Image Classification

CS 6301: Practical Aspects of Data Science

Summer 2020: Assignment 2

Team Member: -
Rajarshi Chattopadhyay (RXC170010)

Introduction

This project involves using deep learning for image classification, using R and Python.

A non-TensorFlow technique for image classification has been used in R and Keras API of Tensorflow has been used in Python.

Data

CIFAR-10 - Object Recognition in Images http://www.cs.utoronto.ca/~kriz/cifar.html
The data is also stored in https://www.utdallas.edu/~rxc170010/cifar-10-batches-py/

The CIFAR-10 is labeled subset of the 80 million tiny images dataset.
The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class.

Models

In ***R***, grid search has been used to train models with different hyper-parameters and changing hidden layers. RMSE values for the models based on the validation set has been obtained.

Library used: H2O https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/deep-learning.html?highlight=deeplearning

Model training was limited to the below due to infrastructure limitations:

- Rectified Linear Units as the activation function
- Neuron architecture of 3072-128-64-10
- 20 epochs

Predictions was made using the above model along with obtaining the Confusion matrix.

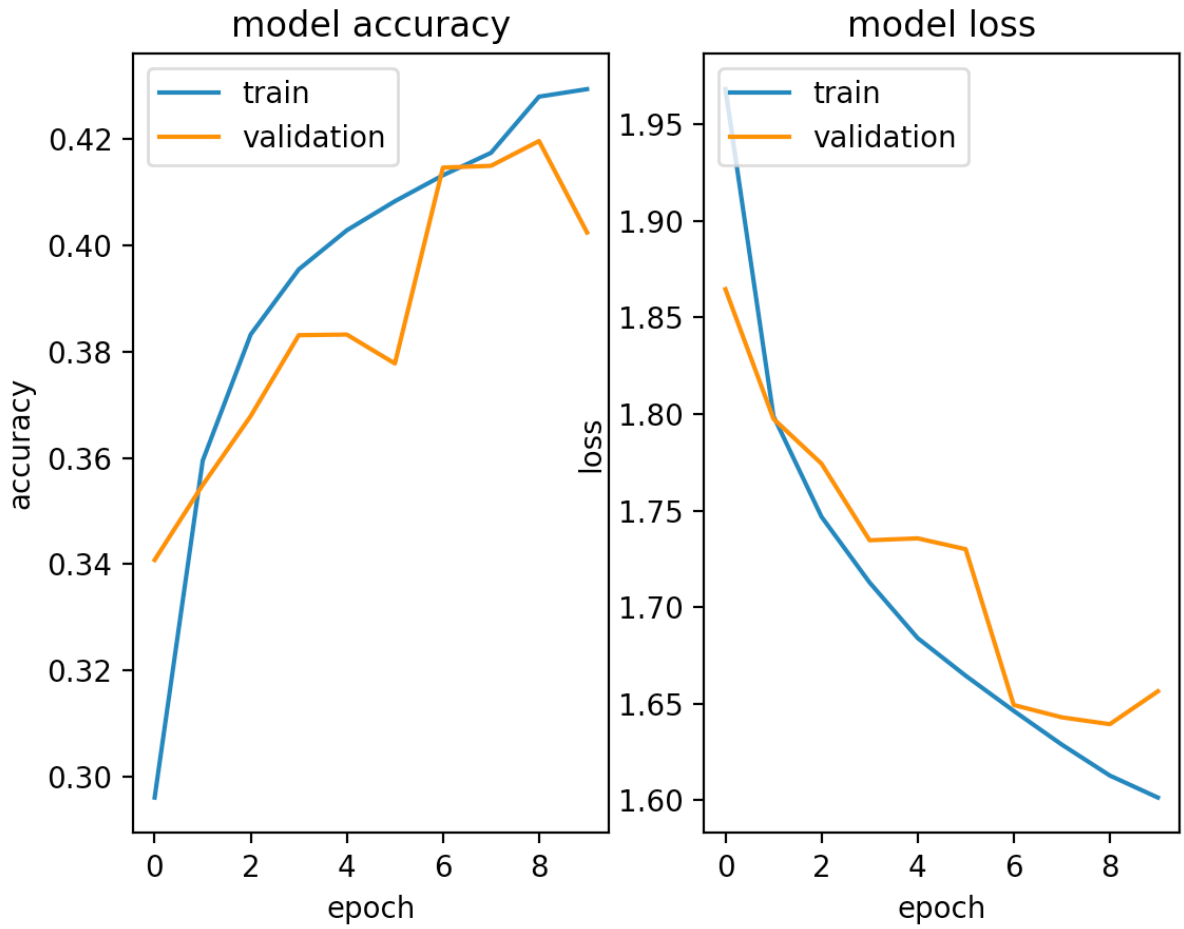
In Python, models with different hidden layers and hyper-parameters were trained. The training and validation accuracy/loss has been plotted related to the epochs for the different models.
The best model has been used to make predictions.

Library used: Keras https://www.tensorflow.org/guide/keras/sequential_model

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 128)	393344
dense_2 (Dense)	(None, 10)	1290

Total params: 394,634
Trainable params: 394,634
Non-trainable params: 0

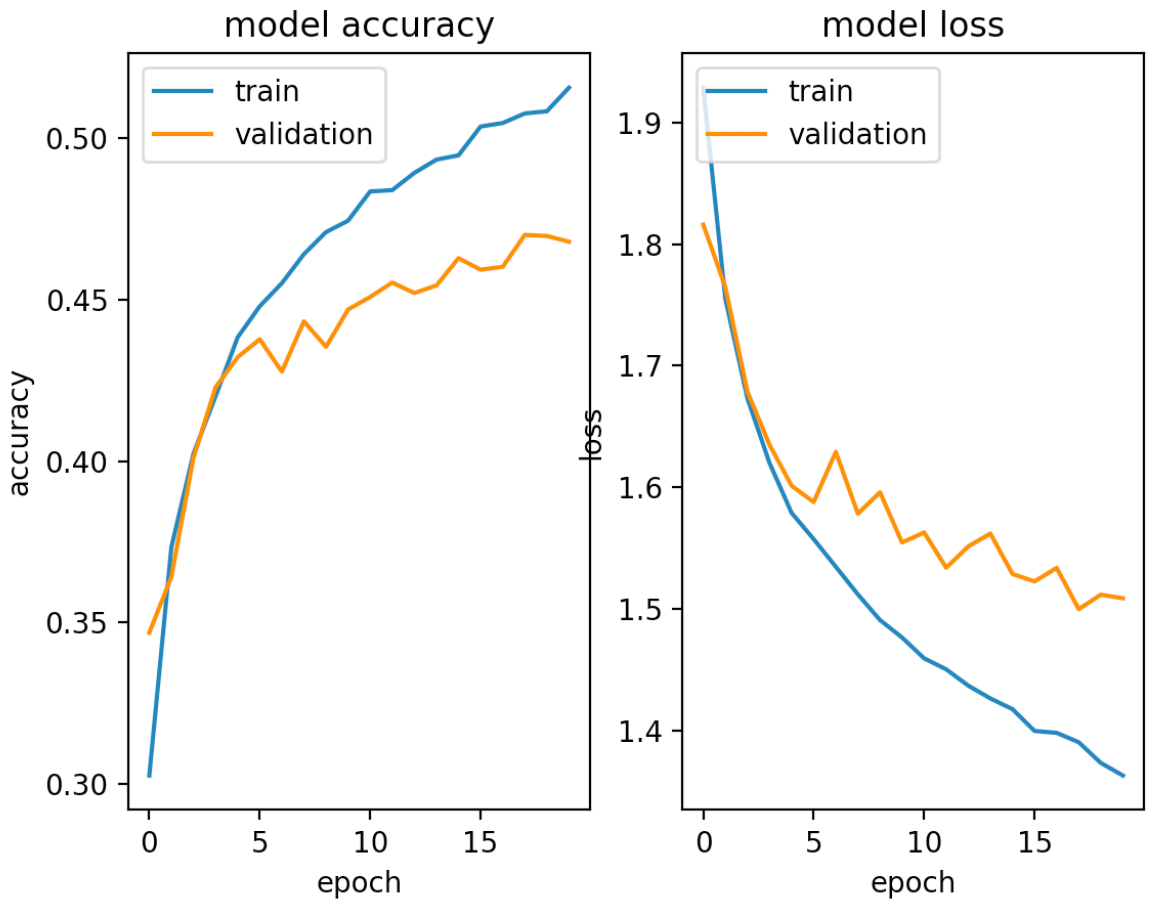


Train result: train_loss: 1.6015 - train_accuracy: 42.9444
Validation result: val_loss: 1.6565 - val_accuracy: 40.2444
Test result: test_loss: 1.6585 - test_accuracy: 39.8600

Model: "sequential_2"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 128)	393344
dense_2 (Dense)	(None, 64)	8256
dense_3 (Dense)	(None, 10)	650

Total params: 402,250
Trainable params: 402,250
Non-trainable params: 0

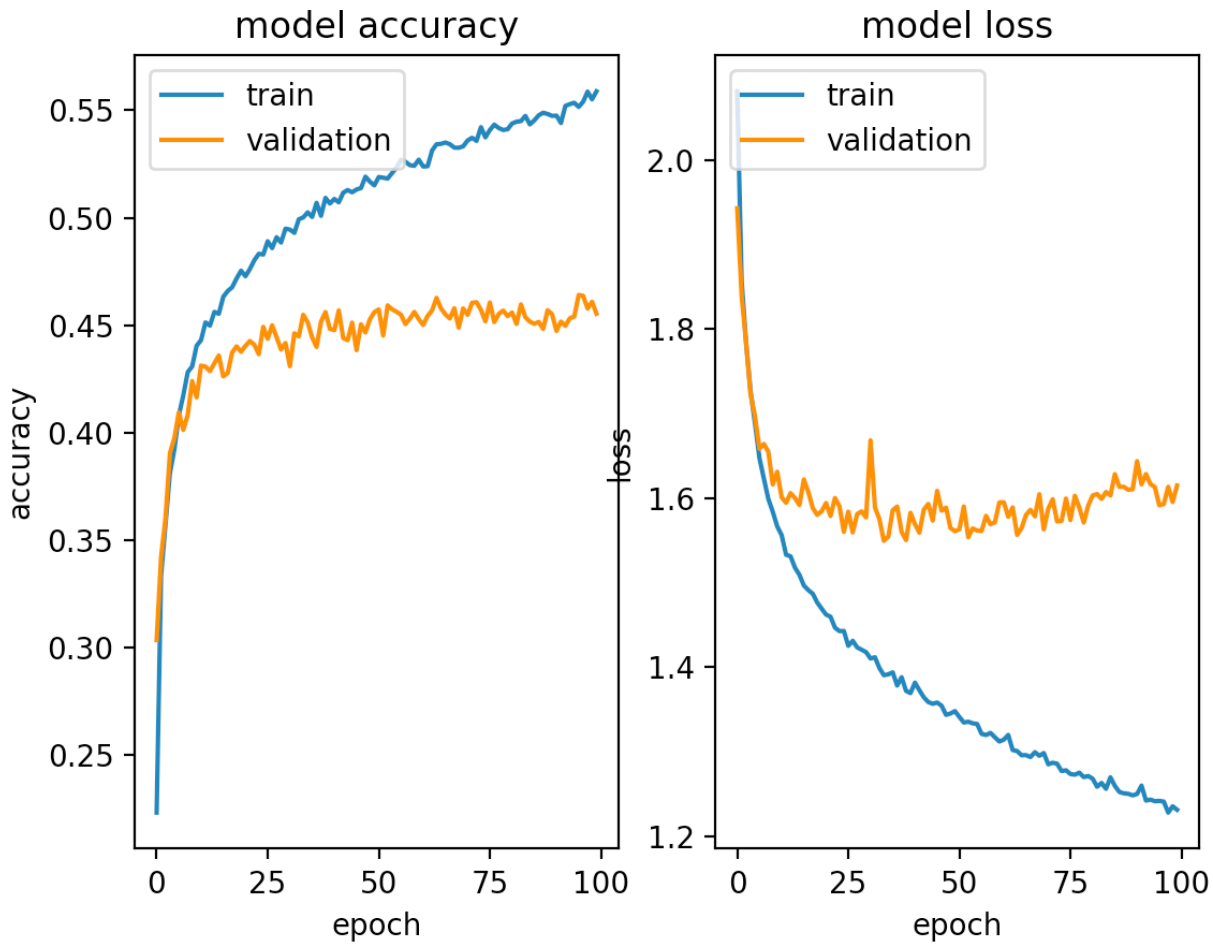


Train result: train_loss: 1.3627 - train_accuracy: 51.5694
Validation result: val_loss: 1.5086 - val_accuracy: 46.8000
Test result: test_loss: 1.4672 - test_accuracy: 47.5800

Model: "sequential_3"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 2048)	6293504
dense_2 (Dense)	(None, 32)	65568
dense_3 (Dense)	(None, 10)	330

Total params: 6,359,402
Trainable params: 6,359,402
Non-trainable params: 0

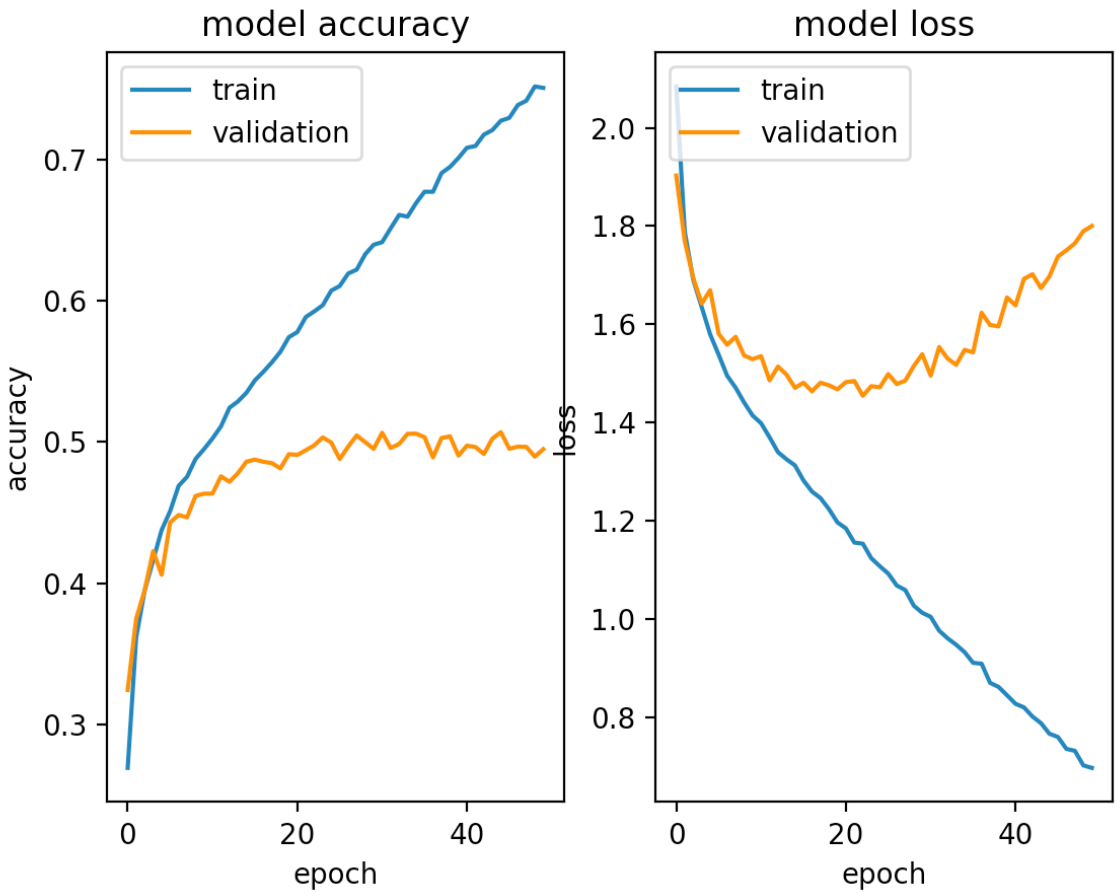


Train result: train_loss: 1.2311 - train_accuracy: 55.8791
Validation result: val_loss: 1.6146 - val_accuracy: 45.5390
Test result: test_loss: 1.5817 - test_accuracy: 46.6600

Model: "sequential_4"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 1024)	3146752
dense_2 (Dense)	(None, 256)	262400
dense_3 (Dense)	(None, 10)	2570

Total params: 3,411,722
Trainable params: 3,411,722
Non-trainable params: 0

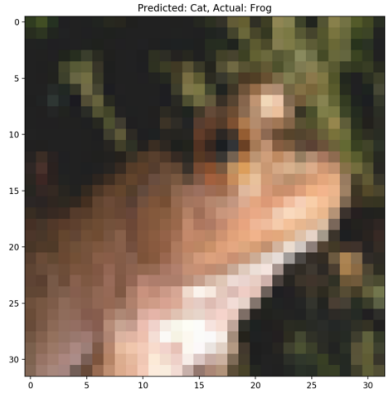


Train result: train_loss: 0.6965 - train_accuracy: 75.1136
Validation result: val_loss: 1.8002 - val_accuracy: 49.4916
Test result: test_loss: 1.7201 - test_accuracy: 50.0000

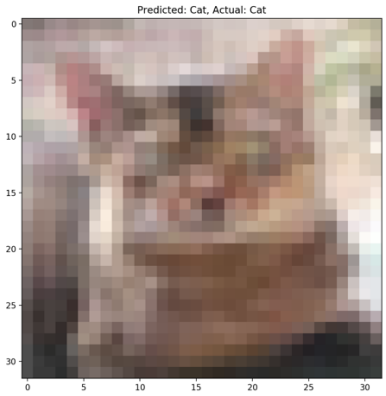
Iteration	Parameters	Accuracy
1	Number of layers = 1 Kernel Size Layer 1 = 128 Activation Function = relu Epochs = 10	Train = 42.94% Validation = 40.24% Test = 39.86%
2	Number of layers = 2 Kernel Size Layer 1 = 128 Kernel Size Layer 2 = 64 Activation Function = relu Epochs = 20	Train = 51.56% Validation = 46.80% Test = 47.58%
3	Number of layers = 2 Kernel Size Layer 1 = 2048 Kernel Size Layer 2 = 32 Activation Function = relu Epochs = 100	Train = 62.88% Validation = 55.53% Test = 56.66%
4	Number of layers = 2 Kernel Size Layer 1 = 1024 Kernel Size Layer 2 = 256 Activation Function = relu Epochs = 50	Train = 75.11% Validation = 59.49% Test = 62.00%

Prediction using best model

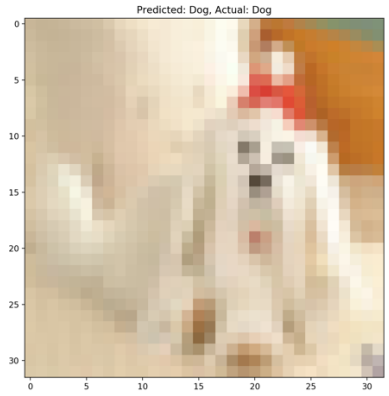
4039: Predicted: Cat, Actual: Frog



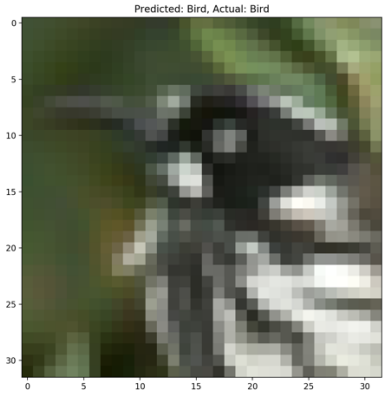
4900: Predicted: Cat, Actual: Cat



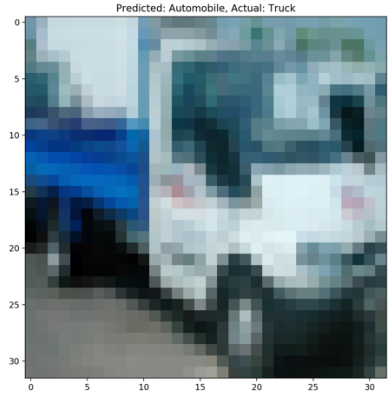
1842: Predicted: Dog, Actual: Dog



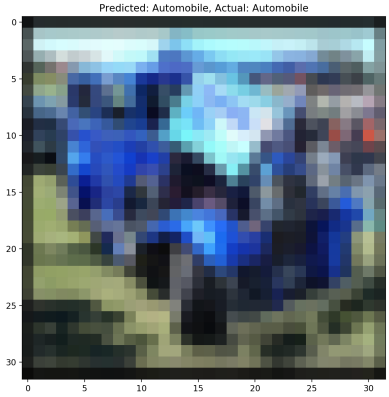
1832: Predicted: Bird, Actual: Bird



734: Predicted: Automobile, Actual: Truck



1107: Predicted: Automobile, Actual: Automobile



Conclusion

Predictions are made using the best model. The training accuracy of the best model is ~75%. Further training of more deep models could not be done due to infrastructure limitations.

References

- Image plotting: https://matplotlib.org/api/_as_gen/matplotlib.pyplot.imshow.html
- Keras: https://www.tensorflow.org/guide/keras/sequential_model
- H2O: https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/deep-learning.html