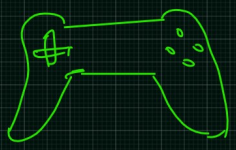
```
def se (y, ŷ):
    return (y - ŷ)^2
```

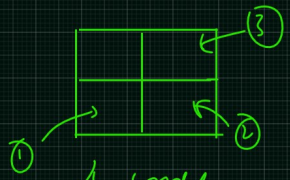
→ keras

. pytorch lightening



with respect to deep learning training

CPU



4 core

parallel processing

GPU

cuda cores → 1000s

$W \cdot x$

$$\underset{W}{\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}^{n} = \underbrace{1 \times 1}_{1} + \underbrace{2 \times 1}_{1} + \underbrace{3 \times 1}_{1}$$

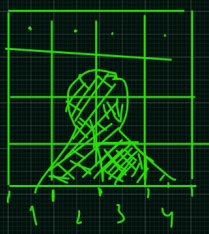
assume for each multiplication 1 sec

if using single core

total time = 3 sec

$$\begin{bmatrix} \quad \end{bmatrix}_{1 \times n} \begin{bmatrix} \quad \\ \quad \end{bmatrix}_{n \times 1} = (n)$$

$n = 100$ → 4 → 25 sec



4x4 pixels

1920 x 1080 , 3
R, G, B

60,000, 28, 28 → 1 channel → Black & White image

60,000
└ 1 [ ]28x28

8 bit image each pixel $2^8 = 256$

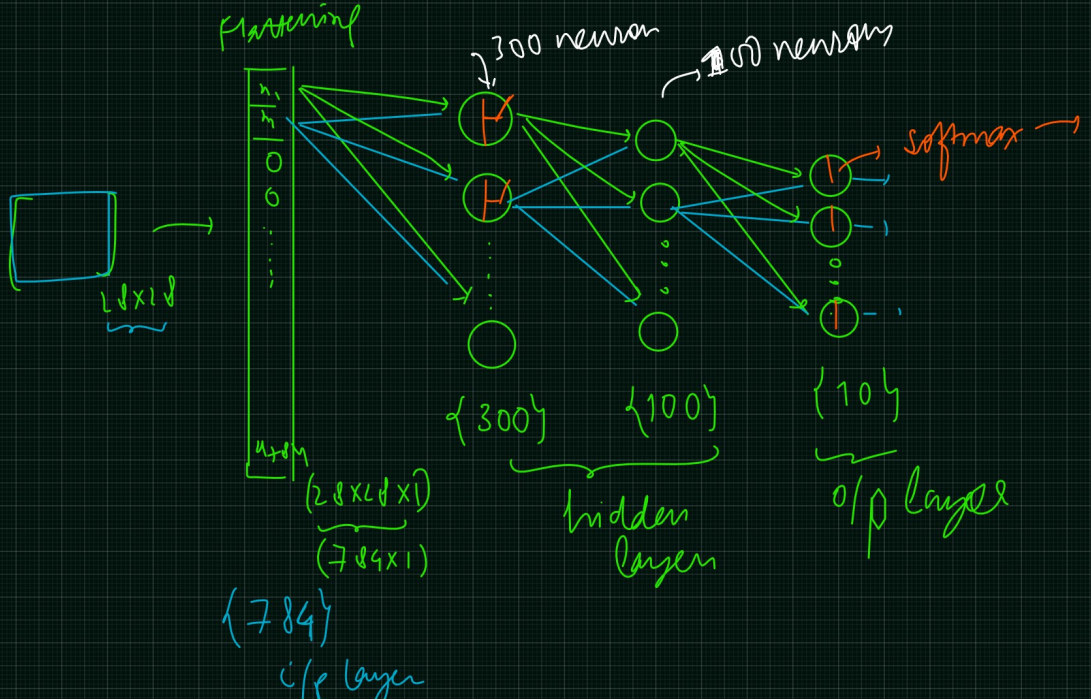
|||||

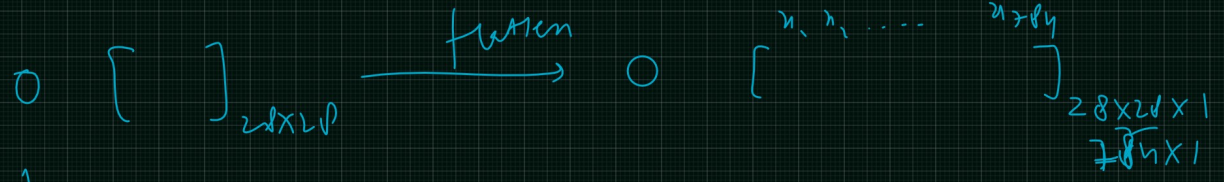
0 → 256 - 1
0 → 255

49 47 / 255 255

255 / 255 = 1

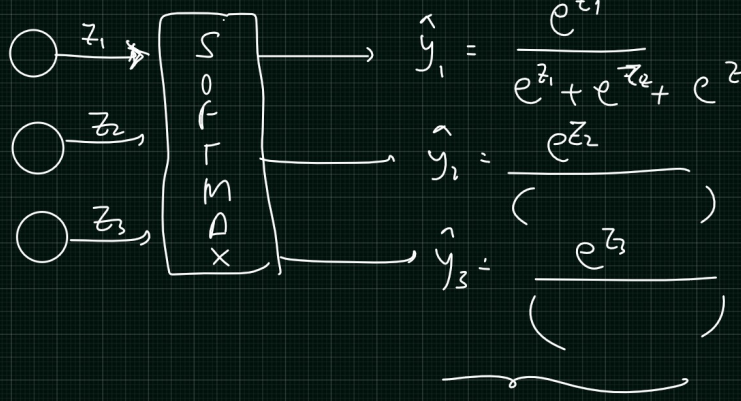
0 → 1





0
1
2

Softmax

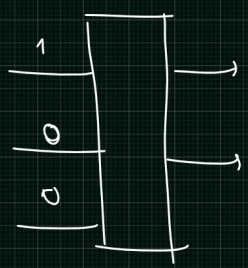


$$\hat{y}_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}} \quad D_r$$

$$\hat{y}_2 = \frac{e^{z_2}}{(\quad)}$$

$$\hat{y}_3 = \frac{e^{z_3}}{(\quad)}$$

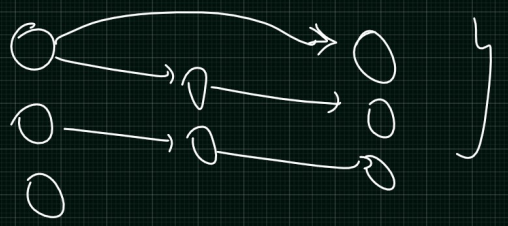
$$\frac{e^{z_1}}{D_r} + \frac{e^{z_2}}{D_r} + \frac{e^{z_3}}{D_r} = \frac{e^{z_1} + e^{z_2} + e^{z_3}}{D_r} = 1$$

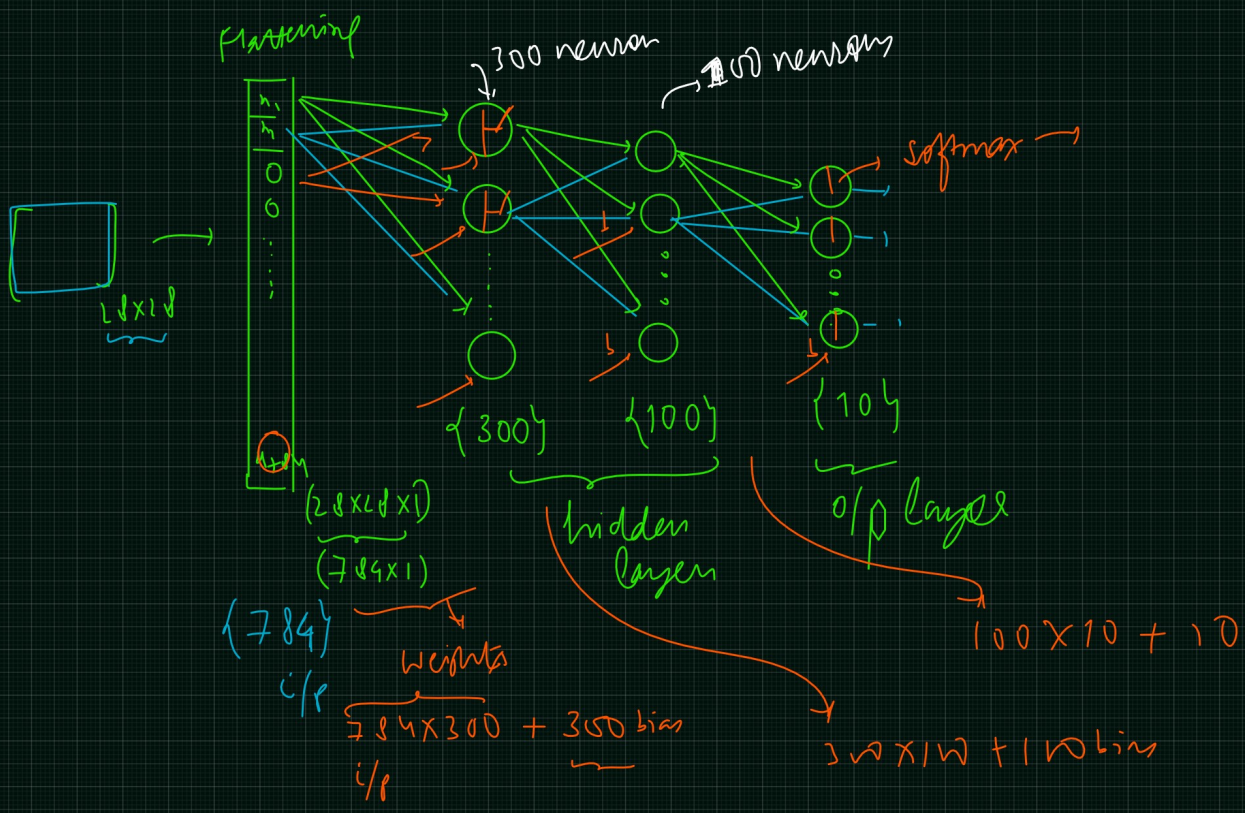


$$\hat{y}_1 = \frac{e^1}{e^1 + e^0 + e^0} = \frac{e^1}{e^1 + 1 + 1} = \frac{e^1}{e^1 + 2} = \frac{2.73}{4.73} \simeq 0.50$$

$$\hat{y}_2 = \frac{e^0}{e^1 + e^0 + e^0} = \frac{1}{e^1 + 2} = \frac{1}{4.73} \simeq 0.25$$

$$\hat{y}_3 = \frac{1}{e^1 + 2} = \frac{1}{4.73} \simeq 0.21$$





→ loss_fn ($y, \hat{y}$) = L ↓

Optimizer (= SGD

metrics = "accuracy"

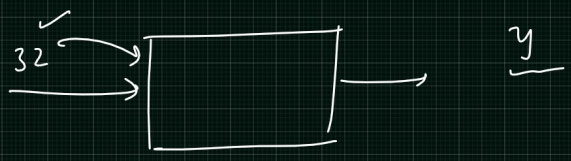
Train → 55,000

Batch size → 32 default.

Epochs = 30

Batches = $\frac{55,000}{32} \simeq 1719$.

every epoch → steps per epoch → 1719



Epoch : 30

Train data = 55,000

Batch size = 32

Steps per epoch = $\frac{\text{total data point}}{\text{Batch size}} \simeq 1719$

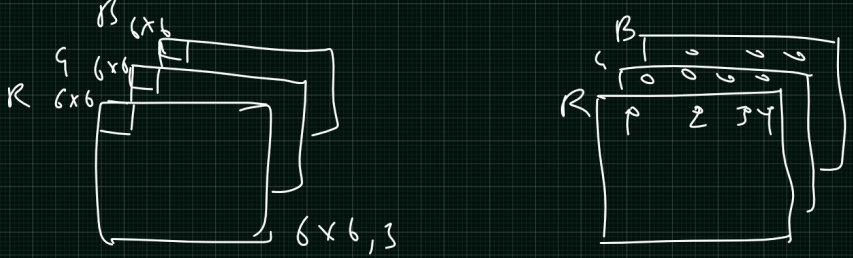
$32, (28, 28)$ — 1 batch

32×784

e_1
e_2
e_2
⋮
e_{32}

back propagation ← av. error

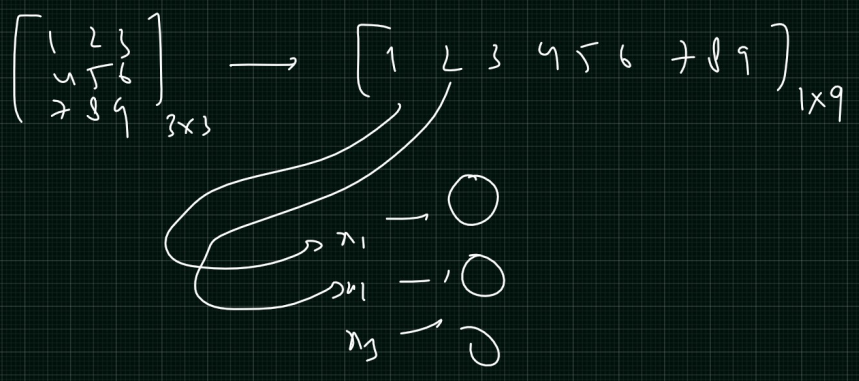
back prop = 1719 per epoch

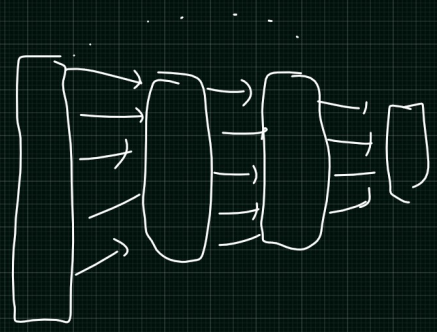


(r, g, b, a)

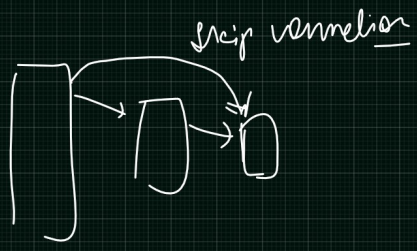
↑ transparency

$(\bar{r}, \bar{g}, \bar{b})$ -





Sequential



Functional

ResNet

100 ~~four~~ parameters → W, b → responsible for inferencing.

↓

W_1 = float → float32 float64

32bit × 1W 3200 bits

keras.h5 → 3200 bits = 3.2 kbits

Classification

- binary
  - softmax, 2 neuron
  - Sigmoid, 1 neuron
- multiclass
  - softmax

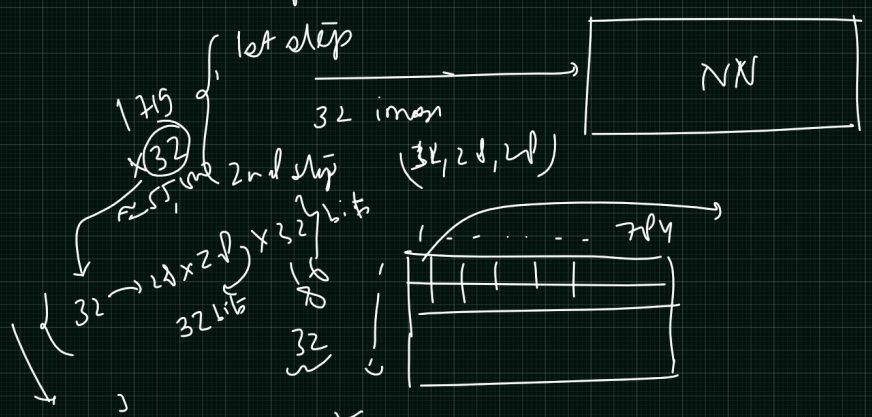
Regression

- Not required

55000 → 1719 → 32 images

1st epoch

1st step



32, 16, 2

binary 2

if else

1

0

1

2

0

1

Linh

-1

1M 10^6 10^5

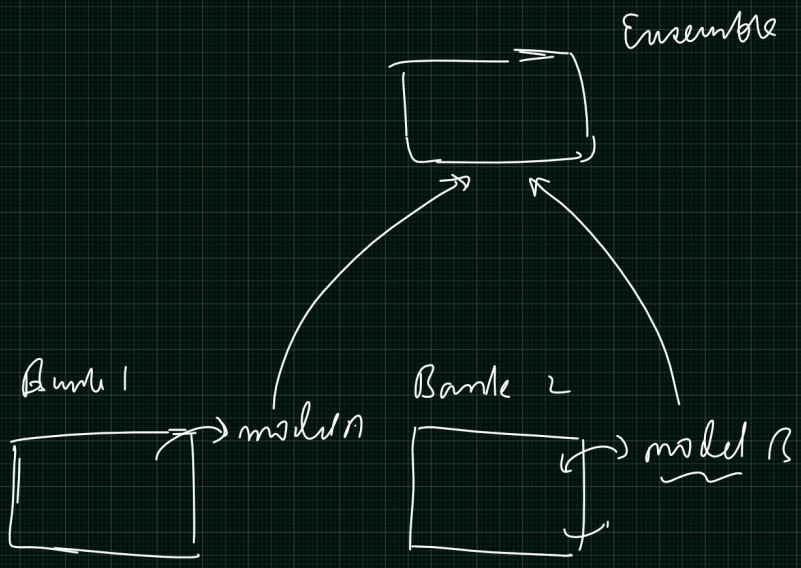
10,000

70% 30%

97% 5%

ANN → CNN → RNN → Transformers

VGG 16
Architecture



X_

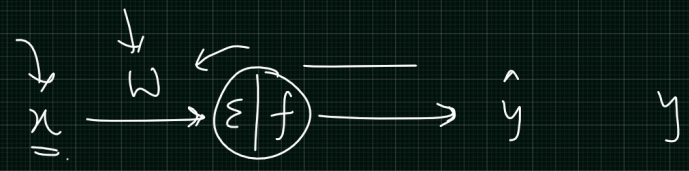
epochs = 30

x	y	pred ŷ

100 ⋮ 1

(32) = batch

$\frac{100}{32} \approx 3$ ✓

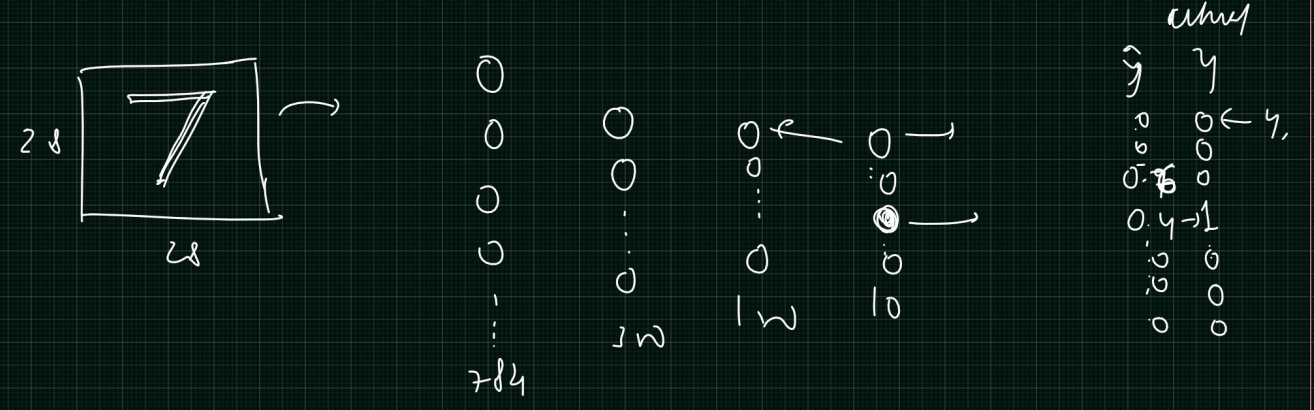


$\hat{y} = f(wx)$

error $= (\hat{y} - y) = e$

$w = w - \eta \frac{\partial e}{\partial w}$ → 0

$w = w$

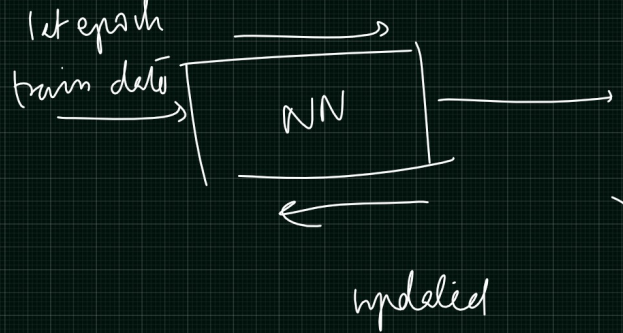


$$y \quad \hat{y} \qquad -\sum_{0}^{9} y \ln \hat{y}$$

$$-y_0 \ln \hat{y}_0 - y_1 \ln \hat{y}_1 - \cdots - y_7 \ln \hat{y}_7$$

$$-1 \ln 0.9 =$$

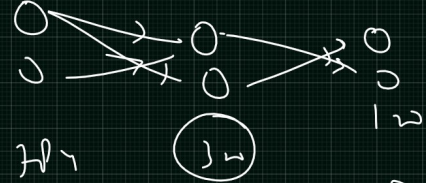
$$-1 \,(-0.9) = 0.91 \text{ error}$$



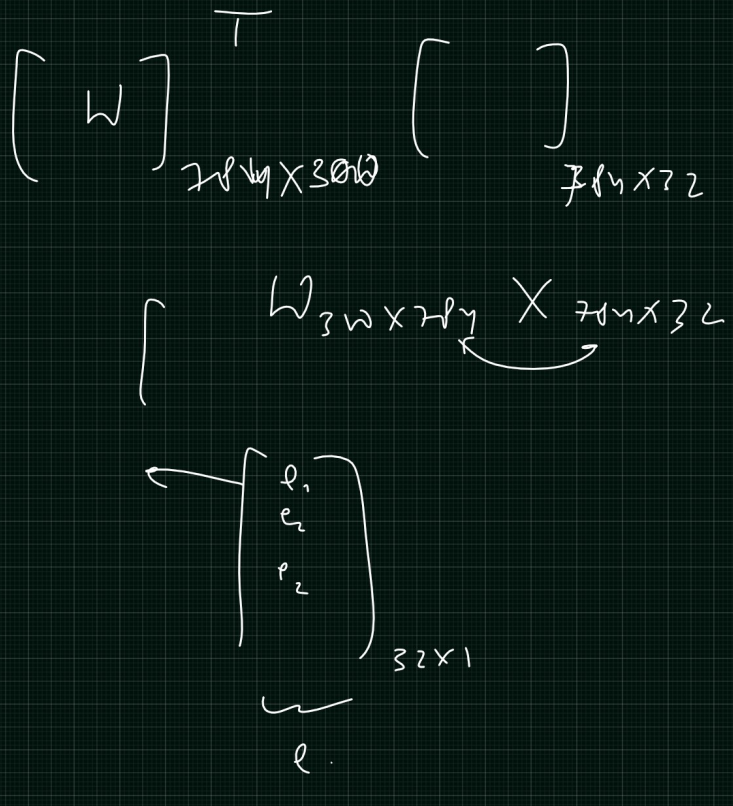
$$x_v W_1 = z \Rightarrow f(z) = \hat{y} - y_{valid}$$

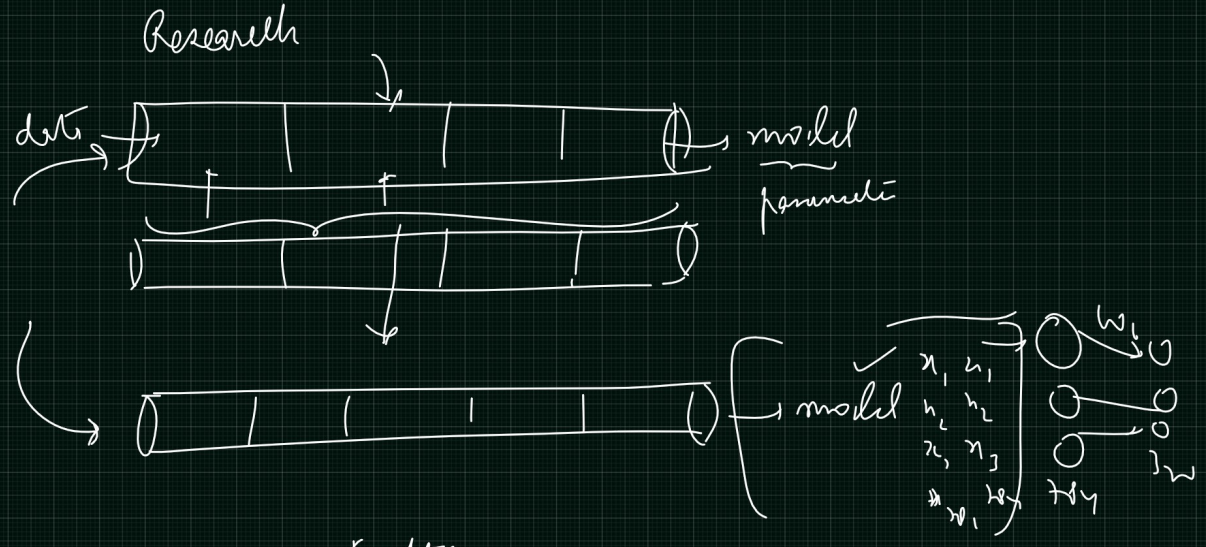
error

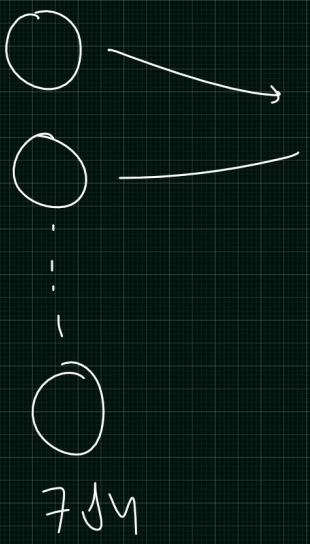
Callbacks → Early stopping

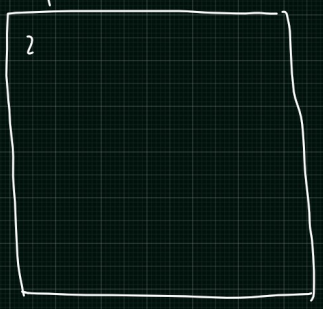


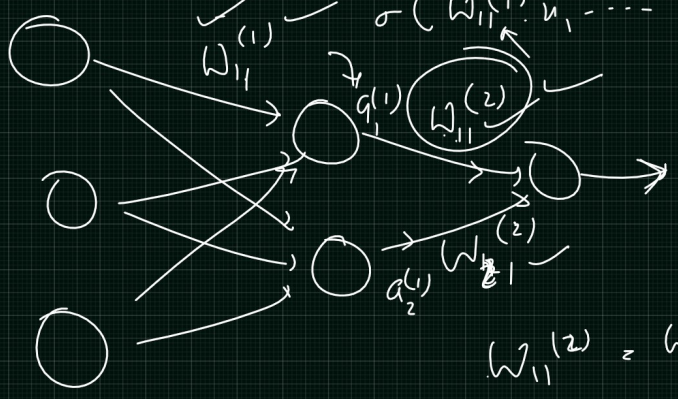
$$W = \begin{bmatrix} \\ \\ \end{bmatrix}_{300 \times 10}$$




Research
data
model
parameter
model
flatten
32,(28,28)
(32,784)
W
X
+ [ ]
3w x 784
784 x 32
3w x 32
A 3w x 32 + B 3w x 32
W 2x3
X 3x32
+ [ ] 2x32 =
= Z 3w x 32
A 2x32 + B 2x32 = Z 2x32
2x32
1x32
= Z 2x32







$e = y - \hat{y}$

$= y - y$

$f(x_1, x_2, x_3)$

$\left.\frac{\partial f}{\partial x_3}\right|_{x=x_3}$

$w_{11}^{(1)} = w_{11}^{(1)} - \eta \left.\frac{\partial e}{\partial w}\right|_{w=w_{11}^{(1)}}$

$w_{11}^{(2)} = w_{11}^{(2)} - \eta \left.\frac{\partial e}{\partial w}\right|_{w=w_{11}^{(2)}}$

$w_{11}^{(2)} = w_{11}^{(2)} - \Delta w$

$$\underline{\theta} \equiv (\tilde{W}_{11}, \tilde{W}_{12}, \cdots \quad \tilde{b}_1, \tilde{b}_2 \cdots \tilde{b}_L)$$

trainable parameters