DELE CA1

Tan Yu Hoe	**P2026309**
Diploma in Applied AI and Analytics	DAAA/FT/2B/04
School of Computing	Singapore Polytechnic
tanyh.20@ichat.sp.edu.sg	
DELE	ST1504
CA1 Assignment	

Table of Contents

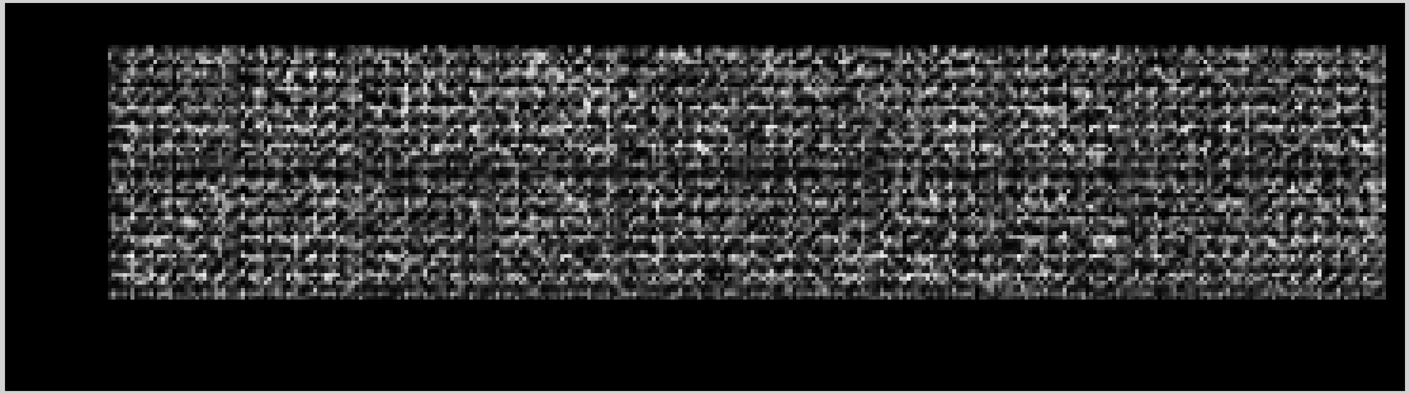
Methodology and Motivation

In the assignment, both Part A (Fashion MNIST) and Part B (CIFAR-10) provided are a dataset of images and labels, and the task is to create a Convolutional Neural Network to perform image classification. Since both task are similar, I will be using the **same approach** (e.g. image pre-processing, augmentation, CNN architectures) for both task.

Personal Objectives

- Obtain a **high test accuracy** relative to the public benchmarks
- Experiment with multiple **training strategies** and **CNN architectures**

Fashion MNIST

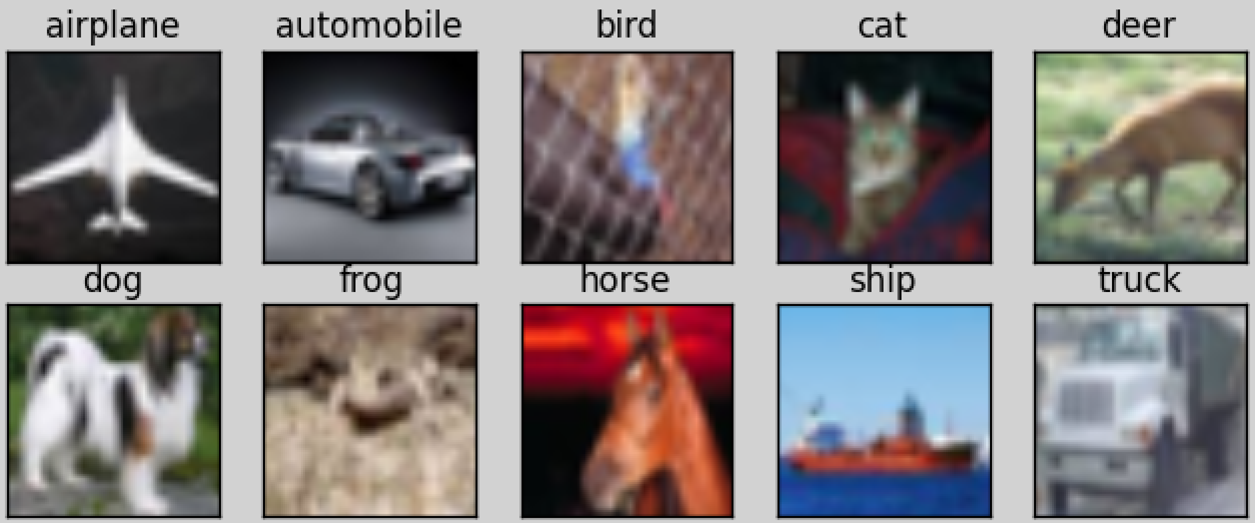


Fashion-MNIST is a dataset of Zalando's (German multinational E-commerce company) article images - consisting of a **training set of 60,000 examples** and a **test set of 10,000 examples**. Each image is a **28x28 grayscale image**, associated with a label from 10 classes. It shares the same image size and structure of training and testing split.

Fashion MNIST

Fashion MNIST

CIFAR-10



CIFAR-10 dataset (Canadian Institute for Advanced Research) is a collection of images that are commonly used to train and benchmark image classification algorithms. It is a subset of 80 million tiny images and consists of 60,000 instances - 32 by 32 coloured images from 10 different classes. There are 50,000 training examples and 10,000 examples.

CIFAR-10


IDX20851: truck
IDX36016: dog
IDX5: automobile
IDX15116: frog
IDX7337: truck
IDX4616: bird
IDX9313: bird
IDX17278: dog
IDX19838: cat
IDX26940: frog
IDX20959: bird
IDX34260: horse
IDX10222: deer
IDX43905: frog
IDX1369: horse

Fashion MNIST

Pixel Normalization / Rescaling

The purpose of normalisation in image processing is to “attempt” to bring the scale of pixels down to a normal distribution $N(0,1)$, in order to mitigate the strong influence of very large or very small pixels. One way to do this is **Z-Score Standardisation** $X' = \frac{X-\mu}{\sigma}$, however, doing this would not provide us any information about the range of values.

The closest thing to $N(0,1)$ would be **Min-Max Normalisation** (also called Unity-Based Normalisation), but bringing the range to $[-1,1]$ instead of $[0,1]$. This is a necessary image pre-processing step to improve better optimization within the Neural Network.

$$X' = a + \frac{(X - X_{min})(b - a)}{X_{max} - X_{min}} = -1 + \frac{X}{127.5}$$

where a and b are lower and upper bound of a predefined range.

Reshape and RGB Conversion

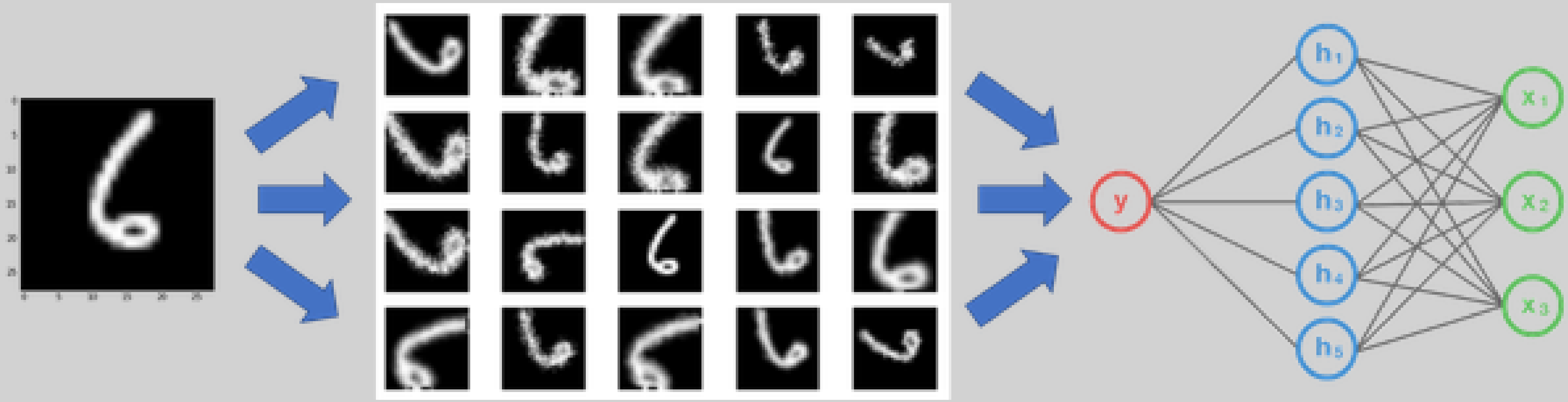
For Fashion MNIST, the convolution layer provided in the Keras API is unable to take in 2D tensors (28, 28) as a single instance, as such I converted the images into a 3D tensor (28, 28, 1), with the last dimension representing the dataset only has one colour channel – grayscale.

Furthermore, most modern architectures (e.g. Inception, ResNet, EfficientNet) only supports images with 3 colour channels. To avoid any future implications, I converted the images within Fashion MNIST to RGB colour space in order to avoid any implications (28, 28, 3).

Image Augmentation

Convolutional Neural Network has an ability to greatly generalise/capture the underlying structure of the training images provided through the use of it's convolution operations. This creates a strong tendency of overfitting within the model.

Using Image Augmentation, we can induce noise into the training data (or also called corruption the training data), in order for the model to have better generalisation. Usually, Image Augmentation is used to mitigate overfitting problems for image classification tasks.



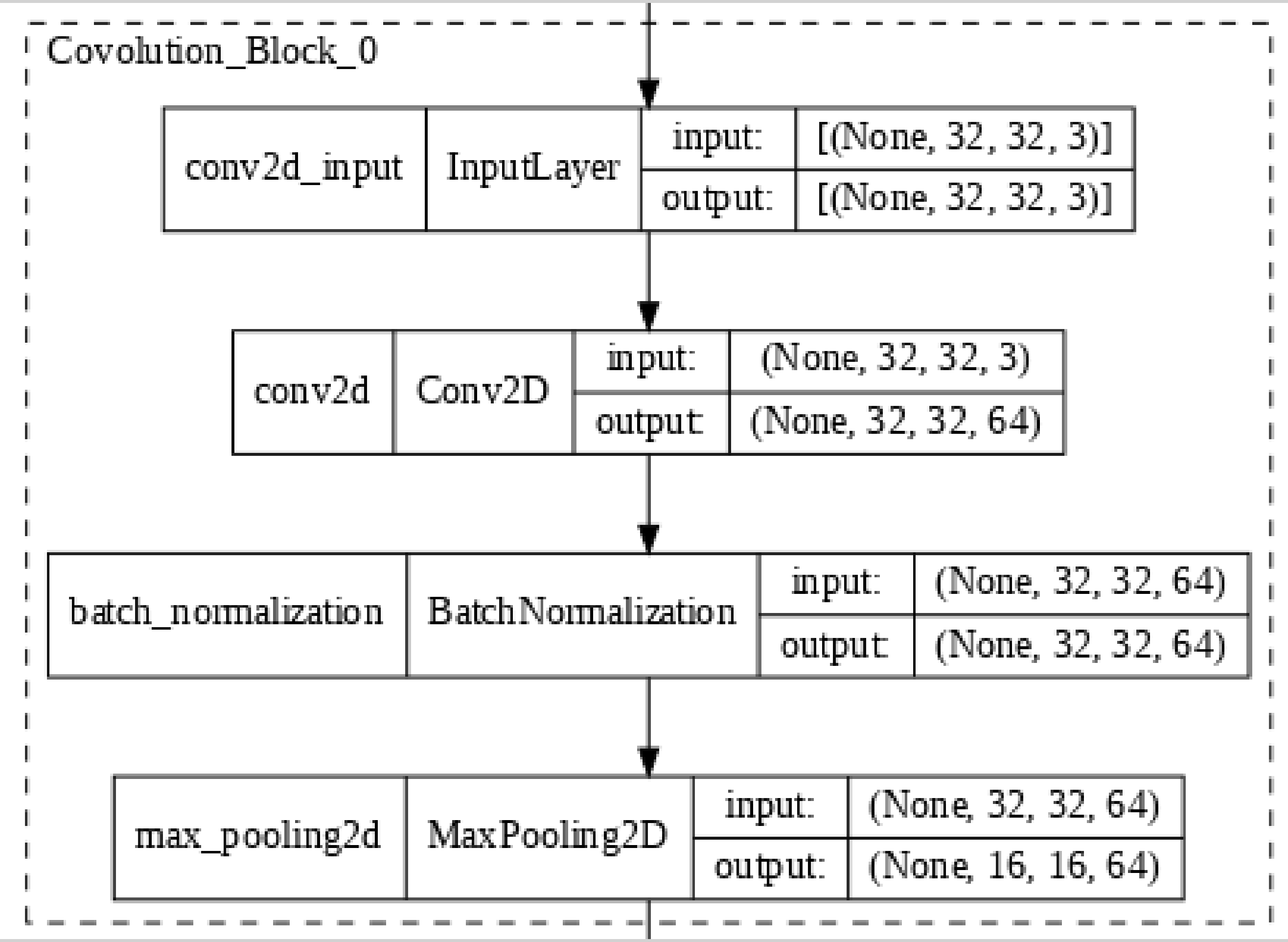
Cutout Augmentation

Cutout Augmentation is a type of image augmentation for Convolutional Neural Networks. It regularize the model by adding random masking into the training images. This prepares the model by learning to take in more features into consideration.

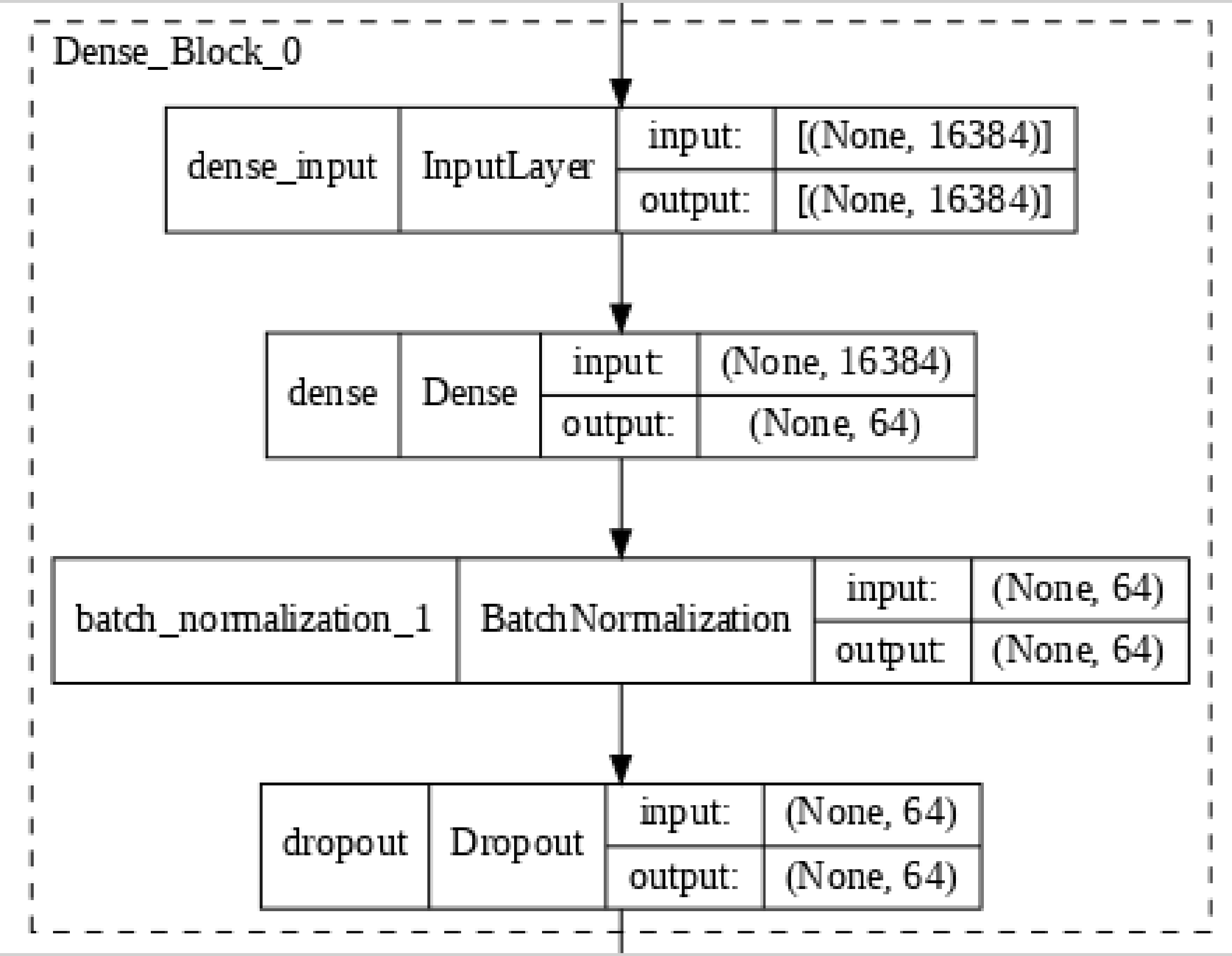
Image Augmentation

Image Augmentation

Baseline ConvNet



Baseline ConvNet

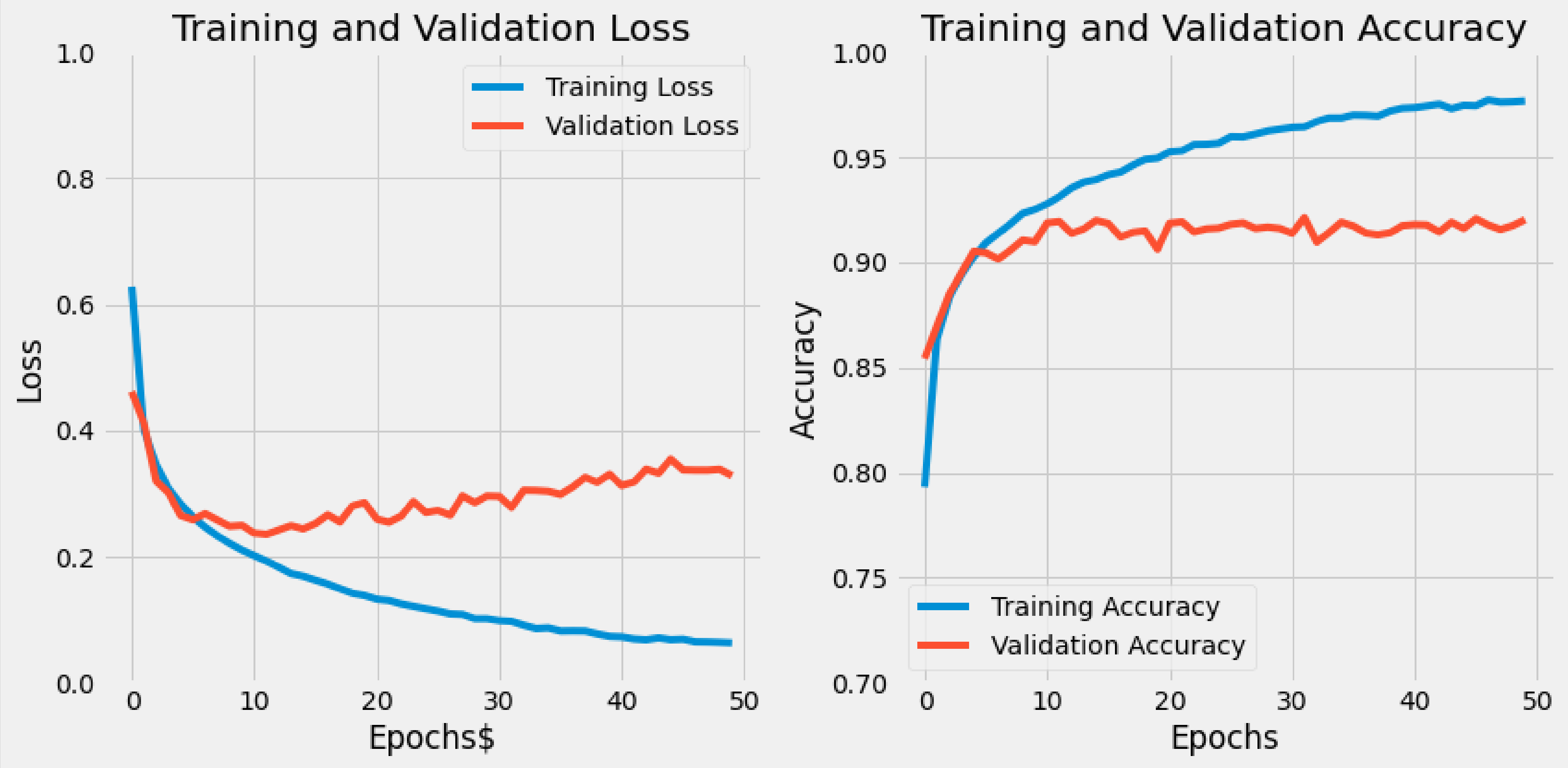


Baseline ConvNet

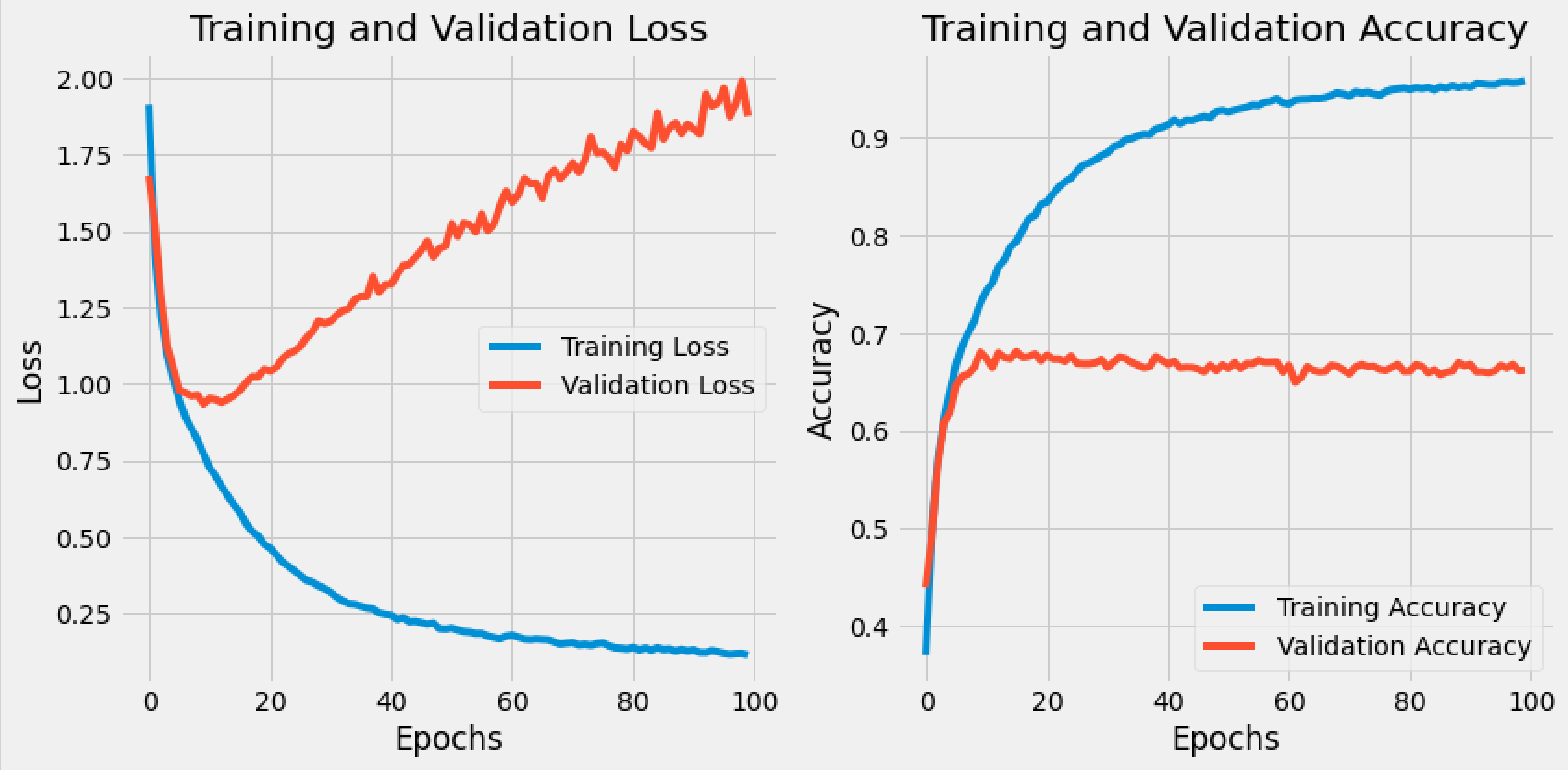
Adam Optimizer
A simple optimizer with great performance without requiring much tuning, which reaches convergence faster than SGD.

Sparse Categorical Cross Entropy aka Log Loss
The task is a multi-class image classification problem, thus I used categorical cross entropy to compute the loss of multiple classes. Since the labels are provided as integers, I used Sparse Categorical Cross Entropy as there is no need to encode the labels.

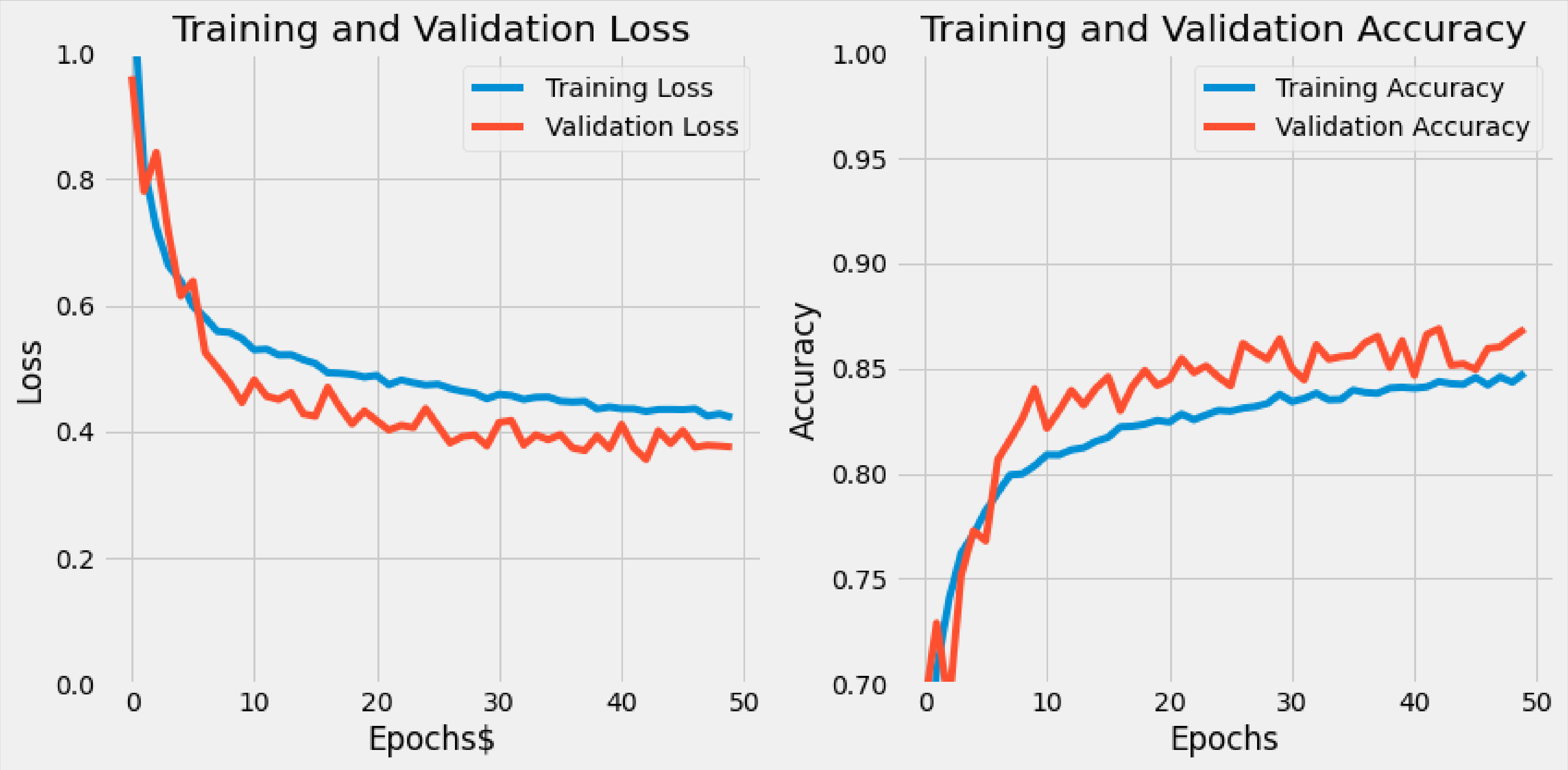
Baseline ConvNet Fashion MNIST



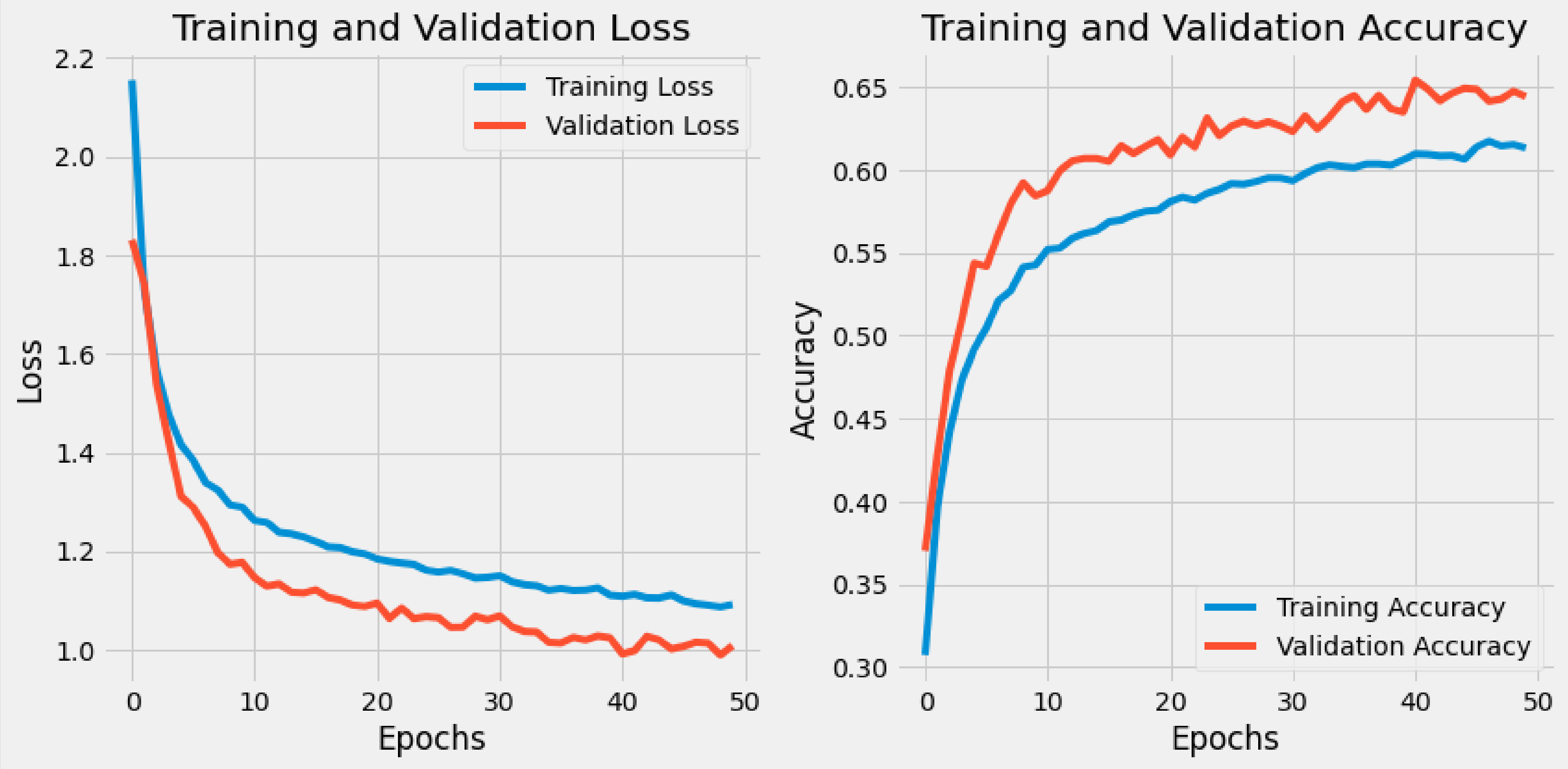
Baseline ConvNet CIFAR-10



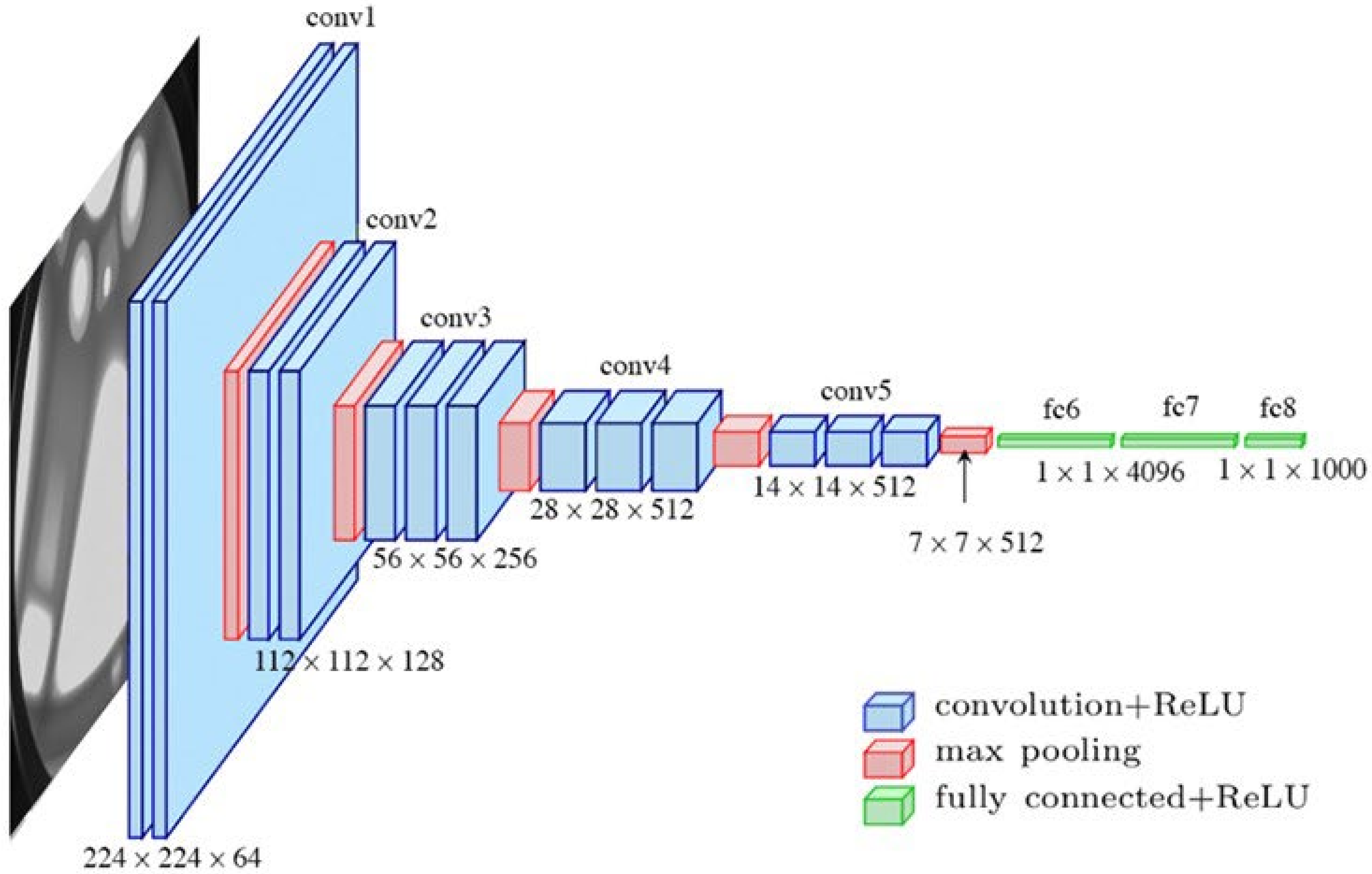
Baseline ConvNet w/ Augmentation - Fashion MNIST



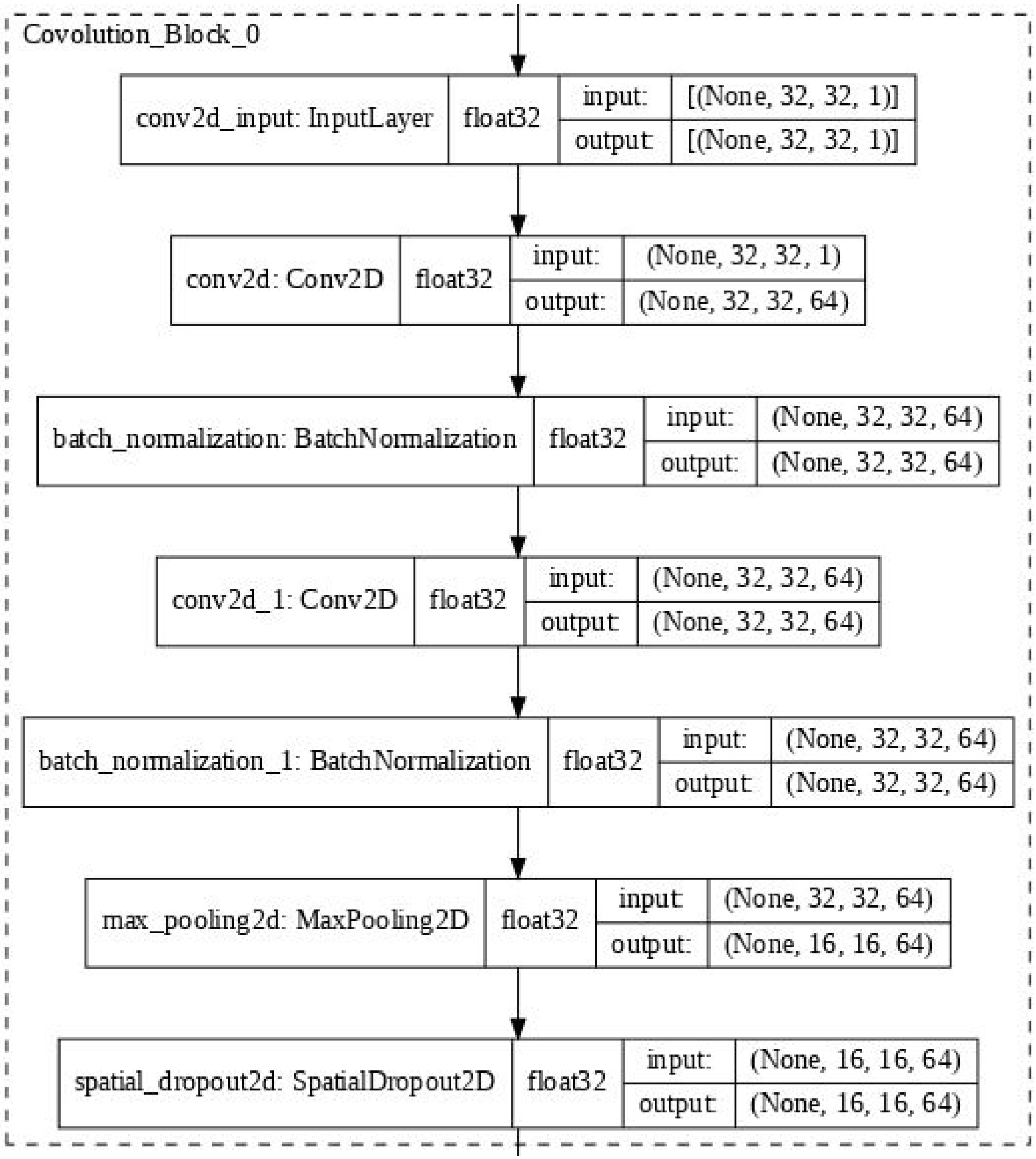
Baseline ConvNet w/ Augmentation - CIFAR-10



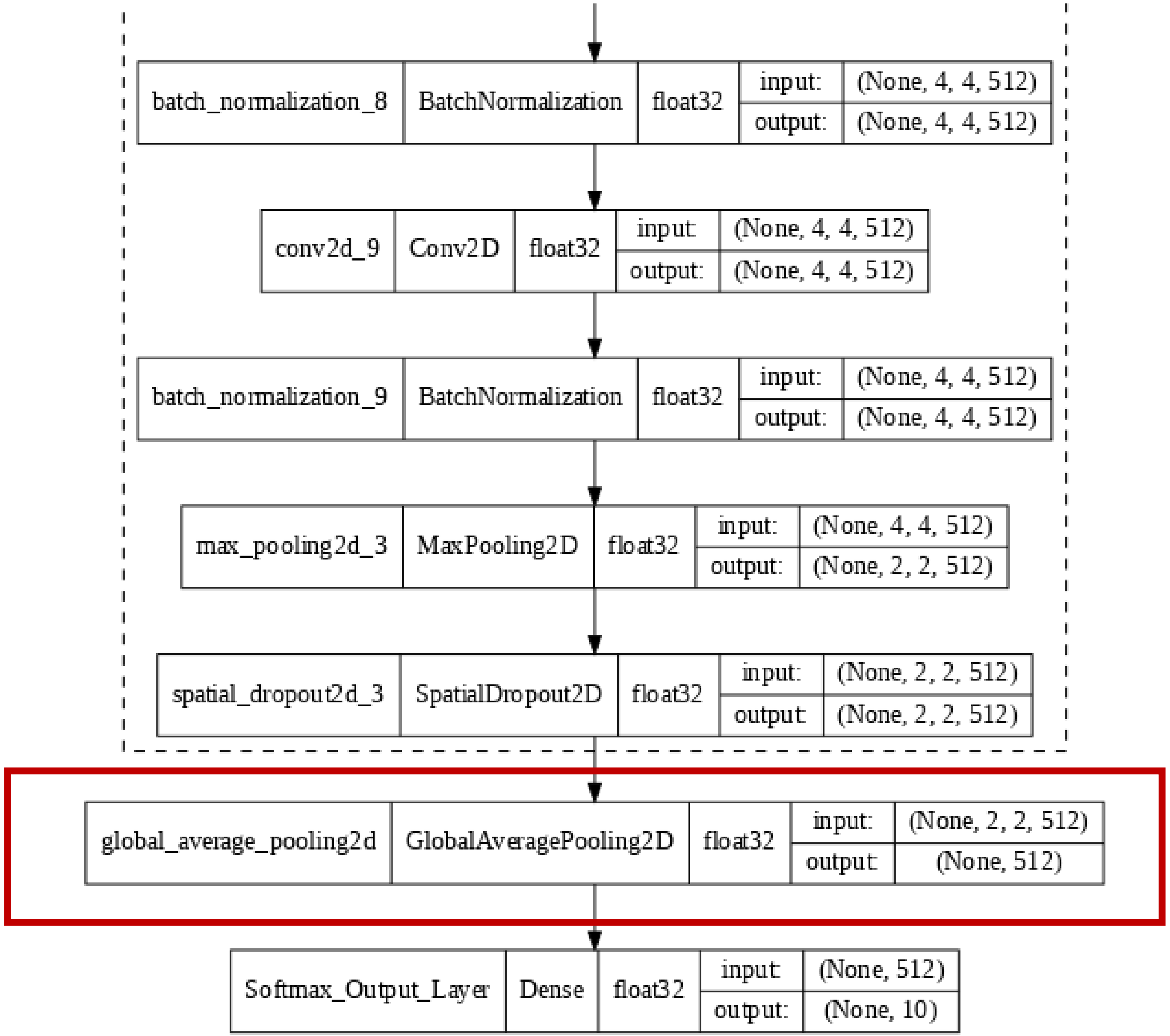
Modified VGG16



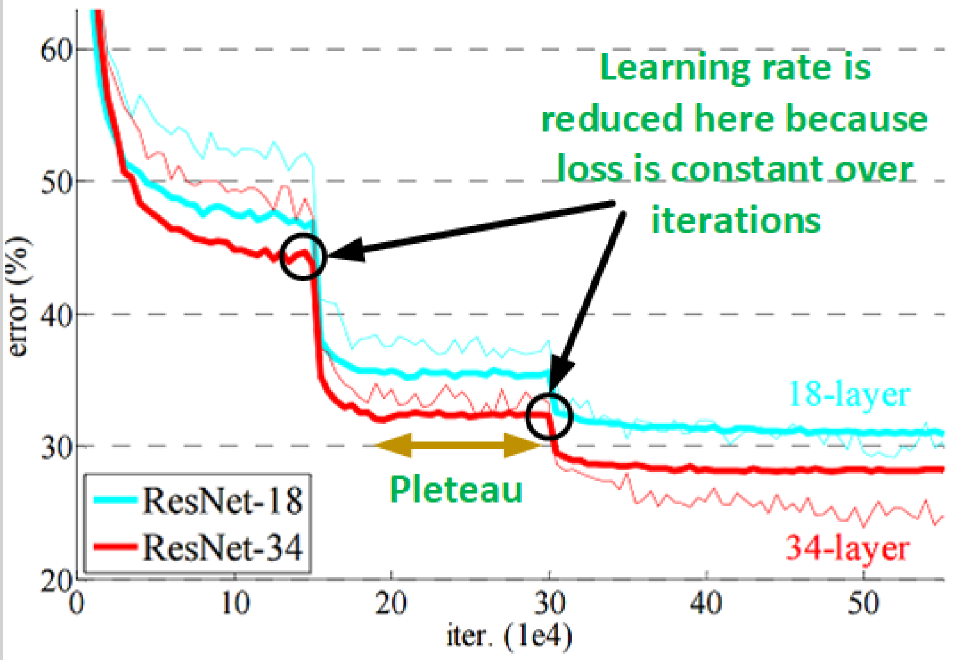
Modified VGG16 Block



Global Average Pooling

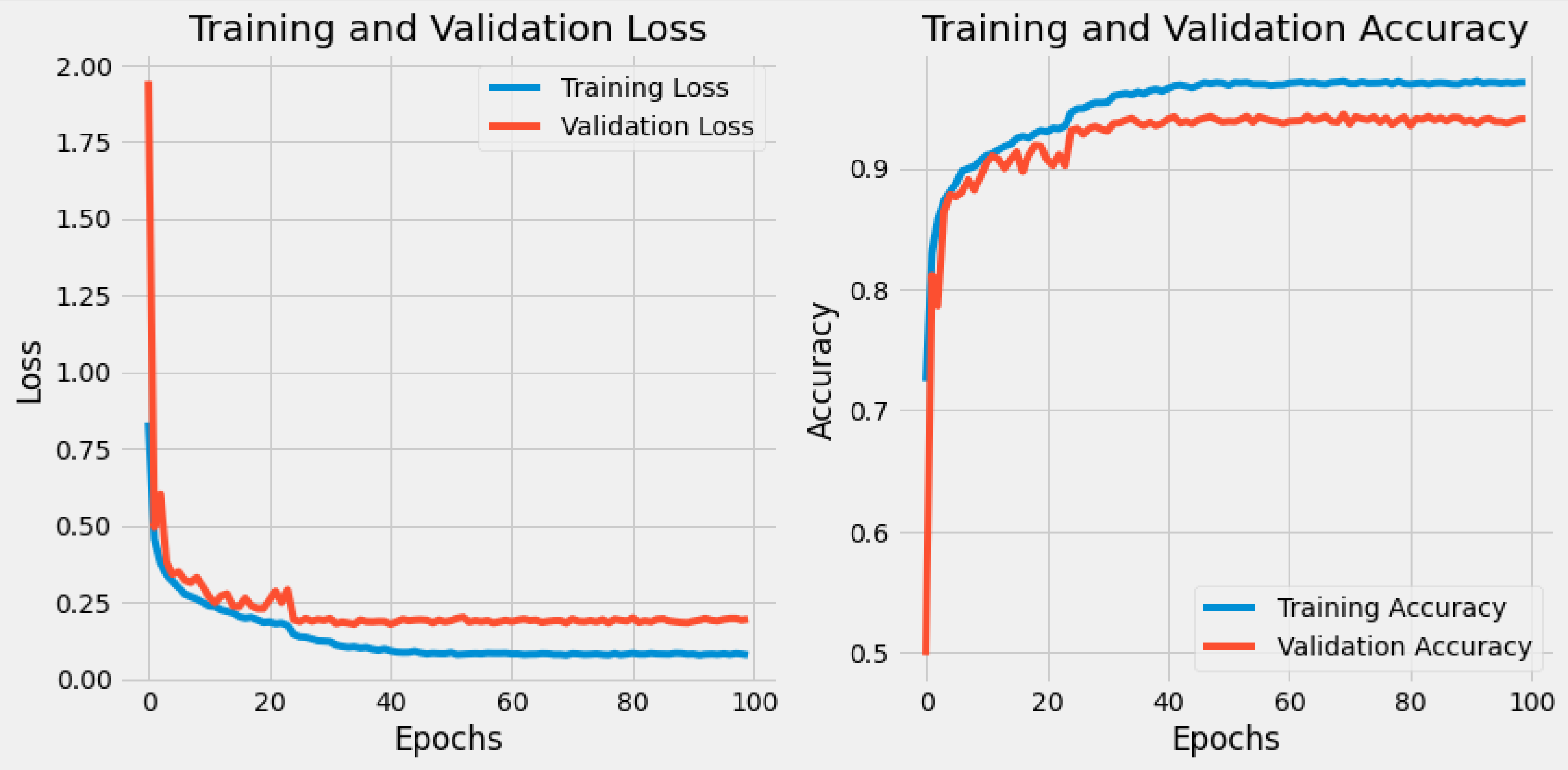


Reduce Learning Rate on Plateau

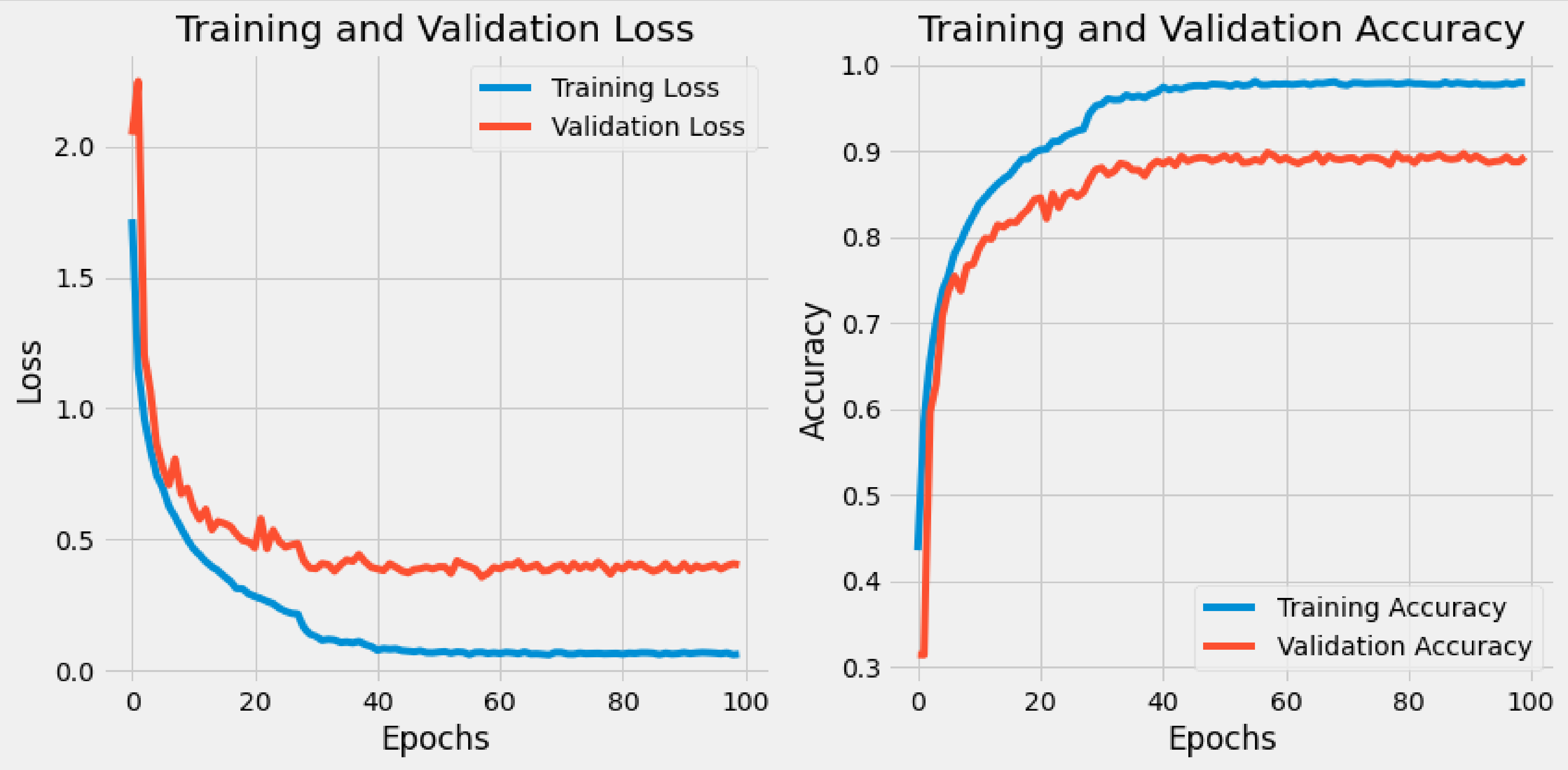


The intuition is to start with a large learning rate to quickly approach a local minima, and reduce it by a certain factor when the validation loss reaches a plateau. This method of having a dynamic learning rate can helps to get out of a local minima.

Modified VGG16 Fashion MNIST



Modified VGG16
CIFAR-10



Issues with VGG16

There is an issue with deeper neural networks like VGG16. As the Neural Network gets deeper (stacking more layers), the architecture would reach a point during training when the gradients would be infinitely large or become zero due to chain rule computation in backpropagation. These issues are commonly referred as **Exploding Gradient** or **Vanishing Gradient** problem.

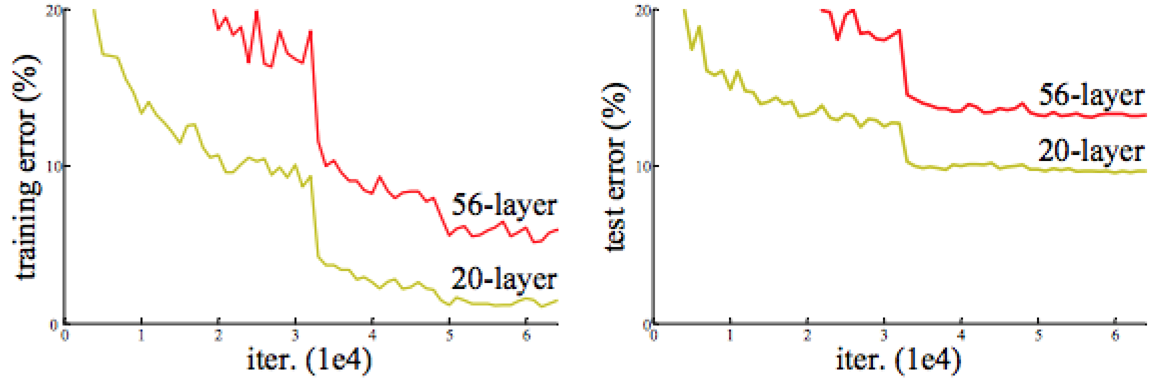
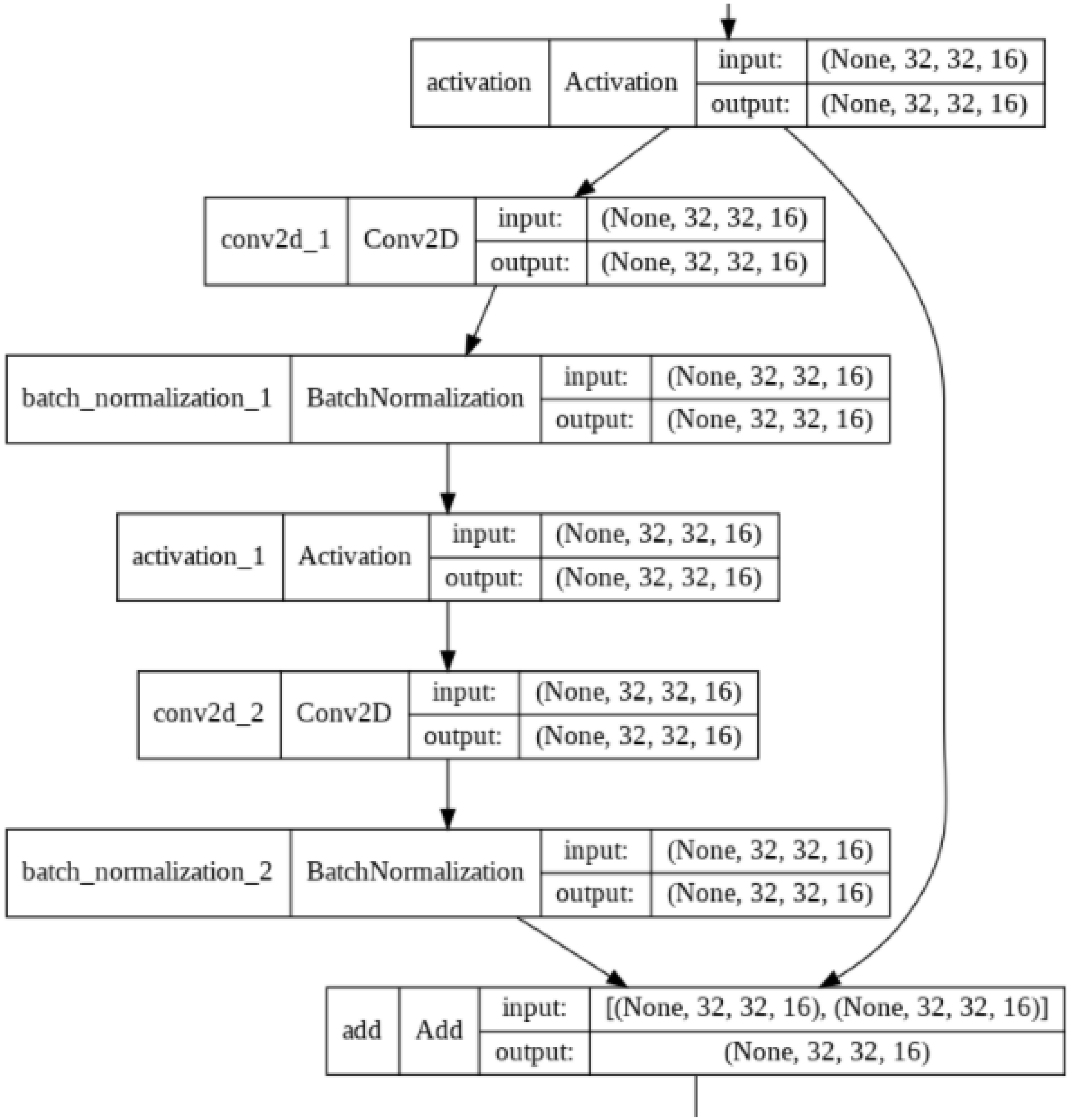
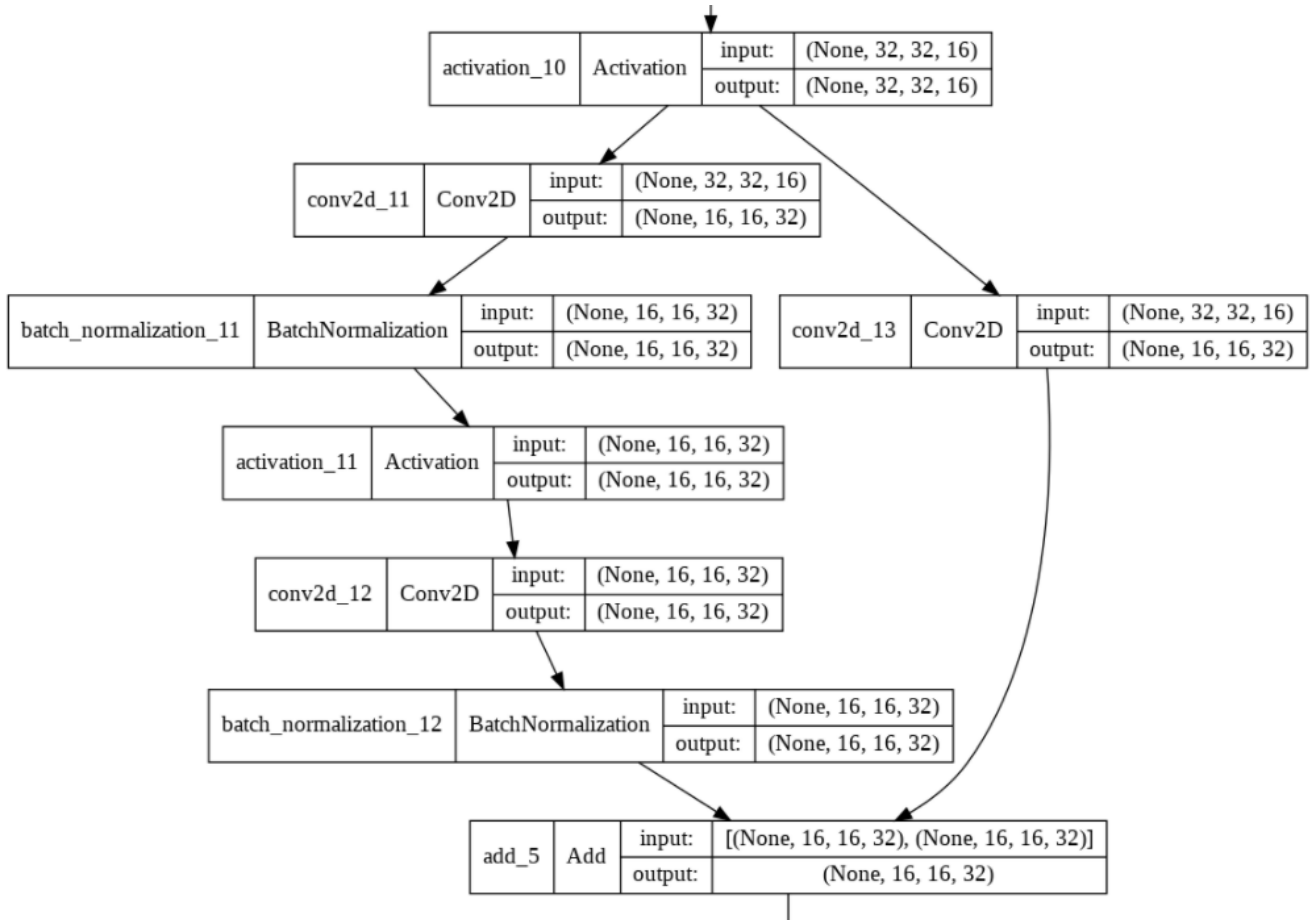


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

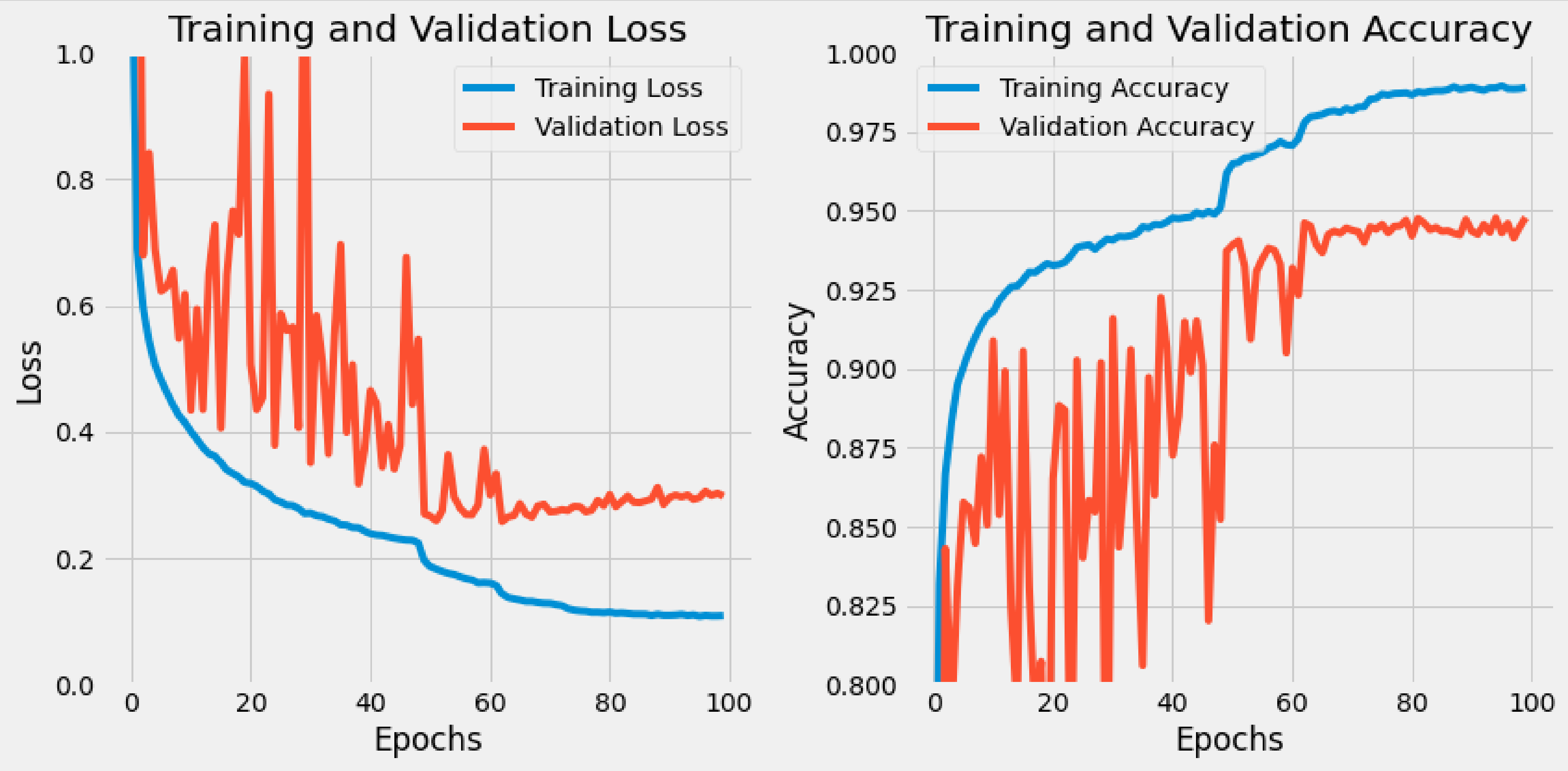
Residual Module



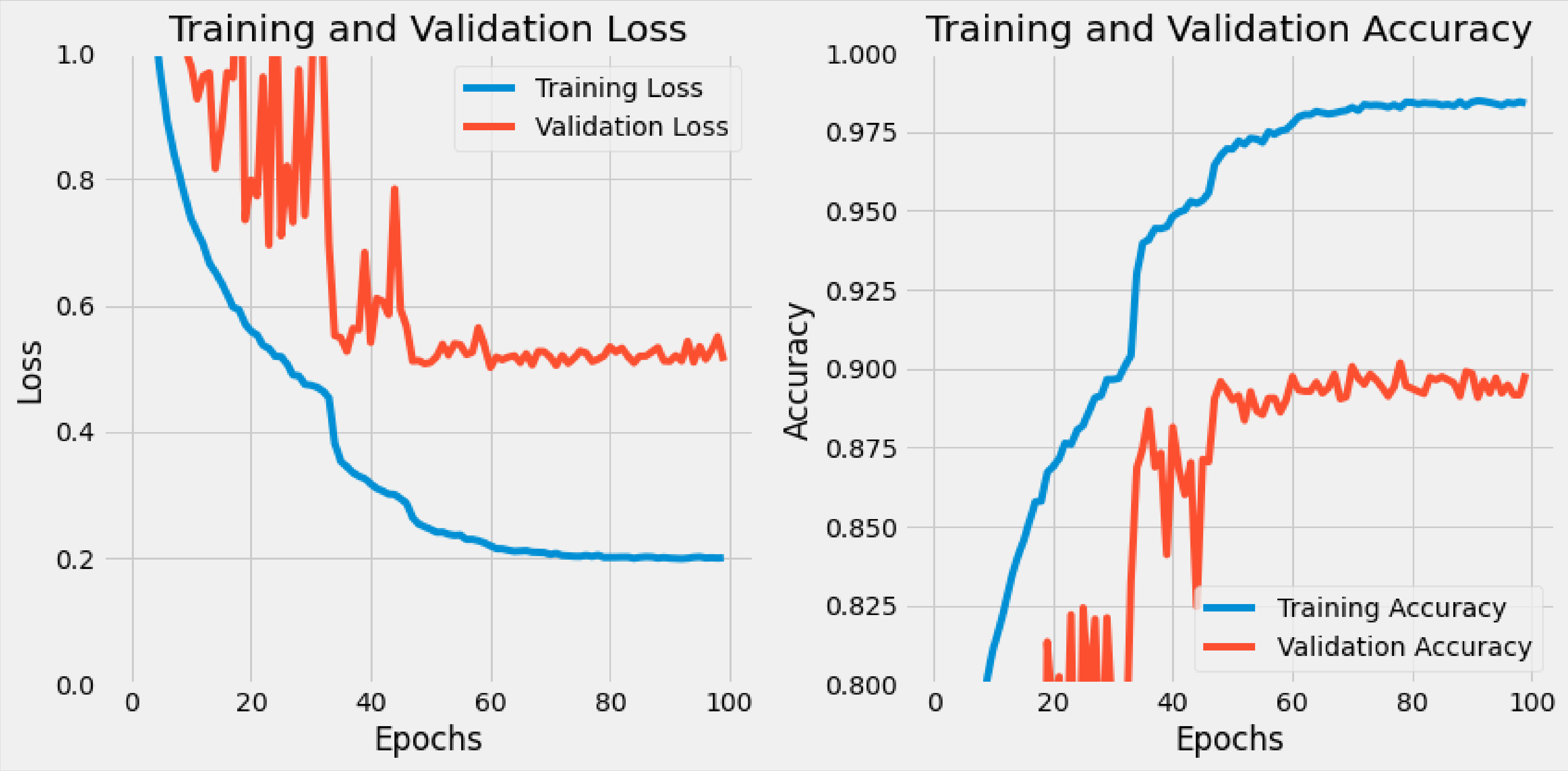
Residual Module



ResNet
Fashion MNIST



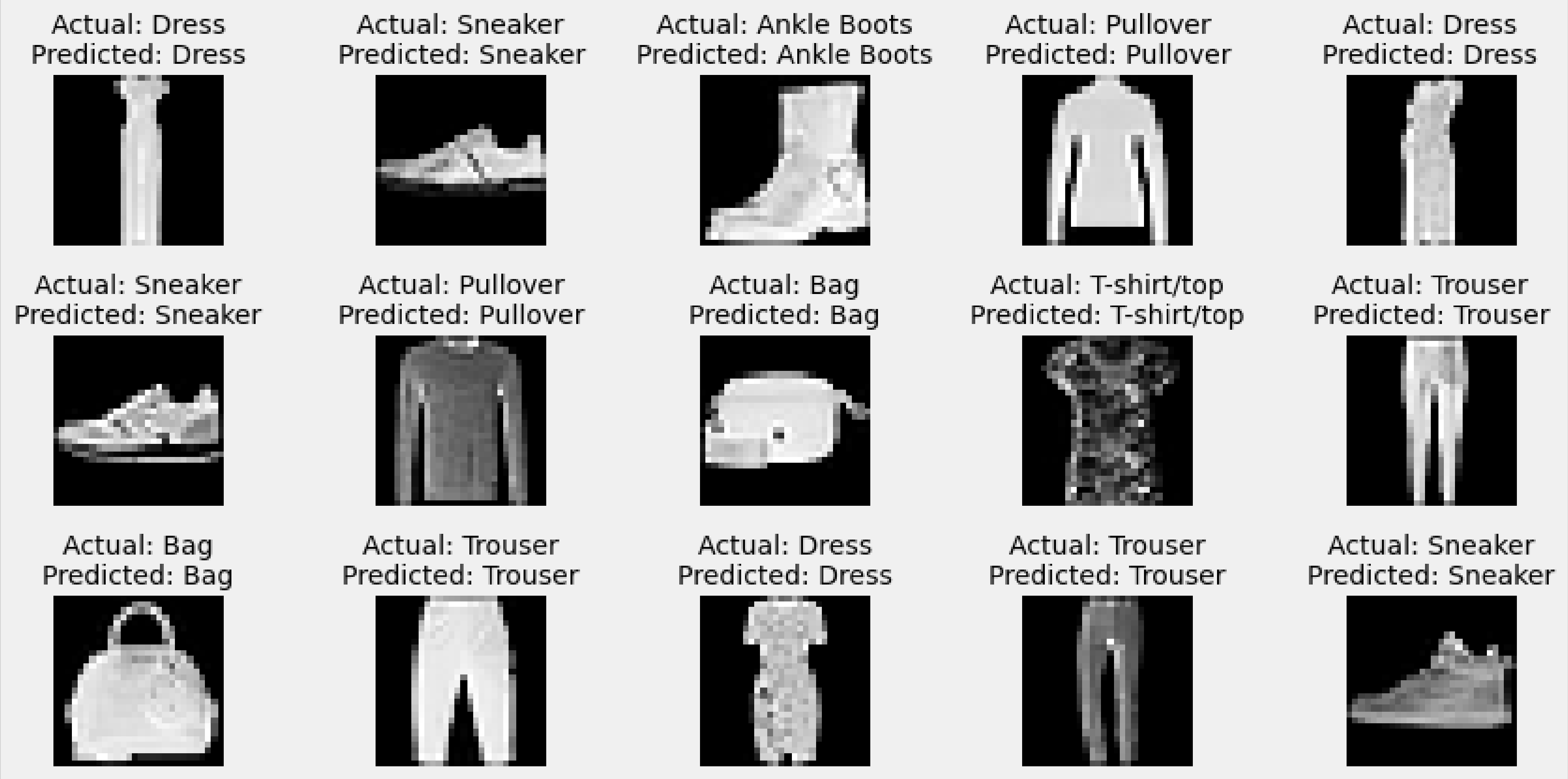
ResNet
CIFAR-10



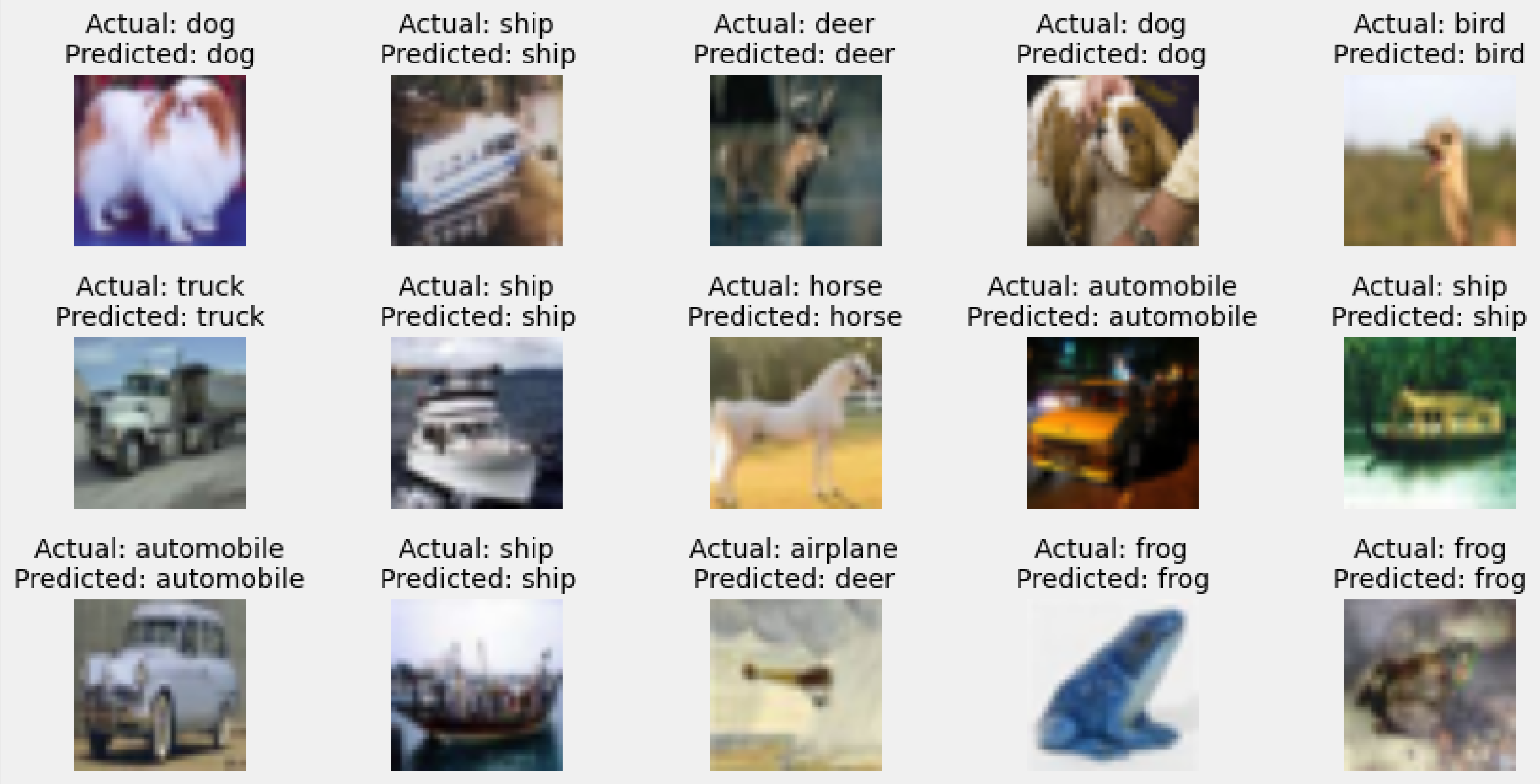
Fashion MNIST Architecture	Training Accuracy (%) [1]	Validation Accuracy (%)	Train Accuracy (%) [2]	Test Accuracy (%)
Baseline Convolutional Neural Network	97.66	92.02	98.66	91.38
Baseline CNN with Image Augmentation	84.78	86.87	89.50	88.09
Modified VGG16	97.06	94.07	98.02	**94.71**
ResNet	**98.90**	94.78	98.19	93.92
Modified VGG16 w/ Full Train Data	96.59	**94.69**	98.00	94.69
ResNet w/ Full Train Data	98.63	94.31	**98.37**	94.31

CIFAR-10 Architecture	Training Accuracy (%) [1]	Validation Accuracy (%)	Train Accuracy (%) [2]	Test Accuracy (%)
Baseline Convolutional Neural Network	95.86	66.22	96.59	65.62
Baseline CNN with Image Augmentation	61.34	64.44	69.71	67.61
Modified VGG16	97.94	89.32	99.02	**90.80**
ResNet	98.39	89.84	97.34	89.00
Modified VGG16 w/ Full Train Data	96.44	89.86	**99.35**	89.86
ResNet w/ Full Train Data	**99.40**	**90.48**	99.13	90.48

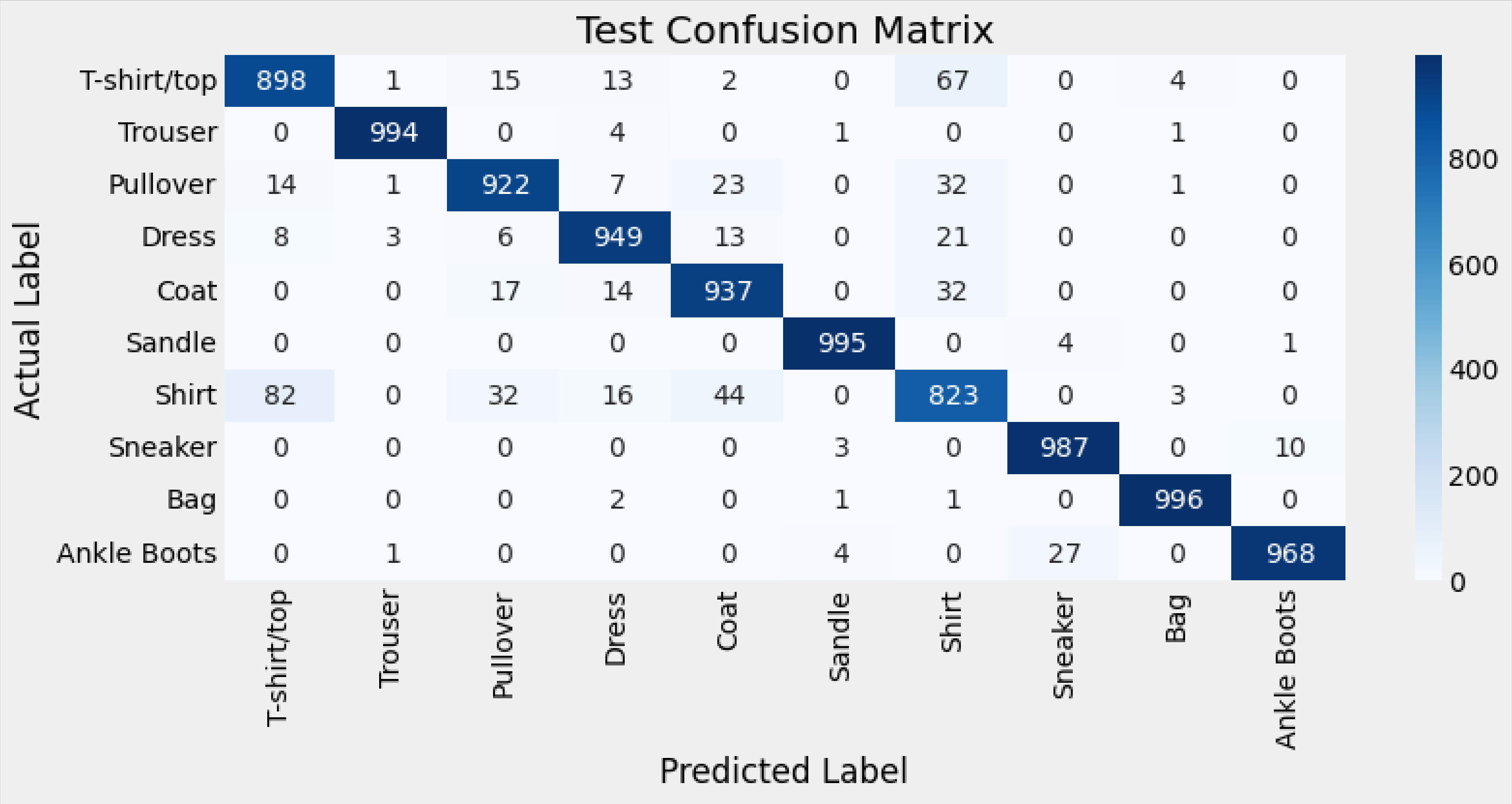
Fashion MNIST



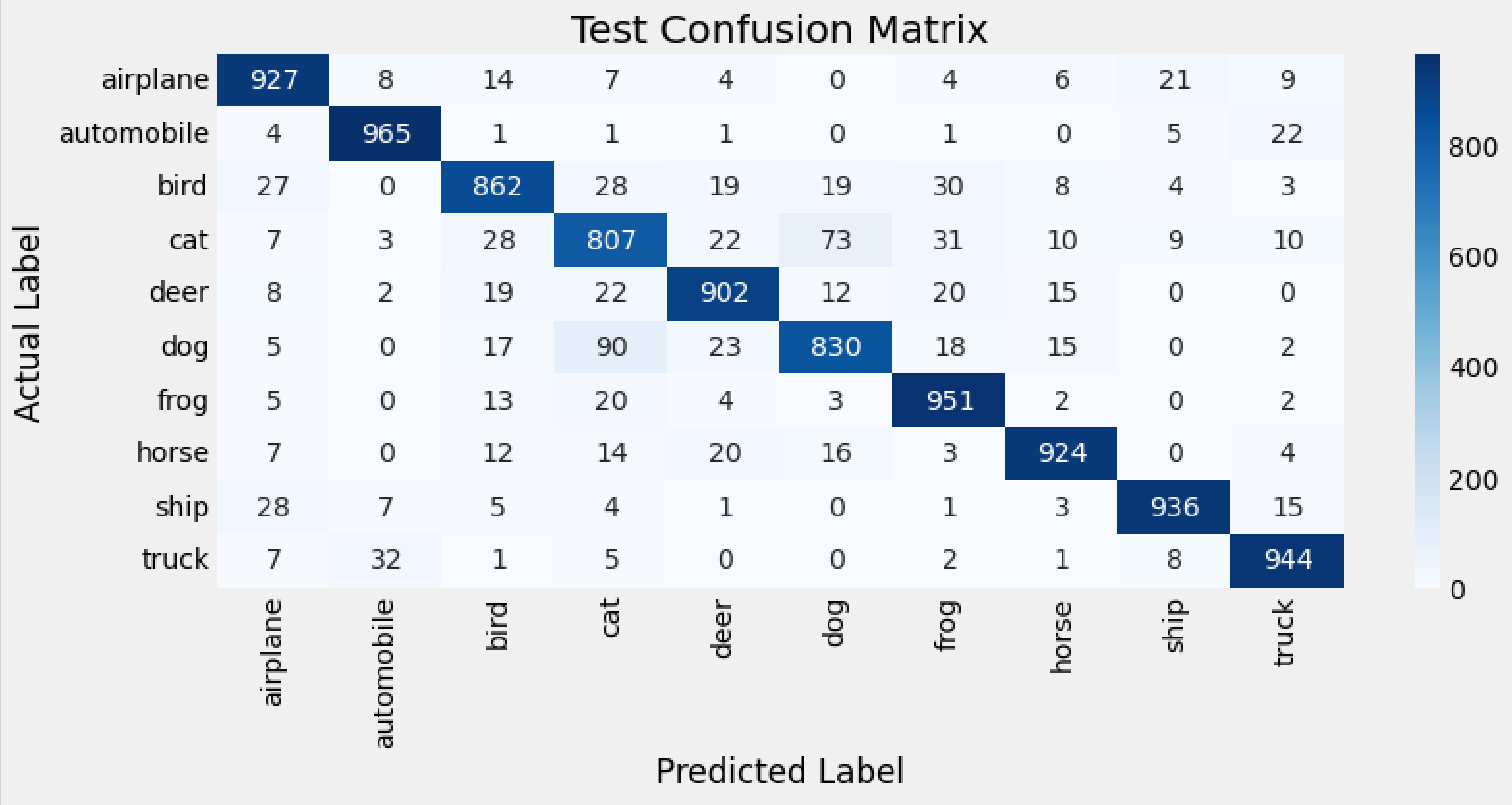
CIFAR-10


Actual: dog
Predicted: dog
Actual: ship
Predicted: ship
Actual: deer
Predicted: deer
Actual: dog
Predicted: dog
Actual: bird
Predicted: bird
Actual: truck
Predicted: truck
Actual: ship
Predicted: ship
Actual: horse
Predicted: horse
Actual: automobile
Predicted: automobile
Actual: ship
Predicted: ship
Actual: automobile
Predicted: automobile
Actual: ship
Predicted: ship
Actual: airplane
Predicted: deer
Actual: frog
Predicted: frog
Actual: frog
Predicted: frog

Fashion MNIST



CIFAR-10


Test Confusion Matrix
Actual Label
Predicted Label
airplane 927 8 14 7 4 0 4 6 21 9
automobile 4 965 1 1 1 0 1 0 5 22
bird 27 0 862 28 19 19 30 8 4 3
cat 7 3 28 807 22 73 31 10 9 10
deer 8 2 19 22 902 12 20 15 0 0
dog 5 0 17 90 23 830 18 15 0 2
frog 5 0 13 20 4 3 951 2 0 2
horse 7 0 12 14 20 16 3 924 0 4
ship 28 7 5 4 1 0 1 3 936 15
truck 7 32 1 5 0 0 2 1 8 944
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck
800
600
400
200
0

Conclusion/Further Improvements

- Both personal objectives of mine has been successfully fulfilled.
- Classes of **similar features** (e.g. Fashion MNIST: upper body clothing, CIFAR-10: smaller animals) has a tendency to be misclassified in CNN.

- For further improvements, I could have done a few more things.
- Try **modern augmentation techniques** such as AugMix, CutMix
- Try **the latest model architectures** such as EfficientNet as VGG16 and ResNet came about 2014 – 2015.

- Moving forward in life, Image Classification is just a small part of Computer Vision, I would like to try out more projects such as Object Localization (R-CNN), OCR, and Semantic Segmentation that might have more meaningful experiences.