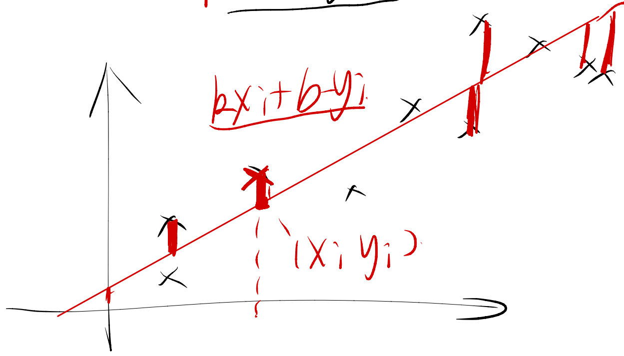
Intro to Neural Network.

Review: Linear Regression.

Dataset: $(x_i, y_i)_{i=1}^{n}$

Model: $y = kx + b$

$f: \mathbb{R} \to \mathbb{R}$



Fit LR model:

$$\text{Error} = \sum_{i=1}^{n} (kx_i + b - y_i)^2$$

k b

$$\frac{\sum x_i}{2} - \frac{x_i^2}{2}$$

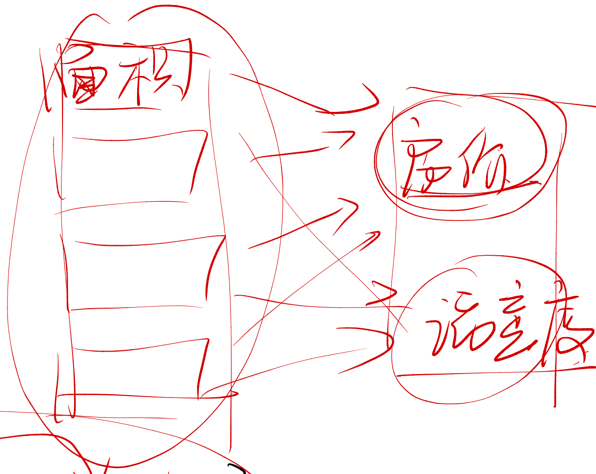
Higher dimension: Linear Regression (eg 房价)

$f: \mathbb{R}^n \to \mathbb{R}^m$

$y = k \cdot x + b$

y: m, x: n, b: m

k: matrix $m \times n$



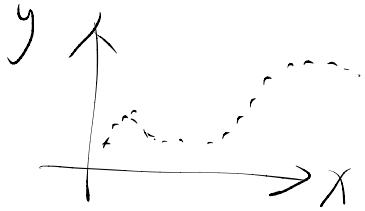
$$\text{Error} = \sum_{i=1}^{n} (\|kx_i + b - y_i\|)^2$$

Cost function

$\{(x_i, y_i)\}_{i=1}^{n}$

$\|(a_1, a_2, a_3)\|_2 = \sqrt{a_1^2 + a_2^2 + a_3^2}$ → L2-norm

Nonlinear model; one example.

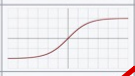
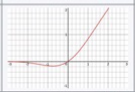


$y = \sigma(kx+b)$

σ : activation function

(nonlinear)

for example

Name	Plot	Function, $f(x)$	Derivative of f, $f'(x)$	Range	Order of continuity	Mono-tonic
Identity		x	1	$(-\infty, \infty)$	C^∞	Yes
Binary step		$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	$\begin{cases} 0 & \text{if } x \neq 0 \\ \text{undefined} & \text{if } x = 0 \end{cases}$	$\{0, 1\}$	C^{-1}	Yes
Logistic, sigmoid, or soft step		$\sigma(x) = \frac{1}{1+e^{-x}}$[1]	$f(x)(1-f(x))$	$(0, 1)$	C^∞	Yes
tanh		$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - f(x)^2$	$(-1, 1)$	C^∞	Yes
Rectified linear unit (ReLU)[11]		$\begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ $= \max\{0, x\} = x\mathbf{1}_{x>0}$	$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ \text{undefined} & \text{if } x = 0 \end{cases}$	$[0, \infty)$	C^0	Yes
Gaussian Error Linear Unit (GELU)[6]		$\frac{1}{2}x\left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$ $= x\Phi(x)$	$\Phi(x) + x\phi(x)$	$(-0.17\ldots, \infty)$	C^∞	No
Softplus[12]		$\ln(1 + e^x)$	$\frac{1}{1+e^{-x}}$	$(0, \infty)$	C^∞	Yes
Exponential linear unit (ELU)[13]		$\begin{cases} \alpha(e^x - 1) & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ with parameter α	$\begin{cases} \alpha e^x & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ 1 & \text{if } x = 0 \text{ and } \alpha = 1 \end{cases}$	$(-\alpha, \infty)$	$\begin{cases} C^1 & \text{if } \alpha = 1 \\ C^0 & \text{otherwise} \end{cases}$	Iff $\alpha \geq$
Square linear		$\begin{cases} x & \text{if } x > 0.0 \end{cases}$	$\begin{cases} 1 & \text{if } x > 0.0 \end{cases}$			

Sigmoid.

More nonlinearity !! Input: x output: y

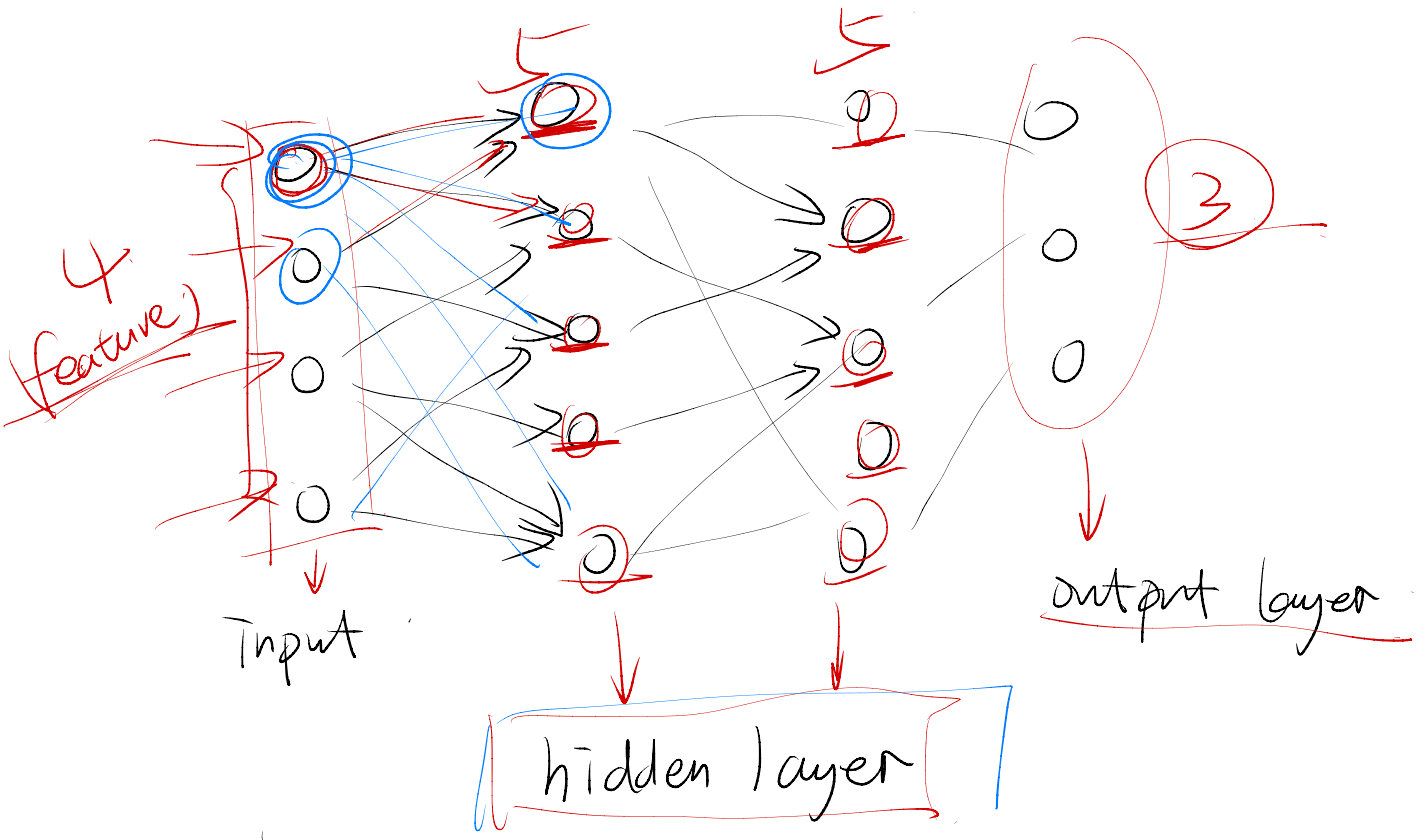
$$y_1 = \sigma(w_1 x + b_1)$$

$$y_2 = \sigma(w_2 y_1 + b_2)$$

$$y = y_3 = \sigma(w_3 y_2 + b_3)$$

w_1 b_1 $\rightarrow$ parameter

A graph of Neural network



parameter

hyper parameter ← 超参数

condition dependent

$N(\mu, \sigma^2)$

input: condition

output: μ, σ^2

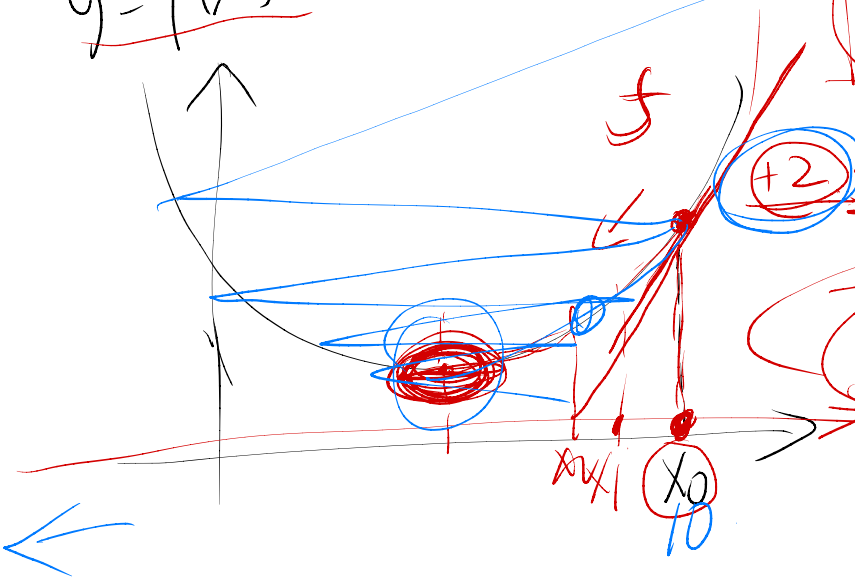
How to optimize (train) a NN?

Gradient descent

How to solve

$$\min_x f(x)$$

$y = f(x)$



Start from x_0

$$x_{i+1} = x_i - s \cdot \nabla f(x_i)$$

s step size

$s > 0$

$s = 100$

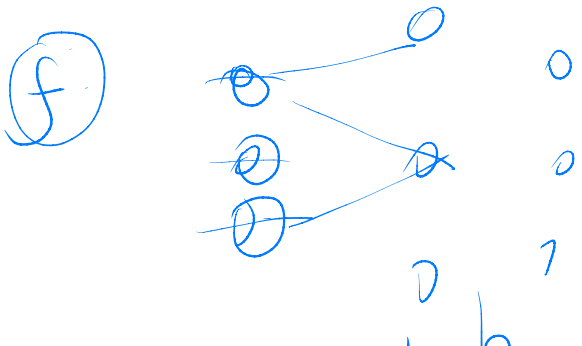
$$x_1 = x_0 - s \cdot \nabla f(x_0)$$

$$x_n = x_1 - s \cdot \nabla f(x_1)$$

$$x_1 = 10 - 100 \cdot 2 = -190$$

learning rate

Chain Rule:



$$\theta = (w_1\ w_2\ w_3 \cdots\ b_1\ b_2\ b_3)$$

$$C = \sum_{i=1}^{n} \left(f_\theta(x_i) - y_i \right)^2$$

$\nabla_\theta C$

$$\nabla_\theta C = \sum_{i=1}^{n} 2\left(f_\theta(x_i) - y_i\right)$$

- $\nabla_\theta\left(f_\theta(x_i) - y_i\right)$

$$= \nabla_\theta f_\theta(x_i)$$

ReLU

$\sigma(z) =$

$$\nabla_{W_2}\sigma\left(W_2 \cdot \sigma(W_1 x + b_1) + b_1\right)$$

$$= \nabla_{W_2}\sigma(z) \qquad z = W_2 \cdot \sigma(W_1 x + b_1) + b_1$$

- $\nabla_{W_2}\left(W_2\sigma(W_1 x + b_1) + b_1\right)$

$$= \sigma(W_1 x + b_1)$$

How to Implement ?!

PyTorch → TensorFlow → pytho

```
model = Sequential( )
model.add_layer("dense", __)
model. _____
model. _____
```

cost = [ ]

model.initialize( )

for i in 100 :

model.descent( )

GAN / VAE / NF

Learn C D with high dim

w/ $Y \mid x=x_1$ $Y \mid x=x_2$ quality

x_1 x_2 Regularization ~~with~~ NF

wrong

$(x_i, y_i)_{i=1}^{n}$

2

0 1


KDE