

# Task\_1.3C\_BalramNaidu(220580881)

July 26, 2022

####Github Link:

Welcome to your assignment this week!

## 1 Classification task

In this task you are asked to build a simple Feed Forward Neural Network, train it and test it!

**After this assignment you will be able to:**

- Load a dataset.
- Train a Feed Forward Neural Network.
- Test a Feed Forward Neural Network.

Let's get started! Run the following cell to install all the packages you will need.

```
[1]: #!pip install numpy
#!pip install keras
#!pip install tensorflow
#!pip install pandas
#!pip install matplotlib
```

if you are using GoogleColab, please install the following packages and mount your Google drive:

```
[2]: # !apt-get install texlive-xetex texlive-fonts-recommended
↪texlive-generic-recommended 2> /dev/null > /dev/null
# !apt-get install pandoc 2> /dev/null > /dev/null

# from google.colab import drive
# drive.mount('/content/drive')
```

Run the following cell to load the packages you will need.

```
[3]: import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import keras
from keras.models import Sequential
from keras.layers import Dense
```

The dataset we will use consists of 4500 examples with 512 features. A label is given for each example to indicate positive and negative instances.

Let's read the data.

```
[4]: df = pd.read_csv('data.csv')
      df.set_index('id', inplace=True)
```

Now, let's split the data into training and test sets.

```
[5]: X_train, X_test, y_train, y_test = train_test_split(
      df.index.values,
      df.label.values,
      test_size=0.15,
      random_state=17,
      stratify=df.label.values
    )
    df['data_type'] = ['note_set']*df.shape[0]
    df.loc[X_train, 'data_type'] = 'train'
    df.loc[X_test, 'data_type'] = 'test'

    ## The data to use:

    X_train = df[df['data_type']=='train'].iloc[:, :512].values
    X_test = df[df['data_type']=='test'].iloc[:, :512].values
    y_train = df[df['data_type']=='train'].iloc[:, 512:513].values
    y_test = df[df['data_type']=='test'].iloc[:, 512:513].values
```

## 2 Task 1

Build a Feed Forward Neural Network to address this classification task using the Keras framework.

```
[6]: # START YOUR CODE HERE
      model = Sequential() #Creating an empty sequential model

      #Adding layers to the sequential model
      model.add(Dense(180, activation='relu'))
      model.add(Dense(60, activation='relu'))
      model.add(Dense(1, activation='sigmoid'))
      model.compile(optimizer='adam', loss='binary_crossentropy',
        ↪metrics=['accuracy'])
```

## 3 Training

Now, let's start our training.

```
[7]: history = model.fit(X_train, y_train, epochs=200, batch_size=64, verbose=1)
```

Epoch 1/200  
60/60 [=====] - 1s 2ms/step - loss: 0.2508 - accuracy: 0.9061

Epoch 2/200  
60/60 [=====] - 0s 2ms/step - loss: 0.0089 - accuracy: 0.9995

Epoch 3/200  
60/60 [=====] - 0s 2ms/step - loss: 0.0022 - accuracy: 1.0000

Epoch 4/200  
60/60 [=====] - 0s 2ms/step - loss: 0.0011 - accuracy: 1.0000

Epoch 5/200  
60/60 [=====] - 0s 2ms/step - loss: 6.6164e-04 - accuracy: 1.0000

Epoch 6/200  
60/60 [=====] - 0s 2ms/step - loss: 4.5411e-04 - accuracy: 1.0000

Epoch 7/200  
60/60 [=====] - 0s 2ms/step - loss: 3.3099e-04 - accuracy: 1.0000

Epoch 8/200  
60/60 [=====] - 0s 2ms/step - loss: 2.5168e-04 - accuracy: 1.0000

Epoch 9/200  
60/60 [=====] - 0s 2ms/step - loss: 1.9715e-04 - accuracy: 1.0000

Epoch 10/200  
60/60 [=====] - 0s 2ms/step - loss: 1.5824e-04 - accuracy: 1.0000

Epoch 11/200  
60/60 [=====] - 0s 2ms/step - loss: 1.2944e-04 - accuracy: 1.0000

Epoch 12/200  
60/60 [=====] - 0s 2ms/step - loss: 1.0740e-04 - accuracy: 1.0000

Epoch 13/200  
60/60 [=====] - 0s 3ms/step - loss: 9.0319e-05 - accuracy: 1.0000

Epoch 14/200  
60/60 [=====] - 0s 3ms/step - loss: 7.6790e-05 - accuracy: 1.0000

Epoch 15/200  
60/60 [=====] - 0s 4ms/step - loss: 6.5853e-05 - accuracy: 1.0000

Epoch 16/200  
60/60 [=====] - 0s 4ms/step - loss: 5.6931e-05 - accuracy: 1.0000

Epoch 17/200  
60/60 [=====] - 0s 4ms/step - loss: 4.9530e-05 -  
accuracy: 1.0000  
Epoch 18/200  
60/60 [=====] - 0s 3ms/step - loss: 4.3391e-05 -  
accuracy: 1.0000  
Epoch 19/200  
60/60 [=====] - 0s 3ms/step - loss: 3.8169e-05 -  
accuracy: 1.0000  
Epoch 20/200  
60/60 [=====] - 0s 2ms/step - loss: 3.3786e-05 -  
accuracy: 1.0000  
Epoch 21/200  
60/60 [=====] - 0s 2ms/step - loss: 3.0022e-05 -  
accuracy: 1.0000  
Epoch 22/200  
60/60 [=====] - 0s 2ms/step - loss: 2.6772e-05 -  
accuracy: 1.0000  
Epoch 23/200  
60/60 [=====] - 0s 2ms/step - loss: 2.3973e-05 -  
accuracy: 1.0000  
Epoch 24/200  
60/60 [=====] - 0s 2ms/step - loss: 2.1557e-05 -  
accuracy: 1.0000  
Epoch 25/200  
60/60 [=====] - 0s 2ms/step - loss: 1.9428e-05 -  
accuracy: 1.0000  
Epoch 26/200  
60/60 [=====] - 0s 2ms/step - loss: 1.7576e-05 -  
accuracy: 1.0000  
Epoch 27/200  
60/60 [=====] - 0s 2ms/step - loss: 1.5943e-05 -  
accuracy: 1.0000  
Epoch 28/200  
60/60 [=====] - 0s 2ms/step - loss: 1.4501e-05 -  
accuracy: 1.0000  
Epoch 29/200  
60/60 [=====] - 0s 2ms/step - loss: 1.3231e-05 -  
accuracy: 1.0000  
Epoch 30/200  
60/60 [=====] - 0s 2ms/step - loss: 1.2101e-05 -  
accuracy: 1.0000  
Epoch 31/200  
60/60 [=====] - 0s 2ms/step - loss: 1.1087e-05 -  
accuracy: 1.0000  
Epoch 32/200  
60/60 [=====] - 0s 3ms/step - loss: 1.0181e-05 -  
accuracy: 1.0000

Epoch 33/200  
60/60 [=====] - 0s 3ms/step - loss: 9.3770e-06 -  
accuracy: 1.0000

Epoch 34/200  
60/60 [=====] - 0s 3ms/step - loss: 8.6466e-06 -  
accuracy: 1.0000

Epoch 35/200  
60/60 [=====] - 0s 3ms/step - loss: 7.9888e-06 -  
accuracy: 1.0000

Epoch 36/200  
60/60 [=====] - 0s 4ms/step - loss: 7.3940e-06 -  
accuracy: 1.0000

Epoch 37/200  
60/60 [=====] - 0s 3ms/step - loss: 6.8556e-06 -  
accuracy: 1.0000

Epoch 38/200  
60/60 [=====] - 0s 4ms/step - loss: 6.3651e-06 -  
accuracy: 1.0000

Epoch 39/200  
60/60 [=====] - 0s 3ms/step - loss: 5.9235e-06 -  
accuracy: 1.0000

Epoch 40/200  
60/60 [=====] - 0s 3ms/step - loss: 5.5166e-06 -  
accuracy: 1.0000

Epoch 41/200  
60/60 [=====] - 0s 2ms/step - loss: 5.1459e-06 -  
accuracy: 1.0000

Epoch 42/200  
60/60 [=====] - 0s 2ms/step - loss: 4.8055e-06 -  
accuracy: 1.0000

Epoch 43/200  
60/60 [=====] - 0s 2ms/step - loss: 4.4932e-06 -  
accuracy: 1.0000

Epoch 44/200  
60/60 [=====] - 0s 2ms/step - loss: 4.2071e-06 -  
accuracy: 1.0000

Epoch 45/200  
60/60 [=====] - 0s 2ms/step - loss: 3.9429e-06 -  
accuracy: 1.0000

Epoch 46/200  
60/60 [=====] - 0s 2ms/step - loss: 3.7011e-06 -  
accuracy: 1.0000

Epoch 47/200  
60/60 [=====] - 0s 2ms/step - loss: 3.4758e-06 -  
accuracy: 1.0000

Epoch 48/200  
60/60 [=====] - 0s 2ms/step - loss: 3.2697e-06 -  
accuracy: 1.0000

Epoch 49/200  
60/60 [=====] - 0s 2ms/step - loss: 3.0760e-06 -  
accuracy: 1.0000  
Epoch 50/200  
60/60 [=====] - 0s 2ms/step - loss: 2.8983e-06 -  
accuracy: 1.0000  
Epoch 51/200  
60/60 [=====] - 0s 2ms/step - loss: 2.7325e-06 -  
accuracy: 1.0000  
Epoch 52/200  
60/60 [=====] - 0s 2ms/step - loss: 2.5783e-06 -  
accuracy: 1.0000  
Epoch 53/200  
60/60 [=====] - 0s 3ms/step - loss: 2.4350e-06 -  
accuracy: 1.0000  
Epoch 54/200  
60/60 [=====] - 0s 4ms/step - loss: 2.3016e-06 -  
accuracy: 1.0000  
Epoch 55/200  
60/60 [=====] - 0s 3ms/step - loss: 2.1775e-06 -  
accuracy: 1.0000  
Epoch 56/200  
60/60 [=====] - 0s 3ms/step - loss: 2.0609e-06 -  
accuracy: 1.0000  
Epoch 57/200  
60/60 [=====] - 0s 3ms/step - loss: 1.9526e-06 -  
accuracy: 1.0000  
Epoch 58/200  
60/60 [=====] - 0s 4ms/step - loss: 1.8508e-06 -  
accuracy: 1.0000  
Epoch 59/200  
60/60 [=====] - 0s 3ms/step - loss: 1.7557e-06 -  
accuracy: 1.0000  
Epoch 60/200  
60/60 [=====] - 0s 4ms/step - loss: 1.6662e-06 -  
accuracy: 1.0000  
Epoch 61/200  
60/60 [=====] - 0s 2ms/step - loss: 1.5824e-06 -  
accuracy: 1.0000  
Epoch 62/200  
60/60 [=====] - 0s 2ms/step - loss: 1.5037e-06 -  
accuracy: 1.0000  
Epoch 63/200  
60/60 [=====] - 0s 2ms/step - loss: 1.4302e-06 -  
accuracy: 1.0000  
Epoch 64/200  
60/60 [=====] - 0s 2ms/step - loss: 1.3604e-06 -  
accuracy: 1.0000

Epoch 65/200  
60/60 [=====] - 0s 2ms/step - loss: 1.2945e-06 -  
accuracy: 1.0000  
Epoch 66/200  
60/60 [=====] - 0s 2ms/step - loss: 1.2324e-06 -  
accuracy: 1.0000  
Epoch 67/200  
60/60 [=====] - 0s 2ms/step - loss: 1.1740e-06 -  
accuracy: 1.0000  
Epoch 68/200  
60/60 [=====] - 0s 2ms/step - loss: 1.1189e-06 -  
accuracy: 1.0000  
Epoch 69/200  
60/60 [=====] - 0s 2ms/step - loss: 1.0666e-06 -  
accuracy: 1.0000  
Epoch 70/200  
60/60 [=====] - 0s 2ms/step - loss: 1.0174e-06 -  
accuracy: 1.0000  
Epoch 71/200  
60/60 [=====] - 0s 2ms/step - loss: 9.7106e-07 -  
accuracy: 1.0000  
Epoch 72/200  
60/60 [=====] - 0s 3ms/step - loss: 9.2694e-07 -  
accuracy: 1.0000  
Epoch 73/200  
60/60 [=====] - 0s 4ms/step - loss: 8.8543e-07 -  
accuracy: 1.0000  
Epoch 74/200  
60/60 [=====] - 0s 3ms/step - loss: 8.4577e-07 -  
accuracy: 1.0000  
Epoch 75/200  
60/60 [=====] - 0s 3ms/step - loss: 8.0846e-07 -  
accuracy: 1.0000  
Epoch 76/200  
60/60 [=====] - 0s 3ms/step - loss: 7.7283e-07 -  
accuracy: 1.0000  
Epoch 77/200  
60/60 [=====] - 0s 3ms/step - loss: 7.3940e-07 -  
accuracy: 1.0000  
Epoch 78/200  
60/60 [=====] - 0s 3ms/step - loss: 7.0714e-07 -  
accuracy: 1.0000  
Epoch 79/200  
60/60 [=====] - 0s 5ms/step - loss: 6.7682e-07 -  
accuracy: 1.0000  
Epoch 80/200  
60/60 [=====] - 0s 2ms/step - loss: 6.4795e-07 -  
accuracy: 1.0000

Epoch 81/200  
60/60 [=====] - 0s 2ms/step - loss: 6.2044e-07 -  
accuracy: 1.0000  
Epoch 82/200  
60/60 [=====] - 0s 2ms/step - loss: 5.9444e-07 -  
accuracy: 1.0000  
Epoch 83/200  
60/60 [=====] - 0s 2ms/step - loss: 5.6934e-07 -  
accuracy: 1.0000  
Epoch 84/200  
60/60 [=====] - 0s 2ms/step - loss: 5.4572e-07 -  
accuracy: 1.0000  
Epoch 85/200  
60/60 [=====] - 0s 2ms/step - loss: 5.2324e-07 -  
accuracy: 1.0000  
Epoch 86/200  
60/60 [=====] - 0s 2ms/step - loss: 5.0160e-07 -  
accuracy: 1.0000  
Epoch 87/200  
60/60 [=====] - 0s 2ms/step - loss: 4.8112e-07 -  
accuracy: 1.0000  
Epoch 88/200  
60/60 [=====] - 0s 2ms/step - loss: 4.6142e-07 -  
accuracy: 1.0000  
Epoch 89/200  
60/60 [=====] - 0s 2ms/step - loss: 4.4292e-07 -  
accuracy: 1.0000  
Epoch 90/200  
60/60 [=====] - 0s 2ms/step - loss: 4.2507e-07 -  
accuracy: 1.0000  
Epoch 91/200  
60/60 [=====] - 0s 2ms/step - loss: 4.0806e-07 -  
accuracy: 1.0000  
Epoch 92/200  
60/60 [=====] - 0s 2ms/step - loss: 3.9194e-07 -  
accuracy: 1.0000  
Epoch 93/200  
60/60 [=====] - 0s 2ms/step - loss: 3.7639e-07 -  
accuracy: 1.0000  
Epoch 94/200  
60/60 [=====] - 0s 3ms/step - loss: 3.6160e-07 -  
accuracy: 1.0000  
Epoch 95/200  
60/60 [=====] - 0s 3ms/step - loss: 3.4739e-07 -  
accuracy: 1.0000  
Epoch 96/200  
60/60 [=====] - 0s 3ms/step - loss: 3.3386e-07 -  
accuracy: 1.0000



Epoch 97/200  
60/60 [=====] - 0s 3ms/step - loss: 3.2100e-07 -  
accuracy: 1.0000  
Epoch 98/200  
60/60 [=====] - 0s 3ms/step - loss: 3.0850e-07 -  
accuracy: 1.0000  
Epoch 99/200  
60/60 [=====] - 0s 4ms/step - loss: 2.9667e-07 -  
accuracy: 1.0000  
Epoch 100/200  
60/60 [=====] - 0s 4ms/step - loss: 2.8532e-07 -  
accuracy: 1.0000  
Epoch 101/200  
60/60 [=====] - 0s 3ms/step - loss: 2.7444e-07 -  
accuracy: 1.0000  
Epoch 102/200  
60/60 [=====] - 0s 2ms/step - loss: 2.6403e-07 -  
accuracy: 1.0000  
Epoch 103/200  
60/60 [=====] - 0s 2ms/step - loss: 2.5407e-07 -  
accuracy: 1.0000  
Epoch 104/200  
60/60 [=====] - 0s 2ms/step - loss: 2.4455e-07 -  
accuracy: 1.0000  
Epoch 105/200  
60/60 [=====] - 0s 2ms/step - loss: 2.3544e-07 -  
accuracy: 1.0000  
Epoch 106/200  
60/60 [=====] - 0s 2ms/step - loss: 2.2663e-07 -  
accuracy: 1.0000  
Epoch 107/200  
60/60 [=====] - 0s 2ms/step - loss: 2.1818e-07 -  
accuracy: 1.0000  
Epoch 108/200  
60/60 [=====] - 0s 2ms/step - loss: 2.1015e-07 -  
accuracy: 1.0000  
Epoch 109/200  
60/60 [=====] - 0s 2ms/step - loss: 2.0240e-07 -  
accuracy: 1.0000  
Epoch 110/200  
60/60 [=====] - 0s 2ms/step - loss: 1.9496e-07 -  
accuracy: 1.0000  
Epoch 111/200  
60/60 [=====] - 0s 2ms/step - loss: 1.8781e-07 -  
accuracy: 1.0000  
Epoch 112/200  
60/60 [=====] - 0s 2ms/step - loss: 1.8098e-07 -  
accuracy: 1.0000

Epoch 113/200  
60/60 [=====] - 0s 2ms/step - loss: 1.7443e-07 -  
accuracy: 1.0000  
Epoch 114/200  
60/60 [=====] - 0s 3ms/step - loss: 1.6808e-07 -  
accuracy: 1.0000  
Epoch 115/200  
60/60 [=====] - 0s 3ms/step - loss: 1.6209e-07 -  
accuracy: 1.0000  
Epoch 116/200  
60/60 [=====] - 0s 3ms/step - loss: 1.5621e-07 -  
accuracy: 1.0000  
Epoch 117/200  
60/60 [=====] - 0s 3ms/step - loss: 1.5065e-07 -  
accuracy: 1.0000  
Epoch 118/200  
60/60 [=====] - 0s 3ms/step - loss: 1.4526e-07 -  
accuracy: 1.0000  
Epoch 119/200  
60/60 [=====] - 0s 3ms/step - loss: 1.4012e-07 -  
accuracy: 1.0000  
Epoch 120/200  
60/60 [=====] - 0s 3ms/step - loss: 1.3518e-07 -  
accuracy: 1.0000  
Epoch 121/200  
60/60 [=====] - 0s 3ms/step - loss: 1.3042e-07 -  
accuracy: 1.0000  
Epoch 122/200  
60/60 [=====] - 0s 2ms/step - loss: 1.2582e-07 -  
accuracy: 1.0000  
Epoch 123/200  
60/60 [=====] - 0s 2ms/step - loss: 1.2139e-07 -  
accuracy: 1.0000  
Epoch 124/200  
60/60 [=====] - 0s 2ms/step - loss: 1.1715e-07 -  
accuracy: 1.0000  
Epoch 125/200  
60/60 [=====] - 0s 2ms/step - loss: 1.1307e-07 -  
accuracy: 1.0000  
Epoch 126/200  
60/60 [=====] - 0s 2ms/step - loss: 1.0914e-07 -  
accuracy: 1.0000  
Epoch 127/200  
60/60 [=====] - 0s 2ms/step - loss: 1.0534e-07 -  
accuracy: 1.0000  
Epoch 128/200  
60/60 [=====] - 0s 2ms/step - loss: 1.0172e-07 -  
accuracy: 1.0000

Epoch 129/200  
60/60 [=====] - 0s 2ms/step - loss: 9.8196e-08 -  
accuracy: 1.0000  
Epoch 130/200  
60/60 [=====] - 0s 2ms/step - loss: 9.4849e-08 -  
accuracy: 1.0000  
Epoch 131/200  
60/60 [=====] - 0s 2ms/step - loss: 9.1564e-08 -  
accuracy: 1.0000  
Epoch 132/200  
60/60 [=====] - 0s 2ms/step - loss: 8.8454e-08 -  
accuracy: 1.0000  
Epoch 133/200  
60/60 [=====] - 0s 2ms/step - loss: 8.5433e-08 -  
accuracy: 1.0000  
Epoch 134/200  
60/60 [=====] - 0s 2ms/step - loss: 8.2545e-08 -  
accuracy: 1.0000  
Epoch 135/200  
60/60 [=====] - 0s 3ms/step - loss: 7.9768e-08 -  
accuracy: 1.0000  
Epoch 136/200  
60/60 [=====] - 0s 3ms/step - loss: 7.7067e-08 -  
accuracy: 1.0000  
Epoch 137/200  
60/60 [=====] - 0s 3ms/step - loss: 7.4472e-08 -  
accuracy: 1.0000  
Epoch 138/200  
60/60 [=====] - 0s 4ms/step - loss: 7.1992e-08 -  
accuracy: 1.0000  
Epoch 139/200  
60/60 [=====] - 0s 4ms/step - loss: 6.9578e-08 -  
accuracy: 1.0000  
Epoch 140/200  
60/60 [=====] - 0s 3ms/step - loss: 6.7257e-08 -  
accuracy: 1.0000  
Epoch 141/200  
60/60 [=====] - 0s 4ms/step - loss: 6.5013e-08 -  
accuracy: 1.0000  
Epoch 142/200  
60/60 [=====] - 0s 3ms/step - loss: 6.2852e-08 -  
accuracy: 1.0000  
Epoch 143/200  
60/60 [=====] - 0s 2ms/step - loss: 6.0785e-08 -  
accuracy: 1.0000  
Epoch 144/200  
60/60 [=====] - 0s 2ms/step - loss: 5.8762e-08 -  
accuracy: 1.0000

Epoch 145/200  
60/60 [=====] - 0s 2ms/step - loss: 5.6837e-08 -  
accuracy: 1.0000  
Epoch 146/200  
60/60 [=====] - 0s 2ms/step - loss: 5.4986e-08 -  
accuracy: 1.0000  
Epoch 147/200  
60/60 [=====] - 0s 2ms/step - loss: 5.3202e-08 -  
accuracy: 1.0000  
Epoch 148/200  
60/60 [=====] - 0s 2ms/step - loss: 5.1467e-08 -  
accuracy: 1.0000  
Epoch 149/200  
60/60 [=====] - 0s 2ms/step - loss: 4.9791e-08 -  
accuracy: 1.0000  
Epoch 150/200  
60/60 [=====] - 0s 2ms/step - loss: 4.8174e-08 -  
accuracy: 1.0000  
Epoch 151/200  
60/60 [=====] - 0s 2ms/step - loss: 4.6622e-08 -  
accuracy: 1.0000  
Epoch 152/200  
60/60 [=====] - 0s 2ms/step - loss: 4.5125e-08 -  
accuracy: 1.0000  
Epoch 153/200  
60/60 [=====] - 0s 2ms/step - loss: 4.3682e-08 -  
accuracy: 1.0000  
Epoch 154/200  
60/60 [=====] - 0s 2ms/step - loss: 4.2293e-08 -  
accuracy: 1.0000  
Epoch 155/200  
60/60 [=====] - 0s 2ms/step - loss: 4.0953e-08 -  
accuracy: 1.0000  
Epoch 156/200  
60/60 [=====] - 0s 2ms/step - loss: 3.9650e-08 -  
accuracy: 1.0000  
Epoch 157/200  
60/60 [=====] - 0s 3ms/step - loss: 3.8391e-08 -  
accuracy: 1.0000  
Epoch 158/200  
60/60 [=====] - 0s 3ms/step - loss: 3.7176e-08 -  
accuracy: 1.0000  
Epoch 159/200  
60/60 [=====] - 0s 3ms/step - loss: 3.5997e-08 -  
accuracy: 1.0000  
Epoch 160/200  
60/60 [=====] - 0s 3ms/step - loss: 3.4857e-08 -  
accuracy: 1.0000

Epoch 161/200  
60/60 [=====] - 0s 3ms/step - loss: 3.3766e-08 -  
accuracy: 1.0000  
Epoch 162/200  
60/60 [=====] - 0s 3ms/step - loss: 3.2703e-08 -  
accuracy: 1.0000  
Epoch 163/200  
60/60 [=====] - 0s 3ms/step - loss: 3.1680e-08 -  
accuracy: 1.0000  
Epoch 164/200  
60/60 [=====] - 0s 3ms/step - loss: 3.0694e-08 -  
accuracy: 1.0000  
Epoch 165/200  
60/60 [=====] - 0s 3ms/step - loss: 2.9747e-08 -  
accuracy: 1.0000  
Epoch 166/200  
60/60 [=====] - 0s 2ms/step - loss: 2.8835e-08 -  
accuracy: 1.0000  
Epoch 167/200  
60/60 [=====] - 0s 2ms/step - loss: 2.7952e-08 -  
accuracy: 1.0000  
Epoch 168/200  
60/60 [=====] - 0s 2ms/step - loss: 2.7089e-08 -  
accuracy: 1.0000  
Epoch 169/200  
60/60 [=====] - 0s 2ms/step - loss: 2.6256e-08 -  
accuracy: 1.0000  
Epoch 170/200  
60/60 [=====] - 0s 2ms/step - loss: 2.5442e-08 -  
accuracy: 1.0000  
Epoch 171/200  
60/60 [=====] - 0s 2ms/step - loss: 2.4669e-08 -  
accuracy: 1.0000  
Epoch 172/200  
60/60 [=====] - 0s 2ms/step - loss: 2.3919e-08 -  
accuracy: 1.0000  
Epoch 173/200  
60/60 [=====] - 0s 2ms/step - loss: 2.3195e-08 -  
accuracy: 1.0000  
Epoch 174/200  
60/60 [=====] - 0s 2ms/step - loss: 2.2485e-08 -  
accuracy: 1.0000  
Epoch 175/200  
60/60 [=====] - 0s 2ms/step - loss: 2.1802e-08 -  
accuracy: 1.0000  
Epoch 176/200  
60/60 [=====] - 0s 2ms/step - loss: 2.1142e-08 -  
accuracy: 1.0000

Epoch 177/200  
60/60 [=====] - 0s 2ms/step - loss: 2.0500e-08 -  
accuracy: 1.0000  
Epoch 178/200  
60/60 [=====] - 0s 3ms/step - loss: 1.9885e-08 -  
accuracy: 1.0000  
Epoch 179/200  
60/60 [=====] - 0s 4ms/step - loss: 1.9292e-08 -  
accuracy: 1.0000  
Epoch 180/200  
60/60 [=====] - 0s 3ms/step - loss: 1.8724e-08 -  
accuracy: 1.0000  
Epoch 181/200  
60/60 [=====] - 0s 3ms/step - loss: 1.8168e-08 -  
accuracy: 1.0000  
Epoch 182/200  
60/60 [=====] - 0s 3ms/step - loss: 1.7633e-08 -  
accuracy: 1.0000  
Epoch 183/200  
60/60 [=====] - 0s 3ms/step - loss: 1.7109e-08 -  
accuracy: 1.0000  
Epoch 184/200  
60/60 [=====] - 0s 3ms/step - loss: 1.6598e-08 -  
accuracy: 1.0000  
Epoch 185/200  
60/60 [=====] - 0s 3ms/step - loss: 1.6108e-08 -  
accuracy: 1.0000  
Epoch 186/200  
60/60 [=====] - 0s 2ms/step - loss: 1.5633e-08 -  
accuracy: 1.0000  
Epoch 187/200  
60/60 [=====] - 0s 2ms/step - loss: 1.5182e-08 -  
accuracy: 1.0000  
Epoch 188/200  
60/60 [=====] - 0s 2ms/step - loss: 1.4741e-08 -  
accuracy: 1.0000  
Epoch 189/200  
60/60 [=====] - 0s 2ms/step - loss: 1.4316e-08 -  
accuracy: 1.0000  
Epoch 190/200  
60/60 [=====] - 0s 2ms/step - loss: 1.3902e-08 -  
accuracy: 1.0000  
Epoch 191/200  
60/60 [=====] - 0s 2ms/step - loss: 1.3503e-08 -  
accuracy: 1.0000  
Epoch 192/200  
60/60 [=====] - 0s 2ms/step - loss: 1.3115e-08 -  
accuracy: 1.0000

```

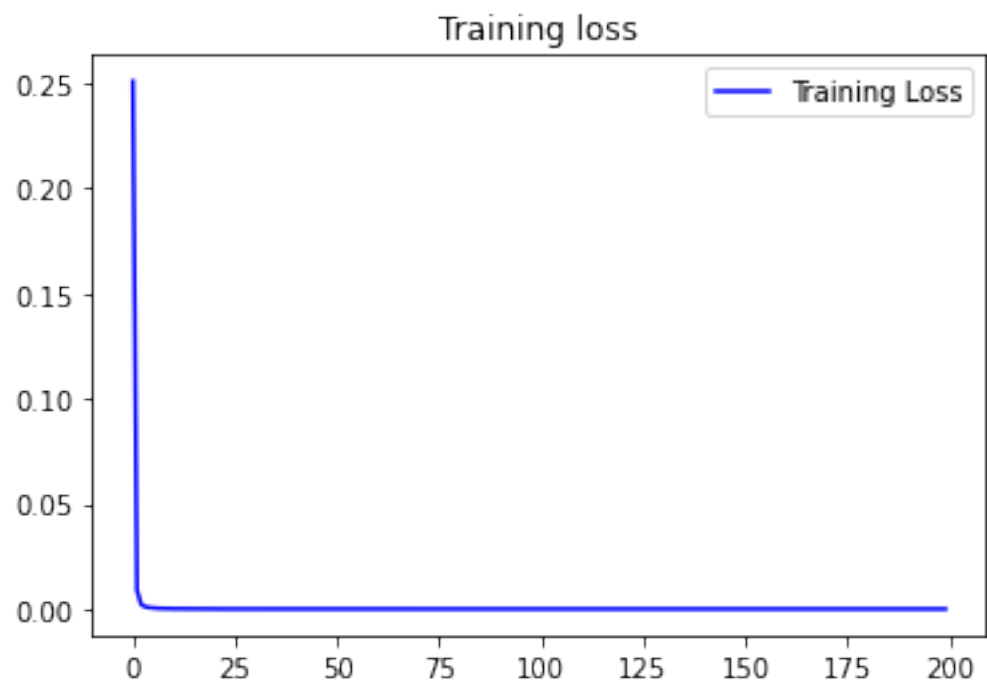
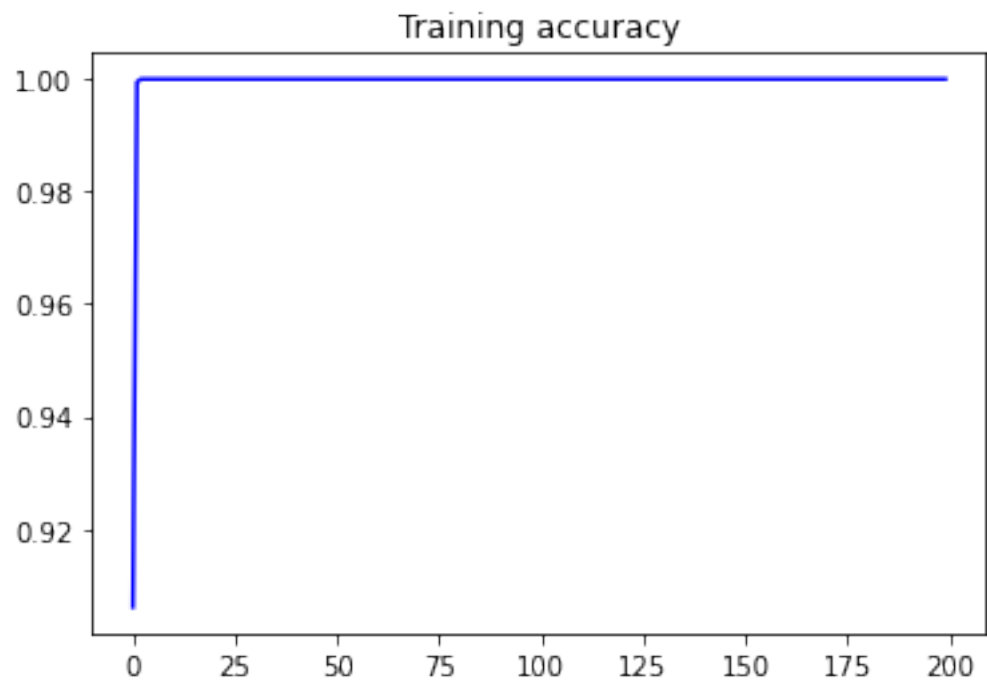
Epoch 193/200
60/60 [=====] - 0s 2ms/step - loss: 1.2745e-08 -
accuracy: 1.0000
Epoch 194/200
60/60 [=====] - 0s 2ms/step - loss: 1.2378e-08 -
accuracy: 1.0000
Epoch 195/200
60/60 [=====] - 0s 2ms/step - loss: 1.2027e-08 -
accuracy: 1.0000
Epoch 196/200
60/60 [=====] - 0s 2ms/step - loss: 1.1692e-08 -
accuracy: 1.0000
Epoch 197/200
60/60 [=====] - 0s 2ms/step - loss: 1.1368e-08 -
accuracy: 1.0000
Epoch 198/200
60/60 [=====] - 0s 2ms/step - loss: 1.1058e-08 -
accuracy: 1.0000
Epoch 199/200
60/60 [=====] - 0s 3ms/step - loss: 1.0756e-08 -
accuracy: 1.0000
Epoch 200/200
60/60 [=====] - 0s 3ms/step - loss: 1.0462e-08 -
accuracy: 1.0000

```

```

[8]: acc = history.history['accuracy']
loss = history.history['loss']
epochs = range(len(acc))
plt.plot(epochs, acc, 'b', label='Training accuracy')
plt.title('Training accuracy')
plt.figure()
plt.plot(epochs, loss, 'b', label='Training Loss')
plt.title('Training loss')
plt.legend()
plt.show()

```





## 4 Task 2

Test the model on the test set and report Precision, Recall, F1-Score, and Accuracy.

```
[16]: # START YOUR CODE HERE
from sklearn.metrics import classification_report
predictions = (model.predict(X_test) > 0.5) #Threshold being 0.5
print(classification_report(y_test, predictions))
```

```
22/22 [=====] - 0s 1ms/step
              precision    recall  f1-score   support

         0           1.00       1.00       1.00        247
         1           1.00       1.00       1.00        428

 accuracy                   1.00        675
 macro avg           1.00       1.00       1.00        675
weighted avg           1.00       1.00       1.00        675
```

Export your notebook to a pdf document

## 5 Congratulations!

You've come to the end of this assignment, and you have built your first neural network.

Congratulations on finishing this notebook!