



**Sexto**  
**Congreso Nacional de**  
**Riego, Drenaje y Biosistemas**  
COMEII- 2021 / Hermosillo, Sonora



**Artículo: COMEII-21005**

*Hermosillo, Son., del 9 al 11 de junio de 2021*

## **ALGORITMO ADAM EN LA INTELIGENCIA ARTIFICIAL**

**Abraham Rojano Aguilar<sup>1\*</sup>; Raquel Salazar Moreno<sup>1</sup>; Luis Miranda <sup>2</sup>, Waldo Ojeda Bustamante<sup>1</sup>**

<sup>1</sup> Universidad Autónoma Chapingo, Chapingo,  
C.P. 56230, Estado de México, México.

<sup>2</sup> Hochschule für Nachhaltige Entwicklung Eberswalde (HNEE)  
Alfred-Möller-Str. 116225 Eberswalde, Germany.

abrojano@hotmail.com - 5534386912 (\*Autor de correspondencia)

### **Resumen**

El algoritmo Adam es una de las herramientas actuales más usadas para los procesos de optimización. Su propuesta data tiene menos de una década, pero se ha posicionado entre los más usados debido a su competitividad con otros métodos. El algoritmo recoge las ventajas de otros algoritmos basados en descenso rápido y momentos adaptándose muy bien para procesos estocásticos.

La inteligencia artificial entendida como el proceso cognitivo de resolver problemas, al principio requiere de una base de datos con buena calidad no solo para el entrenamiento, sino también para verificar si las decisiones que se toman con la inteligencia natural no solo coinciden con la artificial, sino también en la búsqueda de mejores alternativas. En el trabajo se muestran dos aplicaciones, empezando por la búsqueda del mínimo de una función complicada relacionada con Rosenbrock y una de identificación de figuras.

**Palabras claves:** Descenso rápido, redes neuronales, convoluciones, softmax, Relu.



## Introducción

El método Adam es un poderoso y sencillo algoritmo que permite hacer búsquedas de con el menor tiempo y/o número de pasos o de la mejor forma posible o conocida hasta nuestros tiempos. Dichas características caen dentro del campo de la optimización y es una ruta de conexión natural con las aplicaciones de inteligencia artificial donde el objetivo es hacer decisiones correctas, lógicas o razonables en el menor tiempo posible (Kingma, et.al, 2015).

La inteligencia artificial actualmente ha invadido numerosos campos de la vida, sin embargo, la idea primaria de resolver problemas o tomar decisiones razonables basados en algunas premisas de la lógica sigue siendo válida en cualquier escenario ya sea de identificación, clasificación a nivel de sistemas expertos, reconocimiento del lenguaje, de voz, visión o en robótica. En estos casos, los esquemas lógicos son sujetos a un intenso proceso de cálculo donde generalmente con una base de datos se entrena, se verifica, y se evalúan los componentes de una red neuronal, así como sus resultados.

En la inteligencia artificial se sigue el camino heurístico empezando con entradas como bases de datos y terminando con unas funciones con salidas como decisiones o extrapolaciones. En este camino la base de datos es ordenada, refinada, limpiada y preparada para los propósitos específicos de las lógicas de redes. Numerosas bases existen en la red, siendo las de Facebook muy heterogéneas y unas de las que crecen en forma impresionante; las cuales generan desafíos constantes para problemas de identificación, y clasificación; debido a eso, para propósitos educativos recurrimos a bases de datos como mnist, cifar10, y grandes bases de datos entre otras.

Los tipos o arquitecturas de redes neuronales han crecido por su gran cantidad de variantes. Siendo uno de los propósitos describir el estado del arte, entonces recurrimos a cnn para el propósito de este trabajo. La idea básica de cnn data de raíces biológicas descritas en los 50s, pero la implementación computacional empieza en los 90s con la base de datos mnist, y posteriormente cifar10, cifar 100, imagenet con más de 14 millones de imágenes y miles de clases, entre otras, hasta llegar a una red como alexnet(Krizhevsky, A. et. al., 2017).

## Materiales y métodos

El camino tradicional de buscar valores extremos en problemas que podían escribirse como funciones reales de variable real sin restricciones sobre un dominio infinito era el de derivar la función, igualar a cero y obtener el punto donde la función alcanzaba un extremo o un punto silla. Un camino alternativo para problemas con restricciones, desde los tiempos de Euler y luego utilizado el método de los multiplicadores de Lagrange, ha sido muy exitoso. La enorme cantidad de problemas abordados para casos de desigualdades, o de información incompleta ha llevado a una formalización de la teoría de optimización para problemas descritos con funciones continuas, como es el método simplex, métodos no suaves o no lineales.



La aparición de problemas donde algunos de los componentes son de alguna forma aleatorios, entonces se han creado estrategias que siguen aportando nuevos caminos de solución. La idea de los momentos data de 1887 por Chebyshev en la demostración del teorema central de limite, y luego continuada en diversas aplicaciones por Pearson después de 1895, en problemas donde es común contar con algunos datos que pueden ser usados para generar modelos que le den sentido a la descripción, explicación o entendimiento de la información. Así, el algoritmo adam hace uso de funciones reales de variable real o vectorial, con algunos componentes parciales totales de forma aleatoria, no solo para el dominio de definición sino también para las funciones de costo, de pérdida u objetivo. Una vez que la búsqueda empieza debe hacerse con un arranque semilla para llevar a cabo el proceso de continuo descenso hasta llegar a la meta óptima.

El algoritmo adam hace uso de un conjunto de estrategias heurísticas, donde primero para resolver un problema se plantean cuatro pasos empezando por comprender el problema, concebir un plan de solución, ejecutarlo y analizar los resultados. Como cualquier otro problema de la ciencia la solución puede no existir o ser múltiple, sin embargo, cuando entendemos que los problemas de minimización en forma analógica requieren un proceso de descenso, ahora es necesario trasladarlo a un lenguaje simbólico con un plan de solución con posibles y múltiples estrategias, partiendo de la función a optimizar con algún criterio como de likelihood, bootstrap, o el de las funciones logísticas de las redes neuronales, entre otras.

## Aspectos básicos

La explicación generalizada recurre a una analogía con la bola que desciende rápidamente por una superficie hasta llegar a los mínimos locales o globales, donde el problema de las oscilaciones es algo que debe controlarse y es el motivo de esta presentación (Ruder, S., 2017; Duchi et. al. 2011)

Dicha idea con mucho sentido intuitivo data desde los años 60s donde Polyak propuso para el proceso de optimización de primer orden una secuencia  $w_0, w_1, \dots, w_n$ , tal que

$$w_{t+1} = w_t - \alpha_t \nabla_{w_t} f \quad (1)$$

Hasta que

$$|w_{t+1} - w_t| \leq \varepsilon_t \quad (2)$$

Cuando  $f$  es convexa, la ecuación (1) converge en (2). Por analogía, observemos que en las aproximaciones de primer orden por diferencia finita en el método de Euler, el parámetro  $\alpha_t$  corresponde con el número de Courant y  $-\nabla_{w_t} f$  al sentido contrario del gradiente, lo cual le da el nombre al método de una forma natural. En el campo de la inteligencia artificial el parámetro  $\alpha_t$  corresponde al factor de aprendizaje.

Ahora la idea es mejorar el método de descenso rápido como el de momento clásico introduciendo el parámetro  $\mu$  para controlar o modificar los movimientos de zig-zag.



$$w_{t+1} = w_t + v_{t+1} \quad (3)$$

$$v_{t+1} = \mu v_t - \alpha_t \nabla_{w_t} f \quad (4)$$

Y una actualización del gradiente por Nesterov genera el método de gradiente acelerado

$$v_{t+1} = \mu v_t - \alpha_t \nabla_{w_t + \mu v_t} f \quad (5)$$

El método Adam es un método de dos momentos adaptivos en el campo de la optimización estocástica propuesto por los autores Diederik P. Kingma de OpenAI y Jimmy Lei Ba de University of Toronto, pero también se puede usar en la optimización determinista. El método fue presentado en un artículo de la conferencia ICLR 2015 con el título "Adam: A method for Stochastic Optimization". Los valores sugeridos por los autores son  $\alpha_t$ ,  $\mu_1$  y  $\mu_2$  como  $10^{-3}$ , 0.9, y 0.999, respectivamente.

El procedimiento generalizado para alimentar la solución del algoritmo Adam es como sigue para un dominio cuadrado de  $2 \times 2$ , como

```
#Definir la semilla con un generador de números aleatorios,
Semilla()
# define el rango de entrada
bounds = asarray([-1.0, 1.0], [-1.0, 1.0])
# define el total de iteraciones
n_iter = 60
# tamaño de paso
alpha = 0.002
# factor de gradiente promedio
mu_1 = 0.9
# factor de gradiente al cuadrado
mu_2 = 0.999
# Realiza la búsqueda de descenso con la función adam
solucion = adam(objective, derivative, bounds, n_iter, alpha, mu_1, mu_2)
```

y el algoritmo adam o de los momentos adaptivos, tal como

$$w_{t+1} = w_t - \alpha_t \frac{\tilde{u}}{\sqrt{\tilde{v} + \epsilon}} \quad (6)$$

$$u_{t+1} = \mu_1 u_t + (1 - \mu_1) \nabla_{w_t} f \quad (7)$$

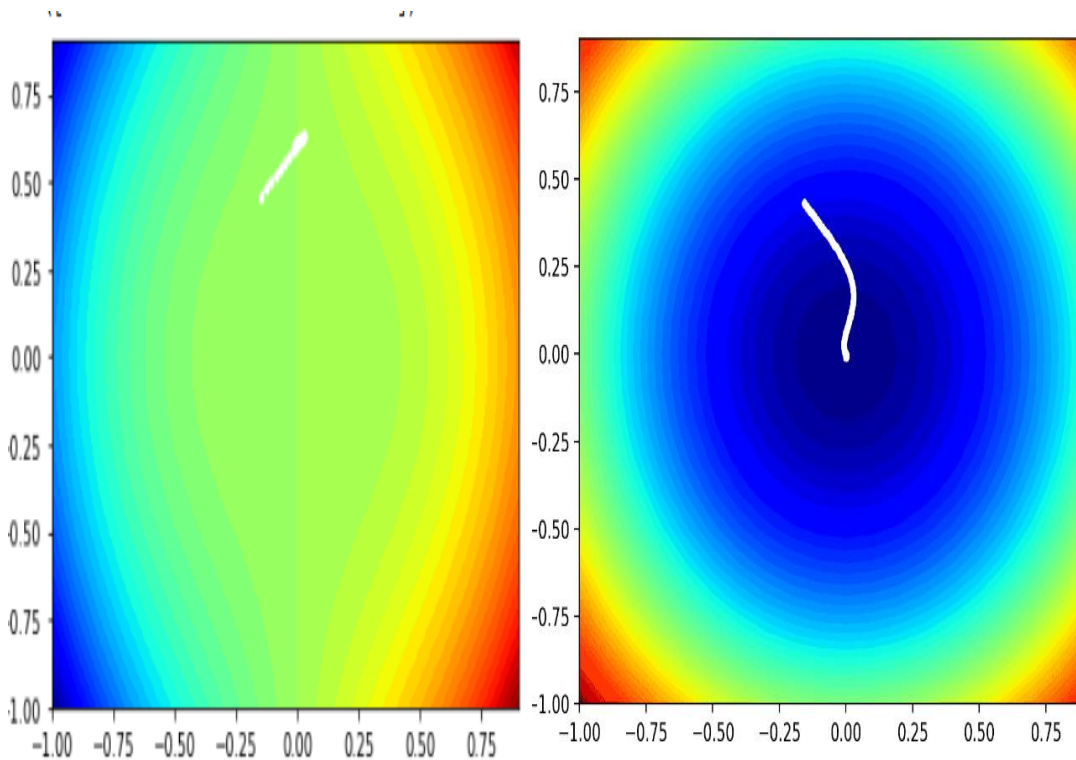
$$v_{t+1} = \mu_2 v_t + (1 - \mu_2) (\nabla_{w_t} f)^2 \quad (8)$$

$$(9) \quad \tilde{u}_{t+1} = \frac{u_{t+1}}{(1-\mu_1)}$$

$$(10) \quad \tilde{v}_{t+1} = \frac{v_{t+1}}{(1-\mu_2)}$$

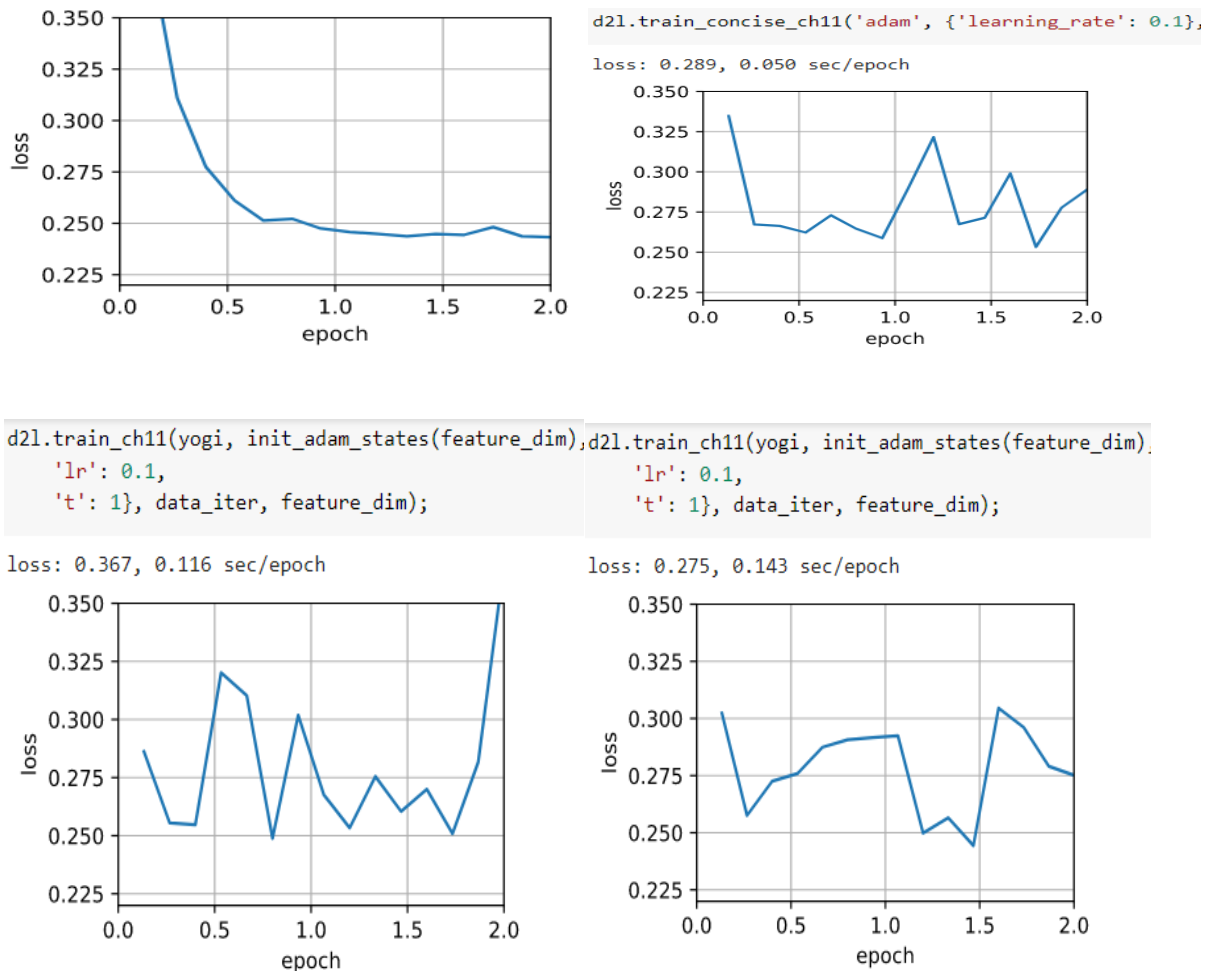
## Resultados

El algoritmo es descrito con numerosas ventajas que van desde la simplicidad, eficiencia computacional, poca memoria, invariancia, objetivos no estacionarios, y con parámetros intuitivos fáciles de calibrar hasta su aplicación en problemas deterministas de gran complejidad como el caso la función de la banana o de Rosenbrock. El algoritmo ha sido probado exitosamente no solo en condiciones estocásticas sino también determinista como es el caso del problema complicado de Rosenbrock entre otras. El algoritmo es programado en python 3.7 y corrido en google colab, pero jupiter y la aplicación ide de spyder también es viable.



**Figura 1.** Lado izquierdo, problema complicado de una función de utilidad  $F(x,y) = x(x^{**2} + y^{**2}.0)$  en un dominio de  $2 \times 2$ ; lado derecho, problema de  $F(x,y) = x^{**2}.0 + y^{**2}.0$  en un dominio de  $2 \times 2$  con optimización determinista.

El algoritmo Adam ha sido probado exitosamente, también dentro de las funciones de utilidad encontradas en las aplicaciones en las siguientes bases de datos. La base de datos mnist consiste de 10 patrones de dígitos del 0 al 9 hechos a mano y es un reto atractivo para principiantes no solo en la construcción y programación de la red, sino que también para un entrenamiento con errores menores al 5%. (<https://findanyanswer.com/what-is-mnist-in-machine-learning>)



**Figura 2.** Resultados obtenidos del programa adam.ipynb con una base de datos mxnet. Arriba, izquierda es con el algoritmo Adam y factor de aprendizaje 0.01. Arriba, derecha, es con factor de aprendizaje 0.1. Abajo, izquierda, es con corrección convergente yogi y aprendizaje 0.1 y abajo, derecha, con corrección convergente yogi y aprendizaje 0.1

La base de datos cifar-10 es un conjunto de 60000 imágenes a color pequeñas de 32x32 píxeles, ordenadas en 10 clases excluyentes entre seres vivos y máquinas de transporte donde se sugiere entrenarse con 5000 y verificar con 1000. Big-data-bases es una base de datos de ajedrez comercial de 8.4 millones de juegos desde 1560 hasta 2020 con la



calidad estándar definida por el motor ChessBase con clasificación de aperturas en mas de 100000 posiciones con acceso a torneos, medio juego, y finales. Dicho sistema requiere al menos Pentium PC, 2 GB RAM, Windows 10, 8 o 7, y los motores Fritz 13, 14, 15, 16 o ChessBase 15, 14, 13 or 12 y drive DVD, con un costo de 80 dólares(<https://shop.chess.com/big-database-2021/>)

En medio del proceso de modelacion, existen las redes neuronales que conectan los datos de la base, por medio de un muestreo, un entrenamiento, y finalmente una decisión. Las decisiones pueden ser validadas, y contrastadas con otra parte de la base de datos con algún criterio o función de utilidad, y para el caso de una pobre correlación o errores grandes se deben buscar caminos de mejora. Ahora bien, la construcción de la red para mnist o el uso de una ya probada como es el caso de grandes bases de datos en ajedrez. Alexnet ha sido propuesta desde 2012 como un perfeccionamiento de lenet5, y ganado desafíos de reconocimiento visual. Dicha red tiene 8 capas, con 5 de muestreo y 3 completamente conectadas, usando funciones de activación Relu, excepto en la última que generalmente es softmax.

Un resumen numérico de la ruta de red neuronal alexnet es desde la entrada como: [227x227x3] ENTRADA; [55x55x96] CONV1: 96 11x11 filtro a paso 4, bloque 0; [27x27x96] MAX POOL1: 3x3 filtro a paso 2; [27x27x96] NORM1: capa normalizadora; [27x27x256] CONV2: 256 5x5 filtros a paso 1, bloque 2; [13x13x256] MAX POOL2: 3x3 filtros a paso 2; [13x13x256] NORM2: capa de normalización; [13x13x384] CONV3: 384 3x3 filtros a paso 1, bloque 1; [13x13x384] CONV4: 384 3x3 filtros a paso 1, bloque 1; [13x13x256] CONV5: 256 3x3 filtros a paso 1, bloque 1; [6x6x256] MAX POOL3: 3x3 filtros a paso 2; [4096] FC6: 4096 neuronas; [4096] FC7: 4096 neuronas; y finalmente [1000] FC8: SALIDA 1000 neuronas que definen las clases(Krizhevsky, A. et.al., 2017).

Cualquier tamaño diferente de las imágenes debe ser ajustado con un escalamiento, para poder empatar con la red y la salida es fija con 1000 clases o categorías. La función de utilidad o perdida original fue la divergencia de Jensen-Shannon, pero ahora es la entropía cruzada para regresión; el error cuadrado medio para la segmentación de imágenes o refuerzo de imágenes. El proceso de minimización del error es hecho por medio del descenso rápido con tres opciones, SGDM que es descenso con gradiente estocástico con optimizador de momentum, RMSProp que es el optimizador de propagación de la raíz del error medio cuadrático, y Adam el optimizador del momento adaptivo.

## Conclusiones

El método de adam tiene grandes ventajas por su simpleza, pero sin embargo existen variantes muy competitivas para casos especiales donde los mínimos locales no coinciden con los globales. En la analogía de la bola que desciende el método Adam está muy relacionado con la bola pesada que baja con fricción. Esto ha generado toda una familia de métodos cercanos al algoritmo adam, como son adamax, adaGrad, RMSprop y adaDelta que en algunas circunstancias pueden ser más efectivos.



La gran flexibilidad de este algoritmo parece muy idónea para los problemas de inteligencia artificial que está basada en modelos de caja negra con redes neuronales, las cuales forman árboles de nodos y capas entre las entradas y salidas. Los criterios de acercamiento entre las salidas de la red con los objetivos deseados generan un modelo de discriminación el cual debe estar constantemente optimizándose no solo con un algoritmo de optimización como el de adam, sino que adicionalmente, el modelo requiere una construcción de la función de pérdida, una medida de la distancia y la precisión deseada. Aplicaciones en inteligencia artificial por medio de redes neuronales para identificar o clasificar imágenes es muy común como se puede constatar en el uso de la función objetivo definida en la librería scikit, keras, pytorch y tensorflow de python para mxnet, y en los ensayos de regresión logística del negativo de log likelihood con imágenes de MNIST e IMDB.

### Agradecimiento

Se agradece a las personas involucradas desde la logística, revisores, y ponentes en este congreso por su paciencia y dedicación para sacar adelante el conjunto de tareas necesarias todas cuesta arriba para llegar a un resultado óptimo en tiempo y forma.

### Referencias Bibliográficas

Duchi, J et al. (2011) *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization*. Available at: <http://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf>

Kingma, D and Ba, J . (2015) *Adam: A method for Stochastic Optimization*. Available at: <https://arxiv.org/pdf/1412.6980.pdf>

Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017-05-24). "[ImageNet classification with deep convolutional neural networks](#)" (PDF). *Communications of the ACM*. **60** (6): 84–90. [doi:10.1145/3065386](https://doi.org/10.1145/3065386). ISSN 0001-0782. S2CID 195908774.

Ruder, S. (2017) *An overview of gradient descent optimization algorithms*. Available at: <https://arxiv.org/pdf/1609.04747.pdf>