Critical Thinking Module 6 Option 1

Benjamin Gutierrez

Colorado State University Global

CSC 580: Applying Machine Learning and Neural Networks - Capstone

Dr. Lori Farr

February 5, 2023

My code uses TensorFlow to train a Convolutional Neural Network (CNN) to categorize photos from the CIFAR10 dataset. The CIFAR10 dataset contains 60,000 color pictures divided into ten categories, with 6,000 images in each (Krizhevsky, 2009). The script begins by importing the CIFAR10 dataset, establishing a reproducibility seed, and disabling SSL verification.

My code then divides each pixel value by 255.0 to normalize the train and test datasets. This phase is critical in guaranteeing that all values in the input are between 0 and 1, as needed by the model's activation functions (Bhandari, 2020).

The TensorFlow Sequential API is then used to define the model. It is made up of many Conv2D layers that apply filters to the input pictures in order to extract useful features (Rosebrock, 2018). The MaxPooling2D layers are used to downsample the picture, decreasing its size while keeping critical information. The Flatten layer converts tensors into 1-dimensional arrays, which are then transferred to a fully linked Dense layer, which produces the final prediction. The leaky relu activation function, a variation of the ReLU activation function that enables tiny negative values to pass through, is used in the Dense layers (Chaudhary, 2020). The Dropout layer prevents overfitting by setting input values to zero at random during training (Brownlee, 2018). The final Dense layer consists of 10 units and a softmax activation function, ensuring that the output is a probability distribution across the ten categories (Saxena, 2021).

After that, the model is constructed using the Adam optimizer and the sparse categorical crossentropy loss, which is used for multi-class classification tasks (Moreno, 2021). The model is then trained on the training dataset for 15 epochs.

Finally, the correctness of the model is tested on the test dataset and printed. The algorithm also predicts the test pictures and presents the first 25 photos from the test dataset with their true and expected labels as seen in Figure 3. If the forecast is right, the label is green; otherwise, it is red. The model's overall accuracy is 74.77% as seen in Figure 2, which is reasonable for a multi-class picture classification issue.

Figures

```
PS C:\Users\Benjamin Gutierrez\Documents\Miscellaneous\csu\csc580\CSC580_CTA_6_1_Gutierrez_Benjamin> python .\CSC580_CTA
_6_1_Gutierrez_Benjamin.py
2023-02-05 22:05:31.871082: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized wit
h oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:
 AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Epoch 1/15
1563/1563 [==============================] - 44s 28ms/step - loss: 1.4950 - accuracy: 0.4600
Epoch 2/15
1563/1563 [==============================] - 41s 26ms/step - loss: 1.1380 - accuracy: 0.6013
Epoch 3/15
1563/1563 [==============================] - 45s 29ms/step - loss: 0.9954 - accuracy: 0.6545
Epoch 4/15
1563/1563 [==============================] - 38s 24ms/step - loss: 0.9065 - accuracy: 0.6864
Epoch 5/15
1563/1563 [==============================] - 47s 30ms/step - loss: 0.8353 - accuracy: 0.7099
Epoch 6/15
1563/1563 [==============================] - 38s 24ms/step - loss: 0.7791 - accuracy: 0.7309
Epoch 7/15
1563/1563 [==============================] - 42s 27ms/step - loss: 0.7312 - accuracy: 0.7469
Epoch 8/15
1563/1563 [==============================] - 40s 25ms/step - loss: 0.6941 - accuracy: 0.7585
Epoch 9/15
1563/1563 [==============================] - 42s 27ms/step - loss: 0.6569 - accuracy: 0.7726
Epoch 10/15
1563/1563 [==============================] - 42s 27ms/step - loss: 0.6220 - accuracy: 0.7828
Epoch 11/15
1563/1563 [==============================] - 41s 26ms/step - loss: 0.5895 - accuracy: 0.7947
Epoch 12/15
1563/1563 [==============================] - 43s 28ms/step - loss: 0.5608 - accuracy: 0.8039
Epoch 13/15
1563/1563 [==============================] - 42s 27ms/step - loss: 0.5307 - accuracy: 0.8124
Epoch 14/15
1563/1563 [==============================] - 45s 29ms/step - loss: 0.5100 - accuracy: 0.8207
Epoch 15/15
1563/1563 [==============================] - 46s 29ms/step - loss: 0.4890 - accuracy: 0.8273
313/313 [==============================] - 3s 10ms/step - loss: 0.8302 - accuracy: 0.7477
Test Accuracy: 74.77%
313/313 [==============================] - 3s 10ms/step
```

Figure 1: Running my code on the CIFAR10 dataset

Figure 2: Test accuracy after training my model

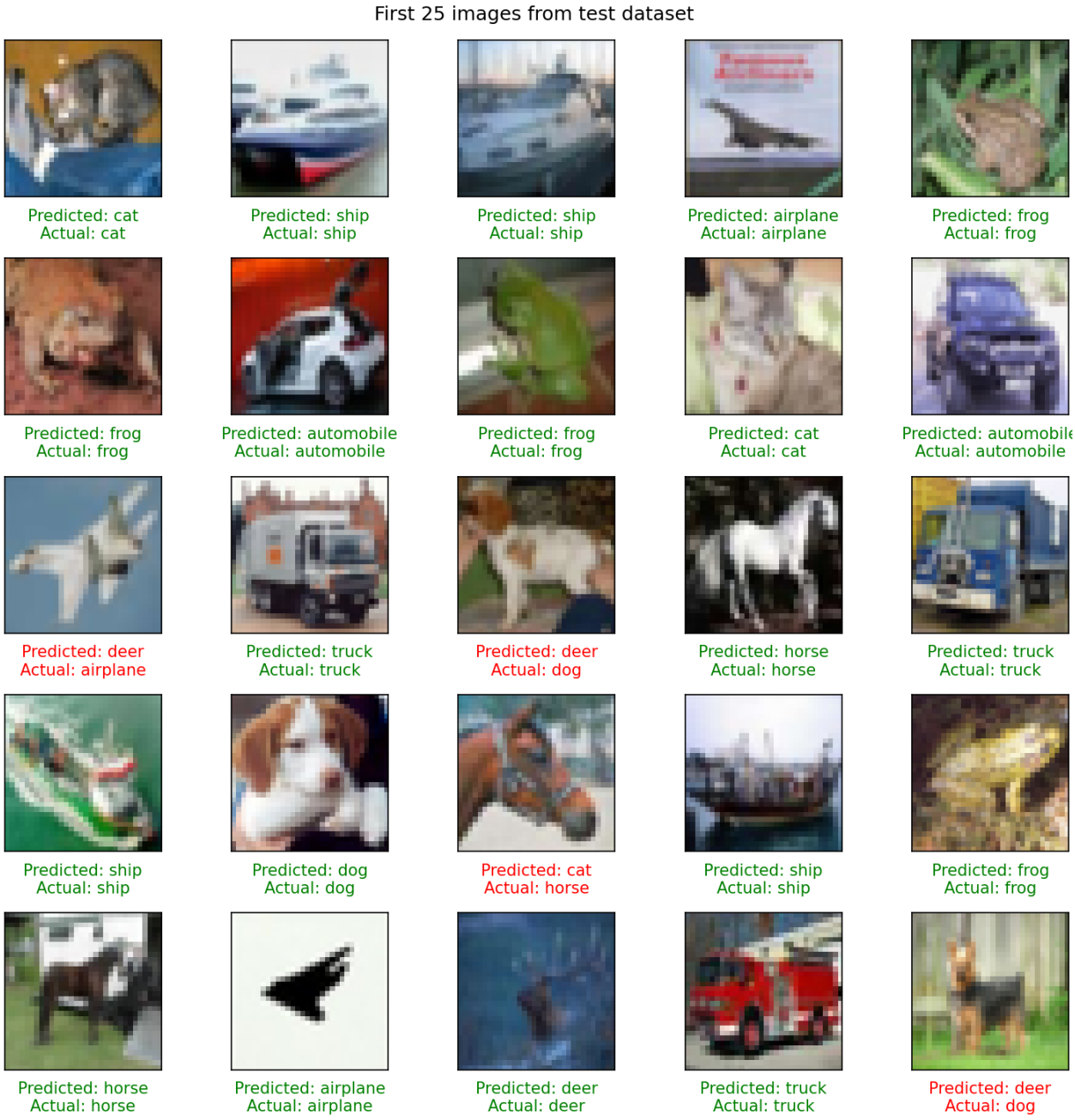

Figure 3: First 25 images in the test set with both actual and predicted labels.

References

Bhandari, A. (2020, April 3). *Feature Scaling | Standardization Vs Normalization*. Analytics Vidhya. Retrieved February 5, 2023, from https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

Brownlee, J. (2018, December 3). *A Gentle Introduction to Dropout for Regularizing Deep Neural Networks - MachineLearningMastery.com*. Machine Learning Mastery. Retrieved February 5, 2023, from https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/

Chaudhary, M. (2020, August 27). *Activation Functions: Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax | by Mukesh Chaudhary*. Medium. Retrieved February 5, 2023, from https://medium.com/@cmukesh8688/activation-functions-sigmoid-tanh-relu-leaky-relu-softmax-50d3778dcea5

Krizhevsky, A. (2009, December 31). *The CIFAR-10 dataset*. CIFAR-10 and CIFAR-100 datasets. Retrieved February 5, 2023, from https://www.cs.toronto.edu/~kriz/cifar.html

Moreno, F. A. (2021, November 12). *Sparse Categorical Cross-Entropy vs Categorical Cross-Entropy | by Felipe A. Moreno | Medium*. Felipe A. Moreno. Retrieved February 5, 2023, from https://fmorenovr.medium.com/sparse-categorical-cross-entropy-vs-categorical-cross-entropy-ea01d0392d28

Rosebrock, A. (2018, December 31). *Keras Conv2D and Convolutional Layers*. PyImageSearch. Retrieved February 5, 2023, from https://pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/

Saxena, S. (2021, April 5). *Softmax | What is Softmax Activation Function | Introduction to Softmax*. Analytics Vidhya. Retrieved February 5, 2023, from

https://www.analyticsvidhya.com/blog/2021/04/introduction-to-softmax-for-neural-network/