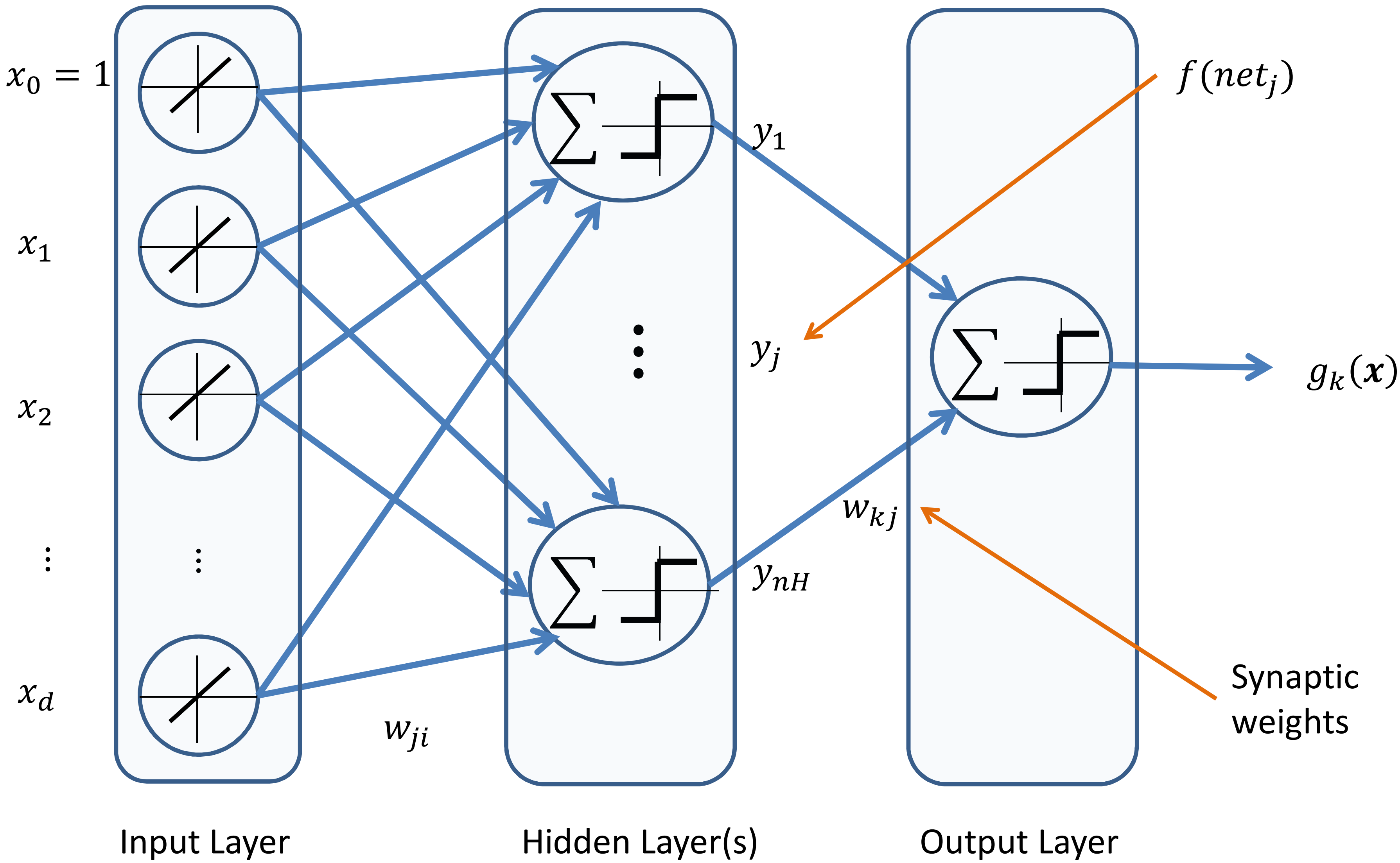
Statistical Methods in Artificial Intelligence
CSE471 - Monsoon 2016 : Lecture 07

Avinash Sharma
CVIT, IIIT Hyderabad

Lecture 07: Plan

- Recap
- Backpropagation in NN
- Learning Curves
- Practical Aspects of Backpropagation

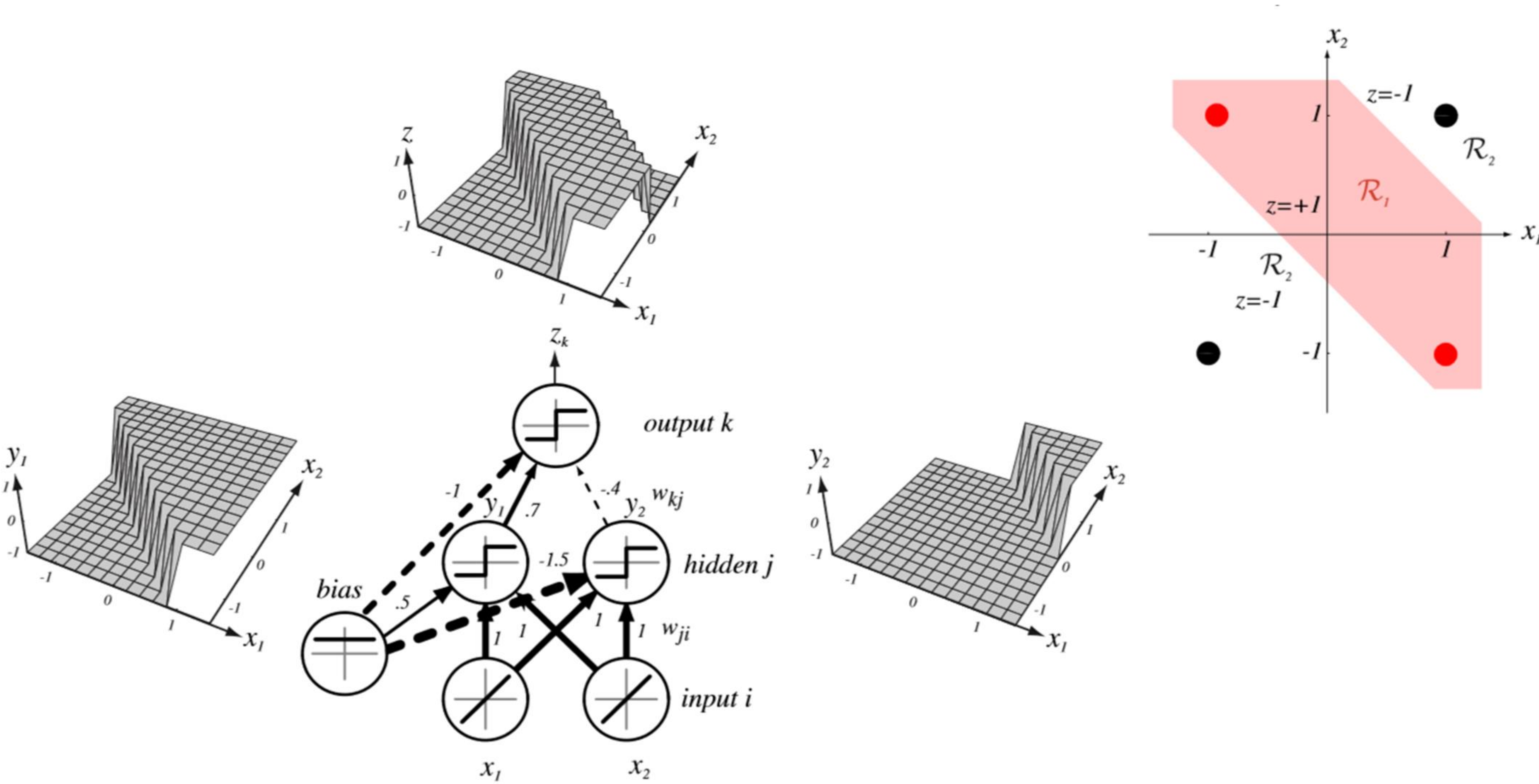
Construction of NN Classifier



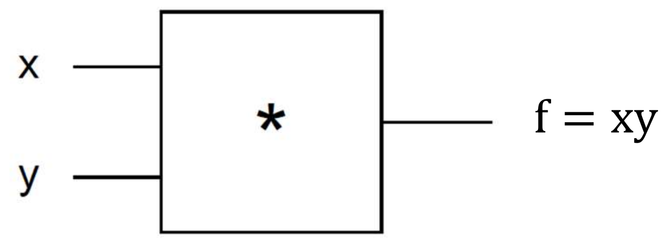
Feed Forward Operation in NN

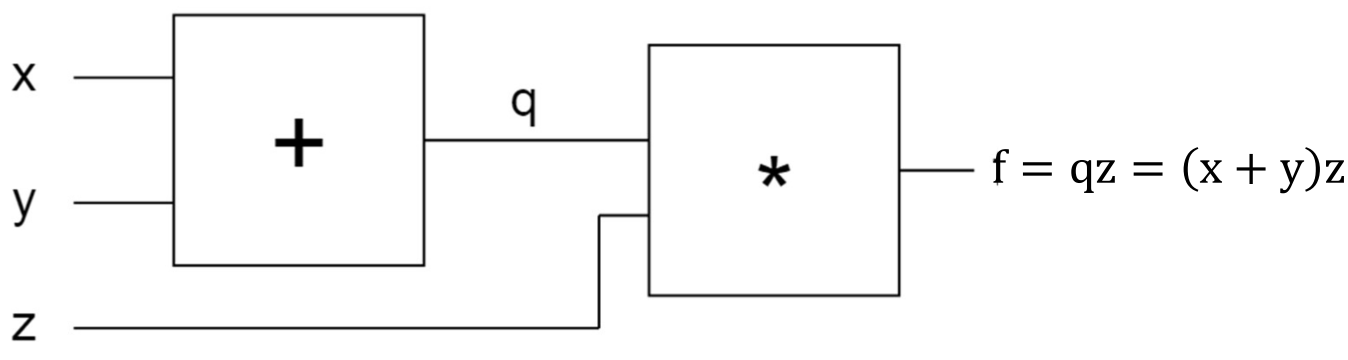
- $net_j = \sum_{i=1}^{d} x_i w_{ji} + w_{j0} = \sum_{i=0}^{d} x_i w_{ji} = \mathbf{w}^T{}_j \mathbf{x}$
- $y_j = f(net_j)$
- $y_j = sgn(net_j)$
- $net_k = \sum_{j=1}^{nH} y_j w_{kj} + w_{k0} = \sum_{j=0}^{nH} y_j w_{kj} = \mathbf{w}^T{}_k y$
- $z_k = f(net_k) = sgn(net_k)$
- $g_k(\mathbf{x}) = z_k = \sum_{j=1}^{nH} w_{kj}\, f(\sum_{i=1}^{d} x_i w_{ji} + w_{j0}) + w_{k0}$

Modelling the Non-linearity

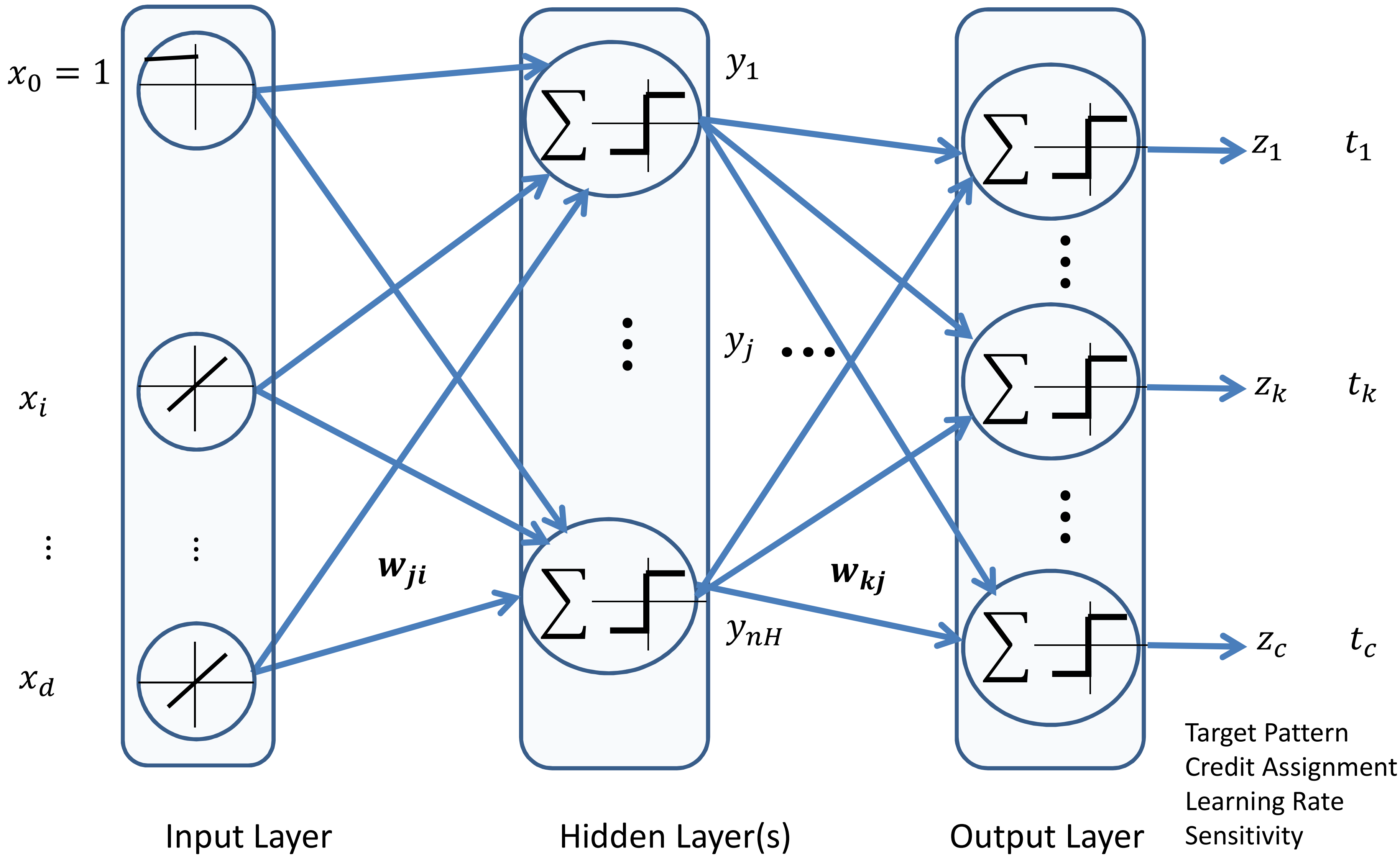


Understanding Backpropagation





Backpropagation in NN



Backpropagation in Neural Networks

- $J(\mathbf{w}) = \frac{1}{2}\sum_{k=1}^{c}(t_k - z_k)^2 = \frac{1}{2}\|\mathbf{t} - \mathbf{z}\|^2$

- $\Delta\mathbf{w} = -\eta\frac{\partial J}{\partial \mathbf{w}}$, $\Delta w_{pq} = -\eta\frac{\partial J}{\partial w_{pq}}$
- $\mathbf{w}(m+1) = \mathbf{w}(m) + \Delta\mathbf{w}$
- $\boxed{\Delta w_{kj} = \eta\delta_k y_j = \eta(t_k - z_k)f'(net_k)y_j}$

Backpropagation in Neural Networks

- $\Delta w_{kj} = -\eta \frac{\partial J}{\partial w_{kj}} = -\eta \left(\frac{\partial J}{\partial net_k} * \frac{\partial net_k}{\partial w_{kj}} \right)$
- $\Delta w_{kj} = -\eta \left(-\frac{\partial J}{\partial net_k} * -\frac{\partial net_k}{\partial w_{kj}} \right)$
- $\Delta w_{kj} = -\eta \left(\left(-\frac{\partial J}{\partial z_k} * \frac{\partial z_k}{\partial net_k} \right) * -\frac{\partial net_k}{\partial w_{kj}} \right)$
- $\Delta w_{kj} = -\eta \left(\underline{(t_k - z_k) * f'(net_k)} * -y_j \right)$
- $\Delta w_{kj} = -\eta \left(\underline{\delta_k} * -y_j \right) = \eta \delta_k y_j$

$$\delta_k = -\frac{\partial J}{\partial net_k} = (t_k - z_k) * f'(net_k)$$

Backpropagation in Neural Networks

- $J(\mathbf{w}) = \frac{1}{2}\sum_{k=1}^{c}(t_k - z_k)^2 = \frac{1}{2}\|\mathbf{t} - \mathbf{z}\|^2$
- $\Delta\mathbf{w} = -\eta\frac{\partial J}{\partial \mathbf{w}}, \Delta w_{pq} = -\eta\frac{\partial J}{\partial w_{pq}}$
- $\mathbf{w}(m+1) = \mathbf{w}(m) + \Delta\mathbf{w}$
- $\Delta w_{kj} = \eta\delta_k y_j = \eta(t_k - z_k)f'(net_k)y_j$
- $\boxed{\Delta w_{ji} = \eta\delta_j x_i = \eta\left[\sum_{k=1}^{c} w_{kj}\,\delta_k\right]f'(net_j)x_i}$

Backpropagation in Neural Networks

- $\Delta w_{ji} = -\eta \frac{\partial J}{\partial w_{ji}} = -\eta \left(\frac{\partial J}{\partial net_j} * \frac{\partial ne\ \ {}_j}{\partial w_{ji}} \right)$
- $\Delta w_{ji} = -\eta \left(\left(\frac{\partial J}{\partial y_j} * \frac{\partial y_j}{\partial net_j} \right) * \frac{\partial ne\ \ {}_j}{\partial w_{ji}} \right)$
- $\Delta w_{ji} = -\eta \left(\left(\left(\frac{\partial J}{\partial z_k} * \frac{\partial z_k}{\partial net_k} * \frac{\partial net_k}{\partial y_j} \right) * \frac{\partial y_j}{\partial net_j} \right) * \frac{\partial net_j}{\partial w_{ji}} \right)$
- $\Delta w_{ji} = -\eta \left(\underline{\left(\left(-\sum_{k=1}^{c} (t_k - z_k) * f'(net_k) * w_{kj} \right) * f'\left(net_j\right) \right)} * x_i \right)$
- $\Delta w_{ji\text{`}} = \eta \underline{\delta_j} x_i = \eta \left(\underline{\left[\sum_{k=1}^{c} w_{kj}\, \delta_k \right] f'\left(net_j\right)} \right) x_i$

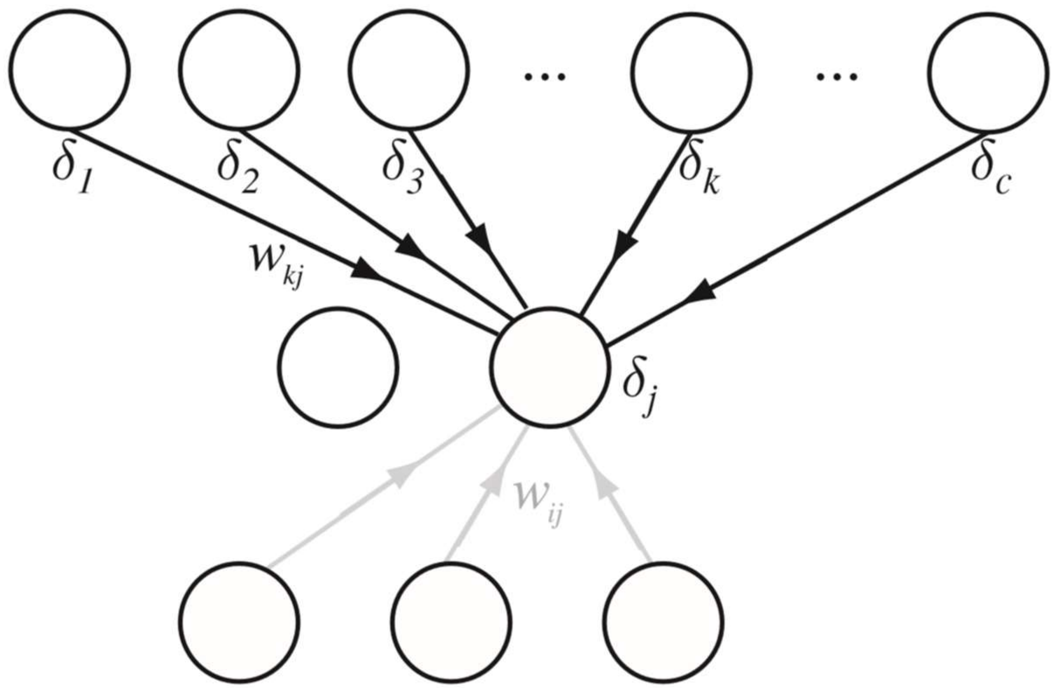
Where, $\delta_k = (t_k - z_k) * f'(net_k)$

Backpropagation in Neural Networks

- Sensitivity Backpropagation:

$$\delta_k = (t_k - z_k) * f'(net_k),$$

$$\delta_j = \left[\sum_{k=1}^{c} w_{kj}\, \delta_k\right] * f'\left(net_j\right)$$



Output Layer

Hidden Layer

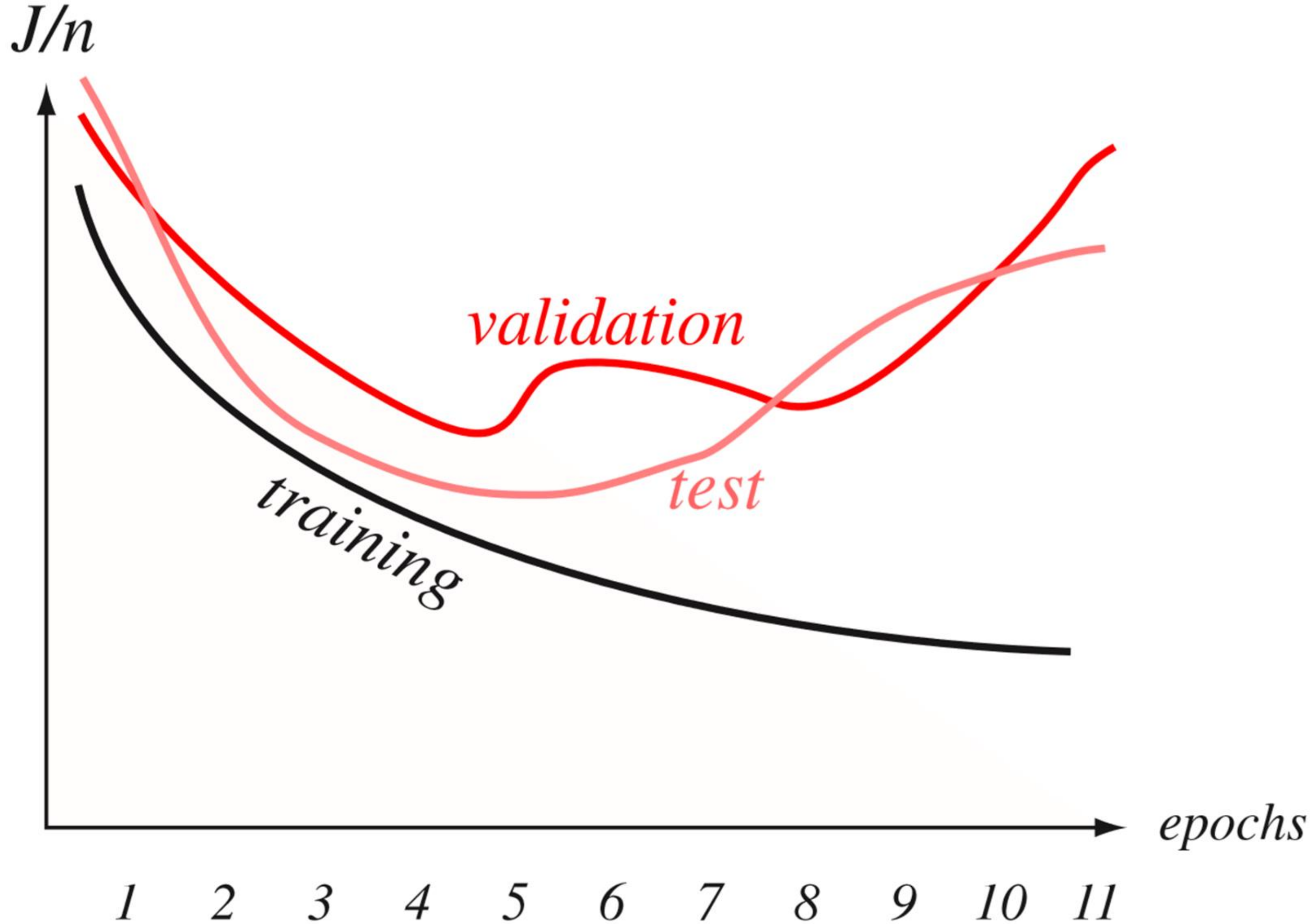
Input Layer

Backpropagation in NN

- Stochastic Backpropagation

1. Initialize $nH, \mathbf{w}, \theta$ (threshold), $\eta, m = 0$
2. do m= $m + 1$
3. randomly choose $\mathbf{x}^m$
4. $w_{ji} = w_{ji} + \eta \delta_j x_i$
5. $w_{kj} = w_{kj} + \eta \delta_k y_j$
6. until $|\nabla J(\mathbf{w})| < \theta$
7. return $\mathbf{w}$

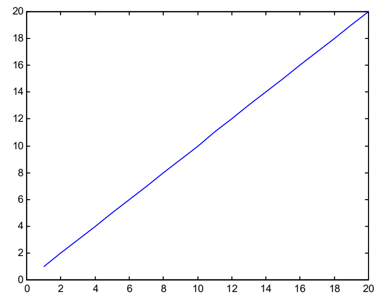
Learning Curves



Practical Aspects of Backpropagation

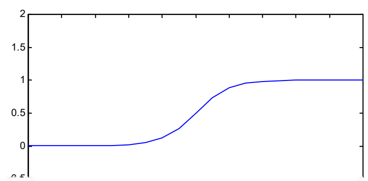
- Activation Function
  - $f(\cdot)$ should be ***Non-linear***
  - $f(\cdot)$ should ***Saturate***
  - $f(\cdot)$ should be ***Continuous & Smooth***
  - $f'(\cdot)$ should be ***Defined***
  - $f(\cdot)$ can have ***Monotonicity***
  - $f(\cdot)$ can be ***linear for small values of*** $\boldsymbol{net}$

Practical Aspects of Backpropagation



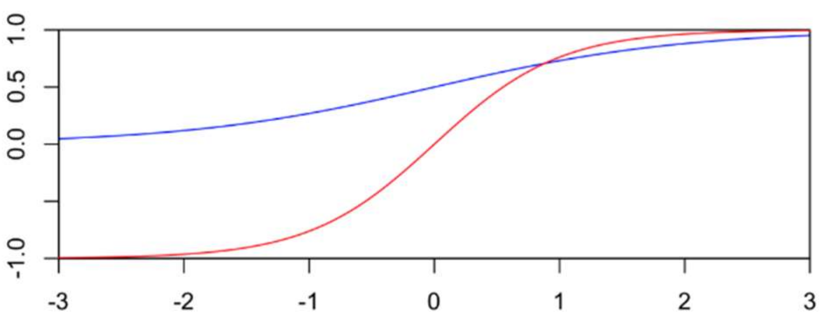
Linear

$$y = x$$

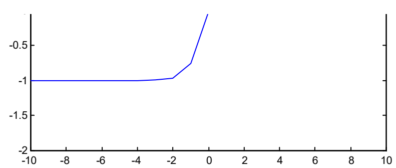


Logistic (Sigmoid) O/P : 0 to 1

$$y = \frac{1}{1 + \exp(-x)} = \frac{\exp(x)}{1 + \exp(x)}$$

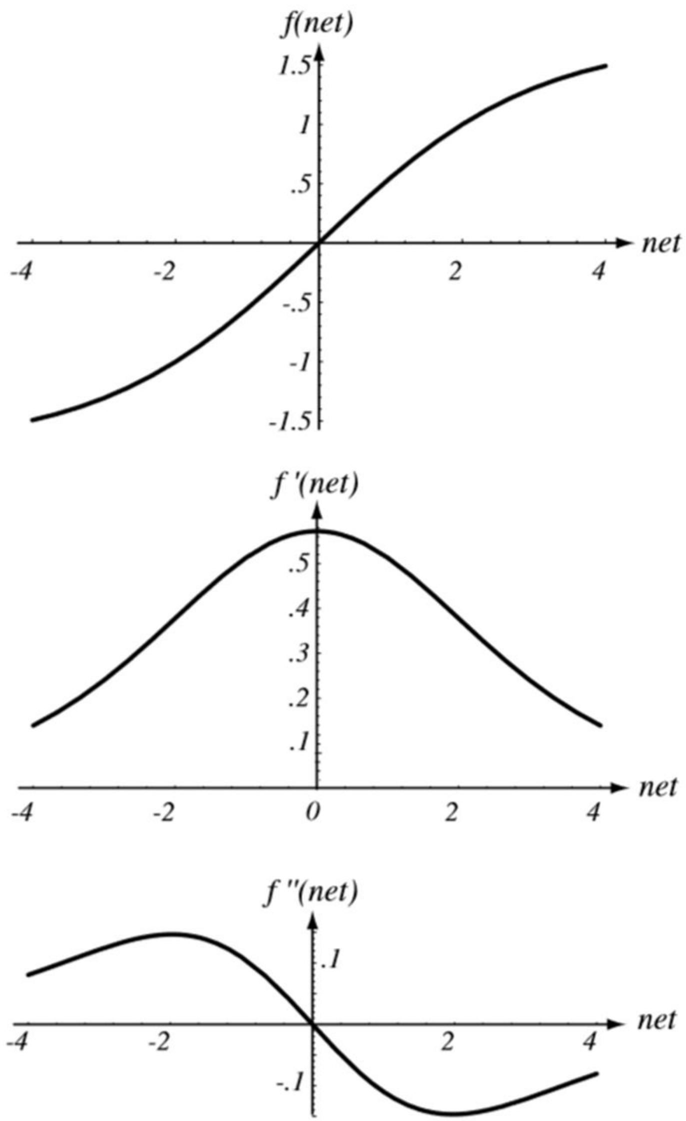


Hyperbolic Tangent O/P : -1 to +1



$$y = a\left[\frac{\exp(bx) - \exp(-bx)}{\exp(bx) + \exp(-bx)}\right]$$

Practical Aspects of Backpropagation



$a = 1.716, b = 2/3$

$$y = a\left[\frac{e^{bx} - e^{-bx}}{e^{bx} + e^{-bx}}\right]$$

Activation Function centered on zero and antisymmetric leads to faster convergence

Practical Aspects of Backpropagation

- Sigmoid as activation function
  - Smooth, Differentiable, Nonlinear and Saturating
  - Admit to linear model for small network weights
  - Derivative of Sigmoid function can be represented in terms of function itself.

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

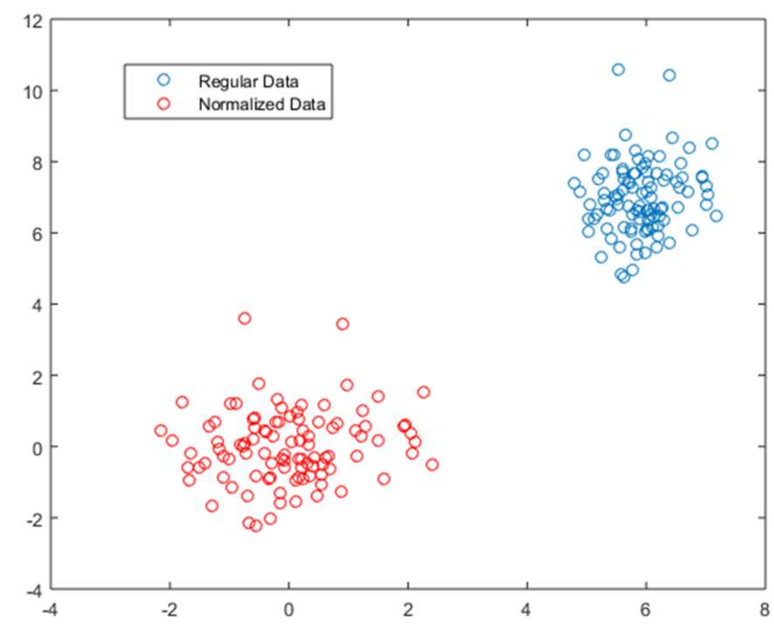
$$\sigma'(x) = -\left\{\frac{1}{(1 + e^{-x})}\right\}^2 e^{-x} \, (-1)$$

$$\sigma'(x) = \frac{1}{(1 + e^{-x})} \times \frac{e^{-x}}{(1 + e^{-x})}$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

Practical Aspects of Backpropagation

- **Scaling of Input**
  - $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m]^T{}_{m \times d}$
  - $\widetilde{\boldsymbol{X}} = \boldsymbol{X} - mean(\boldsymbol{X})$ (Centering of Data)
  - $\boldsymbol{X}_{Norm} = \widetilde{\boldsymbol{X}} / \boldsymbol{\sigma}$ (Column-wise division by Standard Deviation of each dimension)



Practical Aspects of Backpropagation

- **Scaling of Input (*Standardize*)**
  - $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m]^T{}_{m \times d}$
  - $\widetilde{\boldsymbol{X}} = \boldsymbol{X} - mean(\boldsymbol{X})$ (Centering of Data)
  - $\boldsymbol{X}_{Norm} = \widetilde{\boldsymbol{X}}/\boldsymbol{\sigma}$ (Column-wise division by Standard Deviation of each dimension)
- **Target Values**
  - Use $+1$ and -1 or any real value in this range as output
  - Related to saturation value of the activation function
  - Probabilistic Interpretation is not valid for category labels.
- **Training with Noise**
  - Add random noise to original training samples for generating more training samples

Practical Aspects of Backpropagation

- **Manufacturing Data**
  - Add translation and rotation transforms to original training data to generate more rich training data samples
- **Number of Hidden Units**
  - Too few leads to high test error due to lack of expressibility
  - Too many leads to overfitting to training data
  - Choose such that total number of weights = $n/10$.
- **Weight Initialization**
  - Do not initialize with zero weights
  - Use both random positive & negative weights as data is standardized
  - Consider the saturation value of net activation function while choosing range for initial weights.
  - $-1/\sqrt{d} < w_{ji} < +1/\sqrt{d}$ and $-1/\sqrt{nH} < w_{kj} < +1/\sqrt{nH}$