

Diseño de redes neuronales y aprendizaje profundo (Gpo 201)



**Tecnológico
de Monterrey**

MA2003B.201

**Ricardo Camacho Castillo
A01654132**

Proyecto de Aprendizaje Profundo

Profesores: Dr. Santiago Enrique Conant Pablos

Índice

Introducción	1
Modelo 1	2
Modelo 2	3
Conclusiones	4

Introducción:

Fashion-MNIST es un nuevo conjunto de datos estándar que se utiliza en visión artificial y aprendizaje profundo. El conjunto de datos consiste en imágenes de artículos de Zalando Research que consta de un conjunto de formación de 60.000 ejemplos y un conjunto de prueba de 10.000 ejemplos. Cada ejemplo es una imagen en escala de grises de 28x28, asociada con una etiqueta de 10 clases. Utilizando este conjunto de datos exploraremos arquitecturas y/o modelos de redes neuronales para clasificar y/o agrupar las prendas de vestir ahí contenidas.

Zalando Research propuso el Fashion-MNIST como un reemplazo directo del conjunto de datos MNIST original para la evaluación comparativa de los algoritmos de aprendizaje automático. El conjunto original contiene muchos dígitos escritos a mano y a los miembros de la comunidad AI / ML / Data Science les encanta por lo que lo utilizan como punto de referencia para validar sus algoritmos. Ambos conjuntos comparten el mismo tamaño de imagen y estructura de divisiones de entrenamiento y prueba.

El aprendizaje automático es la disciplina científica que se centra en cómo las computadoras aprenden de los datos. Sin embargo las redes neuronales convolucionales, o CNN por sus siglas en inglés, son las arquitecturas que en los últimos años han mostrado una gran mejora en relación a los modelos de aprendizaje automático tradicional. Las CNN son arquitecturas que están dentro del área llamada aprendizaje profundo (deep learning), ya que se incluyen una gran cantidad de capas profundas, lo cual a su vez implica tener una mayor cantidad de datos de entrenamiento para encontrar los miles o millones de pesos involucrados.

Modelo 1:

Para el primer modelo se utilizará una red neuronal con dos capas densas para poder clasificar las imágenes de Fashion-MNIST y poder identificar qué tipo de ropa son.

Para preparar los datos se bajó la base de datos desde los datasets de keras para de esta manera tener tanto los datos de entrenamiento como los de prueba.

Después se escalan los datos a escalas de grises para poder trabajar con vectores sin colores.

Después se imprimen algunas imágenes para conocer un poco más cómo es que se ven las imágenes de la base de datos.

Después se crea un modelo que tiene una capa de entrada en la que se usa `FLatten` para convertir la imagen a un vector y posteriormente se agregan dos capas densas, una con la función `relu` y la siguiente con la función `softmax`.

Se utiliza el optimizador `adam` usando `'compile'` para el modelo.

Después se entrena el modelo usando 20 épocas de entrenamiento.

Finalmente, ya con el modelo entrenado se crean funciones para mostrar tanto la imagen la cual se quiere clasificar junto con una gráfica de probabilidades con los grupos que existen para clasificar y se pone en azul las que tuvieron clasificación correcta y en rojo las que fallaron.

Modelo 2:

Para el segundo modelo, en vez de clasificar, se va a usar una red GAN para generar imágenes de la base de datos. Esto se decidió ya que en el bloque que se tuvo este periodo surgieron dudas en la clase con este tipo de redes.

Se tenía una base de datos de variables meteorológicas y contaminantes y al tener datos faltantes surgió la idea de generar dichos datos con una red GAN. Al ser un tema interesante, pero después encontrar estaciones cercanas de monitoreo ambiental con las que se pudieron rellenar los datos faltantes, se decidió aplicar este tipo de red para este ejercicio.

Los datos se cargaron de la misma manera que con el modelo 1 desde keras.

Después se seleccionó un número random de imágenes para poder entrenar el modelo con ellas.

Posteriormente se agregaron capas al modelo, entre ellas capas densas, capas de Batch Normalization, función de activación Relu, capas convolucionales, y al final se usó la función tanh para la última capa.

Después se generó una imagen sin el modelo entrenado.

Se usó también un modelo discriminador agregando tanto capas densas, como secuenciales, convolucionales, Flatten (convierten imagen a vector), y capas ReLu.

Después se realizaron funciones para determinar el loss tanto del generador como del discriminador.

Se usaron 100 épocas de entrenamiento generando 4 imágenes. También se creó una función para guardar las imágenes de cada época y poder acceder a ellas posteriormente.

Imprimiendo las imágenes nos damos cuenta cómo mientras más épocas hay, mejor se van viendo las imágenes.

En este paso del entrenamiento se cambió a entorno de ejecución GPU ya que estaban tardando aproximadamente 120 segundos cada época. Con GPU tardaban solamente 5-6 segundos cada una.

Finalmente, con el modelo ya entrenado se generó una imagen aleatoria de la cual obtuvimos un sneaker.

Conclusiones.

En este caso no podemos ver algunas comparaciones ya que los objetivos para cada modelo fueron diferentes. Por un lado con uno se hicieron clasificaciones y con otro se generaron nuevas imágenes a partir de imágenes ya procesadas, algo así como cuando se rellena una foto con el fondo después de eliminar personas por ejemplo.

Algo que podría servir sería probar los dos modelos juntos, primero generar algunas imágenes con la base de datos, después entrenar el modelo de clasificación para posteriormente predecir las imágenes generadas con el clasificador.

Este proyecto fue muy interesante ya que no aprendes sólo a usar una red neuronal, algo así como importar de scikit learn y después usar el modelo. Aquí, aunque no se entiende 100% la funcionalidad completa de las redes neuronales, se aprende a estructurarlas para adaptarlas a problemas posteriores que se puedan tener. Dependiendo del problema puedes añadir o quitar capas, modificar tamaños, tipos de capas, funciones de activación, optimizadores...

Es un trabajo que nos deja mucho aprendizaje lo cual sólo fue el inicio, ya que después, ya sabiendo la estructura y la forma en la que se usan, se pueden crear redes neuronales para cualquier tipo de problema que se tenga.

Para futuros proyectos se puede utilizar las redes GAN que en lo personal fueron las más interesantes para de esta manera completar o rellenar bases de datos que se tengan incompletas.