Fundamentos Matemáticos de la Red Neuronal Perceptrón Multicapa

Salvador Pintos

Abril-2000

Resumen

Se presenta aquí una formulación vectorial del MLP con tres capas y de los procesos de retropropagación asociados.

Índice

1. Introducción

La red neuronal tipo perceptrón multicapa MLP (multilayer perceptron) tanto por su rol histórico como por las innumerables aplicaciones que en ella se fundamentan, es la más citada y expuesta en la literatura específica. Sin embargo, la mayoría de las veces su exposición incluye su analogía biológica y la presentación del perceptrón, y se elude la formulación matemática del MLP o se presenta con unas complejas sumatorias basadas en escalares. El propósito de este trabajo es presentar la formulación del MLP basada en análisis vectorial. Se presenta de manera sintética la función hacia adelante (*forward*) y se derivan los gradientes necesarios para los algoritmos de retropropagación (*backpropagation*). Por último, se determinan el gradiente de la función red neuronal respecto del vector de entrada, necesario para cualquier análisis de sensibilidad.

2. Nociones matemáticas preliminares

2.1. Gradiente de una Función

Sea $F(x)$ una función vectorial: $F : R^j \longrightarrow R^k$. Su gradiente es la matriz $(j\ x\ k)$:

$$\nabla F = [\nabla F_1, ..., \nabla F_k] \tag{1}$$

donde $F_j\ j = 1....k$ es el j-esimo componente de F.
si $F(x) = Ax$ (donde A es una matriz (k x j)), entonces:

$$\nabla F = A^T \tag{2}$$

donde se notaráA^T, la traspuesta de la matriz A.

2.1.1. Corolarios

- Si c es un vector de igual dimensión que x, y $F(x) = c^T x$ entonces $\nabla F = c$.
- Si $F(x) = kx$ e I_j la matriz identidad, entonces: $\nabla F = kI_j$

2.2. Gradiente de la composición de funciones

Sean $F : R^j \longrightarrow R^k$, $H(y)$ una función vectorial: $H : R^k \longrightarrow R^l$ y $G = H \circ F$ la composición de F y H, entonces se cumple:

$$\nabla G(x) = \nabla F(x)\ \nabla H\left(F\left(x\right)\right) \tag{3}$$

nótese el orden inverso de la multiplicación conforme de matrices (j x k)*(k x l) .

Si F y H son funciones lineales : $F(x) = Ax, \quad H(y) = By$, entonces por Eqn. 2 y Eqn. 3

$$\nabla G = (B\ A)^T = A^T B^T \tag{4}$$

de nuevo nótese el orden de la multiplicación matricial.

Si una misma función escalar, h, se aplica a cada variable x_j de x de modo que

$$F(x) = \begin{pmatrix} h(x_1) \\ ..... \\ h(x_k) \end{pmatrix} \Longrightarrow \nabla F(x) = \begin{bmatrix} h'(x_1) & 0 & 0 \\ ... & ... & ... \\ 0 & 0 & h'(x_k) \end{bmatrix} \tag{5}$$

Para abreviar la notación de matrices que solo tienen valores en la diagonal se notará: DIAG(z) a la matriz que tiene al vector z formando su diagonal y los restantes elementos 0 . Por ejemplo, la expresión anterior se expresa como $\nabla F(x) = DIAG(h\prime(x))$.

El producto AB es el producto habitual de matrices, para representar el producto componente a componente de dos matrices se usará la notación A#B. Como consecuencia de la notación se tiene que si z y w son 2 vectores de igual dimensión$DIAG(w) * z = w\#z$.

2.3. Función logística

La función logística de parámetro β está dada por:

$$y = \frac{1}{1+e^{-\beta x}} \quad cuya \quad derivada \quad es \quad y' = \beta y(1-y) \tag{6}$$

Si β tiende a infinito la logística tiende a la función escalón. Nótese que la derivada se expresa en función de la imagen, propiedad que, simplifica los procesos de derivación hacia atrás(backpropagation).

Habitualmente β es igual a 1. En dicho caso: $y - \frac{1}{2} = \frac{1}{2}\tanh(\frac{x}{2})$ luego, salvo una transformación lineal la logística y la tangente hiperbólica son equivalentes.

3. Red Neuronal con una capa oculta

Las Redes Neuronales de alimentación hacia adelante, perceptrón multicapa, con tres capas (una capa oculta), son las más citadas en las aplicaciones y por su sencillez servirán aquí para exponer los principios matemáticos de una Red Neuronal

3.1. Formulación de una Red Neuronal

Se considera cada patrón individual como un vector columna tanto para la entrada como para la salida. Se asume una red con 3 capas cuyas dimensiones son: I, H, O neuronas en las capas de entrada, oculta, y de salida, respectivamente. El umbral de excitación de las neuronas de las capas oculta y de salida admite una representación más elegante mediante la inclusión de neuronas artificiales. A la capa de entrada y a la capa oculta se le agrega una neurona

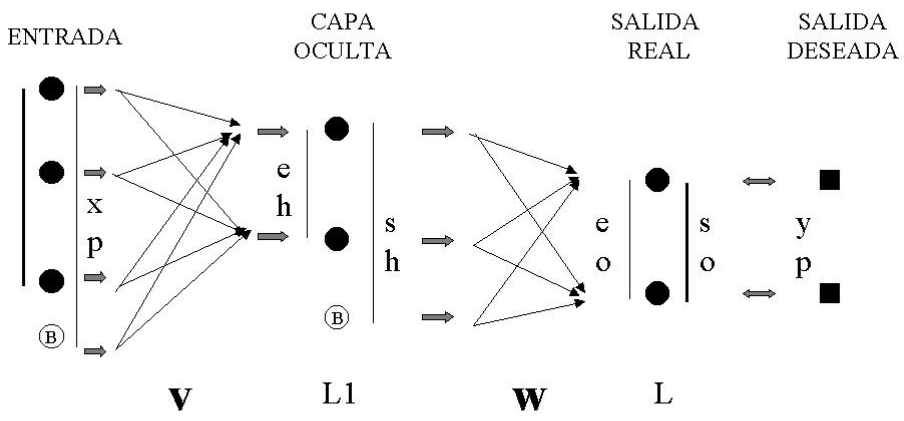

Figura 1: Vectores de la propagación hacia adelante

simbólica, (bias), a cada una, cuya salida es siempre 1 . Los pesos que unen esta neurona con las neuronas de la capa siguiente expresan el umbral de excitación de cada neurona de esta última capa. Con el agregado de éstas últimas las dimensiones ampliadas de cada capa son: $I+1, H+1, O$.

Las neuronas representa un procesador físico o simbólico donde se realizan dos operaciones específicas: primero, una aplicación lineal que sumariza el aporte de las neuronas de la capa anterior, seguido de,segundo, una transformación no lineal, función de activación, (habitualmente la función logística), que representa la respuesta de esta neurona.

El procesamiento en cada neurona es independiente del de cualquier otra neurona de su misma capa. Luego, una red neuronal permite el procesamiento en paralelo.

3.1.1. Definiciones básicas:

- xp patrón de entrada que incluye el 1(constante del bias). Dimensión: $Dim(xp) = I + 1$.
- eh vector resultante de la aplicación lineal en la capa oculta. $Dim(eh) = H$.

- sh vector de salida de la capa oculta. que incluye el 1(constante del bias) $Dim(sh) = h + 1$.
- eo vector resultante de la aplicación lineal en la capa de salida . $Dim(eo) = O$.
- so vector de salida de la capa de salida . $Dim(so) = O$.
- yp patrón de salida. $Dim(yp) = O$.
- V matriz de pesos entre la capa de entrada y la oculta. $Dim(V) = H \times (I + 1)$.
- V_{jh} es el peso (parámetro) que refleja la intensidad de la asociación entre la neurona h de la capa de entrada con la neurona j de la capa oculta.
- W matriz de pesos entre la capa oculta y la capa de salida . $Dim(W) = O \times (H + 1)$.
- W_{kj} es el peso (parámetro) que refleja la intensidad de la asociación entre la neurona j de la capa oculta con la neurona k de la capa de salida.

Formular V de esta manera permite expresar la transformación lineal en la capa oculta como el producto de una matriz por un vector: $eh = V\ xp$. De manera que la fila j de V contiene los pesos correspondientes a la neurona j de la capa oculta que ponderan la salida de la capa de entrada para formar la respuesta lineal de la neurona j de la capa oculta.

La justificación de los W_{kj} es idéntica a la de V.

Se notará como $L(z)$ la función vectorial de activación que a cada componente del vector z le aplica la logística. $L1(z)$ la misma función pero agregando un 1 en el vector de las imágenes (para representar la salida de la neurona bias). Por ejemplo, si eh es el resultado de la aplicación lineal en eh, el vector de salida de la capa oculta es $sh = L1(eh)$. que difiere de $sa = L(eh)$ en el 1que agrega al final. ($Dim(sa) = h,\ Dim(sh) = h + 1$). En cambio, $so = L(eo)$ ya que la capa de salida no contiene neurona bias.

Nota: en lo que sigue se usará la Logística pero un análisis totalmente similar puede realizarse con cualquier otra función de activación. Por ejemplo, en el perceptrón clásico se usa la función escalón.

3.2. Propagación hacia adelante: funciones y gradientes

La propagación hacia adelante (Forward) del patrón xp hasta obtener la salida so, se expresa mediante la composición de funciones elementales. En la tabla siguiente se incluyen estas funciones -en el orden en que se realiza su composición- así como sus gradientes y las dimensiones asociadas:

FUNCIÓN	DIMENSION	GRADIENTE	DIMENSION
$eh = V\ xp$	$I+1 \to H$	V^T	$(I+1) \times H$
$sh = L1(eh)$	$H \to H+1$	$DIAG(sa\#(1-sa)) \,\|\| \, 0$	$H \times (H+1)$
$eo = W\ sh$	$H+1 \to O$	W^T	$(H+1) \times O$
$so = L(eo)$	$O \to O$	$DIAG(so\#(1-so))$	$O \times O$
$Error = \|\|so - yp\|\|^2$	$O \to 1$	$2\ (so - yp)$	O

En los gradientes anteriores se ha supuesto que las matrices de pesos V y W son constantes.

En el caso de la función $sh = L1(eh)$, a la matriz del gradiente se le ha agregado una columna adicional con un vector de ceros que corresponde a la derivada de la constante 1 (última neurona, bias, tiene salida 1). Se incluye también la función error que se usará más adelante.

3.2.1. Familia de funciones Red Neuronal

Si se expresa la salida so de la Red Neuronal en función del patrón de entrada xp se tiene:

$$so(xp) = L[W\ (L1(V\ xp))] \tag{7}$$

luego, una Red Neuronal es una función de la forma dada por la Eqn 7. Las funciones de este tipo constituyen una familia que dependen de los parámetros que conforman las matrices de pesos V y W. En lo sucesivo se usará en forma indistinta el término Red Neuronal tanto para la familia como para uno de sus miembros.

La importancia de las redes neuronales se debe al hecho que esta familia ha demostrado ser robusta tanto para estimar funciones (determinísticas o aleatorias), como para problemas de clasificación supervisada. En este último caso los patrones de salida son discretos: habitualmente están incluidos en el conjunto $\{0,1\}^O$ (como máximo existen 2^O clases).

4. Formulación del problema

El objetivo, tanto de estimación como de clasificación mediante redes neuronales, es determinar, por el método de mínimos cuadrados, el conjunto óptimo de $(I+1) \times H$ pesos de V, y $(H+1) \times O$ pesos de W para aproximar una función dada por un conjunto de patrones de entrada y salida, mediante una Red Neuronal MLP. Más precisamente:

Dado un conjunto de N patrones de entrada y salida, $\{(xp_k, yp_k)\,, k = 1, ..., N\}$,hallar las matrices de pesos V y W de modo de minimizar la siguiente función:

$$\text{mín}\, Error(V,W) = \sum_{k=1}^{N} \|so\,(xp_k) - yp_k\|^2 \tag{8}$$

donde $so\,(xp_k)$ es la salida de la red.

Theorem 1 *Teorema de existencia (Kolmogorov). Asumiendo una capa oculta de* $H = 2I+1$ *neuronas, existe una red neuronal que aproxima cualquier función continua* $F : R^I \rightarrow R^O$, *(dado.*$\varepsilon > 0, \exists(V,W)$ *tal que* $\min Error(V,W) < \varepsilon$*).*

El resultado es más general: esta familia de redes neuronales es densa en L2. Nótese que las funciones Redes Neuronales son continuas por lo que la aproximación a una función no continua no es puntual sino bajo la métrica de cuadrado integrable.

5. Fase de aprendizaje (retropropagación)

La naturaleza no lineal de la Redes Neuronales como función de los pesos V y W , y el excesivo número de pesos $(I+O+1)\times H+O$, hace que la minimización de la función error Eqn 8 sea un complejo problema de optimización no lineal. La superficie de la función Error a minimizar es extremadamente no convexa y multimodal, y es factible que el método de optimización converja a un mínimo local pero no global.

El proceso iterativo, (debido al método de optimización), para la determinación de V y W se denomina habitualmente *fase de aprendizaje*. Esta fase se caracteriza por el algoritmo de optimización elegido. En general es un algoritmo local, de gradiente o derivados de este, que se basa en la determinación de las derivadas de la función Error, Eqn 8, respecto de cada uno de los pesos de V y W con el próposito de modificarlos en una dirección de descenso de la superficie de la función Error.

Por lo expuesto en Eqn 3 las derivadas se determinan de atrás hacia adelante, aplicando el gradiente de la composición de funciones dando origen al nombre del método: retropropagación, (backptopagation). Además, los vectores auxiliares necesarios para la derivación respecto a los W_{kj}participan "posteriormente" en la determinación de las derivadas respecto a los V_{jh}.

Puesto que la ,minimización en Eqn 8 es aditiva se determinarán las derivadas de la función Error para un patrón y luego se sumarán para hallar la suma de las derivadas.

5.1. Cálculo de los gradientes hacia atrás (Backpropagations)

En lo que sigue se aplicará sistemáticamente la Eqn 3 y los resultados de la sección 3.2.

$$\frac{\partial error}{\partial eo} = DIAG[so\#(1-so)]\ 2(so-yp) = 2so\#(1-so)\#(so-yp) = aux \quad (9)$$

Se designa como aux al vector gradiente ya que interviene en varias expresiones futuras.

5.2. Derivadas respecto de los pesos de W

Para la determinación de las derivadas respecto de los pesos de W se notará como ${}^{o}u_i$ al vector fila de dimensión o que tiene todos sus elementos nulos salvo un 1 en la posición i. Como $eo = Wsh$ y W_{ij} sólo participa en la i esima componente de eo con coeficiente sh_j, las derivadas respecto a W_{ij} están dadas por:

$$\frac{\partial eo}{\partial W_{ij}} = sh_j \ {}^{o}u_i \tag{10}$$

que implica:

$$\frac{\partial Error}{\partial W_{ij}} = sh_j \ {}^{o}u_i \ aux = sh_j \ aux_i = (aux \ sh^T)_{ij} \tag{11}$$

entonces, cada derivada respecto de W_{ij} es el elemento i, j de la matriz $aux \ sh^T$. Luego esta es la matriz de las derivadas que en lo sucesivo notaremos:

$$\partial W = aux \ sh^T \tag{12}$$

Como para toda salida so, ésta pertenece al intervalo $[0, 1]$, entonces: $so(1 - so) \leq \frac{1}{4}$. De las ecuaciones 9 y 12 se deduce que cada elemento i,j de la Eqn 12 está acotado en valor absoluto por:

$$abs(\partial W) \leq \frac{1}{2}(abs(so - yp) \ sh^T) \tag{13}$$

De esta última desigualdad se deriva: primero, la magnitud del incremento del peso es pequeño cuando la salida de la neurona de la capa de salida es próxima al patrón deseado, o , la salida de la neurona de la capa oculta es débil (si una neurona oculta tiene su salida próxima a 0 los pesos que surgen de ella reciben cambios insignificantes.) ; segundo, en el caso de hipotético de clasificación, donde yp_i es 1 , los elementos de aux son no positivos y para disminuir el error es necesario aumentar W_{ij}.

5.3. Derivadas respecto a los pesos de V

Ahora W es constante de modo que por la tabla de gradientes ya vista y la Eqn 9:

$$\frac{\partial error}{\partial sh} = \frac{\partial eo}{\partial sh} \quad \frac{\partial error}{\partial eo} = W^T \ aux \tag{14}$$

Aplicando el teorema del gradiente de la función compuesta:

$$\frac{\partial error}{\partial eh} = \frac{\partial sh}{\partial eh} \quad \frac{\partial error}{\partial sh} \tag{15}$$

se obtiene:

$$\frac{\partial error}{\partial eh} = [DIAG(sa\#(1-sa)) \mid\mid 0]\ W^T\ aux \tag{16}$$

Si observamos las dimensiones de las matrices en la Eqn 16 $H \times (H+1)$, $(H+1) \times O$, y $O \times 1$ y el hecho que la última columna de la primera es 0, puede simplificarse la ecuación anterior eliminando la última fila a la matriz W^T, reduciéndola a la matriz de dimensión $H \times O$, WR^T. Nótese que a la matriz W se le quita su última columna para obtener WR.

$$\frac{\partial error}{\partial eh} = DIAG(sa\#(1-sa))\ \ WR^T\ aux \tag{17}$$

o de otra forma:

$$\frac{\partial error}{\partial eh} = sa\#(1-sa)\#(WR^T\ aux) = q \tag{18}$$

Este vector gradiente, de dimensión H, se le denomina q, ya que juega un rol importante en las derivadas respecto a los V_{ij}.

Como $eh = V\ xp$ las derivadas respecto a V_{kl} están dadas por:

$$\frac{\partial eh}{\partial V_{kl}} = xp_l\ {}^H u_k \tag{19}$$

$$\frac{\partial Error}{\partial V_{kl}} = \frac{\partial eh}{\partial V_{kl}} \frac{\partial error}{\partial eh} = xp_l\ {}^H u_k\ q \tag{20}$$

$$\frac{\partial Error}{\partial V_{kl}} = xp_l\ q_k = \left(q\ xp^T\right)_{kl} \tag{21}$$

luego, cada derivada respecto de V_{kl} es el elemento k,l de la matriz $q\ xp^T$. Esta es la matriz de las derivadas que en lo sucesivo notaremos:

$$\partial V = q\ xp^T \tag{22}$$

5.3.1. Condiciones necesarias de optimización

Como la función error, Eqn 8, es continua y derivable en un mínimo local debe cumplirse que $\partial W = 0$ y $\partial V = 0$, sin embargo del análisis de estas matrices surgen consideraciones interesantes acerca de la introducción de soluciones espúreas.

$$\partial W = aux\ sh^T = (2so\#(1-so)\#(so-yp))\ \ sh^T \tag{23}$$

Luego, cuatro factores distintos conducen a que la derivada de un peso sea cercana a 0, y sólo uno de ellos, $so - yp$, expresa que la salida de la red so es próximo al patron yp deseado. Los restantes factores $so, (1-so)$, $y\ sh$ pueden

tender a 0 debido a pesos en valor absolutos muy grandes que producen valores en la logística próximos a 0 o a 1.

$$\partial V = q\ xp^T = (sa\#(1-sa)\#(WR^T\ aux))\ xp^T \tag{24}$$

Similar análisis vale ∂V.

Las condiciones anteriores explican los grandes valles casi estacionarios de la superficie de la función Error,.donde los algoritmos de aprendizaje se estancan al realizar excesivas iteraciones, produciendo cambios insignificantes en los pesos $\partial W \cong 0$ y $\partial V \cong 0$, sin que se obtenga el óptimo.

En la fase de entrenamiento de una red neronal es conveniente observar el comportamiento de los vectores $so,\ sh,\ WR^T\ aux$, para detectar soluciones casi estacionarias debidas a las condiciones recién expuestas.

5.4. algoritmos de optimización

5.4.1. Retropropagación estándar

El cambio en los pesos se realiza luego de evaluar cada patrón. Los nuevos pesos Vnu, Wnu se expresan como:

$$Vnu = V - \lambda\,\partial V\ ; \qquad Wnu = W - \lambda\,\partial W\ ; \qquad \lambda > 0 \tag{25}$$

nótese que para decrecer en la función error es necesario dirigirse en el sentido contrario al del gradiente. El coeficiente λ se denomina *tasa de aprendizaje.*

5.4.2. Retropropagación por lote

El cambio en los pesos se realiza luego de evaluar todos los patrones: se suman las matrices ∂V_k y ∂W_k, obtenidas para cada patrón, ya que el error total, $\sum_{k=1}^{N} \|so\,(xp_k) - yp_k\|^2$ es la suma de los errores de cada patrón .

Theorem 2 *Si las matrices* $\sum_{k=1}^{N} \partial V_k$ *y* $\sum_{k=1}^{N} \partial W_k$ *no se anulan, existe* $\lambda > 0$ *que garantiza el decrecimiento de la función error.*

Sin embargo, dado que no se conoce el λ apropiado suele usarse valores fijos de la tasa de aprendizaje

6. Análisis del gradiente de la función Red Neuronal

Una vez entrenada la red se desea estudiarla como función **so**(**xp**), (salida *so* de la red en función de la entrada *xp*). La sensibilidad de la Red Neuronal (los cambios en la respuesta debidos a los cambios en las variables de entrada) pueden analisarse observando el gradiente de *so* respecto de *xp*. A partir de la tabla de gradientes y de las ecuaciones derivadas en la sección previa se tiene que el gradiente de la salida final respecto de la salida de la capa oculta es:

$$\frac{\partial so}{\partial sh} = W^T \; DIAG\,[so\#(1-so)]$$

y el gradiente de la salida de la capa oculta respecto de la entrada xp :

$$\frac{\partial sh}{\partial xp} \; = V^T \; [\; DIAG\;[sa\,\#\,(1-sa)\,]\;||\;0]$$

Luego multiplicando los gradientes se tiene que:

$$\frac{\partial so}{\partial xp} \;=\; V^T \; [\; DIAG\;[sa\,\#\,(1-sa)\,]\;||\;0] \quad W^T \; DIAG\;[so\#(1-so)]$$

que se puede resumir en:

$$\frac{\partial so}{\partial xp} \; = V^T \; DIAG\;[sa\,\#\,(1-sa)\,]\; WR^T \; DIAG\;[so\#(1-so)]$$

Nótese que las matrices de pesos V, W son constantes y que para un xp dado:

$$sa = L(Vxp) \qquad so(xp) = L[W \;\; (L1(V \;\; xp))]$$

de modo que el cálculo del gradiente $\frac{\partial so}{\partial xp}$ es muy simple.

Fijado el valor de las variables salvo una, puede representarse el gradiente anterior en función de esta última variable para realizar el análisis de sensibilidad.