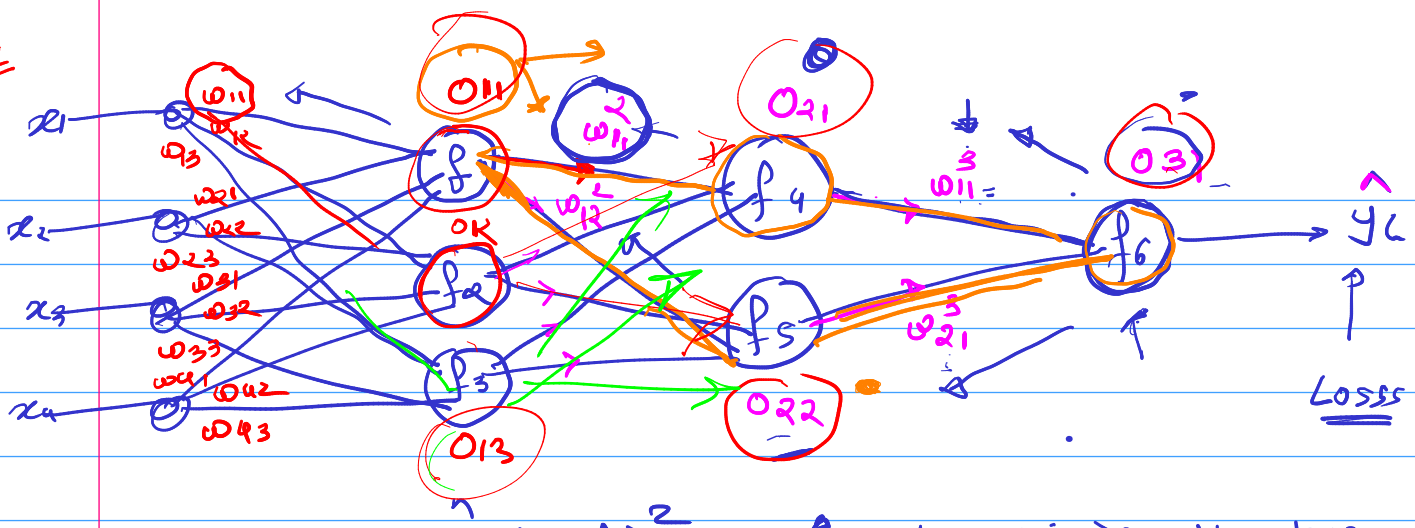
AWN



$$\text{Loss} = \sum_{i=1} (y - \hat{y})^2$$

↑ to minimize the loss, we use optimizer.

Back propagation → change / update, weights & bias.

$$\omega_{11}^{3}{}_{new} = \omega_{11}^{3}{}_{old} - \eta \frac{dL}{d\omega_{11}^{3}{}_{old}}$$

→ chain Rule

$$\frac{dL}{d\omega_{11}^{3}{}_{old}} = \frac{dL}{dO_{31}} \cdot \frac{dO_{31}}{d\omega_{11}^{3}} = \text{chain rule}$$

$$\frac{dL}{d\omega_{21}^{3}{}_{old}} = \frac{dL}{dO_{31}} \cdot \frac{dO_{31}}{d\omega_{21}^{3}{}_{old}}$$

$$\frac{dL}{d\omega_{11}^{2}} = \left[\frac{dL}{dO_{31}} \cdot \frac{dO_{31}}{dO_{21}} \cdot \frac{dO_{21}}{d\omega_{11}^{2}}\right] + \left[\frac{dL}{dO_{31}} \cdot \frac{dO_{31}}{dO_{22}} \cdot \frac{dO_{22}}{d\omega_{11}^{2}}\right] \rightarrow$$

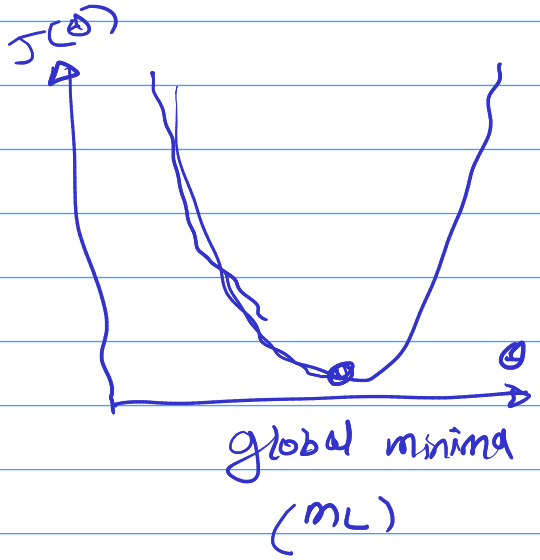
Chain Rule = we find $\frac{dL}{d\omega_{11}}$ on chain rule

epoch → 1 forward & 1 backward.

Optimizer (Reduce the loss)

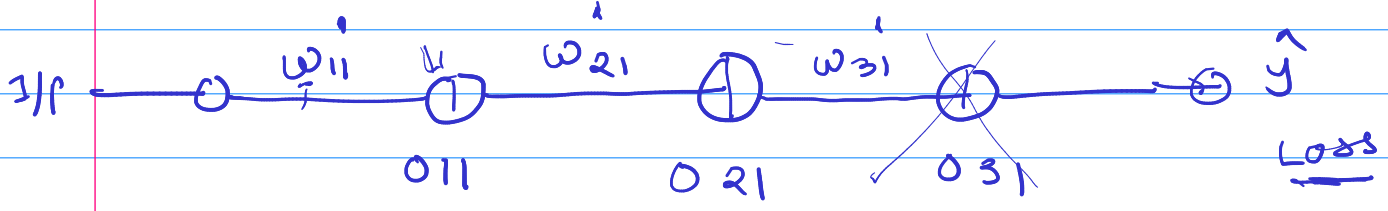
1) Gradient Decent

y & ŷ similar →



(2) Exploding Gradient Decent (problem)

↳ because of weight.



$$w'_{11\,new} = w'_{11\,old} - \frac{dL}{dw'_{11\,old}}$$

475 500

$$\frac{dL}{dw'_{11\,old}} = \frac{dL}{dO_{31}} \cdot \frac{dO_{31}}{dO_{21}} \cdot \frac{dO_{21}}{dO_{11}} \cdot \frac{dO_{11}}{dw'_{11}}$$

Sigmoid = $\frac{1}{1-e^{-z}}$ = 0 to 1

ex Loss = 500

$\frac{dL}{dO_3}$

= 0 < 0.25

= 0.25 × 500

= 125 ⟵ very hight

old & new.

③ Vanishing Gradient Problem

(when Rule

derivative of Sigmoid is alway (0 to 0.25) → explonday

so donivate alway smaller

$x0 \times 5w = 0$

$w_{new} \approx w_{old}$

Relu = (0 to 2)

$\frac{d}{dz}(zero)$ = not define.

dean = dead neon

Sig = 0 to 1

tanh = -1 to 1

Relu = 0 to 2

Leaky Relu = 0.01 to 2

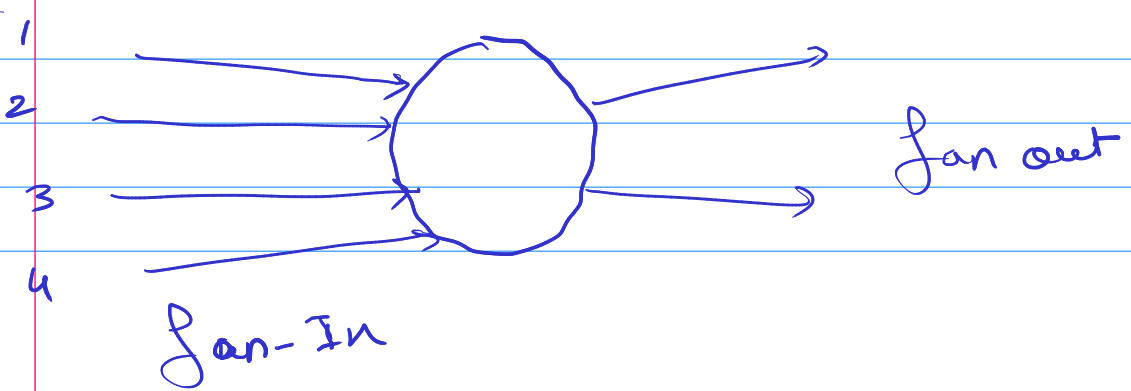
Softmax = multi

Weight Initialization

① Weight should be small ✓

② Weight should not be same ✓

③ should hae good variance. ✓



① Uniform Distribution

$$w_{ij} \sim \text{Uniform}\left[\frac{-1}{\sqrt{fan\text{-}In}}, \frac{1}{\sqrt{fan\text{-}In}}\right]$$

② Xavier / Glorot Distribution

$$w_{ij} \sim N(0, \sigma)$$

(0 = mean; σ = std. deviation)

Normal

(Sigma) $\sigma = \sqrt{\frac{2}{(fanIn + fanOut)}}$

$$w_{ij} \sim U\left[\frac{-\sqrt{6}}{\sqrt{fanIn + fanout}}, \frac{\sqrt{6}}{\sqrt{fanIn + fanout}}\right]$$

(six)

Uniform

④ He init

He uniform $\quad w_{ij} \sim U\left[-\sqrt{\frac{6}{fanin}}, \sqrt{\frac{6}{fanin}}\right]$

He init Normal $\quad w_{ij} \sim N(0, \sigma)$ (sigma)

$$\sigma_{sigma} = \sqrt{\frac{2}{fanin}}$$

③ SGD (Stochastic Gradient Decent)

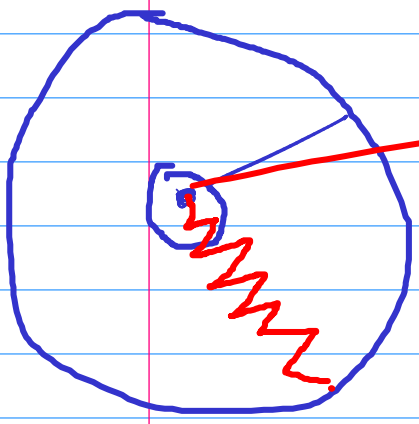
$$w_{new} = w_{old} - \eta \frac{dL}{dw_{old}}$$

$\frac{dL}{dw_{old}}$ → when we consider all data point called as Gradient decent optimizer.

$$\text{Loss} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

SGD → consider only 1 point

$$\text{Los} = (y_1 - \hat{y}_1)^2$$ → it take time to reach global mining



mini batch gradient decent = k is batch size

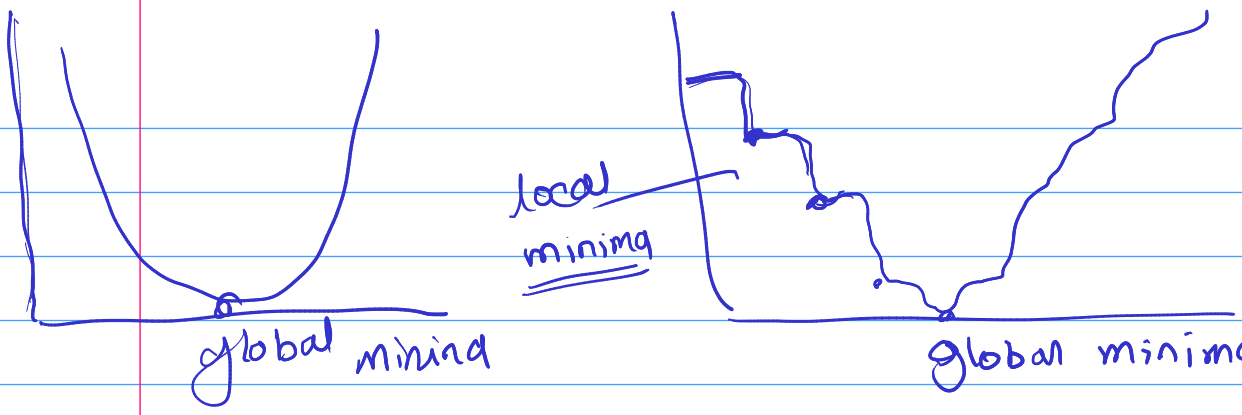
k < n n- data point
k batch

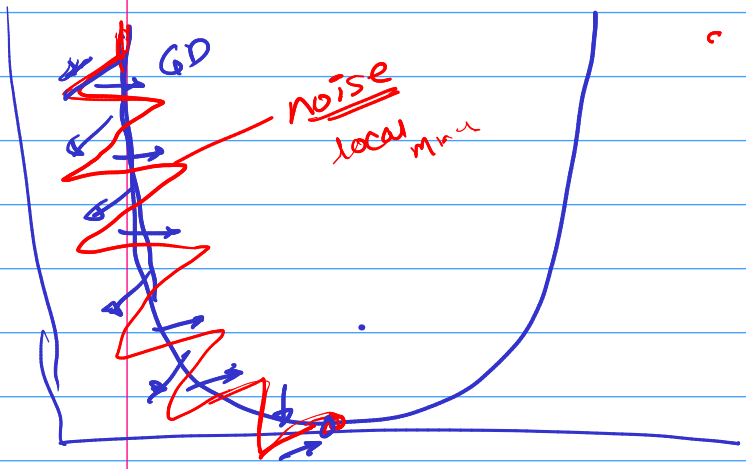
dL/dwold (mini batch) ≃ Gradient Decent

Loss in mini batch

$$\text{Loss} = \sum_{i=1}^{k} (y - \hat{y})^2$$

neg-mean-square-error.





° 1 batch m

④ SGD with momentum →

Since d_1 t_2 t_3 -- t_n → momentum add in Loss function

noise Reduce

↓

minimize loss not possible by SGD.

⑤ Adagrad optimizer (minimize the loss)

$$w_{new} = w_{old} - \eta \frac{dL}{dw_{old}}$$

$$w_{t}^{new} = w_{t-1} - \eta \frac{dL}{dw_{t-1}}$$

Can we use different learning rate for each & every neuron of hidden layer.

why we required ?

in ANN → 2 factor DENSE & SPARSE

DENSE → most factor is non zero

SPARSE → most factor is zero.

for Dense we use different learning rate

Sparse layer we use different learning rate for each.

$$\eta_t' = \frac{\eta}{\sqrt{\alpha_t + \epsilon}}$$

epsilon always small. use for not to zero

Constant Learning rate

$$\alpha_t = \sum_{i=1}^{t} \left[\frac{dL}{dw_t} \right]^2$$

t = time / epoch ep =

disadvantage = when iteration ↑ α_t is also high ↑

η is very small

⑥ Adadelta & RMSPROP

$$\eta_t = \eta \Big/ \sqrt{w_{Avg_t} + \epsilon}$$

i

= GD
Vanishi
explo
↓
SGD
↓
1 batch
↓
(noise) mini batch
↓
(noise) momentum
↓
Adagrad
α_t
↑ ↓ η ↓
Adadelta / RMS PROP

⑦ Adam

(Adaptive movement estimation)

momentum + RMSPROP

$$\eta_t = \frac{\eta}{\sqrt{w_{Avg_t} + \epsilon}} + M \leftarrow \text{momentum}$$

$-1/9$

Loss function (cost function)

loss $= (y - \hat{y})^2$

Loss $= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

ANN — Tabular data

Regression

Square error loss

$L = (y - \hat{y})^2$

(neg-mean-sq-error.)

Absolute

$L = |y - \hat{y}|^2$

RMS $= \sqrt{(y - \hat{y})^2}$

— classification

entropy

Cross-entropy

(binary-cross-entropy)

$\hat{y}$ = Sigmoid for binary

Los $= -y \log(\hat{y}) - (1-y) \log(1-\hat{y})$

multiclass — Cross-entropy

one-hot encoding

Categorical Cross entropy

Softmax

Sparse Categorical Cross entropy

Lable-encoding.

ANN