Universidade Federal do Tocantins - Campus Palmas
Ciência da Computação/9º Período
Redes Neurais
Prof. Marcelo Lisboa
Ítalo Machado Vilarino

Relatório: Treinamento de Rede Neural para Previsão do Tipo Primário de Pokémons

1. INTRODUÇÃO

O objetivo deste relatório é apresentar o trabalho realizado utilizando redes neurais para prever o tipo primário de Pokémons. O trabalho foi desenvolvido em linguagem Python, versão 3.10.0, utilizando as bibliotecas Keras e Scikit-learn. O conjunto de dados consiste em um arquivo CSV contendo informações sobre Pokémons da primeira à sétima geração, incluindo nome, tipo primário e tipo secundário. Além disso, o conjunto de dados também contém um diretório com imagens correspondentes a cada Pokémon.

2. METODOLOGIA

Sobre o pré-processamento dos dados: antes de treinar a rede neural, foi realizado. Isso envolveu a leitura do arquivo CSV e a carga das imagens dos Pokémons. O CSV foi lido utilizando bibliotecas como Pandas para obter as informações dos Pokémons, incluindo os tipos primários. O diretório contendo as imagens foi percorrido para carregar as imagens correspondentes a cada Pokémon, utilizando bibliotecas como OpenCV e NumPy para manipulação e redimensionamento das imagens. Além disso, foram criadas 100 novas imagens manipuladas de cada um dos pokémons, através da rotação, aumento de tamanho ou zoom da mesma, visando escalonar o treinamento e obter melhores resultados. Também foi aplicada a codificação one-hot aos tipos primários para transformá-los em vetores binários.

A arquitetura da rede neural foi implementada usando a biblioteca Keras. Optou-se por uma arquitetura convolucional simples, que consiste em camadas convolucionais, camadas de normalização em lote (batch normalization) e camadas de max pooling. Essas camadas foram seguidas por camadas densas para a classificação final. A quantidade de camadas e o número de filtros foram ajustados empiricamente durante o experimento.

O modelo, por sua vez, foi treinado utilizando a biblioteca Keras com o otimizador Adam e a função de perda “categorical crossentropy”. O conjunto de treinamento foi dividido em conjuntos de treinamento e validação para monitorar o desempenho do modelo durante o treinamento. Foram ajustados hiperparâmetros como taxa de aprendizado, tamanho do lote (batch size) e número de épocas para encontrar a combinação ideal que maximizasse a precisão (acurácia) do modelo.

Após o treinamento, o modelo foi avaliado no conjunto de testes para estimar sua capacidade de generalização. A métrica de avaliação utilizada foi a acurácia, que mede a porcentagem de classificações corretas.

3. RESULTADOS E DISCUSSÃO

Os resultados obtidos mostraram uma acurácia de números superiores a 70% no conjunto de teste, o que indica que o modelo foi capaz de prever corretamente o tipo primário de Pokémons com uma taxa significativa de sucesso. No entanto, é importante destacar que esses resultados podem variar dependendo da complexidade do conjunto de dados, da qualidade das imagens e da quantidade de dados disponíveis para treinamento.

Neste experimento, retirei o pokémon “Abra” do dataset e o usei como modelo para testar a predição:

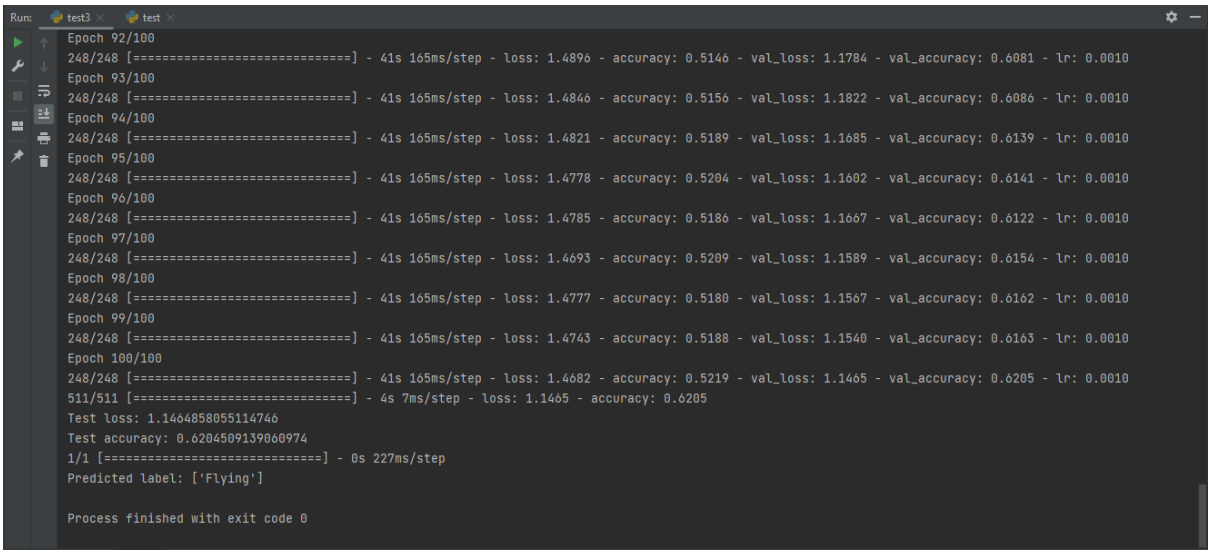
```
187/187 [==============================] - 9s 48ms/step - loss: 1.7926 - accuracy: 0.4086 - val_loss: 1.5788 - val_accuracy: 0.4701 - lr: 0.0010
Epoch 50/50
187/187 [==============================] - 9s 49ms/step - loss: 1.7842 - accuracy: 0.4101 - val_loss: 1.5775 - val_accuracy: 0.4701 - lr: 0.0010
385/385 [==============================] - 1s 2ms/step - loss: 1.5775 - accuracy: 0.4701
Test loss: 1.5775210857391357
Test accuracy: 0.47011464834213257
Saved model to disk
1/1 [==============================] - 0s 116ms/step
Predicted label: ['Psychic']
```

O experimento, mesmo com uma precisão um pouco abaixo da expectativa (foram utilizadas poucas épocas e menores tamanhos de imagem, para agilizar o processo), conseguiu prever com sucesso o tipo do pokémon (‘Psychic’).

Outro detalhe percebido foi que ao diminuir a taxa de aprendizado do otimizador Adam, os números de precisão parecem se tornar mais equilibrados/normalizados, por vezes ocorrendo até um “salto” no seu valor:

```
Epoch 9/20
248/248 [==============================] - 57s 230ms/step - loss: 2.4075 - accuracy: 0.2835 - val_loss: 2.6439 - val_accuracy: 0.2481 - lr: 0.1000
Epoch 10/20
248/248 [==============================] - 60s 242ms/step - loss: 2.3261 - accuracy: 0.2990 - val_loss: 2.8937 - val_accuracy: 0.2676 - lr: 0.1000
Epoch 11/20
248/248 [==============================] - 61s 246ms/step - loss: 2.0855 - accuracy: 0.3624 - val_loss: 1.8953 - val_accuracy: 0.4003 - lr: 0.0100
Epoch 12/20
248/248 [==============================] - 60s 242ms/step - loss: 2.0019 - accuracy: 0.3786 - val_loss: 2.0174 - val_accuracy: 0.4200 - lr: 0.0100
Epoch 13/20
```

O teste abaixo foi um dos que mais foi permitido evoluir. Chegando a 100 épocas e atingindo uma precisão de cerca de 62%, o modelo foi capaz de prever com exatidão o tipo do Pokémon que havia sido colocado na área de avaliação (‘pidgeot’), com o tipo voador.



Portanto, é justo afirmar que o trabalho realizado demonstrou a aplicação e efetividade das redes neurais. A arquitetura da rede neural foi projetada e ajustada para obter o melhor desempenho possível e os resultados obtidos demonstraram a capacidade do modelo de fazer previsões precisas.