Implementacion de una Red Neuronal Convolucional para la clasificación de rostros

Mauricio Pinto Larrea
(dept. Computer Science)
UTEC
Lima, Peru
mauricio.pinto@utec.edu.pe

Francesco Ucelli Meneses
(dept. Computer Science)
UTEC
Lima, Peru
francesco.uccelli@utec.edu.pe

Juan Manuel Navarro Nieto
(dept. Computer Science)
UTEC
Lima, Peru
juan.navarro@utec.edu.pe

I. Introducción

Los algoritmos de *Machine Learning* (ML) son utilizados para una gran variedad de aplicaciones. En este proyecto, utilizamos una red neuronal convolucional (CNN) para clasificación imágenes de rostros de acuerdo a tres criterios: edad, género y expresión. La clasificación de imágenes es la tarea de extraer información detallada y poder emparejar estas según similitudes que se puedan observar. Mas concretamente, es el proceso en el cual se categorizan tanto pixeles o vectores en una imagen. Esta técnica, como se ha visto en previos proyectos, se utiliza ampliamente en diversas áreas de la ciencia como la medicina, la biología y la química.

Como se mencionó, en este proyecto se hará uso de una CNN, con el objetivo final de encontrar los hiperparámetros óptimos para que nuestro modelo pueda alcanzar una precisión de al menos setenta por ciento.

II. Explicación

A. Arquitectura de la Red

Una Red Neuronal Convolucional (CNN por sus siglas en ingles) es un tipo de red neuronal mas frecuentado en problemas de clasificación y reconocimiento de imágenes. De la misma manera que cualquier otra red, esta se enfoca en resolver el problema encontrando una función que se aproxime lo mas posible al resultado deseado y, de la misma manera que el MLP, logra este resultado utilizando pesos (weights), biases y una técnica conocida como backpropagation[2].

En una red neuronal, cada capa consiste de una seria de nodos conocidos como neuronas, los cuales están conectados a todas las otras neuronas de la capa previa. En el caso de las CNNs, estas utilizan capas tridimensionales, en las cuales solo algunas neuronas están conectadas a la capa anterior. Estas redes están formadas por las siguientes capas:

- Capa convolucional: Esta se encarga de detectar conjunciones locales de las características de la capa previa y mapearlas.
- Capa de agrupación: también conocida como pooling layer o capa de disminución, esta se encarga de reducir el tamaño espacial de los mapas de activación
- ReLU: una implementación que combina capas de rectificación con un umbral de cero, la cual esta definida como:

$$relu(x) = max(0, x) \tag{1}$$

- Capa completamente conectada: tiene el objetivo de ajustar los parámetros de peso y crear una stochastic likelyhood

B. Red Neuronal Residual (Resnet)

Una red neuronal residual (también conocida como Resnet) es una red neuronal artificial que se se ensambla sobre las construcciones que se obtienen de las células piramidales. Su uso e importancia se centran mas que nada en resolver problemas complejos apilando layers adicionales en las redes neuronales, lo que resulta en una mejor precisión y performance del modelo. Básicamente, añadiendo mas capas lo que se logra es que estas puedan progresivamente aprender características mas complejas del input[8].

En nuestro caso de reconocimiento de imágenes, la primera capa podría, por ejemplo, aprender a detectar bordes, la siguiente texturas, luego objetos, etc. Se utilizara Resnet en el modelo con el objetivo de reducir el error porcentual del modelo tanto en el entrenamiento como con las pruebas.

III. Dataset

El dataset utilizado [6] incluye 72 imágenes, que incluyen a personas jóvenes, de mediana edad y ancianas (entre 20 y 77 años), hombres y mujeres, y cada una con seis expresiones diferentes: ira, disgusto, miedo, felicidad, tristeza y neutralidad.

Para esto, dividimos el dataset según los parámetros indicados y procedimos a entrenar nuestro modelo con el objetivo de que pueda reconocer estas expresiones, edades y géneros según la imagen que se le vaya a introducir. Esto se puede ver reflejado en la funcion `generate_dataframe` en nuestro código, aprovechando que cada sigla en el nombre de los archivos corresponde a algún tipo de expresión/edad/genero.

IV. Experimentación

Para la experimentación, se dividió la data del dataset proporcionado en un setenta por ciento para entrenamiento y treinta por ciento para las pruebas, esto segun cada una de las

características previamente mencionadas. El trabajo realizado se encuentra publicado en [9].

Para la experimentación medimos tanto el error (CrossEntropyLoss) como el accuracy. A continuación se muestran los gráficos de error con 20 epochs para los modelos de edad, género y expresión. En el gráfico se muestran los steps. En el caso del modelo de clasificación de expresiones, se tiene la diferencia de que se usan los modelos resnext50 y resnet18 con el parámetro de pretrained. Esto quiere decir que al cargar los modelos se cargan ajustados para una clasificación anterior. En este caso, la predeterminada de la librería pytorch es ImageNet, que es una competencia de clasificación de todo tipo de imágenes. Dado que la cantidad de datos en este proyecto es bastante limitada, el preentrenamiento de los modelos tiene un impacto significativo en su desempeño. Se puede notar una diferencia de hasta 30 o 40% con el modelo sin preentrenar.

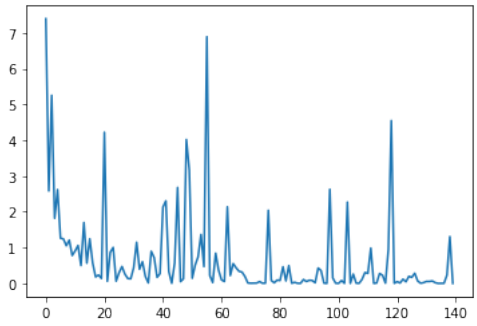

Fig. 1. Perdida en la categoría de edad

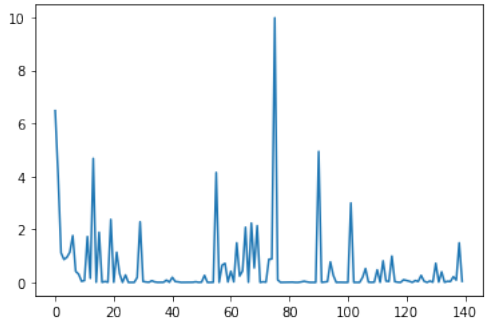

Fig. 2. Precisión en la categoría de edad

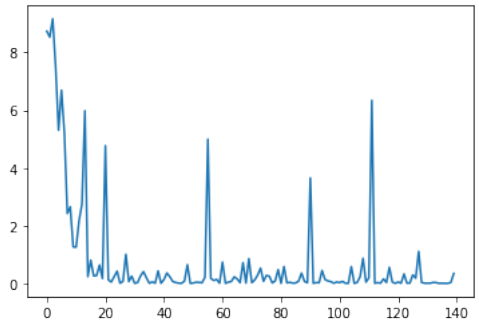

Fig. 3. Perdida en la categoría de genero

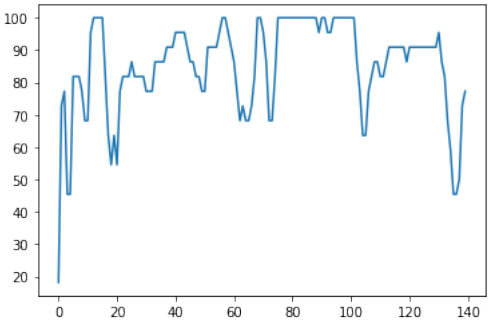

Fig. 4. Precisión en la categoría de genero

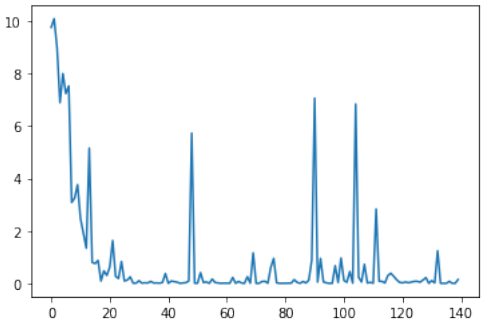

Fig. 5. Perdida en la categoría de expresiones

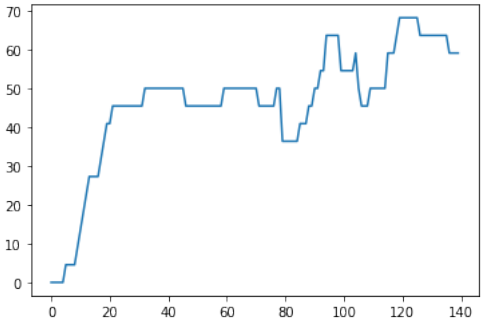

Fig. 6. Precisión en la categoría de expresiones

En la siguiente tabla podemos observar los cambios de las precisiones y los resultados en las pruebas finales. Como se

observa, el modelo fue 100% preciso para clasificar a las personas por edad. Para las demás categorías se alcanzaron valores altos en el entrenamiento, pero las predicciones en la etapa de prueba no fueron tan precisas. Esto posiblemente se deba a la limitada cantidad de datos.

	Época 0	Época 5	Época 10	Época 15	Prueba
Edad	40.9090	77.2727	95.4545	95.4545	100
Género	18.1818	86.3636	95.4545	63.6363	77.2727
Expresión	9.0909	63.6363	77.2727	63.6363	68.1818

TABLE I
VALORES DE PRECISIÓN DE LOS MEJORES MODELOS, PARA LAS ÉPOCAS 0, 5, 10, 15 Y LA ETAPA DE PRUEBA.

V. CONCLUSIONES

Como podemos ver en la sección de experimentación, las tareas de clasificación de edad y de género llegan en algún momento de testo hasta el 100% de precisión y en la final siempre se acercan. De igual manera influye bastante la varianza ya que la cantidad de datos no es lo suficientemente grande como para garantizar buena generalización. Por otro lado, vemos que el modelo de Resnet para clasificación de expresiones funciona significativamente mejor si se usa un modelo preentrenado, de igual manera esto está relacionado con los escasos datos. Concluimos que esto se debe a la efectividad del proceso de entrenamiento, y una mejora en relación a cantidad de datos y a épocas de entrenamiento (con la capacidad computacional correspondiente) podrían garantizar que los modelos mejoren mucho su precisión.

VI. REFERENCIAS

1 Sathyanarayana, Shashi. (2014). A Gentle Introduction to Backpropagation. Numeric Insight, Inc Whitepaper.
2 https://scriptreference.com/neural-networks-from-scratch/neural-network-gradient-descent
3 http://alexlenail.me/NN-SVG/index.html
4 https://patrickhoo.wixsite.com/diveindatascience/single-post/2019/06/13/activation-functions-and-when-to-use-them
5 https://deepai.org/machine-learning-glossary-and-terms/softmax-layer
6 FACES Dataset, https://faces.mpdl.mpg.de/imeji/collection/IXTdg721TwZwyZ8e?q=
7 PyTorch vision models, https://pytorch.org/vision/stable/models.html
8 https://towardsdatascience.com/introduction-to-resnets-c0a830a288a4
9 Kaggle Repository, https://www.kaggle.com/flrotm/proyecto5