Proyecto Final

Integrantes

Juan David Ríos Rodríguez

Prof. Raul Ramos Pollan

Fundamentos de Deep Learning

Universidad de Antioquia

2023-2

Introducción

En el contexto de la creciente importancia de la inteligencia artificial y el aprendizaje profundo, el presente proyecto se sumerge en el ámbito de la clasificación de emociones a partir de rostros humanos. El objetivo primordial es desarrollar un sistema capaz de discernir y comprender las expresiones emocionales de las personas en diversas situaciones. Este sistema no solo tiene aplicaciones prácticas, como la detección de estados de ánimo en tiempo real y la retroalimentación en interacciones en línea, sino que también contribuye a la investigación en psicología al ofrecer una herramienta automatizada para el análisis de expresiones faciales.

El proyecto se apoya en el conjunto de datos "fer2013", que proporciona imágenes en escala de grises de rostros humanos clasificados en siete categorías emocionales: enojo, disgusto, miedo, felicidad, neutral, tristeza y sorpresa. La tarea principal es entrenar un modelo de aprendizaje profundo que pueda aprender patrones complejos en estas imágenes y realizar predicciones precisas de las emociones asociadas.

Este informe ejecutivo abordará la estructura y desarrollo de los notebooks proporcionados, describirá la solución implementada, detallará las iteraciones llevadas a cabo durante el desarrollo y presentará los resultados obtenidos. A través de este documento, buscamos proporcionar una visión completa del proceso, desde la preparación de datos hasta la evaluación del modelo, destacando las decisiones clave tomadas y reflexionando sobre los resultados alcanzados.

Descripción de la estructura de los notebooks entregados

Notebook #1: 01-Red-convolucional

Paso-1: Importar dataset desde Kagle.

1.1 Preparación del entorno en Google Colab para interactuar con Kaggle: Esta sección tiene como objetivo facilitar la interacción con la plataforma Kaggle. Primero, se permite la carga de archivos mediante files.upload(). Luego, se configura la autenticación con Kaggle al crear un directorio oculto llamado .kaggle, mover el archivo de configuración kaggle.json y establecer permisos de seguridad. Por último, se ajusta una variable de entorno para indicar a Kaggle la ubicación de las credenciales.

1.2 Descargar el dataset: Se utiliza el comando !kaggle datasets download para descargar el conjunto de datos "fer2013" de Kaggle.

1.3 Descomprimir el Dataset y crear directorio de datos: Se descomprime el archivo "fer2013.zip" en el directorio "/content/fer2013/". Luego, con !ls, se muestra el contenido del directorio recién creado para verificar la extracción correcta del conjunto de datos. Posteriormente, se asigna la ruta del directorio a la variable dataset_dir para facilitar el acceso a los archivos del conjunto de datos en el código.

Paso-2: Lectura y Etiquetado de Imágenes.

2.1 lectura y etiquetado: Se crean las listas para almacenar las imágenes y etiquetas, posteriormente se añade el diccionario de etiquetas donde a cada emoción le corresponde un número como valor.

2.2 Redimensión a 48px y conversión a escala de grises: Se establece el valor deseado de 48 para el tamaño de las imágenes, luego se crea un bucle que recorre las etiquetas y archivos los agregue a la dirección correcta del diccionario, las redimensiona a 48x48 y las pasa a escala de grises.

Se convierten ambas listas en matrices Numpy y se normalizan los pixeles de las imágenes en el rango [0-1].

2.3 Crear conjuntos de entrenamiento y prueba: Se divide el conjunto de datos en conjuntos de entrenamiento y prueba. La función toma como entrada las listas de imágenes (images) y etiquetas (labels) y las divide en cuatro conjuntos: train_images (imágenes de entrenamiento), test_images (imágenes de prueba), train_labels (etiquetas de entrenamiento) y test_labels (etiquetas de prueba).

Paso-3: Crear el modelo de red convolucional.

3.1 Modelo de red neuronal convolucional: Definición de un modelo de red neuronal convolucional utilizando la API Sequential de Keras.

3.2 Compilación y entrenamiento del modelo: En la compilación, se utiliza el optimizador 'adam', la función de pérdida 'sparse_categorical_crossentropy' (adecuada para la clasificación de múltiples categorías) y se monitorea la métrica de precisión ('accuracy'). Posteriormente, el modelo se entrena utilizando el generador de aumento de datos (datagen) mediante la función fit. Se especifica el número de épocas (12 en este caso) y se proporcionan los datos de entrenamiento y prueba.

3.3 Precisión del conjunto de prueba: Porcentaje de precisión para el conjunto de prueba.

Paso-4: Métricas de Desempeño.

4.1 Matriz de Confusión: Se genera la matriz de confusión comparando las etiquetas reales con las predichas. Finalmente, se genera una visualización de la matriz de confusión utilizando ConfusionMatrixDisplay de scikit-learn y la biblioteca seaborn.

4.2 Reporte de Clasificación: Métricas detalladas del rendimiento del modelo para este caso (Precisión, Recall, F1-Score, support) este reporte brinda una visión más completa del rendimiento del modelo en comparación con métricas simples como la precisión.

4.3 Gráficos de Pérdida y Precisión: La figura generada incluye dos subgráficos. El primero muestra la pérdida en las épocas de entrenamiento y validación, proporcionando información sobre cómo la pérdida del modelo evoluciona a lo largo del tiempo. El segundo subgráfico representa la precisión del modelo en las épocas de entrenamiento y validación, lo que ofrece perspectivas sobre la mejora del rendimiento en la clasificación de las emociones a medida que avanza el entrenamiento.

Paso-5: Métricas de Negocio.

5.1 Tiempo de Inferencia: Toma las primeras 10 imágenes del conjunto de prueba como ejemplo, mide el tiempo que le lleva al modelo realizar predicciones para estas imágenes y luego calcula el tiempo promedio por predicción.

5.2 Sensibilidad y Especificidad:
La sensibilidad se calcula como el número de verdaderos positivos dividido por la suma de verdaderos positivos y falsos negativos.
La especificidad se calcula como el número de verdaderos negativos dividido por la suma de verdaderos negativos y falsos positivos.

Notebook #2: 02-Iteraciones

Paso-1: Importar dataset desde Kagle.

En esta sección se desarrollan los pasos para importar,descomprimir y descargar el dataset desde Kaggle, agrupa en un mismo bloque los pasos 1.1 a 1.3 del primer Notebook.

- **Preparación del entorno en Google Colab para interactuar con Kaggle.**
- **Descargar el dataset.**

- **Descomprimir el Dataset y crear directorio de datos.**

Paso-2: Lectura y Etiquetado de Imágenes.

En esta sección se agrupan el código del paso 2 todos en un solo bloque lo que corresponde a la fase de preprocesamiento de datos, para este caso serían:

- **lectura y etiquetado.**
- **Redimensión a 48px y conversión a escala de grises.**
- **Crear conjuntos de entrenamiento y prueba.**

Paso-3: Modelos Alternativos para la red convolucional.

3.1 Primer Modelo que implemente: Creación, compilación y entrenamiento de un modelo de red neuronal convolucional (CNN) utilizando la biblioteca Keras. Después de definir el modelo, se compila utilizando el optimizador 'adam', la función de pérdida 'sparse_categorical_crossentropy' y se selecciona la métrica de precisión ('accuracy'). Luego, se entrena el modelo utilizando datos de entrenamiento y validación, con un total de 30 épocas y un tamaño de lote de 64.

3.2 Segundo Modelo que implemente: Creación, compilación y entrenamiento de otro modelo de red neuronal convolucional (CNN) utilizando Keras. Este tiene una arquitectura más compleja que el modelo anterior y utiliza capas de dropout para reducir el sobreajuste.

3.3 Tercer Modelo que implemente: Creación, compilación y entrenamiento de otro modelo de red neuronal convolucional (CNN) utilizando Keras. La red incluye capas convolucionales seguidas de capas de agrupación máxima Después de estas capas, se utiliza una capa de aplanado para preparar los datos para capas densas, que incorporan funciones de activación ReLU y una capa de dropout para evitar el sobreajuste. La capa de salida utiliza softmax para asignar probabilidades a cada clase emocional.

Descripción de la Solución

Preprocesamiento de Datos:

Las imágenes se cargan, redimensionan a 48x48 píxeles y se normalizan para ajustar los valores de píxeles en el rango [0, 1].

Arquitectura de la Red Neuronal:

La CNN se compone de varias capas:

Convolucionales y Pooling: Se utilizan tres bloques de capas convolucionales, cada uno seguido por una capa de pooling (max pooling) para extraer características relevantes de las imágenes.

Capa Flatten: La salida de la última capa de pooling se aplana para convertirla en un vector unidimensional, preparándola para la entrada a las capas totalmente conectadas.

Capas Densas (Totalmente Conectadas): Se incluyen dos capas densas, la primera con 512 neuronas y la función de activación ReLU, y la segunda con 7 neuronas de salida representando las clases de emociones posibles, con la función de activación Softmax.

Compilación y Entrenamiento del Modelo:

Se compila el modelo utilizando el optimizador 'adam' y la función de pérdida 'sparse_categorical_crossentropy', ya que se trata de un problema de clasificación múltiple. Durante el entrenamiento, se utiliza el generador de aumento de datos para mejorar la generalización del modelo. El entrenamiento se realiza a lo largo de 12 épocas.

Evaluación del Modelo:

Se evalúa el modelo en el conjunto de prueba para calcular la precisión y la pérdida. Además, se generan métricas de negocio, como el tiempo de inferencia, la sensibilidad y la especificidad, para evaluar su aplicabilidad en aplicaciones en tiempo real y la capacidad de detectar emociones específicas.

Iteraciones

Entre las posibles modificaciones en el desarrollo del modelo, la única que demostró un impacto significativo en los resultados del proyecto fue la fase de implementación de diferentes arquitecturas. En este caso, se llevaron a cabo un total de tres

implementaciones distintas, y se evaluaron utilizando la métrica de exactitud (accuracy). El propósito principal de estas iteraciones fue identificar la arquitectura de modelo que exhibiera la capacidad de predicción más destacada. La información detallada de cada iteración se presenta a continuación:

1- Primera iteración:

Descripción: Proporciona un modelo simple con una estructura de red neuronal convolucional básica para la clasificación de emociones.

Arquitectura implementada: En la sección 2.2 Arquitectura de la red neuronal en el módulo 2 Descripción de la solución se encuentra de forma detallada la estructura y capas que componen este modelo.

Resultados: El proceso de entrenamiento toma en total 12 épocas para alcanzar un porcentaje de precisión en promedio para el conjunto de prueba de 55%, el proceso de entrenamiento es relativamente rápido comparado con sus sucesores y alcanza en pocas épocas el pico máximo de precisión haciéndolo el modelo más eficiente por su simplicidad, velocidad y precisión.

2- Segunda Iteración:

Descripción: presenta un modelo más complejo con respecto al anterior. Busca una mejora en la capacidad de generalización del modelo mediante el uso de capas de dropout adicionales.

Arquitectura Implementada:

Capas Convolucionales, Max Pooling y Dropout: Se utilizan tres bloques de capas convolucionales, cada uno seguido por una capa de max pooling y una capa de dropout. Las capas de dropout ayudan a reducir el sobreajuste al apagar aleatoriamente un porcentaje de las neuronas durante el entrenamiento.

Capa Flatten: La salida de la última capa de pooling se aplana para prepararla para las capas totalmente conectadas.

Capas Densas (Totalmente Conectadas) con Dropout: Se incluyen dos capas densas, la primera con 512 neuronas y la función de activación ReLU, junto con una capa de dropout para reducir el sobreajuste. La segunda capa densa tiene 7 neuronas de salida representando las clases de emociones, con la función de activación Softmax.

Resultados: Este modelo presenta varios problemas ya que es demasiado lento durante el proceso de entrenamiento, tiene un total de 12 capas y el promedio de predicción es aproximadamente es del 54% igual que el anterior, pero el tiempo para alcanzar ese resultado es 4 veces mayor..

3- Tercera Iteración:

Descripción: Presenta una arquitectura más compleja y completa comparado con sus antecesores, utiliza tres capas convolucionales y de agrupación máxima para extraer características de las imágenes faciales. Luego, se aplica una capa de aplanado seguida de una capa densa de 128 unidades con desactivación Dropout para evitar el sobreajuste.

Arquitectura Implementada:

Capas Convolucionales y de Agrupación Máxima:

- La primera capa convolucional tiene 32 filtros (3x3) con activación ReLU, seguida de una capa de agrupación máxima.
- La segunda capa convolucional tiene 64 filtros (3x3) con activación ReLU, seguida de otra capa de agrupación máxima.
- La tercera capa convolucional tiene 128 filtros (3x3) con activación ReLU, seguida de una capa de agrupación máxima.

Capa de Aplanado:

Una capa de aplanado transforma la salida de las capas convolucionales en un vector unidimensional.

Capa Densa y Desactivación Dropout:

- Una capa densa de 128 unidades con activación ReLU ayuda a aprender características más abstractas y complejas.
- Se añade una capa de desactivación Dropout con una tasa del 0.5 para reducir el sobreajuste al eliminar aleatoriamente conexiones durante el entrenamiento.

Capa de Salida:

La capa de salida tiene 7 unidades con activación softmax, representando las siete emociones posibles. Esta capa devuelve las probabilidades de pertenencia a cada clase emocional.

Resultados: A pesar de tener una arquitectura más elaborada el modelo no entrega resultados mejores, el promedio de precisión es el mismo y el tiempo que toma entrenarlo es mayor por lo que sigue siendo para este problema de clasificación ineficiente comparado con el primer modelo que implemente.

Descripción de los resultados

Nota: Cada que se ejecuta el notebook los resultados pueden sufrir leves variaciones.

Precisión en el Conjunto de Prueba: Los resultados muestran que el modelo tiene una precisión general del 55%, lo que significa que clasifica correctamente el 55% de las muestras en el conjunto de prueba.

Matriz de Confusión: Las clases "angry", "happy" y "surprise" muestran una cantidad significativa de clasificaciones correctas, ya que los valores en sus filas y columnas son relativamente altos. Sin embargo, las clases "disgust" y "fear" tienen tasas de clasificación más bajas, indicando desafíos en la identificación precisa de estas emociones.

Reporte de Clasificación: La clase "happy" tiene la precisión y recall más altos, con un F1-score del 77%, indicando una buena capacidad del modelo para identificar esta emoción. Sin embargo, las clases "disgust" y "fear" tienen un rendimiento relativamente inferior, con F1-scores del 45% y 40%, respectivamente.

Tiempo de Inferencia Promedio: El tiempo de inferencia promedio por predicción es bastante bajo, con 0.0099 segundos, lo que sugiere que el modelo es rápido en hacer predicciones en datos de prueba individuales

Sensibilidad y Especificidad: La sensibilidad es del 57.89%, lo que significa que el modelo es capaz de identificar correctamente el 57.89% de las instancias positivas. La especificidades del 98.22%, indicando una alta capacidad del modelo para identificar correctamente las instancias negativas.

Datos

1- En el repositorio se adjunta el archivo ‘Kaggle.json’ necesario para importar el dataset desde kaggle.

2- En caso de que lo solicite al descomprimir los archivos del dataset ingresar la letra ‘A’ para descomprimir todos los archivos.

Conclusiones

1- Tras hacer una evaluación de los resultados y la comparación de los modelos implementados se puede inferir que el modelo carece del número de imágenes necesarias para poder alcanzar un grado de precisión más alta, las clases que poseían conjuntos de imágenes más grandes fueron las que mejores resultados obtuvieron.

2- Una arquitectura más compleja no entrega necesariamente mejores resultados en este caso el modelo más sencillo tardó menos en ser entrenado y alcanzó el pico de precision relativamente rápido (muy pocas épocas).

3- La matriz de confusión y el reporte de clasificación revelan dificultades particulares en la clasificación de emociones específicas, como "disgust" y "fear". Estas emociones muestran tasas de clasificación más bajas, indicando que el modelo tiene problemas para identificar patrones distintivos asociados con estas categorías.

Referencias

Referencias y resultados previos:

Curso de Fundamentos de Deep Learning (UdeA) del profesor Raúl Ramos Pollán: 04 - CONVOLUTIONAL NETWORKS — Fundamentos de Deep Learning (rramosp.github.io)

"Deep Learning" de Ian Goodfellow, Yoshua Bengio y Aaron Courville.
Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.