In this video we will compare our model performance on different datasets.
We will see how changing the hidden units affects our result.

## Other Datasets

```
from sklearn.datasets import make_moons,make_circles,make_classification
```

```
def load_dataset(dataset):
    if dataset=='moons':
        X,Y = make_moons(n_samples=500,noise=0.2,random_state=1) #Perceptron
    elif dataset=='circles':
        X,Y = make_circles(n_samples=500, shuffle=True, noise=0.2, random_state=1, factor=0.2)
    elif dataset=='classification':
        X,Y = make_classification(n_samples=500,n_classes=2,n_features=2,n_informative=2,n_redundant=0,random_state=1)
    else:
        #Create XOR Dataset
        X = np.array([[0,0],
                      [0,1],
                      [1,0],
                      [1,1]])
        Y = np.array([0,1,1,0])

    return X,Y
```

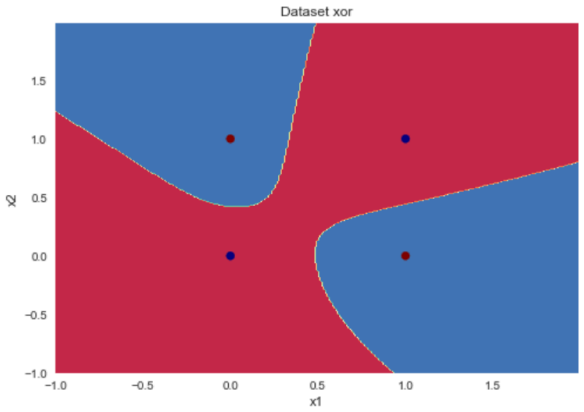
we used it in perceptron learning algorithm.

We are using 4 datasets here.

```
datasets = ["xor","classification","moons","circles"]

for d in datasets:
    model = NeuralNetwork(input_size=2,layers=[4,3],output_size=2)
    X,Y  = load_dataset(d)
    train(X,Y,model,1000,0.001,logs=False)
    outputs = model.predict(X)

    training_accuracy = np.sum(outputs==Y)/Y.shape[0]
    print("Training Acc %.4f"%training_accuracy)

    plt.title("Dataset "+d)
    plot_decision_boundary(lambda x:model.predict(x),X,Y)
    plt.show()
```

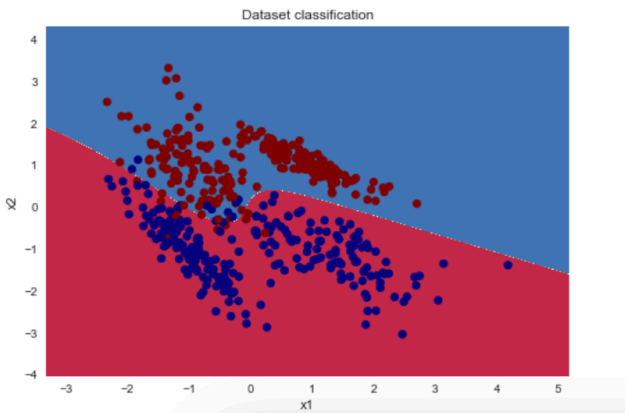
One model for every dataset
Now lets take 4 and 3 units in hidden layer $H_1$ and $H_2$.

```
Training Acc 1.0000
```



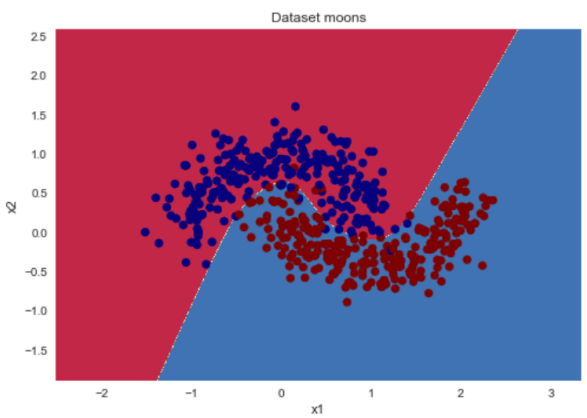

for XOR Dataset Training Accuracy is 100%

```
Training Acc 0.9600
```



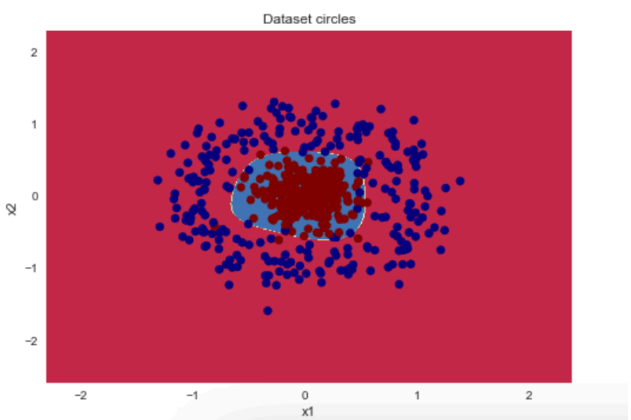

For classification dataset training accuracy is 96%.

```
Training Acc 0.9740
```




For moons dataset, training accuracy is 97%.
(which was about 60% in case of linear classifier perceptron)

```
Training Acc 0.9640
```




In circles dataset accuracy is 96%.

You can change the no. of units (neurons) in hidden layer and see what impact it has on the training accuracy.

Lets make layers = [8, 4]

on changing units in hidden layer

for XOR Dataset variation is more because there are very less training examples from which model can learn. It doesnot know what an appropriate boundary should be but it is able to give 100% accuracy by minimizing the loss.