

# COEN 140 Machine Learning and Data Mining

## Lab Assignment #8: Convolutional Neural Network

Guideline: Submit a pdf report to Camino. Also submit all the source code needed to generate the results as a separate zip file to Camino.

**Problem:** Build a convolutional neural network for the recognition task with the fashion MNIST data set. Use the “sparse\_categorical\_crossentropy” loss function, use ‘adam’ as the optimizer, and train your model for 5 epochs.

Adopt the following convolutional neural network structure:

1. Input layer
2. 2d-convolutional layer: filter size 3x3, depth=32, padding='same', strides = (1,1), ReLU activation function
3. 2x2 max pooling layer, strides = (2,2), padding = 'same'
4. 2d-convolutional layer: filter size 3x3, depth=64, padding='same', strides = (1,1), ReLU activation function
5. 2x2 max pooling layer, strides = (2,2), padding = 'same'
6. 2d-convolutional layer: filter size 3x3, depth=64, padding='same', strides = (1,1), ReLU activation function
7. Flattening layer
8. Fully-connected layer: 64 nodes, ReLU activation function
9. (output) Fully-connected layer: 10 nodes, softmax activation function

The following code snippet is for your reference:

```
from tensorflow import keras

from tensorflow.keras import layers, models

fashion_mnist = keras.datasets.fashion_mnist

(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()

train_images = train_images.reshape((60000, 28, 28, 1))

test_images = test_images.reshape((10000, 28, 28, 1))

# Normalize pixel values to be between 0 and 1

train_images, test_images = train_images / 255.0, test_images / 255.0

model = models.Sequential()

model.add(layers.Conv2D(32, (3, 3), activation='relu', \

                        padding='same', strides = (1,1), input_shape=(28, 28, 1)))

# ... to be completed by yourself
```

**Include in the report (70%):**

1. Give the recognition accuracy rate, and show the confusion matrix, both for the test set.
2. For each layer of the network:

**2.a** Manually calculate the number of parameters of that layer. That is, the number of weight elements (**include the bias terms**). Show your work. Verify whether your results are the same as those given by `model.summary()`.

**2.b** Write out the output dimension of that layer.

**2.c** Manually calculate the number of multiplications required to generate that layer. The weights **have bias terms**. You only need to consider the multiplications required to calculate the weighted sums. You don't need to consider the multiplications involved in the softmax function. Show your work.

3. Compare the recognition accuracy rate of the test set, total number of parameters, and total number of multiplications of this CNN to those in Lab 7 (neural network). Analyze your findings and explain why you obtain different (or similar) results.

**Demo/Explain to TA (10%):**

1. How do you build the layers of the convolutional neural network?
2. How do you do prediction for the test samples?
3. How do you calculate the number of parameters and the number of multiplications?

**Grading:**

Report (70%)

Source code (20%)

Demo (10%)