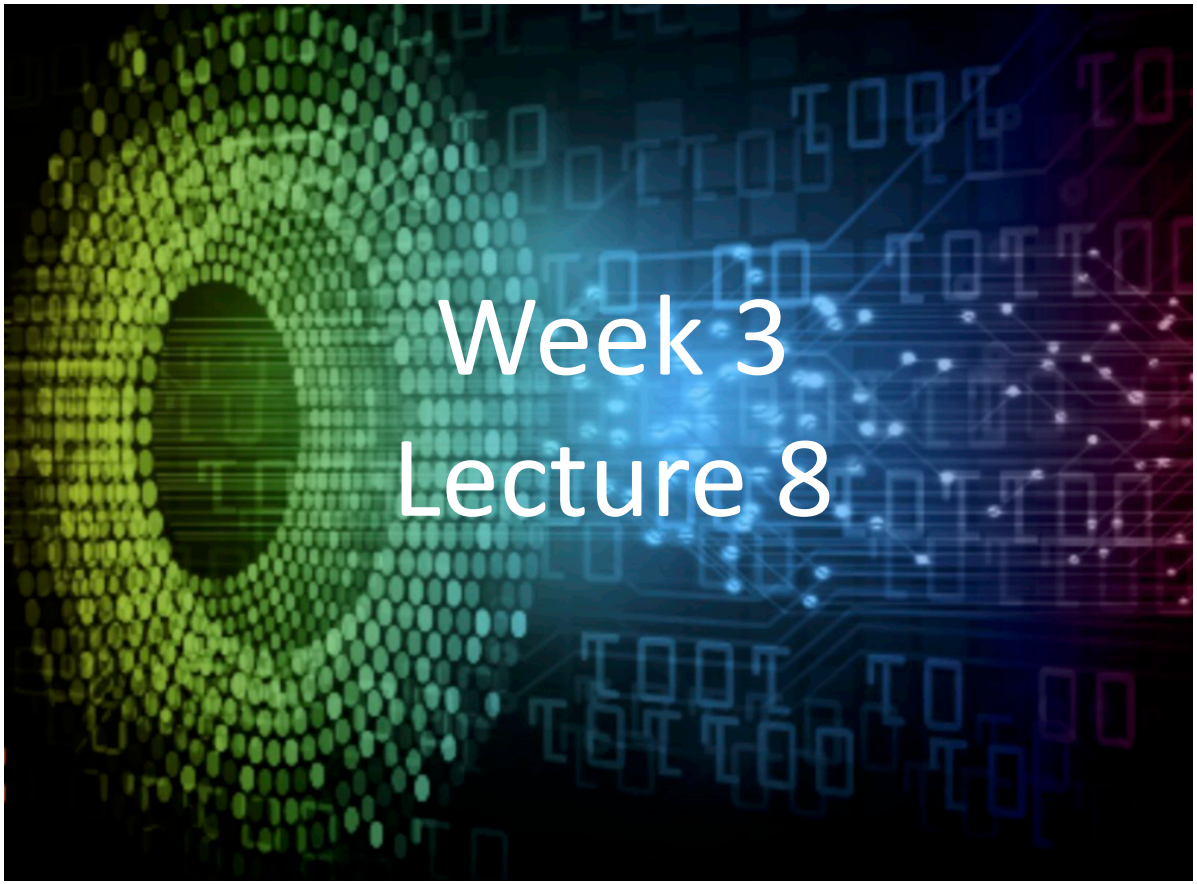
Introduction to Deep Learning
Applications and Theory



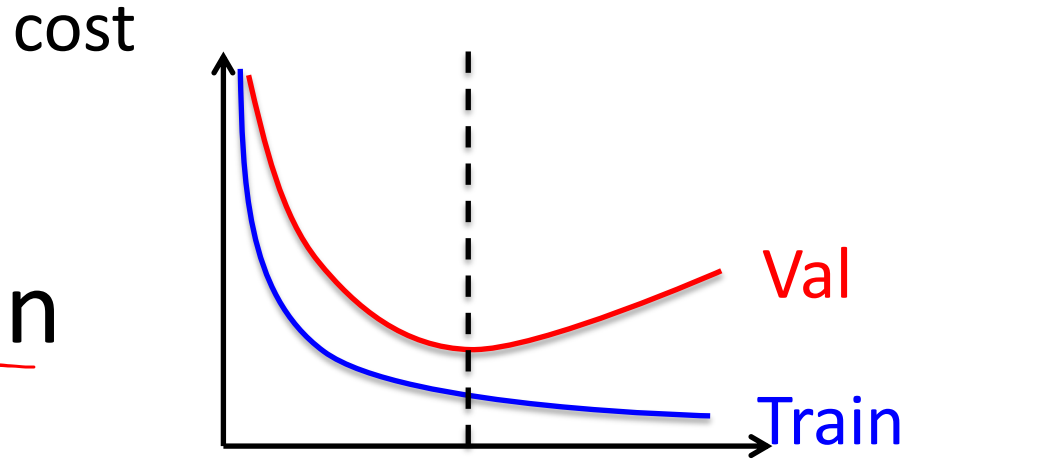
ECE 596 / AMATH 563

Previous Lecture: **Optimization of Stochastic Data**

1. Extensions of Gradient Descent for Stochastic Data

- optimization

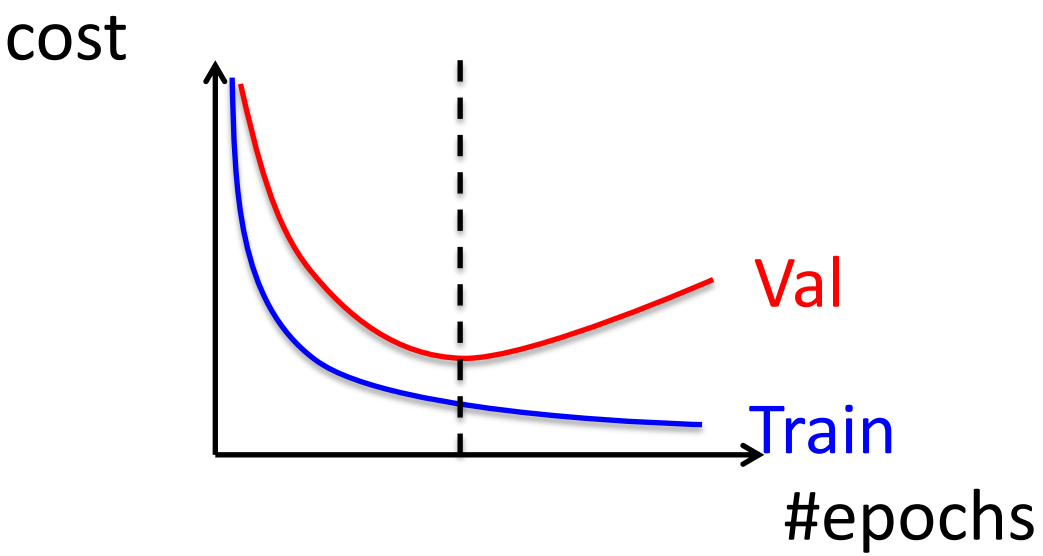
2. Cross Validation



- Data
≫ Overfitting
Under

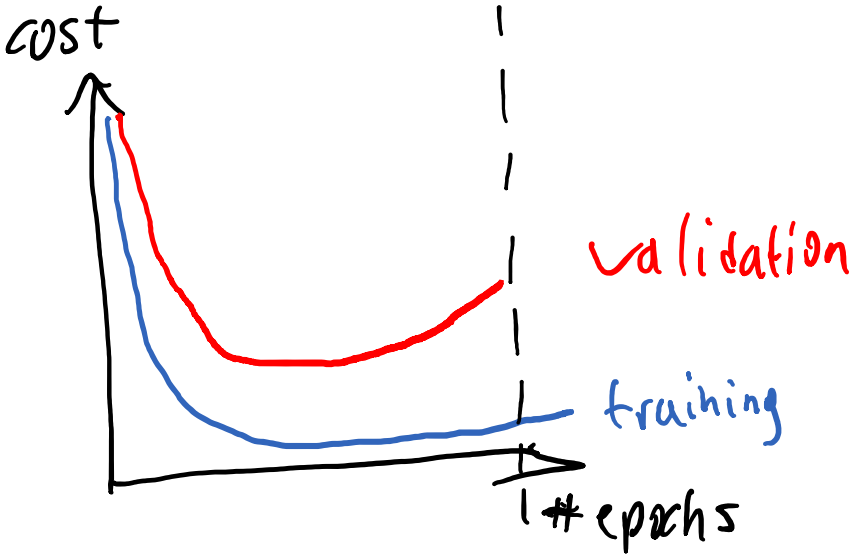
Current Lecture: **Regularization**

1. Regularization
   - L2/L1
   - Dropout
   - Additional



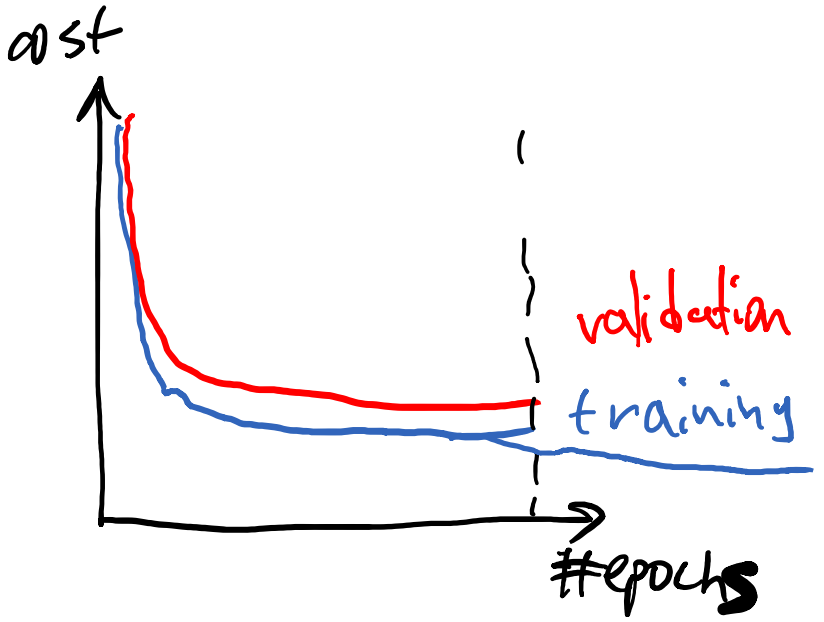
High Variance vs High Bias

Overfitting



High-variance

Underfitting



High-bias

High Variance vs High Bias

Overfitting

"Not enough data for parameters"

More Data

Regularization

Dropout

Initialization

Optimization solutions

Underfitting

"Not enough parameters"

More Layers/Nodes

Longer Training

Architecture — Deep vs. CNN

Hyperparam optim

Over- and Under-Determined

$$\vec{p}^* = \arg\min_{\vec{p}} L(\vec{p}) \qquad A\vec{p} = \vec{b}$$

Over-determined:

$$\arg\min_{\vec{p}} \left(||A\vec{p} - \vec{b}||_2 + \lambda g(p) \right)$$

objective regularizer

$\vec{p}$

$g(p) = \|p\|_2$ <<<

$= \|p\|_1$

Regularization

Add regularization term to the cost

L2 or L1

For single layer

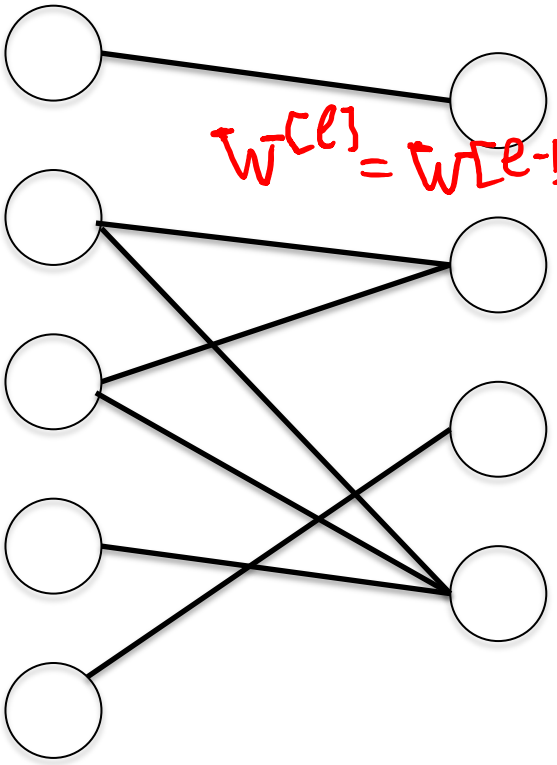
$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m} ||\vec{w}||_2^2$$

$$+ \frac{\lambda}{m} ||\vec{w}||_1$$

MultiLayered Regularization

Layer: $l-1$ l

$\vec{z}^{[l-1]}$ $\vec{a}^{[l-1]}$ $\vec{z}^{[l]}$ $\vec{a}^{[l]}$



$W^{[l]} = W[l-1, l]$

$\vec{z}^{[l]} = W^{[l]} \cdot \vec{a}^{[l-1]} + \vec{b}^{[l]}$

$\vec{a}^{[l]} = F(\vec{z}^{[l]})$

$\#n^{l-1}$ $\#n^{l}$

Multilayered Regularization

$$J(W^{[1]}, b^{[1]}, ..., W^{[L]}, b^{[L]}) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}\sum_{l=1}^{L} ||W^{[l]}||_F^2$$

$$||W^{[l]}||_F^2 = \sum_{i=1}^{n^{[l]}} \sum_{j=1}^{n^{[l-1]}} (w_{ij}^{[l]})^2$$

Frobenius

Regularization effect in GD

$$w_{k+1}^{[l]} = w_k^{[l]} - \alpha \nabla_{w_k^{[l]}} J^R$$

GD

$$\nabla_{w_k^{[l]}} J^R = \nabla_{w_k^{[l]}} J + \frac{\lambda}{m} w_k^{[l]}$$

additional

$$w_{k+1}^{[l]} = \left(1 - \frac{\alpha\lambda}{m}\right) w_k^{[l]} - \alpha \nabla_{w_k^{[l]}} J$$

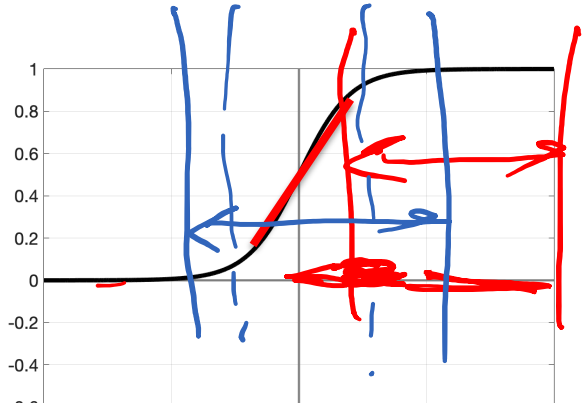
damping

L2 Regularization Intuition

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m} ||\vec{w}||_2^2$$

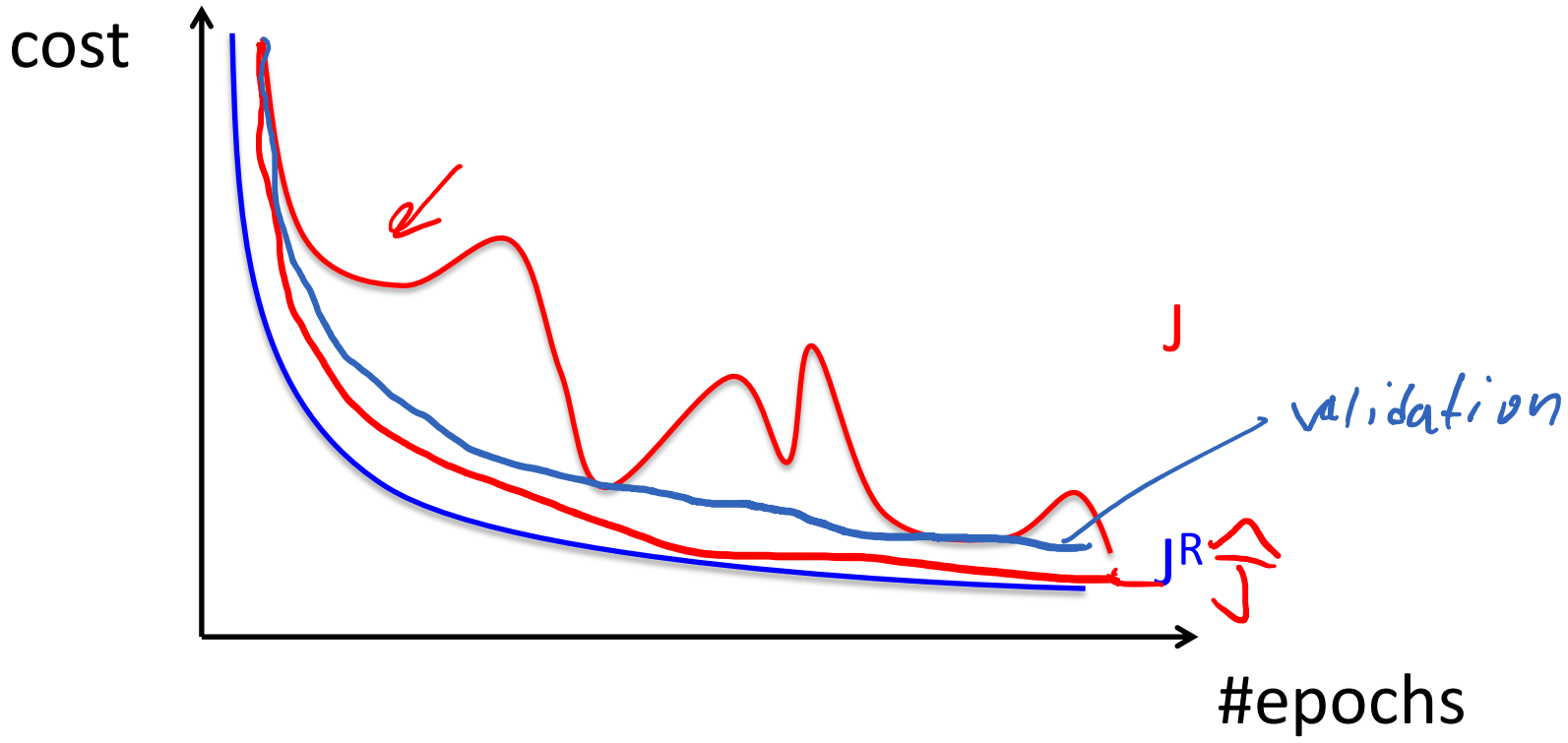
$\lambda \rightarrow$ large

$w^{[l]} \approx 0$

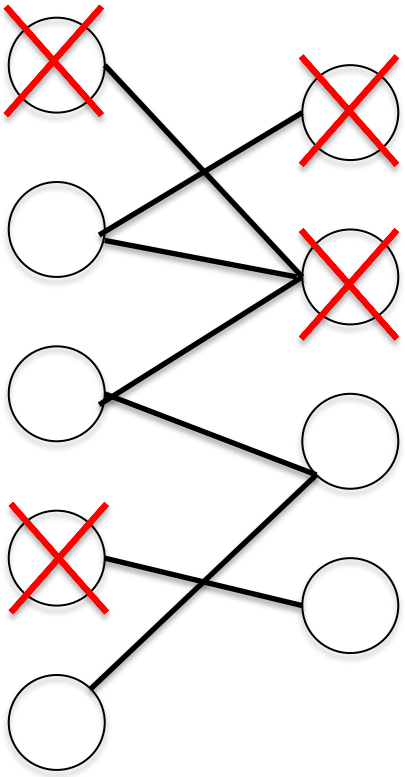


Have to be careful that $\vec{w} \neq 0$.

Cost curve



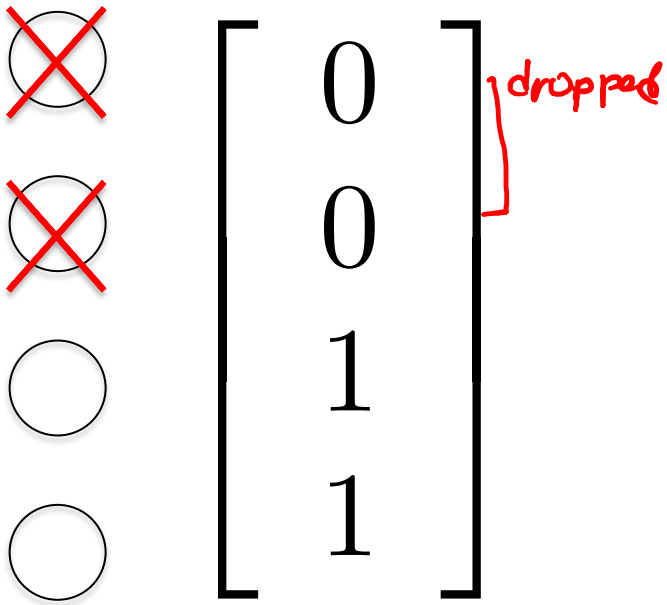
Dropout Regularization



$P_d^{[\ell-1]}=0.5$

Dropout - disabling activation

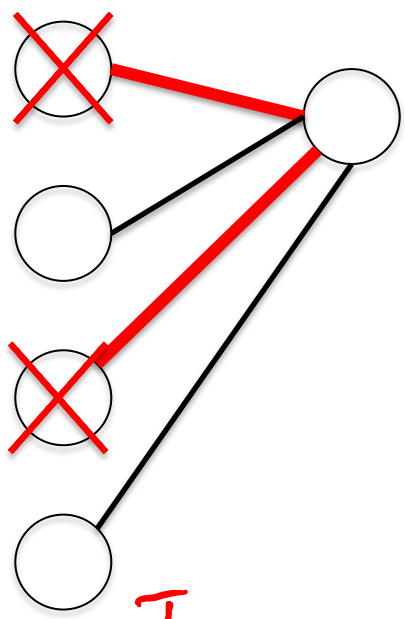
Inverted Dropout



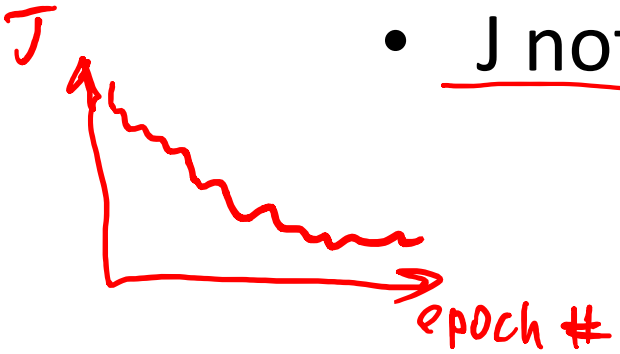
$$\vec{a}_l^d = \vec{a}_l \times \vec{d}_l / (1 - p_d)$$

Dropout intuition

"restricting optimization to fewer params at a time"



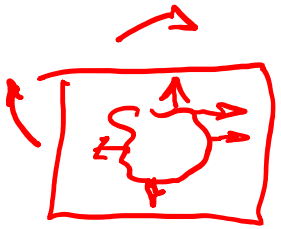
- Effectively spreading the weights
- Similar to L2 reg
- Testing with dropout p_d=0
- Can be W dim dependent
- J not well defined



Additional Regularization

- **Data Augmentation**

Enhance the data set with additional manipulations



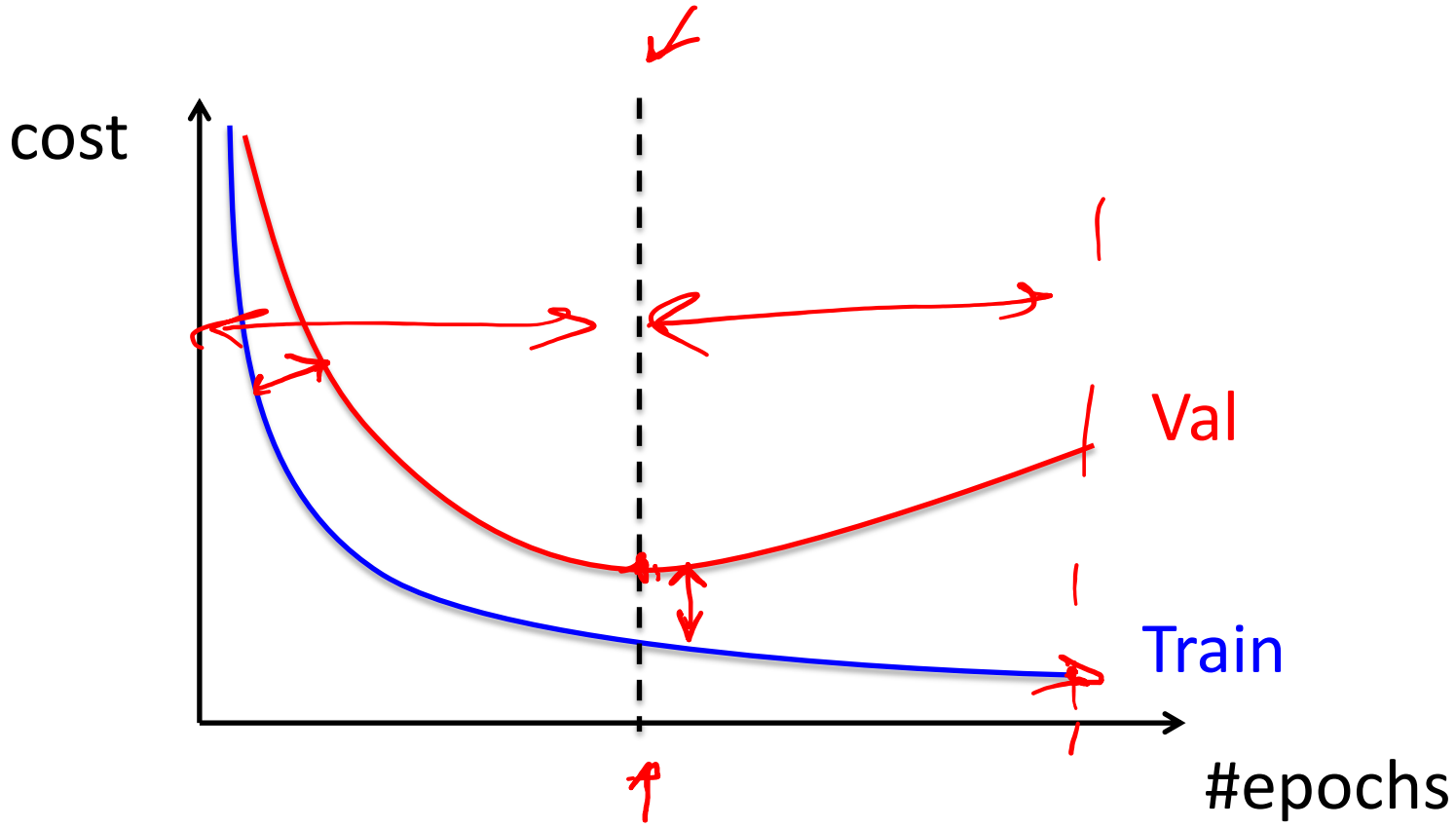
General input: add noise/distortions, synthetic

Images: resolutions, rotate, add symmetries

Shapes/digits: distort

Additional Regularization

- **Early Stopping:**



Balance

Optimization of J

GD

Over-fitting

Regularization