**CSCE 5215 - MACHINE LEARNING**

**FINAL PROJECT**

PRATHYUSHA JAMPALA

11648909

I have used three deep learning models in this project. Each model is trained on training images and tested on testing images of the fashion\_mnist dataset. I have imported the fashion\_mnist dataset from keras datasets.

The three models used in this project are multilayer Perceptron(MLP) model, Convolutional Neural Network(CNN) model and Simple Recurrent Neural Network (Simple RNN) model.

**Model 1 : Multilayer Perceptron Model**

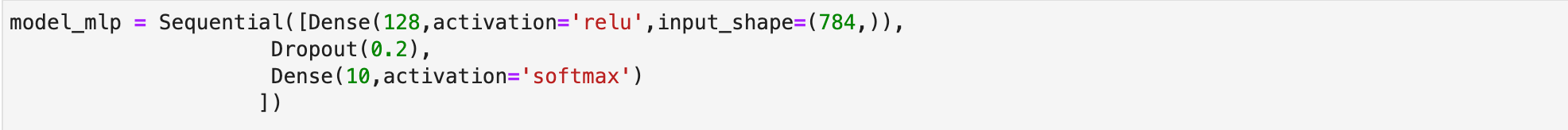
First load the fashion MNIST data into training and test variables.Normalize the data according to the pixel values by didviding train and test data by 255 to improve the quality of data.Also reshape the X\_train and X\_test data into one dimensional arrays.

Now the data is ready, we can implement the model.

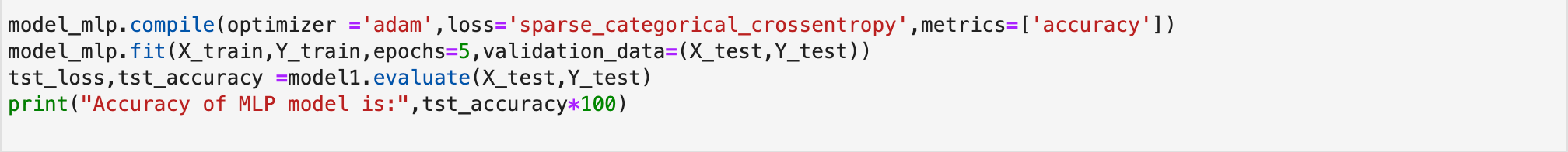


Our first model is multilayer perceptron. For this model, we have imported Dense and Dropout layers from keras of tensorflow.

First we are initializig a Sequential model which has one hideen layer, a dropout layer and a final output layer.Layers are connected with each other having input and output.The first layer of mlp sequential model is Dense layer with relu as activation function of the model with 128 number of units and input shape to (784,). The Dense layer is followed by dropout layer with rate as 0.2which means 20% input is assigned as 0 randomly at every update.The next layer is Dense layer with 10 units and activation function as softmax.



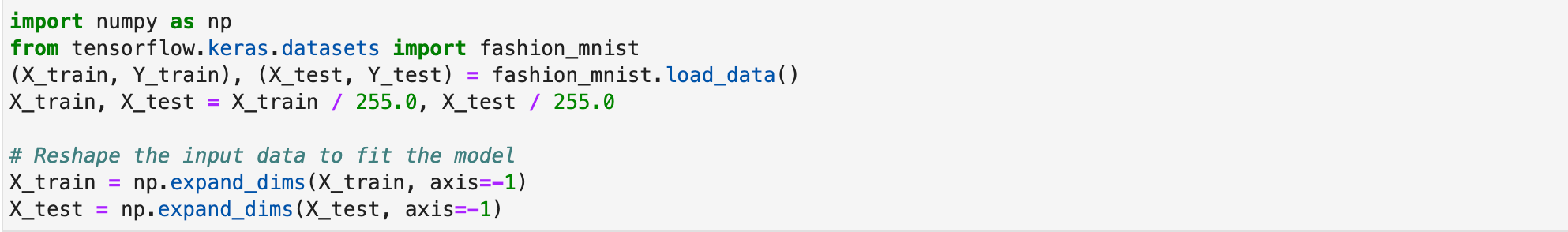
Next step in mlp implementation is compiling the model.We compile the moedl using adam optimization algorithm and sparse categorical cross entropy as loss function to predict the different between actual and predicted data.Now, fit the data using training data and run 5 forward and backward passes on the data in the mlp model by specifying number of epochs and check the validata on testing data.



Now, evaluate the performance of model using accuracy and loss function using evaluate() function on the model.The accuracy of the model when evaluated is given as 87.57 which means the multilayer perceptron model can classify the images 87% correctly.

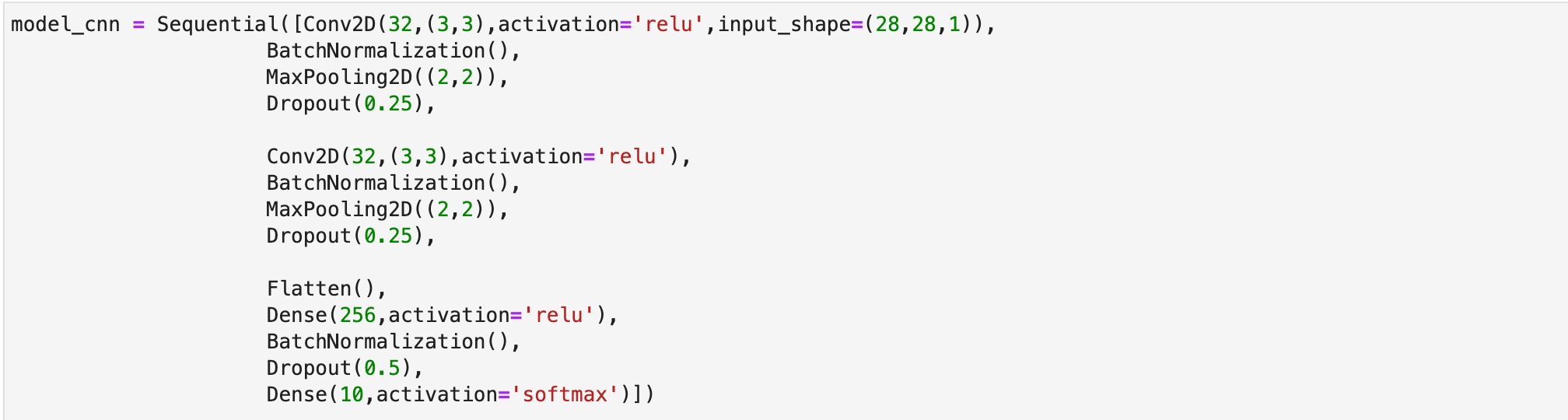
**Model 2: Computational Neural Networks**

First load the fashion MNIST data into training and test variables.Normalize the data according to the pixel values by didviding train and test data by 255 to improve the quality of data.Also reshape the X\_train and X\_test data to fit the model.



Our second model is Computational Neural Networks. For this model, we have imported Conv2D,BatchNormalization, Maxpooling2D, Flatten, Dense and Dropout layers from keras of tensorflow and Sequential model of models from keras of tensorflow.

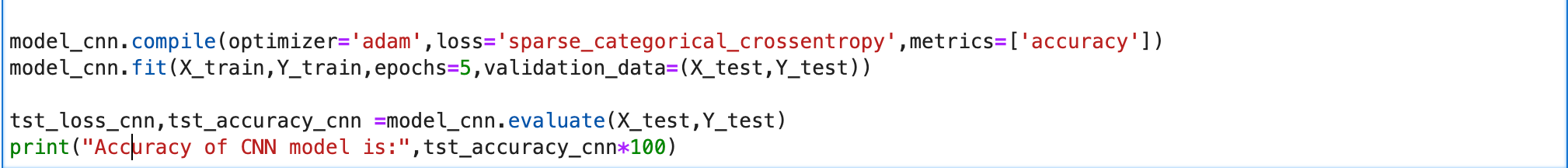
First we are initializing a Sequential model with conv2D as first layer with 332 units and 3X3 size with relu as its activation function and input shape as (28,28,1) followed by batchNormalization layer which normalizes the data foolwed by Maxpooling2D layer in size (2,2) and dropout layer with 0.25 dropout which means 25% input is assigned as 0 randomly at every update.



Add another 2D convolutional layer which is similar to first convolutional layer. Next we added a Flatten layer which converts multidimensional data into one dimension.Next we add a completely connected dense layer with 256 units and relu function followed by batch normalization followed by dropout layer with 0.5 dropout which means 50% input is assigned as 0 randomly at every update and final dense layer with 10 units and softmax function.

Next step in CNN implementation is compiling the model.We compile the model using adam optimization algorithm and sparse categorical cross entropy as loss function to predict the different between actual and predicted data.Now, fit the data using training data and run 5 forward and backward passes on the data in the CNN model by specifying number of epochs and check the validation on testing data.

Now, evaluate the performance of model using accuracy and loss function using evaluate() function on the model.The accuracy of the model when evaluated is given as 89.70 which means the Convolutional Neural Networks model can classify the images 89% correctly.



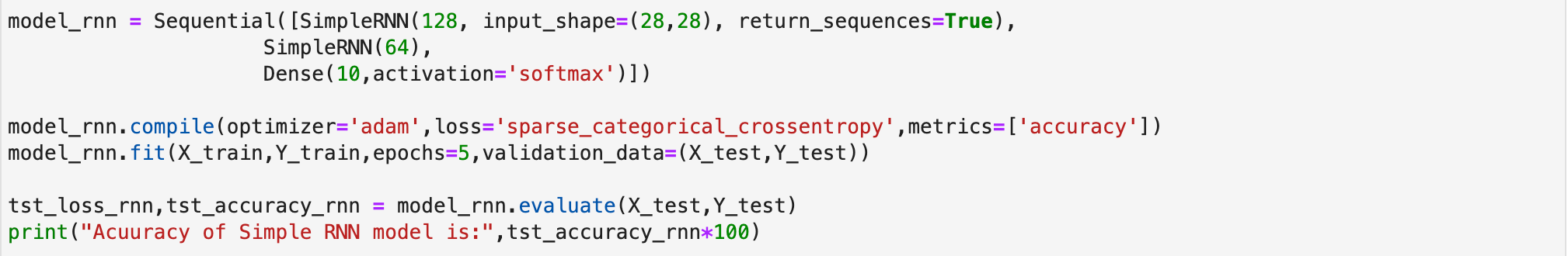
**Model 3 : Simple Recurrent Neural Networks**

First load the fashion MNIST data into training and test variables.Normalize the data according to the pixel values by didviding train and test data by 255 to improve the quality of data.Also reshape the X\_train and X\_test data to fit the model.



Our second model is Simple Recurrent Neural Networks. For this model, we have imported SimpleRNN and Dense layers from keras of tensorflow and Sequential model of models from keras of tensorflow.

First we are initializing a Sequential model with SimpleRNN as firs layer with 128 units , input shape as (28,28) and make sure sequences by giving return sequences as true are true as the next simple RNN layer accepts sequential data.The next layer is also a simple RNN which has 64 units and takes a sequential inputs.This layer is followed by dense layer with 10 units as softmax as activation function which is the final layer of the simple RNN model.



Next step in Simple RNN implementation is compiling the model.We compile the model using adam optimization algorithm and sparse categorical cross entropy as loss function to predict the different between actual and predicted data.Now, fit the data using training data and run 5 forward and backward passes on the data in the Simple RNN model by specifying number of epochs and check the validation on testing data.

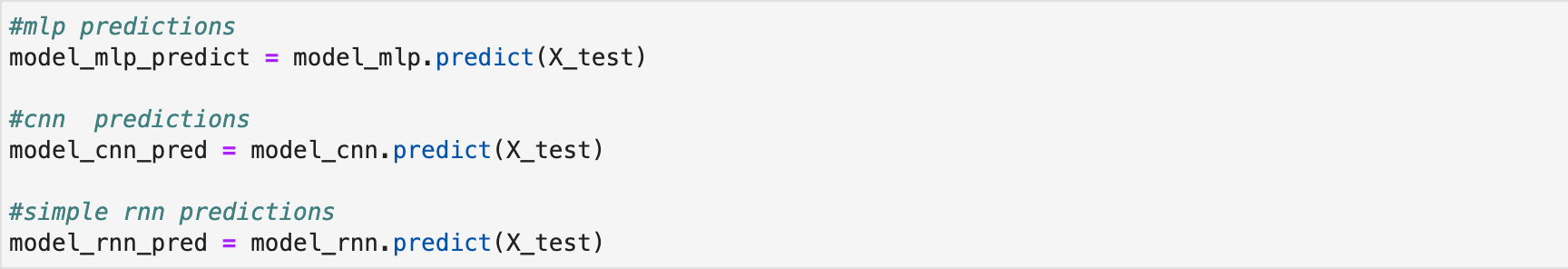
Now, evaluate the performance of model using accuracy and loss function using evaluate() function on the model.The accuracy of the model when evaluated is given as 79.83 which means the Simple RNN model can classify the images approximately 80% correctly.

**Combining Deep Learning models:**

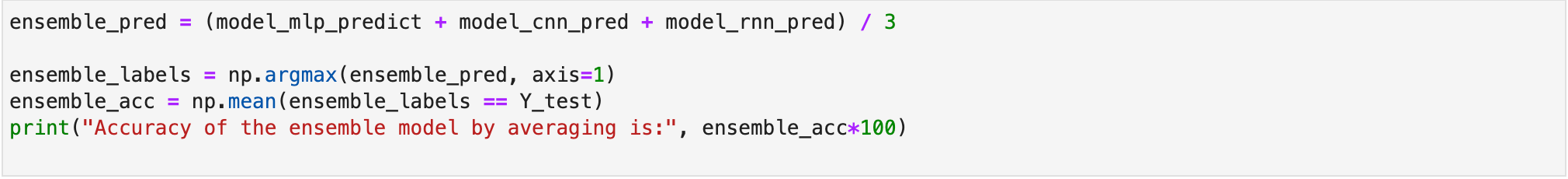
The models can be combined by averaging or multiplying the output of probability values of the three individual deep learning models(MLP, CNN and Simple RNN) we have implemented. We have implemented each model individually and trained the models using fashion mnist data and made predictions. In forming a committee we can combine the models using average or by multiplication.

**Committe of Models by Averaging:**

For designing the committe of MLP, CNN and Simple RNN, first we need to find the predictions of each model on the test data.

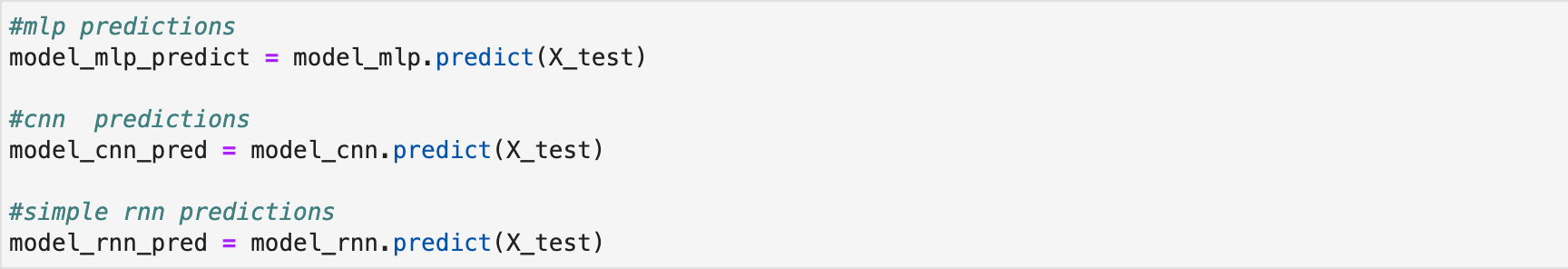


Next step is to find average of predictions made by each deep neural network model and then extract the high probability of each data using argmax() function from numpy libaray.Now match the predicted labels with the actual labels in test data to find the accuracy of the committee.It returns true if predicted labels are actual labels and fasle if not. And we get the accuracy of ensemble model by averaging MLP, CNN and Simple RNN as 89.15% which means the ensemble model can classify the images approximately 89% correctly.

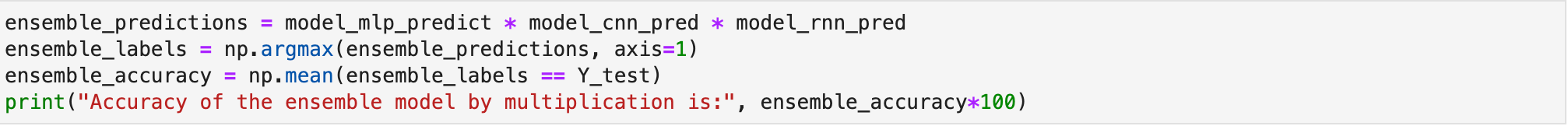


**Committe of Models by Multiplication:**

For designing the committe of MLP, CNN and Simple RNN, first we need to find the predictions of each model on the test data.



Next step is model make multiplication element wise to make predictions for each class made by each deep neural network model and and then extract the high probability of each data using argmax() function from numpy libaray.Now match the predicted labels with the actual labels in test data to find the accuracy of the committee.It returns true if predicted labels are actual labels and fasle if not. And we get the accuracy of ensemble model by averaging MLP, CNN and Simple RNN as 89.04% which means the ensemble model by multiplication can classify the images approximately 89% correctly.



**Accuracy , Precision, Recall and F1 score of implemented models :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 score** |
| Multi Layer Perceptron | 87.35 | 87.45 | 87.35 | 87.32 |
| Convolutional Neural Network | 89.96 | 89.93 | 89.96 | 89.85 |
| Simple RNN | 83.92 | 83.79 | 83.92 | 83.75 |
| Committe by averaging | 89.21 | 89.16 | 89.21 | 89.10 |
| committe by multiplication | 89.32 | 89.28 | 89.32 | 89.22 |