

# Machine Learning

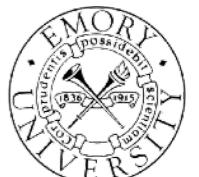
Computational Linguistics

Emory University

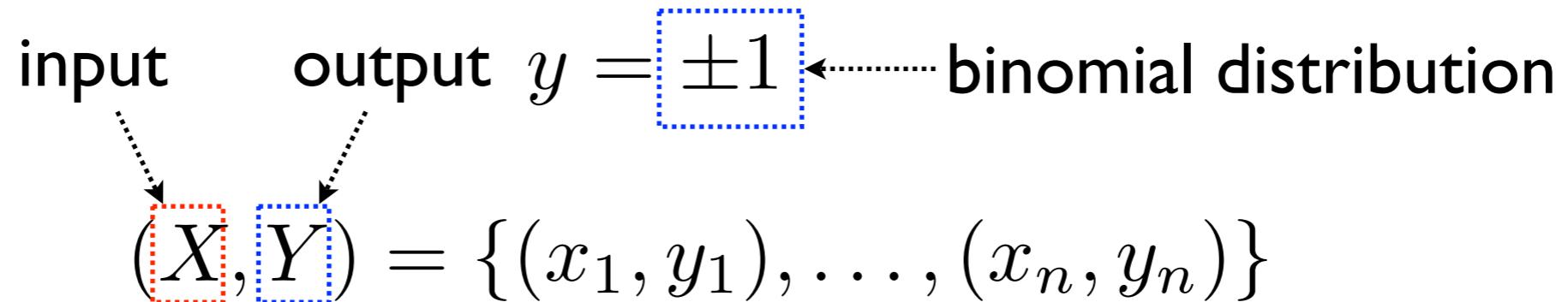
Jinho D. Choi



**EMORY**  
UNIVERSITY



# Supervised Learning



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

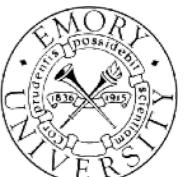
Expected risk  $\rightarrow E(f) = \int \boxed{\ell(\hat{y}; y)} \cdot \boxed{P(x, y)}$

loss function      joint distribution **unknown!**

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



EMORY  
UNIVERSITY



# Linear Prediction

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

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

$$\hat{y} = f(x) = \mathbf{w}^T \Phi(x) = \mathbf{w}^T \mathbf{x}$$

↑  
feature vector

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

Find a **weight vector** that minimizes the **loss**.

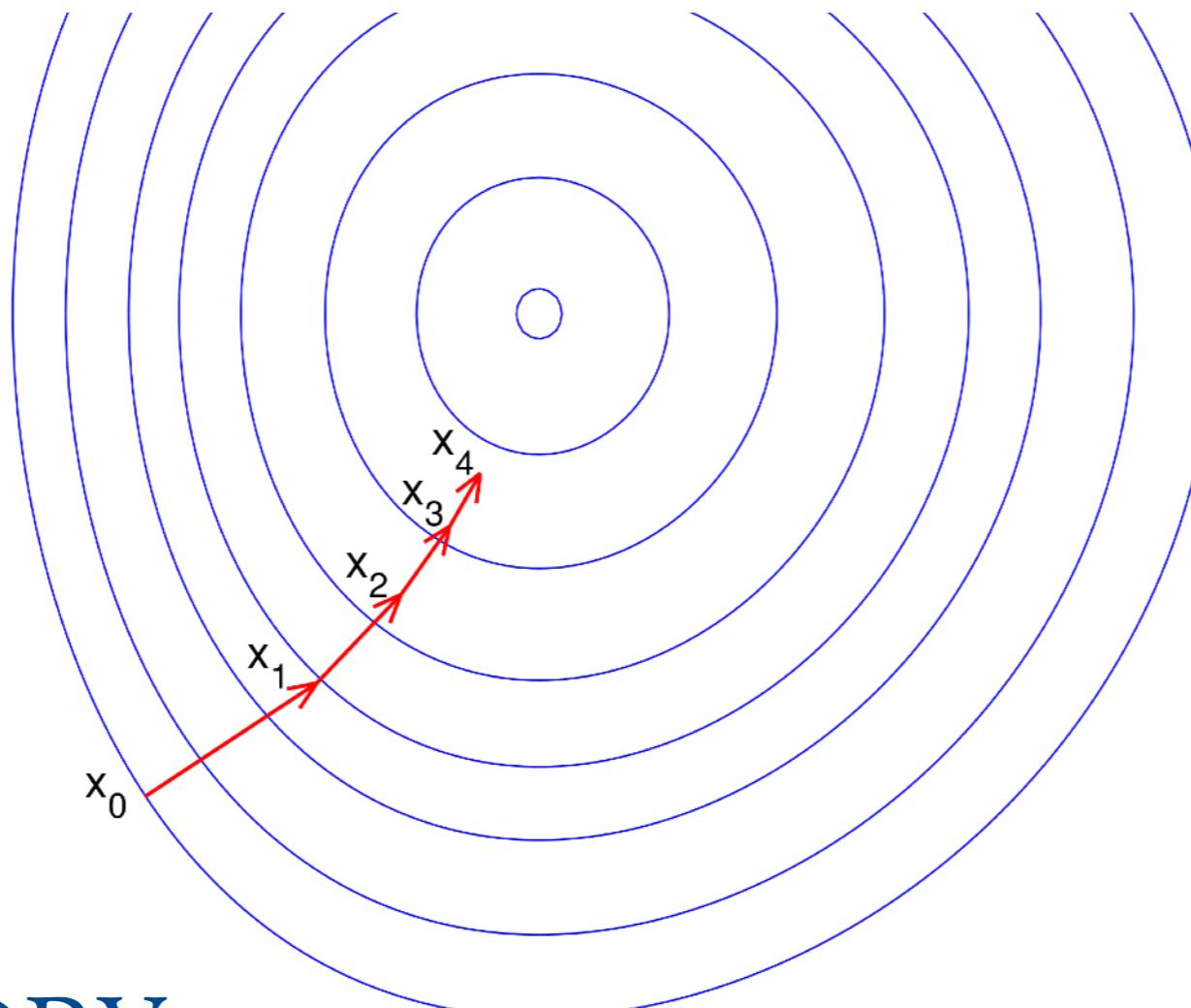


# Gradient Descent

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

learning rate

derivative of the loss



Minimize loss

Derivative  $\rightarrow 0$

Convex optimization

Global optimum?



# Gradient Descent

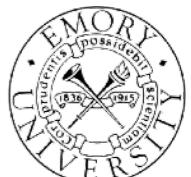
$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{w}_t, \mathbf{x}_i; y_i)$$

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

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

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

How often is the **weight vector** updated?



# Stochastic Gradient Descent

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

updated for **every** instance

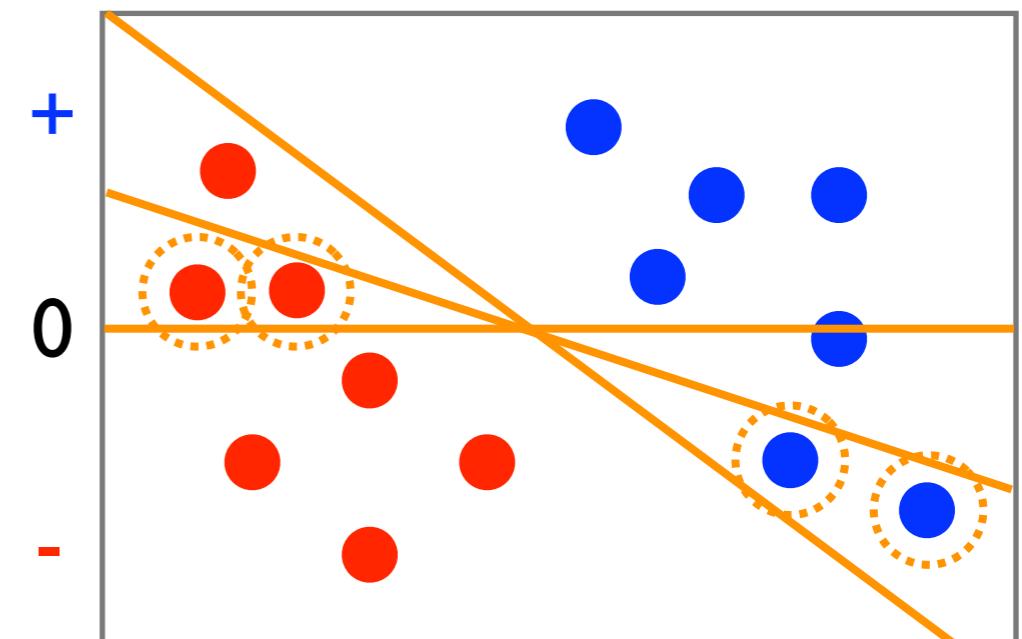
$$\mathbf{w}_0^T \mathbf{x}_1 > 0 \quad \mathbf{w}_1 \leftarrow \mathbf{w}_0 - \eta (\oplus + 1) \mathbf{x}_1$$

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

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

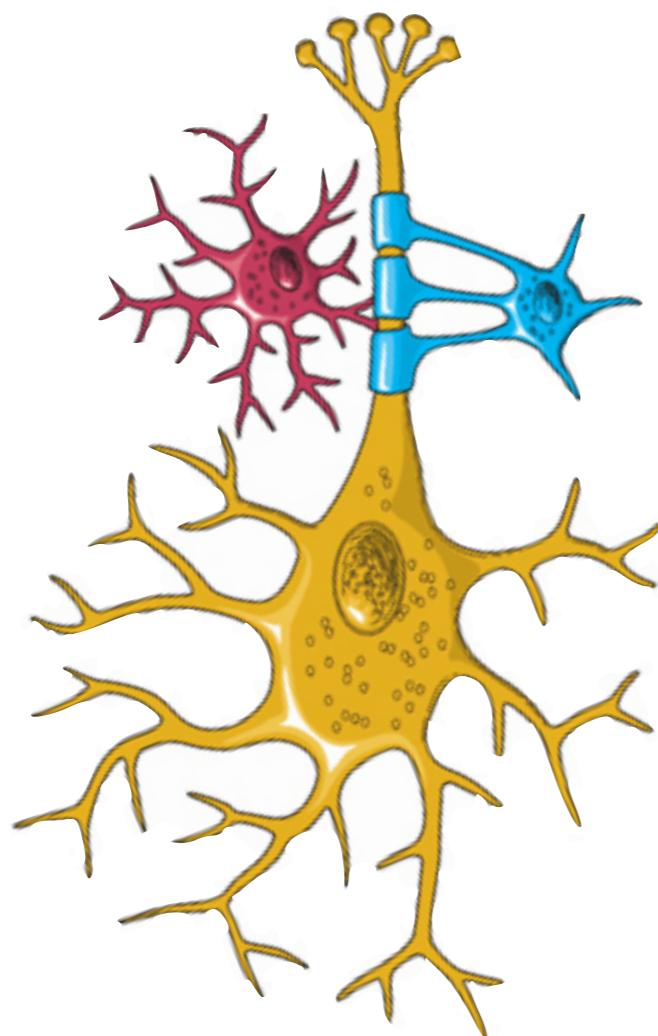
$$\mathbf{w}_3^T \mathbf{x}_4 > 0 \quad \mathbf{w}_4 \leftarrow \mathbf{w}_3 - \eta (\oplus - 1) \mathbf{x}_4$$

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

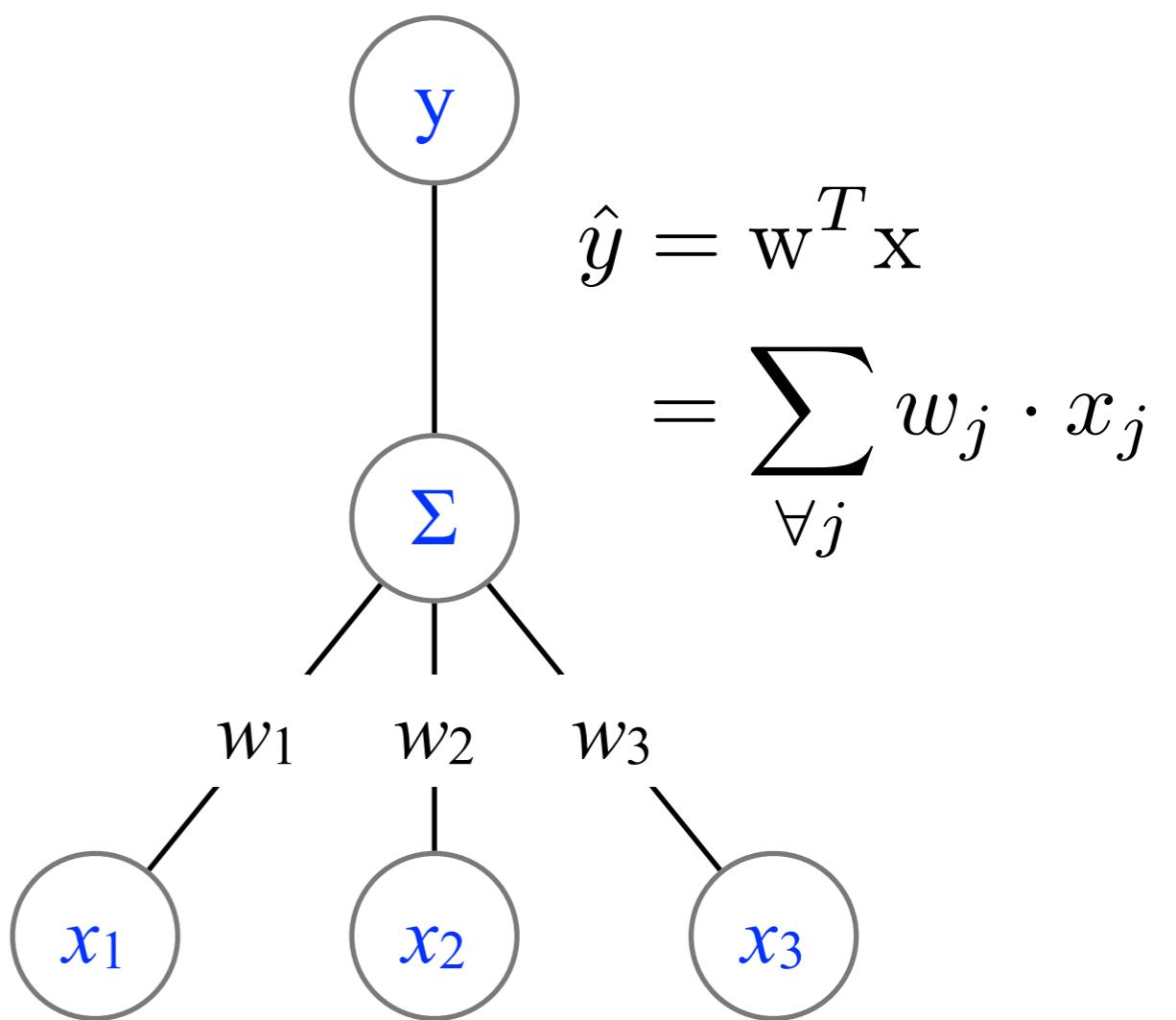


# Perceptron

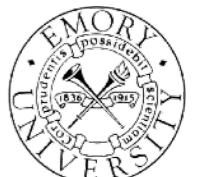
Neuron



Perceptron



EMORY  
UNIVERSITY



# Perceptron

Stochastic gradient descent

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

Least squares

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

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

Perceptron

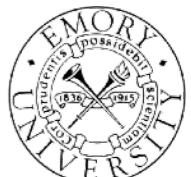
$$\ell(\mathbf{w}, \mathbf{x}; y) = \boxed{\max\{0, -\mathbf{w}^T \mathbf{x} \cdot y\}}$$

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

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

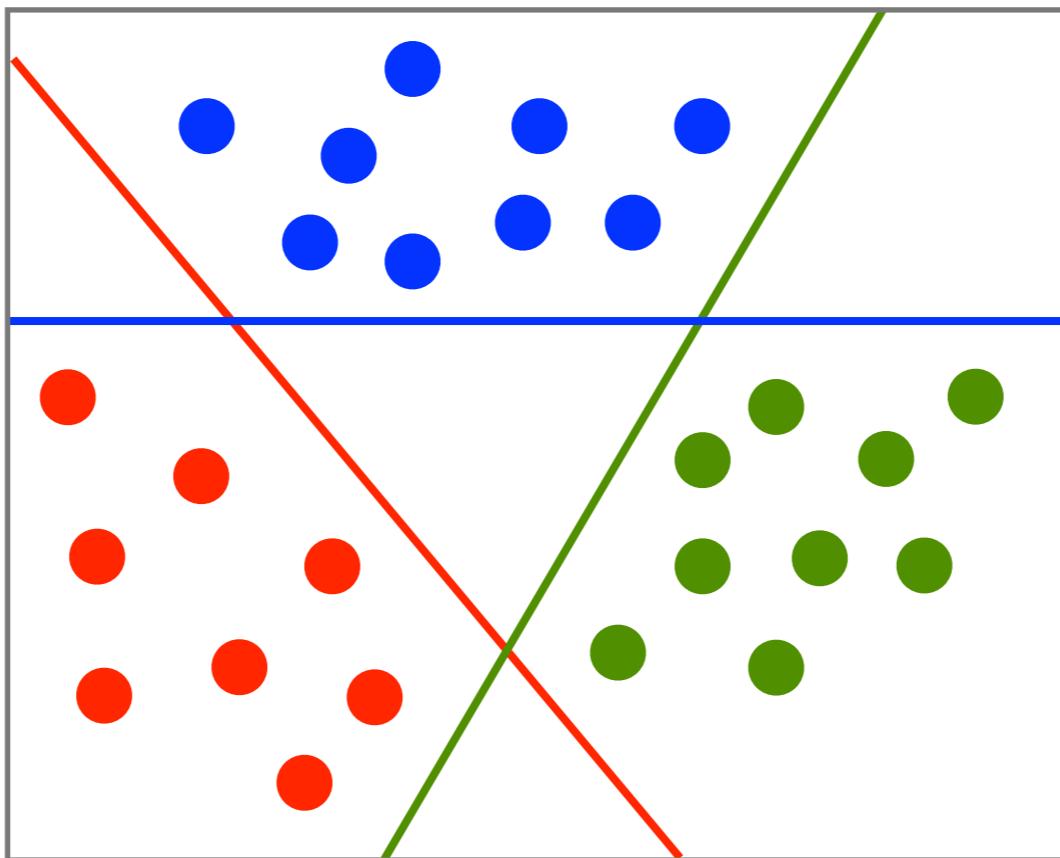


EMORY  
UNIVERSITY



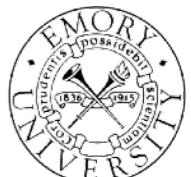
# Multinomial Perceptron

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

$$\mathbf{x} = \boxed{1} \boxed{0} \boxed{0} \boxed{1} \boxed{0}$$

5 features (including bias)

Binomial

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

$$\mathbf{w} = \boxed{a} \boxed{b} \boxed{c} \boxed{d} \boxed{e}$$

$$\mathbf{w}^T \mathbf{x} = a + d \quad \hat{y} = \begin{cases} 1 & \mathbf{w}^T \mathbf{x} \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Multinomial

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

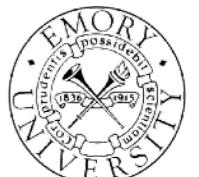
$$\mathbf{w} = \boxed{a_0} \boxed{a_1} \boxed{a_2} \boxed{a_3} \boxed{} \boxed{b_0} \boxed{b_1} \boxed{b_2} \boxed{b_3} \boxed{} \boxed{c_0} \boxed{c_1} \boxed{c_2} \boxed{c_3} \boxed{} \boxed{d_0} \boxed{d_1} \boxed{d_2} \boxed{d_3} \boxed{} \boxed{e_0} \boxed{e_1} \boxed{e_2} \boxed{e_3}$$

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

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



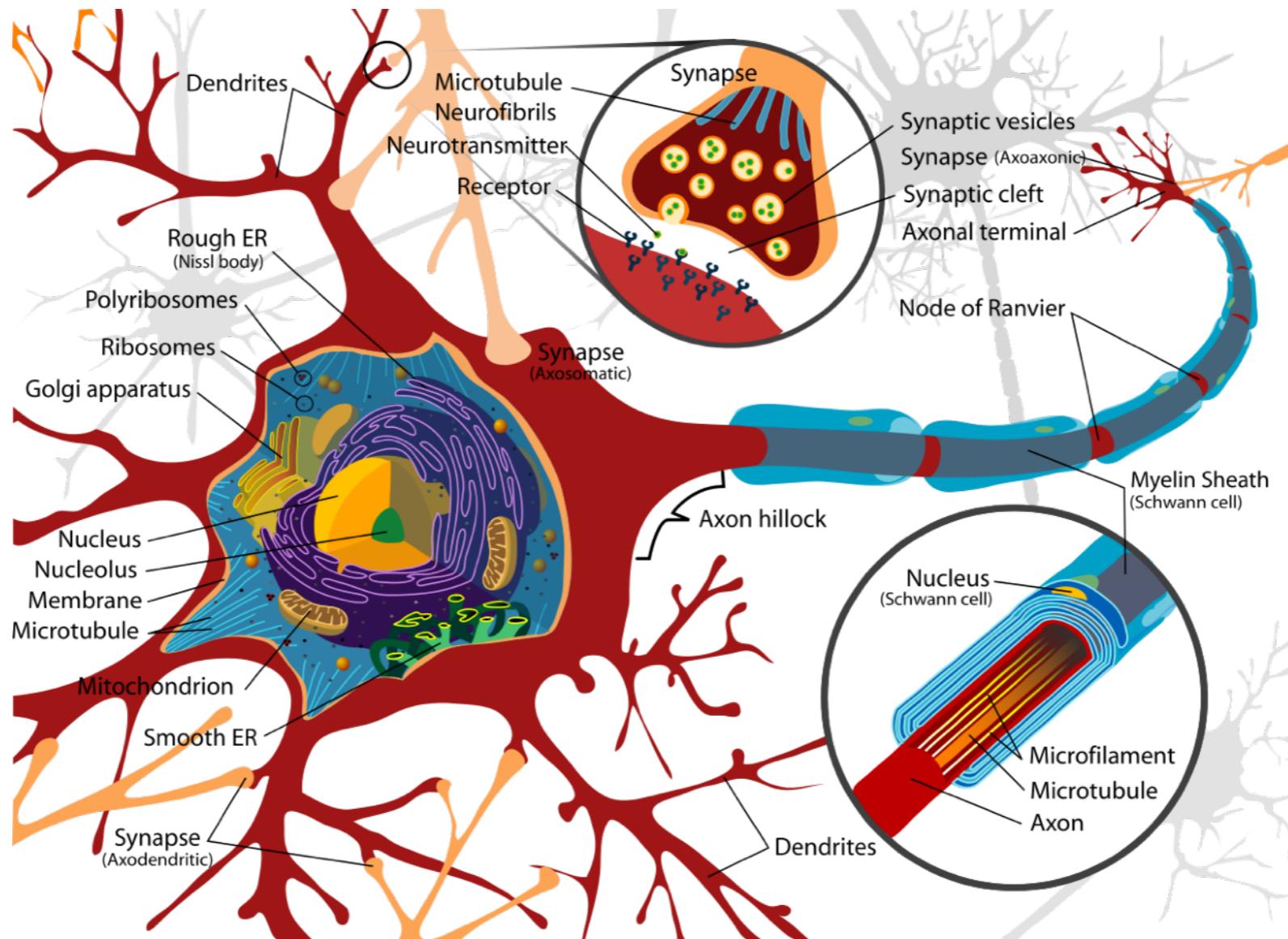
EMORY  
UNIVERSITY



# Human Brain

Learn  
Adapt

Robust  
Fault  
Tolerant

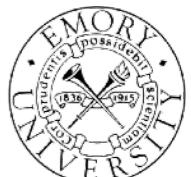


How many **neurons**?

How many **synapsis** per **neuron**?



EMORY  
UNIVERSITY



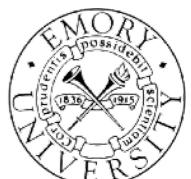
# Hebbian Learning

## Hebb's Learning Rule

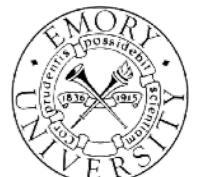
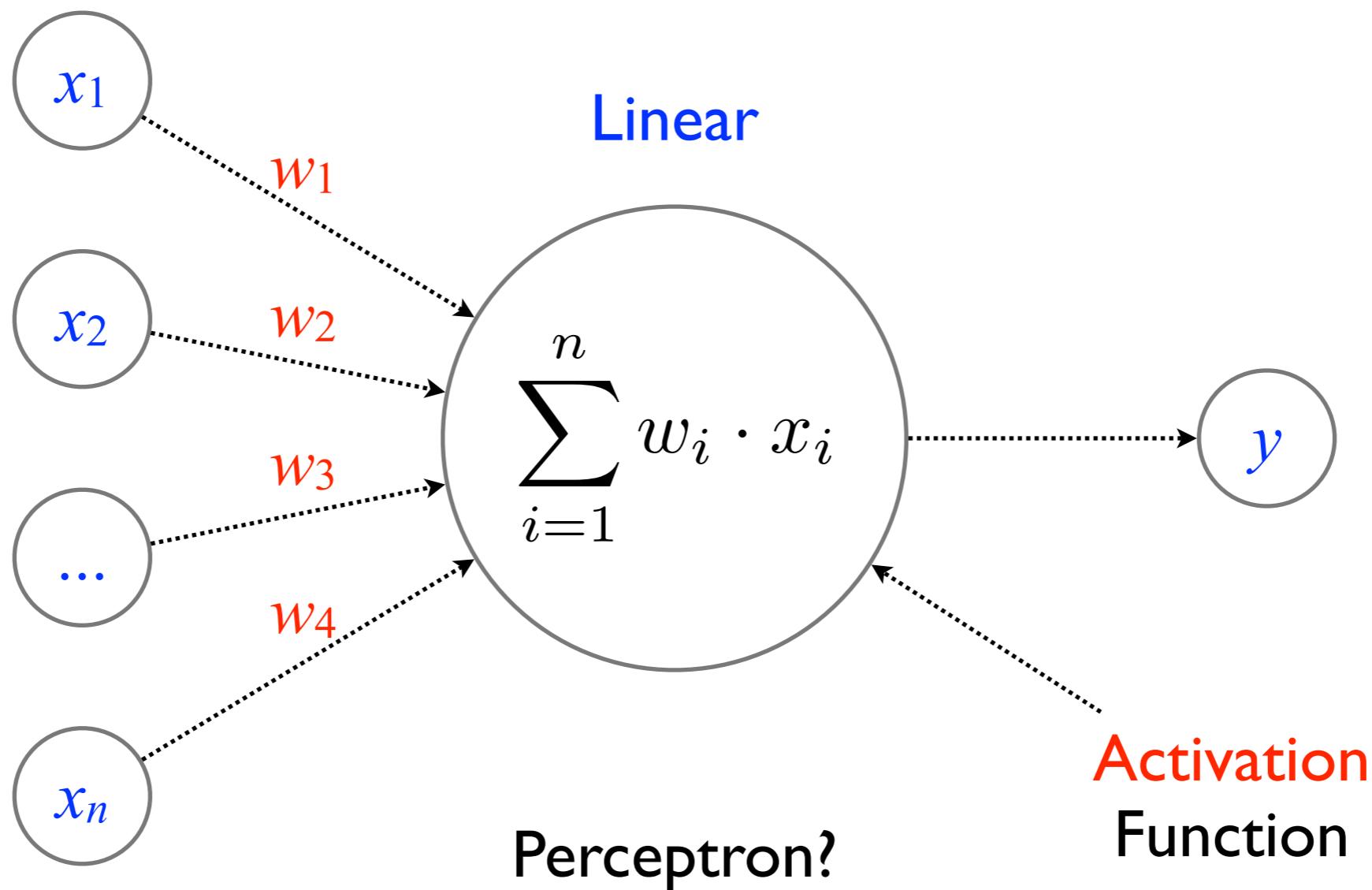
The **connections** between two **neurons** is strengthened if the **neurons** fire **simultaneously**.

**Synchronous** activation  
**increases** the synaptic strength.

**Asynchronous** activation  
**decreases** the synaptic strength.

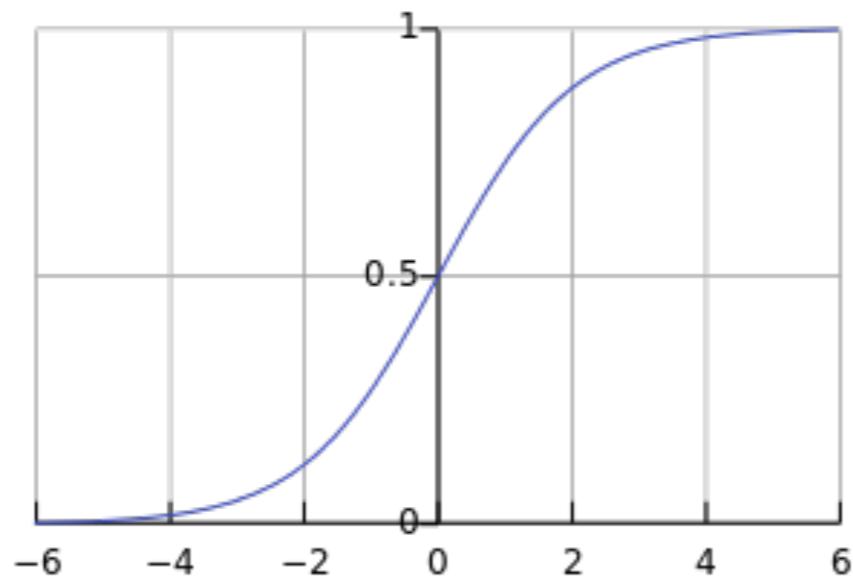


# Single-layer Neural Network

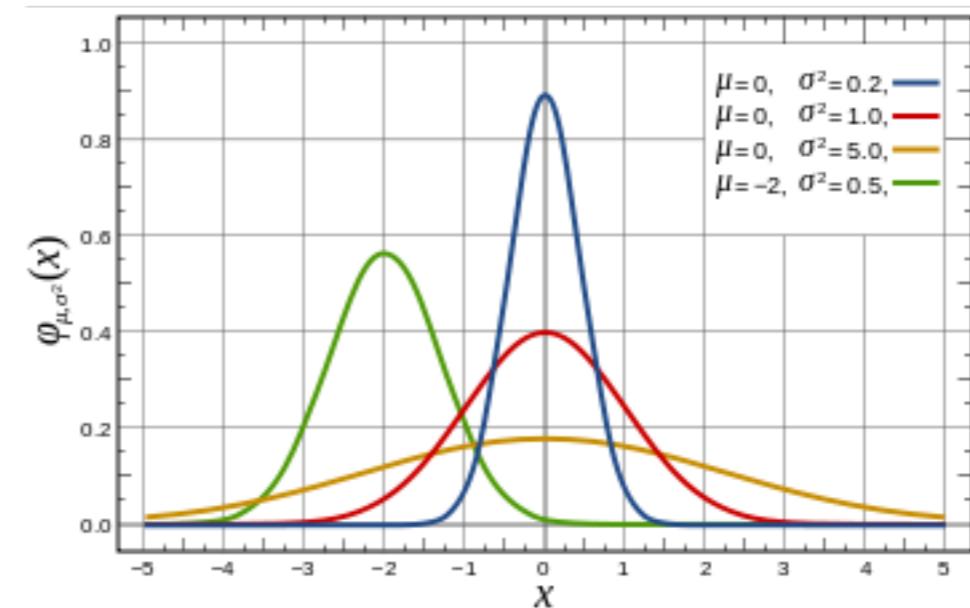


# Non-Linear Function

Sigmoid

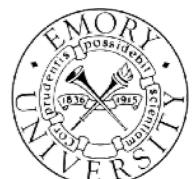


Gaussian



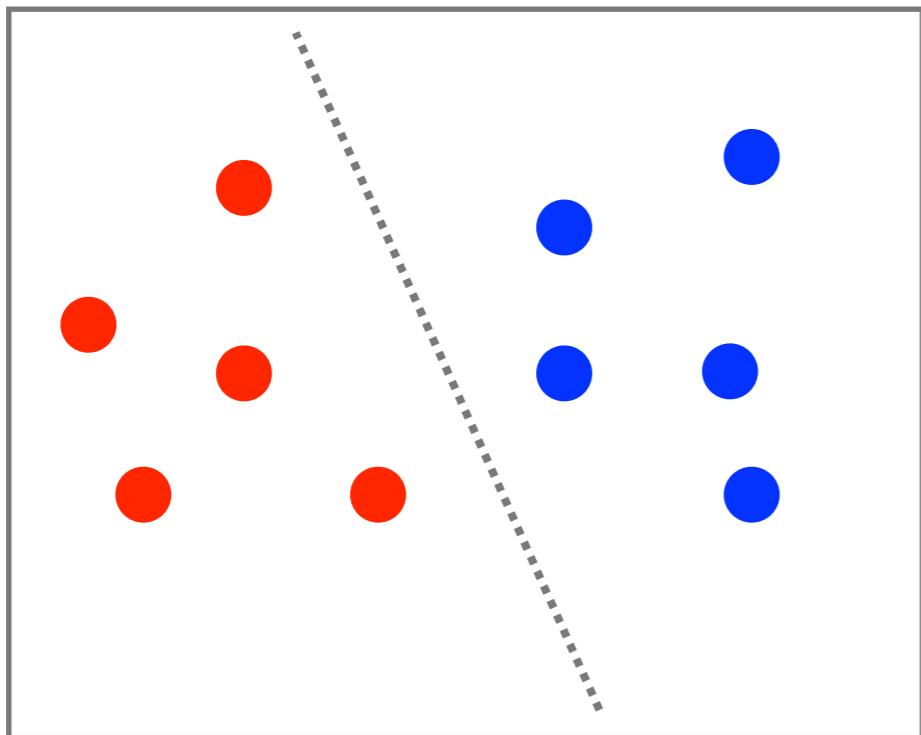
$$\frac{1}{1 + e^{-w \cdot x}}$$

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

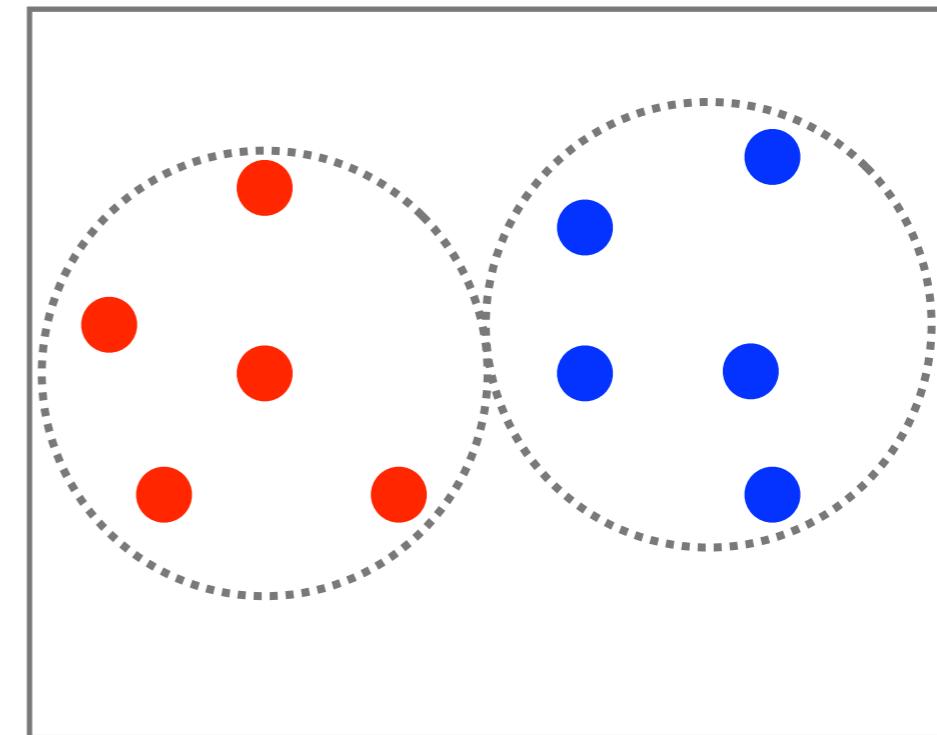


# Sigmoid vs. Gaussian

Sigmoid



Gaussian

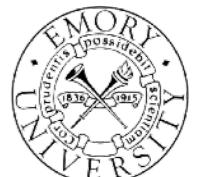


Find [hyperplanes](#)

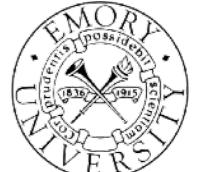
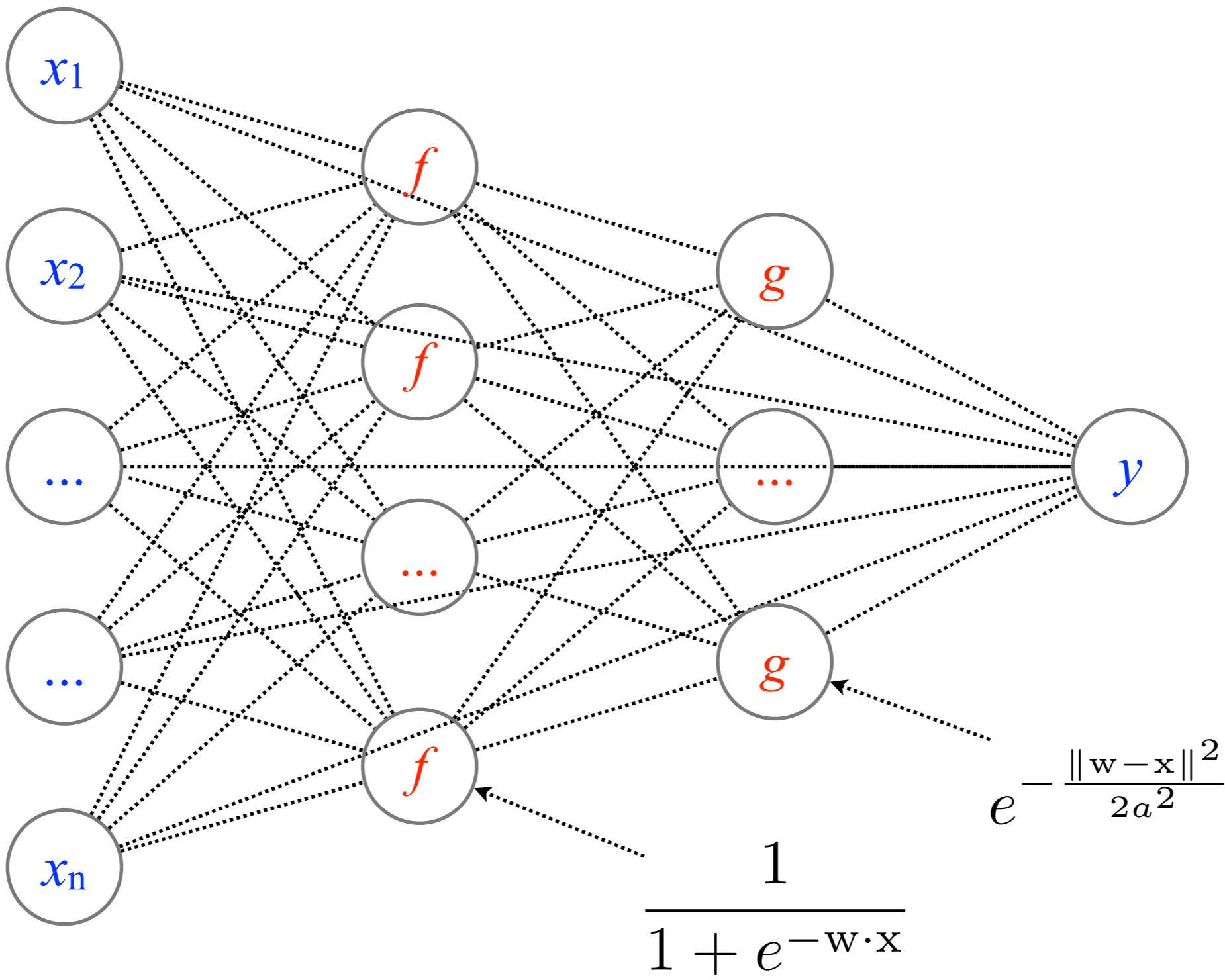
Find [hyperspheres](#)



EMORY  
UNIVERSITY

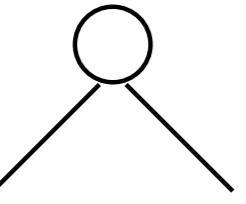
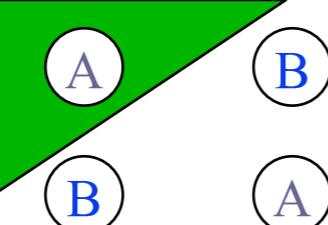
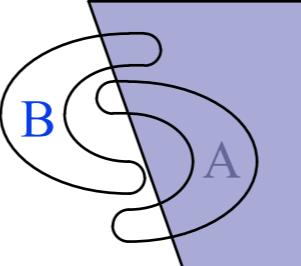
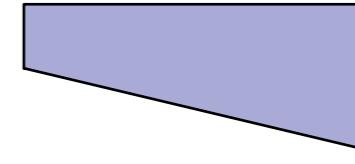
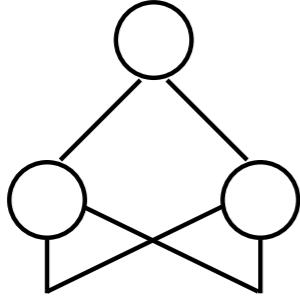
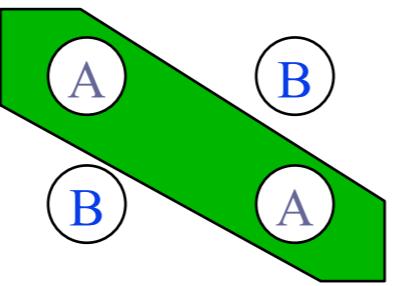
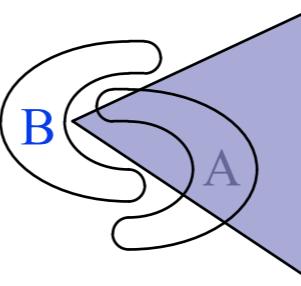
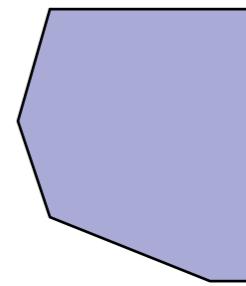
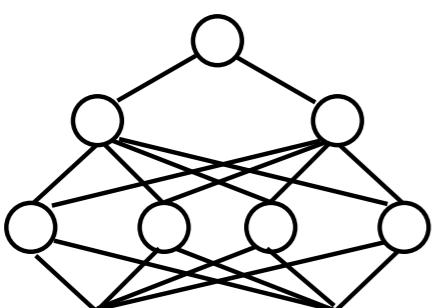
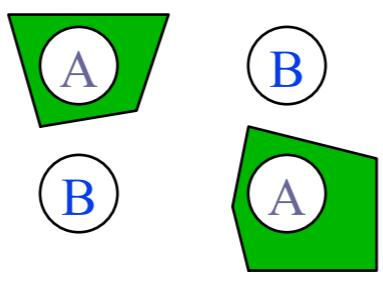
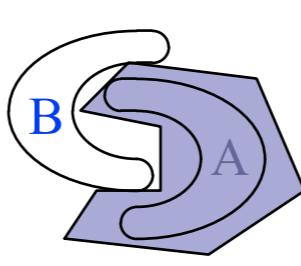
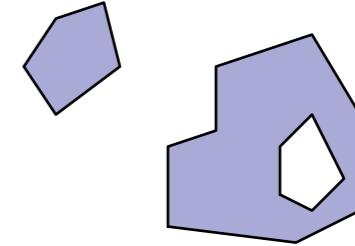


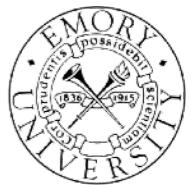
# Multi-layer Neural Network



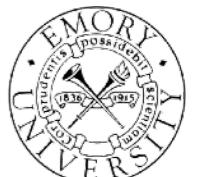
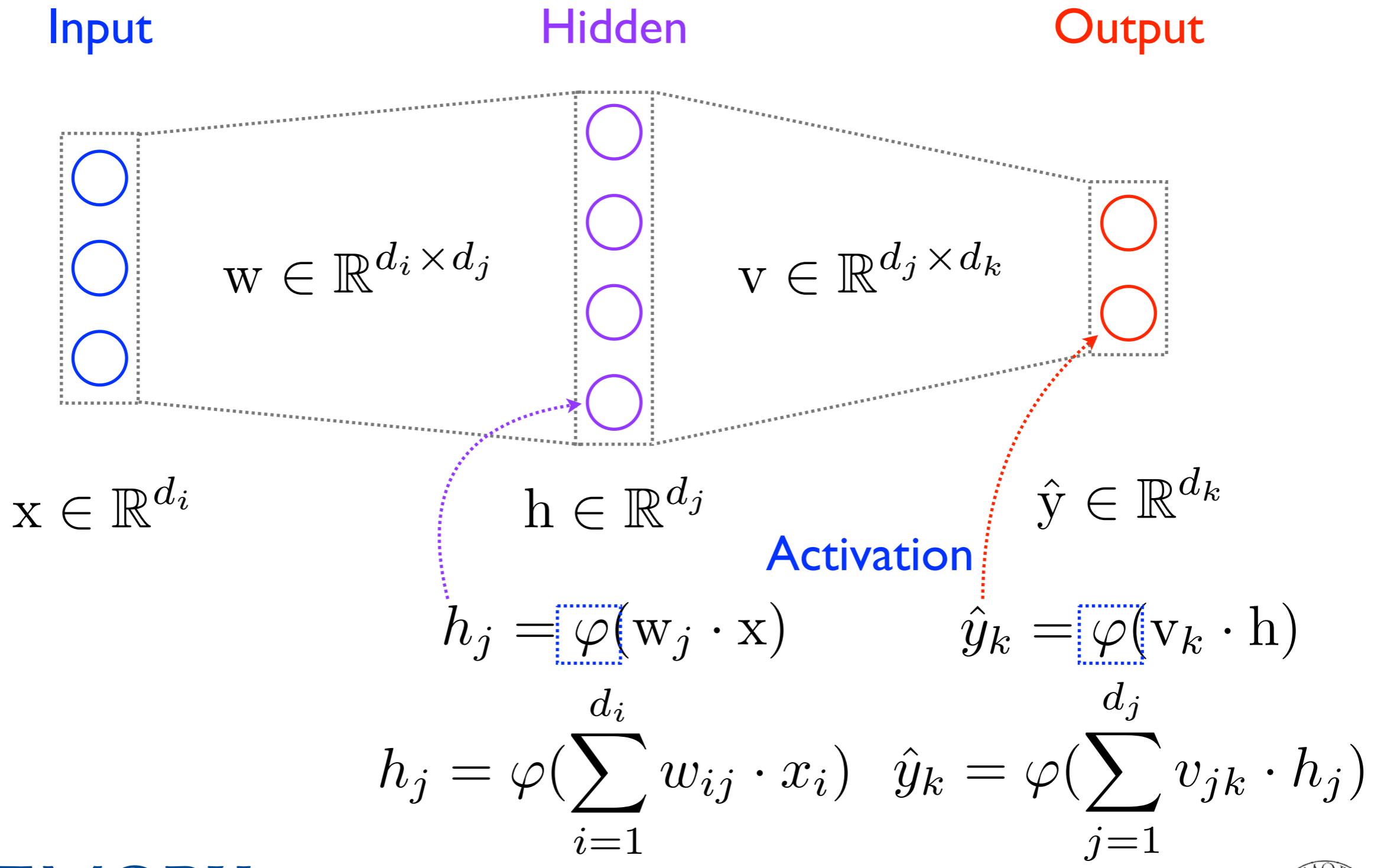
# Multi-layer Neural Network

## Comparison between different layers

# of Layers	Exclusive OR	Meshed Regions	General Regions
			
			
			



# Forward Propagation



# 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

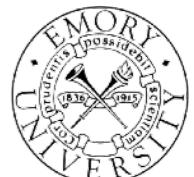
$$\mathcal{E}(y_k, \hat{y}_k) = \frac{1}{2} (\hat{y}_k - y_k)^2$$

Least Square

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

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

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



# Backward Propagation

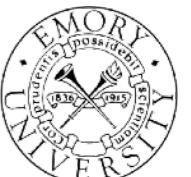
$$\frac{\partial \mathcal{E}}{\partial v_{jk}} = \boxed{\frac{\partial \mathcal{E}}{\partial \hat{y}_k}} \cdot \boxed{\frac{\partial \hat{y}_k}{\partial z_k}} \cdot \boxed{\frac{\partial z_k}{\partial v_{jk}}} \quad \boxed{\varphi(z_k) = \frac{1}{1 + e^{-z_k}}}$$

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

$$\begin{aligned} \frac{\partial}{\partial z_k} \boxed{\hat{y}_k} &= \frac{\partial}{\partial z_k} \boxed{\varphi(z_k)} = \varphi(z_k)(1 - \varphi(z_k)) \\ &= \hat{y}_k(1 - \hat{y}_k) \end{aligned}$$

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

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



# Backward Propagation

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

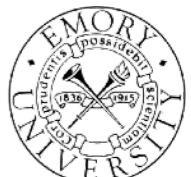


$$\hat{y}_k = \varphi\left(\sum_{j=1}^{d_j} v_{jk} \cdot h_j\right)$$

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



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



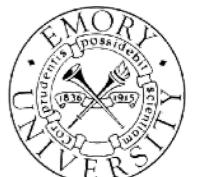
# Backward Propagation

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

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

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

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

Hidden → Output

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

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

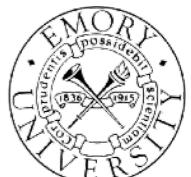
$$\Delta \mathcal{E}_{ij} = \sum_{l=1}^{d_k} (\delta_l \cdot v_{jl}) \cdot h_j(1 - h_j) \cdot x_i \quad \Delta \mathcal{E}_{jk} = \delta_k \cdot h_j$$

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

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



EMORY  
UNIVERSITY



# Distributional Semantics

Quantify and categorize semantic similarities between items.

Distributional properties in Big Data.

Never-Ending Language Learning

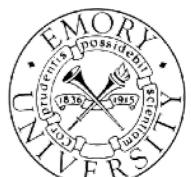
Entity extraction

<http://rtw.ml.cmu.edu/rtw>

ConceptNet

Relation extraction

<http://conceptnet5.media.mit.edu>



# Word Embeddings

Distributional semantics using **neural network**.

“King”

0	1	1	1	0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

“man”

0	0	0	0	0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

“woman”

0	0	0	0	0	1	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

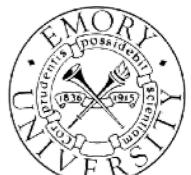
“queen”?

0	1	1	1	0	1	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

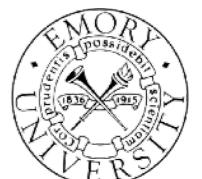
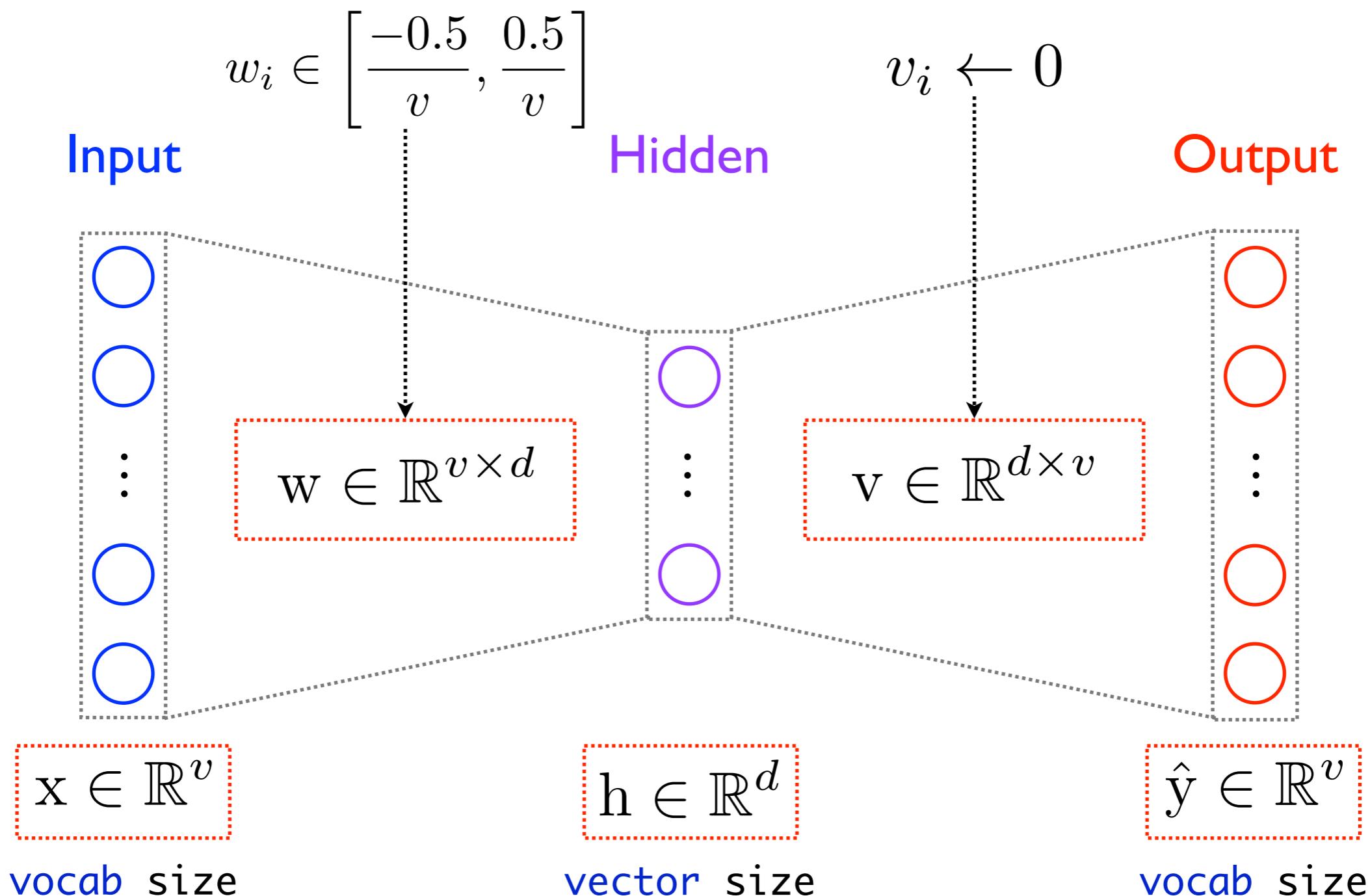
<https://code.google.com/p/word2vec/>



EMORY  
UNIVERSITY

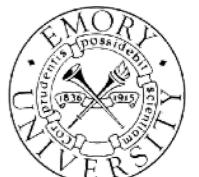
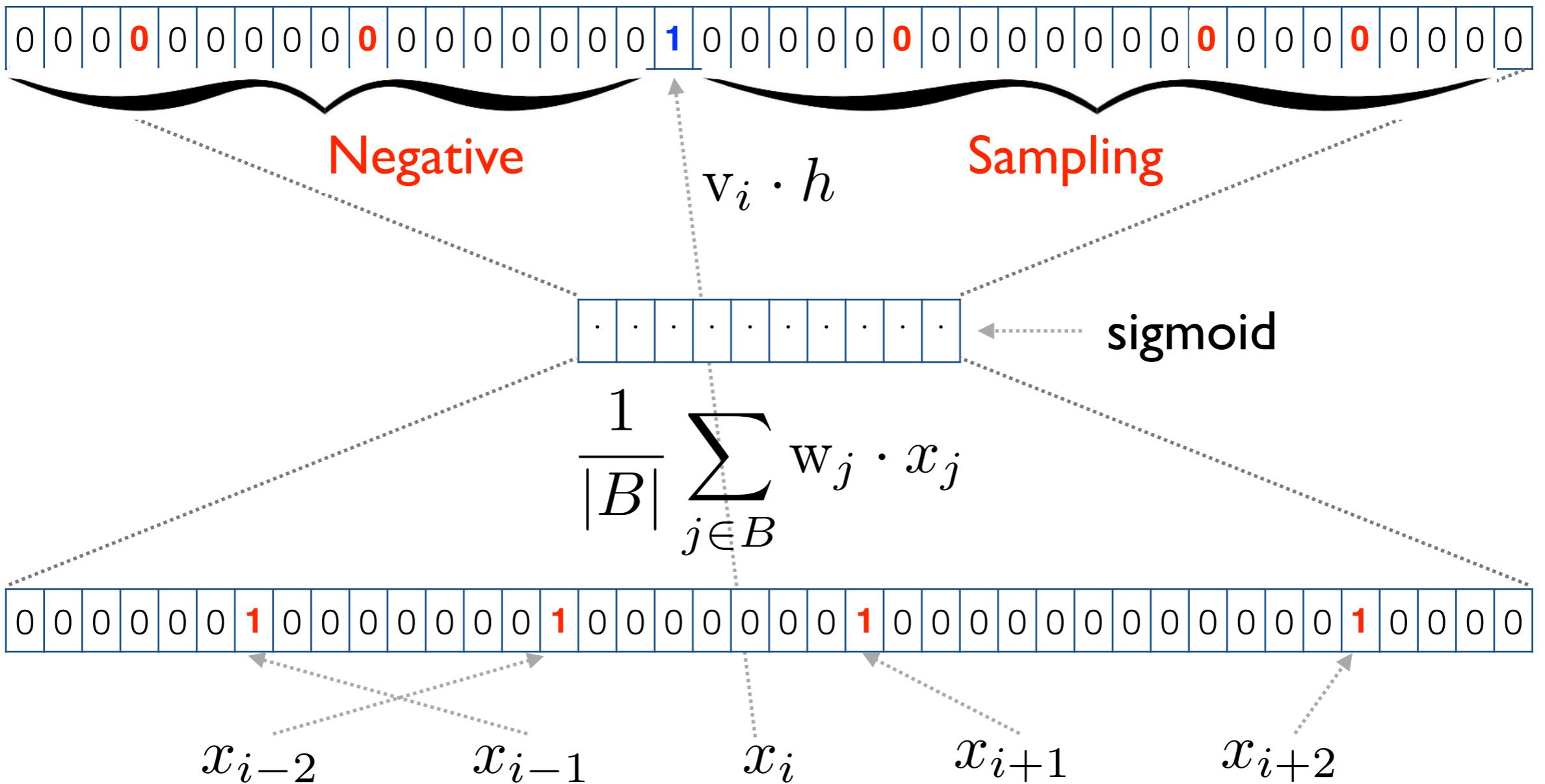


# Word2Vec

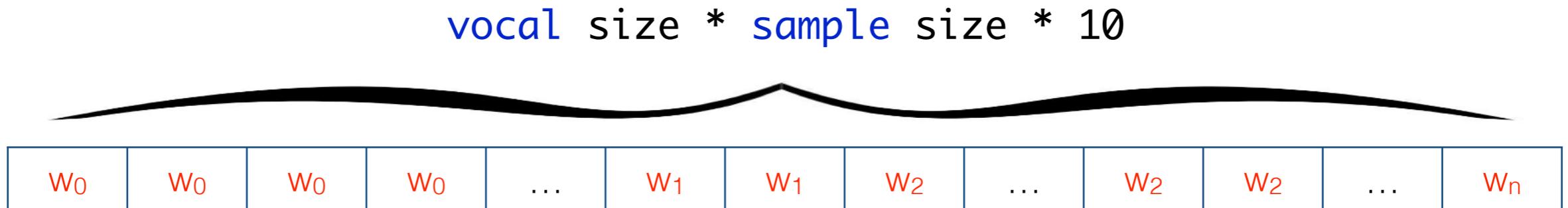


# Bag-of-Words

Predict  $w_i$  given  $\{w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}\}$



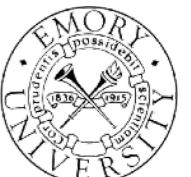
# Negative Sampling



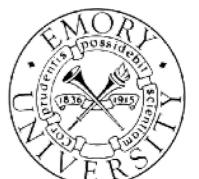
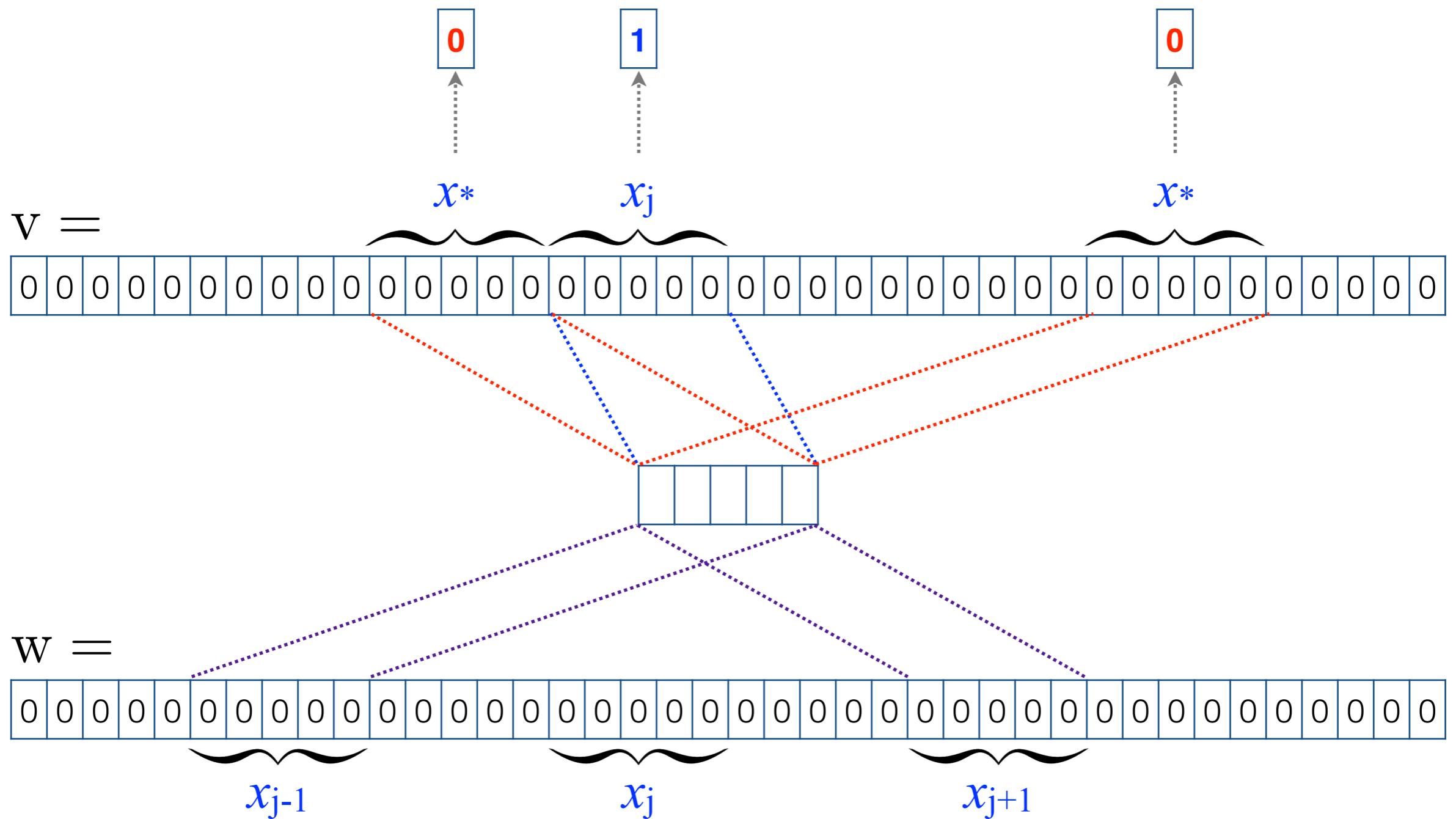
Words are proportionally distributed by their counts.

$$\text{dist}(w_i) = \frac{|w_i|^{\frac{3}{4}}}{\sum_{\forall j} |w_j|^{\frac{3}{4}}} \quad \text{distribution ratio}$$

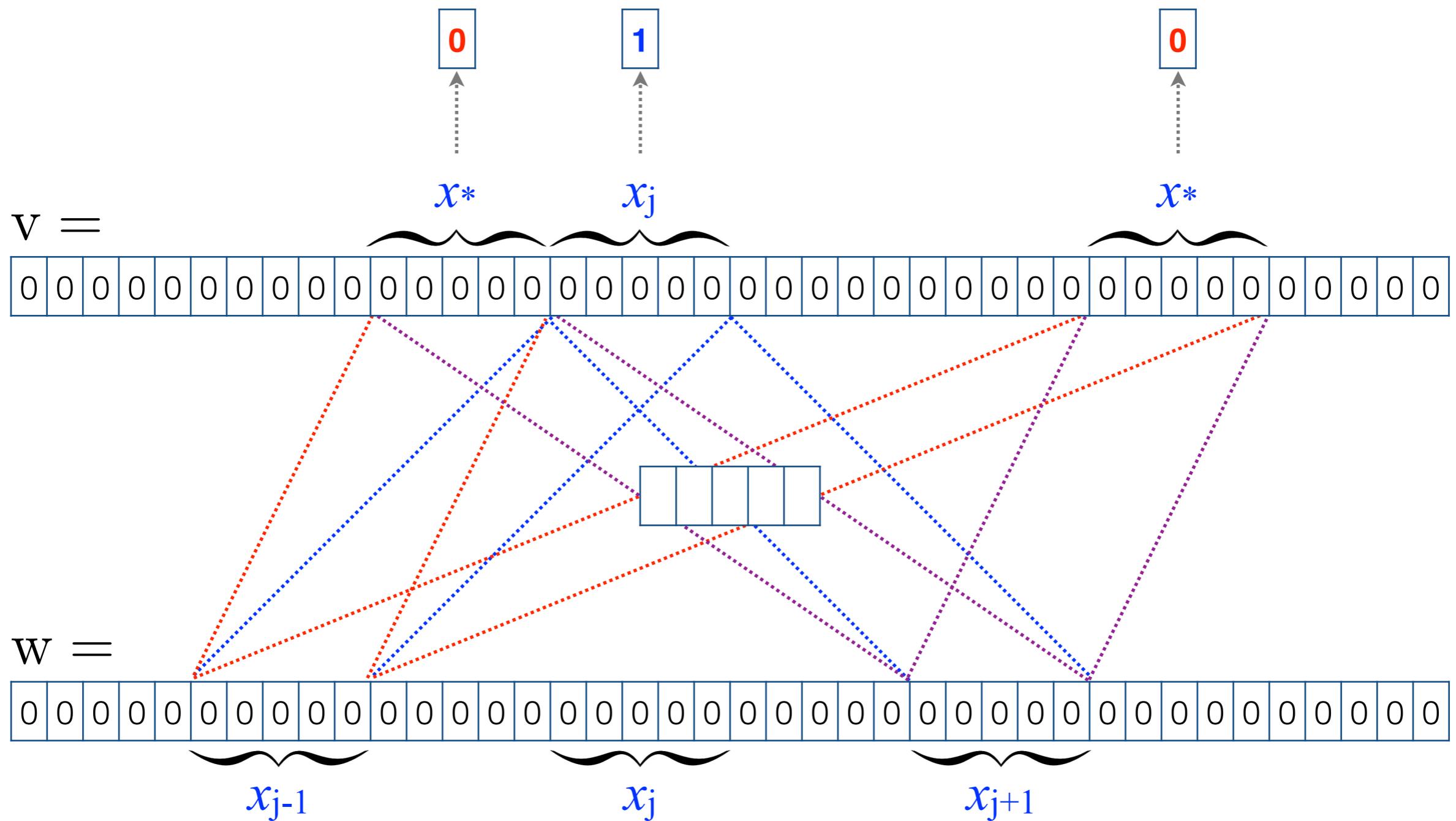
Randomly select negative words from the distribution.



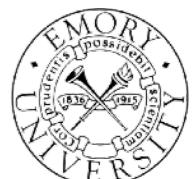
# Bag-of-Words



# Skip-Gram



**EMORY  
UNIVERSITY**



# Neural Language Modeling

Dr. ? is a professor at Emory University

$w_i$

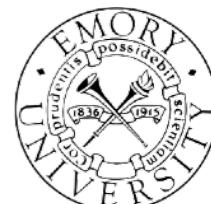
$$P(w_i | w_1, \dots, w_{i-1}, w_{i+1}, w_n)$$

$$= \frac{\#(w_1, \dots, w_{i-1}, w_i, w_{i+1}, w_n)}{\sum_{\forall k} \#(w_1, \dots, w_{i-1}, w_k, w_{i+1}, w_n)}$$

Too sparse!

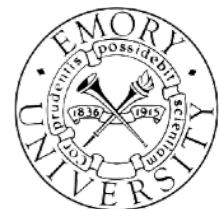
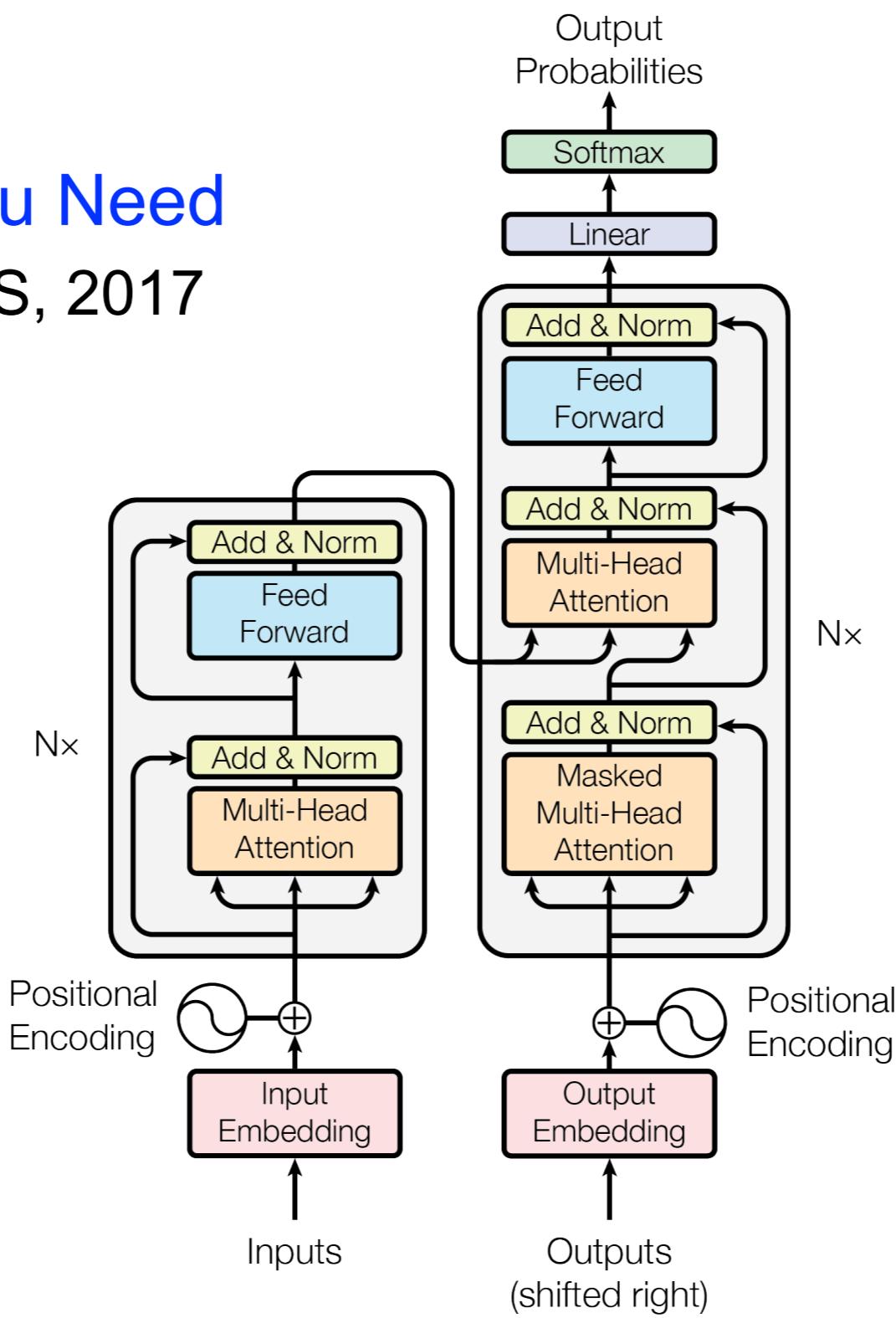
Dr. Choi is a professor at Emory University  
Jinho Choi is an associate professor  
Professor Choi works at Emory in Atlanta

A	AN	ASSOCIATE	AT	ATLANTA	CHOI	DR.	EMORY	IN	IS	JINHO	PROFESSOR	UNIVERSITY	WORKS
1	1	1	2	1		1	2	1	1	1	3	1	1



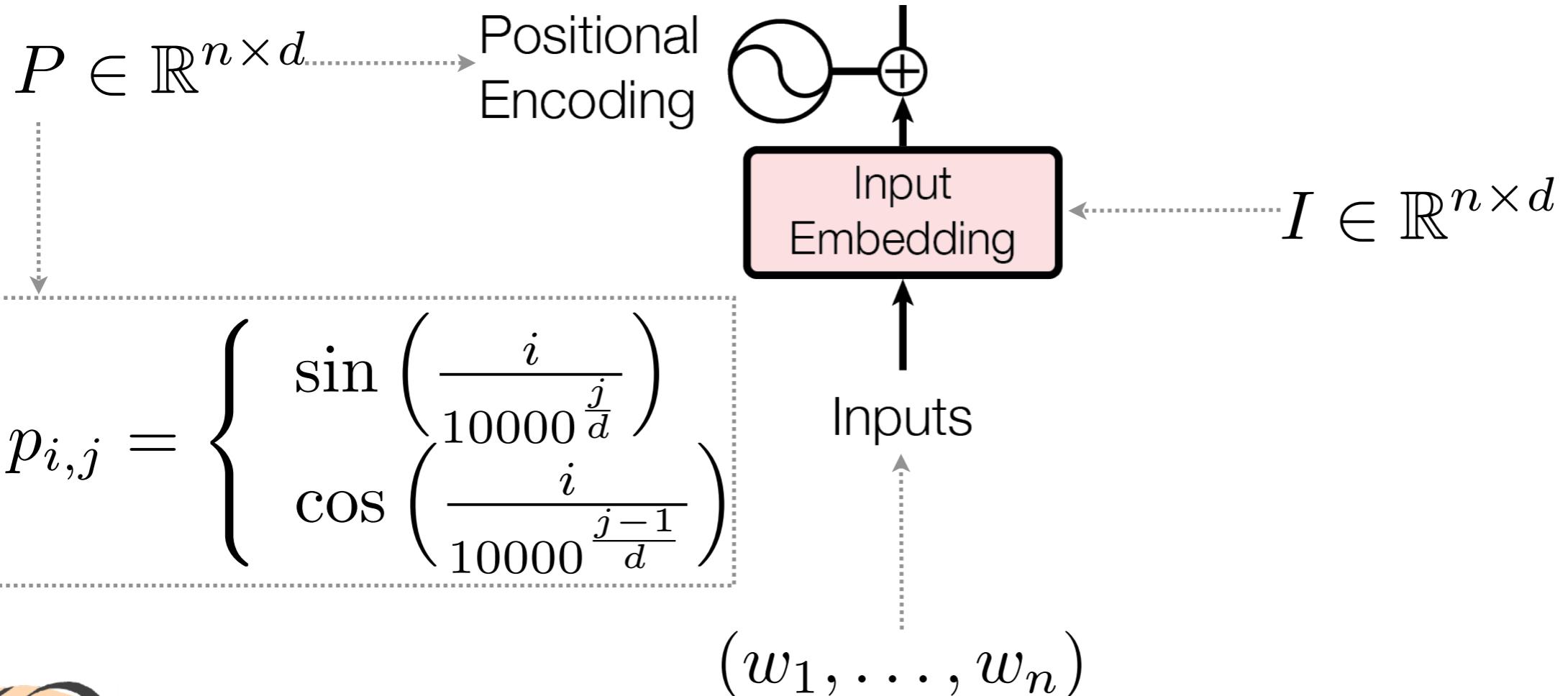
# Transformer

Attention Is All You Need  
Vaswani et al., NIPS, 2017

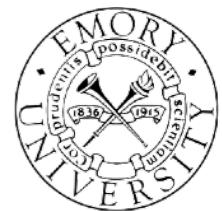
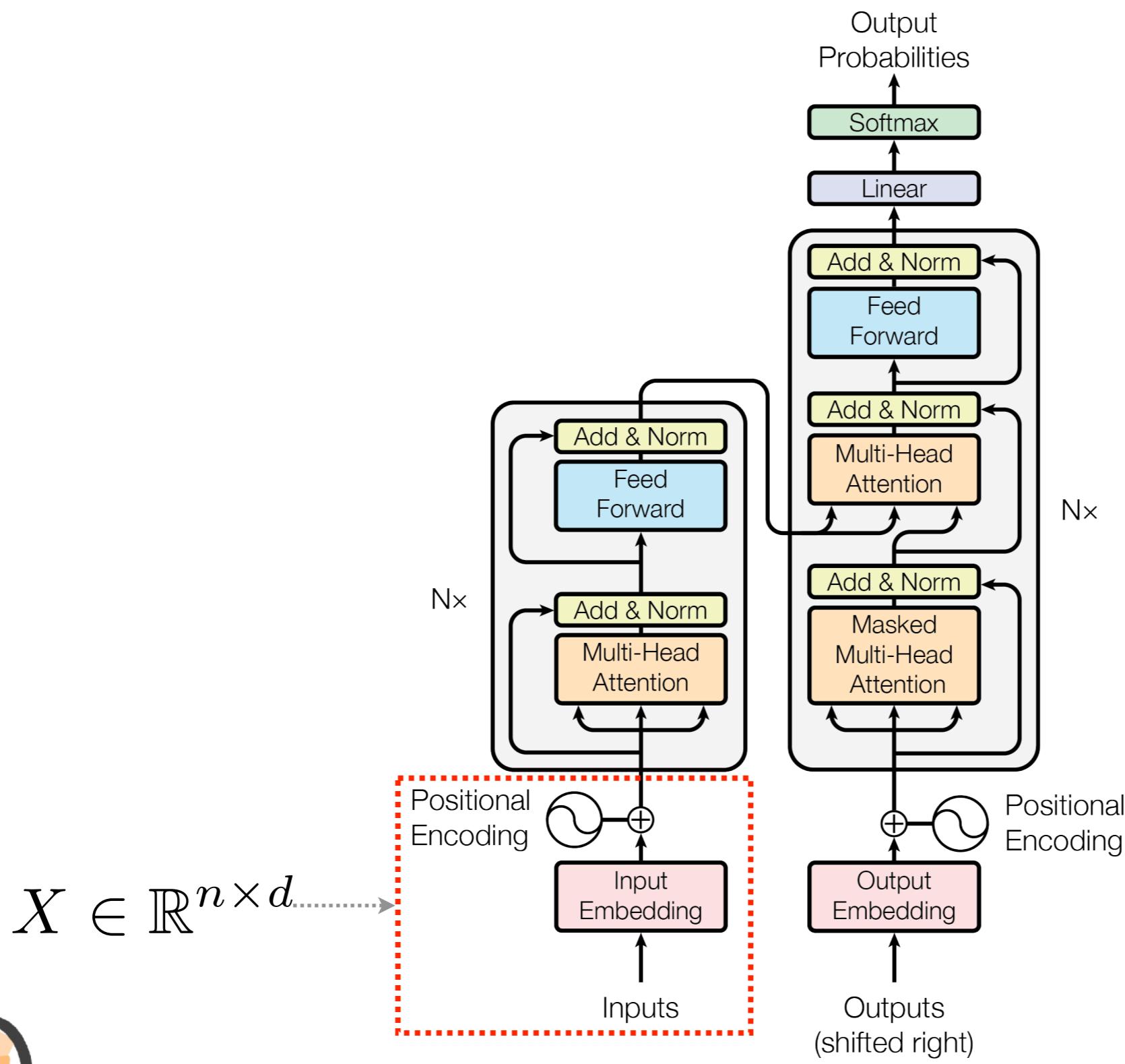


# Input Layer

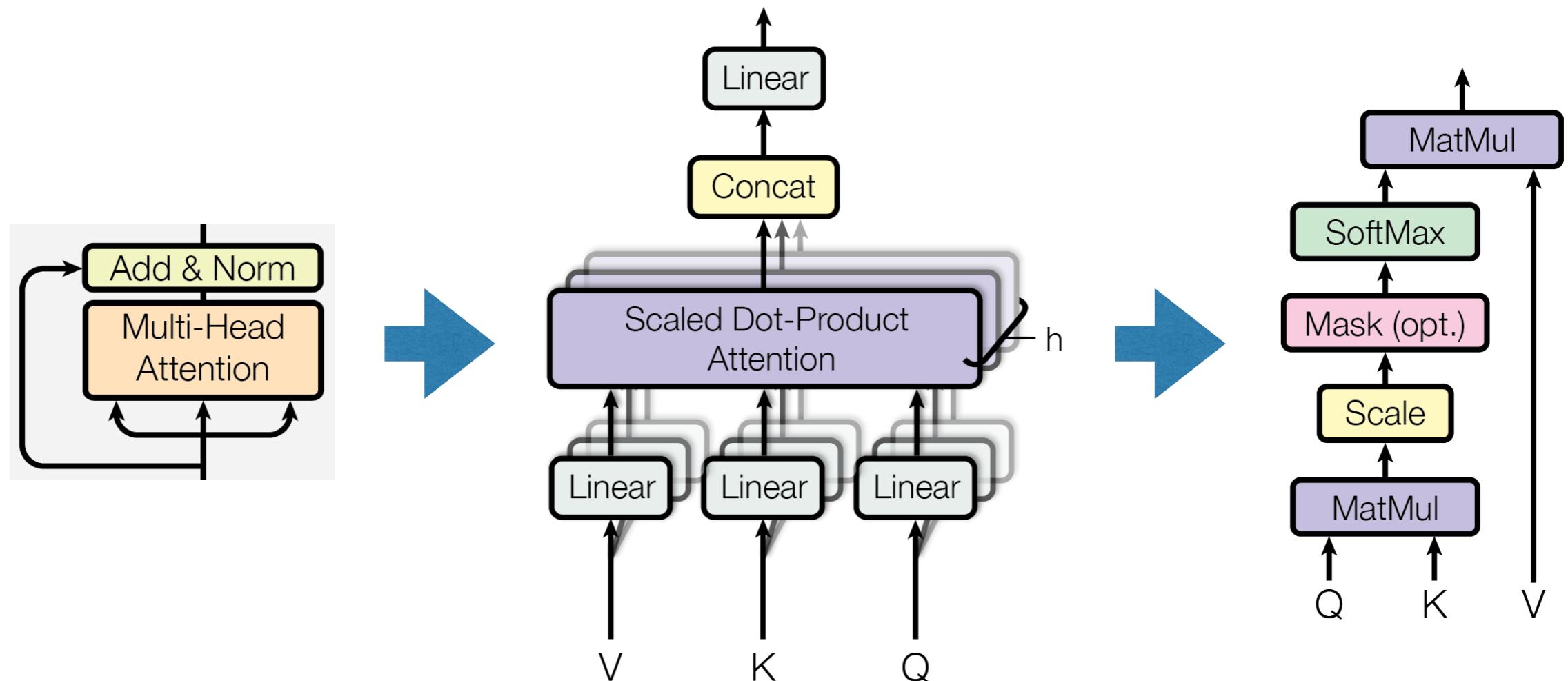
$$X = (I + P) \in \mathbb{R}^{n \times d}$$



# Transformer



# Encoder



# Scaled Dot Product Attention

$$\text{head}_i = \mathcal{S}(VW_i^V) \in \mathbb{R}^{n \times d_v}$$

$$\mathcal{S} = \text{Softmax}_{rw}(\mathcal{A}^s) \in \mathbb{R}^{n \times n}$$

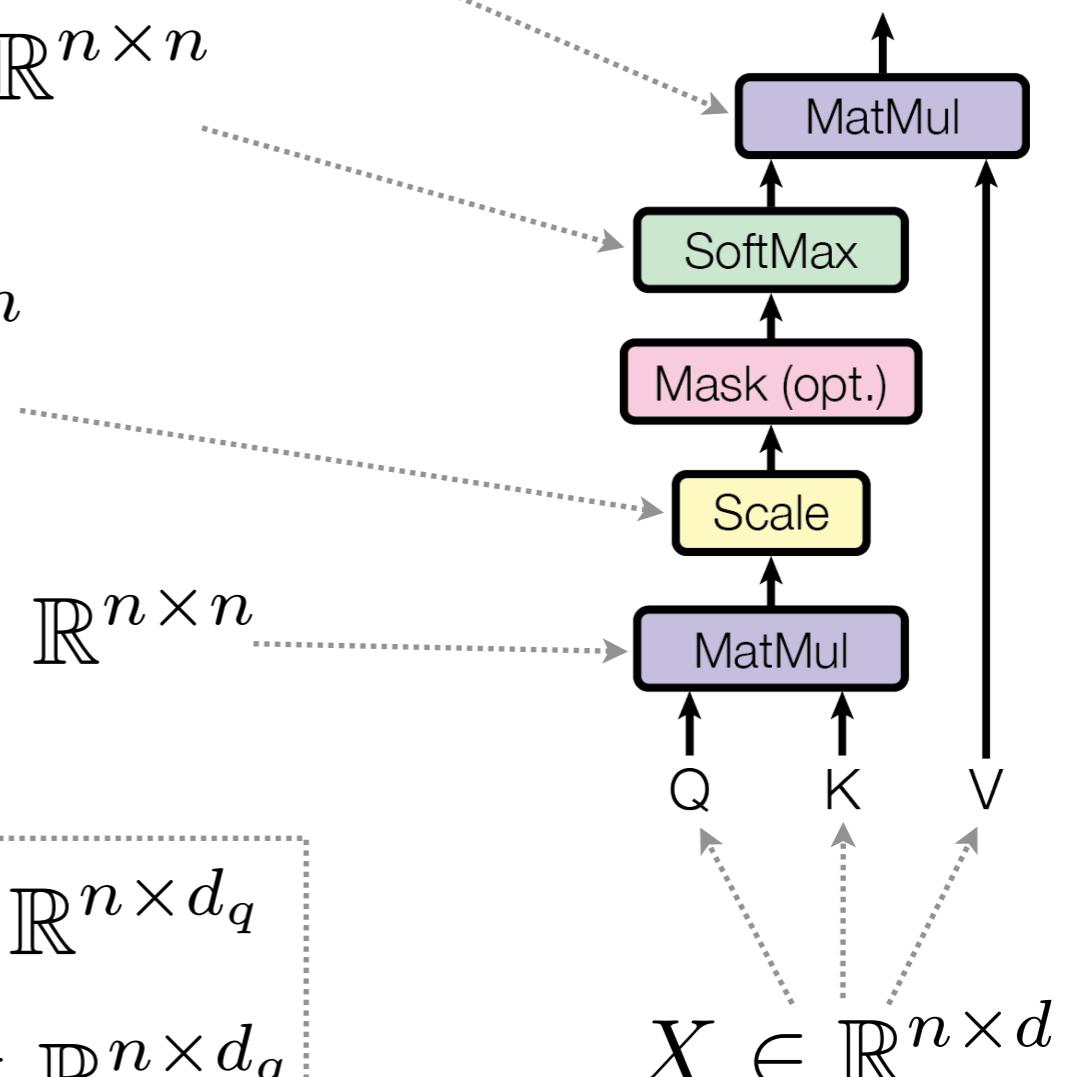
$$\mathcal{A}^s = \frac{\mathcal{A}}{\sqrt{d_q}} \in \mathbb{R}^{n \times n}$$

$$\mathcal{A} = (QW_i^Q)(KW_i^K)^T \in \mathbb{R}^{n \times n}$$

$$W_i^Q \in \mathbb{R}^{d \times d_q} \Rightarrow QW_i^Q \in \mathbb{R}^{n \times d_q}$$

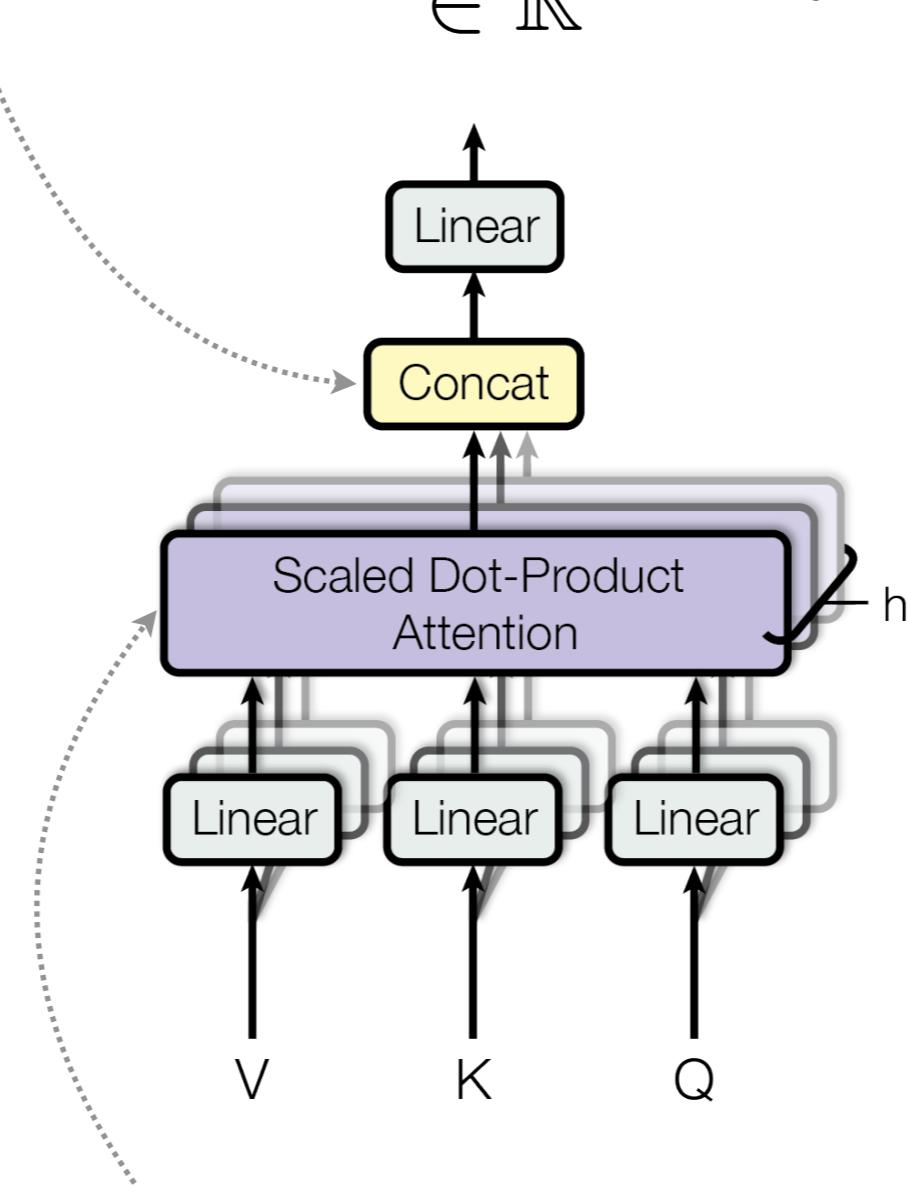
$$W_i^K \in \mathbb{R}^{d \times d_q} \Rightarrow KW_i^K \in \mathbb{R}^{n \times d_q}$$

$$W_i^V \in \mathbb{R}^{d \times d_v} \Rightarrow VW_i^V \in \mathbb{R}^{n \times d_v}$$



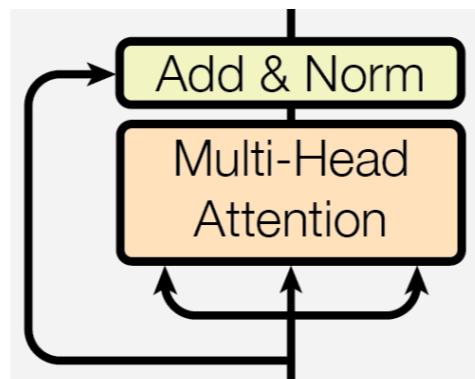
# Multi-Head Attention

$$\begin{aligned}\text{MultiHead}(Q, K, V) &= (\text{head}_1 \oplus \dots \oplus \text{head}_h)W^O \\ &\in \mathbb{R}^{n \times d} \qquad \qquad \qquad \in \mathbb{R}^{n \times h \cdot d_v} \qquad \qquad \qquad \in \mathbb{R}^{h \cdot d_v \times d}\end{aligned}$$



$$\text{head}_i = \mathcal{S}(VW_i^V) \in \mathbb{R}^{n \times d_v}$$

# Encoder



$\text{MultiHead}(Q, K, V) \in \mathbb{R}^{n \times d}$

