First Lecture of Machine Learning

Hung-yi Lee

Learning to say “yes/no”

Binary Classification

Learning to say yes/no

- **Spam filtering**
  - Is an e-mail spam or not?
- **Recommendation systems**
  - recommend the product to the customer or not?
- **Malware detection**
  - Is the software malicious or not?
- **Stock prediction**
  - Will the future value of a stock increase or not with respect to its current value?

Binary Classification

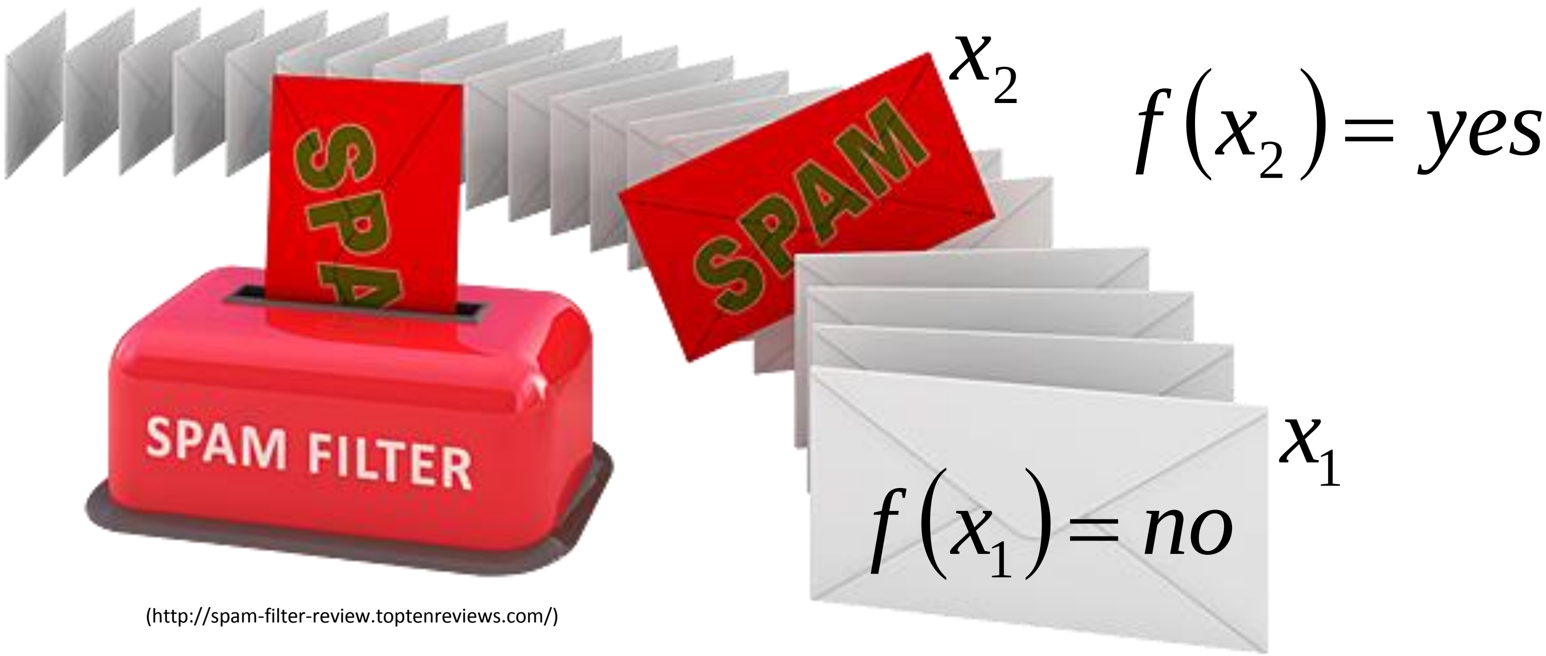
Example Application: Spam filtering

$$f : X \rightarrow \mathrm{Y} = \{yes, no\}$$

E-mail

Spam

Not spam



(http://spam-filter-review.toptenreviews.com/)

Example Application: Spam filtering

$$f : X \rightarrow Y = \{yes, no\}$$

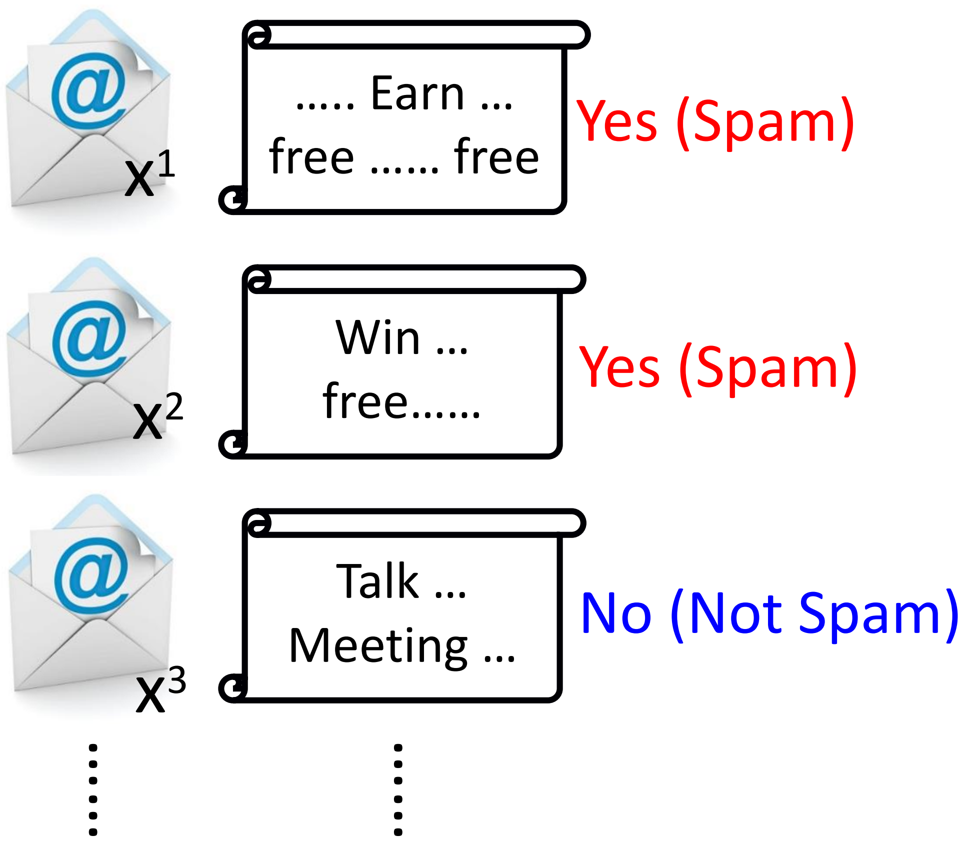
- What does the function f look like?

$$y = f(x) = \begin{cases} yes & P(yes \mid x) \geq 0.5 \\ no & P(yes \mid x) < 0.5 \end{cases}$$

How to estimate P(yes|x)?

Example Application: Spam filtering

- To estimate P(yes|x), collect examples first

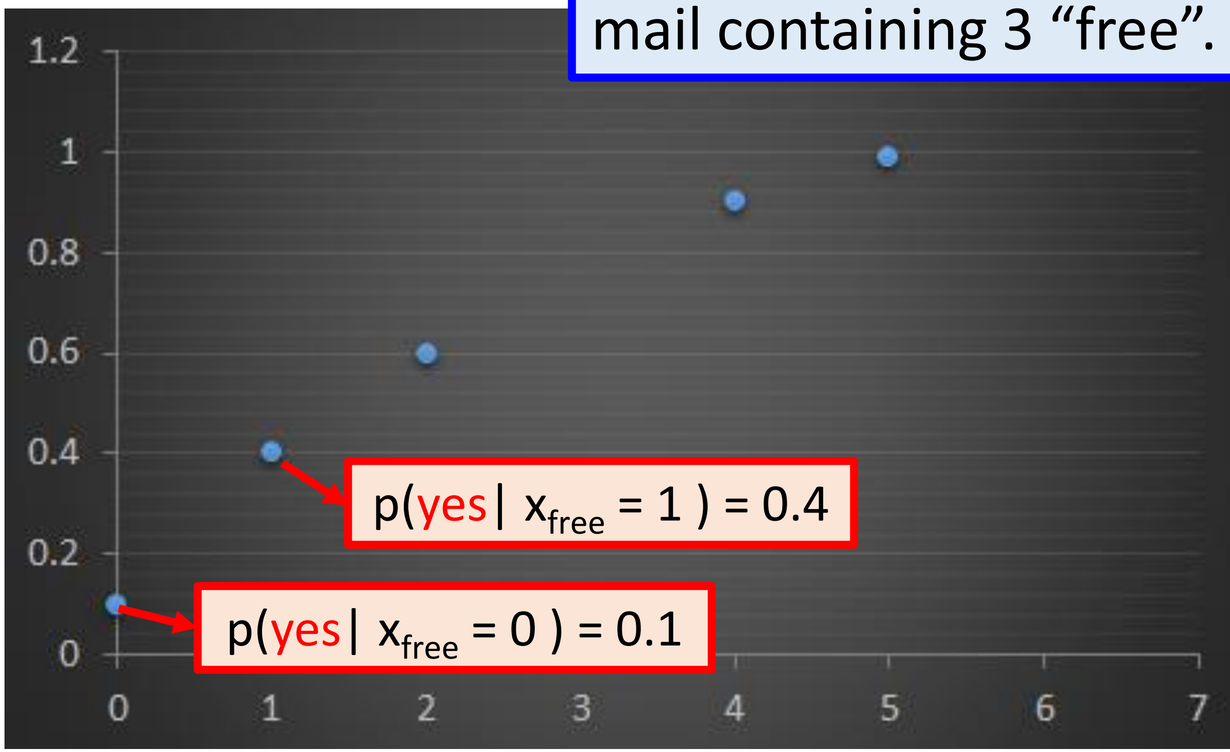


- ➢ Some words frequently appear in the spam
  e.g., “free”
- ➢ Use the frequency of “free” to decide if an e-mail is spam
- ➢ Estimate $P(yes|x_{free} = k)$
  - x_{free} is the number of “free” in e-mail x

Regression

In training data, there is no e-mail containing 3 “free”.

$p(\text{yes} \mid x_{free})$



Frequency of “Free” (x_{free}) in an e-mail x

Problem: What if one day you receive an e-mail with 3 “free” ….

Regression

$f(x_{free})$ = $\mathbf{w}x_{free}$ + $\mathbf{b}$
($f(x_{free})$ is an estimate of $p(yes|x_{free})$)

Store **w** and **b**

$p(yes|x_{free})$

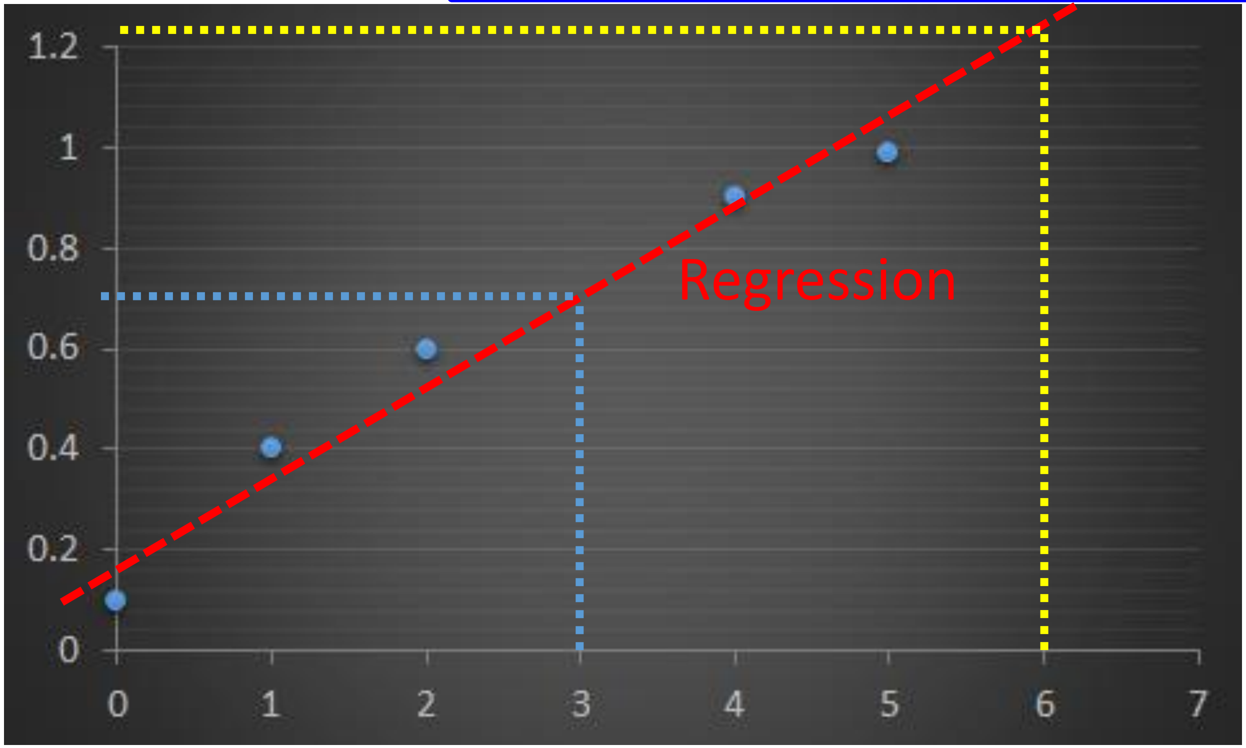
Frequency of “Free” (x_{free}) in an e-mail x

Regression

$f(x_{free}) = wx_{free} + b$

The output of **f** is not between 0 and 1

$p(\text{yes} | x_{free})$

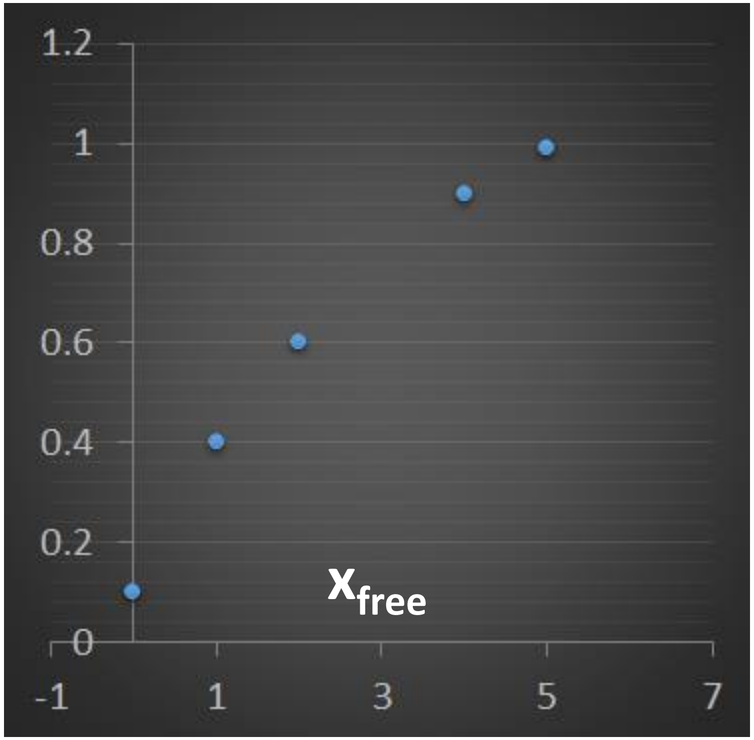


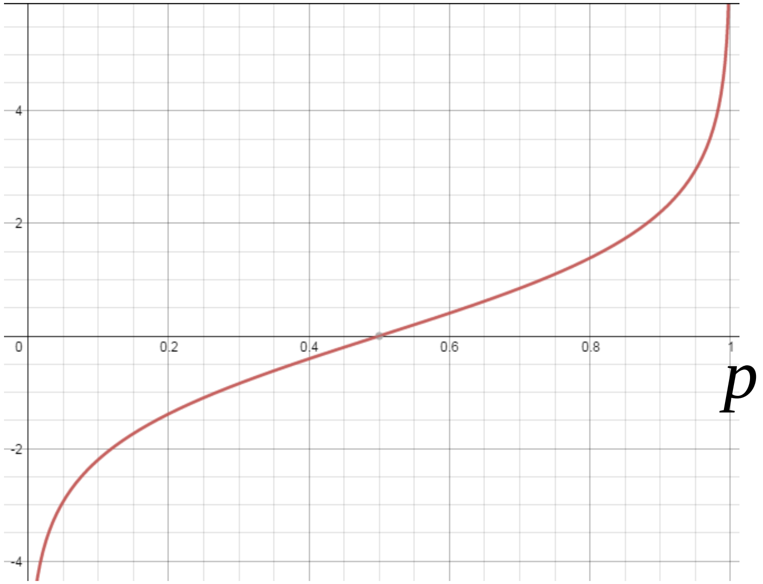
Frequency of "Free" (x_{free}) in an e-mail x

Problem: What if one day you receive an e-mail with 6 "free" ….

Logit

$$\ln\left(\frac{p}{1-p}\right)$$





vertical line: Probability to be spam p(yes|x_{free}) (p)

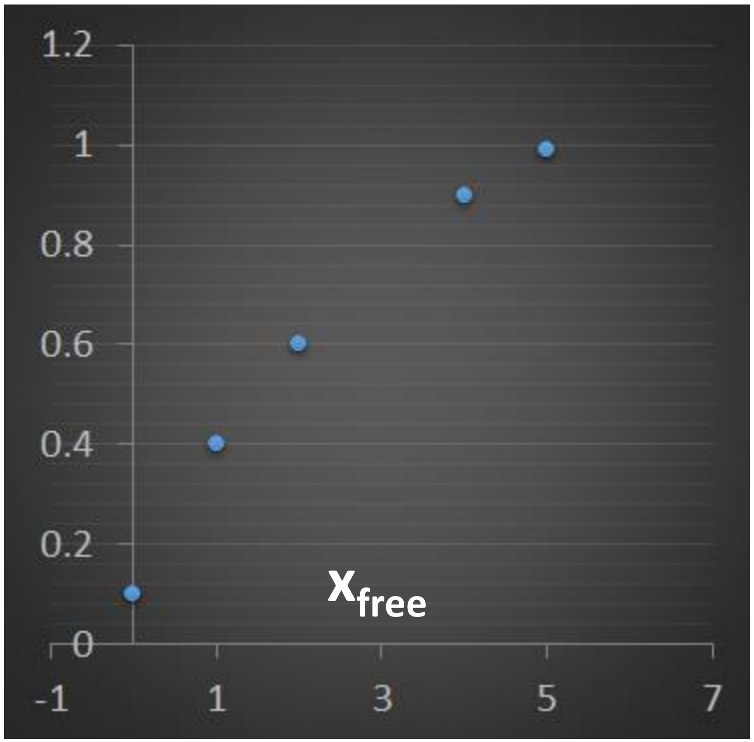
p is always between 0 and 1

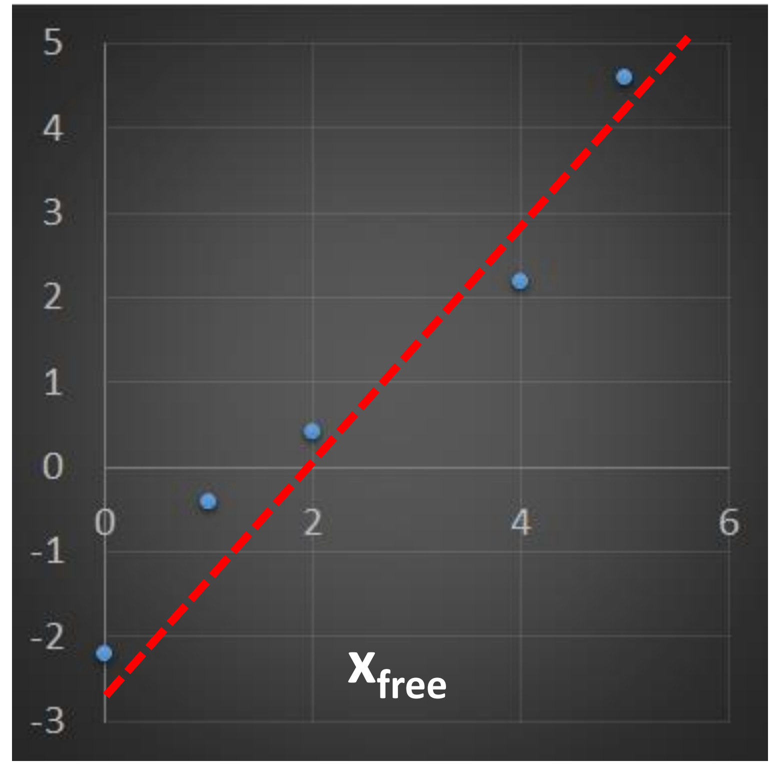
vertical line: logit(p)

$$\text{logit}(p)=\ln\left(\frac{p}{1-p}\right)$$

Logit

$f'(x_{free}) = w'x_{free} + b'$
($f'(x_{free})$ is an estimate of logit(p))





vertical line: Probability to be spam p(yes|x_{free}) (p)

p is always between 0 and 1

vertical line: logit(p)

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$$

Logit

Store w′ and b′

$$x_{free} = 3$$

$$\Rightarrow f'(x_{free}) = w' \times 3 + b' = 1.5$$

$$\Rightarrow \text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = 1.5$$

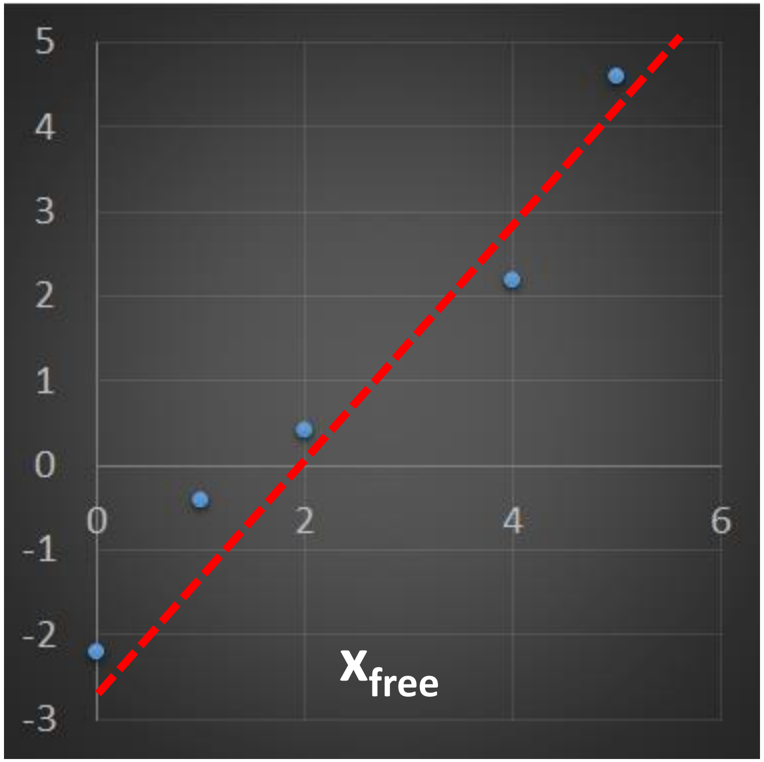
$\Rightarrow p = 0.817$ > 0.5, so "yes"

$$f'(x_{free}) = w' x_{free} + b' > 0$$

$$\Rightarrow \ln\left(\frac{p}{1-p}\right) > 0$$

$\Rightarrow p > 0.5 \Rightarrow$ "yes"

f′(x$_{free}$) = w′x$_{free}$ + b′
(f′(x$_{free}$)) is an estimate of logit(p))



vertical line: logit(p)

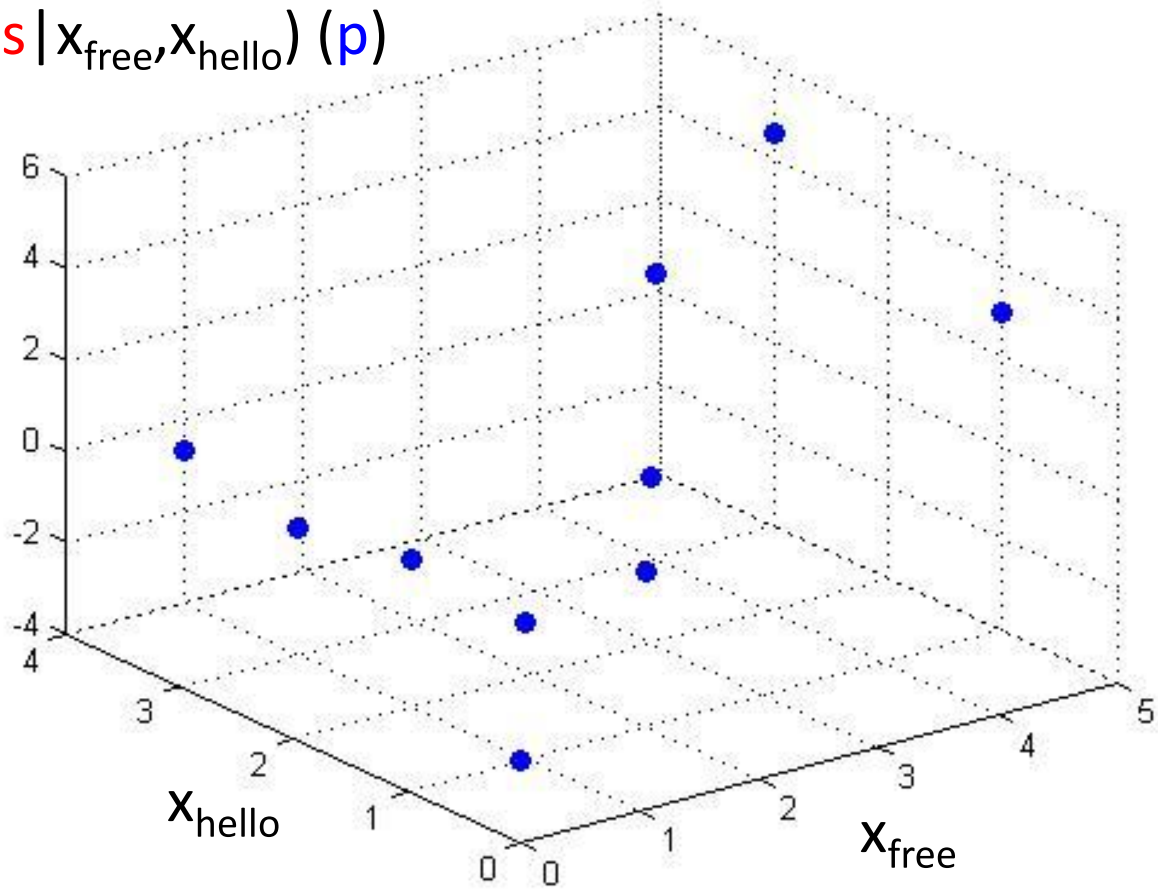
$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$$

Multiple Variables

Consider two words "free" and "hello"

compute $p(\text{yes} \mid x_{free}, x_{hello})$ (p)

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$$



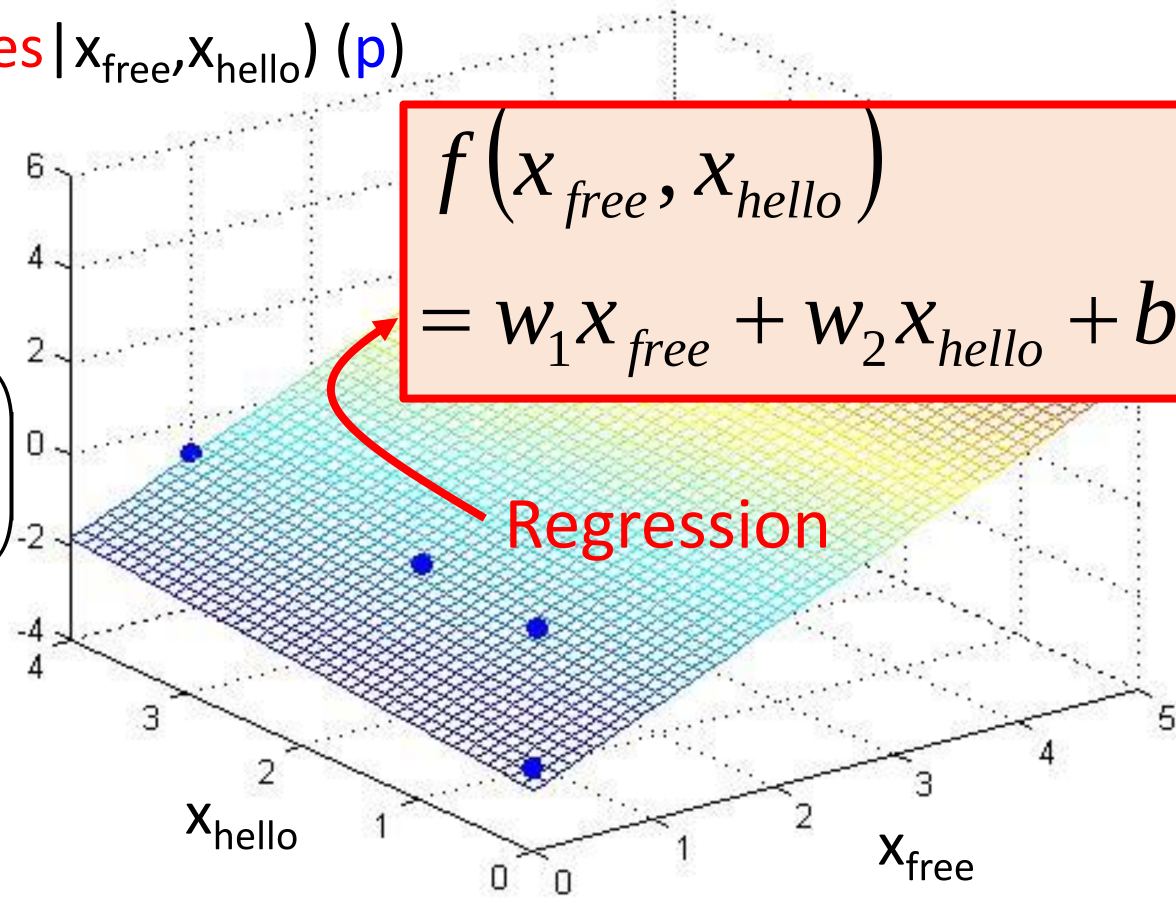
Multiple Variables

Consider two words "free" and "hello"

compute $p(\text{yes}|x_{free}, x_{hello})$ (p)

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$$

$$f(x_{free}, x_{hello}) = w_1 x_{free} + w_2 x_{hello} + b$$



Multiple Variables

- Of course, we can consider all words $\{t_1, t_2, \dots t_N\}$ in a dictionary

$$\mathrm{p} : \mathrm{P}\left(yes \mid x_{t_1}, x_{t_2} \cdots x_{t_N}\right)$$

$$f\left(x_{t_1}, x_{t_2} \cdots x_{t_N}\right) = z = w_1 x_{t_1} + w_2 x_{t_2} + \cdots + w_N x_{t_N} + b$$

$$= \vec{w} \cdot \vec{x} + b$$

z is to approximate logit(p)

$$\vec{x} = \begin{bmatrix} x_{t_1} \\ x_{t_2} \\ \vdots \\ x_{t_N} \end{bmatrix} \quad \vec{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}$$

Logistic Regression

$$z = \vec{w} \cdot \vec{x} + b \xrightarrow{\text{approximate}} \operatorname{logit}(p) = \ln\left(\frac{p}{1-p}\right)$$

$$\mathrm{p} : \mathrm{P}\left(yes \mid x_{t_1}, x_{t_2} \cdots x_{t_N}\right)$$

- If the probability p = 1 or 0, ln(p/1-p) = +infinity or –infinity
- Can not do regression

➢ The probability to be spam p is always 1 or 0.

x 

t_1 appears 3 times
t_2 appears 0 time
…
t_N appears 1 time

$$\mathrm{P}\left(yes \,\middle|\, \begin{array}{l} x_{t_1} = 3 \\ x_{t_2} = 0 \\ \vdots \\ x_{t_N} = 1 \end{array} \right)$$

Logistic Regression

$$z = \vec{w} \cdot \vec{x} + b \longrightarrow \ln\left(\frac{p}{1-p}\right)$$

$$e^{z} = e^{\vec{w}\cdot\vec{x}+b} \longrightarrow \frac{p}{1-p}$$

$$\frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}} \longrightarrow p$$

$$e^{z} = \frac{p}{1-p}$$

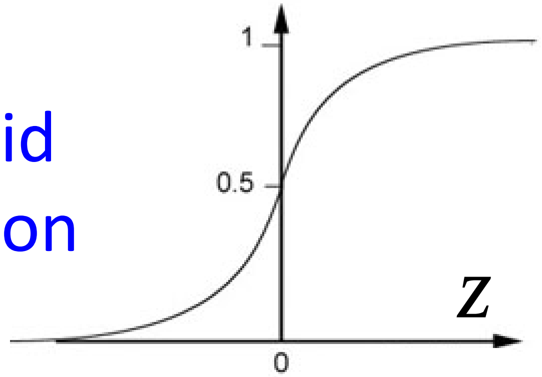
$$e^{z}(1-p) = p$$

$$e^{z} - e^{z}p = p$$

$$e^{z} = (1+e^{z})p$$

$$p = \frac{e^{z}}{1+e^{z}} = \frac{1}{1+e^{-z}}$$

Sigmoid Function



Logistic Regression

$$\frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}} \longrightarrow p$$

x^1

Yes (Spam)

$$\vec{x}^1 = \begin{bmatrix} x^1_{t_1} = 3 \\ x^1_{t_2} = 0 \\ \vdots \\ x^1_{t_N} = 7 \end{bmatrix} \qquad \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}^1+b)}}$$

close to **1**

x^2

No (not Spam)

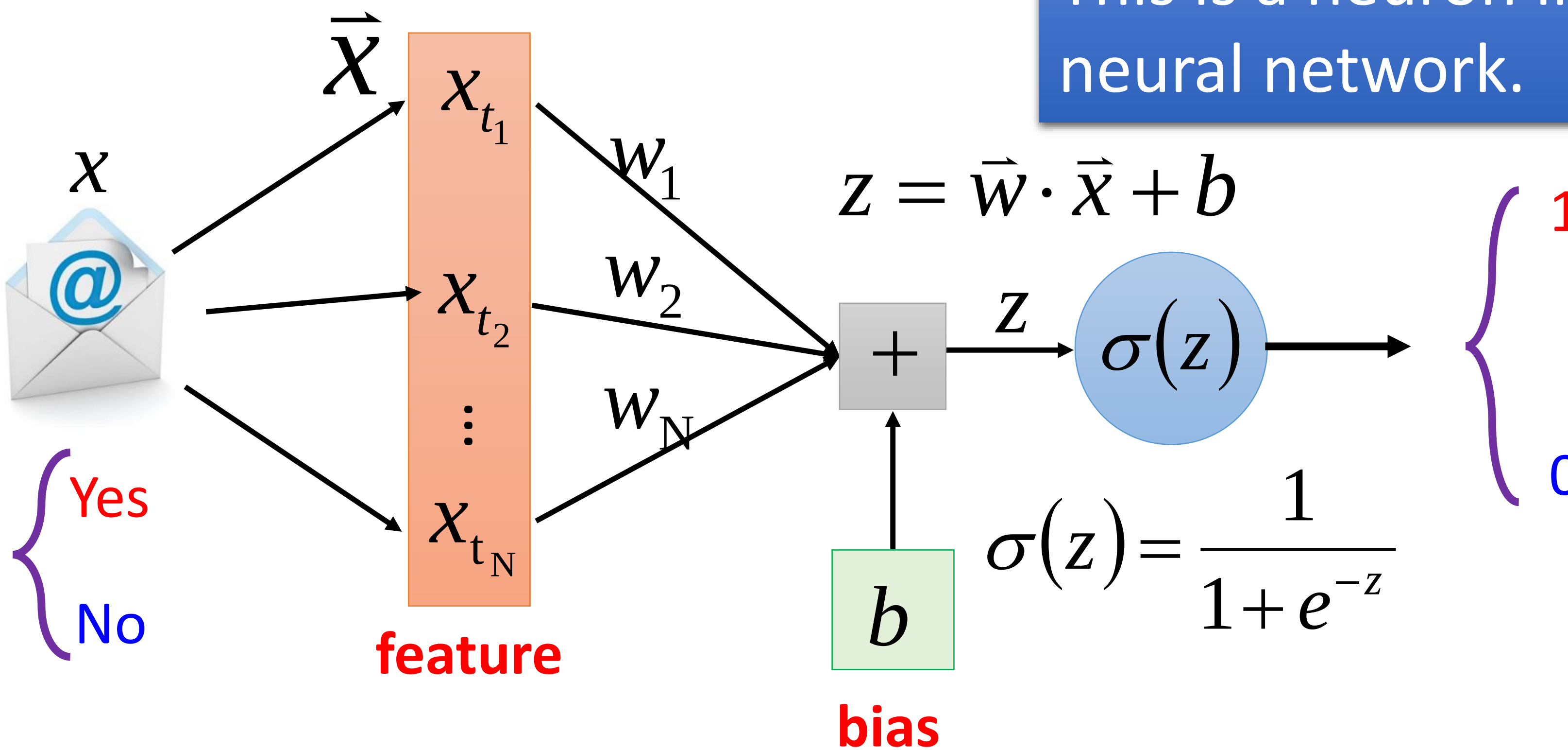
$$\vec{x}^2 = [\vdots] \qquad \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}^2+b)}}$$

close to **0**

Logistic Regression

$$\frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}} \longrightarrow p$$

This is a neuron in neural network.

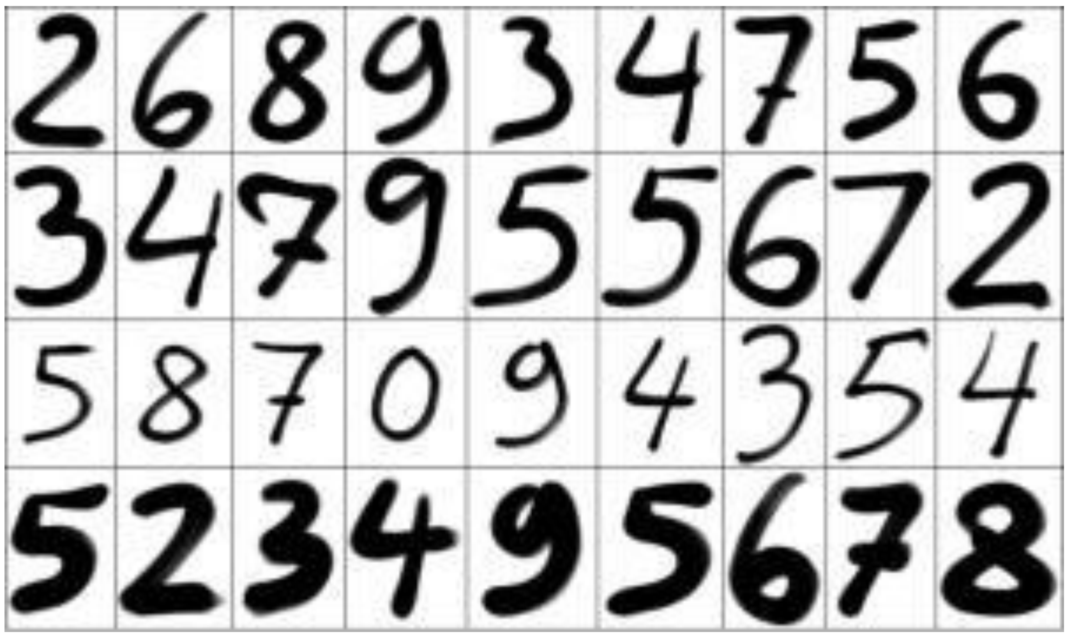


More than saying “yes/no”

Multiclass Classification

More than saying “yes/no”

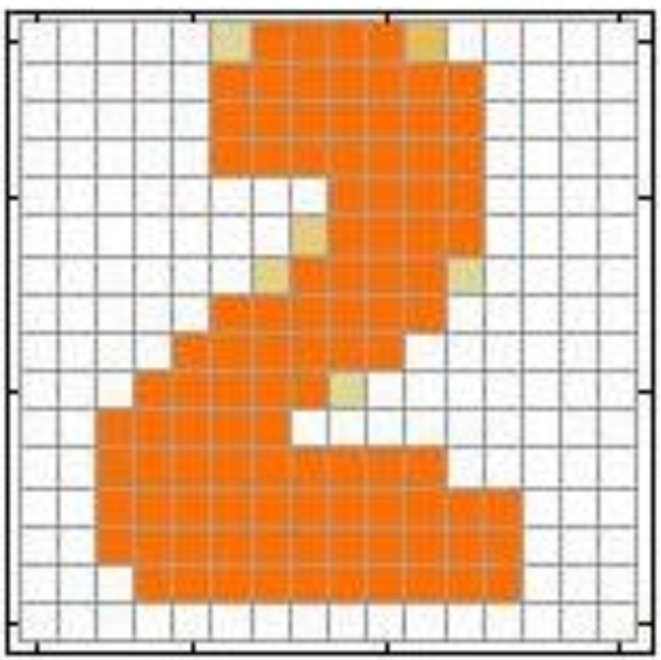
- Handwriting digit classification



This is Multiclass Classification

More than saying “yes/no”

- Handwriting digit classification
  - Simplify the question: whether an image is “2” or not



Describe the characteristics of input object

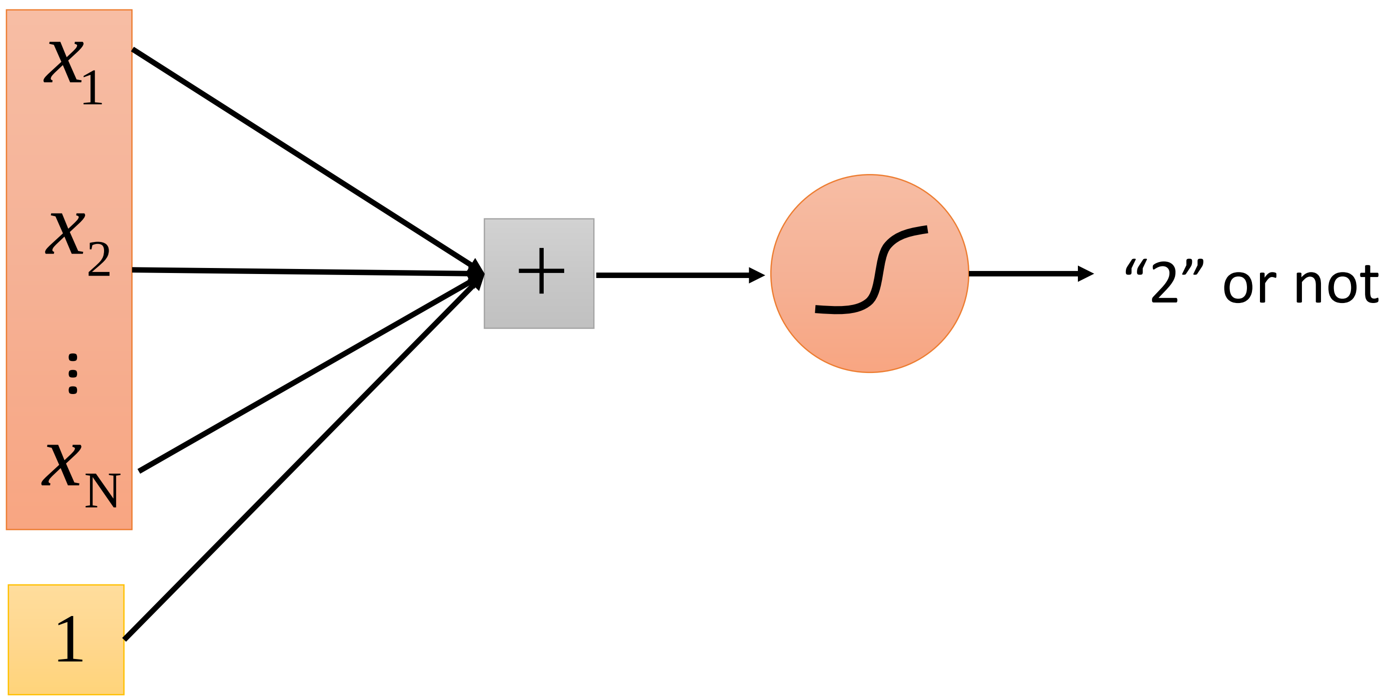
Each pixel corresponds to one dimension in the feature

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

feature of an image

More than saying “yes/no”

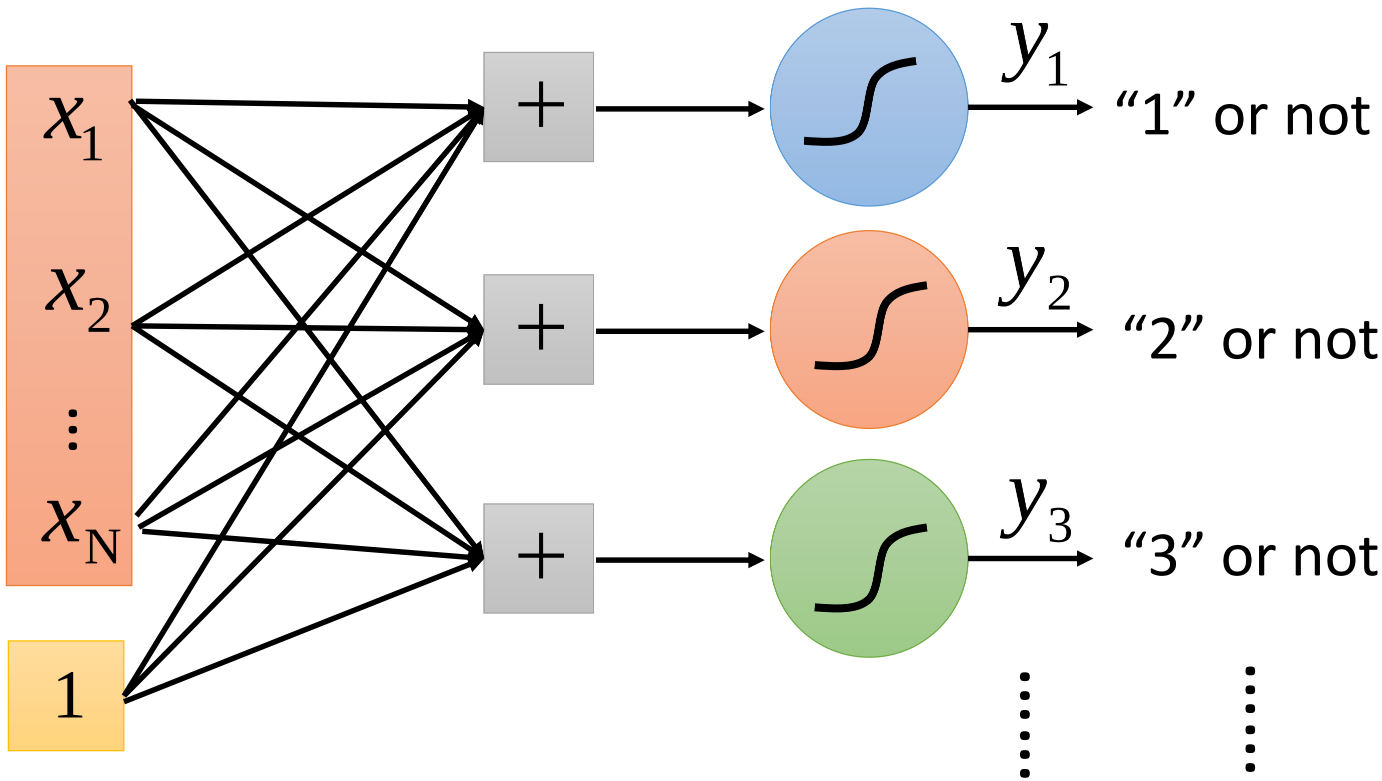
- Handwriting digit classification
  - Simplify the question: whether an image is “2” or not



More than saying "yes/no"

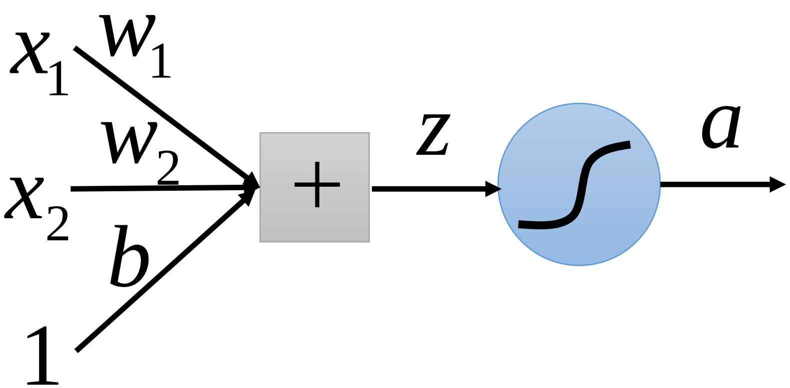
- Handwriting digit classification
  - Binary classification of 1, 2, 3 …

If y_2 is the max, then the image is "2".



This is not good enough …

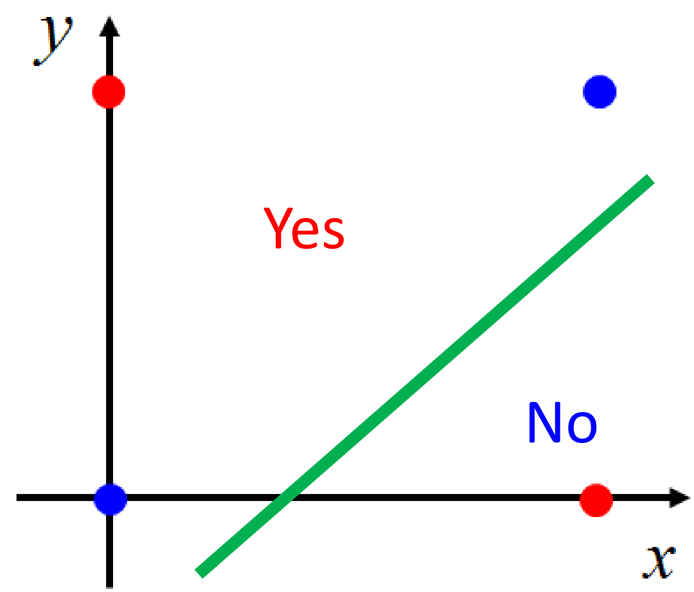
Limitation of Logistic Regression



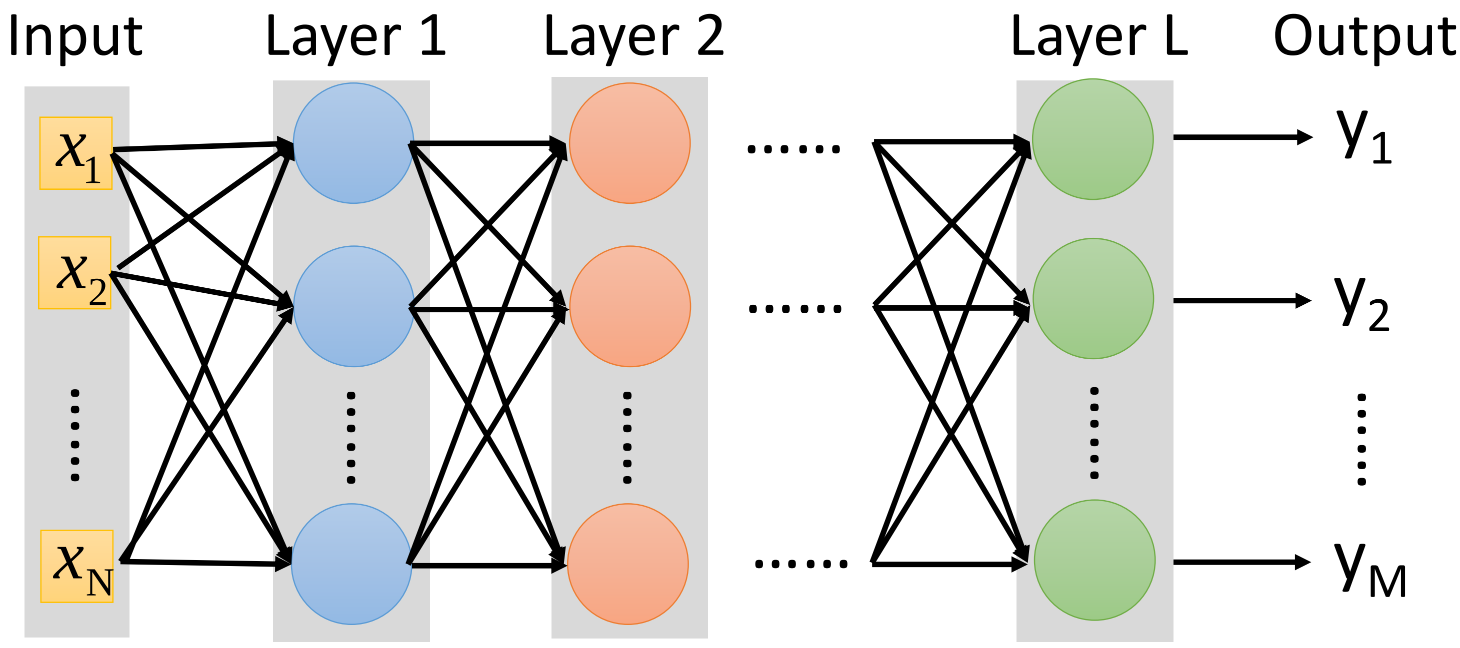
$$\begin{cases} yes & a \geq 0.5 \\ no & a < 0.5 \end{cases} \qquad \begin{cases} yes & z \geq 0 \\ no & z < 0 \end{cases}$$

$$z = w_1 x_1 + w_2 x_2 + b$$

Input		Output
x_1	x_2	
0	0	No
0	1	Yes
1	0	Yes
1	1	No



So we need neural network ……



Deep means many layers

Thank you
for your listening!

Appendix

More reference

- http://www.ccs.neu.edu/home/vip/teach/MLcourse/2_GD_REG_pton_NN/lecture_notes/logistic_regression_loss_function/logistic_regression_loss.pdf
- http://mathgotchas.blogspot.tw/2011/10/why-is-error-function-minimized-in.html
- https://cs.nyu.edu/~yann/talks/lecun-20071207-nonconvex.pdf
- http://www.cs.columbia.edu/~blei/fogm/lectures/glms.pdf
- http://grzegorz.chrupala.me/papers/ml4nlp/linear-classifiers.pdf