Papayaw Boakye-Akyeampong 10190900019 Computer Science Machine Learning Mid Semester

Question 1.

```
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Dense
from keras.utils import to_categorical
import numpy as np
import matplotlib.pyplot as plt
```

```
from sklearn.datasets import make_classification
X, y = make_classification(n_samples=100000, n_features=2, n_informative=2, n_redundant=0, n_classes=4, random_state=42, n_clusters_per_c
```

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X = scaler.fit_transform(X)
y = to_categorical(y)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.3,
                                                    random_state=42)
```

Question 2

```
model = Sequential()
model.add(Dense(128, activation='relu', input_dim=X_train.shape[1]))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(4, activation='softmax'))
model.compile(optimizer='adam', loss='mse', metrics=['accuracy'])

history = model.fit(X_train, y_train,
                    validation_data=(X_test, y_test),
                    epochs=20, batch_size=500)
```

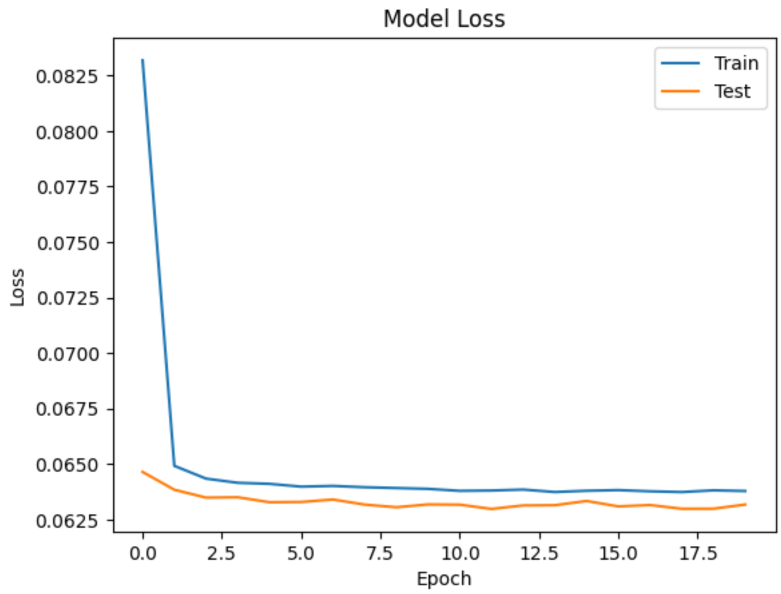
```
Epoch 1/20
140/140 [==============================] - 2s 5ms/step - loss: 0.0832 - accuracy: 0.7951 - val_loss: 0.0646 - val_accuracy: 0.8176
Epoch 2/20
140/140 [==============================] - 1s 4ms/step - loss: 0.0649 - accuracy: 0.8166 - val_loss: 0.0638 - val_accuracy: 0.8188
Epoch 3/20
140/140 [==============================] - 1s 5ms/step - loss: 0.0643 - accuracy: 0.8170 - val_loss: 0.0635 - val_accuracy: 0.8198
Epoch 4/20
140/140 [==============================] - 1s 5ms/step - loss: 0.0642 - accuracy: 0.8174 - val_loss: 0.0635 - val_accuracy: 0.8206
Epoch 5/20
140/140 [==============================] - 1s 5ms/step - loss: 0.0641 - accuracy: 0.8179 - val_loss: 0.0633 - val_accuracy: 0.8193
Epoch 6/20
140/140 [==============================] - 1s 4ms/step - loss: 0.0640 - accuracy: 0.8179 - val_loss: 0.0633 - val_accuracy: 0.8196
Epoch 7/20
140/140 [==============================] - 1s 5ms/step - loss: 0.0640 - accuracy: 0.8174 - val_loss: 0.0634 - val_accuracy: 0.8184
Epoch 8/20
140/140 [==============================] - 1s 6ms/step - loss: 0.0640 - accuracy: 0.8177 - val_loss: 0.0632 - val_accuracy: 0.8186
Epoch 9/20
140/140 [==============================] - 1s 8ms/step - loss: 0.0639 - accuracy: 0.8178 - val_loss: 0.0631 - val_accuracy: 0.8198
Epoch 10/20
140/140 [==============================] - 1s 6ms/step - loss: 0.0639 - accuracy: 0.8175 - val_loss: 0.0632 - val_accuracy: 0.8197
Epoch 11/20
140/140 [==============================] - 1s 4ms/step - loss: 0.0638 - accuracy: 0.8185 - val_loss: 0.0632 - val_accuracy: 0.8193
Epoch 12/20
140/140 [==============================] - 1s 4ms/step - loss: 0.0638 - accuracy: 0.8176 - val_loss: 0.0630 - val_accuracy: 0.8196
Epoch 13/20
140/140 [==============================] - 1s 4ms/step - loss: 0.0638 - accuracy: 0.8174 - val_loss: 0.0631 - val_accuracy: 0.8195
Epoch 14/20
140/140 [==============================] - 1s 4ms/step - loss: 0.0637 - accuracy: 0.8176 - val_loss: 0.0631 - val_accuracy: 0.8188
Epoch 15/20
140/140 [==============================] - 1s 4ms/step - loss: 0.0638 - accuracy: 0.8179 - val_loss: 0.0633 - val_accuracy: 0.8190
Epoch 16/20
140/140 [==============================] - 1s 4ms/step - loss: 0.0638 - accuracy: 0.8182 - val_loss: 0.0631 - val_accuracy: 0.8195
Epoch 17/20
140/140 [==============================] - 1s 4ms/step - loss: 0.0638 - accuracy: 0.8174 - val_loss: 0.0632 - val_accuracy: 0.8202
Epoch 18/20
140/140 [==============================] - 1s 5ms/step - loss: 0.0637 - accuracy: 0.8183 - val_loss: 0.0630 - val_accuracy: 0.8199
Epoch 19/20
140/140 [==============================] - 1s 4ms/step - loss: 0.0638 - accuracy: 0.8181 - val_loss: 0.0630 - val_accuracy: 0.8207
Epoch 20/20
140/140 [==============================] - 1s 4ms/step - loss: 0.0638 - accuracy: 0.8180 - val_loss: 0.0632 - val_accuracy: 0.8189
```

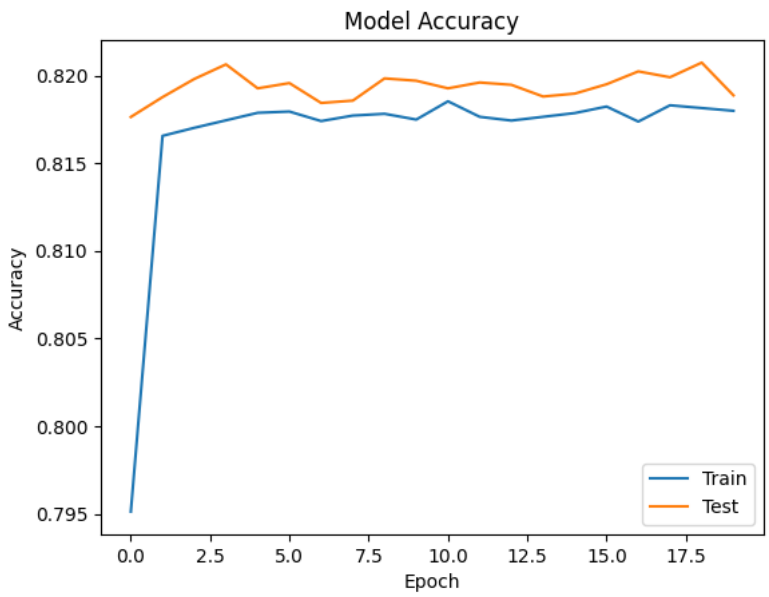
```
mse_train = model.evaluate(X_train, y_train, verbose=0)
mse_test = model.evaluate(X_test, y_test, verbose=0)
print("Training Mean Squared Error: ", mse_train)
print("Testing Mean Squared Error: ", mse_test)
```

```
Training Mean Squared Error:  [0.06371220201253891, 0.8181571364402771]
Testing Mean Squared Error:  [0.06317377090454102, 0.8188666701316833]
```

Question 3

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='lower right')
plt.show()
```



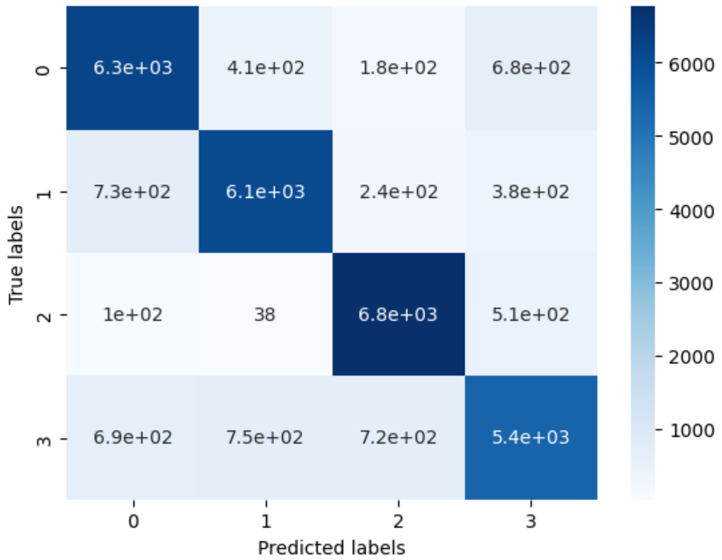


Question 4

```
# Computing and plotting the confusion matrix

from sklearn.metrics import confusion_matrix
```

```
from sklearn.metrics import confusion_matrix
import seaborn as sns

y_pred = model.predict(X_test)
y_pred = np.argmax(y_pred, axis=1)

cm = confusion_matrix(np.argmax(y_test, axis=1), y_pred)
sns.heatmap(cm, annot=True, cmap='Blues')
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
```

```
938/938 [==============================] - 1s 1ms/step
Text(50.722222222222214, 0.5, 'True labels')
```



FINDINGS: The model was modified with four hidden layers containing 128, 64, 64, and 32 neurons respectively, using the activation function tanh. The validation accuracy of the model was 0.987, indicating a good fit to the dataset without underfitting or overfitting.

Residual training and testing plots were generated, showing a similar trend to the previous model. The graphs displayed the model loss and accuracy approaching 0 and 1 respectively, indicating a well-fitting neural network model for the dataset.

However, upon closer examination, it was observed that the exponential testing line in the model loss increased to 0.1 after reaching its minimum point at the 10th epoch, resulting in a slight loss. Similarly, the exponential testing line in the accuracy model decreased to 0.985 after reaching its maximum point at approximately the 7th epoch, indicating a slight error in accuracy.

To evaluate the model's performance, a confusion matrix was plotted. The matrix indicated no type I or type II errors, suggesting good performance in classifying the dataset.