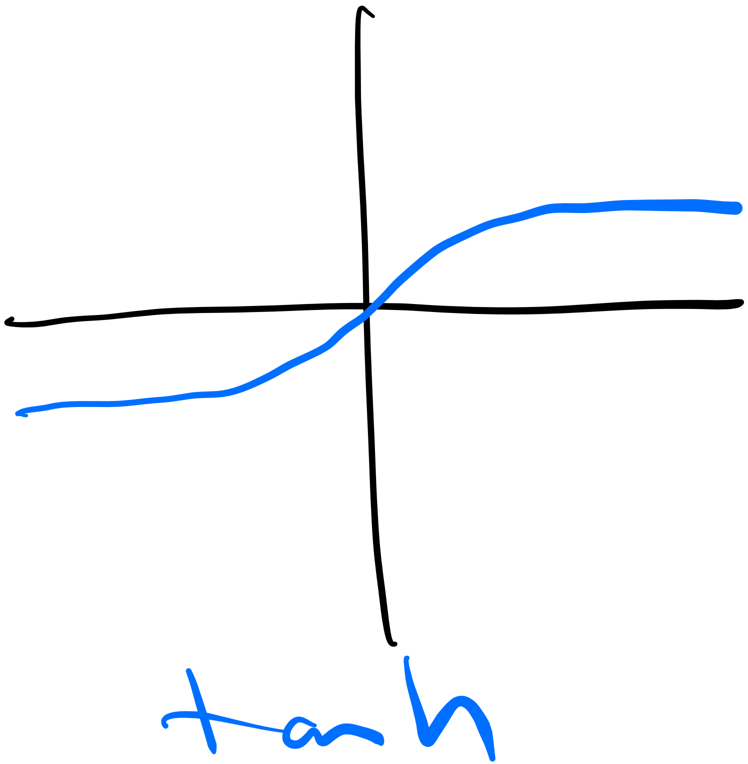
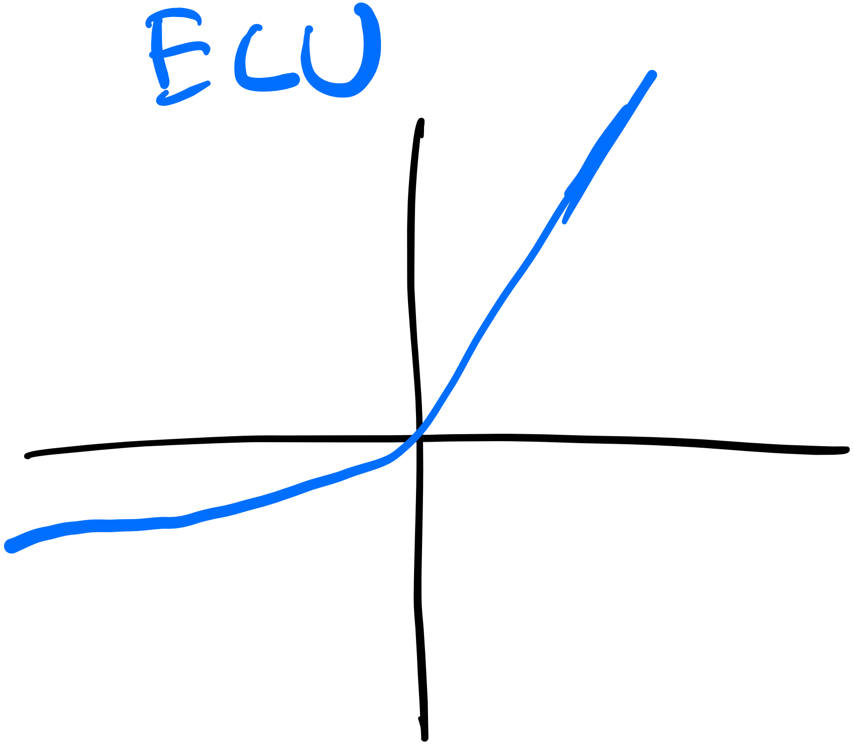
Activation functions





Neural Net Model

a → activation.

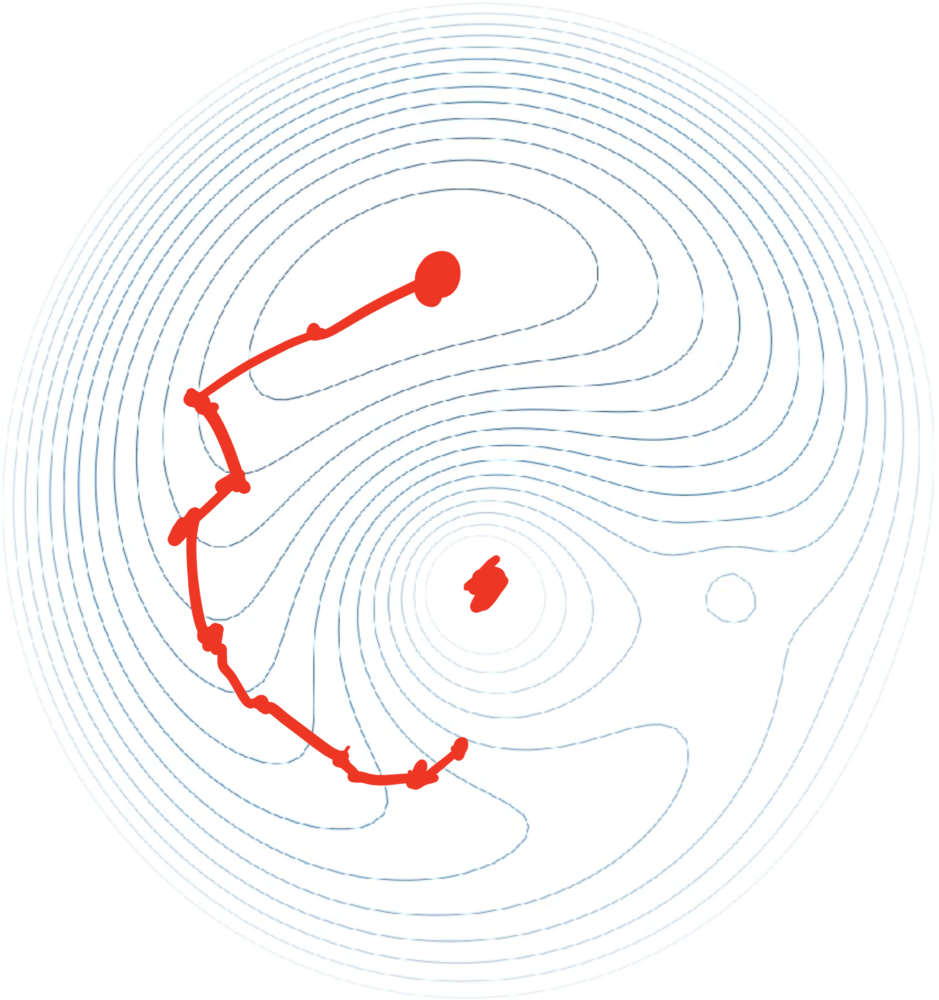
linear layer → $l = Wx + b$

$$\hat{y}(x;\theta) = \left(l^{(n)} \cdots a^{(2)} \circ l^{(2)} \circ a^{(1)} \circ l^{(1)}\right)(x)$$

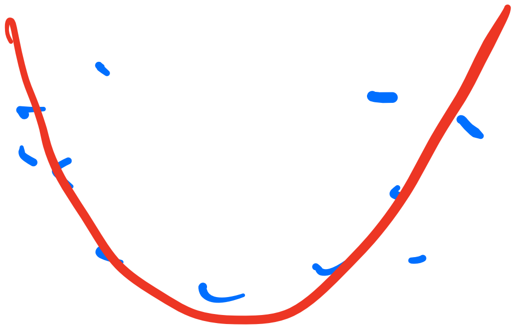
(unconstrained) Optimization

minimize $L(\theta)$

with respect to θ



Train Model: Objective or Loss Function



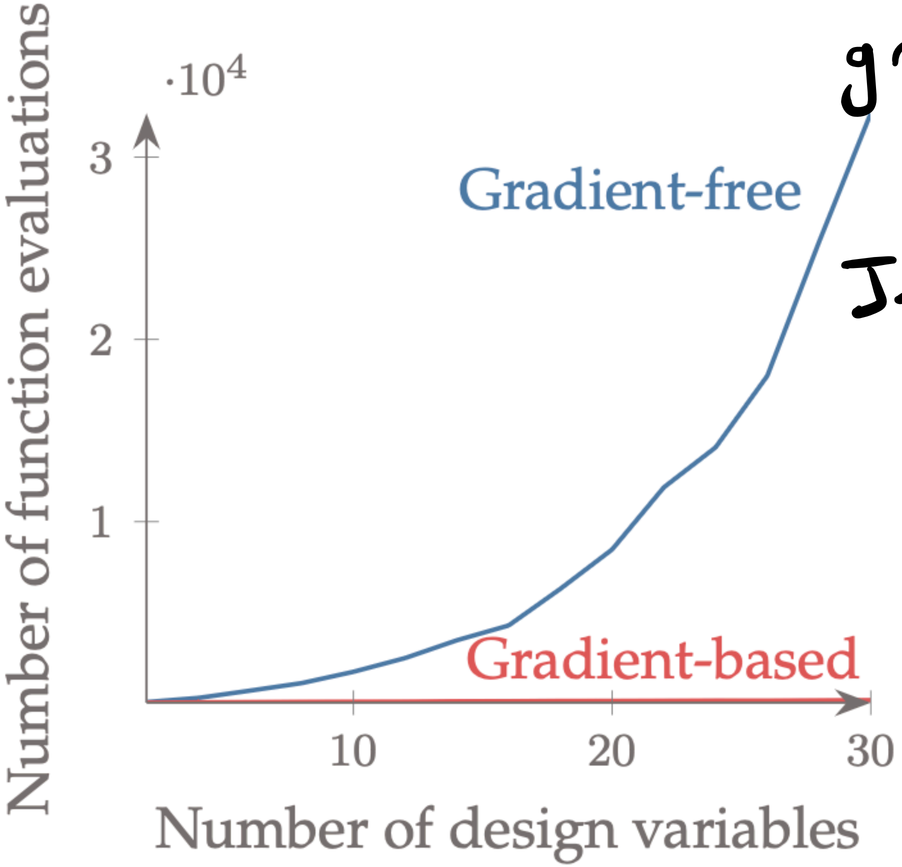
$$L(\theta) = \frac{1}{m}\sum_{i=1}^{m} \lambda(\hat{y}_i, y_i)$$

Mean square error:

$$\Rightarrow L = \frac{1}{m}\sum_{i=1}^{m} (\hat{y}_i - y_i)^2$$

$$\lambda(\hat{y}, y) = (\hat{y}_i - y_i)^2$$

Gradient-based optimization



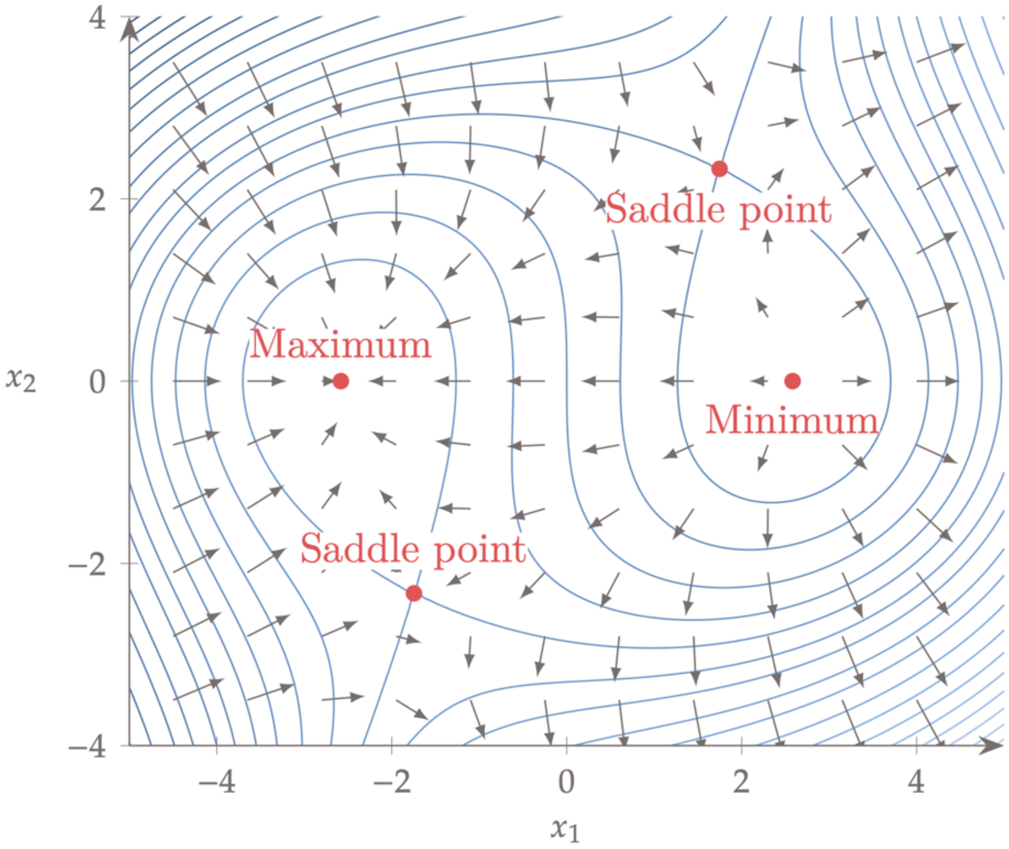
derivative. $\frac{df}{dx}$

gradient $\frac{df}{dx_i}$

Jacobian $\frac{df_i}{dx_j}$

Backpropagation

Gradient Descent



Gradient-Based Optimization

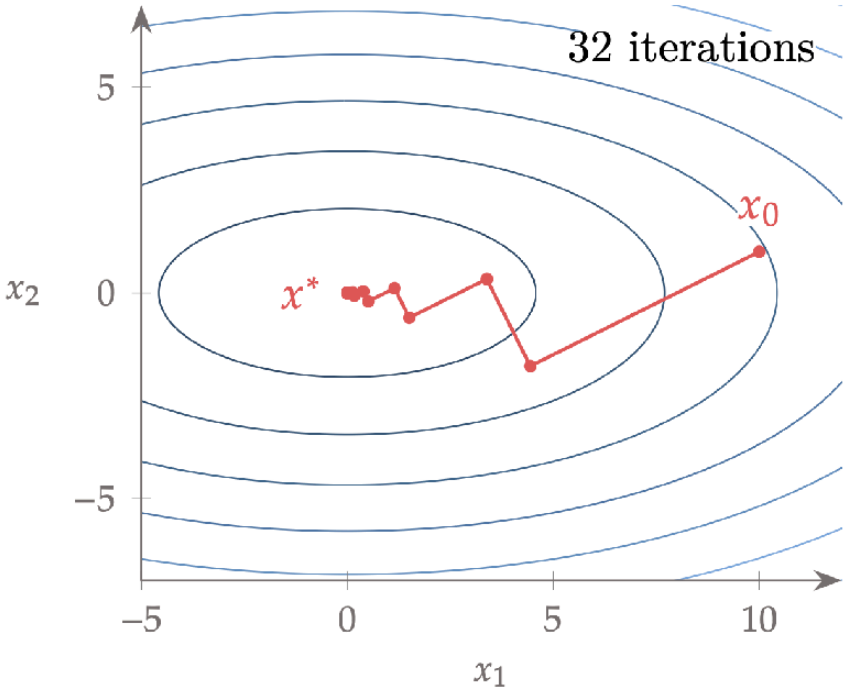
direction.

$$\theta_{k+1} = \theta_k + \alpha p_k$$

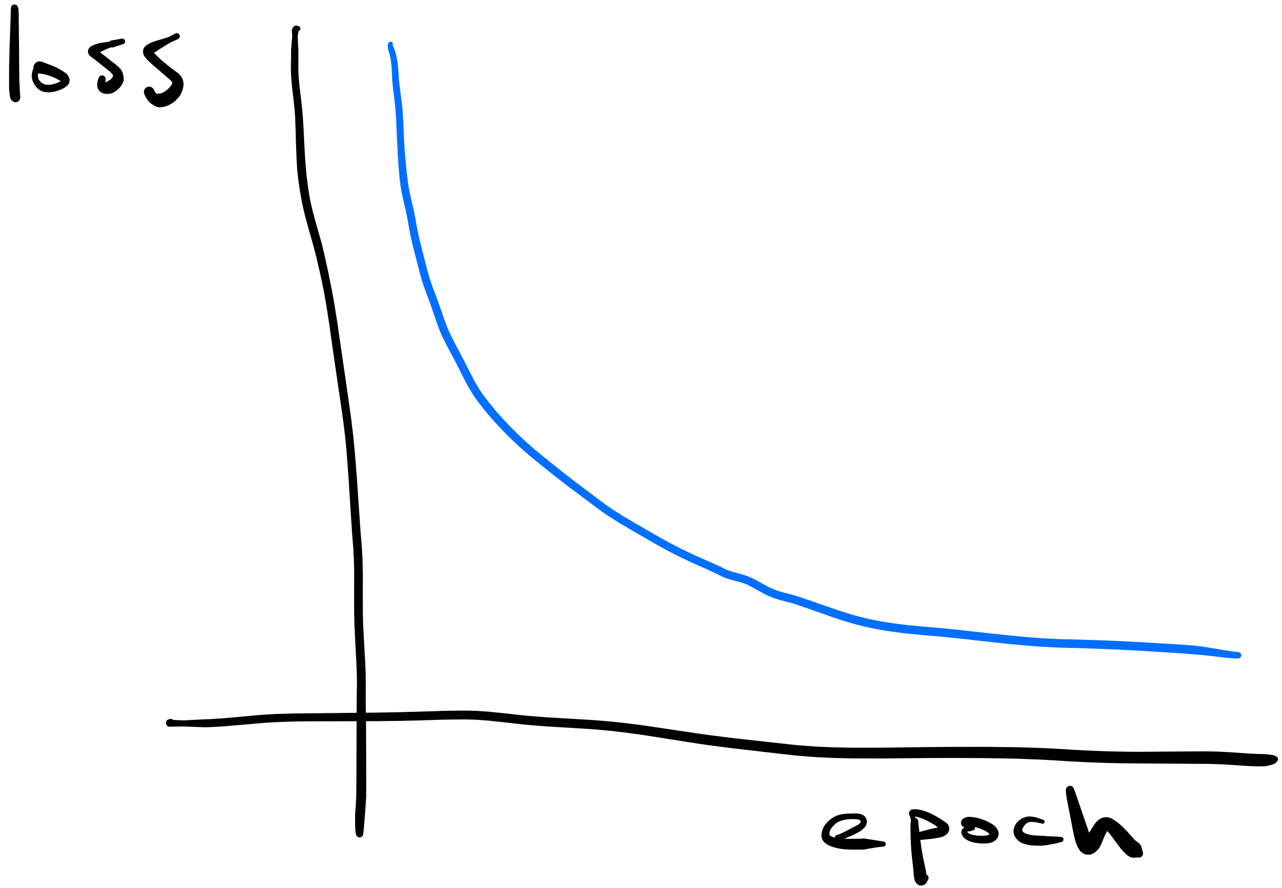
learning rate

$$\begin{bmatrix} \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix} + \alpha \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

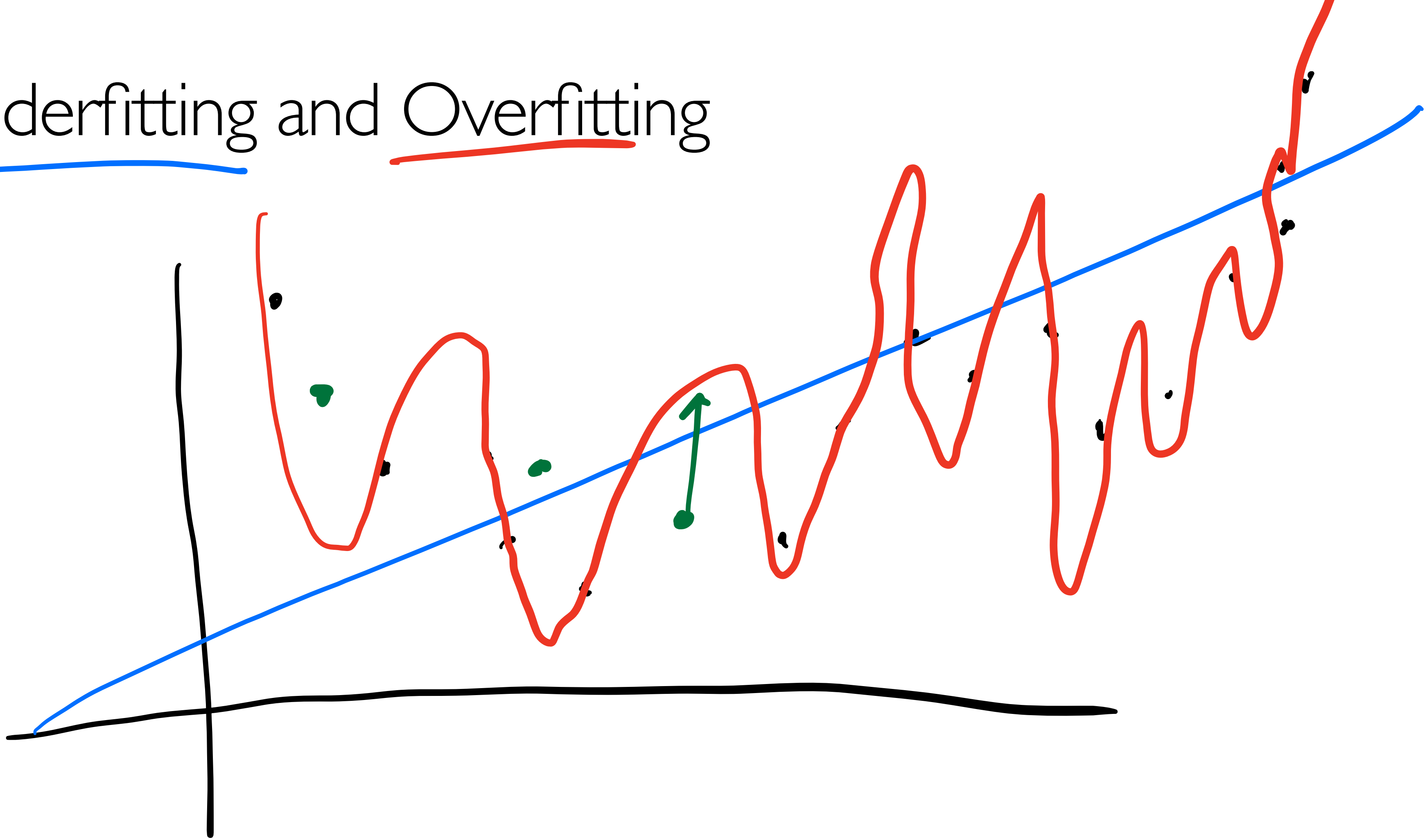
Gradient Descent (or Steepest Descent)



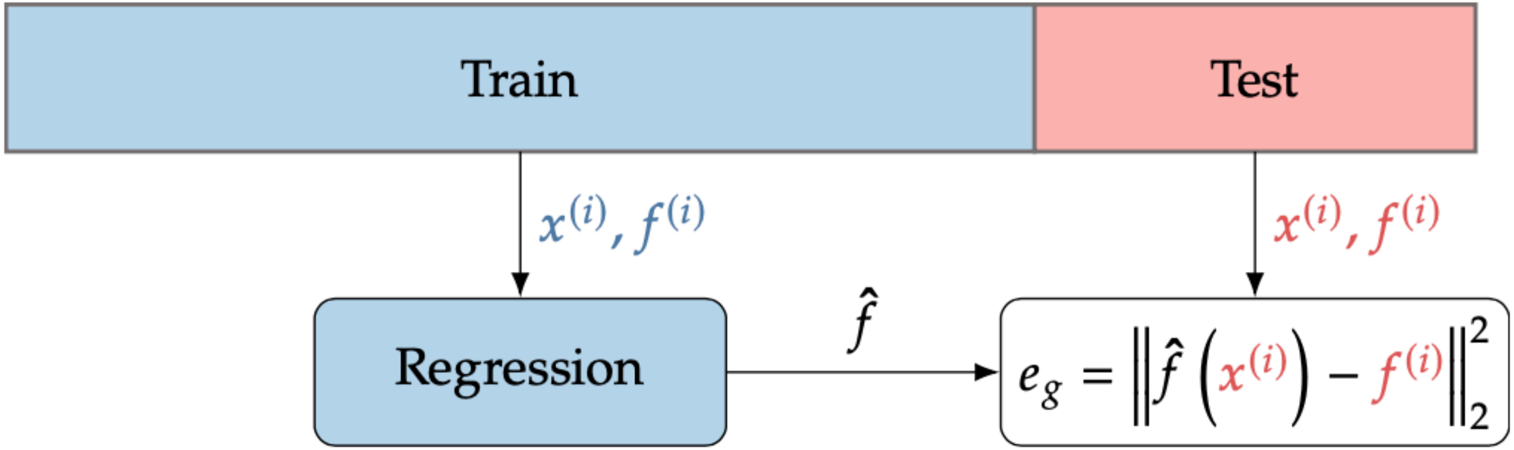
Convergence



Underfitting and Overfitting



Cross Validation

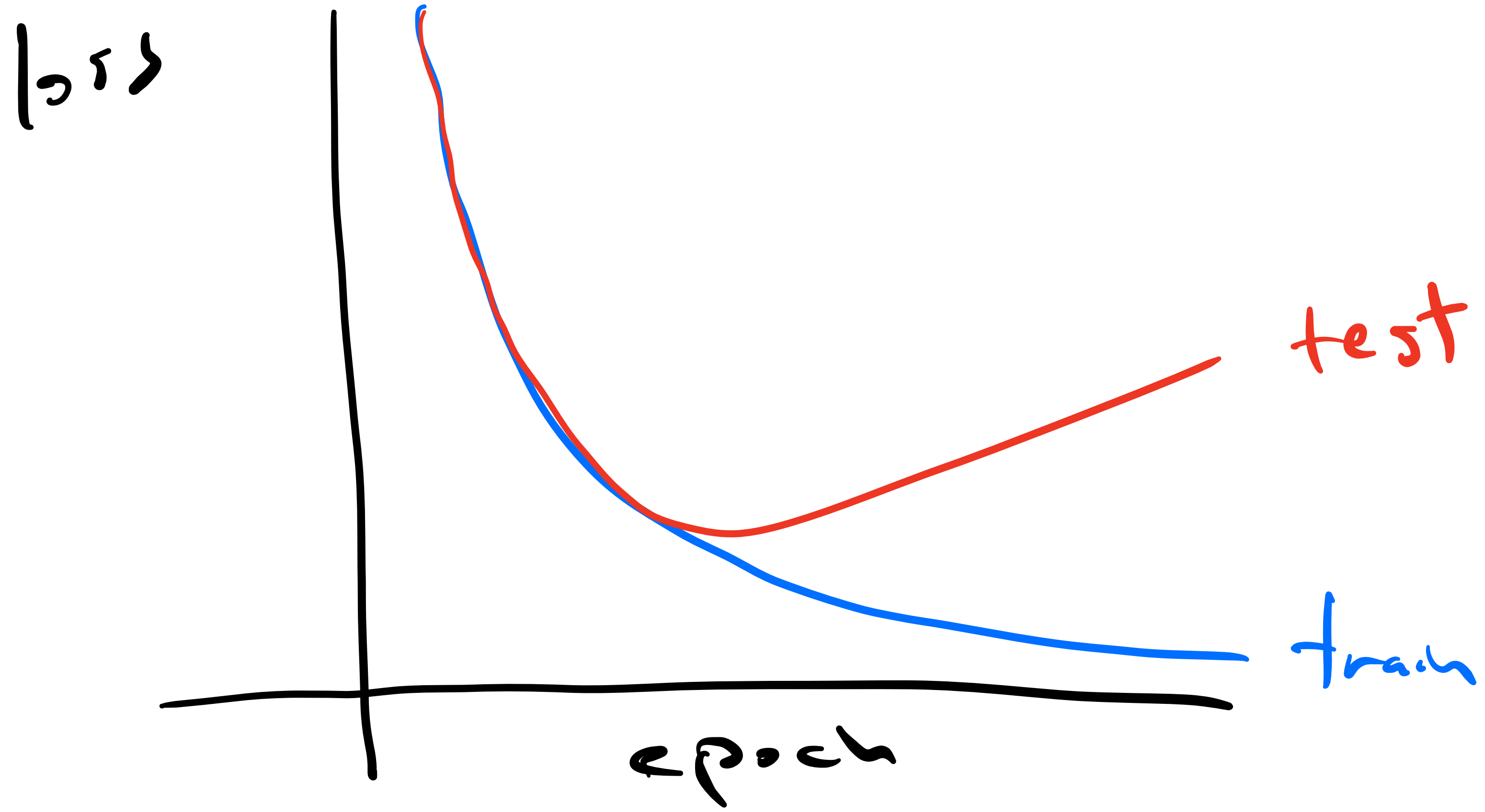


Train, Test, Validation

optimize θ

"optimize" hyperparameters

Underfitting and Overfitting

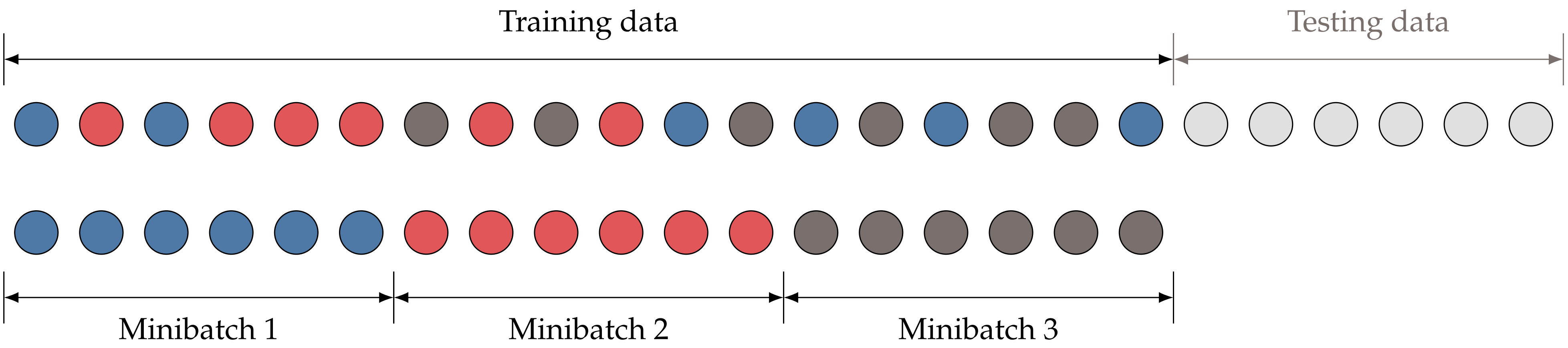


Batching

$$L = \frac{1}{m}\sum_{i=1}^{m} \lambda_i(\theta) = E(\lambda(\theta))$$

$$\frac{dL}{d\theta} = \frac{1}{m}\sum_{i=1}^{m} \frac{d\lambda_i}{d\theta} = E\left(\frac{d\lambda}{d\theta}\right)$$

$$\frac{dL}{d\theta} \approx \frac{1}{n}\sum_{i=1}^{n} \frac{d\lambda_i}{d\theta} \qquad n \ll m$$


Training data
Testing data
Minibatch 1
Minibatch 2
Minibatch 3