¿Cómo funciona el scanner de letras?

...

Por: Roger Santiago Miranda Hoyos 1152363

Redes neuronales convolucionales

Es una red neuronal especializada en el procesamiento de imágenes, diseñada para identificar patrones visuales complejos mediante el uso de capas de convolución y pooling que extraen y procesan características de la entrada. Los filtros pueden empezar con características sencillas como brillo, bordes e ir creciendo en complejidad hasta que definan un objeto de forma singular.

¿Que es un tensor?

Es un objeto matemático que generaliza los conceptos de escalares, vectores y matrices a dimensiones superiores, representando un conjunto de números o funciones en un arreglo multidimensional.

Es decir, los tensores nos permite representar elementos de 0D, 1D y 2D, los cuales pueden ser, secuencias de textos, imágenes y audios.

En Python, TensorFlow, nos facilita su uso, para machine learning, proporcionando herramientas para entrenar redes neuronales.

TensorFlow_datasets...

Además, la API de TensorFlow integra un listado de datasets que podemos usar en nuestros proyectos. En mi caso, el dataset a usar se llama “emnist/letters”, que básicamente son distintas imágenes de letras escritas a mano, con fondos de color negro, y de tamaño 28x28.

Lo importante de estos datasets, es que TensorFlow no solo incluye el entrenamiento, también atribuye una serie de pruebas para que su entrenamiento sea mejor. Tal que se consigue un 80% de entrenamiento y un 20% de prueba.

Ahora hablemos de Keras...

Keras es una librería que viene en TensorFlow la cual nos permite crear y construir modelos de aprendizaje profunde de manera rápida y sencilla.

En nuestro caso la usamos para usar su modo secuencial, el cual es un modelo lineal donde cada capa alimenta la siguiente directamente.

Es decir, las capas están apiladas una después de la otra, por tanto, la salida de una capa, es la entrada para la siguiente.

Cuando decimos capa nos referimos a un bloque de procesamiento dentro de la red neuronal, cada una tiene parámetros ajustables.

Batch y Épocas

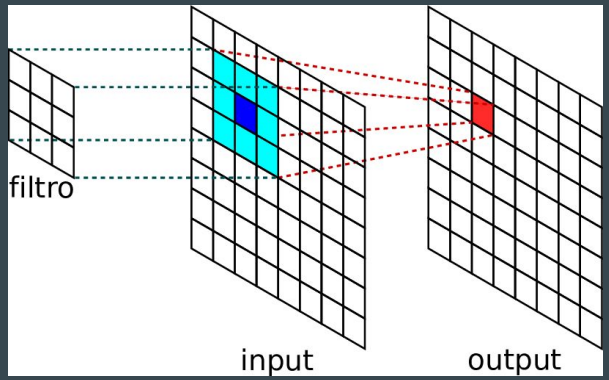
Se les dice Batch a los lotes en los que se separa la cantidad de archivos totales a evaluar, ya que las redes neuronales no van uno por uno, ni van con todo el conjunto de datos de una vez. Por ejemplo, tengo 88800 archivos en imnist, y estoy usando un batch de 128, entonces 88800/128 = 693.75, aproximadamente 694, entonces al momento de entrenar tendrá que pasar por 694 lotes por cada Época.

Una época es el número de pasadas que se le da a un dataset completo, en mi caso había usado 50 pasadas.

Capas y convoluciones

Las capas son aquellas encargadas de recibir información en una red neuronal, hacen interconexiones entre sí, cada una puede tener sus propias propiedades.

Las convoluciones son operaciones matemáticas de procesamiento de señales e imágenes, donde se combinan dos funciones para producir una tercera resultante, que se ve como una forma de desplazamiento y multiplicación



Normalización y Pooling

La normalización es la forma de optimizar nuestras redes neuronales, en el caso durante el proceso secuencial en capas, busca que la media sea más o menos igual a 0, y la desviación estándar sea igual a 1, estabilizando y acelerando el entrenamiento.

Pooling lo que hace reducir la dimensión de la imagen, es decir, su ancho y alto, para de esta forma capturar características más globales, y no se aprenda el mismo patrón todo el rato.

Dropout y Dense

Dropout apaga el 20% de las neuronas durante el entrenamiento, previniendo un problema llamado overfitting, donde el modelo aprende patrones irrelevantes.

Dense crea una capa donde cada neurona está conectada a todas las salidas de la capa anterior, la cantidad de valores que se definan en Dense se refieren a la cantidad de salidas que tendrá esa capa. Además, en Dense nosotros usamos softmax que convierte la salida en probabilidad para cada letra.

Optimizer

Un optimizer en redes neuronales es el que decide como se actualizan los pesos a partir de calculos de gradientes de funcion de perdida, en nuestro caso usamos uno llamado Adam (Adaptive Moment Estimation), el cual combina dos ideas, el Momentoum y RMSProp, en el primer guarda un promedio de gradientes para evitar que la actualizacion de valores se vuelva erratica y en el segundo ajusta la tasa de aprendizaje según la magnitud de sus gradientes.

Un gradiente es un vector que indica direccion y magnitud del cambio que deben tener los pesos, si el gradiente es positivo significa que aumentar el peso aumenta la perdida, por tanto hay disminuir, y viceversa.

Funciones de perdida

Las funciones de perdida miden que tan lejos están las predicciones del modelo respecto a las etiquetas reales, nosotros usamos sparce categorical, porque usamos distintas clases, es decir, del 0-25. Es decir, dentro de la red hay una serie de probabilidades (softmax) de que diga una letra, loss lo que hace es calcular con base en la probabilidad actual que tan cerca está de ser segura y correcta, entre menor sea la perdida más segura es.

Callbacks

Las callbacks son funciones que se ejecutan mientras se entrena la IA, en nuestro caso tenemos 3, EarlyStopping, que básicamente la precisión no mejora luego de 10 épocas, detiene el entrenamiento y guarda.

ReduceLROnPlateau, reduce la taza de aprendizaje si la perdida no mejora después de 5 épocas.

ModelCheckpoint, guarda el mejor modelo conseguido actualmente, basándonos en el valor de su precisión.