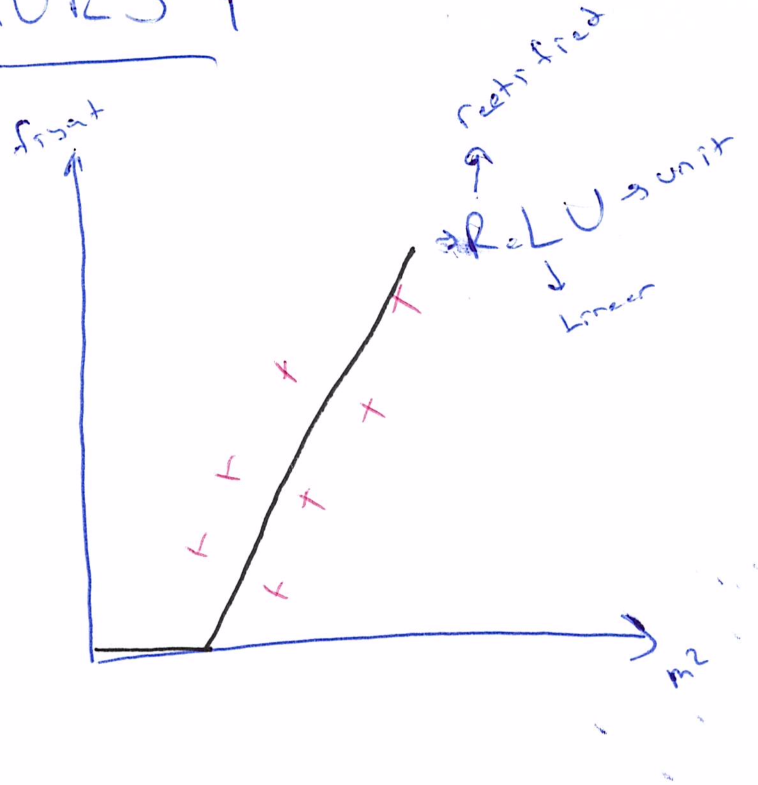
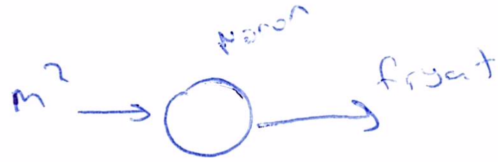
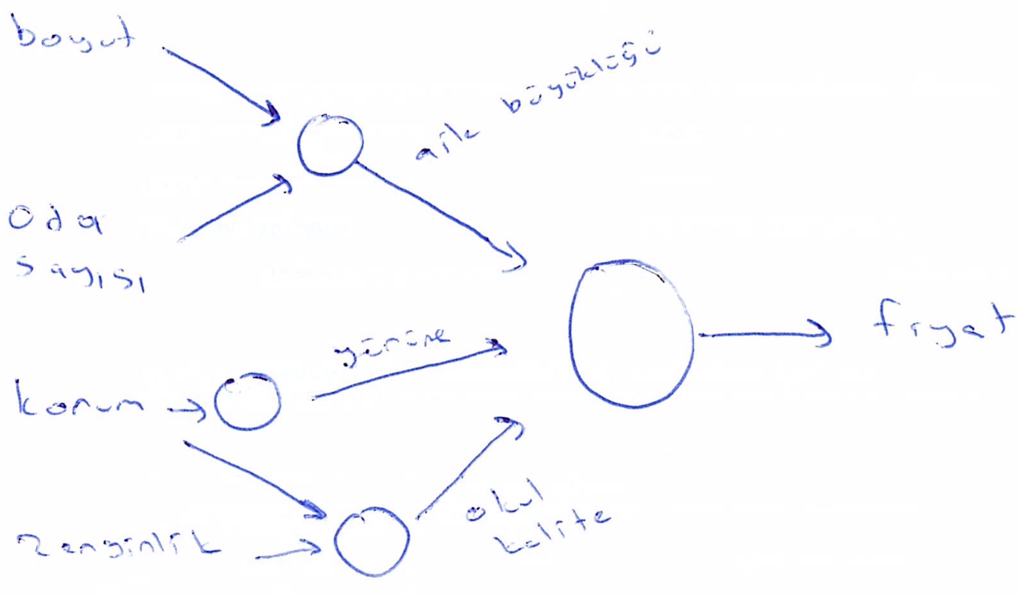
KURS 1

HAFTA 1





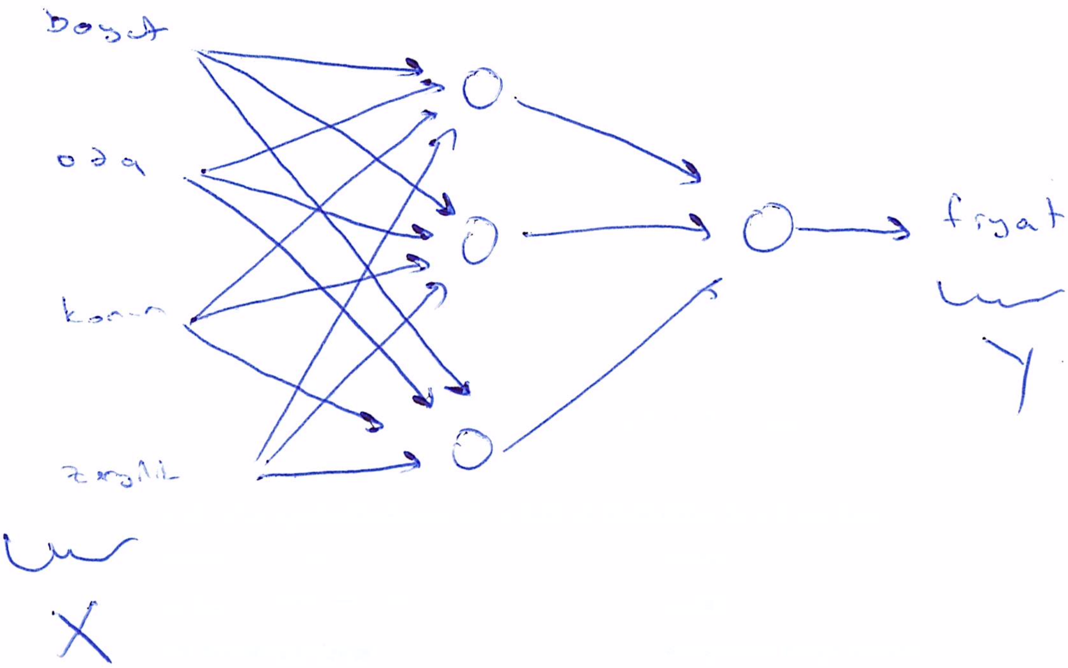


a	b	c	d	e	f
Banyo sayısı	boyut	oda	konum	zeng	fiyat

$$f = ax + by + cz + dg + e.h$$

$$\underset{\substack{\shortparallel \\ 0.001}}{x}, y, z, g, h = ?$$

$$f(a, b, c, d, e)$$



SNN

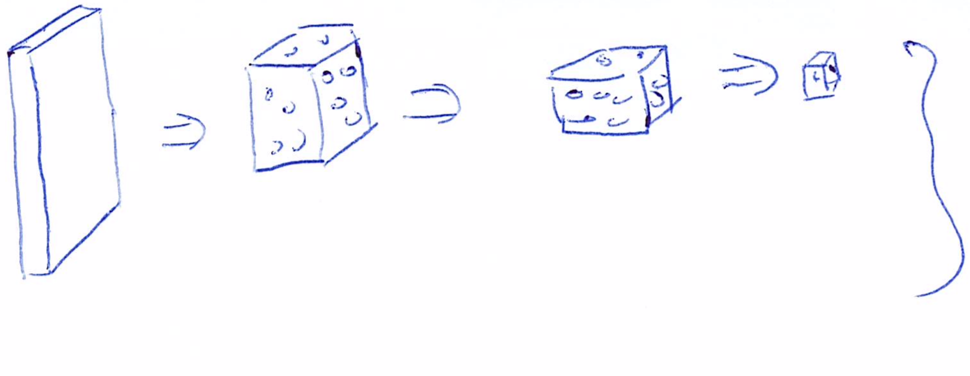
Standart
Neural
Network

SNN
- ★ fiyat tahmini
- ★ reklam tıklama öngörme

CNN ★ resim sınıflandırma

RNN
- ★ sesi yazıya dönüştürme
- ★ tercüme

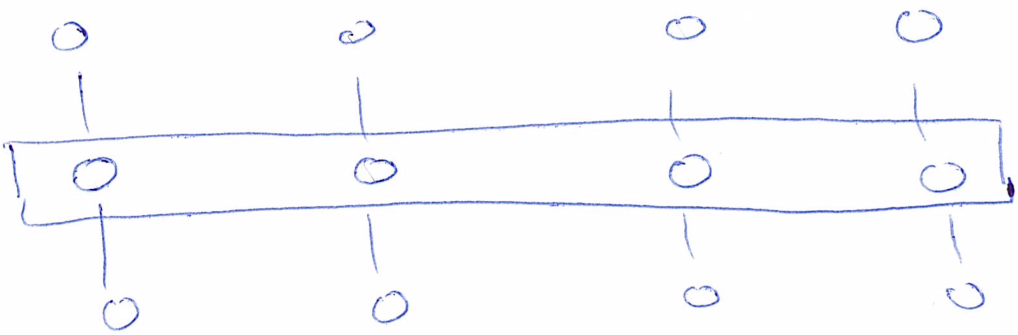
karma ★ otonom sürüş



CNN

görüntü

Convolutional
Neural
Network



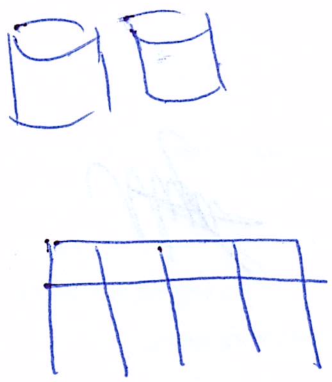
RNN

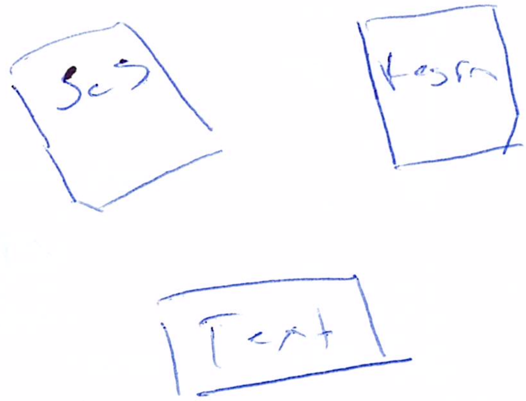
[illegible]

Recurrent
Neural
Network

I Yapılandırılmış | II Yapılandırılmamış

DATA

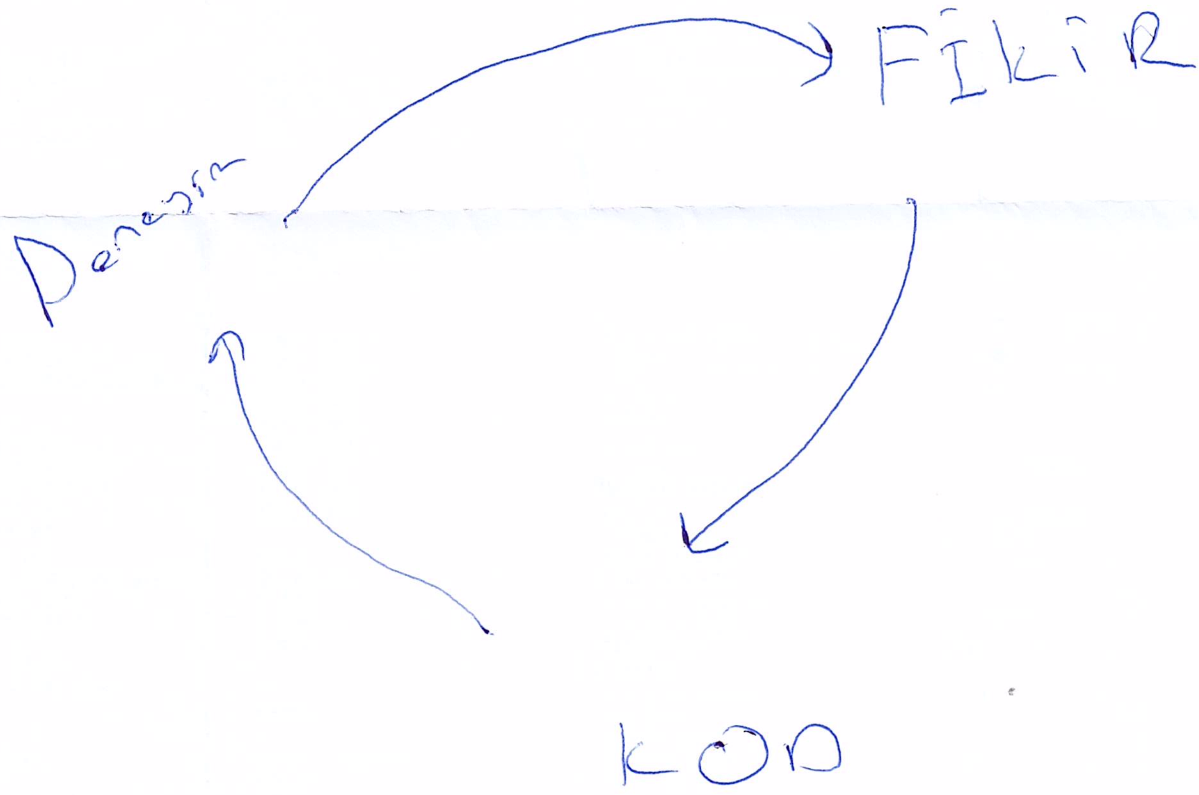


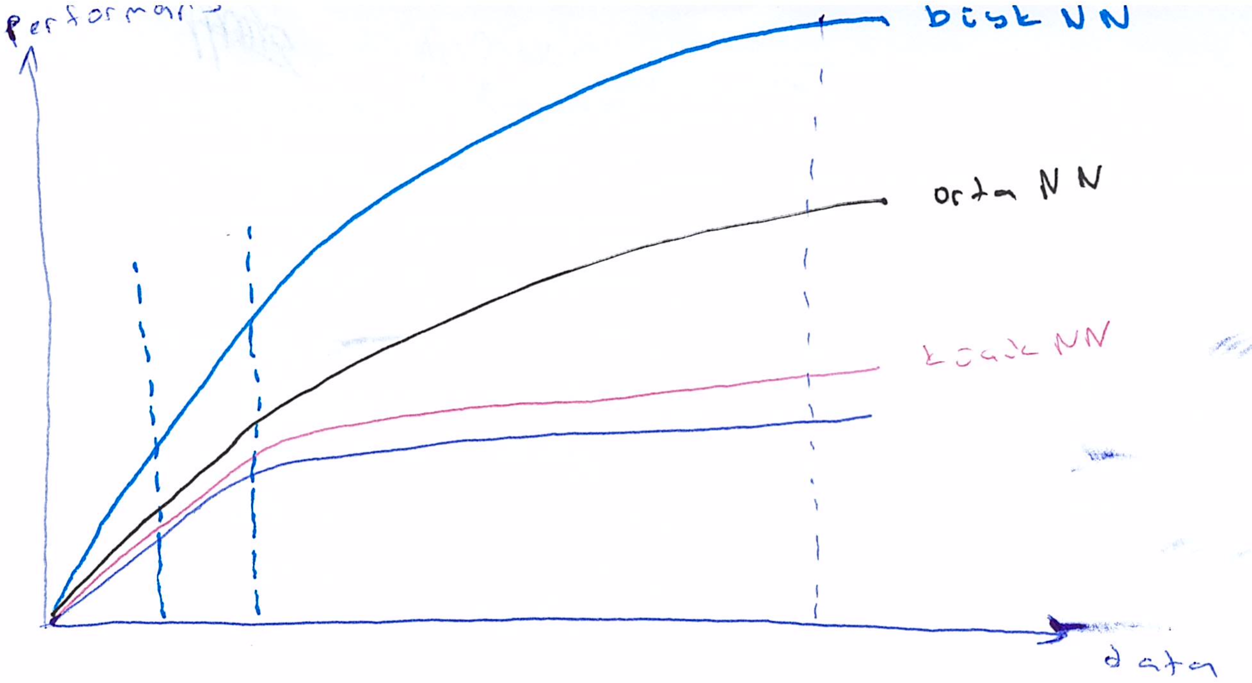


AI Neden Hortladı

* bazı yz algoritmaları 1960 da yazılmıştır!

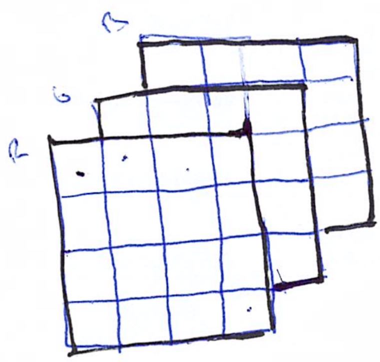
1- Data arttı
2- İşlem kaabiliyeti arttı
3- Algoritmalar gelişti





Lojistik Regresyon — var / yok (kedi)

İkili Sınıflandırma — tıkla / tıklama



$$X = \begin{bmatrix} [R1, R2, R3 \ldots R(64)], \\ [G1, \ldots, G(64)], \\ [B1, \ldots, B(64)] \end{bmatrix}$$

$$X = \begin{bmatrix} R1 \\ \vdots \\ R(64)^2 \\ G1 \\ \vdots \\ G(64)^2 \\ B1 \\ \vdots \\ B(64)^2 \end{bmatrix} \Rightarrow n_x = 12288$$

Jadan

$(x, y) \quad x \in \mathbb{R}^{n_x}, \quad y \in \{1, 0\}$

eğitim satır (m) $\Rightarrow (x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots (x^{(m)}, y^{(m)})$

m_{train}, m_{test}

(Satır, sutun)

$$X = \begin{bmatrix} | & | & & | \\ x_1, & x_2, & \ldots & x_m \\ | & | & & | \end{bmatrix} \Rightarrow (n_x, m)$$

$$Y = [Y_1, Y_2, \ldots Y_m] \Rightarrow (1, m)$$

$$\hat{y} = w^T . x + b \qquad S(z) = \frac{1}{1 + e^{-z}}$$

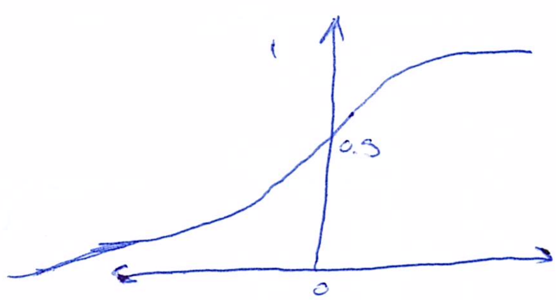
$\downarrow$

tahmin

$$\hat{y} = \sigma(w^T . x + b)$$

$\downarrow$

$0 \leq y \leq 1$



İleri Yayılım

Kayıp Fonksiyonu : tahmin olan $\hat{y}$ 'nın gerçekte ne kadar doğru olduğunu hesaplar

$$\mathcal{L}(\hat{y}, y) = -\left[y \log \hat{y} + (1-y)\log(1-\hat{y})\right]$$

Maliyet Fonksiyonu : toplamda ne kadar tutarlı tahmin yaptığını ölçer

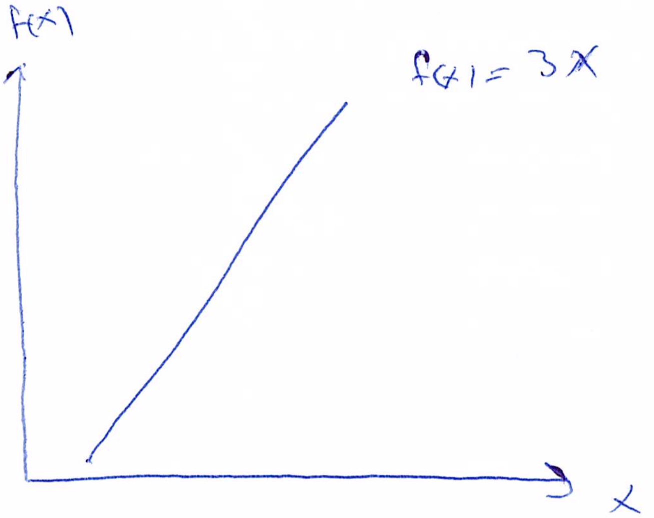
$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\hat{y}^{(i)}, y^{(i)})$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log \hat{y}^{(i)} + (1-y^{(i)})\log(1-\hat{y}^{(i)})\right]$$

☆ hesapladık, sonra hesabımız gerçek ile karşılaştırdık. Dedik ki hesabımız 0.3 doğru. Sonra w ve b'yi doğru yönde* güncelledik. Sonra tekrar hesapladık; Dedik ki hesabımız 0.4 doğru. Bu şekilde ~~tekrarlar~~ adım adım eğitip doğru ağırlıkları bulduk. Eğitim tamamlandı

* doğru yöne gitmeyi geri yayılım ile bulacağız

bir parantez açıp türev bakalım [ ]

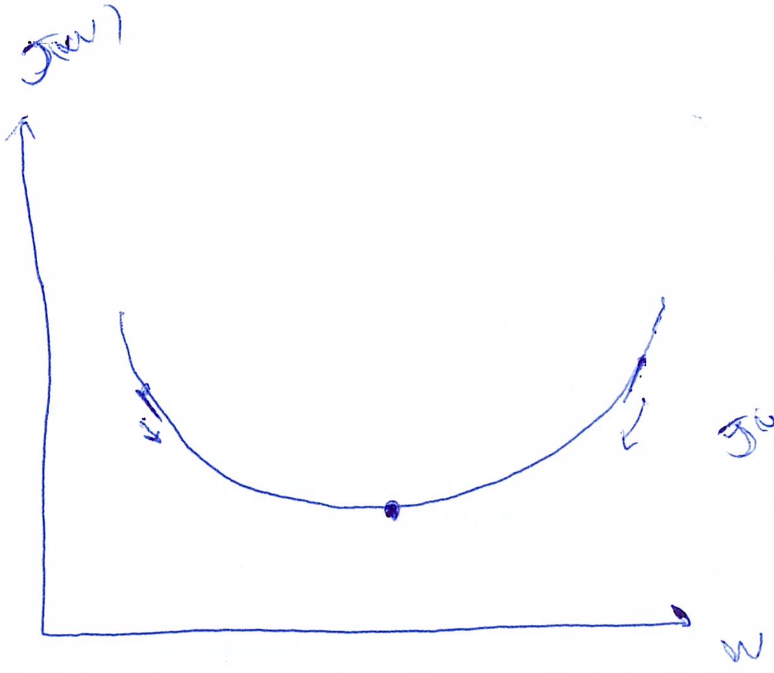


eğim = $f(x)' = 3$

$\Downarrow$

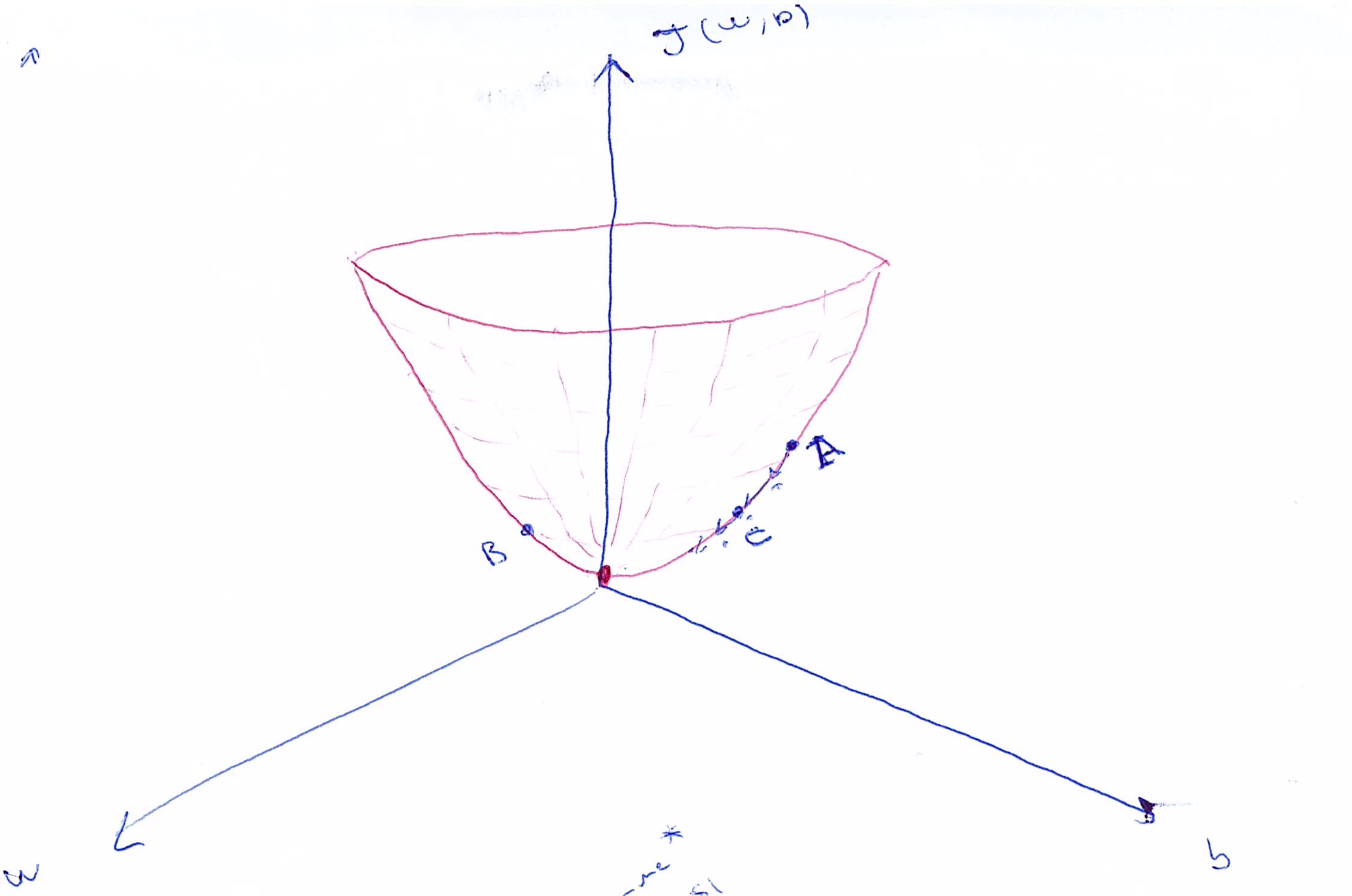
$f(x) = 3x^2$ olsaydı

eğim = $6x$



$J(w)' = \frac{dJ(w)}{dw}$

w'yu türev yönünde (yani eğim) biraz azalt.



öğrenme katsayısı *

$$w = w - \alpha \frac{\partial J(w,b)}{\partial w} = w - \alpha \, \partial w$$

$$b = b - \alpha \frac{\partial J(w,b)}{\partial b} = w - \alpha \cdot \partial b$$

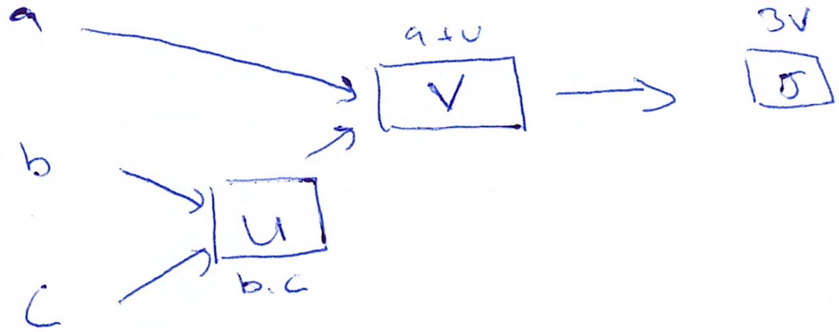
* Eğer küçük bir öğrenme katsayısı olmaz ise grafik üzerinde A dan B'ye B den C'ye atlar dururuz. Öğrenme katsayısını küçük seçiyoruz ki adım adım dibe yaklaşalım

$$J(a,b,c) = 3(\underbrace{a+\underbrace{bc}_{u}}_{v})$$

$$J(a,b,c) = 3v$$

$$v = a + u$$

$$u = b \cdot c$$



$$\boxed{\frac{\partial u}{\partial b} = c}$$

$$\boxed{\frac{\partial v}{\partial a} = 1}$$

$$\boxed{\frac{\partial u}{\partial c} = b}$$

$$\boxed{\frac{\partial J}{\partial a}} = \frac{\partial J}{\partial V} \cdot \frac{\partial V}{\partial a} = 3 \cdot 1 = 3 //$$

$$\frac{\partial J}{\partial V} = 3$$

$$\boxed{\frac{\partial J}{\partial b}} = \frac{\partial J}{\partial V} \cdot \frac{\partial V}{\partial u} \cdot \frac{\partial u}{\partial b} = 3 \cdot 1 \cdot c$$

$$= 3c //$$

$$\frac{\partial V}{\partial u} = 1$$

$$\boxed{\frac{\partial J}{\partial c}} = \frac{\partial J}{\partial V} \cdot \frac{\partial V}{\partial u} \cdot \frac{\partial u}{\partial c} = 3 \cdot 1 \cdot b$$

$$= 3b //$$

Parentezi kapatıp düzelin

$$z = w^T \cdot x + b$$

$$\hat{y} = \sigma(z) = a$$

$$L(a, y) = -\left[y \log(a) + (1-y) \log(1-a) \right]$$

$$dz = a - y$$

$$dw = x \cdot dz$$

$$db = dz$$

$J = 0 \qquad dw = 0 \qquad db = 0$

for $i = 1$ to m

$$z^{(i)} = w^T \cdot x^{(i)} + b$$

$$a^{(i)} = \sigma(z^{(i)})$$

$$J \mathrel{+}= -\left[y^{(i)} \log(a)^{(i)} + (1 - y^{(i)}) \log(1 - \hat{a}^{(i)}) \right]$$

$$dz^{(i)} = a^{(i)} - y^{(i)}$$

$$dw \mathrel{+}= x^{(i)} \cdot dz^{(i)}$$

$$db \mathrel{+}= dz^{(i)}$$

$J/m, \quad dw/m, \quad db/m$

$w = w - \alpha\, dw$

$b = b - \alpha\, db$

Bir parantez daha açıp vektörleştirmeye bakalım

Normalde

```
z = 0
  for i=1 to 1000000
      z += x[i] * w[i]
z += b
```

$\}$ 450 ms

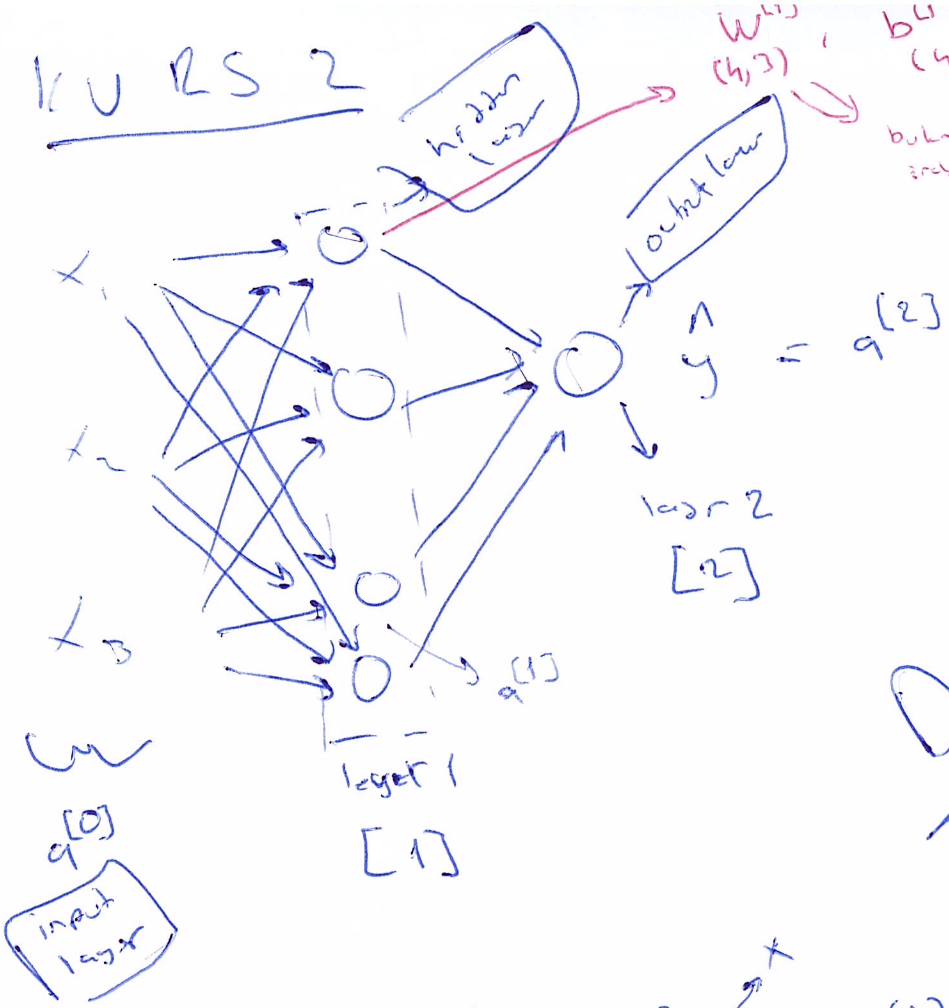
vektörlü

$z = np.dot(w, x) + b \; \}$ 1.5 ms

$$w \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \qquad x \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \qquad b \begin{bmatrix} \vdots \\ \vdots \end{bmatrix}$$

KURS 2

HAFTA 3



Derin

$$[1]\begin{cases} z^{[1]} = W^{[1]} a^{[0]} + b^{[1]} \\ a^{[1]} = \sigma(z^{[1]}) \end{cases}$$

x

$$[2]\begin{cases} z^{[2]} = W^{[2]} a^{[1]} + b^{[2]} \\ a^{[2]} = \sigma(z^{[2]}) \end{cases}$$

$$[l]\begin{cases} \boxed{z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}} \\ \boxed{a^{[l]} = \sigma(z^{[l]})} \end{cases}$$

ileriki derslerin konusu

$$a^{[l](i)} = \boxed{a_i^{[l]}}$$

$[l]$ akt.

i node in layer

$$\begin{bmatrix} - & w_1^{[1]T} & - \\ - & w_2^{[1]T} & - \\ - & w_3^{[1]T} & - \\ - & w_4^{[1]T} & - \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix} = \begin{bmatrix} - & w_1^{[1]T} x + b_1^{[1]} & - \\ - & w_2^{[1]T} x + b_2^{[1]} & - \\ - & w_3^{[1]} x + b_3^{[1]} & - \\ - & w_4^{[1]} x + b_4^{[1]} & - \end{bmatrix}$$

$(4, 3)$

$\Downarrow$

$z^{[1]} \Rightarrow$ ~~b~~

$*$ broadcast ile $(4, 3)$

(x)

$(4, 1)$

$$\begin{bmatrix} a_0^{[0]} \\ a_1^{[0]} \\ a_2^{[0]} \end{bmatrix}$$

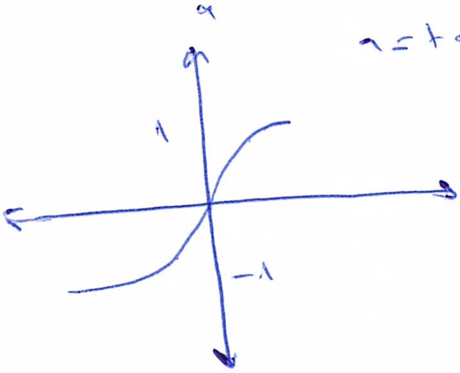
$\Downarrow$

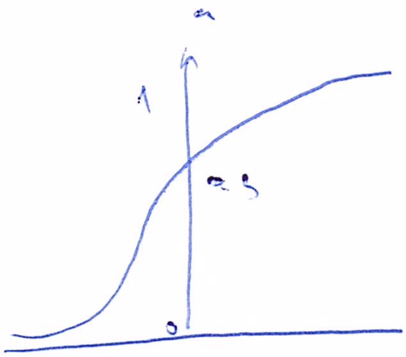
$$\begin{bmatrix} - & z_1^{[1]} & - \\ - & z_2^{[1]} & - \\ - & z_3^{[1]} & - \\ - & z_4^{[1]} & - \end{bmatrix}$$

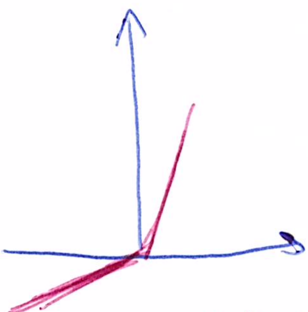
Aktivasyon fonksiyonları

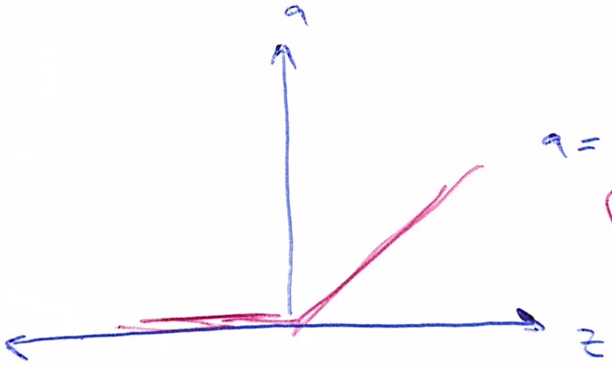
$a = \text{sigmoid}(a) = \frac{1}{1+e^{-z}}$

$a = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$









$a = \max(0, z)$

ReLU

Leaky ReLU

$a = (0.01z, z)$

Sonuç eğer {0, 1} ise sigmoid, ayrıca fiyat bilgisi gibi $0 < y < \mathbb{R}^+$ ise ReLU kullanılabilir.

İki katmanlı ağlar için

$$Z^{[1]} = W^{[1]} \cdot X + b^{[1]}$$
$$A^{[1]} = g(Z^{[1]})$$
$$Z^{[2]} = W^{[2]} \cdot A^{[1]} + b^{[2]}$$
$$A^{[2]} = g(Z^{[2]}) = \sigma(Z^{[2]})$$

forward

$$dZ^{[2]} = A^{[2]} - Y \qquad Y = [y^{(1)}, y^{(2)}, \ldots y^{(m)}]$$
$$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}$$
$$db^{[2]} = \frac{1}{m} \text{np.sum}(dZ^{[2]}, \text{axis}=1, \text{keepdims}=\text{True})$$
$$dZ^{[1]} = \underbrace{W^{[2]T} \cdot dZ^{[2]}}_{(n^{[1]}, m)} * \overbrace{g^{[1]'}(Z^{[1]})}^{(n^{[1]}, m)}$$
$$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$$
$$db^{[1]} = \frac{1}{m} \text{np.sum}(dZ^{[1]}, \text{axis}=1, \text{keepdims}=\text{True})$$

ırastgele ağırlıklarla başlatılmalı

$$W^{[1]} = \text{np.random.rand}((2,2)) * 0.01$$

kurs 1

hafta 3

$n^{[l]}$ katmandaki node sayısı

$n^{[0]} = n_x = 3$ $n^{[1]} = 5$ $n^{[2]} = 5$

$L = 4$

$X = a^{[0]}$

$\hat{y} = a^{[L]}$

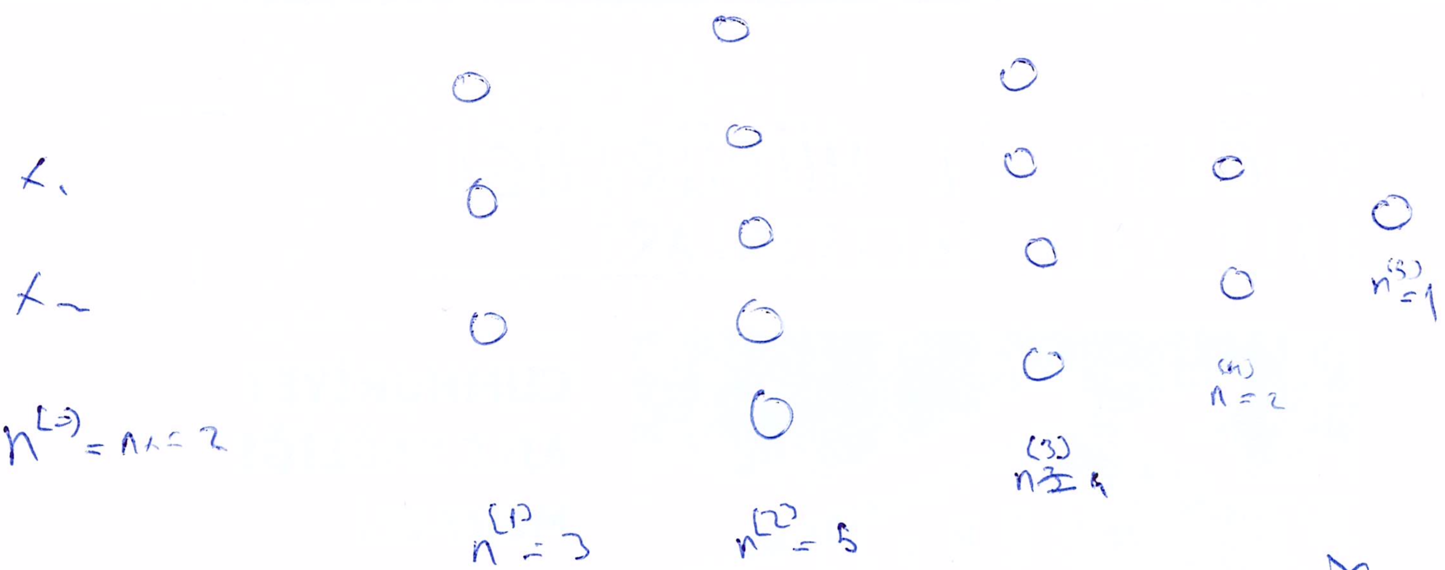
$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$$

forward

$$A^{[l]} = g^{[l]}(Z^{[l]})$$

giriş katmanlarda genelde tanh

gizli katmanda sigmoid yada softmax



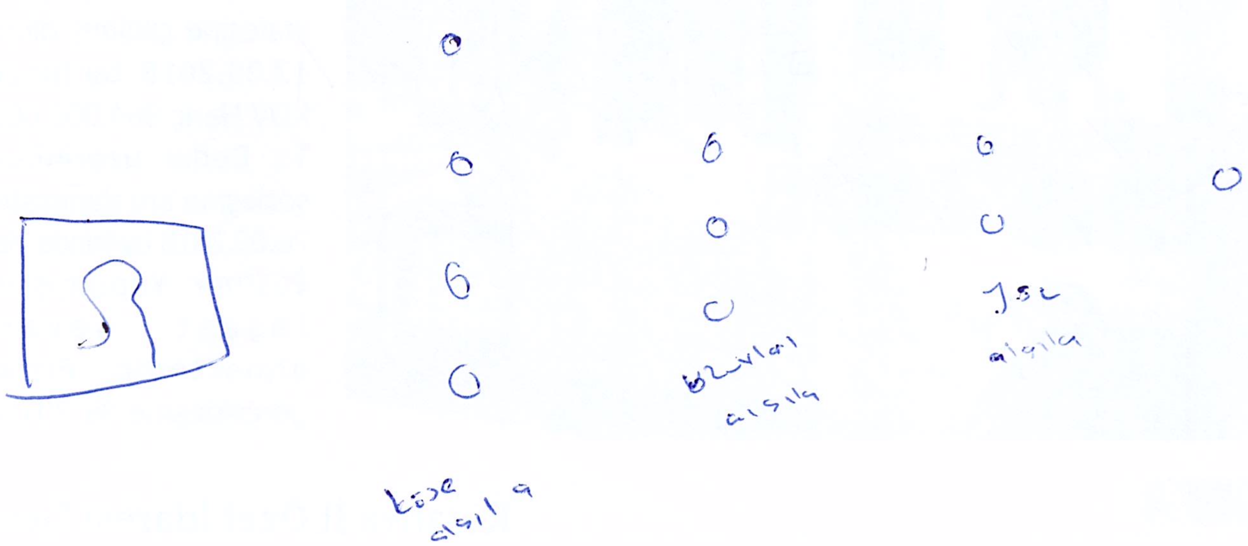
$$W^{[l]} = (n^{[l]}, n^{[l-1]}) \Rightarrow dW^{[l]}$$

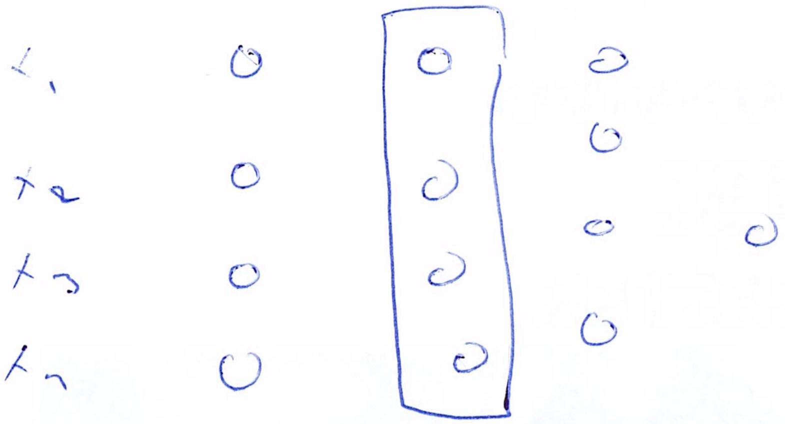
$$b^{[l]} = (n^{[l]}, 1) \Rightarrow db^{[l]}$$

$$A^{[l]} = (n^{[l]}, m) \Rightarrow dA^{[l]}$$

$$Z^{[l]} = (n^{[l]}, m) \Rightarrow dZ^{[l]}$$

Backward





layer l: $w^{[l]}$, $b^{[l]}$

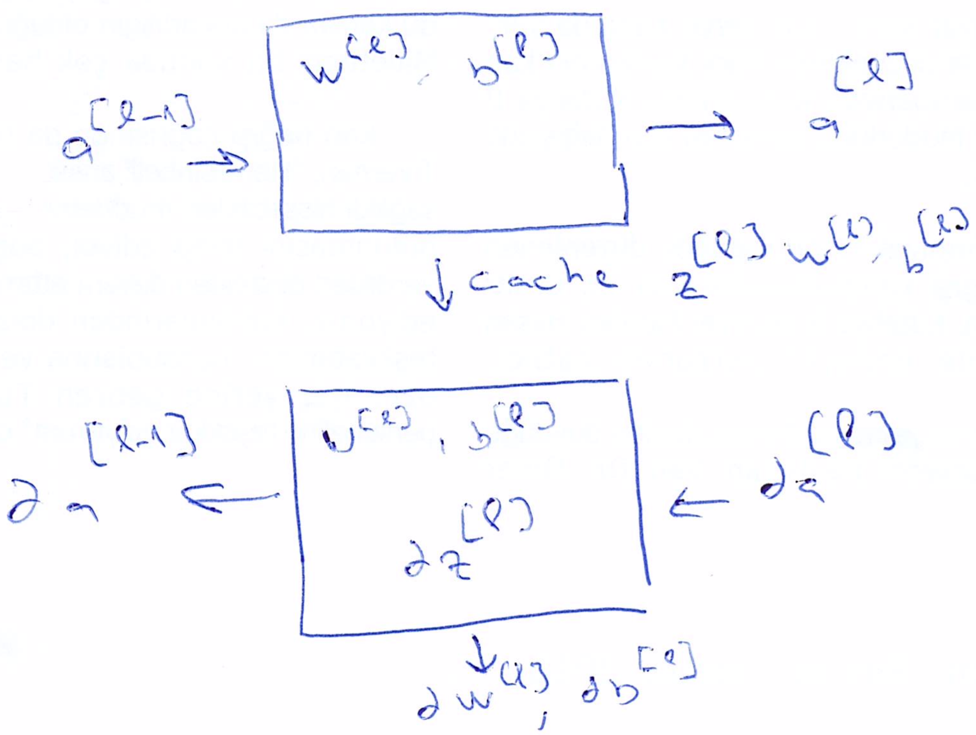
forward: Input $a^{[l-1]}$, output $a^{[l]}$

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$$

$$a^{[l]} = g^{[l]}(Z^{[l]})$$

backward: input $dA^{[l]}$ output $dA^{[l-1]}$

layer l



$$dz^{[l]} = da^{[l]} * g^{[l]'}(z^{[l]})$$

Backward

$$dw^{[l]} = dz^{[l]} \cdot a^{[l-1]}$$

$$db^{[l]} = dz^{[l]}$$

$$da^{[l-1]} = w^{[l]T} \cdot dz^{[l]}$$

$$\boxed{da^{[l]} = w^{[l+1]T} \cdot dz^{[l+1]}}$$

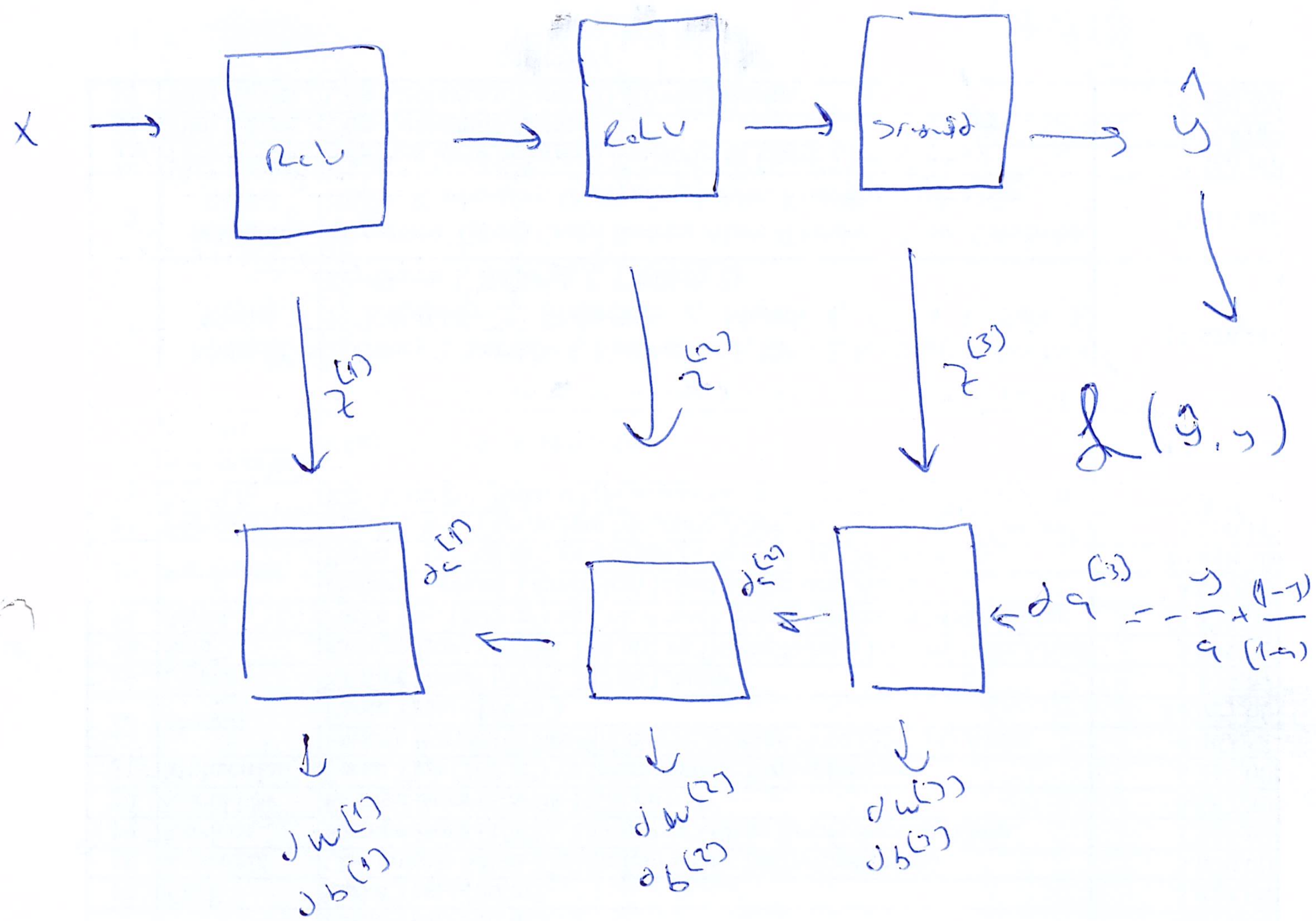
Vectorized

$$dZ^{[l]} = dA^{[l]} * g^{[l]'}(Z^{[l]})$$

$$dW^{[l]} = \frac{1}{m} dZ^{[l]} \cdot A^{[l-1]T}$$

$$db^{[l]} = \frac{1}{m} np.sum(dZ^{[l]}, axis=1, keepdims=True)$$

$$dA^{[l-1]} = W^{[l]T} \cdot dZ^{[l]}$$



Hiper parametreler

learning rate α

iterasyon sayısı

gizli katman

gizli units $n^{[1]}, n^{[2]} \ldots$

aktivasyon fonksiyonları