|  |  |  |
| --- | --- | --- |
| **Nombre**: Ruben Alejandro Deambrossi | | **Matrícula**: 3012348 |
| **Nombre del curso:**  Aprendizaje Automático | **Nombre del profesor**:  Rolando Evelio Pérez Versón | |
| **Módulo**: 1 | **Actividad**: 3 | |
| **Fecha**: 11-Jun-2022 | | |
| **Bibliografía**:  Rosebrock, A. (2021). Backpropagation from scratch with python. *pyimagesearch*. Recuperado de <https://pyimagesearch.com/2021/05/06/backpropagation-from-scratch-with-python>. | | |

**Título**: Cálculo vectorial y probabilidades.

**Introducción**:

El siguiente reporte corresponde a la actividad 3 de Aprendizaje Automático.

**Desarrollo**:

1. Operaciones de Primer y Segunda Derivada

Notebook link: [notebook-3.1](https://github.com/Robbende/tecm-ai-master/blob/main/Aprendizaje_Automatico/Actividad-3/Act-3.1.ipynb)

1. Operaciones de Primer y Segunda Derivada Parciales

Notebook link: [notebook-3.2](https://github.com/Robbende/tecm-ai-master/blob/main/Aprendizaje_Automatico/Actividad-3/Act-3.2.ipynb)

1. Algoritmo de Propagación Inversa

El algoritmo de propagación inversa es un tipo de algoritmo usado en el aprendizaje supervisado para redes neuronales. El objetivo del algoritmo es de afinar las funciones de peso y así mejorar la precisión del modelo.

Para ello se emplea un método de descenso de gradiente para reducir la función de costo.

También se puede pensar a la propagación inversa como la propagación hacia atrás de los errores para lograr una mayor precisión.

La derivada parcial es el corazón de la propagación inversa donde se utiliza para determinar qué tan rápido cambia el costo cuando se cambia los pesos y el bias.

Para aplicar la propagación inversa la función de activación debe ser derivable pudiéndose calcular la derivada parcial del error con respecto al peso dado (*Rosebrock, 2021*).

Notebook link: [notebook-3.3](https://github.com/Robbende/tecm-ai-master/blob/main/Aprendizaje_Automatico/Actividad-3/Act-3.3.ipynb)

1. Cálculo de Probabilidades
2. Estadística Descriptiva