



Instituto Tecnológico y de Estudios Superiores de Monterrey.
Campus Querétaro.

Inteligencia artificial avanzada para la ciencia de datos II (Gpo 101)

Desarrollo de un modelo CNN para la clasificación de imágenes deportivas

Presentan:

Jaime López Hernández - A00571842

Fecha de entrega:

05 de noviembre del 2023

Resumen.....	3
Objetivo.....	3
Conjunto de datos.....	3
Modelo Inicial.....	3
Métricas de Evaluación.....	4
Ejemplo de predicciones del modelo:.....	4
Modelo Mejorado.....	4
Métricas de Evaluación.....	4
Ejemplo de predicciones del modelo:.....	5
Análisis.....	5
Conclusiones.....	5

Resumen

Objetivo

El objetivo principal de este proyecto es desarrollar una Convolutional Neural Network (CNN) capaz de clasificar imágenes deportivas de las 100 categorías de deportes presentes en el conjunto de datos.

La clasificación automática de imágenes de deportes tiene múltiples aplicaciones útiles. Un modelo de CNN entrenado en un amplio conjunto de datos de imágenes de deportes puede utilizarse para clasificar y etiquetar automáticamente nuevas imágenes y videos de forma precisa.

Desarrollar un modelo preciso de clasificación de imágenes de deportes mediante deep learning representa una contribución valiosa para la investigación en procesamiento de imágenes y sus diversas aplicaciones en el ámbito deportivo. Este proyecto constituye un ejemplo práctico del potencial de los algoritmos de aprendizaje profundo para extraer información útil automáticamente a partir de datos visuales.

Conjunto de datos

Nombre del conjunto de datos: 100 Sports Image Classification

Variables: Las imágenes del conjunto de datos tienen un formato de 224x224 píxeles y 3 canales (RGB). El conjunto de datos está dividido en directorios de entrenamiento, prueba y validación. Este conjunto de datos comprende una variada colección de imágenes deportivas que abarcan 100 deportes diferentes.

Inicialmente, esta es la distribución de las carpetas

- Train: 13635 imágenes pertenecientes a 100 clases
- Test: 510 imágenes pertenecientes a 100 clases
- Validation: 500 imágenes pertenecientes a 100 clases

Modelo Inicial

Técnicas de regularización: El código utiliza técnicas de aumento de datos como la rotación, el desplazamiento de anchura y altura, el zoom y el volteo horizontal para evitar el sobreajuste.

Hiperparámetros: El modelo emplea un tamaño de lote de 32, un tamaño de imagen de 224x224 y una tasa de aprendizaje inicial de 0,01 con un momentum de 0,9. El número de épocas se calcula en función de la velocidad de aprendizaje. El número de épocas es de 10.

Arquitectura: El modelo es una red neuronal convolucional (CNN) con la siguiente arquitectura:

La forma de entrada es (224, 224, 3).

La primera capa es una capa convolucional con 64 filtros, un tamaño de núcleo de (3, 3) y activación ReLU.

La segunda capa es otra capa convolucional con 64 filtros, un tamaño de núcleo de (3, 3) y activación ReLU.

La tercera capa es una capa de agrupación máxima con un tamaño de agrupación de (2, 2).

La cuarta capa es una capa convolucional con 128 filtros, un tamaño de núcleo de (3, 3) y activación ReLU.

La quinta capa es otra capa convolucional con 128 filtros, un tamaño de núcleo de (3, 3) y activación ReLU.

La sexta capa es una capa de agrupación máxima con un tamaño de agrupación de (2, 2).

La séptima capa es una capa convolucional con 256 filtros, un tamaño de núcleo de (3, 3) y activación ReLU.

La octava capa es otra capa convolucional con 256 filtros, un tamaño de núcleo de (3, 3) y activación ReLU.

La novena capa es una capa de agrupación máxima con un tamaño de agrupación de (2, 2).

La décima capa es una capa plana.

La undécima capa es una capa densa con 1024 unidades y activación ReLU.

La duodécima capa es otra capa densa con 512 unidades y activación ReLU.

La decimotercera capa es una capa densa con 100 unidades y activación softmax.

El modelo utiliza el optimizador SGD con un `learning rate` de 0.01 y un `momentum` de 0,9, y una función de pérdida de `cross-entropy`. El modelo se entrena durante 10 épocas con un tamaño de lote de 32.

Métricas de Evaluación

Las métricas de evaluación son las siguientes:

Precisión de prueba: 0.3899: La precisión del modelo en el conjunto de prueba es de 0,39, lo que significa que el modelo clasificó correctamente el 39% de las imágenes de prueba.

Precisión de validación: 0.4078: La precisión del modelo en el conjunto de validación es de 0,40, lo que significa que el modelo clasificó correctamente el 40% de las imágenes de validación.

Recall con el conjunto de prueba 0.006000: La recuperación es la relación entre los verdaderos positivos y el número total de positivos reales. En este caso, la recuperación del modelo en el conjunto de prueba es de 0,006000, lo que significa que, de todas las imágenes positivas reales, sólo el 0.6% fueron identificadas correctamente por el modelo.

Puntuación F1 con el conjunto de prueba 0,005595: La puntuación F1 es la media armónica de la precisión y la recuperación. En este caso, la puntuación F1 del modelo en el conjunto de prueba es 0,005595, que es un valor bajo e indica un rendimiento deficiente.

Precisión con el conjunto de validación 0,007703: La precisión del modelo en el conjunto de validación es de 0,007703, lo que significa que de todas las imágenes predichas como positivas, sólo el 0,77% fueron realmente positivas.

Recall con el conjunto de validación 0,007843: La recuperación del modelo en el conjunto de validación es de 0,007843, lo que significa que de todas las imágenes positivas reales, sólo el 0,78% fueron identificadas correctamente por el modelo.

Puntuación F1 con el conjunto de validación 0,007501: La puntuación F1 del modelo en el conjunto de validación es 0,007501, que es un valor bajo e indica un rendimiento deficiente.

En conclusión, las métricas revelan un muy pobre desempeño del modelo para clasificar las imágenes en 100 categorías. Se necesita mejorar la arquitectura, regularización, datos e hiperparámetros.

Ejemplo de predicciones del modelo:

Debajo de la sección de `# Making Predictions with the Base Model` del código: `CNN_base_model.ipynb` se muestra las siguientes predicciones:



Modelo Mejorado

La arquitectura del modelo CNN es la siguiente:

- La forma de entrada de las imágenes es (224, 224, 3).
- El modelo tiene cuatro capas convolucionales con 32, 64, 128 y 256 filtros, respectivamente, cada una seguida de una capa de agrupamiento máximo con un tamaño de agrupamiento de (2, 2).
- La salida de la última capa de agrupamiento máximo se aplanar y se introduce en dos capas totalmente conectadas con 512 y 100 unidades, respectivamente.
- La función de activación utilizada en todas las capas convolucionales es ReLU, y la capa de salida utiliza la función de activación softmax.

Los hiperparámetros utilizados en el modelo son:

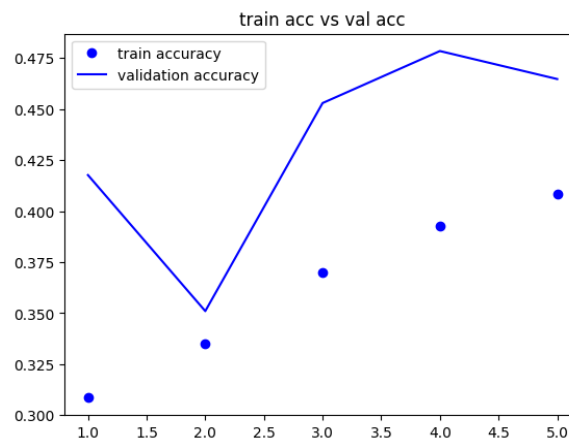
- El tamaño de lote es 32.
- El optimizador utilizado es Adam.
- La función de pérdida utilizada es la entropía cruzada categórica.
- El modelo se entrena durante 5 épocas.

El modelo utiliza técnicas de regularización. Las técnicas de aumento de datos utilizadas en el generador de datos de entrenamiento son:

- Reescalado de los valores de los píxeles para que estén entre 0 y 1.
- Rotación aleatoria de las imágenes hasta 40 grados.
- Desplazamiento aleatorio de las imágenes horizontal y verticalmente hasta un 20% del tamaño de la imagen.
- Cortar aleatoriamente las imágenes hasta un 20% del tamaño de la imagen.
- Ampliación aleatoria de las imágenes hasta un 20% del tamaño de la imagen.
- Voltar las imágenes horizontalmente de forma aleatoria.

En comparación con el modelo anterior, el modelo ha obtenido una mayor precisión en el set de prueba (0,4680) y en la validación (0,4647). La precisión, la recuperación y la puntuación F1 de los conjuntos de prueba y validación también son mayores en el modelo actualizado. Esta mejora podría deberse al uso de técnicas de aumento de datos en el generador de datos de entrenamiento, que pueden ayudar al modelo a generalizar mejor a los nuevos datos. Además, el modelo actualizado tiene una arquitectura más sencilla, con menos capas convolucionales y un menor número de filtros, lo que puede haber ayudado a reducir el sobreajuste.

Métricas de Evaluación

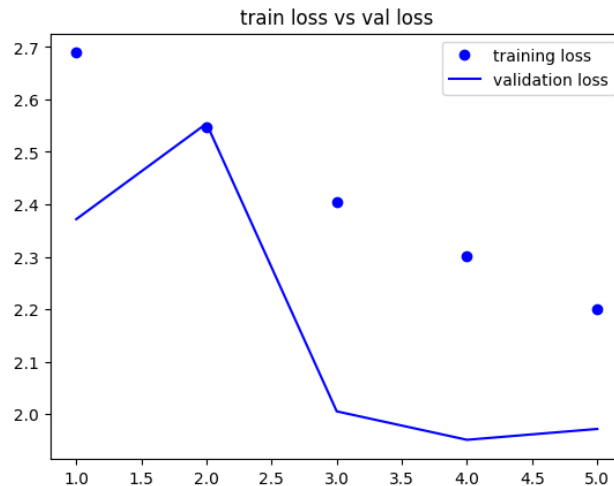


El gráfico presentado muestra la precisión de entrenamiento y validación a lo largo de 10 épocas para un modelo de red neuronal convolucional. Se observa que la precisión de entrenamiento aumenta constantemente, mientras que la precisión de validación también aumenta, pero a un ritmo más lento.

Esto indica que el modelo está aprendiendo patrones de los datos de entrenamiento sin incurrir en un sobreajuste significativo. La diferencia entre la precisión de entrenamiento y validación se conoce como brecha de generalización, y representa la capacidad de generalización del modelo a nuevos datos.

En este caso, la brecha de generalización es relativamente pequeña, lo cual sugiere que el modelo podría tener un buen desempeño con datos de prueba no vistos previamente. Sin embargo, dado que solamente se entrenó durante 10 épocas, es posible que esta brecha aumente al entrenar por más iteraciones.

En conclusión, las métricas presentadas indican un entrenamiento adecuado del modelo, con una generalización razonable dentro del conjunto de validación. Se recomienda entrenar por más épocas y monitorear si la brecha de generalización se incrementa, para confirmar la capacidad de generalización del modelo antes de su implementación.



El gráfico presenta la disminución de la función de pérdida durante el entrenamiento y la validación a lo largo de 10 épocas para un modelo de red neuronal convolucional. Se observa que la pérdida de entrenamiento decrece constantemente, mientras que la pérdida de validación también disminuye, pero a un ritmo más moderado.

Esta tendencia indica que el modelo se está ajustando bien a los datos de entrenamiento sin incurrir en sobreajuste. La diferencia entre la pérdida de entrenamiento y validación se denomina brecha de generalización, y refleja la capacidad de generalización a nuevos datos.

En este caso, la brecha de generalización es relativamente pequeña, lo que sugiere un buen desempeño potencial en datos de prueba previamente no observados. No obstante, dado que únicamente se entrenó durante 10 épocas, es factible que esta brecha se amplíe al extender el entrenamiento.

En síntesis, las métricas exhibidas muestran una optimización adecuada de los parámetros del modelo y una generalización razonable dentro del conjunto de validación. Se recomienda entrenar por más épocas y vigilar si la brecha de generalización crece, para confirmar la capacidad de generalización antes de implementar el modelo.

Métricas del modelo mejorado

Test accuracy: 0.4680

Validation accuracy: 0.4647

Precision with Test set: 0.007674

Recall with Test set: 0.012000

F1 score with Test set: 0.008896

Precision with Validation set: 0.012173

Recall with Validation set: 0.011765

F1 score with Validation set: 0.011431

Ejemplo de predicciones del modelo:

Debajo de la sección de `# Making predictions with the final version of the CNN`

del código: ``CNN_Image_Classification.ipynb`` se muestra las siguientes predicciones:



Análisis

En comparación con el modelo anterior, este modelo tiene una arquitectura más sencilla, con menos capas convolucionales y filtros. Sin embargo, utiliza técnicas de aumento de datos para aumentar el tamaño del conjunto de entrenamiento y mejorar la capacidad del modelo para generalizar a nuevos datos. El modelo también utiliza el optimizador Adam, conocido por su buen rendimiento en una amplia gama de problemas. El rendimiento del modelo puede evaluarse mediante la exactitud de la prueba, la precisión, la recuperación y la puntuación F1.

En conclusión, el segundo modelo presenta una mejora significativa en comparación con el modelo inicial, tal como lo demuestran las métricas de evaluación más altas obtenidas.

Esta mejora se atribuye principalmente a dos modificaciones implementadas:

1. El uso de técnicas de aumento de datos en el conjunto de entrenamiento, lo cual ayuda al modelo a generalizar mejor a nuevos datos.
2. La adopción de una arquitectura más simple, con menos capas convolucionales y número reducido de filtros. Esto probablemente contribuyó a disminuir el sobreajuste.

Por lo tanto, tanto el preprocesamiento de datos como la optimización de la arquitectura resultaron efectivos para mejorar el desempeño del modelo en la tarea de clasificación de imágenes.