Eficacia de diferentes reglas hebbianas en el Aprendizaje Supervisado
Efficacy of different Hebbian rules in Supervised Learning

Fernando Javier Aguilar Canto [1]

[1] Facultad de Matemáticas, Universidad Autónoma de Yucatán,
Calle 17 no. 106 F x 26 y 28
Colonia Chuburná, Mérida, Yucatán.
Correo: pherjev@gmail.com

Fecha de recepción: 15 de julio de 2019
Fecha de aceptación: 30 de agosto de 2019

Resumen. Desde 1949, numerosos neurocientíficos han estudiado los principios implícitos del llamado aprendizaje asociativo (o hebbiano) entre neuronas. Varios algoritmos basados en Hebb (incluyendo la Regla de Hebb Simple, la Regla de la Covarianza y la Regla de Oja) han sido propuestos como modelos teóricos del cambio de la fuerza de conexión sináptica (o pesos) y son aplicables para entrenar redes neuronales en lugar de usar algoritmos artificiales como Descenso de Gradiente. En este artículo, se evaluó el rendimiento de los algoritmos basados en Hebb en diferentes contextos. La simple aplicación de los mencionados métodos fue incapaz de alcanzar la exactitud de los métodos existentes. No obstante, algunos algoritmos combinados con el uso de k-celdas arrojaron mejores resultados comparativos, obteniendo 97.6% de exactitud en tareas como el problema de clasificación de círculos mientras que otros métodos arrojaron 97.4% o menos, haciendo de esta estrategia una alternativa adecuada a algunos métodos clásicos.

Palabras clave: Aprendizaje hebbiano, Aprendizaje Supervisado, Redes Neuronales Artificiales, Redes Neuronales Biológicas, Regla de Oja.

Sumary. Since 1949, many neuroscientists have studied the underlying principles of the so-called associative learning (or Hebbian learning) between neurons. Several Hebbian-based algorithms (including the Simple Hebb Rule, the Covariance Rule, and the Oja Rule) have been proposed as theoretical models of the change of synaptic strength (or weights) and can be applied to train Neural Networks rather than using artificial algorithms such as Gradient Descend. However, these Hebbian-bases algorithms still not being considered in many Supervised Learning tasks. In this paper, we evaluated the performance of Hebbian-based algorithms in different contexts. The simple application of the mentioned algorithms was unable to reach the accuracy of current methods. Nevertheless, some combined algorithms with k-cells yielded better comparative results, getting 97.6% of accuracy in tasks such as circle classification problem whereas other methods yielded 97.4% or less, making this strategy a suitable alternative for some classical methods.

Keywords: Hebbian learning, Supervised learning, Artificial Neural Networks, Biological Neural Networks, Oja Rule.

1 Introducción

En 1949, Donald Hebb [1] postuló que el encendido simultáneo de dos neuronas incrementaba la fuerza de conexión entre ellas, idea que recibió una fuerte evidencia experimental en la década de los 60's tras el descubrimiento del fenómeno conocido como Potenciación a Largo Plazo (*Long-Term Potentiation,* o LTP) [2]. Posteriores avances neurocientíficos ayudaron a la formulación de modelos más precisos de aprendizaje asociativo entre neuronas, entre los cuales se incluyen la Regla de la Covarianza, la Regla de Oja, entre otros (*v. infra*).

Las mencionadas reglas de aprendizaje asociativo anteriores (*learning rules*) consisten de modelos de actualización de los pesos de una red neuronal fija, por lo que estructuralmente no difieren de los modelos de redes neuronales artificiales (*artificial neural networks* o ANN), las cuales han sido empleadas para resolver problemas de clasificación, correspondiendo el principal objetivo de este trabajo. Sin embargo, en contraste, las ANN emplean métodos como Descenso del Gradiente para la actualización de pesos [3].

Dos métodos son propuestos para implementar a las reglas de Hebb y compararlas con los métodos existentes: el primero consiste en utilizar una red de una capa (1-NN) cuyas entradas son las intensidades de cada pixel y entrenar a una sola época para emular condiciones de entrenamiento en tiempo real. Otro método propuesto es segmentar el espacio de datos (vectores en $\mathbb{R}^m$) en *m*-celdas que corresponden a las entradas de la red. Este método no es directamente aplicable para imágenes debido a la alta dimensión de los vectores.

1.1 Redes Neuronales de una capa

Consideremos el escenario más simple de redes neuronales consistente en una sola capa [2] con *n* neuronas cuyos valores de activación son $u_1,\cdots,u_n$, conectadas a una neurona con valor de activación v, donde para cada i, $u_i, v \in \{0,1\}$ (modelo discreto) o bien $u_i, v \in [0,1]$ (modelo continuo). Sea w_i el peso de la *i*-ésima entrada[4]. Una versión lineal del modelo de tasa de disparo [5] está dada por la ecuación diferencial (Regla de la Suma):

$$\tau_r \frac{dv}{dt} = -v + \mathbf{w} \cdot \mathbf{u} \tag{1}$$

donde la constante de control $\tau_r > 0$. Observaciones experimentales indican que $\tau_r \approx 0$, lo cual implica que

$$v = \mathbf{w} \cdot \mathbf{u} \tag{2}$$

Lo cual implica que el estado de la neurona con activación *v* es la suma ponderada de sus entradas. Sin embargo, para cumplir que *v* se encuentre en el mismo espacio que las entradas, debemos exigir que $v = f(\mathbf{w} \cdot \mathbf{u})$, donde *f* es una función de activación. Para el caso discreto (el cual se estudiará), podemos considerar a la función de activación umbral

$$u_\theta(x) = \begin{cases} 1 & x > \theta \\ 0 & else \end{cases} \tag{3}$$

1.2 Regla de Hebb Simple y Regla de la Covarianza

La formulación original de la regla de Hebb implica básicamente que si $v \approx 1$ y $u_i \approx 1$ entonces

$$\frac{dw_i}{dt} > 0 \tag{4}$$

Una formulación simple de lo anterior es la siguiente ecuación diferencial:

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u} \tag{5}$$

donde $\tau_w > 0$ es una constante de control. La Regla de Hebb Simple considera únicamente el proceso de la LTP. Sin embargo, la evidencia experimental también encontró un fenómeno análogo conocido como Depresión a Largo Plazo (LTD, por sus siglas en inglés), el cual consiste en una depresión de la fuerza sináptica si la baja actividad presináptica ocurre simultáneamente a una alta actividad postsináptica. Una aproximación a este fenómeno queda descrita por la siguiente ecuación diferencial, conocida como la Regla de la Covarianza:

$$\tau_w \frac{d\mathbf{w}}{dt} = (v - \theta_v)\mathbf{u} \tag{6}$$

donde $0 < \theta_v < 1$ es un umbral que determina el cambio de LTD a LTP [5].

[4] En todos los modelos se considerará que los pesos son una variable continua.

1.3 Regla de Oja

Las reglas de aprendizaje anteriores presentan el problema del crecimiento no acotado de los pesos. Esto podría causar problemas de *overfitting* al tener pesos con valores distantes. Dos formas para reducir este problema son el empleo de un umbral θ adaptativo y la normalización de los pesos. Ejemplo de la segunda solución es la Regla de Oja [6], cuya expresión queda descrita por la ecuación diferencial siguiente [5]:

(7)

2 Estado de Arte

En la actualidad, a pesar de que el aprendizaje asociativo entre neuronas (modelado con las reglas hebbianas) sea fundamental en las modernas neurociencias, ha encontrado un reducido espacio en el contexto actual de Machine Learning [7]. Esto puede obedecer, entre otras cosas, a la efectividad de métodos basados en el Gradiente y derivados, incluyendo las Redes Neuronales Convolucionales (CNN), en estructuras de Redes de Alimentación Hacia Adelante (*Feedforward Neural Networks*, FNN). En estructuras de Redes Recurrentes (RNN), dado que la existencia de un algoritmo de entrenamiento efectivo continúa en discusión, se ha propuesto el empleo de reglas de aprendizaje como la regla Bienenstock-Cooper-Munro (BCM) y anti-Oja para resolver problemas de clasificación y regresión [8]. En el contexto de FNN, destaca [7] quien emplea un Aprendizaje Hebbiano Adaptativo con una estructura de varias capas (incluyendo convolución) con el fin de resolver problemas de clasificación de imágenes.

2.1 Eficacia de los clasificadores lineares clásicos en el conjunto MNIST

Con el fin de probar la eficacia de los algoritmos propuestos se utilizó la base de datos Modified-NIST [9]. Si bien el problema de clasificar dígitos de MNIST puede ser eficazmente resuelto mediante CNN [10,11], para esta ocasión consideraremos arquitecturas similares, que son las redes de una sola capa, los cuales son clasificadores lineales. De acuerdo con la comparación de LeCun *et al* [12], una red neuronal de una capa (1-NN) sin procesamiento alcanzó un porcentaje de error de 12% sin preprocesamiento, mientras que aplicando *deskewing* se logró un error de 8.4%. Otros clasificadores lineales lograron un porcentaje de error de 7.6%.

3 Metodología

Las ecuaciones planteadas en la Sección 1 inducen a un algoritmo de clasificación por Aprendizaje Supervisado, aunque el paso hacia el mismo no es trivial. En general si consideramos un conjunto de pares ordenados $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ donde $\mathbf{x}$ es el vector de entrada y y la salida deseada, queremos encontrar una función que disminuya el error de clasificación.

Las Reglas de Hebb Simple, Covarianza y de Oja se implementaron como métodos para actualización de pesos en cinco diferentes datasets: tres de ellos son artificiales y generados por la librería Scikit, uno es de caracteres impresos con diferente tipografía pero escasa variabilidad y el último es la base de datos MNIST. Los tres primeros datasets se evaluaron mediante la Regla de Hebb Simple frente a un clasificador de kernel lineal, la Regresión Logística y k-Vecinos Más Cercanos (k-NN). Los últimos dos *datasets* se evaluaron mediante todas las reglas mencionadas y una red neuronal de una capa con una época.

3.2 Algoritmo de actualización de pesos

Para actualizar discretamente los pesos, en general para cada $i = 1, \ldots, n$ (n es el número de datos) se efectúa

$$w \rightarrow w + \alpha \Delta w \tag{8}$$

El valor $\Delta \mathtt{w}$ varía de acuerdo a la Regla empleada. En el caso de la Regla de Hebb Simple se tiene $\Delta \mathtt{w} = y\mathtt{x}$; para la Regla de Covarianza $\Delta \mathtt{w} = (y - \theta_v)\mathtt{x}$; para la Regla de Oja se tiene $\Delta \mathtt{w} = y\mathtt{x} - \beta v^2 \mathtt{w}$. A continuación, se proponen dos métodos para aplicar dichas reglas de aprendizaje en datasets específicos.

3.3 Clasificación por *m*-celdas del Espacio Vectorial

Consideremos al espacio vectorial $\mathbb{E}^m$ (en este caso, $\mathbb{R}^m$), donde *m* es el número de entradas de cada vector $\mathtt{u}_i$ de datos con etiqueta v_i . Podemos particionar el espacio vectorial en *m*-celdas. Para ello, asumiremos que sólo es necesario un número finito de celdas donde se encuentran los posibles vectores de entrada. Cada celda I_j puede considerarse como una entrada “sensorial” de una red de una capa, tal que si $\mathtt{u}_i \in I_j$ entonces la neurona correspondiente $x_j = 1$ y $x_q = 0$ para toda $q \neq j$ y $y = v_i$. De esta forma, aplicando la anterior regla biyectiva construimos un nuevo conjunto de datos $\{(\mathtt{x}_1, y_1), \cdots, (\mathtt{x}_n, y_n)\}$, el cual es entrenable utilizando actualización hebbiana. Notemos que las celdas más pequeñas permiten una separación más fina de los conjuntos etiquetados, sin embargo también se requiere un mayor número de datos puesto que algunas celdas pequeñas podrían quedar sin incidencia (*overfitting*).

3.4 Clasificación de patrones morfológicos

Para el caso de patrones tipo imagen (como caracteres) podemos considerar la imagen binarizada (para el caso discreto) y aplicar la estructura de red neuronal de una capa con $a \times b$ entradas donde (a, b) es la dimensión de la imagen y una capa de salida de 10 neuronas en el caso de los caracteres, procediendo de forma muy similar a la clasificación de caracteres con una red neuronal de una capa. Posteriormente se actualizan los pesos de acuerdo con el algoritmo analizado.

3.4.1 Reglas Escaladas

Una observación realizada *a posteriori* en la aplicación de las Reglas de Hebb y de Oja, es la importancia de reescalar el vector de pesos por el inverso multiplicativo de la suma de sus entradas de la forma siguiente:

$$\mathtt{w} \leftarrow \frac{1}{\sum w_i} \mathtt{w} \tag{9}$$

4 Resultados experimentales

Para evaluar los diferentes algoritmos basados en reglas de aprendizaje se utilizaron tanto datos reales como generados. Para el primer caso, se generó un dataset de 2500 imágenes de caracteres impresos y con escasa variabilidad, el 90% formaron el conjunto de entrenamiento y el 10% el de prueba. El segundo dataset, como se mencionó, corresponde a los caracteres manuscritos MNIST. Para este primer caso se empleó la clasificación de patrones morfológicos. En el segundo caso se generaron tres conjuntos de datos mediante la librería *scikit-learn* de Python, los cuales consisten en el Problema de clasificación de *blobs,* el Problema de clasificación de *moons* y el Problema de clasificación de círculos. Como se mencionó en la introducción, los entrenamientos se evaluaron a una sola época con el fin de emular las condiciones en tiempo real en las que los datos se reciben en los sensores.

Los resultados se resumen en la tabla 1. En general, la regla de Hebb Simple y de la Covarianza presentaron la misma efectividad tanto en la base de datos generada como e MNIST. Se observó que la Regla de Oja ofrecía mejores resultados que la aplicación simple de la Regla de Hebb, no así la regulada, que ofreció resultados considerablemente mejores. En contextos de escasa varianza, la Regla de Hebb Regulada ofrece un buen rendimiento, no así en entornos más variables como en la base de datos MNIST donde su efectividad fue de apenas 73.27%, aunque mayor que los otros algoritmos hebbianos. Esto contrasta con el 91.39% alcanzado usando Descenso de Gradiente en una época.

Tabla 1. Eficacia de los diferentes algoritmos y reglas de aprendizaje

Algoritmo de Aprendizaje	1		2		3		4		MNIST	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Tamaño de la muestra	900000	100000	900000	100000	900000	100000	2250	250	60000	10000
Hebb Simple	0.99993	0.97143	0.99952	0.99836	0.97936	0.97605	0.878	0.871	.4099	0.439
Hebb Regulada	-	-	-	-	-	-	0.952	0.991	.7541	0.7327
Covarianza	-	-	-	-	-	-	0.878	0.871	.4099	0.439
Oja (β = 0.01)	-	-	-	-	-	-	0.884	0.871	.5621	0.6031
Oja Regulada (β = 0.01)	-	-	-	-	-	-	.9524	0.991	0.698	0.6715
Kernel lineal	0.99999	0.99999	0.98799	0.98669	0.9473	0.94124	-	-	-	-
Regresión Logística	0.99996	0.99997	0.88332	0.8834	0.47865	0.47699	-	-	-	-
k-NN	0.99996	0.99996	0.99943	0.99929	0.98026	0.97446	-	-	-	-
Gradient Descent (1-NN)	-	-	-	-	-	-	.9938	1	0.878	0.9139

No obstante, para el caso de los métodos de clasificación por *m*-celdas el panorama fue considerablemente mejor[5]. Para las tres tareas mostró un comportamiento eficaz, inclusive en problemas complejos mostró un mejor rendimiento que en los otros métodos. Los resultados muestran el promedio de diez pruebas realizadas. Su eficacia es comparable con k-NN.

5 Conclusiones y futuras líneas de investigación

Los resultados anteriores muestran que la aplicación trivial de los algoritmos basados en Hebb para resolver problemas complejos en una red de una capa no manifiesta un rendimiento similar a los métodos tradicionales, aunque la Regla de Hebb Regulada mostró un comportamiento eficaz en datos de escasa varianza. ¿A qué se debe el hecho que métodos artificiales como Descenso de Gradiente hayan tenido resultados mejores que los modelos de aprendizaje neuronal? La segunda parte del trabajo parece indicar que el problema se debe a la carencia de una red adecuada, logrando un efectivo empleo de la regla de Hebb con las *m*-celdas para clasificación en $\mathbb{R}^2$. Su generalización constituye una tarea pendiente y podría posibilitar la solución del problema de clasificación en imágenes usando reglas de Hebb.

Agradecimientos

Principalmente quisiera agradecer al Dr. Carlos Francisco Brito Loeza por el seguimiento de la investigación.

Referencias

[1] Hebb, D.O. *The Organization of the Behavior. A Neuropsycological Theory*. John Wiley & Sons, Inc. Estados Unidos de América. 1949

[2] Hartz, B. P., y L.C. B. Rønn. *NCAM in Long-Term Potentiation and Learning. Structure and Function of the Neural Cell Adhesion Molecule NCAM*. Springer, New York, E.U.A., 257-270. 2010.

[3] Cömert, Z. y A.F. Kocamaz. *A study of artificial neural network training algorithms for clasification of cardiotocography signals*. Bitlis Ergen University. En *Journal of Science and Technology,* vol. 7, no. 2, 93-103. 2017.

[4] Aghdam, H.H. y E.J. Heravi. *Guide to Convolutional Neural Networks*. New York, NY: Springer. Vol. 10, p. 978-3. 2017.

[5] Dayan, P. y L.F. Abbott. *Theoretical Neuroscience. Computational and Mathematical Modeling of Neural Systems*. The MIT Press. Massachusetts, Estados Unidos de América. 2001.

[6] Choe, Y. *Hebbian Learning*. En *Encyclopedia of Computational Neuroscience*. D. Jaeger y R. Jung (editores). Springer. Nueva York, Estados Unidos de América. 2015.

[7] Wadhwa, A. y U. Madhow *Bottom-up Deep Learning using the Hebbian Principle*. 2016. Disponible en https://www.ece.ucsb.edu/wcsl/people/aseem/Aseem_stuff/hebbian_preprint.pdf

[5] El código programado en Python puede consultarse en el repositorio https://github.com/Pherjev/hebbian-m-cells

[8] Yusoff, M.H, J. Chrol-Cannon y Y. Jin. *Modeling neural plasticity in echo state networks for classification and regression*. En *Information Sciences,* 364, 184-196. 2016.
[9] LeCun, Y., C. Cortes y C.J.C. Burges. *The MNIST Database of handwritten digits* Disponible en http://yann.lecun.com/exdb/mnist/.
[10]Cireşan, D.C. *et al*. *High-performance neural networks for visual object classification*. IDSIA / USI-SUPSI. 2011. Disponible en https://arxiv.org/pdf/1102.0183.pdf
[11] Cireşan, D.C., U. Meier y J. Schmidhuber. *Multi-column deep neural networks for image classification*. IDSIA / USI-SUPSI. 2012 Disponible en https://arxiv.org/pdf/1202.2745.pdf
[12]LeCun, Y. *et al*. *Gradient-based learning applied to document recognition*. *Proceedings of the IEEE*. 86 (11), 2278-2324. 1998.