Report

Introduction

Object recognition can be described as spotting the objects from a digital image (Brownlee, 2019). Image classification is a process of forecasting the category of one object in a photograph (Brownlee, 2019). Object is located in a provided image and framed to the extent of same (Brownlee, 2019). Image classification can be challenging in various aspects especially when data is gigantic in nature. To accomplish these tasks several machine learning algorithms can aid. In our experiment we will be demonstrating the outcome of two machine learning algorithms. In this experiment we worked on CIFAR-100 data set. CIFAR-100 data set consists 100 categories or we may also refer it as classes of animals, fish, objects and many more. Each category includes 600 images (Krizhevsky, 2017). These 600 images are divided into training and testing images (Krizhevsky, 2017).500 images are allotted as training images and 100 images are utilized as testing images (Krizhevsky, 2017). Further the 100 classes are assembled into 20 superclass group (Krizhevsky, 2017). The label for every image is categorized into 2 types fine and coarse (Krizhevsky, 2017). Fine label belongs to the class which are 100 and coarse label belongs to designated 20 superclass (Krizhevsky, 2017). The size of each image is 32x32x3 representing height, width and color channel. The data is stockpiled in the form of 4D tensor with four dimensions as height, width, color channel and image samples. The image data is segregated into training and testing splits. The training data is utilized to tarin the model with respective features involved in it. The testing data is used to gain the estimation and performance of the model in classifying the objects in the images. The respective labels are also fed as input to the model for training as we have imposed supervised learning. Our task is to classify the images to appropriate classes. For the classification of the images, we implemented two machine learning algorithms neural network and Convolutional neural network. We calculated the accuracy and loss of respective algorithm for the provided data set. We also computed the respective confusion matrices of both the algorithms. We will be going through the detailed overview of the same in next sections.

Method

Preliminary data Processing

We started by importing all the essential packages required in our code. Followed by that we loaded the training and testing image data into respective variables. We loaded the respective training and testing labels. The labels were loaded separately as training fine, testing fine, training coarse and testing coarse respectively.

We learnt about the shapes of the training and testing data and labels. Details of the shapes are mentioned as below

```
Training Images Shape: (32, 32, 3, 50000)
Testing Images Shape: (32, 32, 3, 10000)
Images Fine train Labels Shape: (50000,)
Images Coarse train Labels Shape: (50000,)
Images Fine test Labels Shape: (10000,)
```

```
Images Coarse test Labels Shape: (10000,)
```

Further we rearranged the training and testing image data to get below mentioned shapes

```
Shape of transposed: (50000, 32, 32, 3)
Shape of transposed: (10000, 32, 32, 3)
```

Neural network (NN)

In this machine learning algorithm, we developed a fully connected neural network. We created a sequential model from tensor flow open-source platform. Neural network has various interconnected layers and output of one-layer acts as input to following layer. In this sequential model we created two hidden layers and one output layer. In our experiment we implemented dense layers. The sequential model enables us to provide the number of neurons we desire to have in our neural network. We have developed our neural network with 4000 neurons and 2000 neurons by feeding it as input parameter to our first and second dense layer respectively. Also, we loaded relu (rectified linear activation function) activation function along with neurons in the dense layers. In the output layer we inserted 100 as input parameter as we have 100 unique classes. Further we designed compile strategy for our model. For optimization approach for data training we utilized Stochastic Gradient Descent for fine labels and Adam() for coarse label. We used Sparse Categorical Cross-entropy for loss as the labels are integer values. We used Sparse Catergorical Accuracy as our accuracy metric to estimate appropriate forecasting of input data to corresponding labels. Following this we loaded our transposed train data, train labels, epoch and validation split into fit () function. Epoch can be described as one total transaction of training data through the designed model. In our experiment we have used epoch of value 20. Validation spit helps in splitting the training data into training and validation data. Validation data and testing data are entirely different. Further training and accuracy curves were plotted. Later we evaluated the model on test data and labels and corresponding accuracy was captured.(Entire paragraph citation (Google Colaboratory, 2022))

Convolutional neural network (CNN)

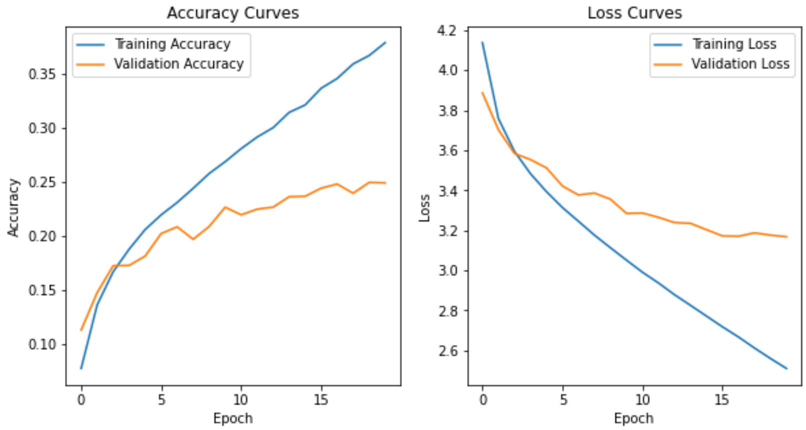
The architecture of CNN is similar to NN. We have added two layers of convolutional and max pooling. In convolutional layer we loaded 32 filters in first layer and 128 filters in second layer which can detect 32 and 128 different features respectively in the particular image. Size of filters is 3 by 3. We have added one dense layer with 400 neurons.

Result

The evaluation metrics we have considered in our experiment is accuracy and loss. Mentioned below are the loss and accuracy stats and respective Accuracy and loss curves for training and validation respectively.

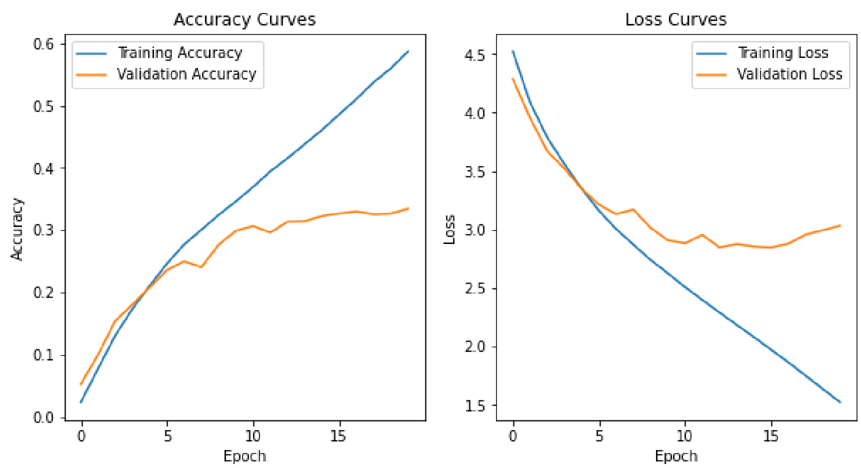
Neural Network (NN)(Fine labels)

```
313/313 [==============================] - 7s 22ms/step - loss: 3.1601 - sparse_categorical_accuracy: 0.2564
Test set loss: 3.16, test set accuracy: 25.64%
```



Convolutional Neural Network(CNN)(Fine Labels)

```
313/313 [==============================] - 4s 11ms/step - loss: 2.9542 - sparse_categorical_accuracy: 0.3431
Test set loss: 2.95, test set accuracy: 34.31%
```



Conclusion

We demonstrated the behavior of two machine learning algorithms on CIFAR-100 data set. We started by understanding the shapes of the data segments and preprocessing the data. We designed training model and further constructed compile model. We trained the created model by passing training data, training labels, epoch and validation split. We visualized loss and accuracy curves for training and validation data which gave us insight of development of training model. We evaluated the model by passing test data images and label. We computed loss and accuracy for the same. From the results we were able to discover that CNN performed better than NN on the provided data set. Drawback of the methods would fall in accuracy. Accuracy needs to be enhanced through investigating by implementing other algorithms and having trail error scenario.

References

Krizhevsky, A., 2017. *CIFAR-10 and CIFAR-100 datasets*. [online] Cs.toronto.edu. Available at: <https://www.cs.toronto.edu/~kriz/cifar.html> [Accessed 20 April 2022].

Brownlee, J., 2019. *A Gentle Introduction to Object Recognition With Deep Learning*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/object-recognition-with-deep-learning/> [Accessed 20 April 2022].

Colab.research.google.com. 2022. *Google Colaboratory*. [online] Available at: <https://colab.research.google.com/drive/1yAEMt0P0TreMrDaR3N7rOFQG6YsQd1ue#scrollTo=c0Zy7WKq04NS> [Accessed 21 April 2022].(Swansea university Machine leaning lab)