# ECE 657A Assignment 3

**Submitted By :**
**Amandeep Kaur (21044104)**
**Bhupesh Dod (21046099)**

## Question 1 : Default Network

**Solution :**

To run the CNNs, we used Google Colab and Keras Library. We used Google Colab GPU to run the CNNs for faster computations. The steps followed to create and implement the default CNN are as follows:

1. Load all the necessary libraries.
2. Import the Datasets.
3. Data pre-processing. : Reshape the data to 28*28
4. Create compile and summarize the CNN network.
5. Fit the CNN network.
6. Calculate Accuracy and Loss.

The network contains:

a) CNN layer with filters = 32, kernels = 3*3, strides = 1 and padding = 'same'
b) MaxPool layer with pool size = (2,2)
c) CNN layer with filters = 32, kernels = 3*3, strides = 1 and padding = 'same'
d) Flatten layer
e) Fully connected layer
f) Fully connected output layer with 5 outputs.

All the hidden layers used the activation function 'Relu' while the output layer used 'Softmax' activation function as it normalizes the output to a probability distribution over output clusters. We used Stochastic Gradient Descent as the optimizer.The structure of the CNN is shown in Fig.1

Fig.1 Structure of default CNN

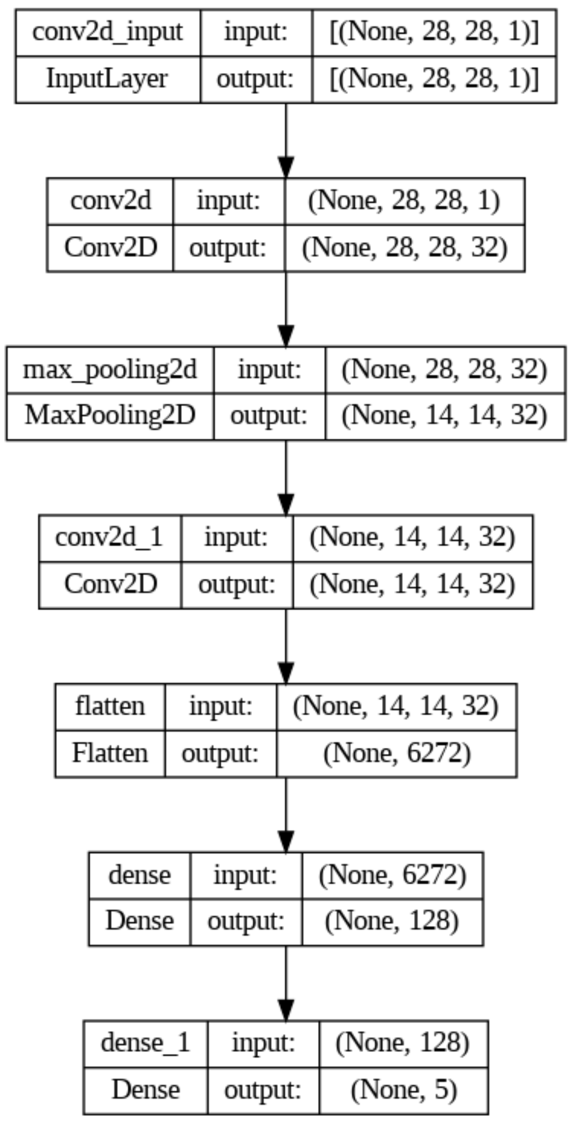
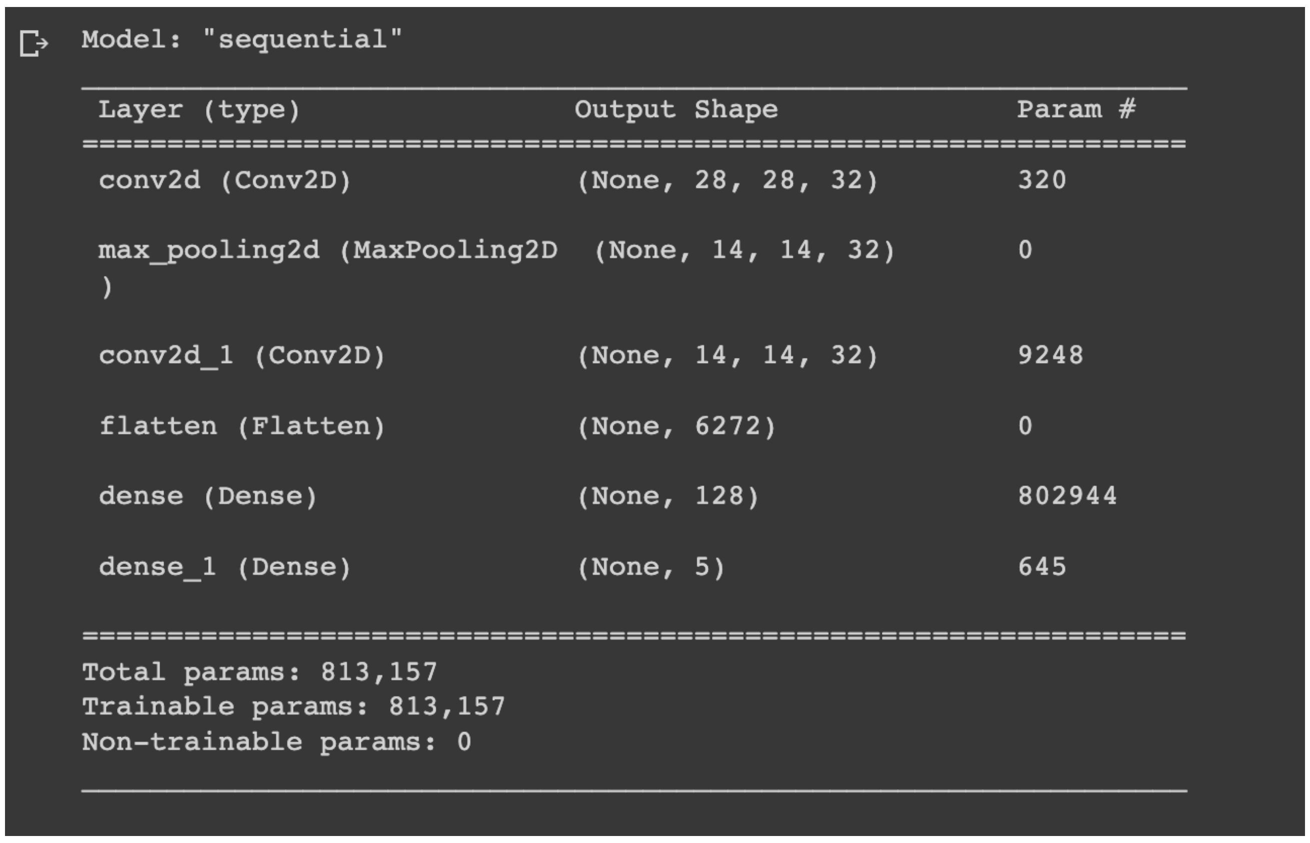

conv2d_input | input: | [(None, 28, 28, 1)]
InputLayer | output: | [(None, 28, 28, 1)]
conv2d | input: | (None, 28, 28, 1)
Conv2D | output: | (None, 28, 28, 32)
max_pooling2d | input: | (None, 28, 28, 32)
MaxPooling2D | output: | (None, 14, 14, 32)
conv2d_1 | input: | (None, 14, 14, 32)
Conv2D | output: | (None, 14, 14, 32)
flatten | input: | (None, 14, 14, 32)
Flatten | output: | (None, 6272)
dense | input: | (None, 6272)
Dense | output: | (None, 128)
dense_1 | input: | (None, 128)
Dense | output: | (None, 5)

Fig. 2 Model Summary of default CNN



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 32)        320

 max_pooling2d (MaxPooling2D  (None, 14, 14, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 14, 14, 32)        9248

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 128)               802944

 dense_1 (Dense)             (None, 5)                 645

=================================================================
Total params: 813,157
Trainable params: 813,157
Non-trainable params: 0
_________________________________________________________________
```

### Summary of the model

- We reshaped the data to 28*28 pixels and input it to the CNN layer 1, this layer has 32 filters with 3*3 kernel, stride of 1 and padding as 1*1. This layer has converted the data to a shape of 28*28 with 32 activation maps. This layer uses RelU as the activation function. Output is (28,28,32).
- Next hidden layer is the Maxpool layer with pool_size = 2,2. This layer has converted the input to (14,14,32).
- Next hidden layer is the second CNN layer with 32 filters, 3*3 kernel size, stride of 1 and padding as 1*1 with RelU as activation function The output of this layer is (14,14,32).
- Next layer flattens the output into one column of 6272 rows.
- Add a fully connected layer with 128 neuron's.
- For the final fully connected output layer with 5 outputs, we used the 'softmax' activation function.

- Parameters are weights that are learnt during training. For our default model, there are 813,157 trainable parameters.

## Runtime Information

For training the CNN model we used EarlyStopping with monitor variable as validation loss and patience = 2, this helps in stopping the training of the model when the change in monitor variable is not significant. The accuracy of the model is 90.429% and the loss is 26.95 %. The time lapsed in training the mode with 100 epochs and batch size of 128 is 49.79 seconds. The time taken in testing the model is 1.24 seconds.

## Mystery Models

After visualizing the x_train and x_test images with respect to y_train and y_test values. The mystery labels for the dataset could be defined as follows:

| Label | Fashion Item |
|---|---|
| 0 | T-shirt/Top and Shirt |
| 1 | Sandal |
| 2 | Pullover and Trouser |
| 3 | Sneaker and Bag |
| 4 | Dress, Ankle Boot and Coat |