Supervised learning

Four ingradients of supervised learning

- Input feature x
- Output target y
- Function hypothesis f
- Loss function

$\Big\} \; y = f(x)$

ex: House price prediction

size $= x_1$

age $= x_2$ (Input)

$\Big\} \rightarrow$ price $= y$ (Output)

Data : $\{(x_1^{\hat{i}}, x_2^{\hat{i}}, y^{\hat{i}})\}_{\hat{i}=1}^{N}$

Goal: To find h s.t. $y^{\hat{i}} = h(x_1^{\hat{i}}, x_2^{\hat{i}})$, $\forall \hat{i}$.

Assume $h(x_1, x_2) = b + w_1 x_1 + w_2 x_2$, $b, w_1, w_2 \in \mathbb{R}$.

$$\Rightarrow h(x_1, x_2) = \begin{pmatrix} 1 & x_1 & x_2 \end{pmatrix} \begin{pmatrix} b \\ w_1 \\ w_2 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{pmatrix} = \begin{pmatrix} 1 & x_1^1 & x_2^1 \\ 1 & x_1^2 & x_2^2 \\ & \vdots & \\ 1 & x_1^N & x_2^N \end{pmatrix} \begin{pmatrix} b \\ w_1 \\ w_2 \end{pmatrix}$$

Y X θ

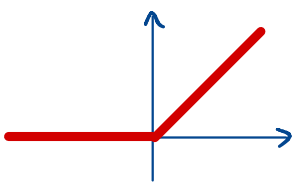
$$\min \|Y - X\theta\|_2^2$$

$$\Rightarrow \theta^* = (X^T X)^{-1} X^T Y$$

mean squared error loss (MSE) : $\frac{1}{N} \sum_{i=1}^{N} |y^i - h(x_1^i, x_2^i)|^2$

Relu activation function

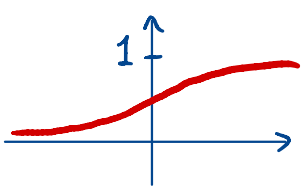
$$\sigma(x) = \begin{cases} x, & x > 0 \\ 0, & x \le 0 \end{cases}$$



hypothesis 2 : $h_2(x_1, x_2) = \sigma(b + w_1 x_1 + w_2 x_2)$

Sigmoid activation function

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



hypothesis 3 : $h_3(x_1, x_2) = w_3 \sigma(b + w_1 x_1 + w_2 x_2)$

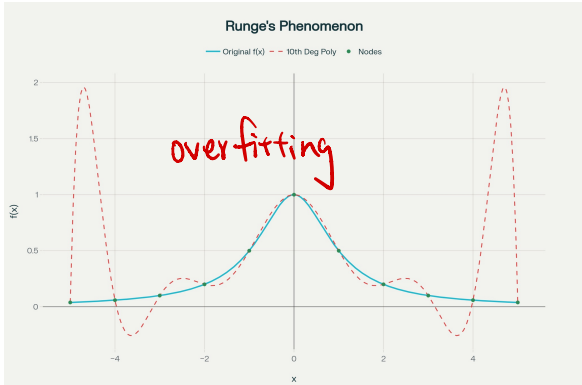
Training / Validation / Test set

make sure training loss $\approx$ validation loss

then we except test loss to be in similar order

Runge phenomena

$$f(x) = \frac{1}{1+x^2}, \quad -5 \le x \le 5.$$



Input x

Output y

Hypothesis polynomial: $h(x) = a_0 + a_1 x + \cdots + a_n x^n$

Supervised learning

hypothesis: $h(x;\theta) = h_\theta(x) = h(x)$

$$\text{Loss}(\theta) = \frac{1}{N} \sum_{i=1}^{N} |y^i - h(x^i;\theta)|^2$$

$\theta \in \mathbb{R}^M$, $\text{Loss}: \mathbb{R}^M \to \mathbb{R}$

$$\theta^* = \underset{\theta}{\text{argmin}}\ \text{Loss}$$

Gradient descent method (GD)

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

$\nabla f(x_0)$ is the stepest ascent direction at x_0.

$-\nabla f(x_0)$ is the stepest descent direction at x_0.

GD algorithm

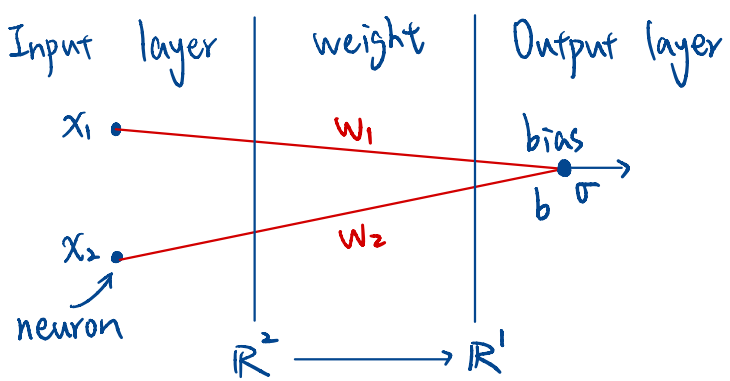
$$\theta^{n+1} = \theta^n - \alpha \nabla_\theta \text{Loss}, \quad \alpha > 0 : \text{learning rate}$$

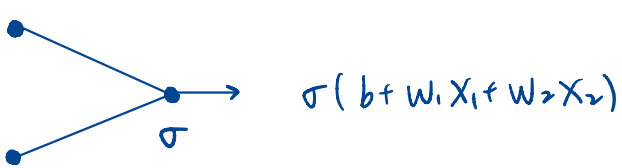
$$\theta^{n+1} = \theta^n + \frac{2\alpha}{m} \sum_{\hat{\imath}=1}^{m} (y^{\hat{\imath}} - h(x^{\hat{\imath}}; \theta^n)) \cdot \nabla_\theta h(x^{\hat{\imath}}; \theta^n)$$

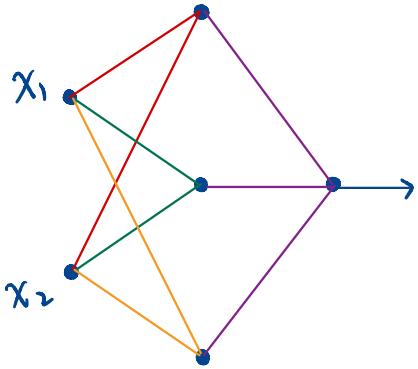
- $m = N$ Batch gradient descent
- $m = 1$ stochastic gradient descent (SGD)
- $m < N$ mini-batch gradient descent

epoch : one full pass through the entire data set

$$h(x_1, x_2) = \sigma(b + w_1 x_1 + w_2 x_2)$$







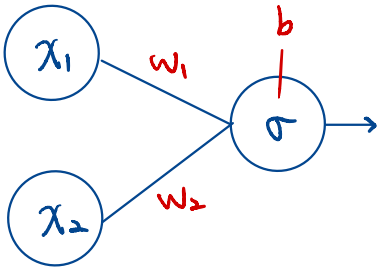
$\rightarrow$: $\sigma(w_{11}x_1 + w_{21}x_2 + b_1)$

$\rightarrow$: $\sigma(w_{21}x_1 + w_{22}x_2 + b_2)$

$\rightarrow$: $\sigma(w_{31}x_1 + w_{32}x_2 + b_3)$

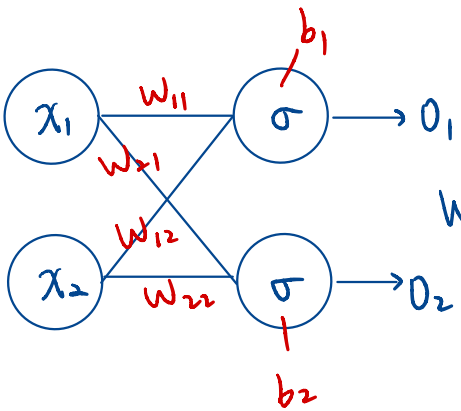
$$\sigma\left(W^{[3]}\sigma\left(\underset{\parallel}{W^{[2]}}\vec{x} + \underset{\parallel}{b^{[2]}}\right) + b^{[3]}\right)$$

$$\begin{pmatrix} w_{11} & w_{21} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix} \qquad \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$



affine linear transformation

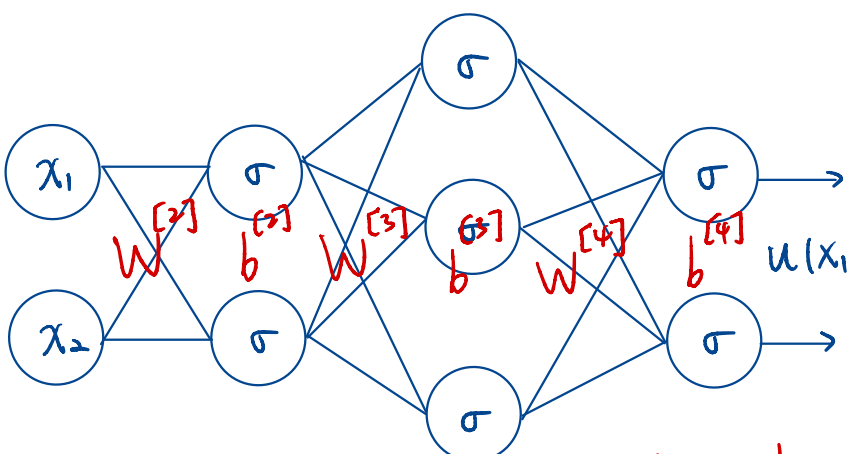
$$\sigma(w_1 x_1 + w_2 x_2 + b) = \sigma(Wx + b)$$



$$W = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix}$$

$M_{2\times 2}$ $\mathbb{R}^2$ $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

$$\sigma(Wx + b) = \begin{pmatrix} \sigma(O_1) \\ \sigma(O_2) \end{pmatrix}$$



$$u(x) = \sigma\left(W^{[4]}\sigma\left(W^{[3]}\sigma\left(W^{[2]}x + b^{[2]}\right) + b^{[3]}\right) + b^{[4]}\right)$$

Supervised learning : data $\{x^i, y^i\}_{i=1}^N$

Loss : $\frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\| y^i - u(x^i)\|_2^2$

SGD : $C(x) = \frac{1}{2}\| y - u(x)\|_2^2$

$$\theta^{n+1} = \theta^n - \alpha \nabla_\theta C(x^i)$$
$$= \theta^n - \alpha (y^i - u(x^i)) \cdot (-\nabla_\theta u(x^i))$$
$$= \theta^n + \alpha (y^i - u(x^i)) \nabla_\theta u(x^i)$$

$$\nabla_\theta u = \left(\frac{\partial u}{\partial w_1}, \cdots, \frac{\partial u}{\partial w_j}, \frac{\partial u}{\partial b_1}, \cdots, \frac{\partial u}{\partial b_k} \right)$$

Backpropogation

$$u(x) = \sigma\left(W^{[4]} \sigma\left(W^{[3]} \sigma\left(W^{[2]} x + b^{[2]} \right) + b^{[3]} \right) + b^{[4]} \right)$$

z_2 z_3 z_4

$$\delta_i^{[4]} = \frac{\partial C}{\partial z_i^{[4]}} = \frac{\partial C}{\partial u_i} \frac{\partial u_i}{\partial z_i^{[4]}} = (u(x)_i - y_i)\, \sigma'(z_i^{[4]})$$

Hadamard product (componentive product)

$(\vec{x} \circ \vec{y})_i = x_i y_i$

$\Rightarrow \delta^{[4]} = \sigma'(z^{[4]}) \circ (u(x) - y)$

$$\delta^{[3]} = \frac{\partial C}{\partial z^{[3]}} = \frac{\partial C}{\partial z^{[4]}} \frac{\partial z^{[4]}}{\partial z^{[3]}} = \underbrace{\delta^{[4]}}_{2\times 1} \cdot \underbrace{W^{[4]}}_{2\times 3} \underbrace{\sigma'(z^{[3]})}_{3\times 1}$$

$$= \sigma'(z^{[3]}) \circ \left((W^{[4]})^T \delta^{[4]} \right)$$

$$\delta^{[2]} = \frac{\partial C}{\partial z^{[2]}} = \frac{\partial C}{\partial z^{[3]}} \cdot \frac{\partial z^{[3]}}{\partial z^{[2]}} = \sigma'(z^{[2]}) \circ \left((W^{[3]})^T \delta^{[3]} \right)$$

$$\frac{\partial C}{\partial W^{[4]}} = \frac{\partial C}{\partial z^{[4]}} \cdot \frac{\partial z^{[4]}}{\partial W^{[4]}} = \delta^{[4]} \sigma(z^{[3]})^T$$

$$\frac{\partial C}{\partial b^{[4]}} = \frac{\partial C}{\partial z^{[4]}} \cdot \frac{\partial z^{[4]}}{\partial b^{[4]}} = \delta^{[4]}$$

$$\frac{\partial C}{\partial W^{[3]}} = \frac{\partial C}{\partial z^{[3]}} \cdot \frac{\partial z^{[3]}}{\partial W^{[3]}} = \delta^{[3]} \sigma(z^{[2]})^T$$

$$\frac{\partial C}{\partial b^{[3]}} = \frac{\partial C}{\partial z^{[3]}} \cdot \frac{\partial z^{[3]}}{\partial b^{[3]}} = \delta^{[3]}$$