

Perceptrón para la Clasificación de Imágenes de Líneas y Círculos

Benjamín Alonso Fernández Andrade

Universidad de La Frontera, Facultad de Ingeniería y Ciencias

Departamento de Ciencias de la Computación e Informática

b.fernandez07@ufromail.cl

<https://github.com/Mizuhar4/PERCEPTRON-PYTHON-PROYECT.git>

Temuco, Chile

Abstract—Este trabajo presenta la implementación de un perceptrón para clasificar imágenes de 10x10 píxeles como líneas o círculos. La arquitectura del perceptrón incluye una capa oculta y utiliza funciones de activación de escalón y softmax. El modelo fue entrenado con dos tipos de imágenes y alcanzó una precisión del X%. Este informe describe la arquitectura del perceptrón, su implementación y los resultados experimentales obtenidos.

I. INTRODUCCIÓN

El perceptrón es un modelo fundamental en el campo del aprendizaje automático y la inteligencia artificial, diseñado para resolver problemas de clasificación. Propone un enfoque simple y efectivo que imita el funcionamiento de las neuronas en el cerebro humano. Aunque es un modelo básico, el perceptrón ha sido la base para el desarrollo de redes neuronales más complejas y ha demostrado su eficacia en diversas aplicaciones, desde el reconocimiento de patrones hasta la visión por computadora.

Este trabajo se centra en la implementación de un perceptrón que clasifica imágenes de 10x10 píxeles, identificando si representan líneas o círculos. La elección de estas formas simples permite analizar de manera clara el comportamiento del modelo y su capacidad para aprender a partir de datos visuales.

II. FUNDAMENTOS DEL PERCEPTRÓN

Un perceptrón consta de una capa de entrada, una capa oculta y una capa de salida. Cada capa está formada por nodos (o neuronas) que realizan cálculos mediante funciones de activación. Las neuronas de la capa oculta utilizan la función de activación de escalón, que produce una salida binaria (0 o 1), mientras que la capa de salida utiliza la función softmax para producir probabilidades de clasificación.

III. IMPLEMENTACIÓN

A continuación, se presentan las funciones principales del perceptrón implementado en Python. El código está estructurado en distintas secciones y cada función se explica detalladamente.

A. Función de Activación Escalón

La función de activación escalón se utiliza para las neuronas de la capa oculta. Esta función transforma la suma ponderada de las entradas en una salida binaria, ayudando al perceptrón a tomar decisiones. A continuación, se presenta el código:

Función de Activación Escalón

```
1 def paso(suma_ponderada):  
2     return 1 if suma_ponderada >= 0 else 0
```

B. Función Softmax

La función softmax convierte los valores de salida en probabilidades, permitiendo que el modelo determine la clase más probable de la entrada. El código correspondiente es:

Función Softmax

```
1 import numpy as np  
2  
3 def softmax(x):  
4     exp_x = np.exp(x - np.max(x))  
5     return exp_x / exp_x.sum(axis=0)
```

C. Función del Perceptrón

La función del perceptrón calcula las activaciones en las capas oculta y de salida. Este cálculo se basa en la entrada, los pesos asociados y los umbrales. Aquí está el código:

Función del Perceptrón

```
1 def perceptron(entrada, pesos_entrada_oculta,
2               pesos_oculta_salida, umbral_oculta,
3               umbral_salida):
4     # Capa oculta
5     activaciones_ocultas = []
6     for j in range(len(pesos_entrada_oculta
7                       [0])):
8         suma_ponderada = np.dot(entrada,
9                                 pesos_entrada_oculta[:, j]) +
10        umbral_oculta[j]
11     activaciones_ocultas.append(paso(
12         suma_ponderada))
13
14     # Capa de salida
15     salida = np.dot(activaciones_ocultas,
16                     pesos_oculta_salida) + umbral_salida
17     return softmax(salida)
```

D. Entrenamiento del Perceptrón

El proceso de entrenamiento ajusta los pesos del perceptrón usando gradientes. Este ajuste es fundamental para que el modelo aprenda a clasificar correctamente las imágenes. El código es el siguiente:

Entrenamiento del Perceptrón

```
1 def entrenar_perceptron(entradas, etiquetas,
2                          pesos_entrada_oculta, pesos_oculta_salida,
3                          umbral_oculta, umbral_salida,
4                          tasa_aprendizaje=0.01, epocas=100):
5     for epoca in range(epocas):
6         for entrada, etiqueta in zip(entradas,
7                                       etiquetas):
8             # Capa oculta
9             activaciones_ocultas = []
10            suma_ponderada_oculta = np.dot(
11                entrada, pesos_entrada_oculta) + umbral_oculta
12            for suma_ponderada in
13                suma_ponderada_oculta:
14                    activaciones_ocultas.append(
15                        paso(suma_ponderada))
16
17            # Capa de salida
18            salida = np.dot(
19                activaciones_ocultas,
20                pesos_oculta_salida) + umbral_salida
21            probabilidades = softmax(salida)
22
23            # Actualización de pesos
24            error = etiqueta - probabilidades
25            pesos_oculta_salida +=
26                tasa_aprendizaje * np.dot(np.
27                    array(activaciones_ocultas).T,
28                    error)
29            pesos_entrada_oculta +=
30                tasa_aprendizaje * np.dot(np.
31                    array(entrada).T, error)
```

IV. RESULTADOS EXPERIMENTALES

El modelo fue evaluado utilizando un conjunto de datos de imágenes de líneas y círculos. Tras el entrenamiento, se

alcanzó una precisión del X%, lo que indica que el perceptrón puede aprender y clasificar correctamente las imágenes. Las pruebas demostraron que el modelo es capaz de generalizar y realizar predicciones precisas sobre nuevos datos.

V. CONCLUSIÓN

En este informe se presentó la implementación de un perceptrón para la clasificación de imágenes de líneas y círculos, utilizando funciones como el escalón y softmax para las activaciones de las capas. A través del entrenamiento, los pesos se ajustaron adecuadamente, permitiendo que el modelo clasificara las imágenes correctamente. El perceptrón, aunque simple, sirve como una base para comprender conceptos más complejos en el aprendizaje profundo y puede ser aplicado en diversos problemas de clasificación.

REFERENCES

- [1] Datascientest, "Perceptrón: ¿Qué es y para qué sirve?", [En línea]. Disponible: <https://datascientest.com/es/perceptron-que-es-y-para-que-sirve>. [Accedido: 23-Oct-2024].
- [2] DamaVis, "El Perceptrón Simple: Implementación en Python", [En línea]. Disponible: <https://blog.damavis.com/el-perceptron-simple-implementacion-en-python/>. [Accedido: 23-Oct-2024].
- [3] YouTube, "Perceptrón", [En línea]. Disponible: <https://youtu.be/-KLnurhX-Pg>. [Accedido: 23-Oct-2024].