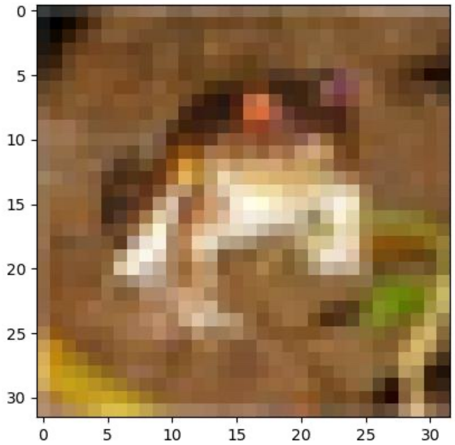
466 Project

Introduction

In this project, I am going to use 3 different algorithms to train 1 dataset, and analyze the performance between them.

Dataset

In this project I am going to use the built-in dataset in tensorflow. The "CIFAR-10" dataset. This dataset contains 50000 colored pictures for training, with each picture as airplanes, cats, dogs...etc. Each picture is labeled from 0 to 9 based on what they were. One sample from the dataset looks like this, just to show some visualizations:



(The first picture in the dataset)
(This is a frog(labeled 6))

Then, there are 10000 colored pictures for testing. In conclusion, the dataset is of this format:
x train shape: (50000, 32, 32, 3)
y train shape: (50000, 1)
x test shape: (10000, 32, 32, 3)
y test shape: (10000, 1)
(where 50000, 32, 32, 3 means it is of 50000 numbers of 32*32 pixels, 3 channel RGB pictures.)
The output will be one number indicating which class this data belongs to.
In order to validate the performance, I split the training dataset into 2 parts. 80% are for training, and 20% are for validating, The one with higher validation accuracy will imply better performance of the model.

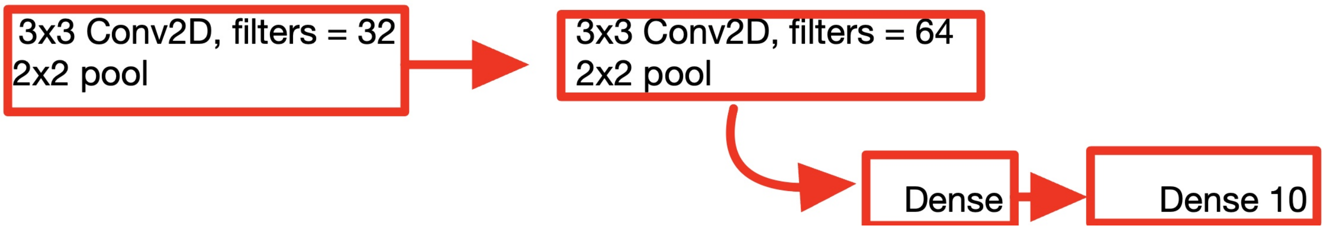
Baseline

I add a baseline just to compare the performance of my models to it, the baseline is implemented by assigning a picture(input) to a random class (0-9), and we get the test accuracy = 0.0959

1. CNN

The first model I will use is the Convolutional Neural Network, I will use it for feature extraction. After that I will feed the features into a Fully Connected Neural Network, I will then tune the hyper-parameters of the network with its 1.number of hidden layers[1,2] 2.number of neurons[64,128] 3. dropout rate[0.3,0.5](the percentage of neurons deactivated in each layer to prevent overfitting)
The evaluation metric used in this method is accuracy. Accuracy is a measure of success for classification tasks and represents the proportion of correctly classified instances out of the total number of instances.

Accuracy is a reasonable approximation of success for general classification tasks where the goal is to correctly classify instances into different classes. It is reasonable because accuracy is calculated as the ratio of correctly predicted instances to the total number of instances, providing a measure of the model's overall correctness.



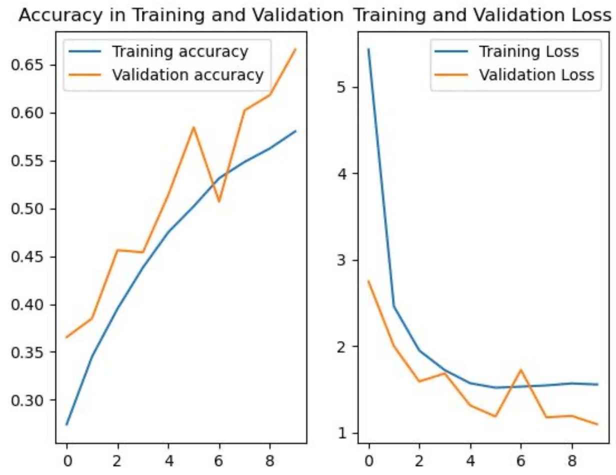
3x3 Conv2D, filters = 32
2x2 pool

3x3 Conv2D, filters = 64
2x2 pool

Dense

Dense 10

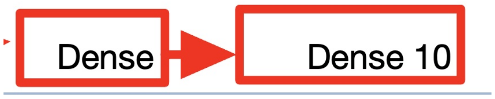
The hyperparameters are tuned in an exhaustive search way, where I try every combination of them and find the ones that give the best results.
After training, I found that the optimized hyperparameters are:
1 hidden layer
64 of dense units for each hidden layer
0.5 dropout rate
The test performance(accuracy) on this optimized model is:0.6650000214576721, this is very high as compared to the baseline.
The plot of the training/validation accuracy/loss of the best hyperparameters is:



Accuracy in Training and Validation Training and Validation Loss

2. Full-Connected Neural Network
The second model I will use is the Fully Connected Network, or Fully Connected NN. I will feed the flattened raw data into the network, and then tune the same hyperparamters as of CNN; The number of hidden layers[1,2], the number of neurons[64,128], and the dropout rate[0.3,0.5]
The evaluation metric used in this method is accuracy. Accuracy is a measure of success for classification tasks and represents the proportion of correctly classified instances out of the total number of instances.
Accuracy is a reasonable approximation of success for general classification tasks where the goal is to correctly classify instances into different classes. It is reasonable because accuracy is calculated

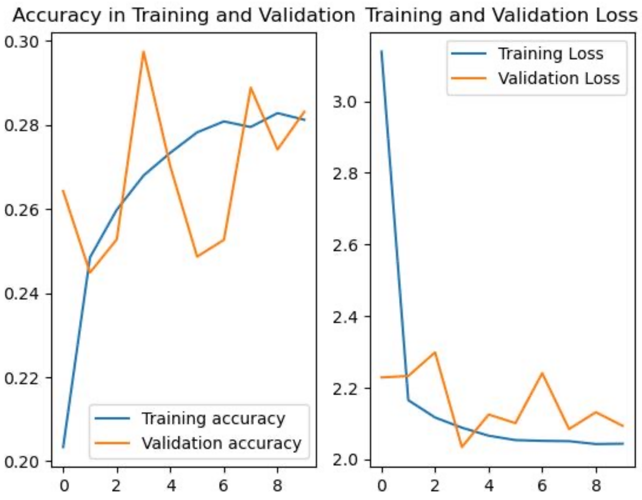
as the ratio of correctly predicted instances to the total number of instances, providing a measure of the model's overall correctness.



Dense

Dense 10

The hyperparameters are tuned in an exhaustive search way, where I try every combination of them and find the ones that give the best results.
After training, I found that the optimized hyperparameters are:
1 layer
128 of dense units for each hidden layer
0.3 dropout rate
The test performance(accuracy) on this optimized model is:0.2833000123500824, this is still higher than the baseline, although not as good as CNN
The plot of the training/validation accuracy/loss of the best hyperparameters is:



3. Random Forest Classifier
The third method I will use is the RF Classifier, I will flatten the data and design a random forest classifier. The hyperparameters I use are 1. number of trees in the forest:[50,100] 2. the maximum depth of each tree, to prevent overfitting[10,20]
The evaluation metric used in this method is accuracy. Accuracy is a measure of success for classification tasks and represents the proportion of correctly classified instances out of the total number of instances.
Accuracy is a reasonable approximation of success for general classification tasks where the goal is to correctly classify instances into different classes. It is reasonable because accuracy is calculated as the ratio of correctly predicted instances to the total number of instances, providing a measure of the model's overall correctness.

After training, I found that the optimized hyperparameters are:
100 trees
20 depth

and the test accuracy(performance) and the optimized hyperparameters is: 0.4601, this is a pretty good model as compared to the baseline, even though it is lower than CNN, it is still considered a high accuracy

I want to get the test accuracy as high as possible, this is the measure of the model performance on unseen data, the higher it is, the better the model is.

Conclusion

In conclusion, CNN > RF > FNN in model performance, model performances are measured by test accuracy, as stated in the previous pages.