Red Neuronal

*Tarea04 del 22/10/2018

1st Astrid Carolina Estrada Trejo
Vision por Computadora
Centro de Innovación y Desarrollo Tecnológico en Cómputo
Cdmx, Mexico
astrich_star93@hotmail.com

***Abstract*—Las redes neuronales son un modelo matemático que nos permite aproximar una función, trata de crear modelos artificiales que solucionen problemas difíciles de resolver mediante técnicas algorítmicas convencionales En este articulo .**

Keywords—red neurronal, sistema lineal, pytorch, .

I. Introduccion

El primer modelo matemático de una neurona artificial, creado con el fin de llevar a cabo tareas simples, fue presentado en el año 1943 en un trabajo conjunto entre el psiquiatra y neuroanatomista Warren McCulloch y el matemático Walter Pitts.

La siguiente figura muestra un ejemplo de modelo neuronal con n entradas, que consta de:

- Un conjunto de entradas x1, …xn.
- Los pesos sinápticos w1, …wn, correspondientes a cada entrada.
- Una función de agregación, Σ.
- Una función de activación, f.
- Una salida, Y.

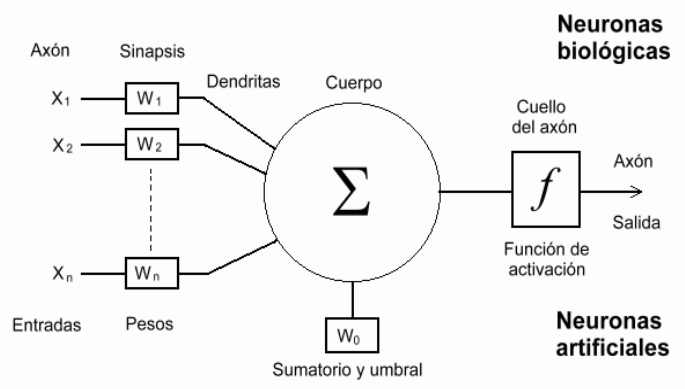


Fig1.Red Neuronal.

Las entradas son el estímulo que la neurona artificial recibe del entorno que la rodea, y la salida es la respuesta a tal estímulo. La neurona puede adaptarse al medio circundante y aprender de él modificando el valor de sus pesos sinápticos, y por ello son conocidos como los **parámetros libres** del modelo, ya que pueden ser modificados y adaptados para realizar una tarea determinada.
En este modelo, la salida neuronal Y está dada por:

$$Y = f(\sum_{i=1}^{n} w_i x_i)$$

Es una función de activación, la cual representa simultáneamente la salida de la neurona y su estado de activación.

II. Perceptron

Para realizar tareas de clasificación en el plano, se usa el perceptron, el simple consta de una capa de salida de n neuronas y otra de salida con m neuronas.Fue introducido por Rosenblatt en 1962 es un modelo unidireccional expresada:

$$y_i = f\left(\sum_{j=1}^{n} w_{ij} x_j - \theta_i\right)$$

Con i= 1,…,m.
Como herramienta de clasificador, es importante destacar su uso de entrenamientoya que permite determiner automáticamente los pesos sinápticos que clasifican un conjunto de patrones a partir de un conjunto de ejemplos etiquetados.PErmite dsicriminar entre clases linealmente separables, es decir clases cuyas regiones de decisión pueden ser separadas mediante una única condición linela o hiperplana.

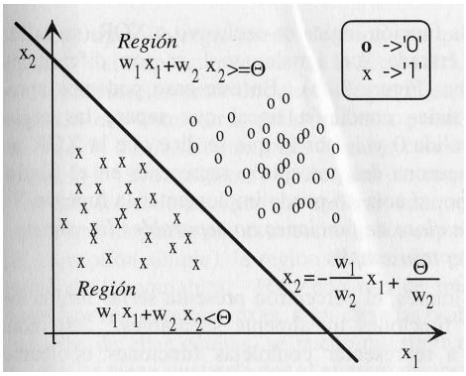


Fig2.Region de decision.

III. Pytorch

Es un paquete de Python diseñado para realizar cálculos numéricos haciendo uso de la programación de tensores. Además permite su ejecución en GPU para acelerar los cálculos.
Normalmente PyTorch es usado tanto para sustituir numpy y procesar los cálculos en GPU como para la investigación y desarrollo en el campo del machine learning, centrado principalmente en el desarrollo de redes neuronales.
Al contrario que otros paquetes como Tensorflow, PyTorch trabaja con grafos dinámicos en vez de estáticos. Esto significa que en tiempo de ejecución se pueden ir

modificando las funciones y el cálculo del gradiente variará con ellas. PyTorch dispone de soporte para su ejecución en tarjetas gráficas (GPU), utiliza internamente CUDA, una API que conecta la CPU con la GPU que ha sido desarrollado por NVIDIA

IV. Procedimiento y Resultados

Descargamos el conjunto de datos de entrenamiento Fashion Mnist, cada imagen contiene prendas como ropa o zapatos, y su conjunto de prueba respectivamente, ordenándolos por parejas. A partir del conjunto de ejemplos esperamos que la red aprenda una función f(x) tal que la salida de la red imite el patrón en los datos.

Se inicializa la red neuronal dándole un valor aleatorio al sesgo (constante que da a la f(x) un mejor acercamiento a la función objetiva) y a los pesos (una representación numérica de lo que influye una entrada en una red neuronal). Muestreamos desde una distribución normal con media cero y desviación estándar = 0.01, porque los valores para inicialización debe estar en el orden de las milésimas en el peso. En seguida se realiza el pase frontal que convertir la imagen a un tensor y pasarla a través de la red. Entonces hace el back propagation para ajustar los pesos para obtener un mejor resultado lo hace con cada una de las imágenes que tiene el dataset, al terminar tendremos nuestro entrenamiento, lo que significa que tenemos nuestro modelo de red neuronal.

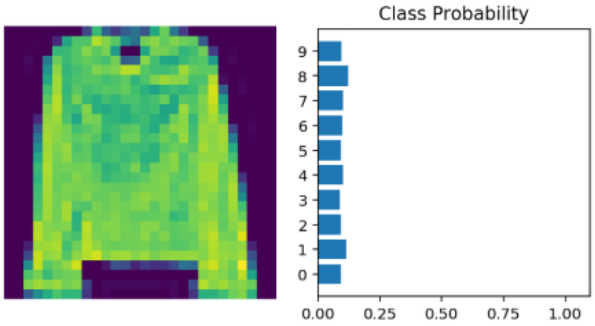
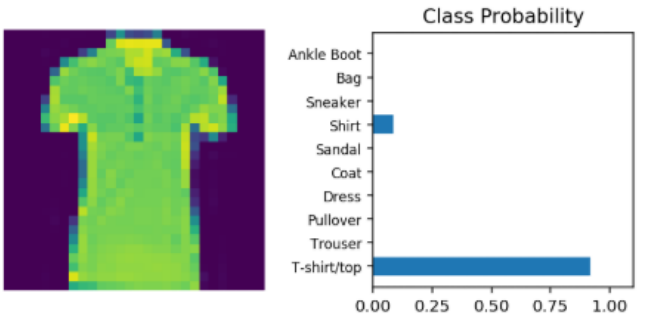


Imagen de entrenamiento

Después se evalúa con el conjunto de prueba.



Prueba sobre el modelo creado

References

[1] M. Minsky, S. Papert (1969). Perceptrons: An Introduction to Computational Geometry, MIT Press Open Source Computer Vision Library, https://docs.opencv.org/3.3.0/dc/dc3/tutorial_py_matcher.html ,2018.

[2] Pytorch.https://pytorch.org