Classification: Logistic Regression

Hung-yi Lee

李宏毅

Step 1: Function Set

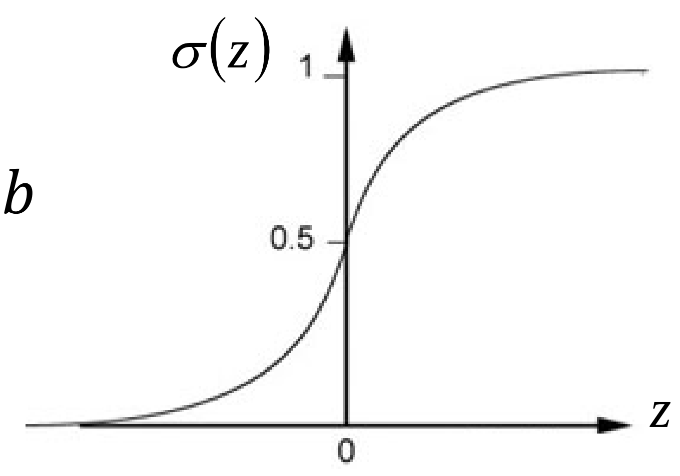
Function set: Including all different w and b

$$\begin{cases} z \geq 0 & \text{class 1} \\ z < 0 & \text{class 2} \end{cases}$$

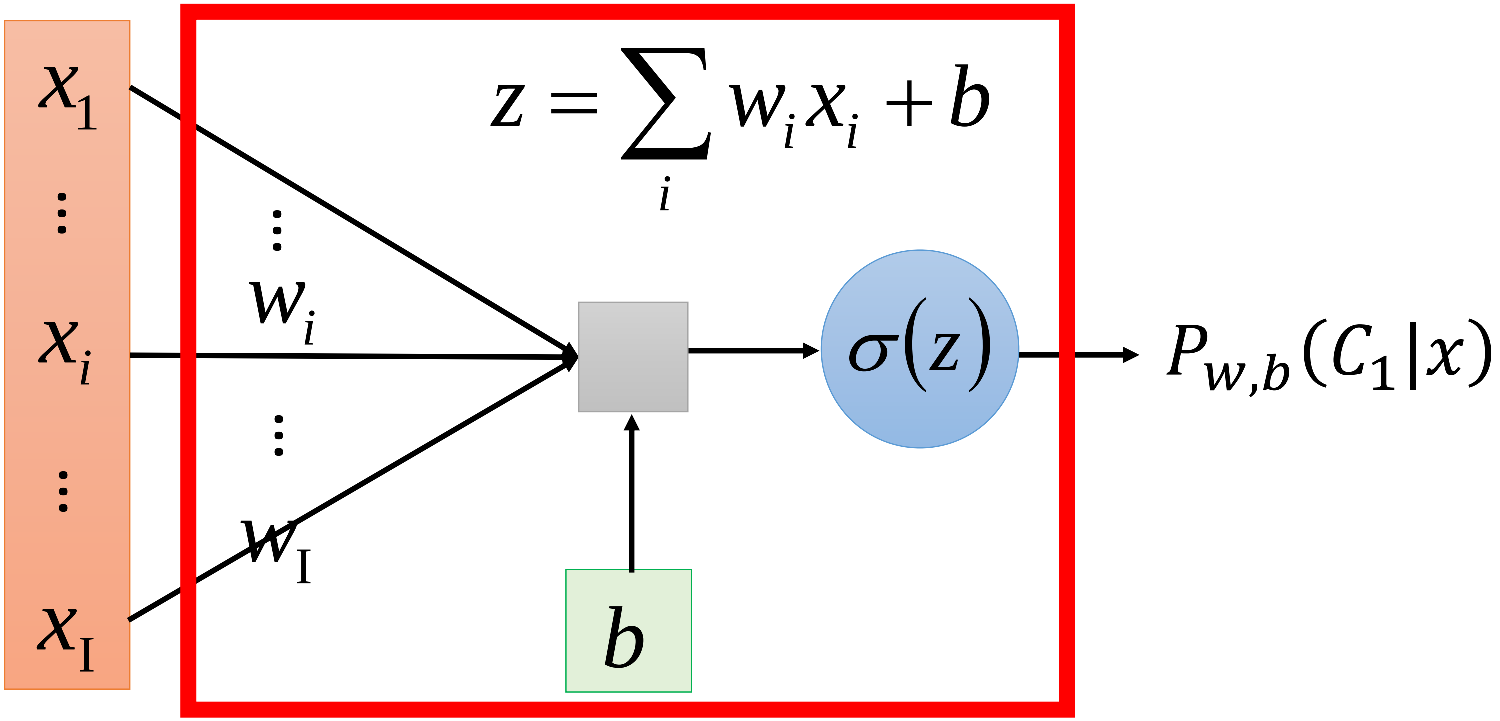
$$P_{w,b}(C_1|x) = \sigma(z)$$

$$z = w \cdot x + b = \sum_i w_i x_i + b$$

$$\sigma(z) = \frac{1}{1 + exp(-z)}$$



Step 1: Function Set



Step 2: Goodness of a Function

Training Data	x^1	x^2	x^3	……	x^N
	C_1	C_1	C_2		C_1

Assume the data is generated based on $f_{w,b}(x) = P_{w,b}(C_1|x)$

Given a set of w and b, what is its probability of generating the data?

$$L(w,b) = f_{w,b}(x^1)f_{w,b}(x^2)\left(1 - f_{w,b}(x^3)\right) \cdots f_{w,b}(x^N)$$

The most likely w* and b* is the one with the largest $L(w,b)$.

$$w^*, b^* = arg \max_{w,b} L(w,b)$$

Step 2: Goodness of a Function

$$L(w,b) = f_{w,b}(x^1) f_{w,b}(x^2) \left(1 - f_{w,b}(x^3)\right) \cdots f_{w,b}(x^N)$$

$$-lnL(w,b) = ln f_{w,b}(x^1) + ln f_{w,b}(x^2) + ln\left(1 - f_{w,b}(x^3)\right) \cdots$$

$\hat{y}^n$: 1 for class 1, 0 for class 2

$$= \sum_n \underbrace{-\left[\hat{y}^n ln f_{w,b}(x^n) + (1 - \hat{y}^n) ln\left(1 - f_{w,b}(x^n)\right)\right]}_{\text{Cross entropy between two Bernoulli distribution}}$$

Distribution p:	cross entropy	Distribution q:
$p(x=1) = \hat{y}^n$		$q(x=1) = f(x^n)$
$p(x=0) = 1 - \hat{y}^n$		$q(x=0) = 1 - f(x^n)$

$$H(p,q) = -\sum_x p(x) ln\left(q(x)\right)$$

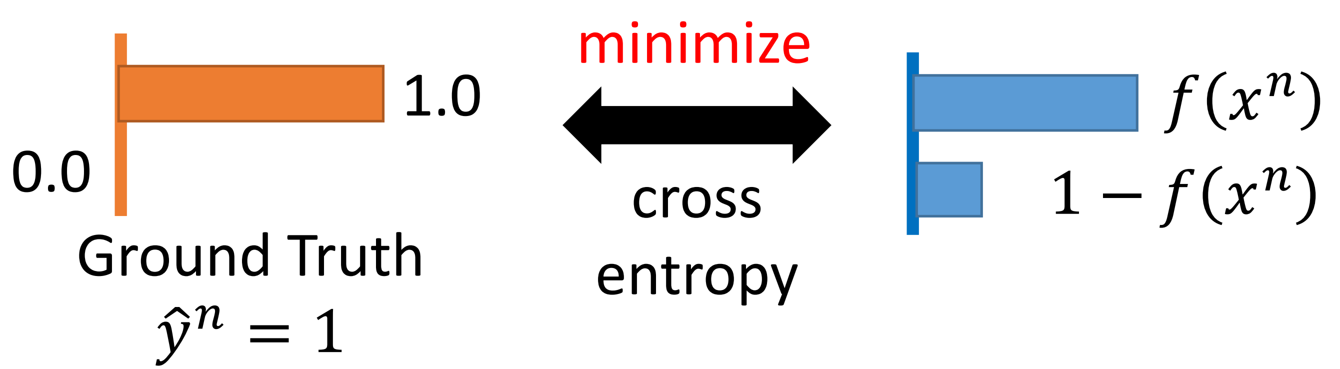
Step 2: Goodness of a Function

$$L(w,b) = f_{w,b}(x^1) f_{w,b}(x^2) \left(1 - f_{w,b}(x^3)\right) \cdots f_{w,b}(x^N)$$

$$-lnL(w,b) = lnf_{w,b}(x^1) + lnf_{w,b}(x^2) + ln\left(1 - f_{w,b}(x^3)\right) \cdots$$

$\hat{y}^n$: 1 for class 1, 0 for class 2

$$= \sum_n \underbrace{-\left[\hat{y}^n lnf_{w,b}(x^n) + (1-\hat{y}^n) ln\left(1 - f_{w,b}(x^n)\right)\right]}_{\text{Cross entropy between two Bernoulli distribution}}$$



Step 3: Find the best function

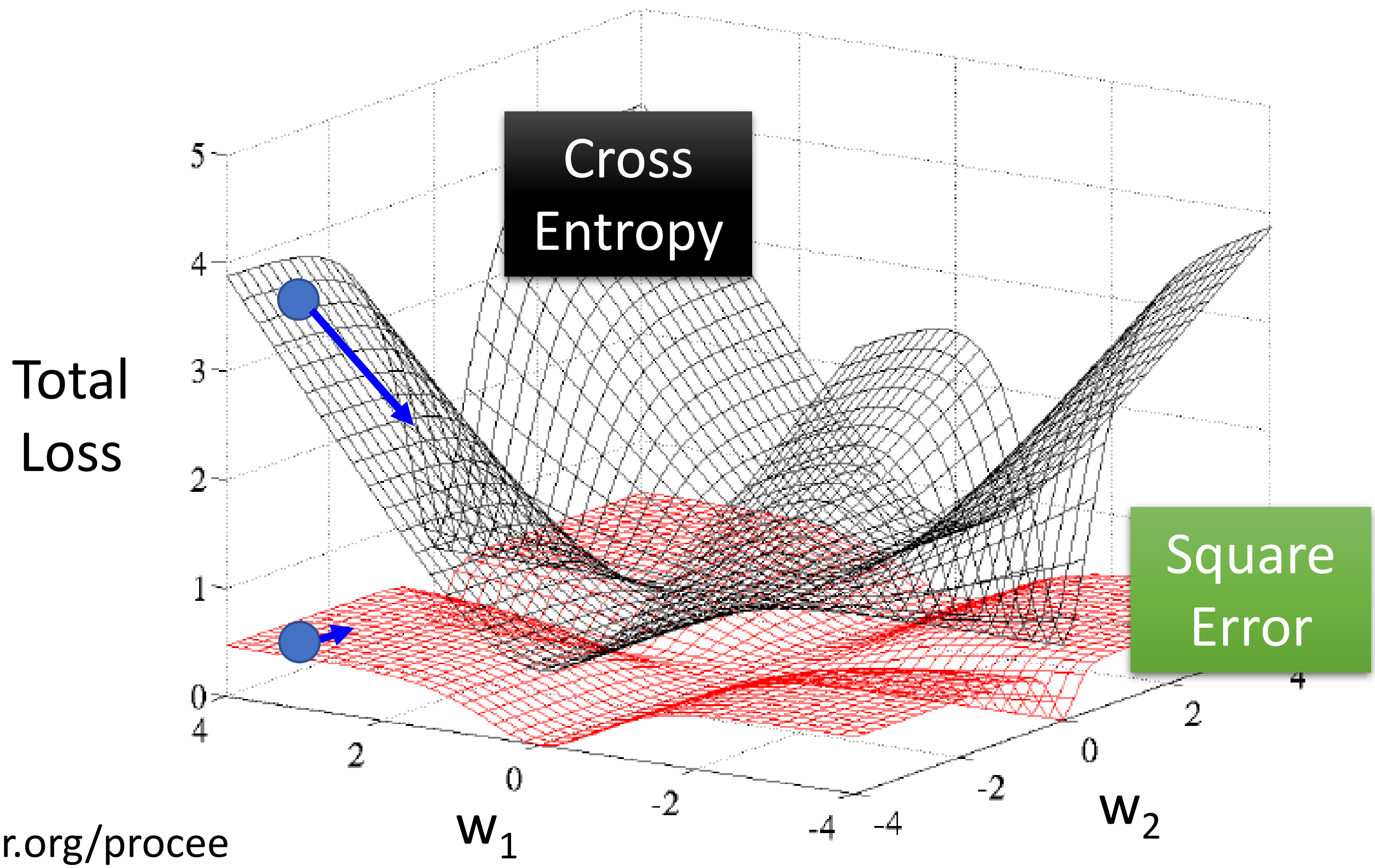
$$\frac{-lnL(w,b)}{\partial w_i} = \sum_n -\left[\hat{y}^n \overset{\left(1-f_{w,b}(x^n)\right)x_i^n}{\boxed{\frac{lnf_{w,b}(x^n)}{\partial w_i}}} + (1-\hat{y}^n)\overset{-f_{w,b}(x^n)x_i^n}{\boxed{\frac{ln\left(1-f_{w,b}(x^n)\right)}{\partial w_i}}}\right]$$

$$= \sum_n -\left(\hat{y}^n - f_{w,b}(x^n)\right)x_i^n$$

$$f_{w,b}(x) = \sigma(z)$$
$$= 1/1 + exp(-z)$$

$$z = w \cdot x + b = \sum_i w_i x_i + b$$

Cross Entropy v.s. Square Error



http://jmlr.org/proceedings/papers/v9/glorot10a/glorot10a.pdf

	<u>***Logistic Regression***</u>	<u>***Linear Regression***</u>
Step 1:	$f_{w,b}(x) = \sigma\left(\sum_i w_i x_i + b\right)$ Output: between 0 and 1	$f_{w,b}(x) = \sum_i w_i x_i + b$ Output: any value
Step 2:	Training data: $(x^n, \hat{y}^n)$ $\hat{y}^n$: 1 for class 1, 0 for class 2 $L(f) = \sum_n l(f(x^n), \hat{y}^n)$	Training data: $(x^n, \hat{y}^n)$ $\hat{y}^n$: a real number $L(f) = \frac{1}{2}\sum_n \left(f(x^n) - \hat{y}^n\right)^2$

Step 3:

Logistic regression: $w_i \leftarrow w_i - \eta \sum_n -\left(\underline{\hat{y}^n - f_{w,b}(x^n)}\right) x_i^n$

Linear regression: $w_i \leftarrow w_i - \eta \sum_n -\left(\underline{\hat{y}^n - f_{w,b}(x^n)}\right) x_i^n$

Discriminative v.s. Generative

$$P(C_1|x) = \sigma(w \cdot x + b)$$

directly find ***w*** and b

Find $\mu^1, \mu^2, \Sigma^{-1}$

$$w^T = (\mu^1 - \mu^2)^T \Sigma^{-1}$$

$$b = -\frac{1}{2}(\mu^1)^T(\Sigma^1)^{-1}\mu^1 + \frac{1}{2}(\mu^2)^T(\Sigma^2)^{-1}\mu^2 + ln\frac{N_1}{N_2}$$

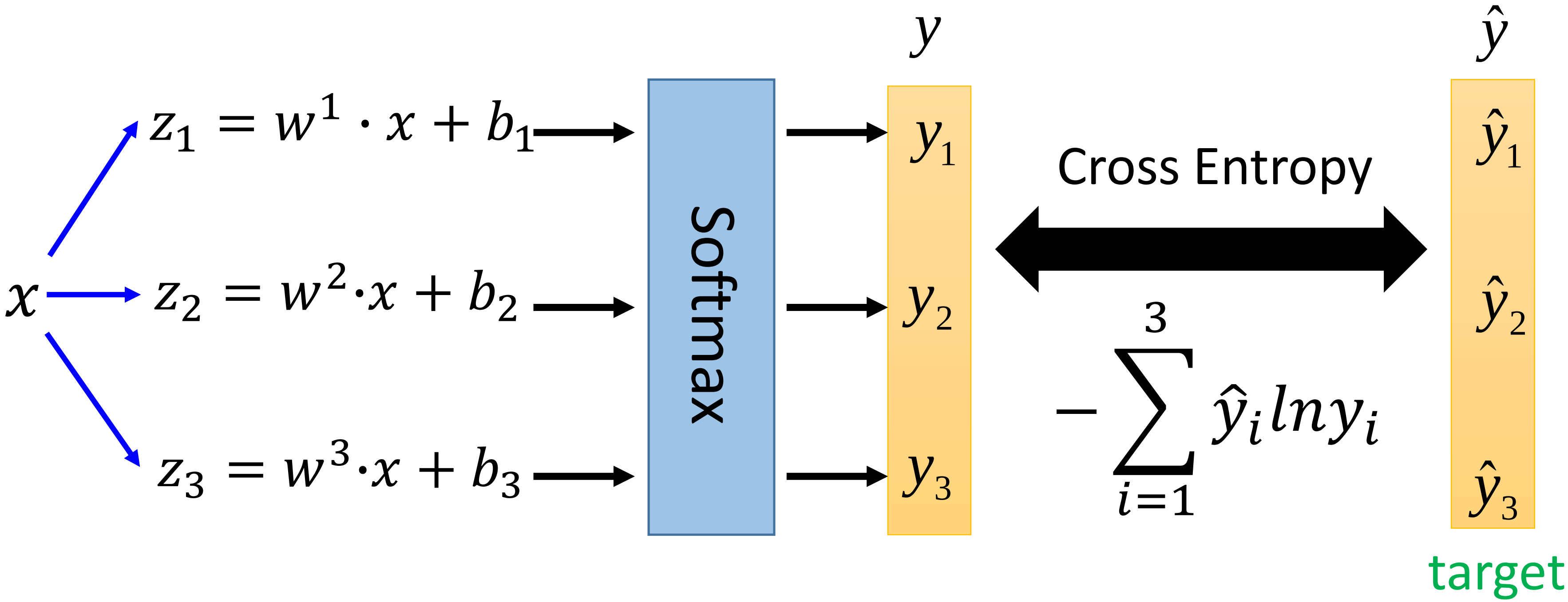
The same model (function set), but different function may be selected by the same training data.

Generative v.s. Discriminative

- Usually people believe discriminative model is better
- Benefit of generative model
  - With the assumption of probability distribution
    - less training data is needed
    - more robust to the noise
  - Priors and class-dependent probabilities can be estimated from different sources.

Multi-class Classification (3 classes as example)

[Bishop, P209-210]



If x ∈ class 1

$$\hat{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$-ln y_1$$

If x ∈ class 2

$$\hat{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$-ln y_2$$

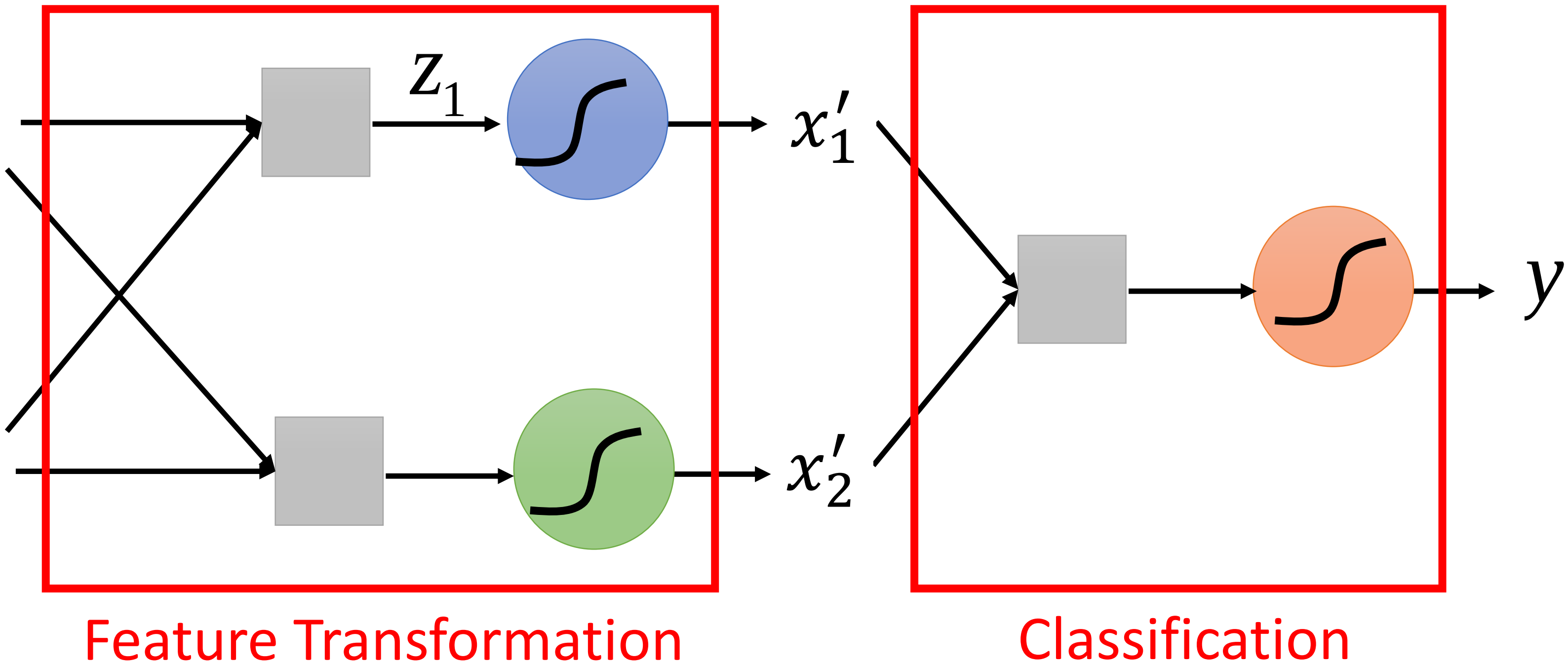
If x ∈ class 3

$$\hat{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$-ln y_3$$

Limitation of Logistic Regression

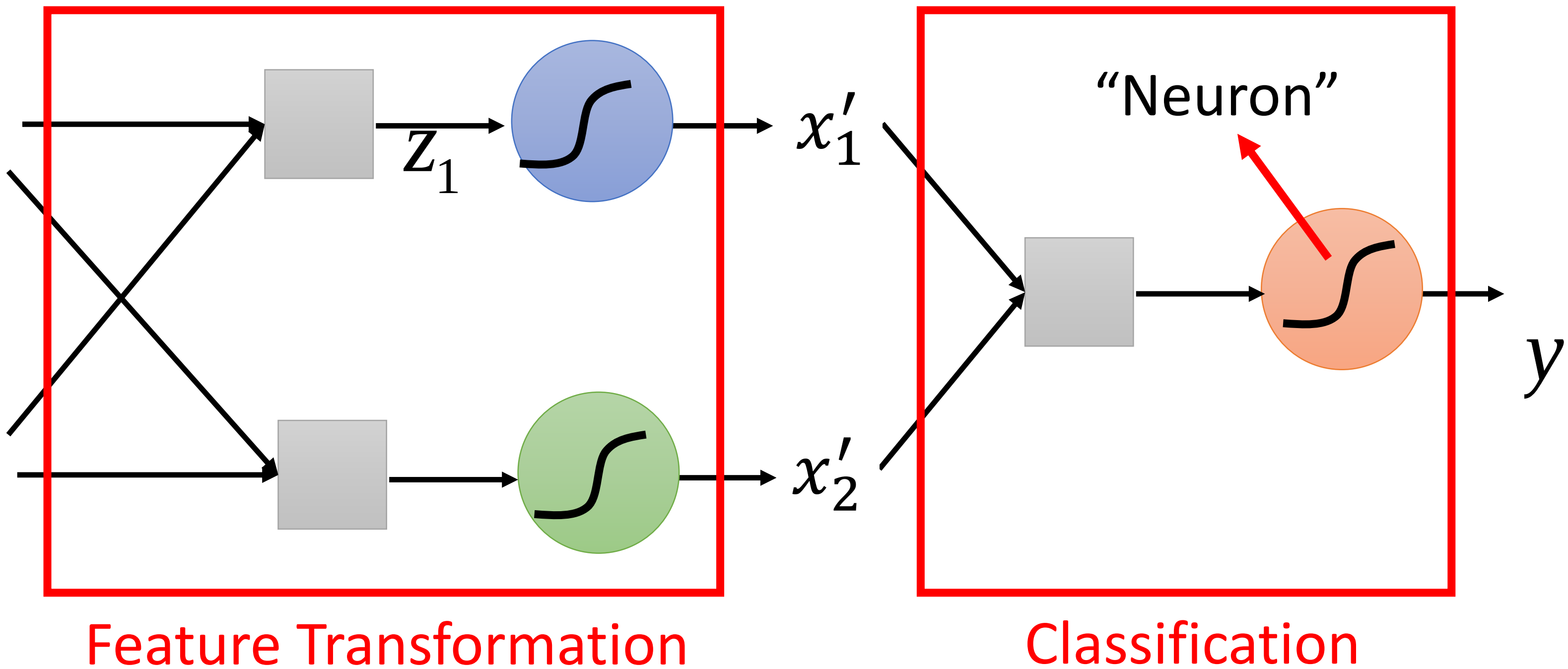
- ***Feature transformation: change the data feature by regression***
- Cascading logistic regression models



(ignore bias in this figure)

Deep Learning!

All the parameters of the logistic regressions are jointly learned.



Neural Network