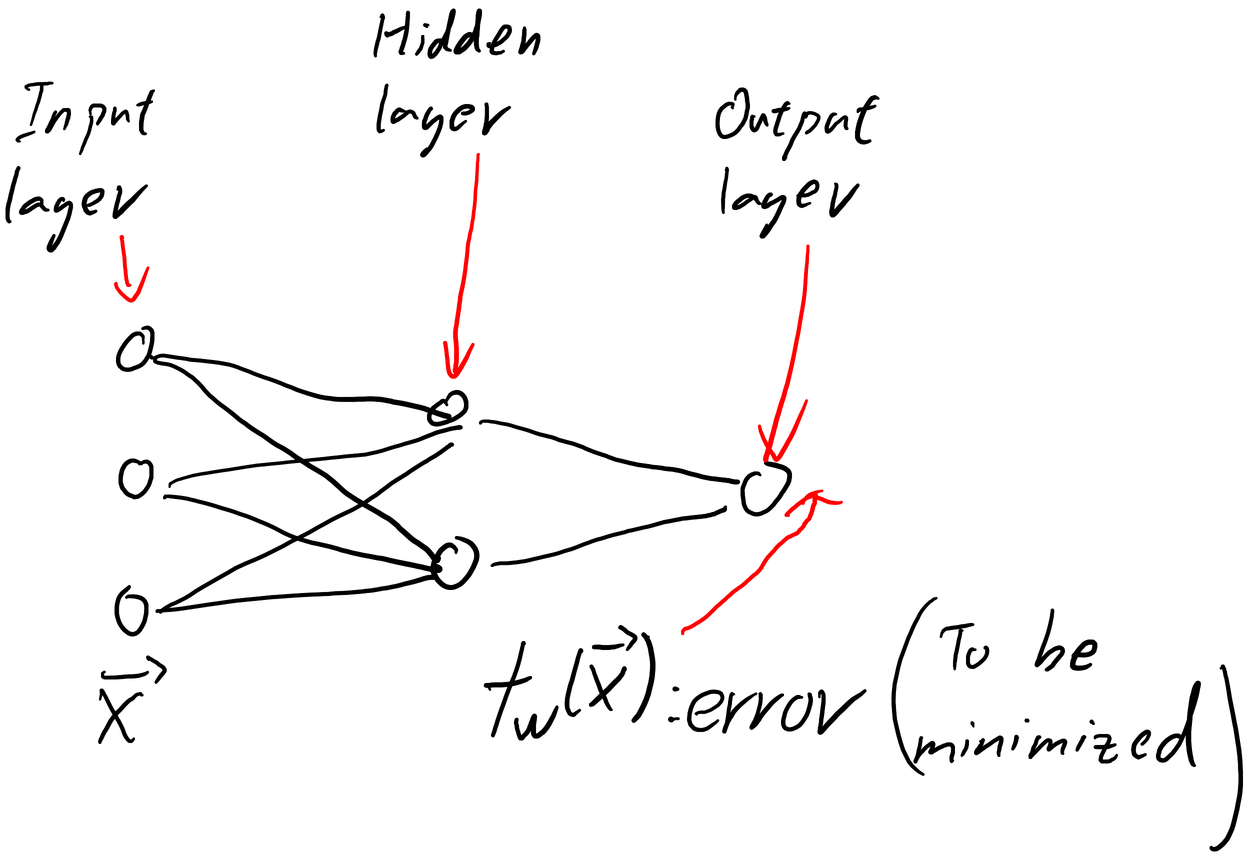
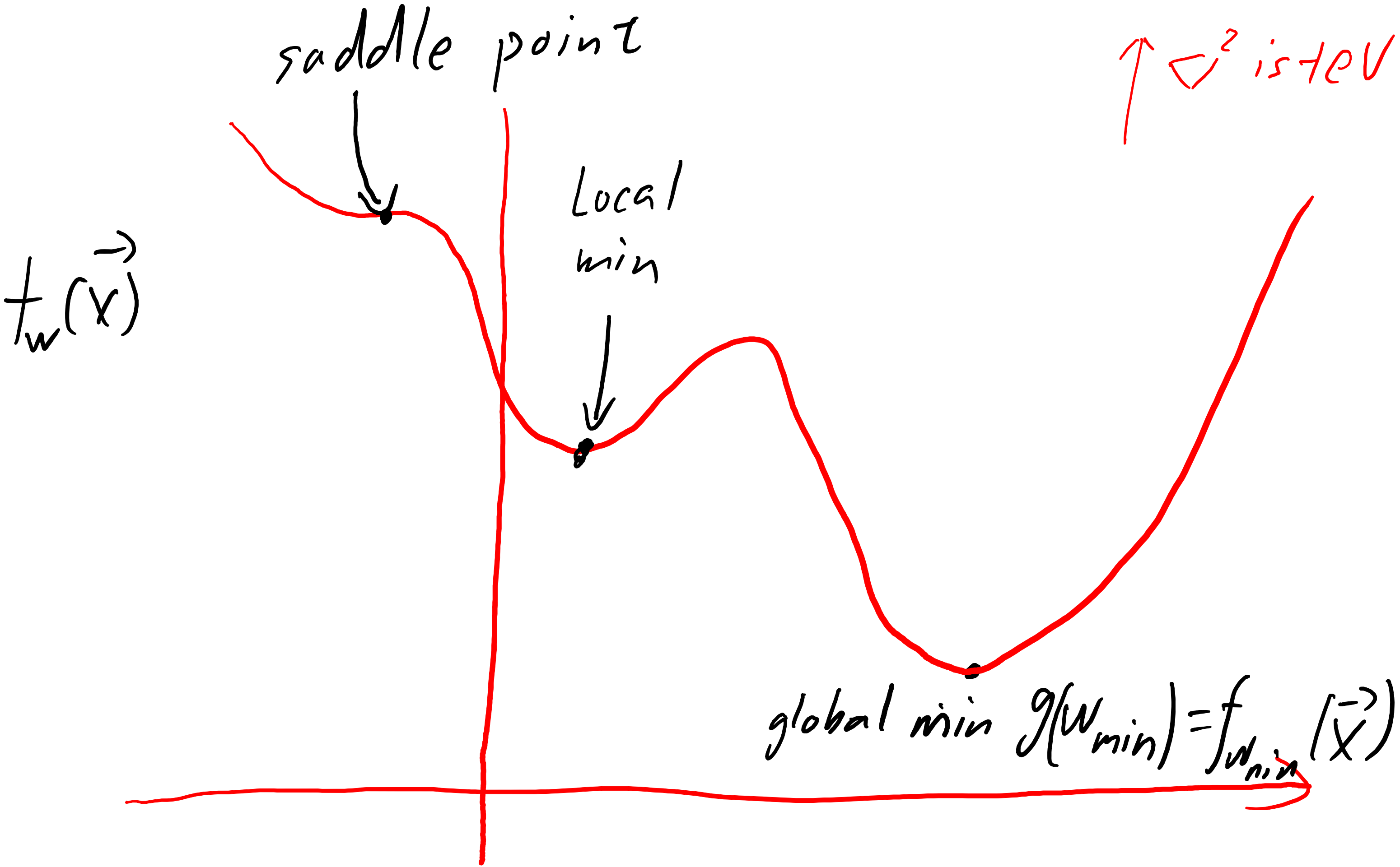
1/26 EC500K



Goal: minimize $f_w(\vec{x}) = g(\vec{w})$ w.r.t. $\vec{w}$



$\nabla g(\vec{w}_{min}) = 0$ at those three points

To find $\vec{w}_{min}$; Use GD

$$\vec{w}_{t+1} = \vec{w}_t - \epsilon \cdot \nabla g(\vec{w}_t)$$

Issue:
- cold start
- ϵ
- local min

Solution: adaptive learning rate

line search

→ Optimization methods:

- zero order:

  evaluate $g(\vec{w})$ at any $\vec{w}$

- 1st order:

  GD, KKT

- 2nd order:

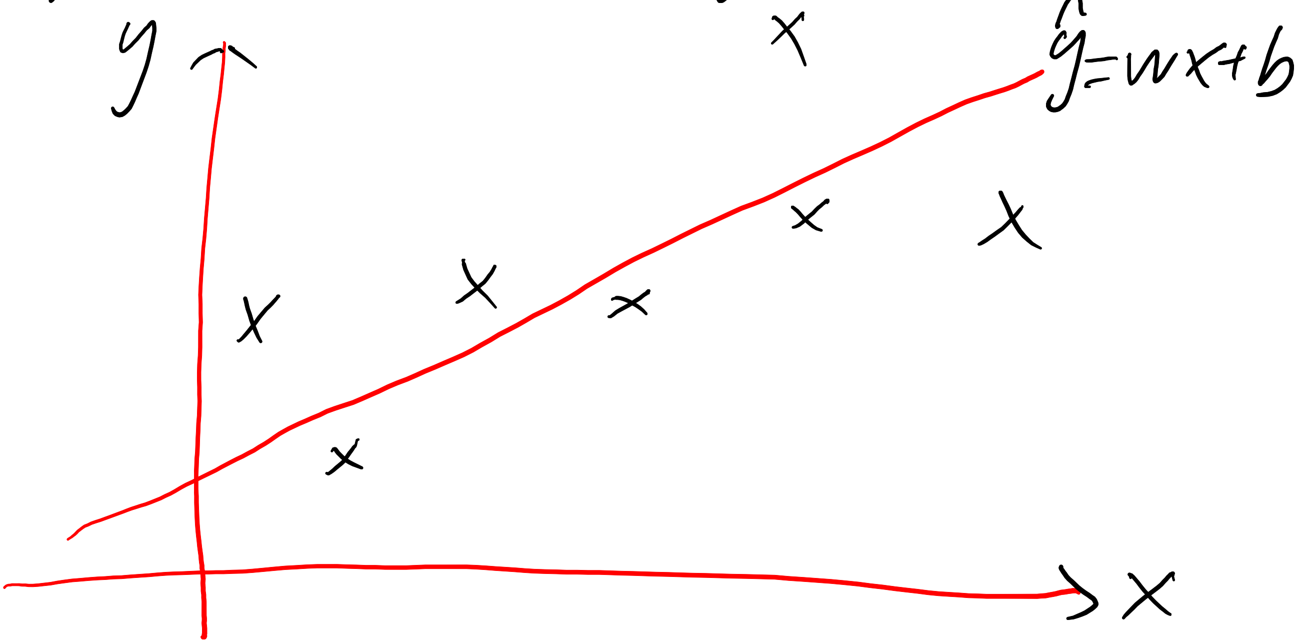
  Newton's method

  (Quasi-Newton's method)

  $$\vec{W}_{t+1} = \vec{w}_t - \epsilon_t \cdot \nabla^2 g(\vec{w}_t)^{-1} \nabla g(\vec{w}_t)$$

→ Example: Linear regression

1-D



Training set $(x_1, y_1) \cdots\cdots (x_n, y_n)$

Higher dimension:

Least square error

$$\sum_{i=1}^{n} \left(\vec{y}_i - \vec{w}^T \vec{x}_i \right)$$

Note: LR actually has closed form solution

Using GD:

At time t:

$$\nabla g(\vec{w}) = -\sum \left(y_i - \vec{w}^T \vec{x}_i\right) \cdot \vec{x}_i$$

Update $\vec{x}_{t+1}$

→ Error func

- It's usually decomposed into sum of data

$$g(\vec{w}) = \sum g_i(w_i \cdot \vec{x}_{ij} \cdot y_i)$$

- It is costly to compute in high dimension

→ SGD:

- Only compute part of $g(\vec{w})$

$$\vec{W}_{t+1} = \vec{W}_t - \epsilon_t \nabla g_i(\vec{w}_t), \text{ for some } i \in \text{range}(n)$$

- In the LR example, SGD can do n iterations in the time of $\underline{1}$ GD update

- Minibatch update

→ Regularization

Again LR example:

With L2 norm regulation:

Error func $g(\vec{w}) = \sum (y_i - \vec{w}^T x_i)^2 + \frac{1}{2} \lambda \|\vec{w}\|_2^2$