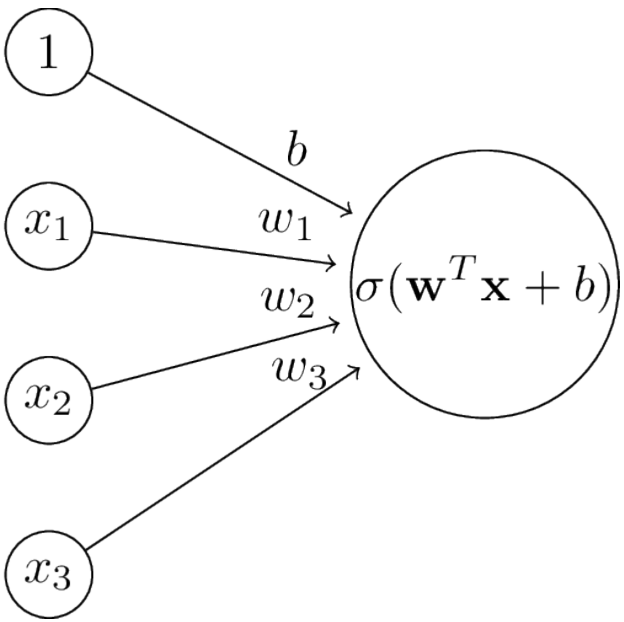
Multi-Layer Neural Networks

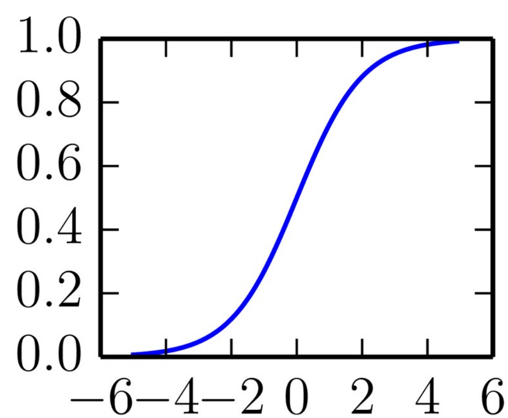
Review: Logistic Regression

<u>"Neuron"</u>

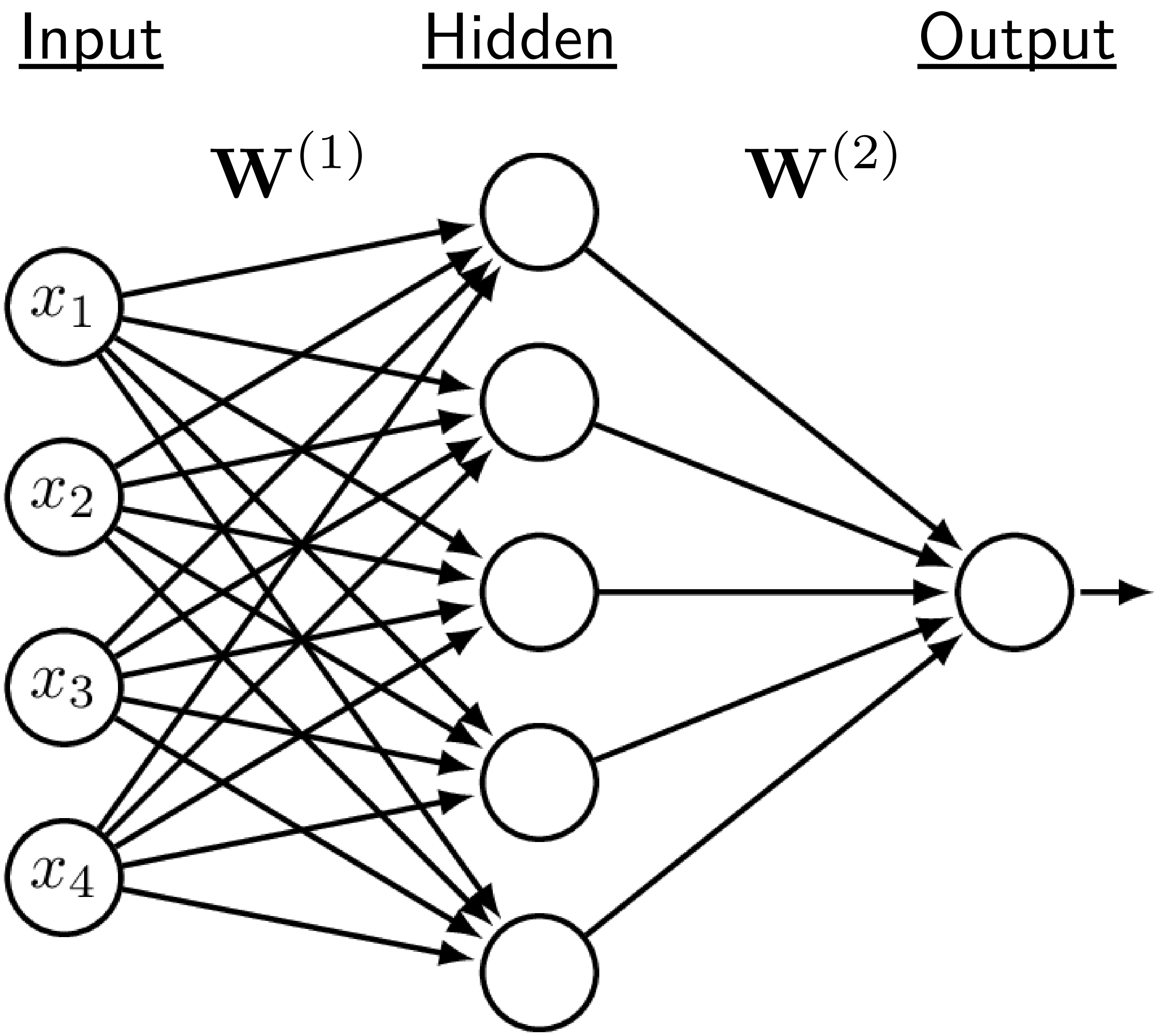


<u>Non-linearity</u>

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$



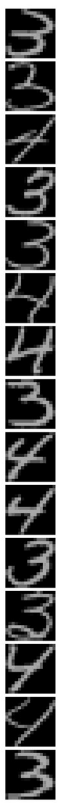
Multi-Layer Networks



Neural Network Example

Training Data

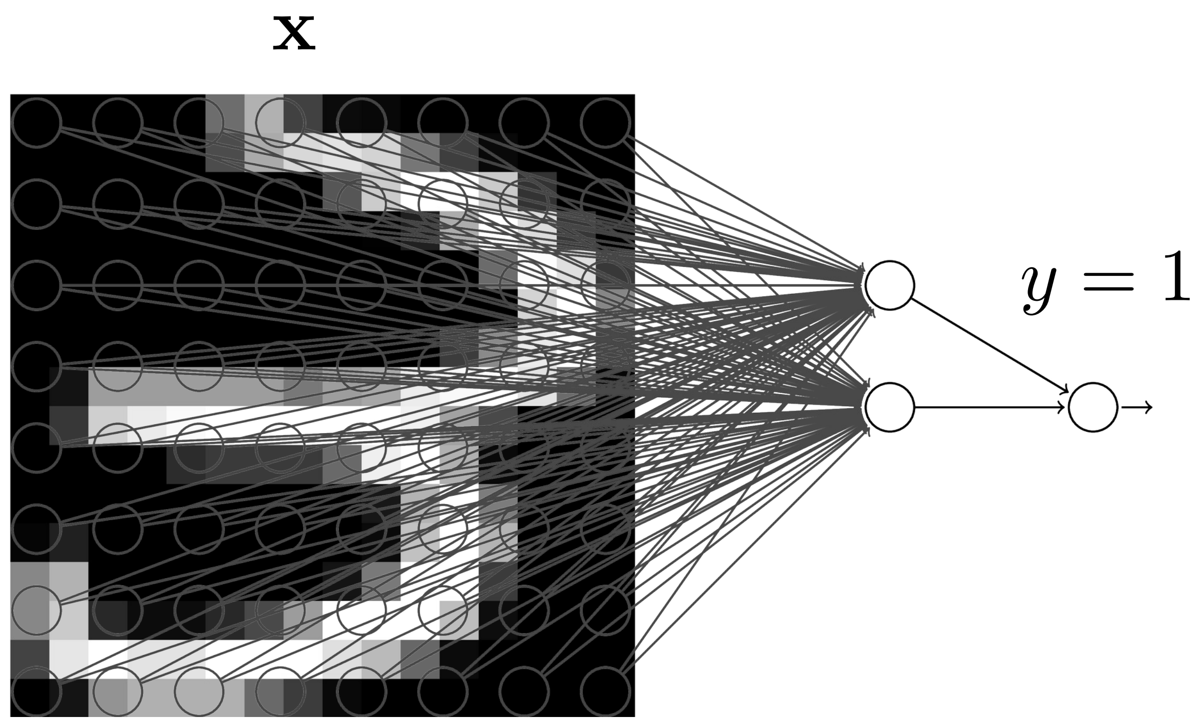
$\mathbf{x}$ $\quad y$



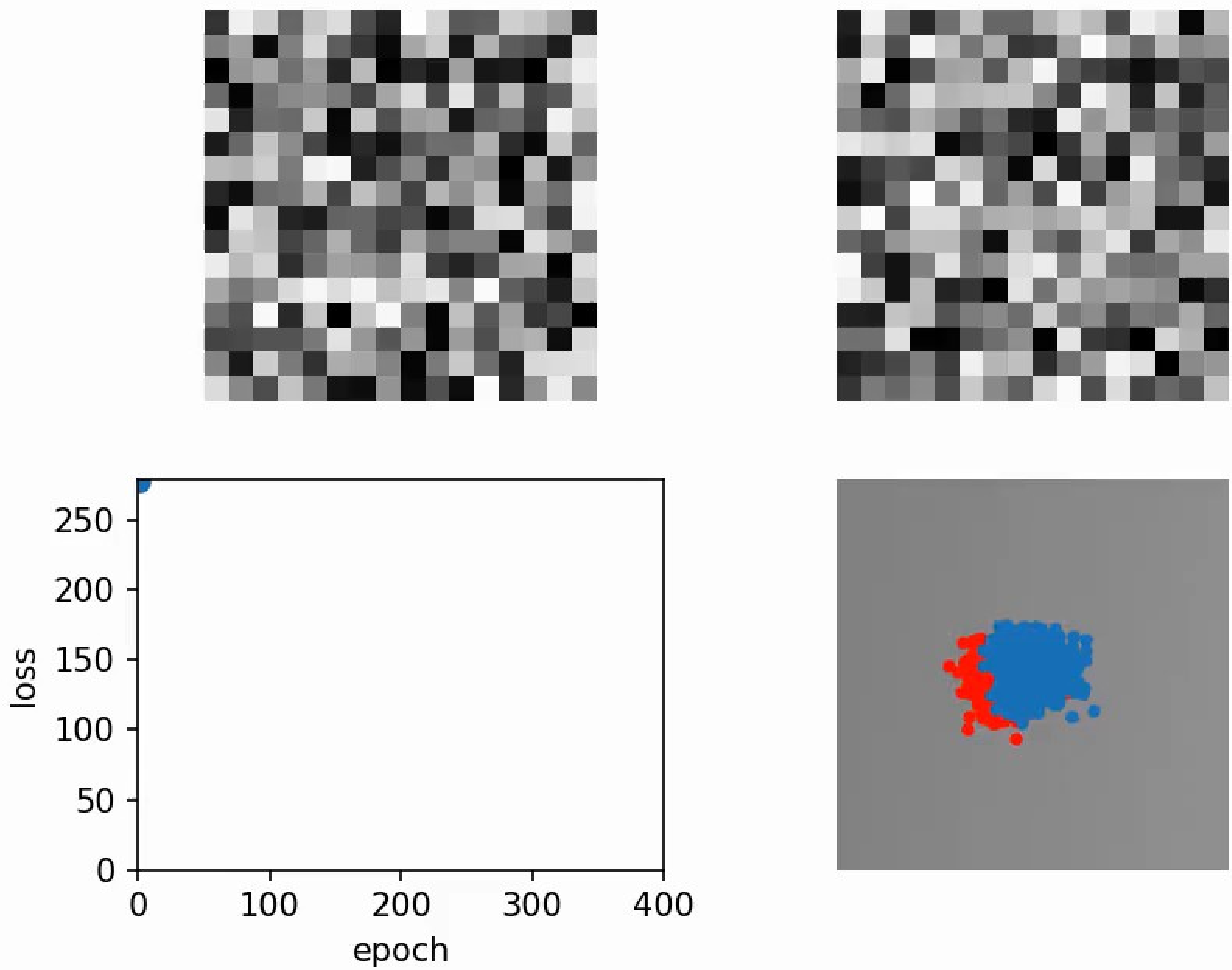
$\rightarrow 1$
$\rightarrow 1$
$\rightarrow 0$
$\rightarrow 1$
$\rightarrow 1$
$\rightarrow 0$
$\rightarrow 0$
$\rightarrow 1$
$\rightarrow 0$
$\rightarrow 0$
$\rightarrow 1$
$\rightarrow 1$
$\rightarrow 0$
$\rightarrow 0$
$\rightarrow 1$

$\vdots$

Network

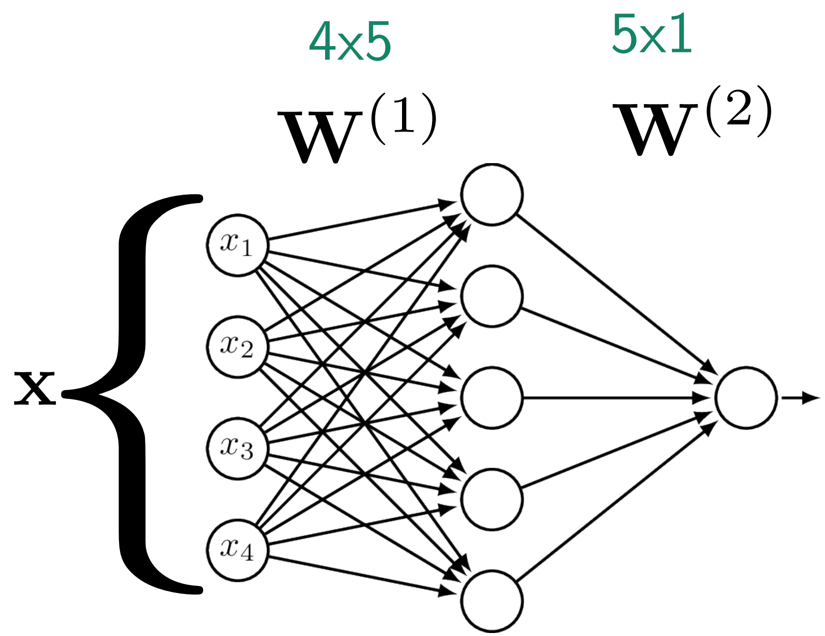


Training Example



Computation Example

4x1
column
vector

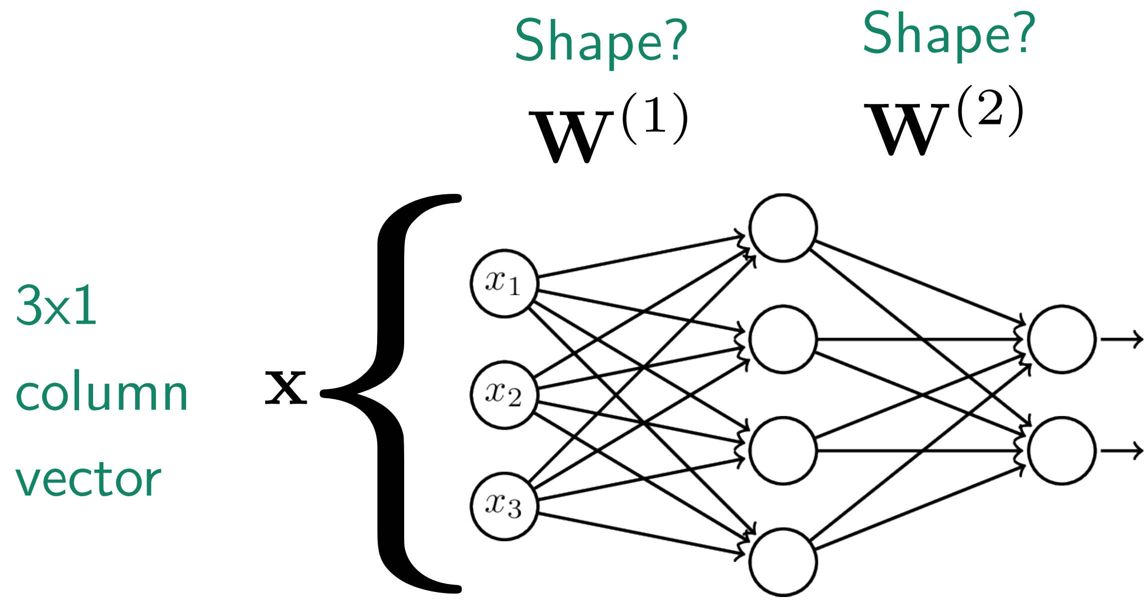


Hidden activation: $h(\mathbf{x}^\top \mathbf{W}^{(1)})$

Output activation: $\sigma\left(h(\mathbf{x}^\top \mathbf{W}^{(1)})\mathbf{W}^{(2)}\right)$

(h is the non-linearity at the hidden layer. *σ* is the non-linearity at the output. Applied element-wise.)

QUIZ

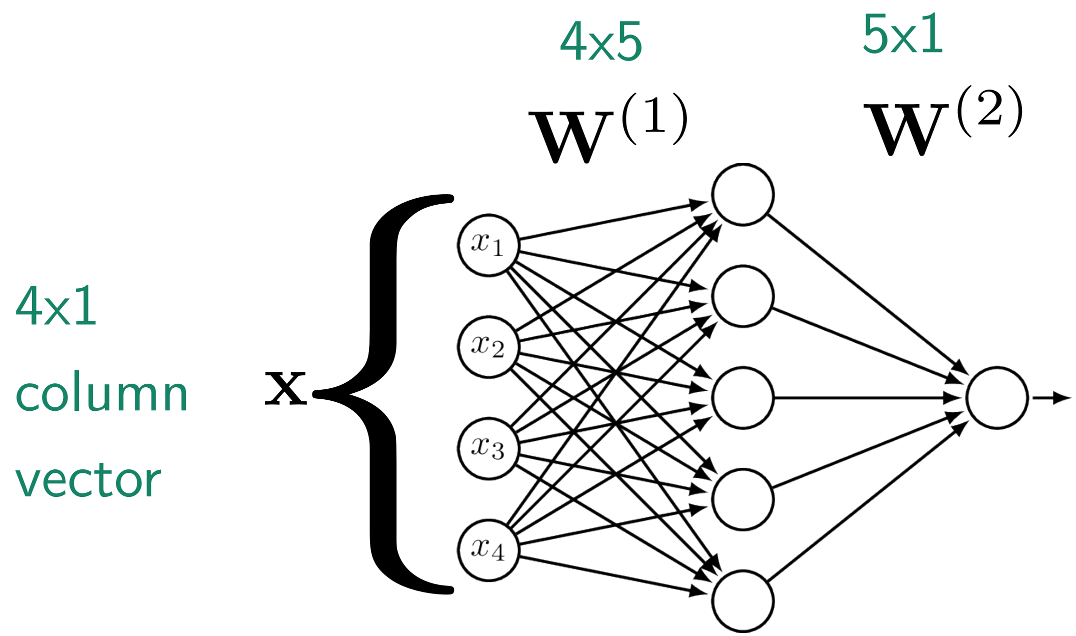


Hidden activation: $h(\mathbf{x}^\mathsf{T}\mathbf{W}^{(1)})$

Output activation: $\sigma\left(h(\mathbf{x}^\mathsf{T}\mathbf{W}^{(1)})\mathbf{W}^{(2)}\right)$

(h is the non-linearity at the hidden layer. *σ* is the non-linearity at the output. Applied element-wise.)

Bias Weights



Hidden activation: $h(\mathbf{x}^\mathsf{T}\mathbf{W}^{(1)} + \mathbf{b}^{(1)})$

Output activation: $\sigma\left(h(\mathbf{x}^\mathsf{T}\mathbf{W}^{(1)} + \mathbf{b}^{(1)})\mathbf{W}^{(2)} + b^{(2)}\right)$

(h is the non-linearity at the hidden layer. *σ* is the non-linearity at the output. Applied element-wise.)

Softmax Activation

- What if we have a multi-class problem?
- **Softmax** for K classes:

$$\sigma(\mathbf{a})_i = \frac{e^{a_i}}{\sum_{j=1}^{K} e^{a_j}}$$

- The (non-squashed) activations a_i are called **logits**.
- Cross-entropy loss function. The target vector $\boldsymbol{y}$ is "one-hot encoded".

$$Loss = -\sum_{i=1}^{K} y_i \log(\sigma(\mathbf{a})_i)$$

Gradient Descent Reminder

- Define a Loss Function:

$$L(\mathbf{w}, D) = -\sum_{(\mathbf{x}_i,\mathbf{y}_i)\in D}\sum_{j=1}^{K} y_{i,j}\log(\sigma(\mathbf{a})_j)$$

- Find the gradient of the error function with respect to the weights:

$$\nabla_{\mathbf{w}} L(\mathbf{w}, D)$$

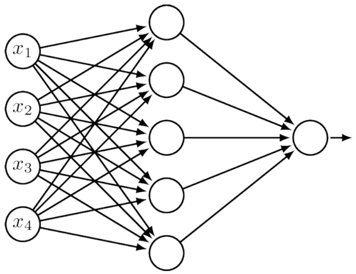
- Take small steps in the direction of the gradient:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha\nabla_{\mathbf{w}} L(\mathbf{w}, D)$$

Backpropagation/Reverse Mode Automatic Differentiation

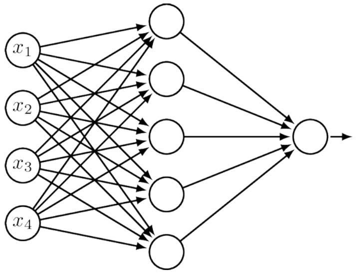
- Efficient algorithm for implementing gradient descent in neural networks.
- Forward Pass:

Activation



- Backward Pass:

Error Signal



Backpropagation/Reverse Mode Automatic Differentiation

- Invented:

> Linnainmaa, S. (1970). *The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors*. Master's thesis, Univ. Helsinki.

- First applied to neural networks:

> P. J. Werbos. (1982) Applications of advances in nonlinear sensitivity analysis. In R. Drenick, F. Kozin, (eds): System Modeling and Optimization: Proc. IFIP, Springer

- Shown to create useful representations in the hidden layer

> DE Rumelhart, GE Hinton, RJ Williams (1985). Learning Internal Representations by Error Propagation. TR No. ICS-8506, California Univ San Diego La Jolla Inst for Cognitive Science.

- Detailed history:

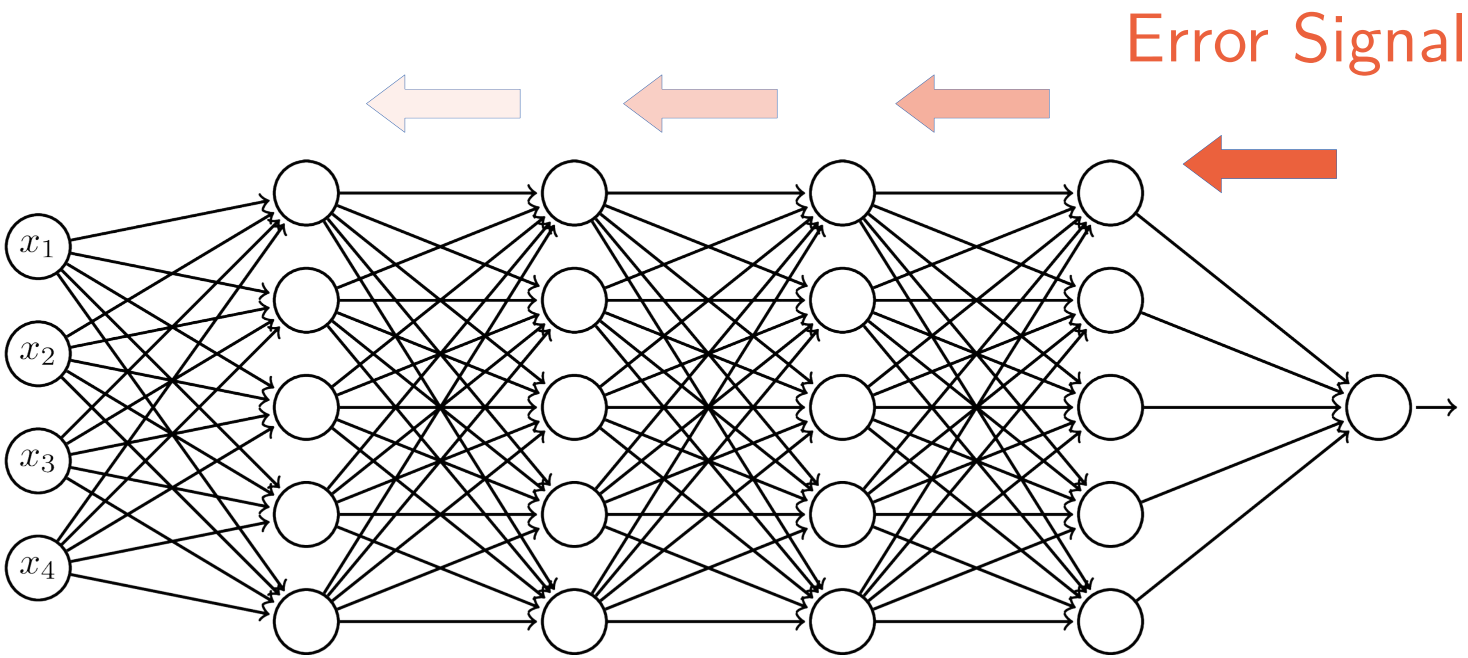
> Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

- Modern ML libraries like PyTorch and TensorFlow automate the implementation of the backward pass.

Deep vs. Shallow Networks

- How best to add capacity?
  - More units in a single hidden layer?
    - Three layer networks are universal approximators: with enough units any continuous function can be approximated
    - Adding layers makes the learning problem harder…

Vanishing Gradients



Advantages of Deep Architectures

- There are tasks that require exponentially many hidden units for a three-layer architecture, but only polynomially many with more hidden layers
- The best hand-coded image processing algorithms have deep structure
- The brain has a deep architecture
- MORE SOON.