Project-6

CIS 666 Artificial Intelligence

Image classification using Convolutional Neural Network (CNN)

Fathima Syeda

Student ID: 2790024

Abstract

In this project we take the 15- scene classification data and train it using the Convolutional Neural Network (CNN) to be able to classify images into 15 scenes. The scene classification data consists of a total of 4485 images of 15 scenes. We set aside 30% of the images for testing and use the remaining 3140 images in training the CNN. Apart from the 15- scene data we also use the MNIST dataset for classification using CNN. We resize the 15-scene dataset images to 32x32 and the MNIST dataset images to 28x28 for faster computation. The accuracy of the classification of both the datasets is found and compared. We use the ReLu activation function in the convolution layers and soft-max activation in the final fully connected layer.

Contents

1. Introduction:

A convolutional neural network is a special kind of Artificial neural network that specialises in detecting patterns. A CNN network consists of three kinds of layers: Convolutional layer, Pooling Layer, Fully connected Layer.

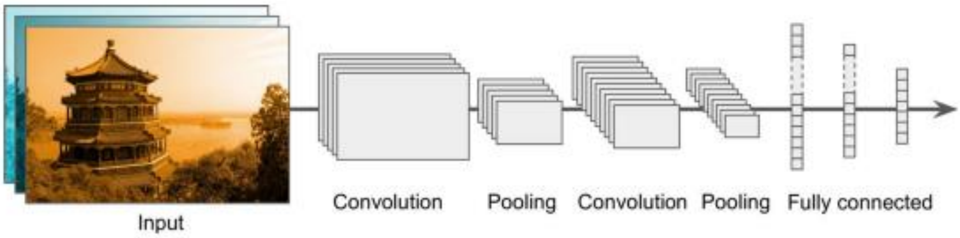


Figure 14-11. Typical CNN architecture

Convolutional Layer:

The most important building block of a CNN is the convolutional layer. neurons in the first convolutional layer are not connected to every single pixel in the input image but only to pixels in their receptive fields. In turn, each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer. This architecture allows the network to concentrate on small low-level features in the first hidden layer, then assemble them into larger higher-level features in the next hidden layer, and so on.

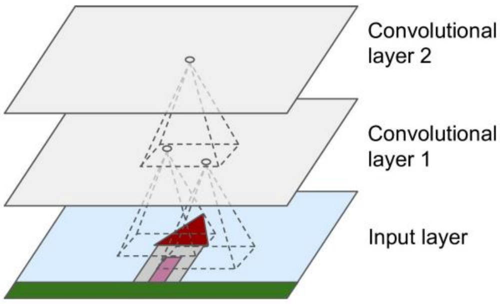


Figure 14-2. CNN layers with rectangular local receptive fields

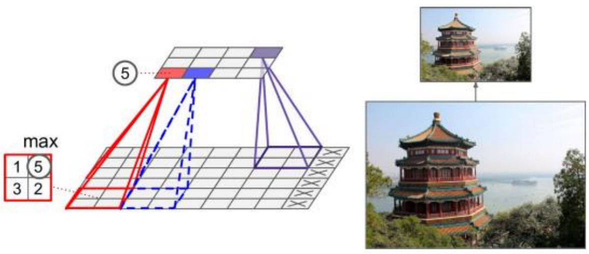


Figure 14-8. Max pooling layer (2 × 2 pooling kernel, stride 2, no padding)

Pooling Layers:

The goal of these layers is to subsample (i.e., shrink) the input image in order to reduce the computational load, the memory usage, and the number of parameters (thereby limiting the risk of overfitting). Just like in convolutional layers, each neuron in a pooling layer is connected to the outputs of a limited number of neurons in the previous layer, located within a small rectangular receptive field.

Fully Connected Layer:

This layer is heavy data driven layer . It gives the top best or most probable classification. In this layer the actual learning of the non-linear combination of the features occurs.

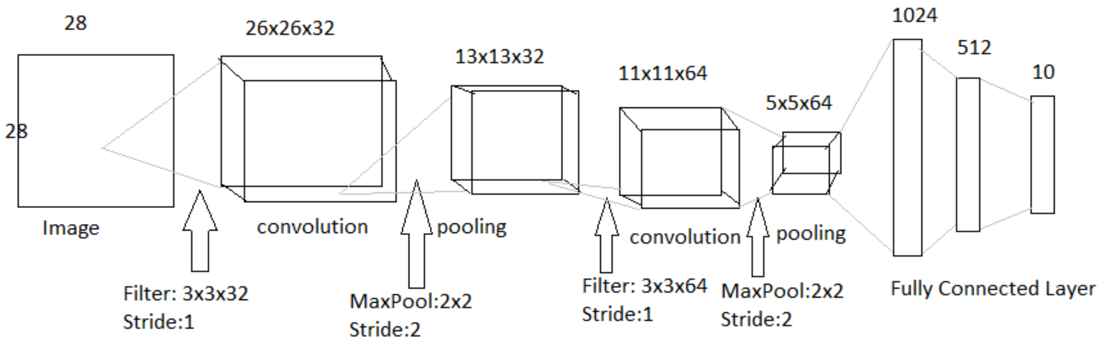
Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (pp. 456-457). O'Reilly Media. Kindle Edition

2. Steps involved in Convolutional Neural Networks.

1. Read the MNIST dataset and the 15-scene dataset from the MNIST website and https://www.kaggle.com/zaiyankhan/zaiyan-assignment-5 respectively.
2. Set aside 70 % of the data from each of the 15 categories of the scene dataset for training and the rest are used for testing.
3. Resize the images of the scene dataset to 32x32x1.
4. Set up the parameters for the Convolution layer 1 as follows:
    a. Filters dimension=[3,3,D], Input=[wxHxD], **number of filters =32**, stride =1
    b. Set the activation function as inside the convolution layer as **ReLu**.
5. Set up the parameters for the Pooling Layer as **Max Pooling : 2x2 with stride =2**
6. Again, add another Convolution layer with the same parameters as the first one but the **number of filters =64** and the input is the result of pooling from the previous layer.
7. Add another Pooling layer with the same parameters as those in the previous one.
8. Finally add three Dense() of 1024,512,10 to form the Fully Connected Layer.
9. Compile the network to train over the CNN and test the data on the model to get the accuracy.

3. Program Outline

In this project we make use of python’s Keras Deep learning library to make the CNN network. Since there are two datasets MNIST and 15-scene dataset we must build two separate CNN networks to build their respective models.



The above figure shows the steps of CNN for a single MNIST image.

The CNN model for the MNIST dataset consists of a convolution layer with parameters as filters = 32, kernel size=(3,3), padding = 'Same', input shape=28x28x1 stride=1 . An activation layer of ReLU is included in the convolution layer itself. This layer outputs a 26x26x32 convoluted image. The next layer is a polling layer , where we use a MaxPool :2x2 with stride =2 whose output is 13x13x32. The third layer is again a convolution layer where the number of filters is 64 . Then again, a second

MaxPool layer is added followed by a Fully connected layer made of Dense(1024), Dense(512), Dense(10) with activation function as SoftMax.

The CNN network for the 15-scene dataset is similar to the one built for the MNIST dataset where it has 1 convolution layer, MaxPool layer, convolution layer, MaxPool Layer and then a Fully connected layer of Dense(1024), Dense(512), Dense(15) with activation function as SoftMax.

4. Program Implementation

The program consists of 2 functions to implement CNN network using python's deep learning Keras library on the MNIST and 15 scene datasets for classification.

- trainMNISTData (x_train, y_train, x_test, y_test)
- 15SceneCNN(X_train, Y_train, X_test, Y_test)

4.1 trainMNISTData (x_train, y_train, x_test, y_test)

This function takes in the train and testing values and builds the CNN network for the MNIST dataset . It then complies the training data to build a model. Then uses this model to test the testing data and print the accuracy.

4.2 15SceneCNN(X_train, Y_train, X_test, Y_test)

This function takes in the train and testing values and builds the CNN network for the 15-Scene dataset . It then complies the training data to build a model. Then uses this model to test the testing data and print the accuracy.

5 Testing

We train 15-scen dataset by adding three convolution layers (filters=32,64,86) and 3 max pooling layers to the CNN network. We notice that as more layers are added that is the deeper the network gets the higher the accuracy (54%) but the takes more computation time.

```
Epoch 1/100
50/49 [==============================] - 10s 197ms/step - loss: 3.1192 - acc: 0.1516
Epoch 2/100
50/49 [==============================] - 11s 225ms/step - loss: 2.4034 - acc: 0.2283
Epoch 3/100
50/49 [==============================] - 11s 225ms/step - loss: 2.1864 - acc: 0.2701
Epoch 4/100
50/49 [==============================] - 12s 235ms/step - loss: 2.0009 - acc: 0.3564
Epoch 5/100
50/49 [==============================] - 12s 238ms/step - loss: 1.9910 - acc: 0.3385
Epoch 6/100
50/49 [==============================] - 11s 227ms/step - loss: 1.8897 - acc: 0.3758
Epoch 7/100
50/49 [==============================] - 12s 241ms/step - loss: 1.8569 - acc: 0.3879
Epoch 8/100
50/49 [==============================] - 11s 220ms/step - loss: 1.8201 - acc: 0.3987
Epoch 9/100
50/49 [==============================] - 11s 228ms/step - loss: 1.7460 - acc: 0.4236
Epoch 10/100
50/49 [==============================] - 12s 243ms/step - loss: 1.6947 - acc: 0.4274
Epoch 11/100
50/49 [==============================] - 11s 228ms/step - loss: 1.6654 - acc: 0.4417
Epoch 12/100
50/49 [==============================] - 11s 224ms/step - loss: 1.6012 - acc: 0.4608
Epoch 13/100
```

```
50/49 [==============================] - 12s 236ms/step - loss: 1.6298 - acc: 0.4701
Epoch 14/100
50/49 [==============================] - 11s 219ms/step - loss: 1.5578 - acc: 0.4850
Epoch 15/100
50/49 [==============================] - 11s 219ms/step - loss: 1.5066 - acc: 0.4841
Epoch 16/100
50/49 [==============================] - 11s 220ms/step - loss: 1.4747 - acc: 0.5232
Epoch 17/100
50/49 [==============================] - 11s 227ms/step - loss: 1.5355 - acc: 0.4898
Epoch 18/100
50/49 [==============================] - 11s 227ms/step - loss: 1.4406 - acc: 0.5146
Epoch 19/100
50/49 [==============================] - 11s 229ms/step - loss: 1.4235 - acc: 0.5223
Epoch 20/100
50/49 [==============================] - 11s 224ms/step - loss: 1.3596 - acc: 0.5532
Epoch 21/100
50/49 [==============================] - 11s 225ms/step - loss: 1.3893 - acc: 0.5455
Epoch 22/100
50/49 [==============================] - 11s 222ms/step - loss: 1.3677 - acc: 0.5596
Epoch 23/100
50/49 [==============================] - 11s 225ms/step - loss: 1.2931 - acc: 0.5646
Epoch 24/100
50/49 [==============================] - 11s 229ms/step - loss: 1.2780 - acc: 0.5790
Epoch 25/100
50/49 [==============================] - 11s 226ms/step - loss: 1.3539 - acc: 0.5589
Epoch 26/100
50/49 [==============================] - 11s 227ms/step - loss: 1.2959 - acc: 0.5764
Epoch 27/100
50/49 [==============================] - 11s 225ms/step - loss: 1.2474 - acc: 0.5901
Epoch 28/100
50/49 [==============================] - 11s 226ms/step - loss: 1.2754 - acc: 0.5949
Epoch 29/100
50/49 [==============================] - 11s 228ms/step - loss: 1.1945 - acc: 0.6013
Epoch 30/100
50/49 [==============================] - 11s 227ms/step - loss: 1.2345 - acc: 0.5997
Epoch 31/100
50/49 [==============================] - 11s 228ms/step - loss: 1.1733 - acc: 0.6080
Epoch 32/100
50/49 [==============================] - 11s 223ms/step - loss: 1.1364 - acc: 0.6137
Epoch 33/100
50/49 [==============================] - 11s 225ms/step - loss: 1.1050 - acc: 0.6306
Epoch 34/100
50/49 [==============================] - 11s 226ms/step - loss: 1.1171 - acc: 0.6299
Epoch 35/100
50/49 [==============================] - 11s 225ms/step - loss: 1.1538 - acc: 0.6185
Epoch 36/100
50/49 [==============================] - 11s 227ms/step - loss: 1.1219 - acc: 0.6280
Epoch 37/100
50/49 [==============================] - 11s 228ms/step - loss: 1.1402 - acc: 0.6178
Epoch 38/100
50/49 [==============================] - 11s 225ms/step - loss: 1.1192 - acc: 0.6245
Epoch 39/100
50/49 [==============================] - 12s 247ms/step - loss: 1.1089 - acc: 0.6287
Epoch 40/100
50/49 [==============================] - 11s 229ms/step - loss: 1.0702 - acc: 0.6481
Epoch 41/100
50/49 [==============================] - 12s 231ms/step - loss: 1.0213 - acc: 0.6516
Epoch 42/100
50/49 [==============================] - 12s 235ms/step - loss: 1.0564 - acc: 0.6433
```

```
Epoch 43/100
50/49 [==============================] - 11s 227ms/step - loss: 1.0199 - acc: 0.6599
Epoch 44/100
50/49 [==============================] - 12s 249ms/step - loss: 1.1175 - acc: 0.6621
Epoch 45/100
50/49 [==============================] - 14s 271ms/step - loss: 1.0695 - acc: 0.6452
Epoch 46/100
50/49 [==============================] - 13s 269ms/step - loss: 1.0391 - acc: 0.6618
Epoch 47/100
50/49 [==============================] - 12s 248ms/step - loss: 0.9939 - acc: 0.6580
Epoch 48/100
50/49 [==============================] - 13s 265ms/step - loss: 0.9651 - acc: 0.6755
Epoch 49/100
50/49 [==============================] - 13s 250ms/step - loss: 1.0130 - acc: 0.6704
Epoch 50/100
50/49 [==============================] - 12s 241ms/step - loss: 0.9677 - acc: 0.6752
Epoch 51/100
50/49 [==============================] - 12s 232ms/step - loss: 1.0213 - acc: 0.6611
Epoch 52/100
50/49 [==============================] - 12s 237ms/step - loss: 0.9658 - acc: 0.6729
Epoch 53/100
50/49 [==============================] - 11s 225ms/step - loss: 0.9918 - acc: 0.6605
Epoch 54/100
50/49 [==============================] - 11s 225ms/step - loss: 0.9779 - acc: 0.6564
Epoch 55/100
50/49 [==============================] - 11s 225ms/step - loss: 0.9205 - acc: 0.6739
Epoch 56/100
50/49 [==============================] - 11s 227ms/step - loss: 0.9334 - acc: 0.6879
Epoch 57/100
50/49 [==============================] - 11s 227ms/step - loss: 0.9433 - acc: 0.6662
Epoch 58/100
50/49 [==============================] - 11s 227ms/step - loss: 0.9210 - acc: 0.6892
Epoch 59/100
50/49 [==============================] - 12s 232ms/step - loss: 0.8978 - acc: 0.6927
Epoch 60/100
50/49 [==============================] - 12s 233ms/step - loss: 0.9035 - acc: 0.6930
Epoch 61/100
50/49 [==============================] - 11s 226ms/step - loss: 0.8795 - acc: 0.7064
Epoch 62/100
50/49 [==============================] - 12s 240ms/step - loss: 0.8766 - acc: 0.6987
Epoch 63/100
50/49 [==============================] - 12s 238ms/step - loss: 0.8911 - acc: 0.7025
Epoch 64/100
50/49 [==============================] - 12s 244ms/step - loss: 0.9276 - acc: 0.6911
Epoch 65/100
50/49 [==============================] - 11s 228ms/step - loss: 0.8772 - acc: 0.7054
Epoch 66/100
50/49 [==============================] - 11s 227ms/step - loss: 0.9211 - acc: 0.6997
Epoch 67/100
50/49 [==============================] - 11s 222ms/step - loss: 0.8286 - acc: 0.7140
Epoch 68/100
50/49 [==============================] - 11s 221ms/step - loss: 0.8483 - acc: 0.7204
Epoch 69/100
50/49 [==============================] - 11s 220ms/step - loss: 0.8325 - acc: 0.7271
Epoch 70/100
50/49 [==============================] - 11s 228ms/step - loss: 0.9001 - acc: 0.7010
Epoch 71/100
50/49 [==============================] - 11s 226ms/step - loss: 0.8544 - acc: 0.7137
Epoch 72/100
```

```
50/49 [==============================] - 11s 221ms/step - loss: 0.9054 - acc: 0.7013
Epoch 73/100
50/49 [==============================] - 11s 220ms/step - loss: 0.8618 - acc: 0.7258
Epoch 74/100
50/49 [==============================] - 12s 235ms/step - loss: 0.8665 - acc: 0.7115
Epoch 75/100
50/49 [==============================] - 13s 257ms/step - loss: 0.8348 - acc: 0.7169
Epoch 76/100
50/49 [==============================] - 15s 298ms/step - loss: 0.8025 - acc: 0.7296
Epoch 77/100
50/49 [==============================] - 12s 246ms/step - loss: 0.7917 - acc: 0.7382
Epoch 78/100
50/49 [==============================] - 11s 226ms/step - loss: 0.7980 - acc: 0.7357
Epoch 79/100
50/49 [==============================] - 11s 229ms/step - loss: 0.7853 - acc: 0.7277
Epoch 80/100
50/49 [==============================] - 12s 235ms/step - loss: 0.7553 - acc: 0.7424
Epoch 81/100
50/49 [==============================] - 12s 231ms/step - loss: 0.7635 - acc: 0.7414
Epoch 82/100
50/49 [==============================] - 12s 234ms/step - loss: 0.7674 - acc: 0.7414
Epoch 83/100
50/49 [==============================] - 11s 228ms/step - loss: 0.7982 - acc: 0.7382
Epoch 84/100
50/49 [==============================] - 12s 233ms/step - loss: 0.7891 - acc: 0.7376
Epoch 85/100
50/49 [==============================] - 12s 232ms/step - loss: 0.7674 - acc: 0.7455
Epoch 86/100
50/49 [==============================] - 12s 232ms/step - loss: 0.7837 - acc: 0.7363
Epoch 87/100
50/49 [==============================] - 12s 232ms/step - loss: 0.7413 - acc: 0.7366
Epoch 88/100
50/49 [==============================] - 11s 227ms/step - loss: 0.7273 - acc: 0.7586
Epoch 89/100
50/49 [==============================] - 11s 227ms/step - loss: 0.7421 - acc: 0.7500
Epoch 90/100
50/49 [==============================] - 12s 230ms/step - loss: 0.7486 - acc: 0.7500
Epoch 91/100
50/49 [==============================] - 11s 229ms/step - loss: 0.7161 - acc: 0.7513
Epoch 92/100
50/49 [==============================] - 11s 230ms/step - loss: 0.7370 - acc: 0.7564
Epoch 93/100
50/49 [==============================] - 11s 229ms/step - loss: 0.6945 - acc: 0.7688
Epoch 94/100
50/49 [==============================] - 12s 231ms/step - loss: 0.8049 - acc: 0.7420
Epoch 95/100
50/49 [==============================] - 11s 225ms/step - loss: 0.7567 - acc: 0.7551
Epoch 96/100
50/49 [==============================] - 11s 223ms/step - loss: 0.6973 - acc: 0.7659
Epoch 97/100
50/49 [==============================] - 11s 224ms/step - loss: 0.7149 - acc: 0.7589
Epoch 98/100
50/49 [==============================] - 11s 225ms/step - loss: 0.7172 - acc: 0.7685
Epoch 99/100
50/49 [==============================] - 11s 229ms/step - loss: 0.7432 - acc: 0.7525
Epoch 100/100
50/49 [==============================] - 12s 231ms/step - loss: 0.6910 - acc: 0.7631
1345/1345 [==============================] - 1s 1ms/step
Accuracy: 0.5412639379501343
```

6 Obtained Results

The following is the result when the CNN network is used to train and model **the MNIST Dataset** with 2 alternate convolution and pooling layers with filters =32,64 with epoch =100.

```
Epoch 1/10
60000/60000 [==============================] - 135s 2ms/step - loss: 0.1176 - accuracy: 0.9637
Epoch 2/10
60000/60000 [==============================] - 135s 2ms/step - loss: 0.0475 - accuracy: 0.9859
Epoch 3/10
60000/60000 [==============================] - 124s 2ms/step - loss: 0.0359 - accuracy: 0.9895
Epoch 4/10
60000/60000 [==============================] - 129s 2ms/step - loss: 0.0291 - accuracy: 0.9912
Epoch 5/10
60000/60000 [==============================] - 150s 3ms/step - loss: 0.0226 - accuracy: 0.9932
Epoch 6/10
60000/60000 [==============================] - 140s 2ms/step - loss: 0.0211 - accuracy: 0.9938
Epoch 7/10
60000/60000 [==============================] - 133s 2ms/step - loss: 0.0170 - accuracy: 0.9948
Epoch 8/10
60000/60000 [==============================] - 130s 2ms/step - loss: 0.0167 - accuracy: 0.9954
Epoch 9/10
60000/60000 [==============================] - 134s 2ms/step - loss: 0.0156 - accuracy: 0.9960
Epoch 10/10
60000/60000 [==============================] - 133s 2ms/step - loss: 0.0137 - accuracy: 0.9962
```

We see that the accuracy is 99%.

The following is the result when the CNN network is used to train and model the 15- scene Dataset with 2 alternate convolution and pooling layers with filters =32,64 with epoch =100.

We see that the accuracy is 48% .

```
Epoch 1/100
50/49 [==============================] - 6s 113ms/step - loss: 3.1121 - acc: 0.1691
Epoch 2/100
50/49 [==============================] - 5s 108ms/step - loss: 2.2854 - acc: 0.2682
Epoch 3/100
50/49 [==============================] - 5s 108ms/step - loss: 2.1240 - acc: 0.3226
Epoch 4/100
50/49 [==============================] - 6s 117ms/step - loss: 2.0144 - acc: 0.3602
Epoch 5/100
50/49 [==============================] - 6s 129ms/step - loss: 1.8793 - acc: 0.3809
Epoch 6/100
50/49 [==============================] - 7s 134ms/step - loss: 1.8633 - acc: 0.3863
Epoch 7/100
50/49 [==============================] - 7s 135ms/step - loss: 1.8462 - acc: 0.4083
Epoch 8/100
50/49 [==============================] - 8s 164ms/step - loss: 1.7414 - acc: 0.4417
Epoch 9/100
50/49 [==============================] - 7s 140ms/step - loss: 1.6778 - acc: 0.4443
Epoch 10/100
50/49 [==============================] - 7s 138ms/step - loss: 1.6243 - acc: 0.4771
Epoch 11/100
```

```
50/49 [==============================] - 7s 134ms/step - loss: 1.6179 - acc: 0.4576
Epoch 12/100
50/49 [==============================] - 7s 136ms/step - loss: 1.6154 - acc: 0.4688
Epoch 13/100
50/49 [==============================] - 7s 134ms/step - loss: 1.5600 - acc: 0.4933
Epoch 14/100
50/49 [==============================] - 7s 133ms/step - loss: 1.5362 - acc: 0.4981
Epoch 15/100
50/49 [==============================] - 7s 134ms/step - loss: 1.4668 - acc: 0.5118
Epoch 16/100
50/49 [==============================] - 6s 129ms/step - loss: 1.4918 - acc: 0.5150
Epoch 17/100
50/49 [==============================] - 6s 127ms/step - loss: 1.4398 - acc: 0.5287
Epoch 18/100
50/49 [==============================] - 6s 128ms/step - loss: 1.3786 - acc: 0.5557
Epoch 19/100
50/49 [==============================] - 7s 131ms/step - loss: 1.4497 - acc: 0.5382
Epoch 20/100
50/49 [==============================] - 6s 128ms/step - loss: 1.4406 - acc: 0.5287
Epoch 21/100
50/49 [==============================] - 6s 128ms/step - loss: 1.4133 - acc: 0.5299
Epoch 22/100
50/49 [==============================] - 6s 127ms/step - loss: 1.4097 - acc: 0.5357
Epoch 23/100
50/49 [==============================] - 7s 135ms/step - loss: 1.3469 - acc: 0.5525
Epoch 24/100
50/49 [==============================] - 6s 129ms/step - loss: 1.3322 - acc: 0.5564
Epoch 25/100
50/49 [==============================] - 6s 127ms/step - loss: 1.3705 - acc: 0.5392
Epoch 26/100
50/49 [==============================] - 7s 133ms/step - loss: 1.3448 - acc: 0.5529
Epoch 27/100
50/49 [==============================] - 7s 133ms/step - loss: 1.2725 - acc: 0.5720
Epoch 28/100
50/49 [==============================] - 7s 133ms/step - loss: 1.3245 - acc: 0.5650
Epoch 29/100
50/49 [==============================] - 7s 140ms/step - loss: 1.2935 - acc: 0.5672
Epoch 30/100
50/49 [==============================] - 7s 141ms/step - loss: 1.2860 - acc: 0.5732
Epoch 31/100
50/49 [==============================] - 7s 142ms/step - loss: 1.2306 - acc: 0.5876
Epoch 32/100
50/49 [==============================] - 7s 140ms/step - loss: 1.2734 - acc: 0.5895
Epoch 33/100
50/49 [==============================] - 7s 136ms/step - loss: 1.3287 - acc: 0.5806
Epoch 34/100
50/49 [==============================] - 7s 136ms/step - loss: 1.2388 - acc: 0.5882
Epoch 35/100
50/49 [==============================] - 7s 135ms/step - loss: 1.2207 - acc: 0.5873
Epoch 36/100
50/49 [==============================] - 6s 130ms/step - loss: 1.2866 - acc: 0.5713
Epoch 37/100
50/49 [==============================] - 6s 128ms/step - loss: 1.2668 - acc: 0.6019
Epoch 38/100
50/49 [==============================] - 6s 129ms/step - loss: 1.2123 - acc: 0.5997
Epoch 39/100
50/49 [==============================] - 7s 130ms/step - loss: 1.2230 - acc: 0.6057
Epoch 40/100
50/49 [==============================] - 6s 127ms/step - loss: 1.2069 - acc: 0.6073
```

```
Epoch 41/100
50/49 [==============================] - 6s 127ms/step - loss: 1.1645 - acc: 0.6000
Epoch 42/100
50/49 [==============================] - 6s 126ms/step - loss: 1.1999 - acc: 0.6076
Epoch 43/100
50/49 [==============================] - 6s 127ms/step - loss: 1.4049 - acc: 0.5602
Epoch 44/100
50/49 [==============================] - 7s 131ms/step - loss: 1.2436 - acc: 0.5892
Epoch 45/100
50/49 [==============================] - 6s 130ms/step - loss: 1.1484 - acc: 0.6162
Epoch 46/100
50/49 [==============================] - 7s 131ms/step - loss: 1.1653 - acc: 0.6080
Epoch 47/100
50/49 [==============================] - 7s 136ms/step - loss: 1.1386 - acc: 0.6092 0s - loss: 1.13
36 - acc: 0.61
Epoch 48/100
50/49 [==============================] - 7s 150ms/step - loss: 1.1428 - acc: 0.6232
Epoch 49/100
50/49 [==============================] - 7s 133ms/step - loss: 1.1837 - acc: 0.6086
Epoch 50/100
50/49 [==============================] - 7s 140ms/step - loss: 1.0913 - acc: 0.6373
Epoch 51/100
50/49 [==============================] - 7s 133ms/step - loss: 1.1675 - acc: 0.6118
Epoch 52/100
50/49 [==============================] - 7s 131ms/step - loss: 1.0824 - acc: 0.6341
Epoch 53/100
50/49 [==============================] - 7s 131ms/step - loss: 1.1546 - acc: 0.6140
Epoch 54/100
50/49 [==============================] - 7s 132ms/step - loss: 1.1608 - acc: 0.6194
Epoch 55/100
50/49 [==============================] - 7s 132ms/step - loss: 1.0796 - acc: 0.6462
Epoch 56/100
50/49 [==============================] - 7s 132ms/step - loss: 1.0768 - acc: 0.6455
Epoch 57/100
50/49 [==============================] - 7s 131ms/step - loss: 1.1076 - acc: 0.6315
Epoch 58/100
50/49 [==============================] - 7s 136ms/step - loss: 1.0615 - acc: 0.6596
Epoch 59/100
50/49 [==============================] - 7s 137ms/step - loss: 1.1241 - acc: 0.6395
Epoch 60/100
50/49 [==============================] - 7s 136ms/step - loss: 1.0868 - acc: 0.6401
Epoch 61/100
50/49 [==============================] - 7s 132ms/step - loss: 1.0145 - acc: 0.6541
Epoch 62/100
50/49 [==============================] - 7s 133ms/step - loss: 0.9878 - acc: 0.6605
Epoch 63/100
50/49 [==============================] - 7s 135ms/step - loss: 1.0363 - acc: 0.6570
Epoch 64/100
50/49 [==============================] - 7s 135ms/step - loss: 1.1259 - acc: 0.6338
Epoch 65/100
50/49 [==============================] - 7s 138ms/step - loss: 1.0197 - acc: 0.6490
Epoch 66/100
50/49 [==============================] - 7s 137ms/step - loss: 1.0435 - acc: 0.6580
Epoch 67/100
50/49 [==============================] - 7s 132ms/step - loss: 0.9736 - acc: 0.6701
Epoch 68/100
50/49 [==============================] - 7s 137ms/step - loss: 1.0532 - acc: 0.6490
Epoch 69/100
50/49 [==============================] - 7s 132ms/step - loss: 1.0336 - acc: 0.6592
```

```
Epoch 70/100
50/49 [==============================] - 6s 128ms/step - loss: 0.9494 - acc: 0.6745
Epoch 71/100
50/49 [==============================] - 6s 129ms/step - loss: 1.0099 - acc: 0.6729
Epoch 72/100
50/49 [==============================] - 7s 136ms/step - loss: 0.9601 - acc: 0.6739
Epoch 73/100
50/49 [==============================] - 7s 136ms/step - loss: 0.9879 - acc: 0.6777
Epoch 74/100
50/49 [==============================] - 7s 133ms/step - loss: 0.9771 - acc: 0.6640
Epoch 75/100
50/49 [==============================] - 7s 135ms/step - loss: 0.9913 - acc: 0.6799
Epoch 76/100
50/49 [==============================] - 7s 135ms/step - loss: 0.9736 - acc: 0.6717
Epoch 77/100
50/49 [==============================] - 7s 133ms/step - loss: 0.9367 - acc: 0.6793
Epoch 78/100
50/49 [==============================] - 8s 150ms/step - loss: 0.9586 - acc: 0.6799
Epoch 79/100
50/49 [==============================] - 7s 146ms/step - loss: 1.0392 - acc: 0.6627
Epoch 80/100
50/49 [==============================] - 7s 137ms/step - loss: 0.9367 - acc: 0.6828
Epoch 81/100
50/49 [==============================] - 7s 138ms/step - loss: 0.9777 - acc: 0.6780
Epoch 82/100
50/49 [==============================] - 7s 134ms/step - loss: 0.9894 - acc: 0.6758
Epoch 83/100
50/49 [==============================] - 7s 142ms/step - loss: 0.8991 - acc: 0.7067
Epoch 84/100
50/49 [==============================] - 7s 135ms/step - loss: 0.9568 - acc: 0.6748
Epoch 85/100
50/49 [==============================] - 6s 129ms/step - loss: 0.9404 - acc: 0.6911
Epoch 86/100
50/49 [==============================] - 7s 131ms/step - loss: 0.9088 - acc: 0.6869
Epoch 87/100
50/49 [==============================] - 7s 132ms/step - loss: 0.9320 - acc: 0.6946
Epoch 88/100
50/49 [==============================] - 7s 135ms/step - loss: 0.9228 - acc: 0.7025
Epoch 89/100
50/49 [==============================] - 7s 134ms/step - loss: 1.0287 - acc: 0.6752
Epoch 90/100
50/49 [==============================] - 7s 135ms/step - loss: 0.9145 - acc: 0.6968
Epoch 91/100
50/49 [==============================] - 7s 134ms/step - loss: 0.9306 - acc: 0.7003
Epoch 92/100
50/49 [==============================] - 7s 132ms/step - loss: 0.9524 - acc: 0.6914
Epoch 93/100
50/49 [==============================] - 7s 132ms/step - loss: 0.9236 - acc: 0.7016
Epoch 94/100
50/49 [==============================] - 7s 131ms/step - loss: 0.8705 - acc: 0.7051
Epoch 95/100
50/49 [==============================] - 7s 136ms/step - loss: 0.8615 - acc: 0.7029
Epoch 96/100
50/49 [==============================] - 7s 131ms/step - loss: 0.8466 - acc: 0.7131
Epoch 97/100
50/49 [==============================] - 7s 133ms/step - loss: 0.9094 - acc: 0.7016
Epoch 98/100
50/49 [==============================] - 7s 141ms/step - loss: 0.8691 - acc: 0.7159
Epoch 99/100
```

```
50/49 [==============================] - 7s 138ms/step - loss: 0.8550 - acc: 0.7086
Epoch 100/100
50/49 [==============================] - 7s 143ms/step - loss: 0.8252 - acc: 0.7175
1345/1345 [==============================] - 1s 624us/step

Accuracy: 0.4877323508262634
```

7 Conclusion

- We notice that the MNIST dataset get an accuracy of 98% when trained on the CNN network which is higher than what we get when trained on SLP or MLP neural network.
- The more layers there are in the CNN network the higher the accuracy.
- A smaller filter for example filter= 3x3 will use fewer parameters and require fewer computations, and it will usually perform better when compared to a filter=5x5
- Unlike a regular neural network , once a CNN has learned to recognize a pattern in one location, it can recognize it in any other location.
- As the CNN has partially connected layers and weight sharing it can work for large datasets too and not just small datasets like MNIST.

8. References

- Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (pp. 456-457). O'Reilly Media. Kindle Edition
- Class Notes