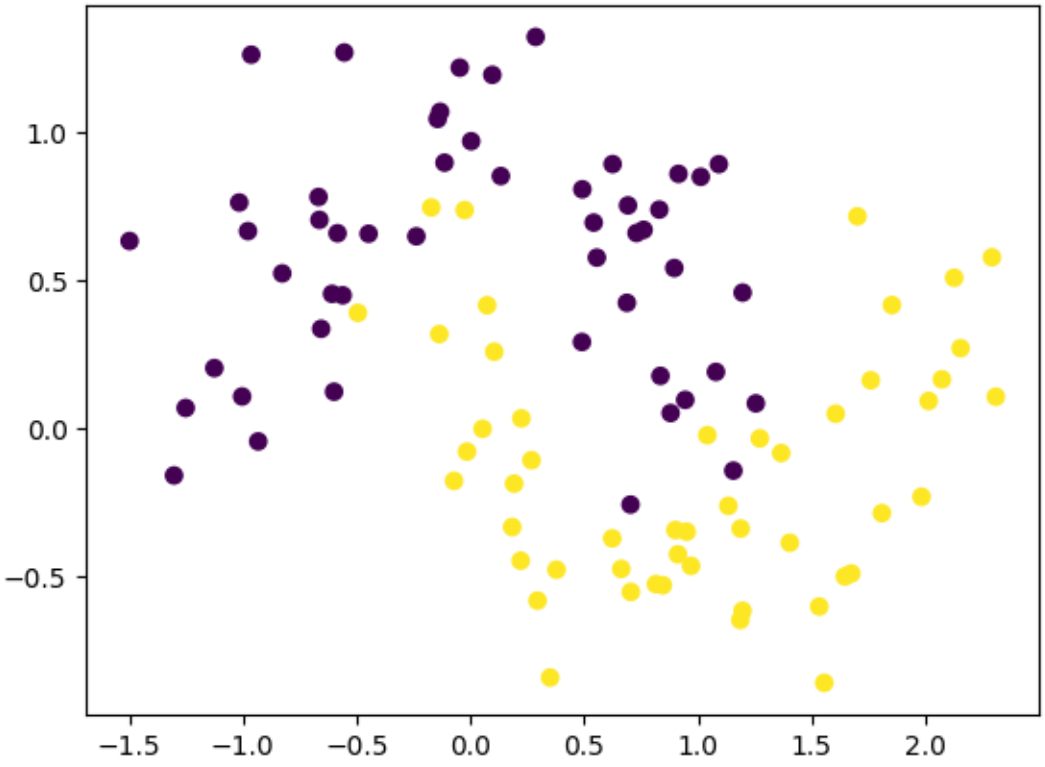
12-dropout-layers-in-nn

May 15, 2023

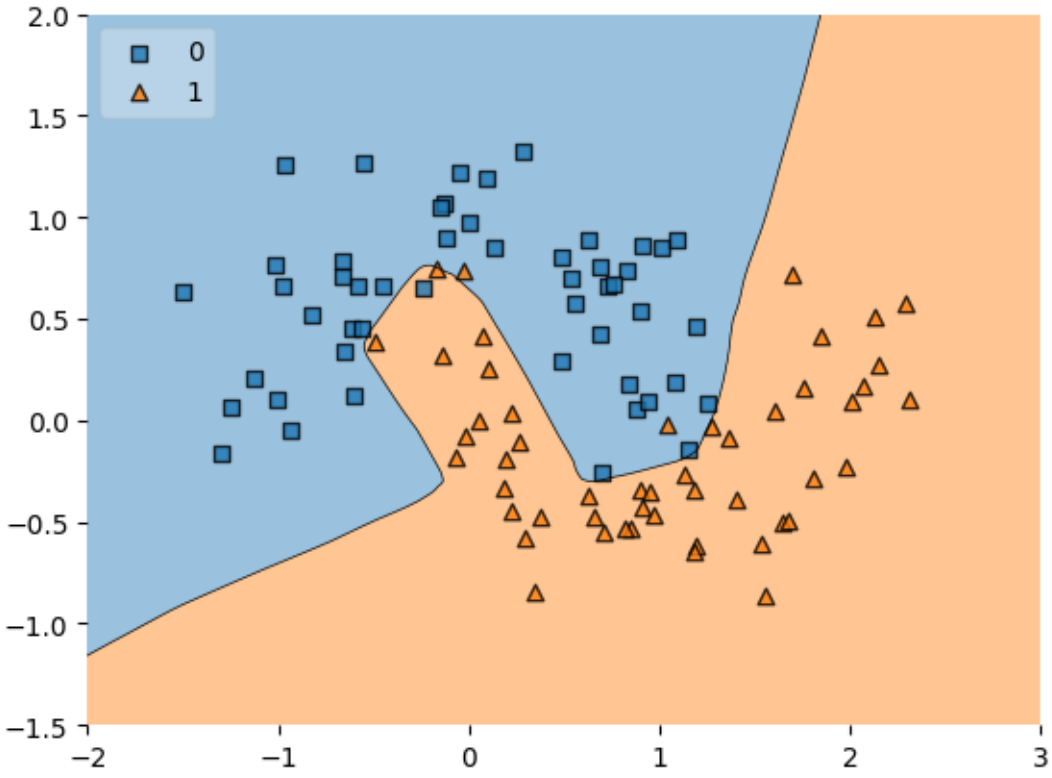
[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_moons
import seaborn as sns
from mlxtend.plotting import plot_decision_regions

import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dropout
from tensorflow.keras.optimizers import Adam

X, y = make_moons(100, noise=0.25, random_state=2)

# Visualize the data
plt.scatter(X[:,0], X[:,1], c=y)
plt.show()

# Build the model with dropout layers
model = Sequential()

model.add(Dense(128, input_dim=2, activation="relu"))
model.add(Dropout(0.5))
model.add(Dense(128, activation="relu"))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

model.summary()

adam = Adam(learning_rate=0.01)
model.compile(loss='binary_crossentropy', optimizer=adam, metrics=['accuracy'])

history = model.fit(X, y, epochs=2000, validation_split=0.2, verbose=0)

# Visualize the decision boundary
plot_decision_regions(X, y.astype('int'), clf=model, legend=2)
plt.xlim(-2,3)
```

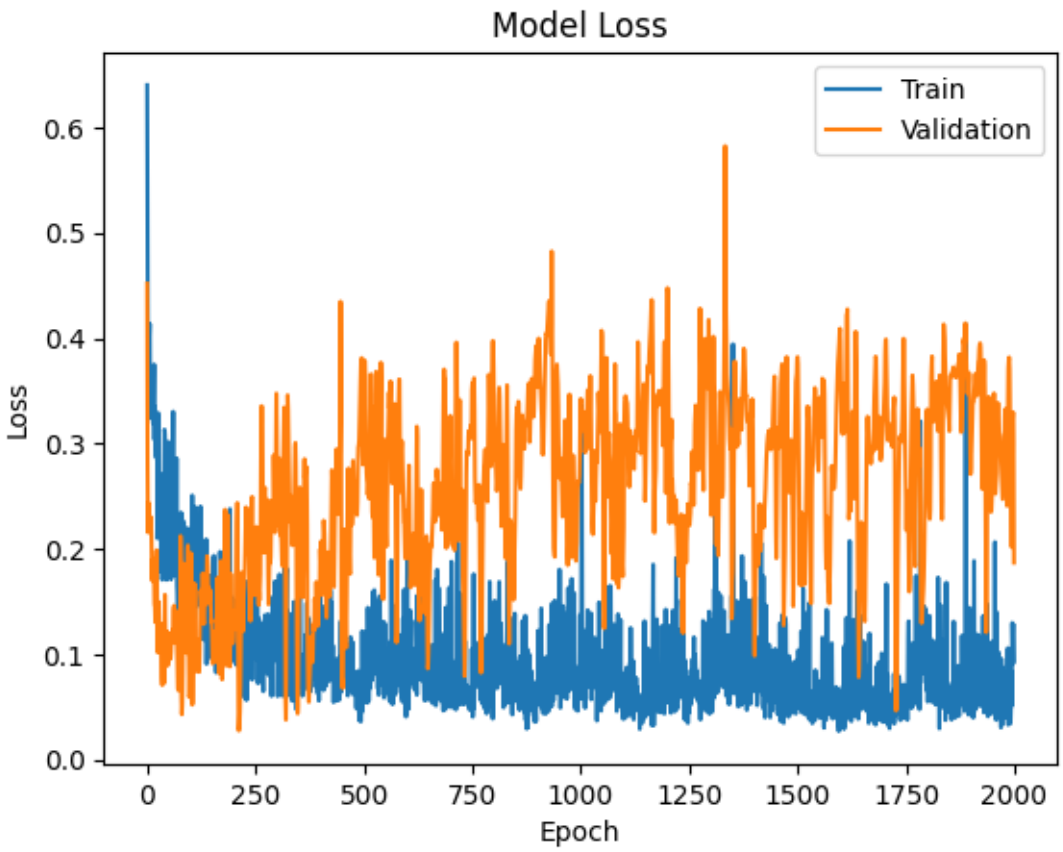
```
plt.ylim(-1.5,2)
plt.show()

# Plot the loss curve
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```




```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               384

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 128)               16512
```

```
 dropout_1 (Dropout)          (None, 128)               0

 dense_2 (Dense)              (None, 1)                 129

=================================================================
Total params: 17,025
Trainable params: 17,025
Non-trainable params: 0

_________________________________________________________________
9600/9600 [==============================] - 22s 2ms/step
```





[2]:
```
# Calculation of accuarcy of each model

# Calculate the accuracy for model1
acc_model1 = history.history['accuracy'][-1] * 100
acc_model1
```

[2]: 93.75

[ ]:

[ ]: