ONLINE MASTERS IN **DATA SCIENCE**

DSC 255 - MACHINE LEARNING FUNDAMENTALS

SOME ISSUES IN TRAINING NEURAL NETS

SANJOY DASGUPTA, PROFESSOR

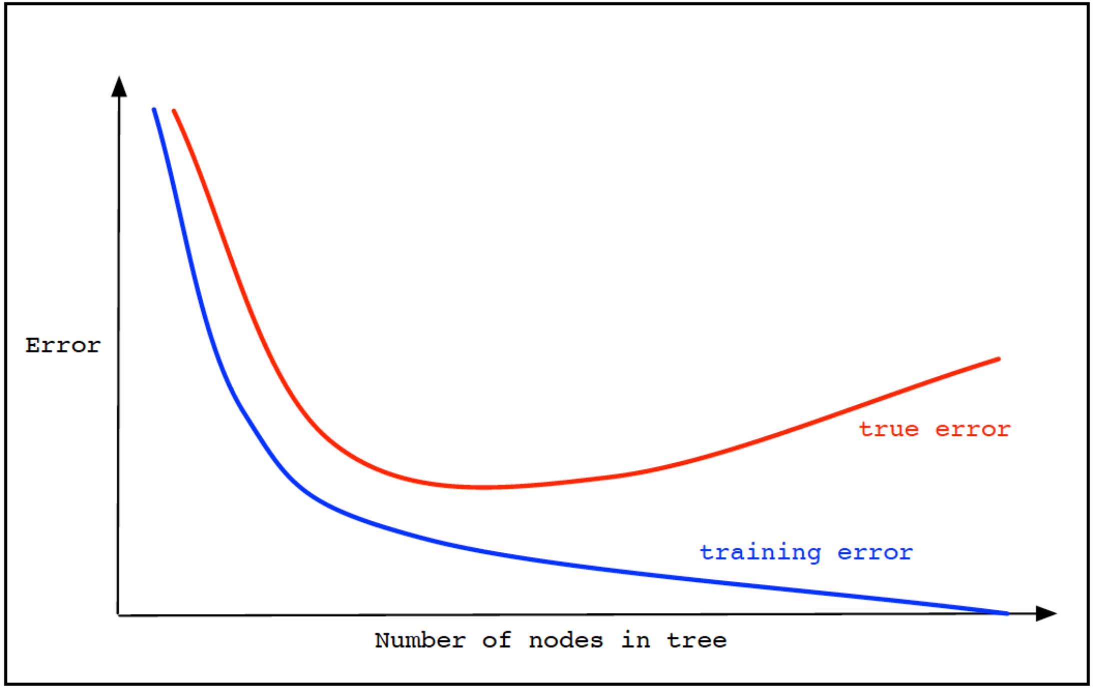
UC San Diego

COMPUTER SCIENCE & ENGINEERING

HALICIOĞLU DATA SCIENCE INSTITUTE

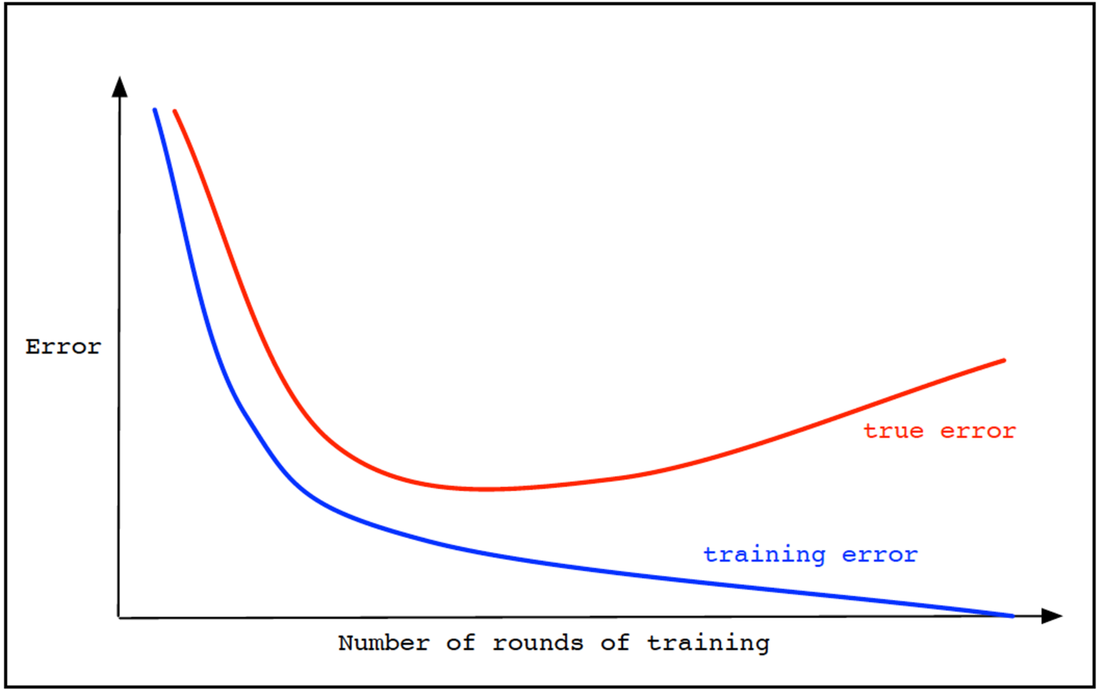
Improving Generalization 1: Early Stopping

- Validation set to better track error rate
- Revert to earlier model when recent training hasn't improved error



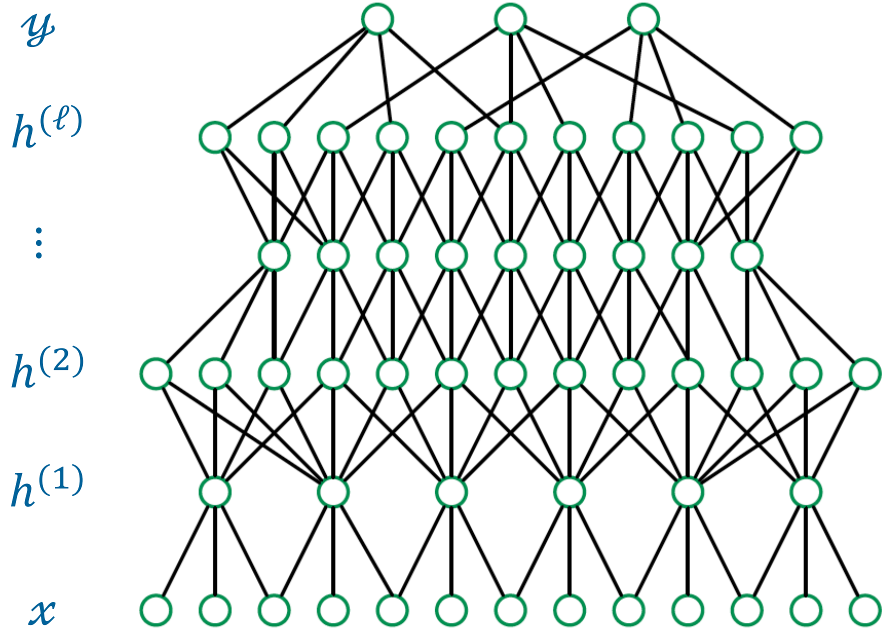
Improving Generalization 1: Early Stopping

- Validation set to better track error rate
- Revert to earlier model when recent training hasn't improved error



Improving Generalization 2: Dropout

During training, delete each hidden unit with probability 1/2, independently.



What does this remind you of?

Facilitating Optimization: Batch Normalization

The distribution of inputs to a particular **layer** of the net can change dramatically during training: **internal covariate shift**.

Mitigate this with an additional normalization step.

For each layer $x_1, \dots, x_p$ in the net, and each *mini-batch* B,

- Compute the mean $m_i^{(B)}$ and variance $v_i^{(B)}$ of each x_i in the mini-batch.
- Replace x_i by

$$x_i' = \frac{x_i - m_i^{(B)}}{\sqrt{v_i^{(B)} + \epsilon}}$$

before feeding to the next layer. This x_i' has mean 0 and variance ≈ 1.

Variants of SGD

Suppose we have parameters θ and loss $\ell(x, y; \theta)$. Usual SGD update:

$$\theta^{(t+1)} = \theta^{(t)} - \eta_t g^{(t)}$$

where $g^{(t)} = \nabla \ell(x_t, y_t; \theta^{(t)})$ is the gradient at time t.

- **Momentum:** Accumulate gradients. For $g^{(t)}$ as above, and $v^{(0)} = 0$,

$$v^{(t)} = \mu v^{(t-1)} + \eta_t g^{(t)}$$
$$\theta^{(t+1)} = \theta^{(t)} - v^{(t)}$$

- **AdaGrad:** Different learning rate for each parameter, automatically tuned.

$$\theta_j^{(t+1)} = \theta_j^{(t)} - \frac{\eta}{\sqrt{\sum_{t'<t} \left(g_j^{(t')}\right)^2 + \epsilon}} g_j^{(t)}$$

Many others: **Adam**, **RMSProp**, etc.