# CS 579 Trusworthy Machine Learning - Homework 1

Eugene Yong

April 11, 2023

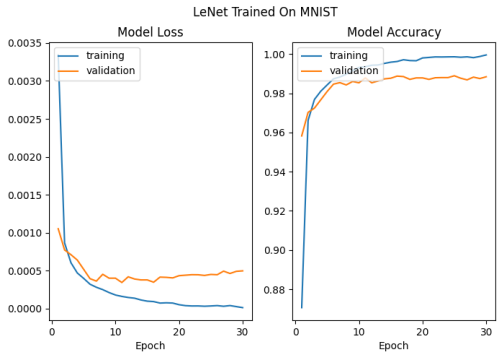
## Task 1: Train and Evaluate Your Models

All base models used Adam optimizer with amsgrad and Cross-Entropy loss fuction. The training data used were not normalized and were split further into 8:2 training data and validation data. Random seed 113 was used to keep consistent result.

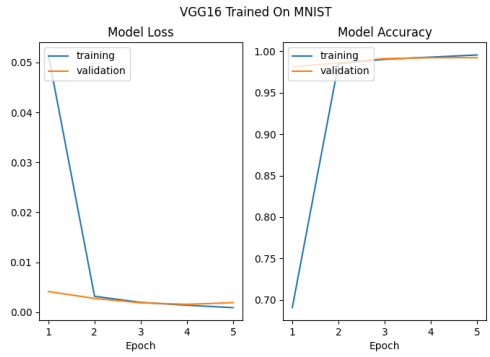
### 1.1 MNIST

| Parameter | LeNet | VGG16 | ResNet18 |
|---|---|---|---|
| Batch Size | 128 | 16 | 128 |
| Learning Rate | 1e-3 | 5e-5 | 1e-4 |
| Epochs | 30 | 5 | 20 |

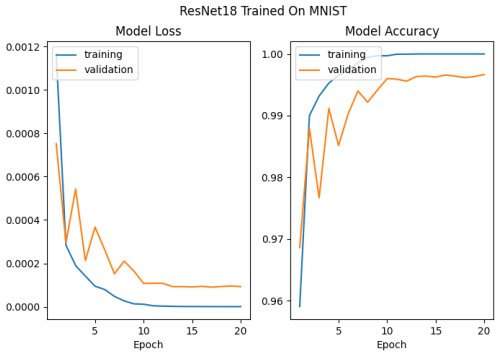
Table 1: Hyperparaters for LeNet, VGG16 and ResNet18 on MNIST dataset




(a) LeNet




(b) VGG16



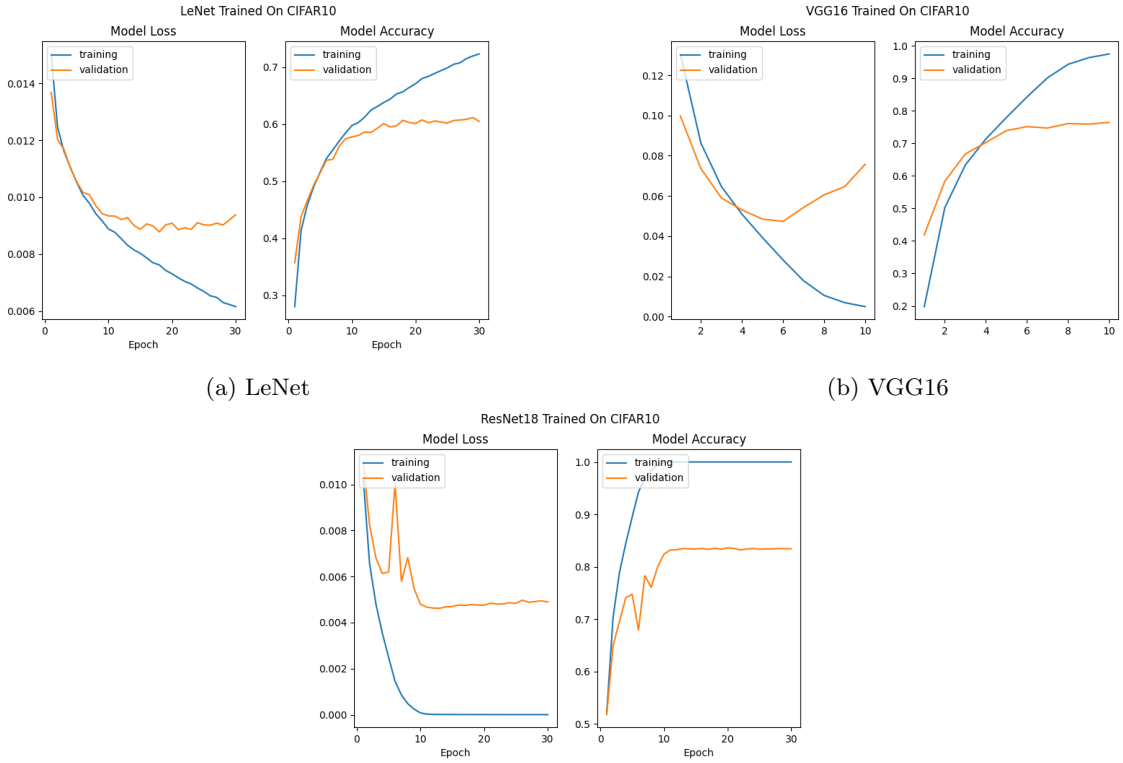

(c) ResNet18

Figure 1: Base model performance on CIFAR10

## 1.2 CIFAR10

| Parameter | LeNet | VGG16 | ResNet18 |
|---|---|---|---|
| Batch Size | 128 | 16 | 128 |
| Learning Rate | 1e-3 | 5e-5 | 1e-4 |
| Epochs | 30 | 10 | 30 |

Table 2: Hyperparaters for LeNet, VGG16 and ResNet18 on CIFAR10 dataset




(a) LeNet

(b) VGG16

(c) ResNet18

Figure 2: Base model performance on CIFAR10

## Analysis

All the models achieved lower loss and higher accuracy in MNIST compared to CIFAR10. For LeNet and ResNet18 for both dataset, the validation loss and validation accuracy graph appear in a more zig-zag shape compared to VGG16.

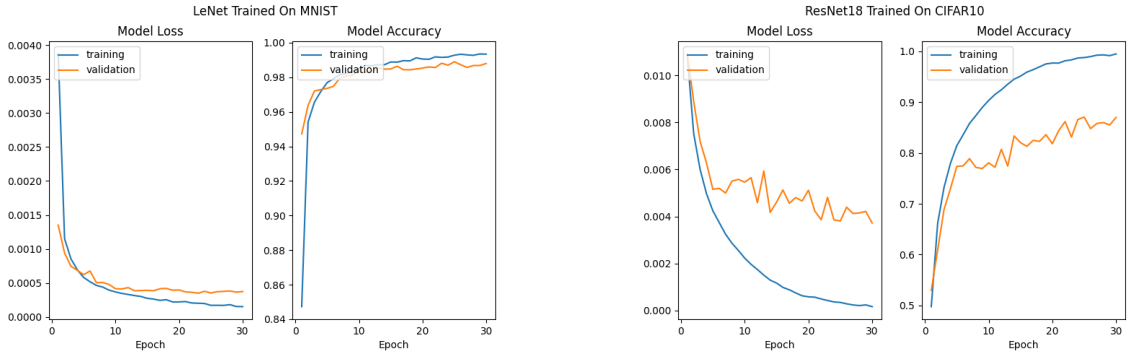
Within models in MNIST dataset, ResNet18 and VGG16 perform roughly the same, getting 99.5% and 99.2% validation accuracy respectively. Whereas LeNet is doing just a little bit worse at 98.5% validation accuracy.

Similar trend was observed models CIFAR10 dataset. The performance of ResNet18 was the best, achieving lowest validation loss and validation accuracy of 83%. Following by VGG16 at 74% validation accuracy with a bell shape validation loss. LeNet perform the worst getting 60% validation accuracy and not be able to fully fit the training set.

All models were able to perform similarly on MNIST because it is a relatively easy to learn dataset compared to CIFAR10. In particular, LeNet was designed for MNIST and was not be able to adapt to the more complicated CIFAR10 that have 2 more color channels than MNIST, thus did worst. ResNet18 did better than VGG16 because it has a work around to the gradient vanishing problem for deep neural network, making it easier to train. My mistake of not normalizing the data could also be one of the reason why VGG16 did poorly for CIFAR10. It could also be that my VGG16 was unlucky and get stuck in a local minima.

# Task 2: Analyze the Impact of Your Training Techniques on Models

## 2.1 Data Augmentation: Rotation (+-20°)

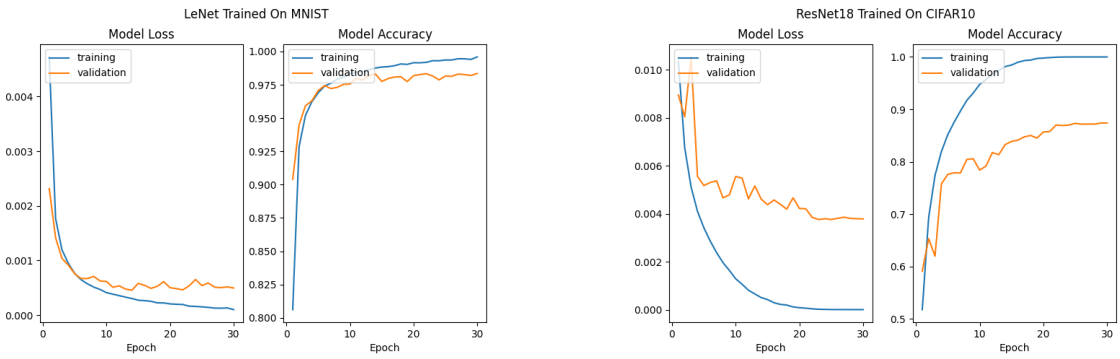



(a) LeNet (b) ResNet18

Figure 3: LeNet on MNIST and ResNet on CIFAR10 with +-20°image rotation on training

Rotation increased the validation accuracy of ResNet18 but not LeNet. My speculation is that LeNet already fit so well into MNIST so rotation hardly help generalize the model. ResNet18 on the other hand improved because rotation helps the model recognize the same object at different angle, preventing overfit.

## 2.2 Data Augmentation: Horizontal Flip




(a) LeNet (b) ResNet18

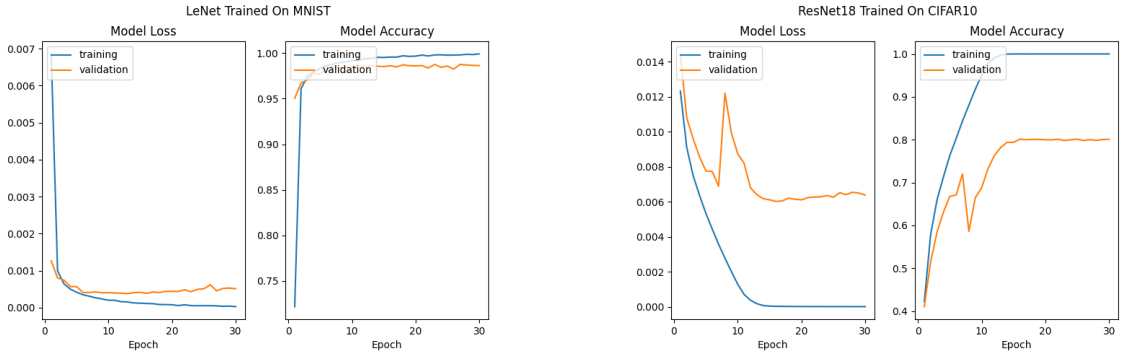
Figure 4: LeNet on MNIST and ResNet on CIFAR10 with 50% chance image gets horizontally flip on training

LeNet perform worse in this case probably because numbers in itself requires right orientation to be recognized. For example a poor written 5 could be mistaken as 2 if flipped. ResNet18 perform better with

horizontal flip because animals and objects do not rely on their orientation to be recognized.

## 2.3 Optimization: SGD

The learning rate of LeNet and ResNet18 was changed to 0.01 and 0.001 respectively.



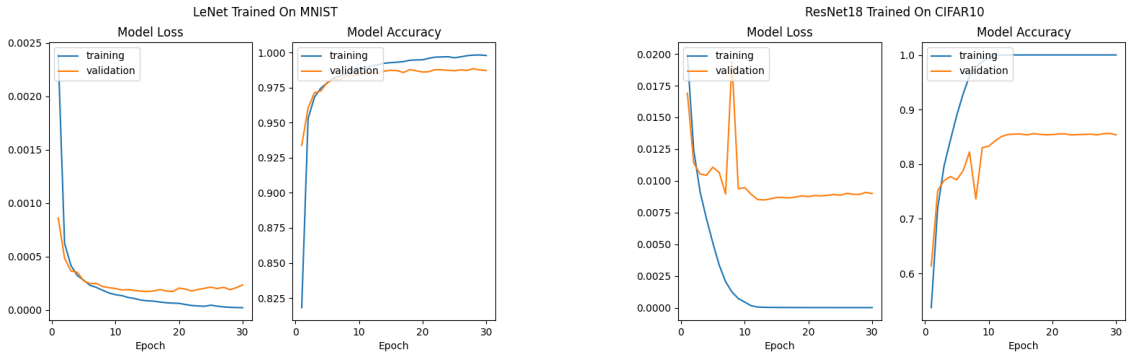

(a) LeNet (b) ResNet18

Figure 5: LeNet on MNIST and ResNet on CIFAR10 with SGD optimizer

The performance of LeNet is about the same but ResNet18 is a bit worse. I saw some information around saying that SGD generalized better than Adam, so more investigation might be needed to reason my result. My guess is that LeNet already did good enough that it's hard to see improvement. For the ResNet18 case, it might be a result of poor learning rate choice and unlucky training.

## 2.4 Batch Size

The batch size of LeNet and ResNet18 was changed to 256 and 64 respectively.



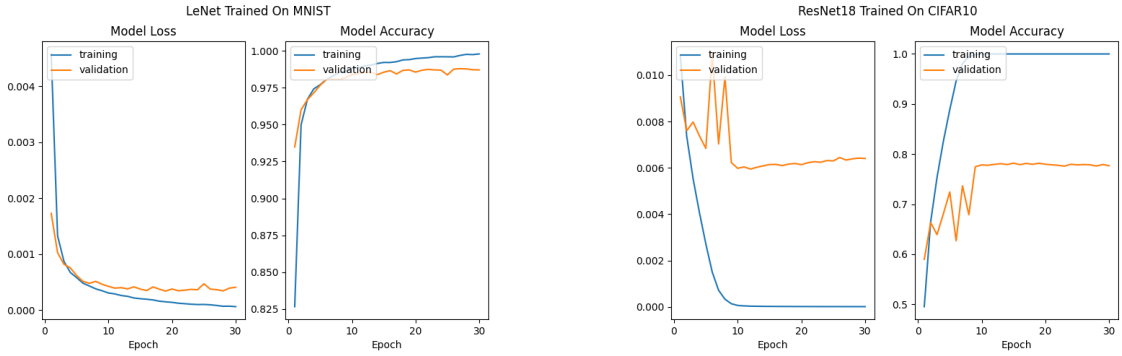

(a) LeNet (b) ResNet18

Figure 6: LeNet on MNIST and ResNet on CIFAR10 with different batch sizes

Batch size is a hyper-parameter to tune. LeNet still perform the same because higher or lower batch size doesn't guarantee better convergence. ResNet18, however, get lucky and reach a better convergence because of different batch size.

## 2.5 Learning Rate

The batch size of LeNet and ResNet18 was changed to 0.0005 and 0.00005 respectively.




(a) LeNet

(b) ResNet18

Figure 7: LeNet on MNIST and ResNet on CIFAR10 SGD optimizer

The model did about the same for LeNet, again, probably because it's easier to fit to MNIST and lowering the learning rate doesn't help it converge to better minima. ResNet18 did worse however, probably because of the lower learning rate make it gets stuck into some worse local minima, unable to escape out.