Ian Pope 700717419
Big Data Analytics - ICP 5
https://youtu.be/JUCtV2d7fD4

Ian Pope 700717419 Big Data Analytics ICP 5

Diabetes - Not Normalized

```
[8]  import keras
     import pandas
     from keras.models import Sequential
     from keras.layers import Dense, Activation

     # load dataset
     from sklearn.model_selection import train_test_split
     import pandas as pd
     import numpy as np

     # get dataset
     dataset = pd.read_csv('/content/drive/MyDrive/Colab_Notebooks/diabetes.csv', header=None).values

     X_train, X_test, Y_train, Y_test = train_test_split(dataset[:,0:8], dataset[:,8],
                                                         test_size=0.25, random_state=87)
     np.random.seed(155)
     my_first_nn = Sequential() # create model
     my_first_nn.add(Dense(20, input_dim=8, activation='relu')) # hidden layer

     #Add four more hidden layers
     my_first_nn.add(Dense(20, activation='relu')) # hidden layer
     my_first_nn.add(Dense(20, activation='relu')) # hidden layer
     my_first_nn.add(Dense(20, activation='relu')) # hidden layer
     my_first_nn.add(Dense(20, activation='relu')) # hidden layer

     my_first_nn.add(Dense(1, activation='sigmoid')) # output layer
     my_first_nn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])
     my_first_nn_fitted = my_first_nn.fit(X_train, Y_train, epochs=100,
                                          initial_epoch=0)
     print(my_first_nn.summary())
     print(my_first_nn.evaluate(X_test, Y_test))
```

```
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 0.7959 - loss: 0.4370
Epoch 85/100
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 0.7515 - loss: 0.4882
```

```
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 0.7954 - loss: 0.4269
Epoch 99/100
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 0.8024 - loss: 0.4481
Epoch 100/100
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 0.7896 - loss: 0.4188
Model: "sequential_6"
```

Layer (type)	Output Shape	Param #
dense_28 (Dense)	(None, 20)	180
dense_29 (Dense)	(None, 20)	420
dense_30 (Dense)	(None, 20)	420
dense_31 (Dense)	(None, 20)	420
dense_32 (Dense)	(None, 20)	420
dense_33 (Dense)	(None, 1)	21

```
 Total params: 5,645 (22.05 KB)
 Trainable params: 1,881 (7.35 KB)
 Non-trainable params: 0 (0.00 B)
 Optimizer params: 3,764 (14.71 KB)
None
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - acc: 0.6594 - loss: 0.6661
[0.6418289542198181, 0.7083333134651184]
```

```
import keras
import pandas
from keras.models import Sequential
from keras.layers import Dense, Activation
from sklearn.preprocessing import StandardScaler

# load dataset
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np

# Get dataset
dataset = pd.read_csv('/content/drive/MyDrive/Colab_Notebooks/diabetes.csv', header=None).values

X_train, X_test, Y_train, Y_test = train_test_split(dataset[:,0:8], dataset[:,8],
                                                    test_size=0.25, random_state=87)

# Normalize the X values using standard scaler
sc = StandardScaler()
sc.fit(X_train)
X_train = sc.transform(X_train)
X_test = sc.transform(X_test)

np.random.seed(155)
my_first_nn = Sequential() # create model
my_first_nn.add(Dense(20, input_dim=8, activation='relu')) # hidden layer

#Adds four more hidden layers
my_first_nn.add(Dense(20, activation='relu')) # hidden layer
my_first_nn.add(Dense(20, activation='relu')) # hidden layer
my_first_nn.add(Dense(20, activation='relu')) # hidden layer
my_first_nn.add(Dense(20, activation='relu')) # hidden layer

my_first_nn.add(Dense(1, activation='sigmoid')) # output layer
my_first_nn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])
my_first_nn_fitted = my_first_nn.fit(X_train, Y_train, epochs=100,
                                     initial_epoch=0)
print(my_first_nn.summary())
print(my_first_nn.evaluate(X_test, Y_test))
```

```
Epoch 99/100
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 0.9664 - loss: 0.1073
Epoch 100/100
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 0.9841 - loss: 0.0809
Model: "sequential_15"
```

Layer (type)	Output Shape	Param #
dense_78 (Dense)	(None, 20)	180
dense_79 (Dense)	(None, 20)	420
dense_80 (Dense)	(None, 20)	420
dense_81 (Dense)	(None, 20)	420
dense_82 (Dense)	(None, 20)	420
dense_83 (Dense)	(None, 1)	21

```
 Total params: 5,645 (22.05 KB)
 Trainable params: 1,881 (7.35 KB)
 Non-trainable params: 0 (0.00 B)
 Optimizer params: 3,764 (14.71 KB)
None
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 0.6769 - loss: 1.3125
[1.237978458404541, 0.6875]
```

Diabetes

With one hidden layer accuracy = 0.6666

With five hidden layers the accuracy = 0.6927

Noramlized data with one hidden layer accuaracy = 0.7604

Normalized data with five hidden layers accuracy = 0.6875

Beast Cancer not normalized

```
[9] import keras
    import pandas
    from keras.models import Sequential
    from keras.layers import Dense, Activation

    # load dataset
    from sklearn.model_selection import train_test_split
    import pandas as pd
    import numpy as np

    # Get dataset
    dataset = pd.read_csv('/content/drive/MyDrive/Colab_Notebooks/breastcancer.csv').values

    # Transform categorical data into numerical
    dataset[dataset == 'M'] = 1
    dataset[dataset == 'B'] = 0

    X_train, X_test, Y_train, Y_test = train_test_split(dataset[:,2:32], dataset[:,1],
                                                        test_size=0.25, random_state=87)

    # Convert values to float32 types to make work in NN
    X_train = np.asarray(X_train).astype(np.float32)
    Y_train = np.asarray(Y_train).astype(np.float32)
    X_test = np.asarray(X_test).astype(np.float32)
    Y_test = np.asarray(Y_test).astype(np.float32)


    np.random.seed(155)
    my_first_nn = Sequential() # create model
    my_first_nn.add(Dense(30, input_dim=30, activation='relu')) # hidden layer

    #Add four more hidden layers
    my_first_nn.add(Dense(25, activation='relu')) # hidden layer
    my_first_nn.add(Dense(20, activation='relu')) # hidden layer
    my_first_nn.add(Dense(15, activation='relu')) # hidden layer
    my_first_nn.add(Dense(10, activation='relu')) # hidden layer

    my_first_nn.add(Dense(1, activation='sigmoid')) # output layer
    my_first_nn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])
```

```
my_first_nn.add(Dense(20, activation='relu')) # hidden layer
my_first_nn.add(Dense(15, activation='relu')) # hidden layer
my_first_nn.add(Dense(10, activation='relu')) # hidden layer

my_first_nn.add(Dense(1, activation='sigmoid')) # output layer
my_first_nn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])
my_first_nn_fitted = my_first_nn.fit(X_train, Y_train, epochs=100,
                                     initial_epoch=0)
print(my_first_nn.summary())
print(my_first_nn.evaluate(X_test, Y_test))
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 0.9281 - loss: 0.1794
Epoch 85/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 0.9335 - loss: 0.2062
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - acc: 0.9390 - loss: 0.1701
Epoch 100/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 0.9237 - loss: 0.1810
Model: "sequential_7"
```

Layer (type)	Output Shape	Param #
dense_34 (Dense)	(None, 30)	930
dense_35 (Dense)	(None, 25)	775
dense_36 (Dense)	(None, 20)	520
dense_37 (Dense)	(None, 15)	315
dense_38 (Dense)	(None, 10)	160
dense_39 (Dense)	(None, 1)	11

```
 Total params: 8,135 (31.78 KB)
 Trainable params: 2,711 (10.59 KB)
 Non-trainable params: 0 (0.00 B)
 Optimizer params: 5,424 (21.19 KB)
None
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 32ms/step - acc: 0.9033 - loss: 0.3682
[0.31708231568336487, 0.9090909361839294]
```

Breast Cancer Normalized

```
import keras
import pandas
from keras.models import Sequential
from keras.layers import Dense, Activation
from sklearn.preprocessing import StandardScaler

# load dataset
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np

# Get dataset
dataset = pd.read_csv('/content/drive/MyDrive/Colab_Notebooks/breastcancer.csv').values

# Convert categorical data into numerical
dataset[dataset == 'M'] = 1
dataset[dataset == 'B'] = 0

X_train, X_test, Y_train, Y_test = train_test_split(dataset[:,2:32], dataset[:,1],
                                                    test_size=0.25, random_state=87)

# Transform data types to make work in NN
X_train = np.asarray(X_train).astype(np.float32)
Y_train = np.asarray(Y_train).astype(np.float32)
X_test = np.asarray(X_test).astype(np.float32)
Y_test = np.asarray(Y_test).astype(np.float32)

# Normalize the X values using standard scaler
sc = StandardScaler()
sc.fit(X_train)
X_train = sc.transform(X_train)
X_test = sc.transform(X_test)
```

```
np.random.seed(155)
my_first_nn = Sequential() # create model
my_first_nn.add(Dense(30, input_dim=30, activation='relu')) # hidden layer

#Add four more hidden layers
my_first_nn.add(Dense(25, activation='relu')) # hidden layer
my_first_nn.add(Dense(20, activation='relu')) # hidden layer
my_first_nn.add(Dense(15, activation='relu')) # hidden layer
my_first_nn.add(Dense(10, activation='relu')) # hidden layer

my_first_nn.add(Dense(1, activation='sigmoid')) # output layer
my_first_nn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])
my_first_nn_fitted = my_first_nn.fit(X_train, Y_train, epochs=100,
                                     initial_epoch=0)
print(my_first_nn.summary())
print(my_first_nn.evaluate(X_test, Y_test))
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 1.0000 - loss: 9.0067e-05
Epoch 85/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 1.0000 - loss: 1.4191e-04
```

```
Epoch 99/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 1.0000 - loss: 4.8438e-05
Epoch 100/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - acc: 1.0000 - loss: 6.1298e-05
Model: "sequential_5"
```

Layer (type)	Output Shape	Param #
dense_22 (Dense)	(None, 30)	930
dense_23 (Dense)	(None, 25)	775
dense_24 (Dense)	(None, 20)	520
dense_25 (Dense)	(None, 15)	315
dense_26 (Dense)	(None, 10)	160
dense_27 (Dense)	(None, 1)	11

```
 Total params: 8,135 (31.78 KB)
 Trainable params: 2,711 (10.59 KB)
 Non-trainable params: 0 (0.00 B)
 Optimizer params: 5,424 (21.19 KB)
None
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - acc: 0.9603 - loss: 0.7083
[0.43157821893692017, 0.9720279574394226]
```

Breast Cancer

Not Normalized, 1 Hidden Layer: .8951

Not Normalized, 5 hidden Layers: .9231

Normalized, 1 Hidden Layer: .9650

Normalized, 5 Hidden Layers .9720