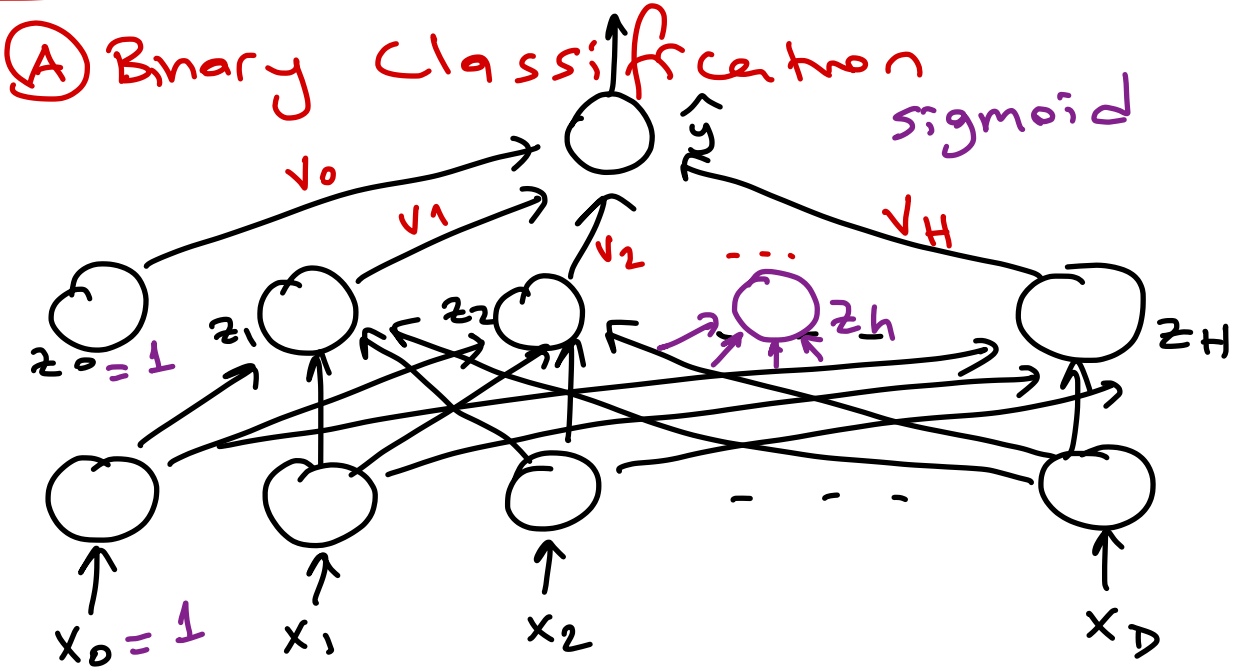
Multilayer Perceptrons

(A) Binary Classification



$$\hat{y}_i = \text{sigmoid}(v^T \cdot z_i)$$

$$z_{ih} = \text{sigmoid}(w_h^T \cdot x_i)$$

sigmoid

weights for all incoming edges to z_h

→ binary cross-entropy

$$\text{Error}_i = -\left[y_i \log(\hat{y}_i) + (1-y_i) \log(1-\hat{y}_i) \right]$$

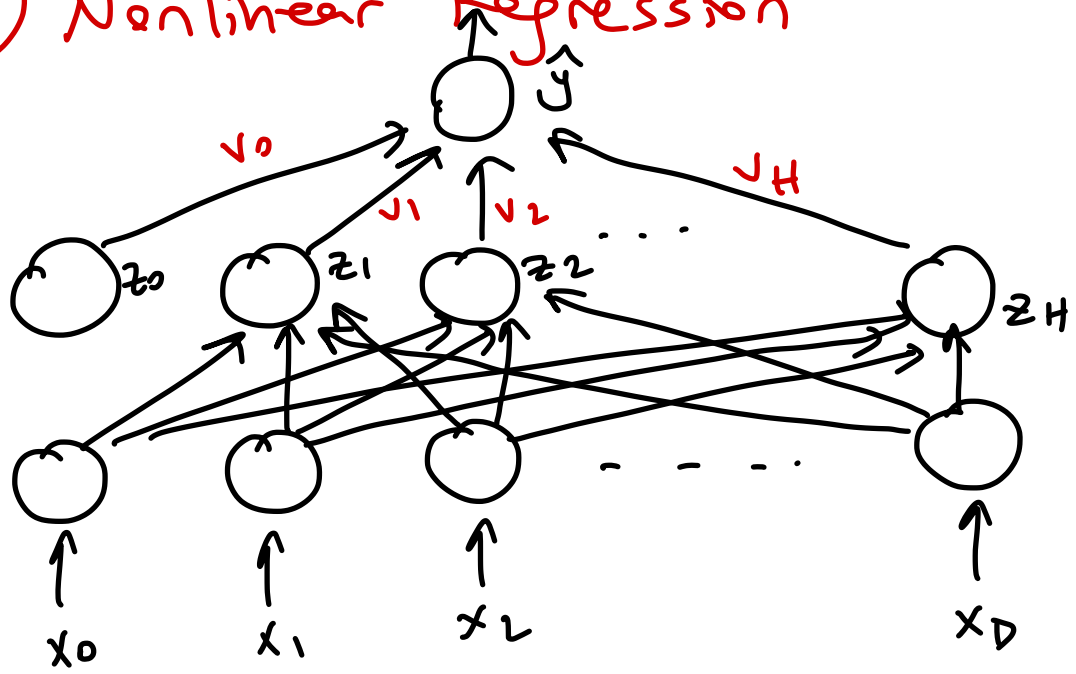
$$\Delta v_h = \eta \cdot (y_i - \hat{y}_i) \cdot z_{ih}$$

$$\Delta w_{hd} = \eta \, (y_i - \hat{y}_i) \, v_h \cdot z_{ih} (1 - z_{ih}) \cdot x_{id}$$

y_i's are either 0 or 1

$\hat{y}_i$'s are between 0 and 1.

(B) Nonlinear Regression



identity $\hat{y}_i = v^T \cdot z_i$

sigmoid $z_{ih} = \text{sigmoid}(w_h^T \cdot x_i)$

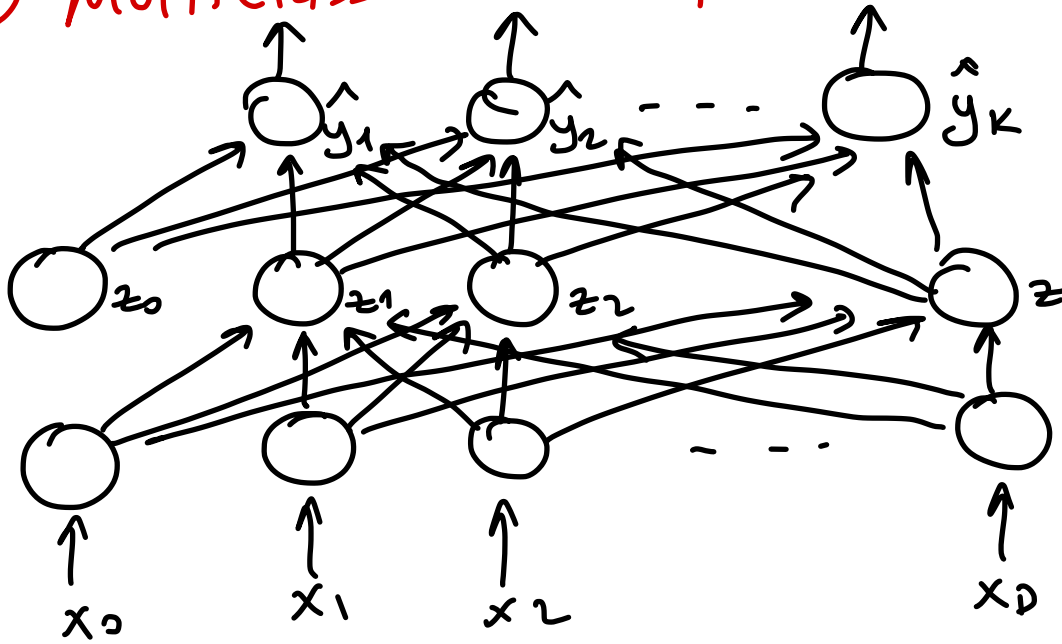
$$\text{Error}_i = \frac{1}{2}(y_i - \hat{y}_i)^2$$

$$\Delta v_h = \eta \cdot (y_i - \hat{y}_i) \cdot z_{ih}$$

$$\Delta w_{hd} = \eta (y_i - \hat{y}_i) \, v_h \, z_{ih}(1 - z_{ih}) \, x_{id}$$

y_i's and $\hat{y}_i$'s are real numbers.

© Multiclass Classification



softmax

$$\hat{y}_{ic} = \frac{\exp\left[v_c^T \cdot z_i\right]}{\sum_{d=1}^{K} \exp\left[v_d^T \cdot z_i\right]}$$

sigmoid

$$z_{ih} = \text{sigmoid}\left(w_h^T \cdot x_i\right)$$

$$\text{Error}_i = -\sum_{c=1}^{K} y_{ic} \log\left(\hat{y}_{ic}\right)$$

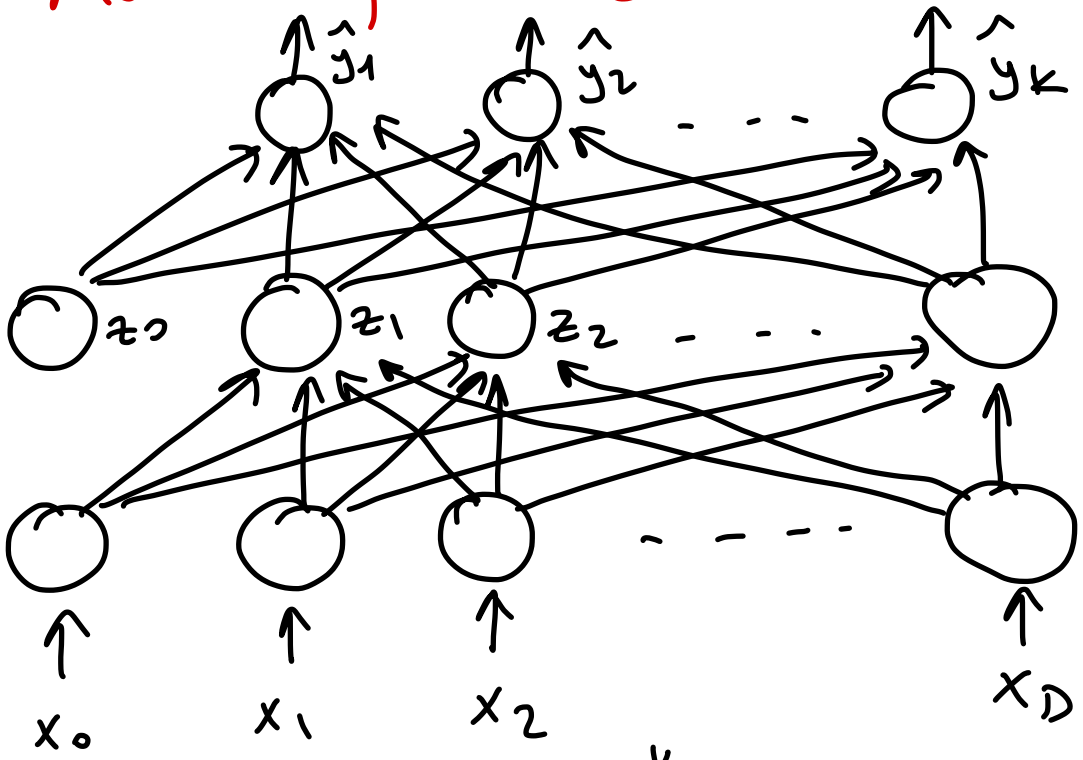
$$\Delta v_{ch} = \eta \left(y_{ic} - \hat{y}_{ic}\right) \cdot z_{ih}$$

$$\Delta w_{hd} = \eta \left[\sum_{c=1}^{K} \left(y_{ic} - \hat{y}_{ic}\right) \cdot v_{ch}\right] \cdot z_{ih}\left(1 - z_{ih}\right) \cdot x_{id}$$

y_{ic}'s are either 0 or 1.

$\hat{y}_{ic}$'s are between 0 and 1.

⑮ Multi-output Regression



identity $\quad \hat{y}_{ic} = v_c^T \cdot z_i$

sigmoid $\quad z_{ih} = \text{sigmoid}(w_h^T \cdot x_i)$

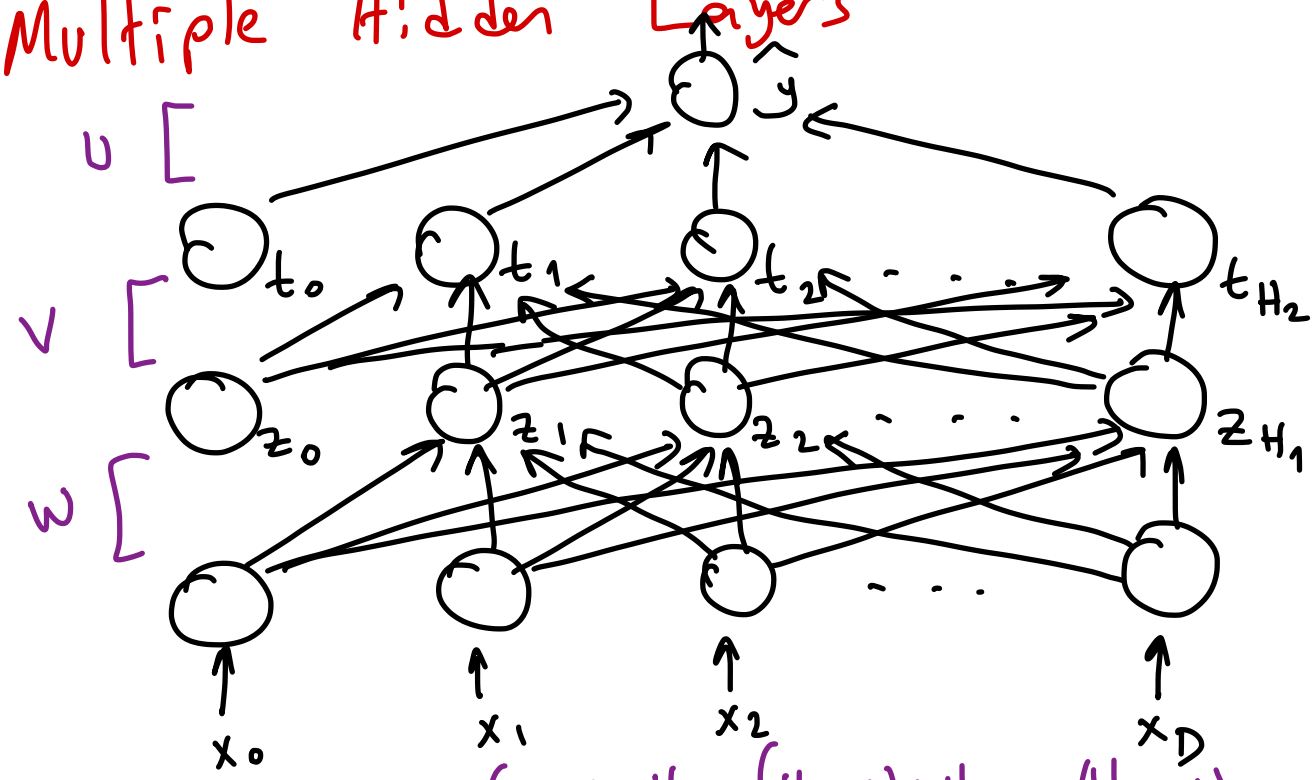
$$\text{Error}_i = \frac{1}{2} \sum_{c=1}^{K} (y_{ic} - \hat{y}_{ic})^2$$

$$\Delta v_{ch} = \eta (y_{ic} - \hat{y}_{ic}) \cdot z_{ih}$$

$$\Delta w_{hd} = \eta \left[\sum_{c=1}^{K} (y_{ic} - \hat{y}_{ic}) \cdot v_{ch} \right] \cdot z_{ih} (1 - z_{ih}) \cdot x_{id}$$

y_{ic}'s and $\hat{y}_{ic}$'s are real numbers.

Multiple Hidden Layers



$$\hat{y}_i = \text{sigmoid}(U^T \cdot t_i)$$

$$t_{ih} = \text{sigmoid}(V_h^T \cdot z_i)$$

$$z_{ih} = \text{sigmoid}(W_h^T \cdot x_i)$$

# of parameters $= \underbrace{(D+1)\times H_1}_{W} + \underbrace{(H_1+1)\times H_2}_{V} + \underbrace{(H_2+1)}_{U}$

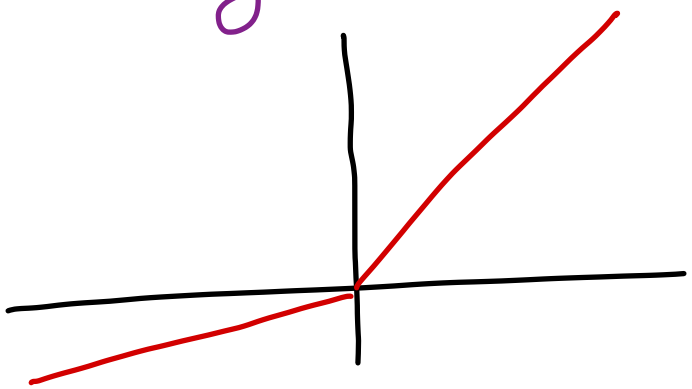
"vanishing gradients"
↳ approaching to 0

Rectified Linear Unit (ReLU)

$$s(a) = \begin{cases} a & \text{if } a > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$s'(a) = \begin{cases} 1 & \text{if } a > 0 \\ 0 & \text{otherwise} \end{cases}$$

Leaky ReLU



$$s(a) = \begin{cases} a & \text{if } a > 0 \\ \alpha a & \text{otherwise} \end{cases}$$

$$s'(a) = \begin{cases} 1 & \text{if } a > 0 \\ \alpha & \text{otherwise} \end{cases}$$

usually $\alpha = 0.01$

TRAINING PROCEDURES

Momentum:

$$s_h^{(t)} = \alpha \underbrace{s_h^{(t-1)}}_{\text{storage variable of the previous step}} + (1-\alpha) \underbrace{\frac{\partial \text{Error}^{(t)}}{\partial w_h}}_{\text{gradient of the current step.}}$$

memory — current opinion

$$\Delta w_h^{(t)} = -\eta s_h^{(t)}$$

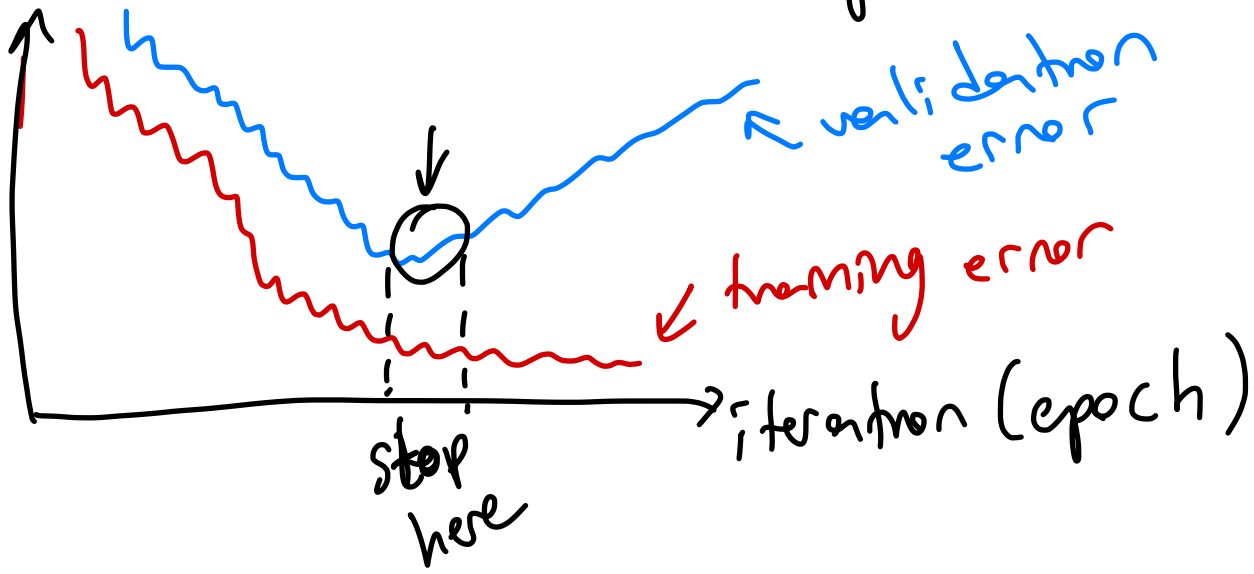
Adaptive Learning Rate: $\eta = ?$

→ starts with a large value

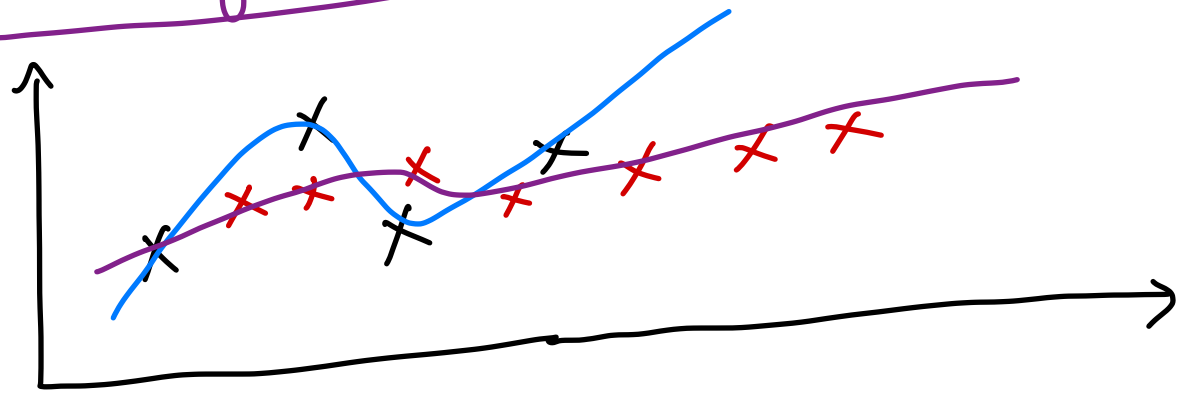
→ decrease if error increases

Early stopping:

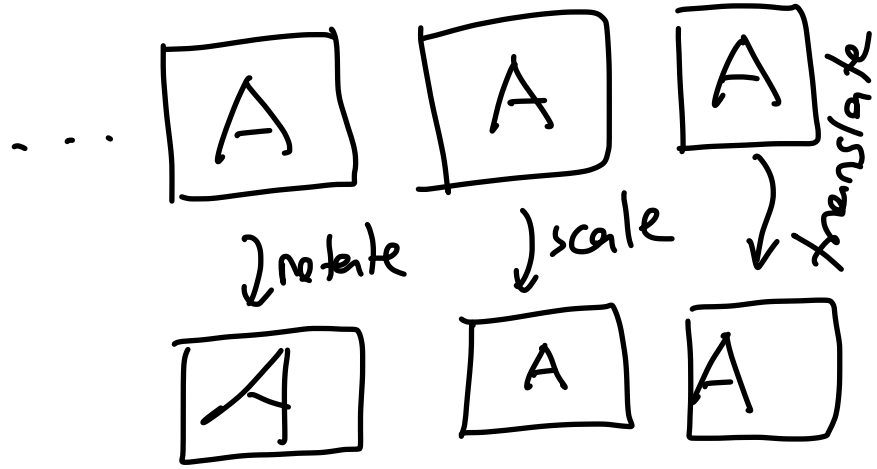
Stop if you think that your algorithm is overfitting.



Increasing Data Set Size:



Data augmentation



Weight Decay:

weight decay

$$Error' = Error + \frac{\lambda}{2} \sum_{h=1}^{H} w_h^2$$

l_2-norm regularization

$$\frac{\partial Error'}{\partial w_h} = \frac{\partial Error}{\partial w_h} + \lambda . w_h$$

$$\sum_{h=1}^{H} w_h^2 = \|w\|_2^2$$

$$\Delta w_h = -\eta \left[\frac{\partial Error}{\partial w_h} + \lambda w_h \right]$$