

Deep Learning Report

Lopez Flores Royer Amed¹

¹Facultad de Ciencias, Universidad Nacional de Ingenieria
¹royer.lopez.f@uni.pe

February 28, 2022

Resumen

En este reporte se presenta la comparación de dos modelos para la clasificación de imágenes en 2022, usando como framework Pytorch, el cual nos facilita la creación de modelos con buenos resultados con el cual podemos crear fácilmente pues nos evita el cálculo del backpropagation y nos permite la creación de distintas capas para un modelo de AI. Por ello planteamos dos modelos con arquitectura de capas convolucionales y Resnet.

1 Metodología

En el presente trabajo compraremos dos clasificadores de imágenes con diferentes arquitecturas una basada en redes neuronales convolucionales y otra basada en redes neuronales residuales, esto para ver los cambios y precisión que cada una de estas capas nos ofrece. Usaremos como dataset el "PokemonDataset" de Kaggle, todas estas imágenes serán tratadas antes de entrar a nuestra red neuronal con un tamaño de 224x224 y obtendremos 150 tipos de pokémones para ambos modelos.

1.1 Data augmentation

Se usa esta técnica para poder obtener una mayor cantidad de datos para tener un modelo más robusto, pues nuestro dataset proporcionado por "Kaggle", es de 35 imágenes por cada tipo de pokémon para ello a todas las imágenes crearemos una transformación como una rotación aleatoria, un Flip horizontal y un Flip vertical obteniendo 150 imágenes de cada clase.

1.2 Modelo CNN

Este modelo está conformado por 5 capas, en cada capa tenemos una convolución, un batch normalization,

una función de activación ReLU y como finalmente un MaxPool, todo esto para obtener después una red neuronal lineal para la clasificación de las 150 clases de pokémones. Este modelo es muy importante pues nos permite extraer patrones en las imágenes las cuales después serán pasadas a una red neuronal que clasifique en 150 clases.

1.3 Modelo ResNet

Este modelo está conformado por 4 capas de redes residuales, para lo cual cada capa residual tiene la siguiente secuencia de capas: una capa convolucional, batch normalization, la función de activación ReLU, donde repetimos la misma secuencia de capas mencionadas, para agregarle una capa de de que unira la entrada con la salida de nuestra capa residual. Esta es una arquitectura más compleja que la anterior pues una sola capa de RNN tiene varias capas de un convolucional además de tener conexiones con las capas anteriores para no tener pérdidas en los gradientes del backpropagation.

2 Resultados

2.1 Model CNN

Como se explico este es un modelo básico de 5 capas de capas convolucionales, con este modelo pudimos obtener el siguiente gráfico de precisión en al entrenar el modelo Figura 1 y las predicciones que realiza Figura 2.

2.2 Modelo ResNet

Este modelo tuvo un rendimiento similar al anterior modelo pues como se observa en la Figura 3 y observamos otras predicciones en la Figura 4.

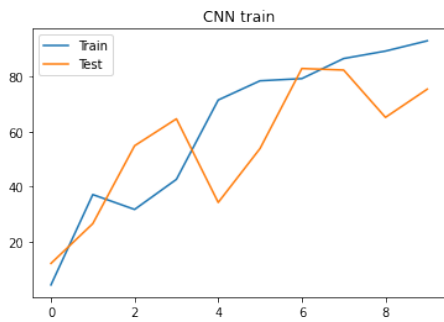


Figura 1: train accuracy vs test accuracy del modelo CNN

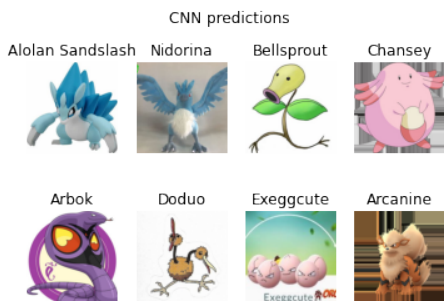


Figura 2: predicciones del modelo CNN

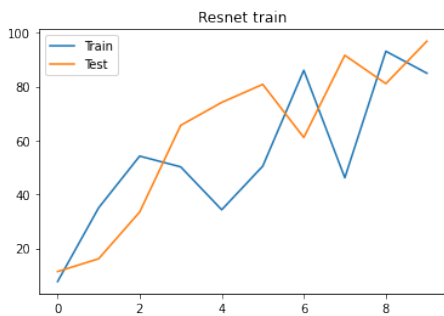


Figura 3: train accuracy vs test accuracy del modelo ResNet

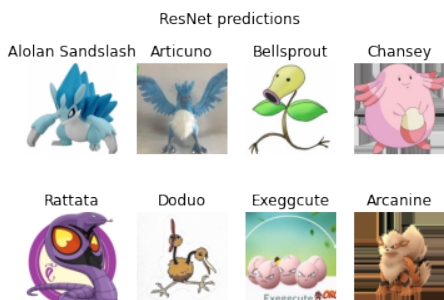


Figura 4: predicciones del modelo ResNet

3 Discusión de resultados

Se observó que ambos modelos al tener el data augmentation, estas llegaban al punto de overfitting pues el train loss era de un 98 % mientras que en el test loss fue de un 50 % con lo cual los modelos mostrados en la sección anterior fue sin la data tratada. Como se ve en el primer modelo este 7 de 8 imágenes bien predecidas, lo cual es un modelo regulamente aceptado además de que su arquitectura no es tan compleja se obtuvo buenas predicciones de parte de este clasificador. Ahora en el caso de el modelo ResNet que necesita de más imágenes y esta arquitectura es más compleja que el CNN pero se obtuvo la misma precisión con las mismas imágenes pero esta red aprendió de una manera distinta a la primera pues en el ejemplo de las 8 imágenes estas predicen no coinciden en la segunda imagen en la cual la primera red falla y la otra acierta con articuno, pero en la quinta imágenes la segunda falla, lo cual nos dice que estas redes aprende de formas distintas.

Conclusiones

Es posible la obtención de overfitting con el uso de data augmentation pues nuestros dos modelos llegaron a tener este problema cuando la data fue aumentada con este método. Las redes neuronales con en una misma situación aprenden de distintas formas pues estas varían en su arquitectura, aunque la ResNet obtuvo un mayor test accuracy lo cual nos dice que la ResNet aprendió a predecir mejor.

Agradecimientos

Un profundo agradecimiento a nuestro profesor Elvin Mark Muños Vega quien compartió sus conocimientos sobre este tema tan intrigante. Además también al grupo de estudiantes del AEPIF por organizar esta escuela de verano de la cual me llevo muchos conocimientos en la cual espero aplicarlo en mi vida profesional.

Referencias

- [1] PyTorch. (2019). Documentación de PyTorch. 27/02/2022, de PyTorch Sitio web: <https://pytorch.org/docs/stable/index.html>
- [2] lantian773030. (2019). 7,000 Labeled Pokemon. 2019-11-09, de Kaggle Sitio web: <https://www.kaggle.com/lantian773030/pokemonclassification>