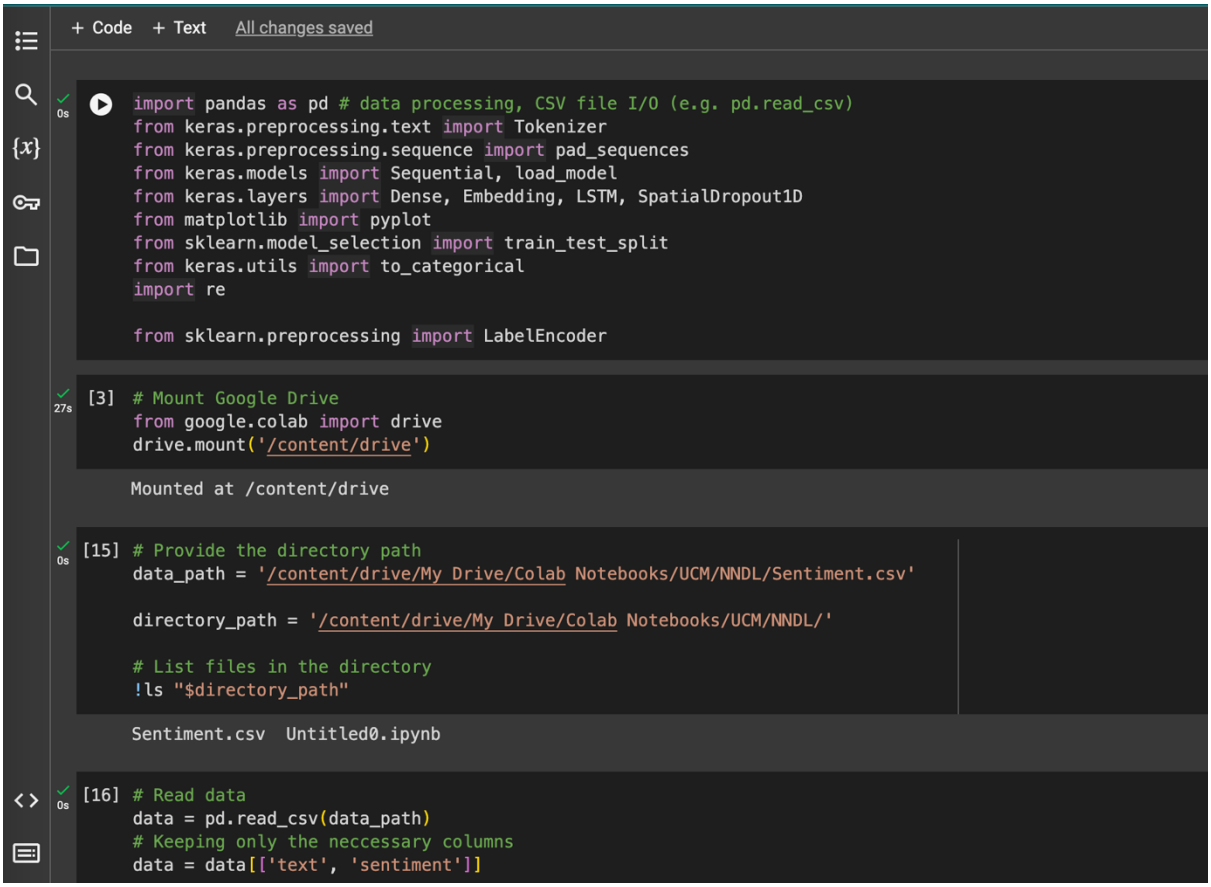
CS 5720 Neural Network Deep Learning
ICP-9

CRN	23441
NAME	MANOJ BALA
EMAIL	mxb40210@ucmo.edu
STUDENT_ID	700754021

GitHub Repository:
https://github.com/mxb40210/700754021-NeuralNetworkDeepLearning

Assignment 9:
https://github.com/mxb40210/700754021-NeuralNetworkDeepLearning/tree/main/assignments/assignment9

Screenshots:



+ Code + Text All changes saved

```
[17] # Process data
     data['text'] = data['text'].apply(lambda x: x.lower())
     data['text'] = data['text'].apply((lambda x: re.sub('[^a-zA-z0-9\s]', '', x)))

     for idx, row in data.iterrows():
         row[0] = row[0].replace('rt', ' ')

     max_fatures = 2000
     tokenizer = Tokenizer(num_words=max_fatures, split=' ')
     tokenizer.fit_on_texts(data['text'].values)
     X = tokenizer.texts_to_sequences(data['text'].values)
     X = pad_sequences(X)
```

```
# Model
embed_dim = 128
lstm_out = 196
def createmodel():
    model = Sequential()
    model.add(Embedding(max_fatures, embed_dim,input_length = X.shape[1]))
    model.add(LSTM(lstm_out, dropout=0.2, recurrent_dropout=0.2))
    model.add(Dense(3,activation='softmax'))
    model.compile(loss = 'categorical_crossentropy', optimizer='adam',metrics = ['accuracy'])
    return model
# print(model.summary())
```

```
[20] # Encode and test train split data
     labelencoder = LabelEncoder()
     integer_encoded = labelencoder.fit_transform(data['sentiment'])
     y = to_categorical(integer_encoded)
     X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size = 0.33, random_state = 42)
```

+ Code + Text All changes saved

```
[20] # Encode and test train split data
     labelencoder = LabelEncoder()
     integer_encoded = labelencoder.fit_transform(data['sentiment'])
     y = to_categorical(integer_encoded)
     X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size = 0.33, random_state = 42)
```

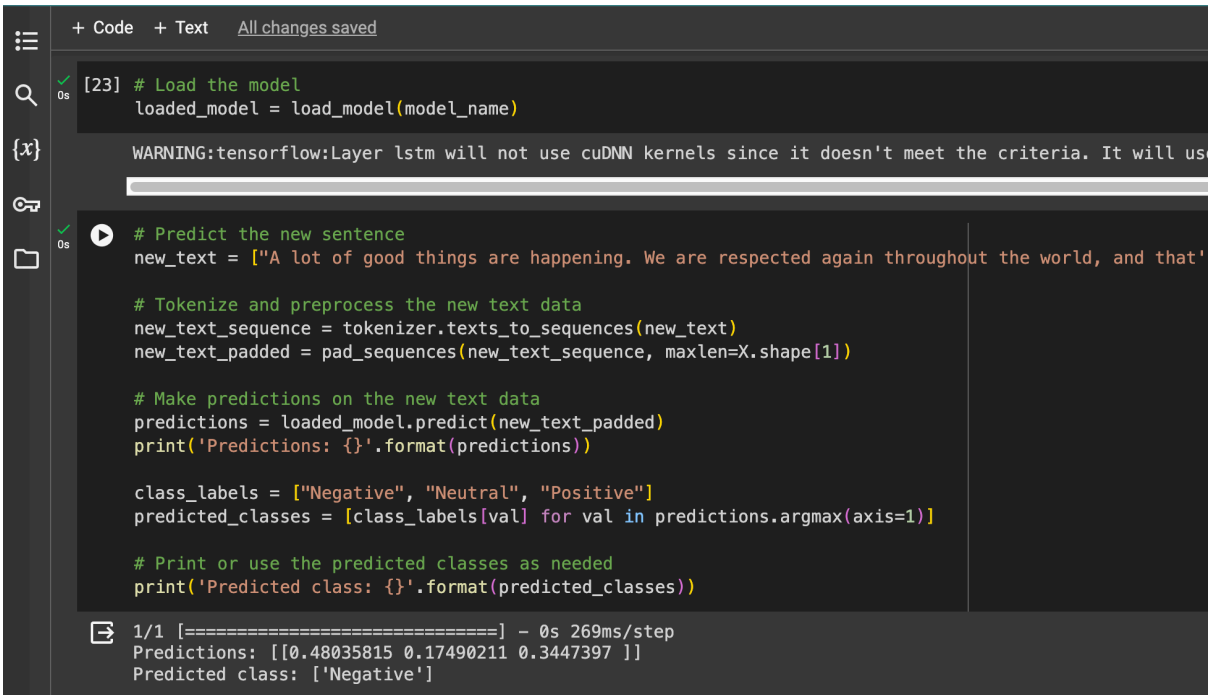
```
[21] # Create & Fit Model
     batch_size = 32
     model = createmodel()
     model.fit(X_train, Y_train, epochs = 1, batch_size=batch_size, verbose = 2)
     score, acc = model.evaluate(X_test,Y_test,verbose=2,batch_size=batch_size)
     print('Score: {}, Accuracy: {}'.format(score, acc))
     print('Model metric_names: {}'.format(model.metrics_names))
```

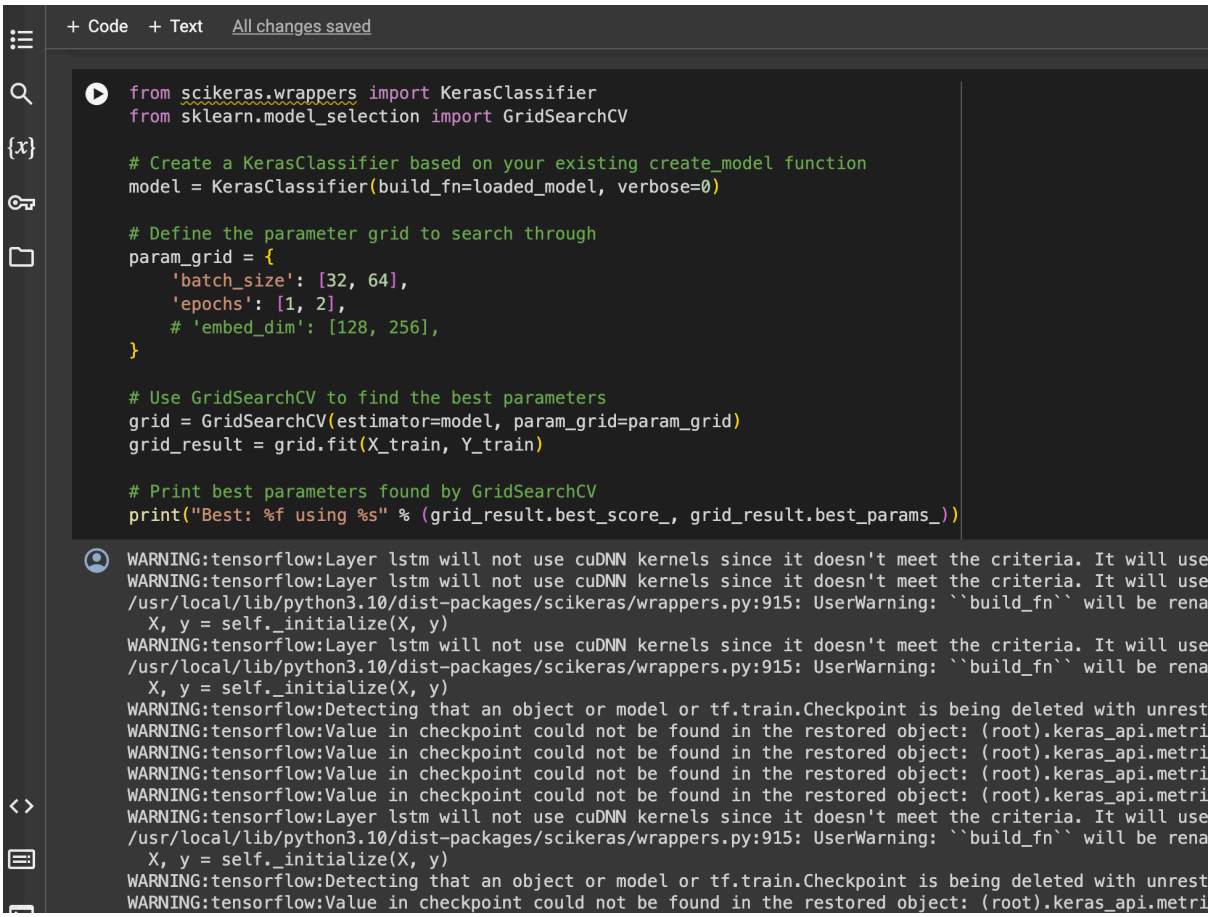
```
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use
291/291 - 41s - loss: 0.8246 - accuracy: 0.6449 - 41s/epoch - 139ms/step
144/144 - 2s - loss: 0.7513 - accuracy: 0.6778 - 2s/epoch - 12ms/step
Score: 0.7512960433959961, Accuracy: 0.6778069138526917
Model metric_names: ['loss', 'accuracy']
```

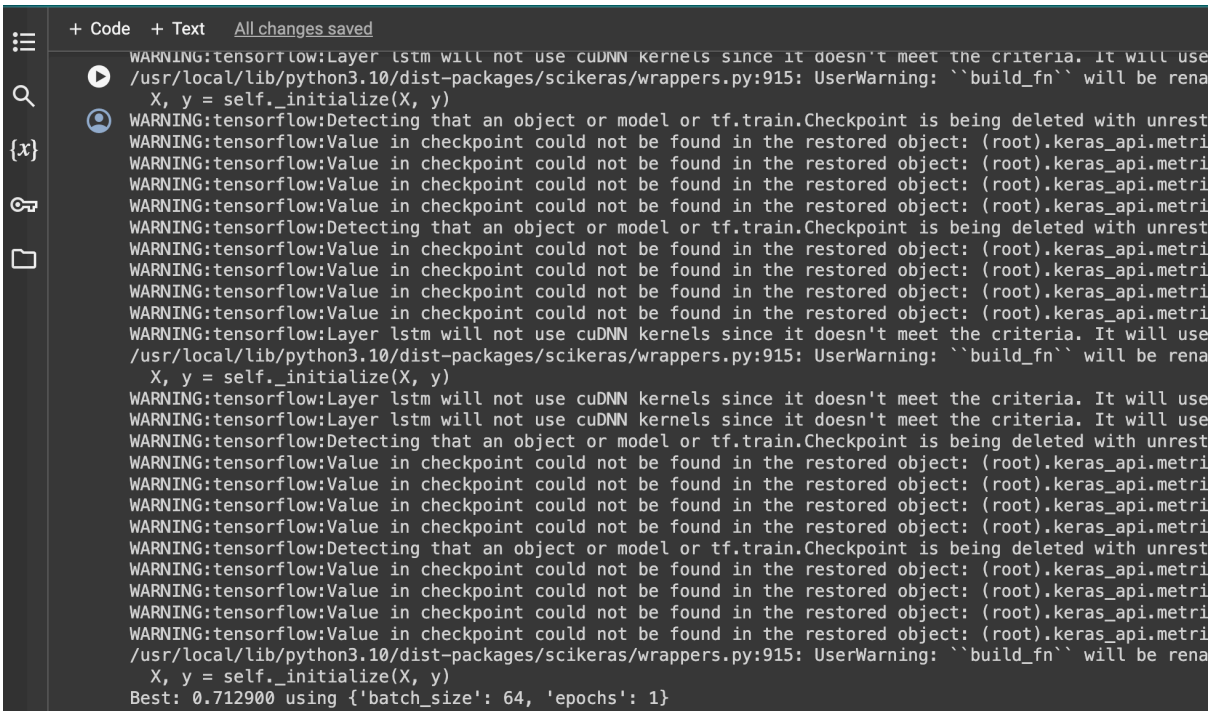
```
# Saving the model
model_name = 'LSTM_MODEL.keras'
model.save(model_name)
```

```
[23] # Load the model
     loaded_model = load_model(model_name)
```

```
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use
```


+ Code
+ Text
All changes saved
[23] # Load the model
loaded_model = load_model(model_name)
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will us
# Predict the new sentence
new_text = ["A lot of good things are happening. We are respected again throughout the world, and that'
# Tokenize and preprocess the new text data
new_text_sequence = tokenizer.texts_to_sequences(new_text)
new_text_padded = pad_sequences(new_text_sequence, maxlen=X.shape[1])
# Make predictions on the new text data
predictions = loaded_model.predict(new_text_padded)
print('Predictions: {}'.format(predictions))
class_labels = ["Negative", "Neutral", "Positive"]
predicted_classes = [class_labels[val] for val in predictions.argmax(axis=1)]
# Print or use the predicted classes as needed
print('Predicted class: {}'.format(predicted_classes))
1/1 [==============================] - 0s 269ms/step
Predictions: [[0.48035815 0.17490211 0.3447397]]
Predicted class: ['Negative']


+ Code
+ Text
All changes saved
from scikeras.wrappers import KerasClassifier
from sklearn.model_selection import GridSearchCV
# Create a KerasClassifier based on your existing create_model function
model = KerasClassifier(build_fn=loaded_model, verbose=0)
# Define the parameter grid to search through
param_grid = {
'batch_size': [32, 64],
'epochs': [1, 2],
# 'embed_dim': [128, 256],
}
# Use GridSearchCV to find the best parameters
grid = GridSearchCV(estimator=model, param_grid=param_grid)
grid_result = grid.fit(X_train, Y_train)
# Print best parameters found by GridSearchCV
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be rena
X, y = self._initialize(X, y)
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be rena
X, y = self._initialize(X, y)
WARNING:tensorflow:Detecting that an object or model or tf.train.Checkpoint is being deleted with unrest
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be rena
X, y = self._initialize(X, y)
WARNING:tensorflow:Detecting that an object or model or tf.train.Checkpoint is being deleted with unrest
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri


+ Code
+ Text
All changes saved
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be rena
X, y = self._initialize(X, y)
WARNING:tensorflow:Detecting that an object or model or tf.train.Checkpoint is being deleted with unrest
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Detecting that an object or model or tf.train.Checkpoint is being deleted with unrest
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be rena
X, y = self._initialize(X, y)
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use
WARNING:tensorflow:Detecting that an object or model or tf.train.Checkpoint is being deleted with unrest
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Detecting that an object or model or tf.train.Checkpoint is being deleted with unrest
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).keras_api.metri
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be rena
X, y = self._initialize(X, y)
Best: 0.712900 using {'batch_size': 64, 'epochs': 1}