CS391L Machine Learning HW6: Deep Learning

Zeyuan Hu, iamzeyuanhu@utexas.edu

EID:zh4378 Fall 2017

1 Introduction

In this task, we are asked to apply the neural networks to handwritten image data[1] to recognize the image identity. The handwritten digit image is described by an 28×28 array of brightness values. We are given 60000 exemplars of images with known labels of the digits. Our objective is to take a new image and identify the digit in the image. We choose track II and implement a Feedforward Neural Network (FFNN) and a Convolutional Neural Network (CNN). We choose these two neural network architectures because they are known to be effective to the image classification task.

2 Method

Both FFNN and CNN are implemented using Keras[2] framework. The network architectures and hyperparameters are detailed below.

2.1 FeedForward Neural Network

The architecture of FFNN is shown in Figure 3. We implement a network with two hidden layers and the hyperparameters are shown in Table 1. We use raw pixel values as input to the network. The images are matrices of size 28×28. Thus, we rehape the image matrix to an array of size 784 (i.e., $28 \times 28 = 784$) and feed this array to the network. The output layer will have 10 neurons for the 10 digits. The implementation is rather straightforward. We convert each image matrix to an array and then we convert the data to float and scale the values between 0 and 1. We then convert the labels from integer to categorical (i.e., one-hot) vectors. Then, we build the model. Note that we need to prevent the overfitting, which is done by adding dropout for each layer. Dropout means a fraction of neurons we randomly turn off during the training process.

[1] `http://yann.lecun.com/exdb/mnist/`

[2] `https://keras.io/`

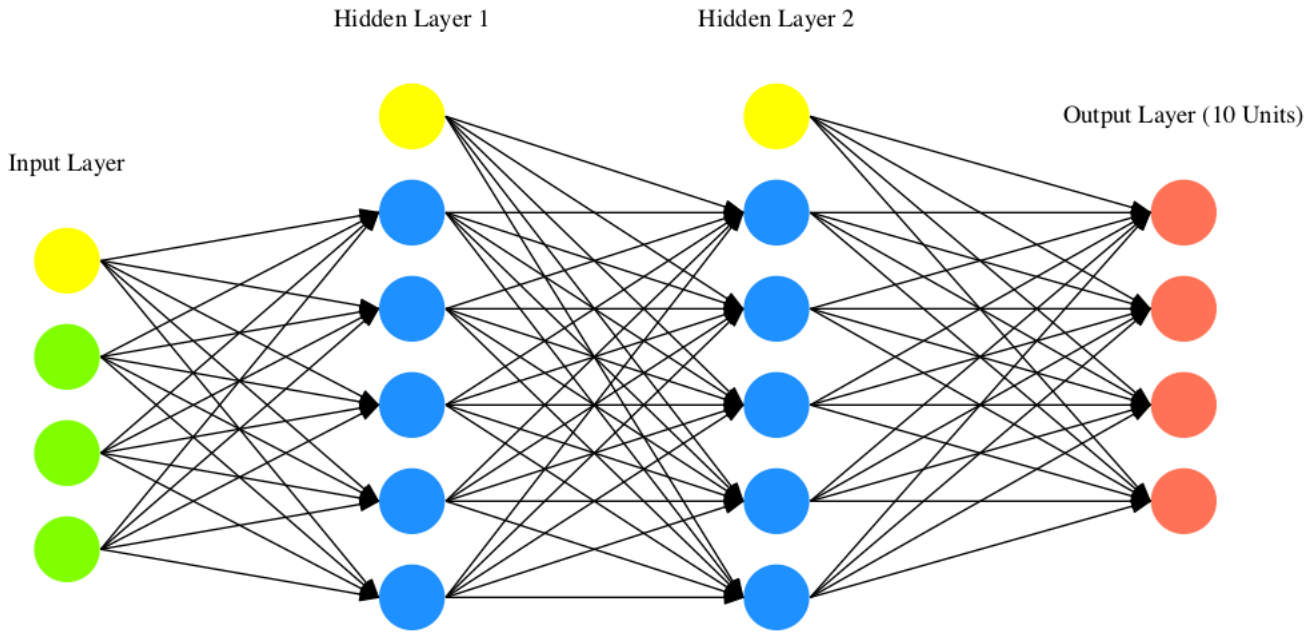

Figure 1: FeedForward Neural Network Architecture

Table 1: FFNN Configuration

Description	Values
batch size	256
number of hidden layer	2
number of hidden units for Hidden Layer 1	512
number of hidden units for Hidden Layer 2	512
dropout rate	0.5
number of epochs	20
learning rate	0.001
optimizer	RMSprop

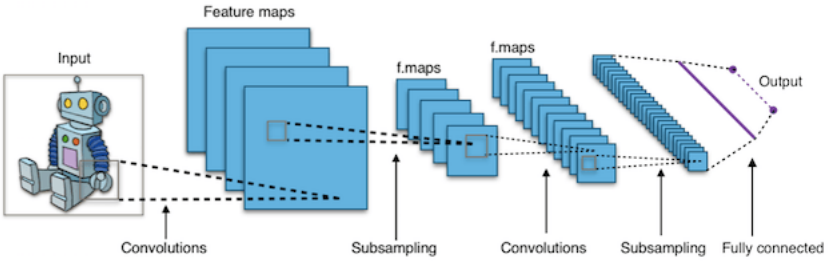

Figure 2: Convolutional Neural Network Architecture

Table 2: CNN Configuration

Description	Values
batch size	128
number of convolution layer	2
number of max pooling layer	1
feature maps for convolution layer 1	32
feature maps for convolution layer 2	64
filter region size	(3,3)
pooling region size	(2,2)
dropout rate	0.5
number of epochs	12
learning rate	1.0
optimizer	Adadelta

2.2 Convolutional Neural Network

The architecture of CNN is shown in Figure 2. We implement a network with two convolution layers and one max pooling layer. Then we include dropout to avoid overfitting and we add a dense layer followed by a softmax layer to get our final classification result. The hyperparameters are shown in Table 2. The implementation is similar to the FFNN one. The only difference is that we now use CNN instead of FFNN.

3 Results

The performance of FFNN is shown in Figure 4 and 5. The accuracy for the test set is 98.36% and the loss is 0.079. The accuracy for the test set is 98.98% and the loss is 0.031.

4 Discussion and Conclusion

In this task, I implement a FFNN and a CNN for handwritten digit image classification task. As shown by the result, the classification accuracy is very high for both the FFNN

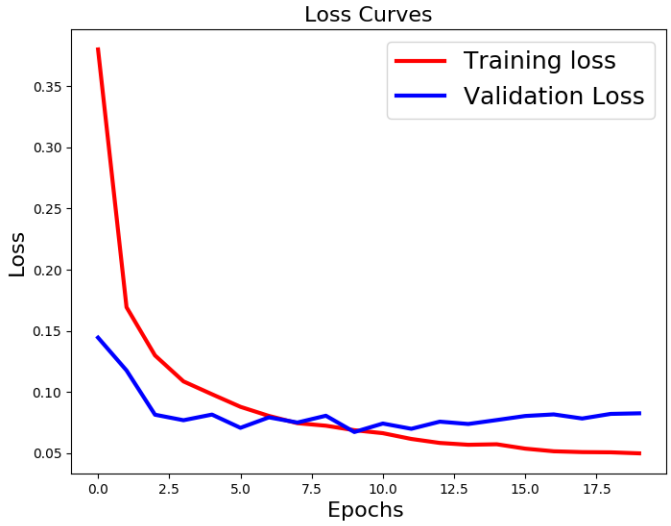

Figure 3: Loss for Feedfoward Neural Network

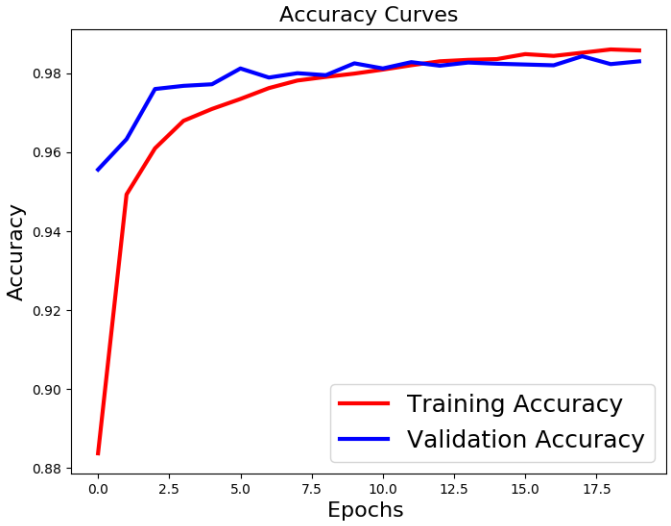

Figure 4: Accuracy for Feedforward Neural Network

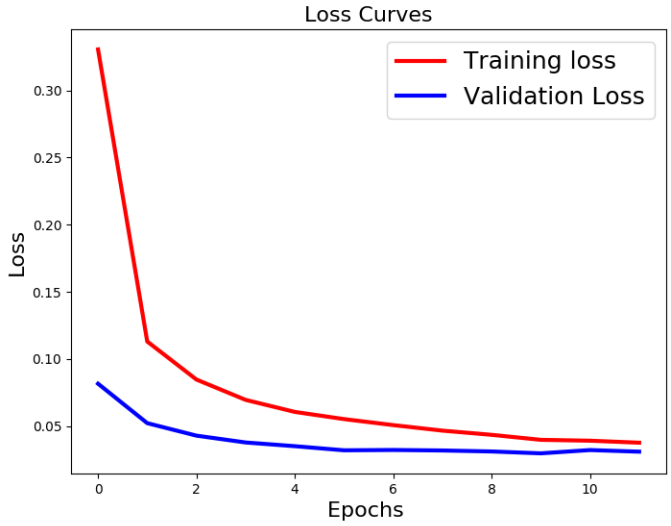

Figure 5: Loss for Convolutional Neural Network

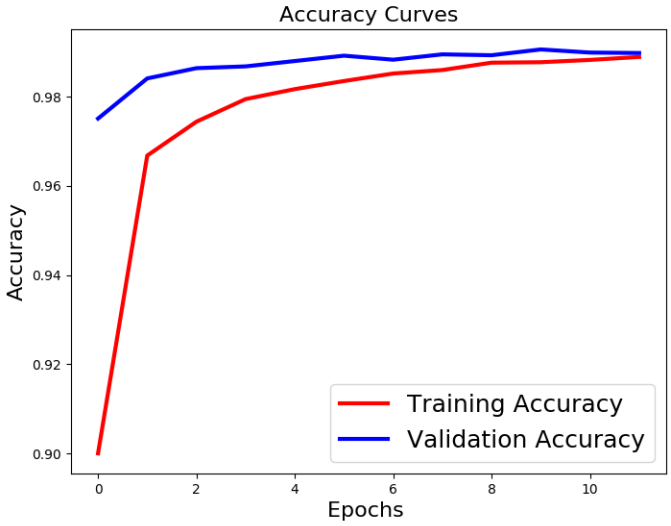

Figure 6: Accuracy for Convolutional Neural Network

and CNN. The main reason for neural networks that can have a good result on this task is that the neural networks are really good at nonlinear classification. In addition, the use of dropout prevents us from overfitting, which helps us to achieve a good accuracy on the test set as well.

Appendices

A How to run the code

To run my code, unzip the `hw6.zip` and get `ffnn.py` and `cnn.py`. Then, you can run the code with the following commands:

- `Python ffnn.py`
- `Python cnn.py`

My code is heavily commented and please take a look if there are any type of questions.