Assignment 5 - Convolutional Neural Network

Author: Rudraksh Kapil - 177154

In this notebook, I implement a convolutional neural network for classifying images using Keras. We carry out two tasks:

1. Handwritten Digit Classification
2. Handwritten Letter Classification

In [20]:
```
# keras import statements
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.utils import np_utils

# other imports
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn import model_selection
```

1. Handwritten digit recognition

Get digits from MNIST Dataset

We also show some images for visualisation. The data is loaded a smaller version of the mnist dataset becuase the actual files are too large to download and reupload during submission.

In [21]:
```
# Loading data from CSV file
X = pd.read_csv('mnist.csv')
y = X['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33
, random_state=42)
```

Preprocessing and Visualizing the Training Data

In [22]:
```
# Storing the labels in a separate vector y_train
y_train = np.array(X_train['label'])

# Dropping the labels column
X_train = X_train.drop(['label'], axis=1)

# Normalizing the values of the pixels
X_train = X_train / 255

# Converting X_train dataframe to numpy array
X_train = np.array(X_train)

print("-------------------------")
print("SAMPLE OF TRAINING DATA")
print("-------------------------")

# Plotting the data
plt.figure(figsize=(8,8))
for i in range(16):
    plt.subplot(4,4,i+1)
    plt.axis('off')
    r = np.random.randint(X_train.shape[0])   # Get a random image to sh
ow
    plt.title('True Label: '+str(y_train[r])) # Show its label as title
    plt.imshow(X_train[r].reshape(28,28), cmap='magma')    # Plotting th
e image
plt.show()

# Converting the labels vector into one hot encoded form
oh = np.zeros((y_train.size, y_train.max()+1))
oh[np.arange(y_train.size), y_train] = 1
y_train = np.array(oh)

# Reshaping X_train to the format of the network
X_train = X_train.reshape(X_train.shape[0], 28, 28, 1).astype('float32')
```

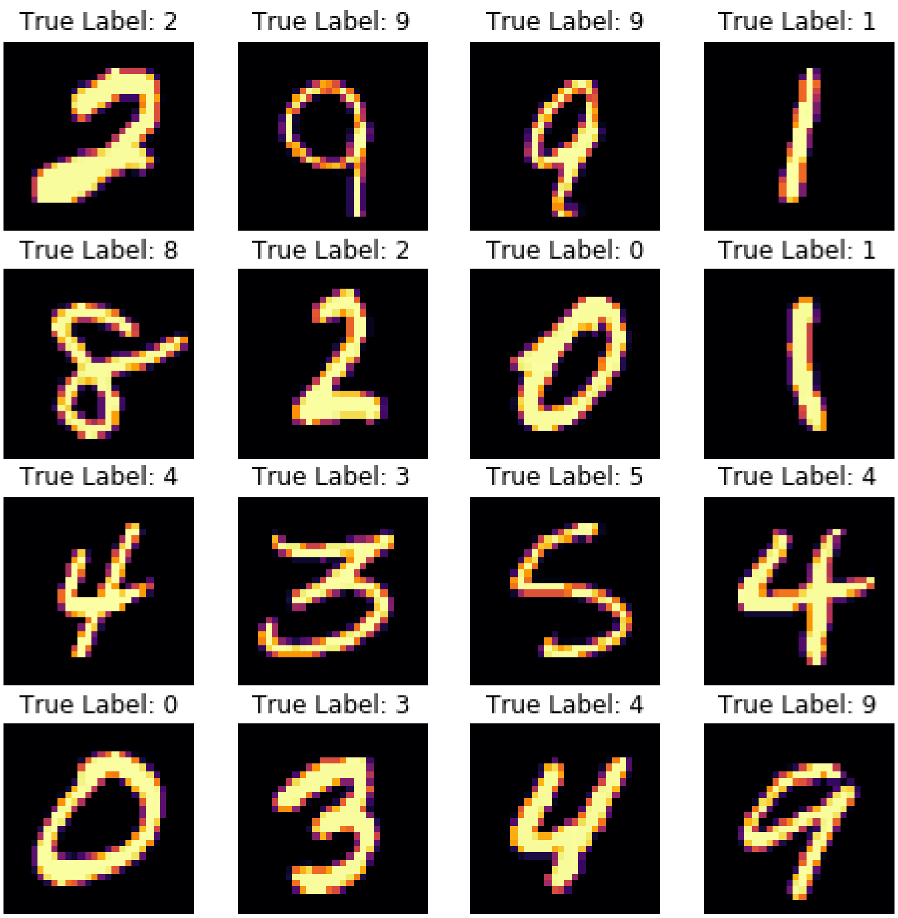
```
-------------------------
SAMPLE OF TRAINING DATA
-------------------------
```

Preprocessing and Visualizing the Testing Data

In [23]:

```
# Storing the labels in a separate vector y_train
y_test = np.array(X_test['label'])

# Dropping the labels column
X_test = X_test.drop(['label'], axis=1)

# Normalizing the values of the pixels
X_test = X_test/ 255

# Converting X_train dataframe to numpy array
X_test = np.array(X_test)

print("-------------------------")
print("SAMPLE OF TESTING DATA")
print("-------------------------")

# Plotting the data
plt.figure(figsize=(8,8))
for i in range(16):
    plt.subplot(4,4,i+1)
    plt.axis('off')
    r = np.random.randint(X_test.shape[0])   # Get a random image to sho
w
    plt.title('True Label: '+str(y_test[r])) # Show its label as title
    plt.imshow(X_test[r].reshape(28,28), cmap='inferno')    # Plotting t
he image
plt.show()

# Converting the labels vector into one hot encoded form
oh = np.zeros((y_test.size, y_test.max()+1))
oh[np.arange(y_test.size), y_test] = 1
y_test = np.array(oh)

# Reshaping X_train to the format of the network
X_test = X_test.reshape(X_test.shape[0], 28, 28, 1).astype('float32')

# number of classes
no_of_label_classes = y_test.shape[1]
```

```
-------------------------
SAMPLE OF TESTING DATA
-------------------------
```



Designing the Convolutional Neural Network

For the architecture, I use a simple convolutional model due to hardware constraints, but it gives good performance anyway. The model has the following architecture:

- 2D convolutional layer of size 5x5 filters, stride 1 and 32 filters. This is followed by an activation function layer of ReLU.
- Max pooling layer of size 2x2
- Dropout for normalisation with factor = 0.2
- Affine (fully-connected) layer of 128 units, followed by a relu activation.
- Finally, another affine layer with the number of desired output units is used with a softmax activation, since it is a classification task.

I use the adam optimiser and categorical cross entropy loss in the model.

In [25]:
```
def create_model():
    '''
        Function to create a sequential keras model
    '''

    model = Sequential()
    model.add(Conv2D(32, (5, 5), input_shape=(28, 28, 1), activation='re
lu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.2))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dense(no_of_label_classes, activation='softmax'))

    # Compile model
    model.compile(loss='categorical_crossentropy', optimizer='adam', met
rics=['accuracy'])
    return model
```

In [26]:
```
# initialising hyperparameters
iterations = 20
b_size = 16
```

Training the Model

Here I train the model, and for each epoch I also print out the time taken, loss, and accuracy. Likewise, after training, I print out the model summary and all of the weights of the neural network, as required for the assignment.

In [27]:
```
# Creating the model
model = create_model()
# Fitting the model
history = model.fit(X_train, y_train,epochs=iterations, batch_size=b_siz
e, verbose=2)

model.summary()
for layer in model.layers: print(layer.get_config(), layer.get_weights
())
```

```
Epoch 1/20
1759/1759 - 12s - loss: 0.1706 - accuracy: 0.9493
Epoch 2/20
1759/1759 - 11s - loss: 0.0607 - accuracy: 0.9808
Epoch 3/20
1759/1759 - 15s - loss: 0.0434 - accuracy: 0.9853
Epoch 4/20
1759/1759 - 15s - loss: 0.0287 - accuracy: 0.9904
Epoch 5/20
1759/1759 - 15s - loss: 0.0200 - accuracy: 0.9937
Epoch 6/20
1759/1759 - 15s - loss: 0.0183 - accuracy: 0.9939
Epoch 7/20
1759/1759 - 16s - loss: 0.0143 - accuracy: 0.9951
Epoch 8/20
1759/1759 - 11s - loss: 0.0119 - accuracy: 0.9954
Epoch 9/20
1759/1759 - 10s - loss: 0.0106 - accuracy: 0.9960
Epoch 10/20
1759/1759 - 11s - loss: 0.0101 - accuracy: 0.9966
Epoch 11/20
1759/1759 - 11s - loss: 0.0069 - accuracy: 0.9976
Epoch 12/20
1759/1759 - 11s - loss: 0.0067 - accuracy: 0.9978
Epoch 13/20
1759/1759 - 10s - loss: 0.0061 - accuracy: 0.9979
Epoch 14/20
1759/1759 - 11s - loss: 0.0075 - accuracy: 0.9974
Epoch 15/20
1759/1759 - 11s - loss: 0.0071 - accuracy: 0.9978
Epoch 16/20
1759/1759 - 10s - loss: 0.0047 - accuracy: 0.9985
Epoch 17/20
1759/1759 - 10s - loss: 0.0050 - accuracy: 0.9982
Epoch 18/20
1759/1759 - 10s - loss: 0.0066 - accuracy: 0.9982
Epoch 19/20
1759/1759 - 11s - loss: 0.0039 - accuracy: 0.9988
Epoch 20/20
1759/1759 - 10s - loss: 0.0026 - accuracy: 0.9990
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_1 (Conv2D)            (None, 24, 24, 32)        832       
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 32)        0         
_________________________________________________________________
dropout_1 (Dropout)          (None, 12, 12, 32)        0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 4608)              0         
_________________________________________________________________
dense_2 (Dense)              (None, 128)               589952    
_________________________________________________________________
dense_3 (Dense)              (None, 10)                1290      
=================================================================
Total params: 592,074
```

```
Trainable params: 592,074
Non-trainable params: 0
_________________________________________________________________
```


```
{'name': 'conv2d_1', 'trainable': True, 'batch_input_shape': (None, 28,
28, 1), 'dtype': 'float32', 'filters': 32, 'kernel_size': (5, 5), 'stri
des': (1, 1), 'padding': 'valid', 'data_format': 'channels_last', 'dila
tion_rate': (1, 1), 'activation': 'relu', 'use_bias': True, 'kernel_ini
tializer': {'class_name': 'GlorotUniform', 'config': {'seed': None}},
'bias_initializer': {'class_name': 'Zeros', 'config': {}}, 'kernel_regu
larizer': None, 'bias_regularizer': None, 'activity_regularizer': None,
'kernel_constraint': None, 'bias_constraint': None} [array([[[[ 5.24529
628e-02, -4.64680605e-04, -7.53076136e-01,
           1.92914367e-01, -1.89401761e-01, -7.46479571e-01,
           7.18029216e-02, -6.87670052e-01, -2.13625859e-02,
           4.29382510e-02,  1.76176399e-01, -8.17698240e-02,
           1.27413467e-01,  2.73956750e-02, -6.05267920e-02,
          -9.59541798e-02, -2.99366206e-01,  7.77364522e-03,
          -5.98835014e-02, -2.43176475e-01,  2.41856892e-02,
          -1.03013381e-01,  5.02482653e-02,  1.87961608e-02,
           1.25783607e-01,  7.64282867e-02, -1.64243933e-02,
          -5.81158735e-02, -1.11878231e-01,  2.34076697e-02,
           1.81910753e-01,  1.32324025e-01]],

        [[ 2.19053656e-01, -2.92159915e-01, -3.99688601e-01,
           4.73521054e-02, -3.25724542e-01, -3.29612851e-01,
           7.95652904e-03, -2.43406311e-01, -1.21542020e-02,
           1.25900537e-01,  1.48201287e-01, -3.43313903e-01,
           2.30803251e-01, -4.28690538e-02,  1.14967898e-01,
          -1.63485527e-01, -1.92707002e-01,  4.96234000e-02,
           8.25004503e-02, -2.92814314e-01, -1.88132912e-01,
          -1.71064019e-01,  1.34135410e-01,  7.55613763e-03,
           3.27372551e-02,  1.78714126e-01,  1.37815088e-01,
           9.58437249e-02, -5.12790726e-03,  1.61051080e-01,
           1.38481542e-01,  1.37570677e-02]],

        [[ 2.61290725e-02, -6.32672787e-01, -2.06264004e-01,
          -8.11664090e-02, -1.36358291e-01, -2.33640790e-01,
           8.71480852e-02,  1.90226600e-01, -1.43144198e-03,
          -3.25497761e-02, -6.37161732e-02, -3.10079366e-01,
           8.12821463e-02, -3.68102491e-01,  1.43063799e-01,
          -5.22230268e-01, -1.36254150e-02,  1.34067714e-01,
           1.17477156e-01, -1.93398803e-01, -1.66874260e-01,
          -3.81522477e-01,  8.97943005e-02,  3.48141119e-02,
           2.97379214e-02,  1.43587157e-01,  5.89065403e-02,
          -9.08486992e-02, -2.33203739e-01,  1.89778805e-01,
           7.78683275e-03,  2.61445463e-01]],

        [[ 7.33065233e-02, -2.43413404e-01, -1.46827519e-01,
          -8.18649307e-02, -2.84508526e-01,  2.72230171e-02,
          -3.14654149e-02,  3.03701758e-02,  2.32274067e-02,
          -1.89303905e-01, -1.58120587e-01,  2.43930165e-02,
          -1.82449624e-01, -5.12744486e-01, -1.74075365e-02,
          -3.28608245e-01,  9.31163207e-02, -1.19657576e-01,
          -6.40039295e-02,  2.67323628e-02, -3.75347078e-01,
          -3.01454008e-01,  3.94611433e-02, -3.79993580e-02,
          -2.86901444e-01, -3.61492783e-01, -2.27368414e-01,
          -5.66061512e-02, -2.74994969e-01, -3.28800604e-02,
```

```
        -2.71305263e-01,  1.39494330e-01]],

       [[-1.99323874e-02,  6.01961873e-02,  1.55007213e-01,
         -1.11964950e-03, -4.08608198e-01,  1.67729035e-01,
          7.57615268e-02, -2.13802889e-01,  1.28425732e-01,
         -7.21780956e-01, -2.13342905e-01,  1.42504796e-01,
         -6.21294864e-02, -4.55804586e-01, -2.79873818e-01,
          2.91826725e-01,  1.68478444e-01,  8.93824399e-02,
         -2.42380828e-01,  1.55168653e-01, -2.94604301e-01,
         -4.96665865e-01,  2.96410650e-01,  6.20897673e-02,
         -3.25877130e-01, -3.61729503e-01,  1.90082356e-01,
          1.12088338e-01,  4.84054796e-02, -5.79910398e-01,
         -2.12292030e-01, -1.99456260e-01]]],


      [[[-1.77545827e-02,  4.66027111e-02, -6.48007030e-03,
          5.22891916e-02,  3.18517983e-01, -1.80407494e-01,
         -1.56093221e-02, -8.13968778e-02,  7.69163445e-02,
          1.15390733e-01, -1.23316258e-01, -2.79592991e-01,
          1.36221973e-02, -5.13517797e-01,  5.10258339e-02,
          1.31745666e-01, -4.90041822e-01, -3.08548391e-01,
          1.12547195e-02, -1.97823402e-02,  1.31267890e-01,
         -3.84194762e-01, -1.26351550e-01, -1.73478480e-02,
          9.72010717e-02, -6.42515272e-02,  1.55666713e-02,
         -5.42068303e-01,  1.94122866e-01, -5.02448201e-01,
          4.36228551e-02,  7.25007132e-02]],

       [[-8.76247138e-02, -5.99299550e-01,  6.25390839e-03,
         -3.29562664e-01,  1.77739978e-01, -5.97663708e-02,
          1.22882165e-01,  3.68787318e-01,  2.17720821e-01,
          2.21316889e-01,  1.90143734e-01, -2.06897318e-01,
          4.16268893e-02, -3.08875382e-01,  1.64838001e-01,
         -1.50805548e-01,  1.57474971e-03, -2.66256273e-01,
          5.32155000e-02, -6.83939755e-02,  3.52617294e-01,
         -8.69729966e-02,  2.02132285e-01,  3.19166332e-02,
          1.49922654e-01,  6.71647936e-02,  1.35607868e-01,
         -1.74723603e-02,  6.17253780e-02,  4.71998230e-02,
          7.66673833e-02, -2.06641823e-01]],

       [[-3.46928090e-02, -1.39086828e-01, -1.75021976e-01,
         -2.30287418e-01,  1.86814338e-01, -3.39910239e-01,
          1.58105701e-01,  1.54264480e-01,  1.65047199e-01,
         -1.34396300e-01,  2.71784961e-01, -1.95286214e-01,
         -4.58661914e-01, -1.18549652e-01,  9.02122483e-02,
         -2.55440980e-01,  1.96860954e-02, -4.74254489e-01,
          2.76809279e-02, -2.37270445e-01,  1.75428003e-01,
         -3.54959629e-02,  2.06330225e-01, -9.79258046e-02,
          1.49159282e-01,  1.92343891e-01, -9.93972570e-02,
          9.72034633e-02,  9.50040221e-02,  2.77297497e-01,
          1.23204812e-01, -7.44048133e-02]],

       [[-1.09919626e-02,  1.49039775e-01, -3.97890747e-01,
          9.42627788e-02, -1.70714885e-01, -6.48741186e-01,
          1.69799000e-01,  3.09766345e-02,  7.61868283e-02,
         -7.31019318e-01, -5.17105013e-02, -1.82830945e-01,
         -3.22536290e-01,  8.23383257e-02, -1.11803919e-01,
         -2.35869512e-01,  5.26864976e-02, -1.53416157e-01,
```

```
         8.40137899e-03, -5.97468913e-02,  2.90692687e-01,
        -6.82024583e-02,  1.79307908e-01, -4.57686782e-02,
        -9.62733105e-02, -1.42884357e-02, -3.55065972e-01,
        -5.93714276e-03,  1.64355248e-01,  2.36442864e-01,
        -2.76195198e-01, -2.74544686e-01]],

       [[ 6.96453359e-03,  9.71365720e-02, -3.90643239e-01,
          2.03552499e-01, -4.19326395e-01, -3.51971209e-01,
          1.32118091e-01, -2.51373559e-01,  2.09721755e-02,
         -2.81529218e-01, -2.39746779e-01,  1.95400655e-01,
          1.97938755e-01, -1.84195191e-02, -3.32314312e-01,
         -1.22321181e-01,  1.12828642e-01, -2.79615633e-02,
         -1.43292740e-01,  1.79844230e-01,  1.95187554e-01,
         -1.10984974e-01,  1.18284356e-02,  3.21458913e-02,
         -4.28043783e-01, -4.75837663e-02,  2.15203375e-01,
         -1.91792846e-03,  2.39805669e-01, -1.16470478e-01,
         -2.02858329e-01, -2.76864529e-01]]],


      [[[-1.73467577e-01, -1.44284949e-01,  1.86567903e-01,
         -2.26639941e-01,  1.62434191e-01,  1.22755200e-01,
         -3.42417687e-01,  3.78359556e-01,  4.88282181e-02,
          3.00298303e-01, -1.84499338e-01,  5.57540879e-02,
         -8.16413760e-02, -2.55915821e-01,  4.90209572e-02,
          2.26076916e-01, -3.78545411e-02, -3.88885289e-01,
         -8.13015085e-03, -1.18034910e-02, -8.23507085e-02,
         -4.19032753e-01, -4.41356808e-01,  4.65674289e-02,
         -1.15211613e-01, -3.23440507e-02,  1.74140930e-01,
         -3.42253685e-01,  1.80151910e-01, -5.89555562e-01,
          1.18205592e-01, -1.99819788e-01]],

       [[-1.96036696e-01, -5.00450969e-01,  2.99332291e-01,
         -3.54469270e-01,  1.50010020e-01,  9.88579541e-02,
          9.61227790e-02,  1.60647243e-01,  1.72812551e-01,
          2.66703695e-01, -5.95737547e-02,  9.04820859e-02,
         -1.14357024e-01, -1.92195140e-02,  1.77302554e-01,
          4.03489828e-01,  1.81679681e-01, -3.48292857e-01,
         -5.24929054e-02, -5.15026450e-02,  4.75090668e-02,
          7.07654208e-02, -1.25289157e-01,  1.87007952e-02,
          1.24714069e-01, -1.09128237e-01,  1.27508852e-03,
          3.54745179e-01,  6.53471798e-02,  1.03116140e-01,
          2.13004157e-01, -2.60271579e-01]],

       [[-1.70630023e-01,  2.17057198e-01,  1.94527730e-01,
         -1.49210263e-02, -2.16777697e-02,  1.32741287e-01,
         -1.47026591e-02, -1.90322444e-01,  5.40067144e-02,
         -5.69405675e-01,  3.82782817e-01, -3.36313069e-01,
         -7.88222849e-02,  2.33937398e-01,  1.09710187e-01,
          2.99743086e-01,  2.42268905e-01,  3.03320549e-02,
          2.09683571e-02, -2.97250450e-01,  1.59786105e-01,
          2.61677504e-01,  2.76999831e-01, -4.05944437e-02,
          3.35191190e-01,  1.46827638e-01, -3.49062294e-01,
          1.87019795e-01,  1.98669165e-01,  1.32648394e-01,
          1.98609587e-02, -7.47084618e-02]],

       [[ 6.79278001e-02,  2.26997286e-01,  1.14998206e-01,
          2.44851232e-01, -1.30637318e-01, -1.41886711e-01,
```

```
          1.14795513e-01, -1.41719401e-01,  9.92183313e-02,
         -4.20613676e-01,  2.55600125e-01, -4.49086815e-01,
          2.22709924e-01,  3.25732410e-01, -1.19739361e-01,
          3.30159292e-02,  2.07307354e-01, -9.06230584e-02,
         -2.59778891e-02, -4.89825383e-02,  1.36035681e-01,
          1.81165516e-01,  4.55668643e-02,  5.22742160e-02,
          1.49611086e-01,  2.70733207e-01,  6.31808192e-02,
         -2.55585480e-02,  1.33495599e-01,  2.65480101e-01,
         -3.83121818e-01,  1.29884049e-01]],

       [[-4.21172231e-02,  1.30746320e-01, -1.77662924e-01,
          1.53021380e-01, -6.07840084e-02, -1.83085397e-01,
         -1.11906091e-02,  5.14066778e-02, -2.47453034e-01,
          2.85738885e-01,  1.23087447e-02,  2.40717173e-01,
          2.73344606e-01,  2.94499099e-01, -3.97218406e-01,
         -1.40909314e-01, -8.24432895e-02,  2.42871389e-01,
         -5.12566715e-02,  1.61841542e-01, -5.01634143e-02,
          6.44142255e-02, -4.61912572e-01, -8.01951736e-02,
         -2.41899818e-01,  4.12087031e-02,  8.80600587e-02,
         -1.23194315e-01, -1.01327868e-02, -9.14993212e-02,
         -1.32787481e-01,  2.75084674e-01]]],


      [[[-4.34009880e-01, -3.83318722e-01,  4.45106812e-02,
         -1.67736337e-01, -4.13009405e-01,  1.33693799e-01,
         -7.18048215e-01,  5.37674613e-02, -3.54602158e-01,
          1.06296405e-01, -4.28610921e-01,  3.05253953e-01,
         -5.50434142e-02,  1.02736749e-01,  8.75505153e-03,
         -1.04285330e-02,  1.23591885e-01, -1.27581522e-01,
         -5.23906127e-02, -1.18739670e-02, -2.09899753e-01,
         -3.92425768e-02, -5.50727963e-01, -6.12670742e-02,
         -3.24600607e-01,  4.53347489e-02,  1.91792294e-01,
         -2.29378954e-01, -2.45945558e-01, -4.09027100e-01,
         -3.01140919e-02,  1.75705984e-01]],

       [[-2.36768976e-01, -2.59747535e-01,  1.50051355e-01,
          6.23806380e-02, -1.59040377e-01,  4.11780290e-02,
         -1.77585810e-01, -3.25141728e-01,  1.46637000e-02,
         -3.73687655e-01, -1.92152426e-01,  1.40459180e-01,
          1.26982436e-01,  1.59289673e-01,  1.92969367e-01,
          2.30897069e-01,  9.52387750e-02, -5.54224774e-02,
         -4.43238132e-02, -1.20651662e-01, -3.09730321e-01,
          2.73521543e-01, -7.86366880e-01,  8.28892458e-03,
          3.53535265e-02, -5.80390878e-02, -1.26399413e-01,
          3.59690964e-01,  8.58796015e-02, -2.46076938e-02,
          9.87975001e-02, -2.03348193e-02]],

       [[ 1.05237611e-01,  2.85133481e-01,  1.14861816e-01,
          2.91018188e-01,  4.76617627e-02,  2.02615291e-01,
         -1.38904704e-02, -3.07599097e-01,  8.52200016e-02,
         -7.24286795e-01, -2.63927951e-02, -3.83843213e-01,
          2.27552444e-01,  1.21126659e-01,  1.14919662e-01,
          1.98619992e-01,  1.24650896e-01,  1.20262854e-01,
          3.51580083e-02, -3.64731133e-01, -4.29095626e-01,
          1.08454034e-01, -5.97085118e-01, -8.79477784e-02,
          2.74401814e-01, -5.01111820e-02, -2.72738993e-01,
          5.45515493e-02,  1.78768188e-01,  3.16933617e-02,
```

```
        -1.96572855e-01,  1.92812353e-01]],

       [[ 9.81594026e-02,  1.64728776e-01,  5.76331280e-02,
          1.77317426e-01, -2.69686710e-02,  1.32770181e-01,
         -6.73043355e-02, -1.23576820e-01, -2.71054715e-01,
          1.09490663e-01,  2.70639777e-01, -3.73691261e-01,
          1.55811146e-01,  5.63968532e-02, -1.24953397e-01,
          1.43317983e-01,  2.43317802e-02,  2.64560670e-01,
         -8.69006035e-04,  6.07958622e-03, -2.89075285e-01,
          1.32183567e-01, -4.34512228e-01, -8.69849324e-02,
          2.79855669e-01,  1.98766246e-01, -1.08258091e-02,
         -1.26862109e-01,  5.87431453e-02,  8.43889490e-02,
         -4.96364325e-01,  2.67583191e-01]],

       [[-9.00157988e-02,  9.04410183e-02, -1.69018079e-02,
         -2.22317964e-01, -5.92068359e-02, -1.86932221e-01,
         -1.52867332e-01,  1.03414178e-01, -5.75809360e-01,
          2.06499711e-01,  1.00580342e-01,  1.42656401e-01,
         -1.60061464e-01, -1.81272570e-02, -4.27930266e-01,
         -2.38873251e-02, -3.38699281e-01,  1.54039681e-01,
          6.32544383e-02,  2.32734606e-01, -1.91482976e-01,
          4.46928479e-02, -5.17552853e-01,  6.35691732e-02,
         -1.62860066e-01, -3.59707475e-02,  1.06332712e-01,
         -3.87486666e-01, -3.50101709e-01, -9.77838412e-02,
          6.78690821e-02,  2.79461086e-01]]],


      [[[ 6.94897696e-02, -4.56465155e-01, -4.78669368e-02,
          1.70147881e-01, -2.67105490e-01,  1.95284620e-01,
         -1.31940946e-01, -3.93551774e-02, -4.01075184e-01,
         -5.87399125e-01, -1.19585104e-01,  3.52404535e-01,
          1.94345549e-01,  1.21695541e-01,  1.49443313e-01,
         -4.64245111e-01,  1.26615241e-01, -6.73554167e-02,
         -1.11112064e-02,  1.94472909e-01,  3.44417468e-02,
          1.35204559e-02,  3.69025201e-01,  4.17443551e-02,
         -4.44194943e-01, -7.74747282e-02,  1.79393485e-01,
         -1.66174565e-02, -2.50142813e-01, -1.70370668e-01,
         -5.59278019e-02,  9.61616635e-02]],

       [[ 1.27359331e-01, -3.01042795e-01, -9.75781977e-02,
          6.65487275e-02, -2.07983598e-01,  1.05241708e-01,
         -3.16370040e-01, -1.37010902e-01, -4.71739292e-01,
         -6.81548715e-01, -2.00591683e-01, -3.66233150e-03,
          9.99188945e-02, -1.00562572e-01,  7.71338269e-02,
         -3.30850005e-01,  9.19390004e-03,  1.33350283e-01,
          1.25099653e-02, -8.52252245e-02, -1.28456295e-01,
         -6.24202304e-02,  3.88428271e-02, -3.85829285e-02,
         -1.66093856e-01, -5.03325947e-02, -1.25724301e-01,
          1.56014666e-01, -4.98696536e-01, -1.42603954e-02,
         -9.52607468e-02,  2.55258046e-02]],

       [[ 7.18368068e-02,  3.75352688e-02,  3.25325765e-02,
          4.18917462e-02, -7.96908364e-02, -3.27906455e-03,
         -5.25300562e-01,  3.93787883e-02, -5.03541946e-01,
         -2.11107284e-01, -3.07447404e-01, -5.88431299e-01,
         -4.86466847e-02, -2.10774884e-01, -5.47463521e-02,
         -9.07892436e-02, -1.27245605e-01, -1.95403416e-02,
```

```
        -1.10881485e-01, -3.30216676e-01, -1.73106134e-01,
         3.16825174e-02, -2.50107855e-01, -7.54563557e-03,
        -9.42679867e-03, -1.05316326e-01, -1.44612547e-02,
        -6.76659048e-02, -5.32359540e-01, -1.96105074e-02,
        -5.78470707e-01,  1.94161590e-02]],

      [[-3.80605496e-02,  1.15597621e-01,  2.40781292e-01,
        -2.15775073e-01, -1.08158421e-02,  2.50523448e-01,
        -2.11744651e-01,  1.59278288e-01, -1.02526389e-01,
         1.83125898e-01, -1.41086608e-01, -2.69224402e-02,
        -1.44201607e-01, -2.38584906e-01, -2.56093681e-01,
         1.09456867e-01, -2.14618966e-01, -3.72877233e-02,
         5.02127968e-02, -1.49150744e-01, -3.85982603e-01,
         1.31096002e-02, -1.62528232e-01, -9.44341198e-02,
         1.99074000e-01, -2.62909289e-02,  3.22503597e-02,
        -2.31052667e-01, -3.10773790e-01,  7.71962851e-02,
        -3.82670254e-01, -1.52855575e-01]],

      [[-4.80562933e-02,  1.28194824e-01,  8.04379135e-02,
        -3.30098629e-01,  6.94305375e-02,  8.49042386e-02,
        -7.37621933e-02, -1.19480841e-01,  1.78001776e-01,
         2.51053780e-01,  2.64935374e-01,  9.32559595e-02,
        -1.89254865e-01, -1.19727887e-01, -2.49206737e-01,
         3.92570980e-02, -3.79686832e-01, -3.05933714e-01,
        -5.06319888e-02,  2.63252616e-01, -4.32694584e-01,
         8.73524323e-02,  6.51656389e-02, -1.51857316e-01,
         1.28908902e-01,  2.25458071e-02,  1.37101591e-01,
         4.70965765e-02, -2.60380119e-01,  1.04226425e-01,
         1.03774220e-02, -5.79924285e-01]]]], dtype=float32), array
([-0.34538507, -0.0651003 , -0.0605657 , -0.22909553, -0.14077169,
       -0.05809003, -0.05415076, -0.11472243, -0.05088018,  0.0056802 ,
       -0.41413873, -0.09826721, -0.37634802, -0.05388484, -0.26262867,
       -0.0139049 , -0.30295607, -0.18053962, -0.2624145 , -0.14987054,
       -0.05122993, -0.08681966, -0.05492871, -0.26623186, -0.3613396 ,
       -0.41412297, -0.34026945, -0.08845586, -0.00496876, -0.24378292,
       -0.0872119 , -0.21576062], dtype=float32)]
{'name': 'max_pooling2d_1', 'trainable': True, 'dtype': 'float32', 'poo
l_size': (2, 2), 'padding': 'valid', 'strides': (2, 2), 'data_format':
'channels_last'} []
{'name': 'dropout_1', 'trainable': True, 'dtype': 'float32', 'rate': 0.
2, 'noise_shape': None, 'seed': None} []
{'name': 'flatten_1', 'trainable': True, 'dtype': 'float32', 'data_form
at': 'channels_last'} []
{'name': 'dense_2', 'trainable': True, 'dtype': 'float32', 'units': 12
8, 'activation': 'relu', 'use_bias': True, 'kernel_initializer': {'clas
s_name': 'GlorotUniform', 'config': {'seed': None}}, 'bias_initialize
r': {'class_name': 'Zeros', 'config': {}}, 'kernel_regularizer': None,
'bias_regularizer': None, 'activity_regularizer': None, 'kernel_constra
int': None, 'bias_constraint': None} [array([[-0.00244963,  0.0204884 ,
-0.01477033, ..., -0.01727296,
         0.0037416 ,  0.07159058],
       [-0.04051248, -0.0091156 , -0.02415105, ...,  0.07753382,
         0.02495002,  0.1313915 ],
       [-0.09868282,  0.04286613,  0.00943325, ...,  0.00970718,
        -0.01077031,  0.1859574 ],
       ...,
       [ 0.08220999,  0.00320657,  0.05784059, ...,  0.00958308,
```

```
          -0.00750509, -0.00433638],
        [-0.08231065,  0.00303738, -0.08532829, ...,  0.11781107,
          0.01646915, -0.05295718],
        [ 0.03747772,  0.00230891,  0.2194466 , ...,  0.0810186 ,
          0.00917124, -0.03949664]], dtype=float32), array([-0.0697494 ,
-0.12933622,  0.02400744, -0.02914514, -0.01153096,
       -0.07374071, -0.05685407, -0.05014405, -0.045807  , -0.02366221,
        0.05426114, -0.11542899, -0.02937198, -0.06945445, -0.10507935,
       -0.06166892, -0.03792334, -0.09612294,  0.07923929, -0.03742353,
       -0.21819237, -0.04855415, -0.02130529, -0.00544576, -0.05491989,
       -0.00558628,  0.00938595, -0.00460963, -0.08569646, -0.06724875,
        0.09909341,  0.00317494, -0.05115647, -0.04006927, -0.09451327,
       -0.00363668,  0.10011022, -0.03769823, -0.04692935, -0.00753925,
       -0.0365366 , -0.04764347,  0.04364438, -0.02741652, -0.05189038,
        0.02228923, -0.05266039, -0.11620439, -0.05586323, -0.12111288,
       -0.02341688, -0.00785092, -0.0858429 , -0.03339842,  0.06113833,
        0.07796517,  0.17715423, -0.09414646, -0.01163498, -0.02284488,
       -0.03692288, -0.05826164, -0.03130798, -0.12237389, -0.11213206,
       -0.18661796, -0.06618702,  0.00515154,  0.06594225,  0.00077354,
       -0.11601824,  0.00183565, -0.07537822, -0.0673568 , -0.04044918,
       -0.07828008, -0.2309237 ,  0.01849574, -0.06394696, -0.00176849,
       -0.06980103, -0.07419752, -0.04606844, -0.01546294, -0.12819   ,
        0.04635283, -0.01610494, -0.15522572, -0.03245178,  0.09932876,
        0.0386508 , -0.00923306,  0.03146196, -0.09106684, -0.12961072,
       -0.0678592 , -0.00645411, -0.00943616, -0.07351997, -0.00754467,
       -0.0719175 , -0.03496964, -0.06499901, -0.01579884, -0.10952124,
       -0.0332803 , -0.01139016, -0.02190196, -0.00710684,  0.00220169,
       -0.01999539, -0.0340815 , -0.04948583, -0.07451184, -0.05747848,
       -0.04641361, -0.14643288, -0.03070576,  0.05953415, -0.12134353,
       -0.09211281,  0.01866474,  0.05525352, -0.08645079, -0.10176076,
       -0.11381823, -0.03654835,  0.07518332], dtype=float32)]
{'name': 'dense_3', 'trainable': True, 'dtype': 'float32', 'units': 10,
'activation': 'softmax', 'use_bias': True, 'kernel_initializer': {'clas
s_name': 'GlorotUniform', 'config': {'seed': None}}, 'bias_initialize
r': {'class_name': 'Zeros', 'config': {}}, 'kernel_regularizer': None,
'bias_regularizer': None, 'activity_regularizer': None, 'kernel_constra
int': None, 'bias_constraint': None} [array([[ 0.22751303,  0.07116753,
-0.12112398, ...,  0.0310067 ,
          0.29751632,  0.14487289],
        [ 0.04177473, -0.05423146, -0.07803074, ..., -0.19545197,
         -0.01323551,  0.05512254],
        [-0.09823686,  0.18048772,  0.26633352, ..., -0.24154374,
          0.16609193,  0.06753043],
        ...,
        [ 0.22901817, -0.36596105, -0.13597994, ..., -0.01908575,
          0.09681717,  0.11688822],
        [-0.04842698, -0.0732211 ,  0.19431911, ..., -0.0610413 ,
          0.11255985, -0.05603024],
        [-0.23551854,  0.22178076, -0.48258162, ..., -0.11463284,
         -0.04866566, -0.43127036]], dtype=float32), array([-0.0429344 ,
0.01867466, -0.03304278,  0.01511194,  0.00222329,
       -0.00061798, -0.07435245,  0.0095069 ,  0.00911702,  0.0261830
5],
      dtype=float32)]
```

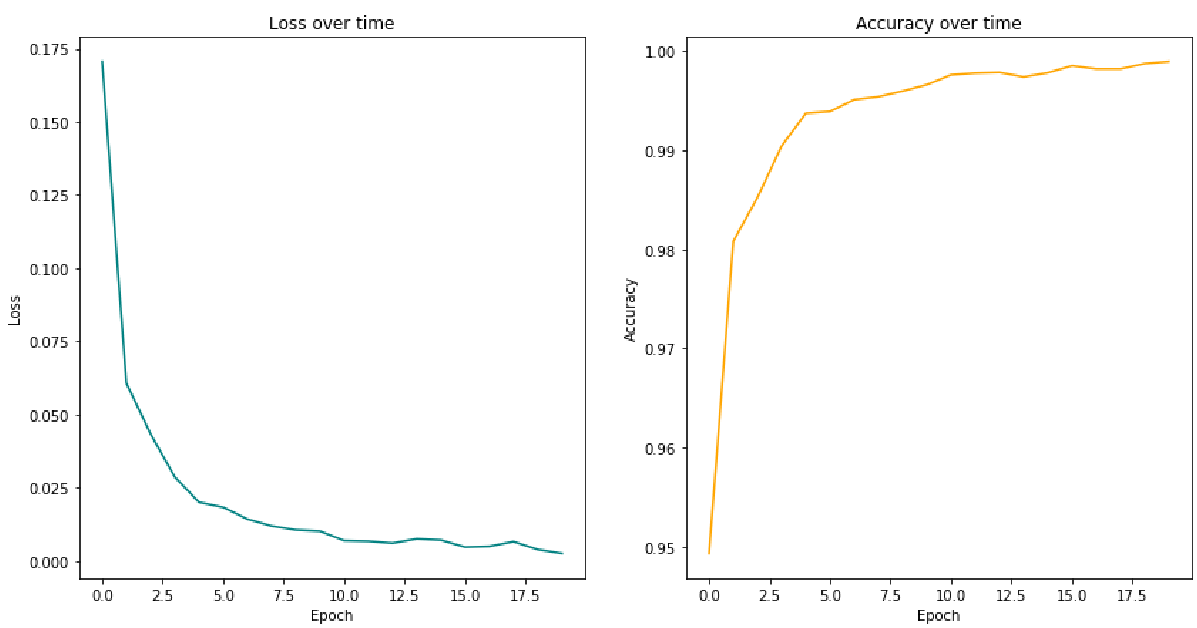
Testing the Model

In [28]:
```
print("Evaluating on Test Data")
results = model.evaluate(X_test, y_test, batch_size=b_size,verbose=0)
print("Test Accuracy:", results[1])
```

```
Evaluating on Test Data
Test Accuracy: 0.9852092266082764
```

Visualizing the Performance

In [31]:
```
# accuracy vs epochs
plt.figure(figsize=(14,7))

plt.subplot(1,2,1)
plt.plot(history.history['loss'], c='teal')
plt.title('Loss over time')
plt.ylabel('Loss')
plt.xlabel('Epoch')

plt.subplot(1,2,2)
plt.plot(history.history['accuracy'], c='orange')
plt.title('Accuracy over time')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.show()
```



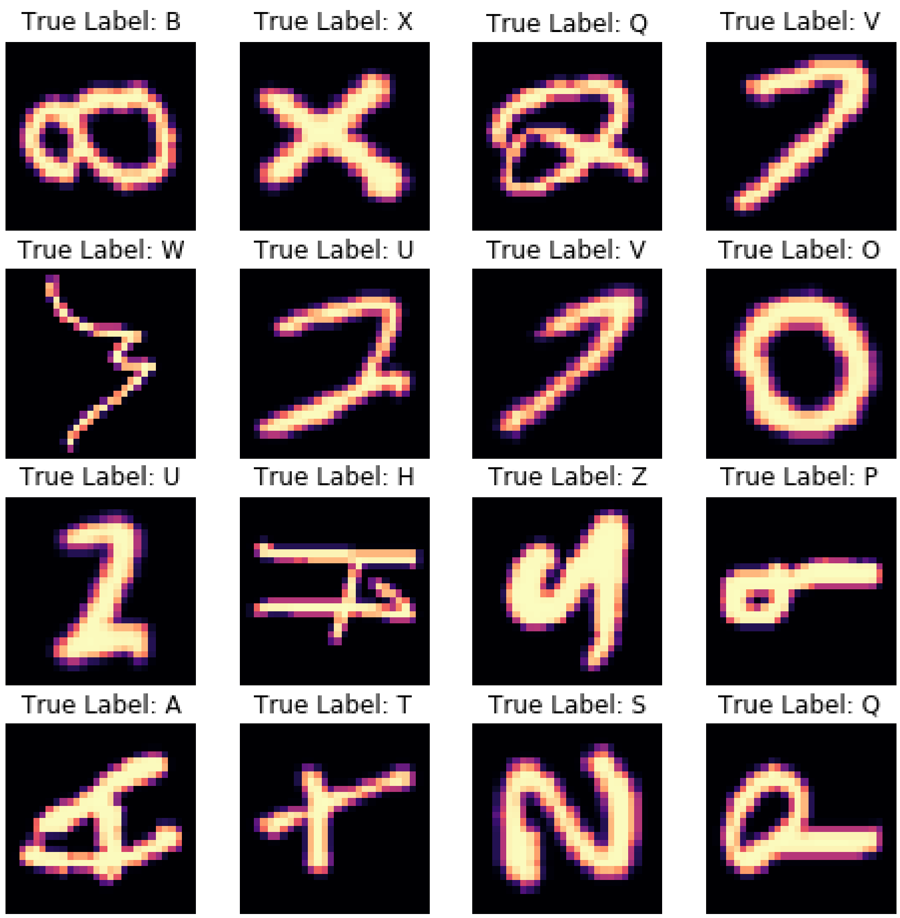
2. Handwritten character recognition

In [57]:
```
# Loading data from CSV file
X = pd.read_csv('emnist-letters.csv')
y = X['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33
, random_state=42)
```

Preprocessing and Visualizing the Training Data

In [58]:
```
# Storing the labels in a separate vector y_train
y_train = np.array(X_train['label'])

# Dropping the labels column
X_train = X_train.drop(['label'], axis=1)

# Normalizing the values of the pixels
X_train = X_train / 255

# Converting X_train dataframe to numpy array
X_train = np.array(X_train)

print("-------------------------")
print("SAMPLE OF TRAINING DATA")
print("-------------------------")

# Plotting the data
plt.figure(figsize=(8,8))
for i in range(16):
    plt.subplot(4,4,i+1)
    plt.axis('off')
    r = np.random.randint(X_train.shape[0])   # Get a random image to sh
ow
    plt.title('True Label: '+str(chr(ord('@')+y_train[r]))) # Show its l
abel as title
    plt.imshow(X_train[r].reshape(28,28), cmap='magma')    # Plotting th
e image
plt.show()

# Converting the labels vector into one hot encoded form
oh = np.zeros((y_train.size, y_train.max()+1))
oh[np.arange(y_train.size), y_train] = 1
y_train = np.array(oh)

# Reshaping X_train to the format of the network
X_train = X_train.reshape(X_train.shape[0], 28, 28, 1).astype('float32')
```

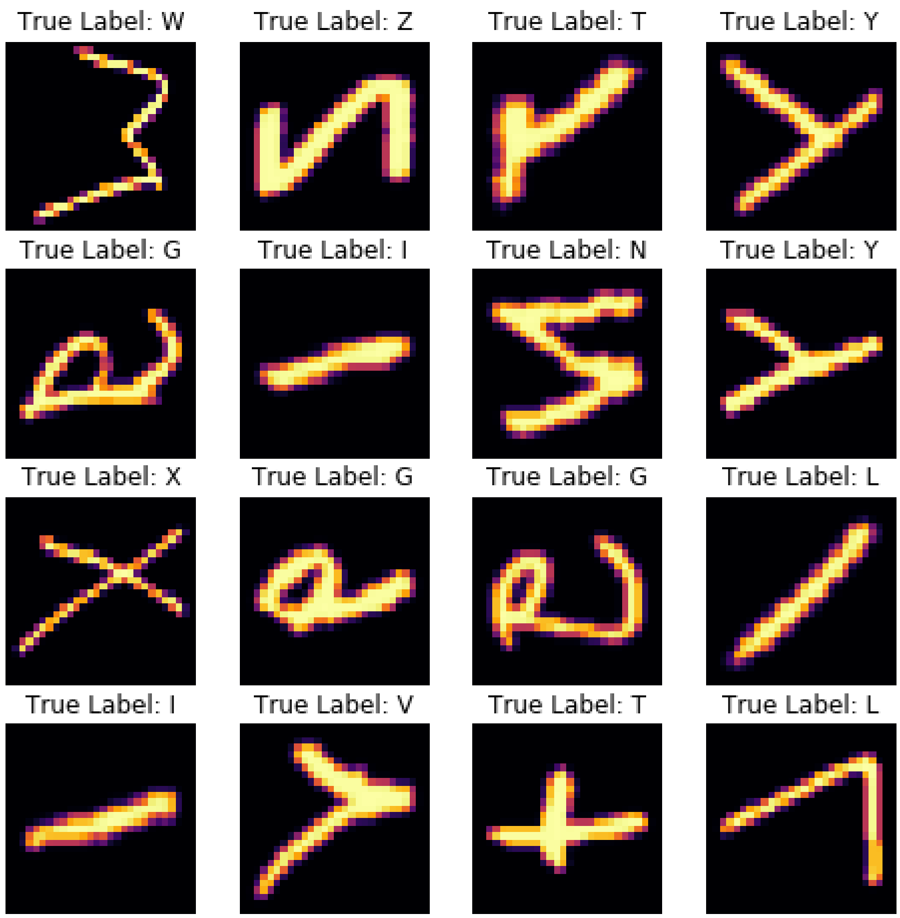
```
-------------------------
SAMPLE OF TRAINING DATA
-------------------------
```



Preprocessing and Visualizing the Testing Data

In [59]:
```
# Storing the labels in a separate vector y_train
y_test = np.array(X_test['label'])

# Dropping the labels column
X_test = X_test.drop(['label'], axis=1)

# Normalizing the values of the pixels
X_test = X_test/ 255

# Converting X_train dataframe to numpy array
X_test = np.array(X_test)

print("-------------------------")
print("SAMPLE OF TESTING DATA")
print("-------------------------")

# Plotting the data
plt.figure(figsize=(8,8))
for i in range(16):
    plt.subplot(4,4,i+1)
    plt.axis('off')
    r = np.random.randint(X_test.shape[0])   # Get a random image to sho
w
    plt.title('True Label: '+str(chr(ord('@')+y_test[r]))) # Show its la
bel as title
    plt.imshow(X_test[r].reshape(28,28), cmap='inferno')    # Plotting t
he image
plt.show()

# Converting the labels vector into one hot encoded form
oh = np.zeros((y_test.size, y_test.max()+1))
oh[np.arange(y_test.size), y_test] = 1
y_test = np.array(oh)

# Reshaping X_train to the format of the network
X_test = X_test.reshape(X_test.shape[0], 28, 28, 1).astype('float32')

# number of class labels
no_of_label_classes = y_test.shape[1]
```

```
-------------------------
SAMPLE OF TESTING DATA
-------------------------
```



Initializing the Hyperparameters

In [60]:
```
iterations = 10
b_size = 20
```

Training the Model

Here I train the model, and for each epoch I also print out the time taken, loss, and accuracy. Likewise, after training, I print out the model summary and all of the weights of the neural network, as required for the assignment.

We use the same network as the first task.

In [66]:
```
# Creating the model
model = create_model()
model.summary()

# Fitting the model
history = model.fit(X_train, y_train,epochs=iterations,batch_size=b_size
, verbose=2)

for layer in model.layers: print(layer.get_config(), layer.get_weights
())
```

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_4 (Conv2D)            (None, 24, 24, 32)        832       
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 12, 12, 32)        0         
_________________________________________________________________
dropout_4 (Dropout)          (None, 12, 12, 32)        0         
_________________________________________________________________
flatten_4 (Flatten)          (None, 4608)              0         
_________________________________________________________________
dense_8 (Dense)              (None, 128)               589952    
_________________________________________________________________
dense_9 (Dense)              (None, 27)                3483      
=================================================================
Total params: 594,267
Trainable params: 594,267
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
235/235 - 2s - loss: 1.5737 - accuracy: 0.5335
Epoch 2/10
235/235 - 2s - loss: 0.7698 - accuracy: 0.7673
Epoch 3/10
235/235 - 2s - loss: 0.5056 - accuracy: 0.8443
Epoch 4/10
235/235 - 2s - loss: 0.3846 - accuracy: 0.8795
Epoch 5/10
235/235 - 2s - loss: 0.2851 - accuracy: 0.9104
Epoch 6/10
235/235 - 2s - loss: 0.2123 - accuracy: 0.9298
Epoch 7/10
235/235 - 2s - loss: 0.1798 - accuracy: 0.9386
Epoch 8/10
235/235 - 2s - loss: 0.1404 - accuracy: 0.9526
Epoch 9/10
235/235 - 2s - loss: 0.1153 - accuracy: 0.9616
Epoch 10/10
235/235 - 2s - loss: 0.0990 - accuracy: 0.9648
{'name': 'conv2d_4', 'trainable': True, 'batch_input_shape': (None, 28, 
28, 1), 'dtype': 'float32', 'filters': 32, 'kernel_size': (5, 5), 'stri
des': (1, 1), 'padding': 'valid', 'data_format': 'channels_last', 'dila
tion_rate': (1, 1), 'activation': 'relu', 'use_bias': True, 'kernel_ini
tializer': {'class_name': 'GlorotUniform', 'config': {'seed': None}}, 
'bias_initializer': {'class_name': 'Zeros', 'config': {}}, 'kernel_regu
larizer': None, 'bias_regularizer': None, 'activity_regularizer': None, 
'kernel_constraint': None, 'bias_constraint': None} [array([[[[-2.62659
162e-01, -2.86796778e-01, -5.96503094e-02,
           2.13486105e-01, -2.03730986e-02, -8.42066482e-02,
           1.15707956e-01,  4.39602472e-02,  5.15466370e-02,
           3.97382118e-02,  3.03045753e-02,  6.94067106e-02,
           7.69493356e-02, -9.37363412e-03, -7.71879330e-02,
           1.08277172e-01, -2.13169232e-02, -2.05471348e-02,
          -3.67316790e-02, -1.96057558e-01, -1.72141954e-01,
          -2.77983636e-01, -1.60927162e-01, -8.70682225e-02,
          -1.16195105e-01, -2.96903163e-01, -1.48989871e-01,
```

```
        -2.05770791e-01, -2.99408108e-01, -2.52330862e-03,
         3.11799459e-02,  2.86261346e-02]],

       [[-2.74173677e-01, -2.79942483e-01,  1.57165080e-02,
         2.40237638e-01,  1.83474422e-02, -7.56371242e-04,
         7.60851055e-02,  4.97812107e-02,  7.05849379e-02,
        -7.49548431e-04,  3.55659798e-02,  1.35732278e-01,
         1.13593563e-01, -1.68401957e-01, -1.58404946e-01,
         9.38318297e-02,  5.29651567e-02, -3.08635719e-02,
        -1.42153695e-01, -1.48698866e-01, -2.29519710e-01,
        -4.61056888e-01, -1.71591237e-01,  1.11111782e-01,
        -2.55402237e-01, -2.80899797e-02, -1.10367425e-01,
        -2.05127060e-01, -3.66590470e-01, -6.11406527e-02,
        -6.11772574e-02,  8.84649083e-02]],

       [[-1.05772443e-01, -2.18333676e-01,  4.47129942e-02,
         2.07482457e-01,  2.21011713e-01,  5.41245118e-02,
         2.29674637e-01, -2.15152698e-03,  3.64839435e-02,
         8.87693465e-02,  2.15575114e-01, -1.52344666e-02,
        -1.94838177e-02, -8.54951516e-02, -1.11961327e-01,
         2.40080804e-01,  1.48383290e-01,  3.02308630e-02,
        -3.89017969e-01,  1.52964324e-01, -2.14367528e-02,
        -2.74275929e-01, -1.28551871e-01,  9.36241224e-02,
        -3.14569354e-01,  2.78608680e-01, -2.58944929e-04,
         1.00292712e-01, -1.79553479e-01,  9.45853069e-02,
        -6.61514774e-02,  1.19843893e-01]],

       [[-5.80292754e-02, -2.49524415e-01,  2.71763802e-02,
        -2.63858959e-02,  2.25128010e-01,  1.31004294e-02,
         1.32255942e-01, -1.19979545e-01, -4.89309393e-02,
         1.38367210e-02,  1.51144832e-01, -6.32454222e-03,
        -4.31733839e-02,  8.10490847e-02, -1.86602741e-01,
         2.04728439e-01, -1.21874074e-02, -6.57166913e-02,
        -3.71662885e-01,  1.95477188e-01,  2.62798071e-02,
         5.63968346e-02,  1.44009247e-01,  1.80196553e-03,
        -3.21606487e-01,  1.41824841e-01,  6.04636362e-03,
         1.79051027e-01, -4.57627289e-02,  2.21532837e-01,
        -7.58857355e-02,  1.22849055e-01]],

       [[ 9.55766439e-02, -3.08572084e-01, -7.56677613e-02,
        -3.27113308e-02,  4.67826277e-02, -1.70172349e-01,
         7.37870932e-02, -1.04867863e-02, -4.93294327e-04,
         3.24250869e-02,  1.59173205e-01, -4.78907786e-02,
        -2.15266854e-01,  1.58834532e-01, -8.28122497e-02,
         1.87717646e-01, -1.37381345e-01, -1.59224197e-01,
        -2.37653270e-01,  1.06023595e-01,  7.85220638e-02,
         2.11046293e-01,  2.32665259e-02, -1.33995488e-01,
        -3.50138843e-01, -2.03159302e-01, -7.65269175e-02,
         2.51578659e-01,  8.17424133e-02,  1.29557505e-01,
         5.60957473e-03, -6.05687313e-02]]],


      [[[-1.35615408e-01, -1.27587110e-01,  8.55320394e-02,
          1.53800607e-01, -2.36871883e-01, -6.97078034e-02,
          1.86518773e-01,  4.31037098e-02,  3.44631937e-03,
          7.51786232e-02,  3.03977840e-02,  3.45581286e-02,
          1.28091825e-02, -1.33735716e-01, -1.69286579e-02,
```

```
 -7.32660219e-02,  1.40635088e-01,  7.00218827e-02,
  1.56782582e-01, -3.03819478e-01, -1.09061837e-01,
 -2.72165328e-01, -1.19991057e-01, -6.86534913e-03,
 -3.84455398e-02, -1.09113723e-01, -6.86644092e-02,
 -4.52649407e-02, -1.05756544e-01, -1.65130913e-01,
 -8.58709123e-03, -5.01008220e-02]],

[[-1.16132163e-01, -2.73844510e-01,  1.33469805e-01,
  1.88565224e-01, -2.04132751e-01,  6.04696795e-02,
  1.56646162e-01, -4.88011912e-03,  5.46953715e-02,
 -6.00561723e-02,  1.11008875e-01,  1.48113489e-01,
  1.68293566e-01, -7.00355470e-02,  3.59472865e-03,
  1.29857987e-01,  1.21785969e-01,  8.60592872e-02,
  8.80878344e-02, -1.75672993e-01, -1.66547015e-01,
 -3.64009649e-01, -6.73314333e-02,  4.02637981e-02,
 -1.34429052e-01, -5.67913130e-02, -4.73518334e-02,
 -1.18136115e-01,  3.46367285e-02, -1.11098319e-01,
  9.32278559e-02, -8.56781825e-02]],

[[ 1.00850299e-01, -2.99411535e-01, -1.18365753e-02,
 -1.21116843e-02,  8.91422331e-02,  1.75691932e-01,
  1.94646090e-01,  9.06918719e-02,  6.75702766e-02,
 -9.58003104e-03,  4.64516059e-02,  1.52792960e-01,
 -2.35740021e-02, -1.13378361e-01, -9.43327323e-04,
  1.91207856e-01,  3.89800556e-02,  9.59627330e-03,
 -1.09257191e-01,  2.15609714e-01,  1.17117144e-01,
  1.80152934e-02, -4.32841405e-02, -1.41552808e-02,
 -4.90938090e-02,  2.01063648e-01,  1.19261459e-01,
 -1.41455010e-01,  4.09114473e-02,  4.04385105e-02,
  3.19142826e-02, -7.80320242e-02]],

[[ 1.34671763e-01, -2.84518272e-01, -6.13896511e-02,
 -1.34139657e-01,  1.99804410e-01,  2.89837476e-02,
  1.31802842e-01, -1.16519323e-02,  8.39101821e-02,
  6.77514896e-02,  1.63867697e-01,  2.08902493e-01,
 -1.13301210e-01,  1.61879107e-01,  8.97370353e-02,
  1.97659940e-01, -1.98154196e-01, -8.04706365e-02,
 -1.02754042e-01,  1.26881480e-01,  7.87253529e-02,
  1.52747661e-01,  1.57910734e-01, -4.66593467e-02,
 -1.05718590e-01, -4.22094745e-04,  1.08486392e-01,
 -5.77023961e-02,  2.42475644e-01,  2.87644953e-01,
 -2.85092853e-02, -2.54289061e-02]],

[[ 5.91534004e-02, -2.54712284e-01,  4.09256555e-02,
 -1.64234400e-01,  1.86743379e-01, -2.58700579e-01,
  2.70058401e-02, -5.82129695e-02,  5.85606284e-02,
 -3.03171128e-02,  1.49575487e-01,  1.60868078e-01,
 -2.53512114e-01,  1.67986989e-01,  1.66076764e-01,
  2.78130230e-02, -2.73983151e-01, -1.25928922e-02,
 -2.99690485e-01,  6.92425445e-02,  1.08949699e-01,
  2.14452252e-01, -9.37197264e-03, -1.63747147e-01,
 -1.37229621e-01, -1.39971584e-01,  5.31838201e-02,
  2.24838361e-01,  1.93996295e-01,  2.00219914e-01,
  2.80684326e-02, -9.21344385e-02]]],


[[[ 1.13677911e-01,  5.98952174e-02,  8.89724270e-02,
```

```
   1.77398309e-01, -2.53527015e-01, -1.95676789e-01,
  -3.87208685e-02, -4.90595996e-02, -8.62180814e-03,
   4.13912795e-02, -6.68181330e-02, -4.69542406e-02,
   2.74125393e-02, -1.77915439e-01, -8.48564878e-02,
  -1.42271638e-01,  2.68108517e-01, -3.44536379e-02,
   1.99945420e-01, -3.39091301e-01, -7.69308656e-02,
  -3.74482691e-01,  4.08051200e-02,  8.34314674e-02,
   5.74265681e-02, -1.00186162e-01,  1.88841987e-02,
  -3.60563700e-03,  1.06279962e-01, -2.13442639e-01,
  -2.94251721e-02,  4.15813830e-03]],

[[ 1.65234655e-01, -1.80843901e-02, -1.34326564e-02,
   1.06972620e-01, -2.42887735e-01,  1.34591967e-01,
   3.48634156e-03,  3.85014527e-02, -8.60608071e-02,
   3.40871029e-02, -1.79856643e-01,  1.34009078e-01,
   2.06215292e-01, -5.92581555e-02,  8.00753087e-02,
  -1.19512551e-01,  2.34500095e-01,  2.98365392e-02,
   3.12632322e-01, -4.06346321e-02,  1.67404469e-02,
  -3.97950828e-01,  5.81528917e-02,  6.29952103e-02,
  -3.25238183e-02, -2.80236341e-02,  1.13554187e-01,
  -4.43345755e-02,  4.05011140e-02, -2.98723012e-01,
   1.43072203e-01,  3.27277742e-02]],

[[ 1.19816817e-01, -1.34647325e-01, -5.81295639e-02,
  -1.42520815e-01, -3.75352651e-02,  1.67114764e-01,
   3.46442349e-02,  1.11377493e-01,  7.29165152e-02,
   5.29730581e-02, -3.19445640e-01,  2.14206442e-01,
   1.29756257e-01, -4.54512611e-02,  6.64651021e-02,
  -1.19237565e-01, -4.54472750e-02,  1.02998257e-01,
   1.94629446e-01,  1.99923098e-01,  2.23926693e-01,
  -4.36312631e-02,  2.61085536e-02,  4.73002568e-02,
   1.45351238e-04,  1.26317948e-01,  1.29567564e-01,
  -2.02536322e-02,  1.86657399e-01,  2.99911108e-02,
   1.29414335e-01,  3.16935144e-02]],

[[ 1.01807088e-01, -9.45778042e-02, -7.80034661e-02,
  -2.15662271e-01,  2.77773082e-01,  5.48660085e-02,
  -6.95882142e-02,  6.12227507e-02, -2.03209883e-03,
  -1.72556024e-02, -3.58385533e-01,  7.48506635e-02,
  -9.47060362e-02,  1.69833854e-01,  1.15562789e-01,
  -1.13943994e-01, -3.00621957e-01,  7.48182759e-02,
   8.03980753e-02,  2.04784498e-01,  1.64917022e-01,
   2.34666809e-01, -1.15321745e-04, -1.00103475e-01,
   8.88334736e-02, -4.46435623e-02,  9.01877582e-02,
  -2.16104865e-01,  9.19323117e-02,  1.02828227e-01,
  -5.54352030e-02, -9.19570960e-03]],

[[-3.33392108e-03,  3.02629564e-02,  3.86429131e-02,
  -2.83232689e-01,  2.50221789e-01, -3.32195431e-01,
  -2.02989846e-01,  6.52677640e-02,  1.10026442e-01,
  -9.37612727e-03, -2.26410255e-01,  1.04694746e-01,
  -2.03726307e-01,  8.59235823e-02,  4.95339036e-02,
   3.51525173e-02, -2.87060142e-01, -1.12331100e-01,
  -4.87355031e-02, -6.64591491e-02,  3.96784320e-02,
   3.02388668e-01, -6.54734746e-02, -1.20600551e-01,
  -1.43786706e-02,  5.25536612e-02,  5.97640686e-02,
  -1.13742739e-01,  1.20522957e-02,  2.73099065e-01,
```

```
         9.84830558e-02, -3.02547812e-02]]],


       [[[ 1.66483015e-01,  6.89576045e-02,  8.57578516e-02,
           6.70876205e-02, -1.50828153e-01, -1.04648963e-01,
          -1.22031219e-01,  2.64707673e-02,  5.78464158e-02,
          -4.16798657e-03, -1.34903431e-01, -1.78213865e-01,
           9.88570675e-02, -1.44564703e-01,  8.95365551e-02,
          -1.70211360e-01,  3.16950887e-01,  4.69639078e-02,
           7.52942860e-02, -3.30230504e-01, -9.39007699e-02,
          -2.53229797e-01, -6.89247102e-02,  9.44090113e-02,
           1.50562122e-01, -1.32451147e-01,  8.50838572e-02,
           2.11223394e-01,  1.48062810e-01, -1.06895141e-01,
           5.28408177e-02,  6.99358210e-02]],

        [[ 1.83349699e-01,  5.66084534e-02, -1.19302429e-01,
          -1.16853133e-01, -3.66479188e-01,  1.52540296e-01,
          -2.18576819e-01,  7.75028020e-02, -3.72556522e-02,
           6.89183101e-02, -1.52770385e-01, -1.67087957e-01,
           1.83120474e-01, -1.12878688e-01,  3.79815623e-02,
          -2.42515668e-01,  2.19392523e-01,  4.66349423e-02,
           1.42321035e-01, -1.43494278e-01,  1.50398046e-01,
          -2.69408345e-01, -1.21298134e-02,  1.16838571e-02,
           1.27909347e-01, -6.48622140e-02,  1.09666817e-01,
           6.29280284e-02,  1.01741008e-01, -3.27377945e-01,
          -6.56959489e-02,  3.75649035e-02]],

        [[-1.79627836e-02,  1.47019133e-01,  3.38745862e-03,
          -2.39950716e-01, -8.94018933e-02,  2.25262821e-01,
          -2.23478898e-01, -5.41968420e-02, -1.06160626e-01,
          -1.25734387e-02, -1.91965401e-01, -4.32798900e-02,
           2.93370970e-02, -5.66029549e-02,  9.34825167e-02,
          -3.41895342e-01, -4.24660221e-02,  1.04058504e-01,
           2.21587345e-01,  1.79928094e-01,  1.09370075e-01,
           6.93826983e-03,  4.58716974e-02, -2.28842553e-02,
           1.63205415e-01, -7.41138831e-02,  1.63062494e-02,
          -3.25407274e-03,  1.47588432e-01, -1.74707472e-01,
           8.52701142e-02,  9.36009064e-02]],

        [[-2.38645971e-02,  1.28944576e-01,  2.37401444e-02,
          -1.56137854e-01,  7.19244480e-02,  6.54591545e-02,
          -3.47693264e-01,  4.23857868e-02, -5.18962108e-02,
           6.66230023e-02, -2.01441243e-01, -1.64086774e-01,
          -2.06408232e-01,  1.07706450e-01,  6.32021651e-02,
          -1.85568348e-01, -2.88134277e-01,  1.01520844e-01,
           1.62370935e-01,  7.77530745e-02,  2.58034039e-02,
           2.38797054e-01,  6.63897842e-02,  6.70166239e-02,
           1.36711821e-01, -6.02002628e-02, -3.86222713e-02,
          -3.53026032e-01, -1.30234763e-01,  8.10424536e-02,
          -4.57605235e-02,  1.34125143e-01]],

        [[-1.10553548e-01,  2.02968240e-01,  8.27477053e-02,
          -2.26392180e-01,  1.13612242e-01, -3.19933534e-01,
          -2.52651274e-01,  9.51175764e-02,  7.51160383e-02,
           2.93049309e-02, -1.98385432e-01, -1.56639948e-01,
          -2.52513379e-01,  1.26719579e-01, -8.16111863e-02,
          -2.26818904e-01, -1.58489585e-01,  4.26483862e-02,
```

```
        2.23337971e-02,  3.87800485e-03, -1.75147608e-01,
        1.87814668e-01,  6.15669824e-02, -2.62696259e-02,
        1.58558726e-01,  1.29794672e-01, -9.27553177e-02,
       -1.81734696e-01, -2.44190216e-01,  1.65678546e-01,
        1.28542796e-01,  3.00317891e-02]]],


     [[[ 4.49034944e-03, -3.26143801e-02,  3.20552625e-02,
       -1.38056383e-01, -8.29127431e-02, -1.61337461e-02,
       -2.21479982e-01, -9.82442200e-02, -3.01513206e-02,
       -5.31897433e-02,  4.02822904e-02, -2.55720377e-01,
        6.59745559e-02, -6.81136549e-02,  9.26379710e-02,
       -3.21709216e-01,  2.03363642e-01, -4.89337370e-02,
       -1.91918746e-01, -2.02263221e-01,  1.43022329e-01,
       -2.02380821e-01,  7.78121203e-02,  1.17201768e-01,
        6.94472343e-02,  8.37581232e-03,  4.01134677e-02,
        3.17623079e-01,  5.65646254e-02, -1.31521448e-01,
       -1.26611693e-02,  4.67897430e-02]],

      [[ 1.46308271e-02, -1.67352557e-02, -7.60942996e-02,
       -2.76473373e-01, -3.29332829e-01,  1.57207862e-01,
       -2.64782697e-01, -1.57225847e-01, -1.28814289e-02,
        6.33320957e-02,  1.59012169e-01, -3.65610182e-01,
        2.40351856e-01, -1.95309117e-01, -7.47222751e-02,
       -1.43935397e-01, -1.99875492e-03, -7.52519369e-02,
       -7.23101478e-03, -9.41338390e-02,  2.87770815e-02,
       -2.43260011e-01, -1.01595142e-04,  9.01878551e-02,
        1.89031571e-01, -1.06932715e-01, -3.92927788e-02,
        3.29116702e-01,  5.93598559e-02, -3.73231441e-01,
       -1.24981388e-01, -1.36712655e-01]],

      [[ 1.16621256e-02,  9.95317847e-02,  7.74843544e-02,
       -3.22906703e-01, -2.75310367e-01,  9.99361426e-02,
       -2.81053543e-01, -4.37528826e-02, -3.48962136e-02,
        7.80442497e-03,  2.85527766e-01, -3.78047347e-01,
        2.01708585e-01, -7.91985989e-02, -5.69458194e-02,
       -1.41971394e-01, -2.59458661e-01,  6.54081777e-02,
       -3.06003746e-02,  7.07594156e-02,  5.99960573e-02,
       -3.83427329e-02, -2.03571599e-02, -5.18420115e-02,
        8.00544545e-02,  8.39608908e-02,  3.26460563e-02,
       -1.27736358e-02, -4.24704514e-02, -4.13143516e-01,
       -1.06598087e-01,  2.12545004e-02]],

      [[-1.79678857e-01,  2.92040944e-01,  3.94303389e-02,
       -2.83910096e-01, -1.19689628e-01, -1.01726074e-02,
       -1.69434786e-01,  7.62162283e-02,  1.16324602e-02,
        9.77107212e-02,  2.40018353e-01, -3.10143054e-01,
       -7.12881684e-02,  1.15302324e-01, -4.60326336e-02,
       -2.14348406e-01, -3.29549879e-01,  6.34265915e-02,
       -9.25727282e-03,  1.40772268e-01, -1.36457488e-01,
        2.08576128e-01,  1.13361694e-01,  5.36531443e-03,
        8.83093849e-02,  1.00816801e-01, -1.38089821e-01,
       -1.91532731e-01, -3.25783521e-01, -2.40728021e-01,
        7.33572468e-02,  7.71664530e-02]],

      [[-1.84878647e-01,  2.38010392e-01,  3.67400534e-02,
       -2.20994070e-01,  1.03567854e-01, -2.35417947e-01,
```

```
            -2.02162400e-01,  5.23276329e-02, -9.37771946e-02,
             1.51067879e-02,  1.46084920e-01, -1.55896708e-01,
            -2.61327922e-01,  1.41152665e-01, -1.57011561e-02,
            -1.00559548e-01, -1.58729285e-01,  6.32182583e-02,
            -4.32793889e-03, -2.21676473e-03, -7.76459351e-02,
             1.42884538e-01,  4.88758832e-02,  2.26881392e-02,
             1.19593412e-01,  2.83851158e-02, -1.54858232e-01,
            -3.10419649e-01, -1.85245752e-01, -1.71365235e-02,
             2.72380840e-02, -1.00449361e-02]]]], dtype=float32), array
([-0.07303513, -0.00490717, -0.10133684, -0.00198583, -0.01215528,
       -0.0096582 , -0.02553517, -0.12782653, -0.1458519 , -0.16512218,
       -0.02873799, -0.03154898, -0.02204422, -0.00796115, -0.11212323,
       -0.01043833, -0.0158116 , -0.1209944 , -0.05459162, -0.00738281,
       -0.15833126, -0.0057979 , -0.09082346, -0.11932043, -0.0627428 ,
       -0.00505546, -0.07251032, -0.00159513, -0.04219499, -0.00361742,
       -0.10584531, -0.10525089], dtype=float32)]
{'name': 'max_pooling2d_4', 'trainable': True, 'dtype': 'float32', 'poo
l_size': (2, 2), 'padding': 'valid', 'strides': (2, 2), 'data_format':
'channels_last'} []
{'name': 'dropout_4', 'trainable': True, 'dtype': 'float32', 'rate': 0.
2, 'noise_shape': None, 'seed': None} []
{'name': 'flatten_4', 'trainable': True, 'dtype': 'float32', 'data_form
at': 'channels_last'} []
{'name': 'dense_8', 'trainable': True, 'dtype': 'float32', 'units': 12
8, 'activation': 'relu', 'use_bias': True, 'kernel_initializer': {'clas
s_name': 'GlorotUniform', 'config': {'seed': None}}, 'bias_initialize
r': {'class_name': 'Zeros', 'config': {}}, 'kernel_regularizer': None,
'bias_regularizer': None, 'activity_regularizer': None, 'kernel_constra
int': None, 'bias_constraint': None} [array([[-0.00815926,  0.08774213,
-0.11695202, ..., -0.0234357 ,
         0.07634362, -0.01249438],
       [-0.00873554,  0.06557878, -0.02145915, ..., -0.0061143 ,
         0.07373478, -0.09546915],
       [-0.02885514,  0.03065909, -0.02891535, ...,  0.01816612,
         0.00132762, -0.04716184],
       ...,
       [ 0.01958124,  0.06900159, -0.04689107, ...,  0.02337674,
         0.02043335, -0.021426  ],
       [-0.01256807,  0.14285229, -0.02170241, ...,  0.03707835,
        -0.03222957,  0.04770225],
       [-0.02687442,  0.00697224, -0.02735196, ...,  0.01298561,
        -0.01323174, -0.06714709]], dtype=float32), array([-0.00659706,
0.0416751 ,  0.07021477, -0.00706996,  0.04073195,
        0.0359399 ,  0.01087742, -0.00874299,  0.02181394,  0.06509011,
        0.01457337, -0.00385433, -0.01634564, -0.0777375 , -0.02900325,
       -0.03676903, -0.00620523,  0.02034576, -0.00560687,  0.01320661,
       -0.00887606,  0.01304791, -0.00753049,  0.03849972, -0.03112127,
       -0.011364  ,  0.00211957, -0.0062126 , -0.02746272,  0.01914381,
        0.00323765,  0.00130585, -0.007411  , -0.01311271,  0.01354863,
        0.01319304,  0.00383648, -0.01295357,  0.01619292,  0.02162124,
        0.0117601 ,  0.02799085, -0.00966535, -0.01614583, -0.01100206,
        0.01858019, -0.00839692,  0.00483095, -0.02768645,  0.03888199,
       -0.03685933, -0.00600476, -0.04267141,  0.00347424, -0.00030342,
       -0.00637326, -0.00714785,  0.00079179,  0.02420245, -0.00835755,
       -0.00162547, -0.01956173,  0.01403755,  0.00921485, -0.00847038,
        0.06304734, -0.03262193, -0.00870738, -0.00600491,  0.00189905,
       -0.01654987, -0.01197515, -0.02278707,  0.0237876 , -0.00927266,
```

```
         0.03180752, -0.01124519,  0.02044147, -0.0071295 ,  0.03804911,
        -0.0082936 , -0.02067241, -0.06974455,  0.00350823, -0.07550107,
         0.04133981,  0.03957261,  0.0068303 ,  0.01396197,  0.0013601 ,
        -0.01478914, -0.02594884, -0.00701967, -0.00815306, -0.00745951,
         0.00091213,  0.02380663, -0.01860054, -0.02184186,  0.03885571,
        -0.00957522, -0.00723644, -0.00600427, -0.00536228,  0.01449386,
        -0.00569397,  0.05645052,  0.03401814, -0.02619813,  0.00871371,
        -0.02000757, -0.01051402, -0.00600493,  0.03218004,  0.02556206,
         0.00035873, -0.0249726 , -0.00600472,  0.01188981,  0.02483101,
        -0.01875519,  0.03655513, -0.01807713,  0.0261963 , -0.01062996,
        -0.01615136,  0.00185746,  0.02275585], dtype=float32)]
{'name': 'dense_9', 'trainable': True, 'dtype': 'float32', 'units': 27,
'activation': 'softmax', 'use_bias': True, 'kernel_initializer': {'clas
s_name': 'GlorotUniform', 'config': {'seed': None}}, 'bias_initialize
r': {'class_name': 'Zeros', 'config': {}}, 'kernel_regularizer': None,
'bias_regularizer': None, 'activity_regularizer': None, 'kernel_constra
int': None, 'bias_constraint': None} [array([[-0.18956852, -0.12878886,
-0.13308418, ..., -0.19249791,
          0.15416545,  0.11489559],
        [-0.19633599, -0.00625877, -0.2623217 , ...,  0.08538906,
         -0.18273   , -0.30853972],
        [-0.11087029,  0.16754349,  0.1463467 , ..., -0.08401053,
         -0.04022557,  0.10567338],
        ...,
        [ 0.04598832,  0.11488138, -0.18002984, ...,  0.11282329,
          0.00366834, -0.14849477],
        [-0.11026984, -0.28745806,  0.01281498, ...,  0.00085452,
          0.07904769, -0.10444328],
        [ 0.09512308,  0.1621041 , -0.09759017, ..., -0.06388108,
         -0.2179139 ,  0.0881637 ]], dtype=float32), array([-0.02759628,
-0.00595153, -0.0305569 ,  0.04634005, -0.02396287,
        -0.00543478,  0.04288201,  0.01470611,  0.0116201 ,  0.05414284,
         0.05297694, -0.00332547,  0.02356804, -0.02277661,  0.00214141,
        -0.01522343, -0.02662523, -0.02751798, -0.02260997,  0.00063586,
        -0.02574153,  0.0004231 ,  0.02960827, -0.0428015 , -0.04717439,
         0.03066854, -0.01553019], dtype=float32)]
```

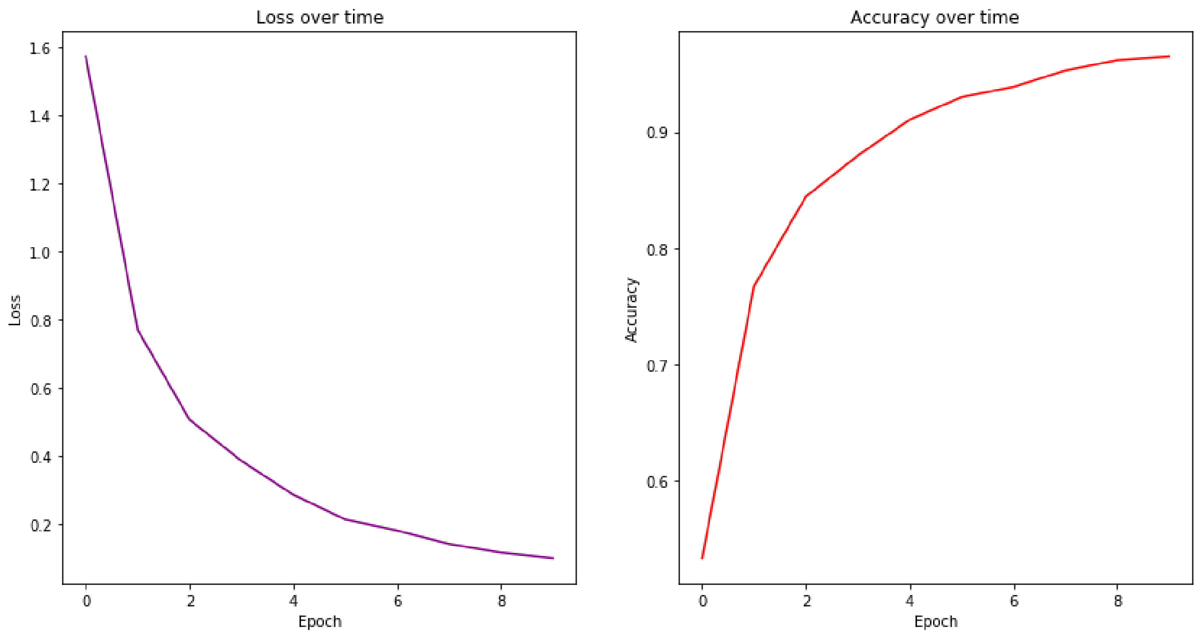
Testing the Model

In [67]:
```python
print("Evaluating on Test Data")
results = model.evaluate(X_test, y_test, batch_size=b_size,verbose=0)
print("Test Accuracy:", results[1])
```

```
Evaluating on Test Data
Test Accuracy: 0.8246753215789795
```

Visualizing the Performance

In [69]:
```
# accuracy vs epochs
plt.figure(figsize=(14,7))

plt.subplot(1,2,1)
plt.plot(history.history['loss'], c='purple')
plt.title('Loss over time')
plt.ylabel('Loss')
plt.xlabel('Epoch')

plt.subplot(1,2,2)
plt.plot(history.history['accuracy'], c='red')
plt.title('Accuracy over time')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.show()
```



END OF ASSIGNMENT

Author: Rudraksh Kapil - 177154

Thank you for reading :)