1. Network parameters [1.5 pt]

Below is the architecture of a network. "FC 256" denotes a fully connected layer with 256 units. "FC 10" denotes a fully connected layer with 10 units. In all layers of Module 1 and Module 2, the padding is set to "same" and strides are set to 1×1. The activation functions of all hidden layers are ReLU. The activation function of the last FC layer is Softmax. Compute the number (NO.) of parameters and NO. multiplications in each layer of the network (ignore the bias) given the single input. Show your working in the writing report.
Note: the NO.parameters and multiplications in each layer of the module 1 and 2 should be separately given.

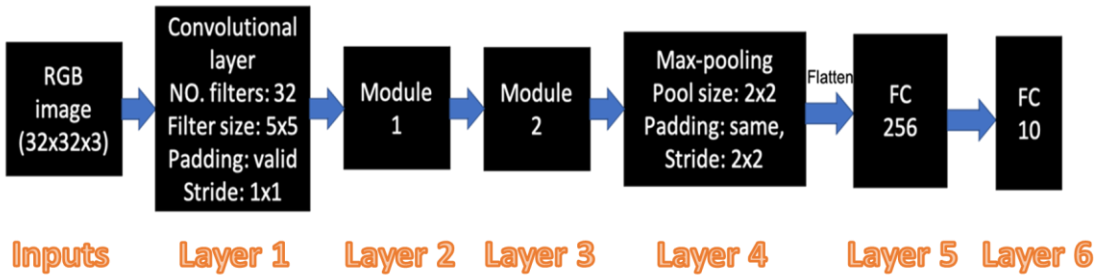


Fig 1.0.1 Network Diagram

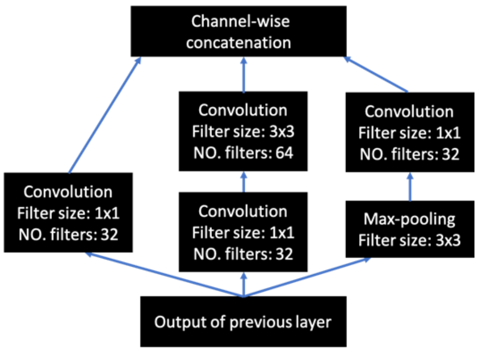


Fig 1.0.2 Module 1 (Layer 2)

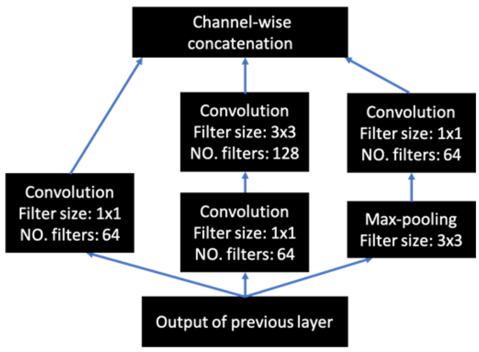


Fig 1.0.3 Module 2 (Layer 3)

Layer 1:

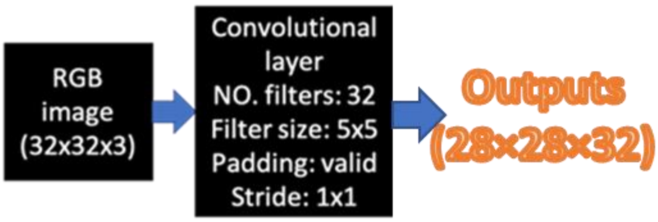


Fig 1.1.1 Inputs & Outputs of Layer 1

As shown in Fig. 1.1, the padding is set to be "valid" in the layer 1. Hence,

$$Output\ Size\ of\ One\ Chanel = Ceiling\left(\frac{Input\ Size - Kernel\ Size + 1}{Stride}\right) = Ceiling\frac{32-5+1}{1} = 28$$

As shown in Fig. 1.1, there are 32 filters (Channels) in layer 1, then:

$$Output \;=\; 28 \times 28 \times 32$$

$Number\ of\ Parameters\ = w_{Kernel} \times h_{Kernel} \times Input\ Chanels\ \times No.Filters =$
$5 \times 5 \times 3 \times 32 = 2{,}400$

$$\begin{aligned} Number\ of\ Multiplications & \\ &= Input\ Channels \times w_{Kernel} \times h_{Kernel} \times w_{output} \times h_{output} \times No.Filters \\ &= 3 \times 5 \times 5 \times 28 \times 28 \times 32 = 1{,}881{,}600 \end{aligned}$$

Layer 2:

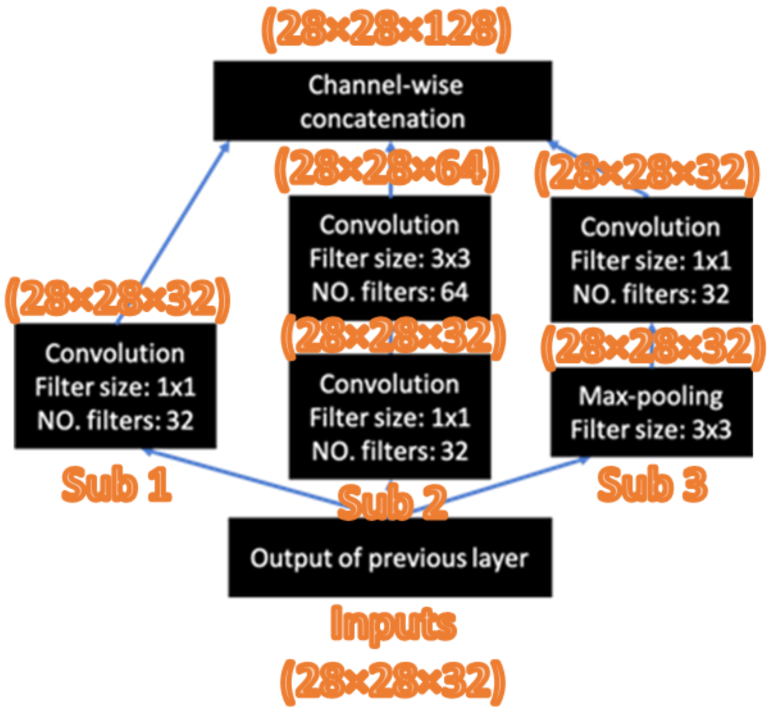


Fig. 1.2.1 Inputs & Outputs of Layer 2

For Sub 1:

The padding is set to “same” and strides are set to 1×1. Hence,

$$Output\ Size\ of\ One\ Chanel = Ceiling\left(\frac{Input\ Size}{Stride}\right) = Ceiling\left(\frac{28}{1}\right) = 28$$

$$Number\ of\ Channels = No.Filters = 32$$

$$Output\ of\ Sub\ 1 = 28 \times 28 \times 32$$

$$Number\ of\ Parameters\ = w_{Kernel} \times h_{Kernel}\ \times No.of\ Input\ Chanels\ \times No.Filters$$
$$= 1 \times 1 \times 32 \times 32 = 1{,}024$$

$$Number\ of\ Multiplications$$
$$= No.of\ Input\ Channels \times w_{Kernel} \times h_{Kernel} \times w_{output} \times h_{output} \times No.Filters = 32 \times 1 \times 1 \times 28 \times 28 \times 32 = 802{,}816$$

For Sub 2:

The **first layer in sub 2** is the same as sub 1. Hence,

$$Output\ of\ Layer\ 1, Sub\ 2\ = 28 \times 28 \times 32$$

$$Number\ of\ Parameters = 1{,}024$$

$$Number\ of\ Multiplications = 802{,}816$$

The **second layer in sub 2** is a convolution layer with 64 3x3 filters, where the padding is set to "same" and strides are set to 1×1. Hence,

$$Output\ Size\ of\ One\ Chanel = Ceiling\left(\frac{Input\ Size}{Stride}\right) = Ceiling\left(\frac{28}{1}\right) = 28$$

$$Number\ of\ Channels = No.Filters = 64$$

$$Output\ of\ Sub\ 2 = 28 \times 28 \times 64$$

$$Number\ of\ Parameters\ = w_{Kernel} \times h_{Kernel}\ \times No.of\ Input\ Chanels\ \times No.Filters$$
$$= 3 \times 3 \times 32 \times 64 = 18{,}432$$

$$Number\ of\ Multiplications$$
$$= No.of\ Input\ Channels \times w_{Kernel} \times h_{Kernel} \times w_{output} \times h_{output} \times No.Filters = 32 \times 3 \times 3 \times 28 \times 28 \times 64 = 14{,}450{,}688$$

For Sub 3:

The **first layer in sub 3** is a max pooling filter, and its padding is set to "same" and strides are set to 1×1. Hence,

$$Output\ Size\ of\ One\ Chanel = Ceiling\left(\frac{Input\ Size}{Stride}\right) = Ceiling\left(\frac{28}{1}\right) = 28$$

$$Number\ of\ Channels = No.of\ Input\ Channels = 32$$

$$Output\ of\ Layer\ 1, Sub\ 3\ = 28 \times 28 \times 32$$

$$Number\ of\ Parameters = 0$$

$$Number\ of\ Multiplications = 0$$

The **second layer in sub 3** is a convolution layer with 32 1x1 filters, and its padding is set to “same” and strides are set to 1×1. Hence,

$$Output\ Size\ of\ One\ Chanel = Ceiling\left(\frac{Input\ Size}{Stride}\right) = Ceiling\left(\frac{28}{1}\right) = 28$$

$$Number\ of\ Channels = No.Filters = 32$$

$$Output\ of\ Sub\ 3 = 28 \times 28 \times 32$$

$$\begin{aligned} Number\ of\ Parameters\ &= w_{Kernel} \times h_{Kernel}\ \times No.of\ Input\ Chanels\ \times No.Filters \\ &= 1 \times 1 \times 32 \times 32 = 1{,}024 \end{aligned}$$

$$\begin{aligned} &Number\ of\ Multiplications \\ &\quad = No.of\ Input\ Channels \times w_{Kernel} \times h_{Kernel} \times w_{output} \times h_{output} \\ &\quad \times No.Filters = 32 \times 1 \times 1 \times 28 \times 28 \times 32 = 802{,}816 \end{aligned}$$

Concatenation:

$$Output\ of\ Sub\ 1 = 28 \times 28 \times 32$$

$$Output\ of\ Sub\ 2 = 28 \times 28 \times 64$$

$$Output\ of\ Sub\ 3 = 28 \times 28 \times 32$$

$$Output\ of\ Concatenation = 28 \times 28 \times (32 + 64 + 32) = 28 \times 28 \times 128$$

$$Number\ of\ Parameters = 0$$

$$Number\ of\ Multiplications = 0$$

Thus, the outputs of each filter can be calculated as shown in Fig.2.1

Layer 3:

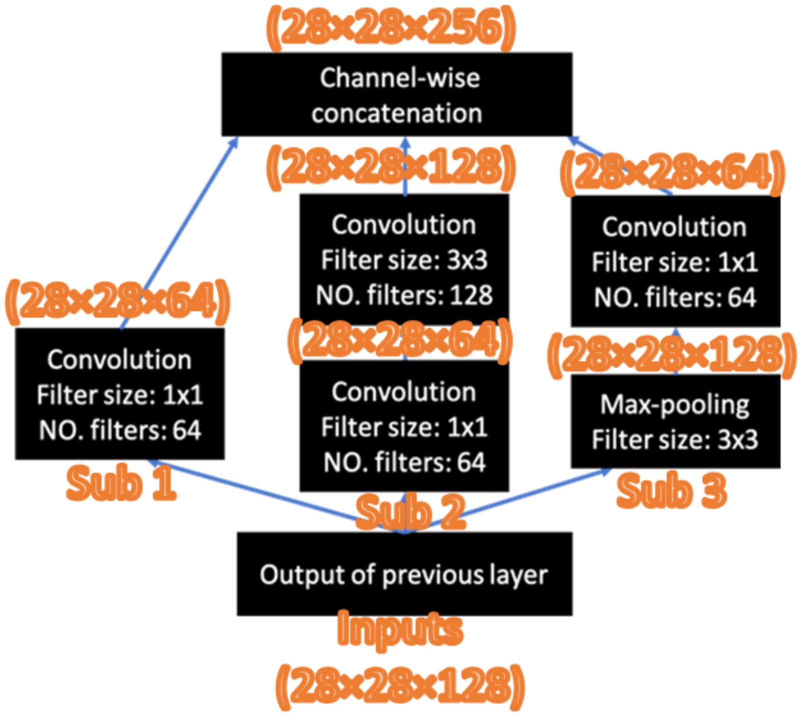


Fig. 1.3.1 Inputs & Outputs of Layer 3

For Sub 1:

The padding is set to “same” and strides are set to 1×1. Hence,

$$Output\ Size\ of\ One\ Chanel = Ceiling\left(\frac{Input\ Size}{Stride}\right) = Ceiling\left(\frac{28}{1}\right) = 28$$

$$Number\ of\ Channels = No.Filters = 64$$

$$Output\ of\ Sub\ 1 = 28 \times 28 \times 64$$

$$\begin{aligned} Number\ of\ Parameters\ &= w_{Kernel} \times h_{Kernel}\ \times No.of\ Input\ Chanels\ \times No.Filters \\ &= 1 \times 1 \times 128 \times 64 = 8{,}192 \end{aligned}$$

$$\begin{aligned} &Number\ of\ Multiplications \\ &\quad = No.of\ Input\ Channels \times w_{Kernel} \times h_{Kernel} \times w_{output} \times h_{output} \\ &\quad \times No.Filters = 128 \times 1 \times 1 \times 28 \times 28 \times 64 = 6{,}422{,}528 \end{aligned}$$

For Sub 2:

The **first layer in sub 2** is the same as sub 1. Hence,

$$Output\ of\ Layer\ 1, Sub\ 2\ = 28 \times 28 \times 64$$

$$Number\ of\ Parameters = 8{,}192$$

$$Number\ of\ Multiplications = 6{,}422{,}528$$

The **second layer in sub 2** is a convolution layer with 128 3x3 filters, where the padding is set to "same" and strides are set to 1×1. Hence,

$$Output\ Size\ of\ One\ Chanel = Ceiling\left(\frac{Input\ Size}{Stride}\right) = Ceiling\left(\frac{28}{1}\right) = 28$$

$$Number\ of\ Channels = No.Filters = 128$$

$$Output\ of\ Sub\ 2 = 28 \times 28 \times 128$$

$$Number\ of\ Parameters\ = w_{Kernel} \times h_{Kernel}\ \times No.of\ Input\ Chanels\ \times No.Filters = 3 \times 3 \times 64 \times 128 = 73{,}728$$

$$Number\ of\ Multiplications = No.of\ Input\ Channels \times w_{Kernel} \times h_{Kernel} \times w_{output} \times h_{output} \times No.Filters = 64 \times 3 \times 3 \times 28 \times 28 \times 128 = 57{,}802{,}752$$

For Sub 3:

The **first layer in sub 3** is a max pooling filter, and its padding is set to "same" and strides are set to 1×1. Hence,

$$Output\ Size\ of\ One\ Chanel = Ceiling\left(\frac{Input\ Size}{Stride}\right) = Ceiling\left(\frac{28}{1}\right) = 28$$

$$Number\ of\ Channels = No.of\ Input\ Channels = 128$$

$$Output\ of\ Layer\ 1, Sub\ 3\ = 28 \times 28 \times 128$$

$$Number\ of\ Parameters = 0$$

$$Number\ of\ Multiplications = 0$$

The **second layer in sub 3** is a convolution layer with 64 1x1 filters, and its padding is set to "same" and strides are set to 1×1. Hence,

$$Output\ Size\ of\ One\ Chanel = Ceiling\left(\frac{Input\ Size}{Stride}\right) = Ceiling\left(\frac{28}{1}\right) = 28$$

$$Number\ of\ Channels = No.Filters = 64$$

$$Output\ of\ Sub\ 3 = 28 \times 28 \times 64$$

$$\begin{aligned} Number\ of\ Parameters\ &= w_{Kernel} \times h_{Kernel}\ \times No.of\ Input\ Chanels\ \times No.Filters \\ &= 1 \times 1 \times 128 \times 64 = 8{,}192 \end{aligned}$$

$$\begin{aligned} &Number\ of\ Multiplications \\ &\quad = No.of\ Input\ Channels \times w_{Kernel} \times h_{Kernel} \times w_{output} \times h_{output} \\ &\quad \times No.Filters = 128 \times 1 \times 1 \times 28 \times 28 \times 64 = 6{,}422{,}528 \end{aligned}$$

Concatenation:

$$Output\ of\ Sub\ 1 = 28 \times 28 \times 64$$

$$Output\ of\ Sub\ 2 = 28 \times 28 \times 128$$

$$Output\ of\ Sub\ 3 = 28 \times 28 \times 64$$

$$Output\ of\ Concatenation = 28 \times 28 \times (64 + 128 + 64) = 28 \times 28 \times 256$$

$$Number\ of\ Parameters = 0$$

$$Number\ of\ Multiplications = 0$$

Thus, the outputs of each filter can be calculated as shown in Fig.3.1

Layer 4:

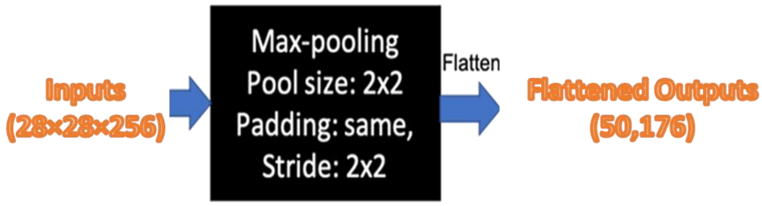


Fig. 1.4.1 Inputs & Outputs of Layer 4

The layer 4 is a max pooling filter, and its padding is set to "same" and strides are set to 2×2. Hence,

$$Output\ Size\ of\ One\ Chanel = Ceiling\left(\frac{Input\ Size}{Stride}\right) = Ceiling\left(\frac{28}{2}\right) = 14$$

$$Number\ of\ Channels = No.of\ Input\ Channels = 256$$

$$Output\ of\ Layer\ 4\ = 14 \times 14 \times 256$$

$$Number\ of\ Parameters = 0$$

$$Number\ of\ Multiplications = 0$$

After **Flattening**,

$$Flattened\ Output\ of\ Layer\ 4 = 14 \times 14 \times 256 = 50{,}176$$

$$Number\ of\ Parameters = 0$$

$$Number\ of\ Multiplications = 0$$

Layer 5:

Layer 5 is a fully connected layer with 256 units. Hence,

$$Output\ of\ Layer\ 5 = Number\ of\ FC\ Units = 256$$

$$Number\ of\ Parameters = Number\ of\ Inputs \times Number\ of\ FC\ Units = 50176 \times 256 = 12{,}845{,}056$$

$$Number\ of\ Multiplications = Number\ of\ Inputs \times Number\ of\ FC\ Units = 50176 \times 256 = 12{,}845{,}056$$

Layer 6:

Layer 6 is also a fully connected layer with 10 units. Similarly,

$$Output\ of\ Layer\ 6 = 10$$

$$Number\ of\ Parameters = 256 \times 10 = 2{,}560$$

$$Number\ of\ Multiplications = 256 \times 10 = 2{,}560$$

2. Classification on the CIFAR-10 subset [4.5 pts]

   1. (1.5 pts) Implement the CNN shown in Q1 in the provided Jupyter notebook and use it to classify the provided dataset. Plot the training accuracy and testing accuracy under different epochs. Use “adam” as the optimization algorithm (set learning rate to 0.001, and leave other hyperparameters of Adam fixed at the default for the library you choose to use). Set batch size to 32 and epoch to 20. In your written report, present the recognition results (show the plot of the training accuracy and testing accuracy under different epochs) and explain why the testing accuracy of this network is poor.

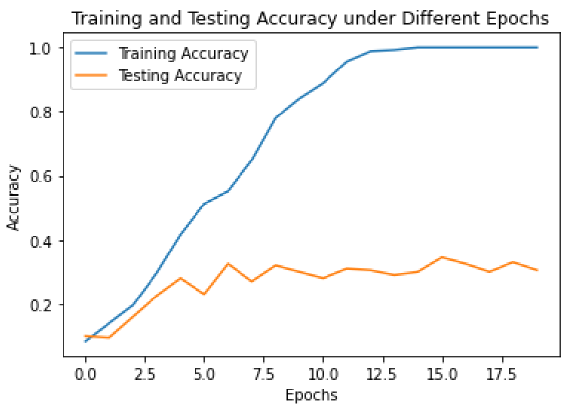


Fig. 2.1.1 Training & Testing Accuracy under Different Epochs

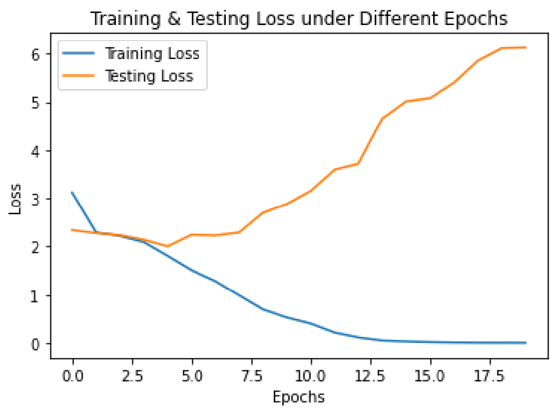
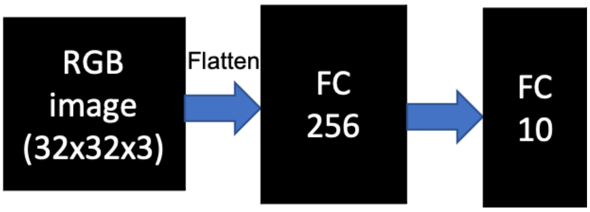


Fig. 2.1.2 Training & Testing Loss under Different Epochs

As shown in Fig. 2.1.1, the testing accuracy stops going up after 7 epochs while the training accuracy approaches 100% after 20 epochs. This indicates the model fits to the training set too much and lost its generalisation. The model is overfitting the training data after 7 epochs.

2. (1 pt) Based on the given dataset (subset of CIFAR-10 with the provided 250/200 train/test split), you consider using the following multi-layer perceptron (MLP) to conduct classification on the raw images with the same training settings, i.e., same optimisation algorithm, batch size and epoch as the above CNN model. Implement the model and use it to classify the given dataset. Plot the training accuracy and testing accuracy under different epochs. In your written report, present the recognition results (show the plot of the training accuracy and testing accuracy under different epochs) and explain why the testing accuracy of the MPL is poor.



MLP

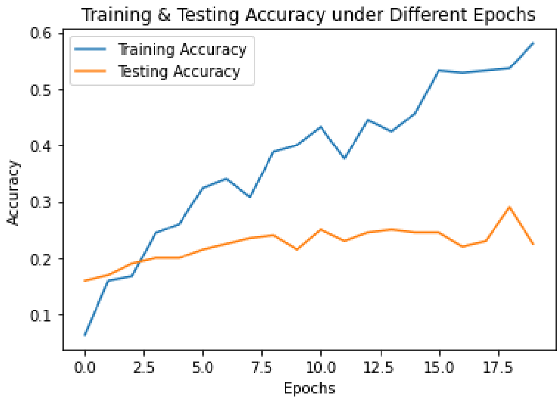


Fig. 2.2.1 Training & Testing Accuracy under Different Epochs

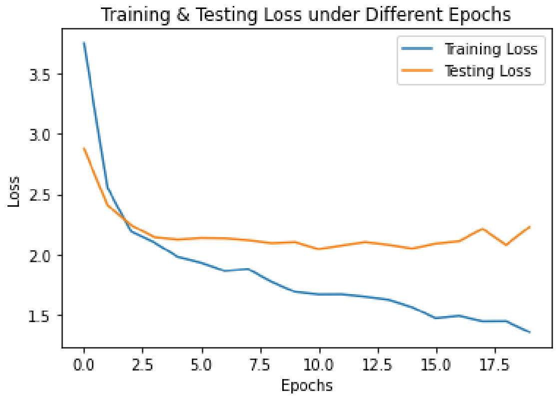


Fig. 2.2.2 Training & Testing Loss under Different Epochs

As shown in Fig. 2.2.1, the testing accuracy stops going up after 7 epochs while the training accuracy approaches 60% after 20 epochs. This indicates the model fits to the training set too much and lost its generalisation. On the other hand, the training set has only 250 sets of data, 25 data sets per type on average, which is too small for a (10 type) classification problem.

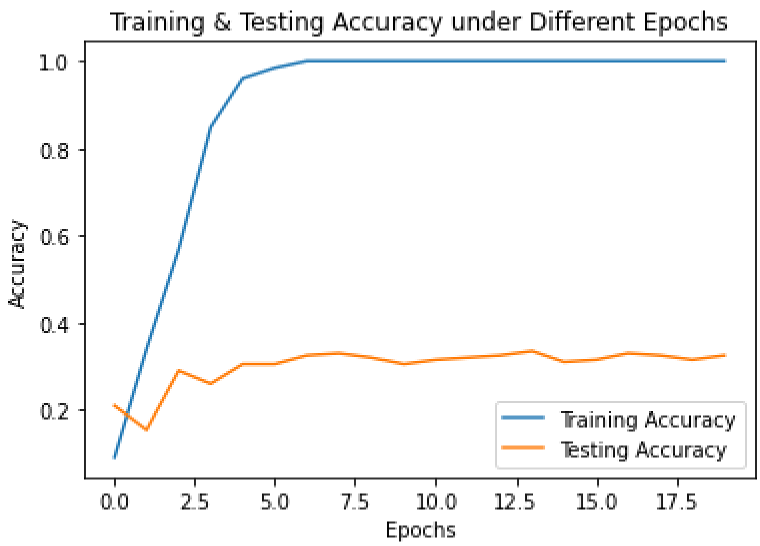
3. (2 pts) Assume that you don't have other extra CIFAR-10 image data except the given subset. You now consider how to improve the testing accuracy of the above MLP model without modifying the model, i.e., without changing the model architecture, hyperparameters, or training settings in any way. One strategy is to use features of the images as the input of the MLP to improve the performance. How will you extract the features from the raw images? Implement the feature extraction and classification on the given dataset, and plot the training accuracy and testing accuracy under different epochs. In your written report, present the recognition results (i.e., show the plot of the training accuracy and testing accuracy under different epochs), explain how you extract the features to conduct classification and why this method can improve the testing accuracy of the MLP.

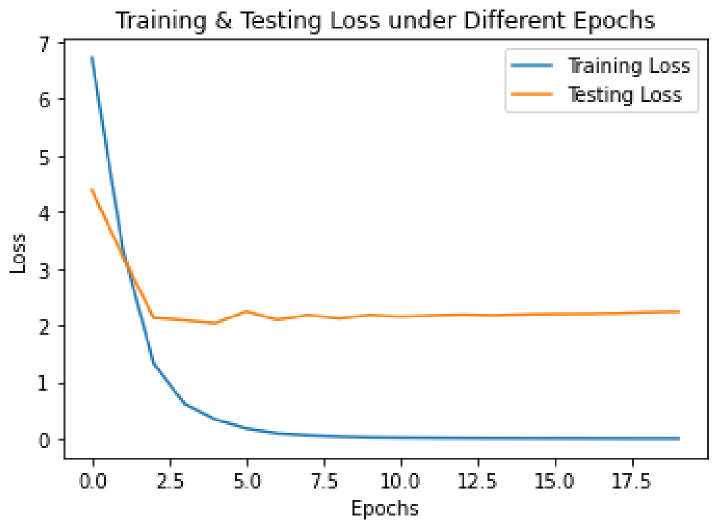
Feature Candidates: edges; Corners; Color; Hog, Sift

By applying the hog (Histogram of Oriented Gradients) on the images, the shape of the edges can be obtained and the accuracy can be increased to

```
max(Q2_3_model.history['val_accuracy'])
```

```
0.33500000834465027
```




Training & Testing Loss under Different Epochs
Training Loss
Testing Loss
Loss
Epochs
0
1
2
3
4
5
6
7
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

Submission

You should make two submissions on the LMS: your code and a short written report with answers to the above two questions. Please note both questions require written response. In Q1, show the formulas for each computation. In Q2, the explanations to each sub question should be less than five sentences.

Submission will be made via the Canvas LMS. Please submit your code and written report separately under the Assignment 2: Code and the Assignment 2: Report links on Canvas.

- Your code submission should include the Jupyter Notebook (please use the provided template) with your code.

- Your written report should be a .pdf with your answers to each of the questions. The report should address the questions posed in this assignment and include any images, diagrams, or tables required by the question.

Evaluation

Your submission will be marked on the correctness of your code/method, including the quality and efficiency of your code. You should use built-in Python functions where appropriate and use descriptive variable names. Your written report should clearly include all of the specific outputs required by the question (e.g., images, diagrams, tables, or responses to sub-questions).

Late submission

The submission mechanism will stay open for one week after the submission deadline. Late submissions will be penalised at 10% of the total possible mark per 24-hour period after the original deadline. Submissions will be closed 7 days (168 hours) after the published assignment deadline, and no further submissions will be accepted after this point.

Updates to the assignment specifications

If any changes or clarifications are made to the project specification, these will be posted on the LMS.

Academic misconduct

You are welcome — indeed encouraged — to collaborate with your peers in terms of the conceptualisation and framing of the problem. For example, we encourage you to discuss what the assignment specification is asking you to do, or what you would need to implement to be able to respond to a question.

However, sharing materials — for example, showing other students your code or colluding in writing responses to questions — or plagiarising existing code or material will be considered cheating. Your submission must be your own original, individual work. We will invoke University's Academic

Misconduct policy (http://academichonesty.unimelb.edu.au/policy.html) where inappropriate levels of plagiarism or collusion are deemed to have taken place.