

Apuntes de la clase del 25 de setiembre de 2025

Kevin Carranza Jimenez
 Escuela de Ingeniería en Computación
 Tecnológico de Costa Rica
 kcaranza@estudiantec.cr

Abstract—This document summarizes the lecture held on September 25, 2025, which included the presentation of the company Skild.ia, focused on applying artificial intelligence algorithms for robot control. It also provides a review of the previous lecture, covering neural networks from logistic regression to their application in binary classifiers using multinomial expressions. The session then introduces the perceptron model, defined as a linear regression with a Hinge Loss function. It is emphasized that a single perceptron cannot solve non-linear functions, although multiple perceptrons can be combined to achieve this. Finally, the multilayer perceptron is introduced as a form of deep neural network with biological inspiration.

Index Terms—Skild, Regresión lineal, multinomial red neuronal, perceptrón.

I. INTRODUCTION

EN el desarrollo del curso, las clases recientes han abordado los fundamentos de las redes neuronales y su evolución hacia modelos más complejos. La sesión del 25 de septiembre de 2025 incluyó como tema de interés un video de la empresa Skild.ia, que tiene como objetivo utilizar algoritmos de inteligencia artificial para controlar robots y estos puedan emplear cualquier tarea, trayendo la inteligencia artificial al mundo físico. También se incluyó tanto la revisión de conceptos previamente estudiados, entre ellos la regresión logística para desarrollar expresiones multinomiales y con multiples capas de estas, desarrollar redes neuronales. También la introducción del perceptrón, considerado el punto de partida para las redes neuronales profundas.

II. SKILD.IA

Skild AI es una startup emergente dedicada al desarrollo de una inteligencia artificial de propósito general para el control de robots de múltiples tipos (humanoides, brazos robóticos, plataformas de locomoción, etc.).

La propuesta central de Skild AI es crear un “cerebro robótico omni-corporal” —denominado Skild Brain— que permita que un mismo modelo de IA controle diferentes cuerpos robóticos sin necesidad de reentrenamientos específicos para cada hardware.

Un aspecto clave de su diseño es la capacidad de adaptación a fallos o cambios drásticos en la morfología del robot: cuando un robot pierde una extremidad o sufre un daño, el modelo puede reorganizar su control para seguir operando, aprovechando la experiencia aprendida previamente [1].

III. REDES NEURONALES

Una Red Neuronal Artificial (RNA) es un modelo computacional inspirado en la estructura y funcionamiento del cerebro humano, compuesto por nodos (neuronas artificiales) organizados en capas y conectados entre sí mediante pesos. Estas redes aprenden patrones complejos a partir de datos de entrada a través de un proceso iterativo de ajuste de pesos, permitiendo resolver tareas de clasificación, predicción y reconocimiento en diversos dominios [2].

A. Clasificador de MNIST

MNIST es un dataset con 60k muestras de números del 0 al 9 en un solo canal. En el resumen de la clase se da a entender que se está intentando desarrollar un clasificador utilizando este dataset. Que cada una de estas imágenes está compuesta por un grupo de píxeles.

B. Regresión Logística

En el resumen de la clase anterior se menciona que para el clasificador de MNIST se comenzaba tratando de hacer una clasificación binaria respecto a la imagen. Hasta el momento de esta clase el único algoritmo conocido para desarrollar esta clasificación es la regresión logística. Para esto se pasan la información de cada uno de los píxeles de la imagen como entrada para la regresión logística. La situación es que el problema no puede ser resuelto con una regresión logística únicamente, si no con una regresión logística multinomial, ya que requerimos 10 clases y la regresión logística solo permite 1.

C. Multinomial

Para esto, al problema requerir 10 clases, se desarrollan 10 regresiones logísticas, una por cada clase y a cada una se le pasa como entrada la información de los píxeles de la imagen, por lo que una de las regresiones logísticas dará mayor probabilidad que las demás. Y en este punto tenemos en la Figura 1 una arquitectura que ya podría llamarse red neuronal, aunque todavía faltaría agregar una siguiente capa para poder resolver problemas no lineales.

También se menciona en el resumen de la clase anterior que en lugar de calcular cada regresión lineal de forma vectorial, cambiamos los vectores por matrices para hacer 1 sola operación y no N. Donde N es el tamaño de la capa siguiente utilizando conceptos de álgebra lineal. En cada una de las filas será representado las neuronas para la siguiente capa y las entradas las columnas.

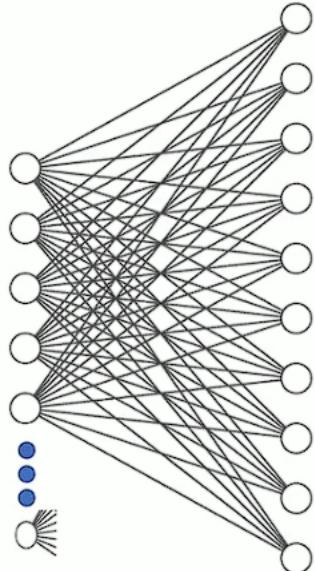


Fig. 1: Primer Red Neuronal.

El resumen de la clase anterior concluye definiendo algunas características de las redes neuronales, las cuales son que al no ser lineales nos permite atacar problemas complejos, está compuesta por capas, estas capas son el hiper parámetro de la red neuronal y es importante que sean diferenciables. Si la red neuronal se puede derivar se puede optimizar y que en cada capa hay neuronas.

IV. EL PERCEPTRÓN

El perceptrón es uno de los modelos más simples de red neuronal artificial, propuesto por Frank Rosenblatt en 1958. Consiste en una unidad de procesamiento que recibe un conjunto de entradas ponderadas, las combina linealmente y aplica una función de activación para producir una salida binaria. Su objetivo principal es clasificar patrones linealmente separables. Aunque limitado para problemas no lineales, constituye la base conceptual de arquitecturas más complejas como el perceptrón multicapa y las redes neuronales profundas [3].

A. Invierno de la AI

El invierno de la inteligencia artificial hace referencia a períodos históricos en los que las expectativas generadas alrededor de la investigación en IA no se cumplieron, provocando una disminución drástica en la financiación, el interés académico y el desarrollo industrial en este campo. Durante estos períodos, los avances en IA se ralentizaron debido a limitaciones tecnológicas, falta de resultados prácticos y críticas hacia la viabilidad de los enfoques predominantes. Se reconocen principalmente dos inviernos de la IA: el primero a mediados de los años 1970, y el segundo a finales de los años 1980 hasta principios de los 1990 [4].

En 1969, Marvin Minsky y Seymour Papert publicaron el libro *Perceptrons*, en el que señalaron limitaciones fundamentales del perceptrón simple. Entre los problemas destacados,

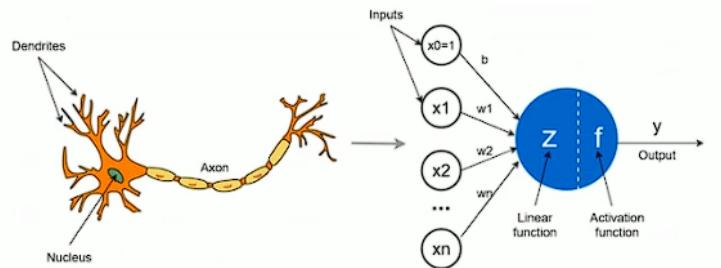


Fig. 2: Inspiración biológica de la red neuronal.

señalaron que este modelo no podía resolver funciones no linealmente separables, siendo el ejemplo clásico la función lógica XOR. Además, advirtieron sobre su limitada expresividad computacional y su escasa capacidad para generalizar en problemas más complejos, lo que contradecía las expectativas iniciales de que los perceptrones pudieran resolver tareas de visión y reconocimiento de patrones. Estas observaciones demostraron que, aunque los perceptrones eran útiles para ciertos problemas lineales, su aplicación práctica era muy limitada. El impacto de estas críticas fue significativo, contribuyendo al primer invierno de la inteligencia artificial, hasta que el desarrollo del perceptrón multicapa y el algoritmo de retropropagación permitieron superar estas restricciones [5].

B. Inspiración Biológica

Las redes neuronales artificiales se inspiran en el funcionamiento de las neuronas del cerebro humano, donde cada neurona recibe señales de múltiples conexiones sinápticas, las procesa y genera una respuesta que se transmite a otras neuronas. De manera análoga, en las redes neuronales artificiales, cada nodo o “neurona” recibe entradas ponderadas, aplica una función de activación y transmite su salida a las siguientes capas, reproduciendo de forma simplificada el procesamiento distribuido y paralelo del sistema nervioso biológico. Esta inspiración biológica se ilustra en la Figura 2, donde se muestra la correspondencia entre una neurona biológica y su modelo artificial.

C. Función de Activación

En regresión logística se llama función no-lineal (sigmoid). Esta depende de si la señal activa o no la neurona. Dependiendo de la intensidad de la señal que se haya recibido, esta dejará pasar la información, la bloqueará o la transformará y existen varias funciones de activación.

1) *Función Sigmoidal:* La función sigmoidal transforma un valor de entrada en un rango entre 0 y 1, lo que permite interpretarla como una probabilidad. Su desventaja principal es la saturación de gradientes en valores extremos, lo que dificulta el entrenamiento en redes profundas [6].

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

2) *Función Tangente Hiperbólica (tanh)*: La tangente hiperbólica es similar a la sigmoide, pero su rango va de -1 a 1, lo que permite que las salidas estén centradas en cero. Esto ayuda a mitigar algunos problemas de gradientes en comparación con la sigmoide, aunque aún puede sufrir de saturación [6].

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

3) *Función ReLU (Rectified Linear Unit)*: La función ReLU es una de las más utilizadas en redes neuronales modernas. Define la salida como 0 para valores negativos y como la propia entrada para valores positivos. Es computacionalmente eficiente y mitiga en gran parte el problema del desvanecimiento del gradiente, aunque puede presentar el problema de “neurona muerta” [7].

$$f(x) = \max(0, x)$$

4) *Función Leaky ReLU*: La función Leaky ReLU es una variante de la ReLU que permite pequeños valores negativos en la salida (usualmente multiplicados por una constante pequeña, como 0.01). Esto evita el problema de neuronas muertas al asegurar un gradiente no nulo para entradas negativas [8].

$$f(x) = \begin{cases} x & \text{si } x \geq 0 \\ \alpha x & \text{si } x < 0 \end{cases}$$

5) *Función Softmax*: La función Softmax convierte un vector de valores reales en una distribución de probabilidad, donde cada valor queda entre 0 y 1 y la suma total es igual a 1. Se utiliza principalmente en la capa de salida de clasificadores multiclas [6].

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad \text{para } i = 1, \dots, K$$

V. PERCEPTRÓN MULTICAPA

El Perceptrón Multicapa (MLP, por sus siglas en inglés) es una arquitectura fundamental dentro de las redes neuronales artificiales. Está compuesto por una capa de entrada, una o más capas ocultas y una capa de salida. A diferencia del perceptrón simple, que solo puede resolver problemas linealmente separables, el MLP utiliza funciones de activación no lineales en sus neuronas ocultas, lo que le permite aproximar funciones complejas y resolver problemas no lineales. Su entrenamiento se realiza comúnmente mediante el algoritmo de retropropagación (backpropagation), el cual ajusta los pesos de las conexiones minimizando el error entre la salida predicha y la deseada. Esta arquitectura constituye la base de los modelos modernos de aprendizaje profundo [6], [9].

A. Maldición de dimensionalidad

A mayor cantidad de dimensiones, aumenta la complejidad, a su vez, aumentando la computabilidad y se vuelve más complicado encontrar patrones. Para esto existen algoritmos de deducción de dimensiones como el PCA.

1) *PCA*: El Análisis de Componentes Principales (PCA, por sus siglas en inglés: Principal Component Analysis) es un método estadístico ampliamente utilizado para la reducción de dimensionalidad, que transforma un conjunto de variables posiblemente correlacionadas en un nuevo conjunto de variables no correlacionadas denominadas componentes principales. El procedimiento consiste en centrar los datos, calcular la matriz de covarianza, obtener sus autovalores y autovectores, y seleccionar los vectores asociados a los mayores autovalores para proyectar los datos en un subespacio de menor dimensión que conserva la mayor varianza posible de la información original [10], [11], [12].

B. Comportamiento Jerárquico

Los humanos aprenden cosas simples para transformarlo en algo más complejo, tal es el caso del MLP conformado por múltiples regresiones lineales, de lo cual se obtienen ganancias exponenciales en algunas funciones, como polinomios, la composición de funciones que permite reusar funciones simples otras de orden superior y que mediante una representación compacta, en la que pocos pesos se pueden modelar funciones complejas, como por ejemplo, una red neuronal que se aproxime a otra.

VI. CONCLUSION

La clase permitió la comprensión de los fundamentos de las redes neuronales, resaltando su estructura jerárquica al final y las motivaciones biológicas que inspiran su arquitectura. A partir del análisis del perceptrón y de sus limitaciones, se introdujo la necesidad de arquitecturas más complejas, como el MLP, que posibilitan la resolución de problemas no lineales. Esta sesión trató tanto el potencial como los desafíos de las redes neuronales, entre ellos la maldición de la dimensionalidad y la importancia de un diseño acorde al problema en cuestión en términos de capas y neuronas. Así, la clase proporcionó las bases para comprender las arquitecturas modernas de aprendizaje profundo.

REFERENCES

- [1] K. Wiggers, “Skild ai emerges from stealth with \$300m to build a general-purpose ai brain for robots.” *TechCrunch*, Sep. 2025, accessed: 2025-10-02. [Online]. Available: <https://techcrunch.com/2025/09/16/skild-ai-emerges-from-stealth-with-300m-to-build-a-general-purpose-ai-brain-for-robots/>
- [2] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Prentice Hall, 2009.
- [3] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- [4] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.
- [5] M. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.
- [6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [7] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010.
- [8] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

- [9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [10] I. T. Jolliffe and J. Cadima, “Principal component analysis: a review and recent developments,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.
- [11] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [12] J. Shlens, “A tutorial on principal component analysis,” *arXiv preprint arXiv:1404.1100*, 2014.