1.3C - Classification using FFNN

July 18, 2025

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.

[27]:
```
# !pip install numpy
# !pip install keras
# !pip uninstall TensorFlow
# !pip install pandas
# !pip install matplotlib
# !pip list
```

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

[30]:
```
# !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.

[110]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import Dense
```

```
from sklearn.metrics import classification_report, confusion_matrix,␣
↪roc_auc_score
```

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.

[113]:
```
df = pd.read_csv('data.csv')
df.set_index('id', inplace=True)
```

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

[405]:
```
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.

[407]:
```
# Model architecture
model = Sequential([
    Dense(128, input_dim=512, activation='relu'),
    Dense(64, activation='relu'),
    Dense(1, activation='sigmoid')
])

# Compile model
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

```
C:\Users\micha\anaconda3\Lib\site-packages\keras\src\layers\core\dense.py:93:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When
```

```
using Sequential models, prefer using an `Input(shape)` object as the first
layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

3 Training

Now, let's start our training.

[409]:
```python
history = model.fit(X_train, y_train, epochs=200, batch_size=64, verbose=1)
```

```
Epoch 1/200
60/60                 1s 1ms/step -
accuracy: 0.7680 - loss: 0.4446
Epoch 2/200
60/60                 0s 1ms/step -
accuracy: 0.9991 - loss: 0.0178
Epoch 3/200
60/60                 0s 1ms/step -
accuracy: 1.0000 - loss: 0.0035
Epoch 4/200
60/60                 0s 1ms/step -
accuracy: 1.0000 - loss: 0.0015
Epoch 5/200
60/60                 0s 1ms/step -
accuracy: 1.0000 - loss: 8.7458e-04
Epoch 6/200
60/60                 0s 987us/step -
accuracy: 1.0000 - loss: 5.2560e-04
Epoch 7/200
60/60                 0s 1ms/step -
accuracy: 1.0000 - loss: 4.1031e-04
Epoch 8/200
60/60                 0s 1ms/step -
accuracy: 1.0000 - loss: 3.1117e-04
Epoch 9/200
60/60                 0s 954us/step -
accuracy: 1.0000 - loss: 2.3969e-04
Epoch 10/200
60/60                 0s 1ms/step -
accuracy: 1.0000 - loss: 1.9884e-04
Epoch 11/200
60/60                 0s 1ms/step -
accuracy: 1.0000 - loss: 1.6393e-04
Epoch 12/200
60/60                 0s 1ms/step -
accuracy: 1.0000 - loss: 1.2752e-04
Epoch 13/200
60/60                 0s 969us/step -
```

```
accuracy: 1.0000 - loss: 1.1669e-04
Epoch 14/200
60/60             0s 984us/step -
accuracy: 1.0000 - loss: 9.3810e-05
Epoch 15/200
60/60             0s 989us/step -
accuracy: 1.0000 - loss: 8.1465e-05
Epoch 16/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 7.4656e-05
Epoch 17/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 6.5677e-05
Epoch 18/200
60/60             0s 973us/step -
accuracy: 1.0000 - loss: 5.7126e-05
Epoch 19/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 5.2066e-05
Epoch 20/200
60/60             0s 979us/step -
accuracy: 1.0000 - loss: 4.4256e-05
Epoch 21/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 4.4416e-05
Epoch 22/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 3.7006e-05
Epoch 23/200
60/60             0s 980us/step -
accuracy: 1.0000 - loss: 3.5563e-05
Epoch 24/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 3.2510e-05
Epoch 25/200
60/60             0s 981us/step -
accuracy: 1.0000 - loss: 2.9850e-05
Epoch 26/200
60/60             0s 965us/step -
accuracy: 1.0000 - loss: 2.6887e-05
Epoch 27/200
60/60             0s 955us/step -
accuracy: 1.0000 - loss: 2.3522e-05
Epoch 28/200
60/60             0s 982us/step -
accuracy: 1.0000 - loss: 2.2019e-05
Epoch 29/200
60/60             0s 1ms/step -
```

```
accuracy: 1.0000 - loss: 2.0624e-05
Epoch 30/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.8042e-05
Epoch 31/200
60/60            0s 974us/step -
accuracy: 1.0000 - loss: 1.8212e-05
Epoch 32/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.5615e-05
Epoch 33/200
60/60            0s 983us/step -
accuracy: 1.0000 - loss: 1.6242e-05
Epoch 34/200
60/60            0s 983us/step -
accuracy: 1.0000 - loss: 1.4226e-05
Epoch 35/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.3581e-05
Epoch 36/200
60/60            0s 995us/step -
accuracy: 1.0000 - loss: 1.2872e-05
Epoch 37/200
60/60            0s 968us/step -
accuracy: 1.0000 - loss: 1.2021e-05
Epoch 38/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.1515e-05
Epoch 39/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.0573e-05
Epoch 40/200
60/60            0s 979us/step -
accuracy: 1.0000 - loss: 9.7810e-06
Epoch 41/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 9.3660e-06
Epoch 42/200
60/60            0s 985us/step -
accuracy: 1.0000 - loss: 8.8939e-06
Epoch 43/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 8.4549e-06
Epoch 44/200
60/60            0s 972us/step -
accuracy: 1.0000 - loss: 8.0713e-06
Epoch 45/200
60/60            0s 1ms/step -
```

```
accuracy: 1.0000 - loss: 7.4155e-06
Epoch 46/200
60/60              0s 986us/step -
accuracy: 1.0000 - loss: 6.9909e-06
Epoch 47/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 6.8596e-06
Epoch 48/200
60/60              0s 979us/step -
accuracy: 1.0000 - loss: 6.4662e-06
Epoch 49/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 5.8371e-06
Epoch 50/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 5.9828e-06
Epoch 51/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 5.4859e-06
Epoch 52/200
60/60              0s 990us/step -
accuracy: 1.0000 - loss: 5.2344e-06
Epoch 53/200
60/60              0s 985us/step -
accuracy: 1.0000 - loss: 5.2282e-06
Epoch 54/200
60/60              0s 979us/step -
accuracy: 1.0000 - loss: 5.0305e-06
Epoch 55/200
60/60              0s 973us/step -
accuracy: 1.0000 - loss: 4.3817e-06
Epoch 56/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 4.3445e-06
Epoch 57/200
60/60              0s 980us/step -
accuracy: 1.0000 - loss: 4.0462e-06
Epoch 58/200
60/60              0s 966us/step -
accuracy: 1.0000 - loss: 4.0747e-06
Epoch 59/200
60/60              0s 982us/step -
accuracy: 1.0000 - loss: 3.5963e-06
Epoch 60/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 3.6088e-06
Epoch 61/200
60/60              0s 969us/step -
```

```
accuracy: 1.0000 - loss: 3.3899e-06
Epoch 62/200
60/60              0s 972us/step -
accuracy: 1.0000 - loss: 3.2449e-06
Epoch 63/200
60/60              0s 985us/step -
accuracy: 1.0000 - loss: 3.1562e-06
Epoch 64/200
60/60              0s 973us/step -
accuracy: 1.0000 - loss: 3.0953e-06
Epoch 65/200
60/60              0s 971us/step -
accuracy: 1.0000 - loss: 2.8641e-06
Epoch 66/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 2.7520e-06
Epoch 67/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 2.6426e-06
Epoch 68/200
60/60              0s 987us/step -
accuracy: 1.0000 - loss: 2.4933e-06
Epoch 69/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 2.4035e-06
Epoch 70/200
60/60              0s 951us/step -
accuracy: 1.0000 - loss: 2.3503e-06
Epoch 71/200
60/60              0s 935us/step -
accuracy: 1.0000 - loss: 2.1940e-06
Epoch 72/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 2.2894e-06
Epoch 73/200
60/60              0s 969us/step -
accuracy: 1.0000 - loss: 2.2100e-06
Epoch 74/200
60/60              0s 966us/step -
accuracy: 1.0000 - loss: 1.9108e-06
Epoch 75/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 1.7654e-06
Epoch 76/200
60/60              0s 970us/step -
accuracy: 1.0000 - loss: 1.7701e-06
Epoch 77/200
60/60              0s 951us/step -
```

```
accuracy: 1.0000 - loss: 1.7642e-06
Epoch 78/200
60/60            0s 998us/step -
accuracy: 1.0000 - loss: 1.5489e-06
Epoch 79/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.6258e-06
Epoch 80/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.5658e-06
Epoch 81/200
60/60            0s 971us/step -
accuracy: 1.0000 - loss: 1.5132e-06
Epoch 82/200
60/60            0s 985us/step -
accuracy: 1.0000 - loss: 1.4605e-06
Epoch 83/200
60/60            0s 986us/step -
accuracy: 1.0000 - loss: 1.3932e-06
Epoch 84/200
60/60            0s 954us/step -
accuracy: 1.0000 - loss: 1.3733e-06
Epoch 85/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.3351e-06
Epoch 86/200
60/60            0s 971us/step -
accuracy: 1.0000 - loss: 1.1814e-06
Epoch 87/200
60/60            0s 970us/step -
accuracy: 1.0000 - loss: 1.1419e-06
Epoch 88/200
60/60            0s 963us/step -
accuracy: 1.0000 - loss: 1.1111e-06
Epoch 89/200
60/60            0s 991us/step -
accuracy: 1.0000 - loss: 1.0649e-06
Epoch 90/200
60/60            0s 970us/step -
accuracy: 1.0000 - loss: 1.0030e-06
Epoch 91/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 9.7913e-07
Epoch 92/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 9.6112e-07
Epoch 93/200
60/60            0s 1ms/step -
```

```
accuracy: 1.0000 - loss: 9.5988e-07
Epoch 94/200
60/60              0s 988us/step -
accuracy: 1.0000 - loss: 9.1791e-07
Epoch 95/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 8.2610e-07
Epoch 96/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 8.3160e-07
Epoch 97/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 7.4845e-07
Epoch 98/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 7.8170e-07
Epoch 99/200
60/60              0s 968us/step -
accuracy: 1.0000 - loss: 7.5341e-07
Epoch 100/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 7.7163e-07
Epoch 101/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 6.4930e-07
Epoch 102/200
60/60              0s 958us/step -
accuracy: 1.0000 - loss: 6.6738e-07
Epoch 103/200
60/60              0s 978us/step -
accuracy: 1.0000 - loss: 6.2874e-07
Epoch 104/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 6.1707e-07
Epoch 105/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 5.9496e-07
Epoch 106/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 5.8928e-07
Epoch 107/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 5.5969e-07
Epoch 108/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 5.3308e-07
Epoch 109/200
60/60              0s 991us/step -
```

```
accuracy: 1.0000 - loss: 5.2904e-07
Epoch 110/200
60/60             0s 991us/step -
accuracy: 1.0000 - loss: 5.1597e-07
Epoch 111/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 4.8484e-07
Epoch 112/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 4.7962e-07
Epoch 113/200
60/60             0s 997us/step -
accuracy: 1.0000 - loss: 4.4328e-07
Epoch 114/200
60/60             0s 969us/step -
accuracy: 1.0000 - loss: 4.4721e-07
Epoch 115/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 4.0425e-07
Epoch 116/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 4.3131e-07
Epoch 117/200
60/60             0s 982us/step -
accuracy: 1.0000 - loss: 3.8256e-07
Epoch 118/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 3.8613e-07
Epoch 119/200
60/60             0s 965us/step -
accuracy: 1.0000 - loss: 3.7271e-07
Epoch 120/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 3.5107e-07
Epoch 121/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 3.3674e-07
Epoch 122/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 3.3116e-07
Epoch 123/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 3.4239e-07
Epoch 124/200
60/60             0s 1ms/step -
accuracy: 1.0000 - loss: 3.1255e-07
Epoch 125/200
60/60             0s 1ms/step -
```

```
accuracy: 1.0000 - loss: 2.9532e-07
Epoch 126/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 2.8318e-07
Epoch 127/200
60/60            0s 997us/step -
accuracy: 1.0000 - loss: 2.7981e-07
Epoch 128/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 2.6772e-07
Epoch 129/200
60/60            0s 980us/step -
accuracy: 1.0000 - loss: 2.5104e-07
Epoch 130/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 2.4410e-07
Epoch 131/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 2.3575e-07
Epoch 132/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 2.4196e-07
Epoch 133/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 2.2179e-07
Epoch 134/200
60/60            0s 985us/step -
accuracy: 1.0000 - loss: 2.0261e-07
Epoch 135/200
60/60            0s 990us/step -
accuracy: 1.0000 - loss: 1.9935e-07
Epoch 136/200
60/60            0s 1000us/step -
accuracy: 1.0000 - loss: 1.9620e-07
Epoch 137/200
60/60            0s 977us/step -
accuracy: 1.0000 - loss: 2.0571e-07
Epoch 138/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.8991e-07
Epoch 139/200
60/60            0s 965us/step -
accuracy: 1.0000 - loss: 1.9424e-07
Epoch 140/200
60/60            0s 971us/step -
accuracy: 1.0000 - loss: 1.7096e-07
Epoch 141/200
60/60            0s 993us/step -
```

```
accuracy: 1.0000 - loss: 1.6612e-07
Epoch 142/200
60/60            0s 977us/step -
accuracy: 1.0000 - loss: 1.6549e-07
Epoch 143/200
60/60            0s 989us/step -
accuracy: 1.0000 - loss: 1.5163e-07
Epoch 144/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.5969e-07
Epoch 145/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.4125e-07
Epoch 146/200
60/60            0s 996us/step -
accuracy: 1.0000 - loss: 1.4649e-07
Epoch 147/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.3836e-07
Epoch 148/200
60/60            0s 987us/step -
accuracy: 1.0000 - loss: 1.3479e-07
Epoch 149/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.2993e-07
Epoch 150/200
60/60            0s 991us/step -
accuracy: 1.0000 - loss: 1.2464e-07
Epoch 151/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.2541e-07
Epoch 152/200
60/60            0s 958us/step -
accuracy: 1.0000 - loss: 1.1747e-07
Epoch 153/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.0349e-07
Epoch 154/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.0946e-07
Epoch 155/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 1.0715e-07
Epoch 156/200
60/60            0s 968us/step -
accuracy: 1.0000 - loss: 9.8765e-08
Epoch 157/200
60/60            0s 1ms/step -
```

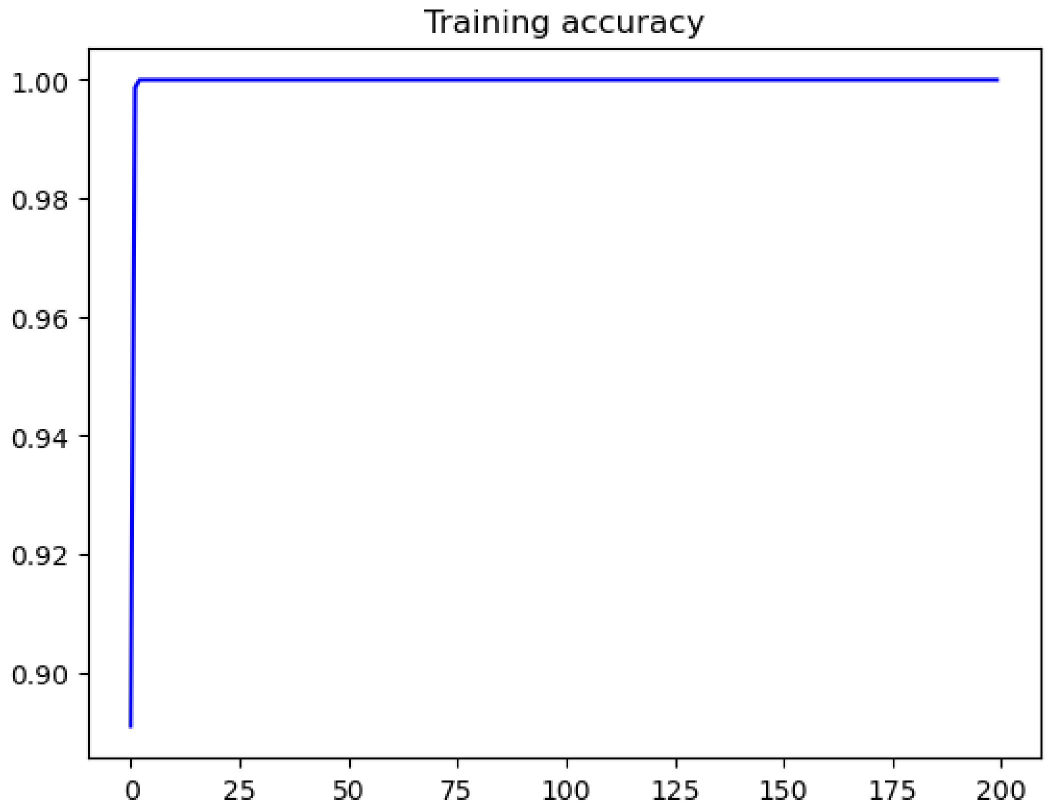
```
accuracy: 1.0000 - loss: 9.6503e-08
Epoch 158/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 9.3582e-08
Epoch 159/200
60/60            0s 996us/step -
accuracy: 1.0000 - loss: 9.3092e-08
Epoch 160/200
60/60            0s 981us/step -
accuracy: 1.0000 - loss: 8.5161e-08
Epoch 161/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 9.1349e-08
Epoch 162/200
60/60            0s 974us/step -
accuracy: 1.0000 - loss: 8.9542e-08
Epoch 163/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 8.4733e-08
Epoch 164/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 7.3321e-08
Epoch 165/200
60/60            0s 973us/step -
accuracy: 1.0000 - loss: 7.9278e-08
Epoch 166/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 7.4152e-08
Epoch 167/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 6.8223e-08
Epoch 168/200
60/60            0s 1ms/step -
accuracy: 1.0000 - loss: 6.6386e-08
Epoch 169/200
60/60            0s 995us/step -
accuracy: 1.0000 - loss: 6.9653e-08
Epoch 170/200
60/60            0s 976us/step -
accuracy: 1.0000 - loss: 6.5924e-08
Epoch 171/200
60/60            0s 987us/step -
accuracy: 1.0000 - loss: 6.0330e-08
Epoch 172/200
60/60            0s 981us/step -
accuracy: 1.0000 - loss: 5.9798e-08
Epoch 173/200
60/60            0s 966us/step -
```

```
accuracy: 1.0000 - loss: 6.0019e-08
Epoch 174/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 5.5136e-08
Epoch 175/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 5.5412e-08
Epoch 176/200
60/60              0s 968us/step -
accuracy: 1.0000 - loss: 5.1816e-08
Epoch 177/200
60/60              0s 972us/step -
accuracy: 1.0000 - loss: 5.0043e-08
Epoch 178/200
60/60              0s 996us/step -
accuracy: 1.0000 - loss: 4.9655e-08
Epoch 179/200
60/60              0s 986us/step -
accuracy: 1.0000 - loss: 4.6239e-08
Epoch 180/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 4.7704e-08
Epoch 181/200
60/60              0s 981us/step -
accuracy: 1.0000 - loss: 4.5087e-08
Epoch 182/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 4.1918e-08
Epoch 183/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 4.0682e-08
Epoch 184/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 4.1135e-08
Epoch 185/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 3.8379e-08
Epoch 186/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 3.8235e-08
Epoch 187/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 3.8696e-08
Epoch 188/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 3.3959e-08
Epoch 189/200
60/60              0s 1ms/step -
```

```
accuracy: 1.0000 - loss: 3.3789e-08
Epoch 190/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 3.4892e-08
Epoch 191/200
60/60              0s 972us/step -
accuracy: 1.0000 - loss: 3.3722e-08
Epoch 192/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 3.2305e-08
Epoch 193/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 3.1339e-08
Epoch 194/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 3.1288e-08
Epoch 195/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 3.0814e-08
Epoch 196/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 2.8833e-08
Epoch 197/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 2.5910e-08
Epoch 198/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 2.6467e-08
Epoch 199/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 2.5199e-08
Epoch 200/200
60/60              0s 1ms/step -
accuracy: 1.0000 - loss: 2.4849e-08
```

[410]:

```python
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()
```


Training accuracy
1.00
0.98
0.96
0.94
0.92
0.90
0
25
50
75
100
125
150
175
200

4 Task 2

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

[412]:
```
# Evaluate model
train_score = model.evaluate(X_train, y_train, verbose=0)
test_score = model.evaluate(X_test, y_test, verbose=0)
print(f"Training Accuracy: {train_score[1]:.4f}")
print(f"Test Accuracy: {test_score[1]:.4f}")

# Predictions
y_pred = model.predict(X_test)
y_pred_class = (y_pred > 0.5).astype(int)

# Classification report
print("\nClassification Report:")
print(classification_report(y_test, y_pred_class))

# Confusion matrix
print("\nConfusion Matrix:")
```

```
print(confusion_matrix(y_test, y_pred_class))

# ROC AUC score
roc_auc = roc_auc_score(y_test, y_pred)
print(f"\nROC AUC Score: {roc_auc:.4f}")
```

```
Training Accuracy: 1.0000
Test Accuracy: 0.9985
22/22            0s 1ms/step

Classification Report:
              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


Confusion Matrix:
[[246   1]
 [  0 428]]

ROC AUC Score: 1.0000
```

Export your notebook to a pdf document

[428]:
```
# !jupyter nbconvert --to pdf "C:\Users\micha\OneDrive\Desktop\manual backup␣
↪laptop\deakin stuff\deakintrimester3\SIT799 - Human Aligned\task1.3\1.3C -␣
↪Classification using FFNN.ipynb"
```

5 Congratulations!

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

Congratulations on finishing this notebook!