Deep Learning CNN

Indhira Ramirez
UTEC
Lima, Perú
indhira.ramirez@utec.edu.pe

Mauricio Nieto
UTEC
Lima, Perú
mauricio.nieto@utec.edu.pe

***Abstract*—Este documento es una guia para la implementación de una multilayer perceptron en python**

I. Objetivos

- Analizar los datos según sus características y decidir que modelo utilizar. Proponer una arquitectura.
- Utilizar 70% para el entrenamiento y 30% para testing.
- Realizar una solida experimentación a fin de alcanzar un accuracy mayor a 70%
- Se debe lograr clasificar según emoción, genero y grupo de edad.

II. Marco Teórico

Se utilizaron arquitecturas de CNN junto a MLP para resolver los problemas de clasificación propuestos. Una CNN tiene la característica especial de incorporar convoluciones a sus capas, una convolución es el proceso por el cual se obtiene una nueva imagen o matriz a partir de la primera, dento de la CNN el entrenamiento consistirá en hallar las convoluciones más adecuadas para resolver el problema de clasificación. En adición, una MLP es una red neuronal compuesta únicamente de neuronas que realizan una suma ponderada de los valores que reciben.

A. Arquitecturas

Con el fin de solucionar el problema de clasificación se utilizaron dos arquitecturas, debido a la diferencia que había entre las tareas de clasificar por emoción y clasificar por género y edad, pues se requería de una arquitectura más especializada para la detección de emociones. Asimismo, se utilizó una arquitectura CNN en ambas que constan de una serie de convoluciones seguidas por un MLP(Multilayer Perceptron), pues es conocida que esta presenta muy buenos resultados para el procesamiento de imágenes ya que da importancia espacial a la distribución de los datos, que son imágenes, formato al que corresponde la data utilizada. También, para las capas convolucionales en ambos modelos se siguió una secuencia de convolución, activación y maxpooling.

1) Arquitectura 1:: La arquitectura uno fue utilizada para la clasificación por emoción, esta arquitectura dispone de una capa de entrada, una capa de salida y 4 capas ocultas, donde las primeras 4 capas corresponden a capas convolucionales y la última a un MLP. Lo primero que se encontró en esta arquitectura, a través de la experimentación, es que el kernel es muy importante para el reconocimiento de emociones, pues una expresión facial posee características muy complejas y amplias en el rostro, por lo que solo un kernel amplió puede captar estos patrones, por ello la distribución usada en tamaño de kernel en las cinco primeras capas es de 11, 7, 5, 5 y 3 respectivamente, la razón por las que estas decrecen es que a medida que se aplican convoluciones y maxpooling se va reduciendo la dimensión de la imagen y no da espacio a seguir aplicando un kernel de amplio tamaño. No obstante, el aumento en el tamaño de kernel para las capas dio lugar a un aumento del coste computacional, para reducir este coste se redujo la cantidad de convoluciones en cada capa, dando lugar a una distribución de 32, 32, 32 , 32 y 32 para las primeras cinco capas. Igualmente, con el fin de disminuir el tiempo de computo se aumentó el stride para la primera capa que es la de mayor kernel. Sin embargo, esto redujo tanto la dimensionalidad en las imágenes que las últimas capas no recibían suficiente información, para solucionar esto se aumentó los paddings en las capas de manera proporcional al kernel en la capa, dando como resultado una distribución 5, 3, 2 , 2 y 1 para las primeras cinco capas. Como resultado de estos parámetros se obtuvo una arquitectura capaz de aplicar grandes kernel y a la vez mantener un coste computacional relativamente bajo. Luego, se notaron beneficios a la hora de aplicar un dropout en las capas 3 y 4, precisamente antes de aplicarse la función de activación, esta mejora de predicción se debe a que el dropout evita el overfitting. Por último, la ultima capa se constituyó de 2 capas lineales con su respectiva función de activación y una última capa lineal sin función de activación, la ausencia de función de activación en la última capa aumento la precisión del modelo.

2) Arquitectura 2: La arquitectura 2 se utilizó para la clasificación de género y sexo, la clasificación simultanea de ambas características es resultado de que por separado se obtuvieran precisiones de predicción siempre mayores al 95 por ciento y generalmente del cien por ciento. Esta arquitectura fue más sencilla pues las diferencias entre sexos y edades son más notables y menos amplias que las diferencias presentes entre emociones. Para esta arquitectura se aplicaron 5 capas de convolución seguida de una MLP, los paddings fueron de 1, tan solo para mantener la dimensionalidad al aplicar convolución, el kernel usado para todas las convoluciones tuvo un tamaño de 3 pues estos mantenían un computo computacional bajo y eran suficientes para la clasificación. Finalmente el MLP usado solo presentó una capa de neuronas con una función de activación y una segunda capa sin función de activación pues

esto permitió mejores predicciones.

III. Experimentación

Para la experimentación se dividieron los datos en dos conjuntos, uno para entrenamiento y otro para testeo, la proporción entre ambos conjuntos fue de 7 y 3 respectivamente. Igualmente, se aseguro que hubiera una cantidad mínima de cada conjunto a clasificar en el conjunto de entrenamiento, esto para asegurar que no ocurriera el caso dónde el conjunto de entrenamiento no contase o contase con muy pocas muestras de un grupo a clasificar. De igual manera, para el entrenamiento se aplicó un límite de error, es decir que el entrenamiento se detenía al momento de alcanzar cierto error en el conjunto de entrenamiento. Además, el conjunto de testeo nunca estuvo incluido en el entrenamiento, para que este conjunto cumpla su función de asegurar un testeo adecuado.

El hiper parámetro utilizado para entrenar la red neuronal en el caso de la arquitectura 1 fue 0.0005 pues daba un buen rendimiento computacional y a su vez permitía que el error descendiera de manera adecuada, por las mismas razones se optó por el valor de 0.001 en la arquitectura 2. Asimismo, el optimizador en la gradiente en ambas arquitecturas fue Adam para controlar la caída de la gradiente en tanto aumentaban las repeticiones de entrenamiento, finalmente la función para compara el éxito del modelo en predecir resultados fue cross entropy, especialmente recomendada para fines de clasificación pues produce una curva de error logarítmica que disminuye rápidamente a mayor sea el error.

A. Pre-procesamiento

Para el pre-procesamiento primero se hizo un resize, ya que las imágenes eran muy grandes, y después se aplico una función de MTCNN, que estaba pre entrenada para reconocer rostros. El uso de MTCNN disminuyo el tiempo de ejecucion y también nos ayudo a elevar el accuracy en la clasificación de emociones. Esto se debe a que las emociones eran mucho mas fácil de clasificar teniendo en cuenta solo el rostro.

En la arquitectura 1 se le aumento el brillo y el contraste a las imagenes, para que así las facciones se vean mas prominentes.

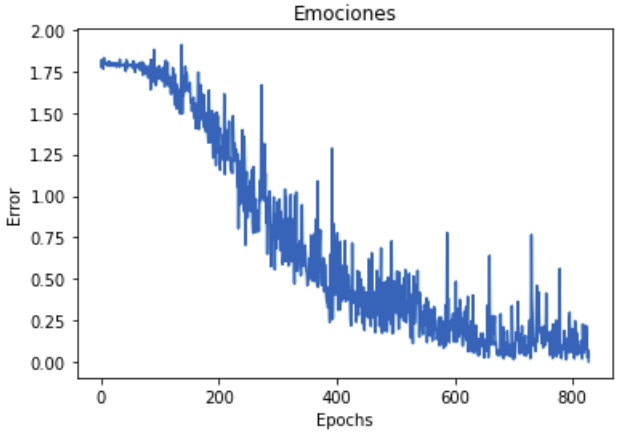
B. Resultados

Como se explico anteriormente, dividimos este problema en dos, por un lado la clasificación de genero y grupo de edad, y por otro lado la emoción.

1) Arquitectura 1: Emoción:

- El acuracy mejoro en poco mas de 10% mas al usar solo el rostro como input y no toda la imagen completa y también al aumentar el contraste.
- Se necesito mas épocas de entrenamiento en comparación de la arquitectura 2.
- Se llego hasta un accuracy de 90% en un caso especial, pero usualmente esta entre 77% y 81%.

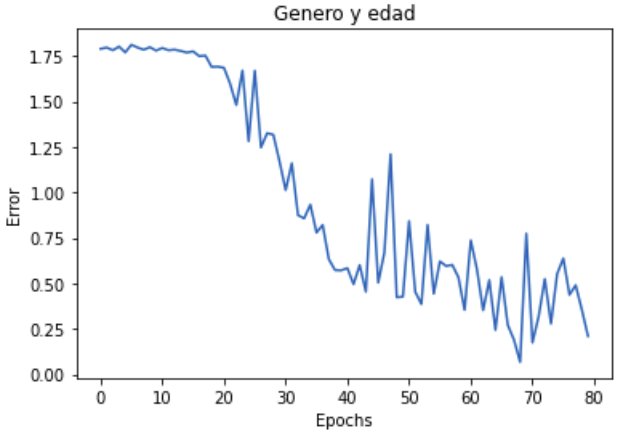
En la siguiente imagen se puede observar la caída del error respecto a las épocas.



2) Arquitectura 2: Genero y grupo de edad:

- No se necesito un entrenamiento tan arduo como la arquitectura 1. Esto se debe a que tanto el género como el grupo de edad son mas fáciles de diferenciar.
- Se llego hasta un accuracy de 95% en un caso especial, pero usualmente esta entre 90% y 81%.

En la siguiente imagen se puede observar la caida del error respecto a las épocas.



IV. Conclusión

- No se pudo usar redes previamente entrenadas porque se uso muy poca data, lo cual hacia que la gradiente desaparezca.
- Fue de gran ayuda usar una MTCN, ya entrenada, para poder trabajar solo con los rostros.
- El genero y el grupo de edad, tanto juntos como por separado, son mucho mas fáciles de clasificar que las emociones.
-

V. Código fuente

Repositorio Colab:

Clasificación de emociones: click aquí

Clasificación de genero y edad: click aquí