neuronal el criterio de optimización de la fase de entrenamiento consiste en predecir la respuesta con el mínimo error posible en los datos de entrenamiento, en un modelo estadístico se suele recurrir a alguna técnica general como la estimación mínimo cuadrática, máximo-verosímil o alguna aproximación no paramétrica (Cheng y Titterington, 1994). En algunos casos, si se asume una distribución del error conocida, el entrenamiento de una red neuronal es equivalente a la estimación máximo-verosímil. Sin embargo, la aproximación tradicional de análisis con redes propone un criterio de optimización del error sin mención de errores aleatorios ni modelos probabilísticos.

Las disimilitudes entre ambas aproximaciones se manifiestan especialmente en la actitud adoptada en el análisis de un determinado problema. Como señala Breiman - comentario en Cheng y Titterington (1994) -, la primera pregunta que se hace un estadístico es: ¿Cuál es el modelo de probabilidad de los datos?, mientras que un usuario de redes pregunta: ¿Cuál es la precisión?. Para un estadístico, alta dimensionalidad (en número de parámetros) es 10 ó 15, mientras que en redes neuronales 100 parámetros es un número moderado, siendo habituales modelos con cientos. En problemas de elevada complejidad, un estadístico intentaría en primer lugar extraer algunas características que abarcaran la mayor parte de información relevante, y una vez reducida la dimensionalidad del problema, usaría alguna técnica estándar de clasificación. Por contra, un usuario de redes pasaría todos los datos directamente a una red con miles de parámetros, dejaría el ordenador trabajando durante varías horas o días y obtendría una función empírica de clasificación. A este respecto, parece que la polémica de si los datos se han de ajustar al modelo o el modelo se debe ajustar a los datos no es tal en el ámbito de las redes neuronales, ya que en éstas el modelo surge directamente de los datos (Stern, 1997).

Siguiendo la sugerencia de Flexer (1995), una revisión a los trabajos publicados que comparan ambos enfoques permite distinguir dos grandes grupos: los que se dedican a hacer comparaciones teóricas y los que, principalmente mediante simulación, se centran en comparaciones empíricas. A continuación se revisan algunas de estas publicaciones.

5.1.1.1. Comparaciones teóricas

Para realizar una comparación teórica de los modelos estadísticos y los modelos de redes neuronales, ambos deben ser descritos en un marco teórico común. La mayor parte de trabajos en esta línea han sido realizados por estadísticos que, naturalmente, hacen uso de la teoría matemático-estadística.

Sarle (1994) constituye un ejemplo paradigmático de este grupo de autores. En su primer trabajo de relevancia proporciona una equivalencia entre los modelos estadísticos y los modelos de redes neuronales (ver Tabla 15):

Tabla 15. Equivalencia entre modelos estadísticos y modelos de red neuronal (Sarle, 1994)

Modelo estadístico	Modelo de red neuronal
Regresión lineal múltiple	Perceptrón lineal simple
Regresión logística	Perceptrón no lineal simple
Función discriminante llineal	Adaline
Regresión no lineal múltiple	Perceptrón multicapa
Análisis de componentes principales	Modelos Hebbianos de aprendizaje asociativo
Análisis de clusters	Redes de Kohonen
Variaciones de K vecinos más cercanos	*Learning vector quantization* (LVQ)
Kernel Regresión	Funciones de base radial (RBF)

Y dos años después el mismo autor (Sarle, 1996) presenta una equivalencia de la terminología empleada en ambas disciplinas, que se resume en la Tabla 16:

Tabla 16. Equivalencia en la terminología estadística y de redes neuronales (Sarle, 1996)

Terminología estadística	Terminología de redes neuronales
Muestra	Datos de entrenamiento
Muestra de validación	Datos de validación, test
Residual	Error
Término de error	Ruido
Término de interacción	Conexión funcional
Variable independiente	Variable independiente, entrada
Variable dependiente	Variable dependiente, salida
Modelo	Arquitectura
Estimación	Entrenamiento, aprendizaje
Valor estimado de un parámetro	Peso de una conexión
Constante	Peso umbral
Función de regresión	Función
Regresión, análisis discriminante	Aprendizaje supervisado
Reducción de dimensionalidad	Aprendizaje no supervisado
Análisis de *clusters*	Aprendizaje competitivo

Ripley (1993) es otro autor que trabaja a nivel de comparaciones teóricas. Se centra en las redes con propagación de la información hacia adelante, explorando el problema de los mínimos locales del algoritmo de propagación del error hacia atrás. Concluye que la minimización del error medio cuadrático puede ser conseguida con más eficacia por un algoritmo de mínimos cuadrados no lineales o por un algoritmo de minimización general como el algoritmo de *Newton-Raphson.*

La principal conclusión que se obtiene de estos trabajos es que los modelos de redes neuronales pueden ser adecuadamente descritos en un marco estadístico, dadas las inherentes similitudes entre ambas aproximaciones.

5.1.1.2. Comparaciones empíricas

La cada día mayor disponibilidad de datos de investigación ha posibilitado que un elevado número de trabajos comparativos de modelos estadísticos y de redes neuronales se realicen aplicando ambas técnicas a una misma matriz de datos. Por otra parte, en el estudio de las características particulares de un algoritmo, el papel de la simulación es crucial, ya que permite controlar determinadas propiedades del conjunto de datos empleado.

Una extensa comparación entre 23 métodos de clasificación, incluyendo modelos estadísticos, algoritmos de inducción de reglas y redes neuronales, en un total de 22 conjuntos de datos diferentes, fue realizada por Michie, Spiegelhalter y Taylor (1994). Los resultados de este vasto estudio indican que no existe un método que en general pueda ser considerado superior, en términos de porcentaje de errores de clasificación, en los diversos conjuntos de datos analizados. La Tabla 17 resume a qué categoría de clasificación de las tres analizadas pertenecen los cinco mejores métodos en cada conjunto de datos.

Tabla 17. Categoría de clasificación de los cinco mejores algoritmos en cada conjunto de datos (Michie, Spiegelhalter y Taylor, 1994)

	Conjunto de datos																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	Total
ESTAD.	5	4	3	3	3	3	3	3	3	2	3	3	2	2	1	2	2	1	1	0	0	0	49
RNA	0	1	2	2	2	2	2	2	2	3	1	1	2	1	2	0	0	1	0	0	0	0	26
IND.R.	0	0	0	0	0	0	0	0	0	0	1	1	1	2	2	3	3	3	4	5	5	5	35

Thrun, Mitchell y Cheng (1991) dirigieron un extenso estudio comparativo de diferentes algoritmos en la clasificación de 432 registros en una de dos categorías a partir de seis atributos. Aunque algunos algoritmos fueron ligeramente

superiores, en las conclusiones del trabajo no se destaca especialmente ninguno de ellos.

Ripley (1993) compara un amplio conjunto de métodos estadísticos y redes neuronales. Aunque las conclusiones se refieran a un único conjunto de datos, algunos resultados son interesantes. En cuanto a la capacidad predictiva, el método de los k vecinos más cercanos, la red perceptrón multicapa y la *Projection pursuit regression* son los mejores, aunque si se tienen en cuenta otros aspectos como la velocidad de convergencia o la interpretabilidad de la función de clasificación, la eficacia de la red se ve sustancialmente disminuida.

En otros estudios, realizados por Bonelli y Parodi (1991), Huang y Lippmann (1987), Sethi y Otten (1990), se concluye que diferentes tipos de redes neuronales ofrecen resultados similares, o ligeramente superiores, a los de los procedimientos estadísticos.

5.1.1.3. Homogeneización de los criterios de comparación

Como se ha comentado, se han realizado numerosos estudios aplicados para comparar algoritmos de clasificación, obteniéndose en general resultados dispares. Las causas de ello pueden ser diversas y con diferentes orígenes. Henery (1994) hace una exhaustiva revisión de los aspectos que pueden conducir a conclusiones tan diferentes.

En primer lugar, algunos sesgos comunes en la elección de los algoritmos comparados y de las matrices de datos empleadas son:

1. La selección de algoritmos es limitada, de manera que los elegidos no son representativos del estado real de desarrollo del ámbito que representan.
2. En muchos casos los autores han desarrollado sus propios algoritmos, y son expertos en ellos, pero no en el uso del resto de métodos comparados.
3. Los conjuntos de datos son usualmente pequeños, y en el caso de ser simulados, frecuentemente se establecen sesgos a favor de un determinado algoritmo.

Puesto que la ausencia de consenso sobre qué método de clasificación es más adecuado parece ser debida, en parte, a las propiedades particulares de cada conjunto de datos utilizado, de manera que un procedimiento u otro resulta mejor según las características específicas de los datos en que se aplique, Henery (1994b) presenta un conjunto de índices descriptivos de una matriz de datos que deberían ser incluidos en cada publicación:

1. Número de casos: es el número total de casos en la muestra.
2. Número de atributos: es el número total de variables registradas, especificando cuales son predictivas y cuales predichas. También se debe

indicar si se incluyen atributos categóricos recodificados a variables ficticias, así como el número de categorías de cada uno.

3. Test de homogeneidad de covariancias: la matriz de covariancias es fundamental en la teoría de discriminación lineal. Su homogeneidad debe ser testada en el conjunto de variables continuas mediante alguna medida como el estadístico M de Box.
4. Media del valor absoluto de los coeficientes de correlación: el conjunto de correlaciones entre todos los pares de variables da alguna indicación sobre su interdependencia. Un sencillo valor de resumen puede ser obtenido promediando el valor absoluto de estos coeficientes.
5. Desviación de la normalidad: la asunción de normalidad multivariada subyace en muchos procedimientos clásicos de discriminación, por lo que es recomendable proporcionar el resultado de una prueba de significación de ajuste a la normal, o simplemente una medida de no normalidad. Algunas de estas pruebas son revisadas por Ozturk y Romeu (1992).
6. Asimetría y apuntamiento: medidos en cada variable continua, además de promediados sobre todas ellas.
7. Entropía de atributos y de clases: es una medida de la aleatoridad de una variable aleatoria, que se calcula a partir de la probabilidad de cada valor o de la prevalencia de cada categoría.

Otra importante posible causa de la falta de acuerdo parece ser el criterio de comparación empleado. En este sentido, Henery (1994) establece tres aspectos básicos a valorar en la evaluación de un sistema de clasificación:

1. Precisión: se refiere a la fiabilidad del modelo, usualmente representada por el porcentaje de casos correctamente clasificados. Cuando algunos errores pueden tener un coste especialmente elevado se utilizan otras medidas que incluyen esta característica.
2. Velocidad: hace referencia a la velocidad tanto en la fase de entrenamiento-estimación del modelo como en la de clasificación. En ocasiones puede ser preferible un clasificador con menor precisión pero mayor velocidad.
3. Comprensión: se refiere a si el modo de funcionamiento del sistema de clasificación es comprensible para un ser humano. Este atributo es fundamental cuando la clasificación ha de ser realizada, en última instancia, por un humano.

Finalmente, el preprocesamiento realizado en los datos también puede influir sobre los resultados de la comparación de diferentes procedimientos. La fase de preprocesamiento puede ocupar más tiempo que la fase de análisis propiamente dicha, y son muy pocos los investigadores que informan de las manipulaciones

realizadas sobre los datos. Henery (1994b) resume los principales aspectos que deben ser incorporados en la publicación de un estudio:

1. Información ausente: operaciones realizadas sobre los datos faltantes de la matriz.
2. Selección de variables: en matrices de datos con muchos atributos es usual hacer una selección previa de las variables que serán susceptibles de ser incorporadas en el modelo predictivo. En dicho caso debe indicarse el método empleado para reducir la dimensionalidad de la matriz así como las variables excluidas.
3. Transformaciones de variables: es usual realizar transformaciones de los datos para mejorar la predicción. Las más frecuentes, como la raíz cuadrada, logaritmos o transformaciones recíprocas están encaminadas a normalizar variables continuas. También es muy habitual, y en general recomendable, generar variables ficticias (*dummy variables*) para codificar las variables categóricas. Los atributos jerárquicos son un caso especial que, además de desinformado, no suele estar correctamente resuelto en la mayoría de trabajos. Por atributos jerárquicos se entienden aquellos que sólo son relevantes en un subconjunto de casos de la muestra; por ejemplo, el número de embarazos es una información de interés sólo en mujeres. Las soluciones habituales son dejar este ítem missing o con el valor cero en hombres, aunque la mejor alternativa sería generar un nuevo atributo con información conjunta de las dos variables implicadas, que permitiera diferenciar cuando un missing o un cero son en realidad un "no aplicable", aspectos que han sido presentados en el capítulo 1.

5.2. REDES NEURONALES CON DATOS INCOMPLETOS

La investigación sobre cómo incorporar información incompleta en las fases de entrenamiento y de predicción de una red neuronal artificial es reciente. Prácticamente no existen trabajos publicados antes de la presente década, si bien es cierto que muchos de los avances actuales se deben al ingente trabajo realizado en los 80 sobre la aproximación estadística al problema de la información ausente.

Puesto que en las redes con aprendizaje supervisado, de elección en problemas de clasificación, los datos missing pueden aparecer sólo en una o bien en las dos fases de análisis con la red (entrenamiento y predicción), y como además sólo se pueden dar en el conjunto de variables independientes (si la variable dependiente de un registro es missing el caso no se ha de incluir en los datos de entrenamiento), algunos autores han desarrollado procedimientos específicos para tratar los valores faltantes de las variables independientes en la fase de

entrenamiento, mientras que otros se han centrado en su influencia durante la fase de predicción, suponiendo que el entrenamiento se ha realizado con un conjunto de datos completos. Sin embargo, la mayoría de propuestas incluyen soluciones válidas para ambas fases.

A grandes rasgos, se pueden distinguir seis enfoques en el tratamiento de datos incompletos con una red neuronal artificial:

1. *Eliminar los casos que contengan algún valor faltante.* Como se ha comentado en capítulos anteriores, este método es el más utilizado en la práctica pero al mismo tiempo el que puede provocar sesgos más importantes, especialmente si los datos perdidos no tienen un origen aleatorio. Además, la reducción de la muestra puede ser tan importante que imposibilite un adecuado entrenamiento y, por otra parte, en estudios en que interesa la predicción individual sobre la colectiva, la eliminación de un registro es inviable.

2. *Usar variables indicadoras* que reflejen para cada dato de entrada si éste es missing o conocido. Evidentemente, la codificación con variables indicadoras se ha de realizar tanto en los datos de entrenamiento como en los de predicción.

3. *Imputar un valor a cada dato faltante.* Consiste en reemplazar cada valor missing por un valor que, o bien no tenga efecto sobre el comportamiento de la red, o bien sea una estimación del verdadero valor desconocido. Se realiza tanto en los datos missing del conjunto de datos de entrenamiento como en los datos de predicción.

4. *Entrenar varias redes*, cada una con una topología específica en función de los posibles patrones de missing de los datos. Esta técnica se denomina *"Network reduction"* y también es aplicable tanto durante el entrenamiento como durante la predicción.

5. *Modificar el algoritmo de aprendizaje de la red* para que soporte información ausente. Este tipo de procedimientos se estructuran en torno a la estimación máximo verosímil con datos incompletos como una alternativa al tradicional cómputo del gradiente del error, y sólo se aplica durante la fase de entrenamiento de la red.

6. *Utilizar el teorema de Bayes* para incorporar información previa en el cálculo de la probabilidad de cada categoría de la variable dependiente, partiendo de un conjunto de datos incompletos. Este procedimiento sólo es válido en la fase de predicción.

En los sucesivos apartados se amplían estos enfoques y se presentan ejemplos aplicados de algunos de ellos.

5.2.1. Variables indicadoras

Una solución para incorporar los valores faltantes en el entrenamiento y en la predicción de una red perceptrón multicapa consiste en usar una topología que duplique el número de unidades de entrada, de manera que cada variable independiente (o categoría de ésta) es representada por dos unidades de la capa de entrada. La primera unidad de cada par (*"flag unit"*) indica si el valor del input es missing (1) o conocido (0), y con ella se pretende que la red aprenda cuando ha de tener en cuenta el valor del input que le acompaña. En el caso de un valor conocido, la segunda unidad tiene ese valor, mientras que en el caso de un valor missing, si la red interpretara correctamente la variable indicadora no tendría importancia que valor se asignara a la segunda unidad, aunque en la práctica se han hallado mejores resultados asignando el valor promedio de la variable.

5.2.2. Imputación de valor

De las diferentes técnicas de imputación de valor a cada dato faltante, en el campo de las redes neuronales artificiales con retropropagación del error se han estudiado una amplia variedad. Así, Vamplew y Adams (1992) y Pitarque y Ruiz (1996) presentan sendos estudios de simulación comparando las siguientes técnicas:

- *Imputación del valor cero (IC)*: En arquitecturas de red con inputs estandarizados, cuando un valor perdido en una variable independiente es sustituido por el valor cero, se anula el efecto que la correspondiente unidad de la capa de entrada pudiera tener sobre el resultado de la red.
- *Imputación incondicional de la media (IM)*: En el caso de variables cuantitativas se reemplaza cada valor ausente con la media, calculada en los casos que sí tienen valor en la variable. En variables categóricas, el valor faltante es sustituido por la moda de la distribución. Puesto que los valores de las variables independientes cuantitativas suelen ser normalizados para que su media sea cero y su variancia uno, en este tipo de variables esta solución es equivalente a imputar el valor cero.
- *Imputación de un valor aleatorio (IA)*: Consiste en imputar a cada dato perdido un valor aleatorio dentro del intervalo de valores de la variable. Si se trata de un dato missing en un input cuantitativo normalizado se imputa un valor aleatorio en el rango [-1,+1], mientras que los valores faltantes en inputs categóricos se sustituyen por un valor aleatorio correspondiente a alguna de sus categorías.
- *Imputación por regresión simple (IRS)*: Consiste en sustituir cada valor missing de una determinada variable *y* con el valor predicho por el modelo de

regresión simple estimado a partir de su mejor covariante x. Cuando en esta variable x se halla un dato perdido, se reemplaza en el modelo de regresión por su valor medio. Esta técnica implica estimar tantos modelos de regresión simple como número de variables independientes con valores faltantes halla.

- *Imputación por regresión múltiple (IRM)*: Cada valor missing en una variable es reemplazado con el valor predicho por el modelo de regresión múltiple estimado a partir del resto de variables. En cada modelo de regresión múltiple, los datos faltantes en las variables predictoras son provisionalmente reemplazados por su valor medio. Esta técnica también implica estimar tantos modelos de regresión como variables independientes con valores faltantes halla.
- *Imputación del valor predicho por una red neuronal (IR)* (*best estimate*): Un método de imputación más sofisticado consiste en sustituir cada valor missing por la predicción resultante de una red neuronal, normalmente del tipo perceptrón multicapa, aunque en ocasiones se utiliza una red autoasociativa. En un problema con veinte variables independientes incompletas se deben crear veinte redes, de manera que en cada una se modifica el rol de una variable de predictora a predicha. Como sucede con los modelos de imputación por regresión, los valores missing en las variables que actúan como inputs en cada "subred" suelen ser sustituidos por el valor promedio de la variable.

5.2.3. Network reduction

El método *Network reduction* consiste en entrenar varias redes para predecir la variable dependiente en estudio, usando en cada una de ellas un conjunto diferente de inputs. El número de redes viene determinado por los posibles patrones de datos faltantes que se puedan dar en el conjunto de variables de entrada. Durante la fase de predicción, el conjunto completo de datos puede ser clasificado mediante la presentación de cada caso a la red correspondiente, en función de qué datos sean missing.

Sharpe y Solly (1995) utilizan esta estrategia en un pequeño sistema diagnóstico para clasificar la función tiroidea en una de tres categorías. Para ello disponen de un total de cuatro pruebas de laboratorio (TT3, TT4, T3u, TBG), dos de las cuales (T3u, TBG) son ocasionalmente missing. La solución pasa por entrenar cuatro redes que cubran el rango completo de combinaciones de datos faltantes (ver Fig. 23).

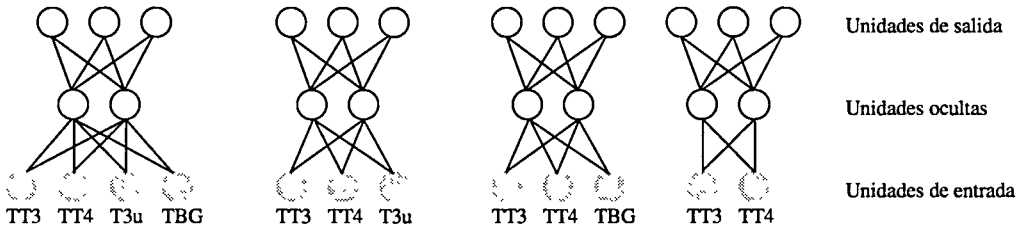


Fig. 23. Redes necesarias para clasificar la función tiroidea (Sharpe y Solly, 1995)

Aunque los autores concluyen que esta técnica es más eficaz que la imputación de valor mediante red neuronal (IR) y los algoritmos de inducción de reglas -por ejemplo NewID o ID3 (Boswell, 1992)-, presenta el inconveniente de ser computacionalmente muy costosa de aplicar cuando el número de patrones de datos faltantes es muy elevado, ya que para cada uno de ellos se debe diseñar y entrenar una red específica. Además, si el número de registros disponibles para el entrenamiento es limitado, posiblemente no hayan suficientes casos en cada patrón de datos ausentes como para entrenar con garantías las diferentes redes (Tresp, Ahmad y Neuneier, 1994).

Es por ello que el procedimiento *Network reduction* es aplicable únicamente cuando el número de variables independientes es limitado o cuando sólo algunas de ellas son susceptibles de presentar valores ausentes.

Pitarque y Ruiz (1996) presentan un estudio de simulación para comparar las técnicas de imputación IC, IM, IA, IRS, IRM, la de eliminación de los registros incompletos y la que utiliza variables indicadoras (también incluyen la imputación mediante una red no supervisada), tanto en la fase de entrenamiento como en la de ejecución de una red neuronal MLP. El objetivo es obtener una clasificación binaria a partir de 15 inputs cuantitativos. Cada muestra simulada contiene un total de 100 registros, 50 de los cuales tienen dos valores de entrada desconocidos, de manera que el porcentaje total de datos missing es del 6.67%. Cada técnica es ensayada en un conjunto de 15 redes perceptrón multicapa 15-6-1 entrenadas durante 2000 épocas. Pitarque y Ruiz (1996) concluyen que durante la fase de entrenamiento los métodos IC, IRS, IM, IRM proporcionan el menor error medio cuadrático, mientras que el peor resultado se obtiene con la técnica de eliminación de registros. Los resultados en la fase de ejecución son similares.

Por otra parte, Vamplew y Adams (1992) comparan las técnicas IC, IM, IR, uso de variables indicadoras y *Network reduction*. Utilizan un conjunto de 398 registros (298 de entrenamiento y 100 de test) para clasificar semillas en uno de diez grupos a partir de siete variables independientes. Eliminando aleatoriamente sólo un valor de entrada en cada registro, los mejores resultados se obtienen con

IR, *Network reduction* y mediante el uso de variables indicadoras. Con dos valores missing en cada registro se excluyó la técnica *Network reduction* por la enorme cantidad de redes que hubieran sido necesarias, obteniendo el mayor porcentaje de clasificaciones correctas con la técnica IR y por medio del uso de variables indicadoras.

5.2.4. Estimación máximo-verosímil

El tradicional algoritmo de descenso del gradiente del error, utilizado en el entrenamiento de redes perceptrón multicapa (MLP), puede ser sustituido por el método de estimación máximo verosímil, ya que en general un descenso en la función de error puede ser interpretado como un ascenso en la función de verosimilitud (White, 1989).

Para obtener la estimación máximo-verosímil en presencia de datos missing en las variables independientes es necesario conocer la función de distribución conjunta de los inputs de la red. En principio, ello puede ser conseguido mediante múltiples redes MLP, cada una de ellas aprendiendo una función de distribución condicional particular. Por ejemplo, si el patrón de datos faltantes es monótono, los datos ausentes pueden ser completados por un conjunto de x-1 redes, donde x es el número de inputs. Sin embargo, para soportar cualquier patrón de datos incompletos, esta aproximación es ineficaz, puesto que el número de redes necesario aumenta exponencialmente con la dimensionalidad de los datos. Una solución más general consiste en utilizar el algoritmo EM presentado en el capítulo 2 para estimar los pesos de la red.

Ghahramani y Jordan (1994) presentan los resultados de un estudio para clasificar una variable categórica con tres categorías a partir de cuatro variables independientes cuantitativas. La muestra estaba formada por 100 registros de entrenamiento y 50 de test. Los autores concluyen que la estimación máximo verosímil de los pesos de la red mediante el algoritmo EM es tan efectiva como los métodos de imputación, cuando el porcentaje de valores faltantes es inferior al 40%, mientras que con porcentajes superiores el algoritmo EM es claramente mejor. A pesar de los resultados, la implementación de este método en programas de ordenador es tan escasa que representa un avance más teórico que aplicado.

5.2.5. Teorema de Bayes

Ahmad y Tresp (1993) discuten técnicas Bayesianas para calcular la probabilidad de cada valor de un output categórico a partir de datos incompletos durante la fase de predicción, suponiendo que la red ha sido entrenada con datos completos.

Un sencillo ejemplo sirve para ilustrar la aplicación del cálculo Bayesiano de la probabilidad posterior a partir de la distribución conjunta de los inputs. Considérese la situación de la Fig. 24. Representa un atributo en un espacio

bidimensional con seis posibles resultados. Suponiendo que la red ha sido entrenada con datos completos, si durante la clasificación de un nuevo caso sólo se tiene el valor z_0 correspondiente al input z, la clasificación debe asignar la misma probabilidad $p(C_i|z_o)$ a las categorías 1, 2 y 3, y una probabilidad cero a las categorías 4, 5 y 6. Cualquier método de imputación de valor al dato missing del input x conduciría a un resultado erróneo; así, si por ejemplo se imputara el valor medio de x, la clase 2 tendría una probabilidad cercana a uno, y las clases 1 y 3 cercana a cero. Para obtener la probabilidad posterior correcta es necesario integrar la predicción de la red respecto a los valores desconocidos de x.

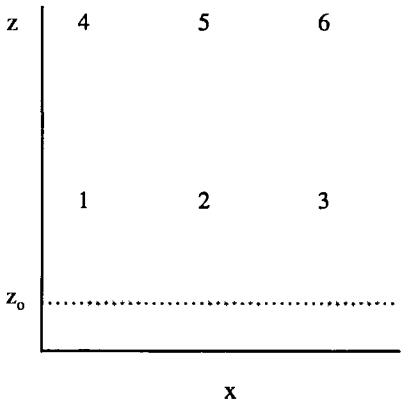


Fig. 24. Representación de un atributo con 6 categorías a partir de los inputs x,z

Además se debe tener en cuenta otro factor: la distribución de probabilidad sobre el input desconocido x puede estar condicionada por el valor del input conocido z. Con una distribución como la de la Fig. 25, la clasificación debe asignar a la clase 1 la mayor probabilidad. Así, la probabilidad posterior se ha de obtener integrando la predicción de la red respecto a los valores de x (inputs desconocidos), ponderando por la distribución conjunta de x,z (todos los inputs). Formalmente, si denominamos X a los datos que se habrían observado en ausencia de valores faltantes, podemos descomponer $X=(X_{obs}, X_{mis})$, donde X_{obs} representa los valores realmente observados y X_{mis} los valores desconocidos. La probabilidad posterior de cada categoría (C_i) se obtiene a partir de:

$$p(C_i|X_{obs}) = \frac{p(C_i, X_{obs})}{p(X_{obs})} = \frac{\int p(C_i, X_{obs}, X_{mis}) dX_{mis}}{p(X_{obs})} =$$

$$= \frac{\int p(C_i|X_{obs}, X_{mis})\, p(X_{obs}, X_{mis})\, dX_{mis}}{p(X_{obs})}$$

Con $p(C_i|X_{obs},X_{mis})$ aproximada por la predicción de la red, y donde es necesario conocer la distribución conjunta de los inputs $p(X_{obs},X_{mis})$.

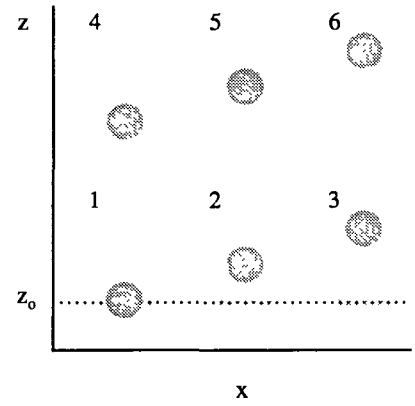


Fig. 25. Representación de un atributo con 6 categorías a partir de los inputs *x,z*. Las zonas oscuras representan regiones de mayor probabilidad