



Introduction to Keras

Keras is an open-source neural network library written in Python that enables developers to easily design, build, and train deep learning models. It is a high-level API built on top of lower-level neural network libraries such as TensorFlow, Theano, and CNTK, providing a simpler and more intuitive way to create deep learning models. Keras allows users to quickly build and prototype deep learning models with minimal coding and provides a range of tools for implementing a variety of neural network architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Additionally, Keras provides easy-to-use interfaces for data preprocessing, model evaluation, and visualization, making it an accessible tool for both beginners and experienced developers alike. With its user-friendly interface, flexibility, and wide range of applications, Keras has become a popular choice for implementing deep learning models in a variety of fields, including computer vision, natural language processing, and robotics.

Using Keras to Build and Train Neural Networks

The purpose of this notebook is to employ a neural network to predict diabetes. To begin, we will establish a performance baseline by training a Random Forest model. Subsequently, we will use the Keras package to construct and train a neural network swiftly and evaluate its performance in comparison to the Random Forest model. Furthermore, we will investigate how diverse network structures influence performance, training time, and the degree of overfitting or underfitting.

Diabetes Dataset

[Download Dataset](#)

Attributes: (all numeric-valued)

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)²)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

The Diabetes Dataset which has 8 numerical predictors and a binary outcome.

```
In [77]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
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, roc_curve, accuracy_score
from sklearn.ensemble import RandomForestClassifier
```

```
In [78]: ## 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
```

```
In [79]: ## Load in the data set
names = ["times_pregnant", "glucose_tolerance_test", "blood_pressure", "skin_thickness", "insulin",
         "bmi", "pedigree_function", "age", "has_diabetes"]
diabetes_df = pd.read_csv('./diabetes.csv', names=names, header=0)
```

```
In [80]: # Take a peek at the data -- if there are lots of "NaN" you may have internet connectivity issues
print(diabetes_df.shape)
diabetes_df.sample(5)
```

```
(768, 9)
```

```
Out[80]:
```

	times_pregnant	glucose_tolerance_test	blood_pressure	skin_thickness	insulin	bmi	pedigree_function	age	has_diabetes
202	0	108	68	20	0	27.3	0.787	32	0
101	1	151	60	0	0	26.1	0.179	22	0
128	1	117	88	24	145	34.5	0.403	40	1
547	4	131	68	21	166	33.1	0.160	28	0
678	3	121	52	0	0	36.0	0.127	25	1

```
In [81]: X = diabetes_df.iloc[:, :-1].values  
y = diabetes_df["has_diabetes"].values
```

```
In [82]: # Split the data to Train, and Test (75%, 25%)  
  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=11111)
```

```
In [83]: np.mean(y), np.mean(1-y)
```

```
Out[83]: (0.3489583333333333, 0.6510416666666666)
```

Above, we see that about 35% of the patients in this dataset have diabetes, while 65% do not. This means we can get an accuracy of 65% without any model - just declare that no one has diabetes. We will calculate the ROC-AUC score to evaluate performance of our model, and also look at the accuracy as well to see if we improved upon the 65% accuracy.

Get a baseline performance using Random Forest

To begin, and get a baseline for classifier performance:

1. Train a Random Forest model with 200 trees on the training data.
2. Calculate the accuracy and roc_auc_score of the predictions.

```
In [84]: ## Train the RF Model  
rf_model = RandomForestClassifier(n_estimators=200)  
rf_model.fit(X_train, y_train)
```

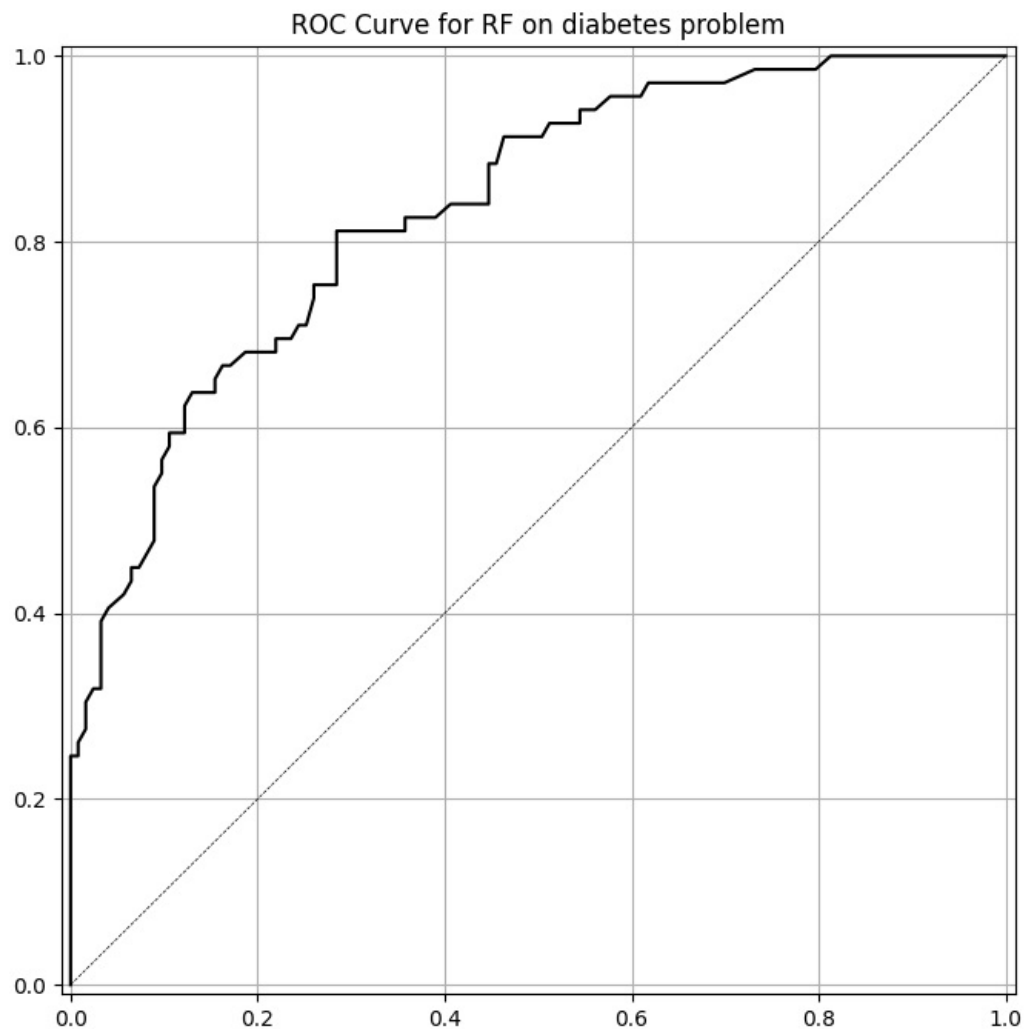
```
Out[84]:
```

RandomForestClassifier

RandomForestClassifier(n_estimators=200)

```
In [85]: # Make predictions on the test set - both "hard" predictions, and the scores (percent of trees voting yes)  
y_pred_class_rf = rf_model.predict(X_test)  
y_pred_prob_rf = rf_model.predict_proba(X_test)  
  
print('accuracy is {:.3f}'.format(accuracy_score(y_test, y_pred_class_rf)))  
print('roc-auc is {:.3f}'.format(roc_auc_score(y_test, y_pred_prob_rf[:, 1])))  
  
accuracy is 0.786  
roc-auc is 0.835
```

```
In [87]: 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 diabetes problem'.format(model_name),  
           xlim=[-0.01, 1.01], ylim=[-0.01, 1.01])  
plot_roc(y_test, y_pred_prob_rf[:, 1], 'RF')
```



Build a Single Hidden Layer Neural Network

We will use the Sequential model to quickly build a neural network. Our first network will be a single layer network. We have 8 variables, so we set the input shape to 8. Let's start by having a single hidden layer with 12 nodes.

```
In [61]: ## First let's normalize the data
## This aids the training of neural nets by providing numerical stability
## Random Forest does not need this as it finds a split only, as opposed to performing matrix multiplications

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

```
In [88]: # Define the Model
# Input size is 8-dimensional
# 1 hidden layer, 12 hidden nodes, sigmoid activation
# Final layer has just one node with a sigmoid activation (standard for binary classification)

model_1 = Sequential()
model_1.add(Dense(12, input_shape = (8,), activation = 'sigmoid'))
model_1.add(Dense(1, activation='sigmoid'))
```

```
In [89]: # This is a nice tool to view the model you have created and count the parameters

model_1.summary()
```

Model: "sequential_6"

Layer (type)	Output Shape	Param #
dense_15 (Dense)	(None, 12)	108
dense_16 (Dense)	(None, 1)	13

```
=====
Total params: 121
Trainable params: 121
Non-trainable params: 0
=====
```

Let's fit our model for 200 epochs

Loss in our model for 200 epochs.

In [96]: # Fit(Train) the Model

```
# Compile the model with Optimizer, Loss Function and Metrics
# Roc-Auc is not available in Keras as an off the shelf metric yet, so we will skip it here.

model_1.compile(SGD(learning_rate = .003), "binary_crossentropy", metrics=["accuracy"])
run_hist_1 = model_1.fit(X_train_norm, y_train, validation_data=(X_test_norm, y_test), epochs=200)
# the fit function returns the run history.
# It is very convenient, as it contains information about the model fit, iterations etc.
```

```
Epoch 1/200
18/18 [=====] - 0s 11ms/step - loss: 0.4865 - accuracy: 0.7708 - val_loss: 0.5082 - val_accuracy: 0.7708
Epoch 2/200
18/18 [=====] - 0s 4ms/step - loss: 0.4863 - accuracy: 0.7708 - val_loss: 0.5081 - val_accuracy: 0.7708
Epoch 3/200
18/18 [=====] - 0s 3ms/step - loss: 0.4862 - accuracy: 0.7708 - val_loss: 0.5080 - val_accuracy: 0.7708
Epoch 4/200
18/18 [=====] - 0s 4ms/step - loss: 0.4861 - accuracy: 0.7708 - val_loss: 0.5078 - val_accuracy: 0.7708
Epoch 5/200
18/18 [=====] - 0s 4ms/step - loss: 0.4860 - accuracy: 0.7743 - val_loss: 0.5077 - val_accuracy: 0.7708
Epoch 6/200
18/18 [=====] - 0s 5ms/step - loss: 0.4858 - accuracy: 0.7726 - val_loss: 0.5076 - val_accuracy: 0.7708
Epoch 7/200
18/18 [=====] - 0s 4ms/step - loss: 0.4857 - accuracy: 0.7726 - val_loss: 0.5075 - val_accuracy: 0.7708
Epoch 8/200
18/18 [=====] - 0s 4ms/step - loss: 0.4856 - accuracy: 0.7726 - val_loss: 0.5074 - val_accuracy: 0.7708
Epoch 9/200
18/18 [=====] - 0s 3ms/step - loss: 0.4854 - accuracy: 0.7743 - val_loss: 0.5073 - val_accuracy: 0.7708
Epoch 10/200
18/18 [=====] - 0s 4ms/step - loss: 0.4853 - accuracy: 0.7743 - val_loss: 0.5072 - val_accuracy: 0.7708
Epoch 11/200
18/18 [=====] - 0s 4ms/step - loss: 0.4852 - accuracy: 0.7743 - val_loss: 0.5071 - val_accuracy: 0.7656
Epoch 12/200
18/18 [=====] - 0s 4ms/step - loss: 0.4850 - accuracy: 0.7743 - val_loss: 0.5070 - val_accuracy: 0.7656
Epoch 13/200
18/18 [=====] - 0s 4ms/step - loss: 0.4849 - accuracy: 0.7743 - val_loss: 0.5069 - val_accuracy: 0.7656
Epoch 14/200
18/18 [=====] - 0s 4ms/step - loss: 0.4848 - accuracy: 0.7743 - val_loss: 0.5068 - val_accuracy: 0.7656
Epoch 15/200
18/18 [=====] - 0s 4ms/step - loss: 0.4847 - accuracy: 0.7743 - val_loss: 0.5066 - val_accuracy: 0.7656
Epoch 16/200
18/18 [=====] - 0s 4ms/step - loss: 0.4845 - accuracy: 0.7743 - val_loss: 0.5065 - val_accuracy: 0.7656
Epoch 17/200
18/18 [=====] - 0s 4ms/step - loss: 0.4844 - accuracy: 0.7743 - val_loss: 0.5064 - val_accuracy: 0.7656
Epoch 18/200
18/18 [=====] - 0s 4ms/step - loss: 0.4843 - accuracy: 0.7743 - val_loss: 0.5063 - val_accuracy: 0.7656
Epoch 19/200
18/18 [=====] - 0s 4ms/step - loss: 0.4842 - accuracy: 0.7743 - val_loss: 0.5062 - val_accuracy: 0.7656
Epoch 20/200
18/18 [=====] - 0s 3ms/step - loss: 0.4840 - accuracy: 0.7743 - val_loss: 0.5061 - val_accuracy: 0.7656
Epoch 21/200
18/18 [=====] - 0s 4ms/step - loss: 0.4839 - accuracy: 0.7743 - val_loss: 0.5060 - val_accuracy: 0.7656
Epoch 22/200
18/18 [=====] - 0s 4ms/step - loss: 0.4838 - accuracy: 0.7743 - val_loss: 0.5059 - val_accuracy: 0.7656
Epoch 23/200
18/18 [=====] - 0s 4ms/step - loss: 0.4837 - accuracy: 0.7743 - val_loss: 0.5058 - val_accuracy: 0.7656
Epoch 24/200
18/18 [=====] - 0s 4ms/step - loss: 0.4836 - accuracy: 0.7743 - val_loss: 0.5057 - val_accuracy: 0.7656
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

Epoch 191/200
18/18 [=====] - 0s 4ms/step - loss: 0.4699 - accuracy: 0.7830 - val_loss: 0.4952 - val_accuracy: 0.7552
Epoch 192/200
18/18 [=====] - 0s 3ms/step - loss: 0.4698 - accuracy: 0.7830 - val_loss: 0.4951 - val_accuracy: 0.7552
Epoch 193/200
18/18 [=====] - 0s 4ms/step - loss: 0.4698 - accuracy: 0.7830 - val_loss: 0.4951 - val_accuracy: 0.7552
Epoch 194/200
18/18 [=====] - 0s 4ms/step - loss: 0.4697 - accuracy: 0.7830 - val_loss: 0.4950 - val_accuracy: 0.7552
Epoch 195/200
18/18 [=====] - 0s 4ms/step - loss: 0.4697 - accuracy: 0.7830 - val_loss: 0.4950 - val_accuracy: 0.7552
Epoch 196/200
18/18 [=====] - 0s 4ms/step - loss: 0.4696 - accuracy: 0.7830 - val_loss: 0.4950 - val_accuracy: 0.7552
Epoch 197/200
18/18 [=====] - 0s 4ms/step - loss: 0.4696 - accuracy: 0.7830 - val_loss: 0.4949 - val_accuracy: 0.7552
Epoch 198/200
18/18 [=====] - 0s 4ms/step - loss: 0.4695 - accuracy: 0.7830 - val_loss: 0.4949 - val_accuracy: 0.7552
Epoch 199/200
18/18 [=====] - 0s 4ms/step - loss: 0.4695 - accuracy: 0.7830 - val_loss: 0.4949 - val_accuracy: 0.7552
Epoch 200/200
18/18 [=====] - 0s 4ms/step - loss: 0.4694 - accuracy: 0.7830 - val_loss: 0.4948 - val_accuracy: 0.7552

```

```

In [97]: ## Like we did for the Random Forest, we generate two kinds of predictions
# One is a hard decision, the other is a probabilistic score.
# Assuming your model is defined as `model_1`

# Make predictions on test data
y_pred_prob_nn_1 = model_1.predict(X_test_norm)

# Obtain predicted class labels by taking the index of the highest probability for each example
y_pred_class_nn_1 = y_pred_prob_nn_1.argmax(axis=-1)

6/6 [=====] - 0s 2ms/step

```

```

In [98]: y_pred_prob_nn_1[:10]

```

```

Out[98]: array([[0.48891714],
 [0.7093221 ],
 [0.2787088 ],
 [0.26989993],
 [0.19304611],
 [0.514679 ],
 [0.08812507],
 [0.31525382],
 [0.8304369 ],
 [0.19541515]], dtype=float32)

```

```

In [99]: y_pred_class_nn_1[:10]

```

```

Out[99]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])

```

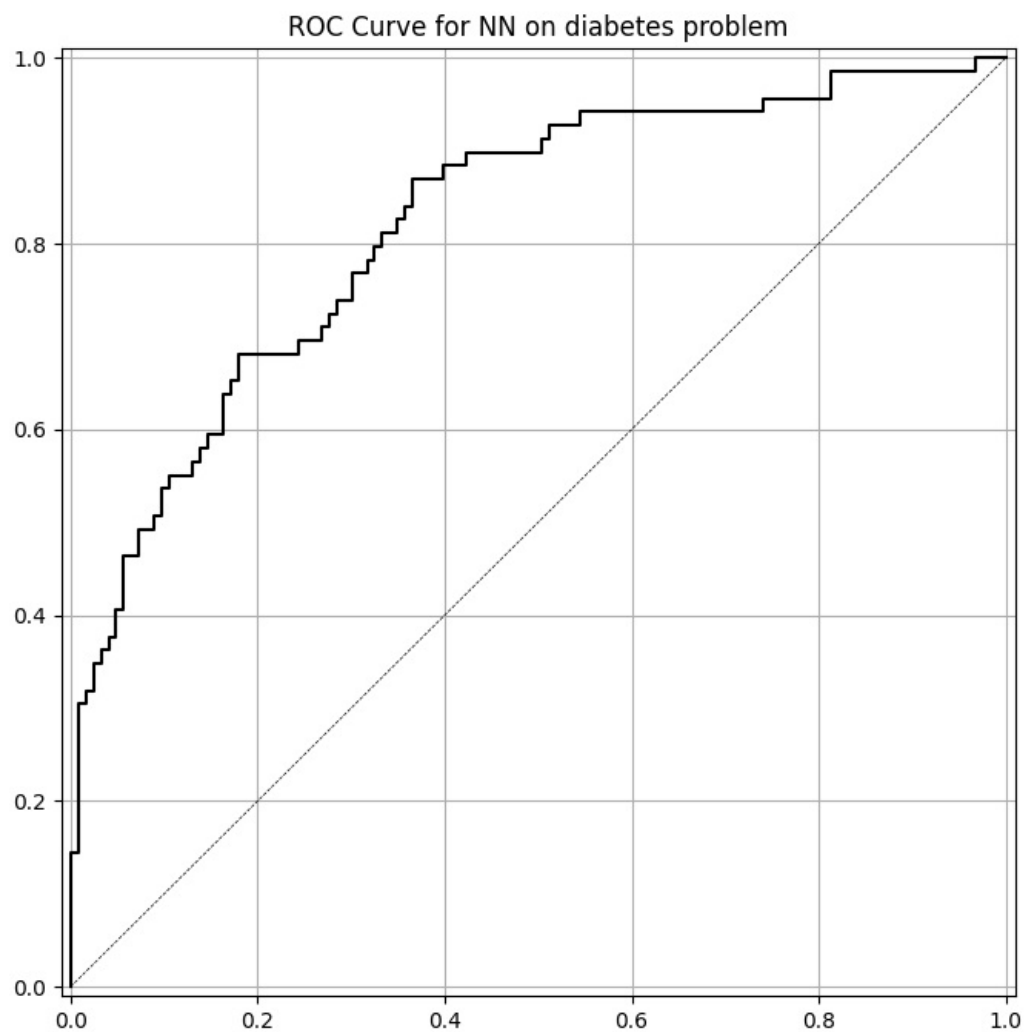
```

In [100]: # Print model performance and plot the roc curve
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')

accuracy is 0.641
roc-auc is 0.821

```



Let's look at the `run_hist_1` object that was created, specifically its `history` attribute.

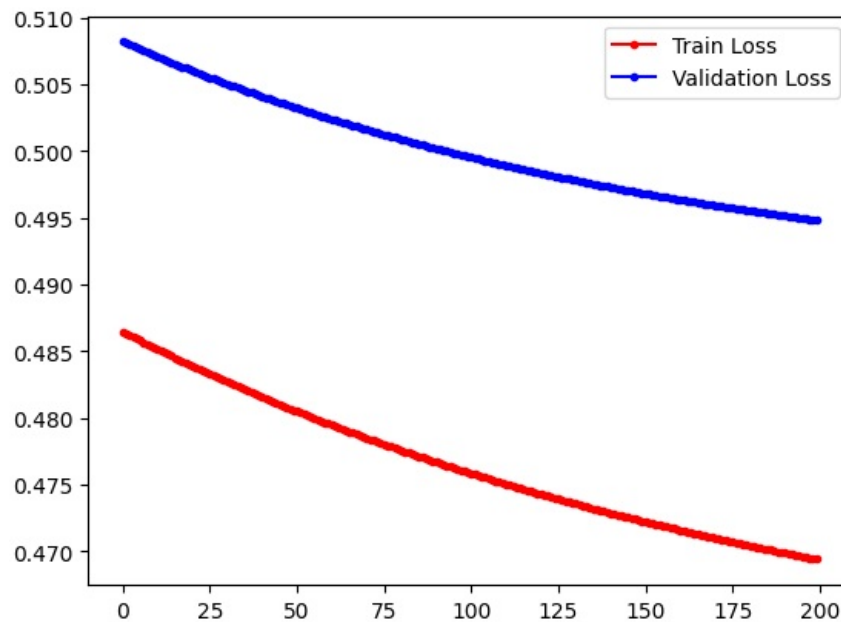
```
In [101]: run_hist_1.history.keys()
```

```
Out[101]: dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

Let's plot the training loss and the validation loss over the different epochs and see how it looks.

```
In [102]: fig, ax = plt.subplots()
ax.plot(run_hist_1.history["loss"], 'r', marker='.', label="Train Loss")
ax.plot(run_hist_1.history["val_loss"], 'b', marker='.', label="Validation Loss")
ax.legend()
```

```
Out[102]: <matplotlib.legend.Legend at 0x7f31be0bc4c0>
```



Looks like the losses are still going down on both the training set and the validation set. This suggests that the model might benefit from further training. Let's train the model a little more and see what happens. Note that it will pick up from where it left off. Train for 1000 more epochs.

```
In [103]: ## Note that when we call "fit" again, it picks up where it left off
run_hist_1b = model_1.fit(X_train_norm, y_train, validation_data=(X_test_norm, y_test), epochs=1000)

Epoch 1/1000
18/18 [=====] - 0s 6ms/step - loss: 0.4694 - accuracy: 0.7830 - val_loss: 0.4948 - val
_accuracy: 0.7552
Epoch 2/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4693 - accuracy: 0.7830 - val_loss: 0.4948 - val
_accuracy: 0.7552
Epoch 3/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4693 - accuracy: 0.7830 - val_loss: 0.4947 - val
_accuracy: 0.7552
Epoch 4/1000
18/18 [=====] - 0s 5ms/step - loss: 0.4692 - accuracy: 0.7830 - val_loss: 0.4947 - val
_accuracy: 0.7552
Epoch 5/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4692 - accuracy: 0.7830 - val_loss: 0.4947 - val
_accuracy: 0.7552
Epoch 6/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4691 - accuracy: 0.7830 - val_loss: 0.4946 - val
_accuracy: 0.7552
Epoch 7/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4691 - accuracy: 0.7830 - val_loss: 0.4946 - val
_accuracy: 0.7552
Epoch 8/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4690 - accuracy: 0.7847 - val_loss: 0.4946 - val
_accuracy: 0.7552
Epoch 9/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4690 - accuracy: 0.7847 - val_loss: 0.4945 - val
_accuracy: 0.7552
Epoch 10/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4689 - accuracy: 0.7847 - val_loss: 0.4945 - val
_accuracy: 0.7552
Epoch 11/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4689 - accuracy: 0.7847 - val_loss: 0.4945 - val
_accuracy: 0.7552
Epoch 12/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4688 - accuracy: 0.7847 - val_loss: 0.4944 - val
_accuracy: 0.7552
Epoch 13/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4688 - accuracy: 0.7830 - val_loss: 0.4944 - val
_accuracy: 0.7552
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```

18/18 [=====] - 0s 4ms/step - loss: 0.4559 - accuracy: 0.7812 - val_loss: 0.4892 - val
_accuracy: 0.7708
Epoch 983/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4559 - accuracy: 0.7812 - val_loss: 0.4892 - val
_accuracy: 0.7708
Epoch 984/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4559 - accuracy: 0.7812 - val_loss: 0.4892 - val
_accuracy: 0.7708
Epoch 985/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4559 - accuracy: 0.7830 - val_loss: 0.4892 - val
_accuracy: 0.7708
Epoch 986/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4558 - accuracy: 0.7830 - val_loss: 0.4892 - val
_accuracy: 0.7708
Epoch 987/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4559 - accuracy: 0.7812 - val_loss: 0.4892 - val
_accuracy: 0.7708
Epoch 988/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4558 - accuracy: 0.7812 - val_loss: 0.4893 - val
_accuracy: 0.7708
Epoch 989/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4558 - accuracy: 0.7830 - val_loss: 0.4893 - val
_accuracy: 0.7708
Epoch 990/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4558 - accuracy: 0.7812 - val_loss: 0.4893 - val
_accuracy: 0.7708
Epoch 991/1000
18/18 [=====] - 0s 3ms/step - loss: 0.4558 - accuracy: 0.7830 - val_loss: 0.4893 - val
_accuracy: 0.7708
Epoch 992/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4558 - accuracy: 0.7830 - val_loss: 0.4893 - val
_accuracy: 0.7708
Epoch 993/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4558 - accuracy: 0.7830 - val_loss: 0.4893 - val
_accuracy: 0.7708
Epoch 994/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4558 - accuracy: 0.7812 - val_loss: 0.4893 - val
_accuracy: 0.7708
Epoch 995/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4558 - accuracy: 0.7812 - val_loss: 0.4893 - val
_accuracy: 0.7708
Epoch 996/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4558 - accuracy: 0.7812 - val_loss: 0.4893 - val
_accuracy: 0.7708
Epoch 997/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4558 - accuracy: 0.7830 - val_loss: 0.4893 - val
_accuracy: 0.7708
Epoch 998/1000
18/18 [=====] - 0s 3ms/step - loss: 0.4558 - accuracy: 0.7812 - val_loss: 0.4893 - val
_accuracy: 0.7708
Epoch 999/1000
18/18 [=====] - 0s 4ms/step - loss: 0.4558 - accuracy: 0.7830 - val_loss: 0.4893 - val
_accuracy: 0.7708
Epoch 1000/1000
18/18 [=====] - 0s 3ms/step - loss: 0.4558 - accuracy: 0.7830 - val_loss: 0.4893 - val
_accuracy: 0.7708

```

```

In [104]: n = len(run_hist_1.history["loss"])
m = len(run_hist_1b.history['loss'])
fig, ax = plt.subplots(figsize=(16, 8))

ax.plot(range(n), run_hist_1.history["loss"], 'r', marker='.', label="Train Loss - Run 1")
ax.plot(range(n, n+m), run_hist_1b.history["loss"], 'hotpink', marker='.', label="Train Loss - Run 2")

ax.plot(range(n), run_hist_1.history["val_loss"], 'b', marker='.', label="Validation Loss - Run 1")
ax.plot(range(n, n+m), run_hist_1b.history["val_loss"], 'LightSkyBlue', marker='.', label="Validation Loss - Run 2")

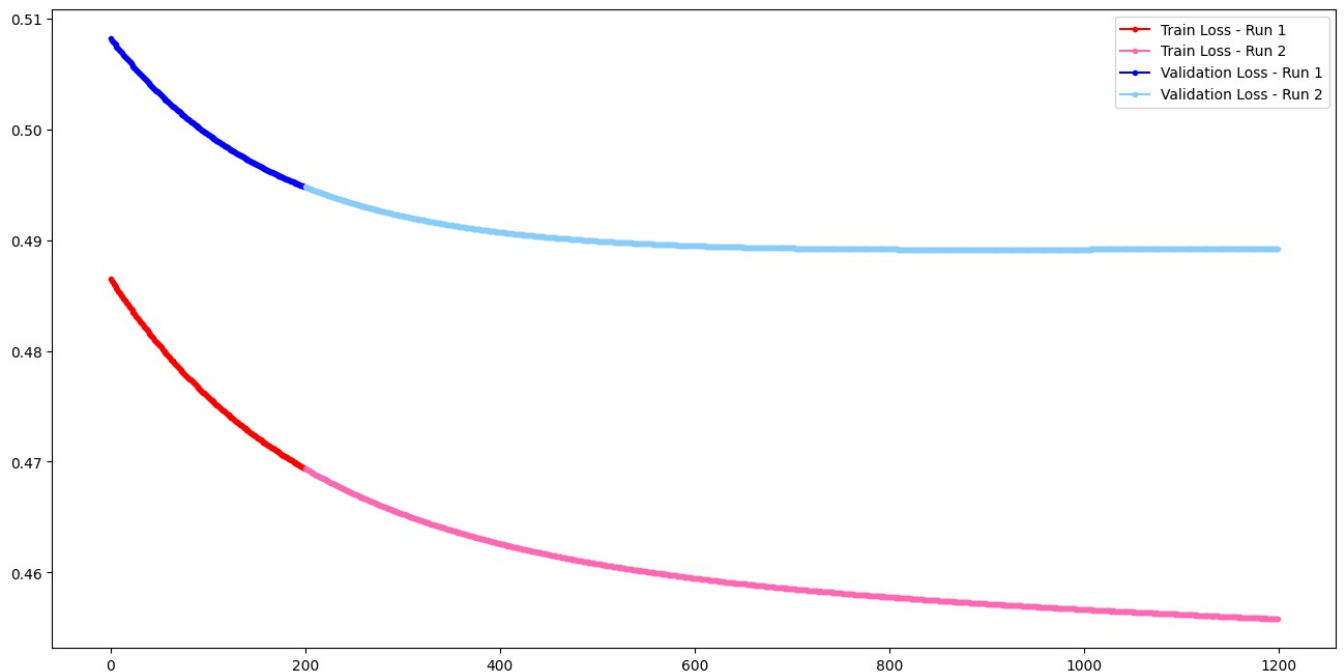
ax.legend()

```

```

Out[104]: <matplotlib.legend.Legend at 0x7f31bcb238b0>

```



Note that this graph begins where the other left off. While the training loss is still going down, it looks like the validation loss has stabilized (or even gotten worse!). This suggests that our network will not benefit from further training. What is the appropriate number of epochs?

Rebuild Neural Network

For this, do the following in the cells below:

- Build a model with two hidden layers, each with 6 nodes
- Use the "relu" activation function for the hidden layers, and "sigmoid" for the final layer
- Use a learning rate of .003 and train for 1500 epochs
- Graph the trajectory of the loss functions, accuracy on both train and test set
- Plot the roc curve for the predictions

Experiment with different learning rates, numbers of epochs, and network structures

```
In [106]: model_2 = Sequential()
model_2.add(Dense(6, input_shape=(8,), activation="relu"))
model_2.add(Dense(6, activation="relu"))
model_2.add(Dense(1, activation="sigmoid"))

model_2.compile(SGD(learning_rate = .003), "binary_crossentropy", metrics=["accuracy"])
run_hist_2 = model_2.fit(X_train_norm, y_train, validation_data=(X_test_norm, y_test), epochs=1500)
```

```
Epoch 1/1500
18/18 [=====] - 1s 11ms/step - loss: 0.9109 - accuracy: 0.3767 - val_loss: 0.9485 - val
_accuracy: 0.3646
Epoch 2/1500
18/18 [=====] - 0s 4ms/step - loss: 0.8947 - accuracy: 0.3819 - val_loss: 0.9308 - val
_accuracy: 0.3698
Epoch 3/1500
18/18 [=====] - 0s 4ms/step - loss: 0.8798 - accuracy: 0.3819 - val_loss: 0.9147 - val
_accuracy: 0.3802
Epoch 4/1500
18/18 [=====] - 0s 4ms/step - loss: 0.8662 - accuracy: 0.3924 - val_loss: 0.8999 - val
_accuracy: 0.3802
Epoch 5/1500
18/18 [=====] - 0s 4ms/step - loss: 0.8538 - accuracy: 0.3958 - val_loss: 0.8864 - val
_accuracy: 0.4010
Epoch 6/1500
```

18/18 [=====] - 0s 4ms/step - loss: 0.8424 - accuracy: 0.4062 - val_loss: 0.8739 - val
_accuracy: 0.4062
Epoch 7/1500
18/18 [=====] - 0s 4ms/step - loss: 0.8319 - accuracy: 0.4149 - val_loss: 0.8623 - val
_accuracy: 0.4115
Epoch 8/1500
18/18 [=====] - 0s 4ms/step - loss: 0.8221 - accuracy: 0.4253 - val_loss: 0.8515 - val
_accuracy: 0.4167
Epoch 9/1500
18/18 [=====] - 0s 4ms/step - loss: 0.8130 - accuracy: 0.4340 - val_loss: 0.8414 - val
_accuracy: 0.4115
Epoch 10/1500
18/18 [=====] - 0s 4ms/step - loss: 0.8046 - accuracy: 0.4392 - val_loss: 0.8320 - val
_accuracy: 0.4167
Epoch 11/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7967 - accuracy: 0.4549 - val_loss: 0.8232 - val
_accuracy: 0.4219
Epoch 12/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7893 - accuracy: 0.4531 - val_loss: 0.8149 - val
_accuracy: 0.4375
Epoch 13/1500
18/18 [=====] - 0s 6ms/step - loss: 0.7823 - accuracy: 0.4583 - val_loss: 0.8072 - val
_accuracy: 0.4375
Epoch 14/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7758 - accuracy: 0.4670 - val_loss: 0.7999 - val
_accuracy: 0.4427
Epoch 15/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7696 - accuracy: 0.4774 - val_loss: 0.7931 - val
_accuracy: 0.4427
Epoch 16/1500
18/18 [=====] - 0s 5ms/step - loss: 0.7638 - accuracy: 0.4913 - val_loss: 0.7866 - val
_accuracy: 0.4375
Epoch 17/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7583 - accuracy: 0.4948 - val_loss: 0.7804 - val
_accuracy: 0.4427
Epoch 18/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7531 - accuracy: 0.5000 - val_loss: 0.7746 - val
_accuracy: 0.4427
Epoch 19/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7481 - accuracy: 0.5087 - val_loss: 0.7691 - val
_accuracy: 0.4531
Epoch 20/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7433 - accuracy: 0.5191 - val_loss: 0.7638 - val
_accuracy: 0.4688
Epoch 21/1500
18/18 [=====] - 0s 5ms/step - loss: 0.7389 - accuracy: 0.5278 - val_loss: 0.7588 - val
_accuracy: 0.4792
Epoch 22/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7345 - accuracy: 0.5347 - val_loss: 0.7540 - val
_accuracy: 0.4896
Epoch 23/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7303 - accuracy: 0.5434 - val_loss: 0.7494 - val
_accuracy: 0.4948
Epoch 24/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7264 - accuracy: 0.5538 - val_loss: 0.7450 - val
_accuracy: 0.5000
Epoch 25/1500
18/18 [=====] - 0s 5ms/step - loss: 0.7225 - accuracy: 0.5608 - val_loss: 0.7408 - val
_accuracy: 0.5052
Epoch 26/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7189 - accuracy: 0.5816 - val_loss: 0.7367 - val
_accuracy: 0.5104
Epoch 27/1500
18/18 [=====] - 0s 5ms/step - loss: 0.7153 - accuracy: 0.5868 - val_loss: 0.7328 - val
_accuracy: 0.5104
Epoch 28/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7118 - accuracy: 0.5938 - val_loss: 0.7290 - val
_accuracy: 0.5156
Epoch 29/1500
18/18 [=====] - 0s 5ms/step - loss: 0.7086 - accuracy: 0.6042 - val_loss: 0.7253 - val
_accuracy: 0.5208
Epoch 30/1500
18/18 [=====] - 0s 4ms/step - loss: 0.7055 - accuracy: 0.6128 - val_loss: 0.7218 - val
_accuracy: 0.5365
Epoch 31/1500
18/18 [=====] - 0s 5ms/step - loss: 0.7024 - accuracy: 0.6146 - val_loss: 0.7184 - val
_accuracy: 0.5469
Epoch 32/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6994 - accuracy: 0.6215 - val_loss: 0.7151 - val
_accuracy: 0.5573
Epoch 33/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6966 - accuracy: 0.6198 - val_loss: 0.7119 - val
_accuracy: 0.5625

Epoch 34/1500
18/18 [=====] - 0s 5ms/step - loss: 0.6938 - accuracy: 0.6198 - val_loss: 0.7088 - val_accuracy: 0.5677
Epoch 35/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6912 - accuracy: 0.6250 - val_loss: 0.7058 - val_accuracy: 0.5729
Epoch 36/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6886 - accuracy: 0.6250 - val_loss: 0.7029 - val_accuracy: 0.5729
Epoch 37/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6861 - accuracy: 0.6285 - val_loss: 0.7000 - val_accuracy: 0.5833
Epoch 38/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6837 - accuracy: 0.6319 - val_loss: 0.6973 - val_accuracy: 0.5938
Epoch 39/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6813 - accuracy: 0.6372 - val_loss: 0.6946 - val_accuracy: 0.5990
Epoch 40/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6790 - accuracy: 0.6424 - val_loss: 0.6919 - val_accuracy: 0.5990
Epoch 41/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6768 - accuracy: 0.6458 - val_loss: 0.6894 - val_accuracy: 0.6042
Epoch 42/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6746 - accuracy: 0.6510 - val_loss: 0.6869 - val_accuracy: 0.6094
Epoch 43/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6725 - accuracy: 0.6632 - val_loss: 0.6845 - val_accuracy: 0.6094
Epoch 44/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6704 - accuracy: 0.6632 - val_loss: 0.6821 - val_accuracy: 0.6146
Epoch 45/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6685 - accuracy: 0.6684 - val_loss: 0.6798 - val_accuracy: 0.6146
Epoch 46/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6665 - accuracy: 0.6684 - val_loss: 0.6776 - val_accuracy: 0.6146
Epoch 47/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6647 - accuracy: 0.6701 - val_loss: 0.6754 - val_accuracy: 0.6354
Epoch 48/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6628 - accuracy: 0.6753 - val_loss: 0.6733 - val_accuracy: 0.6354
Epoch 49/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6610 - accuracy: 0.6771 - val_loss: 0.6712 - val_accuracy: 0.6354
Epoch 50/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6593 - accuracy: 0.6771 - val_loss: 0.6691 - val_accuracy: 0.6406
Epoch 51/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6575 - accuracy: 0.6823 - val_loss: 0.6671 - val_accuracy: 0.6458
Epoch 52/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6559 - accuracy: 0.6875 - val_loss: 0.6651 - val_accuracy: 0.6458
Epoch 53/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6542 - accuracy: 0.6892 - val_loss: 0.6632 - val_accuracy: 0.6510
Epoch 54/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6525 - accuracy: 0.6927 - val_loss: 0.6613 - val_accuracy: 0.6510
Epoch 55/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6510 - accuracy: 0.6944 - val_loss: 0.6595 - val_accuracy: 0.6510
Epoch 56/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6494 - accuracy: 0.6944 - val_loss: 0.6577 - val_accuracy: 0.6510
Epoch 57/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6479 - accuracy: 0.6962 - val_loss: 0.6559 - val_accuracy: 0.6562
Epoch 58/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6464 - accuracy: 0.7014 - val_loss: 0.6541 - val_accuracy: 0.6615
Epoch 59/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6449 - accuracy: 0.6979 - val_loss: 0.6524 - val_accuracy: 0.6615
Epoch 60/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6435 - accuracy: 0.7014 - val_loss: 0.6507 - val_accuracy: 0.6562
Epoch 61/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6420 - accuracy: 0.7014 - val_loss: 0.6491 - val

```
_accuracy: 0.6719
Epoch 62/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6407 - accuracy: 0.7031 - val_loss: 0.6474 - val
_accuracy: 0.6771
Epoch 63/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6393 - accuracy: 0.7083 - val_loss: 0.6458 - val
_accuracy: 0.6823
Epoch 64/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6379 - accuracy: 0.7066 - val_loss: 0.6442 - val
_accuracy: 0.6771
Epoch 65/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6366 - accuracy: 0.7101 - val_loss: 0.6427 - val
_accuracy: 0.6823
Epoch 66/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6353 - accuracy: 0.7135 - val_loss: 0.6411 - val
_accuracy: 0.6771
Epoch 67/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6340 - accuracy: 0.7118 - val_loss: 0.6396 - val
_accuracy: 0.6823
Epoch 68/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6327 - accuracy: 0.7135 - val_loss: 0.6382 - val
_accuracy: 0.6875
Epoch 69/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6315 - accuracy: 0.7170 - val_loss: 0.6367 - val
_accuracy: 0.6875
Epoch 70/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6302 - accuracy: 0.7153 - val_loss: 0.6353 - val
_accuracy: 0.6979
Epoch 71/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6290 - accuracy: 0.7188 - val_loss: 0.6339 - val
_accuracy: 0.7031
Epoch 72/1500
18/18 [=====] - 0s 9ms/step - loss: 0.6278 - accuracy: 0.7170 - val_loss: 0.6325 - val
_accuracy: 0.6979
Epoch 73/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6267 - accuracy: 0.7170 - val_loss: 0.6311 - val
_accuracy: 0.6979
Epoch 74/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6254 - accuracy: 0.7170 - val_loss: 0.6298 - val
_accuracy: 0.6979
Epoch 75/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6243 - accuracy: 0.7153 - val_loss: 0.6284 - val
_accuracy: 0.6927
Epoch 76/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6232 - accuracy: 0.7153 - val_loss: 0.6271 - val
_accuracy: 0.6979
Epoch 77/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6221 - accuracy: 0.7118 - val_loss: 0.6258 - val
_accuracy: 0.6979
Epoch 78/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6210 - accuracy: 0.7135 - val_loss: 0.6245 - val
_accuracy: 0.7083
Epoch 79/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6199 - accuracy: 0.7135 - val_loss: 0.6233 - val
_accuracy: 0.7083
Epoch 80/1500
18/18 [=====] - 0s 5ms/step - loss: 0.6188 - accuracy: 0.7170 - val_loss: 0.6220 - val
_accuracy: 0.7083
Epoch 81/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6178 - accuracy: 0.7170 - val_loss: 0.6208 - val
_accuracy: 0.7135
Epoch 82/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6167 - accuracy: 0.7205 - val_loss: 0.6196 - val
_accuracy: 0.7135
Epoch 83/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6157 - accuracy: 0.7240 - val_loss: 0.6184 - val
_accuracy: 0.7135
Epoch 84/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6147 - accuracy: 0.7257 - val_loss: 0.6173 - val
_accuracy: 0.7135
Epoch 85/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6137 - accuracy: 0.7257 - val_loss: 0.6161 - val
_accuracy: 0.7188
Epoch 86/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6127 - accuracy: 0.7274 - val_loss: 0.6149 - val
_accuracy: 0.7188
Epoch 87/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6118 - accuracy: 0.7309 - val_loss: 0.6138 - val
_accuracy: 0.7188
Epoch 88/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6108 - accuracy: 0.7292 - val_loss: 0.6127 - val
_accuracy: 0.7188
Epoch 89/1500
```

18/18 [=====] - 0s 4ms/step - loss: 0.6098 - accuracy: 0.7309 - val_loss: 0.6116 - val
_accuracy: 0.7188
Epoch 90/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6089 - accuracy: 0.7292 - val_loss: 0.6105 - val
_accuracy: 0.7240
Epoch 91/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6080 - accuracy: 0.7326 - val_loss: 0.6094 - val
_accuracy: 0.7240
Epoch 92/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6070 - accuracy: 0.7309 - val_loss: 0.6084 - val
_accuracy: 0.7240
Epoch 93/1500
18/18 [=====] - 0s 5ms/step - loss: 0.6061 - accuracy: 0.7326 - val_loss: 0.6073 - val
_accuracy: 0.7188
Epoch 94/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6052 - accuracy: 0.7326 - val_loss: 0.6063 - val
_accuracy: 0.7188
Epoch 95/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6043 - accuracy: 0.7309 - val_loss: 0.6052 - val
_accuracy: 0.7188
Epoch 96/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6034 - accuracy: 0.7361 - val_loss: 0.6042 - val
_accuracy: 0.7188
Epoch 97/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6025 - accuracy: 0.7361 - val_loss: 0.6032 - val
_accuracy: 0.7188
Epoch 98/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6017 - accuracy: 0.7344 - val_loss: 0.6022 - val
_accuracy: 0.7135
Epoch 99/1500
18/18 [=====] - 0s 4ms/step - loss: 0.6008 - accuracy: 0.7344 - val_loss: 0.6012 - val
_accuracy: 0.7188
Epoch 100/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5999 - accuracy: 0.7344 - val_loss: 0.6003 - val
_accuracy: 0.7188
Epoch 101/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5991 - accuracy: 0.7292 - val_loss: 0.5993 - val
_accuracy: 0.7188
Epoch 102/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5982 - accuracy: 0.7326 - val_loss: 0.5984 - val
_accuracy: 0.7188
Epoch 103/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5974 - accuracy: 0.7309 - val_loss: 0.5974 - val
_accuracy: 0.7188
Epoch 104/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5965 - accuracy: 0.7396 - val_loss: 0.5965 - val
_accuracy: 0.7188
Epoch 105/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5957 - accuracy: 0.7396 - val_loss: 0.5956 - val
_accuracy: 0.7188
Epoch 106/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5949 - accuracy: 0.7448 - val_loss: 0.5947 - val
_accuracy: 0.7188
Epoch 107/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5942 - accuracy: 0.7413 - val_loss: 0.5938 - val
_accuracy: 0.7188
Epoch 108/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5934 - accuracy: 0.7431 - val_loss: 0.5929 - val
_accuracy: 0.7188
Epoch 109/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5926 - accuracy: 0.7431 - val_loss: 0.5920 - val
_accuracy: 0.7188
Epoch 110/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5918 - accuracy: 0.7413 - val_loss: 0.5912 - val
_accuracy: 0.7188
Epoch 111/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5910 - accuracy: 0.7448 - val_loss: 0.5903 - val
_accuracy: 0.7188
Epoch 112/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5903 - accuracy: 0.7431 - val_loss: 0.5895 - val
_accuracy: 0.7188
Epoch 113/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5895 - accuracy: 0.7465 - val_loss: 0.5887 - val
_accuracy: 0.7188
Epoch 114/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5888 - accuracy: 0.7465 - val_loss: 0.5878 - val
_accuracy: 0.7240
Epoch 115/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5880 - accuracy: 0.7483 - val_loss: 0.5870 - val
_accuracy: 0.7240
Epoch 116/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5873 - accuracy: 0.7500 - val_loss: 0.5862 - val
_accuracy: 0.7240

Epoch 117/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5866 - accuracy: 0.7483 - val_loss: 0.5854 - val_accuracy: 0.7240
Epoch 118/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5859 - accuracy: 0.7483 - val_loss: 0.5846 - val_accuracy: 0.7240
Epoch 119/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5852 - accuracy: 0.7483 - val_loss: 0.5838 - val_accuracy: 0.7240
Epoch 120/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5844 - accuracy: 0.7483 - val_loss: 0.5830 - val_accuracy: 0.7240
Epoch 121/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5838 - accuracy: 0.7465 - val_loss: 0.5822 - val_accuracy: 0.7292
Epoch 122/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5830 - accuracy: 0.7465 - val_loss: 0.5815 - val_accuracy: 0.7292
Epoch 123/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5824 - accuracy: 0.7500 - val_loss: 0.5807 - val_accuracy: 0.7292
Epoch 124/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5817 - accuracy: 0.7500 - val_loss: 0.5799 - val_accuracy: 0.7292
Epoch 125/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5810 - accuracy: 0.7500 - val_loss: 0.5792 - val_accuracy: 0.7292
Epoch 126/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5804 - accuracy: 0.7500 - val_loss: 0.5785 - val_accuracy: 0.7344
Epoch 127/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5797 - accuracy: 0.7483 - val_loss: 0.5777 - val_accuracy: 0.7344
Epoch 128/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5791 - accuracy: 0.7500 - val_loss: 0.5770 - val_accuracy: 0.7344
Epoch 129/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5785 - accuracy: 0.7500 - val_loss: 0.5763 - val_accuracy: 0.7344
Epoch 130/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5778 - accuracy: 0.7500 - val_loss: 0.5757 - val_accuracy: 0.7344
Epoch 131/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5772 - accuracy: 0.7500 - val_loss: 0.5750 - val_accuracy: 0.7292
Epoch 132/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5765 - accuracy: 0.7535 - val_loss: 0.5743 - val_accuracy: 0.7292
Epoch 133/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5759 - accuracy: 0.7552 - val_loss: 0.5736 - val_accuracy: 0.7292
Epoch 134/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5753 - accuracy: 0.7535 - val_loss: 0.5730 - val_accuracy: 0.7292
Epoch 135/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5747 - accuracy: 0.7552 - val_loss: 0.5723 - val_accuracy: 0.7292
Epoch 136/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5741 - accuracy: 0.7552 - val_loss: 0.5717 - val_accuracy: 0.7292
Epoch 137/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5734 - accuracy: 0.7552 - val_loss: 0.5711 - val_accuracy: 0.7292
Epoch 138/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5728 - accuracy: 0.7535 - val_loss: 0.5705 - val_accuracy: 0.7292
Epoch 139/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5723 - accuracy: 0.7535 - val_loss: 0.5698 - val_accuracy: 0.7292
Epoch 140/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5717 - accuracy: 0.7517 - val_loss: 0.5692 - val_accuracy: 0.7292
Epoch 141/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5711 - accuracy: 0.7517 - val_loss: 0.5687 - val_accuracy: 0.7240
Epoch 142/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5705 - accuracy: 0.7517 - val_loss: 0.5681 - val_accuracy: 0.7240
Epoch 143/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5699 - accuracy: 0.7517 - val_loss: 0.5675 - val_accuracy: 0.7240
Epoch 144/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5693 - accuracy: 0.7517 - val_loss: 0.5669 - val_accuracy: 0.7240

```
_accuracy: 0.7240
Epoch 145/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5688 - accuracy: 0.7517 - val_loss: 0.5663 - val
_accuracy: 0.7240
Epoch 146/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5682 - accuracy: 0.7517 - val_loss: 0.5658 - val
_accuracy: 0.7240
Epoch 147/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5676 - accuracy: 0.7535 - val_loss: 0.5652 - val
_accuracy: 0.7292
Epoch 148/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5671 - accuracy: 0.7535 - val_loss: 0.5646 - val
_accuracy: 0.7292
Epoch 149/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5666 - accuracy: 0.7535 - val_loss: 0.5641 - val
_accuracy: 0.7292
Epoch 150/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5660 - accuracy: 0.7517 - val_loss: 0.5636 - val
_accuracy: 0.7292
Epoch 151/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5655 - accuracy: 0.7552 - val_loss: 0.5630 - val
_accuracy: 0.7292
Epoch 152/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5649 - accuracy: 0.7535 - val_loss: 0.5625 - val
_accuracy: 0.7292
Epoch 153/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5644 - accuracy: 0.7552 - val_loss: 0.5620 - val
_accuracy: 0.7344
Epoch 154/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5639 - accuracy: 0.7552 - val_loss: 0.5615 - val
_accuracy: 0.7344
Epoch 155/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5634 - accuracy: 0.7552 - val_loss: 0.5609 - val
_accuracy: 0.7344
Epoch 156/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5629 - accuracy: 0.7552 - val_loss: 0.5604 - val
_accuracy: 0.7344
Epoch 157/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5623 - accuracy: 0.7552 - val_loss: 0.5599 - val
_accuracy: 0.7344
Epoch 158/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5618 - accuracy: 0.7587 - val_loss: 0.5594 - val
_accuracy: 0.7344
Epoch 159/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5613 - accuracy: 0.7587 - val_loss: 0.5589 - val
_accuracy: 0.7292
Epoch 160/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5608 - accuracy: 0.7587 - val_loss: 0.5585 - val
_accuracy: 0.7292
Epoch 161/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5604 - accuracy: 0.7587 - val_loss: 0.5580 - val
_accuracy: 0.7292
Epoch 162/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5598 - accuracy: 0.7587 - val_loss: 0.5575 - val
_accuracy: 0.7292
Epoch 163/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5593 - accuracy: 0.7587 - val_loss: 0.5570 - val
_accuracy: 0.7344
Epoch 164/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5588 - accuracy: 0.7604 - val_loss: 0.5565 - val
_accuracy: 0.7292
Epoch 165/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5584 - accuracy: 0.7587 - val_loss: 0.5560 - val
_accuracy: 0.7292
Epoch 166/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5579 - accuracy: 0.7587 - val_loss: 0.5556 - val
_accuracy: 0.7292
Epoch 167/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5574 - accuracy: 0.7622 - val_loss: 0.5551 - val
_accuracy: 0.7292
Epoch 168/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5569 - accuracy: 0.7604 - val_loss: 0.5546 - val
_accuracy: 0.7292
Epoch 169/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5564 - accuracy: 0.7587 - val_loss: 0.5542 - val
_accuracy: 0.7292
Epoch 170/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5559 - accuracy: 0.7622 - val_loss: 0.5537 - val
_accuracy: 0.7292
Epoch 171/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5555 - accuracy: 0.7622 - val_loss: 0.5533 - val
_accuracy: 0.7240
Epoch 172/1500
```


18/18 [=====] - 0s 4ms/step - loss: 0.5551 - accuracy: 0.7622 - val_loss: 0.5528 - val
_accuracy: 0.7240
Epoch 173/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5546 - accuracy: 0.7622 - val_loss: 0.5524 - val
_accuracy: 0.7240
Epoch 174/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5542 - accuracy: 0.7622 - val_loss: 0.5520 - val
_accuracy: 0.7240
Epoch 175/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5537 - accuracy: 0.7622 - val_loss: 0.5516 - val
_accuracy: 0.7240
Epoch 176/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5532 - accuracy: 0.7622 - val_loss: 0.5512 - val
_accuracy: 0.7240
Epoch 177/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5528 - accuracy: 0.7622 - val_loss: 0.5507 - val
_accuracy: 0.7240
Epoch 178/1500
18/18 [=====] - 0s 5ms/step - loss: 0.5523 - accuracy: 0.7604 - val_loss: 0.5503 - val
_accuracy: 0.7240
Epoch 179/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5519 - accuracy: 0.7604 - val_loss: 0.5499 - val
_accuracy: 0.7240
Epoch 180/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5514 - accuracy: 0.7604 - val_loss: 0.5495 - val
_accuracy: 0.7240
Epoch 181/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5510 - accuracy: 0.7604 - val_loss: 0.5491 - val
_accuracy: 0.7240
Epoch 182/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5505 - accuracy: 0.7604 - val_loss: 0.5487 - val
_accuracy: 0.7240
Epoch 183/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5501 - accuracy: 0.7604 - val_loss: 0.5483 - val
_accuracy: 0.7240
Epoch 184/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5497 - accuracy: 0.7622 - val_loss: 0.5479 - val
_accuracy: 0.7240
Epoch 185/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5492 - accuracy: 0.7587 - val_loss: 0.5475 - val
_accuracy: 0.7240
Epoch 186/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5489 - accuracy: 0.7604 - val_loss: 0.5472 - val
_accuracy: 0.7240
Epoch 187/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5484 - accuracy: 0.7604 - val_loss: 0.5468 - val
_accuracy: 0.7240
Epoch 188/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5480 - accuracy: 0.7604 - val_loss: 0.5464 - val
_accuracy: 0.7240
Epoch 189/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5476 - accuracy: 0.7622 - val_loss: 0.5461 - val
_accuracy: 0.7240
Epoch 190/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5472 - accuracy: 0.7639 - val_loss: 0.5457 - val
_accuracy: 0.7240
Epoch 191/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5467 - accuracy: 0.7622 - val_loss: 0.5454 - val
_accuracy: 0.7240
Epoch 192/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5464 - accuracy: 0.7639 - val_loss: 0.5451 - val
_accuracy: 0.7240
Epoch 193/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5460 - accuracy: 0.7622 - val_loss: 0.5447 - val
_accuracy: 0.7240
Epoch 194/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5456 - accuracy: 0.7622 - val_loss: 0.5444 - val
_accuracy: 0.7240
Epoch 195/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5453 - accuracy: 0.7622 - val_loss: 0.5441 - val
_accuracy: 0.7240
Epoch 196/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5448 - accuracy: 0.7639 - val_loss: 0.5438 - val
_accuracy: 0.7240
Epoch 197/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5445 - accuracy: 0.7639 - val_loss: 0.5435 - val
_accuracy: 0.7240
Epoch 198/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5441 - accuracy: 0.7639 - val_loss: 0.5432 - val
_accuracy: 0.7240
Epoch 199/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5437 - accuracy: 0.7622 - val_loss: 0.5428 - val
_accuracy: 0.7240

Epoch 200/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5434 - accuracy: 0.7656 - val_loss: 0.5425 - val_accuracy: 0.7240
Epoch 201/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5431 - accuracy: 0.7622 - val_loss: 0.5422 - val_accuracy: 0.7292
Epoch 202/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5426 - accuracy: 0.7639 - val_loss: 0.5419 - val_accuracy: 0.7292
Epoch 203/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5423 - accuracy: 0.7604 - val_loss: 0.5416 - val_accuracy: 0.7344
Epoch 204/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5419 - accuracy: 0.7639 - val_loss: 0.5414 - val_accuracy: 0.7344
Epoch 205/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5416 - accuracy: 0.7656 - val_loss: 0.5411 - val_accuracy: 0.7344
Epoch 206/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5412 - accuracy: 0.7604 - val_loss: 0.5408 - val_accuracy: 0.7344
Epoch 207/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5409 - accuracy: 0.7604 - val_loss: 0.5405 - val_accuracy: 0.7344
Epoch 208/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5405 - accuracy: 0.7604 - val_loss: 0.5402 - val_accuracy: 0.7344
Epoch 209/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5401 - accuracy: 0.7622 - val_loss: 0.5399 - val_accuracy: 0.7344
Epoch 210/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5398 - accuracy: 0.7604 - val_loss: 0.5396 - val_accuracy: 0.7344
Epoch 211/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5395 - accuracy: 0.7622 - val_loss: 0.5394 - val_accuracy: 0.7344
Epoch 212/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5391 - accuracy: 0.7604 - val_loss: 0.5391 - val_accuracy: 0.7344
Epoch 213/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5387 - accuracy: 0.7622 - val_loss: 0.5388 - val_accuracy: 0.7344
Epoch 214/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5384 - accuracy: 0.7622 - val_loss: 0.5386 - val_accuracy: 0.7344
Epoch 215/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5381 - accuracy: 0.7622 - val_loss: 0.5383 - val_accuracy: 0.7344
Epoch 216/1500
18/18 [=====] - 0s 5ms/step - loss: 0.5378 - accuracy: 0.7639 - val_loss: 0.5380 - val_accuracy: 0.7344
Epoch 217/1500
18/18 [=====] - 0s 5ms/step - loss: 0.5375 - accuracy: 0.7622 - val_loss: 0.5378 - val_accuracy: 0.7344
Epoch 218/1500
18/18 [=====] - 0s 6ms/step - loss: 0.5371 - accuracy: 0.7639 - val_loss: 0.5375 - val_accuracy: 0.7344
Epoch 219/1500
18/18 [=====] - 0s 5ms/step - loss: 0.5368 - accuracy: 0.7656 - val_loss: 0.5373 - val_accuracy: 0.7344
Epoch 220/1500
18/18 [=====] - 0s 6ms/step - loss: 0.5364 - accuracy: 0.7691 - val_loss: 0.5370 - val_accuracy: 0.7344
Epoch 221/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5362 - accuracy: 0.7639 - val_loss: 0.5368 - val_accuracy: 0.7396
Epoch 222/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5358 - accuracy: 0.7639 - val_loss: 0.5365 - val_accuracy: 0.7396
Epoch 223/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5355 - accuracy: 0.7639 - val_loss: 0.5362 - val_accuracy: 0.7396
Epoch 224/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5352 - accuracy: 0.7639 - val_loss: 0.5360 - val_accuracy: 0.7448
Epoch 225/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5349 - accuracy: 0.7656 - val_loss: 0.5358 - val_accuracy: 0.7448
Epoch 226/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5346 - accuracy: 0.7674 - val_loss: 0.5355 - val_accuracy: 0.7448
Epoch 227/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5343 - accuracy: 0.7691 - val_loss: 0.5353 - val

```
_accuracy: 0.7500
Epoch 228/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5340 - accuracy: 0.7708 - val_loss: 0.5351 - val
_accuracy: 0.7500
Epoch 229/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5336 - accuracy: 0.7691 - val_loss: 0.5348 - val
_accuracy: 0.7500
Epoch 230/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5333 - accuracy: 0.7691 - val_loss: 0.5346 - val
_accuracy: 0.7500
Epoch 231/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5330 - accuracy: 0.7691 - val_loss: 0.5344 - val
_accuracy: 0.7500
Epoch 232/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5327 - accuracy: 0.7674 - val_loss: 0.5341 - val
_accuracy: 0.7500
Epoch 233/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5324 - accuracy: 0.7691 - val_loss: 0.5339 - val
_accuracy: 0.7500
Epoch 234/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5321 - accuracy: 0.7691 - val_loss: 0.5337 - val
_accuracy: 0.7500
Epoch 235/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5318 - accuracy: 0.7691 - val_loss: 0.5334 - val
_accuracy: 0.7552
Epoch 236/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5314 - accuracy: 0.7726 - val_loss: 0.5332 - val
_accuracy: 0.7552
Epoch 237/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5312 - accuracy: 0.7708 - val_loss: 0.5330 - val
_accuracy: 0.7552
Epoch 238/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5309 - accuracy: 0.7708 - val_loss: 0.5328 - val
_accuracy: 0.7552
Epoch 239/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5306 - accuracy: 0.7708 - val_loss: 0.5326 - val
_accuracy: 0.7552
Epoch 240/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5303 - accuracy: 0.7726 - val_loss: 0.5324 - val
_accuracy: 0.7552
Epoch 241/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5300 - accuracy: 0.7708 - val_loss: 0.5321 - val
_accuracy: 0.7552
Epoch 242/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5296 - accuracy: 0.7708 - val_loss: 0.5319 - val
_accuracy: 0.7552
Epoch 243/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5294 - accuracy: 0.7726 - val_loss: 0.5317 - val
_accuracy: 0.7552
Epoch 244/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5291 - accuracy: 0.7726 - val_loss: 0.5315 - val
_accuracy: 0.7552
Epoch 245/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5288 - accuracy: 0.7726 - val_loss: 0.5313 - val
_accuracy: 0.7552
Epoch 246/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5285 - accuracy: 0.7743 - val_loss: 0.5311 - val
_accuracy: 0.7552
Epoch 247/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5282 - accuracy: 0.7743 - val_loss: 0.5309 - val
_accuracy: 0.7552
Epoch 248/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5280 - accuracy: 0.7726 - val_loss: 0.5307 - val
_accuracy: 0.7552
Epoch 249/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5277 - accuracy: 0.7743 - val_loss: 0.5305 - val
_accuracy: 0.7552
Epoch 250/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5274 - accuracy: 0.7708 - val_loss: 0.5303 - val
_accuracy: 0.7552
Epoch 251/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5271 - accuracy: 0.7743 - val_loss: 0.5301 - val
_accuracy: 0.7552
Epoch 252/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5268 - accuracy: 0.7726 - val_loss: 0.5300 - val
_accuracy: 0.7552
Epoch 253/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5265 - accuracy: 0.7726 - val_loss: 0.5298 - val
_accuracy: 0.7552
Epoch 254/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5263 - accuracy: 0.7726 - val_loss: 0.5296 - val
_accuracy: 0.7552
Epoch 255/1500
```

[illegible]

Epoch 283/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5187 - accuracy: 0.7691 - val_loss: 0.5250 - val_accuracy: 0.7448
Epoch 284/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5184 - accuracy: 0.7691 - val_loss: 0.5248 - val_accuracy: 0.7448
Epoch 285/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5182 - accuracy: 0.7691 - val_loss: 0.5247 - val_accuracy: 0.7448
Epoch 286/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5179 - accuracy: 0.7674 - val_loss: 0.5245 - val_accuracy: 0.7448
Epoch 287/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5177 - accuracy: 0.7691 - val_loss: 0.5244 - val_accuracy: 0.7448
Epoch 288/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5174 - accuracy: 0.7674 - val_loss: 0.5243 - val_accuracy: 0.7448
Epoch 289/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5172 - accuracy: 0.7691 - val_loss: 0.5241 - val_accuracy: 0.7448
Epoch 290/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5169 - accuracy: 0.7674 - val_loss: 0.5240 - val_accuracy: 0.7448
Epoch 291/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5166 - accuracy: 0.7691 - val_loss: 0.5239 - val_accuracy: 0.7448
Epoch 292/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5165 - accuracy: 0.7691 - val_loss: 0.5237 - val_accuracy: 0.7500
Epoch 293/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5162 - accuracy: 0.7691 - val_loss: 0.5236 - val_accuracy: 0.7500
Epoch 294/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5160 - accuracy: 0.7674 - val_loss: 0.5235 - val_accuracy: 0.7500
Epoch 295/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5158 - accuracy: 0.7691 - val_loss: 0.5234 - val_accuracy: 0.7500
Epoch 296/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5156 - accuracy: 0.7691 - val_loss: 0.5232 - val_accuracy: 0.7500
Epoch 297/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5153 - accuracy: 0.7691 - val_loss: 0.5231 - val_accuracy: 0.7500
Epoch 298/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5150 - accuracy: 0.7708 - val_loss: 0.5230 - val_accuracy: 0.7500
Epoch 299/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5148 - accuracy: 0.7708 - val_loss: 0.5229 - val_accuracy: 0.7500
Epoch 300/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5146 - accuracy: 0.7708 - val_loss: 0.5228 - val_accuracy: 0.7500
Epoch 301/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5143 - accuracy: 0.7726 - val_loss: 0.5227 - val_accuracy: 0.7500
Epoch 302/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5142 - accuracy: 0.7708 - val_loss: 0.5225 - val_accuracy: 0.7500
Epoch 303/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5139 - accuracy: 0.7691 - val_loss: 0.5224 - val_accuracy: 0.7500
Epoch 304/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5136 - accuracy: 0.7708 - val_loss: 0.5223 - val_accuracy: 0.7500
Epoch 305/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5135 - accuracy: 0.7760 - val_loss: 0.5222 - val_accuracy: 0.7500
Epoch 306/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5132 - accuracy: 0.7760 - val_loss: 0.5221 - val_accuracy: 0.7500
Epoch 307/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5130 - accuracy: 0.7726 - val_loss: 0.5220 - val_accuracy: 0.7500
Epoch 308/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5128 - accuracy: 0.7760 - val_loss: 0.5219 - val_accuracy: 0.7500
Epoch 309/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5126 - accuracy: 0.7760 - val_loss: 0.5218 - val_accuracy: 0.7448
Epoch 310/1500
18/18 [=====] - 0s 4ms/step - loss: 0.5123 - accuracy: 0.7743 - val_loss: 0.5217 - val

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

18/18 [=====] - 0s 5ms/step- loss: 0.4474 - accuracy: 0.7951 - val_loss: 0.5106 - val_accuracy: 0.7344

In [74]: `run_hist_2.history.keys()`

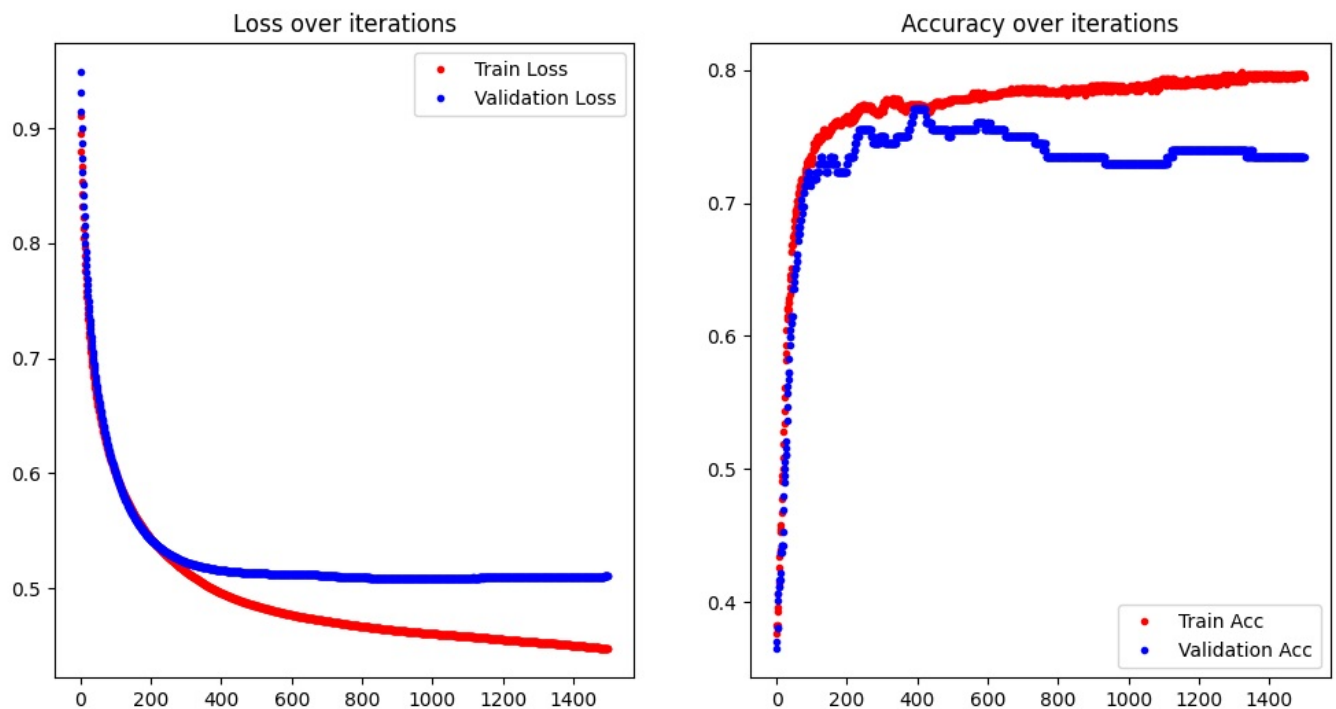
Out[74]: `dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])`

```
In [107]: n = len(run_hist_2.history["loss"])

fig = plt.figure(figsize=(12, 6))
ax = fig.add_subplot(1, 2, 1)
ax.plot(range(n), (run_hist_2.history["loss"]), 'r.', label="Train Loss")
ax.plot(range(n), (run_hist_2.history["val_loss"]), 'b.', label="Validation Loss")
ax.legend()
ax.set_title('Loss over iterations')

ax = fig.add_subplot(1, 2, 2)
ax.plot(range(n), (run_hist_2.history["accuracy"]), 'r.', label="Train Acc")
ax.plot(range(n), (run_hist_2.history["val_accuracy"]), 'b.', label="Validation Acc")
ax.legend(loc='lower right')
ax.set_title('Accuracy over iterations')
```

Out[107]: `Text(0.5, 1.0, 'Accuracy over iterations')`



```
In [108]: # Make predictions on test data
y_pred_prob_nn_2 = model_2.predict(X_test_norm)

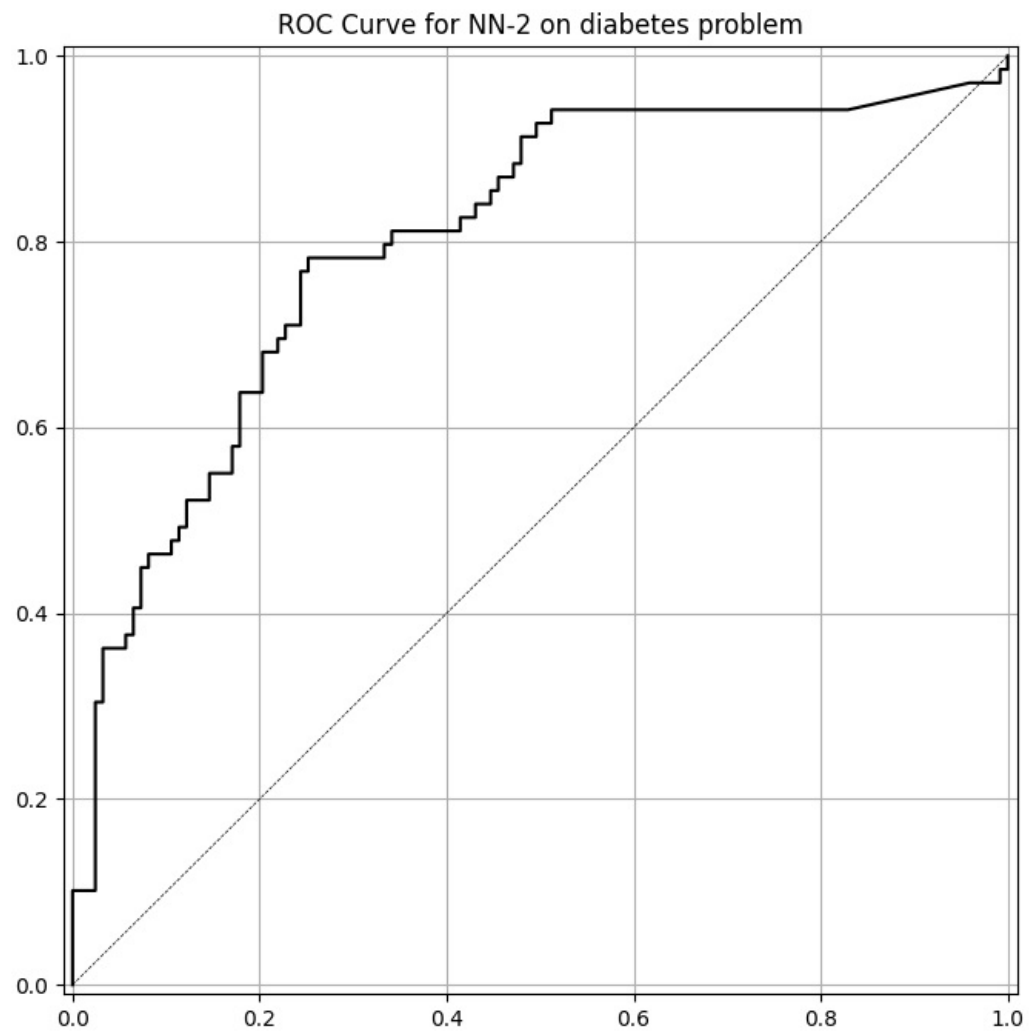
# Obtain predicted class labels by taking the index of the highest probability for each example
y_pred_class_nn_2 = y_pred_prob_nn_2.argmax(axis=-1)

print('')
print('accuracy is {:.3f}'.format(accuracy_score(y_test,y_pred_class_nn_2)))
print('roc-auc is {:.3f}'.format(roc_auc_score(y_test,y_pred_prob_nn_2)))

plot_roc(y_test, y_pred_prob_nn_2, 'NN-2')
```

6/6 [=====] - 0s 2ms/step

accuracy is 0.641
roc-auc is 0.801



Authors

[Moazzam Ali](#) is a Associate ML Engineer and CEO of [MT Learners](#).