Activity 1.2 : Training Neural Networks

Objective(s):

This activity aims to demonstrate how to train neural networks using keras

Intended Learning Outcomes (ILOs):

- Demonstrate how to build and train neural networks
- Demonstrate how to evaluate and plot the model using training and validation loss

Resources:

- Jupyter Notebook

CI Pima Diabetes Dataset

- pima-indians-diabetes.csv

Procedures

Load the necessary libraries

In [16]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, precision_recall_curve, roc_auc_score
from sklearn.ensemble import RandomForestClassifier

import seaborn as sns

%matplotlib inline
```

In [17]:
```
## Import Keras objects for Deep Learning

from keras.models  import Sequential
from keras.layers import Input, Dense, Flatten, Dropout, BatchNormalization
from keras.optimizers import Adam, SGD, RMSprop
```

Load the dataset

In [18]:
```
filepath = "pima-indians-diabetes.csv"
names = ["times_pregnant", "glucose_tolerance_test", "blood_pressure", "skin_thickn
         "bmi", "pedigree_function", "age", "has_diabetes"]
diabetes_df = pd.read_csv(filepath, names=names)
```

Check the top 5 samples of the data

In [19]:
```
print(diabetes_df.shape)
diabetes_df.sample(5)
```

```
(768, 9)
```

Out[19]:

	times_pregnant	glucose_tolerance_test	blood_pressure	skin_thickness	insulin	bmi
330	8	118	72	19	0	23.1
331	2	87	58	16	52	32.7
608	0	152	82	39	272	41.5
309	2	124	68	28	205	32.9
24	11	143	94	33	146	36.6

In [20]:
```
diabetes_df.dtypes
```

Out[20]:
```
times_pregnant              int64
glucose_tolerance_test      int64
blood_pressure              int64
skin_thickness              int64
insulin                     int64
bmi                       float64
pedigree_function         float64
age                         int64
has_diabetes                int64
dtype: object
```

In [21]:
```
X = diabetes_df.iloc[:, :-1].values
y = diabetes_df["has_diabetes"].values
```

Split the data to Train, and Test (75%, 25%)

In [22]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_st
```

In [23]:
```
np.mean(y), np.mean(1-y)
```

Out[23]:
```
(0.3489583333333333, 0.6510416666666666)
```

Build a single hidden layer neural network using 12 nodes. Use the sequential model with single layer network and input shape to 8.

Normalize the data

In [24]:
```
normalizer = StandardScaler()
```

```
X_train_norm = normalizer.fit_transform(X_train)
X_test_norm = normalizer.transform(X_test)
```

Define the model:

- Input size is 8-dimensional
- 1 hidden layer, 12 hidden nodes, sigmoid activation
- Final layer with one node and sigmoid activation (standard for binary classification)

In [25]:
```
model  = Sequential([
    Dense(12, input_shape=(8,), activation="relu"),
    Dense(1, activation="sigmoid")
])
```

```
C:\Python312\Lib\site-packages\keras\src\layers\core\dense.py:85: UserWarning: Do no
t pass an `input_shape`/`input_dim` argument to a layer. When using Sequential model
s, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

View the model summary

In [26]:
```
model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	
dense_2 (Dense)	(None, 12)	
dense_3 (Dense)	(None, 1)	

Total params: 121 (484.00 B)

Trainable params: 121 (484.00 B)

Non-trainable params: 0 (0.00 B)

Train the model

- Compile the model with optimizer, loss function and metrics
- Use the fit function to return the run history.

In [37]:
```
model.compile(SGD(learning_rate =.003), "binary_crossentropy", metrics=["accuracy"]
run_hist_1 = model.fit(X_train_norm, y_train, validation_data=(X_test_norm, y_test)
```

```
Epoch 1/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.7914 - loss: 0.4364 - val_accu
racy: 0.7500 - val_loss: 0.4940
Epoch 2/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7971 - loss: 0.4438 - val_accur
acy: 0.7500 - val_loss: 0.4940
Epoch 3/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7865 - loss: 0.4596 - val_accur
acy: 0.7500 - val_loss: 0.4940
Epoch 4/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7868 - loss: 0.4434 - val_accur
acy: 0.7500 - val_loss: 0.4941
Epoch 5/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8127 - loss: 0.4284 - val_accur
acy: 0.7500 - val_loss: 0.4941
Epoch 6/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8036 - loss: 0.4246 - val_accur
acy: 0.7500 - val_loss: 0.4941
Epoch 7/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8024 - loss: 0.4316 - val_accur
acy: 0.7500 - val_loss: 0.4941
Epoch 8/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7884 - loss: 0.4744 - val_accur
acy: 0.7500 - val_loss: 0.4942
Epoch 9/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7944 - loss: 0.4400 - val_accur
acy: 0.7500 - val_loss: 0.4942
Epoch 10/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7841 - loss: 0.4478 - val_accur
acy: 0.7500 - val_loss: 0.4942
Epoch 11/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7949 - loss: 0.4470 - val_accur
acy: 0.7500 - val_loss: 0.4942
Epoch 12/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8293 - loss: 0.3946 - val_accur
acy: 0.7500 - val_loss: 0.4943
Epoch 13/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8142 - loss: 0.4133 - val_accur
acy: 0.7500 - val_loss: 0.4943
Epoch 14/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7861 - loss: 0.4574 - val_accur
acy: 0.7500 - val_loss: 0.4943
Epoch 15/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7755 - loss: 0.4675 - val_accur
acy: 0.7500 - val_loss: 0.4943
Epoch 16/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8219 - loss: 0.4083 - val_accur
acy: 0.7500 - val_loss: 0.4943
Epoch 17/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8158 - loss: 0.4212 - val_accur
acy: 0.7500 - val_loss: 0.4943
Epoch 18/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7922 - loss: 0.4450 - val_accur
acy: 0.7500 - val_loss: 0.4944
Epoch 19/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7833 - loss: 0.4470 - val_accur
```

```
acy: 0.7500 - val_loss: 0.4944
Epoch 20/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8095 - loss: 0.4370 - val_accur
acy: 0.7500 - val_loss: 0.4944
Epoch 21/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8016 - loss: 0.4302 - val_accur
acy: 0.7500 - val_loss: 0.4945
Epoch 22/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8122 - loss: 0.4242 - val_accur
acy: 0.7500 - val_loss: 0.4945
Epoch 23/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8052 - loss: 0.4450 - val_accur
acy: 0.7448 - val_loss: 0.4945
Epoch 24/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7685 - loss: 0.4806 - val_accur
acy: 0.7448 - val_loss: 0.4945
Epoch 25/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7844 - loss: 0.4448 - val_accur
acy: 0.7448 - val_loss: 0.4945
Epoch 26/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7954 - loss: 0.4667 - val_accur
acy: 0.7500 - val_loss: 0.4946
Epoch 27/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7930 - loss: 0.4375 - val_accur
acy: 0.7500 - val_loss: 0.4946
Epoch 28/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7703 - loss: 0.4537 - val_accur
acy: 0.7500 - val_loss: 0.4946
Epoch 29/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8136 - loss: 0.4106 - val_accur
acy: 0.7500 - val_loss: 0.4946
Epoch 30/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8050 - loss: 0.4273 - val_accur
acy: 0.7500 - val_loss: 0.4947
Epoch 31/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7709 - loss: 0.4660 - val_accur
acy: 0.7500 - val_loss: 0.4947
Epoch 32/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7958 - loss: 0.4502 - val_accur
acy: 0.7500 - val_loss: 0.4947
Epoch 33/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7954 - loss: 0.4351 - val_accur
acy: 0.7500 - val_loss: 0.4947
Epoch 34/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7980 - loss: 0.4178 - val_accur
acy: 0.7500 - val_loss: 0.4947
Epoch 35/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7877 - loss: 0.4496 - val_accur
acy: 0.7500 - val_loss: 0.4948
Epoch 36/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7947 - loss: 0.4539 - val_accur
acy: 0.7500 - val_loss: 0.4948
Epoch 37/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7976 - loss: 0.4461 - val_accur
acy: 0.7500 - val_loss: 0.4948
Epoch 38/200
```

```
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8025 - loss: 0.4421 - val_accur
acy: 0.7500 - val_loss: 0.4948
Epoch 39/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8062 - loss: 0.4289 - val_accur
acy: 0.7500 - val_loss: 0.4948
Epoch 40/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8087 - loss: 0.4400 - val_accur
acy: 0.7500 - val_loss: 0.4948
Epoch 41/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8183 - loss: 0.4182 - val_accur
acy: 0.7500 - val_loss: 0.4949
Epoch 42/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8408 - loss: 0.4062 - val_accur
acy: 0.7500 - val_loss: 0.4949
Epoch 43/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7971 - loss: 0.4407 - val_accur
acy: 0.7500 - val_loss: 0.4949
Epoch 44/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7906 - loss: 0.4470 - val_accur
acy: 0.7552 - val_loss: 0.4949
Epoch 45/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7974 - loss: 0.4348 - val_accur
acy: 0.7552 - val_loss: 0.4949
Epoch 46/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8219 - loss: 0.4343 - val_accur
acy: 0.7552 - val_loss: 0.4950
Epoch 47/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7873 - loss: 0.4327 - val_accur
acy: 0.7552 - val_loss: 0.4950
Epoch 48/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8013 - loss: 0.4316 - val_accur
acy: 0.7552 - val_loss: 0.4950
Epoch 49/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8126 - loss: 0.4234 - val_accur
acy: 0.7552 - val_loss: 0.4950
Epoch 50/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7945 - loss: 0.4447 - val_accur
acy: 0.7552 - val_loss: 0.4950
Epoch 51/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7980 - loss: 0.4565 - val_accur
acy: 0.7552 - val_loss: 0.4951
Epoch 52/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8033 - loss: 0.4418 - val_accur
acy: 0.7552 - val_loss: 0.4951
Epoch 53/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7846 - loss: 0.4675 - val_accur
acy: 0.7552 - val_loss: 0.4951
Epoch 54/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8064 - loss: 0.4236 - val_accur
acy: 0.7552 - val_loss: 0.4951
Epoch 55/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8002 - loss: 0.4412 - val_accur
acy: 0.7552 - val_loss: 0.4951
Epoch 56/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7898 - loss: 0.4449 - val_accur
acy: 0.7552 - val_loss: 0.4951
```

```
Epoch 57/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7976 - loss: 0.4470 - val_accur
acy: 0.7552 - val_loss: 0.4952
Epoch 58/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7802 - loss: 0.4715 - val_accur
acy: 0.7552 - val_loss: 0.4952
Epoch 59/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8227 - loss: 0.4160 - val_accur
acy: 0.7552 - val_loss: 0.4952
Epoch 60/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7872 - loss: 0.4412 - val_accur
acy: 0.7552 - val_loss: 0.4952
Epoch 61/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8139 - loss: 0.4163 - val_accur
acy: 0.7552 - val_loss: 0.4952
Epoch 62/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8099 - loss: 0.4337 - val_accur
acy: 0.7552 - val_loss: 0.4952
Epoch 63/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8047 - loss: 0.4546 - val_accur
acy: 0.7552 - val_loss: 0.4952
Epoch 64/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8124 - loss: 0.4234 - val_accur
acy: 0.7552 - val_loss: 0.4953
Epoch 65/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8185 - loss: 0.4219 - val_accur
acy: 0.7552 - val_loss: 0.4953
Epoch 66/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7961 - loss: 0.4482 - val_accur
acy: 0.7604 - val_loss: 0.4953
Epoch 67/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7815 - loss: 0.4504 - val_accur
acy: 0.7604 - val_loss: 0.4953
Epoch 68/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8142 - loss: 0.4351 - val_accur
acy: 0.7604 - val_loss: 0.4953
Epoch 69/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8156 - loss: 0.4208 - val_accur
acy: 0.7604 - val_loss: 0.4953
Epoch 70/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8146 - loss: 0.4097 - val_accur
acy: 0.7604 - val_loss: 0.4953
Epoch 71/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7755 - loss: 0.4705 - val_accur
acy: 0.7604 - val_loss: 0.4954
Epoch 72/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8045 - loss: 0.4401 - val_accur
acy: 0.7604 - val_loss: 0.4954
Epoch 73/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8260 - loss: 0.4051 - val_accur
acy: 0.7604 - val_loss: 0.4954
Epoch 74/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8051 - loss: 0.4076 - val_accur
acy: 0.7604 - val_loss: 0.4954
Epoch 75/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7923 - loss: 0.4464 - val_accur
```

```
acy: 0.7604 - val_loss: 0.4954
Epoch 76/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8136 - loss: 0.4353 - val_accur
acy: 0.7604 - val_loss: 0.4954
Epoch 77/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7885 - loss: 0.4554 - val_accur
acy: 0.7604 - val_loss: 0.4954
Epoch 78/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7898 - loss: 0.4352 - val_accur
acy: 0.7604 - val_loss: 0.4954
Epoch 79/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7767 - loss: 0.4405 - val_accur
acy: 0.7604 - val_loss: 0.4955
Epoch 80/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8187 - loss: 0.4155 - val_accur
acy: 0.7604 - val_loss: 0.4955
Epoch 81/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7881 - loss: 0.4393 - val_accur
acy: 0.7604 - val_loss: 0.4955
Epoch 82/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8041 - loss: 0.4530 - val_accur
acy: 0.7604 - val_loss: 0.4955
Epoch 83/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8195 - loss: 0.4230 - val_accur
acy: 0.7604 - val_loss: 0.4955
Epoch 84/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8006 - loss: 0.4433 - val_accur
acy: 0.7604 - val_loss: 0.4955
Epoch 85/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8002 - loss: 0.4303 - val_accur
acy: 0.7604 - val_loss: 0.4955
Epoch 86/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7985 - loss: 0.4379 - val_accur
acy: 0.7604 - val_loss: 0.4955
Epoch 87/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7913 - loss: 0.4374 - val_accur
acy: 0.7604 - val_loss: 0.4955
Epoch 88/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7975 - loss: 0.4547 - val_accur
acy: 0.7604 - val_loss: 0.4955
Epoch 89/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7774 - loss: 0.4596 - val_accur
acy: 0.7604 - val_loss: 0.4956
Epoch 90/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8074 - loss: 0.4191 - val_accur
acy: 0.7604 - val_loss: 0.4956
Epoch 91/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7877 - loss: 0.4533 - val_accur
acy: 0.7604 - val_loss: 0.4956
Epoch 92/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8118 - loss: 0.4210 - val_accur
acy: 0.7604 - val_loss: 0.4956
Epoch 93/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7928 - loss: 0.4562 - val_accur
acy: 0.7604 - val_loss: 0.4956
Epoch 94/200
```

```
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7976 - loss: 0.4246 - val_accur
acy: 0.7604 - val_loss: 0.4956
Epoch 95/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7968 - loss: 0.4354 - val_accur
acy: 0.7604 - val_loss: 0.4956
Epoch 96/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8160 - loss: 0.4151 - val_accur
acy: 0.7604 - val_loss: 0.4956
Epoch 97/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7927 - loss: 0.4350 - val_accur
acy: 0.7604 - val_loss: 0.4957
Epoch 98/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7909 - loss: 0.4350 - val_accur
acy: 0.7604 - val_loss: 0.4957
Epoch 99/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7765 - loss: 0.4691 - val_accur
acy: 0.7604 - val_loss: 0.4957
Epoch 100/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7893 - loss: 0.4368 - val_accur
acy: 0.7604 - val_loss: 0.4957
Epoch 101/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8075 - loss: 0.4340 - val_accur
acy: 0.7604 - val_loss: 0.4957
Epoch 102/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7885 - loss: 0.4530 - val_accur
acy: 0.7604 - val_loss: 0.4957
Epoch 103/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8108 - loss: 0.4227 - val_accur
acy: 0.7604 - val_loss: 0.4957
Epoch 104/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7982 - loss: 0.4308 - val_accur
acy: 0.7604 - val_loss: 0.4958
Epoch 105/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7819 - loss: 0.4606 - val_accur
acy: 0.7604 - val_loss: 0.4958
Epoch 106/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7864 - loss: 0.4647 - val_accur
acy: 0.7604 - val_loss: 0.4958
Epoch 107/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8206 - loss: 0.4223 - val_accur
acy: 0.7604 - val_loss: 0.4958
Epoch 108/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8069 - loss: 0.4056 - val_accur
acy: 0.7604 - val_loss: 0.4958
Epoch 109/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8077 - loss: 0.4185 - val_accur
acy: 0.7604 - val_loss: 0.4958
Epoch 110/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8130 - loss: 0.4173 - val_accur
acy: 0.7604 - val_loss: 0.4958
Epoch 111/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7944 - loss: 0.4185 - val_accur
acy: 0.7604 - val_loss: 0.4959
Epoch 112/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8129 - loss: 0.4012 - val_accur
acy: 0.7604 - val_loss: 0.4959
```

```
Epoch 113/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8186 - loss: 0.4307 - val_accur
acy: 0.7604 - val_loss: 0.4959
Epoch 114/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7991 - loss: 0.4451 - val_accur
acy: 0.7604 - val_loss: 0.4959
Epoch 115/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8093 - loss: 0.4234 - val_accur
acy: 0.7604 - val_loss: 0.4959
Epoch 116/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7824 - loss: 0.4508 - val_accur
acy: 0.7604 - val_loss: 0.4959
Epoch 117/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7851 - loss: 0.4580 - val_accur
acy: 0.7604 - val_loss: 0.4959
Epoch 118/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7863 - loss: 0.4553 - val_accur
acy: 0.7604 - val_loss: 0.4959
Epoch 119/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8008 - loss: 0.4444 - val_accur
acy: 0.7604 - val_loss: 0.4960
Epoch 120/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8186 - loss: 0.4183 - val_accur
acy: 0.7604 - val_loss: 0.4960
Epoch 121/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7892 - loss: 0.4397 - val_accur
acy: 0.7604 - val_loss: 0.4960
Epoch 122/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8214 - loss: 0.4078 - val_accur
acy: 0.7604 - val_loss: 0.4960
Epoch 123/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8215 - loss: 0.4100 - val_accur
acy: 0.7604 - val_loss: 0.4960
Epoch 124/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8111 - loss: 0.4295 - val_accur
acy: 0.7604 - val_loss: 0.4960
Epoch 125/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7734 - loss: 0.4620 - val_accur
acy: 0.7604 - val_loss: 0.4961
Epoch 126/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8027 - loss: 0.4236 - val_accur
acy: 0.7604 - val_loss: 0.4961
Epoch 127/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7520 - loss: 0.4957 - val_accur
acy: 0.7604 - val_loss: 0.4961
Epoch 128/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7797 - loss: 0.4621 - val_accur
acy: 0.7604 - val_loss: 0.4961
Epoch 129/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7948 - loss: 0.4392 - val_accur
acy: 0.7604 - val_loss: 0.4961
Epoch 130/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8191 - loss: 0.3988 - val_accur
acy: 0.7604 - val_loss: 0.4961
Epoch 131/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7967 - loss: 0.4279 - val_accur
```

```
acy: 0.7604 - val_loss: 0.4962
Epoch 132/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7852 - loss: 0.4352 - val_accur
acy: 0.7604 - val_loss: 0.4962
Epoch 133/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8333 - loss: 0.3842 - val_accur
acy: 0.7604 - val_loss: 0.4962
Epoch 134/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7965 - loss: 0.4097 - val_accur
acy: 0.7604 - val_loss: 0.4962
Epoch 135/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7852 - loss: 0.4665 - val_accur
acy: 0.7604 - val_loss: 0.4962
Epoch 136/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7934 - loss: 0.4338 - val_accur
acy: 0.7604 - val_loss: 0.4962
Epoch 137/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7837 - loss: 0.4487 - val_accur
acy: 0.7604 - val_loss: 0.4963
Epoch 138/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7939 - loss: 0.4207 - val_accur
acy: 0.7604 - val_loss: 0.4963
Epoch 139/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7969 - loss: 0.4291 - val_accur
acy: 0.7604 - val_loss: 0.4963
Epoch 140/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7782 - loss: 0.4413 - val_accur
acy: 0.7604 - val_loss: 0.4963
Epoch 141/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7874 - loss: 0.4649 - val_accur
acy: 0.7604 - val_loss: 0.4963
Epoch 142/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7905 - loss: 0.4521 - val_accur
acy: 0.7604 - val_loss: 0.4963
Epoch 143/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7872 - loss: 0.4536 - val_accur
acy: 0.7604 - val_loss: 0.4963
Epoch 144/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7879 - loss: 0.4332 - val_accur
acy: 0.7604 - val_loss: 0.4964
Epoch 145/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7873 - loss: 0.5012 - val_accur
acy: 0.7604 - val_loss: 0.4964
Epoch 146/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8143 - loss: 0.4085 - val_accur
acy: 0.7604 - val_loss: 0.4964
Epoch 147/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7963 - loss: 0.4307 - val_accur
acy: 0.7604 - val_loss: 0.4964
Epoch 148/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8141 - loss: 0.4063 - val_accur
acy: 0.7604 - val_loss: 0.4964
Epoch 149/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7956 - loss: 0.4334 - val_accur
acy: 0.7604 - val_loss: 0.4964
Epoch 150/200
```

```
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8087 - loss: 0.4052 - val_accur
acy: 0.7604 - val_loss: 0.4964
Epoch 151/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8087 - loss: 0.4187 - val_accur
acy: 0.7604 - val_loss: 0.4964
Epoch 152/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8158 - loss: 0.4189 - val_accur
acy: 0.7604 - val_loss: 0.4965
Epoch 153/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8080 - loss: 0.4403 - val_accur
acy: 0.7604 - val_loss: 0.4965
Epoch 154/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7852 - loss: 0.4431 - val_accur
acy: 0.7604 - val_loss: 0.4965
Epoch 155/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8047 - loss: 0.4171 - val_accur
acy: 0.7604 - val_loss: 0.4965
Epoch 156/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7735 - loss: 0.4569 - val_accur
acy: 0.7604 - val_loss: 0.4965
Epoch 157/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8124 - loss: 0.4190 - val_accur
acy: 0.7604 - val_loss: 0.4965
Epoch 158/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8078 - loss: 0.4285 - val_accur
acy: 0.7604 - val_loss: 0.4965
Epoch 159/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7881 - loss: 0.4434 - val_accur
acy: 0.7604 - val_loss: 0.4965
Epoch 160/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7992 - loss: 0.4261 - val_accur
acy: 0.7604 - val_loss: 0.4966
Epoch 161/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8148 - loss: 0.4091 - val_accur
acy: 0.7604 - val_loss: 0.4966
Epoch 162/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8150 - loss: 0.4225 - val_accur
acy: 0.7604 - val_loss: 0.4966
Epoch 163/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7668 - loss: 0.4621 - val_accur
acy: 0.7604 - val_loss: 0.4966
Epoch 164/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7895 - loss: 0.4243 - val_accur
acy: 0.7604 - val_loss: 0.4966
Epoch 165/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8244 - loss: 0.4056 - val_accur
acy: 0.7604 - val_loss: 0.4966
Epoch 166/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7884 - loss: 0.4467 - val_accur
acy: 0.7604 - val_loss: 0.4966
Epoch 167/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8153 - loss: 0.3922 - val_accur
acy: 0.7604 - val_loss: 0.4966
Epoch 168/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8175 - loss: 0.4031 - val_accur
acy: 0.7604 - val_loss: 0.4966
```

```
Epoch 169/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8014 - loss: 0.4463 - val_accur
acy: 0.7604 - val_loss: 0.4967
Epoch 170/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8166 - loss: 0.4025 - val_accur
acy: 0.7604 - val_loss: 0.4967
Epoch 171/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8085 - loss: 0.4268 - val_accur
acy: 0.7604 - val_loss: 0.4967
Epoch 172/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8040 - loss: 0.4302 - val_accur
acy: 0.7604 - val_loss: 0.4967
Epoch 173/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8103 - loss: 0.4164 - val_accur
acy: 0.7604 - val_loss: 0.4967
Epoch 174/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7770 - loss: 0.4547 - val_accur
acy: 0.7604 - val_loss: 0.4967
Epoch 175/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8123 - loss: 0.4190 - val_accur
acy: 0.7604 - val_loss: 0.4967
Epoch 176/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8125 - loss: 0.4176 - val_accur
acy: 0.7604 - val_loss: 0.4968
Epoch 177/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7884 - loss: 0.4524 - val_accur
acy: 0.7604 - val_loss: 0.4968
Epoch 178/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8017 - loss: 0.4153 - val_accur
acy: 0.7604 - val_loss: 0.4968
Epoch 179/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8062 - loss: 0.4168 - val_accur
acy: 0.7604 - val_loss: 0.4968
Epoch 180/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7990 - loss: 0.4293 - val_accur
acy: 0.7604 - val_loss: 0.4968
Epoch 181/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7681 - loss: 0.4472 - val_accur
acy: 0.7604 - val_loss: 0.4968
Epoch 182/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8204 - loss: 0.4060 - val_accur
acy: 0.7604 - val_loss: 0.4969
Epoch 183/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7898 - loss: 0.4529 - val_accur
acy: 0.7604 - val_loss: 0.4969
Epoch 184/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8101 - loss: 0.4369 - val_accur
acy: 0.7604 - val_loss: 0.4969
Epoch 185/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7787 - loss: 0.4441 - val_accur
acy: 0.7604 - val_loss: 0.4969
Epoch 186/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8196 - loss: 0.4141 - val_accur
acy: 0.7604 - val_loss: 0.4969
Epoch 187/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8084 - loss: 0.4099 - val_accur
```

```
acy: 0.7604 - val_loss: 0.4969
Epoch 188/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8106 - loss: 0.4184 - val_accur
acy: 0.7604 - val_loss: 0.4969
Epoch 189/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7963 - loss: 0.4314 - val_accur
acy: 0.7604 - val_loss: 0.4969
Epoch 190/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7875 - loss: 0.4494 - val_accur
acy: 0.7604 - val_loss: 0.4970
Epoch 191/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7849 - loss: 0.4512 - val_accur
acy: 0.7604 - val_loss: 0.4970
Epoch 192/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7966 - loss: 0.4197 - val_accur
acy: 0.7604 - val_loss: 0.4970
Epoch 193/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8008 - loss: 0.4206 - val_accur
acy: 0.7604 - val_loss: 0.4970
Epoch 194/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8094 - loss: 0.4308 - val_accur
acy: 0.7604 - val_loss: 0.4970
Epoch 195/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7869 - loss: 0.4434 - val_accur
acy: 0.7604 - val_loss: 0.4970
Epoch 196/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7980 - loss: 0.4150 - val_accur
acy: 0.7604 - val_loss: 0.4970
Epoch 197/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7987 - loss: 0.4321 - val_accur
acy: 0.7604 - val_loss: 0.4971
Epoch 198/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8055 - loss: 0.4267 - val_accur
acy: 0.7604 - val_loss: 0.4971
Epoch 199/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8067 - loss: 0.4232 - val_accur
acy: 0.7604 - val_loss: 0.4971
Epoch 200/200
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8040 - loss: 0.4412 - val_accur
acy: 0.7604 - val_loss: 0.4971
```

In [45]:

```python
## Like we did for the Random Forest, we generate two kinds of predictions
#  One is a hard decision, the other is a probabilitistic score.

y_pred_class_nn_1 = model.predict(X_test_norm) #predict_classes is deprecated
y_pred_prob_nn_1 = model.predict(X_test_norm)
```

```
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
```

In [42]:

```python
# Let's check out the outputs to get a feel for how keras apis work.
y_pred_class_nn_1[:10]
```

Out[42]:

```
array([[0.5993208 ],
       [0.8447913 ],
       [0.30370638],
       [0.17771429],
       [0.22383448],
       [0.49790466],
       [0.02425634],
       [0.2858935 ],
       [0.9277605 ],
       [0.184226  ]], dtype=float32)
```

In [43]:

```
y_pred_prob_nn_1[:10]
```

Out[43]:

```
array([[0.5993208 ],
       [0.8447913 ],
       [0.30370638],
       [0.17771429],
       [0.22383448],
       [0.49790466],
       [0.02425634],
       [0.2858935 ],
       [0.9277605 ],
       [0.184226  ]], dtype=float32)
```

Create the plot_roc function

In [47]:

```
def plot_roc(y_test, y_pred, model_name):
    fpr, tpr, thr = roc_curve(y_test, y_pred)
    fig, ax = plt.subplots(figsize=(8, 8))
    ax.plot(fpr, tpr, 'k-')
    ax.plot([0, 1], [0, 1], 'k--', linewidth=.5)  # roc curve for random model
    ax.grid(True)
    ax.set(title='ROC Curve for {} on PIMA diabetes problem'.format(model_name),
           xlim=[-0.01, 1.01], ylim=[-0.01, 1.01])
```

Evaluate the model performance and plot the ROC CURVE

In [50]:

```
print('accuracy is {:.3f}'.format(accuracy_score(y_test,y_pred_class_nn_1)))
print('roc-auc is {:.3f}'.format(roc_auc_score(y_test,y_pred_prob_nn_1)))

plot_roc(y_test, y_pred_prob_nn_1, 'NN')
```