**Actividad 4.1 - Ejercicio de clasificación con Keras**

Keras es una biblioteca de código abierto escrita en Python que tiene por propósito la generación y experimentación con modelos de aprendizaje profundo (deep learning). Su interfase de alto nivel permite que el usuario se enfoque en la parte productiva del ciclo de desarrollo de Ciencia de Datos; y no necesariamente en los detalles técnicos y complejidades subyacentes de la construcción de algoritmos de aprendizaje profundo. A su vez, esto acelera el ciclo de análisis, pues los científicos de datos pueden pasar más tiempo en la interpretación de resultados y en la aplicación de los modelos, en lugar de pasar tiempo en la codificación y depuración de los algoritmos de bajo nivel.

Esta biblioteca fue desarrollada con el enfoque de permitir la experimentación rápida y el prototipado, con la idea de poder pasar de la idea al resultado con la menor demora posible. Keras se puede ejecutar en varios motores de cálculo de aprendizaje profundo, siendo los más notables TensorFlow, Microsoft Cognitive Toolkit y Theano.

Algunas de las principales características de Keras incluyen:

1. Soporte para redes neuronales convolucionales (CNNs), redes neuronales recurrentes (RNNs), y una combinación de ambas.
2. Soporte para arquitecturas de redes de múltiples entradas y múltiples salidas.
3. Herramientas para trabajar con imágenes y texto, lo que facilita la experimentación con estas formas de datos.
4. Funcionalidades para manejar la regularización, la normalización, las optimizaciones, y mucho más.

Cabe señalar que Keras fue oficialmente adoptado por TensorFlow (una biblioteca de aprendizaje automático de Google) como su interfaz de alto nivel, lo que significa que Keras está integrado en TensorFlow y se beneficia de todas las funciones y mejoras de esta biblioteca.

En este caso, se llevó a cabo la construcción de un modelo de Red Neuronal Convolucional (CNN, por sus siglas en inglés) para la detección de peatones utilizando imágenes. El uso de CNNs se ha vuelto muy popular en la detección de objetos en imágenes, gracias a su capacidad para extraer características importantes a través de diferentes niveles de abstracción.

El proceso inició con el preprocesamiento de las imágenes, que implicó la lectura, redimensionamiento y normalización de las imágenes. Esta etapa es crucial ya que prepara los datos para la alimentación de la red neuronal. Posteriormente, las imágenes preprocesadas fueron divididas en dos categorías, una etiquetada como 'peatón' y la otra como 'no peatón', creando así un conjunto de datos etiquetados que facilita el entrenamiento supervisado.

El diseño de la CNN incluyó varias capas convolucionales intercaladas con capas de agrupamiento o "pooling". Las capas convolucionales son responsables de la detección de características visuales en las imágenes, mientras que las capas de pooling reducen la dimensionalidad de los datos, lo que ahorra tiempo de cálculo y ayuda a evitar el sobreajuste.

Después de las capas de convolución y pooling, la entrada se aplanó para poder ser alimentada a las capas densas, que son responsables de la clasificación de las características aprendidas en las capas previas. Se incorporó una capa de abandono o "dropout" después de las capas densas para mejorar la generalización del modelo y minimizar el sobreajuste. Finalmente, la capa final, una capa densa con una sola neurona y una función de activación sigmoide, generó la salida del modelo, proporcionando la probabilidad de que la imagen contenga un peatón.

El modelo se compiló con una función de pérdida de entropía cruzada binaria, dado que el problema de detección de peatones es un problema de clasificación binaria, y se utilizó el optimizador 'rmsprop'.

Se realizó una validación cruzada estratificada de 5 divisiones para evaluar la eficacia del modelo, proporcionando una visión más precisa de cómo se desempeñará el modelo en datos no vistos.

Los resultados de la validación cruzada, representados en forma de gráficos de precisión y pérdida para cada pliegue, ofrecieron un medio visual para evaluar el rendimiento del modelo a lo largo de las épocas, tanto en el conjunto de entrenamiento como en el de validación. Al final, se calculó la precisión promedio y la desviación estándar de los 5 pliegues, proporcionando una medida general del rendimiento del modelo.

En resumen, la CNN demostró ser una herramienta eficaz para la detección de peatones. El uso de la validación cruzada proporcionó una evaluación robusta del modelo, y la inclusión de capas de dropout y pooling ayudó a minimizar el sobreajuste.