

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



**ANÁLISIS DE LA EFICIENCIA DEL ENTRENAMIENTO BASADO EN
SIMULATED ANNEALING EN REDES NEURONALES PROFUNDAS**

Felipe Alberto Reyes González

Profesor Guía: Victor Parada

Tesis de grado presentada en conformidad a los
requisitos para obtener el grado de Magíster en
Ingeniería Informática

Santiago, Chile

2017

© Felipe Alberto Reyes González- 2017



• Algunos derechos reservados. Esta obra está bajo una Licencia Creative Commons Atribución-Chile 3.0. Sus condiciones de uso pueden ser revisadas en:

<http://creativecommons.org/licenses/by/3.0/cl/>.

TABLA DE CONTENIDOS

1. Introducción	1
1.1. Antecedentes y motivación	1
1.2. Descripción del problema	3
1.3. Solución propuesta	3
1.3.1. Características de la solución	3
1.3.2. Propósito de la solución	3
1.4. Objetivos y alcances del proyecto	3
1.4.1. Objetivo general	3
1.4.2. Objetivos específicos	4
1.4.3. Alcances	4
1.5. Metodología y herramientas utilizadas	4
1.5.1. Metodología de trabajo	4
1.5.2. Herramientas de desarrollo	5
Referencias bibliográficas	6

ÍNDICE DE TABLAS

1.1. Especificaciones del equipo 5

ÍNDICE DE FIGURAS

CAPÍTULO 1. INTRODUCCIÓN

1.1 ANTECEDENTES Y MOTIVACIÓN

Las redes neuronales (*Neural Networks*, NN) son sistemas de procesamiento de información que basan su estructura en una analogía de las redes neuronales biológicas. Consisten en un conjunto de elementos de procesamiento simple llamados nodos, estos nodos están dispuestos en una estructura jerárquica y conectadas entre si por un valor numérico llamado peso que, mediante un proceso de entrenamiento, varia su valor.

La actividad que una neurona realiza en una NN es simple. El proceso consiste en ponderar las entradas de la neurona por los pesos de las conexiones de la neurona para luego ser sumadas y entregadas a la función de activación asociada a la neurona (McCulloch y Pitts, 1943). La salida corresponderá a la respuesta que la neurona genera a la entrada que se presentada.

Las neuronas con una función de activación umbral, fueron estudiadas por Rosenblatt (1962) quién las denominó *perceptrón*. El modelo básico se compone de 3 capas conectadas consecutivamente. La primera capa corresponde a la capa de entrada, que recibe el patrón de entrada a clasificar. La segunda capa contiene las neuronas de asociación o detección de características. Y la tercera capa es la capa de salida, que contiene las neuronas que reconocen los patrones. La limitación de los perceptrones se debe a capacidad de clasificación basado en el umbral, lo que no permite clasificar patrones mas complejos.

El conjunto de n neuronas se llamará capa, y una NN puede estar compuesta de una o más capas. Cada capa estará compuesta por una cantidad de neuronas que no necesariamente será la misma para todas las capas, y estarán dispuesta en forma consecutiva de tal manera que las capas se conecten unas con otras y siempre hacia adelante. La primera capa, la de entrada, recibirá un patrón que será entregado a las distintas neuronas que la capa posea. Cada neurona de la capa de entrada procesará los datos y generará una salida que servirá de entrada para la capa siguiente, repitiendo el proceso para cada una de las capas de la NN hasta llegar a la capa de salida, en cuyo caso la salida representará la respuesta de la red, concretando así el ciclo.

Las NN han sido utilizadas para la clasificación de entradas, y han sido diseñados diversos métodos para entrenar la red y que los pesos se adapten de tal manera que la salida de la red sea representativa de la salida esperada, a esto método de entrenamiento se le llama supervisado. Dentro de los métodos de entrenamiento existentes se encuentra el método del gradiente descendente, el método de Newton, el gradiente conjugado, el método quasi-Newton, o el algoritmo Levenberg-Marquardt. El más utilizado es el método del gradiente, que consiste en

actualizar los pesos de las distintas neuronas en función de la dirección contraria al gradiente de la función de activación, logrando minimizar el error.

El gradiente descendente es el mas simple de los métodos. Este requiere información del gradiente del vector, por lo tanto corresponde a un método de primer orden. El método comienza en un punto inicial y, mientras el criterio de parada no se cumpla, la solución inicial se moverá en la dirección contraria al gradiente a razón de una tasa de aprendizaje establecida. La tasa de aprendizaje es quién decide cuan largo será el paso, si éste es muy grande, puede que la convergencia a la solución sea superada por un paso muy grande, en cambio, una tasa de aprendizaje muy pequeña puede demorar la convergencia.

El método de Newton es un método de segundo orden, pues hace uso de la matriz Hessiana. El objetivo de este método es encontrar una mejor dirección de entrenamiento mediante el uso de la segunda derivada de la función de pérdida. Considera una aproximación cuadrática de la función en la solución inicial utilizando una expansión de la serie de Taylor. De esta manera, el método de Newton permite mover la solución inicial en función de la inversa de la matriz Hessiana, que pasará a ser la dirección de entrenamiento de Newton.

El método del gradiente del conjugado fue propuesto por Fletcher y Reeves (1964), utiliza sucesivas direcciones conjugadas basadas en el gradiente y el residuo. Si la función objetivo es cuadrática y la dirección de búsqueda es minimizada exactamente en cada iteración, el método converge de forma cuadrática. Leonard y Kramer (1990) han utilizado el método para entrenar redes neuronales como alternativa a la retropropagación. Los pesos de una red se actualizan de acuerdo con una búsqueda unidireccional en la dirección de descenso $S(n)$ de la siguiente forma:

$$\Delta W(n) = \lambda(n)S(n) \quad (1.1)$$

Y la dirección de descenso $S(n)$ se calcula a partir del gradiente de iteraciones pasadas y presentes como muestra la ecuación 1.2

$$S(n) = -G(n) + \frac{\|G(n)\|}{\|G(n-1)\|}S(n-1) \quad (1.2)$$

Para que las direcciones de descenso permanezcan conjugadas, la búsqueda unidireccional, en cada iteración, tiene que ser llevada con alta precisión. Sin embargo, después de muchas iteraciones, las direcciones podrían llegar a ser casi paralelas. Para superar esta dificultad, Fletcher y Reeves (1964) sugieren volver a iniciar el procedimiento igualando la dirección de descenso al gradiente en cada iteración $(t + 1)$, donde (t) es el número total de pesos en la red.

1.2 DESCRIPCIÓN DEL PROBLEMA

La retropropagación basa su funcionamiento en la regla de la cadena para poder calcular los gradientes, y a medida que el error se propaga hacia la capa de entrada de la red el gradiente comienza a disminuir su valor por cada capa que atraviesa. Esto significa que el gradiente disminuirá de manera exponencial, lo que representa un problema para una red de muchas capas, ya que las capas mas cercanas a la capa de entrada necesitarán más tiempo para ser entrenadas.

1.3 SOLUCIÓN PROPUESTA

1.3.1 Características de la solución

Mediante el uso de el algoritmo *simulated annealing* se busca analizar la eficiencia que la NN alcanza en una red neuronal profunda frente a otros métodos de aprendizaje.

1.3.2 Propósito de la solución

El propósito de la solución es aportar en el campo de las redes neuronales y la clasificación de datos, proporcionando un análisis comparativo de la convergencia de distintas redes.

1.4 OBJETIVOS Y ALCANCES DEL PROYECTO

1.4.1 Objetivo general

Evaluar el desempeño del algoritmo *simulated annealing* y su efecto sobre entrenamiento de redes neuronales profundas.

1.4.2 Objetivos específicos

Los objetivos establecidos para el presente trabajo son descritos a continuación

1. Definir las reglas de aprendizaje a implementar.
2. Construir los conjuntos de datos de entrada y salida a analizar.
3. Establecer los parámetros de las redes neuronales para la experimentación.
4. Entrenar las redes con los distintos conjuntos de datos.
5. Establecer las conclusiones del trabajo.

1.4.3 Alcances

1.5 METODOLOGÍA Y HERRAMIENTAS UTILIZADAS

1.5.1 Metodología de trabajo

Considerando el aspecto investigativo del trabajo, se considera la utilización del método científico. Entre las actividades que componen la metodología, Sampieri (2006) describe los siguientes pasos para desarrollar una investigación:

- Formulación de la hipótesis: Las redes neuronales que adolecen del desvanecimiento del gradiente se ven beneficiadas por el uso del algoritmo *simulated annealing* en la convergencia.
- Marco teórico: Una revisión de la literatura donde se aborda el problema planteado, para situarse en el contexto actual de los problemas. Se describirán redes neuronales que buscan solucionar el mismo problema.
- Diseño de la solución: Se deberá diseñar el experimento para generar los datos que permitan sustentar las comparaciones entre las distintas redes. Diseñar y ejecutar el experimento basado en entradas equivalentes.

- Análisis y verificación de los resultados: Los resultados se analizarán considerando los valores de convergencia de los distintos métodos.
- Presentación de los resultados: Se presentarán tablas que describan los resultados obtenidos y que se consideren pertinentes.
- Conclusiones obtenidas en el desarrollo de la investigación.

1.5.2 Herramientas de desarrollo

Para el desarrollo y ejecución de los experimentos se utilizará un equipo con las siguientes características

Sistema Operativo	Solus 2017.04.18.0 64-bit
Procesador	Intel® Core™ i5-2450M CPU @ 2.50GHz x 4
RAM	7.7Gb
Gráficos	Intel® Sandybridge Mobile
Almacenamiento	935.6 GB

Tabla 1.1: Especificaciones del equipo

El software que se utilizará es:

- Plataforma de desarrollo: Atom.
- Lenguaje de programación: Python.
- Sistema de redes neuronales: Keras API (Chollet, 2015).
- Herramienta ofimática: \LaTeX .

REFERENCIAS BIBLIOGRÁFICAS

- Chollet, F. (2015). *Keras*. <https://github.com/fchollet/keras>. GitHub.
- Fletcher, R., y Reeves, C. M. (1964, 1 de febrero). Function minimization by conjugate gradients. *The Computer Journal*, 7(2), 149–154. Descargado de <http://dx.doi.org/10.1093/comjnl/7.2.149> doi: 10.1093/comjnl/7.2.149
- Leonard, J., y Kramer, M. (1990). Improvement of the backpropagation algorithm for training neural networks. *Computers & Chemical Engineering*, 14(3), 337 - 341. Descargado de <http://www.sciencedirect.com/science/article/pii/0098135490870706> doi: [http://dx.doi.org/10.1016/0098-1354\(90\)87070-6](http://dx.doi.org/10.1016/0098-1354(90)87070-6)
- 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
- Rosenblatt, F. (1962). *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Spartan Books.
- Sampieri, R. (2006). *Metodología de la investigación*. México: McGraw Hill.