1 CIFAR10

1.1 Preprocessing

Firstly, downloading and loading *CIFAR10* dataset from *Keras* library and using the *train_test_split* function to randomly sample 20 percent of the training set as the new training set in order to save computation.

Secondly, divide the RGB images by 255 to normalize the input vectors.

Thirdly, using a 0.5 ratio to split the testing set into validation and testing.

1.2 Output Layer and Loss Function

The activation function used in the output layer is **softmax** which inputs an N-dimensional vector of real numbers and outputs a probability distribution. And totally 10 neurons are used in the output layer, one represents each category in this dataset. The loss function used in this question is **categorical_crossentropy** as there are 10 classes in this dataset and the labels are in *one_hot* representation.

1.3 MLP

Fig.1 shows the training and validation accuracy for the MLP model with two fully connected layers with 512 units and a sigmoid activation function and an output layer.

The training accuracy after 5 epochs is 35.77% and the validation accuracy is 35.16%. Both of them show an upward trend observed from fig.1 meaning that the result is under-fitting as the training epoch and number of training samples are small. And the testing accuracy provided by the MLP model is 36.52%.

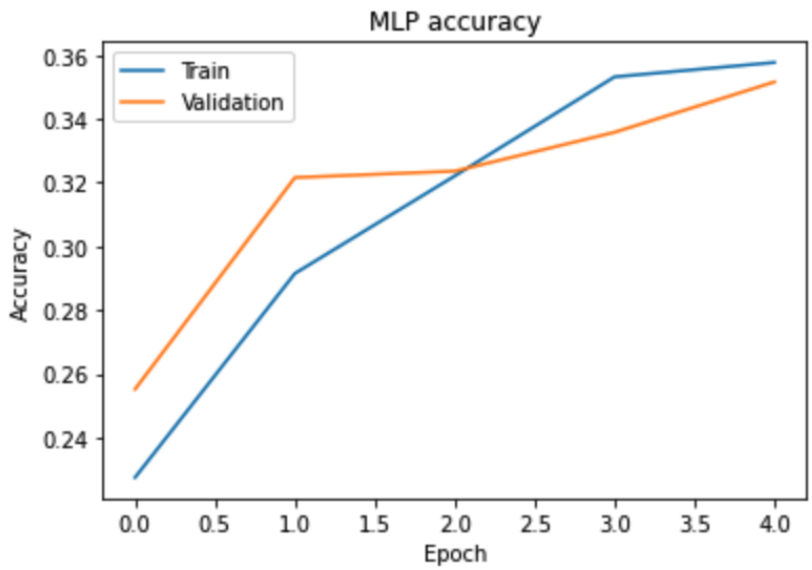

Figure 1: Training and validation accuracy for MLP model

Fig.2 and fig.3 show the training and validation accuracy for the MLP models after changing the number of layers and neurons per layer, where the MLP-v1 model has three fully connected layers with 512, 512, 256 units respectively and MLP-v2 model has only one fully connected layer with only 10 neurons.
The training and validation accuracy for the three fully connected layer MLP model after 5 epochs are 33.26% and 34.44%, and the testing accuracy for this model is 34.06%. The training and validation accuracy for the one fully connected layer MLP model after 5 epochs are 31.60% and 29.98%, and the testing accuracy for this model is 31.26%.

Table 1 summarizes the accuracy of the three models mentioned above. The accuracies for all the three MLP models are very low due to the small number of training epochs and the small number of training samples. And sometimes the validation accuracy and testing accuracy are higher than training accuracy, this is may because the number of

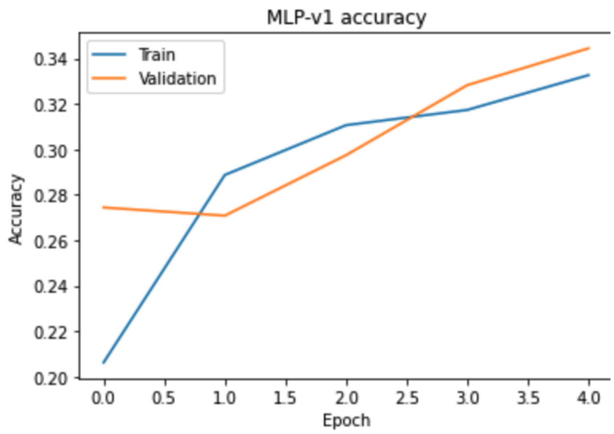

Figure 2: Training and validation accuracy for version1 MLP model

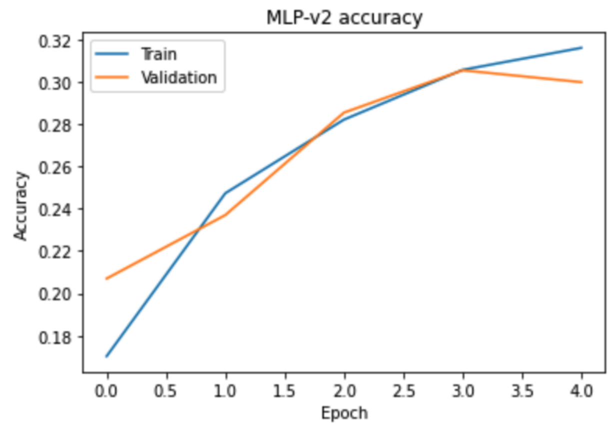

Figure 3: Training and validation accuracy for version2 MLP model

training samples (10000) is insufficient compared with the validation set (5000) and testing set (5000), and its samples' distribution affected the results.

The reason that the three-fully-connected-layer MLP-v1 model's accuracy is lower than the original two-fully-connected-layer model is that the architecture of the network is more complicated and with the same number of training epochs, its result is more under-fitting than the simple model. And the reason that the one-fully-connected-layer MLP-v2 model's accuracy is lower is that the small number of neurons (only 10) cannot perform the classification task very well.

Netword	Training Accuracy	Validation Accuracy	Testing Accuracy
MLP	35.77%	35.16%	36.53%
MLP-v1	33.26%	34.44%	34.06%
MLP-v2	31.06%	29.98%	31.26%

Table 1: Accuracy for three MLP models

1.4 Train and Test Accuracy for MLP and CNNs

Table 2 shows the train and test accuracy for all the three networks. It can be observed that both of the two CNN models perform better than the MLP model based on the testing accuracy. This result is as expected, as now Convolutional Neural Network is mainly used in computer vision.

Netword	Training Accuracy	Validation Accuracy	Testing Accuracy
MLP	35.77%	35.16%	36.53%
CNN1	94.25%	54.38%	55.40%
CNN2	61.78%	57.24%	57.68%

Table 2: Accuracy for MLP and CNNs models

In general, the number of weight parameters in the MLP model would be very large as the hidden layer is fully-connected which would cause inefficiency and requires higher computational and memory resources. Besides, the MLP model inputs flatten vectors ignoring the spatial information in images. Both of these characteristics leading to lower accuracy.

The panning of the filters in the Convolutional Neural Network allows it to share weight parameters and find specific patterns from every corner of the input image. Usually, different layers have different tasks to find different patterns from simple to complicate. The use of filters works well for object detection since it allows the network to find patterns in more than one place in the input image.

The training accuracy (94.25%) for the first CNN model is much higher than the second one. But its validation accuracy and testing accuracy are only 54.38 % and 55.4%. The high variance means this model is overfitting as it doesn't contain any pooling layer and dropout layer to avoid overfitting.

The second CNN model holds the best performance among the three models, which provides a 61.78% training accuracy and 57.68% testing accuracy.

1.5 Training and Validation curves for CNNs

Fig.4 and fig.5 show the training and validation curves for the two CNN models.

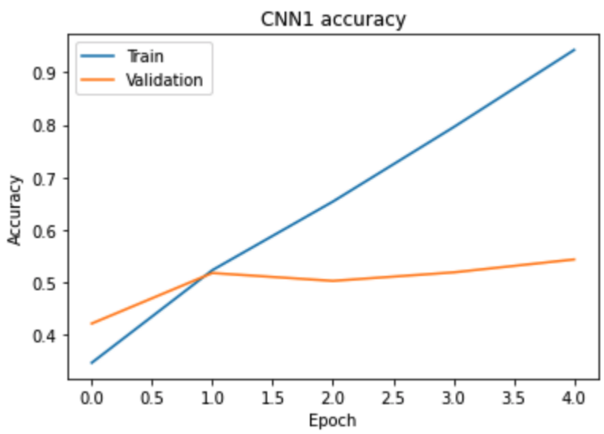

Figure 4: Training and validation accuracy for CNN1 model

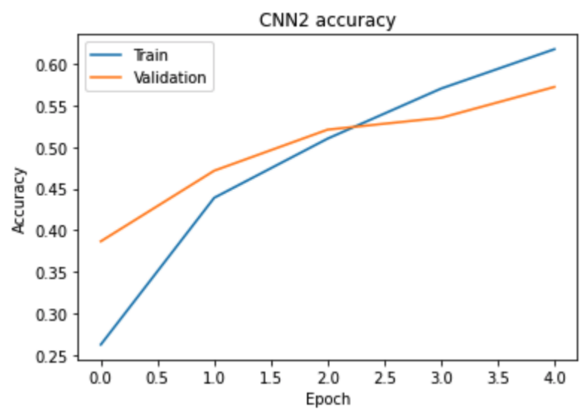

Figure 5: Training and validation accuracy for CNN2 model

It takes around 300 seconds to finish one epoch in the first CNN model, whereas it only takes about 50 seconds to complete one epoch in the second CNN model. This is mainly because the added Maxpooling layers in the second CNN model which reduce the dimensions of the data, thus reduce the amount of the weight parameters and lower the computation.

As already mentioned in the subsection 1.4, the first CNN model is overfitting after 5 epochs, if the networks were trained for more epochs, the results of it would not be better. But as the added pooling layers and dropout layers in the second CNN model help it avoid overfitting, the loss is still decreasing after 5 epochs. Thus, training more epochs (within a rational range)would let the model has higher accuracy (both training and testing). But if it was trained too long, it will also become overfitting.

1.6 Improve Recommendations

For the MLP model, training longer and bigger would help it reduce the avoidable bias as it is still underfitting after 5 epochs. But its characteristics decided that its accuracy would not be very high. For higher accuracy, a better neuron network architecture is required (for example, CNN).

For the first CNN model, several methods can be used to avoid overfitting, thus improve the network. For example, using more training data. Or adding pooling layers. Or implementing regularization, which including add L2 regularization to the Conv2D layer or Dense layer, or add dropout layer. Besides, early stopping can also help to avoid overfitting, which means training fewer epochs.

For the second CNN model, training slightly more epochs would help it perform better. Additionally, using more training samples, tuning the hyperparameters, changing optimizer, and loss function, all might help improve the network.