Deep Learning

Dr. Mehran Safayani

safayani@iut.ac.ir

safayani.iut.ac.ir

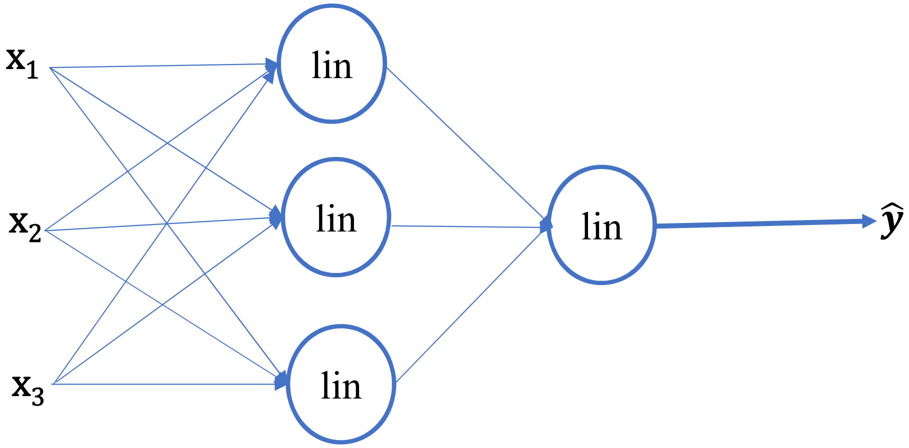
https://www.aparat.com/mehran.safayani

https://github.com/safayani/deep_learning_course

Department of Electrical and computer engineering, Isfahan university of technology, Isfahan, Iran

Activation function

Non-linear activation function



- چرا به توابع فعال‌ساز غیرخطی نیاز داریم؟

- با فرض تابع فعال‌ساز خطی

- $z^{[1]} = w^{[1]}x + b^{[1]}$
- $a^{[1]} = g\left(z^{[1]}\right) = z^{[1]}$
- $z^{[2]} = w^{[2]}a^{[1]} + b^{[2]}$
- $a^{[2]} = g(z^{2}) = z^{[2]}$

$$a^{[1]} = z^{[1]} = w^{[1]}x + b^{[1]}$$

$$a^{[2]} = z^{[2]} = w^{[2]}a^{[1]} + b^{[2]}$$

$$= w^{[2]}(w^{[1]}x + b^{[1]}) + b^{[2]}$$

$$= \underbrace{w^{[2]}w^{[1]}}_{w'}x + \underbrace{w^{[2]}b^{[1]} + b^{[2]}}_{b'}$$

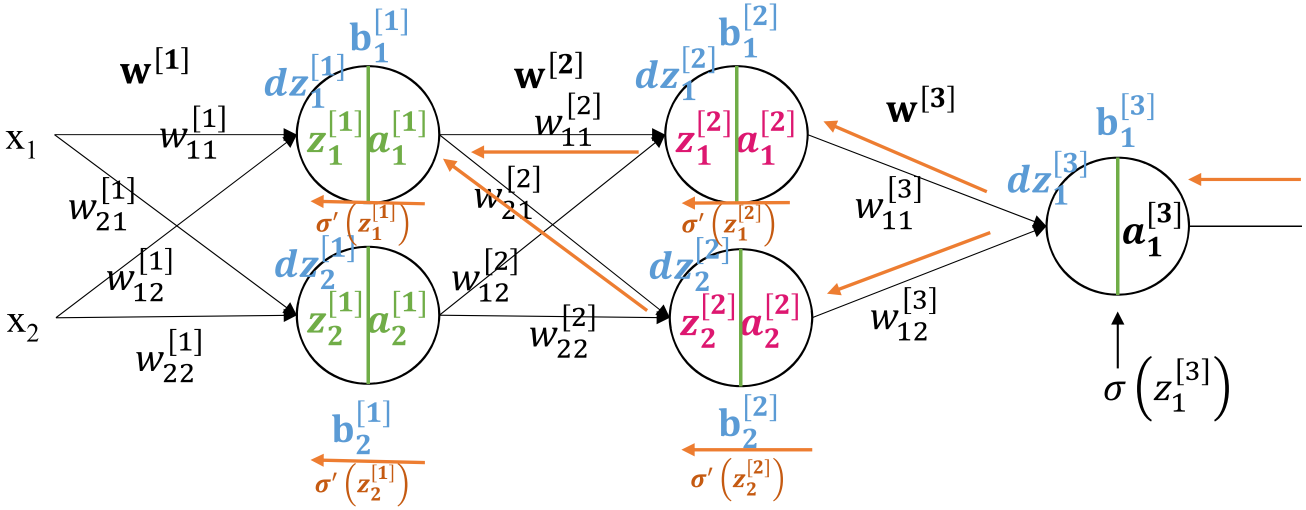
$$= w'x + b'$$

Derivatives of activation functions

- Sigmoid:
- $g(z) = \frac{1}{1+e^{-z}} \qquad g'(z) = \frac{dg(z)}{dz} = \frac{1}{1+e^{-z}}\left(1 - \frac{1}{1+e^{-z}}\right) = g(z)(1 - g(z))$
- Tanh:
- $g(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}} \qquad g'(z) = 1 - \tanh^2(z) = 1 - g^2(z)$
- Relu:
- $g(z) = \max(0, z) = \begin{cases} z & z \geq 0 \\ 0 & z < 0 \end{cases} \qquad g'(z) = \begin{cases} 1 & if\ z \geq 0 \\ 0 & if\ z < 0 \end{cases}$
- Leaky Relu:
- $g(z) = \max(z, 0.01z) = \begin{cases} z & z \geq 0 \\ 0.01z & z < 0 \end{cases} \qquad g'(z) = \begin{cases} 1 & if\ z \geq 0 \\ 0.01 & if\ z < 0 \end{cases}$

Back propagation(Bp)

- پس‌انتشار



$\mathbf{w}_{\mathbf{ij}}^{[l]}$

$\dfrac{dL}{dw_{ij}^{l}}$ for $l = 1 \dots L$

for each i,j

$$\mathbf{w}_{\mathbf{ij}}^{[l]} = \mathbf{w}_{\mathbf{ij}}^{[l]} - \alpha \frac{dL}{dw_{ij}^{l}}$$

- $dw_{11}^{[2]} = \dfrac{L\left(\dots,w_{11}^{[2]}+\varepsilon,\dots\right)-L\left(\dots,w_{11}^{[2]}-\varepsilon,\dots\right)}{2\varepsilon}$

Loss Cross Entropy

- $\frac{dL}{dw_{11}^{[3]}} = \underbrace{\frac{dL}{da_1^{[3]}} \times \frac{da_1^{[3]}}{dz_1^{[3]}}}_{\frac{dL}{dz_1^{[3]}} = dz_1^{[3]}} \times \frac{dz_1^{[3]}}{dw_{11}^{[3]}} = \left(a_1^{[3]} - y\right) a_1^{[2]}$

➢ $z_1^{[3]} = a_1^{[2]} w_{11}^{[3]} + a_2^{[2]} w_{12}^{[3]} + b_1^{[3]}$

- $\frac{dL}{db_1^{[3]}} = \frac{dL}{dz_1^{[3]}} \times \underbrace{\frac{dz_1^{[3]}}{db_1^{[3]}}}_{1} = dz_1^{[3]} \times 1 = dz_1^{[3]}$

- $dz_1^{[2]} = \frac{dL}{dz_1^{[2]}} = \underbrace{\frac{dL}{dz_1^{[3]}}}_{dz_1^{[3]}} \times \underbrace{\frac{dz_1^{[3]}}{da_1^{[2]}}}_{w_{11}^{[3]}} \times \underbrace{\frac{da_1^{[2]}}{dz_1^{[2]}}}_{\sigma'\left(z_1^{[2]}\right)}$ $a_1^{[2]} = \sigma\left(z_1^{[2]}\right)$ $\frac{da_1^{[2]}}{dz_1^{[2]}} = \sigma'\left(z_1^{[2]}\right)$

- $dz_1^{[2]} = dz_1^{[3]} \times w_{11}^{[3]} \times \sigma'\left(z_1^{[2]}\right)$

Loss Cross Entropy

- $dz_2^{[2]} = dz_1^{[3]} \times w_{12}^{[3]} \times \sigma' \left(z_2^{[2]} \right)$
- $dw_{11}^{[2]} = dz_1^{[2]} \times a_1^{[1]}$ $\qquad db_1^{[2]} = dz_1^{[2]}$
- $dw_{12}^{[2]} = dz_1^{[2]} \times a_2^{[1]}$ $\qquad db_2^{[2]} = dz_2^{[2]}$
- $dz_1^{[1]} = dz_1^{[2]} w_{11}^{[2]} + dz_2^{[2]} w_{21}^{[2]}$

Gradient descent

Gradient descent

- Parameters: $\underbrace{w^{[1]}}_{(n^{[1]},n^{[0]})}$, $\underbrace{b^{[1]}}_{(n^{[1]},1)}$, $\underbrace{w^{[2]}}_{(n^{[2]},n^{[1]})}$, $\underbrace{b^{[2]}}_{(n^{[2]},1)}$
- $w^{[1]}, b^{[1]}, w^{[2]}, b^{[2]}$ ← random initialization
- Repeat{

```
    →Forward propagation to compute  $\hat{y}^{(\mathrm{i})}$  i=1, …, m
     Backward prop
     → $dw^{[1]} = \frac{dJ}{dw^{[1]}}, db^{[1]}, dw^{[2]}, db^{[2]}$
     Update $\begin{cases} w^{[1]} = w^{[1]} - \alpha\, dw^{[1]} \\ w^{[2]} = w^{[2]} - \alpha\, dw^{[2]} \\ b^{[1]} = b^{[1]} - \alpha\, db^{[1]} \\ b^{[2]} = b^{[2]} - \alpha\, db^{[2]} \end{cases}$
        }Until Convergence
```

Gradient descent

- *Forward propagation:*

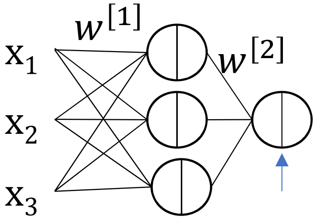
$$z^{[1]} = w^{[1]}X + b^{[1]}$$

$$A^{[1]} = g^{[1]}(z^{[1]})$$

$$z^{[2]} = w^{[2]}A^{[1]} + b^{[2]}$$

$$A^{[2]} = g^{[2]}(z^{[2]})$$

- *Back prop. :*



$$dz^{[2]} = A^{[2]}{}_{1\times m} - Y_{1\times m}$$

$$\underbrace{dw^{[2]}}_{1\times \mathrm{n}^{[1]}} = \frac{1}{m}\underbrace{dz^{[2]}}_{1\times m}\ \underbrace{A^{[1]T}}_{m\times n^{[1]}}$$

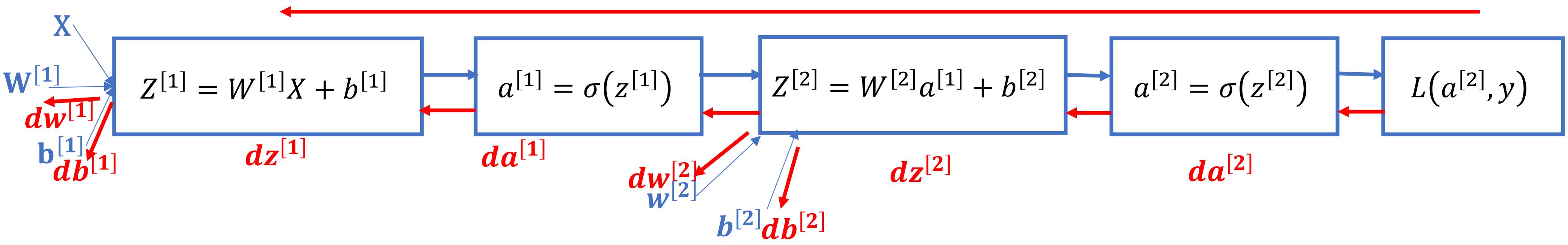
$$\underbrace{db^{[2]}}_{1\times 1} = \frac{1}{m} np \cdot sum\left(\underbrace{dz^{[2]}}_{1\times m}, axis = 1, keepdims = True\right) (1{,}1)$$

Gradient descent

- $\underbrace{dz^{[1]}}_{n^{[1]}\times m} = \underbrace{w^{[2]T}}_{n^{[1]}\times 1}\,\underbrace{dz^{[2]}}_{1\times m} * \underbrace{g^{[1]'}(\underbrace{z^{[1]}}_{n^{[1]}\times m})}_{n^{[1]}\times m}$

  Element wise

- $\underbrace{dw^{[1]}}_{n^{[1]}\times n^{[0]}} = \frac{1}{m}\,\underbrace{dz^{[1]}}_{n^{[1]}\times m}\;\underbrace{X^{T}}_{m\times n^{[0]}}$
- $\underbrace{db^{[1]}}_{n^{[1]}\times 1} = \frac{1}{m} np \cdot sum\left(\underbrace{dz^{[1]}}_{n^{[1]}\times m}, axis = 1, keepdims = True\right)$ $(n^{[1]}, 1)$



Gradient descent

- $dz^{[1]} = w^{[2]T}\, dz^{[2]} * g^{[1]'}(z^{[1]})$ m=1
- $dz^{[2]} = a^{[2]} - y$
- $dw^{[1]} = dz^{[1]} X^{T}$
- $dw^{[2]} = dz^{[2]} {a^{[1]}}^{T}$
- $db^{[1]} = dz^{[1]}$
- $db^{[2]} = dz^{[2]}$

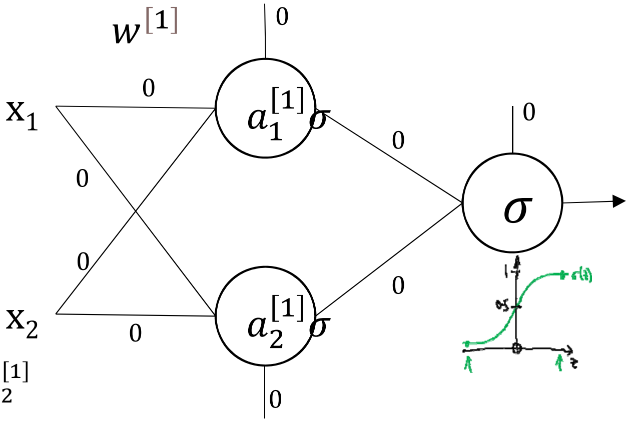
Summary of gradient descent

- $dz^{[2]} = A^{[2]} - Y$
- $dw^{[2]} = \frac{1}{m} dz^{[2]} A^{[1]T}$
- $db^{[2]} = \frac{1}{m} np \cdot sum\left(dz^{[2]}, axis = 1, keepdims = True\right)$
- $dz^{[1]} = w^{[2]T} dz^{[2]} * g^{[1]'}(z^{[1]})$
- $dw^{[1]} = \frac{1}{m} dz^{[1]} X^T$
- $db^{[1]} = \frac{1}{m} np \cdot sum\left(dz^{[1]}, axis = 1, keepdims = True\right)$

Random Initialization

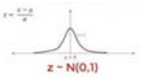
- **Restricted Boltzmann Machine(RBM):** A **restricted Boltzmann machine** is a generative stochastic artificial neural network that can learn a probability distribution over its set of inputs.

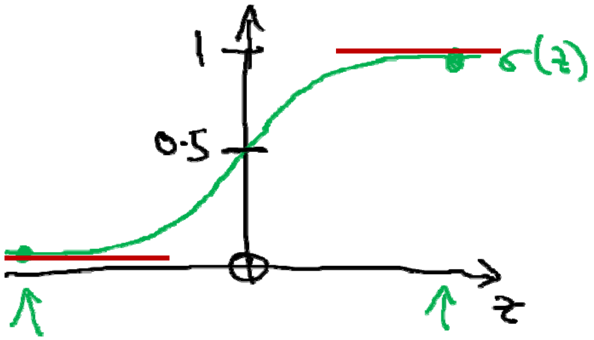
- اگر همه وزن‌های اولیه و بایاس‌ها صفر باشند:



- $a_1^{[1]} = a_2^{[1]}$
- $w^{[1]} = np.random.randn((2,2)) * 0.01$
- $b^{[1]} = np.zeros((2,1))$
- $w^{[2]} = np.random.randn((1,2)) * 0.01$
- $b^{[2]} = 0$

$N(0,1)$





Vanishing gradient