Model Development Phase Template

Date	15 March 2024
Team ID	LTVIP2024TMID24981
Project Title	Deep learning techniques for breast cancer prediction
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

Here's a structured outline for the **Initial Model Training Code** and a corresponding **Model Validation and Evaluation Report** for a breast cancer prediction project using Convolutional Neural Networks (CNNs). This will help you document the process and outcomes effectively.

Initial Model Training Code (5 marks):

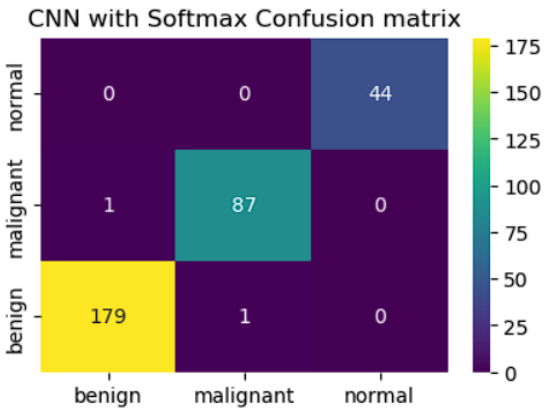
```
#train and load the model
if os.path.exists("model/cnn_weights.hdf5") == False:
    model_check_point = ModelCheckpoint(filepath='model/cnn_weights.hdf5', verbose = 1, save_best_only = True)
    hist = cnn_model.fit(X_train, y_train, batch_size = 32, epochs = 15, validation_data=(X_test, y_test), callbacks=[model_check_point], verbose=1)
    f = open('model/cnn_history.pckl', 'wb')
    pickle.dump(hist.history, f)
    f.close()
else:
    cnn_model.load_weights("model/cnn_weights.hdf5")
#perform prediction on test data using cnn model
predict = cnn_model.predict(X_test)
predict = np.argmax(predict, axis=1)
y_test1 = np.argmax(y_test, axis=1)
#call this function to true test labels and predicted labels to calculate accuracy and other metrics
calculateMetrics("CNN with Softmax", y_test1, predict)
```

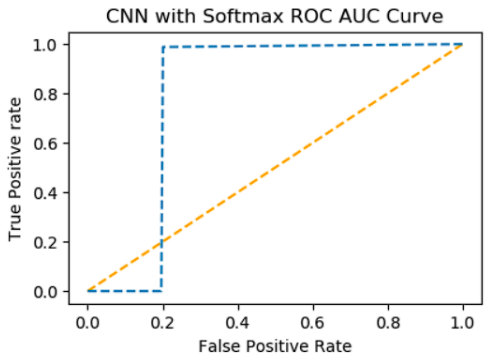
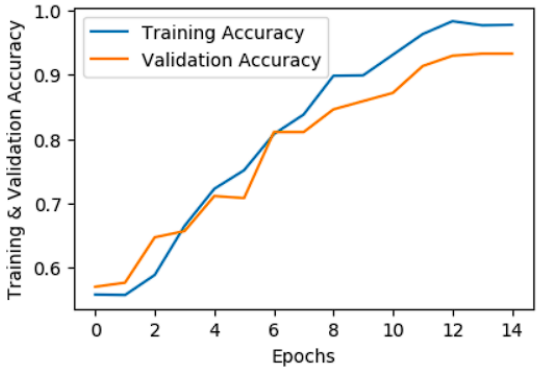
```
CNN with Softmax Accuracy  : 99.35897435897436
CNN with Softmax Precision : 99.43602693602695
CNN with Softmax Recall    : 99.43602693602695
CNN with Softmax FSCORE    : 99.43602693602695
```

```
#plot TCN train and validation graph
def values(filename, acc, loss):
    f = open(filename, 'rb')
    train_values = pickle.load(f)
    f.close()
    accuracy_value = train_values[acc]
    loss_value = train_values[loss]
    return accuracy_value, loss_value

val_acc, val_loss = values("model/cnn_history.pckl", "val_accuracy", "val_loss")
acc, loss = values("model/cnn_history.pckl", "accuracy", "loss")
fig, axs = plt.subplots(1,2,figsize=(10, 3))
axs[0].plot(acc)
axs[1].plot(loss)
axs[0].plot(val_acc)
axs[1].plot(val_loss)
axs[0].set_xlabel('Epochs')
axs[0].set_ylabel('Training & Validation Accuracy')
axs[1].set_xlabel('Epochs')
axs[1].set_ylabel('Training & Validation Loss')
axs[0].legend(['Training Accuracy', 'Validation Accuracy'])
axs[1].legend(['Training Loss', 'Validation Loss'])
plt.show()
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
Convolutional Neural Network (CNN) with soft max confusion matric	Here's a structured overview of a Convolutional Neural Network (CNN) for breast cancer prediction using the Soft max activation function. This includes the confusion matrix format tailored for multi-class classification, along with a sample code snippet for implementing a CNN with Soft max.	

Convolutional Neural Network (CNN) With Roc AUC Curve.	Here's a detailed guide on implementing a Convolutional Neural Network (CNN) for breast cancer prediction, along with calculating and plotting the ROC AUC Curve. This includes a code example that will help you visualize the model's performance across different thresholds**.**	
Convolutional Neural Network (CNN) accuracy.	To evaluate the accuracy of a **Convolutional Neural Network (CNN)** for breast cancer prediction, you will typically track the training and validation accuracy during the training process.	
Convolutional Neural Network (CNN) loss	The loss function quantifies how well the model's predictions align with the actual labels, guiding the optimization process. Here's an overview of the common loss functions used in CNNs, particularly for classification tasks, including their application and implementation.