Linear models leading to subsequent neural network model

INF 552: Machine Learning
Zongdi Xu
March 5, 2019

Unproper simulation with perceptron

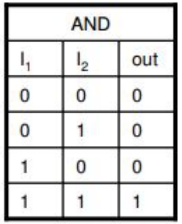
XOR: Non-linear case

Your dataset looks like this:

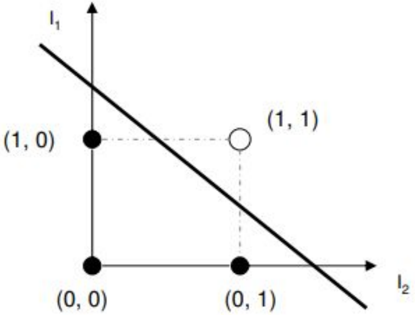
1 0
0 1

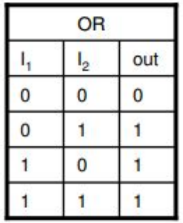
It can't be done.

Not 100% accuracy

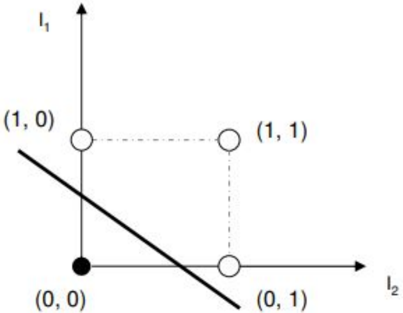


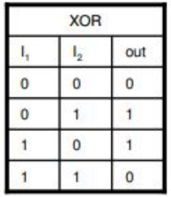
AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1



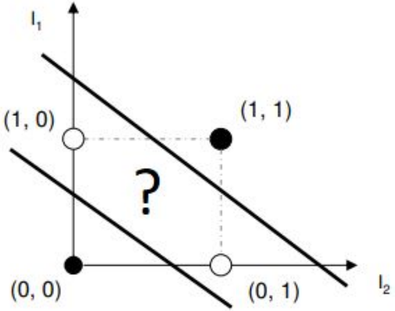


OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1





XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0



Unsuccessful classifier

Retrospect on perceptron

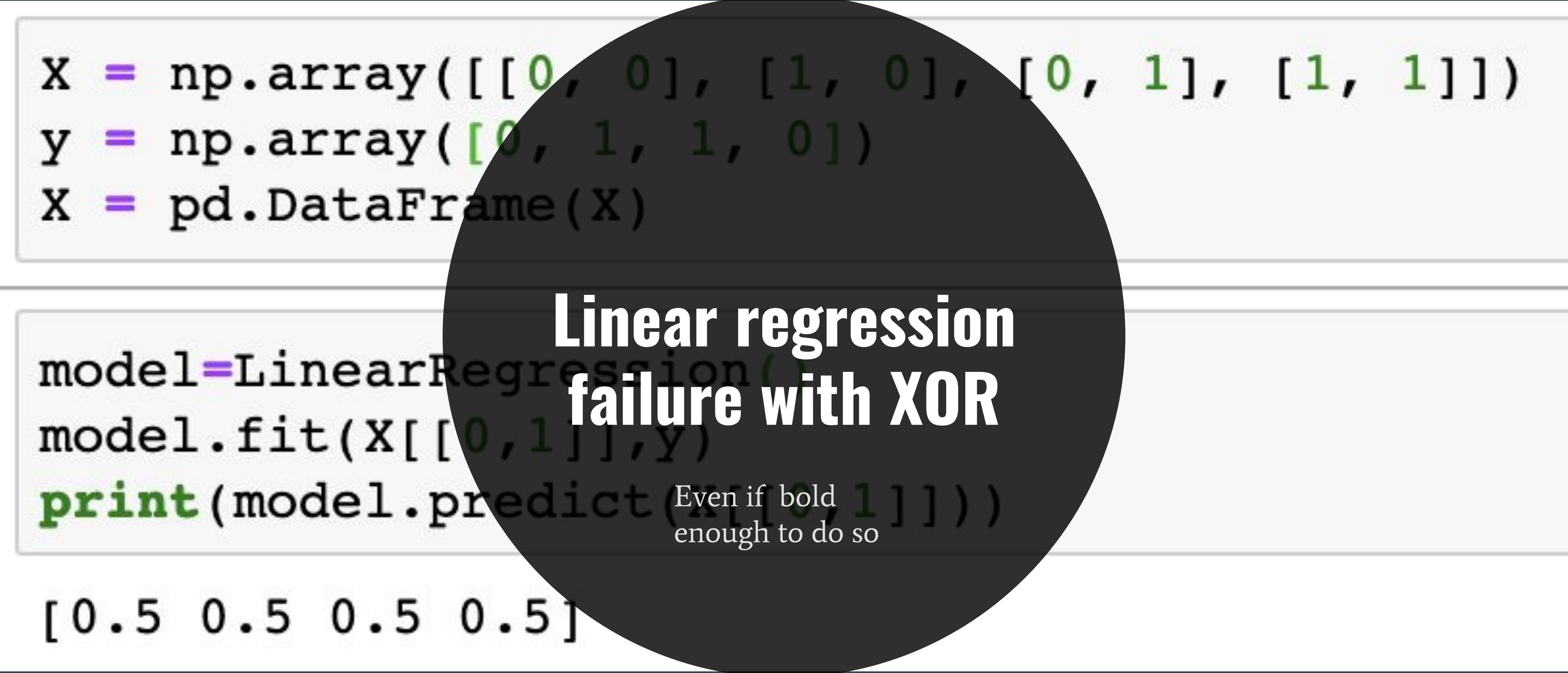
A single perceptron can’t represent the boolean XOR function.

“[XOR] is not linearly separable so the perceptron cannot learn it” -- Artificial Intelligence: A Modern Approach(p.730).

“Single layer perceptrons are only capable of learning linearly separable patterns” -- Wikipedia.

Linear regression failure with XOR

Even if bold enough to do so



```
X = np.array([[0, 0], [1, 0], [0, 1], [1, 1]])
y = np.array([0, 1, 1, 0])
X = pd.DataFrame(X)
```

```
model=LinearRegression()
model.fit(X[[0,1]],y)
print(model.predict(X[[0,1]]))
```

```
[0.5 0.5 0.5 0.5]
```

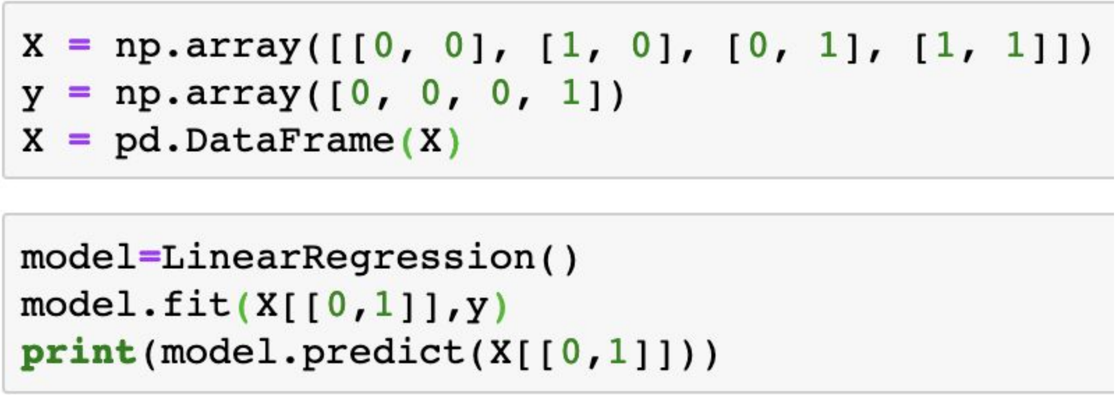
Still worthwhile to take a loot at easy cases

Notice the output of linear regression with OR & AND function.

Apply another threshold function and get the correct result.

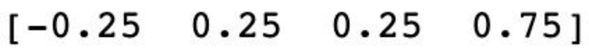
Example of composite function

y=sgn(x1+x2-threshold)



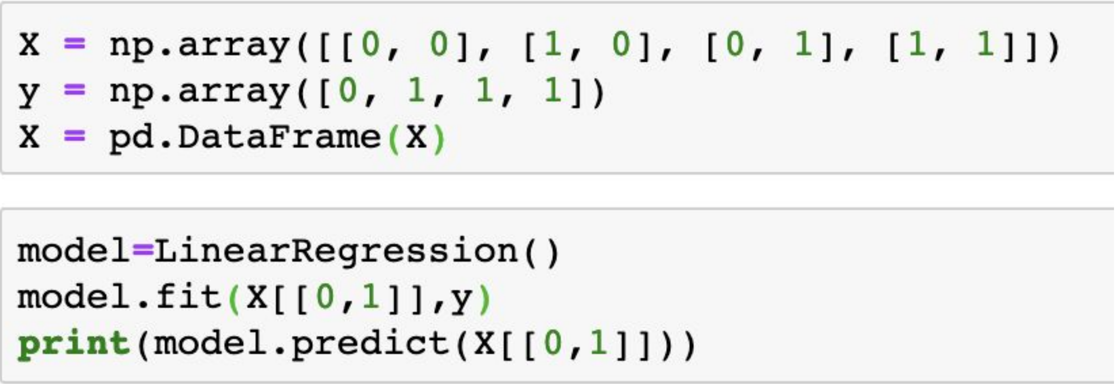
```
X = np.array([[0, 0], [1, 0], [0, 1], [1, 1]])
y = np.array([0, 0, 0, 1])
X = pd.DataFrame(X)
```

```
model=LinearRegression()
model.fit(X[[0,1]],y)
print(model.predict(X[[0,1]]))
```



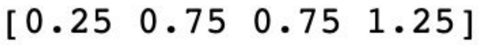
```
[-0.25   0.25   0.25   0.75]
```

“AND”: output 1 if predict > threshold, otherwise 0



```
X = np.array([[0, 0], [1, 0], [0, 1], [1, 1]])
y = np.array([0, 1, 1, 1])
X = pd.DataFrame(X)
```

```
model=LinearRegression()
model.fit(X[[0,1]],y)
print(model.predict(X[[0,1]]))
```



```
[0.25 0.75 0.75 1.25]
```

“OR”: output 1 if predict > threshold, otherwise 0

It is also possible ...

However, we can observe that we can write the XOR function as the following logical expression:

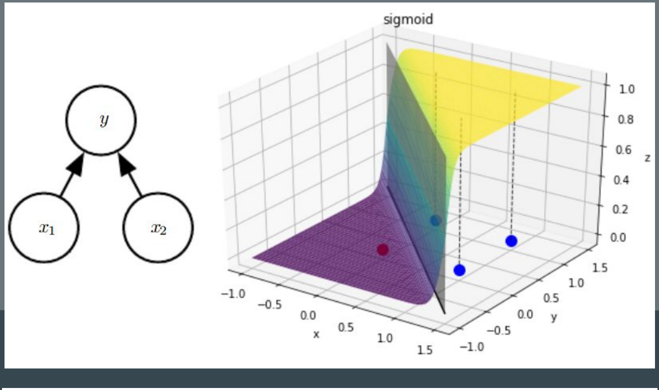
y = (x1 and not (x2)) or (not (x1) and x2).

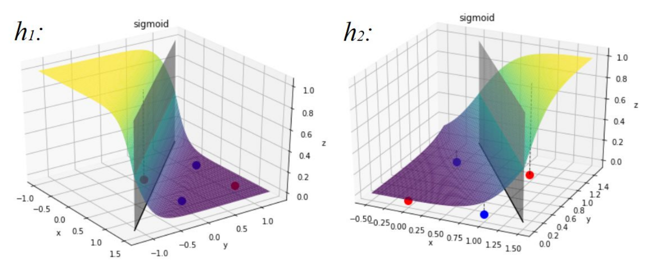
So when typical perceptron model fails, we apply one to another and constitute multilevel models that could make it.

We have been dealing with hyperplanes

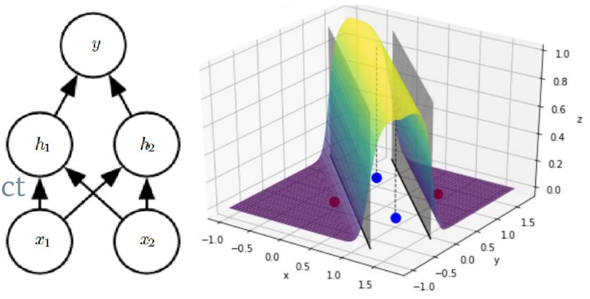
Multiple hyperplanes can combine, not separate lines.

Simple units give something more.





interconnect

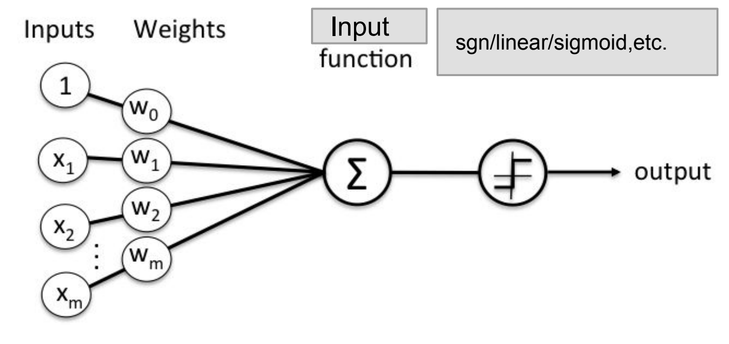


Coincidence

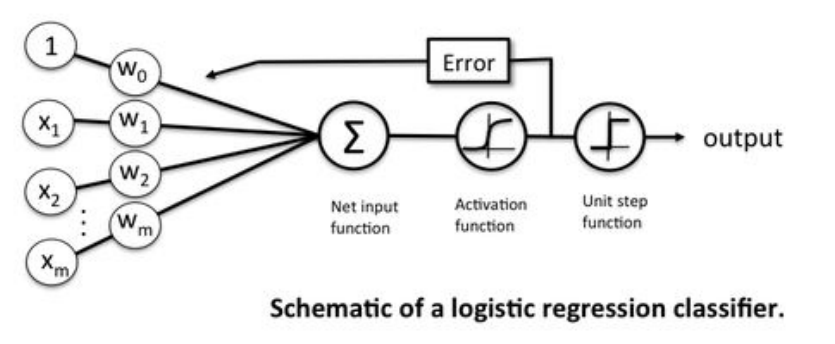
Perceptron/linear/(non-polynomial) logistic regression are just similar to a single neuron model in neural network.

The activation function takes the form of sgn, linear function, sigmoid, etc.

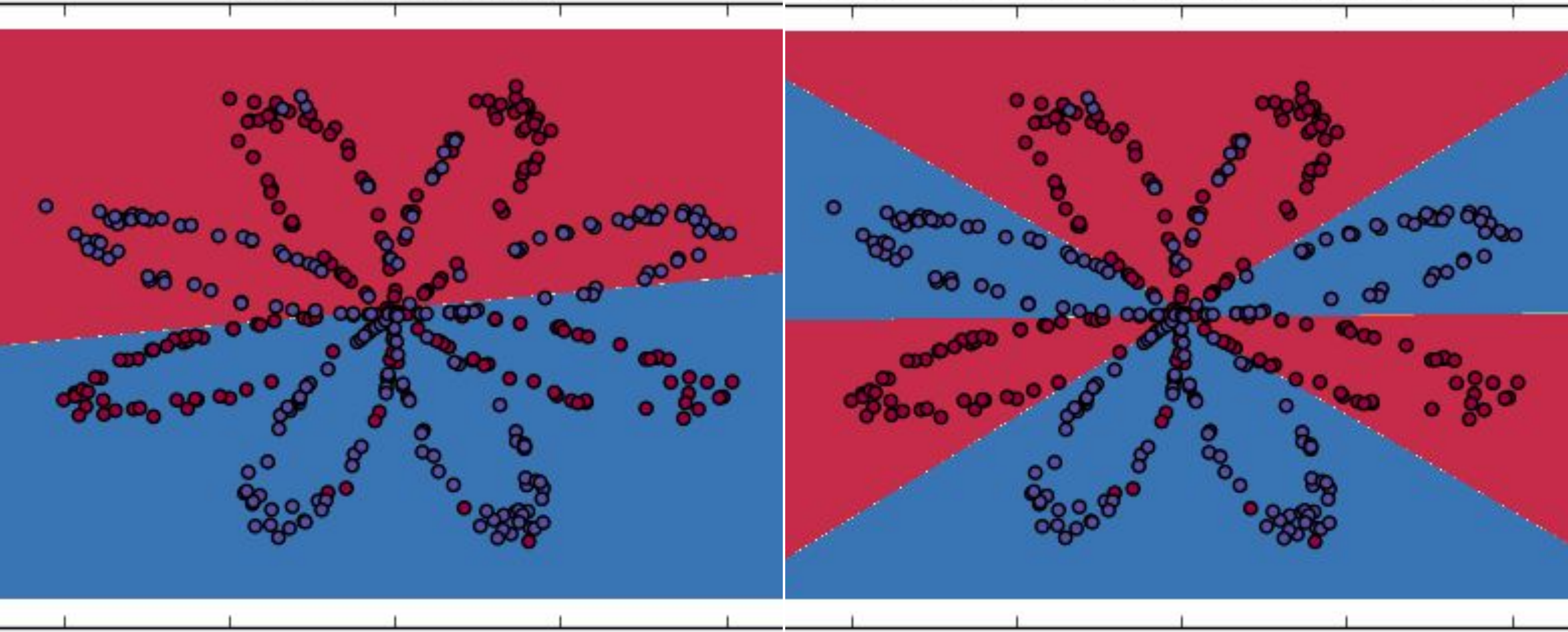
Multiple neuron interconnect.



Perceptron/linear



Neuron



Even more complicated cases

Thank you!

Reference:

Solving XOR with a single Perceptron

https://medium.com/@lucaspereira0612/solving-xor-with-a-single-perceptron-34539f395182

Logistic Regression & Neural Networks

https://courses.cs.washington.edu/courses/cse446/16sp/logistic_regression_3.pdf