

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



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

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 CONTENIDO

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Antecedentes y motivación . . . . .	1
1.2	Descripción del problema . . . . .	2
1.3	Solución propuesta . . . . .	2
1.3.1	Características de la solución . . . . .	2
1.3.2	Propósito de la solución . . . . .	2
1.4	Objetivos y alcances del proyecto . . . . .	3
1.4.1	Objetivo general . . . . .	3
1.4.2	Objetivos específicos . . . . .	3
1.4.3	Alcances . . . . .	3
1.5	Metodología y herramientas utilizadas . . . . .	4
1.5.1	Metodología de trabajo . . . . .	4
1.5.2	Herramientas de desarrollo . . . . .	4
	<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>6</b>

# ÍNDICE DE TABLAS

Tabla 1.1 Especificaciones del equipo . . . . . 4

## ÍNDICE DE ILUSTRACIONES

# CAPÍTULO 1. INTRODUCCIÓN

## 1.1 ANTECEDENTES Y MOTIVACIÓN

Las redes neuronales artificiales (*Artificial Neural Networks*, NN) han sido ampliamente estudiadas y ampliamente utilizadas en muchas aplicaciones de la inteligencia artificial. El problema durante el proceso de aprendizaje de las NN es descrito como un problema de minimización de una función de error, la que depende de los pesos que conforman la red (Rumelhart, Hinton, y Williams, 1986a). Este problema de optimización tiene la desventaja de ser no lineal, no convexo, además de tener mas de un mínimo local. Para solventar este problema se han desarrollado diversos algoritmos (Grippo, 1994; Jacobs, 1988; V. P. Plagianakos, Magoulas, y Vrahatis, 2002; Rumelhart, Hinton, y Williams, 1986b; V. Plagianakos, Sotiropoulos, y Vrahatis, 1998) y su rendimiento varía según el problema a resolver.

El enfoque clásico para el entrenamiento de las NN es la aplicación de algoritmos basados en el gradiente como la retropropagación (Rumelhart y cols., 1986b). El algoritmo de retropropagación minimiza la función de error mediante la dirección de descenso más pronunciada. Aunque la función de error disminuye rápidamente en la dirección del gradiente negativo, la retropropagación es generalmente ineficiente y poco fiable (Gori y Tesi, 1992) debido a la superficie de error. Además, su rendimiento se ve afectado por parámetros que deben ser especificados por el usuario, pues no existe una base teórica para escogerlos (Nguyen y Widrow, 1990). Dichos parámetros tienen una importancia crucial en el buen funcionamiento del algoritmo, por lo que el diseñador está obligado a seleccionar parámetros como los pesos iniciales de la NN, la topología de la red y la tasa de aprendizaje. En diversas investigaciones (Cauchy, 1847; Grippo, 1994; V. Plagianakos y cols., 1998; V. P. Plagianakos y cols., 2002) ha quedado demostrado que pequeñas modificaciones en estos valores influyen en el rendimiento de la NN. Para proporcionar una convergencia más rápida y estable se han desarrollado diversas variaciones y alternativas a la retropropagación.

Varias técnicas se han propuestos, como la adaptación de un término de momento (Jacobs, 1988; Rumelhart y cols., 1986b) o de una tasa variable de aprendizaje (Jacobs, 1988; Vogl, Mangis, Rigler, Zink, y Alkon, 1988). Magoulas, Vrahatis, y Androulakis (1997); V. Plagianakos y cols. (1998) propusieron dos técnicas para evaluar en forma dinámica la tasa de aprendizaje sin el uso de alguna heurística o alguna función adicional y las evaluaciones de gradiente. El primero se basó en el algoritmo de Barzilai y Borwein (Barzilai y Borwein, 1988) que adapta la tasa de aprendizaje sin evaluar la matriz Hessiana; mientras que el segundo utiliza estimaciones de la constante de Lipschitz, eplotando la información local de la superficie de error y los pesos posteriores (Magoulas

y cols., 1997). Hay evidencias (Magoulas y cols., 1997; V. P. Plagianakos y cols., 2002; V. Plagianakos y cols., 1998) que han demostrado que la retropropagación con algoritmos que adaptan la velocidad del aprendizaje son robustas y tienen un buen rendimiento para el entrenamiento de NN.

## **1.2 DESCRIPCIÓN DEL PROBLEMA**

La retropropagación basa su funcionamiento en multiplicaciones sucesivas basadas en el error 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 redes profundas, ya que las capas mas cercanas a la capa de entrada necesitarán más tiempo para ser entrenadas.

El método de aprendizaje basado en *simulated annealing* permite la actualización de los pesos de la red sin mermar la capacidad de adaptación de los pesos. El método supone una alternativa efectiva a los métodos tradicionales de aprendizaje para la convergencia de los métodos debido a la independencia que otorga a la actualización de los pesos de las distintas capas..

## **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 tales como SGD y RMSPROP.

### **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**

En ésta sección se presenta el objetivo general, los objetivos específicos además del alcance y limitaciones de la presenta investigación.

### **1.4.1 Objetivo general**

Evaluar el desempeño del algoritmo *simulated annealing* y su efecto sobre el entrenamiento de redes neuronales profundas en comparación con otros métodos.

### **1.4.2 Objetivos específicos**

Los objetivos establecidos para el presente trabajo son descritos a continuación

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

### **1.4.3 Alcances**

1. Se analizará la misma arquitectura con diferentes reglas de aprendizaje.
2. Los conjunto de datos para el entrenamiento a utilizar son los propuestos en (Morse y Stanley, 2016).



## 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.
- 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:

- Lenguaje de programación: Python.
- Sistema de redes neuronales: Keras API (Chollet, 2015).
- Herramienta ofimática:  $\text{\LaTeX}$ .

## REFERENCIAS BIBLIOGRÁFICAS

- Barzilai, J., y Borwein, J. M. (1988). Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1), 141. doi: 10.1093/imanum/8.1.141
- Cauchy, A.-L. (1847, 18 de octubre). Méthode générale pour la résolution des systèmes d'équations simultanées. *Compte Rendu des S'éances de L'Acad'emie des Sciences XXV, S'erie A*(25), 536–538.
- Chollet, F. (2015). *Keras*. <https://github.com/fchollet/keras>. GitHub.
- Gori, M., y Tesi, A. (1992, Jan). On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1), 76-86. doi: 10.1109/34.107014
- Grippo, L. (1994). A class of unconstrained minimization methods for neural network training. *Optimization Methods and Software*, 4(2), 135-150. doi: 10.1080/10556789408805583
- Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1(4), 295 - 307. doi: [https://doi.org/10.1016/0893-6080\(88\)90003-2](https://doi.org/10.1016/0893-6080(88)90003-2)
- Magoulas, G. D., Vrahatis, M. N., y Androulakis, G. S. (1997). Effective backpropagation training with variable stepsize. *Neural Networks*, 10(1), 69 - 82. Descargado de <http://www.sciencedirect.com/science/article/pii/S0893608096000524> doi: [https://doi.org/10.1016/S0893-6080\(96\)00052-4](https://doi.org/10.1016/S0893-6080(96)00052-4)
- Morse, G., y Stanley, K. O. (2016). Simple evolutionary optimization can rival stochastic gradient descent in neural networks. En *Proceedings of the genetic and evolutionary computation conference 2016* (pp. 477–484). New York, NY, USA: ACM. Descargado de <http://doi.acm.org/10.1145/2908812.2908916> doi: 10.1145/2908812.2908916
- Nguyen, D., y Widrow, B. (1990, June). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. En *1990 ijcnn international joint conference on neural networks* (p. 21-26 vol.3). doi: 10.1109/IJCNN.1990.137819
- Plagianakos, V., Sotiropoulos, D., y Vrahatis, M. (1998). Automatic adaptation of learning rate for backpropagation neural networks. *Recent Advances in Circuits and Systems*, 337.
- Plagianakos, V. P., Magoulas, G. D., y Vrahatis, M. N. (2002, Nov). Deterministic nonmonotone strategies for effective training of multilayer perceptrons. *IEEE Transactions on Neural Networks*, 13(6), 1268-1284. doi: 10.1109/TNN.2002.804225

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

Rumelhart, D. E., Hinton, G. E., y Williams, R. J. (1986b). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. En D. E. Rumelhart, J. L. McClelland, y C. PDP Research Group (Eds.), (pp. 318–362). Cambridge, MA, USA: MIT Press. Descargado de <http://dl.acm.org/citation.cfm?id=104279.104293>

Sampieri, R. (2006). *Metodología de la investigación*. México: McGraw Hill.

Vogl, T. P., Mangis, J. K., Rigler, A. K., Zink, W. T., y Alkon, D. L. (1988). Accelerating the convergence of the back-propagation method. *Biological Cybernetics*, 59(4), 257–263. doi: 10.1007/BF00332914