

```
In [1]: !pip install ann_visualizer
!pip install graphviz
```

```
Collecting ann_visualizer
  Downloading ann_visualizer-2.5.tar.gz (4.7 kB)
Building wheels for collected packages: ann-visualizer
  Building wheel for ann-visualizer (setup.py): started
  Building wheel for ann-visualizer (setup.py): finished with status 'done'
  Created wheel for ann-visualizer: filename=ann_visualizer-2.5-py3-none-any.whl size=4168 sha256=9220b77da755b5c2aa54d4b2bdc087c7d888eba43271df51feafcc4a9863b3c
  Stored in directory: c:\users\aj240\appdata\local\pip\cache\wheels\2d\d8\86\67f4a249969eaaa31c6df569f4ebfa84634fae3af2c627107b
Successfully built ann-visualizer
Installing collected packages: ann-visualizer
Successfully installed ann-visualizer-2.5

WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ConnectTimeoutError(<pip._vendor.urllib3.connection.HTTPSConnection object at 0x0000025A767110D0>, 'Connection to file s.pythonhosted.org timed out. (connect timeout=15)')': /packages/db/51/157be500337fba347e32711aaf9f11c1ba9e1162f486a1d708b4ae594ea4/ann_visualizer-2.5.tar.gz

Collecting graphviz
  Downloading graphviz-0.20.1-py3-none-any.whl (47 kB)
Installing collected packages: graphviz
Successfully installed graphviz-0.20.1
```

```
In [2]: import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
from ann_visualizer.visualize import ann_viz
```

```
In [3]: import os
os.environ["PATH"] += os.pathsep + 'C:/Program Files/Graphviz/bin'
```

```
In [4]: df = pd.read_csv('pima-indians-diabetes.csv')
```

In [5]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 767 entries, 0 to 766
Data columns (total 9 columns):
 #   Column  Non-Null Count  Dtype  
---  -
 0    6      767 non-null    int64  
 1   148     767 non-null    int64  
 2    72     767 non-null    int64  
 3    35     767 non-null    int64  
 4    0      767 non-null    int64  
 5   33.6    767 non-null    float64 
 6   0.627   767 non-null    float64 
 7    50     767 non-null    int64  
 8    1      767 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [6]: df.columns

Out[6]: Index(['6', '148', '72', '35', '0', '33.6', '0.627', '50', '1'], dtype='object')

In [7]: X=df.iloc[:,0:-1].values

In [8]: y=df.iloc[:,8].values

```
In [9]: model=Sequential()
model.add(Dense(12,input_dim=8,activation='relu'))
model.add(Dense(8,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
model.fit(X,y,epochs=100,batch_size=10)
```

```
Epoch 1/100
77/77 [=====] - 1s 1ms/step - loss: 10.2106 - accuracy: 0.3468
Epoch 2/100
77/77 [=====] - 0s 1ms/step - loss: 2.0491 - accuracy: 0.5150
Epoch 3/100
77/77 [=====] - 0s 934us/step - loss: 0.9768 - accuracy: 0.6167
Epoch 4/100
77/77 [=====] - 0s 1ms/step - loss: 0.8587 - accuracy: 0.6037
Epoch 5/100
77/77 [=====] - 0s 1ms/step - loss: 0.8056 - accuracy: 0.6154
Epoch 6/100
77/77 [=====] - 0s 1ms/step - loss: 0.7558 - accuracy: 0.6167
Epoch 7/100
77/77 [=====] - 0s 1ms/step - loss: 0.7161 - accuracy: 0.7161
```

```
In [10]: _,accuracy=model.evaluate(X,y)
print('Accuracy: %.2f' % (accuracy*100))
```

```
24/24 [=====] - 0s 1ms/step - loss: 0.4996 - accuracy: 0.7614
Accuracy: 76.14
```

```
In [11]: ann_viz(model, view=True, filename="Neural_network.pdf", title="MyNeural Network")
```

```
In [12]: import numpy as np
predictions=np.round_(model.predict(X))

df1=pd.DataFrame()

df1.insert(0,'Actual Class',y,True)
df1.insert(1,'Predicted Class',predictions,True)
```

In [13]: `df1.head()`

Out[13]:

	Actual Class	Predicted Class
0	0	0.0
1	1	1.0
2	0	0.0
3	1	1.0
4	0	0.0

In [14]: `print('executed on 15/11/22 by Aarya Jha')`

executed on 15/11/22 by Aarya Jha

In []: