deeplearning-with-keras-and-tensorflow-in-diabetes

April 27, 2023

[38]:
```
# This Python 3 environment comes with many helpful analytics libraries␣
 ↪installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/
 ↪docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list␣
 ↪all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that␣
 ↪gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved␣
 ↪outside of the current session
```

```
/kaggle/input/pima-indians-diabetes-database/diabetes.csv
```

[39]:
```
'''import necessary libraries'''
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
print('all libraries import perfectly')
```

```
all libraries import perfectly
```

[40]:
```
'''loading the dataset'''
```

```
dataset = pd.read_csv('/kaggle/input/pima-indians-diabetes-database/diabetes.
 ↪csv')
dataset
```

[40]:
```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0              6      148             72             35        0  33.6
1              1       85             66             29        0  26.6
2              8      183             64              0        0  23.3
3              1       89             66             23       94  28.1
4              0      137             40             35      168  43.1
..           ...      ...            ...            ...      ...  ...
763           10      101             76             48      180  32.9
764            2      122             70             27        0  36.8
765            5      121             72             23      112  26.2
766            1      126             60              0        0  30.1
767            1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]
```

[41]:
```
'''preprocessing the data'''
mean=[]
for i in dataset.columns:
    if(i=='Outcome'):
        break
    mean.append(dataset[i].mean())

count=0
for i in dataset.columns:
    if(i=='Outcome'):
        break
    dataset.loc[dataset[i]==0,i]=mean[count]
    count+=1
```

[42]:
```
dataset
```

[42]:
```
     Pregnancies  Glucose  BloodPressure  SkinThickness     Insulin   BMI  \
0       6.000000    148.0           72.0      35.000000   79.799479  33.6
1       1.000000     85.0           66.0      29.000000   79.799479  26.6
2       8.000000    183.0           64.0      20.536458   79.799479  23.3
3       1.000000     89.0           66.0      23.000000   94.000000  28.1
4       3.845052    137.0           40.0      35.000000  168.000000  43.1
..           ...      ...            ...            ...         ...   ...
763    10.000000    101.0           76.0      48.000000  180.000000  32.9
764     2.000000    122.0           70.0      27.000000   79.799479  36.8
765     5.000000    121.0           72.0      23.000000  112.000000  26.2
766     1.000000    126.0           60.0      20.536458   79.799479  30.1
767     1.000000     93.0           70.0      31.000000   79.799479  30.4

     DiabetesPedigreeFunction   Age  Outcome
0                       0.627  50.0        1
1                       0.351  31.0        0
2                       0.672  32.0        1
3                       0.167  21.0        0
4                       2.288  33.0        1
..                        ...   ...      ...
763                     0.171  63.0        0
764                     0.340  27.0        0
765                     0.245  30.0        0
766                     0.349  47.0        1
767                     0.315  23.0        0

[768 rows x 9 columns]
```

[43]:
```
'''making of dependent(y) and independent variables(x)'''
X=dataset.iloc[:,:-1]
Y=dataset.iloc[:,-1]
X.shape,Y.shape
```

[43]:
```
((768, 8), (768,))
```

[44]:
```
'''preparing the keras model'''
model = Sequential()
model.add(Dense(12,input_shape=(8,),activation='relu'))
model.add(Dense(8,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
print('model prepare successfully')
```

```
model prepare successfully
```

[45]:
```
'''compile the keras model'''
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
print('model compile successfully')
```

```
model compile successfully
```

[46]:
```
'''after that we have to fit the model with our dataset and run for specific␣
↪number of epochs(iterations) and also tell the batch size in the arguments'''
history=model.fit(X,Y,epochs=150,batch_size=10)
```

```
Epoch 1/150
77/77 [==============================] - 1s 2ms/step - loss: 6.8998 - accuracy:
0.5495
Epoch 2/150
77/77 [==============================] - 0s 2ms/step - loss: 1.6055 - accuracy:
0.6315
Epoch 3/150
77/77 [==============================] - 0s 2ms/step - loss: 1.4536 - accuracy:
0.6406
Epoch 4/150
77/77 [==============================] - 0s 2ms/step - loss: 1.2814 - accuracy:
0.6484
Epoch 5/150
77/77 [==============================] - 0s 2ms/step - loss: 1.0812 - accuracy:
0.6562
Epoch 6/150
77/77 [==============================] - 0s 2ms/step - loss: 0.9534 - accuracy:
0.6549
Epoch 7/150
77/77 [==============================] - 0s 2ms/step - loss: 0.8341 - accuracy:
0.6628
Epoch 8/150
77/77 [==============================] - 0s 2ms/step - loss: 0.7553 - accuracy:
0.6589
Epoch 9/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6900 - accuracy:
0.6901
Epoch 10/150
77/77 [==============================] - 0s 2ms/step - loss: 0.7854 - accuracy:
0.6589
Epoch 11/150
77/77 [==============================] - 0s 2ms/step - loss: 0.7089 - accuracy:
0.6667
Epoch 12/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6704 - accuracy:
0.6719
Epoch 13/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6681 - accuracy:
0.6901
Epoch 14/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6946 - accuracy:
0.6628
```

```
Epoch 15/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6806 - accuracy:
0.6823
Epoch 16/150
77/77 [==============================] - 0s 2ms/step - loss: 0.7306 - accuracy:
0.6784
Epoch 17/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6492 - accuracy:
0.7005
Epoch 18/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6440 - accuracy:
0.6875
Epoch 19/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6141 - accuracy:
0.6927
Epoch 20/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6128 - accuracy:
0.7070
Epoch 21/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6097 - accuracy:
0.6927
Epoch 22/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6358 - accuracy:
0.7005
Epoch 23/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6704 - accuracy:
0.6823
Epoch 24/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6597 - accuracy:
0.6888
Epoch 25/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6292 - accuracy:
0.6953
Epoch 26/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6163 - accuracy:
0.6953
Epoch 27/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6167 - accuracy:
0.7018
Epoch 28/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6569 - accuracy:
0.6875
Epoch 29/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6165 - accuracy:
0.6914
Epoch 30/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6184 - accuracy:
0.6901
```

```
Epoch 31/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6394 - accuracy:
0.7057
Epoch 32/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6178 - accuracy:
0.7044
Epoch 33/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6558 - accuracy:
0.6836
Epoch 34/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6016 - accuracy:
0.7122
Epoch 35/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6150 - accuracy:
0.6940
Epoch 36/150
77/77 [==============================] - 0s 2ms/step - loss: 0.7284 - accuracy:
0.6510
Epoch 37/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6261 - accuracy:
0.6901
Epoch 38/150
77/77 [==============================] - 0s 2ms/step - loss: 0.7021 - accuracy:
0.6615
Epoch 39/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6242 - accuracy:
0.6862
Epoch 40/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6000 - accuracy:
0.7161
Epoch 41/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6054 - accuracy:
0.6862
Epoch 42/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6008 - accuracy:
0.6979
Epoch 43/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6269 - accuracy:
0.7188
Epoch 44/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6603 - accuracy:
0.6719
Epoch 45/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5737 - accuracy:
0.7331
Epoch 46/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6038 - accuracy:
0.6979
```

```
Epoch 47/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6181 - accuracy:
0.6927
Epoch 48/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6143 - accuracy:
0.6940
Epoch 49/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5940 - accuracy:
0.7096
Epoch 50/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5815 - accuracy:
0.6992
Epoch 51/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6104 - accuracy:
0.7201
Epoch 52/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5748 - accuracy:
0.7057
Epoch 53/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5757 - accuracy:
0.7109
Epoch 54/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5812 - accuracy:
0.7188
Epoch 55/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5968 - accuracy:
0.6914
Epoch 56/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6017 - accuracy:
0.7135
Epoch 57/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5736 - accuracy:
0.7240
Epoch 58/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5979 - accuracy:
0.7148
Epoch 59/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5696 - accuracy:
0.7279
Epoch 60/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5653 - accuracy:
0.7070
Epoch 61/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6059 - accuracy:
0.6979
Epoch 62/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6181 - accuracy:
0.7096
```

```
Epoch 63/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5709 - accuracy:
0.7201
Epoch 64/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5974 - accuracy:
0.7135
Epoch 65/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5907 - accuracy:
0.7292
Epoch 66/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5951 - accuracy:
0.6992
Epoch 67/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5918 - accuracy:
0.7161
Epoch 68/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5998 - accuracy:
0.7253
Epoch 69/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5760 - accuracy:
0.7344
Epoch 70/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5571 - accuracy:
0.7331
Epoch 71/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5782 - accuracy:
0.7148
Epoch 72/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5692 - accuracy:
0.7383
Epoch 73/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5585 - accuracy:
0.7279
Epoch 74/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5731 - accuracy:
0.7188
Epoch 75/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5675 - accuracy:
0.7188
Epoch 76/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6255 - accuracy:
0.6875
Epoch 77/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5503 - accuracy:
0.7318
Epoch 78/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5860 - accuracy:
0.7057
```

```
Epoch 79/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5716 - accuracy:
0.7214
Epoch 80/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5419 - accuracy:
0.7643
Epoch 81/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6183 - accuracy:
0.7005
Epoch 82/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5608 - accuracy:
0.7227
Epoch 83/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5672 - accuracy:
0.7292
Epoch 84/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5332 - accuracy:
0.7396
Epoch 85/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5738 - accuracy:
0.7487
Epoch 86/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6058 - accuracy:
0.7214
Epoch 87/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5877 - accuracy:
0.7214
Epoch 88/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5787 - accuracy:
0.7174
Epoch 89/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5289 - accuracy:
0.7474
Epoch 90/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5905 - accuracy:
0.7161
Epoch 91/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5850 - accuracy:
0.7227
Epoch 92/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5607 - accuracy:
0.7279
Epoch 93/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6036 - accuracy:
0.7031
Epoch 94/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5992 - accuracy:
0.6875
```

```
Epoch 95/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6048 - accuracy:
0.7214
Epoch 96/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6447 - accuracy:
0.6641
Epoch 97/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6521 - accuracy:
0.7005
Epoch 98/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5639 - accuracy:
0.7188
Epoch 99/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5189 - accuracy:
0.7461
Epoch 100/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5867 - accuracy:
0.7305
Epoch 101/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5347 - accuracy:
0.7526
Epoch 102/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5522 - accuracy:
0.7266
Epoch 103/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6003 - accuracy:
0.7161
Epoch 104/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5537 - accuracy:
0.7331
Epoch 105/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5568 - accuracy:
0.7383
Epoch 106/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5626 - accuracy:
0.7344
Epoch 107/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5329 - accuracy:
0.7591
Epoch 108/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5714 - accuracy:
0.7148
Epoch 109/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5718 - accuracy:
0.7109
Epoch 110/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5177 - accuracy:
0.7266
```

```
Epoch 111/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5650 - accuracy:
0.7396
Epoch 112/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5697 - accuracy:
0.7318
Epoch 113/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5742 - accuracy:
0.7161
Epoch 114/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5333 - accuracy:
0.7279
Epoch 115/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5308 - accuracy:
0.7487
Epoch 116/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5303 - accuracy:
0.7344
Epoch 117/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5514 - accuracy:
0.7292
Epoch 118/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5429 - accuracy:
0.7435
Epoch 119/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5256 - accuracy:
0.7461
Epoch 120/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5381 - accuracy:
0.7318
Epoch 121/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5969 - accuracy:
0.7174
Epoch 122/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5321 - accuracy:
0.7422
Epoch 123/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5529 - accuracy:
0.7318
Epoch 124/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5139 - accuracy:
0.7591
Epoch 125/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5129 - accuracy:
0.7513
Epoch 126/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5435 - accuracy:
0.7318
```

```
Epoch 127/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5650 - accuracy:
0.7370
Epoch 128/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5321 - accuracy:
0.7370
Epoch 129/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6271 - accuracy:
0.7057
Epoch 130/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5765 - accuracy:
0.7383
Epoch 131/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5375 - accuracy:
0.7526
Epoch 132/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5406 - accuracy:
0.7370
Epoch 133/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5270 - accuracy:
0.7565
Epoch 134/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5249 - accuracy:
0.7591
Epoch 135/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5684 - accuracy:
0.7357
Epoch 136/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5515 - accuracy:
0.7422
Epoch 137/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5223 - accuracy:
0.7552
Epoch 138/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5205 - accuracy:
0.7630
Epoch 139/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5445 - accuracy:
0.7422
Epoch 140/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5132 - accuracy:
0.7474
Epoch 141/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5500 - accuracy:
0.7344
Epoch 142/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5494 - accuracy:
0.7240
```

```
Epoch 143/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5380 - accuracy:
0.7435
Epoch 144/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5734 - accuracy:
0.7253
Epoch 145/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5661 - accuracy:
0.7096
Epoch 146/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5038 - accuracy:
0.7565
Epoch 147/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5276 - accuracy:
0.7604
Epoch 148/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5385 - accuracy:
0.7370
Epoch 149/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5188 - accuracy:
0.7552
Epoch 150/150
77/77 [==============================] - 0s 2ms/step - loss: 0.5338 - accuracy:
0.7461
```

[47]:
```
'''now we have to evaluate our model on above processing'''
accuracy = model.evaluate(X,Y)
print(accuracy)
```

```
24/24 [==============================] - 0s 2ms/step - loss: 0.4771 - accuracy:
0.7760
[0.47707709670066833, 0.7760416865348816]
```

[48]:
```
'''after getting output we have to predict the output based on above output'''
predict_output = model.predict(X)
predict_output.shape
'''rounding up the predictions'''
rounded = [round(x[-1])for x in predict_output]
```

```
24/24 [==============================] - 0s 1ms/step
```

[49]:
```
'''convert data to 1 if prediction more than 0.5 otherwise 0'''
predict_output = (model.predict(X)>0.5).astype(int)
predict_output.shape
```

```
24/24 [==============================] - 0s 1ms/step
```

[49]: (768, 1)

[63]:
```
X.iloc[[0]]
```

[63]:
```
   Pregnancies  Glucose  BloodPressure  SkinThickness    Insulin   BMI  \
0          6.0    148.0           72.0           35.0  79.799479  33.6

   DiabetesPedigreeFunction   Age
0                     0.627  50.0
```

[68]:
```
'''displaying the result'''
for i in range(5):
    print(X.iloc[[i]],Y[i],predict_output[i],end="\n")
    print()
```

```
   Pregnancies  Glucose  BloodPressure  SkinThickness    Insulin   BMI  \
0          6.0    148.0           72.0           35.0  79.799479  33.6

   DiabetesPedigreeFunction   Age
0                     0.627  50.0   1 [1]

   Pregnancies  Glucose  BloodPressure  SkinThickness    Insulin   BMI  \
1          1.0     85.0           66.0           29.0  79.799479  26.6

   DiabetesPedigreeFunction   Age
1                     0.351  31.0   0 [0]

   Pregnancies  Glucose  BloodPressure  SkinThickness    Insulin   BMI  \
2          8.0    183.0           64.0      20.536458  79.799479  23.3

   DiabetesPedigreeFunction   Age
2                     0.672  32.0   1 [1]

   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
3          1.0     89.0           66.0           23.0     94.0  28.1

   DiabetesPedigreeFunction   Age
3                     0.167  21.0   0 [0]

   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
4     3.845052    137.0           40.0           35.0    168.0  43.1

   DiabetesPedigreeFunction   Age
4                     2.288  33.0   1 [1]
```

[70]:
```
output = list(Y)
pred_outpt = list(predict_output)
X['output']=output
```

```
X['prediction']=pred_outpt
```

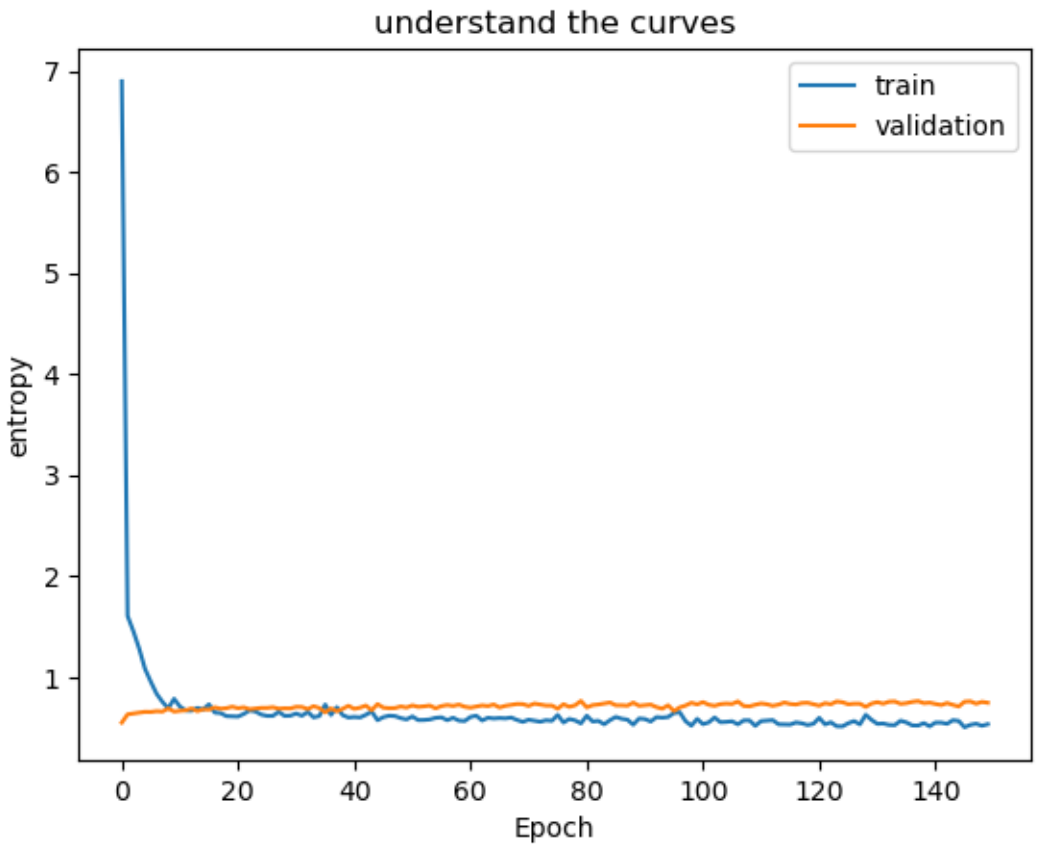
[76]:
```
for i in range(5):
    print(X.iloc[[i]])
```

```
   Pregnancies  Glucose  BloodPressure  SkinThickness    Insulin   BMI  \
0          6.0    148.0           72.0           35.0  79.799479  33.6

   DiabetesPedigreeFunction   Age  output prediction
0                     0.627  50.0       1        [1]
   Pregnancies  Glucose  BloodPressure  SkinThickness    Insulin   BMI  \
1          1.0     85.0           66.0           29.0  79.799479  26.6

   DiabetesPedigreeFunction   Age  output prediction
1                     0.351  31.0       0        [0]
   Pregnancies  Glucose  BloodPressure  SkinThickness    Insulin   BMI  \
2          8.0    183.0           64.0      20.536458  79.799479  23.3

   DiabetesPedigreeFunction   Age  output prediction
2                     0.672  32.0       1        [1]
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
3          1.0     89.0           66.0           23.0     94.0  28.1

   DiabetesPedigreeFunction   Age  output prediction
3                     0.167  21.0       0        [0]
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
4     3.845052    137.0           40.0           35.0    168.0  43.1

   DiabetesPedigreeFunction   Age  output prediction
4                     2.288  33.0       1        [1]
```

[53]:
```
print(history.history.keys())
```

```
dict_keys(['loss', 'accuracy'])
```

[55]:
```
plt.title('understand the curves')
plt.xlabel('Epoch')
plt.ylabel('entropy')
plt.plot(history.history['loss'],label='train')
plt.plot(history.history['accuracy'],label='validation')
plt.legend()
plt.show()
```



[56]:
```
model.save('model.h5')
```

[72]:
```
from tensorflow.keras.models import load_model
model = load_model('model.h5')
pred=[20,200,125,100,850,68,2.45,85]
pred_out = model.predict([pred])
print('Predicted: %.3f' % pred_out[0])
```

```
1/1 [==============================] - 0s 100ms/step
Predicted: 0.999
```