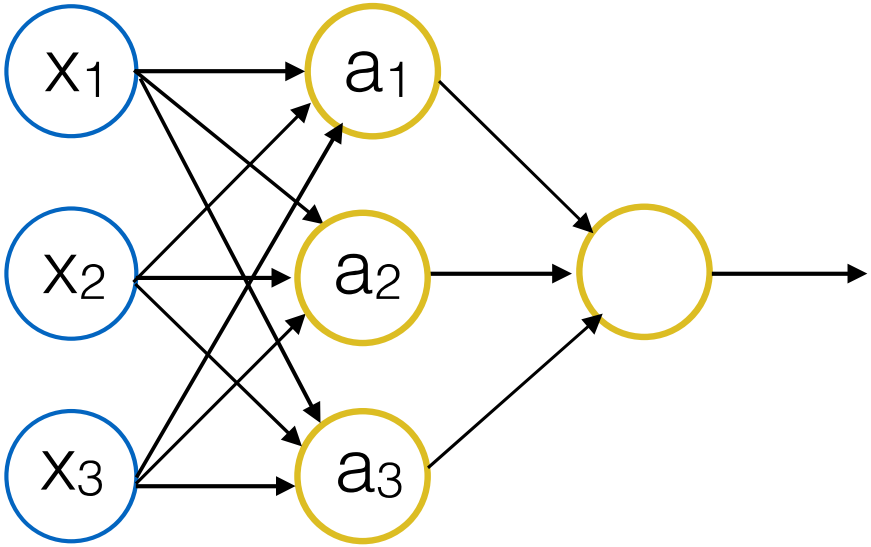
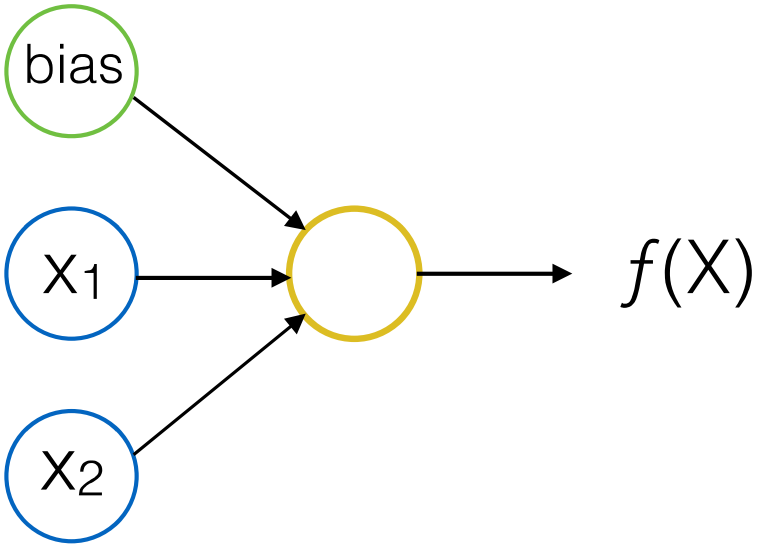
Neural Network

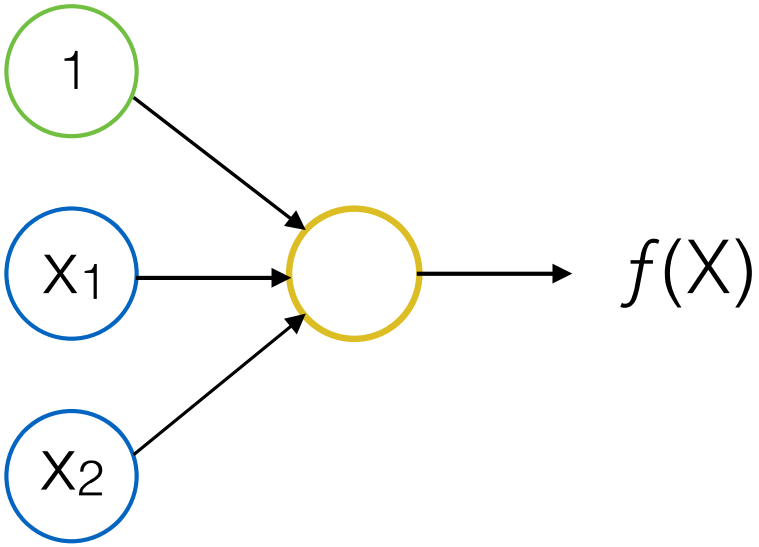
Neural Network



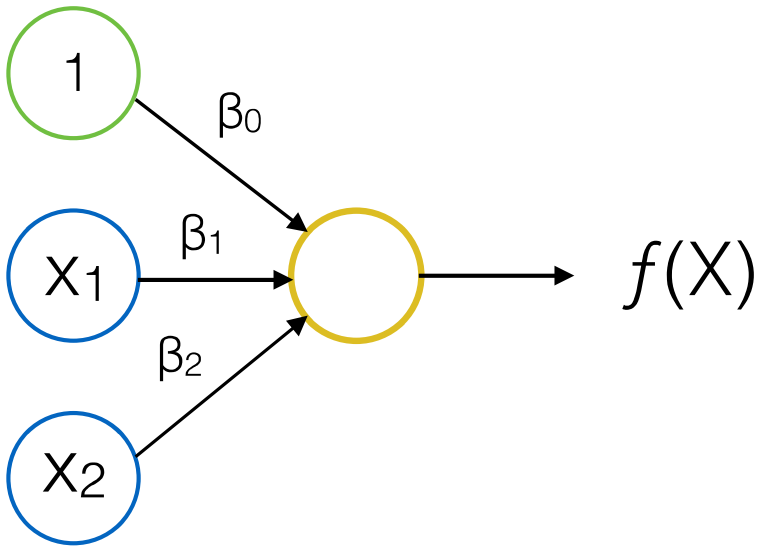
Perceptron



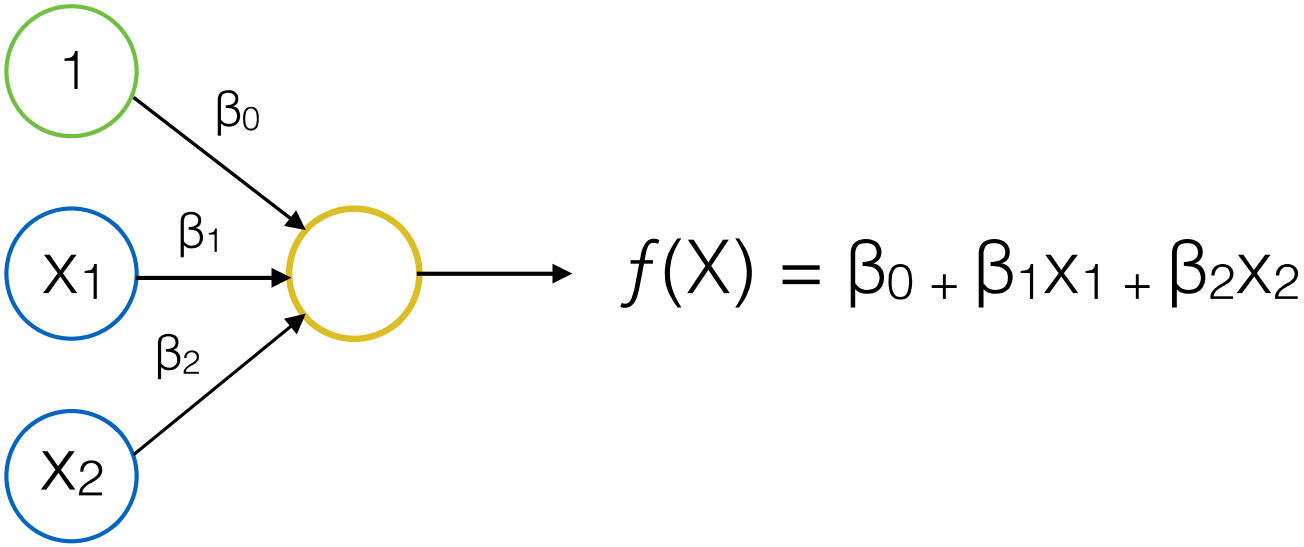
Perceptron



Perceptron

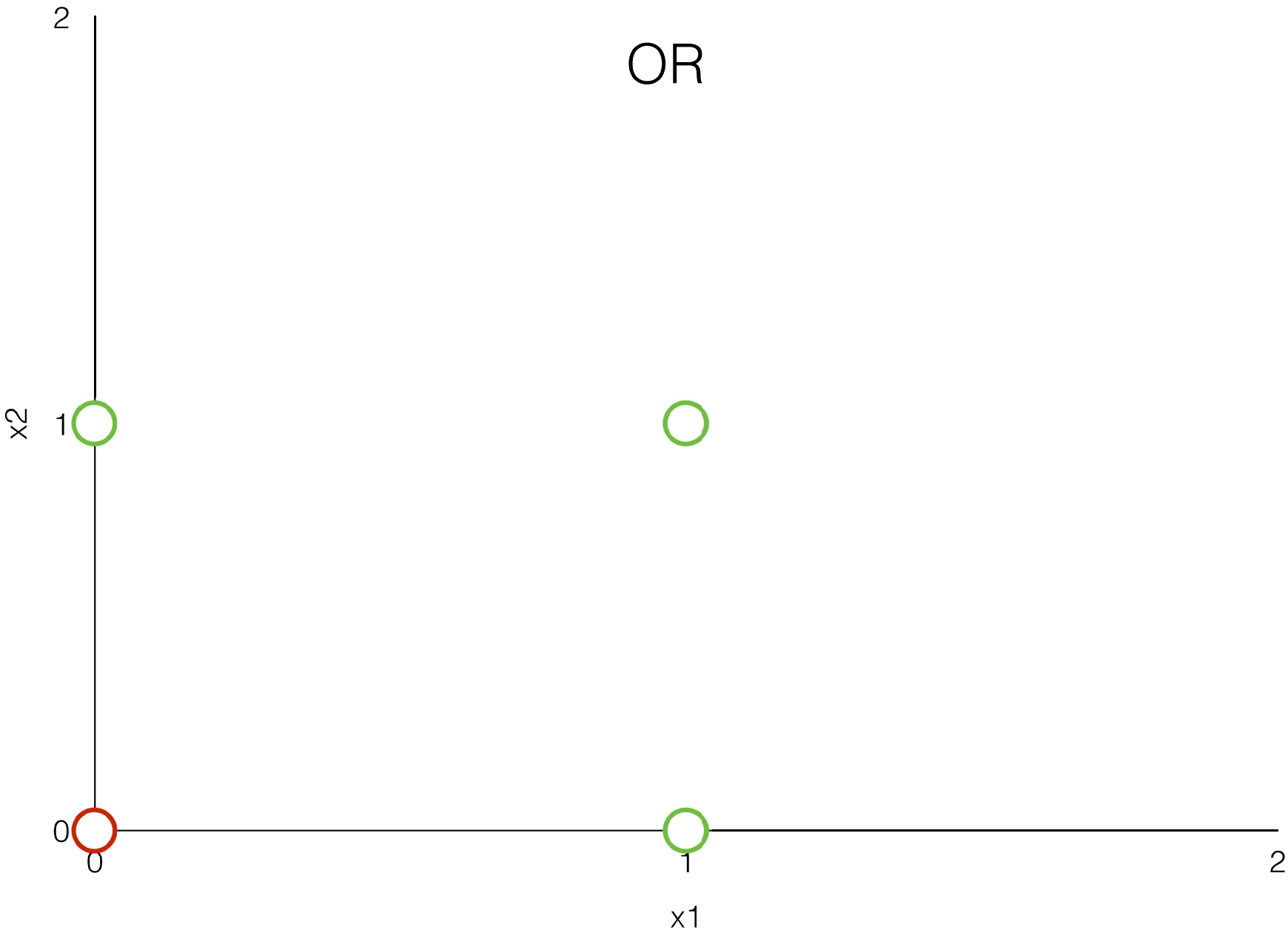


Perceptron

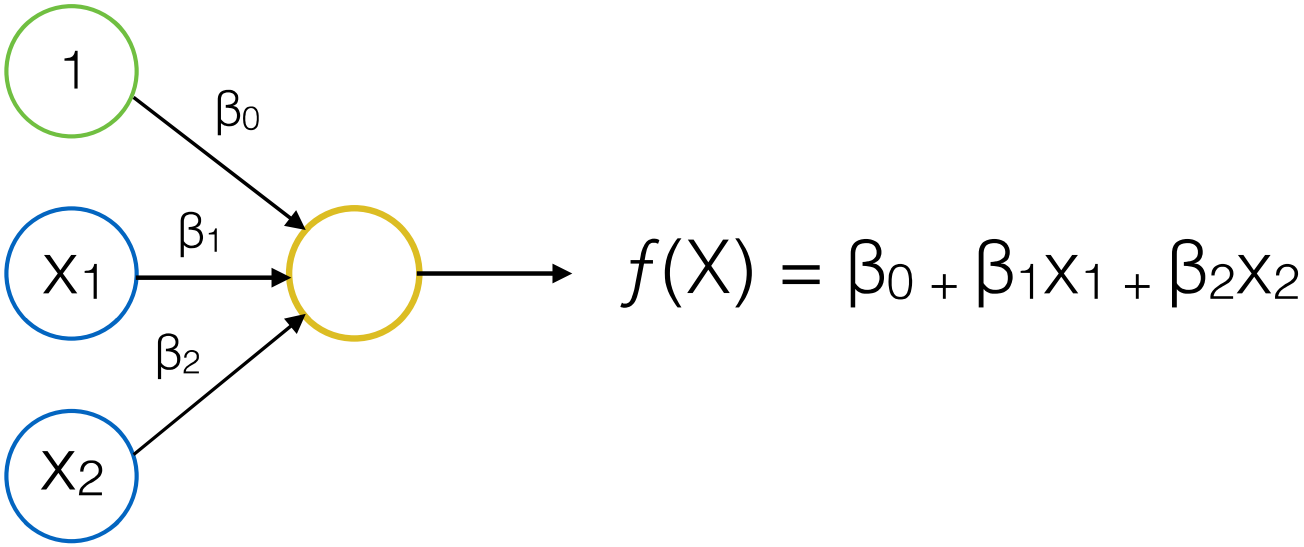


OR

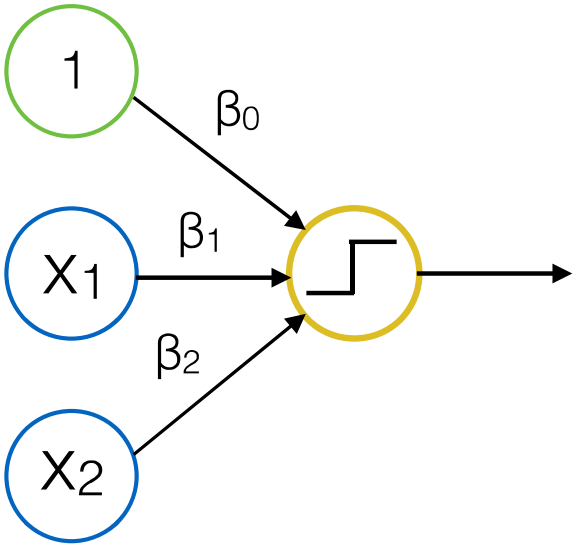
Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1


OR
2
x2
1
0
0
1
2
x1

Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1



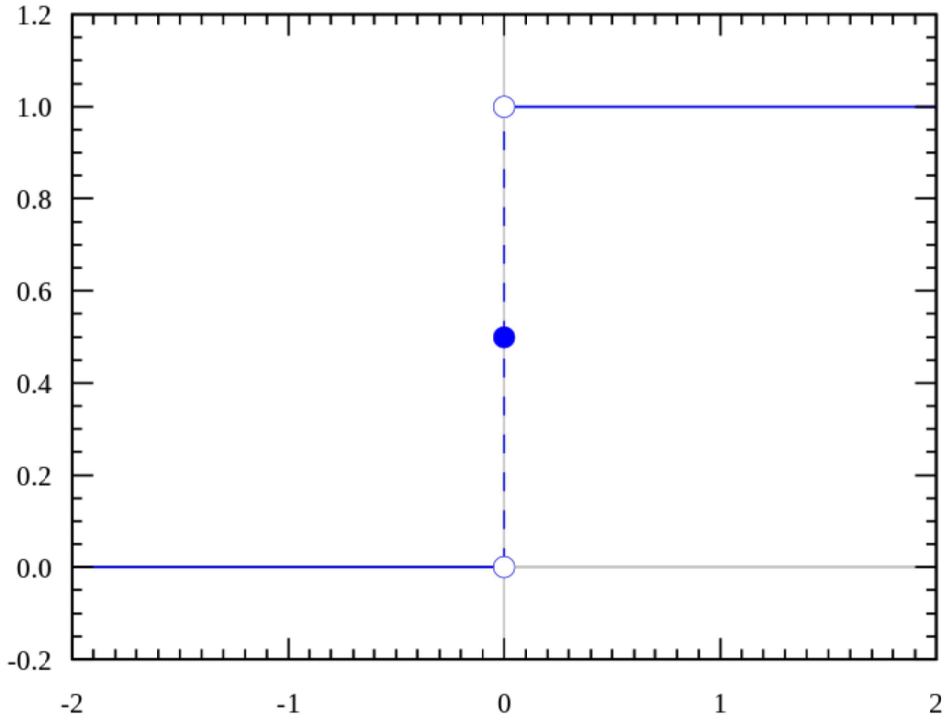
Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1



$$f(X) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

Activation Function: Threshold

step function

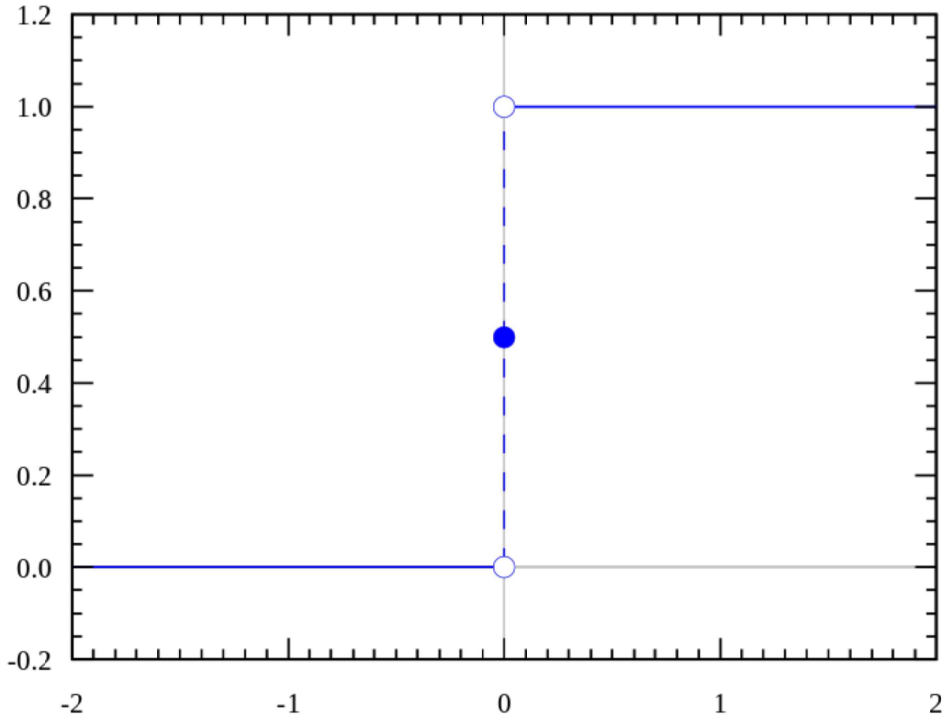


Activation Function: Threshold

if $\beta_0 + \beta_1 x_1 + \beta_2 x_2 > 0$: 1

Else: 0

step function



Activation Function: Threshold

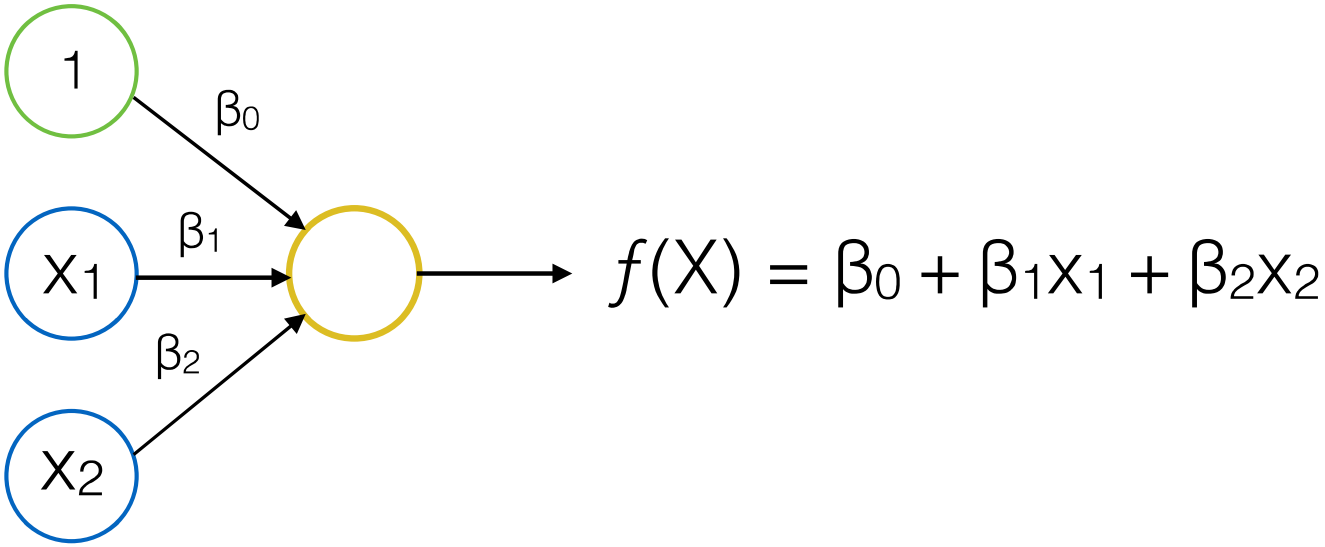
$$\text{if } \beta_0 + \beta_1 x_1 + \beta_2 x_2 > 0: \quad 1$$

$$\text{Else:} \quad 0$$

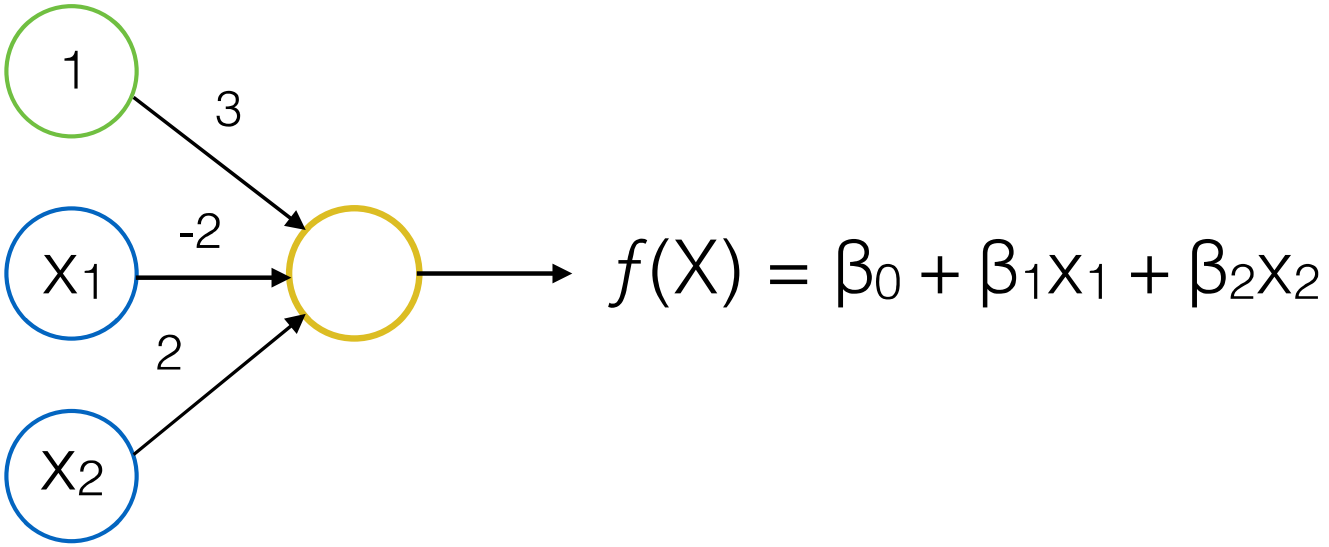
Update Rule:

$$\text{updated weight}_i = \text{weight}_i - (\text{output} - \text{target}) * \text{input}_i$$

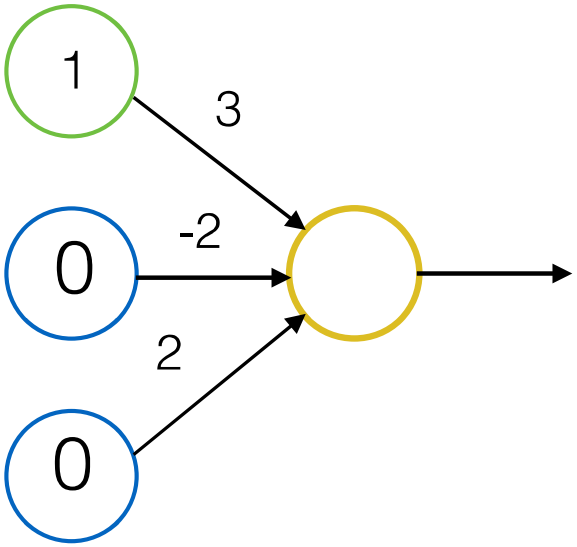
Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1



Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1

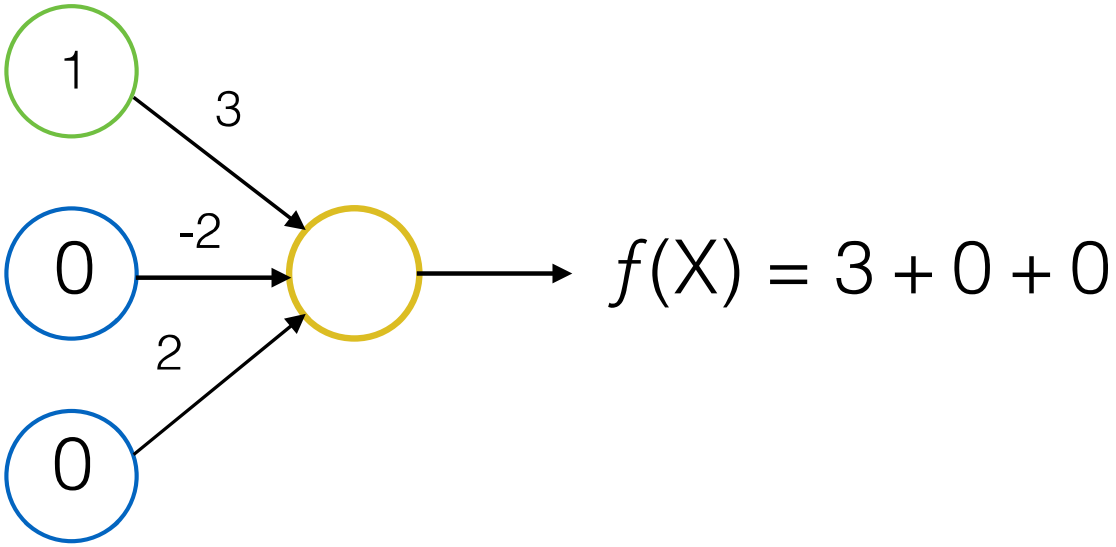


Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1

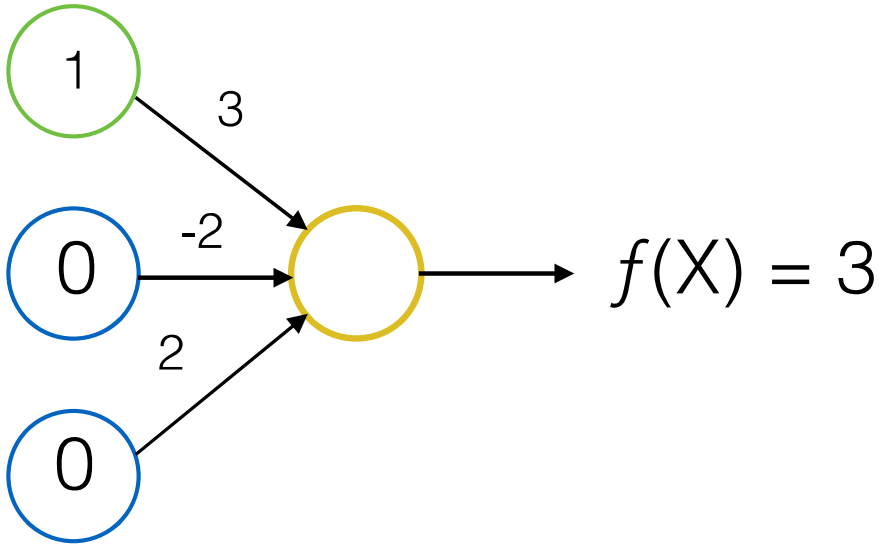


$$f(X) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

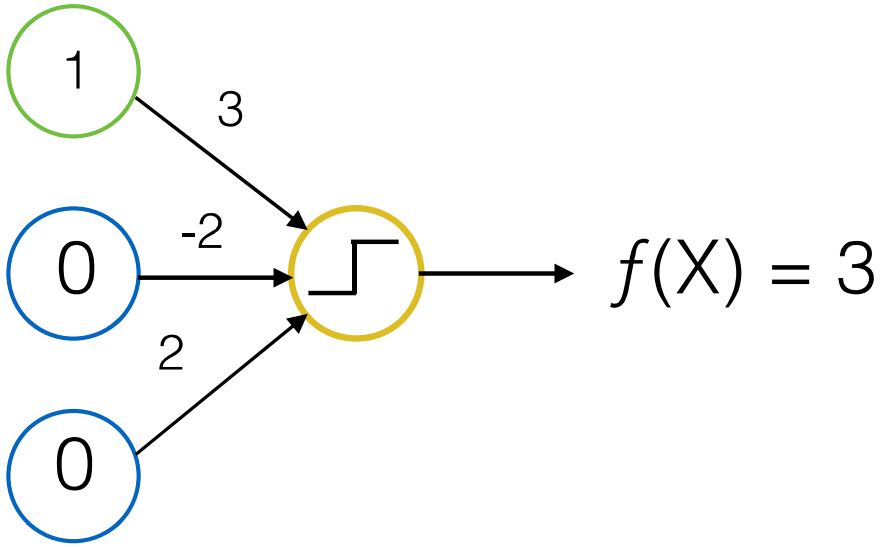
Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1



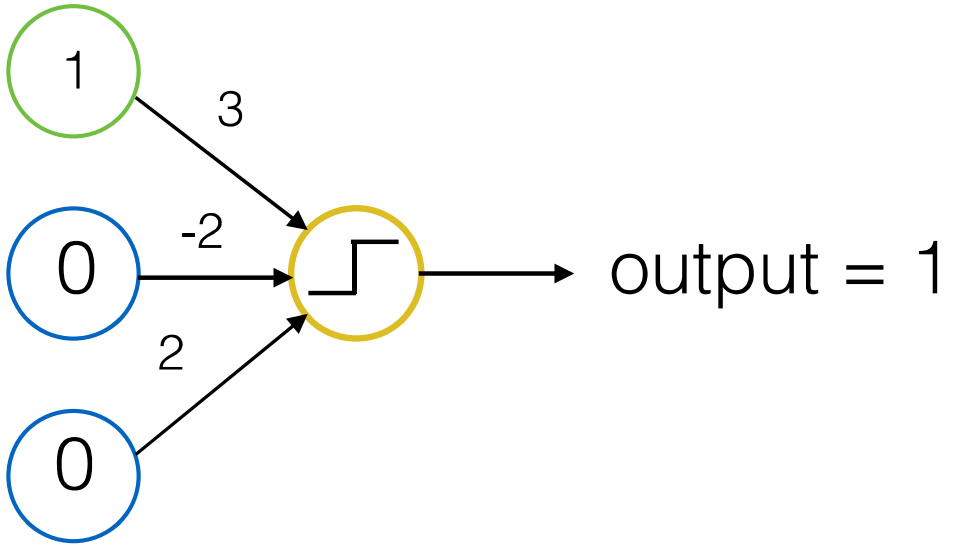
Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1



Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1



Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1



weights: 3, -2, 2

input: 1, 0, 0

output: 1

target: 0

updated $weight_0$ = $weight_0$ - (output - target) * $input_0$

updated $weight_1$ = $weight_1$ - (output - target) * $input_1$

updated $weight_2$ = $weight_2$ - (output - target) * $input_2$

weights: 3, -2, 2

input: 1, 0, 0

output: 1

target: 0

$$\text{updated weight}_0 = 3 - (\text{output} - \text{target}) * \text{input}_0$$

$$\text{updated weight}_1 = -2 - (\text{output} - \text{target}) * \text{input}_1$$

$$\text{updated weight}_2 = 2 - (\text{output} - \text{target}) * \text{input}_2$$

weights: 3, -2, 2

output: 1

input: 1, 0, 0

target: 0

$$\text{updated weight}_0 = 3 - (1 - \text{target}) * \text{input}_0$$

$$\text{updated weight}_1 = -2 - (1 - \text{target}) * \text{input}_1$$

$$\text{updated weight}_2 = 2 - (1 - \text{target}) * \text{input}_2$$

weights: 3, -2, 2

output: 1

input: 1, 0, 0

target: 0

$$\text{updated weight}_0 = 3 - (1 - 0) * \text{input}_0$$

$$\text{updated weight}_1 = -2 - (1 - 0) * \text{input}_1$$

$$\text{updated weight}_2 = 2 - (1 - 0) * \text{input}_2$$

weights: 3, -2, 2

output: 1

input: 1, 0, 0

target: 0

$$\text{updated weight}_0 = 3 - (1 - 0) * 1$$

$$\text{updated weight}_1 = -2 - (1 - 0) * 0$$

$$\text{updated weight}_2 = 2 - (1 - 0) * 0$$

weights: 3, -2, 2

input: 1, 0, 0

output: 1

target: 0

updated $\text{weight}_0 = 3 - 1$

updated $\text{weight}_1 = -2 - 0$

updated $\text{weight}_2 = 2 - 0$

weights: 3, -2, 2

input: 1, 0, 0

output: 1

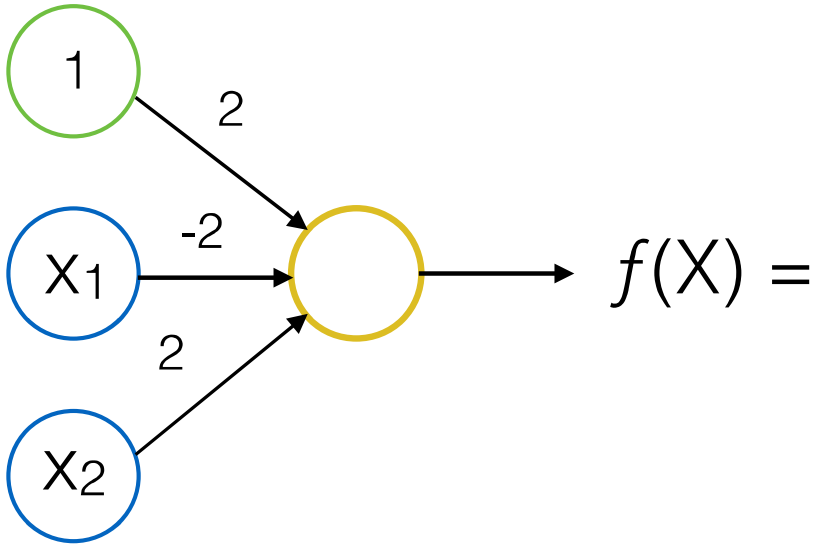
target: 0

updated $\text{weight}_0 = 2$

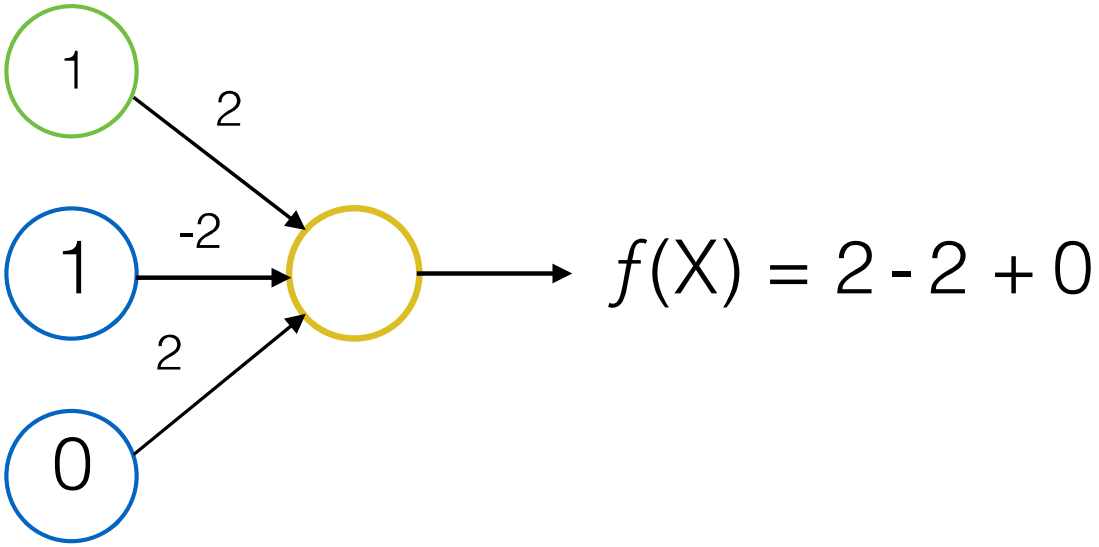
updated $\text{weight}_1 = -2$

updated $\text{weight}_2 = 2$

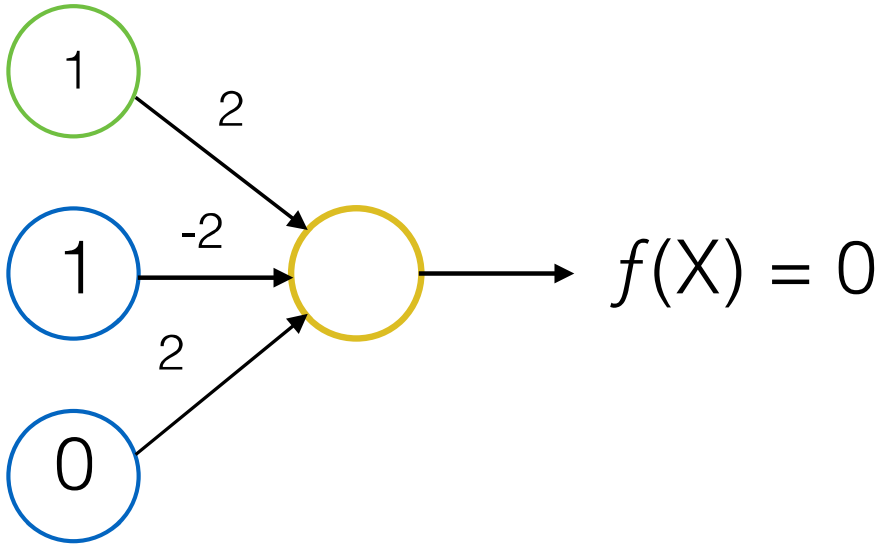
Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1



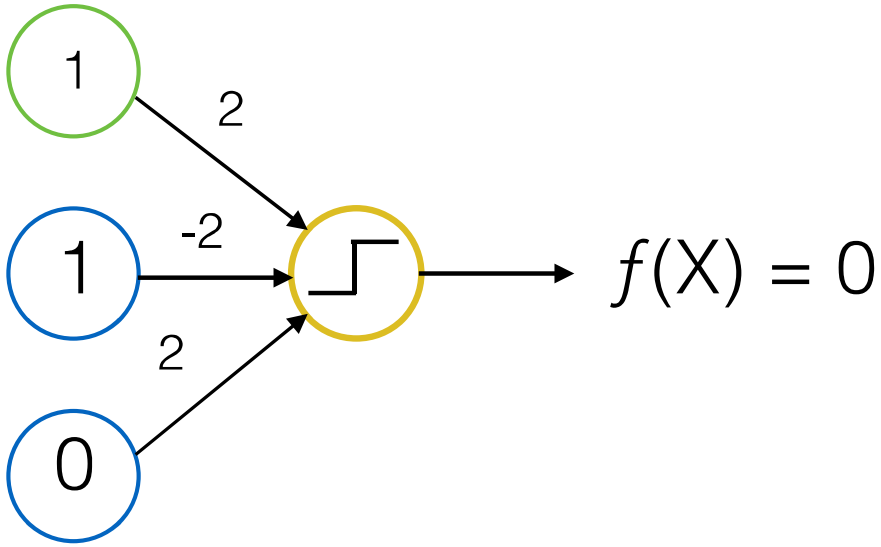
Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1



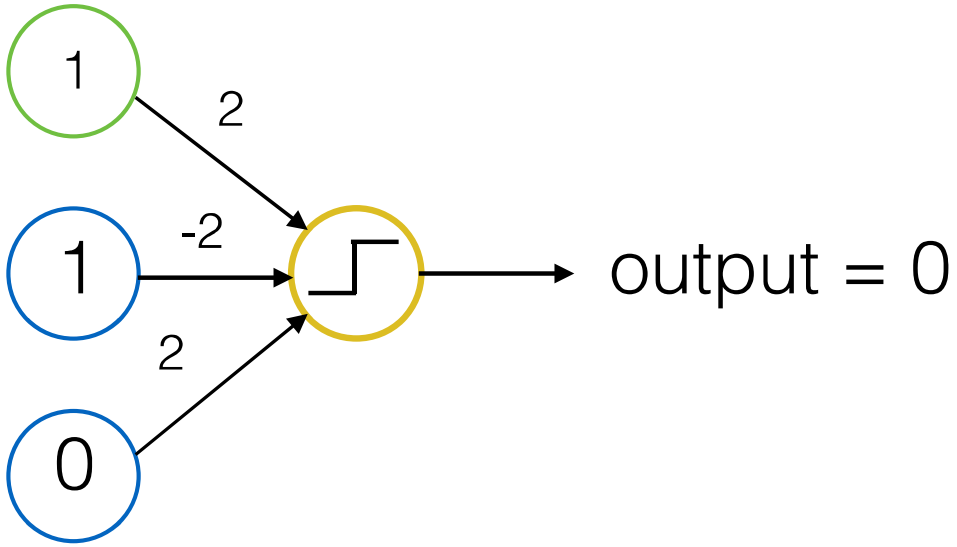
Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1



Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1



Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	1



weights: 2, -2, 2

output: 0

input: 1, 1, 0

target: 1

updated $weight_0$ = $weight_0$ - (output - target) * $input_0$

updated $weight_1$ = $weight_1$ - (output - target) * $input_1$

updated $weight_2$ = $weight_2$ - (output - target) * $input_2$

weights: 2, -2, 2

input: 1, 1, 0

output: 0

target: 1

$$\text{updated weight}_0 = 2 - (\text{output} - \text{target}) * \text{input}_0$$

$$\text{updated weight}_1 = -2 - (\text{output} - \text{target}) * \text{input}_1$$

$$\text{updated weight}_2 = 2 - (\text{output} - \text{target}) * \text{input}_2$$

weights: 2, -2, 2

output: 0

input: 1, 1, 0

target: 1

$$\text{updated weight}_0 = 2 - (0 - \text{target}) * \text{input}_0$$

$$\text{updated weight}_1 = -2 - (0 - \text{target}) * \text{input}_1$$

$$\text{updated weight}_2 = 2 - (0 - \text{target}) * \text{input}_2$$

weights: 2, -2, 2

output: 0

input: 1, 1, 0

target: 1

$$\text{updated weight}_0 = 2 - (0 - 1) * \text{input}_0$$

$$\text{updated weight}_1 = -2 - (0 - 1) * \text{input}_1$$

$$\text{updated weight}_2 = 2 - (0 - 1) * \text{input}_2$$

weights: 2, -2, 2

output: 0

input: 1, 1, 0

target: 1

$$\text{updated weight}_0 = 2 - (0 - 1) * 1$$

$$\text{updated weight}_1 = -2 - (0 - 1) * 1$$

$$\text{updated weight}_2 = 2 - (0 - 1) * 0$$

weights: 2, -2, 2

output: 0

input: 1, 1, 0

target: 1

$$\text{updated weight}_0 = 2 - (-1)$$

$$\text{updated weight}_1 = -2 - (-1)$$

$$\text{updated weight}_2 = 2 - 0$$

weights: 2, -2, 2

input: 1, 1, 0

output: 0

target: 1

$$\text{updated weight}_0 = 2 + 1$$

$$\text{updated weight}_1 = -2 + 1$$

$$\text{updated weight}_2 = 2 - 0$$

weights: 2, -2, 2

output: 0

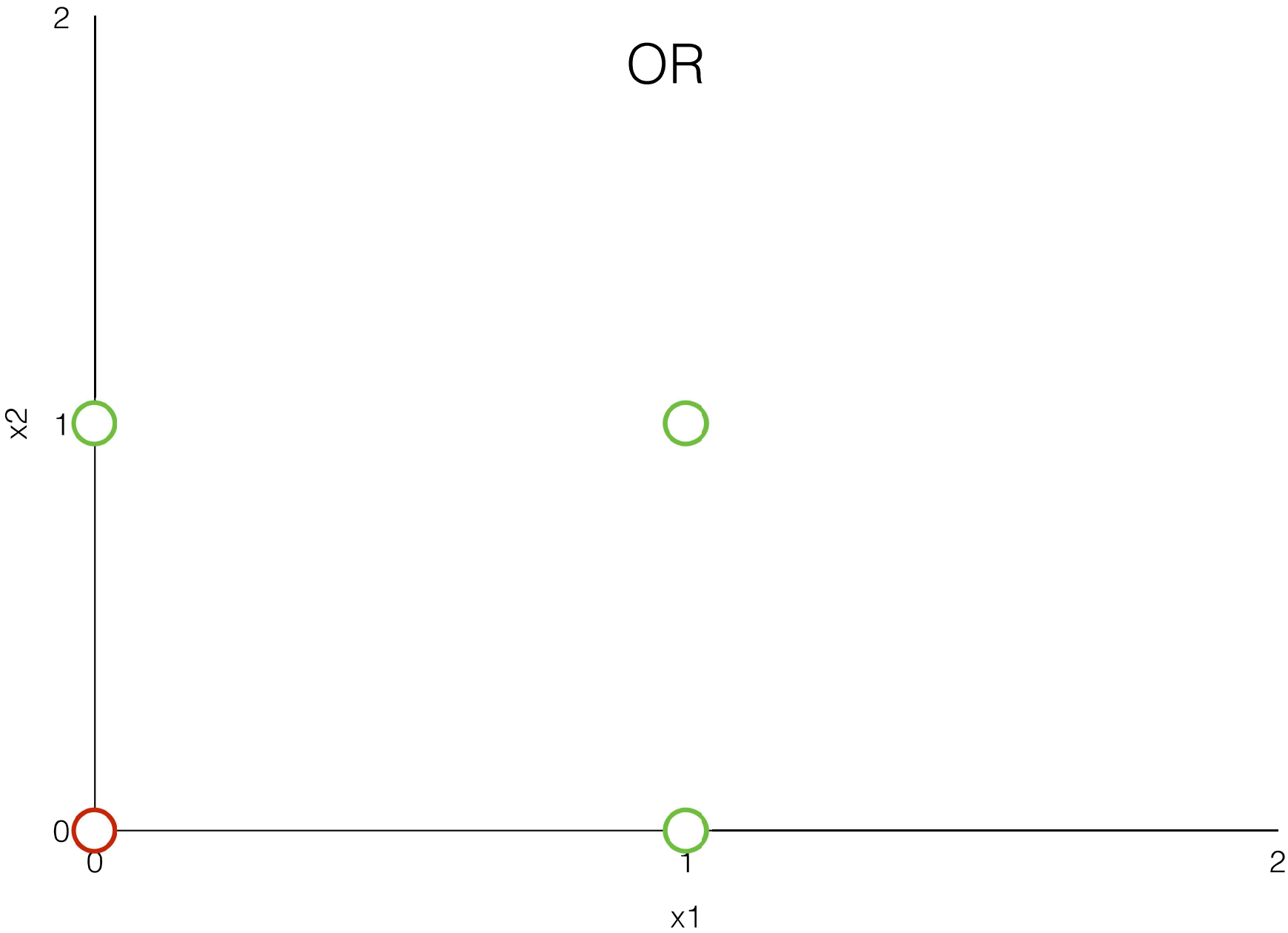
input: 1, 1, 0

target: 1

updated $\text{weight}_0 = 3$

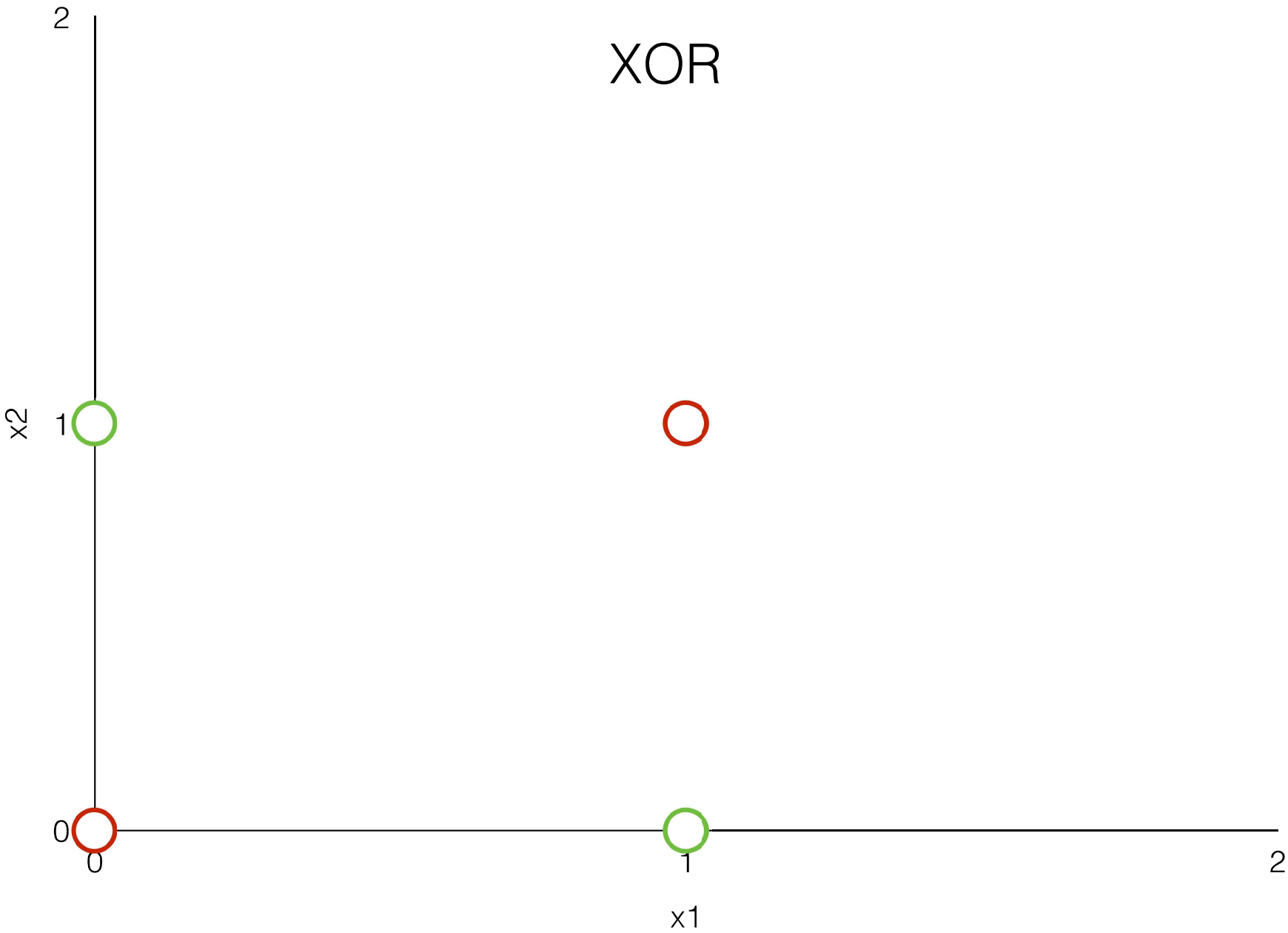
updated $\text{weight}_1 = -1$

updated $\text{weight}_2 = 2$

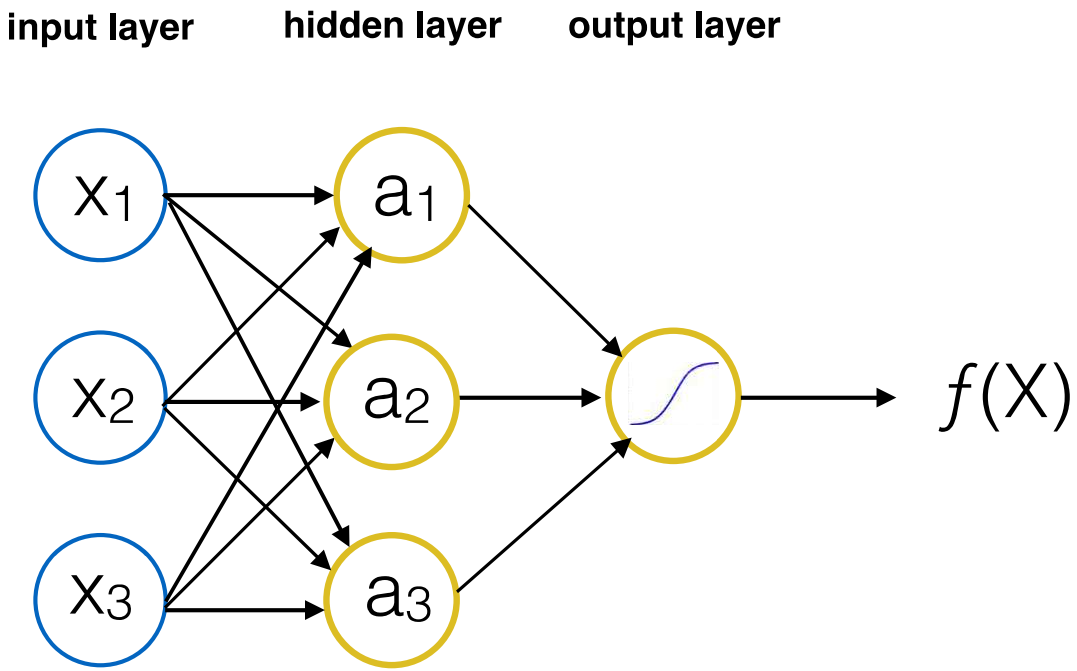

OR
2
x2
1
0
0
1
2
x1

XOR

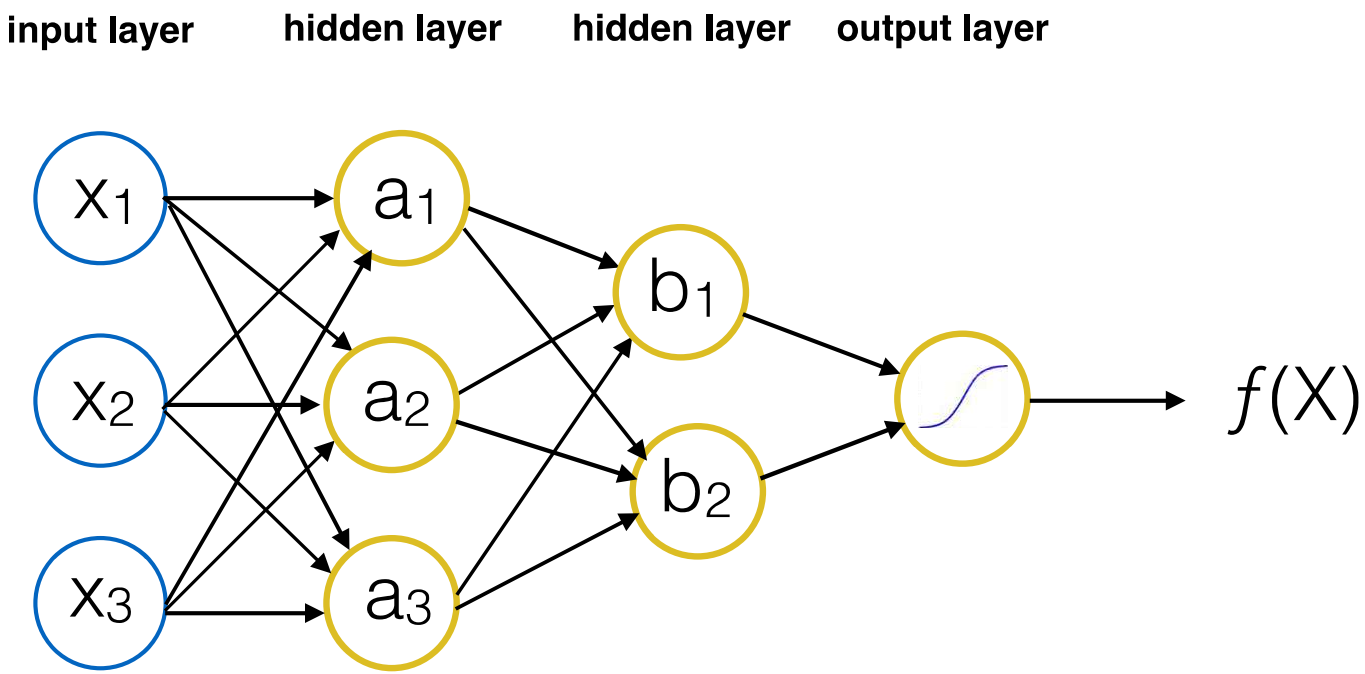
Feature 1	Feature 2	Target
0	0	0
1	0	1
0	1	1
1	1	0


XOR
2
1
0
x2
0
1
2
x1

Multi-Layer Perceptron (MLP)



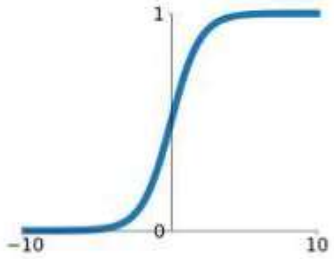
Multi-Layer Perceptron (MLP)



Activation Functions

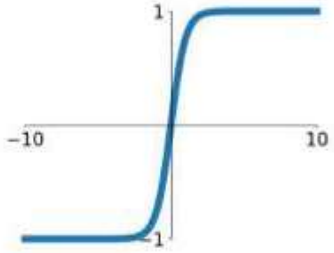
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



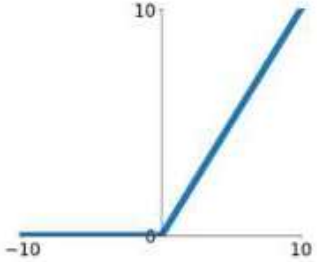
tanh

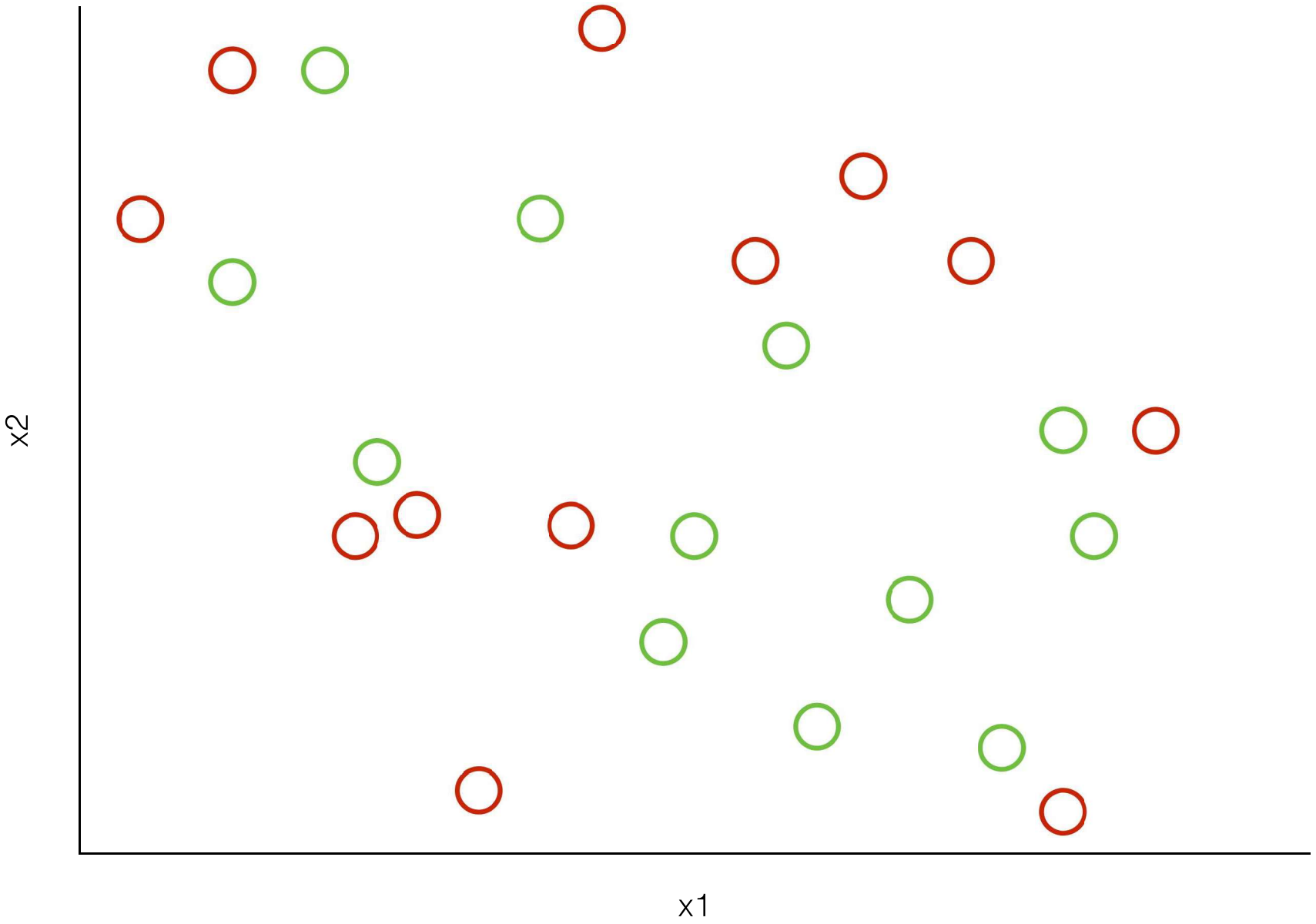
$$\tanh(x)$$

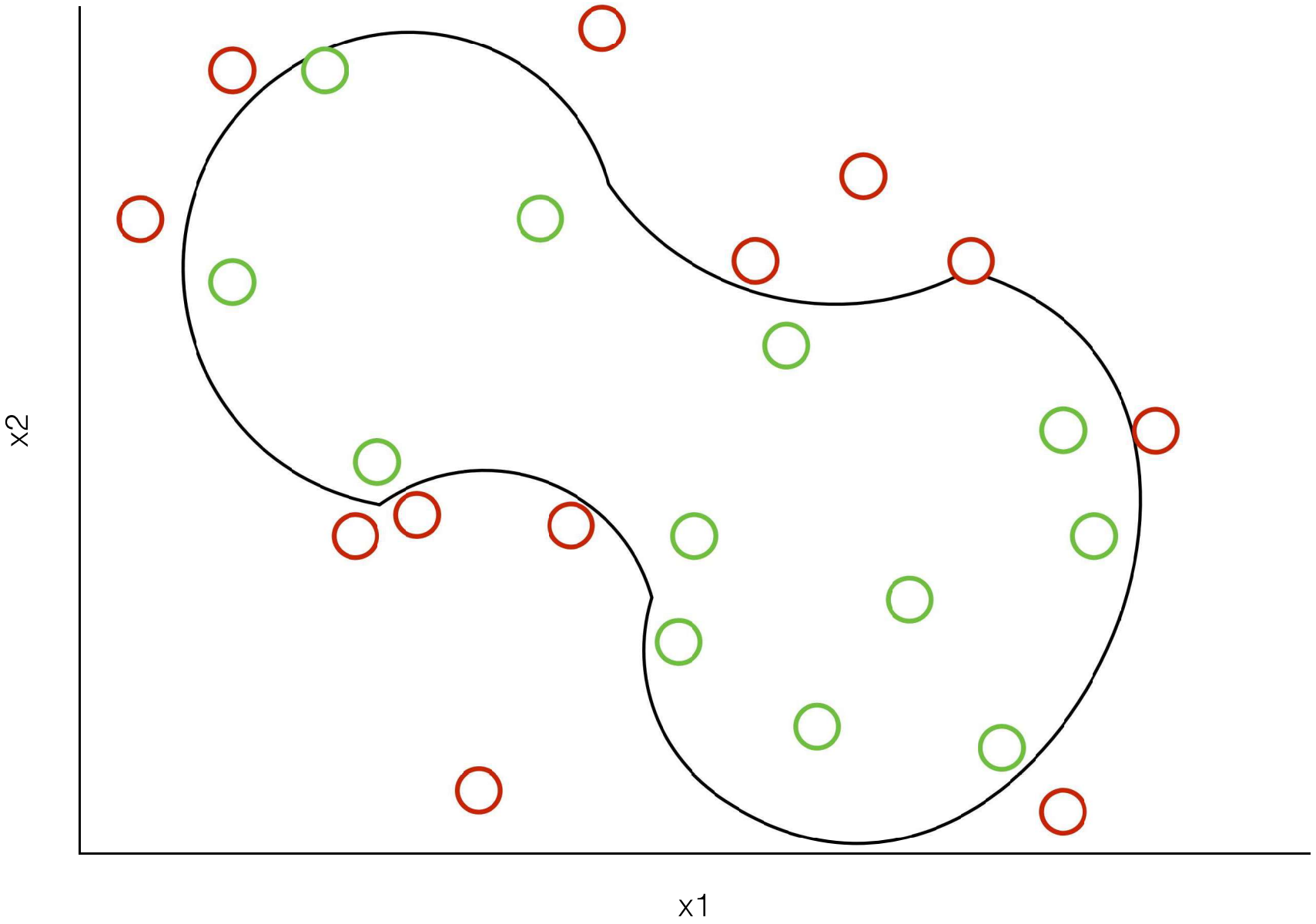


ReLU

$$\max(0, x)$$




x2
x1


x2
x1

Model Selection

Test/Train Split

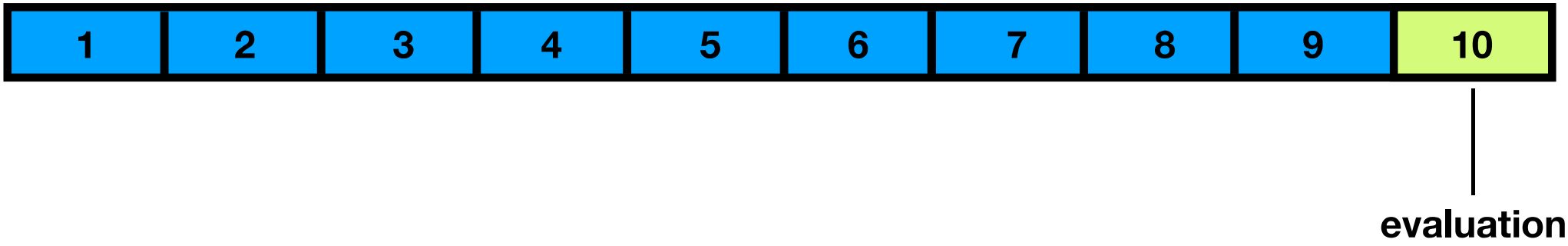
DATA SET

DATA SET

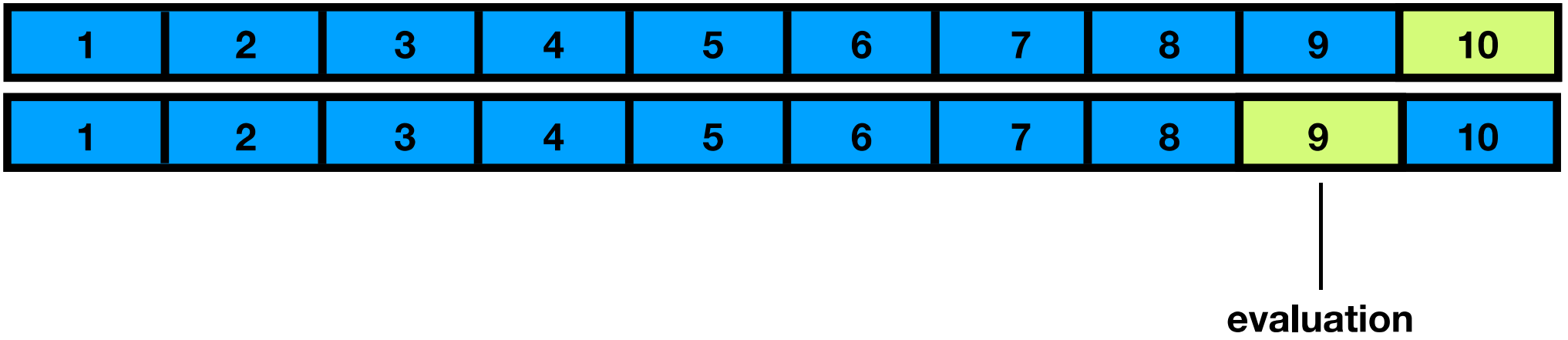
70%

K-fold cross validation

10-Fold Cross Validation

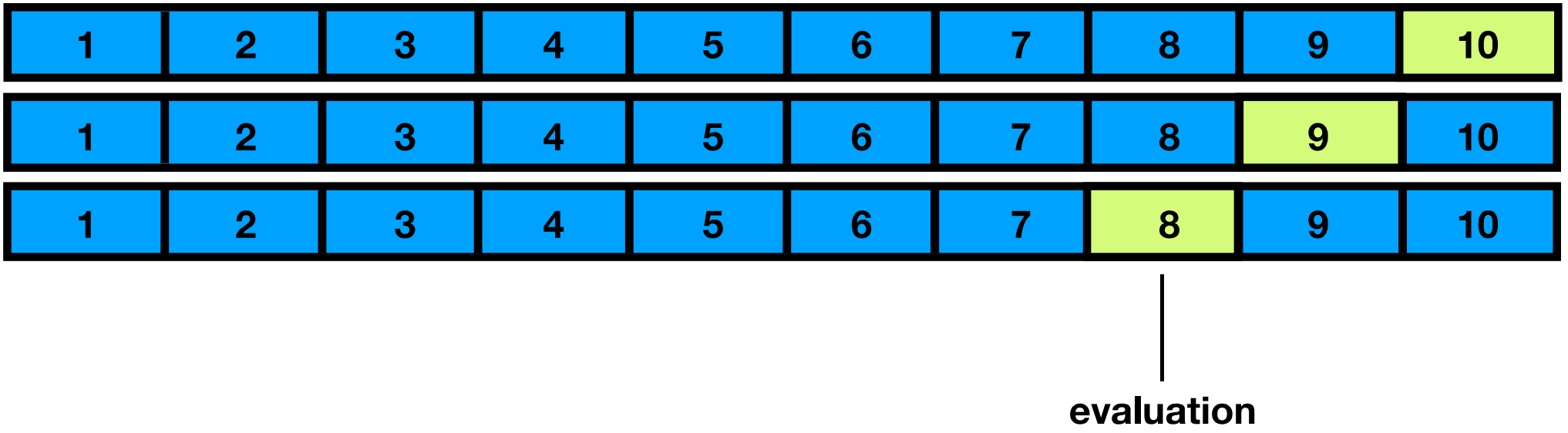


10-Fold Cross Validation



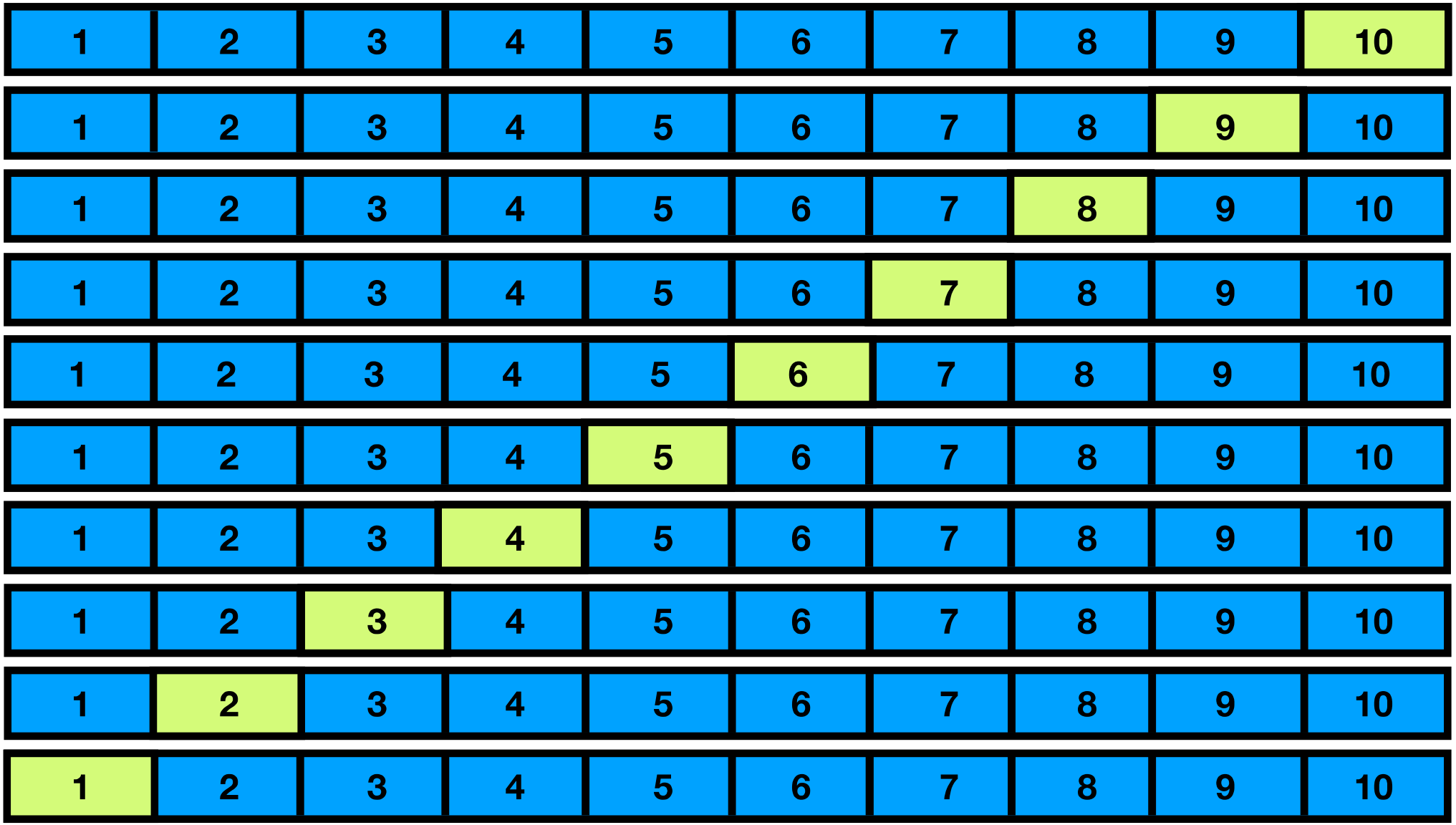
10-Fold Cross Validation

Training Set



10-Fold Cross Validation

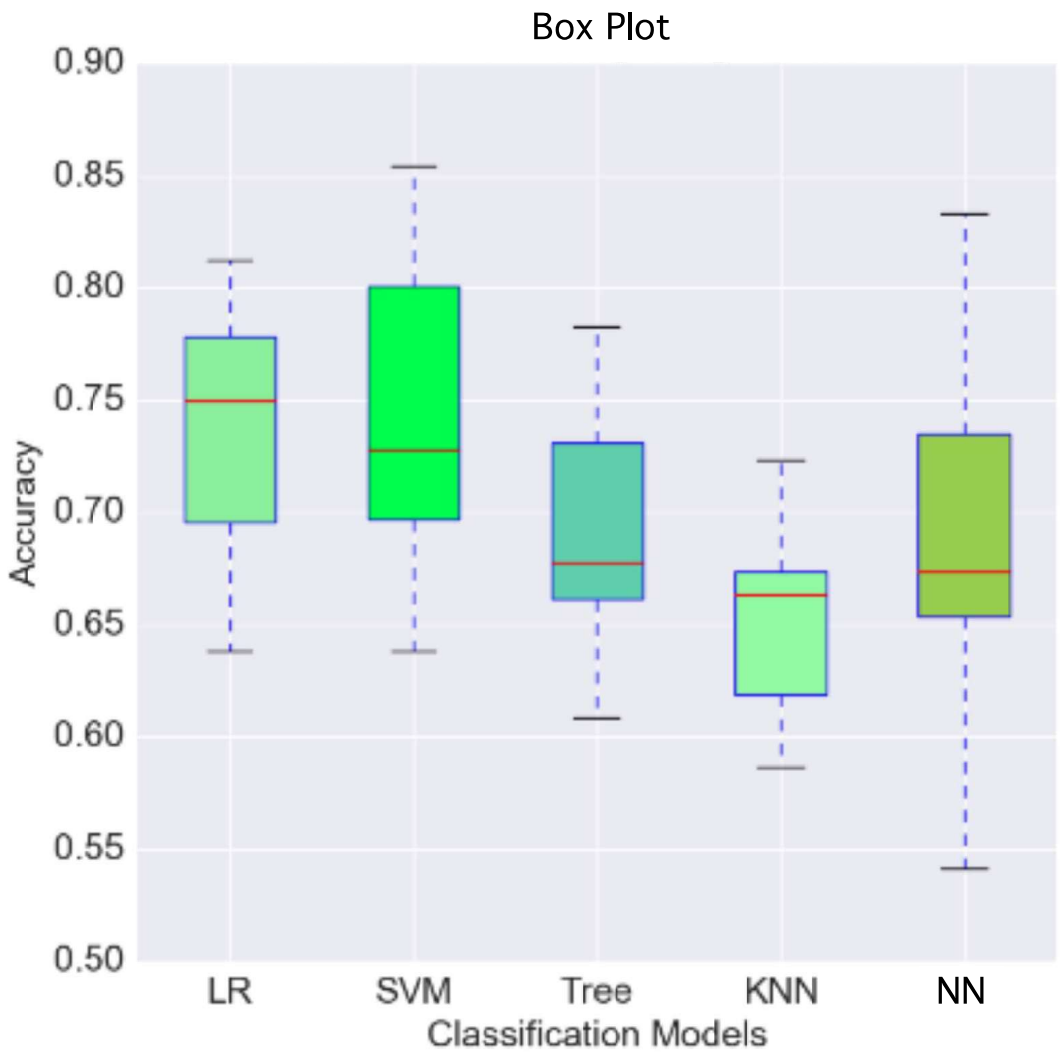
Training Set



Logistic Regression

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	mean
0.69	0.64	0.73	0.82	0.64	0.70	0.68	0.71	0.70	0.69	**0.70**

Logistic Regression	Support Vector Machine	Decision Tree	K-Nearest Neighbor	Neural Network
0.705	0.722	0.635	0.675	0.607



Hyperparameter Tuning

Logistic Regression

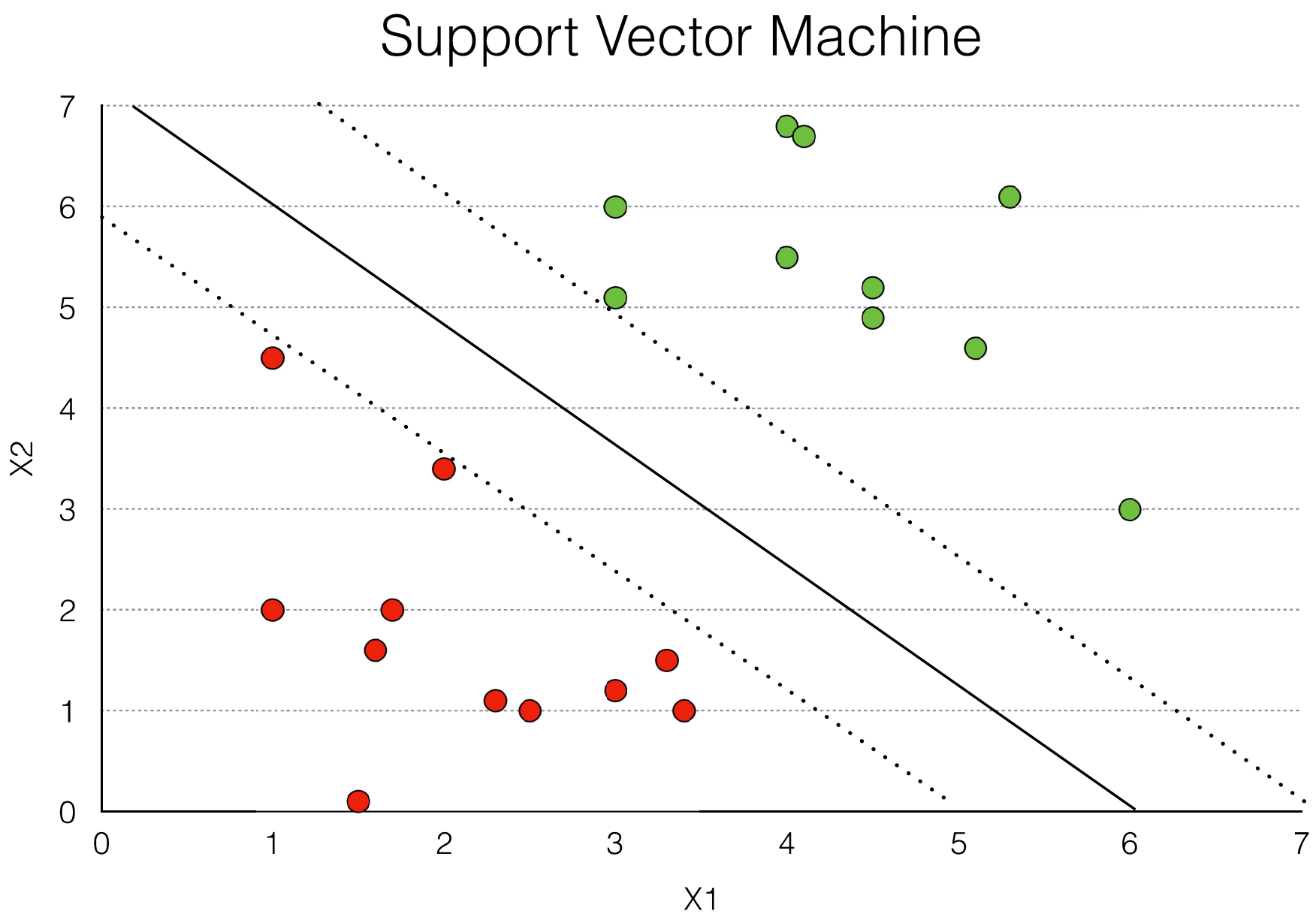
$$\hat{f}(X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2)}}$$

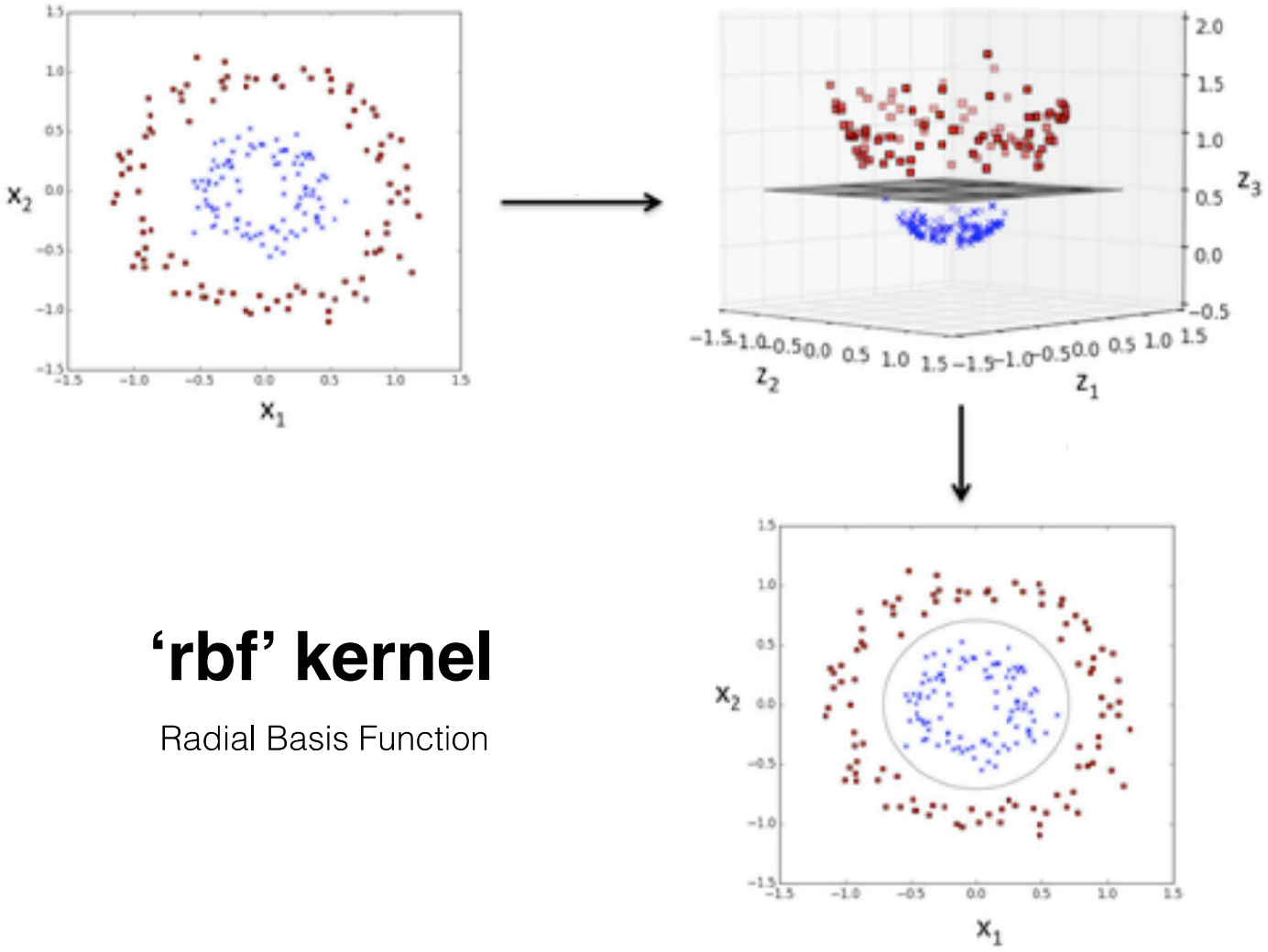
$$LOC = 227.63 + 9.51x_1 + 2.7x_2 - 7.08x_3$$

x_1 = hour pair programming

x_2 = gender (m = 0; f = 1)

x_3 = number of social accounts

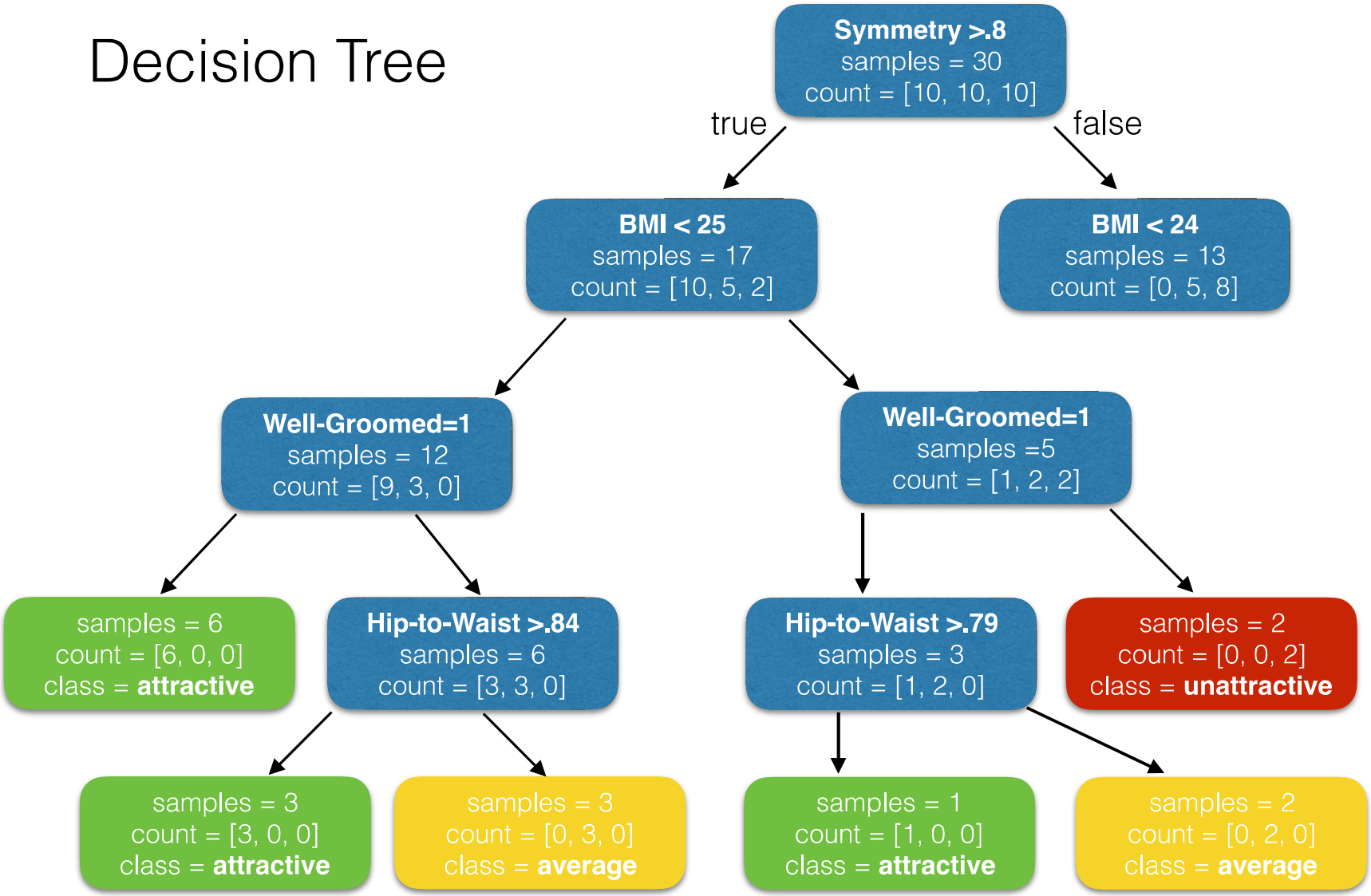

Support Vector Machine
X2
X1
0
1
2
3
4
5
6
7



‘rbf’ kernel

Radial Basis Function

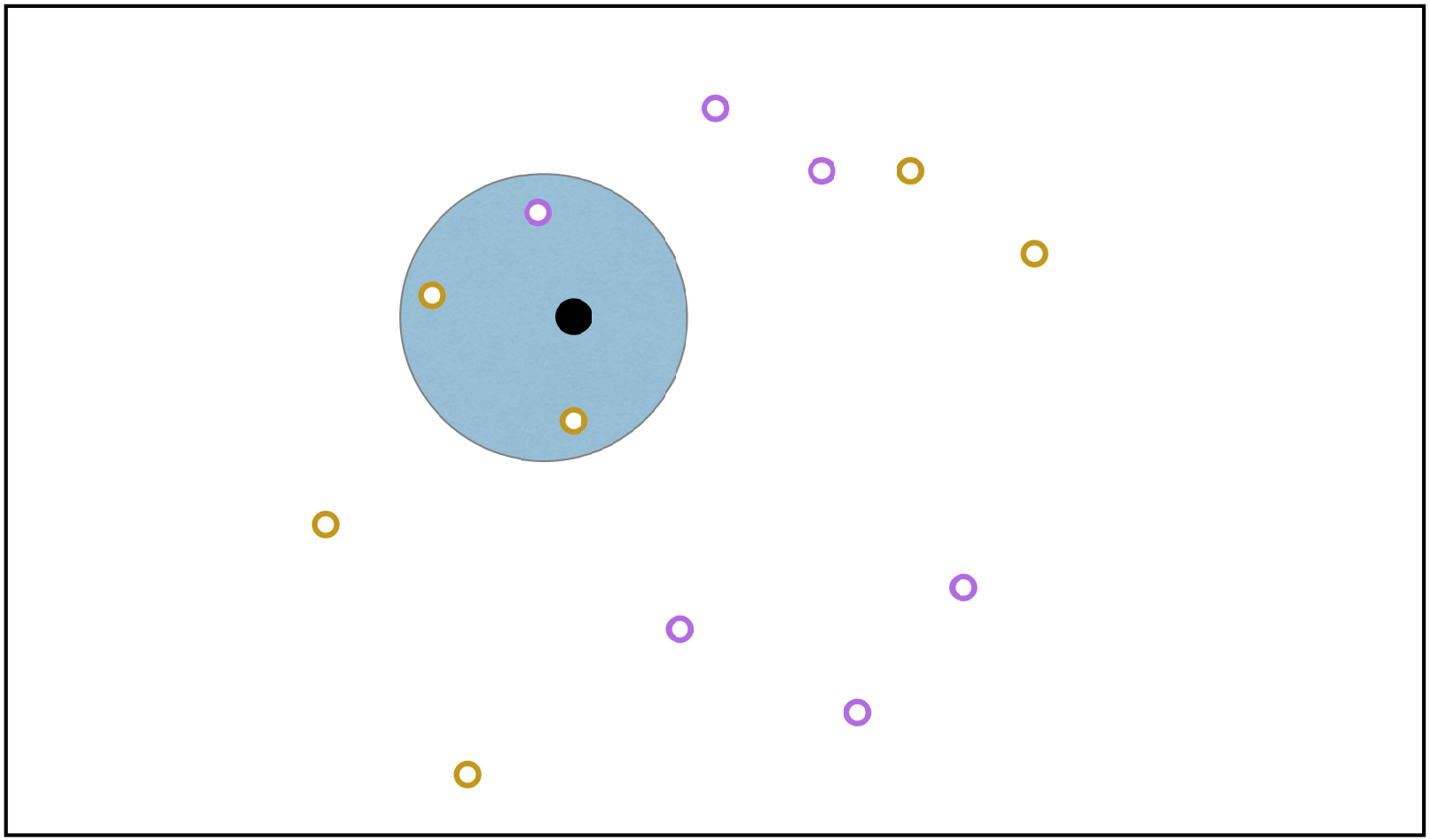
Decision Tree



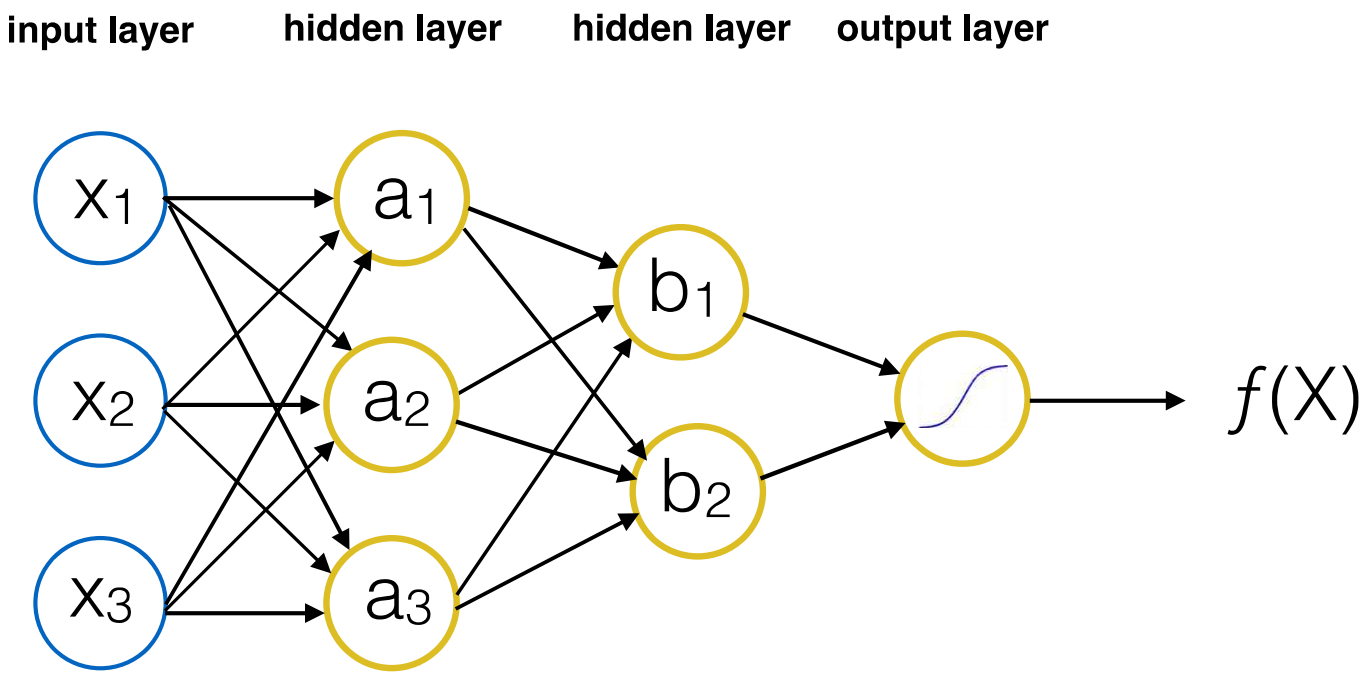
[att, ave, un]

k-Nearest Neighbor

k = 3



Multi-Layer Perceptron (MLP)



Grid Search

Grid Search

Support Vector Machine

Grid Search

Support Vector Machine

param_grid=[{'C': [.1, 1, 10]}], ‘kernel’: [‘linear’, ‘rbf’], cv=3)}]

C	kernel
0.1	linear’
1	linear’
10	linear’
0.1	‘rbf’
1	'rbf’
10	‘rbf’

Training Set

DATA SET

DATA SET

30%