

Tarea 5

Universidad Autónoma De Nuevo León
Facultad de Ciencias Físico-Matemáticas
Procesamiento y clasificación de datos

Redes Neuronales Convolucionales

Nombre: Ricardo Luna Escobedo

Matricula: 1805328

Fecha: 11/03/2025

1. Resumen

En este trabajo observaremos el empleo de la programación en python para poder llevar a cabo un análisis entorno a las imágenes, pero desde un punto de vista que involucre a las redes neuronales.

Siendo más objetivos, construiremos un modelo hecho a base de redes neuronales convolucionales. A partir de un conjunto de 167 imágenes, conformadas por un auto Shelby, un robot batalla y un perico, definiremos 3 categorías que nos permitan diferenciar a un conjunto de imágenes sobre otras (las categorías llevarán los siguientes nombres: Shelby, Robot y Perico). Una vez establecidas las imágenes con sus respectivas categorías, serán introducidas en el modelo; un 80 % de las imágenes lo entrenarán y un 20 % lo probarán. Al final, nuestro modelo deberá ser capaz de poder categorizar, correctamente, cualquier imagen nueva (que no forme parte del conjunto entrenamiento) que le introduzcamos; nos indicará si la imagen introducida pertenece a un auto Shelby, un Perico o un Robot batalla.

2. Introducción

Las nuevas tecnologías nos han brindado un mejoramiento en distintos ámbitos, entre ellos: **la seguridad**. Un ejemplo de ellos es la implementación de huellas y biometrías faciales de los clientes, al momento de querer ingresar a su cuenta bancaria. Pero... ¿nunca se han preguntado como es esto posible?, ¿cómo es que se puede identificar que cierta huella o rostro pertenece a Fulano De tal y no a Mengano De tal? ¿qué sucederá detrás de los ordenadores para que esta tarea pueda ser llevada a cabo? la respuesta a ello es muy simple: **redes neuronales convolucionales (CNN)**.

Si pensabamos que las matemáticas aprendidas en Álgebra lineal no tienen propósito alguno, ahora nos daremos cuenta de lo equivocados que estabamos. Ya que las CNN transforman las imágenes que introducimos al modelo creado en datos matriciales, para de esa manera comprimir la información que tenemos y obtener sus características más importantes. De esta forma, el modelo que vayamos construyendo irá aprendiendo y aprendiendo para poder clasificar correctamente cualquier imagen que le introduzcamos.

3. Investigación

3.1. Datos

Las CNN se distinguen de otras redes neuronales por su mejor desempeño con entradas de señal de imagen, voz o audio. Tienen tres tipos principales de capas, que son:

- Capa convolucional.
- Capa agrupamiento (pooling).
- Capa totalmente conectada.

La **capa convolucional** es el pilar central de una CNN, y es donde ocurre la mayor parte del cálculo. Requiere algunos componentes, que son datos de entrada, un filtro y un mapa de características. También hay un detector de características, conocido como kernel o filtro, que se moverá a través de los campos receptivos de la imagen, verificando si la característica está presente. Este proceso se conoce como *convolución*.

Después de cada operación de convolución, una CNN aplica una transformación de Unidad lineal rectificada (ReLU) al mapa de características, introduciendo la no linealidad en el modelo. De esta manera nos aseguramos que las neuronas más relevantes se activen y dejen pasar solo lo importante.

La **capa agrupamiento (pooling)** reduce la dimensionalidad de los datos entrada. Aunque se pierde mucha información en la capa de agrupación, también tiene una serie de beneficios para la CNN. Ayuda a reducir la complejidad, mejorar la eficiencia y limitar el riesgo de sobreajuste.

La **capa totalmente conectada** realiza la tarea de clasificación basada en las características extraídas a través de las capas anteriores y sus diferentes filtros. Mientras que las capas convolucionales y agrupadas tienden a usar funciones ReLU, estas capas suelen aprovechar una función de activación softmax para clasificar las entradas adecuadamente, produciendo una probabilidad de 0 a 1; es perfecta cuando el modelo elige una clase entre varias opciones. [1]

3.2. Descripción

1. Imágenes utilizadas

Cabe mencionar que no se mostrarán todas las imágenes, solo 3 de cada categoría para poder contextualizar.

1.1. Robot



(a) Vista trasera



(b) Taladro



(c) Vista frontal

1.2. Carro Shelby



(a) Vista superior



(b) Vista frontal



(c) Vista lateral

1.3. Perico



(a) Vista lateral izquierda



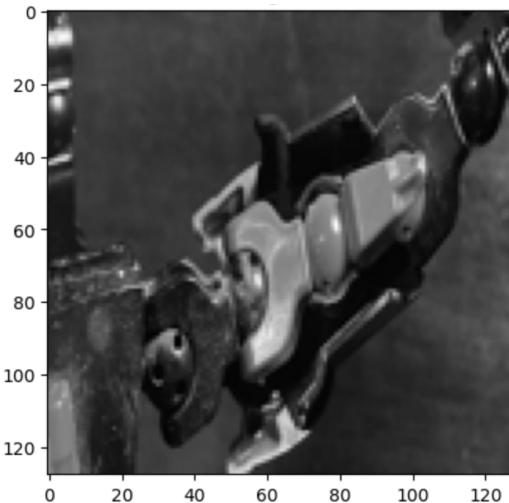
(b) Vista lateral derecha



(c) Vista frontal

2. Procedimiento

- a) Definimos el directorio de donde se van a tomar las imágenes a analizar.
- b) Definimos el nombre de las carpetas en donde se encuentran localizadas las imágenes en el directorio. En esta caso: *Shelby*, *Perico* y *Robot*. El nombre de estas carpetas servirá también para poder definir las etiquetas de donde pertenece la imagen seleccionada.
- c) Cargamos las imágenes al programa en un formato escala grises y nos aseguramos que todas tengan el mismo tamaño para no tener inconvenientes a la hora de ser introducidas al modelo (128 píxeles de largo x 128 píxeles de alto).



- d) Establecemos el modelo entrenamiento/prueba (le llamaremos: A) para poder clasificar imágenes con su respectiva etiqueta: el 80 % de las imágenes conforman el entrenamiento del modelo; el 20 % restante conforman su prueba.
- e) Establecemos el modelo convolucional (le llamaremos: B), para poder ir procesando las imágenes cargadas, de la siguiente forma:
 - **Establecemos la primera capa convolucional:** introducimos las imágenes de 128 x 128 en escala grises, usamos la función ReLU para introducir no linealidad y aplicamos 32 filtros de 3 x 3.
 - **Aplicamos la primera capa Pooling:** reducimos la dimensionalidad de las características extraídas a la mitad, tomando el máximo valor de regiones de 2 x 2.
 - **Establecemos la segunda capa convolucional:** aplicamos 64 filtros de 3 x 3 para detectar características más complejas.
 - **Aplicamos la segunda capa Pooling**
 - **Establecemos la tercera capa convolucional:** aplicamos más filtros, más características aprendidas (128 filtros de 3 x 3).
 - **Aplicamos la tercera capa Pooling**
 - **Establecemos la cuarta capa convolucional:** ahora usamos 256 filtros, profundizando en las características.
 - **Aplicamos la cuarta capa Pooling**
 - **Aplanamiento:** convierte la salida 3D de las convoluciones en un vector 1D, para poder conectarlo a la capa densa.
 - **Aplicamos la capa densa:** realiza la conexión de las 256 neuronas entre sí.
 - **Activación Dropout:** se desactivan aleatoriamente 60 % de las neuronas durante el entrenamiento para evitar el sobreajuste.
 - **Aplicamos la capa de salida:** establecemos 3 neuronas para los grupos a los que pertenecen las imágenes; usamos softmax para transformar las salidas en probabilidades.
- f) Optimizamos el modelo para asegurarnos que pueda ser corrido.
- g) **Corremos el modelo:** establecemos 20 epochs (repeticiones) para que el modelo pueda ajustarse a los datos prueba y entrenamiento.

```
Epoch 20/20
5/5 ━━━━━━━━ 1s 138ms/step - accuracy: 0.9237 - loss: 0.2259 - val_accuracy: 0.6765 - val_loss: 0.9250
```

- Conjunto entrenamiento: 92.37 % precisión.
- Error para hacer predicciones en el conjunto entrenamiento: 22.59 %.
- Conjunto prueba: 67.65 % precisión.
- Error para hacer predicciones en el conjunto prueba: 92.50 %.

Podemos ver que la mayor parte de nuestro modelo es funcional, a excepción por el error fallo en las predicciones prueba. Hagamos el funcionamiento de nuestro modelo con una imagen nueva:



Resultado:

1/1 ————— **0s 47ms/step**
La clase predicha es: Shelby

Nuestro modelo funciona parcialmente. Ya que sí tomó el nombre de cada una de las 3 categorías que le introducimos, pero produce un fallo al momento de asignarlo con la imagen asociada como consecuencia del alto porcentaje error obtenido en las predicciones prueba; se necesita mejorar el modelo aún más.

4. Conclusión

Hemos ahora introducido al análisis visual temas de algoritmos y probabilidad, y como estos nos pueden ayudar a tener una primera idea de lo que consisten los escáneres. Este modelo presentado en el artículo tiene mucha área oportunidad; no habrá hecho la predicción correctamente, pero si pudo leer nuestras imágenes y su respectiva clasificación.

Aun queda un largo camino por recorrer, pero lo importantes es que estamos aprendiendo a construir nuestros propios modelos predictivos-clasificatorios para posteriormente perfeccionarlos. La aplicación de este modelo fue utilizada para juguetes, sin embargo, hay todavía una infinidad de temas/objetos que podríamos usar para introducirlos a las CNN como: análisis faciales, análisis corporales, detección de billetes falsos/auténticos, etc.

5. Referencias

[1] IBM. ¿qué son las redes neuronales convolucionales? <https://www.ibm.com/mx-es/think/topics/convolutional-neural-networks>, 2024. Accedido el 9 de marzo de 2025.