Machine Learning

Computational Linguistics

Emory University

Jinho D. Choi

EMORY
UNIVERSITY

Supervised Learning

input output $y = \pm 1$ ⟵ binomial distribution

$$(X, Y) = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

prediction ⟶ $\hat{y} = f(x)$ ⟵ predicts the output of x

Expected risk ⟶ $E(f) = \int \ell(\hat{y}; y) \cdot P(x, y)$

loss function joint distribution unknown!

Empirical risk ⟶ $\hat{E}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(\hat{y}_i; y_i)$ ⟵ minimize!

EMORY UNIVERSITY

Linear Prediction

$$\hat{E}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(\hat{y}_i; y_i)$$

$$\ell(\hat{y}; y) = \frac{1}{2}(\hat{y} - y)^2$$ ← least squares

$$\hat{y} = f(x) = \mathrm{w}^T \Phi(x) = \mathrm{w}^T \mathrm{x}$$ ← linear function

feature vector

$$\ell(\mathrm{w}, x; y) = \frac{1}{2}(\mathrm{w}^T \mathrm{x} - y)^2$$

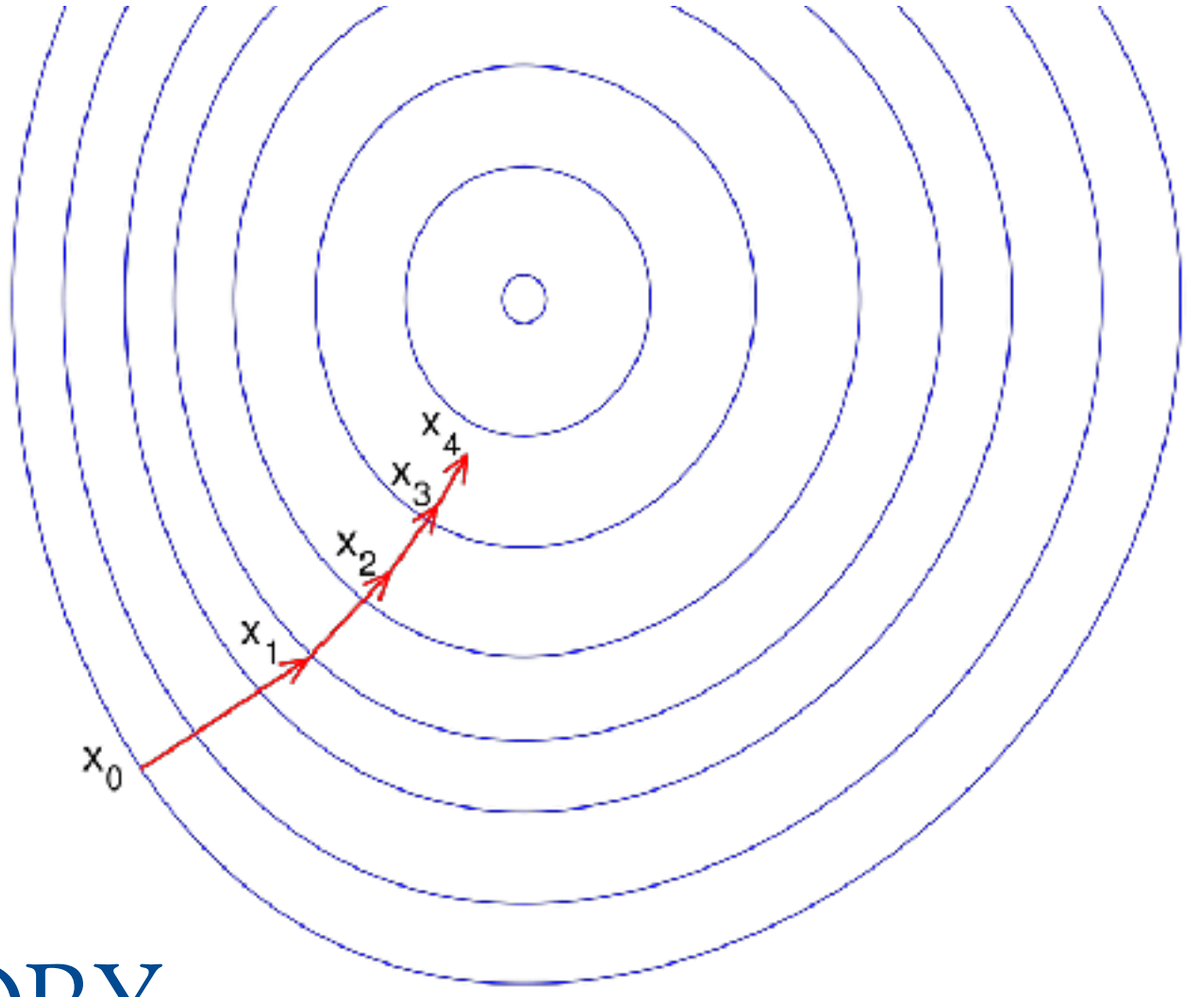
Find a weight vector that minimizes the loss.

Gradient Descent

learning rate

derivative of the loss

$$\mathrm{w}_{t+1} \leftarrow \mathrm{w}_t - \eta_t \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial \mathrm{w}} \ell(\mathrm{w}_t, x_i; y_i)$$



Minimize loss

Derivative → 0

Convex optimization

Global optimum?

Gradient Descent

$$\mathrm{w}_{t+1} \leftarrow \mathrm{w}_t - \eta_t \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial \mathrm{w}} \ell(\mathrm{w}_t, x_i; y_i)$$

$$\ell(\mathrm{w}, x; y) = \frac{1}{2}(\mathrm{w}^T \mathrm{x} - y)^2$$

$$\frac{\partial}{\partial w} \ell(\mathrm{w}, x; y) = \frac{\partial}{\partial \mathrm{w}} \frac{1}{2}(\mathrm{w}^T \mathrm{x} - y)^2 = (\mathrm{w}^T \mathrm{x} - y)\mathrm{x}$$

$$\mathrm{w}_{t+1} \leftarrow \mathrm{w}_t - \eta_t \frac{1}{n} \sum_{i=1}^{n} (\mathrm{w}^T \mathrm{x}_i - y_i)\mathrm{x}_i$$

How often is the weight vector updated?

Stochastic Gradient Descent

$$\mathrm{w}_{t+1} \leftarrow \mathrm{w}_t - \eta_t \frac{1}{n} \sum_{i=1}^{n} (\mathrm{w}^T \mathrm{x}_i - y_i) \mathrm{x}_i$$

$$\mathrm{w}_{t+1} \leftarrow \mathrm{w}_t - \eta_t (\mathrm{w}_t^T \mathrm{x}_i - y_i) \mathrm{x}_i$$

updated for every instance

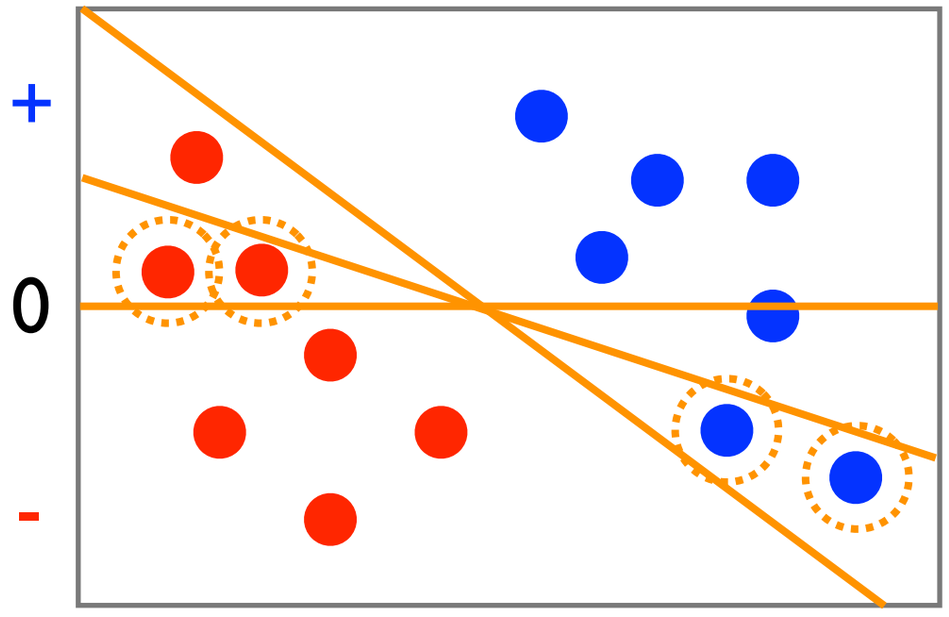
$$\mathrm{w}_0^T \mathrm{x}_1 > 0 \quad \mathrm{w}_1 \leftarrow \mathrm{w}_0 - \eta(\oplus + 1)\mathrm{x}_1$$

$$\mathrm{w}_1^T \mathrm{x}_2 < 0 \quad \mathrm{w}_2 \leftarrow \mathrm{w}_1 - \eta(\ominus + 1)\mathrm{x}_2$$

$$\mathrm{w}_2^T \mathrm{x}_3 < 0 \quad \mathrm{w}_3 \leftarrow \mathrm{w}_2 - \eta(\ominus - 1)\mathrm{x}_3$$

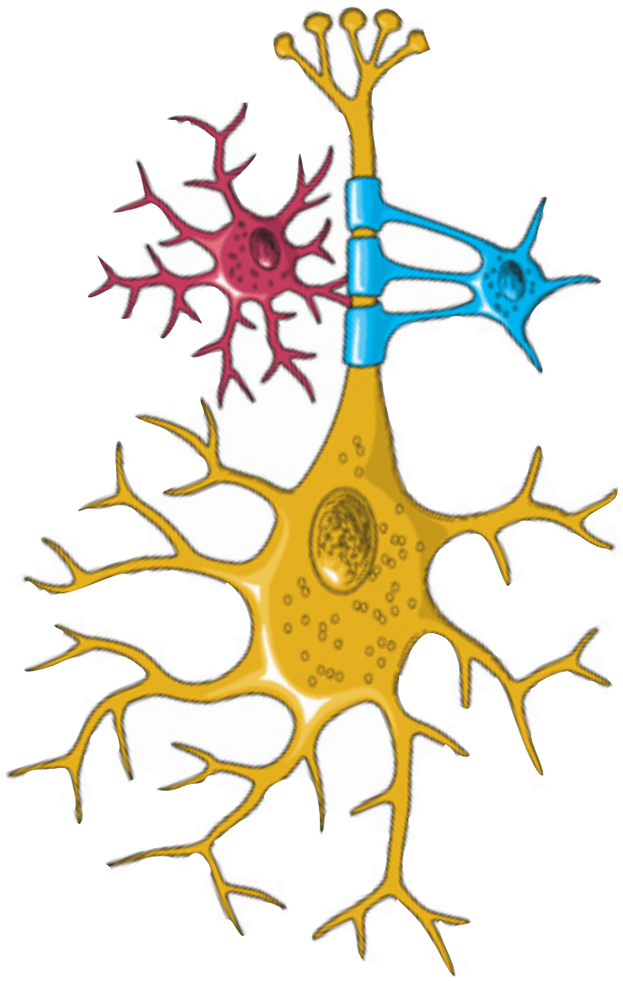
$$\mathrm{w}_3^T \mathrm{x}_4 > 0 \quad \mathrm{w}_4 \leftarrow \mathrm{w}_3 - \eta(\oplus - 1)\mathrm{x}_4$$

$$\mathrm{w}_0 \leftarrow 0$$

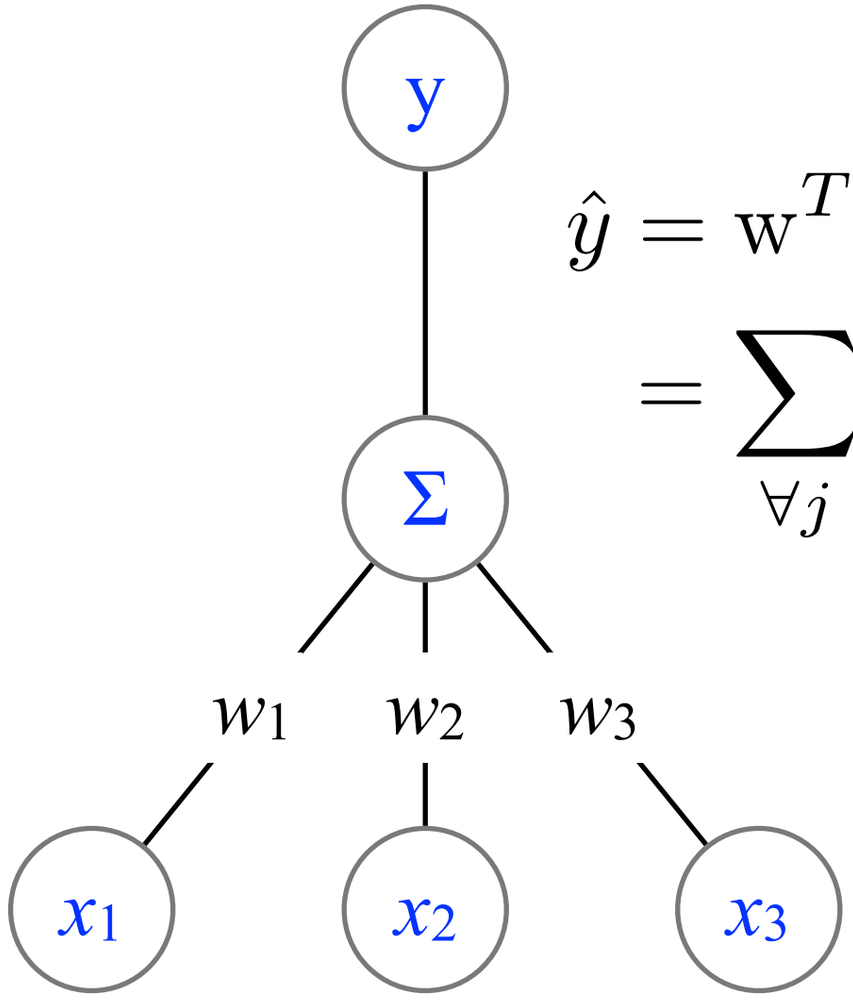


Perceptron

Neuron



Perceptron



Perceptron

Stochastic gradient descent

$$\mathrm{w}_{t+1} \leftarrow \mathrm{w}_t - \eta_t \Delta\ell$$

Least squares

$$\ell(\mathrm{w}, x; y) = \frac{1}{2}(\mathrm{w}^T\mathrm{x} - y)^2$$

$$\Delta\ell = (\mathrm{w}^T\mathrm{x} - y)\mathrm{x}$$

Perceptron

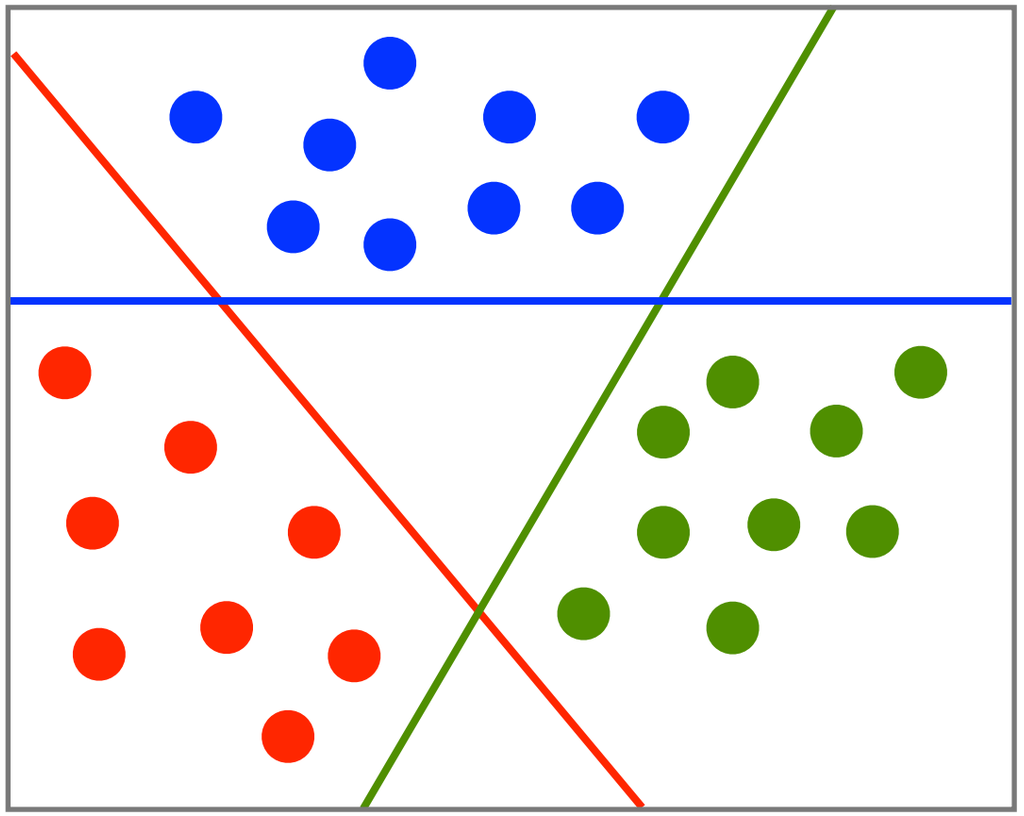
$$\ell(\mathrm{w}, \mathrm{x}; y) = \max\{0, -\mathrm{w}^T\mathrm{x} \cdot y\}$$

$$\Delta\ell = \begin{cases} -\mathrm{x} \cdot y & \mathrm{w}^T\mathrm{x} \cdot y < 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{w}_{t+1} \leftarrow \mathrm{w}_t + \eta_t \begin{cases} \mathrm{x} \cdot y & \mathrm{w}_t^T\mathrm{x} \cdot y < 0 \\ 0 & \text{otherwise} \end{cases}$$

Multinomial Perceptron

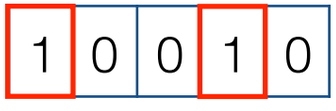
Binomial distribution requires
1 hyperplane to separate 2 classes.



How many for
m classes?

Multinomial distribution requires
m hyperplanes to separate *m* classes.

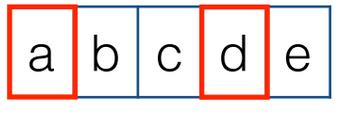
Multinomial Perceptron

$\mathrm{x} =$ 

5 features (including bias)

Binomial

$y = \{-1, 1\}$

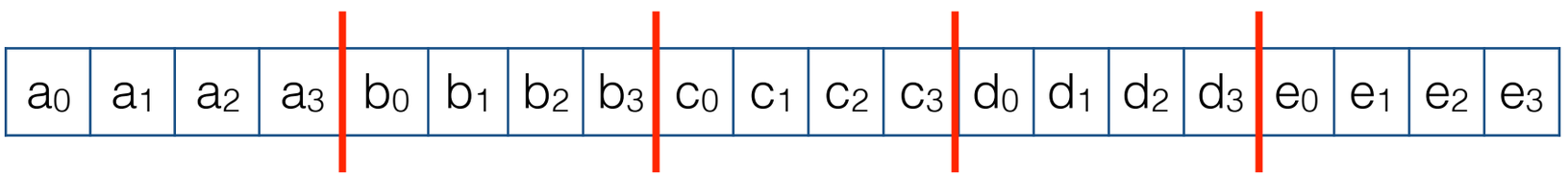
$\mathrm{w} =$ 

$\mathrm{w}^T \mathrm{x} = a + d$

$$\hat{y} = \begin{cases} 1 & \mathrm{w}^T \mathrm{x} \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Multinomial

$y = \{0, 1, 2, 3\}$

$\mathrm{w} =$ 

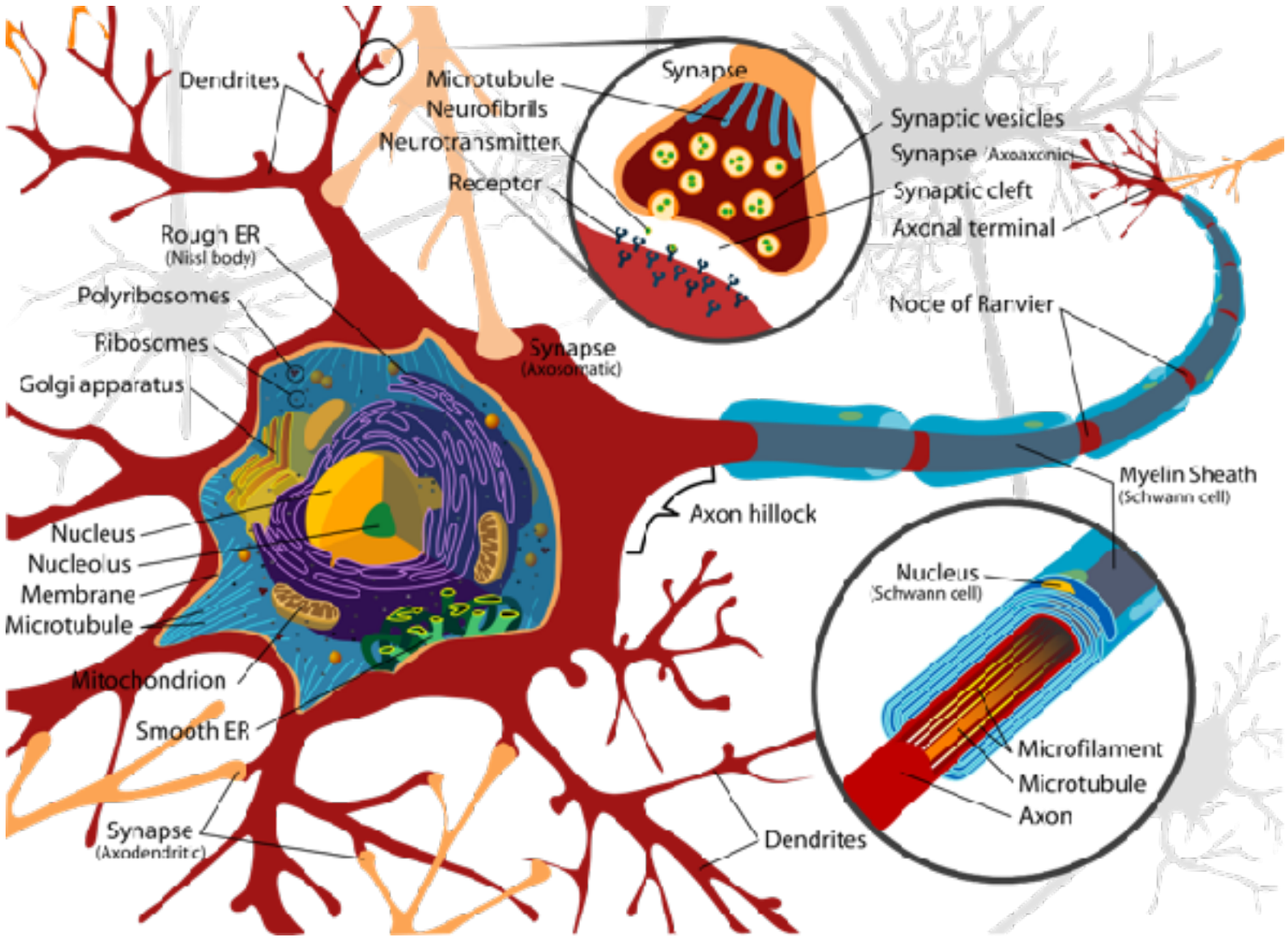
$\mathrm{w}_y^T \mathrm{x} = a_y + d_y$

$$\hat{y} = \arg\max_y \mathrm{w}_y^T \mathrm{x}$$

Human Brain

Learn

Adapt



Robust

Fault Tolerant

How many neurons?

How many synapsis per neuron?

Hebbian Learning

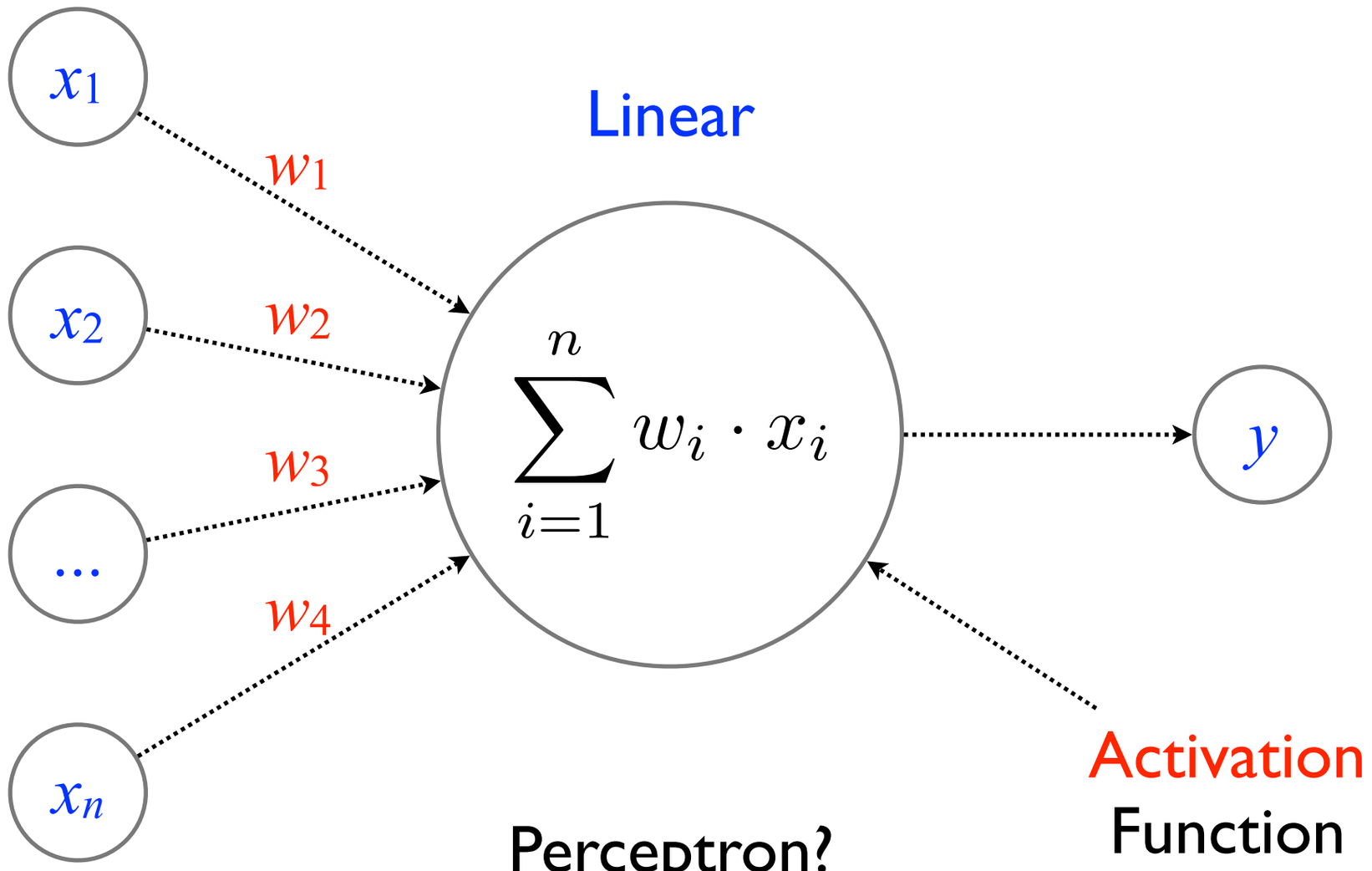
Hebb's Learning Rule

The connections between two neurons is strengthened
if the neurons fire simultaneously.

Synchronous activation
increases the synaptic strength.

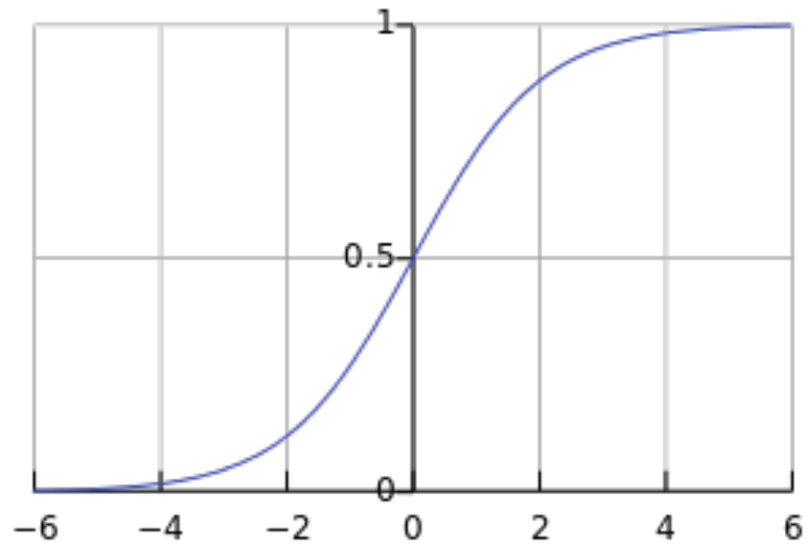
Asynchronous activation
decreases the synaptic strength.

Single-layer Neural Network



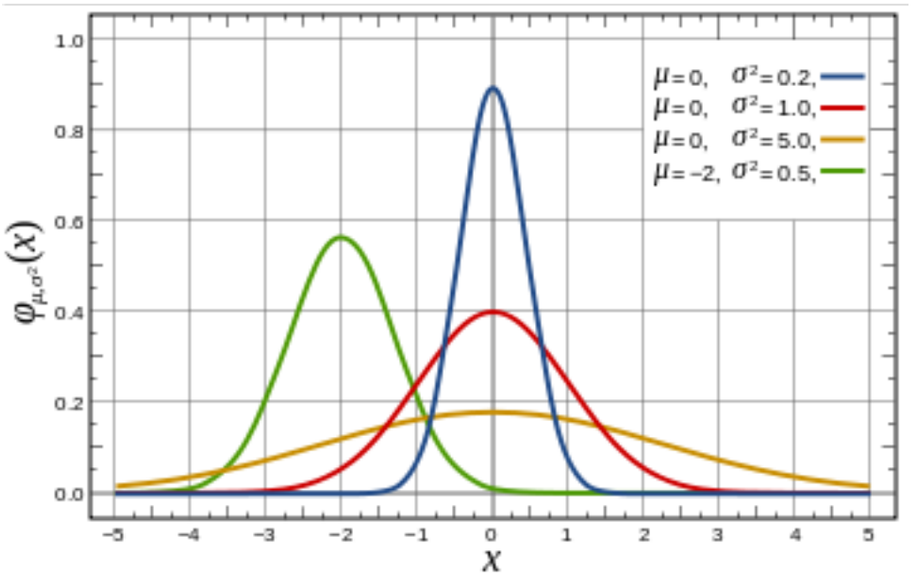
Non-Linear Function

Sigmoid



$$\frac{1}{1 + e^{-\mathrm{w}\cdot\mathrm{x}}}$$

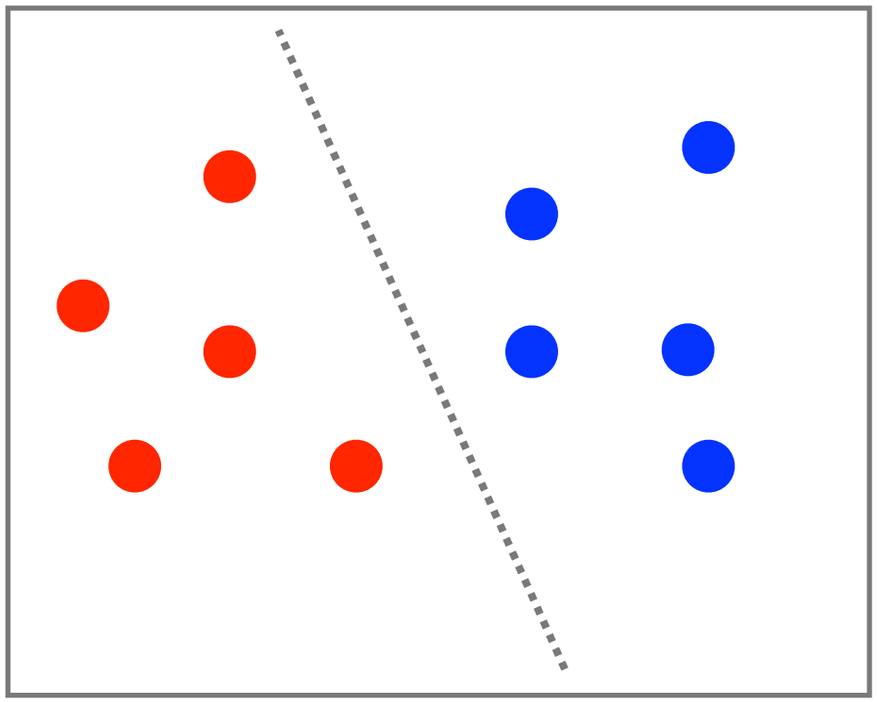
Gaussian



$$e^{-\frac{\|\mathrm{w}-\mathrm{x}\|^2}{2a^2}}$$

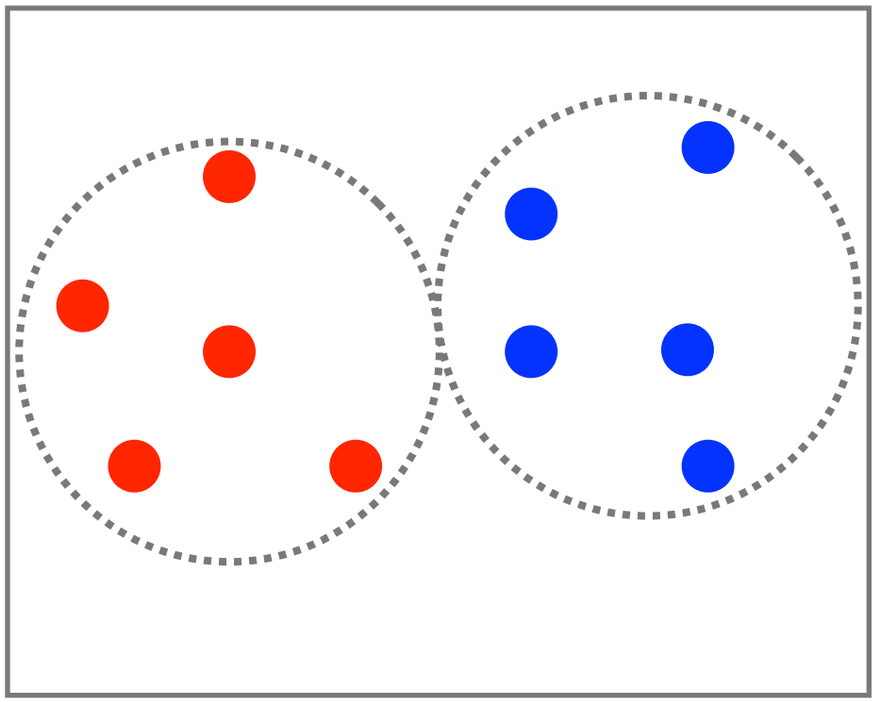
Sigmoid vs. Gaussian

Sigmoid



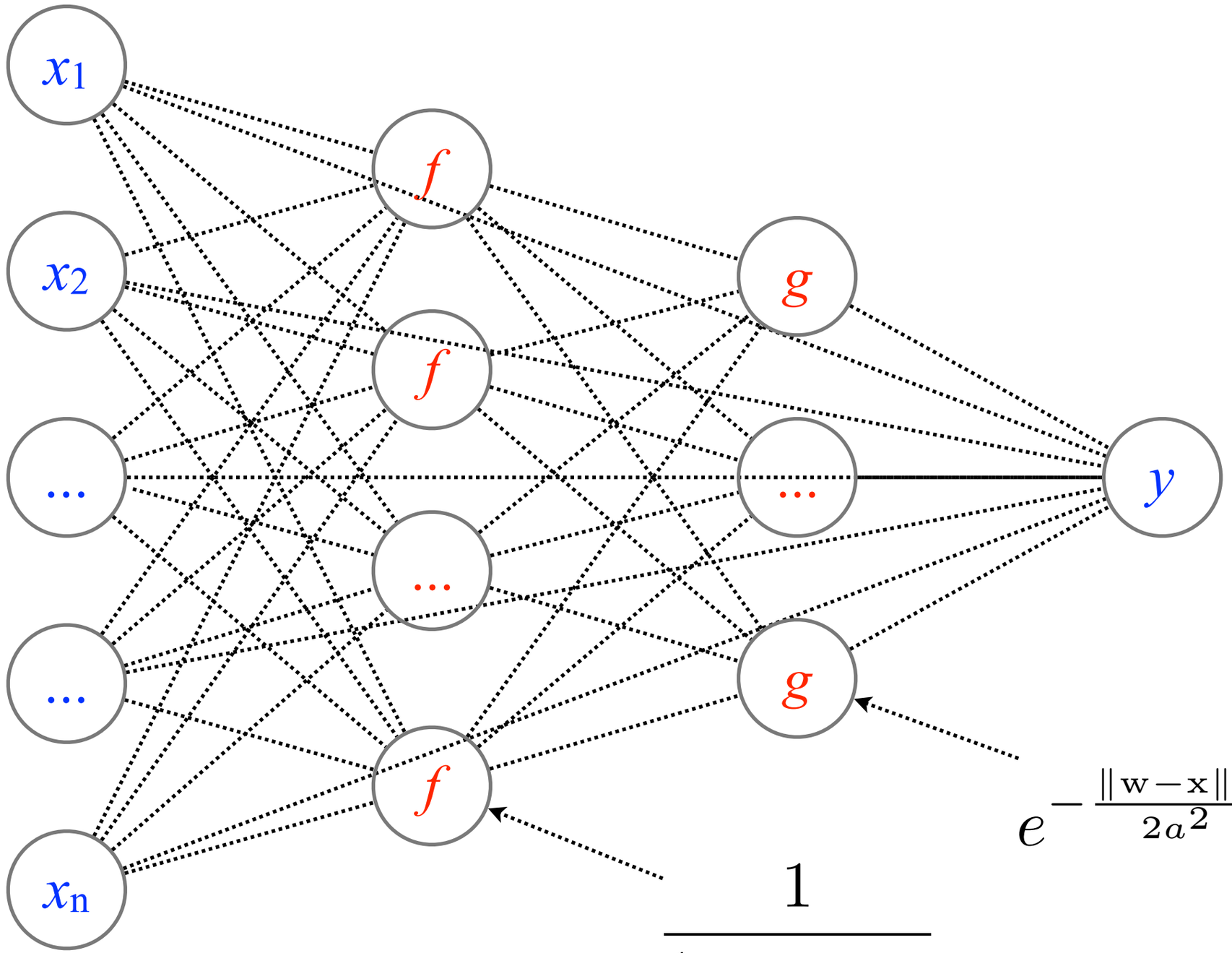
Find hyperplanes

Gaussian



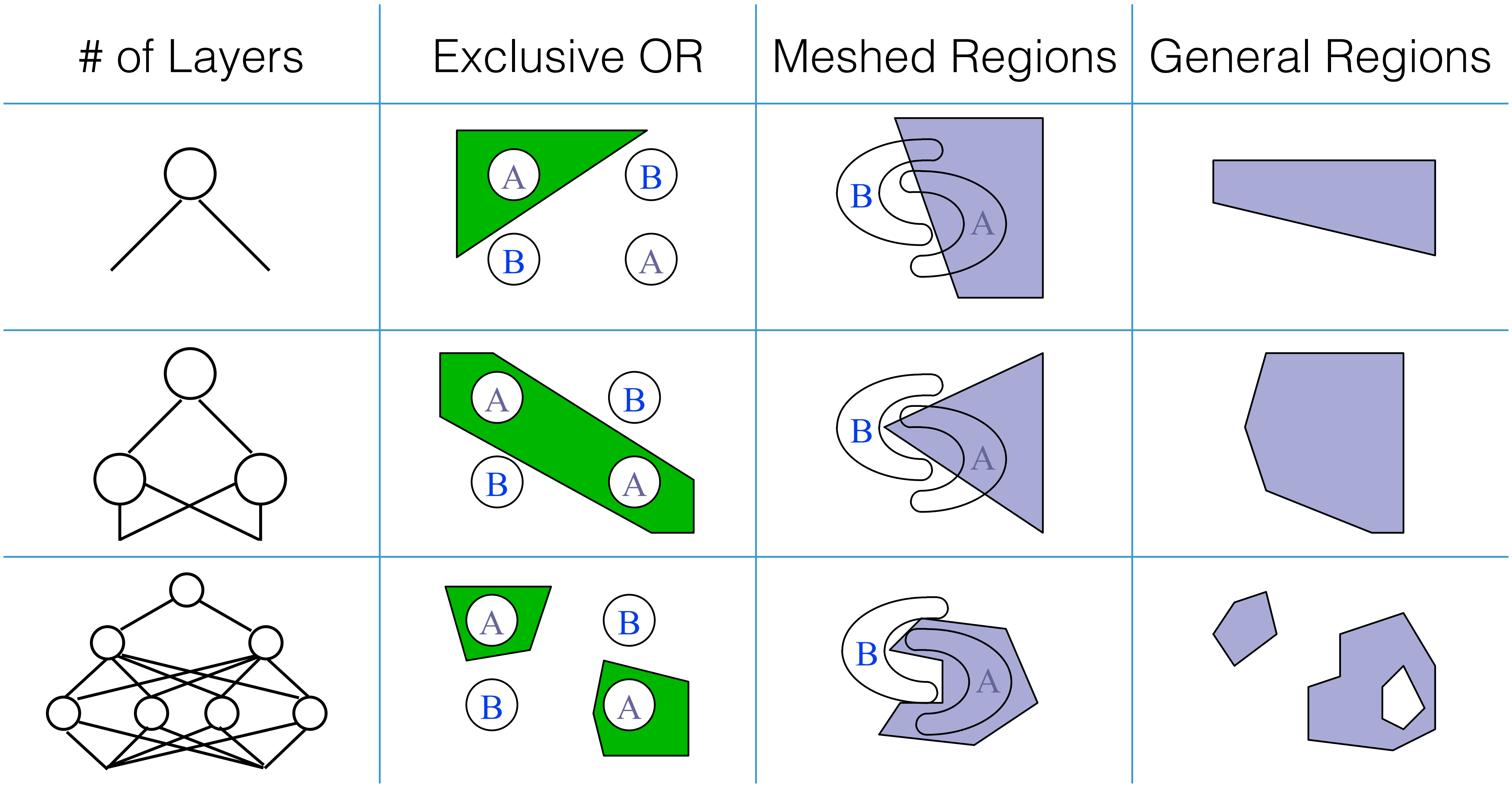
Find hyperspheres

Multi-layer Neural Network



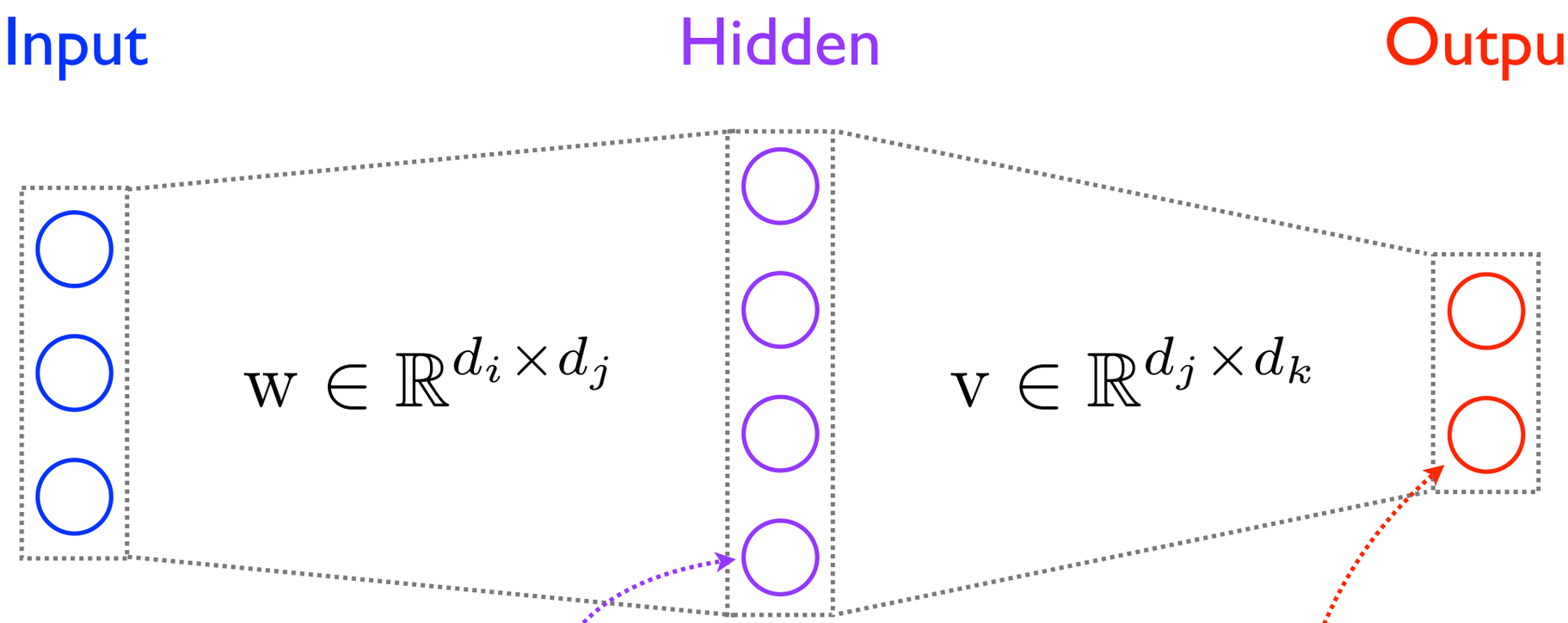
Multi-layer Neural Network

Comparison between different layers



EMORY UNIVERSITY

Forward Propagation



Activation

$$h_j = \varphi(\mathbf{w}_j \cdot \mathbf{x}) \qquad \hat{y}_k = \varphi(\mathbf{v}_k \cdot \mathbf{h})$$

$$h_j = \varphi(\sum_{i=1}^{d_i} w_{ij} \cdot x_i) \qquad \hat{y}_k = \varphi(\sum_{j=1}^{d_j} v_{jk} \cdot h_j)$$

Backward Propagation

$$\hat{y}_k = \varphi\left(\sum_{j=1}^{d_j} v_{jk} \cdot h_j\right)$$

j'th hidden neuron

↓

k'th output neuron

$$\mathcal{E}(y_k, \hat{y}_k) = \frac{1}{2}(\hat{y}_k - y_k)^2$$

Least Square

$$z_k = \mathrm{v}_k \cdot \mathrm{h}$$

$$\mathcal{E} = \frac{1}{2}(\varphi(z_k) - y_k)^2$$

$$\frac{\partial \mathcal{E}}{\partial v_{jk}} = \frac{\partial \mathcal{E}}{\partial \hat{y}_k} \cdot \frac{\partial \hat{y}_k}{\partial z_k} \cdot \frac{\partial z_k}{\partial v_{jk}}$$

Backward Propagation

$$\frac{\partial \mathcal{E}}{\partial v_{jk}} = \frac{\partial \mathcal{E}}{\partial \hat{y}_k} \cdot \frac{\partial \hat{y}_k}{\partial z_k} \cdot \frac{\partial z_k}{\partial v_{jk}} \qquad \varphi(z_k) = \frac{1}{1 + e^{-z_k}}$$

$$\frac{\partial}{\partial v_{jk}} z_k = \frac{\partial}{\partial v_{jk}} \left(\sum_{l=1}^{d_j} v_{lk} \cdot h_l \right) = h_j$$

$$\frac{\partial}{\partial z_k} \hat{y}_k = \frac{\partial}{\partial z_k} \varphi(z_k) = \varphi(z_k)(1 - \varphi(z_k)) = \hat{y}_k(1 - \hat{y}_k)$$

$$\frac{\partial}{\partial \hat{y}_k} \mathcal{E} = \frac{\partial}{\partial \hat{y}_k} \frac{1}{2}(\hat{y}_k - y_k)^2 = \hat{y}_k - y_k$$

$$\Delta \mathcal{E}_{jk} = (\hat{y}_k - y_k) \cdot \hat{y}_k(1 - \hat{y}_k) \cdot h_j$$

EMORY UNIVERSITY

Backward Propagation

$$\Delta \mathcal{E}_{jk} = \boxed{(\hat{y}_k - y_k)} \cdot \boxed{\hat{y}_k(1-\hat{y}_k)} \cdot \boxed{h_j}$$

$$\hat{y}_k = \varphi(\sum_{j=1}^{d_j} v_{jk} \cdot h_j)$$

$$h_j = \varphi(\sum_{i=1}^{d_i} w_{ij} \cdot x_i)$$

$$\Delta \mathcal{E}_{ij} = \boxed{(h_j - ?)} \cdot \boxed{h_j(1-h_j)} \cdot \boxed{x_i}$$

Backward Propagation

$$\Delta\mathcal{E}_{jk} = \boxed{(\hat{y}_k - y_k) \cdot \hat{y}_k(1 - \hat{y}_k)} \cdot h_j$$

$$\delta_k$$

$$\Delta\mathcal{E}_{ij} = \boxed{(h_j - ?)} \cdot h_j(1 - h_j) \cdot x_i$$

$$\frac{\partial\mathcal{E}}{\partial h_j} = \boxed{\sum_{l=1}^{d_k}(\delta_l \cdot v_{jl})}$$

Sum of all the gradients
made by the j'th hidden unit

$$\Delta\mathcal{E}_{ij} = \sum_{l=1}^{d_k}(\delta_l \cdot v_{jl}) \cdot h_j(1 - h_j) \cdot x_i$$

Backward Propagation

Input → Hidden

$$h_j = \varphi(\sum_{i=1}^{d_i} w_{ij} \cdot x_i)$$

Hidden → Output

$$\hat{y}_k = \varphi(\sum_{j=1}^{d_j} v_{jk} \cdot h_j)$$

$$\delta_k = (\hat{y}_k - y_k) \cdot \hat{y}_k(1 - \hat{y}_k)$$

$$\Delta\mathcal{E}_{ij} = \sum_{l=1}^{d_k} (\delta_l \cdot v_{jl}) \cdot h_j(1 - h_j) \cdot x_i$$

$$\Delta\mathcal{E}_{jk} = \delta_k \cdot h_j$$

$$w_{ij}^{t+1} \leftarrow w_{ij}^{t} - \eta_t \Delta_{ij}$$

$$v_{jk}^{t+1} \leftarrow v_{jk}^{t} - \eta_t \Delta_{jk}$$