UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERIA
DEPARTAMENTO DE INGENIERIA INFORMÁTICA

PROPUESTA DE TESIS

Felipe Alberto Reyes González

Profesor Guía: Victor Parada

Memoria para obtener el título de Analista en Computación Científica

Santiago, Chile

2017

Tabla de contenido

Índice de tablas

Índice de figuras

2.1 Aspectos teóricos

2.1.1 Algoritmo de retropropagación

Una regla de aprendizaje es el método que le permite adaptar los parámetros de la red. El perceptrón multicapa actualiza sus pesos en función de la salida obtenida de tal manera que los nuevos pesos permitan reducir el error de salida. Por tanto, para cada patrón de entrada a la red es necesario disponer de un patrón de salida deseada.

El objetivo es que la salida de la red sea lo más próxima posible a la salida deseada, debido a esto la es que el aprendizaje de la red se describe como un problema de minimización de la siguiente maner

$$\min_{W} E$$

donde W es el conjunto de parámetros de la red (pesos y umbrales) y E es una función de error que evalúa la diferencia entre las salidas de la red y las salidas deseadas. en la mayor parte de los casos, la función de error se define como:

$$E = \frac{1}{N}\sum_{i=1}^{N} e(i) \tag{2.1}$$

Donde N es el número de muestras y $e(n)$ es el error cometido por la red para el patrón i, definido de la siguiente manera

$$e(i) = \frac{1}{n_C}\sum_{j=1}^{n_C}(s_j(i) - y^j(n))^2 \tag{2.2}$$

Siendo $Y(i) = (y_1(i), y_2(i), \cdots, y_{n_C}(i))$ y $S(i) = (s_1(i), s_2(i), \cdots, s_{n_C}(i))$ los vectores de salida y salidas deseadas para el patrón i respectivamente.

De esta manera, si W^* es un mínimo de la función de error E, en dicho punto el error será cercano a cero, y en consecuencia, la salida de la red será próxima a la salida deseada.

Así es como el aprendizaje es equivalente a encontrar un mínimo de la función de error. La presencia de funciones de activación no lineales hace que la respuesta de la red sea no lineal respecto a los parámetros ajustables, por lo que el problema de aminimización es un problema no lineal y se hace necesario el uso de técnicas de optización no lineales para su

resolución.

Las técnicas utilizadas suelen basarse en la actualización de los parámetros de la red mediante la determinación de una dirección de búsqueda. En el caso de las redes neuronales multicapa, la dirección de búsqueda más utilizada se basa en la dirección contraria del gradiente de la función de error E, el método de gradiente descendente.

Si bien el aprendizaje de la red busca minimizar el error total de la red, el procedimiento está basado en métodos del agradiente estocástico, que son una sucesión de minimizaciones del error en función de cada patrón $e(i)$, en lugar de minimizar el error total E de la red. Aplicando el método del gradiente estocástico, cada parámetro w se modifica para cada patrón de entrada n según la siguiente regla de aprendizaje

$$w(i) = w(n-1) - \alpha \frac{\partial e(i)}{\partial w} \tag{2.3}$$

donde $e(i)$ es el error para el patrón de entrada i dado por la ecuación 2.2, y α es la tasa de aprendizaje, éste último determina el desplazamiento en la superficie del error.

Como las neuronas están ordenadas por capas y en distintos niveles, es posible aplicar el método del gradiente de forma eficiente, resultando en el *algoritmo de retropropagación* (Rumelhart, Hinton, y Williams, 1986) o *regla delta generalizada*. El término retropropagación es utilizado debido a la forma de implementar el método del gradiente en las redes multicapa, pues el error cometido en la salida de la red es propagado hacia atrás, transformándolo en un error para cada una de las neuronas ocultas de la red.

El algoritmo de retropropagación es el método de entrenamiento más utilizado en redes con conexión hacia adelante. Es un método de aprendizaje supervisado de gradiente descendente, en el que se distinguen claramente dos fases:

1. Se aplica un patrón de entrada, el cual se propaga por las distintas capas que componen la red hasta producir la salida de la misma. Esta salida se compara con la salida deseada y se calcula el error cometido por cada neurona de salida.

2. Estos errores se transmiten desde la capa de salida, hacia todas neuronas de las capas anteriores (Fritsch, 1996). Cada neurona recibe un error que es proporcional a su contribución sobre el error total de la red. Basándose en el error recibido, se ajustan los errores de los pesos sinápticos de cada neurona.

2.1.2 Problema desl desvanecimiento del gradiente

El problema del gradiente desvaneciente nace en las NN profundas, éstas utilizan funciones cuyo gradiente tienden a estar entre 0 y 1. Debido a que estos gradientes pequeños se multiplican durante la retropropagación, tienden a *desvanecerse* a través de las capas, evitando que la red aprenda en redes muy profundas.

Si se tiene una NN, la activación de una neurona de una capa intermedia i con función de activación f_i y con entrada

$$net_i(t) = \sum_j w_{ij} y^j(t-1)$$

es $y^i(t) = f_i(net_i(t))$. Además w_{ij} es el peso de la conexión desde la unidad j hasta la unidad i, $d_k(t)$ será la respuesta esperada de la unidad k de la capa de salida en el tiempo t. Usando el error cuadrático medio (*Mean square error*, MSE), el error de k será

$$E_k(t) = (d_k(t) - y^k(t))^2$$

En un tiempo $\tau \leq t$ cualquiera, el error de una neurona j que no sea una neurona de entrada es la suma de los errores externos y el error propagado hacia atrás desde la neurona previa será

$$\vartheta_j(\tau) = f_j'(net_j(\tau)) \left(E_j(\tau) + \sum_i w_{ij} \vartheta_i(\tau+1) \right)$$

El peso actualizado en el tiempo τ resulta

$$w_{jl}^{new} = w_{jl}^{old} + \alpha \vartheta_j(\tau) y^l(\tau - 1)$$

donde α es la tasa de aprendizaje, y l es una unidad arbitraria conectada a la unidad j.

En un tiempo arbitrario $\tau \leq t$, el error de la señal en una unidad j, que no sea de entrada, será la suma de los errores externos y la señal propagada anterioremente como muestra la ecuación 2.4.

$$\vartheta(\tau) \quad = \quad f_j'(net_j(\tau)) \left(E_j(\tau) + \sum_i w_{ij} \vartheta_i(\tau+1) \right) \tag{2.4}$$

Entonces, los pesos actualizados serán

$$w_{jl}^{new} = w_{jl}^{old} + \alpha \vartheta_j(\tau) y^l(\tau - 1)$$

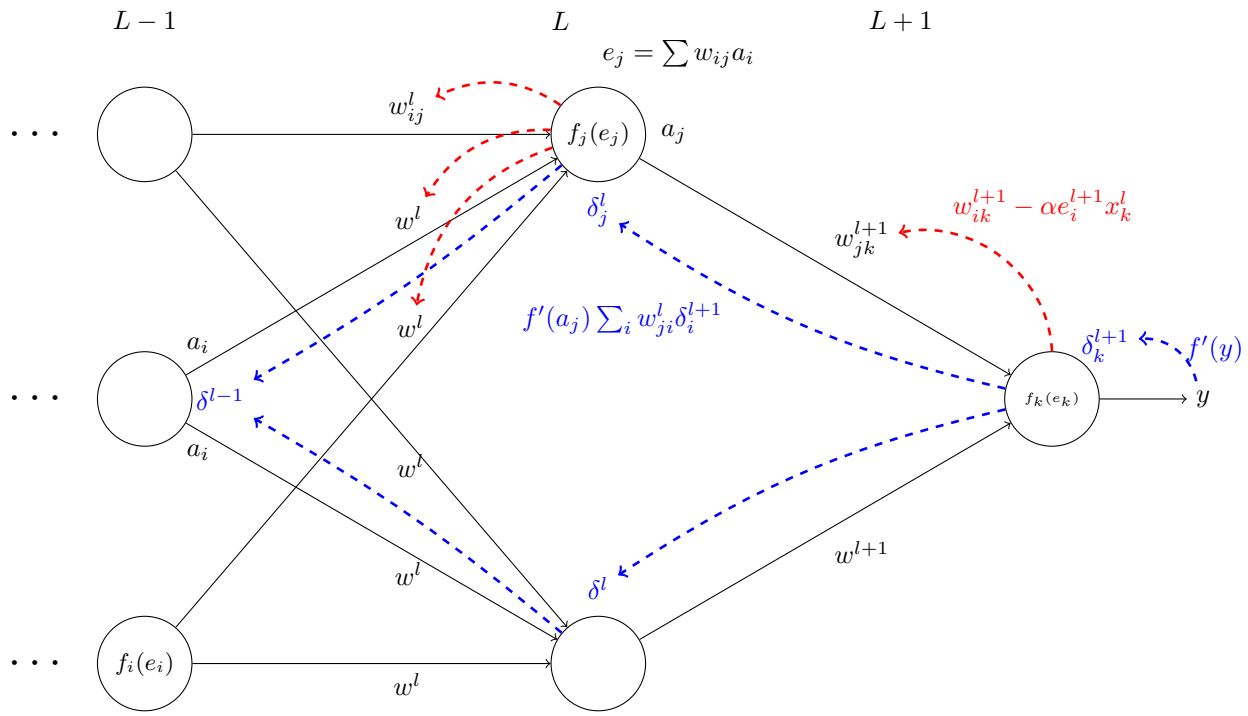

Figura 2.1: Gradiente descendente

2.2 Revisión de la literatura

Diseño de la experimentación

3.1 Lógica del diseño

3.2 Metodología del experimento

3.3 Consideraciones generales

()

Bibliografía

Anderson, J. A. (1968). A memory storage model utilizing spatial correlation functions. *Kybernetik*, *5*, 113–119.

Anderson, J. A. (1970). *Two models for memory organization using interacting traces.* doi: 10 .1016/0025-5564(70)90147-1

Baldi, P. (1995, Jan). Gradient descent learning algorithm overview: a general dynamical systems perspective. *IEEE Transactions on Neural Networks*, *6*(1), 182-195. doi: 10.1109/72.363438

Bengio, Y., Simard, P., y Frasconi, P. (1994, Mar). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157-166. doi: 10.1109/ 72.279181

Chollet, F. (2015). *Keras.* `https://github.com/fchollet/keras`. GitHub.

Cruz, P. (2011). *Inteligencia artificial con aplicaciones a la ingeniería*. Barcelona: Marcombo.

Fritsch, J. (1996). *Modular neural networks for speech recognition* (Masters Thesis). KIT.

Hebb, D. (2002). *The organization of behavior: A neuropsychological theory*. Taylor & Francis.

Hochreiter, S. (1991). *Untersuchungen zu dynamischen neuronalen netzen* (Diploma thesis). Institut f. Informatik, Technische Univ. Munich.

Hochreiter, S. (1998, abril). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, *6*(2), 107–116. Descargado de `http://dx.doi.org/10.1142/S0218488598000094` doi: 10.1142/ S0218488598000094

Hochreiter, S., y Schmidhuber, J. (1997, noviembre). Long short-term memory. *Neural Comput.*, *9*(8), 1735–1780. Descargado de `http://dx.doi.org/10.1162/neco.1997.9.8.1735` doi: 10.1162/neco.1997.9.8.1735

Hopfield, J. J. (1982, 1 de abril). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, *79*(8), 2554–2558.

Huang, G.-B., Saratchandran, P., y Sundararajan, N. (2005, Jan). A generalized growing and pruning rbf (ggap-rbf) neural network for function approximation. *IEEE Transactions on Neural Networks*, *16*(1), 57-67. doi: 10.1109/TNN.2004.836241

ichi Amari, S. (1993). Backpropagation and stochastic gradient descent method. *Neurocomputing*, *5*(4–5), 185 - 196. Descargado de `http://www.sciencedirect.com/science/article/ pii/0925231293900060` doi: http://doi.org/10.1016/0925-2312(93)90006-O

Kohonen, T. (1972, April). Correlation matrix memories. *Computers, IEEE Transactions on*, *C-21*(4), 353-359. doi: 10.1109/TC.1972.5008975

Kohonen, T. (1974, April). An adaptive associative memory principle. *Computers, IEEE Transactions on*, *C-23*(4), 444-445. doi: 10.1109/T-C.1974.223960

Marvin Minsky, S. A. P. (1987). *Perceptrons: An introduction to computational geometry* (Expanded ed.). The MIT Press.

McCulloch, W., y Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115-133. doi: 10.1007/BF02478259

McCulloch, W. S., y Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115–133. doi: 10.1007/BF02478259

Newell, A. (1969). Perceptrons: An introduction to computational geometry. marvin minsky and seymour papert. m.i.t. press, cambridge, mass., 1969. vi + 258 pp., illus. cloth, $12; paper$,4.95. *Science*, *165*(3895), 780–782. Descargado de `http://science.sciencemag.org/content/165/3895/780` doi: 10.1126/science.165.3895.780

Rosenblatt, F. (1957). *The perceptron—a perceiving and recognizing automaton* (Report n.º 85-460-1). Cornell Aeronautical Laboratory.

Rosenblatt, F., y Laboratory, C. A. (1958). *The perceptron: a theory of statistical separability in cognitive systems (project para)*. Cornell Aeronautical Laboratory.

Rumelhart, D. E., Hinton, G. E., y Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Steinbuch, K. (1961). Die lernmatrix. *Kybernetic*, *1*, 36–45.

Steinbuch, K., y Piske, U. A. W. (1963, Dec). Learning matrices and their applications. *Electronic Computers, IEEE Transactions on*, *EC-12*(6), 846-862. doi: 10.1109/PGEC.1963.263588

y. Liang, N., b. Huang, G., Saratchandran, P., y Sundararajan, N. (2006, Nov). A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on Neural Networks*, *17*(6), 1411-1423. doi: 10.1109/TNN.2006.880583