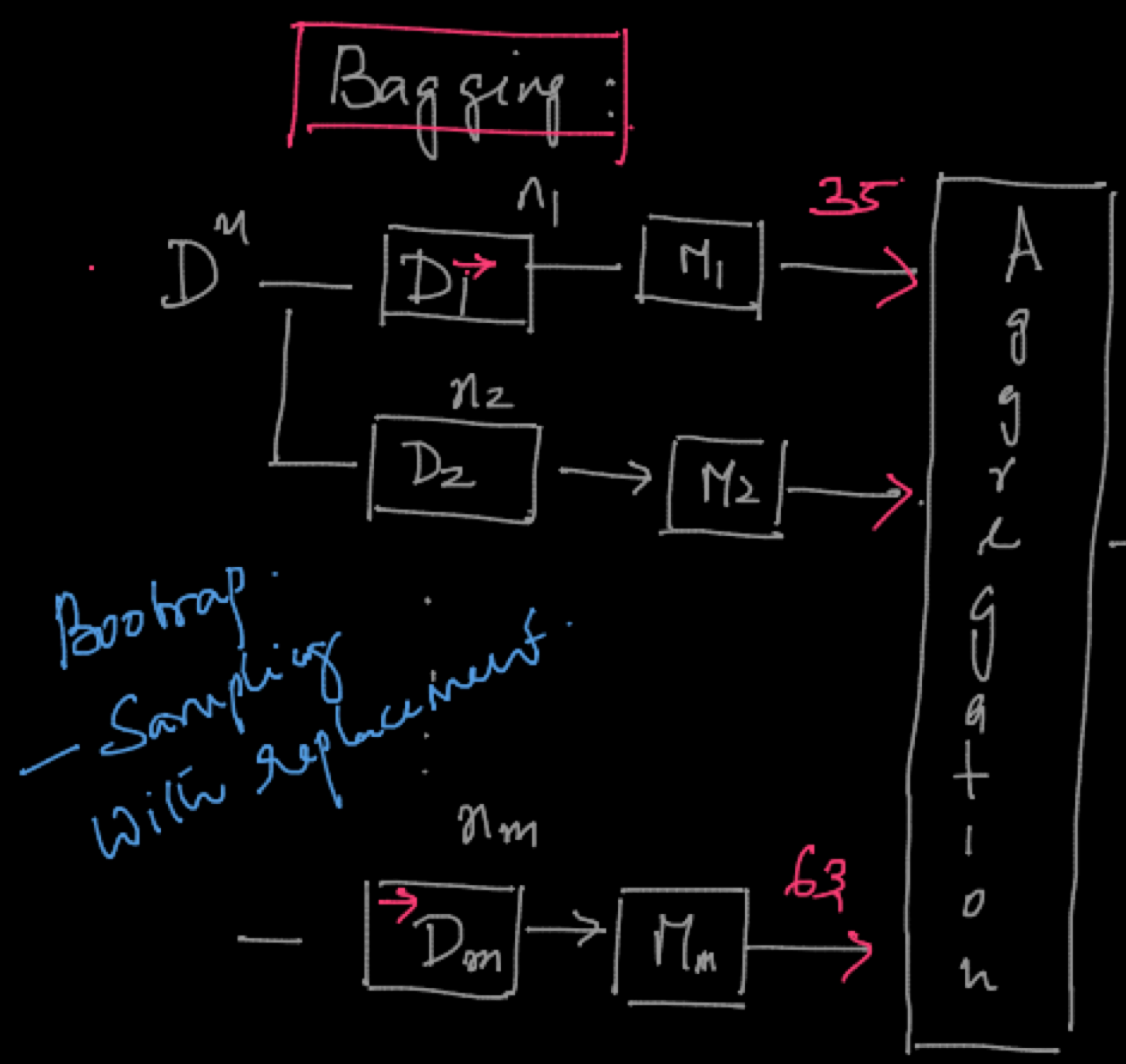
Ensemble Models.

1. Bagging → Bootstrapped Aggregation
2. Boosting.
3. Stacking.

1000's



Regression (Avg / Median)

One o/p.

Classification (Maj. Vote).

Random Forest.

- All base models are DT's.
- Feature Sampling + Row Sampling

→ Base models low bias / High Variance.

→ Aggregation reduces Variance.

→ ... fore parallalizable.

Boosting

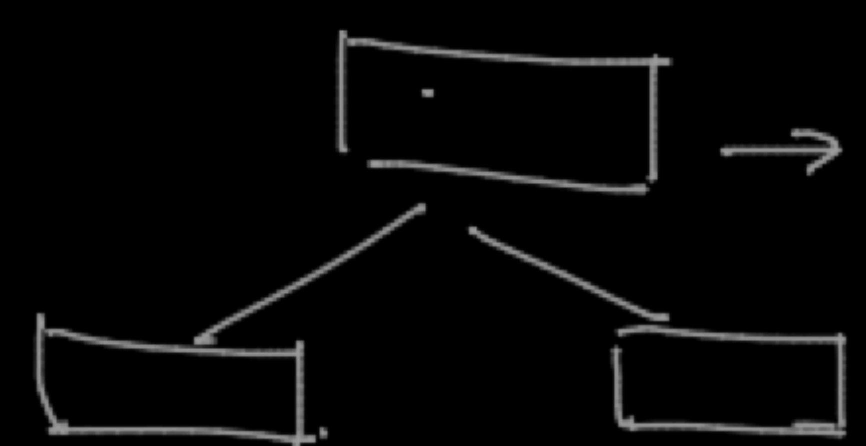
— Sequential Model / Additive Model.

→ Make a rough prediction of y.

— Subsequent models will work to reduce the errors from prev. models.

— Base models: High Bias | Low Variance.

Decision Stump ←



→ only one split / split based on only one feature.

ADABOOST.

→ Classification.

→ Records wrongly classified are given high weightage.

→ Individual models are also given weightage based on their performance

Opera File Edit View History Bookmarks Developer Window Help

Gradient Descent.

— minimize the 'loss function'.
(Errors / Residual)

$$\hat{y} = \beta_0 + \beta_1 x$$

residual $= y - \hat{y}$.

OLS.

→ line of best fit.

→ β_0, β_1 minimum

loss fn. $\Rightarrow (y - \hat{y})^2$.

$\Rightarrow$ squared loss.

Minimum Error in the model.

$$\underset{(L)}{SSE} = \left[y - (\beta_0 + \beta_1 x) \right]^2$$

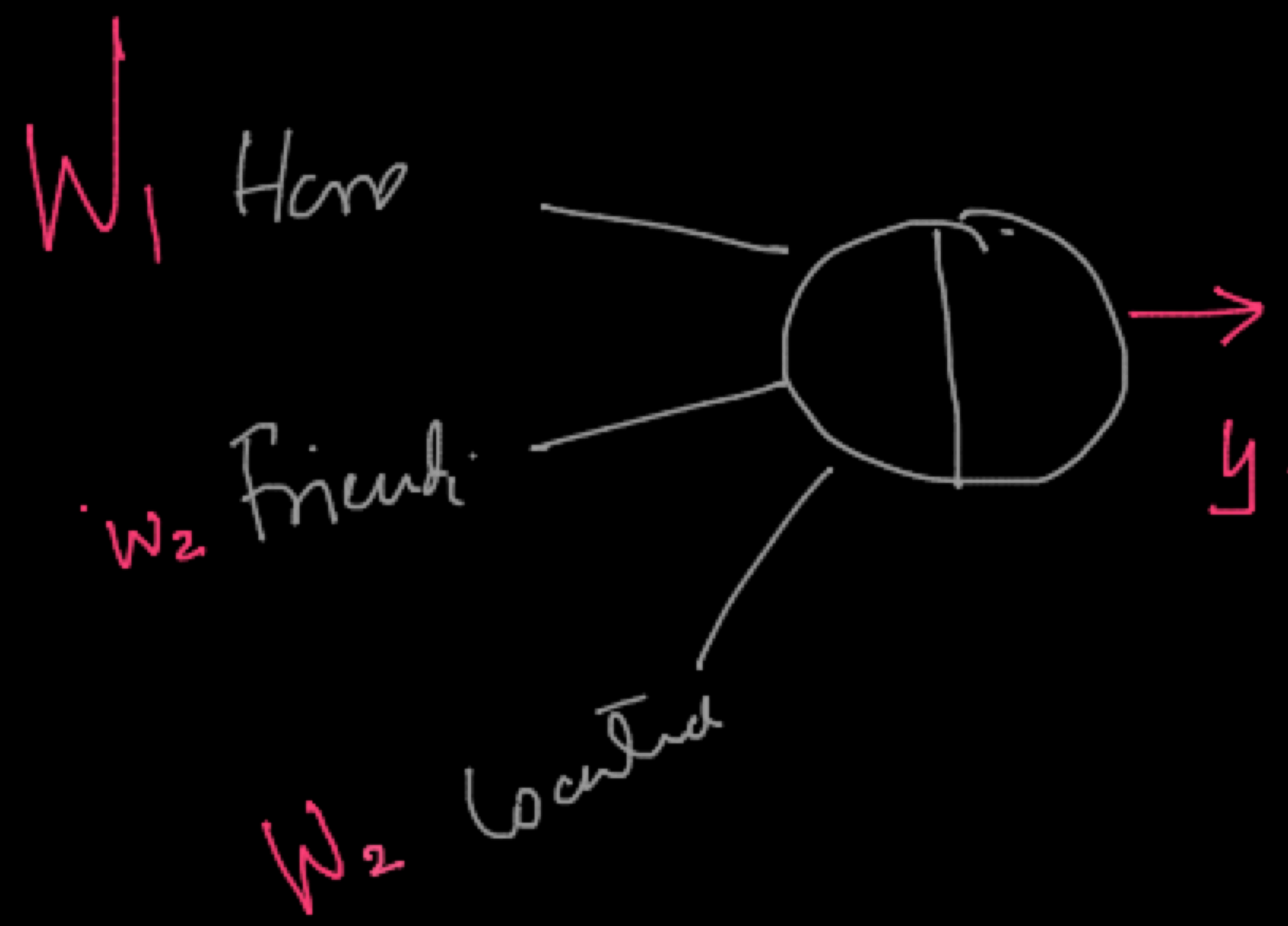
Find β_0, β_1 which will minimize the loss fn.

3 / 3

β's → Weights.

$y = w_1 x_1 + w_2 x_2 + \dots + b.$

↳ Weightage given to feature x_1

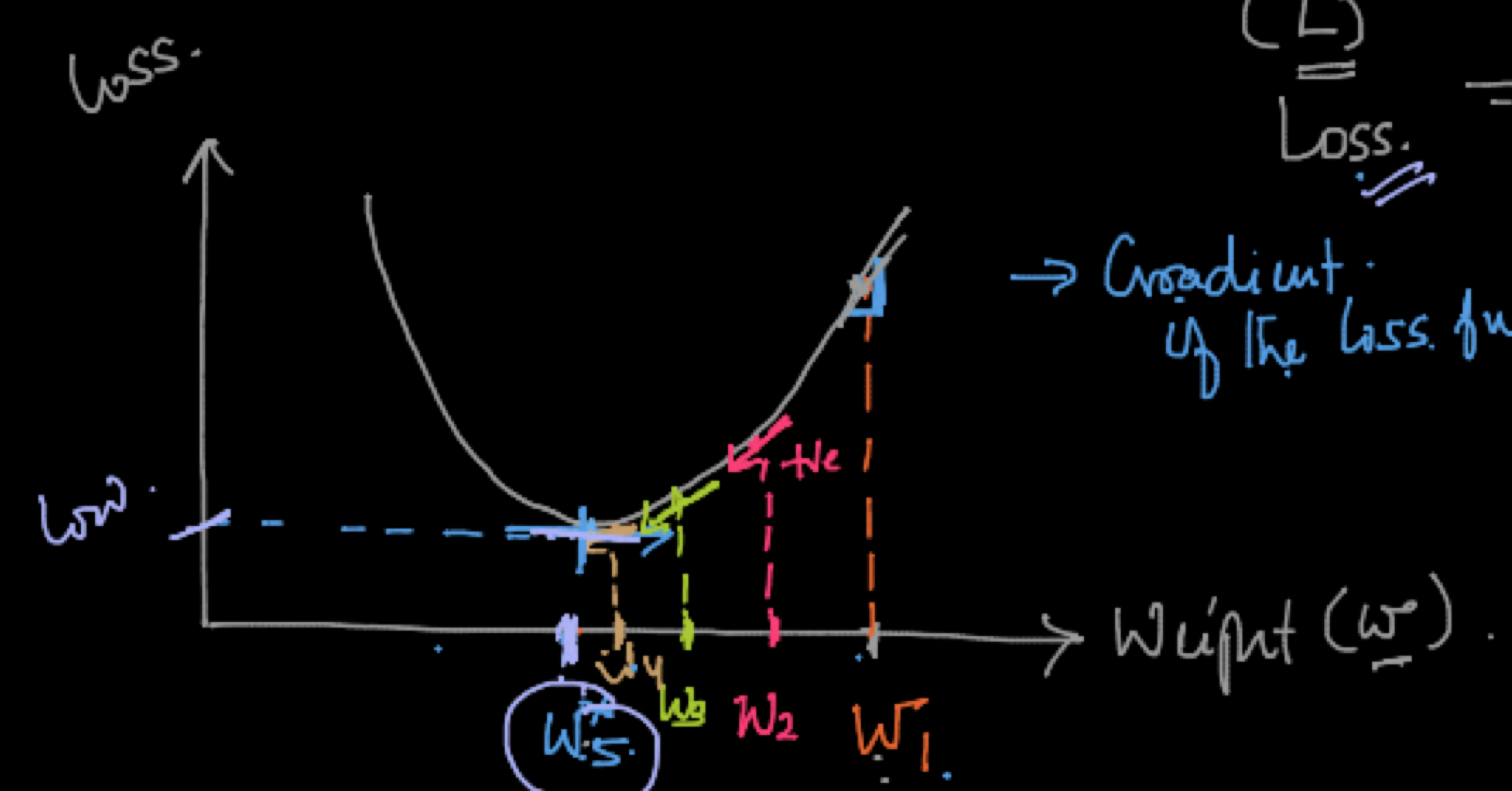


$y \Rightarrow W_1 \times \text{Hero} + W_2 \times \text{Friend} + W_3 \times \text{location}$

$(dy/dx) \rightarrow$ gradient

$$\boxed{y = Wx + b.} \quad Wx \rightarrow W_s, \quad b \rightarrow 10$$

$$\text{SSE (L) Loss} = (y - \hat{y})^2 = [y - (Wx + b)]^2$$



$\left(\frac{\partial L}{\partial W}\right)_{w_1}$ → Gradient of the loss fn w.r.t. the weight at $W = W_1$

Update the Weight

$$W_2 = W_1 + \underbrace{\left(-\frac{\partial L}{\partial W}\right)_{W_1}}_{\Delta W} \quad +ve$$

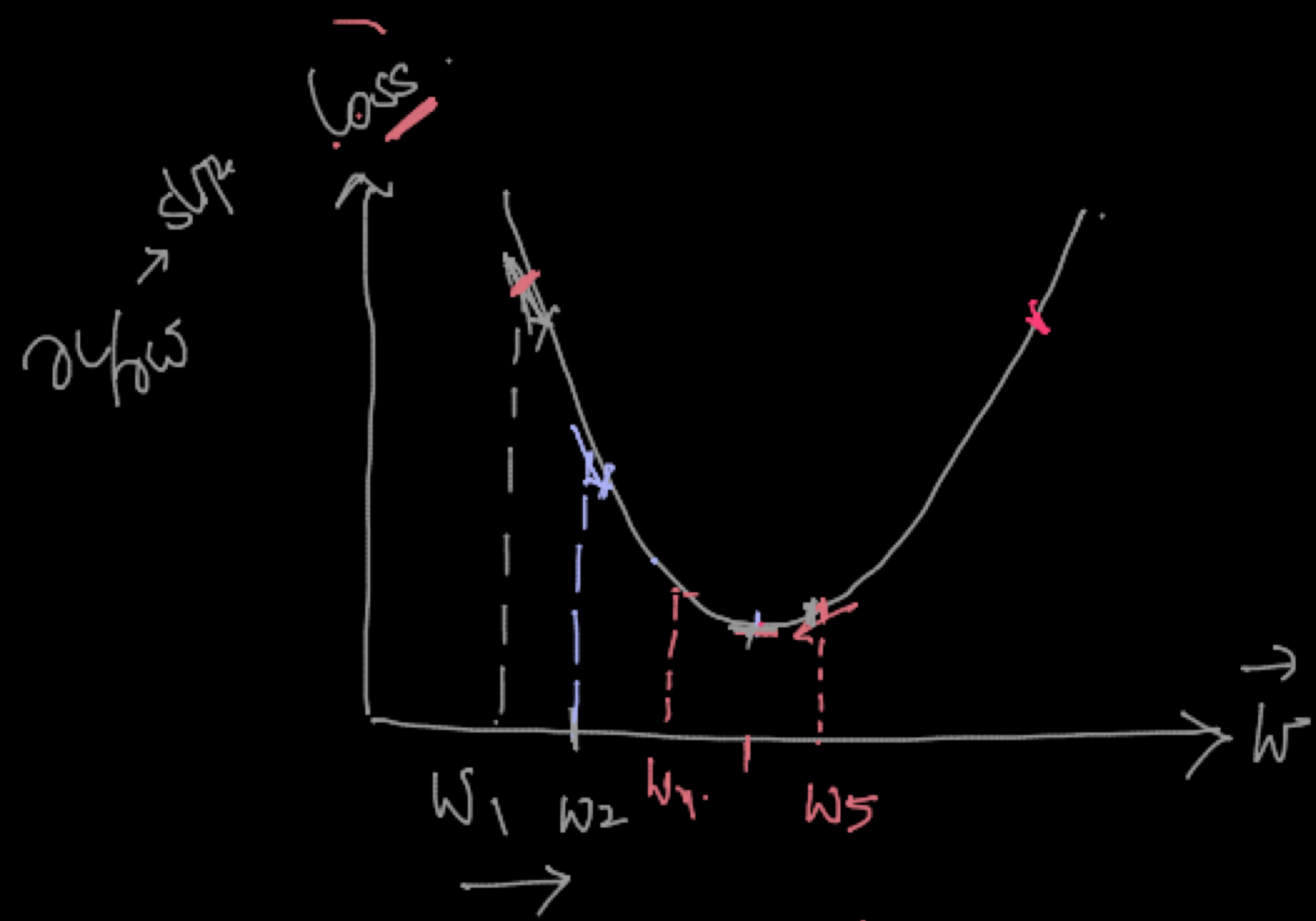
$$W_2 = W_1 - \Delta W$$

$$W_3 = W_2 + -\left(\frac{\partial L}{\partial W}\right)_{W_2} = W_2 - \Delta W$$

$$W_4 = W_3 - \left(\frac{\partial L}{\partial W}\right)_{W_3} = W_3 - \Delta W$$

$$W_6 = W_5 - \left(\partial L / \partial W\right)_{W_5} \rightarrow \doteq 0$$

$$W_6 = W_5$$



Step I : Randomly choose W_1

Repeat until convergence

Step 2 : Find the gradient of the loss fn at that W.

Step 3 : Update the Weights.

$$W_{new} = W_{old} + \left[-\partial L/\partial W\right]_{W_{old}}$$

← ΔW →

Convergence ⇒ $\boxed{\Delta W \approx 0 \pm \delta w}$

Neural →

→ β_0, β_1 →

$$W_2 = W_1 + \left[-\frac{\partial L}{\partial W}\right]$$ ✓

$-(-\nabla L)$

← ΔW →

$$W_2 = W_1 + \Delta W.$$

Loss fn ⇒ Hinge loss

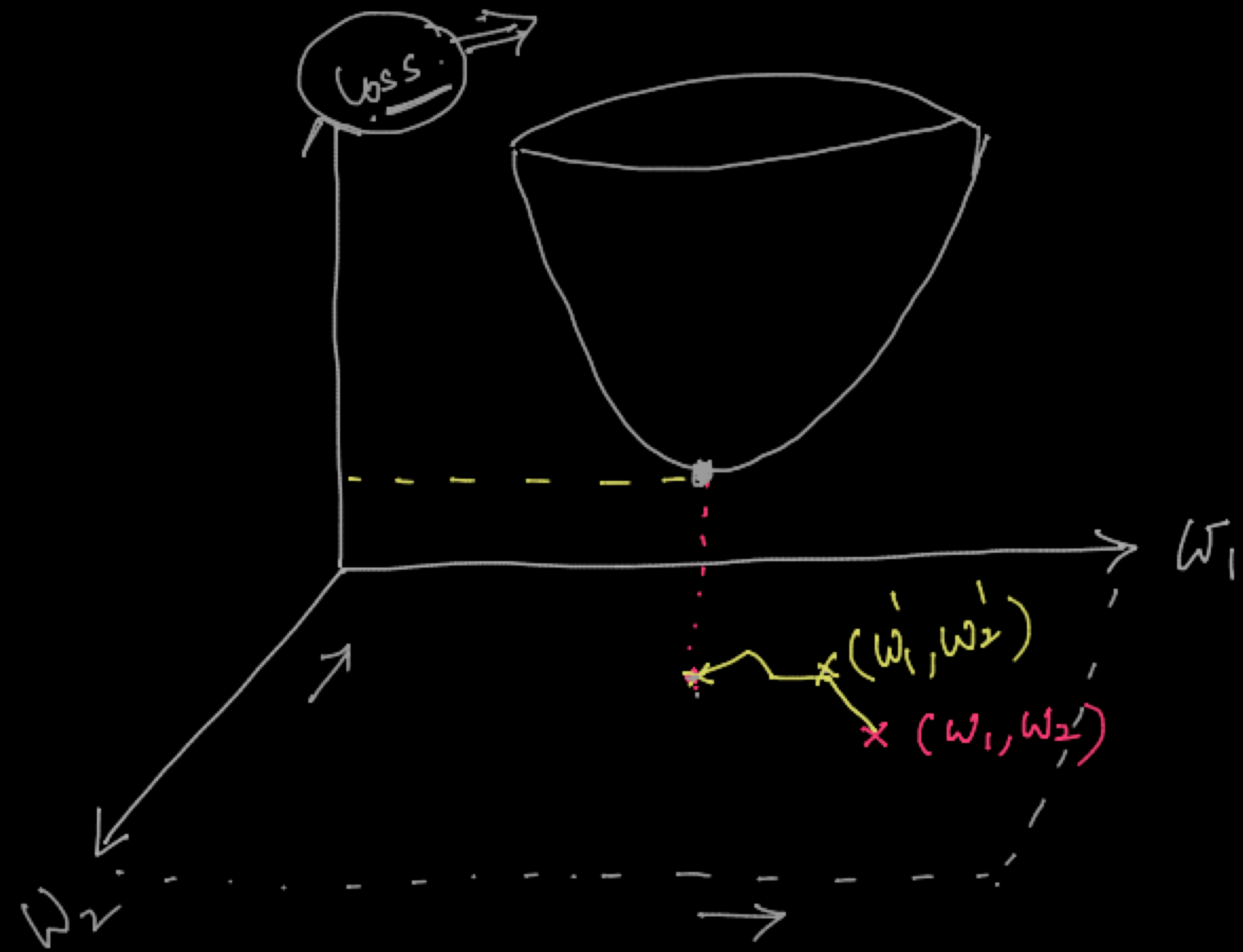
statsmodels.api.fn → ols

Sklearn LinearRegression → Gradient desc.

model = LinearReg()

x | y

[max_iterations = 400]



$w_1, w_2 \rightarrow$ Randomly choose

Update the weights simultaneously.

$$w_1' = w_1 + \left[-\partial L/\partial w_1\right]_{w_1} \rightarrow x_1 \quad x_2 \quad y$$

$$w_2' = w_2 + \left[-\partial L/\partial w_2\right]_{w_2}$$

$(w_1', w_2', w_3', \dots w_{100}')$

$(w_1, w_2, w_3 \dots w_{100})$

$x_1 \quad x_2 \quad x_3 \quad \dots \quad x_d$

$x_0 = 1$

$$y = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

$$y = w_1 x_1 + w_2 x_2 + w_3 x_3 + \dots w_d x_d$$

Logloss.
L.

Any → SVM → .
→ . Logist.
Linear Reg.
Decision.

⟹ Minimize / maximize Obj. fn
— find the model param
Which will give you minimum.

W's → 'Hinge loss' ✓
↳ Gradient.