

Proyecto 2. Filtros sobre imágenes: Clasificación de imágenes con CNN usando optimizadores Adam y SGD en la base Intel Image Classification

Luz Itzel Álvarez Cruz

Diplomado Ciencia de Datos (BUAP)
Abril 2025

1 Introducción

En este proyecto se aborda la tarea de clasificación de imágenes utilizando una red neuronal convolucional (CNN). El objetivo es entrenar un modelo que pueda clasificar imágenes de diferentes clases de acuerdo con el conjunto de datos Intel Image Classification, que contiene imágenes pertenecientes a seis clases distintas: forest, street, mountain, buildings, sea y glacier. Para ello se compara el rendimiento de dos modelos utilizando diferentes optimizadores: Adam y SGD, evaluando su desempeño en términos de precisión y pérdida en el conjunto de entrenamiento y validación.

2 Descripción de los Datos

El conjunto de datos utilizado en este proyecto corresponde al dataset de clasificación de imágenes de Intel. Este conjunto de datos está compuesto por imágenes de seis clases diferentes, cada una con un total de 500 imágenes divididas en dos conjuntos: entrenamiento y prueba.

Las clases son:

- Bosques
- Calles
- Montañas
- Edificios
- Mares
- Glaciares

Las imágenes son de tamaño 150x150 píxeles y tienen una gran variabilidad en cuanto a iluminación, perspectiva y fondo.

3 Preprocesamiento de los Datos

Antes de alimentar los datos a los modelos, se aplicaron varios pasos de preprocesamiento:

Las imágenes fueron cargadas desde dos directorios: uno para el conjunto de entrenamiento (`seg_train`) y otro para el conjunto de validación (`seg_test`). Se dividió el conjunto de entrenamiento en 7000 imágenes y el conjunto de validación en 1500 imágenes, con un número balanceado de imágenes por clase.

Las imágenes fueron redimensionadas a un tamaño estándar de 64x64 píxeles.

Se normalizó la intensidad de los píxeles para que estuvieran en el rango $[0, 1]$, dividiendo por 255. Se dividieron los datos en un conjunto de entrenamiento (80%) y un conjunto de validación (20%).

4 Arquitectura del Modelo

La arquitectura utilizada en este proyecto es una red neuronal convolucional (CNN) básica, diseñada para ser eficiente en la clasificación de imágenes. Esta arquitectura consta de las siguientes capas:

1. Capa convolucional 1: 32 filtros de tamaño 3x3, con activación ReLU.
2. Capa de max-pooling 1: Max-pooling de 2x2 para reducir las dimensiones.
3. Capa convolucional 2: 64 filtros de tamaño 3x3, con activación ReLU.
4. Capa de max-pooling 2: Max-pooling de 2x2.
5. Capa convolucional 3: 128 filtros de tamaño 3x3, con activación ReLU.
6. Capa de max-pooling 3: Max-pooling de 2x2.
7. Capa densa 1: 512 unidades con activación ReLU.
8. Capa de salida: 6 unidades (una para cada clase), con activación softmax.

La salida de la red es una probabilidad para cada clase, la cual es utilizada para determinar la clase final predicha.

5 Parámetros de Entrenamiento

Los modelos fueron entrenados con los siguientes parámetros:

- Número de épocas: 10
- Tamaño de lote: 32
- Función de pérdida: Categorical Crossentropy (para clasificación multiclase).
- Optimización: Se entrenaron dos modelos, uno con el optimizador Adam y otro con el optimizador SGD.
- Métrica de evaluación: Precisión.

6 Resultados del Entrenamiento

1. Modelo con Adam:

- Precisión de entrenamiento:
La precisión comenzó en 36.62% en la primera época y mejoró de manera constante hasta alcanzar 88.94% en la última época.
- Precisión de validación:
La precisión de validación aumentó desde 51.33% en la primera época hasta 66.00% al final.
- Pérdida de entrenamiento:
La pérdida en el conjunto de entrenamiento comenzó en 1.5080 y disminuyó gradualmente hasta 0.3220 al final.
- Pérdida de validación:
La pérdida de validación comenzó en 1.1763 y terminó en 1.1059.

La progresión en la precisión y la pérdida muestra que el modelo Adam convergió de manera eficiente, alcanzando buenos resultados en el conjunto de entrenamiento y mejorando gradualmente su rendimiento en el conjunto de validación.

2. Modelo con SGD:

- Precisión de entrenamiento:
La precisión comenzó en 69.54% en la primera época y aumentó hasta 85.17% al final.
- Precisión de validación:
La precisión de validación comenzó en 64.67% y terminó en 67.20%.
- Pérdida de entrenamiento:
La pérdida en el conjunto de entrenamiento comenzó en 0.9657 y se redujo a 0.4228.
- Pérdida de validación:
La pérdida de validación comenzó en 0.9862 y terminó en 0.9478.

El modelo SGD también mostró una mejora constante en precisión, aunque no alcanzó el mismo rendimiento que el modelo Adam. Esto sugiere que el optimizador Adam pudo haber encontrado una solución más óptima en este caso específico.

7 Evaluación de Desempeño

Matriz de Confusión: La matriz de confusión (Fig. 1) para ambos modelos muestra cómo se distribuyen las predicciones para cada clase. El modelo Adam presentó un desempeño superior, especialmente en la clasificación de las clases de edificios y montañas. Aunque el modelo SGD también fue efectivo, la precisión en la clasificación de algunas clases fue ligeramente inferior.

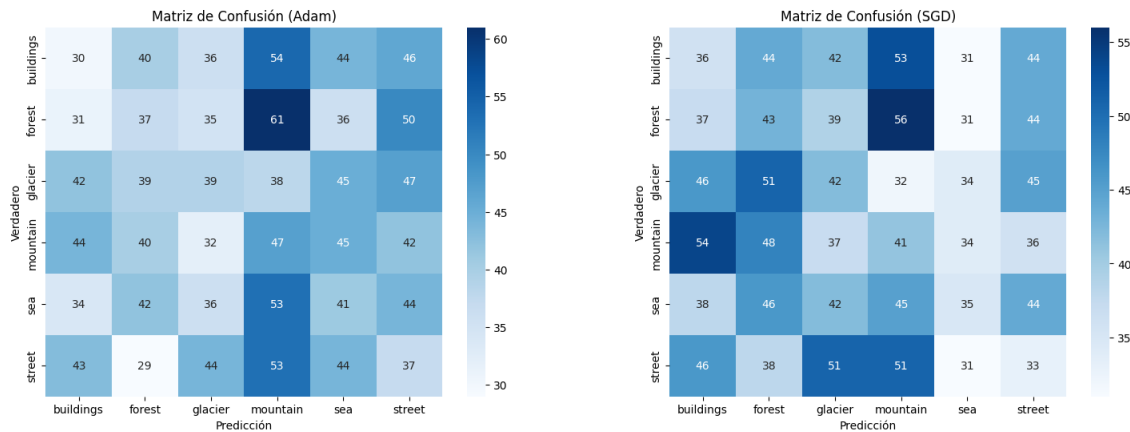


Figure 1: Matrices de confusión para ambos modelos.

8 Conclusiones

- El optimizador Adam superó al optimizador SGD en términos de precisión tanto en el conjunto de entrenamiento como en el de validación.
- El modelo Adam mostró una mejor capacidad para clasificar las imágenes correctamente en la mayoría de las clases, lo que sugiere que Adam es más eficiente para este tipo de tarea de clasificación de imágenes.
- Aunque el modelo con SGD presentó un desempeño competitivo, el modelo con Adam logró una convergencia más rápida y una mayor precisión en