CAPÍTULO 3 MODELIZACIÓN NEURONAL: *APROXIMACIÓN ESTADÍSTICA Y ECONOMÉTRICA*

3.1. Introducción: Las redes neuronales como herramienta de *modelización estadística*.

Los modelos neuronales se conocen por su asombrosa capacidad para aprender, generalizar y retener conocimiento de los datos, por esta razón y desde una óptica econométrica y estadística, pueden ser considerados como modelos de regresión y modelos discriminantes no lineales[1]. A continuación profundizamos en los siguientes aspectos: en primer lugar, realizamos una aproximación de los modelos neuronales a los modelos estadísticos y econométricos[2] (modelos de *Regresión* clásico, modelos de probabilidad, *Probit* y *Logit*, modelos de variables *Latentes* y los modelos *Generalizados*). En segundo lugar, enumeramos los esfuerzos realizados sobre cómo realizar su interpretación estadística y en tercer lugar, planteamos una visión econométrica de los mismos.

Siguiendo a Cheng B. y Titterington[3] (1994), las razones por las cuales los estadísticos deberían considerar la modelización neuronal como una herramienta más son las siguientes: proporcionan topologías muy similares para la mayoría de los modelos estadísticos; muchos de los problemas comunes de modelización e inferencia estadística pueden ser tratadas desde ambas metodologías; las técnicas estadísticas en ocasiones pueden ser realizadas mediante modelos neuronales[4] o de forma híbrida[5]; algunos modelos neuronales poseen en su estructura

[1] Véase Sarle, Waren S. (1994). **Neural Networks and Statistical Models**, *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, pp. 1-13.

[2] Todos los modelos econométricos anteriores pueden ser considerados como casos particulares de los modelos neuronales. Véase una comparativa en Cooper, C.B., J. (1999). **Artificial neural networks versus multivariate statistics: an applications from economics**, *Journal of Applied Statistics*, Vol. 26(8), pp. 909-921.

[3] Véase Cheng, Biang ; Titterington, D.M. (1994). **Neural Networks: A Review from a Statistical Perspective**, *Statistical Science*, Vol. 9, No 1, pp. 2-54.

[4] Pueden construirse reglas de discriminación lineal y cuadrática, reglas para calcular componentes principales y para aproximar probabilidades Bayesianas.

[5] Véase una aplicación mixta entre modelos ARIMA y Back Propagation, ARIMABP, Tseng, F-M.; Yu, H-Cheng, Tzeng, G-Hsiung. (2002). **Combining neural network model with seasonal time series ARIMA mode**l, *Technological Forecasting & Social Change*, 69, pp.71-87.

elementos probabilísticos (modelos de *Hopfield* y *máquina de Boltzmann*); y en último lugar, existe cada vez más, un esfuerzo de vincular dichas disciplinas[6].

3.2. *Paralelismos* entre los Modelos *estadísticos* y Neuronales.

3.2.1. Modelo *Regresión Lineal*.

Los modelos de regresión lineal múltiples pueden ser representados mediante una red neuronal *feedfoward* de dos capas, denominada *Adaline*[7] (Widrow y Hoff (1960)), que posee una función de transferencia lineal o identidad. Su arquitectura es esencialmente la misma que la del modelo *Perceptron*, ya que ambas estructuras utilizan neuronas con funciones de transferencia muy parecidas[8], pero existe una diferencia esencial en el mecanismo de aprendizaje. La red *Adaline* y su versión múltiple, *Madaline*, utilizan la *regla delta* de Hidrow-Hoff o regla del mínimo error cuadrado medio (*LMS Algorithm*)[9], es decir, el error cometido por el modelo compara la diferencia entre el valor deseado y la salida lineal. En cambio para el modelo de *Perceptron* la comparación es respecto a la salida binaria. Dicha diferencia permite que los modelos Adaline / Madaline alcancen el mínimo error de forma más sencilla que el propio modelo Perceptron (la convergencia del proceso de entrenamiento es más fácil). Para poder obtener un modelo *Adaline* a partir de un *Perceptron*, debemos añadir dos componentes, el primero de ellos es un término de tendencia ("*Bias*" (Umbral)), que proporciona un grado de libertad adicional, y el segundo de los componentes consiste en añadir una condición bipolar[10] a la salida, es decir, si el resultado de la red es positivo, adjudicarle el valor (+1) y si es negativa, el valor (–1).

---

[6] Véase en especial, Kay, J.W.; Titterington, D.M. (1999). **Statistics and Neural Networks. Advances at the Interface**, Oxford University Press.

[7] El término *Adaline* ha cambiado ligeramente con el paso de los años, inicialmente se llamaba ADAptive Linear Neuron, posteriormente se definió como ADAptive LINear Element.

[8] Para el primer caso una función de transferencia *lineal* y para el segundo una del tipo, *escalón*. Si bien podemos encontrar en el ámbito de la ingeniería modelos *Adalines* que se les aplica al resultado del modelo un función bipolar, es decir, salidas lineales positivas, (+1) y salidas negativas, (-1) e incluso salidas analógicas utilizando en su fase final, funciones tangente hiperbólicas o exponenciales.

[9] LMS significa *Least Mean Square*, mínimos cuadrados. Supone que la actualización de las ponderaciones es proporcional al error que la neurona comete.

[10] En este caso la función de transferencia asociada a la salida es del tipo *escalón* simétrica.

Su expresión es la siguiente,

$$y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$$
$$X = (x_1, x_2, \cdots, x_p)'$$
$$\{\beta_j, j = 0,1,\cdots, p\}$$

donde "*y*" es el valor de salida, "*x*" es el vector de entrada, β_j es el vector de ponderaciones, (véase ilustración 1.3.2.1.).

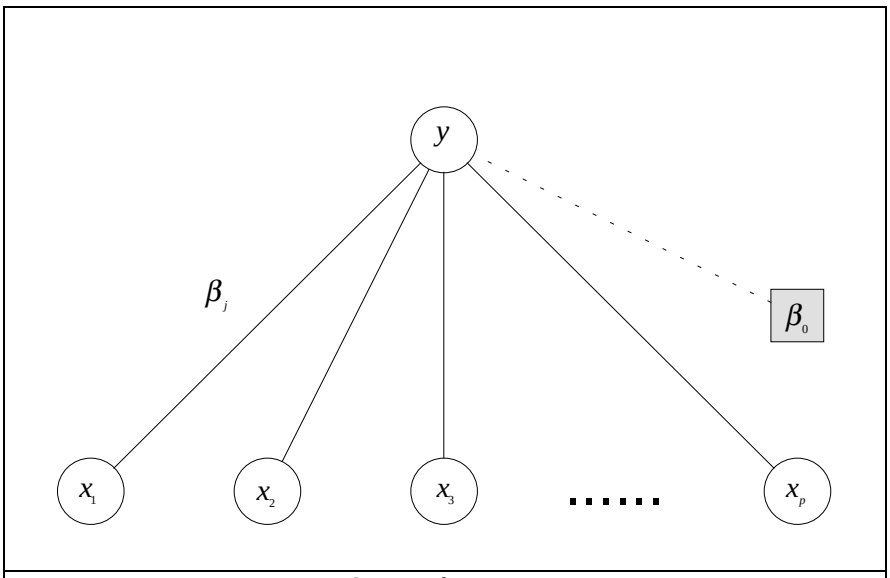


Ilustración 1.3.2.1.
Modelo neuronal *Adaline*

La propia topología del modelo *Adaline*[11] posee un conjunto de desventajas. En primer lugar, no generaliza bien con datos que no se han utilizado en el proceso de aprendizaje. En segundo lugar, es más engorrosa desde la óptica computacional que el modelo de regresión lineal y en último lugar, posee las limitaciones del propio *Perceptron*, como por ejemplo, imposibilidad de calcular la función *XOR* (función lógica or-exclusivo). Sin embargo, posee algunas ventajas, como por ejemplo, no presupone aspectos como la *homoscedasticidad* ni la *ortogonalidad* (premisas del modelo de regresión lineal), permitiendo una mayor robustez en el proceso de estimación.

[11] Respecto a su funcionamiento, se ha comprobado que es útil en varias aplicaciones. Una de las más conocidas es la utilización de dicho modelo como supresor de ecos en los *modems*.

Las limitaciones que posee el modelo *Adaline* pueden ser solucionadas planteando una nueva topología, la red lineal adaptiva múltiple (*Madaline*), véase ilustración 2.3.2.1. Esta red es similar al modelo *Multilayer Perceptron* (MLP) y puede ser utilizada para representar modelos de regresiones aparentemente no relacionadas[12].

Su expresión formal es la siguiente,

$$
\begin{aligned}
&y_1 = \beta_{0i} + \sum_{j=1}^{p} \beta_{1j} x_j \\
&y_2 = \beta_{0i} + \sum_{j=1}^{p} \beta_{2j} x_j \\
&\vdots \\
&y_v = \beta_{0i} + \sum_{j=1}^{p} \beta_{ij} x_j \\
&Y = (y_1, y_2, \cdots, y_v)', \\
&X = (x_1, x_2, \cdots, x_p)' \\
&\left\{ \beta_{ji}, \begin{array}{l} j = 0,1,\cdots,p \\ i = 1,\cdots,v \end{array} \right\}
\end{aligned}
$$

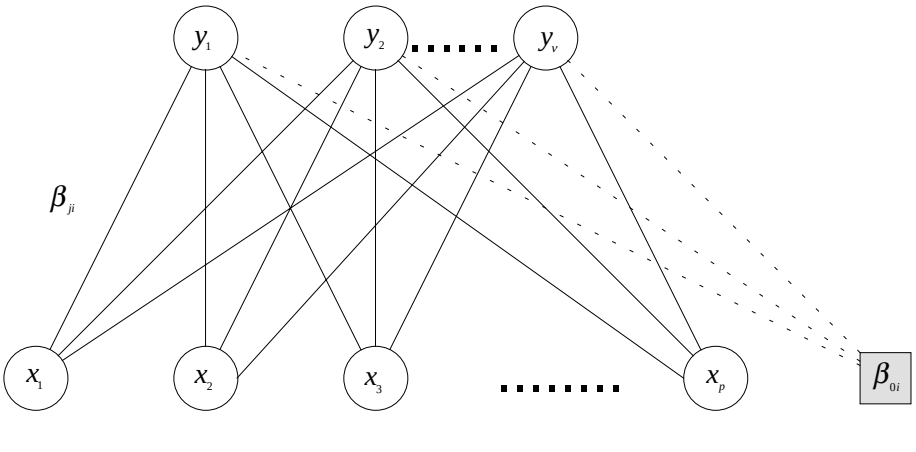


Ilustración 2.3.2.1.
Modelo neuronal *Madaline*

Finalmente, si se utilizan *outputs* retardados como entradas en una red *Adaline*, obtenemos una ecuación formada por elementos temporales de carácter lineal AR(*p*), es decir,

[12] Los modelos de *regresiones aparentemente no relacionadas* se definen así debido a que la relación entre ellas no está explicitada analíticamente sino que viene generada por las correlaciones entre los términos de error.

un modelo *autoregresivo*[13]cuyo orden es igual al número de ponderaciones del modelo neuronal, suponiendo un *bias* nulo, (véase ilustración 3.3.2.1.).

Su expresión formal es,

$$y_t = \beta_0 + \sum_{j=1}^{p} \beta_j y_{t-j}$$
$$X = \left(y_{t-1}, y_{t-2}, \cdots, y_{t-p}\right)'$$
$$\{\beta_j, j = 0,1,\cdots, p\}$$

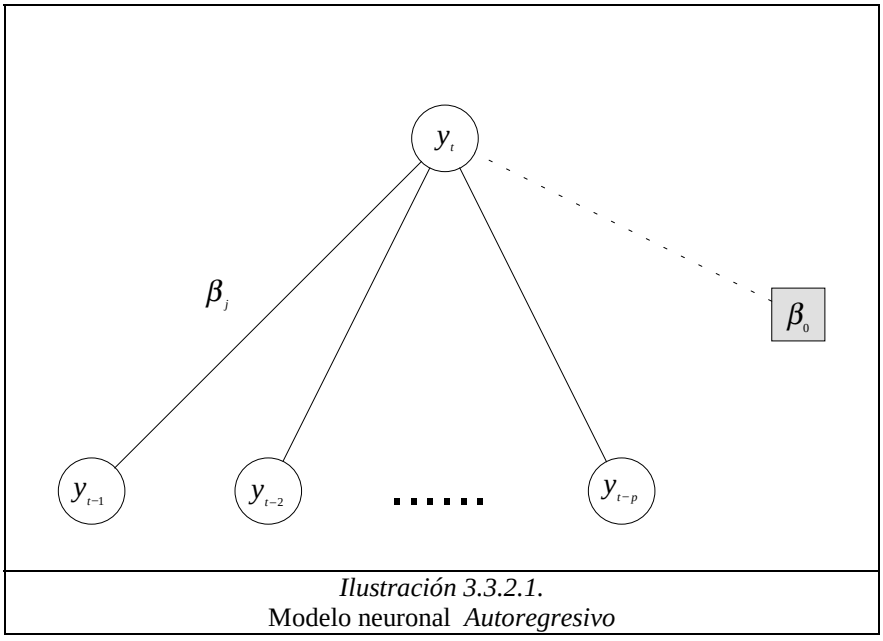


Ilustración 3.3.2.1.
Modelo neuronal *Autoregresivo*

3.2.2. Modelos *Logit* y *Probit*.

Tal y como se ha desarrollado en el punto anterior, la red *Adaline* posee dos capas con una función de transferencia lineal, pero si asociamos a la salida una función bipolar $[0;1]$, es decir, si la salida es positiva (+1) y si es negativa o igual a cero (0), entonces existe plena similitud con los modelos de variables dependientes cualitativas: modelos *Probit* y *Logit*[14] muy adecuados para problemas de clasificación.

[13] Existe la posibilidad de utilizar la modelización neuronal para detectar de forma automática el modelo estocástico ARMA que mejor se ajuste a los datos, véase Jae Kyu Lee; Won Chul Jhee. (1994). **A two-stage neural network approach for ARMA model identification with ESACF**, *Decisión Support Systems*, 11, pp. 461-479.

[14] Véase para mayor desarrollo, Greene, W.H. (1993).**Econometric Analysis**, 2ª Ed. , Macmillan Publishing .

La función de transferencia puede ser una función no decreciente continua, permitiendo representar funciones de *distribución acumulada*. Si ésta función es la distribución acumulada logística, entonces el modelo obtenido representa el valor esperado condicional del modelo *Logit* binario, cuya expresión formal es,

$$y_i = f\left(\sum_{j=0}^{p} \beta_{ij} x_j\right) = \frac{\exp\left(\sum_{j=0}^{p} \beta_{ij} x_j\right)}{1+\exp\left(\sum_{j=0}^{p} \beta_{ij} x_j\right)}$$

En cambio, si utilizamos la distribución acumulada normal, entonces obtenemos el valor esperado condicional de una variable aleatoria binaria generada por un modelo *Probit*,

$$y_i = \Phi\left(\sum_{j=0}^{p} \beta_{ij} x_j\right),$$

donde, Φ es la función de distribución acumulada normal.

Por lo tanto, un modelo neuronal de dos capas permite representar los modelos de regresión *Logit* y *Probit*. Sin embargo, debido a las limitaciones de la red neuronal de dos capas, la mayoría de las aplicaciones de clasificación de las redes neuronales artificiales utilizan una o más capas intermedias, de forma que, se ha demostrado que una red neuronal artificial de dos capas tiene un funcionamiento similar al del análisis discriminante lineal. La incorporación de una capa oculta mejora considerablemente la exactitud de la clasificación, Ripley[15] (1993), detalla de forma clara las relaciones entre las redes neuronales artificiales y los métodos tradicionales.

3.2.3. Modelo de *variables latentes* con indicadores múltiples y causas múltiples (*MIMIC*).

Los modelos causales que contienen variables latentes se han aplicado a varias áreas de las ciencias sociales, como pueden ser, psicología, economía, educación, etc. Dichas variables son observables hipotéticamente y no directamente, pero pueden influenciar sobre las relaciones entre variables observables, de forma que, pueden ser efectos (*indicadores*) o causas de las variables latentes o ambas cosas.

[15] Véase Ripley, B. D. (1993). **Statistical aspects of Neural Networks,** capítulo 2, en Barndorff-Nielsen, O.E. ; Jensen, J.L.; Kendall, W.S. (1993). **Networks and Chaos – Statistical and Probabilistic Aspects,** Chapman & Hall.

Algunas veces los modelos causales con múltiples indicadores y múltiples causas de variables latentes se denominan, modelos *MIMIC*[16]. Estos modelos pueden ser representados mediante un modelo neuronal *feed-forward* de tres capas, cuya expresión formal es la siguiente,

$$y_i = \beta_0 + \sum_{j=1}^{q} \beta_j g\left(\sum_{i=1}^{p} \phi_{ij} x_i + \phi_{0j} \right)$$
$$X = (x_1, x_2, \cdots, x_p)'$$
$$Y = (y_1, y_2, \cdots, y_i)$$
$$\{ \phi_{ij}, i = 0,1, \cdots, p, j = 1, \cdots, q \}$$
$$\{ \beta_j, j = 0,1, \cdots, q \}$$

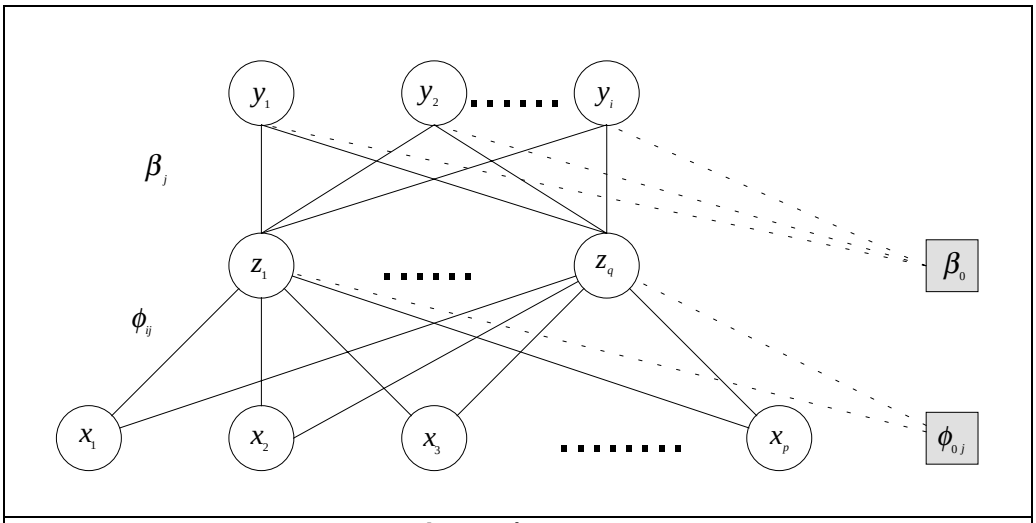


Ilustración 1.3.2.3.
Representación neuronal de un modelo *MIMIC*

La capa oculta del modelo neuronal representado por el vector, $Z = (z_1, \cdots z_q)$ está determinada linealmente por el vector de entrada, $X = (x_1, x_2, \cdots, x_p)'$, que corresponde a un conjunto de causas exógenas observables. A su vez las unidades en la capa oculta determinan linealmente las unidades de salida $Y = (y_1, y_2, \cdots, y_i)$, que corresponde a un conjunto de indicadores endógenos observables. Finalmente el vector, $Z = (z_1, \cdots z_q)$, representa a las variables latentes de un modelo *MIMIC*, (véase la ilustración 1.3.2.3.).

[16] Véase un ejemplo en Roberto Esposti, R.; Pierani, P. (2000). **Modelling technical change in Italian agriculture: a latent variable approach**, *Agricultural Economics*, 22 , pp. 261-270.

3.2.4. **Familia de Modelos** *Generalizados.*

Recordemos que el modelo neuronal *Multilayer Perceptron* (MLP), definido en el apartado 2.5.1, posee la siguiente expresión,

$$\begin{aligned} & y = f\left(\beta_0 + \sum_{j=1}^{q} \beta_j g\left(\sum_{i=1}^{p} \phi_{ij} x_i + \phi_{0j} \right) \right) \\ & X = \left(x_1, x_2, \cdots, x_p\right)' \\ & \left\{ \phi_{ij}, i = 0,1,\cdots, p, j = 1,\cdots, q \right\} \\ & \left\{ \beta_j, j = 0,1,\cdots, q \right\} \end{aligned}$$

donde," y " es el <u>valor de salida</u> (*output*), " x " es el <u>vector de entrada</u> (*inputs*), β_j es el vector de ponderaciones o <u>parámetros</u> a estimar (une las entradas con la capa oculta) y ϕ_{ij} son las <u>ponderaciones</u> que vinculan la capa oculta con la salida. La <u>función de transferencia</u>, $g(\cdot)$, puede poseer características lineales o no lineales y la <u>función de salida</u>, $f(\cdot)$, en la mayoría de los casos es de naturaleza lineal[17]. La expresión anterior no es más que una función de regresión no lineal de cierta complejidad, que puede aglutinar casos particulares muy cercanos a la familia de los modelos de regresión *Aditivos* (AM), cuya expresión es la siguiente,

$$\begin{aligned} & y = \beta_0 + \sum_{j=1}^{q} \beta_j g_j(\cdot) \\ & X = \left(x_1, x_2, \cdots, x_p\right)' \\ & g_j(\cdot) = g_j\left(\sum_{i=1}^{p} \phi_{ij} x_i + \phi_{0j} \right) \\ & \left\{ \phi_{ij}, i = 0,1,\cdots, p, j = 1,\cdots, q \right\} \\ & \left\{ \beta_j, j = 0,1,\cdots, q \right\} \end{aligned}$$

siendo, $g_j(\cdot)$, las funciones no especificadas de carácter no paramétrico. El planteamiento anterior excluye una variedad de modelos importante, como por ejemplo, los modelos para datos de supervivencia o binarios, de ahí, la necesidad de una forma más general, los denominados modelos *Aditivos Generalizados* (GAM)[18].

---

[17] Para un mayor detalle véase el apartado 3.4.3.

[18] Véase Hastie, T.J. ; Tibshirani, R.J. (1990). **<u>Generalized Additive Models</u>**, Chapman, London. Referenciado en Cheng, Biang ; Titterington, D.M. (1994). **Neural Networks: A Review from a Statistical Perspective**, *Statistical Science*, Vol. 9, No 1, pp. 2-54 y en http://www-stat.stanford.edu/~hastie/Papers/.

Cuya expresión es,

$$\begin{aligned} & y = f\left(\beta_0 + \sum_{j=1}^{q} \beta_j g_j(\cdot) \right) \\ & X = \left(x_1, x_2, \cdots, x_p \right)' \\ & g_j(\cdot) = g_j\left(\sum_{i=1}^{p} \phi_{ij} x_i + \phi_{0j} \right) \\ & \left\{ \begin{matrix} \phi_{ij}, i = 0,1,\cdots, p, j = 1,\cdots, q \\ \beta_j, j = 0,1,\cdots, q \end{matrix} \right\} \end{aligned}$$

donde, $f(\cdot)$, es una función "*link*" monótona, conocida a priori que incluye varios casos particulares, uno de ellos es, el modelo *Logístico*[19], cuya forma aditiva posee la siguiente expresión,

$$\begin{aligned} & \log\left(\frac{\Pr(y=1|X)}{1-\Pr(y=1|X)} \right) = \beta_0 + \sum_{j=1}^{q} \beta_j g_j(\cdot) \\ & X = \left(x_1, x_2, \cdots, x_p \right)' \\ & g_j(\cdot) = g_j\left(\sum_{i=1}^{p} \phi_{ij} x_i + \phi_{0j} \right) \\ & \left\{ \begin{matrix} \phi_{ij}, i = 0,1,\cdots, p, j = 1,\cdots, q \\ \beta_j, j = 0,1,\cdots, q \end{matrix} \right\} \end{aligned}$$

Desde una perspectiva más general, podemos definir la media condicional del modelo de regresión aditivo definido como, $E(y|X)$, lo cual nos conduce a,

$$\begin{aligned} & f\left(E(y|X)\right) = \beta_0 + \sum_{j=1}^{q} \beta_j g_j(\cdot) \\ & X = \left(x_1, x_2, \cdots, x_p \right)' \\ & g_j(\cdot) = g_j\left(\sum_{i=1}^{p} \phi_{ij} x_i + \phi_{0j} \right) \\ & \left\{ \begin{matrix} \phi_{ij}, i = 0,1,\cdots, p, j = 1,\cdots, q \\ \beta_j, j = 0,1,\cdots, q \end{matrix} \right\} \end{aligned}$$

donde, si la función *link*, $f\left(E(y|X)\right)$, es la identidad, utilizamos modelos aditivos y lineales para datos de respuesta Gaussiana. En cambio, si la función *link* es un Logit[20] (detallado anteriormente) o un Probit, $f\left(E(y|X)\right) = \Phi^{-1}\left(E(y|X)\right)$ entonces, se utiliza para modelizar

[19] Logaritmo del cociente de disparidad o *Logit* (abreviación del término "*logistic probability unit*"). La transformación *Probit* (abreviación del término "*probability unit*") permite que la relación entre la variables del modelo puedan considerarse lineal.

[20] Véase aplicaciones del modelo *Logit* en su versión Multinomial aplicada al fraude en la cobertura de seguros de automóvil en Ayuso Gutiérrez, M. (1998). **Modelos Econométricos para la detección del fraude en el seguro del automóvil**, Tesis Doctoral, Universidad de Barcelona, Barcelona.

probabilidades de tipo Binomial. Por último, si la función *link* es $f\left(E\left(y|X\right)\right)=\log\left(E\left(y|X\right)\right)$ se denominan modelos log-aditivos para datos de conteo[21] ("*Count Data*").

Los modelos anteriores pueden flexibilizarse aún más a través de modelos mixtos, como por ejemplo, los modelos *Aditivos Parcialmente Lineales* (APLM) y los modelos con *Componentes Bivariantes Aditivas* o modelos aditivos con interacción (GAPLM), (véase tabla 1.3.2.4.).

Tabla 1.3.2.4.

	Modelos	*Nomenclatura*	*Expresión*
1	Modelos Aditivos	AM	$E\left(y\|X\right)=\beta_0+\sum_{j=1}^{q}\beta_j g_j(\cdot)$
2	Modelos Generalizados Aditivos	GAM	$E\left(y\|X\right)=f\left(\beta_0+\sum_{j=1}^{q}\beta_j g_j(\cdot)\right)$
3	Modelos Aditivos parcialmente lineales	APLM	$E\left(y\|X\right)=\beta_0+\sum_{j=1}^{q}\beta_j g_j(\cdot)+\sum_{i=1}^{p}\phi_{ij}x_i$
4	Modelos con componentes bivariantes aditivas o modelos aditivos con interacción	GAPLM	$E\left(y\|X\right)=f\left(\beta_0+\sum_{j=1}^{q}\beta_j g_j(\cdot)+\sum_{i=1}^{p}\phi_{ij}x_i\right)$

Existe un caso particular de mucho interés, dentro del grupo de los modelos aditivos, que es el modelo de regresión *Projection Pursuit* (PPR)[22], donde las funciones $g(\cdot)$, se las define como funciones "*ridge*" que acogen un abanico importante de funciones no lineales. Además destacamos que los modelos *Generalizados Aditivos* (GAM)[23] son la versión no lineal de los modelos *Lineales Generalizados* (GLM)[24], donde el error del modelo puede quedar explicado por otro comportamiento que no los Gaussianos, por ejemplo, *Binomial*, *Poisson*, *Gamma*, *Gausiana Inversa*, etc. Todos los modelos anteriores utilizan funciones no paramétricas para estimar la relación entre *inputs* y el *output*.

---

[21] Véase para una aplicación en el entorno del *credit scoring*, Dionne, G; Artis, M.; Guillén, M. (1996). **Count Data models for a credit scoring system**, *Journal of Empirical Finance*, 3, pp. 303-325.

[22] Desde una perspectiva más amplia de estos modelos, existe una similitud entre los métodos de exploración *Projection Pursuit* y la técnica de análisis componentes independientes (ICA), véase apartado 2.2, de forma que, podemos considerarla como un análisis factorial de naturaleza no Gaussiana.

[23] Véase para procedimientos de estimación y contraste para los modelos (GAM), Yang L.; Sperlich, S.; Härdle, W. (2002). **Derivative estimation and testing in generalized additive models**, *Journal of Statistical Planning and Inference*, Article in press, corrected in Prof., (aceptado Marzo 2002).

[24] Véase para mayor detalle Fahrmeir, L.; Tutz, G. (1994). **Multivariate Statistical Modelling Based on Generalized Linear Models**, Springer. Como posibles aplicaciones de estos modelos, véase Alcañiz Zanón, M. (1996). **Modelos de Poisson generalizados con una variable de exposición al riesgo**, Tesis Doctoral, Universidad de Barcelona, Barcelona; Wood, G. R. (2002). **Generalised linear accident models and goodness of fit testing**, *Accident Analysis and Prevention*, 34, pp. 417-427. Véase aplicación para datos de supervivencia en Biganzoli, E.; Boracchi, P.; Marubini, E. (2002). **A general framework for neural network models on censored survival data,** *Neural Networks*, 15, pp. 209-218.

Finalmente existen otros métodos "relacionados" con los modelos aditivos generalizados con la misma finalidad y que son no menos importantes. Resaltamos el procedimiento adaptativo para la regresión mediante *Splines*, (MARS[25], *Multivariate Adaptive Regression Splines*) como uno de los de mayor utilización.

Todas las expresiones anteriores en general son de difícil interpretación, pero se están registrando avances significativos. En esta línea, Refenes (1995) expresa la similitud de un modelo neuronal con un modelo *Aditivo Generalizado* (GAM). Así para el modelo neuronal especificado en la ilustración 1.3.2.4. y considerando como función de transferencia una *sigmoide*, tenemos que el *output* del modelo neuronal adquiere la forma siguiente,

$$y = \frac{1}{1+e^{-(\beta_0+\beta_1 z_1+\beta_2 z_2)}}$$

y los *outputs* intermedios de la capa oculta, $(z_1; z_2)$, poseen las siguientes expresiones,

$$z_1 = \frac{1}{1+e^{-(\phi_{01}+\phi_{11}x_1+\phi_{12}x_2)}} \quad \text{y} \quad z_2 = \frac{1}{1+e^{-(\phi_{02}+\phi_{21}x_1+\phi_{22}x_2)}}$$

Si consideramos lineal la función de salida en la neurona de la capa output, podemos expresar dicho modelo como,

$$\begin{aligned} y &= \beta_0 + \beta_1 z_1 + \beta_2 z_2 = \\ &= \beta + \beta_1\left(\frac{1}{1+e^{-(\phi_{01}+\phi_{11}x_1+\phi_{12}x_2)}}\right) + \beta_2\left(\frac{1}{1+e^{-(\phi_{02}+\phi_{21}x_1+\phi_{22}x_2)}}\right) \end{aligned}$$

Tal y como comenta, Refenes (1995), una de las transformaciones sobre los datos más utilizada es la logarítmica, si aplicamos dicha transformación para reducir la varianza o extraer el efecto de los outliers, se obtiene $\ln(y) = f(\ln(x_1), \ln(x_2))$, que rescribiendo los términos exponenciales[26],

$$e^{(\phi_{11}\ln(x_1)+\phi_{12}\ln(x_2))} = e^{\left(\ln\left(x_1^{\phi_{11}}\right)+\ln\left(x_2^{\phi_{12}}\right)\right)} = e^{\left(\ln\left(x_1^{\phi_{11}}x_2^{\phi_{12}}\right)\right)} = x_1^{\phi_{11}}x_2^{\phi_{12}}$$

[25] Véase Hastie, T; Tibshirani, R.; Friedman, J. (2001). **The Elements of Statistical Learning. Data Mining, Inference, and Prediction**, pp. 283-289, Springer. Para una adaptación para series temporales, véase De Gooijer, J.G.; Ray, B.K.; Kräger, H. (1998). **Forecasting exchange rates using TSMARS**, *Journal of International Money and Finance*, 17 , pp. 513-534.

[26] Se ha ignorado los términos independientes o *bias* para una exposición más sencilla.

permite definir de forma alternativa el modelo neuronal como un modelo aditivo generalizado, cuya expresión es,

$$\ln(y) = \beta_1\left(\frac{x_1^{\phi_{11}} x_2^{\phi_{12}}}{1 + x_1^{\phi_{11}} x_2^{\phi_{12}}}\right) + \beta_2\left(\frac{x_1^{\phi_{21}} x_2^{\phi_{22}}}{1 + x_1^{\phi_{21}} x_2^{\phi_{22}}}\right)$$

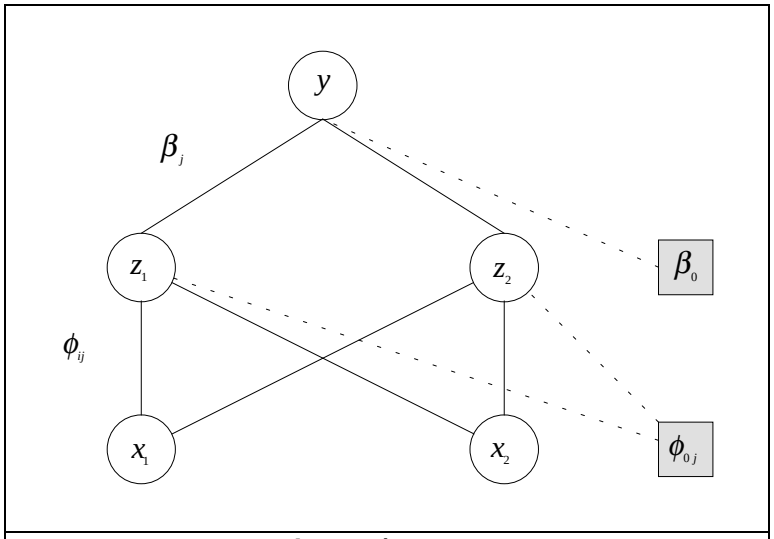


Ilustración 1.3.2.4.
Representación neuronal
de un modelo *aditivo generalizado* (GAM)

Consideramos de mucho interés profundizar aún más en la modelización *no paramétrica* o *semiparamétrica* en donde los modelos neuronales poseen ya un espacio reservado de importancia.

3.3. Interpretación de los modelos de redes neuronales desde la *óptica estadística*.

Las limitaciones de los modelos neuronales puede resumirse en tres[27]. En primer lugar, no existe ninguna teoría formal para determinar la *estructura* óptima de un modelo neuronal[28], así aspectos como, la determinación del número adecuado de capas, el número de neuronas en la capa oculta, etc, deben decidirse en muchos casos de manera heurística. En segundo lugar, no existe algoritmo óptimo que asegure el mínimo global en la superficie de error cuando esta presenta mínimos locales. Por último, las propiedades estadísticas de las redes neuronales generalmente no están disponibles y por lo tanto, no se puede llevar a cabo ninguna inferencia estadística con garantías. Además es difícil llegar a interpretar los parámetros de un modelo neuronal una vez terminado el proceso de aprendizaje.

Frente a todas las anteriores dificultades, existen investigadores como ejemplo, Cheng y Titterington[29] (1994) que están realizando una labor muy importante para conectar las disciplinas de los métodos estadísticos y la tecnología de redes neuronales[30].

Una vez detallado las dificultades presentes en la modelización neuronal, es necesario profundizar un poco más en el desarrollo y realización de los mismos[31]. Con este objetivo debemos tener presente los siguientes aspectos, en primer lugar, la *identificación* correcta de los *inputs* e *outputs* más importantes, en segundo lugar, la elección de su *estructura* adecuada, incluyendo el número necesario de *capas ocultas* y el número de neuronas para cada una de las capas intermedias o ocultas y en último lugar, la definición de los criterios de evaluación de los modelos estimados.

---

[27] Véase Min Qi. (1996). **Financial applications of artificial Neural Networks**. *Handbook of Statistics,* Vol. 14, pp. 537-538 (Edited by G.S. Maddala and C.R.Rao) Elsevier.

[28] Existen mecanismos de búsqueda mediante algoritmos genéticos, *Structure-Adaptive Neural Networks*, véase capítulo 15, Chin-Teng Lin y George Lee, C.S. (1996). **Neural Fuzzy Systems. A Neuro-Fuzzy Synergism to Intelligent Systems**, Prentice Hall PTR, Upper Saddle River, NJ.

[29] Ambos autores presentan como posibles líneas de profundización en el ámbito neuronal: la modelización matemática, la investigación teórica en el campo de la *Neurocomputación* y el desarrollo de herramientas orientadas a la predicción y el reconocimiento de patrones.

[30] Véase Riccia, G.D.; Lenz, H. J.; Kruse, R. (1997). **Learning, Networks and Statistics**, *International Centre for Mechanical Sciences*, *Courses and Lectures*, No. 382, Springer.

[31] Véase una aportación desde la óptica de la simulación, Intrator, OP.; Intrator, N. (2001). **Interpreting neural-network results: a simulation study**, *Computational Statistics & Data Analysis,* 37, pp. 373-393.

Además, existen otros aspectos que deben ser considerados tanto para los métodos tradicionales como para los métodos neuronales y son: la *calidad* de los datos, el grado de *representatividad* de los mismos y el *tamaño* muestral que se posea.

Respecto al primero de los aspectos relevantes, la elección de las variables, lógicamente depende principalmente del objetivo que posea el estudio, este aspecto posee una carga subjetiva que depende de los profesionales que formen parte del grupo de investigación[32]. Este proceso de selección previo puede ser apoyado en métodos tradicionales de reducción de la dimensión, como por ejemplo, la elección de un grupo más reducido de variables estadísticamente significativas.

Los mecanismos que pueden ser utilizados para realizar la preselección pueden ser de diferente índole, a continuación exponemos tres posibilidades: mediante una regresión de las variables dependientes de un grupo extenso de variables independientes, la utilización de técnicas multivariantes (como por ejemplo, el análisis de componentes principales (PCA)) y la regresión s*tepwise* que supone una selección secuencial mediante criterios estadísticos. Para minimizar el efecto del tamaño entre los *inputs* y los *outputs* y así aumentar la efectividad del algoritmo de aprendizaje, se normaliza el conjunto de datos para que esté dentro de un intervalo específico dependiente de la función de transferencia. Por ejemplo, si una red neuronal artificial posee una función de transferencia *Sigmoidal* o *Logística* en la capa de salida, es necesario escalar la salida para estar en el intervalo de [0,1], ya que por lo contrario el proceso de estimación podría quedar alterado y no se podría llegar a la generalización de la relación entre entradas y salidas. Normalmente, las variables se normalizan para tener de media cero y desviación estándar la unidad, si bien existen otros métodos.

[32] Es habitual utilizar variables independientes como inputs o *entradas* de red y utilizar variables dependientes como outputs o *salidas* del modelo, si bien, existen otros casos en donde las salidas se encuentran de forma simultánea tanto la variable explicativa como la reacción de la misma en función de sus valores, es decir, tanto la *predicción* como la *acción* que conlleva dicha predicción, generando reglas de decisión. Véase Beltratti, A.; Margarita, S.; Terna, P. (1996). **Neural Networks for economic and financial modellig**, ITP, pp. 153-158.

El segundo de los aspectos se refiere a la estructura del modelo, para el cual realizamos las siguientes apreciaciones. La primera de ellas consiste en que, de forma habitual, se definen modelos con al menos una capa oculta, debido a la limitación que posee un modelo neuronal con solo dos capas (*inputs* y *outputs*). En segundo lugar, se ha demostrado que los modelos neuronales con un máximo de dos capas ocultas pueden aproximar un conjunto particular de funciones con una exactitud arbitraria y que con una sola capa oculta es suficiente para aproximar cualquier función continua (Hornik, Stinchcombe y White[33], 1989). En tercer lugar, la elección del número de capas ocultas representa un compromiso, de forma que, si es demasiado pequeño, el modelo obtenido puede no aproximar con la exactitud deseada, pero si es demasiado grande, se puede producir un *sobreajuste* que puede evitar el proceso de generalización en la fase de test, es decir, fuera de la muestra utilizada para el aprendizaje, generando un modelo sobreparametrizado. Un método eficiente y muy utilizado en el ámbito anterior es la *validación cruzada*[34], que permite determinar el número de unidades de la capa oculta, de forma que optimice su funcionamiento en una parte de la muestra reservada a tal efecto.

Refenes (1995) ha tratado otros métodos comunes para el diseño de redes óptimas. Pueden ser clasificados en tres grupo, (véase tabla 1.3.3.).

El primer grupo de técnicas descansa sobre la idea de que el número de nodos ocultos de un modelo depende del tamaño de la muestra utilizada en el proceso de estimación y su número se define a priori, como por ejemplo, el número de conexiones debería ser inferior a un 10% del tamaño de la muestra (n) o el número de unidades ocultas del orden de $(n-1)$ o $\log(n)$. El problema principal de estas técnicas es que realizan un análisis estático y precisan de un análisis previo de la dimensionalidad del vector de los *inputs*. Debido a esta limitación sólo pueden proporcionar una estimación muy aproximada del tamaño de la capa oculta.

[33] Véase White, H. (1992). **Artificial Neural Networks. Aproximation and Learning Theory**, Blackwell Publishers, pp. 12-28.

[34] Dentro de los mecanismos de cálculo del error de predicción, el método *bootstrap* propuesto en 1979 por Efron se utiliza en el entorno neuronal para el cálculo de intervalos de confianza, si bien existen otras técnicas, como por ejemplo el *jacknife* (predecesor del *bootstrap*) y la *validación cruzada*. Véase para mayor detalle, Rojas, R. (1996). **Neural Networks. A systematic Introduction**, pp. 233-237.

Tabla 1.3.3.

Métodos (grupo 1)	***Autores***
Estimación analítica	Lippmann, R.P. (1987)
Estimación Heurística[35]	Zanakis,S.H. y Evans, J.R. (1981)
Métodos (grupo 2)	***Autores***
Algoritmo *Tiling*	Mezard, M.;Nadal, J.(1989)
Correlación en cascada	Fahlman, S.E.; Lebiere, C. (1990)
Procedimiento CLS	Refenes, A.N.; Chan, B. (1993)
Combinación discriminantes lineales	Gallant, S.I. (1986)
Método Generación	Honavar, V.; Uhr,L. (1988)
Procedimiento *stepwise*	Kerling, M. (1992)
Métodos de creación de nodos dinámico	Miller, B; Reinhardt (1990)
Algoritmo *upstart*	Frean, M.R.A. (1989)
Métodos (grupo 3)	***Autores***
Reducción en dos etapas	Sietsma, J; Dow, R.F.J. (1991)
Selección artificial	Hergert, F.; Finnoff, W; Zimmermann, H.G. (1992)
Sensibilidad del peso del error	Karnin, E.D. (1990)

Fuente: Refenes, A.P. (1995). **Neural networks in the capital markets**, Wiley, pp. 32-54 y elaboración propia.

El segundo grupo se refiere a técnicas constructivas, como la correlación en cascada (Fahlman y Lebiere, 1990), algoritmo "Tiling" (Mezard y Nadal, 1989), árbol de decisión neuronal (Gallant, 1986), algoritmo "*upstart*" (Frean,1989) y el procedimiento CLS (Refenes y Vithlani, 1991). Estos métodos constructivos de capas ocultas realizan de forma secuencial su proceso, introduciendo una a una las diferentes capas a medida que el modelo lo necesita. Tal y como comenta Refenes (1995), estas técnicas garantizan la convergencia del modelo hacia su generalización pero no su estabilidad.

Por último, las técnicas que suponen una reducción paulatina de los modelos, operan lógicamente en la dirección opuesta, reduciendo la red y eliminando las conexiones redundantes o con menor sensibilidad. Este grupo incluye las siguientes técnicas: reducción de modelo en dos etapas[36](Sietsma y Dow,1991); selección artificial[37] (Hergert, Finnoff y

[35] Véase para más información Díaz, A. y otros. (1996). **Optimización Heurística y redes neuronales en dirección de operaciones e ingeniería**, pp. 24-36, Paraninfo, Madrid.

[36] Véase Sietsma, J.; Dow, R.F.J. (1991). **Creating artificial neural networks that generalize**, *Neural Networks*, 4, pp. 67-79. Referenciado en Refenes, A. P. (1995). **Neural Networks in the Capital Markets**, Wiley.

[37] Véase Hergert, F.; Finnoff, W.; Zimmermann, H.G. (1992). **A comparison of weight elimination methods for reducing complexity in neural networks**, *Interntational Joint Conference on Neural Networks*, Maryland, Vol. 3, pp. 980-987. Referenciado en Refenes, A. P. (1995). **Neural Networks in the Capital Markets**, Wiley.

Zimmermann,1992); y sensibilidad de error al excluir paulatinamente los pesos del modelo[38] (Karnin, E.D., 1990), aunque no siempre es posible una reducción óptima, (véase ilustración 1.3.3.).

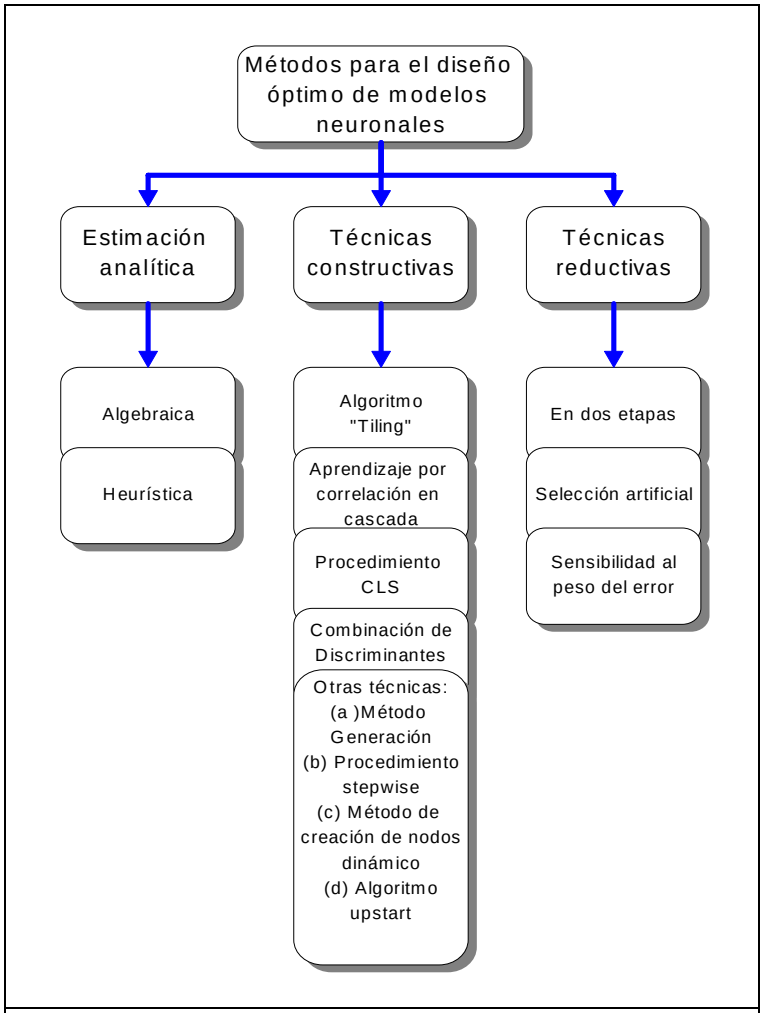


Ilustración 1.3.3.
Métodos de *diseño* óptimo para Modelos Neuronales

En último lugar y de la misma forma que cualquier modelo econométrico, los modelos econométricos neuronales precisan de criterios de evaluación que permitan comparar el funcionamiento de modelos alternativos y la selección del mejor. La tabla 2.3.3. presenta algunos de los criterios más habituales que reflejan intereses diferentes.

[38] Véase Karnin, E.D. (1990). **Simple procedure for pruning backpropagation trained neural networks**, *IEEE. Trans. On Neural Networks*, 1, pp. 20. Referenciado en Refenes, A. P. (1995). **Neural Networks in the Capital Markets,** Wiley.

Tabla 2.3.3.

***Definición* (1)**	***Expresión* (1)**	***Definición* (2)**	***Expresión* (2)**
Mean Square error (*MSE*)	$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2$	Theil´s coefficient of inequality (*U*)	$U = \frac{\text{RMSE}}{\sqrt{\frac{1}{\text{N -1}}\sum_{i=!}^{N}(y_i - y_{i-1})^2}}$
Root Mean Square error (*RMSE*)	$\text{RMSE} = \sqrt{\text{MSE}}$	Akaike information criterion (*AIC*)	$AIC = \text{MSE}\left(\frac{\text{N} + \text{k}}{\text{N - k}}\right)$
Mean absolute error (*MAE*)	$\text{MAE} = \frac{1}{\text{N}}\sum_{i=1}^{N}\lvert y_i - \hat{y}_i\rvert$	Schwarz information criterion (*SIC*) ó Bayesian information criterion (*BIC*)	$SIC = BIC = \ln(MSE) + \frac{\ln(N)}{N}k$
Mean absolute percentage error (*MAPE*)	$\text{MAPE} = \frac{1}{\text{N}}\sum_{i=1}^{N}\left\lvert\frac{y_i - \hat{y}_i}{y_i}\right\rvert$	Predictive stochastic complexity (*PSC*)	$PSC = \frac{1}{N-k}\sum_{i=k+1}^{N}(y_i - \hat{y}_i)^2$
Coefficient of determinations (R^2)	$\text{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2};\ \bar{y} = 1/N\sum_{i=1}^{N} y_i$	Direction accuracy (*DA*)	$DA = \frac{1}{N}\sum_{i=1}^{N} a_i$ $a_i = \begin{cases} 1 & (y_{i+1} - y_i)(\hat{y}_{i+1} - y_i) > 0 \\ 0 & \text{otro caso} \end{cases}$
Pearson correlation (ρ)	$\rho = \frac{\sum_{i=1}^{N}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{N}(\hat{y}_i - \bar{\hat{y}})^2}}$	Confusion rate (*CR*)	$CR = 1 - DA$
Nota: "*N*" es el tamaño de la muestra; $(\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_N)$ son los valores ajustados; $(y_1, y_2, \cdots, y_N)$ son los valores muestrales; se ha mantenido las definiciones en inglés por su mayor difusión.			

Fuente: Min Qi. (1996). **Financial applications of artificial Neural Networks,** *Handbook of Statistics,* Vol. 14, pp. 540-541, (Edited by G.S. Maddala and C.R.Rao) Elsevier y elaboración propia.

Las anteriores medidas suelen estar acompañadas de otras que buscan comprobar la existencia de diferencias significativas entre el funcionamiento de los modelos alternativos, así se suelen utilizar los tests-t o el test de Diebold-Mariano[39] (Diebold y Mariano, 1995) que permiten comprobar la hipótesis nula de que no hay diferencia en los errores cuadrados de dos modelos alternativos. Desde la óptica de la independencia entre las direcciones de la predicción y el real, se puede comprobar mediante el test HM[40] (Henriksson y Merton, 1981; Pesaran y Timmerman,1994).

Tal y como comenta Min Qi (1996), es importante observar que el funcionamiento para la muestra de aprendizaje de cualquier modelo neuronal diseñado adecuadamente y

[39] Véase Diebold, F. ; Mariano, R. (1995). **Comparing Predictive Accuracy,** *Journal of Business & Economic Statistics*, 13(3), pp. 253-263.

[40] Referenciados en Min Qi. (1996). **Financial applications of artificial Neural Networks,** *Handbook of Statistics,* Vol. 14, pp. 540-541, (Edited by G.S. Maddala and C.R.Rao) Elsevier.

evaluado mediante algunas de las medidas reseñadas en la tabla 2.3.3. es normalmente mejor que sus homólogas en modelos estadísticos tradicionales. De hecho este aspecto, no nos debe sorprender dado el poder de aproximación universal de estos modelos y la necesidad de información a priori acerca del objetivo del problema que necesitan los modelos tradicionales.

Existe otro aspecto de gran trascendencia en el proceso especificación de un modelo neuronal, como evitar la predicción *espúrea* o el *sobreajuste*. Para ello, es importante realizar, una vez estimado el modelo neuronal, una evaluación de su capacidad de ajuste con una base de datos distinta del proceso de aprendizaje o estimación. La calidad del modelo dependerá principalmente de su funcionamiento con los datos no utilizados en el proceso de aprendizaje. Para aumentar la fiabilidad del propio modelo se suele realizar un proceso de *validación* durante la propia estimación iterativa de los parámetros del modelo, aumentando su robustez. El proceso que se suele seguir es dividir los datos que se poseen en tres muestras, *aprendizaje*, *test* y *predicción*. La base de datos de *aprendizaje* se utiliza para ajustar el modelo, la base de *test* para estimar el error de predicción del modelo seleccionado, y por último, la base de *predicción*, para gestionar el proceso de generalización[41].

Existen diferentes métodos para subdividir los datos, véase tabla 3.3.3. Algunos de ellos son más apropiados para datos de naturaleza temporal, como puede ser, escoger los primeros *n*-valores para *aprendizaje*, los siguientes *m*-valores para *test*, y el resto para *predicción* (número 3, tabla 3.3.3.).

Tabla 3.3.3.

	Métodos de extracción de datos
1	Un porcentaje de datos para *test*, *n*, otro para la base de datos de *predicción*, *m*, elegidos de forma aleatoria. El resto para *aprendizaje*.
2	Escoger individualmente los datos para *test* y de *predicción*, y el resto para *aprendizaje*.
3	Escoger los primeros *n*-valores para *aprendizaje*, los siguientes *m*-valores para *test*, y el resto para *predicción*.
4	Escoger los últimos *m*-valores para *predicción* y un porcentaje para *test* escogidos de forma aleatoria, el resto para *aprendizaje*.

[41] Una forma general es reservar un 50%, para el aprendizaje, un 25% para el test y un 25% para la predicción, siempre que la cantidad de datos que se posea lo permita.

Una vez dividida la base de datos, el proceso de generalización del modelo conlleva la utilización de métodos de aprendizaje comprobando su potencial sobre datos independientes, donde el proceso de validación varía en función de la complejidad del modelo[42]. Debemos por lo tanto separar claramente la labor de *selección* de los modelos, es decir, escoger el mejor entre diferentes posibilidades y la *gestión* del modelo, que supone para el modelo seleccionado, estimar el error de predicción en una base de datos de test. Podemos observar en la ilustración 3.3.3. como el error cometido en el aprendizaje tiende a cero conforme se incrementa la complejidad del mismo, que es directamente proporcional al número de parámetros. En este punto, el modelo generaliza poco por estar sobreparametrizado.

Para poder controlar el hecho de que el error en el proceso de aprendizaje no es un buen estimador del error cometido con los datos de la base de test, se suele interrumpir el proceso iterativo de estimación al observar que empieza la divergencia entre ambas curvas de aprendizaje, (véase línea vertical en la ilustración 3.3.3.).

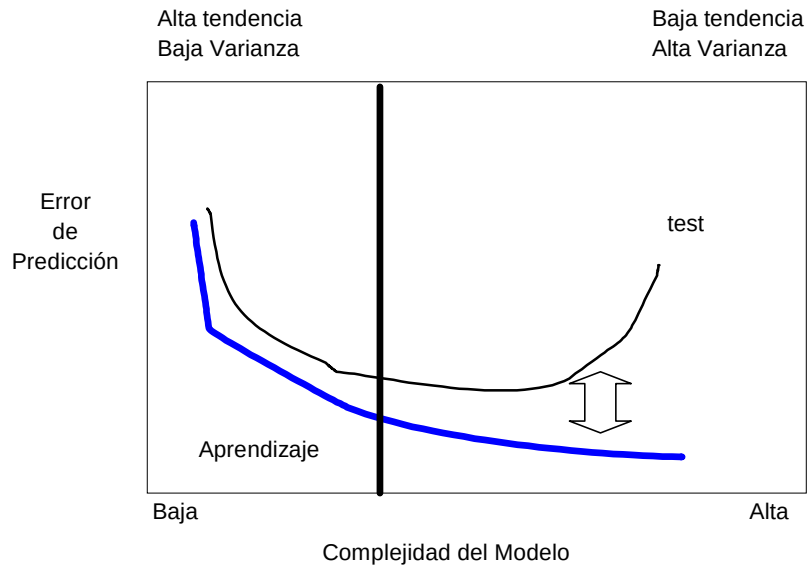


Ilustración 3.3.3.
Comportamiento del proceso de *Validación*

[42] Interesante la visión neuronal desde la óptica de la complejidad, Kárný, M.; Warwick, K.; Kůrková, V. (1998). **Dealing with Complexity. A Neural Networks Approach**, Springer.

Un aspecto importante en el aprendizaje del modelo es como tratar el binomio tendencia o "sesgo" y varianza del error de predicción, orientado a la selección de modelos (*Bias-variance Trade-off*)[43]. La idea proviene de la descomposición del propio error esperado de predicción. Es decir, si tenemos la siguiente expresión, $y = f(X) + \varepsilon$, donde la componente aleatoria se comporta como, $E(\varepsilon) = 0$ y $V(\varepsilon) = \sigma_\varepsilon^2$, se puede expresar el valor esperado del error de predicción en función del valor ajustado," $\hat{y}_0$ ", para el valor, " $X = x_0$ ", de la forma siguiente,

$$\begin{aligned} Error(x_0) &= E\left[\left(f(x_0) - \hat{y}_0\right)^2\right] \\ &= \sigma_\varepsilon^2 + \left[E(\hat{y}_0) - f(x_0)\right]^2 + E\left[\hat{y}_0 - E(\hat{y}_0)\right]^2 \\ &= \sigma_\varepsilon^2 + Tendencia^2(\hat{y}_0) + Varianza(\hat{y}_0) \end{aligned}$$

Dicha expresión está constituida por tres términos. El primero de ellos, la dispersión del error estocástico inherente al modelo," σ_ε^2 ", el cual no puede reducirse y cuyo valor sería nulo si el modelo fuese de naturaleza *determinista*. El segundo de ellos, contiene el error atribuible a la estimación de los parámetros del modelo, cuyo valor sería cero si el método de estimación escogido generase estimadores *insesgados*. En último lugar, la dispersión esperada entre el valor ajustado del modelo y su valor promedio, asociada a la fuente de error que proviene del valor incierto de las variables exógenas en el futuro de predicción. A su vez podemos descomponer la tendencia o "sesgo" al cuadrado en dos componentes,

$$\left[f(x_0) - E(\hat{y}_0)\right]^2 = \left[f(x_0) - E(y_0)\right]^2 + \left[E(y_0) - E(\hat{y}_0)\right]^2$$

El primero de ellos contiene el error entre la mejor aproximación posible y el verdadero valor de $f(x_0)$, que solo puede ser reducido ampliando el número de modelos disponibles[44]. En cambio el segundo componente consiste en el error entre la mejor aproximación posible y el mejor ajuste obtenido[45]. Para modelos de mayor grados de libertad la segunda componente puede ser positivo, permitiendo la posibilidad de reducir la varianza a costa del incremento de la tendencia o "sesgo", (véase ilustración 4.3.3.).

---

[43] Véase para mayor detalle el capítulo 7 de Hastie, T; Tibshirani, R.; Friedman, J. (2001). **The Elements of Statistical Learning. Data Mining, Inference, and Prediction**, Springer y Johnston, J. (1987). **Métodos de Econometría**, pp. 238-241,Vicens Universidad, Barcelona.

[44] Incluyendo las posibles interacciones y transformaciones sobre la variables del modelo.

[45] Cuyo valor es nulo si consideramos el caso de modelos lineales estimados por mínimos cuadrados ordinarios (MCO).

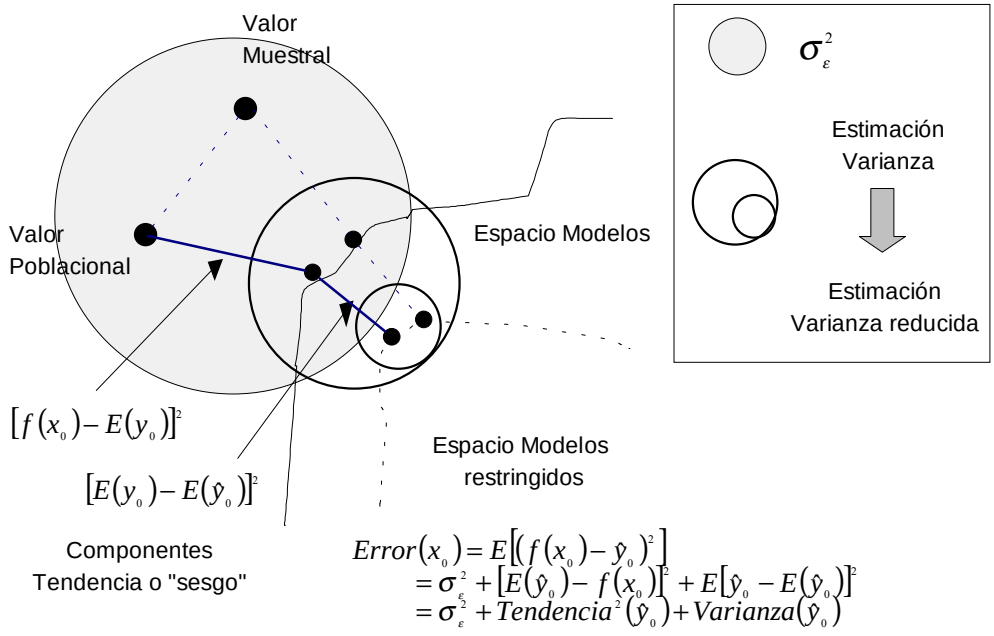


Ilustración 4.3.3.
Representación del comportamiento binomio: "*Bias-variance Tradeoff*" para el error de predicción.

Nota: El espacio de modelos contiene todas aquellas posibles predicciones para el modelo definido y el espacio de modelos restringidos son aquellos que poseen alguna modificación en términos de penalización.

Fuente: Hastie, T; Tibshirani, R.; Friedman, J. (2001). **The Elements of Statistical Learning. Data Mining, Inference, and Prediction**, Springer y elaboración propia.

Por lo todo lo anterior, el proceso de validación es una pieza clave para controlar la gestión del modelo a través del error de predicción. Inicialmente este proceso consiste en comprobar cada cierto número de iteraciones el nivel de error cometido en el proceso de estimación (datos de la base de aprendizaje) sobre los datos de la base de validación o *test*[46]. Si el error es menor que en la validación anterior, el proceso iterativo de estimación continúa, ya que la modificación dinámica de los parámetros permite aún una reducción del error global del modelo. En cambio, si el error no es menor, el proceso iterativo se detiene, en este momento se considera que el modelo ya ha generalizado lo suficiente.

[46] A este aspecto se le define como la *calibración* del modelo, por ejemplo, cada 200 iteraciones.

Pero no debemos confundir el proceso anterior, con el concepto de *validación cruzada* que supone una validación sobre el espacio de modelos entrenados, a diferencia de la anterior que lo es sobre "un" solo modelo. El procedimiento en sí es más robusto y consiste en generar *k*-submuestas de igual tamaño, de forma que, se obtiene el error de predicción del modelo seleccionado con los datos de las *k-1* muestras restantes. Este proceso se realiza varias veces intercambiando las submuestras. Al final se combinan las *k*-estimaciones de error de predicción obtenidas[47].

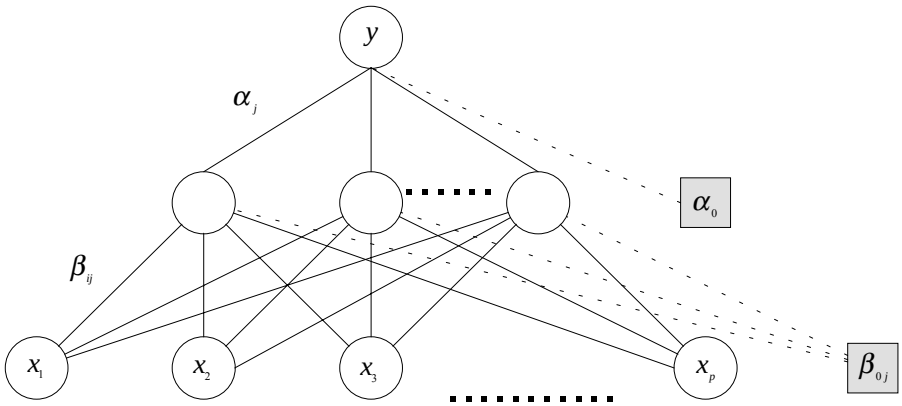
Finalmente, desde la óptica de la inferencia estadística existen muy pocos estudios empíricos sobre aplicaciones de modelos neuronales que establezcan intervalos de confianza o permitan realizar contrastes estadísticos. Debido sobre todo a que normalmente no se dispone de propiedades estadísticas clásicas. Sin embargo y siguiendo a Min Qi[48] (1996), si consideramos un modelo neuronal como un modelo de regresión no lineal, el estimador de "θ" tendrá las propiedades estadísticas de un estimador de mínimos cuadrados no lineales. Este aspecto permite establecer unas primeras pautas para realizar inferencia estadística en este entorno[49].

Se han propuesto varios métodos para interpretar la importancia relativa de cada *input* sobre cada *output*, es decir, su relevancia. El primero de ellos, son las *pseudoponderaciones* (PW) que supone una aproximación de lo que contribuye cada *input* a la explicación del output. En segundo lugar, el *sumatorio de los pesos* en valor absoluto de los *inputs* (SW). Las diferencias entre PW y SW son claras, SW pierde información sobre el efecto negativo de una variable *input* sobre el *output* por escoger valores absolutos. Si todas los pesos son positivos, PW y SW deberían finalizar con el mismo orden de rango de las diferentes variables explicativas, de todos modos, frente a fuertes no linealidades ambas medidas no son relevantes. El tercer método es el *análisis de sensibilidad*, el cual muestra la sensibilidad de

[47] Véase Hastie, T; Tibshirani, R.; Friedman, J. (2001). **The Elements of Statistical Learning. Data Mining, Inference, and Prediction,** Springer, pp. 214-224.
[48] Véase Min Qi. (1996). **Financial applications of artificial Neural Networks,** *Handbook of Statistics,* Vol. 14, pp. 540-541.
[49] Véase Kuan, C.; White, H. (1994). **Artificial neural networks: An econometric perspective,** *Econometric Rev*. 13, pp. 1-91. Existen otros esfuerzos basados en el método de *bootstrap* que permite determinar la calidad y fiabilidad de un modelo neuronal, aunque su coste computacional es alto, proporciona resultados más robustos.

los *outputs* frente a cambios en los *inputs*. Para realizar el análisis de sensibilidad, primero se determinan el mínimo, el máximo y el valor medio de cada *inputs*, dicho valor se modifica uno a uno manteniendo los valores de los otros *inputs* prefijados a su valor medio o mediano.

Finalmente, el *índice de sensibilidad* utiliza dicho índice para descubrir la fuerza relativa de la influencia de cada *input* sobre su *output*. El índice para una variable inputs se calcula obteniendo el porcentaje de los cambios del *output* para un determinado número de cambios en los intervalos definidos iguales para la variable *inputs* estudiada, por lo tanto, proporciona una medida de "significación" de los *inputs* al predecir el *output* deseado, (véase ilustración 5.3.3.).



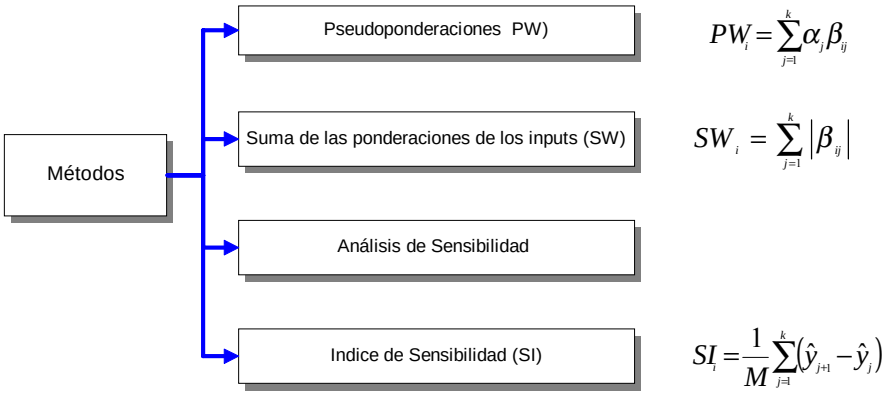


Ilustración 5.3.3.
Métodos para la interpretación relativa de la importancia de cada *inputs* respecto al *output*.

3.4. *Aproximación* econométrica de los modelos neuronales.

3.4.1. Introducción.

Los modelos neuronales son en esencia un mecanismo de inferencia estadística no paramétrica inspirado en los sistemas biológicos, así ciertos algoritmos utilizados en el proceso de aprendizaje se han demostrado que están muy cerca de la especificación de modelos de regresión no lineales[50]. Modelos como por ejemplo, las redes *feed-forward* poseen la capacidad de ser utilizadas como *aproximadores universales* de funciones[51].

Desde la óptica econométrica, a los modelos neuronales se les puede asociar el planteamiento clásico siguiente. Existe una función desconocida a priori, $f(x)$ con cierto componente estocástico[52], donde el proceso de aprendizaje del modelo consiste en calcular un estimador[53] de la función desconocida, $f(x;\omega) \equiv \hat{f}(x)$, siendo "*w*" el vector de parámetros desconocidos[54] y "x" el conjunto de datos observados. El modelo neuronal definido es por lo tanto un estimador no paramétrico de la esperanza matemática de "*x*" condicionada a "*y*", es decir, $E(y|x)$.

Existen muchos otros métodos para capturar aspectos no lineales en la literatura estadística (véase tabla 1.3.4.1.), pero son los propios modelos neuronales quienes dotan a

[50] Véase White, H. (1989). **Learning in artificial neural networks: a statistical perspective**, *Neural Computation*, 1, pp. 425-464.
[51] Comentado en el apartado 2.2.
[52] Existen incluso aproximaciones entre la modelización de ecuaciones simultáneas y la tecnología neuronal, véase Caporaletti, L.E.; Dorsey, R.E.; Johnson, J.D. y Powell, W.A. (1994). **A decision support system for in-sample simultaneous equation systems forecasting using artificial neural systems**, *Decision Support Systems*, 11, pp. 481-495 o aplicaciones en el ámbito macroeconómico, Min Qi. (2001). **Predicting US recessions with leading indicators via neural networks models**, *International Journal of Forecasting*, 17, pp. 383-401.
[53] Los algoritmos de aprendizaje que permiten obtener los parámetros estimados del modelo, pueden dividirse en dos categorías, de primer orden, con todas las variedades de gradiente descendente, (en el caso de los modelos neuronales *feed-forward*, se define como *error backpropagation*) y de segundo orden, que incluyen entre otros, el método de *cuasi-Newton*, *Broyden-Fletcher-Golfarb-Shanno* (BFGS) y los métodos de *gradiente conjugado*. Véase para mayor detalle, Shepherd, A.J. (1997). **Second-Order Methods for Neural Networks, Fast and Reliable Training Methods for Multi-layer Perceptrons**, Springer.
[54] Los cuales son estimados desde los datos de la muestra observada.

este campo de un elegante formalismo que permite unificar todos los anteriores paradigmas[55], en uno solo, véase Zapranis y Refenes (1999).

Tabla 1.3.4.1.

✓	Polynomial Regression	Eubank, R.L.(1999)
✓	Fourier series regression	Eubank, R.L. (1999) Haerdle W. (1990)
✓	Wavelet smoothing	Donoho, D.L. y Johnstone, I.M. (1995) Donoho. D.L., Johnstone I.M., Kerkyacharian, G. y Picard, D. (1995)
✓	K-nearest neighbor regression	Haerdle, W. (1990) Hand, D.J.(1981,1997) Ripley, B.D. (1996)
✓	Kernel regression	Eubank, R.L.(1999), Haerdle, W. (1990) Hand, D.J. (1981,1982,1997), Ripley, B.D. (1996)
✓	Local polynomial smoothing	Eubank, R.L. (1999), Wand, M.P. y Jones, M.C. (1995) Fan, J. and Gijbels, I. (1995)
✓	B-Splines	Eubank, R.L. (1999)
✓	Tree-based models (CART, AID, etc.)	Haerdle, W. (1990); Lim, T.-S., Loh, W.-Y, Shih, Y.-S (1997) Hand, D.J. (1997), Ripley, B.D. (1996)
✓	Multivariate adaptive regression splines (MARS)	Friedman , J.H.(1991)
✓	Proyection pursuit	Friedman, J.H. y Stuetzle, W. (1981) Haerdle, W. (1990) Ripley, B.D. (1996)
✓	Bayesian Methods	Dey, D. (1998)
✓	GMDH	Farlow, S.J. (1984)
✓	Smoothing *splines*	Eubank, R.L. (1999), Wahba, G. (1990), Green, P.J. y Silverman, B.W. (1994) Haerdle, W. (1990)

Fuente: ftp://ftp.sas.com/pub/neural/FAQ.html y elaboración propia.

En la misma línea econométrica anterior es posible que una vez seleccionado un modelo, no sea necesariamente una fiel representación de la esencia de la función desconocida, $f(x)$, donde algunas de las posibles causas de esta divergencia pueden ser las siguientes: omisión de variables relevantes; inclusión de variables irrelevantes; forma funcional incorrecta; errores en la medida de las variables dependientes; especificación incorrecta del término de error del modelo; algoritmo de aprendizaje inadecuado (por ejemplo, problemas de *convergencia*) y métodos ineficaces de selección de modelos (por ejemplo, *overfitting*[56]).

[55] Véase un esfuerzo en unificar los métodos de proyección no lineal en, Bakshi, B.R. Utojo, U. (1998). **Unification of neural and statistical modeling methods that combine inputs by linear projection**, *Computers Chem. Engng*, Vol. 22, No. 12, pp. 1859-1878 y con carácter más general, Creedy, J.; Martín, V.L. (1997). **Nonlinear Economic Models. Cross-sectional, Time Series and Neural Network Applications**, Edward Elgar.

[56] El sobreaprendizaje o *overfitting*, supone que el modelo pierda la capacidad de generalizar y la sustituye por su capacidad de ajustar lo que es posible que suponga una reducción en los resultados de predicción.

Frente a los problemas anteriores los métodos estadísticos clásicos poseen un conjunto de herramientas muy potentes para el proceso de especificación y de diagnóstico. En cambio la metodología neuronal no posee, de momento, mecanismos de control sobre el proceso de aprendizaje suficientemente contrastados, si bien se están realizando esfuerzos muy importantes. A continuación se desarrollarán aquellos aspectos econométricos más relevantes de los modelos neuronales.

El proceso de identificación de un modelo incluye dos etapas, la primera, la comprobación de la *bondad* del mismo y la segunda, el establecimiento de *tests* que permitan diagnosticar la significación de las variables explicativas. La primera de ellas, supone valorar si es o no una fiel representación de, $E(y|x)$ a través del estudio de los residuos. La segunda etapa consiste en satisfacer un conjunto de tests. Este aspecto es una condición necesaria pero no suficiente para considerar como óptimo el modelo propuesto. Cuando se trata de valorar la importancia de las variables independientes, debe tenerse en cuenta los siguientes aspectos,

- Qué se entiende por una variable relevante en el modelo,
- Establecer una medida de la variabilidad en el muestreo de los estimadores,
- Diseño de contrastes de hipótesis para detectar variables irrelevantes.

Todos los anteriores aspectos en el campo neuronal se complican. Por ejemplo, la derivada parcial de "*y*" respecto de "*x*" no es constante, la distribución de los estimadores es desconocida y pueden tomar más de un valor los estimadores, etc. Frente a este vacío formal existen aportaciones de autores que poco a poco van configurando un conjunto de mecanismos que permitan a los modelos neuronales estar a la altura de los modelos econométricos tradicionales. Así, por ejemplo, para el primer punto, existen varias propuestas de autores sobre métodos de selección de variables relevantes. En primer lugar, tenemos a Zapranis y Refenes (1999), véase tabla 2.3.4.1., en la línea de los comentados en el apartado 3.3. (Min Qi (1996)). Y en segundo lugar, Bishop (1995), con la definición de mecanismos de búsqueda secuencial, (véase tabla 3.3.4.1.).

Tabla 2.3.4.1.

✓	Criterio de elasticidad media
✓	Criterio de la máxima sensibilidad
✓	Criterio de sensibilidad del modelo ajustado

Tabla 3.3.4.1.

✓	Criterio secuencial *"forward"*
✓	Criterio secuencial *"backward"*
✓	Criterio " *branch* & *bound*"

La estimación de la variabilidad en el muestreo de los estimadores es necesaria para justificar, mediante la construcción de tests, la importancia o relevancia de las variables. En el ámbito no paramétrico este aspecto se complica en demasía, de forma que, es habitual introducir la simulación estocástica. Tal aspecto genera complejidad tanto en su control, como en la gestión sobre un número elevado de simulaciones. Uno de los métodos que en la actualidad posee más seguidores es el *Bootstrapping*[57] que permite abordar los posibles efectos de los mínimos locales en la fiabilidad de la estimación de los parámetros[58], de todos modos autores como Zapranis y Refenes (1999) recomiendan como mejor método desde la óptica de eficiencia, el muestreo estocástico desde la distribución conjunta multivariante teórica de los parámetros del modelo neuronal, que bajo ciertas hipótesis relajadas, es *Gaussiana* y donde la matriz de covarianzas puede ser calculada analíticamente.

Respecto al último punto, es decir, la elaboración de contrastes de hipótesis de variables irrelevantes, los estudios realizados mediante técnicas de *bootstrap* no garantizan buenos resultados, tanto en términos de convergencia como de indiferencia frente a condiciones iniciales. Zapranis y Refenes (1999) proponen como alternativa trabajar con las distribuciones empíricas, utilizando por ejemplo los percentiles.

La selección adecuada del modelo puede ser asimilable al *trade-off*[59] existente entre la tendencia o "sesgo" y la varianza sobre la diferencia al cuadrado entre $f(x;\omega)$ y $\hat{f}(x)$. Si se cae en una subparametrización del modelo neuronal dominaría la tendencia o sesgo, lo cual ocasionaría, en primer lugar, que la respuesta del modelo fuera diferente desde $\hat{f}(x)$ y en segundo lugar, el estimador fuese sesgado, en cambio si sobreparametrizamos el modelo, el estimador puede ser insesgado pero se vuelve muy sensible a los datos, por el incremento de la varianza.

[57] Para más información, Efron, B and Tibshirani, R. J. (1993). **An introduction to the Bootstrap, Monographs on Statistics and Applied Probability**, Chapman & Hall. Referenciado en Zapranis A. y Refenes, A. P. (1999). **Principles of Neural Model. Identification, Selection and Adequacy**, Springer y apartado 3.4.2.

[58] Existen diferentes métodos para aplicar *bootstrap*, uno de ellos es inicializar el proceso seleccionando de forma aleatoria los parámetros de entre un rango predefinido.

[59] Véase apartado 3.3.

En último lugar, los procedimientos que hasta la fecha se han utilizado para la selección de modelos quedan reflejados en la tabla 4.3.4.1, algunos de los cuales serán desarrollados en el apartado siguiente.

Tabla 4.3.4.1. Procedimientos de *selección* para modelos neuronales

✓	Métodos de regularización	Weigend et al. (1991) Wahba et al (1994)
✓	Métodos de modificación de topologías	Ash (1989) Fahlman and Lebiere (1990) Refenes and Mitrelias (1993)
✓	Principio de duración de la descripción mínima (MDL)	Rissanen (1989) Zemel (1993)
✓	Principio de minimización del riesgo estructural (SRM)	Vapnik and Chervonenkis (1971)
✓	Estimación algebraica	Moody (1992) Amari (1995)
✓	Métodos de muestreo y de validación cruzada	Stone (1974) Geisser (1975) Wahba and Wold (1975)
✓	*Bootstrap*	Efron (1981)
✓	Principio de riesgo mínimo de predicción (MPR)	Stone (1977) Eubank (1988) Murata et al (1991, 1993) Moody (1992) Amari (1995)

Fuente: Zapranis A. y Refenes, A.P. (1999). **Principles of Neural Model. Identification, Selection and Adequacy**, Springer y elaboración propia.

3.4.2. Diseño de Modelos Econométricos Neuronales: *principios de identificación*.

El proceso de identificación de los modelos neuronales[60] debe seguir una estrategia *modular* constituida por tres componentes: la selección del modelo, el contraste de significación de la variables y el contraste de significación del modelo. En este ámbito, no debemos confundir términos como *estimación* (cálculo de los parámetros del modelo), con la *especificación del modelo* (identificar la forma funcional más apropiada)[61], con la propia *selección* del modelo que, desde una óptica estadística, supone la estimación y estudio del término de error del modelo.

[60] Este este apartado está inspirado en Zapranis A. y Refenes, A.P. (1999). **Principles of Neural Model Identification Selection and Adequacy**, Springer.

[61] Importante la síntesis elaborada en el entorno de datos de series temporales, por Min Qi, Zhang, G.P. (2001). **Theory and Methodology. An investigation of model selection criteria for neural network time series forecasting**, *European Journal of Operational Reserach*, 132, pp. 666-680.

La ilustración 1.3.4.2. muestra con mayor detalle las principales áreas del proceso de identificación de modelos neuronales, que coinciden con las utilizadas en el ámbito econométrico tradicional,

- Selección del modelo: escoger la forma funcional más idónea, estimar los parámetros y establecer aquellos criterios de ajuste para la evaluación del propio modelo.
- Contraste de la significación de las variables: medida de la relevancia de cada una de las variables del modelo, estimación de la varianza de la anterior medida y contraste de hipótesis para detectar la irrelevancia de ciertas variables.
- Contraste de relevancia del modelo en su conjunto.

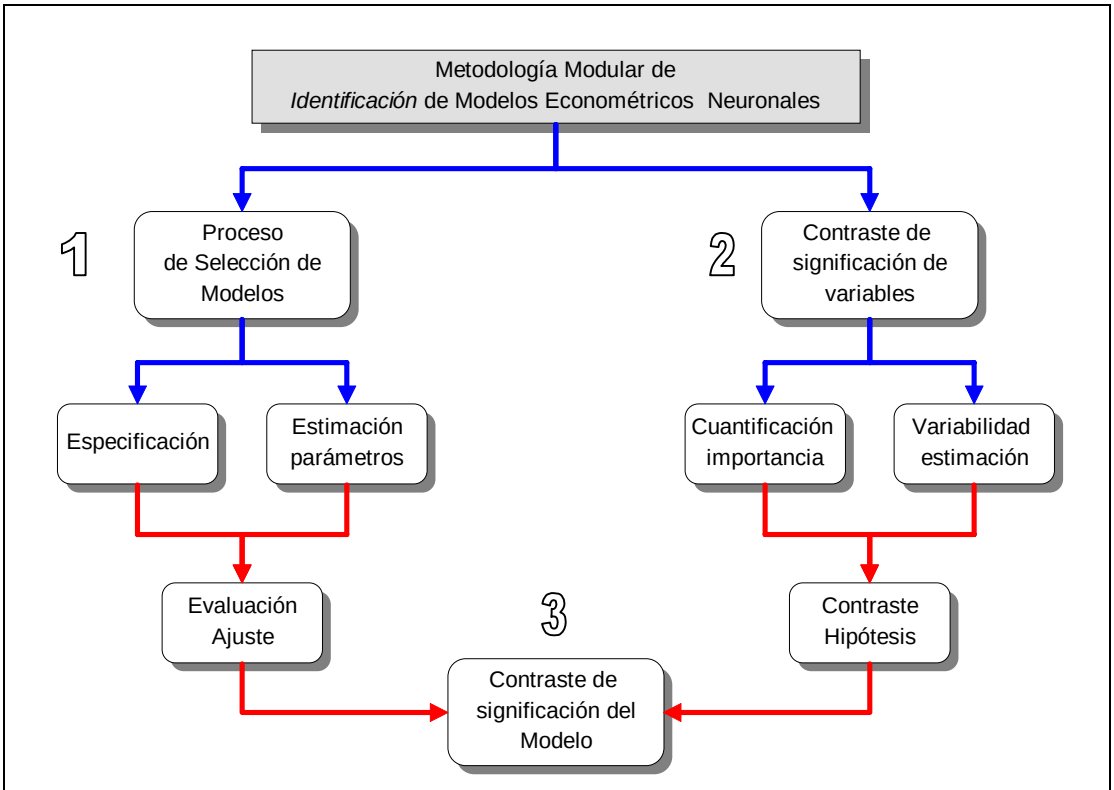


Ilustración 1 3.4.2.
Esquema *Modular* de identificación de modelos neuronales

En este proceso de selección, la <u>especificación</u> consiste en definir los componentes del propio modelo, es decir,

p : número de *inputs*,
A_q : arquitectura o topología del modelo para "q" neuronas en la capa oculta,
$w = \{\beta; \phi\}$: pesos o conexiones del modelo,
m: número de parámetros del modelo[62], $m = h(A_q)$,
S_q : conjunto de modelos neuronales, especificados como,

$$S_q \equiv \{g_q(x; w), \quad x \in R^m, w \in W\}; \; W \subseteq R^p .$$

Una especificación más general podría ser, (véase mayor desarrollo en el apartado 3.4.3.),

$$\hat{y} = g_q(x; w) = \beta_0 + \sum_{j=1}^{q} \beta_j f\left(\sum_{i=1}^{p} \phi_{ij} x_i + \phi_{0j} \right)$$

siendo,

f : función de *transferencia* o *activación*, que puede ser lineal o no lineal, pero es continua y derivable en la mayoría de los casos[63],
ϕ_{ij} : peso correspondiente a la conexión entre el i-ésimo *input* con la j-ésima neurona oculta,
ϕ_{0j} : término independiente o "*bias*" que le corresponde la j-ésima neurona oculta,
β_j : peso correspondiente a la conexión entre la j-ésima neurona oculta y el *output*,
β_0 : término independiente del *output*.

La práctica común de la selección de modelos neuronales, tal y como comenta Refenes (1999), posee dos líneas de actuación, la primera de ellas, supone escoger el modelo más sencillo que sea <u>consistente con los datos</u>[64] y la segunda, descansa en la definición de un <u>criterio de ajuste</u>. Cuando dicho criterio de ajuste es la capacidad esperada del estimador para

[62] La dimensión del modelo en número de parámetros se puede calcular mediante la relación, $m = (p+2)q+1$, para una arquitectura, *single-hidden layer* y con una sola neurona en la capa de salida.
[63] En este caso se considera como función de *salida* el caso lineal.
[64] Un ejemplo en el campo paramétrico del modelo de regresión es el procedimiento de selección "*stepwise*" de variables relevantes.

predecir nuevas observaciones, se define como *riesgo de predicción* y puede ser estimado mediante el método *algebraico* o por métodos de *remuestreo*[65].

El estimador $g_q(x;w)$ es único y está adscrito, a una arquitectura neuronal específica,"A_q", a un vector de parámetros,"w", y a un criterio de ajuste, $r(z;w)$. Su expresión suele ser en forma de función de pérdida o error,

$$L(w)=\frac{1}{n}\sum_{i=1}^{n} r(z;w),$$

siendo "z", la combinación de valores poblacionales entre *inputs* y *output*, $\{x, y\}$, asociada a los parámetros,"w". Una vez observada una muestra de ambos, podemos estimar la función empírica de pérdida o error a través de ellos, cuya expresión es,

$$L_n(w)=\frac{1}{n}\sum_{i=1}^{n} r(z_i;w).$$

En este caso, "z", proviene de la base de aprendizaje, $\{x_i, y_i\}_{i=1}^{n}$. Ambas funciones convergen conforme se incrementa el tamaño de los datos utilizados para el proceso de aprendizaje. Encontrar la solución al planteamiento anterior, supone minimizar la siguiente expresión,

$$\hat{w}_n = \arg\min\{L_n(w): w\in W\};\ W\subseteq \Re^p,$$

donde"$\hat{w}_n$" es el estimador de discrepancia mínima de "w_0". Dicha discrepancia consiste en la diferencia entre la mejor aproximación al modelo $g_q(x;w_0)$ por parte de $g_q(x;\hat{w}_n)$.

Si utilizamos la similitud con el modelo de regresión clásico, en este contexto, la función de pérdida se definiría como,

$$L_n(w)=\frac{1}{2n}\sum_{i=1}^{n}\left[y_i - g(x_i;w)\right]^2$$

y la solución supone obtener "$\hat{w}_n$"[66].

[65] Véase para mayor detalle, Moody, J.M ; Utans, J. **Architecture Selection Strategies for Neural Networks: Application to Corporate Bond Rating Prediction**, referenciado en Refenes, A. P. (1995). **Neural Networks in the Capital Markets**, pp. 276-290, Wiley.

[66] En este caso, su obtención supone resolver el sistema de ecuaciones normales, que se derivan de igualar a cero las derivadas de la función de pérdida respecto a los parámetros, de difícil resolución para los casos no lineales. Por esta razón se suele recurrir a los algoritmos iterativos.

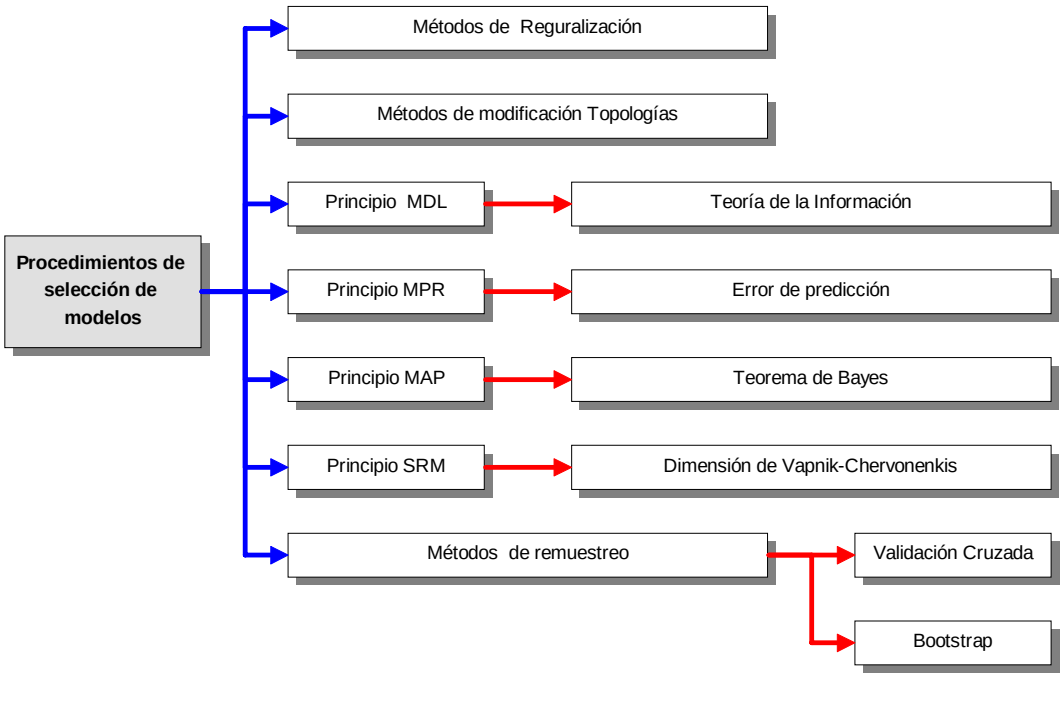


Ilustración 2.3.4.2.
Procedimientos de *selección* de modelos neuronales

Nota: MDL: *Minimum Description Length Principle* ; MPR: *Minimum Prediction Risk Principle*
MAP: *Maximum A Posteriori Probability Principle*; SRM: *Structural Risk Minimization Principle*

Existen diferentes métodos para la selección de modelos. El primero de ellos, son los métodos de regularización que incorporan elementos de penalización sobre la complejidad de los modelos. En segundo lugar existen los algoritmos que modifican la topología y por lo tanto la especificación del modelo. En tercer lugar, el método que utiliza el principio *Minimum Description Length Principle* (MDL) muy vinculado a la teoría de la información. En cuarto lugar, el principio de minimización del *riesgo estructural*[67] (SRM) vinculado a la teoría de *learning machine* y la dimensión de *Vapnik-Chernovenkis*[68]. Desde una óptica Bayesiana y en quinto lugar, tenemos el método *Maximum a Posteriori Probability Principle* (MAP), que permite comprobar la verosimilitud del modelo a partir de las probabilidades a posteriori obtenidas mediante distribuciones a priori de los parámetros del modelo.

[67] Dicho principio ha permitido el diseño de un nuevo modelo neuronal, el modelo *support vector machine* (SVM), con mucho potencial en el ámbito de la predicción de series temporales. Véase Francis E.H. Tay; Lijuan Cao. (2001). **Application of support vector machines in financial time series forecasting**, *Omega*, 29, pp. 309-317.

[68] Ésta dimensión modela la capacidad del modelo para realizar particiones en el espacio.

En sexto lugar, tenemos los métodos de remuestreo, mediante técnicas de *Bootstrap* o de *validación cruzada.* En último lugar, el principio de riesgo de predicción, *Minimum Prediction Risk Principle* (MPR) que supone el cálculo del error cometido en la base de datos de test, (véase la ilustración 2.3.4.2.).

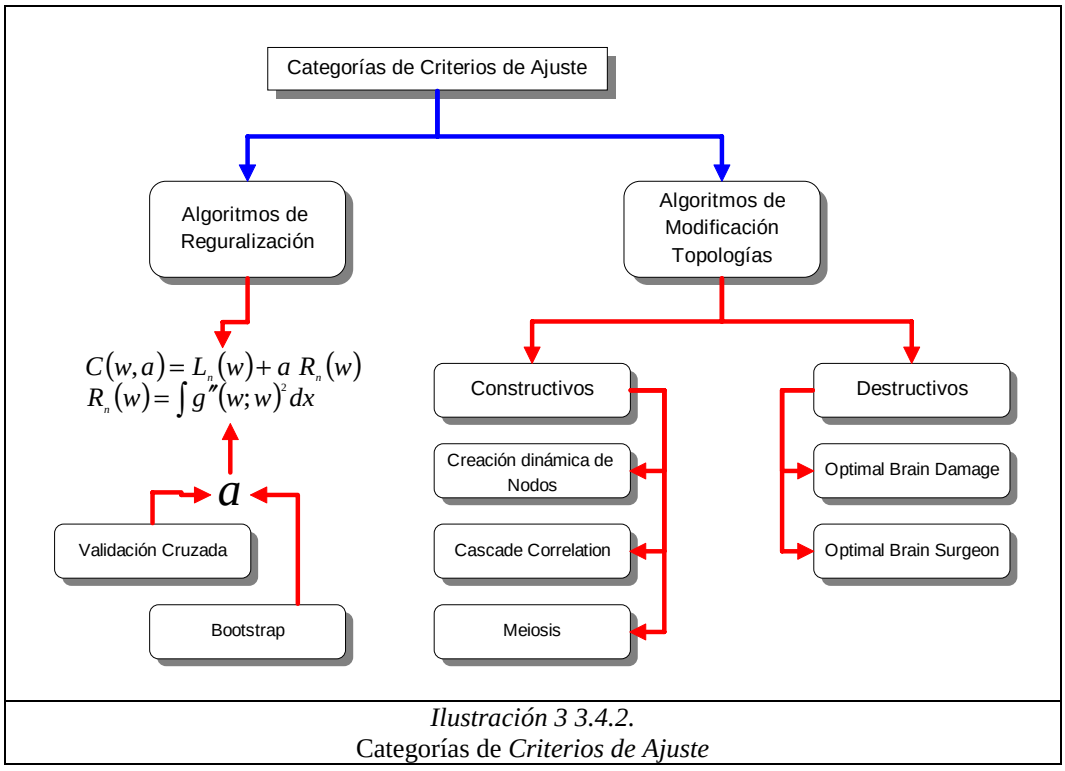


Ilustración 3 3.4.2.
Categorías de *Criterios de Ajuste*

La ilustración 3.3.4.2. muestra las dos categorías existentes de criterios de ajuste, los algoritmos de regularización[69] y los algoritmos de modificación de topologías. Los primeros incorporan un elemento que penaliza la complejidad del modelo, $R_n(w)$, cuya expresión es,

$$C(w,a)=L_n(w)+a\,R_n(w)$$
$$R_n(w)=\int g''(x;w)^2\,dx$$

donde el parámetro, "a", representa el compromiso o *trade-off* entre el error cometido y la calidad del ajuste del mismo. Su valor es obtenido de forma secuencial mediante técnicas de *validación cruzada* o de *bootstrap*[70].

[69] Véase para mayor detalle, Orr, G.B.; Müller, K-R. (1998). **Neural Networks: Tricks of the Trade**, pp. 51-139, Springer.
[70] Véase Hastie, T; Tibshirani, R.; Friedman, J. (2001). **The Elements of Statistical Learning. Data Mining, Inference, and Prediction**, pp. 216-217 y 217-221, Springer.

Los algoritmos que utilizan la modificación de las topologías pueden dividirse entre aquellos que incrementan la complejidad y aquellos que realizan el proceso inverso, es decir, reducir la dimensión del modelo. Para el primer grupo, existen tres categorías, Creación Dinámica de Nodos, Cascade Correlation y Meiosis y para el segundo, dos categorías: Optimal Brain Damage y Optimal Brain Surgeon.

A continuación realizaremos un breve comentario para cada uno de ellos.

En primer lugar, el método de *creación dinámica de nodos* supone un mecanismo que permite el incremento del número de neuronas comprobando la necesidad de aumentar o no la estructura del modelo neuronal[71] de forma secuencial. En segundo lugar, está el algoritmo *cascade correlation*[72](CCA) (Fahlman; Lebiere (1990)) que posee conexiones directas entre *inputs* y *outputs*. A igual que el algoritmo anterior, éste incluye de forma secuencial nuevas conexiones pero con una diferencia clara, el algoritmo propuesto por Fahlman y Lebiere escoge el nuevo nodo a incluir de un grupo de candidatos que maximizan la correlación entre el *output* del modelo y la función de pérdida o error, $L_n(w)$. En último lugar, tenemos *meiosis networks* (Hanson, 1990), topología que utiliza pesos estocásticos, generados a través de una distribución de tipo gaussiano, $N(\mu_{w_{ij}};\sigma_{w_{ij}})$.

El aprendizaje o proceso de estimación se genera modificando los parámetros de la distribución normal para cada uno de los pesos, de forma que se calcula en cada nodo de la capa oculta el coeficiente de variación. Si éste es mayor que un umbral predefinido, dicho nodo se divide en dos y a estos pesos se les asigna la mitad de la variancia, centrados en la misma media. La creación de nodos se detiene cuando la función de pérdida se acerca cada vez menos al umbral predefinido.

---

[71] Las aplicaciones son numerosas en disciplinas ajenas a la Economía, como por ejemplo en el campo de la geofísica, véase Huang, Z.; Williamson, Mark A. (1996). **Artificial neural network modelling as an aid to source rock characterization**, *Marine and Petroleum Geology*, Vol. 13, No. 2, pp. 277-290, donde presenta una simbiosis del algoritmo "quickprop" con la creación dinámica de nodos para conseguir una mayor eficiencia en el aprendizaje.

[72] Véase en el ámbito de aplicaciones, Spoerre, J.K. (1997). **Application of the cascade correlation algorithm (CCA) to bearing fault classification problems**, *Computers in industry*, 32, pp. 295-304; Lacher, R.C.; Cotas, P.K.; Sharma, S.C.; Fant, L. F. (1995). **A neural network for classifying the financial health of a firm**, *European Journal of Operational Research*, 85, pp. 53-65 y desde la óptica algorítmica, Prechelt, L. (1997) **Investigation of the CasCor Family of Learning Algorithms**, *Neural Networks*, Vol. 10, No. 5, pp. 885-896.

Para el grupo de algoritmos que modifican topologías pero desde la óptica destructiva, es decir, reduciendo la dimensión del modelo de forma iterativa, tenemos en primer lugar, el método *optimal brain damage* (OBD) que permite un control sobre la complejidad del modelo neuronal, calculando el coste que supone en términos de la función de pérdida la supresión de un peso o parámetro[73]. El método permite calcular la importancia del peso suprimido sobre el error en el aprendizaje y se aproxima a su valor utilizando los elementos de la diagonal *Hessiana* de $L_n(w)$. La expresión formal de dicha importancia, "s_i", en ciertas condiciones, es,

$$s_i = \left(R_{ii} + \frac{1}{2} H_{ii} \right) w_i^2$$

siendo,

R_{ii}: los elementos de la diagonal de la matriz "R", que recoge las segundas derivadas del término de regularización [74],

H_{ii}: matriz *Hessiana*,

w_i: peso que es suprimido.

El segundo método destructivo es, *Optimal Brain Surgeon* (OBS), cuya idea básica es la siguiente: supone un procedimiento para estimar el incremento del error en el proceso de aprendizaje cuando se suprime un peso o conexión utilizando la información de las derivadas de segundo orden de la superficie de error[75]. La ventaja que posee frente al anterior método es que no solo suprime los pesos sino que además vuelve a estimar todos los demás en la nueva situación. Para ello necesita del cálculo de la inversa de la matriz Hessiana de $L_n(w)$[76].

---

[73] Véase, Tautvydas Cibas, Françoise Fogelman Soulié, Patrick Gallinari, Sarunas Raudys. (1996).**Variable selection with neural networks**, *Neurocomputing*, 12, pp. 223-248. Desde la óptica de su aplicación para problemas de clasificación, véase, Mads Hintz-Madsen, Lars Kai hansen, Jan Larsen, Morten UIT Pedersen, Michael Larsen. (1998). **Neural classifier construction using reguralization, pruning and test error estimation**, *Neural Networks*, 11, pp. 1659-1670.

[74] Permite eliminar el *overfitting* y asegura la estabilidad numérica de la solución, penalizando la curvatura de la función de verosimilitud

[75] La aproximación local a la función de coste se gestiona mediante la expansión de series de *Taylor*, ya que el grado controla la "suavidad".

[76] Véase Chandrasekaran, H.; Chen, Hung-Han; Manry, Michel T. (2000). **Pruning of basis functions in nonlinear approximators**, *Neurocomputing*, 34, pp. 29-53. El método OBS se puede utilizar para mejorar los resultados de técnicas multivariantes, como por ejemplo, regresión de componentes principales (PCR) frente a relaciones no lineales, véase, Poppi, R.J. Massart, D.L. (1998). **The optimal brain surgeon for pruning neural network architecture applied to multivariate calibration**, *Analítica Chimica Acta*, 375, pp. 187-195.

Una vez detallado los métodos de regularización y aquellos que modifican las topologías, vamos a detallar los restantes. En primer lugar, tenemos el método del principio SRM, que consiste en minimizar de forma simultánea la función de pérdida y la dimensión de *Vapnik-Chervonenkis* (VC) como medida de la complejidad del modelo[77]. Para ello se adiciona el término, $\Omega(n/h_k)$, que posee con una forma funcional complicada, pero permite la expresión en términos de intervalo de confianza,

$$L(w_n^k) \leq L_n(w_n^k) + \Omega\left(\frac{n}{h_k}\right)$$

En segundo lugar, el principio de MDL, que proviene de la teoría de la información[78] y consiste en buscar la descripción más simple de los datos. En tercer lugar, el principio MAP, dicho procedimiento consiste en seleccionar el mejor modelo a partir de la probabilidad *bayesiana* del mismo,

$$P(M_i|D) \propto P(D|M_i)P(M_i)$$

donde, "M", es el modelo óptimo y "D", los datos[79]. Finalmente, en el contexto del principio MPR, tenemos el error de predicción.

Fue Moody (1992) el primero de introducir el concepto de *Error de Predicción Generalizado* (GPE), el cual posee claras similitudes con las expresiones homólogas para modelos lineales, como el criterio de Akaike[80].

---

[77] Véase Hastie, T; Tibshirani, R.; Friedman, J. (2001). **The Elements of Statistical Learning. Data Mining, Inference, and Prediction**, pp. 210-213, Springer. Donde se especifica que el principio SRM es mucho más efectivo que los estadísticos tradicionales, AIC y BIC para la selección de modelos.

[78] Es equivalente a minimizar la longitud de un mensaje codificado, de ahí, su utilidad en el ámbito de los protocolos de comunicación.

[79] Véase aplicaciones del principio MDL como soporte de otras técnicas, Thomas C.M. Lee. (2000). **Regression spline smoothing using the minimum length principle**, *Statistics & Probability Letters*, 48, pp. 71-82 y Leonardis, Aleš; Bischof, Horst. (1998). **An efficient MDL-based construction of RBF networks**, *Neural Networks*, 11, pp. 963-973.

[80] *Akaike* en 1973 propuso el criterio AIC, derivado de la teoría de la información, según el cual dado un conjunto de modelos de regresión con distinto número de parámetros," p", elegiremos aquel que tenga un valor mínimo de, $AIC = n\ln\hat{\sigma}^2 + n + 2p$, siendo "$\hat{\sigma}^2$"el estimador máximo verosímil y "p" el número de parámetros.

La expresión formal del *Error de Predicción Generalizado* es,

$$GPE(q) = ASE(q) + \frac{2}{n} tr\mathrm{VG}(q)$$

donde,

q: número de neuronas en la capa oculta,

V: matriz de varianzas-covarianzas de los datos observados,

$ASE(q)$: promedio de los errores al cuadrado cometidos en la fase de aprendizaje,

n: tamaño muestral,

$G(q)$: matriz de influencia generalizada[81],

y en la práctica la matriz "V" y $G(q)$ son desconocidas y deben ser estimadas.

En síntesis, Zapranis y Refenes (1999) proponen como mejor método de selección de modelos neuronales el principio de mínimo riesgo de predicción (MPR) ya comentado. Sustituyen la forma analítica por una forma algebraica del cálculo del error de predicción, a través de un proceso de incremento sucesivo de la topología, muy parecido a los mecanismos de *stepwise* de la modelización clásica de regresión.

Para poder implementar dicho principio se establece un proceso de siete fases, donde el mecanismo de selección de modelos neuronales se establece por incremento iterativo del número de nodos en la capa oculta, "q" y en base al principio de mínimo riesgo de predicción (MPR), (véase ilustración 4.3.4.2.)

[81] La *traza* de esta matriz, simboliza como $tr(\cdot)$, puede ser aproximada mediante el número total de parámetros del modelo neuronal, "*m*". Véase Moody, J.M; Utans, J. **Architecture Selection Strategies for Neural Neetworks: Application to Corporate Bond Rating Prediction**, referenciado en Refenes, A. P. (1995). **Neural Networks in the Capital Markets**, pp. 276-290, Wiley.

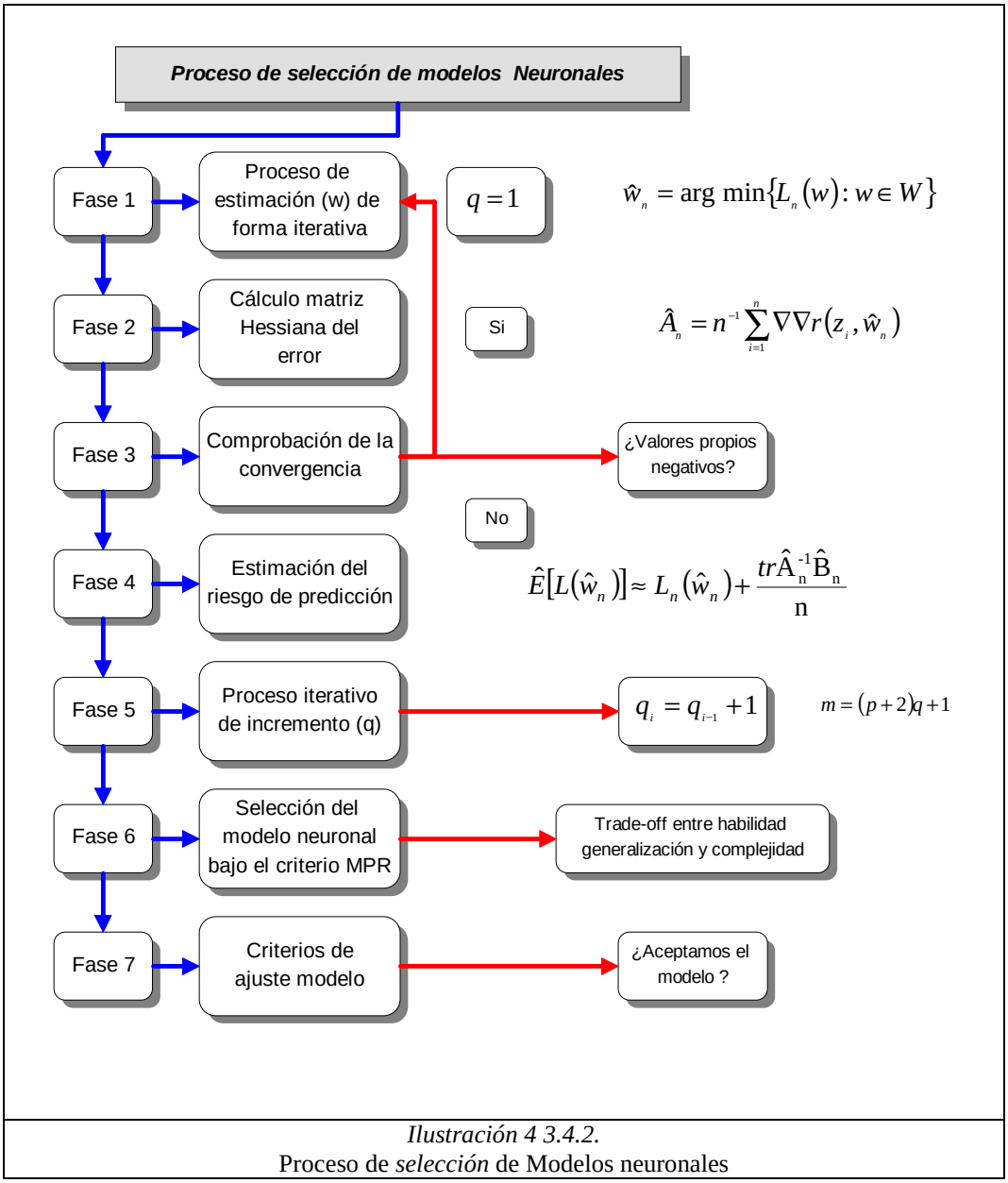


Ilustración 4 3.4.2.
Proceso de *selección* de Modelos neuronales

Así, en la primera fase, se estiman los parámetros, "w", que minimizan la función de pérdida o error, $L_n(w)$, en la segunda fase, se calcula la matriz Hessiana del error, $\hat{A}_n$. Posteriormente en la fase tercera se comprueba la convergencia, es decir, que no existan valores propios negativos[82]. En la cuarta fase, se estima el riesgo de predicción (de forma algebraica o por remuestreo). Dicho proceso se repite de forma iterativa incrementando el

[82] Si existen valores propios negativos es síntoma de que la matriz está mal condicionada y el proceso debe inicializarse otra vez.

número de neuronas en la capa oculta una a una, de forma que, el valor máximo que puede llegar a tomar es "n", el tamaño muestral del aprendizaje[83]. Posteriormente en la sexta fase, se selecciona el modelo neuronal a partir del principio MPR, que expresa el *trade-off* entre la habilidad por generalizar del modelo y su complejidad. En último lugar, se comprueba la adecuación del modelo.

El proceso de identificación anterior posee una pieza esencial para su desarrollo, la estimación del riesgo de predicción. Tal y como se ha comentado puede ser estimado por métodos algebraicos y por métodos de remuestreo, (véase ilustración 5.3.4.2.).

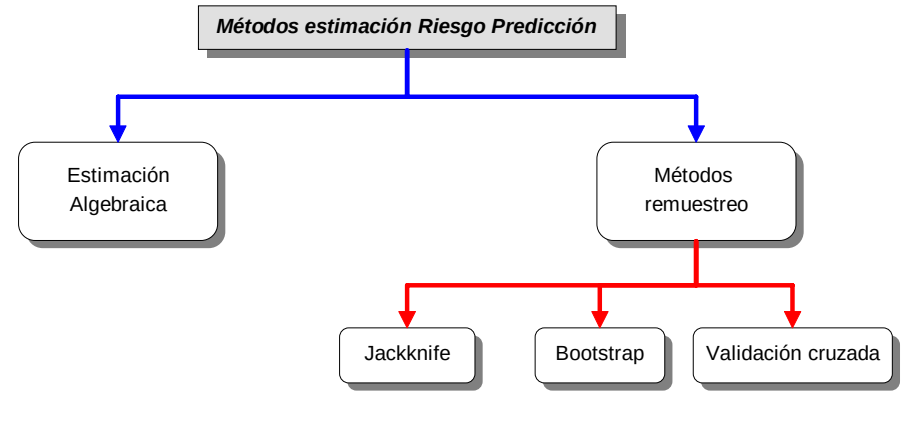


Ilustración 5 3.4.2.
Métodos de estimación *Riesgo de predicción*

En primer lugar, detallamos el método algebraico como alternativa a la versión analítica. Recordemos que en la cuarta fase de la ilustración 4.3.4.2. se describe la necesidad de estimar el riesgo de predicción, que ahora definimos como, PR_q y cuyo valor esperado es, $PR_q \equiv E\left[L\left(\hat{w}_n\right)\right]$. La estrategia utilizada para estimarlo consiste en obtener los dos primeros términos de la expansión de Taylor para la función de pérdida. Si "$\hat{w}_n$" es el estimador de mínima discrepancia, éste converge de forma asintótica a "w_0" cumpliéndose que,

$$\sqrt{n}\left(\hat{w}_n - w_0\right) \rightarrow N\left(0; \hat{C}_n\right)$$

[83] Hay que tener presente que para "*p*" variables el número total de parámetros es, $m = (p+2)q+1$, que suele ser menor que "*n*". En la práctica se acepta como bueno un límite definido por el ratio, "n/m".

donde,

$$\hat{C}_n = \hat{A}_n^{-1} \mathrm{B} \hat{A}_n^{-1}$$

y

$$A = E\left[\nabla\nabla r(z, w_0)\right]$$
$$\mathrm{B} = E\left[\nabla r(z, w_0) \nabla r(z, w_0)^T\right]$$

siendo, "A" y "B matrices no singulares; ∇, gradiente $(mx1)$; $\nabla\nabla$, (mxm)operadores hessianos de "w" y "m", el número de parámetros de la red neuronal. Si tomamos el desarrollo de Taylor para la función de pérdida definida $L(\hat{w}_n)$, evaluada en el punto "w_0", tenemos que[84],

$$L(\hat{w}_n) \approx L(w_0) + \nabla L(w_0) \nabla w^T + \frac{1}{2} \nabla w^T A \nabla w$$

donde, aplicando operadores de esperanza y considerando que, $\nabla L(w_0)$es cero, el valor esperado y la varianza de la función de pérdida fuera de la muestra utilizada para el aprendizaje posee la siguiente expresión formal,

$$E\left[L(\hat{w}_n)\right] \approx L(w_0) + \frac{tr\mathrm{A}^{-1}\mathrm{B}}{2\mathrm{n}}$$

y

$$\mathrm{var}\left[L(\hat{w}_n)\right] \approx \frac{tr\mathrm{A}^{-1}\mathrm{B}\mathrm{A}^{-1}B}{2n^2}$$

si repetimos dicho proceso para $L_n(\hat{w}_n)$ y se sustituye $L(w_0)$ por,

$$E\left[L(\hat{w}_n)\right] + \frac{tr\mathrm{A}^{-1}\mathrm{B}}{2\mathrm{n}}$$

donde,

$$L_n(\hat{w}_n) = \frac{1}{2n} \sum_{i=1}^{n} \left[y_i - g(x_i; \hat{w}_n)\right]^2$$

obtenemos finalmente la siguiente expresión,

$$\hat{E}\left[L(\hat{w}_n)\right] \approx L_n(\hat{w}_n) + \frac{tr\hat{\mathrm{A}}_{\mathrm{n}}^{-1}\hat{\mathrm{B}}_{\mathrm{n}}}{\mathrm{n}}$$

que es la estimación del riesgo de predicción.

[84] En este punto el gradiente de la *función de pérdida* es cero.

En la práctica se desconoce el valor "$tr\mathrm{A}^{-1}\mathrm{B}$" y debe ser estimado, reemplazando "A" y "B" por sus estimaciones, $(\hat{A}_n ; \hat{B}_n)$ que son,

$$\hat{A}_n = n^{-1}\sum_{i=1}^{n} \nabla\nabla r(z_i, \hat{w}_n)$$

$$\hat{B}_n = n^{-1}\sum_{i=1}^{n} \nabla r(z_i, \hat{w}_n)\nabla r(z_i, \hat{w}_n)^T$$

siendo las expresiones resultantes,

$$\hat{V}[L(\hat{w}_n)] \approx \frac{tr\left(\hat{\mathrm{A}}_{\mathrm{n}}^{-1}\hat{\mathrm{B}}_{\mathrm{n}}\right)^2}{2n^2}$$

$$L(\hat{w}_0) \approx L_n(\hat{w}_n) + \frac{tr\hat{\mathrm{A}}_{\mathrm{n}}^{-1}\hat{\mathrm{B}}_{\mathrm{n}}}{2\mathrm{n}}$$

Así la expresión,

$$\frac{tr\hat{\mathrm{A}}_{\mathrm{n}}^{-1}\hat{\mathrm{B}}_{\mathrm{n}}}{\mathrm{n}} \approx \hat{E}[L(\hat{w}_n)] - L_n(\hat{w}_n)$$

nos indica que el término "$tr\mathrm{A}^{-1}\mathrm{B}$" expresa la diferencia entre al error esperado en la muestra no utilizada para el aprendizaje, es decir la de test, y el valor esperado del error en el propio proceso de aprendizaje. Un valor relativamente grande indica la presencia de *overfitting* o *sobreapredizaje*, indicando inestabilidad en el modelo.

La segunda forma de estimar el error de predicción son los métodos de remuestreo, procedimientos habituales en el campo no paramétrico para estimar el error estadístico (véase ilustración 5.3.4.2.). Dos son los métodos esenciales, la *validación cruzada*, que surge de la psicometría y los métodos de *Bootstrap*[85] y *Jackknife* (con reemplazamiento). Estos dos últimos son una clara alternativa a los métodos de Monte-Carlo para realizar inferencia estadística, ya que no necesitan la definición a priori del mecanismo por el cual generamos los datos. En su contra tenemos que necesitan de una computación intensiva.

[85] En esencia el *Bootstrap* consiste en un procedimiento de inferencia cercano al método de máxima verosimilitud, pero con la ventaja de que no necesita que las expresiones estén disponibles. Un mecanismo que simplifica dicha dificultad es la utilización del algoritmo EM o Baum-Welch, vinculado a la inferencia Bayesiana. Véase Hastie, T; Tibshirani, R.; Friedman, J. (2001). **The Elements of Statistical Learning. Data Mining, Inference, and Prediction**, pp. 231-235, Springer.

El método de *bootstrap* propuesto por Efron[86] (1979) consiste en aproximar la función de distribución desconocida, F, de los datos observados mediante muestreo aleatorio con reemplazamiento obteniendo, $\hat{F}$. El procedimiento determina un conjunto de *n*-muestras del mismo tamaño sobre las cuales se calculan el estadístico que deseamos realizar inferencia. Si repetimos dicho proceso, nos permite obtener una distribución simulada del estadístico, (véase ilustración 6.3.4.2.).

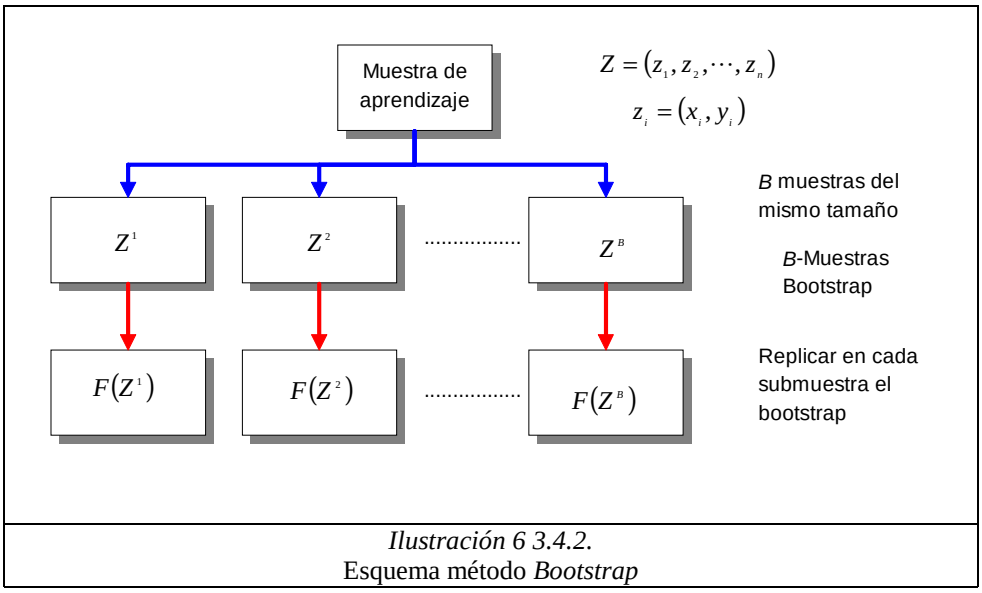


Ilustración 6 3.4.2.
Esquema método *Bootstrap*

El método *jackknife* es mucho más refinado y es anterior al *Bootstrap*[87]. La dificultad de su implementación es computacional ya que los estadísticos son calculados para *n*-posibles muestras de tamaño $(n-1)$, dejando que en cada cálculo fuera una observación distinta.

Algunos paquetes estadísticos recomiendan utilizar en primer lugar el método de *jackknife* para examinar la influencia de las observaciones y posteriormente utilizar el *bootstrap* para estimar el estadístico[88].

[86] Existen nuevas técnicas asociadas al método del *bootstrapping* cuyo objetivo es reducir la componente de varianza del error de predicción por agregación, bootstrap aggregating o "bagging", véase Berthold, M; Hand, D.J. (2003). **Intelligent Data Analysis**. **An Introduction**, Second Edition, Springer.
[87] Si desea implementar los dos métodos, véase capítulo 9 en Masters, T. (1995). **Advanced Algorithms for Neural Networks. A C++ Sourcebook**, Wiley.
[88] Véase capítulo 30 de **S-PLUS4 Guide to Stastistics**, (1997), Data Análisis Products Divison, MathSoft Inc.

Finalmente está el método de *validación cruzada*[89](CV). Inherente al mismo está también la idea de *leave-one-out*, donde la validación consiste en repetir un procedimiento, por ejemplo el cálculo del error de predicción, *n*-veces controlado por criterios de validación estadística, reservando una observación en el proceso. Pero con la importante diferencia de que se realiza para muestras diferentes, una de ellas es la muestra de aprendizaje, $D^{train} = \{x_i, y_i\}_{i=1}^{m}$ y la otra, la muestra de validación simple, $D^{test} = \{x_i, y_i\}_{i=1}^{n-m}$ ($m < n$). Para el caso que nos ocupa, la expresión de la validación cruzada del promedio del error al cuadrado es,

$$CV(\hat{w}_n) = \frac{1}{2n}\sum_{i=1}^{n}\{y_i - g_q(x_i; \hat{w}_{n-1})\}^2$$

3.4.3. *Especificación* econométrica de los modelos neuronales.

La utilización de modelos neuronales como instrumento de modelización econométrica está viviendo en este momento una rápida expansión, tanto en el ámbito académico como en el campo de las aplicaciones industriales y/o en las aplicaciones económico-financieras[90]. El aspecto más remarcable de ellos, ya comentado en apartados anteriores, es su capacidad para extraer procesos no lineales con un mínimo conjunto de hipótesis de partida sobre la naturaleza del proceso generador de datos. Además debemos remarcar que los modelos de redes neuronales constituyen un caso particular de los modelos parámetricos no lineales[91] y su proceso de aprendizaje es paralelo al mecanismo de estimación estadística de los parámetros.

Las aplicaciones en el entorno económico y econométrico poseen un gran potencial de crecimiento, abarcando desde la modelización de series temporales[92], la estimación no paramétrica de los parámetros, la simulación de procesos de aprendizaje por parte de los

---

[89] Véase para mayor detalle, Hastie, T; Tibshirani, R.; Friedman, J. (2001). **The Elements of Statistical Learning. Data Mining, Inference, and Prediction**, pp. 214-217, Springer.

[90] Véase Wong, Bo. K. ; Yakup Selvi. (1998). **Neural network applications in finance: A review and analysis of literature (1990-1996),** *Information & Management*, 34, pp. 129-139.

[91] Véase Kennedy, P. (1998). **A Guide to Econometrics**, 4ª Ed., pp. 307-309, Blackwell Publishers.

[92] Véase por ejemplo, Zhang, G. et al. (1998). **Forecasting with artificial neural networks: The state of the art**, *International Journal of Forecasting*, 14, pp. 35-62; Kanas, A.; Yannopoulos, A. (2001). **Comparing linear and nonlinear forecasts for stock returns**, *International Review of Economics and Finance*, 10, pp. 383-398; Swanson, N.R.; White H. (1997). **Forecasting economic time series using flexible versus fixed specification and linear versus nonlinear econometric models**, *International Journal of Forecasting*, 13, pp. 439-461.

agentes económicos, modelización económica[93], etc. A nuestro entender el crecimiento está asegurado, prueba de ello es que en los últimos años ha existido un creciente interés en buscar paralelismos entre modelos neuronales, economía y econometría. Más específicamente, desde la óptica econométrica, recordamos que la metodología neuronal descansa principalmente en la idea de búsqueda de una forma funcional desconocida a priori entre unos *inputs* o entradas y unos *outpus* o salida. En este ámbito, los esfuerzos académicos se están orientando hacia la relación entre la *modelización econométrica* y las posibles *arquitecturas* o *topologías* de los modelos neuronales artificiales, así si consideramos las diferentes aplicaciones podemos realizar la siguiente clasificación[94], exponiendo aquellos modelos más usuales[95].

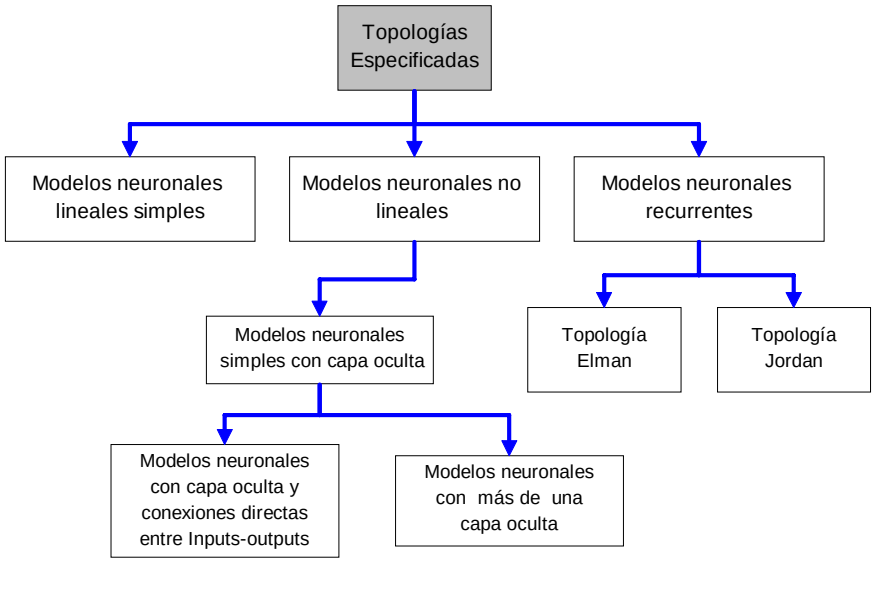


Ilustración 1.3.4.3.
Topologías *especificadas*

[93] Véase los siguientes trabajos, Aiken, M.; Krosp, J.; Vanjani, M.; Govindarajulu, Ch.; Sexton, R. (1994). **A Neural Network for Predicting Total Industrial Production**, *Journal of End User Computing*, Vol. 7, No. 2, pp. 19-23; Worzala, E., Lenk, M., Silva, A. (1995). **An Exploration of Neural Networks and Its Application to Real Estate Valuation**, *The Journal of Real Estate Research*, Vol. 10, No. 2, pp. 185-201; Alon, I.; Min Qi, Sadowski, R.J. (2001). **Forecasting agregate retail sales: a comparison of artificial neural networks and traditional methods**, *Journal of Retailing and Consumer Services*, 8, pp. 147-156 y Tkacz, G. (2001). **Neural networks forecasting of Canadian GDP growth**, *International Journal of Forecasting*, 17, pp. 57-69.

[94] La teoría que justifica estas relaciones se puede encontrar con explicaciones geométricas intuitivas en Duda, R.O.; Stork, D.G.; Hart, P.E. (2000). **Pattern Classification and Scene Analysis: Pattern Classification**, Wiley.

[95] En todos los casos se ha considerado modelos con un solo *output* por sencillez y función de salida, $f(\cdot)$, lineal.

En primer lugar tenemos los modelos que tratan relaciones lineales *input-output* o problemas de clasificación con distribuciones gausianas simples, es decir, *Simple Linear Network*. En segundo lugar, los modelos que generan funciones arbitrarias *input-output* o tratan problemas de clasificación con distribuciones de probabilidad arbitrarias, es decir, *Single Hidden Layer Network* y *Múltiple Hidden Layer*, y en último lugar, los sistemas recurrentes, similares en complejidad a la modelos markovianos arbitrarios, es decir, los modelos con topologías de *Elman* y *Jordan*, (véase ilustración 1.3.4.3.).

La primera de las arquitecturas es la expresión de una red lineal sin capa oculta (*Simple Linear Network*) representada por la ilustración 2.3.4.3. cuya expresión formal es,

$$y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \varepsilon$$
$$X = \left(x_1, x_2, \cdots, x_p\right)'$$
$$\left\{\beta_j, j = 0,1,\cdots, p\right\}$$

donde," y ", es el valor de salida (*output*), " x " es el vector de entrada (*inputs*), β_j es el vector de ponderaciones o parámetros a estimar que une las entradas con la salida.

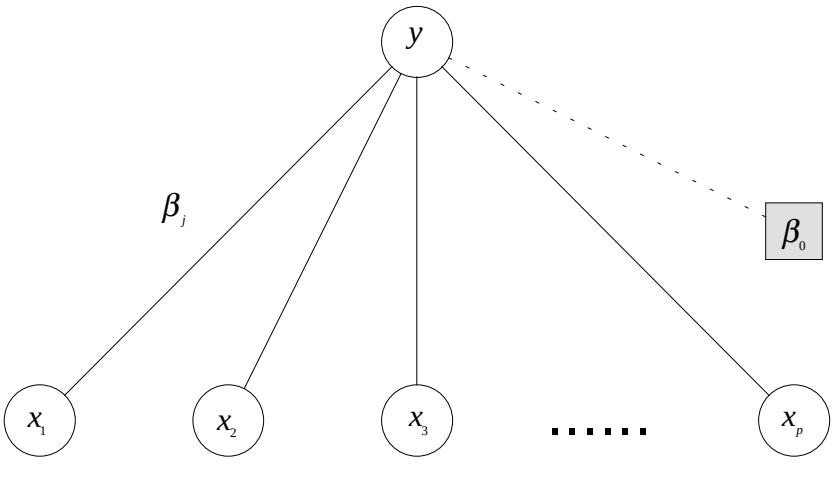


Ilustración 2.3.4.3.
Topología: *Simple Linear Network.*

En segundo lugar, si consideramos características no lineales de la respuesta del modelo neuronal, entonces este aspecto nos lleva a la siguiente expresión,

$$y = \beta_0 + \sum_{j=1}^{q} \beta_j g\left(\sum_{i=1}^{p} \phi_{ij} x_i + \phi_{0j} \right) + \varepsilon$$
$$X = (x_1, x_2, \cdots, x_p)'$$
$$\{ \phi_{ij}, i = 0,1,\cdots, p, j = 1,\cdots, q \}$$
$$\{ \beta_j, j = 0,1,\cdots, q \}$$

donde," *y* ", es el valor de salida (*output*), " *x* " es el vector de entrada (*inputs*), β_j es el vector de ponderaciones o parámetros a estimar que une las entradas con la capa oculta y ϕ_{ij}, las ponderaciones que vinculan la capa oculta con la salida, (véase ilustración 3.3.4.3.). La función de transferencia es " *g* ", que puede poseer características lineales o no. Así por ejemplo, si consideramos la función *logística*, representaríamos un modelo *Logit*, y si utilizamos una *normal acumulada*, entonces estamos frente la presencia de un modelo *Probit*, (véase apartado 3.2.).

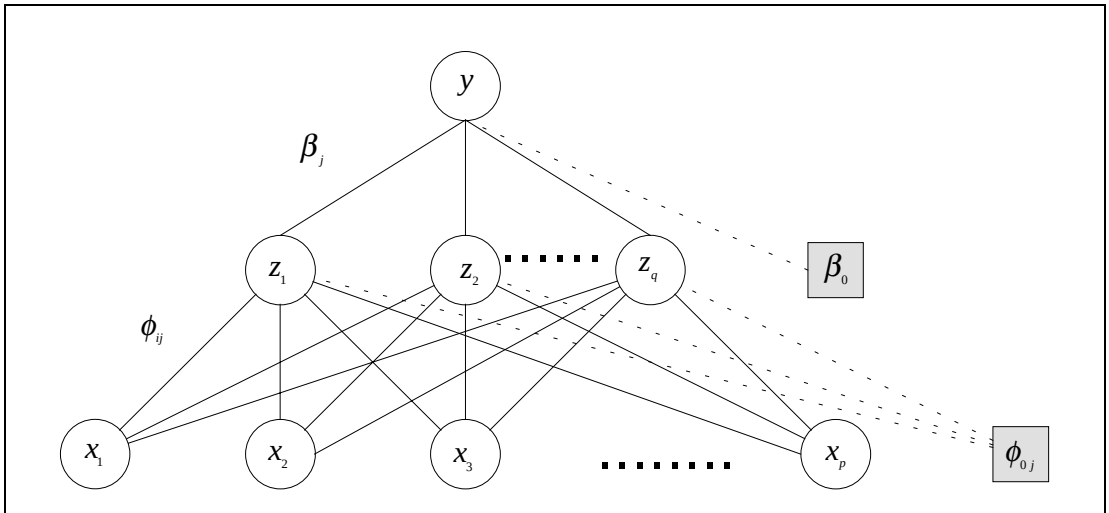


Ilustración 3.3.4.3.
Topología: *Single Hidden Layer Network*

En tercer lugar, una posible variante de la topología anterior *Single Hidden Layer Network* es la presencia de conexiones directas entre *inputs* y *outputs*, *Augmented Hidden Layer Network*, (véase ilustración 4.3.4.3.).

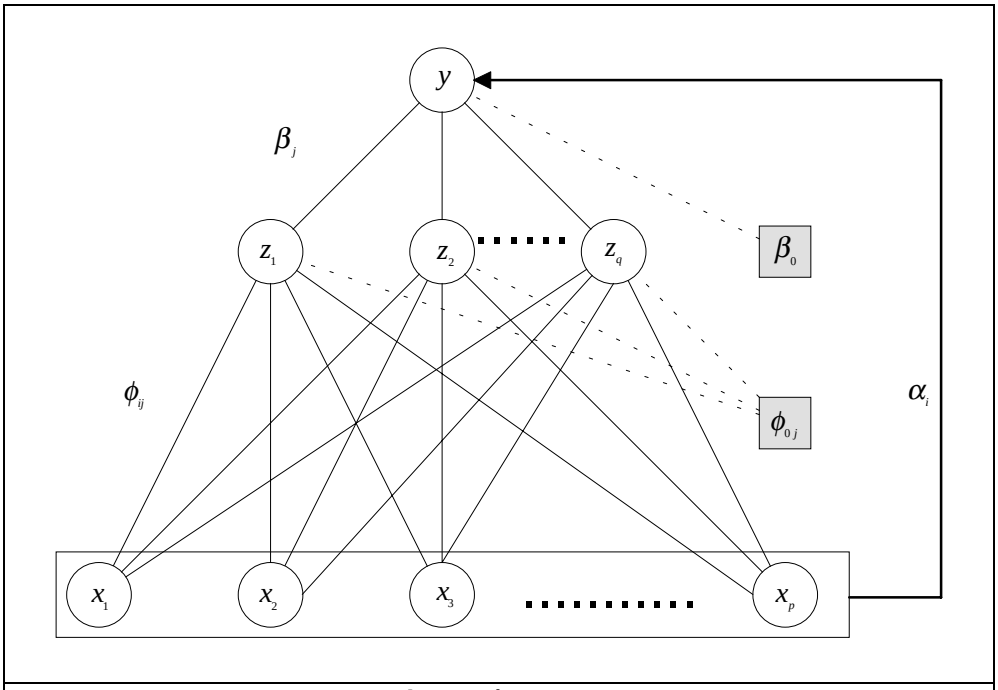


Ilustración 4.3.4.3.
Topología: *Augmented Hidden Layer Network*

Su especificación es la siguiente[96],

$$y = \sum_{i=1}^{p} \alpha_i x_i + \beta_0 + \sum_{j=1}^{q} \beta_j g\left(\sum_{i=1}^{p} \phi_{ij} x_i + \phi_{0j} \right) + \varepsilon$$
$$X = (x_1, x_2, \cdots, x_p)'$$
$$\{\phi_{ij}, i = 0,1,\cdots, p; j = 0,1,\cdots, q\}$$
$$\{\beta_j, j = 0,1,\cdots, q\}$$
$$\{\alpha_i, i = 1,2,\cdots, p\}$$

donde," *y* ", es el valor de salida (*output*), " *x* " es el vector de entrada (*inputs*), β_j es el vector de ponderaciones o parámetros a estimar que une las entradas con la capa oculta, ϕ_{ij}, las ponderaciones que vinculan la capa oculta con la salida y α_i, las ponderaciones que unen de forma directa el vector de entrada con el de salida.

[96] Un aspecto de importancia sería comprobar la eficiencia neuronal frente otros métodos clásicos de aproximación como puede ser las series de *Fourier* o la regresión *Polinómica*. Véase el capítulo 1 de Bishop, C.M. (1995). **Neural Networks for Pattern Recognition**, Clarendon Press-Oxford.

En cuarto lugar, permitimos el aumento de la topología al incrementar el número de capas ocultas, así elevamos el nivel de complejidad del modelo. Su especificación es la siguiente,

$$y = \beta_0 + \sum_{j=1}^{r} \beta_j f\left(\sum_{j=1}^{q} \alpha_{ij} g\left(\sum_{i=1}^{p} \phi_{ij} x_i + \phi_{0j} \right) \right) + \varepsilon$$
$$X = \left(x_1, x_2, \cdots, x_p\right)'$$
$$\left\{ \phi_{ij}, i = 0,1,\cdots, p; j = 1,\cdots, q \right\}$$
$$\left\{ \alpha_{ij}, i = 0,1,\cdots, q; j = 1,\cdots, r \right\}$$
$$\left\{ \beta_j, j = 1,\cdots, r \right\}$$

donde," *y* ", es el valor de salida (*output*), " *x* " es el vector de entrada (*inputs*), ϕ_{ij} es el vector de ponderaciones que une las entradas con la primera capa oculta, α_{ij} las ponderaciones que vinculan la primera capa oculta con la segunda capa oculta y β_j , son las ponderaciones que unen la última capa oculta con la salida, (véase ilustración 5.3.4.3.).

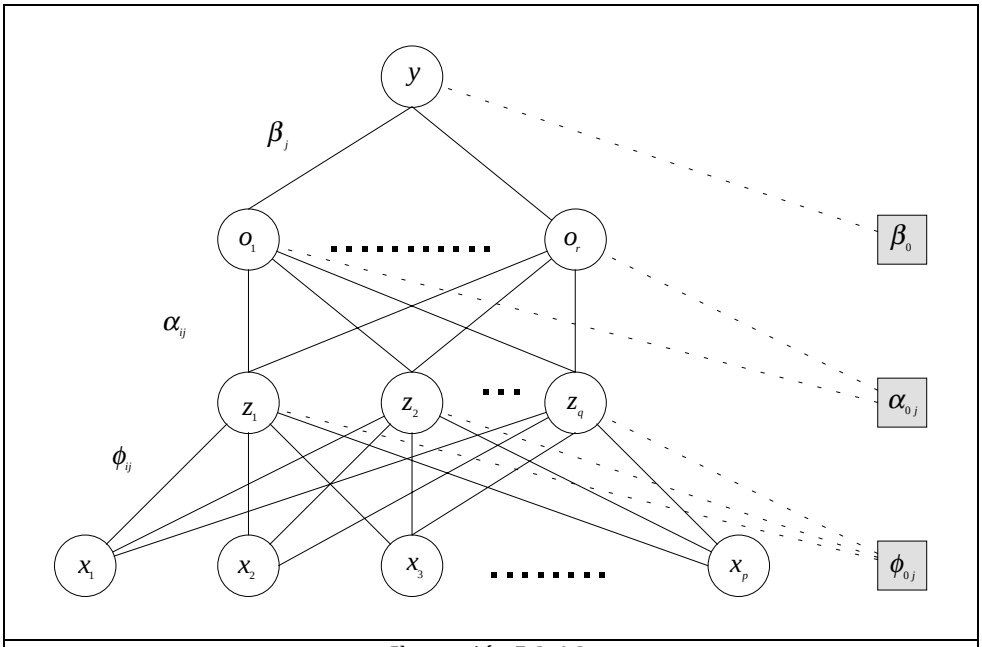


Ilustración 5.3.4.3.
Topología: *Multiple Hidden Layer Network*

En último lugar, presentamos las especificaciones de las redes recurrentes, en especial las redes *Elman* y *Jordan*. La primera de ellas establece su recurrencia en la propia capa oculta, en cambio la segunda topología la establece en la propia capa output.

La expresión del modelo de red recurrente, *Elman*, es,

$$
\begin{aligned}
& y = \beta_0 + \sum_{j=1}^{q} \beta_j z_{j,t} + \varepsilon \\
& z_{j,t} = g\left(\sum_{i=1}^{p} \phi_{ij} x_i + \phi_{0j} + \delta_{ij} z_{j,t-1} \right) \\
& X = \left(x_1, x_2, \cdots, x_p \right)' \\
& \left\{ \phi_{ij}, i = 0,1,\cdots, p, j = 1,\cdots, q \right\} \\
& \left\{ \beta_j, j = 0,1,\cdots, q \right\} \\
& \left\{ \delta_{ij}, i = 1,\cdots, p, j = 1,\cdots, q \right\}
\end{aligned}
$$

donde," y ", es el valor de salida (*output*), " x " es el vector de entrada (*inputs*), ϕ_{ij} es el vector de ponderaciones que une las entradas con la capa oculta, δ_{ij}, son las ponderaciones que vinculan la capa oculta con la capa de *contexto* (capa que recoge la recurrencia sucesiva en cada iteración) y β_j, las ponderaciones que unen la capa oculta con la salida. Podemos observar que dicha topología dependerá de un valor inicial y de los inputs o variables exógenas, (véase ilustración 6.3.4.3.).

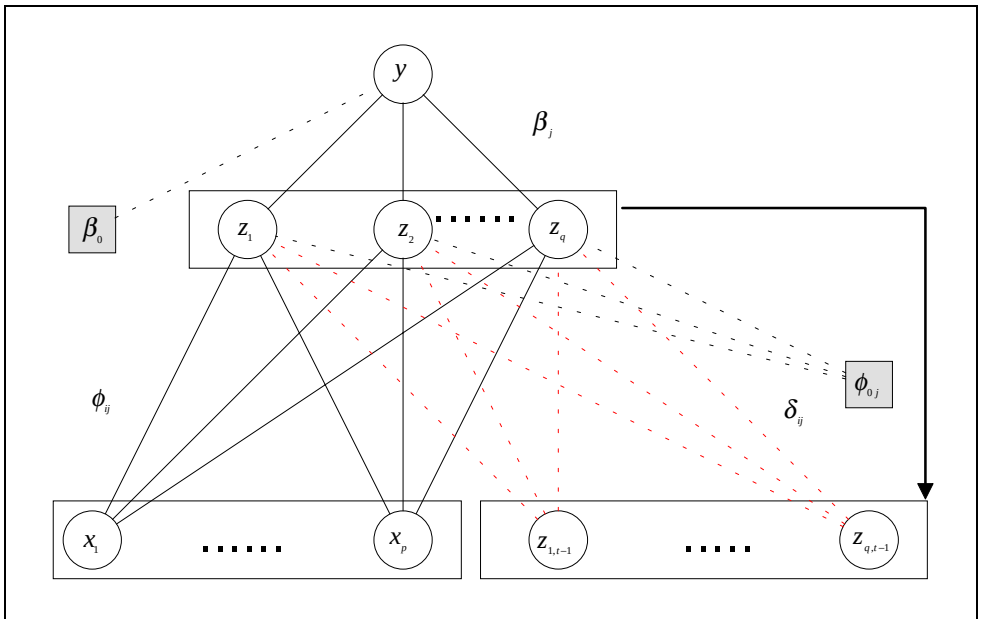


Ilustración 6.3.4.3.
Topología: *Elman Network*

Desde una óptica econométrica el anterior modelo es un modelo dinámico de variables latentes, de forma que, estamos frente a procesos recursivos de estimación cercanos a los filtros de *Kalman*, que permiten obtener estimaciones consistentes de los parámetros.

Finalmente, la especificación de la topología de *Jordan* posee la siguiente expresión econométrica,

$$
\begin{aligned}
& y_t = \beta_0 + \sum_{j=1}^{q} \beta_j z_{tj} + \varepsilon \\
& z_{tj} = g\left(\sum_{i=1}^{p} \phi_{ij} x_i + \phi_{0j} + \delta_{ij} \hat{y}_{t-1} \right) \\
& X = \left(x_1, x_2, \cdots, x_p \right)' \\
& \left\{ \phi_{ij}, i = 0,1,\cdots, p, j = 1,\cdots, q \right\} \\
& \left\{ \beta_j, j = 0,1,\cdots, q \right\} \\
& \left\{ \delta_{ij}, i = 1,\cdots, p, j = 1,\cdots, q \right\}
\end{aligned}
$$

donde," y ", es el valor de salida (*output*), " x " es el vector de entrada (*inputs*), ϕ_{ij} es el vector de ponderaciones que une las entradas con la capa oculta, δ_{ij}, las ponderaciones que vinculan la capa oculta con la capa de *contexto* (capa que recoge la recurrencia sucesiva en cada iteración, en este caso, desde la capa de salida) y β_j, las ponderaciones que unen la capa oculta con la salida, (véase ilustración 7.3.4.3.).

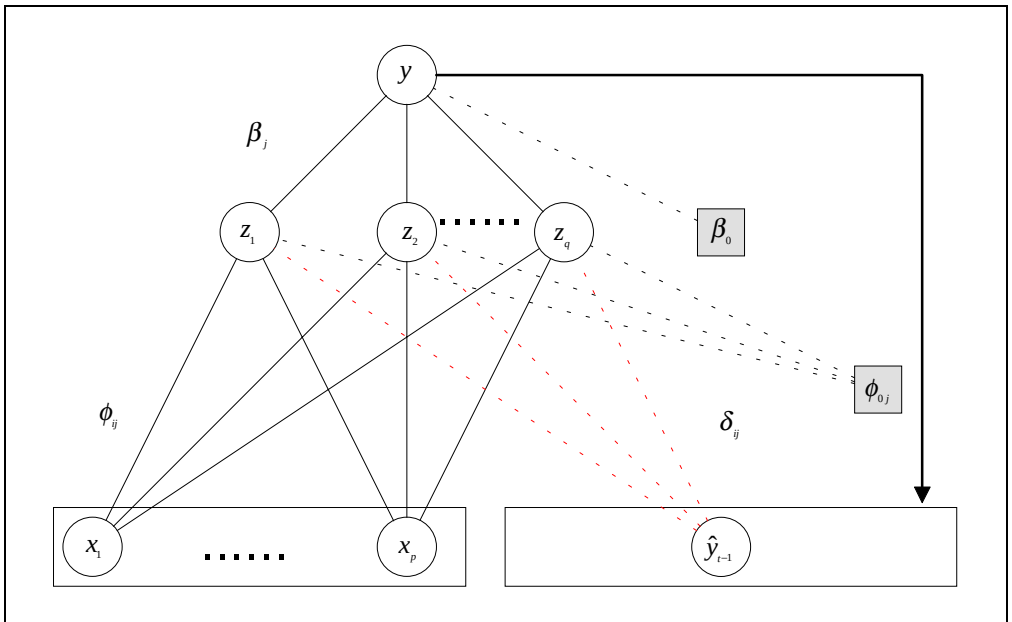


Ilustración 7.3.4.3.
Topología: *Jordan Network*

Una vez especificados los modelos neuronales desde la óptica econométrica, planteamos una aproximación estadística del contraste de variables relevantes y de validación de modelos, siempre desde la óptica neuronal, (véase apartados 3.4.4. y 3.4.5.).

3.4.4. Aproximación estadística del *contraste* de variables relevantes.

Una vez desarrollado en capítulos anteriores los métodos de identificación de los modelos neuronales y sus especificaciones econométricas más habituales, ahora nos acercamos a una aproximación estadística al contraste de variables relevantes[97], siempre en sintonía con el principio de parsimonia[98] o razón de *Occam*.

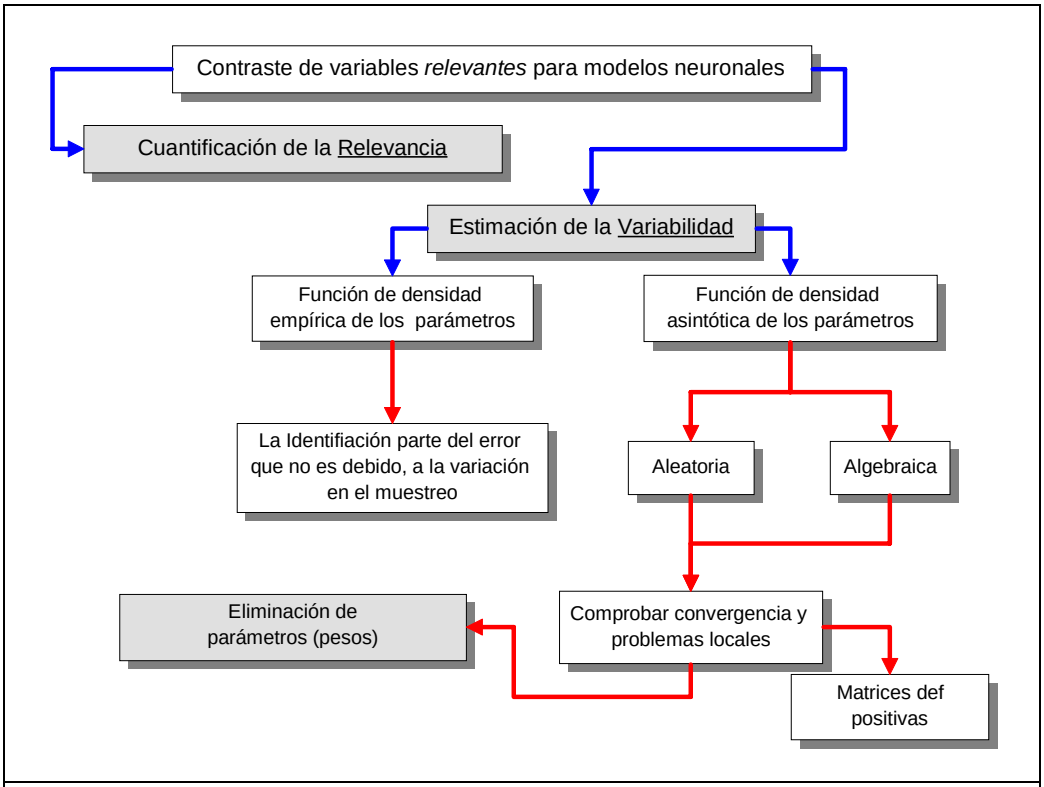


Ilustración 1.3.4.4.
Representación del proceso de *contraste de variables relevantes*

[97] Es necesario recordar que desde el punto vista clásico, la selección de variables relevantes descansa en los contrastes hipótesis sobre los parámetros mediante el diseño de *tests* específicos. En un entorno no paramétrico existen trabajos que proponen metodologías propias, como por ejemplo, véase Vieu, P. (1994). **Choice of regressors in nonparametric estimation**, *Computational Statistics & Data Analysis*, 17, pp. 575-594.

[98] Existen varias razones para desear modelos parsimoniosos, algunas de ellas son, en primer lugar, que son más fáciles de interpretar ya que capturan la esencia de la relación del modelo, en segundo lugar, se evita el problema de las relaciones *espúreas* o en términos neuronales el *sobreaprendizaje*.

La evaluación estadística de la significación del poder explicativo de las variables posee tres aspectos. Definir que se entiende por variable relevante, estimar la variabilidad en el muestreo de los parámetros y el contraste de hipótesis, (véase ilustración 1.3.4.4.).

Respecto al primero de los aspectos, la relevancia de las variables se cuantifica mediante la derivada parcial, $(\partial y/\partial x_j)$, en un contexto de modelos lineales. Es decir, el resultado es el parámetro "b_j" que acompaña al *input*, "x_j", siempre que asumamos que las variables son independientes[99]. Pero en el entorno de los modelos no lineales la derivada definida no es constante y es necesario elaborar nuevas medidas de sensibilidad de "y" respecto de "x_j", (véase tabla 1.3.4.4.).

Tabla 1.3.4.4.

Definición(1)	***Expresión(1)***	***Definición(2)***	***Expresión(2)***
Derivada promedio (AvgD)	$AvgD(x_i)=\frac{1}{n}\sum_{i=1}^{n}\frac{\partial \hat{y}_i}{\partial x_{ij}}$	Valor Absoluto Promedio Elasticidad (AvgLM)	$AvgLM(x_i)=\frac{1}{n}\sum_{i=1}^{n}\left(\left\|\frac{\partial \hat{y}_i}{\partial x_{ij}}\right\|\right)\left(\left\|\frac{x_{ij}}{\hat{y}_i}\right\|\right)$
Valor absoluto Derivada Promedio (AvgDM)	$AvgDM(x_i)=\frac{1}{n}\sum_{i=1}^{n}\left\|\frac{\partial \hat{y}_i}{\partial x_{ij}}\right\|$	Promedio Contribución relativa (AvgSTD)	$AvgSTD(x_i)=\frac{1}{n}\sum_{i=1}^{n}\left(\frac{(\partial \hat{y}_i/\partial x_{ij})^2}{\sum_{i=1}^{m}((\partial \hat{y}_i/\partial x_{ij})^2)}\right)$
Valores Máximos y Mínimos Derivadas (Max/Min)	$MaxD(x_i)=\underset{i=1,\cdots,n}{\text{Max}}\left\{\frac{\partial \hat{y}_i}{\partial x_{ij}}\right\}$ $MinD(x_i)=\underset{i=1,\cdots,n}{\text{Min}}\left\{\frac{\partial \hat{y}_i}{\partial x_{ij}}\right\}$	Dispersión de la sensibilidad (SDD)	$SDD(x_i)=\left(\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\partial \hat{y}_i}{\partial x_{ij}}-AvgD(x_i)\right)^2\right)^{1/2}$
Valores en valor absoluto Máximos y Mínimos Derivadas	$MaxDM(x_i)=\underset{i=1,\cdots,n}{\text{Max}}\left\{\left\|\frac{\partial \hat{y}_i}{\partial x_{ij}}\right\|\right\}$ $MinDM(x_i)=\underset{i=1,\cdots,n}{\text{Min}}\left\{\left\|\frac{\partial \hat{y}_i}{\partial x_{ij}}\right\|\right\}$	Coeficiente de Variación (CVD)	$CVD(x_i)=\frac{SDD(x_i)}{AvgD(x_i)}$
Promedio Elasticidad (AvgL)		$AvgL(x_i)=\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\partial \hat{y}_i}{\partial x_{ij}}\right)\left(\frac{x_{ij}}{\hat{y}_i}\right)$	

Fuente: Zapranis, A. y Refenes, A.P. (1999). **Principles of Neural Model Identification, Selection and Adequacy**, pp. 81, Springer y elaboración propia.

[99] Bajo esta hipótesis podemos ignorar el segundo término de la expansión de Taylor del valor ajustado obtenido por el modelo neuronal para cada una de las variables explicativas, $d\hat{y}_i=\sum_j\frac{\partial \hat{y}_i}{\partial x_{ij}}dx_1+\sum_{ij,j2}\frac{\partial \hat{y}_i}{\partial x_{ij}^2}dx_{j1}dx_{j2}+o(\|dx\|^3)$.

La ilustración 2.3.4.4. muestra el esquema de un modelo econométrico neuronal, donde se expresa la sensibilidad del valor ajustado respecto a las variables explicativas mediante derivadas parciales, cuya expresión es,

$$\frac{\partial \hat{y}}{\partial x_i} = \phi_{i1} g'(\cdot)_{z_1} \beta_1 + \phi_{i2} g'(\cdot)_{z_2} \beta_2 + \phi_{i3} g'(\cdot)_{z_3} \beta_3 + \cdots + \phi_{ij} g'(\cdot)_{z_q} \beta_j$$

donde, $g'(\cdot)_q$ es el valor de la derivada de las funciones de transferencia definidas en cada neurona oculta.

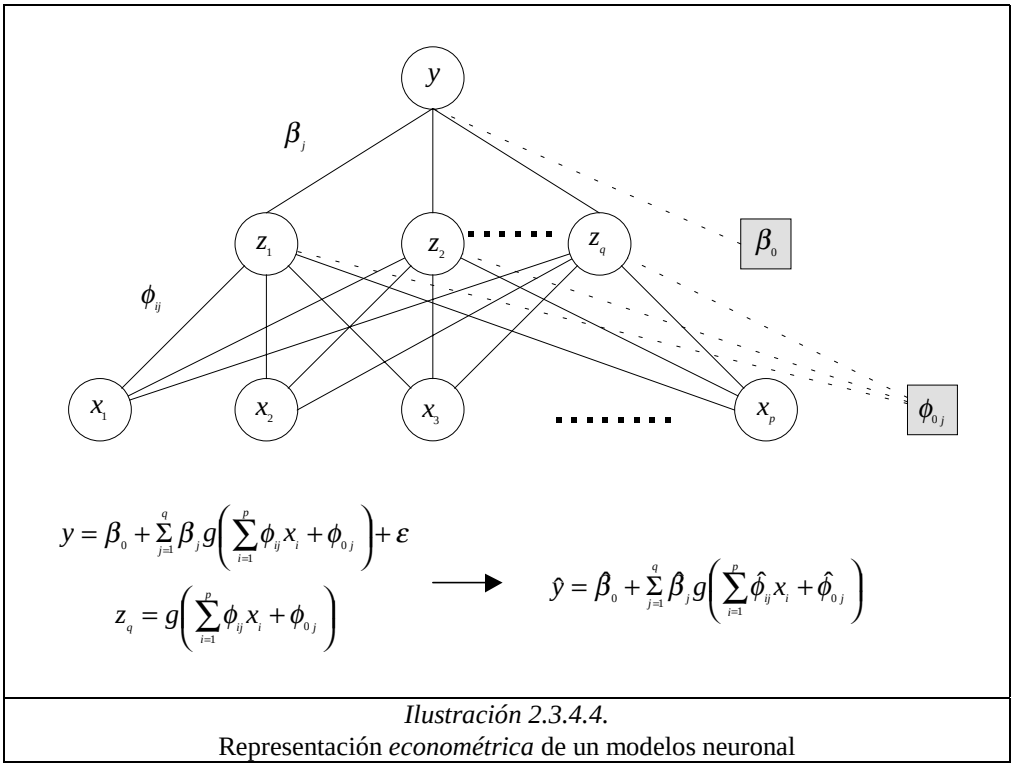


Ilustración 2.3.4.4.
Representación *econométrica* de un modelos neuronal

Pero existen otras formas de aproximar el cálculo de las derivadas parciales requeridas para determinar la importancia de las variables explicativas, una de ellas es,

$$\frac{\partial \hat{y}}{\partial x_i} = \frac{1}{q} k_i \sum_{j=1}^{q} \left| \phi_{ij} \right|$$

donde valor de "k_i" es, $k_i = \sum_{j=1}^{q} g'(\cdot)_j \beta_j$ y la *sensibilidad* se cuantifica, en este caso, mediante la suma de las ponderaciones $\sum_j \left| \phi_{ij} \right|$.

Una segunda alternativa para su cálculo, consiste en definir unos índices de sensibilidad de la forma siguiente,

$$S(x_i) = \sum_{j=1}^{q} |\phi_{ij}| \frac{|\beta_j|}{\sum_{j=1}^{q} |\beta_j|}$$

y último lugar, tenemos una medida que evalúa la sensibilidad sobre el error cometido en el modelo neuronal en su fase de aprendizaje o estimación, al sustituir una variable explicativa, "x_i", por su valor promedio. Su expresión es la siguiente,

$$S(x_i) = \frac{1}{n} \sum_{i=1}^{n} |e_{ij}| = \frac{1}{n} \sum_{i=1}^{n} |SE(\bar{x}_{ij}; \hat{w}_n) - SE(x_{ij}; \hat{w}_n)|$$

Según Zapranis y Refenes (1999), todas las anteriores medidas adolecen de ciertas dificultades, como por ejemplo, problemas de inestabilidad o la existencia de excesivas formas de cuantificar el efecto sobre "$\hat{y}$" de cualquier variable explicativa, "x_i", etc. Aún así los autores proponen nuevas medidas de sensibilidad pero sobre el grado de ajuste del modelo: en primer lugar, el efecto sobre la función de pérdida o error por introducir una pequeña perturbación en los *inputs*, en segundo lugar, el efecto sobre la función de pérdida por reemplazar "x" por su valor medio,

$$\Delta L_n(x_i) = L_n(x; \hat{w}_n) - L_n(\bar{x}^{(i)}; \hat{w}_n)$$

y en tercer lugar, el efecto sobre el coeficiente de determinación, R^2, por un pequeño cambio en "x".

Una vez desarrollado varios aspectos de la relevancia de las variables explicativas en el contexto de la modelización neuronal, debemos ahora concentrarnos en la estimación de la variabilidad en el muestreo de los estimadores. Para ello existe tres métodos, en primer lugar, la técnica de *bootstrap*[100] de carácter local, donde sus resultados acostumbran a sobreestimar el error estándar de los estimadores.

[100] Recordamos que en el ámbito de la inferencia, el *bootstrap* es esencia una implementación no paramétrica o paramétrica del método de máximo verosimilitud, cuya ventaja es no necesitar de expresiones formales.

Una segunda posibilidad consiste en métodos de muestreo desde la distribución asintótica de los parámetros de los modelos neuronales, tal y como se argumentó en el apartado 3.3.4, su distribución asintótica posee la forma siguiente,

$$\sqrt{n}(\hat{w}_n - w_0) \rightarrow N(0; \hat{C}_n)$$

con un comportamiento multivariante normal de media cero y con una matriz de varianzas-covarianzas, $\hat{C}_n$, cuya expresión es,

$$\hat{C}_n = \hat{A}_n^{-1} \hat{B} \hat{A}_n^{-1},$$

siendo, "$\hat{w}_n$" el vector de parámetros estimado y "w_0" el vector de parámetros poblacionales desconocidos.

De todos modos en el contexto de contraste de hipótesis que nos encontramos, es necesario la definición del error estándar estimado de los parámetros, cuya expresión estándar es la siguiente,

$$\hat{\theta} - z_\alpha \hat{\sigma} \leq \theta_0 \leq \hat{\theta} + z_\alpha \hat{\sigma},$$

y su variabilidad,

$$\hat{\sigma} = \left[\frac{1}{k-1} \sum \left(\hat{\theta}^{(a)} - \hat{\theta}(\circ) \right)^2 \right]^{1/2} \quad a = 1,2,\cdots,k$$

siendo,

$$\hat{\theta} = h(\hat{w}_n)$$

$$\hat{\theta}(\circ) = \frac{1}{k} \sum_{a=1}^{k} h(\hat{w}_n^{(a)})$$

y $h(\circ)$, una función arbitraria.

Un último procedimiento consiste en evaluar mediante *n*-simulaciones de *bootstrap*, la variabilidad de los estimadores, generando una distribución empírica de $\hat{\theta}^a$ con los resultados obtenidos[101].

[101] Para nuevos desarrollos, como por ejemplo, los vínculos entre el bootstrap y la inferencia Bayesiana incorporando técnicas de *Bagging* o de búsqueda estocástica (*Bumping*), véase el capítulo 8 de Hastie, T; Tibshirani, R.; Friedman, J. (2001). **The Elements of Statistical Learning. Data Mining, Inference, and Prediction,** Springer.

Para finalizar nos queda mencionar aspectos relacionados con los contrastes de hipótesis. Con esta finalidad, debemos recordar que para comprobar la relevancia de unos *inputs* en un modelo lineal, es necesario establecer una hipótesis nula sobre la que recae el peso del contraste, es decir,

$$H_0 : b_i = 0 .$$

White[102] (1989) propuso elaborar contrastes basados en la distribución asintótica de los parámetros de los modelos neuronales, cuya expresión es la siguiente,

$$H_0 : S\hat{w}_n = 0$$
$$H_1 : S\hat{w}_n \neq 0$$

donde,*" S "*, es la selección de pesos o conexiones elegidas para comprobar su relevancia conjunta. Bajo esta hipótesis nula se cumple que[103],

$$\sqrt{n}(\hat{w}_n - w_0) \rightarrow N(0; \hat{C}_n)$$
$$\hat{C}_n = \hat{A}_n^{-1}\hat{B}\hat{A}_n^{-1}$$

así, para un nivel de significación prefijado, α, no rechazaremos la hipótesis nula si el estadístico no excede, $(1-\alpha)$ percentiles de la distribución de χ_q^2,

$$n\hat{w}_n^T S^T \left(S\hat{C}_n S\right)^{-1} S\hat{w}_n \rightarrow \chi_q^2$$

Para tamaños muestrales grandes, la probabilidad de rechazar correctamente la hipótesis nula tiende a la unidad, además realizar contrastes de significación individuales para los *inputs*, "x_i", requiere "$p+1$" matrices invertidas, el cálculo de $\left(S\hat{C}_n S\right)$ y también la matriz Hessiana,*" A "*.

En este ámbito y utilizando el intervalo definido anteriormente,

$$\hat{\theta} - z_\alpha \hat{\sigma} \leq \theta_0 \leq \hat{\theta} + z_\alpha \hat{\sigma} ,$$

podemos construir el contraste individual siguiente, que dependerá del criterio escogido de sensibilidad,

$$H_0 : \hat{\theta} = \theta_0 \quad \rightarrow \quad H_0 : \left(\partial L_n(\hat{w}_n) / \partial x_j\right) = 0$$
$$H_A : \hat{\theta} \neq \theta_0 \quad \rightarrow \quad H_0 : \left(\partial L_n(\hat{w}_n) / \partial x_j\right) - z_\alpha \hat{\sigma} > 0$$

[102] White, H. (1989). **Learning in artificial neural networks: a statistical perspective**, *Neural Computation*, 1, pp. 425-464.

[103] La hipótesis de un modelo neuronal único local es equivalente a la independencia de los parámetros del modelo lineal.

Si aceptamos la hipótesis nula, significa que podemos suprimir dicha variable del modelo, ya que el efecto sobre el error cometido en el proceso de aprendizaje no es sensiblemente diferente a cero. Por el contrario, si consideramos las técnicas de *bootstrap* podemos utilizar la distribución empírica[104] para generar el intervalo de confianza de los parámetros del modelo neuronal, cuya expresión es,

$$\hat{\theta}(\alpha) \leq \theta_0 \leq \hat{\theta}(1-\alpha).$$

La evaluación del efecto sobre el modelo por suprimir una variable necesita de una medida muy conocida en el ámbito lineal, el *coeficiente de determinación* (R^2), y más apropiadamente, su *versión ajustada* ($\overline{R}^2$), cuya expresión es,

$$\overline{R}^2 = 1 - \frac{(SSR/(n-m))}{(SST/(n-1))}$$

donde,

SSR es la parte no explicada por el modelo neuronal,

SST es la variación total a explicar,

m es el número de parámetros del modelo,

n es el tamaño de la muestra.

Por definición, la parte residual o no explicada (en la base de test) por el modelo es igual a, $2nL_n(\hat{w}_n)$, que debidamente ajustada por el nivel de complejidad del modelo y expresado en función del riesgo de predicción, tenemos que su expresión es, $2nE[L(\hat{w}_n)]$. Este aspecto nos permite redefinir, $\overline{R}^2$, en función de $tr\mathrm{A}^{-1}\mathrm{B}$, que recordamos es la diferencia entre el error esperado en la muestra no utilizada para el aprendizaje y el valor esperado del error en el propio proceso de aprendizaje, siendo el resultado final de la expresión formal,

$$\overline{R}^2 = 1 - \frac{2nL_n(\hat{w}_n) + 2trA_n^{-1}B_n}{SST} = 1 - \frac{SSR + 2trA_n^{-1}B}{SST}.$$

[104] Generada a partir de la obtención de los *percentiles*. Dicha distribución acostumbra a ser sustancialmente diferente a una normal y más cercana a una distribución asimétrica, tipo χ^2.

De nuevo existe un problema computacional, ya que si consideramos que existen *p*-variables potencialmente explicativas, tal aspecto nos generará $(2^p - 1)$ distintas configuraciones a partir de los *inputs*. Es necesario por lo tanto, arbitrar algún mecanismo de selección secuencial no excepto de críticas al no estar definido un cuerpo teórico que describa el fenómeno, (véase ilustración 3.3.4.4.).

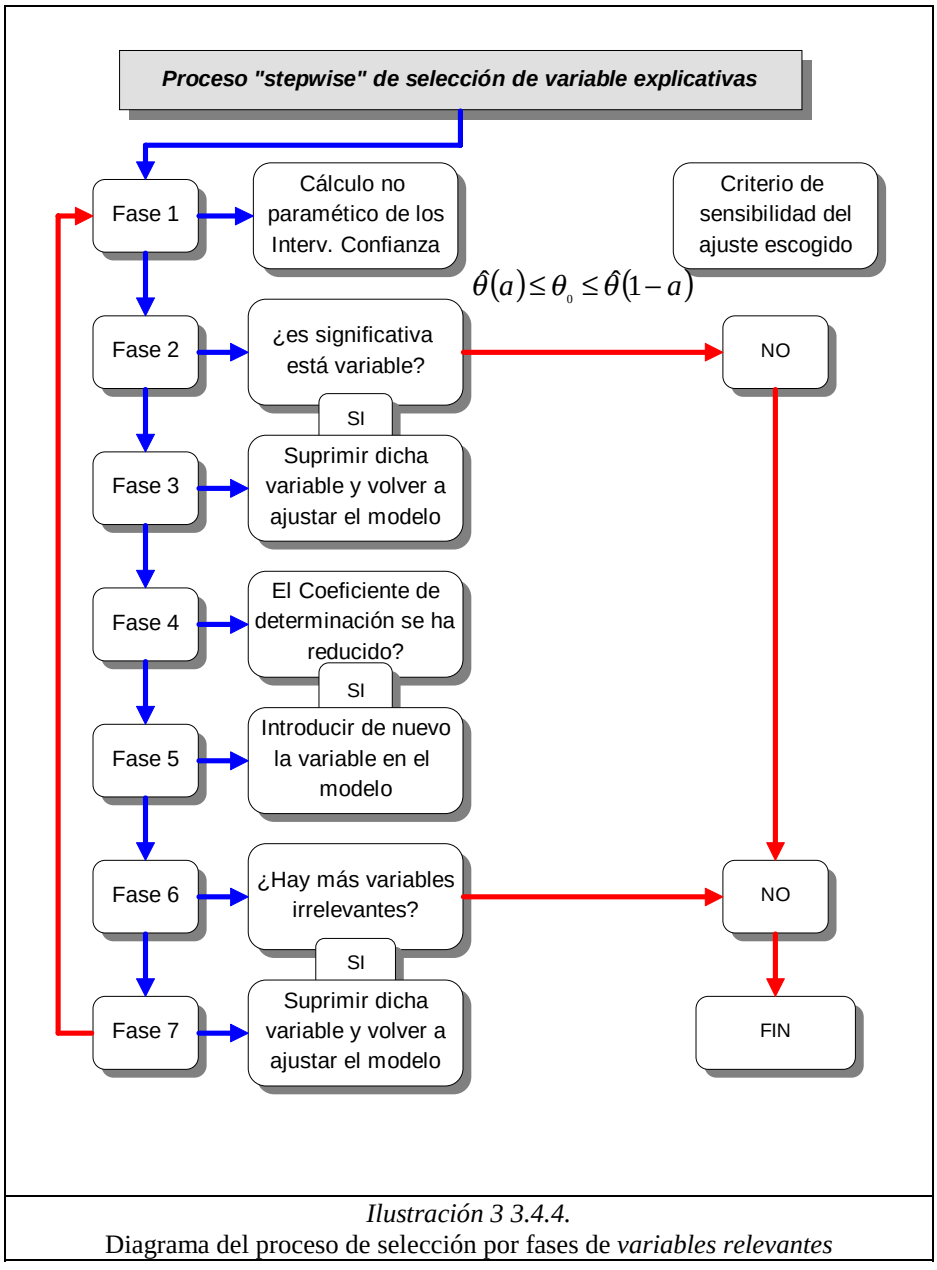


Ilustración 3 3.4.4.
Diagrama del proceso de selección por fases de *variables relevantes*

3.4.5. Test de *contraste de* *hipótesis* de validación de los modelos.

La naturaleza no paramétrica de los modelos neuronales establece que la condición de correcta especificación del modelo es necesaria pero no suficiente para diagnosticar a través de los errores la validez del mismo en su globalidad. Este aspecto nos obliga a redefinir procedimientos de contraste para errores aditivos independientes a partir de los ya conocidos en el ámbito paramétrico.

El primero de ellos es el contraste de autorrelación en los residuos, que suele detectarse su presencia mediante una inspección visual de la función de autocorrelación parcial (FAC), en el caso que nos ocupa las bandas de confianza son,

$$-z_{\alpha}\frac{1}{\sqrt{n}} \leq \hat{\gamma}_k \leq +z_{\alpha}\frac{1}{\sqrt{n}}$$

donde, z_{α}, es el valor del percentil $100(1-\alpha)$ de la distribución Normal estándar. La segunda forma de comprobar la autocorrelación es mediante el test Box-Pierce donde el estadístico es Q y su expresión formal,

$$Q = n\sum_{k=1}^{m}\hat{\gamma}_k^2$$

que bajo la hipótesis nula se distribuye asintóticamente como un χ_m^2. Existe otro contraste habitual que es el contraste de Ljung-Box, cuya expresión es,

$$LB = n(n+2)\sum_{k=1}^{m}\frac{\hat{\gamma}_k^2}{n-k}$$

que bajo la hipótesis nula se distribuye asintóticamente como un χ_m^2.

Finalmente el contraste de Durbin-Watson,

$$d = \frac{\sum_{i=2}^{n}(e_i - e_{i-1})^2}{\sum_{i=1}^{n}e_i^2} \approx 2(1-\hat{\gamma}_1)$$

siendo, $\hat{\gamma}_1$, el coeficiente de autocorrelación muestral de orden 1, que posee su homólogo para los modelos de regresión no lineales, es decir, los modelos neuronales, $g(x;\hat{w}_n)$.

La equivalencia asintótica para estos modelos surge mediante la aproximación alrededor del verdadero valor del parámetro mediante el primer término de la aproximación de Taylor,

$$g(x;w_0)+\frac{\partial(x;w_0)}{\partial\hat{w}_n^T}(\hat{w}_n-w_0)$$

La segunda comprobación habitual es el estadístico-F que necesita la hipótesis de normalidad del término estocástico del modelo y que nos facilita información sobre la fiabilidad global del modelo. En el caso no lineal, se ordenan los errores de menor a mayor cuantía y se genera el estadístico,

$$F_{(k)}=\frac{e_n^2+e_{n-1}^2+\cdots+e_{n-k+1}^2}{e_1^2+e_2^2+\cdots+e_k^2}$$

bajo la hipótesis nula de errores absolutos homocedásticos, su ordenación es aleatoria. Si estos residuos son i.i.d. el estadístico $F_{(k)}$ se distribuye según $F_{(k,k)}$.

Este apartado finaliza el capítulo 3, que junto al capítulo 2, reúnen una parte importante de los esfuerzos metodológicos que en el entorno neuronal se están gestando en últimos años. Investigadores de la talla de **Achilleas Zaprabis**, **Apostolos-Paul Refenes**, **Trevor Hastie**, **Robert Tibshirani**, **Jerome Friedman**, etc. están contribuyendo a desmitificar poco a poco el sombrío campo de la interpretación de los modelos neuronales con sus aportaciones, aunque somos conscientes del largo camino que queda por recorrer. En el capítulo 6, se ha utilizado sólo una parte pequeña del potencial existente, circunscrito a las herramientas informáticas utilizadas.