Brief summary on Previous Lecture

- For a two-class problem, the optimum LDA solution or SLP initialization or *Fisher criterion* can be obtained by:

$$\mathbf{w} = \arg\max_{\mathbf{w}} \left(\frac{\mathbf{w}\mathbf{S}_b\mathbf{w}^T}{\mathbf{w}\mathbf{S}_w\mathbf{w}^T} \right) = \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right) \text{ and SLP } y = \varphi\left(\mathbf{w}^T\left(\mathbf{x} - \mathbf{b}_0\right)\right).$$

- In BP learning algorithm, weight updates are "local":

For output nodes

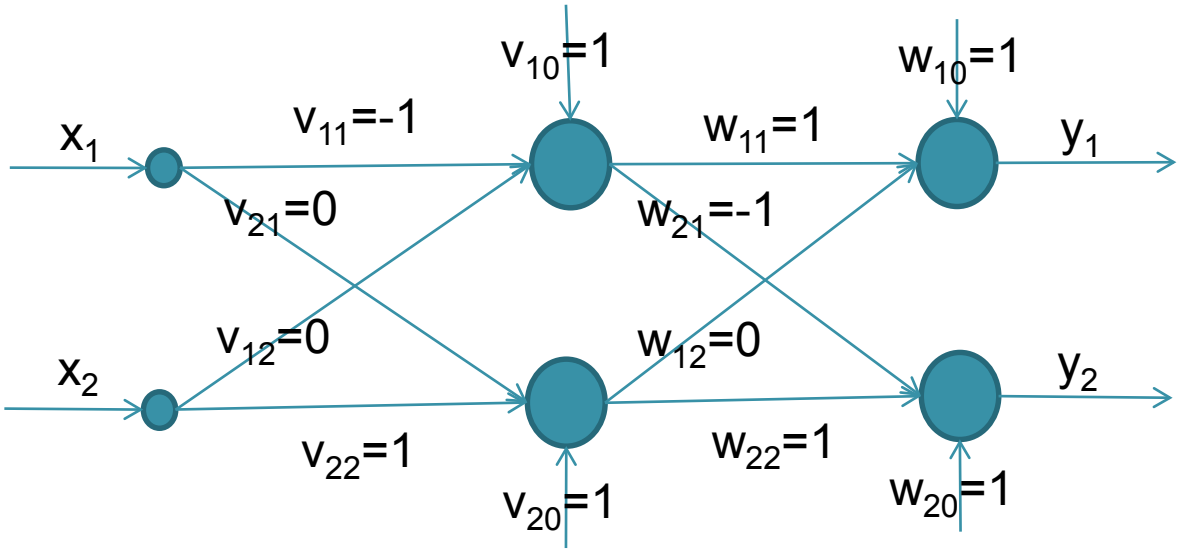
$$w_{k,j}(t+1) = w_{k,j}(t) + \eta\Delta_k(t)z_j(t)$$
$$= w_{k,j}(t) + \eta\varphi'\left(a_k(t)\right)\left(d_k(t) - y_k(t)\right)z_j(t)$$

For hidden nodes

$$v_{j,i}(t+1) = v_{j,i}(t) + \eta\delta_j(t)x_i(t)$$
$$= v_{j,i}(t) + \eta\varphi'\left(u_j(t)\right)\sum_{k=1}^{m2} w_{k,j}\Delta_k(t)x_i(t)$$

MLP for XOR Example

- Use identity activation function $\varphi(a) = a$, for illustration
- Learning rate η=0.1
- Consider input [0 1] with target output [1 0]

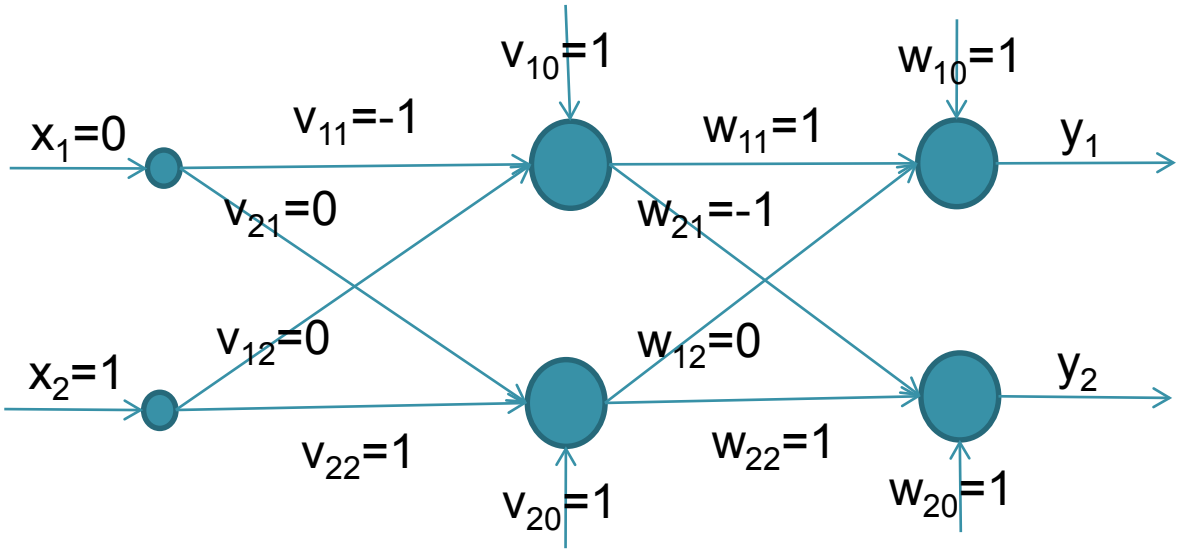


MLP for XOR Example

- Forward pass:

$$u_1 = x_1 v_{11} + x_2 v_{12} + v_{10} = 0 \cdot -1 + 1 \cdot 0 + 1 = 1, \qquad z_1 = u_1 = 1$$

$$u_2 = x_1 v_{21} + x_2 v_{22} + v_{20} = 0 \cdot 0 + 1 \cdot 2 + 1 = 2, \qquad z_2 = u_2 = 2$$

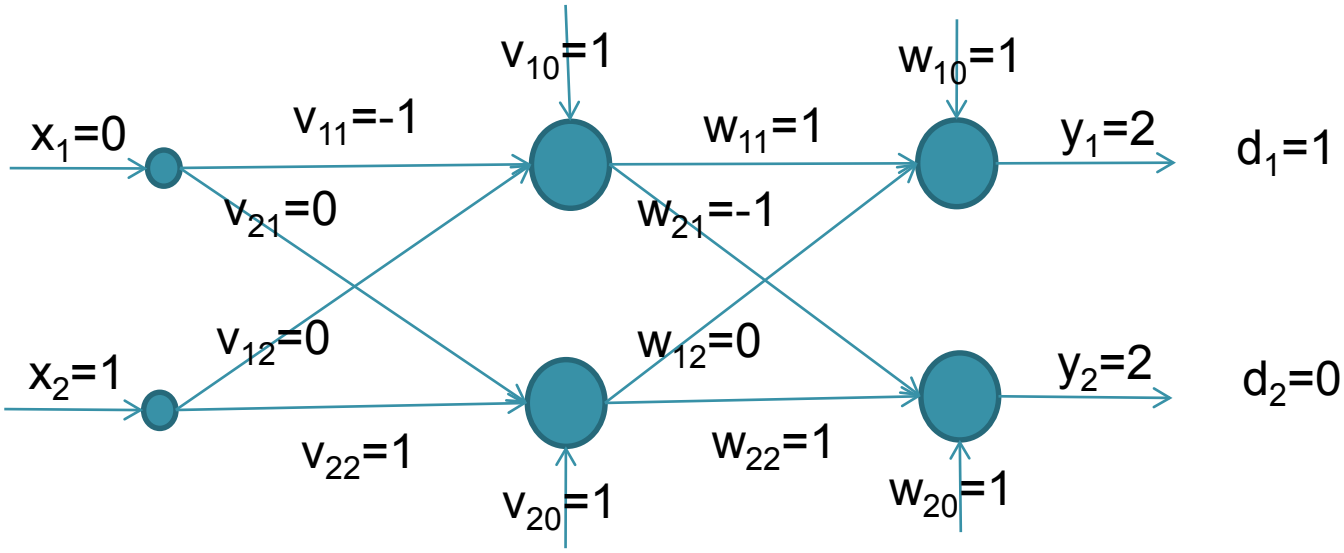


MLP for XOR Example

- Forward pass:

$$a_1 = z_1 w_{11} + z_2 w_{12} + w_{10} = 1 \cdot 1 + 2 \cdot 0 + 1 = 2, \qquad y_1 = a_1 = 2$$

$$a_2 = z_1 w_{21} + z_2 w_{22} + w_{20} = 1 \cdot -1 + 2 \cdot 1 + 1 = 2, \qquad y_2 = a_2 = 2$$



MLP for XOR Example

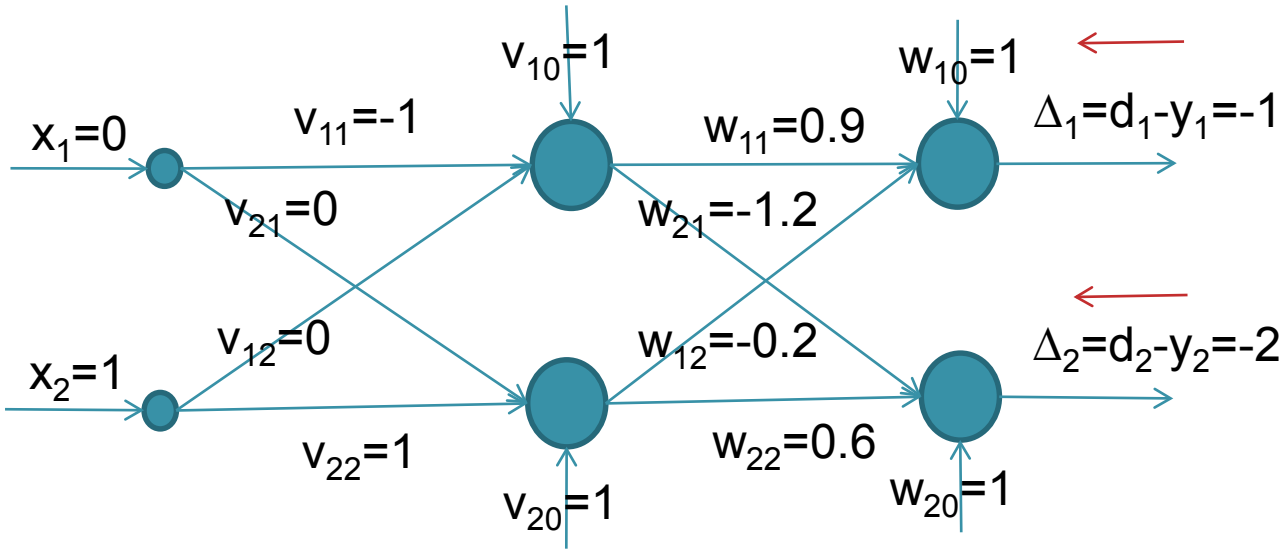
- Backward pass (output-to-hidden):

$$w_{1,1}(t+1) = w_{1,1}(t) + \eta\Delta_1(t)z_1(t) = 1 + 0.1 \cdot -1 \cdot 1 = 0.9$$

$$w_{2,1}(t+1) = w_{2,1}(t) + \eta\Delta_2(t)z_1(t) = -1 + 0.1 \cdot -2 \cdot 1 = -1.2$$

$$w_{1,2}(t+1) = w_{1,2}(t) + \eta\Delta_1(t)z_2(t) = 0 + 0.1 \cdot -1 \cdot 2 = -0.2$$

$$w_{2,2}(t+1) = w_{2,2}(t) + \eta\Delta_2(t)z_2(t) = 1 + 0.1 \cdot -2 \cdot 2 = 0.6$$

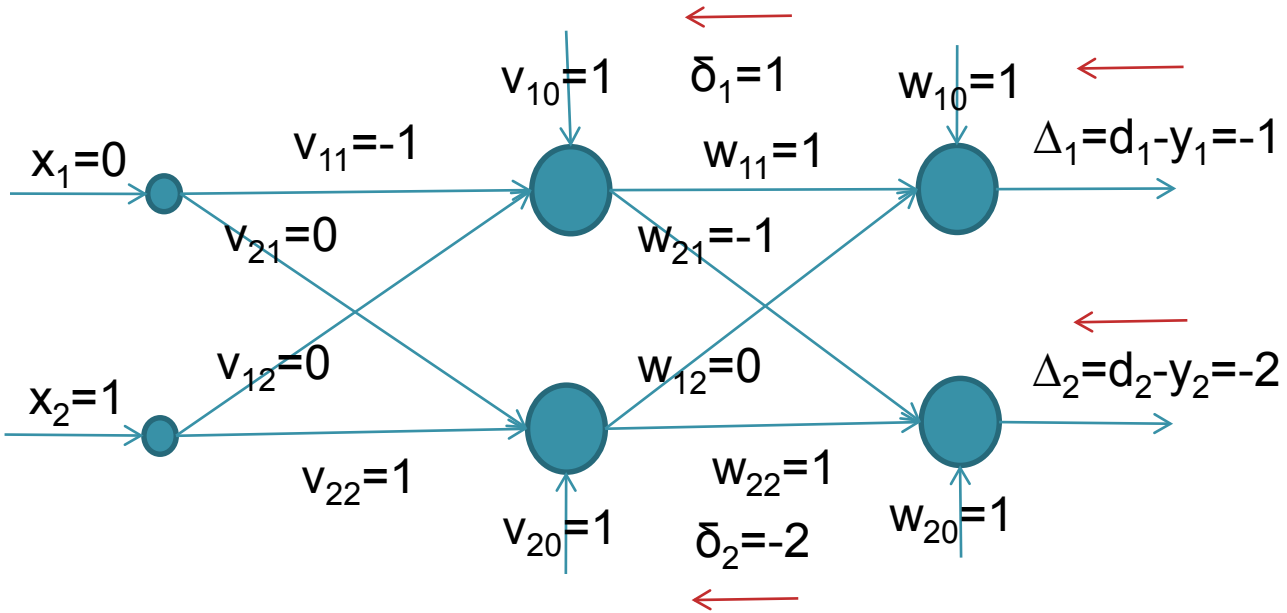


MLP for XOR Example

- Backward pass (hidden-to-input):

$$\delta_1 = \sum_{k=1}^{2} w_{k,1}\Delta_k = w_{1,1}\Delta_1 + w_{2,1}\Delta_2 = 1\cdot -1 + -1\cdot -2 = 1$$

$$\delta_2 = \sum_{k=1}^{2} w_{k,2}\Delta_k = w_{1,2}\Delta_1 + w_{2,2}\Delta_2 = 0\cdot -1 + 1\cdot -2 = -2$$



MLP for XOR Example

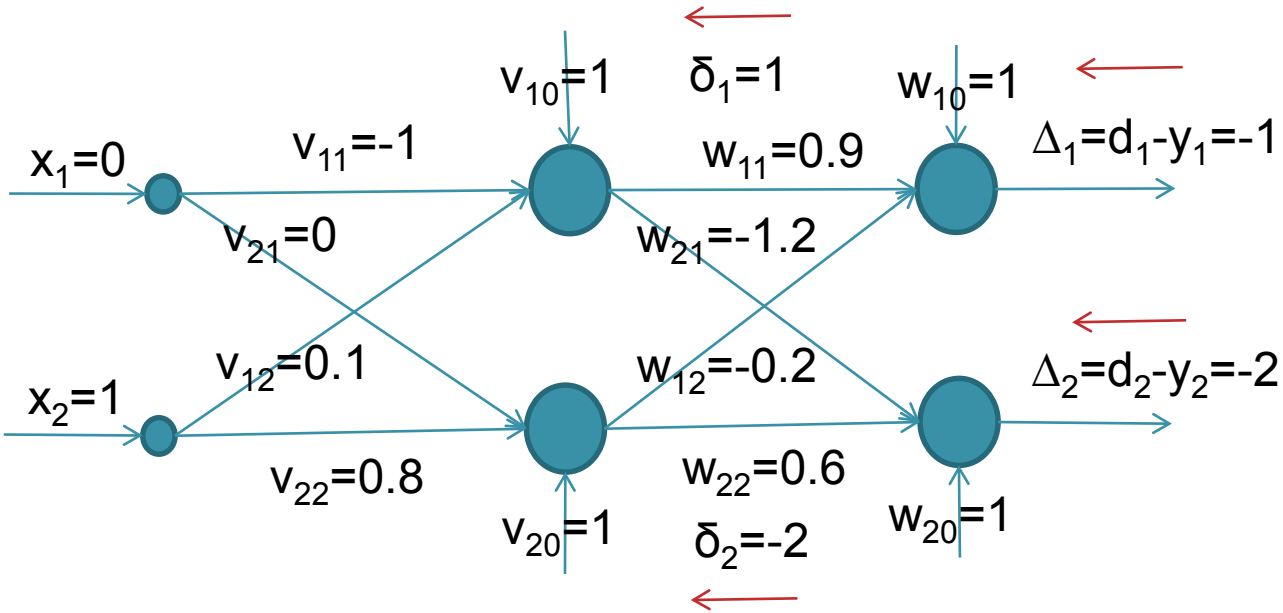
- Backward pass (hidden-to-input):

$$v_{1,1}(t+1) = v_{1,1}(t) + \eta\delta_1(t)x_1(t) = -1 + 0.1 \cdot 1 \cdot 0 = -1$$

$$v_{2,1}(t+1) = v_{2,1}(t) + \eta\delta_2(t)x_1(t) = 0 + 0.1 \cdot -2 \cdot 0 = 0$$

$$v_{1,2}(t+1) = v_{1,2}(t) + \eta\delta_1(t)x_2(t) = 0 + 0.1 \cdot 1 \cdot 1 = 0.1$$

$$v_{2,2}(t+1) = v_{2,2}(t) + \eta\delta_2(t)x_2(t) = 1 + 0.1 \cdot -2 \cdot 1 = 0.8$$



MLP for XOR Example

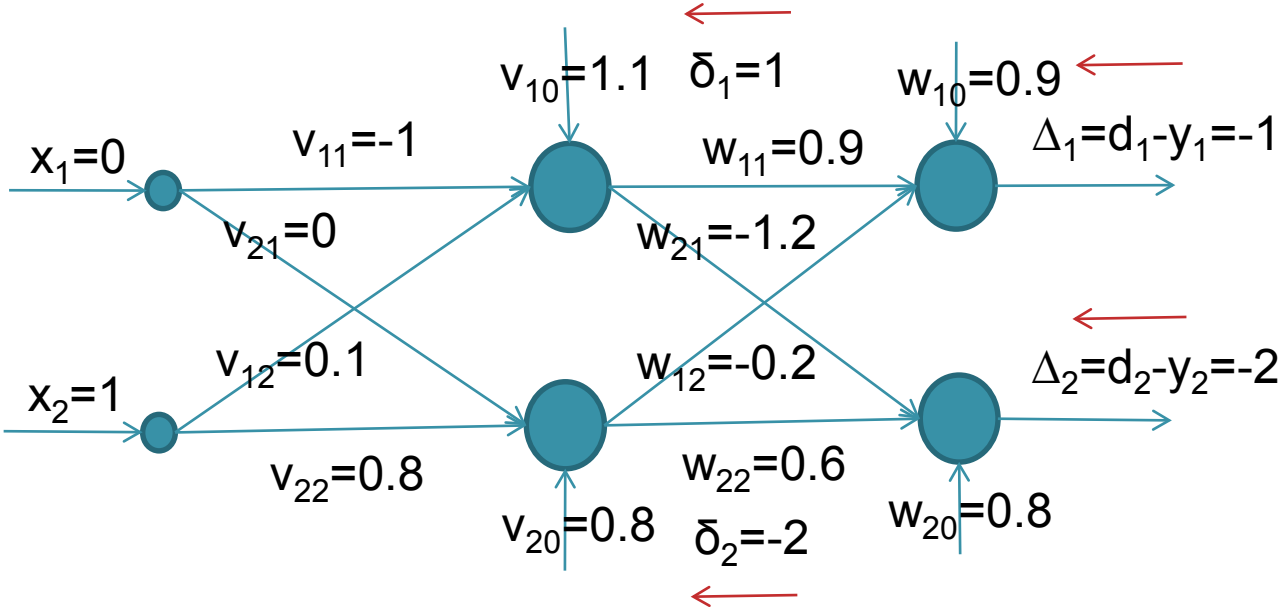
- Remark: the weights multiplied by the zero input are unchanged as they do not contribute to the error (and output signal).

$$v_{1,0}(t+1) = v_{1,0}(t) + \eta\delta_1(t)x_0(t) = 1 + 0.1 \cdot 1 \cdot 1 = 1.1$$

$$v_{2,0}(t+1) = v_{2,0}(t) + \eta\delta_2(t)x_0(t) = 1 + 0.1 \cdot -2 \cdot 1 = 0.8$$

$$w_{1,0}(t+1) = w_{1,0}(t) + \eta\Delta_1(t)z_0(t) = 1 + 0.1 \cdot -1 \cdot 1 = 0.9$$

$$w_{2,0}(t+1) = w_{2,0}(t) + \eta\Delta_2(t)z_0(t) = 1 + 0.1 \cdot -2 \cdot 1 = 0.8$$

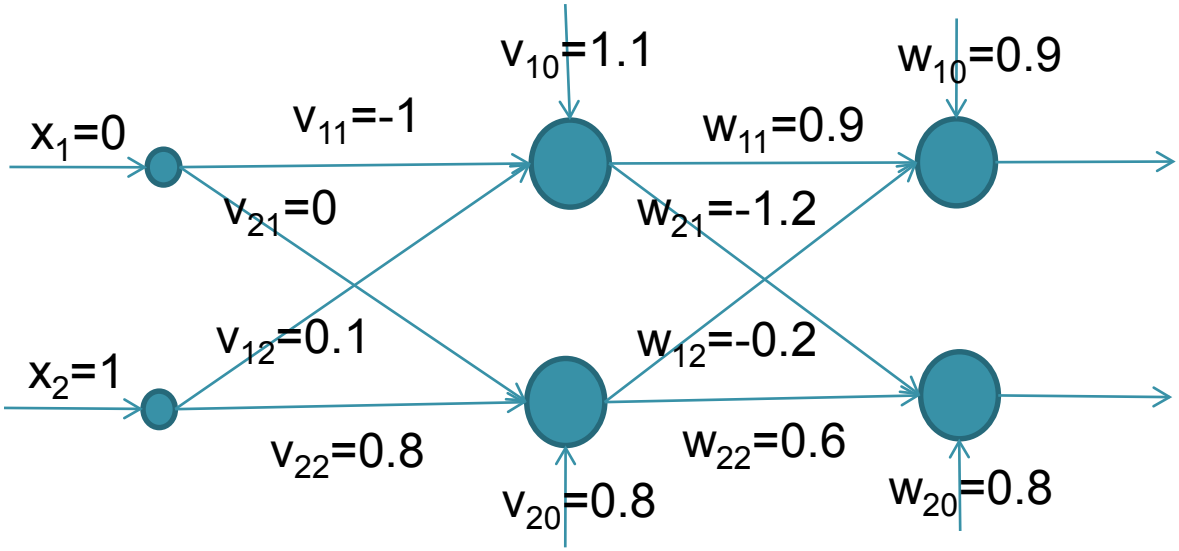


MLP for XOR Example

- Go forward again (normally use a new input):

$$u_1 = x_1 v_{11} + x_2 v_{12} + v_{10} = 0 \cdot -1 + 1 \cdot 0.1 + 1.1 = 1.2, \qquad z_1 = u_1 = 1.2$$

$$u_2 = x_1 v_{21} + x_2 v_{22} + v_{20} = 0 \cdot 0 + 1 \cdot 0.8 + 0.8 = 1.6, \qquad z_2 = u_2 = 1.6$$

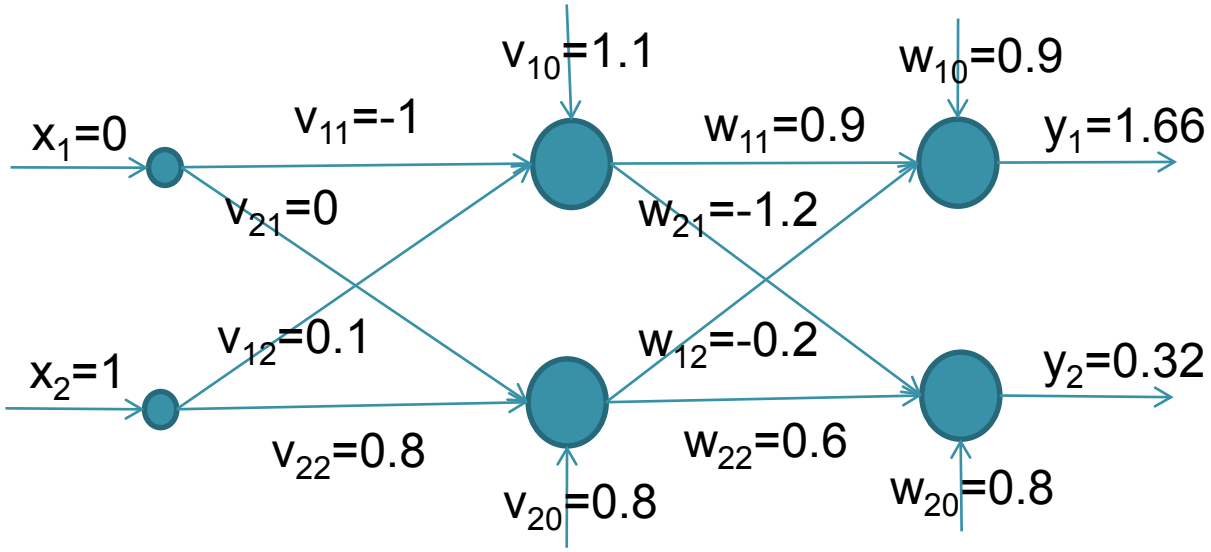


MLP for XOR Example

- Forward pass:

$$a_1 = z_1 w_{11} + z_2 w_{12} + w_{10} = 1.2 \cdot 0.9 + 1.6 \cdot -0.2 + 0.9 = 1.66, \quad y_1 = a_1 = 1.66$$

$$a_2 = z_1 w_{21} + z_2 w_{22} + w_{20} = 1.2 \cdot -1.2 + 1.6 \cdot 0.6 + 0.8 = 0.32, \quad y_2 = a_2 = 0.32$$



- Remark: Outputs now are closer to target value [1 0].

Advantages and Open Issues MLP

- Advantages:
  * Widely applicable (including regression for quantitative traits)
  * Provide posterior Bayes confidence
  * Insensitive to mislabeling and "brain damage"
- Open problems:
  * Nature of "black box" and model complexity
  * Local optima (non-convex, random initialization, learning rate)
  * Overfit (generalization performance, model selection #s)
- Remark:
  * Input range normalization
  * weight pruning (sparse solution, L_1 norm constraint)

Interpretations of MLP

Alternative interpretations of hidden layer/nodes? autoencoder

* 'Local' binary Bayesian classifier (approximation a posterior)
* 'Local' decision boundary
* Space projection/transform (feature extraction)
* What would happen if m1 < m0/m2?