The University of Texas at Austin
Cockrell School of Engineering

GRAPH NEURAL NETWORK AND ITS APPLICATION

Yinshuang Xiao | yinshuangxiao@utexas.edu
System Integration & Design Informatics Laboratory
Walker Department of Mechanical Engineering

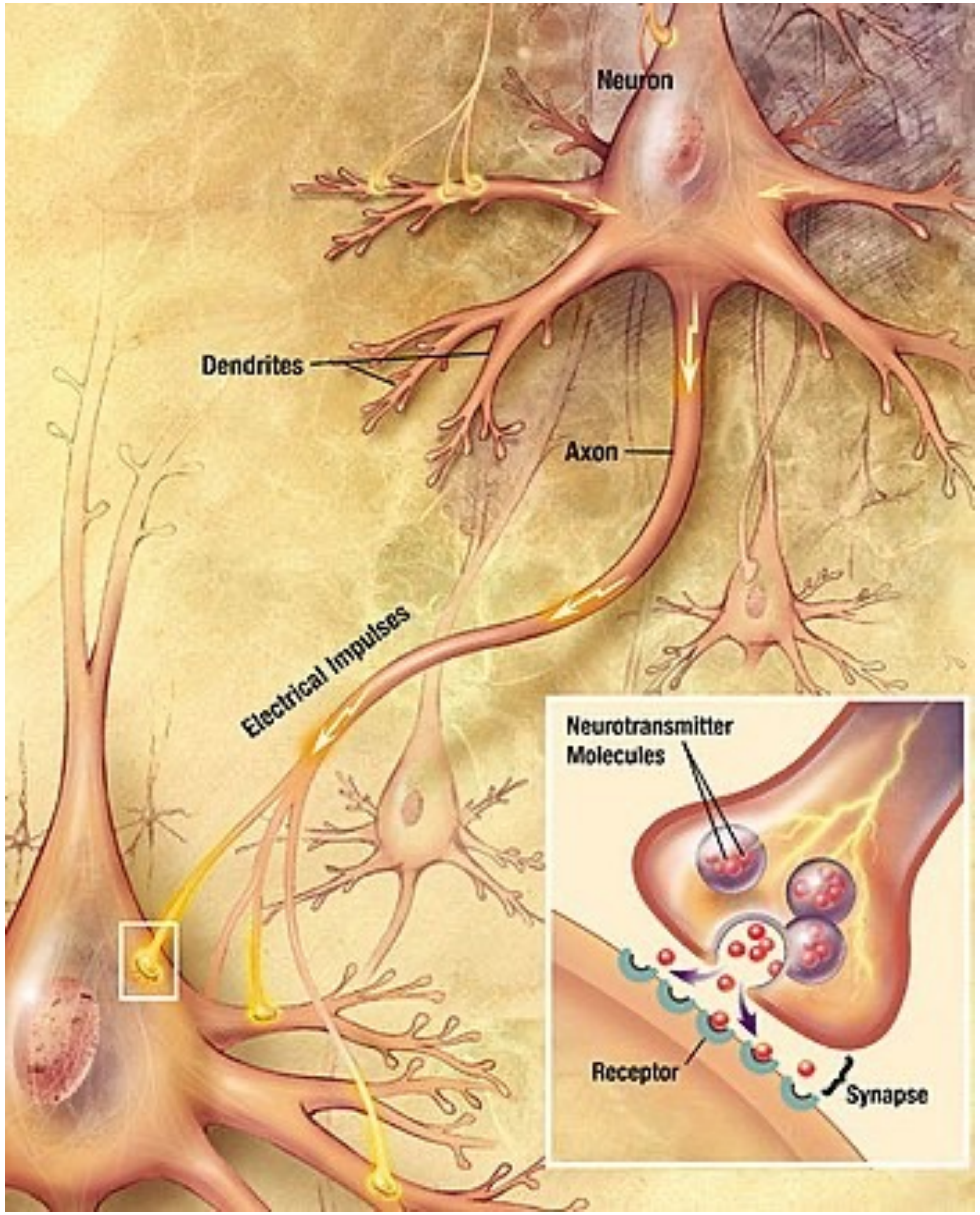
Outline

- **Section 1. Some Basic Concepts of Neural Network**
  - Neuron
  - Activation function
  - How do neural networks work
  - Artificial Neural Network (ANN) example
- **Section 2. Graph Neural Network (GNN)**
  - Graph data and graph tasks
  - How do GNNs work
  - GraphSAGE
- **Section 3. Application: Modeling Shared Mobility System Using GNN**

Outline

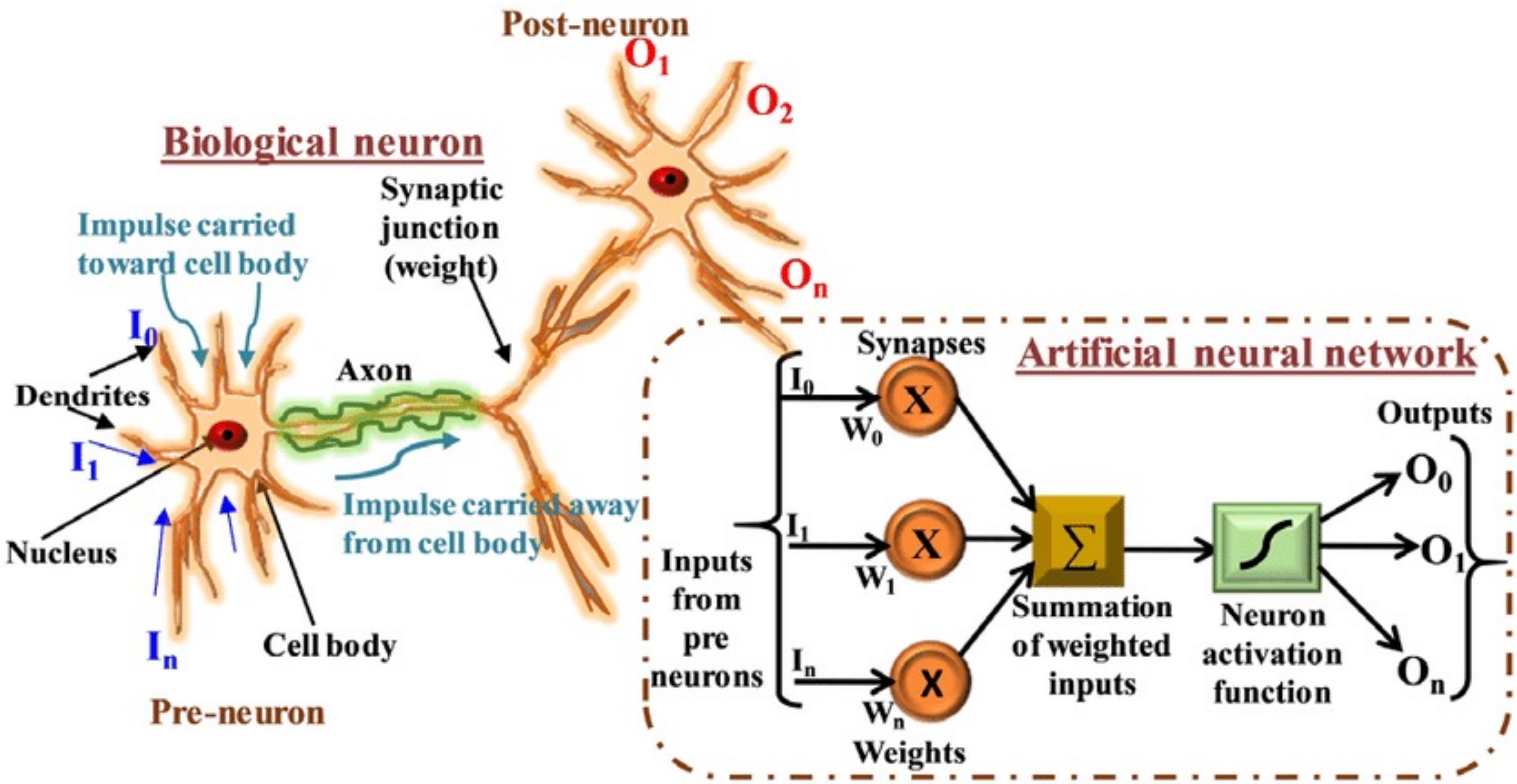
- **Section 1. Some Basic Concepts of Neural Network**
  - Neuron
  - Activation function
  - How do neural networks work
  - Artificial Neural Network (ANN) example
- **Section 2. Graph Neural Network (GNN)**
  - Graph data and graph tasks
  - How do GNNs work
  - GraphSAGE
- **Section 3. Application: Modeling Shared Mobility System Using GNN**

- Biological neuron

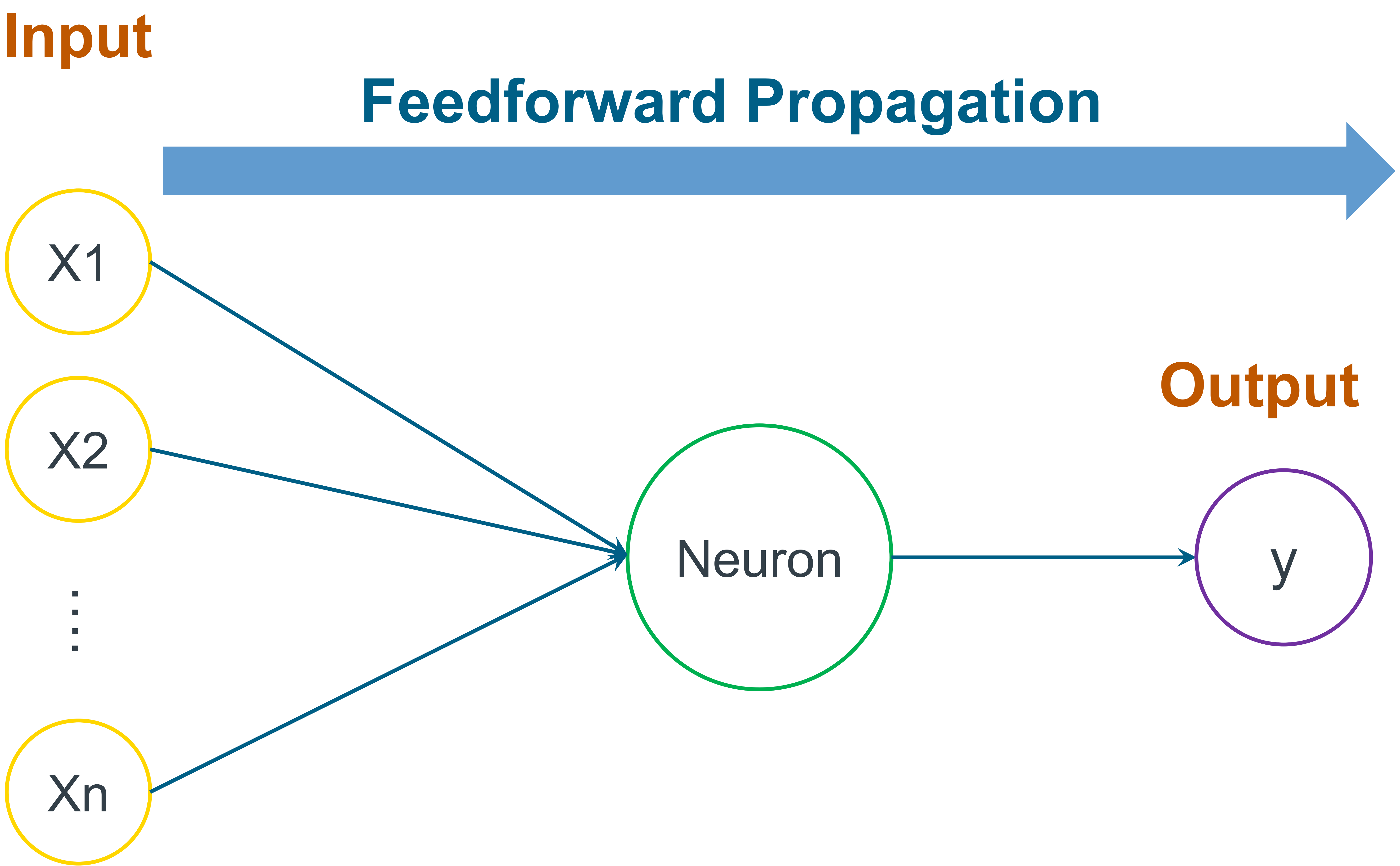


(Wikipedia)

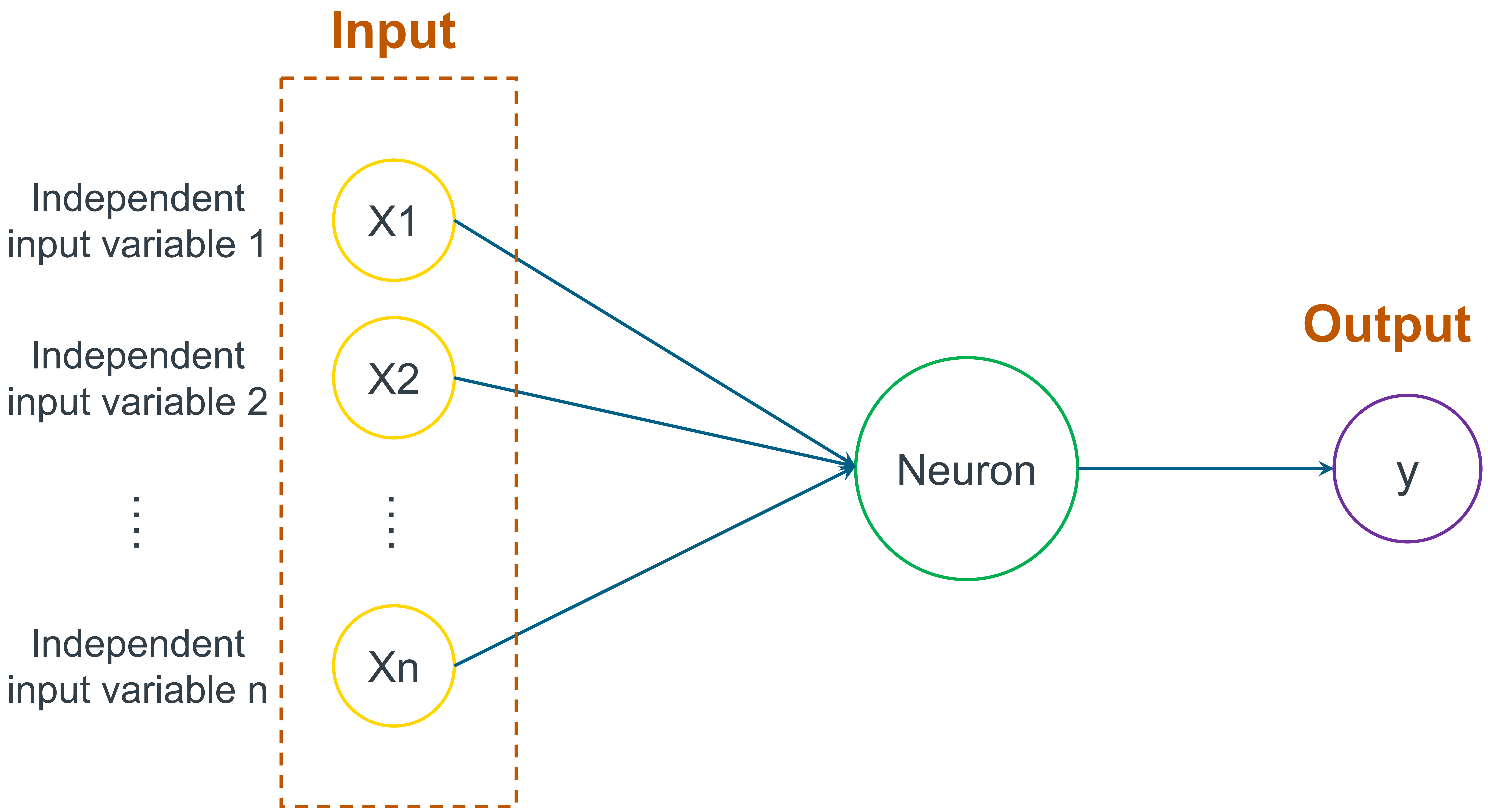
Some Basic Concepts of Neural Network



Reference: Shreya, S., Verma, G., Piramanayagam, S.N. and Kaushik, B.K., 2020. Energy-efficient all-spin BNN using voltage-controlled spin-orbit torque device for digit recognition. *IEEE Transactions on Electron Devices*, *68*(1), pp.385-392.


Input
Feedforward Propagation
X1
X2
⋮
Xn
Neuron
Output
y

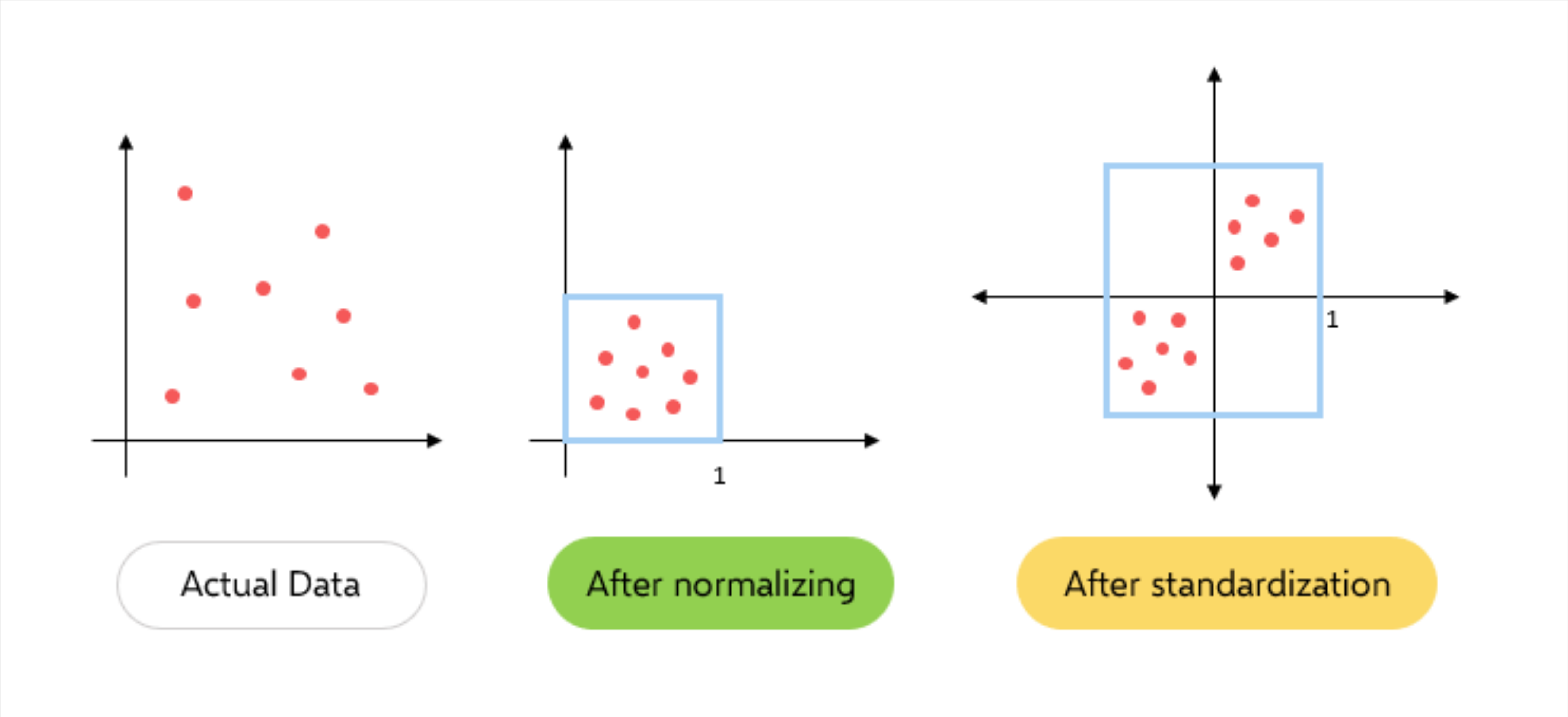
Some Basic Concepts of Neural Network



Standardization / Normalization[1]

[1] LeCun, Y.A., Bottou, L., Orr, G.B. and Müller, K.R., 2012. Efficient backprop. In *Neural networks: Tricks of the trade* (pp. 9-48). Springer, Berlin, Heidelberg.

Standardization / Normalization

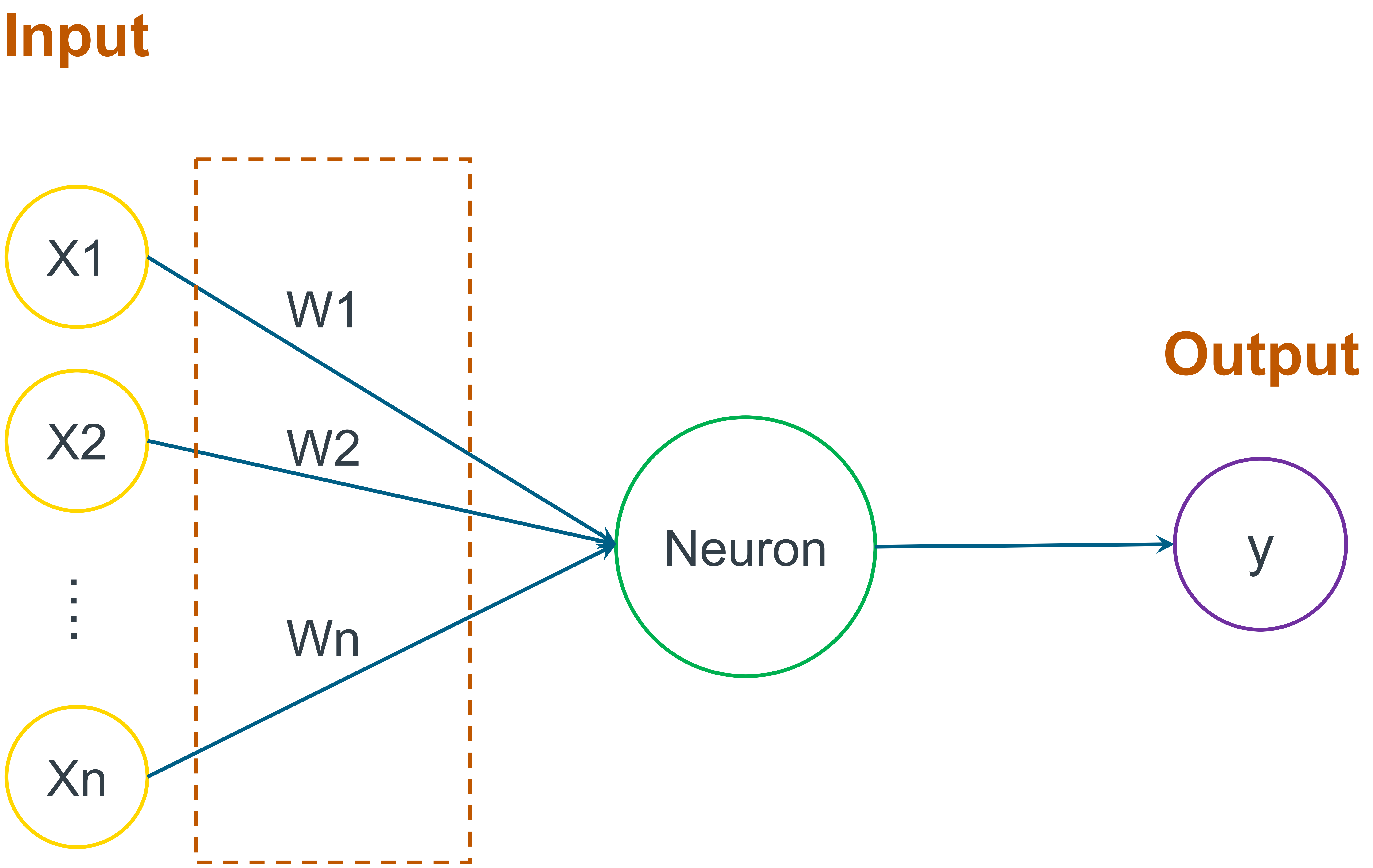


(https://www.someka.net/blog/how-to-normalize-data-in-excel/)

Standardization (Z-score Standardization): $\mathrm{X_i}' = \frac{x_i - \mu}{\sigma}$

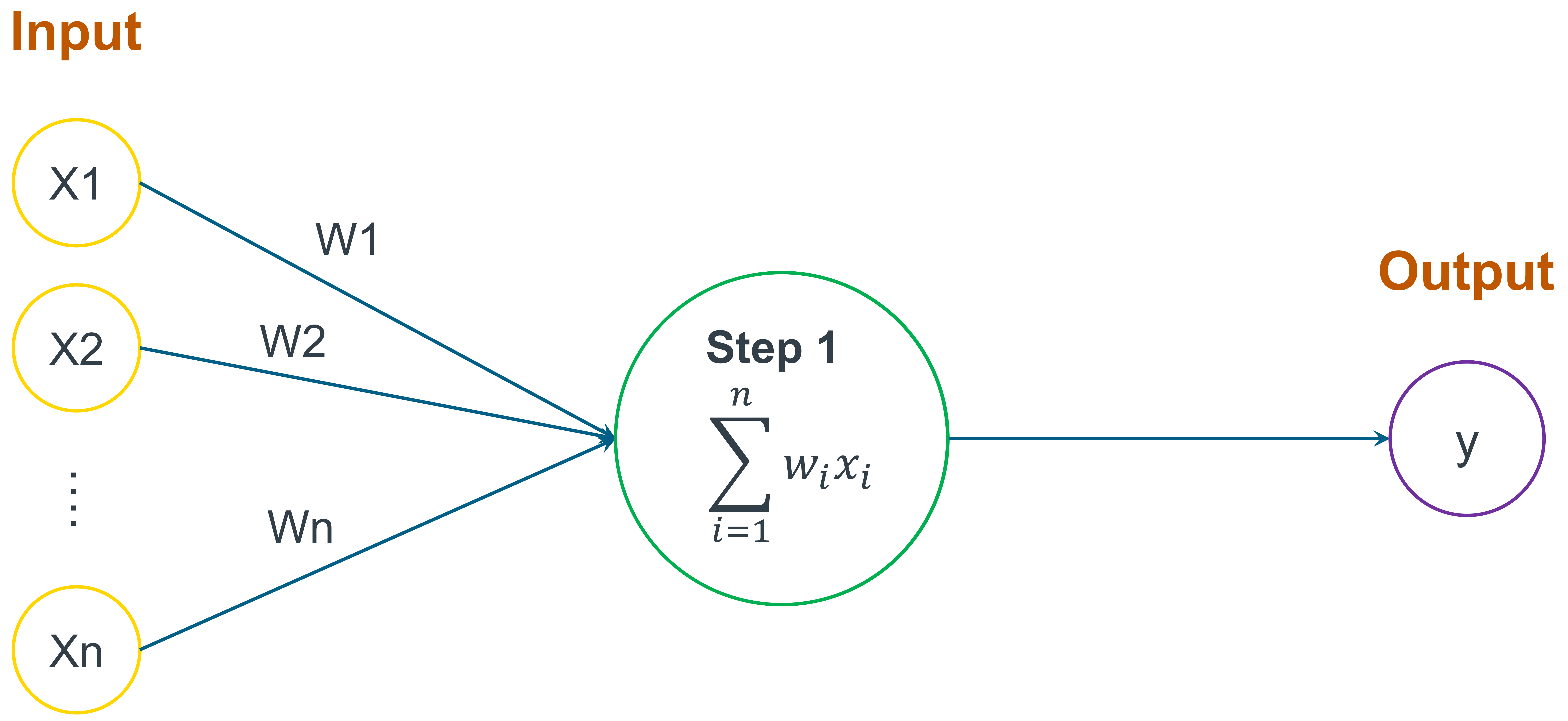
Normalization: $\mathrm{X_i}' = \frac{x_i - \min(X)}{\max(X) - \min(X)}$

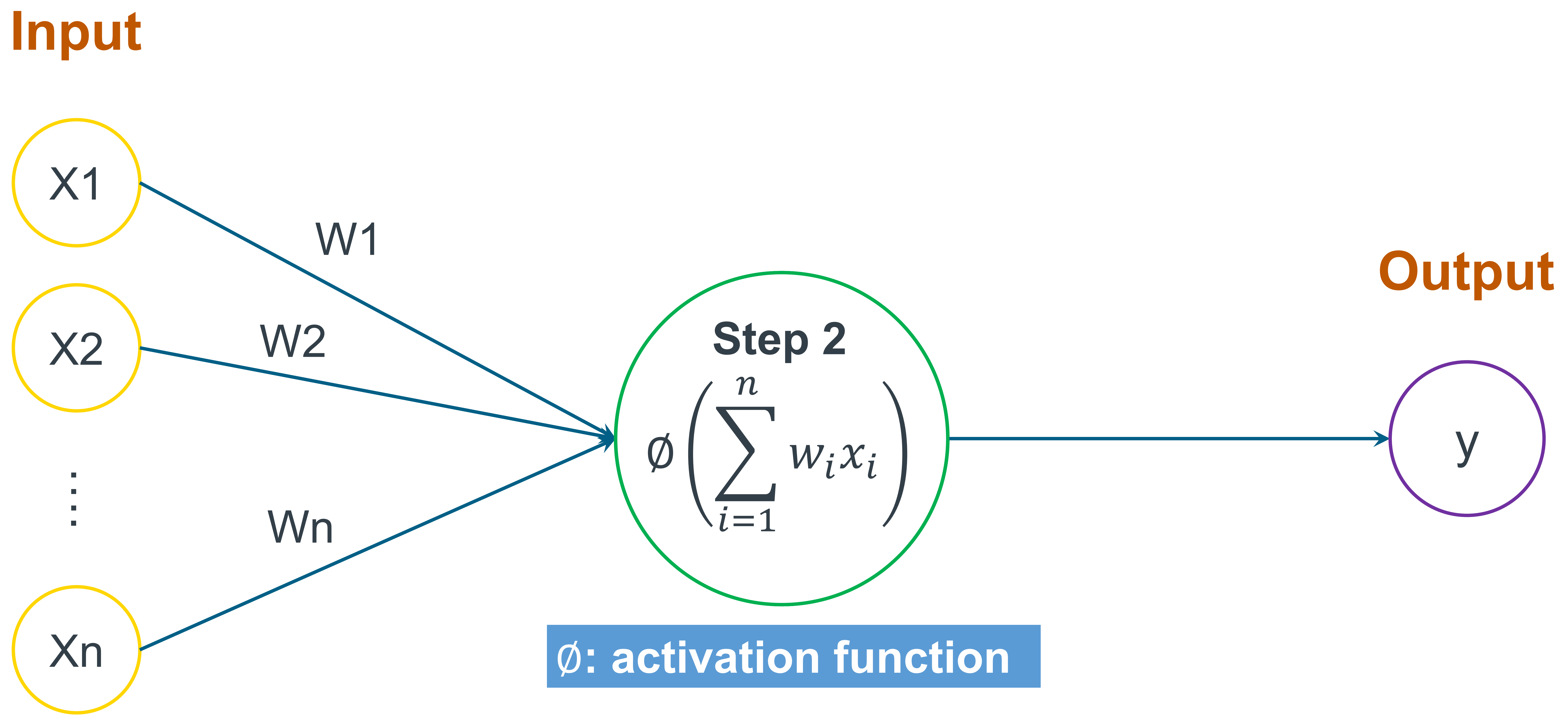
- **Normalization** is used when the data doesn't have Gaussian distribution whereas **Standardization** is used on data having Gaussian distribution.
- **Normalization** scales in a range of [0,1] or [-1,1]. **Standardization** is not bounded by range.
- **Normalization** is highly affected by outliers. **Standardization** is slightly affected by outliers.
- **Normalization** is considered when the algorithms do not make assumptions about the data distribution. **Standardization** is used when algorithms make assumptions about the data distribution.

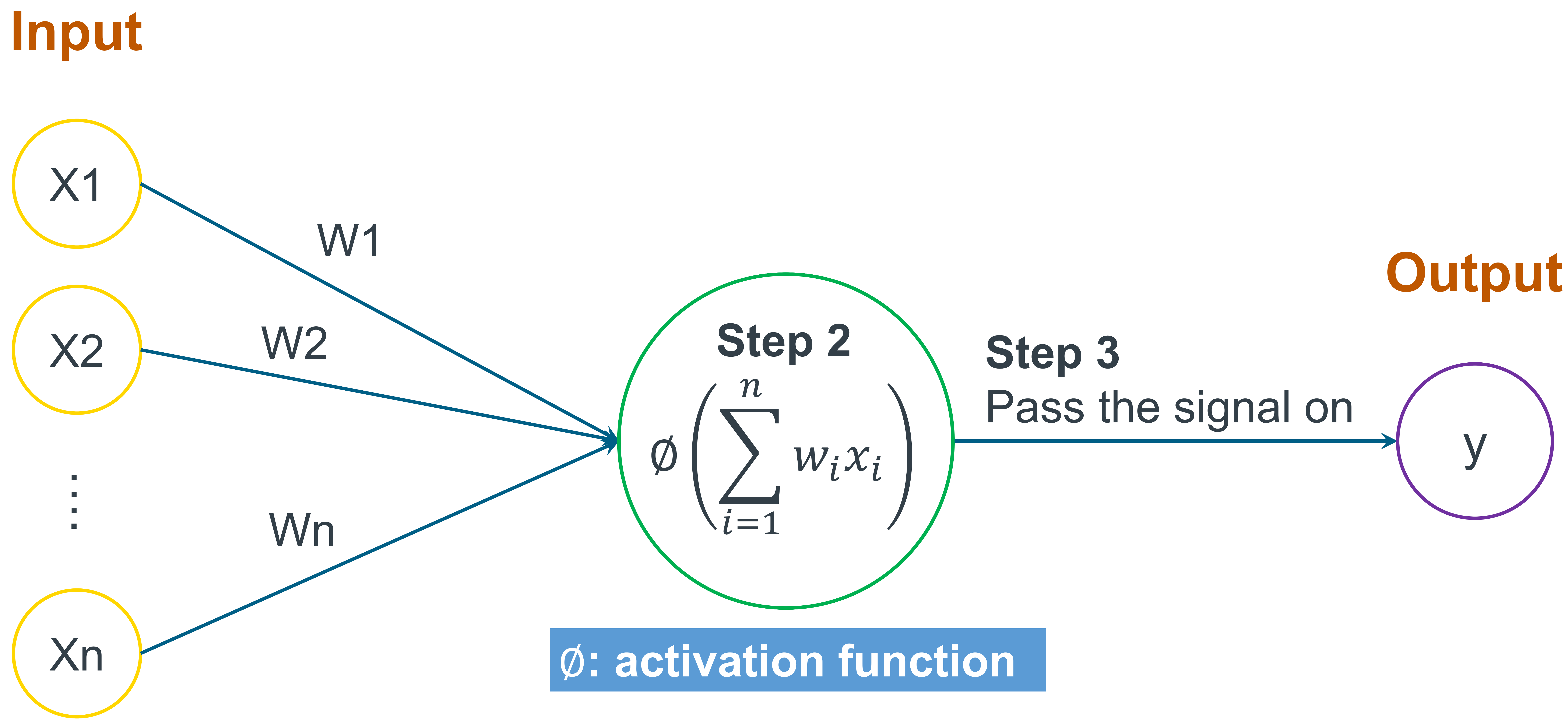


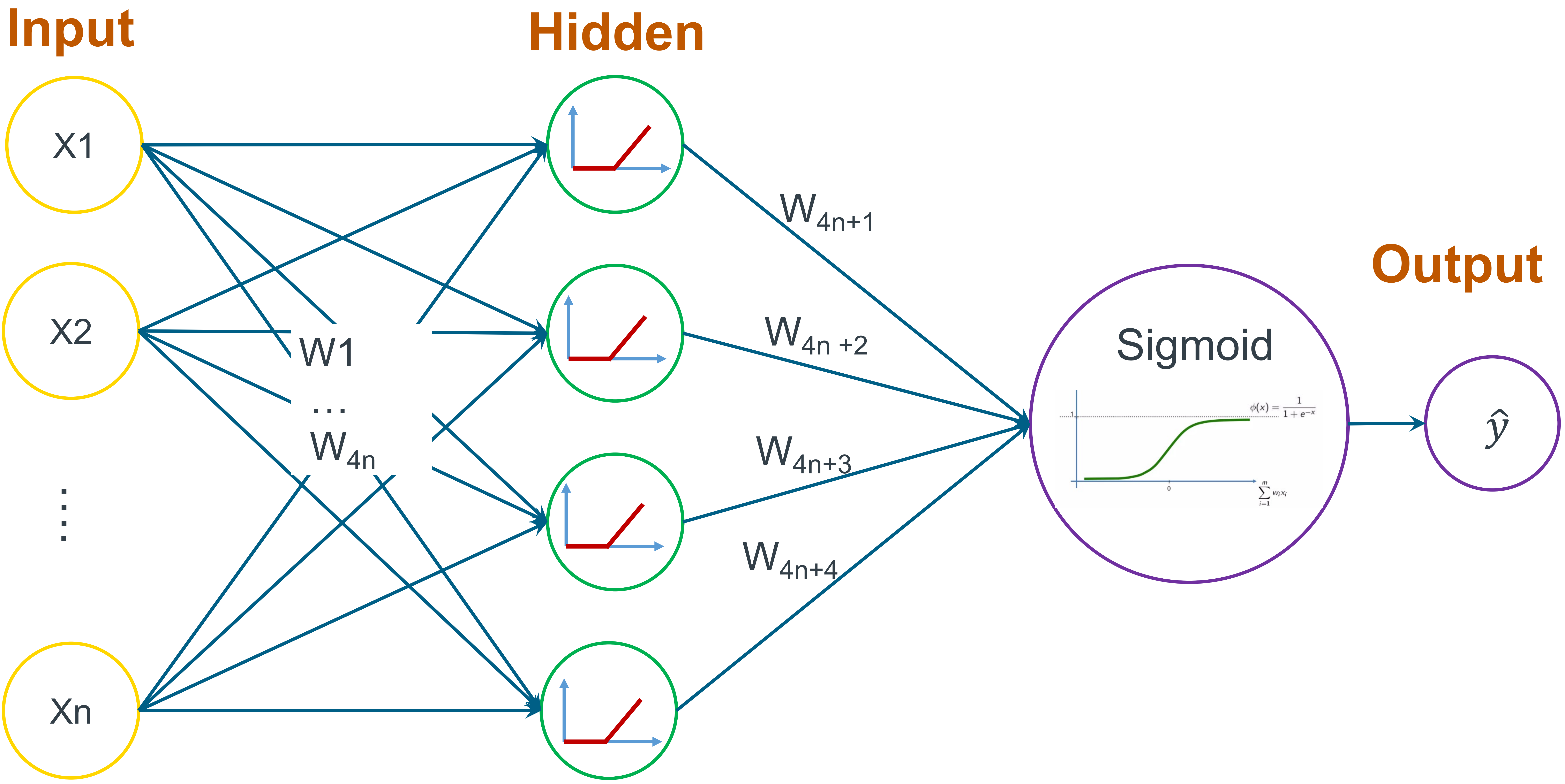
Synapses Weights

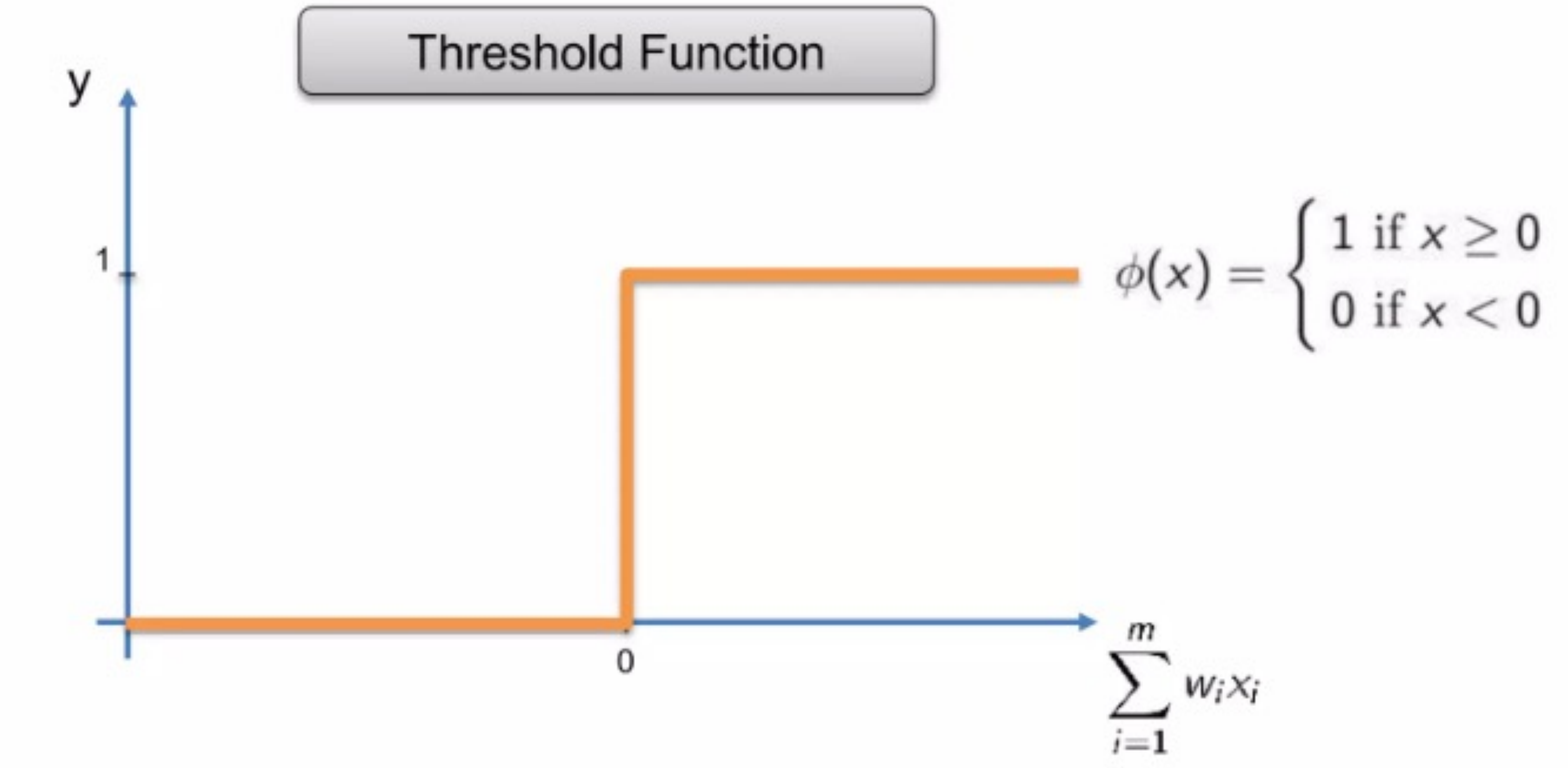
- Are how neural network learn
- Determine which inputs are important and which are not to a certain neuron.


Input
X1
X2
⋮
Xn
W1
W2
Wn
Step 1
$\sum_{i=1}^{n} w_i x_i$
Output
y


Input
X1
X2
⋮
Xn
W1
W2
Wn
Step 2
$\emptyset\left(\sum_{i=1}^{n} w_i x_i\right)$
Output
y
∅: activation function


Input
X1
X2
⋮
Xn
W1
W2
Wn
Step 2
$\emptyset\left(\sum_{i=1}^{n} w_i x_i\right)$
Step 3
Pass the signal on
Output
y
∅: activation function


Input
Hidden
Output
X1
X2
⋮
Xn
W1
…
W_{4n}
W_{4n+1}
W_{4n+2}
W_{4n+3}
W_{4n+4}
Sigmoid
$\phi(x) = \frac{1}{1+e^{-x}}$
$\hat{y}$

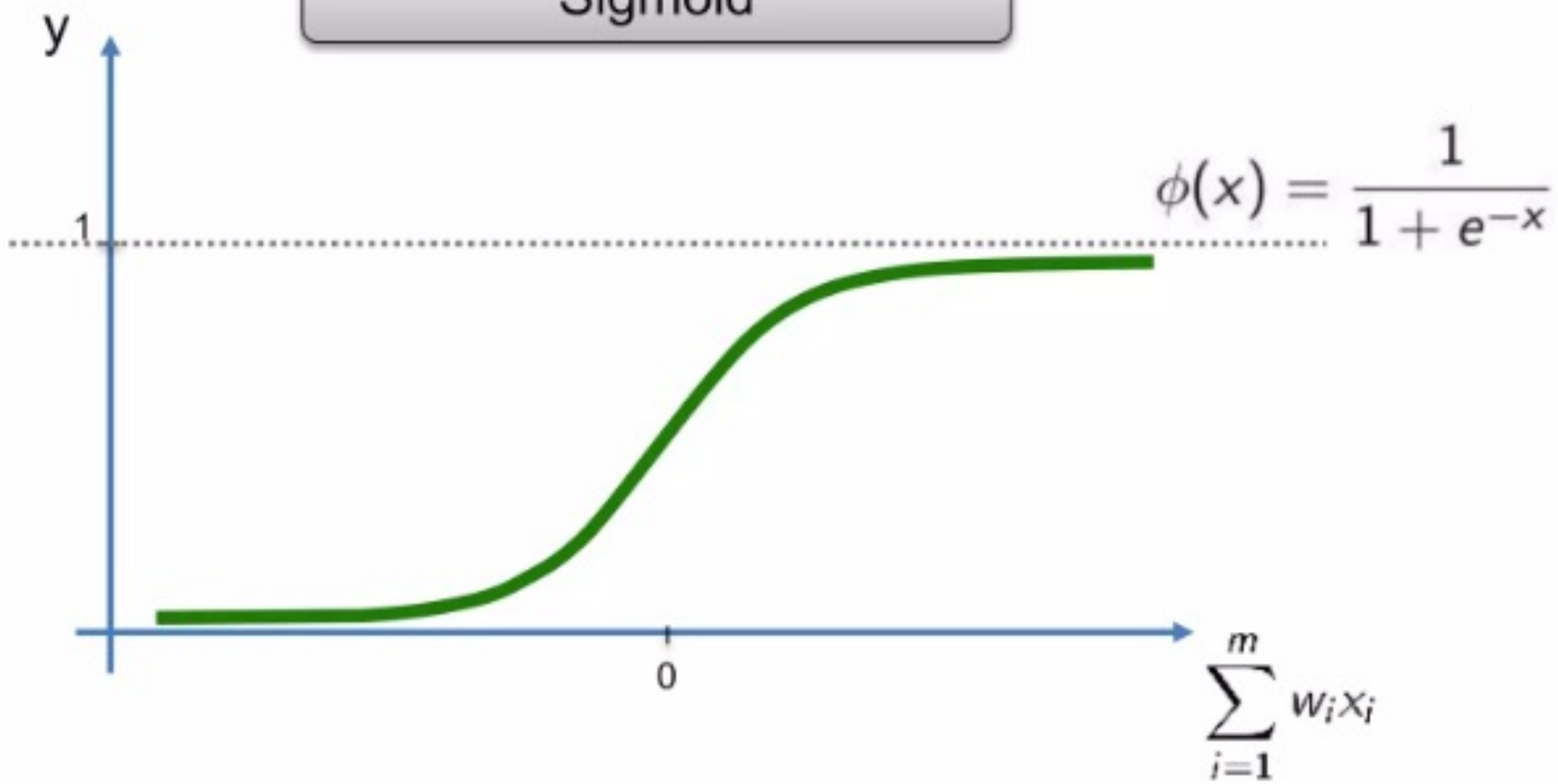


✓ depends on a threshold value that decides whether a neuron should be activated or not.

Limitation

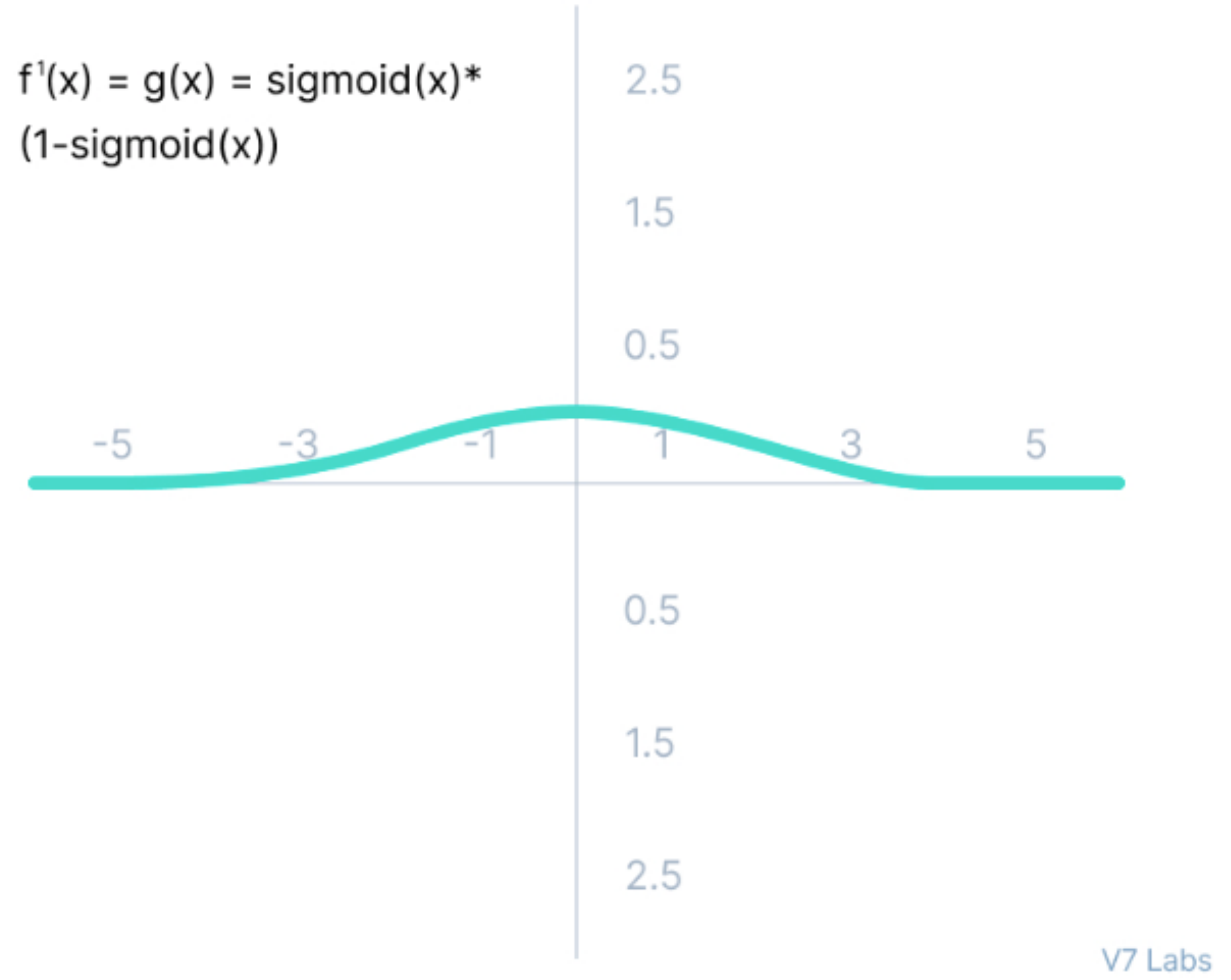
- It cannot provide multi-value outputs—for example, it cannot be used for multi-class classification problems.
- The gradient of the step function is zero, which causes a hindrance in the backpropagation process.

Sigmoid



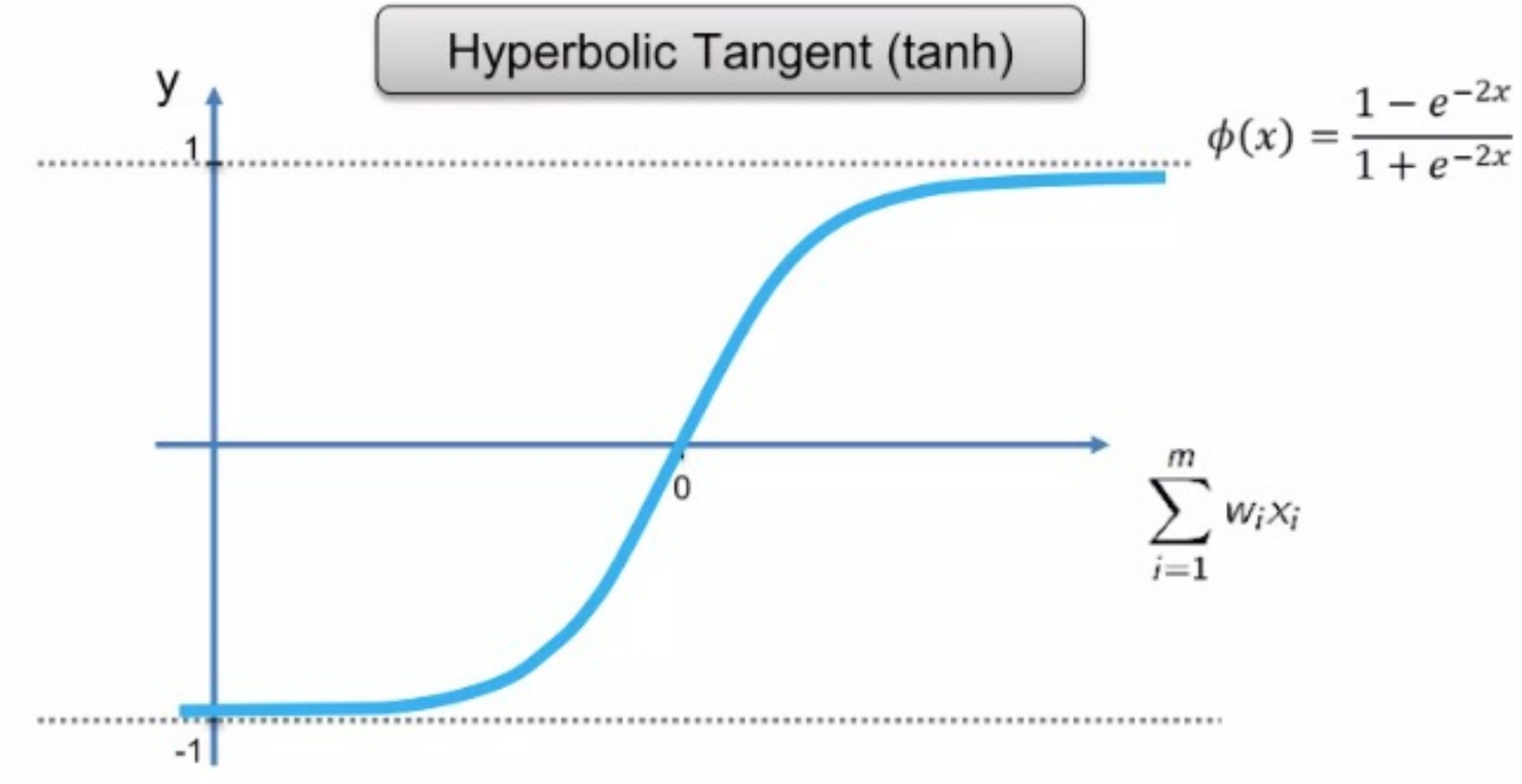
- ✓ Takes any real value as input and outputs values in the range of 0 to 1.
- ✓ One of the most widely used functions
- ✓ Commonly used for models where the probability is an output
- ✓ The function is differentiable and provides a smooth gradient, i.e., preventing jumps in output values. This is represented by an S-shape of the sigmoid activation function.

Limitation



The derivative of the Sigmoid Activation Function

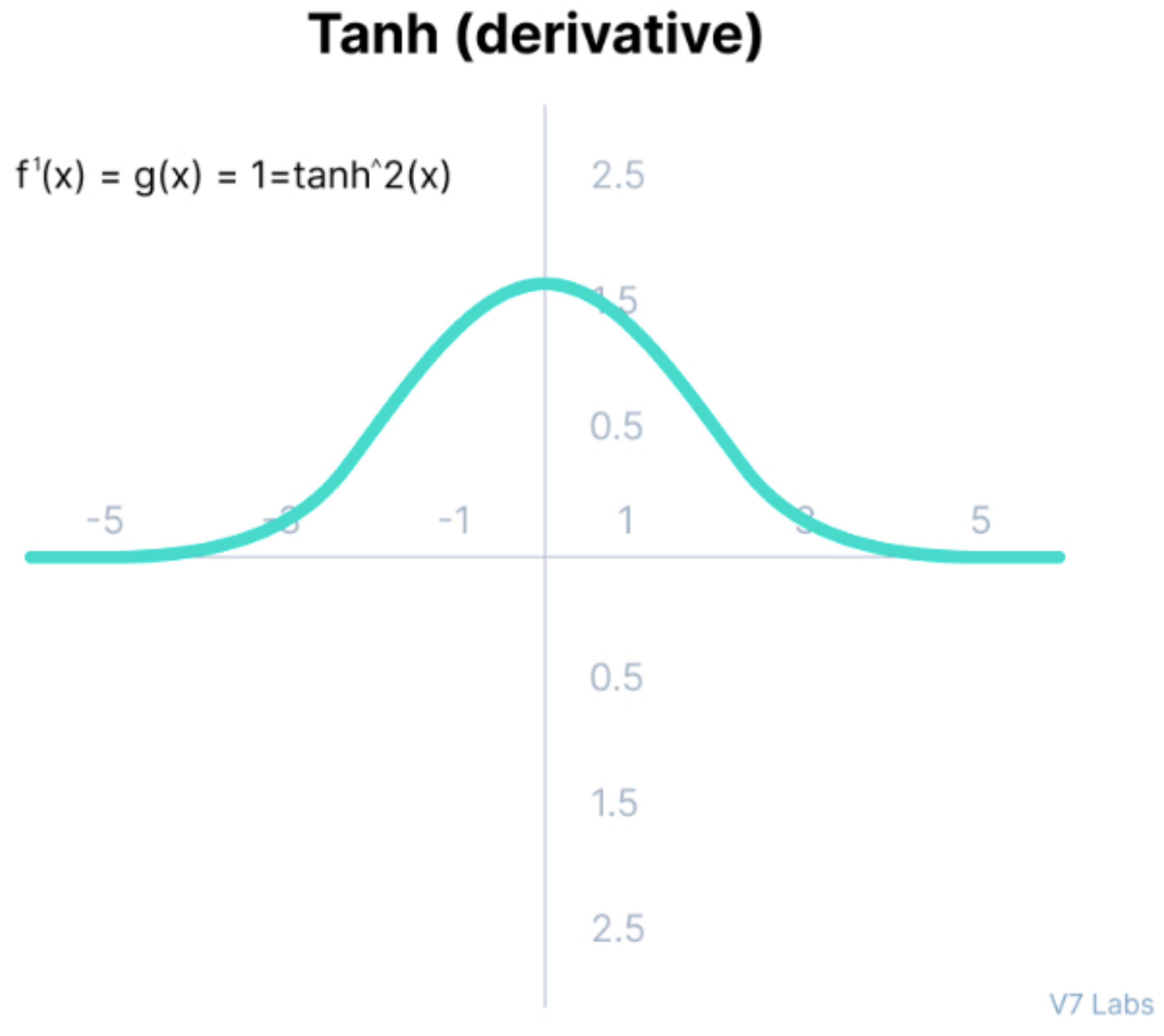
- For values greater than 3 or less than -3, the network ceases to learn and suffers from the Vanishing gradient problem.
- Sigmoid outputs are not zero-centered, which makes the training of the neural network more difficult and unstable.



✓ The output of the tanh activation function is Zero centered;

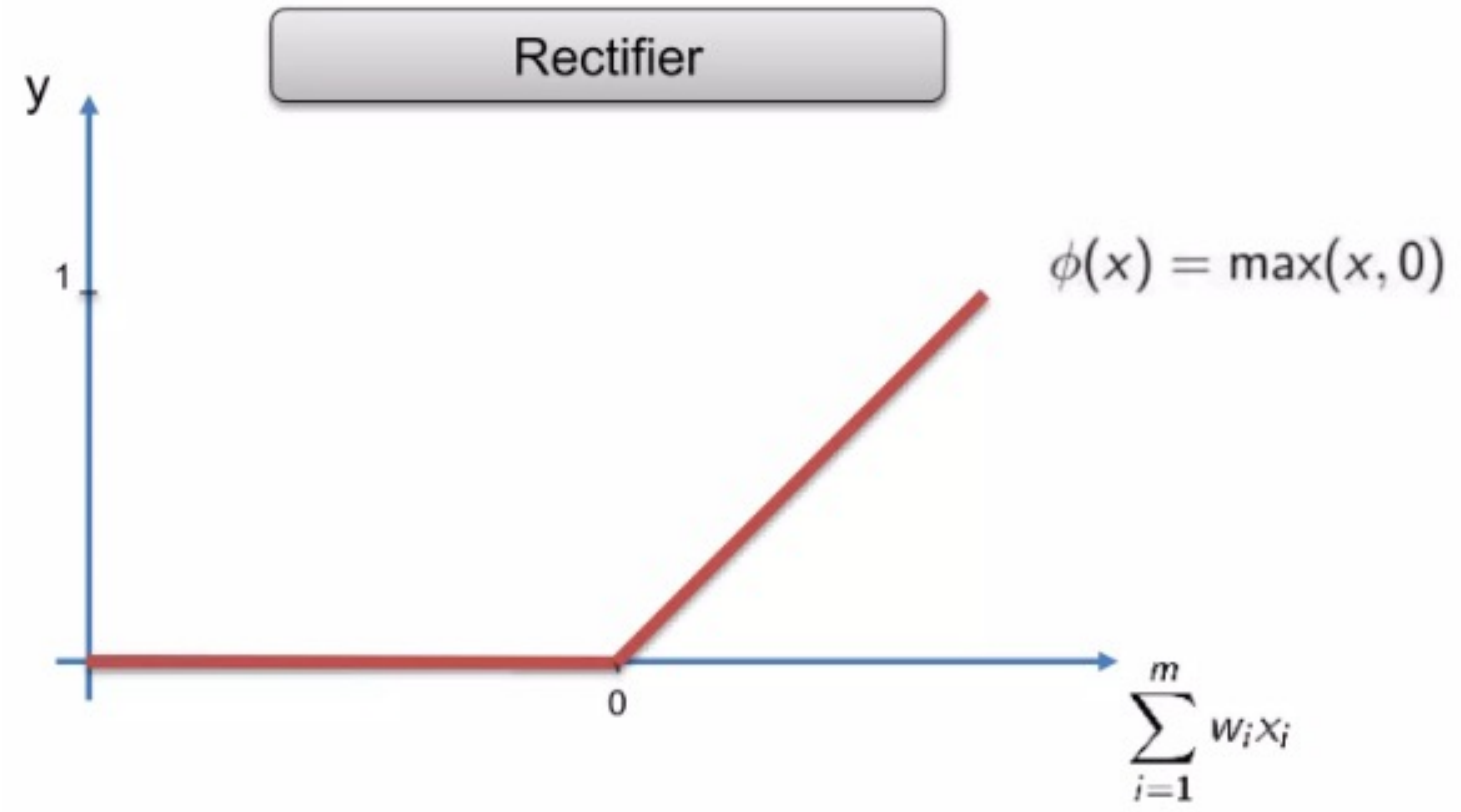
✓ Usually used in hidden layers of a neural network as its values lie between -1 to 1; therefore, the mean for the hidden layer comes out to be 0 or very close to it. It helps in centering the data and makes learning for the next layer much easier.

Limitation



Gradient of the Tanh Activation Function

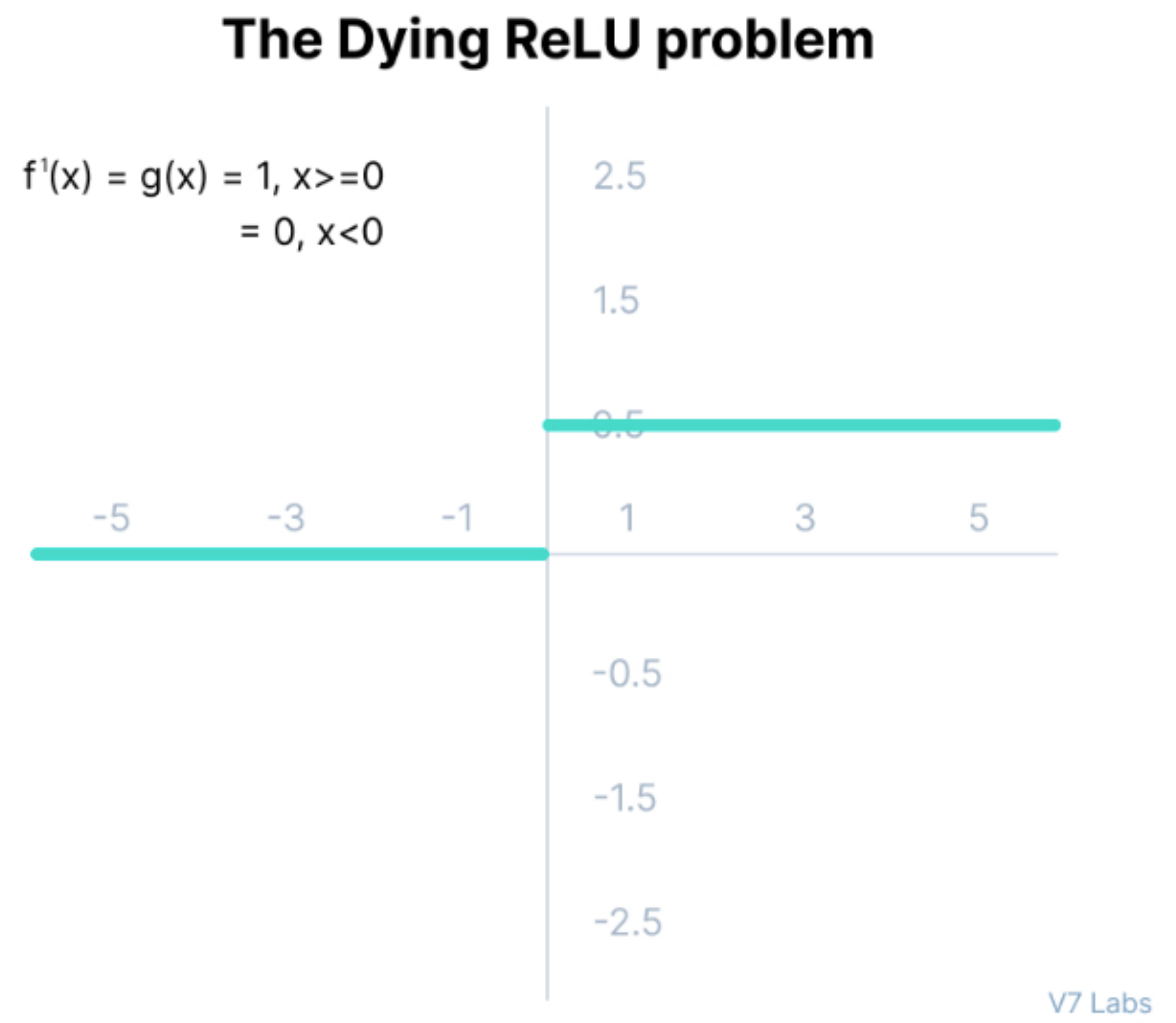
- it also faces the problem of vanishing gradients similar to the sigmoid activation function. Plus, the gradient of the tanh function is much steeper as compared to the sigmoid function.



✓Only a certain number of neurons are activated, making the ReLU function more computationally efficient than the sigmoid and tanh functions.

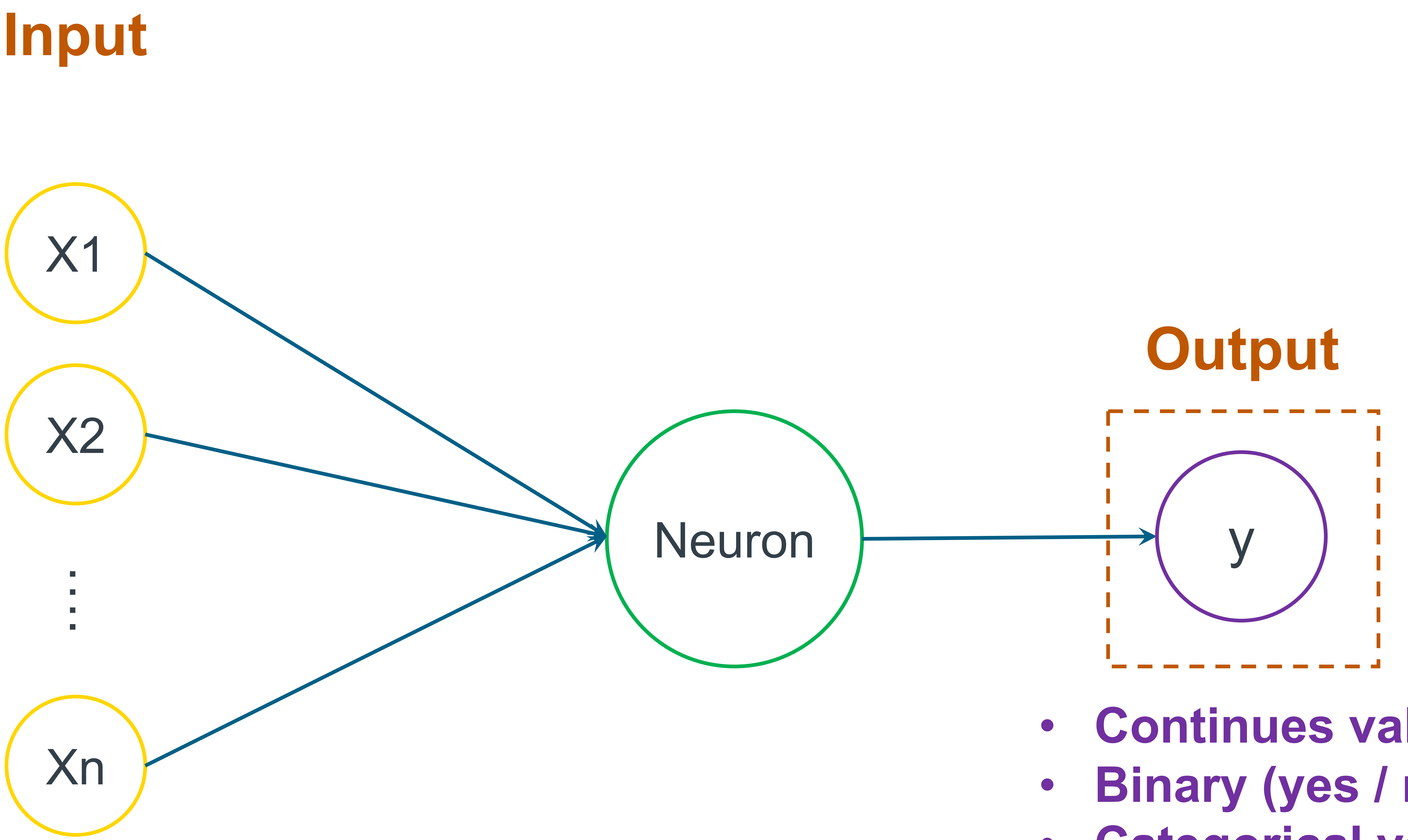
✓ReLU accelerates the convergence of gradient descent towards the global minimum of the loss function due to its linear, non-saturating property.

Limitation



The Dying ReLU problem

- The dying ReLU problem could create dead neurons which never get activated.
- All the negative input values become zero, which decreases the model's ability to fit or train from the data properly.

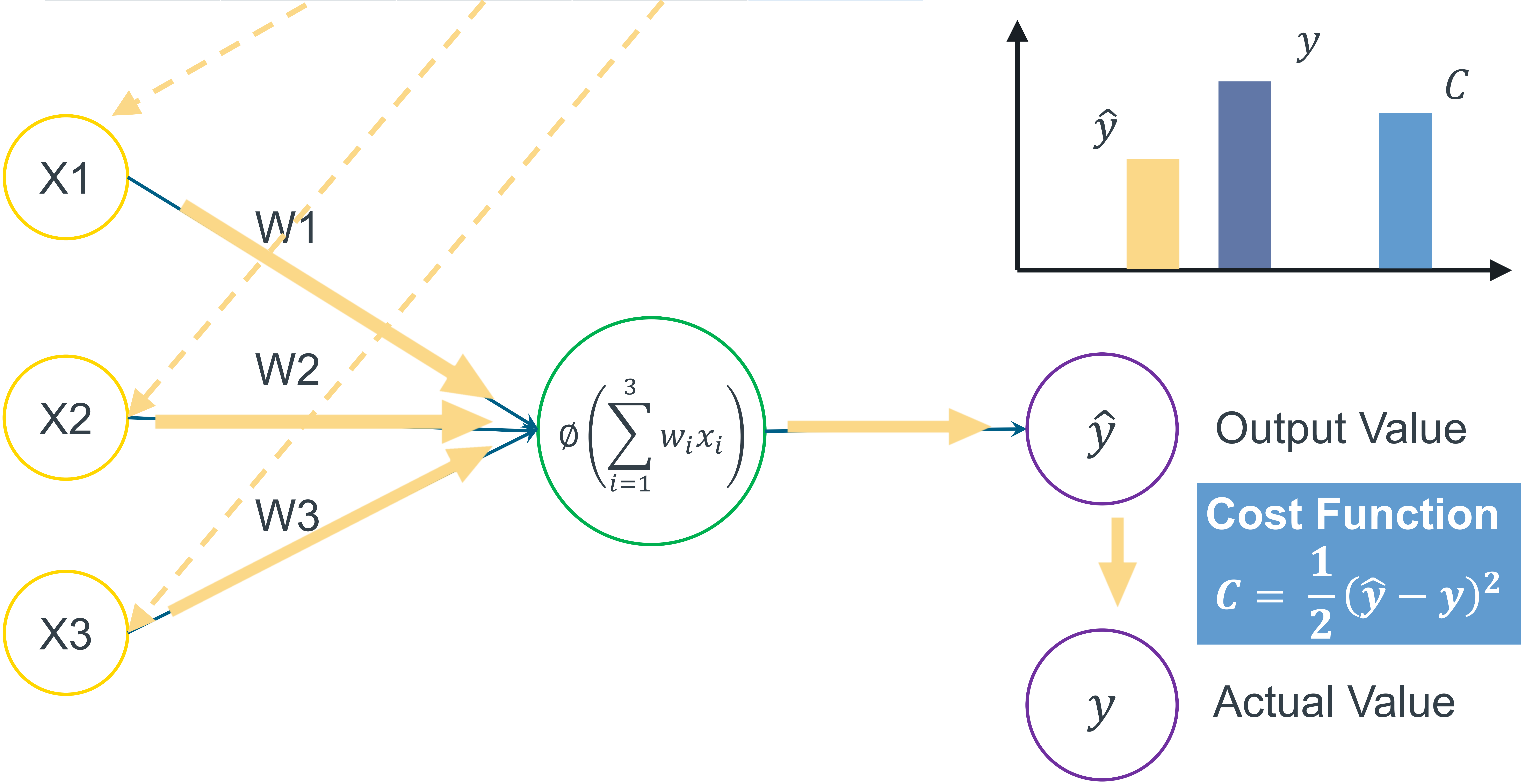


- Continues value
- Binary (yes / no)
- Categorical value: y1, y2, y3,…

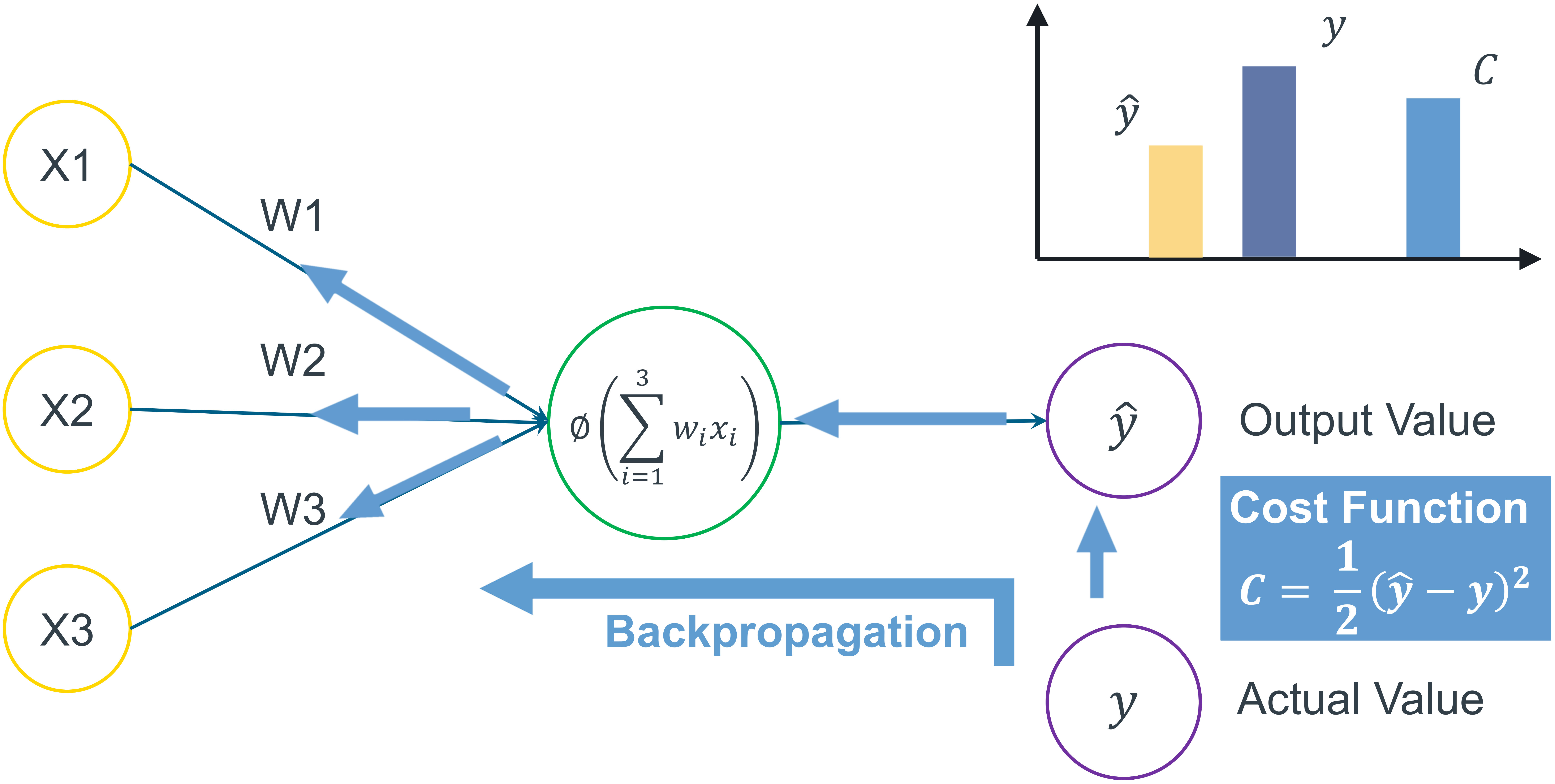
Some Basic Concepts of Neural Network

How Do Neural Networks Work

Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	90%



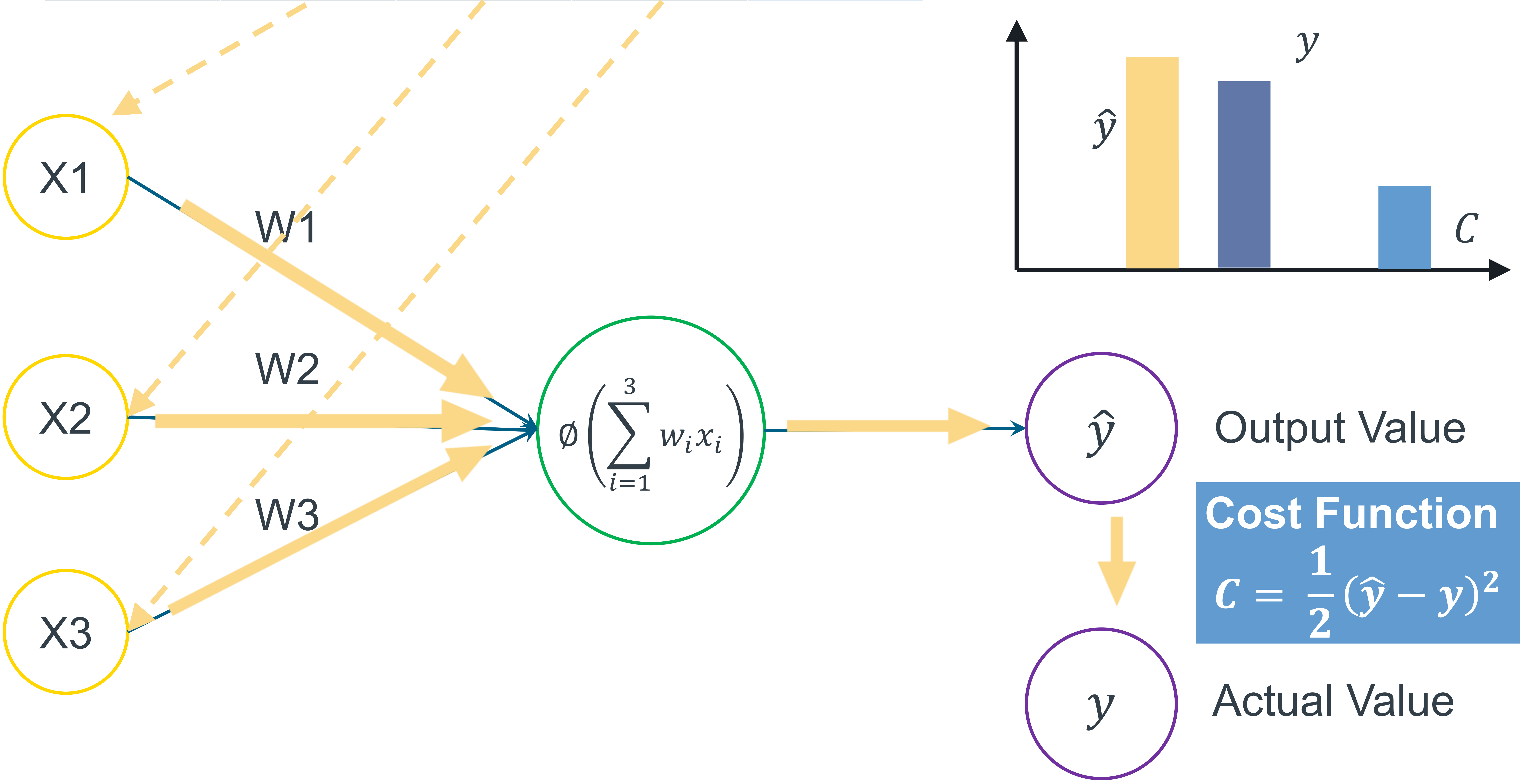
Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	90%



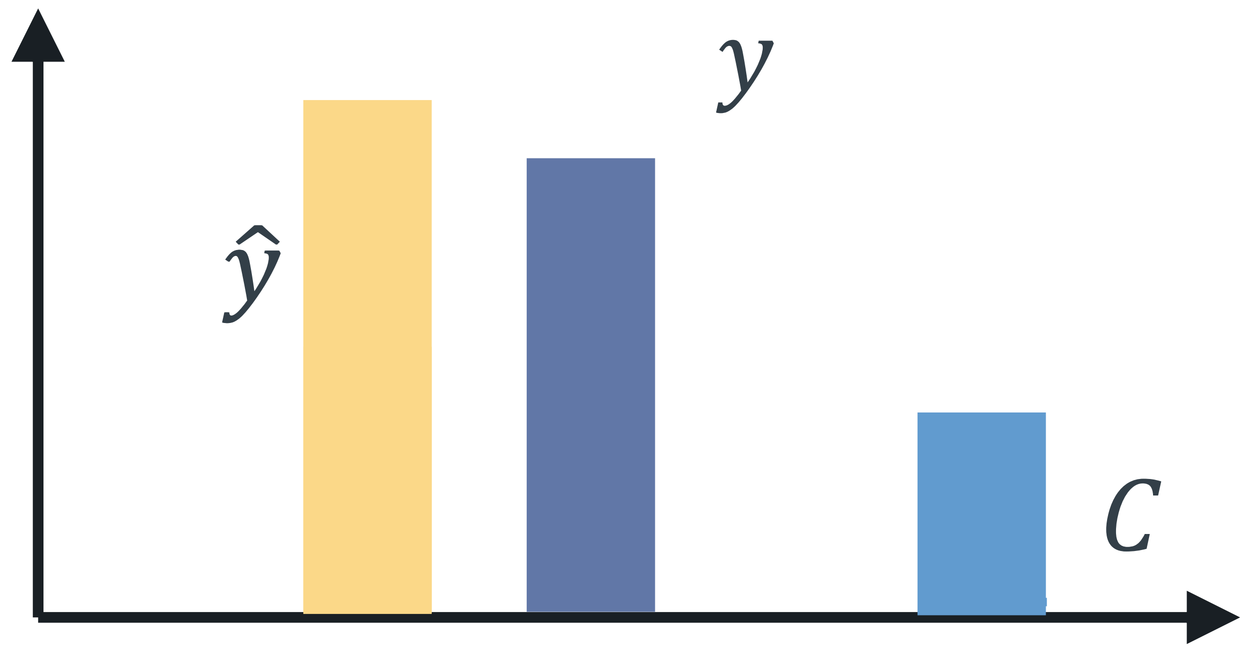
Some Basic Concepts of Neural Network

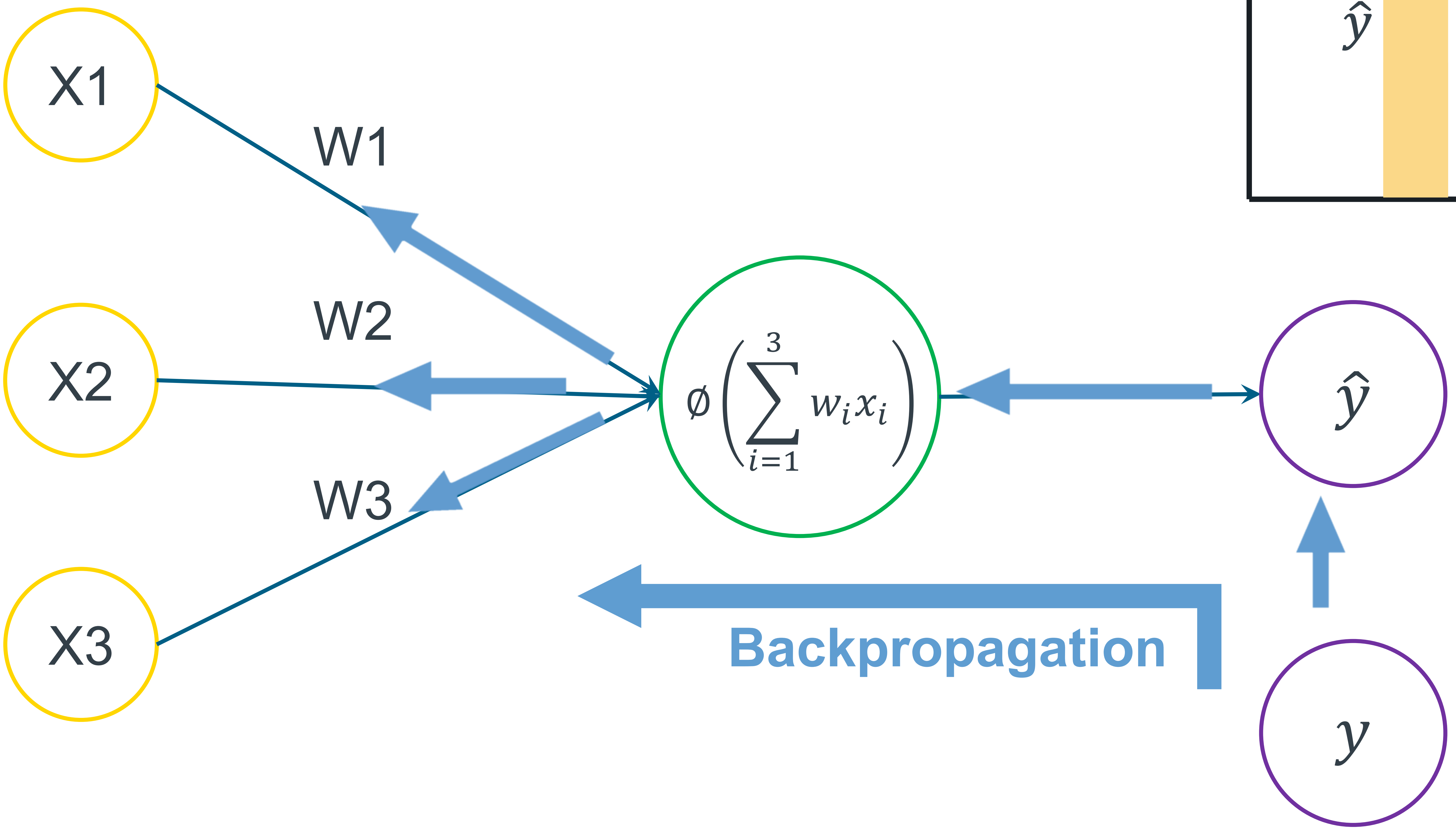
How Do Neural Networks Work

Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	90%



Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	90%

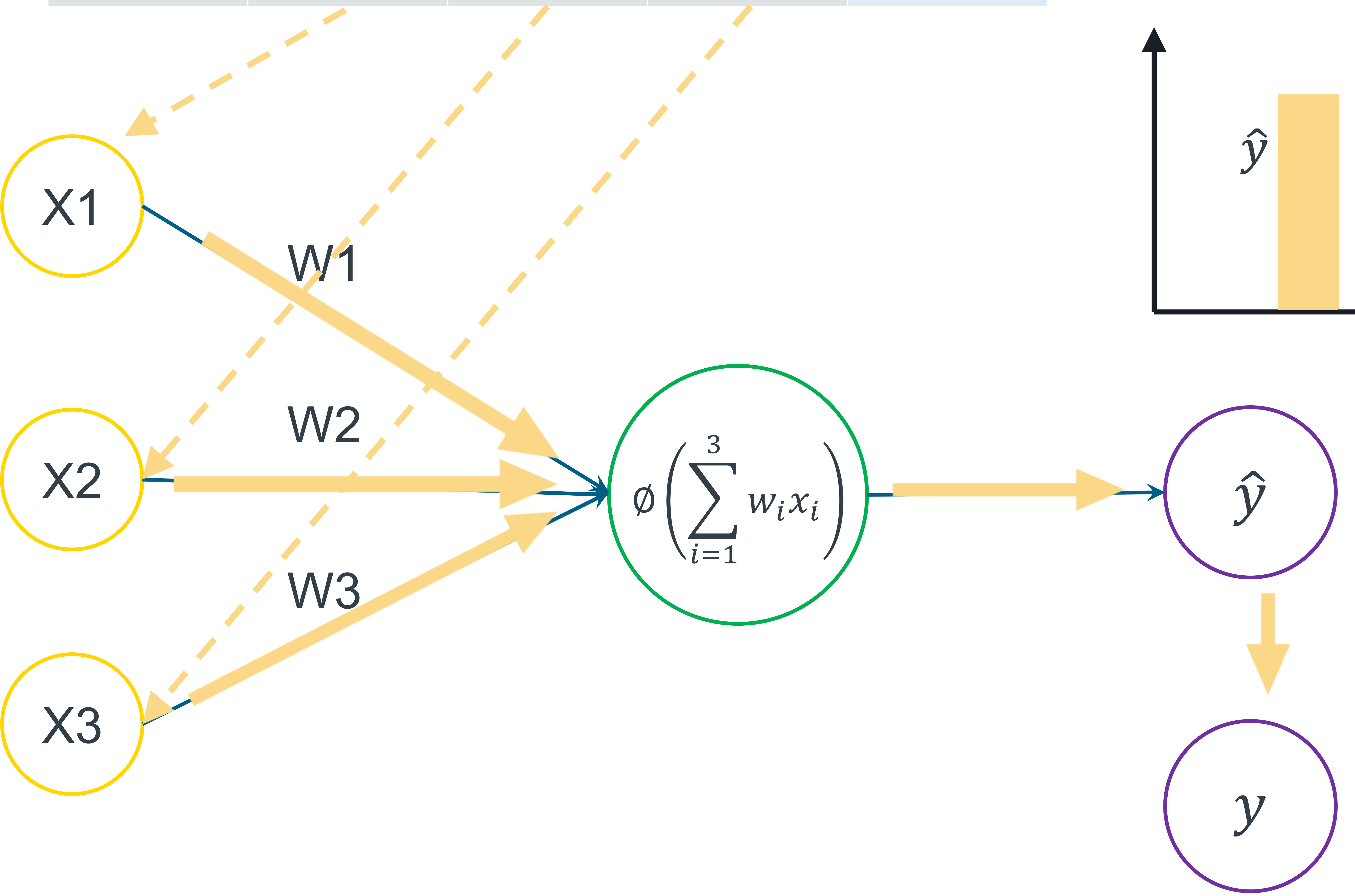


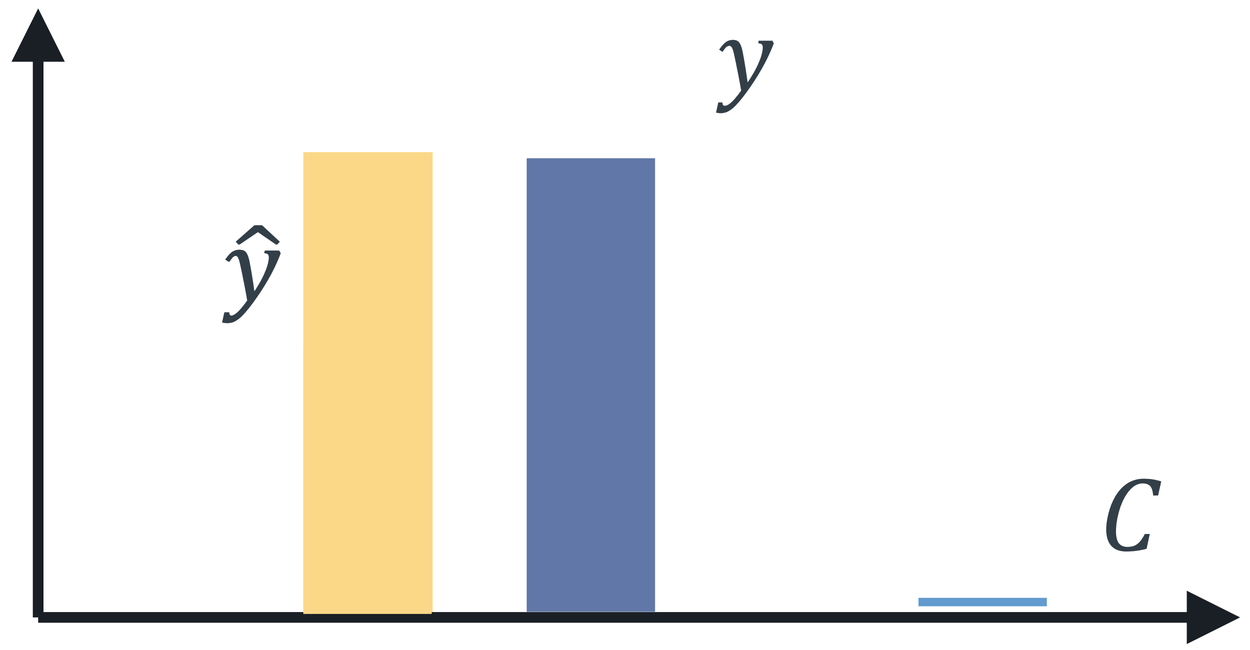


Cost Function

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

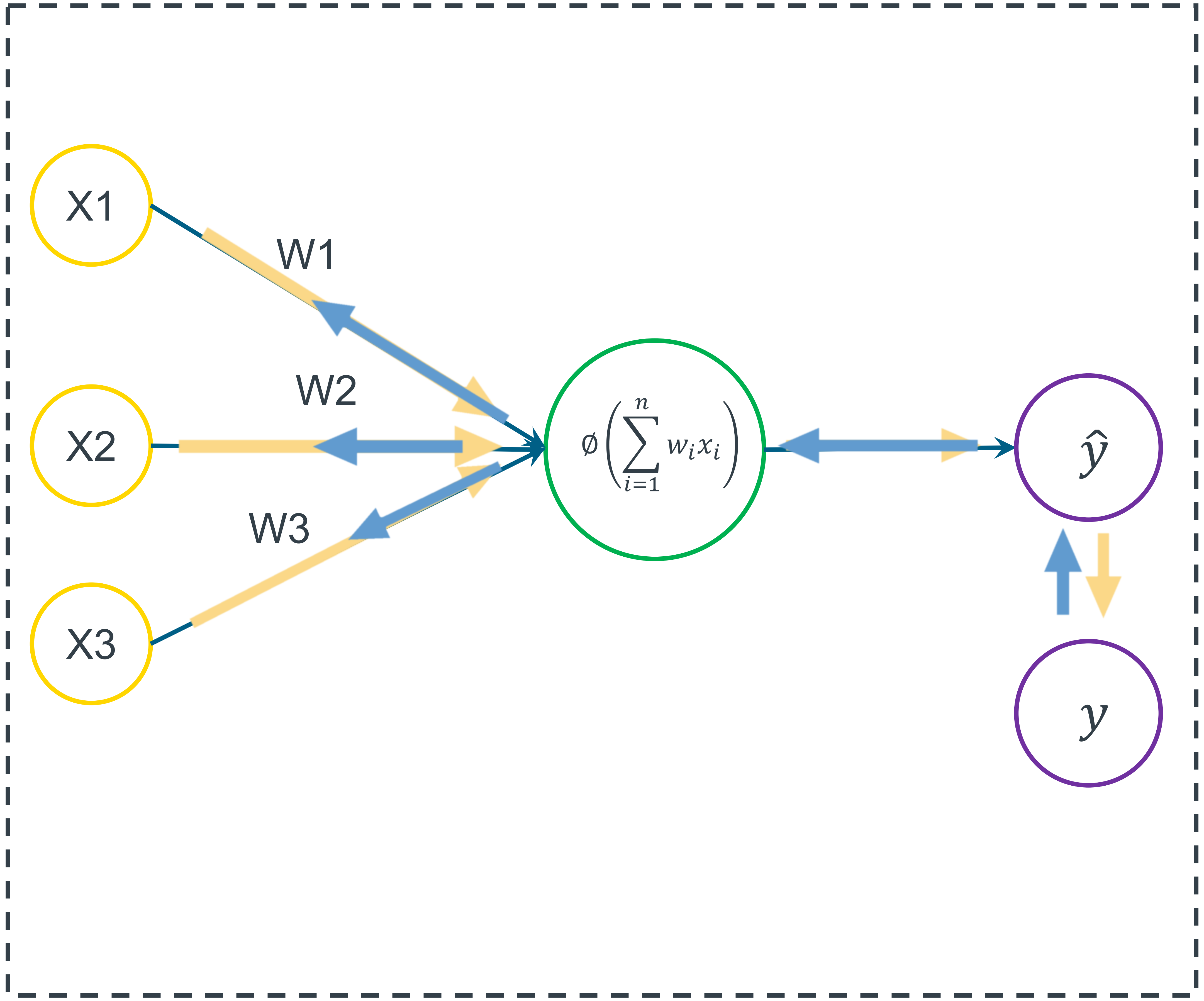
Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	90%





Cost Function

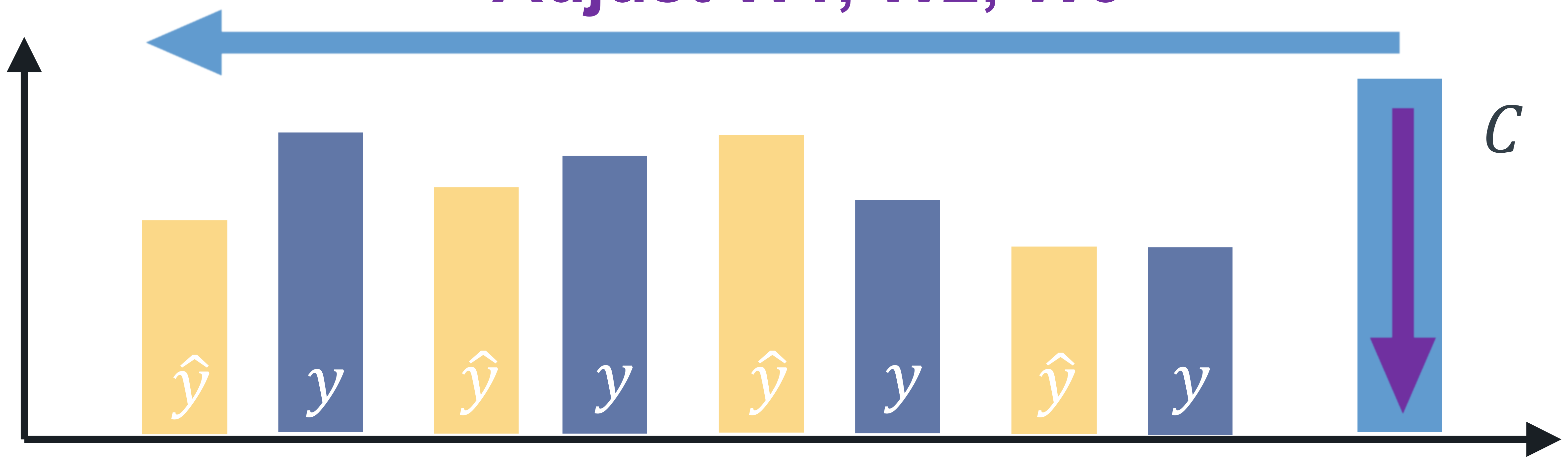
$$C = \frac{1}{2}(\hat{y} - y)^2$$



Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	90%
2	11	8	77%	88%
3	8	8	80%	75%
4	6	12	70%	68%

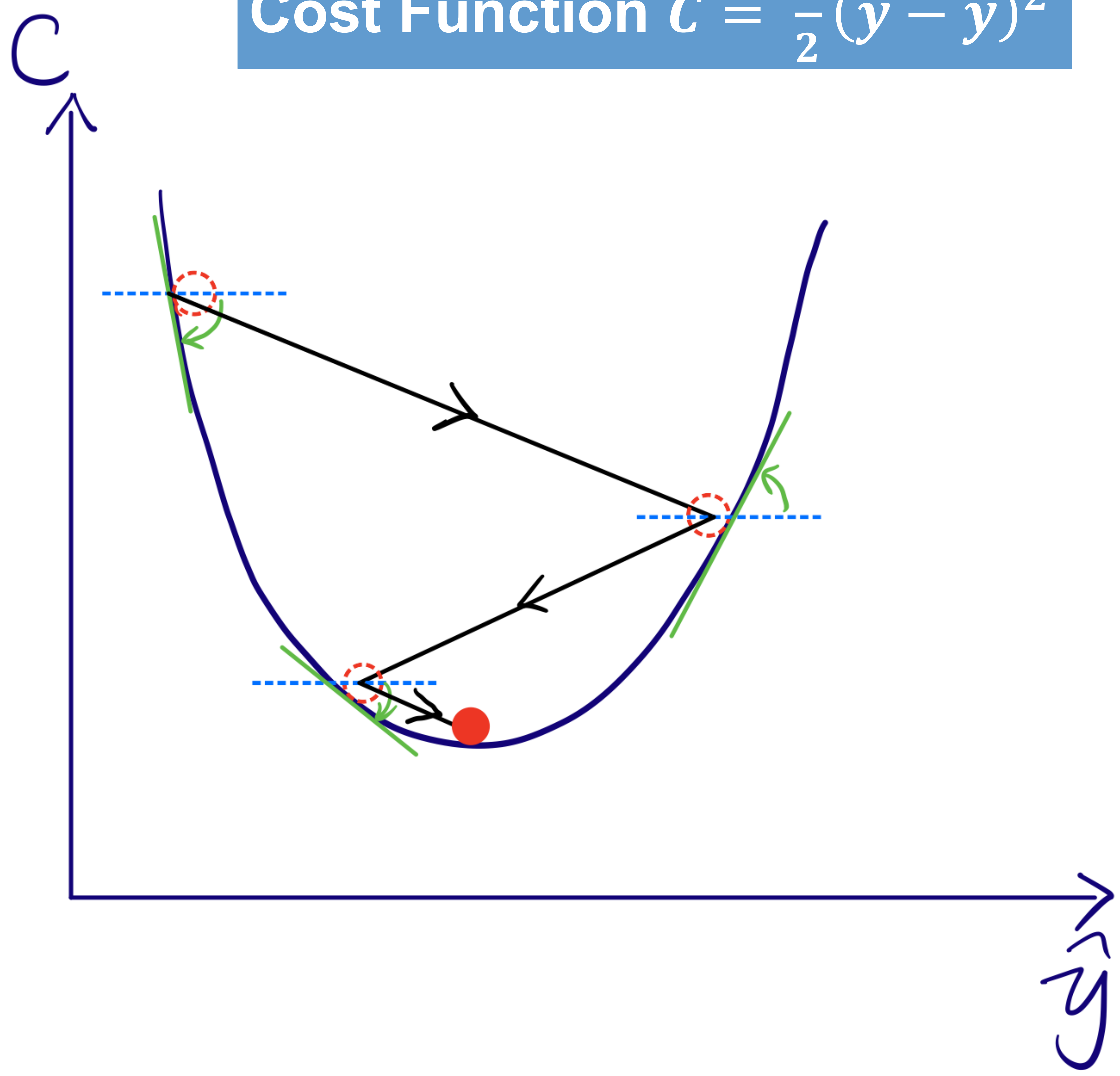
Cost Function: $C = \sum \frac{1}{2}(\hat{y} - y)^2$

Adjust W1, W2, W3



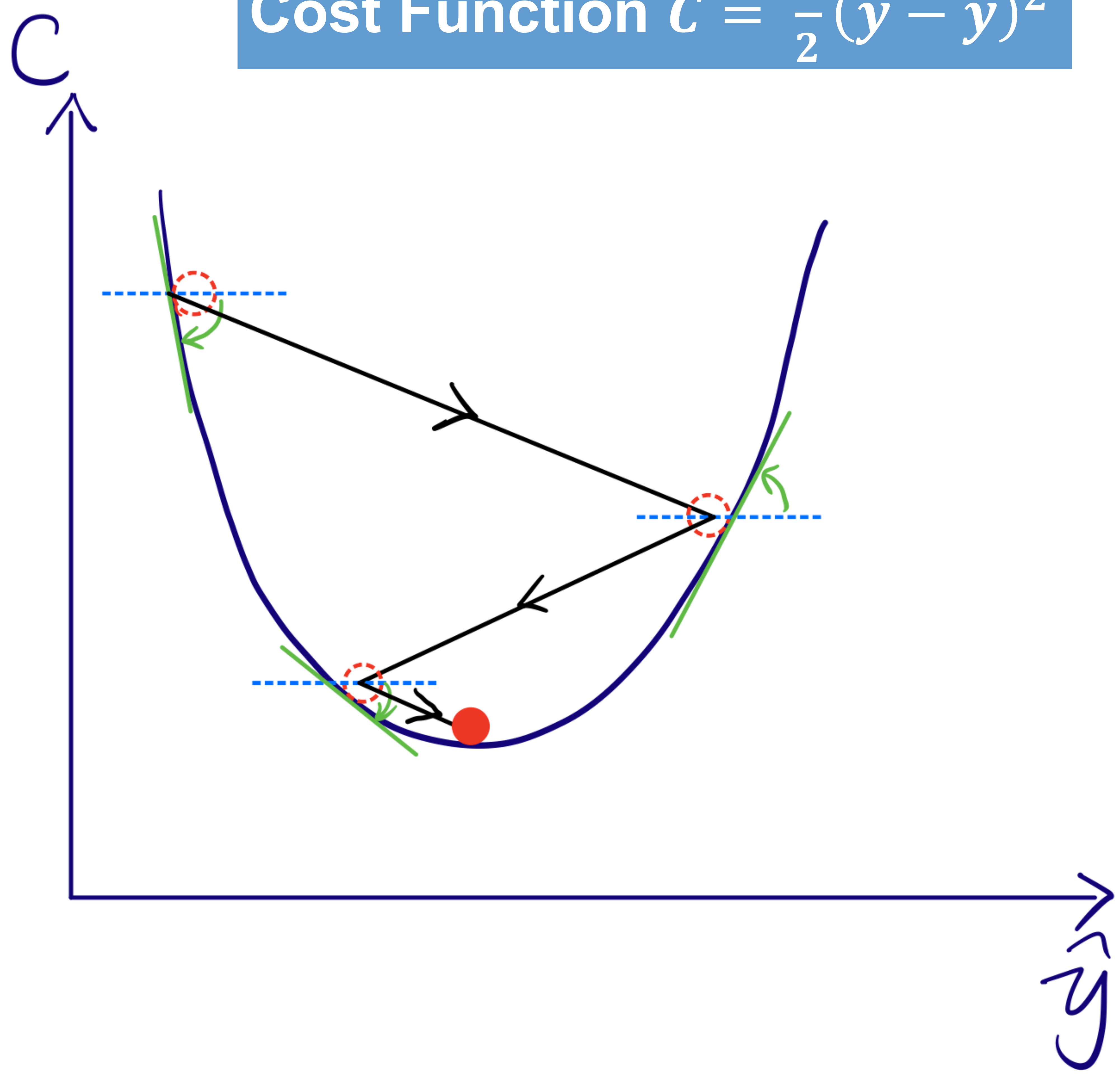
➢ **Gradient Descent**

Cost Function $C = \frac{1}{2}(\hat{y} - y)^2$

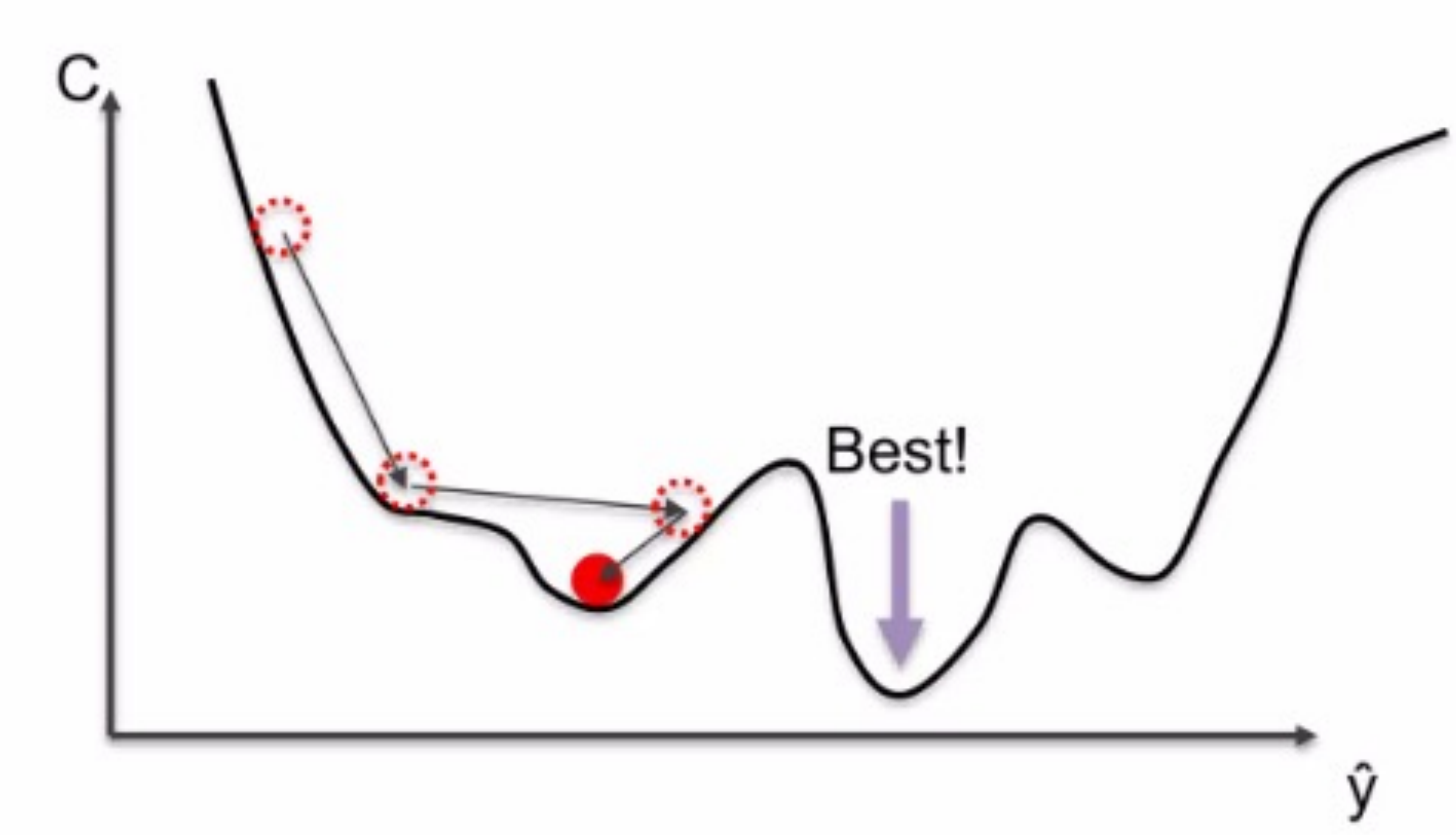


➢Gradient Descent

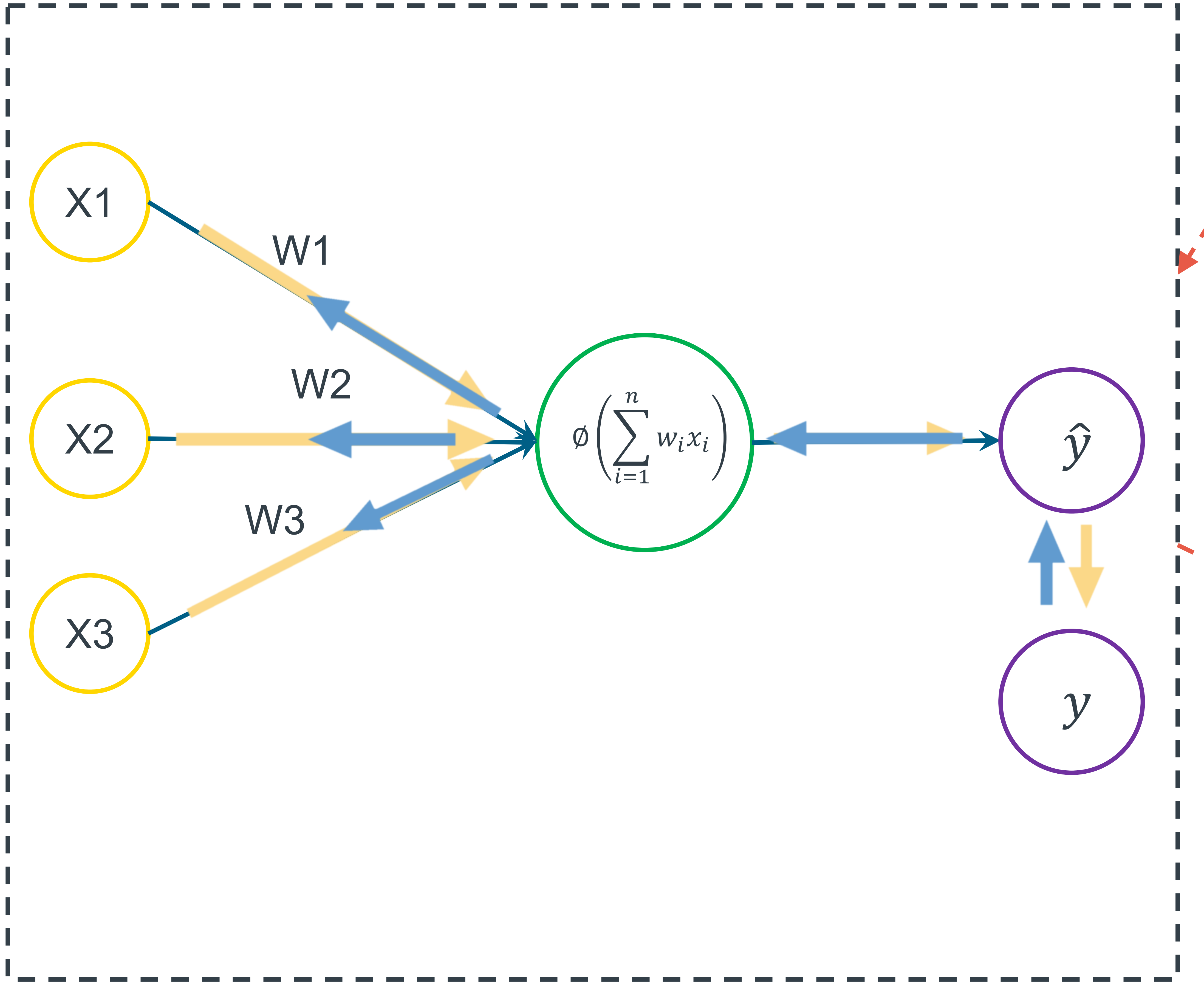
Cost Function $C = \frac{1}{2}(\hat{y} - y)^2$



However,….



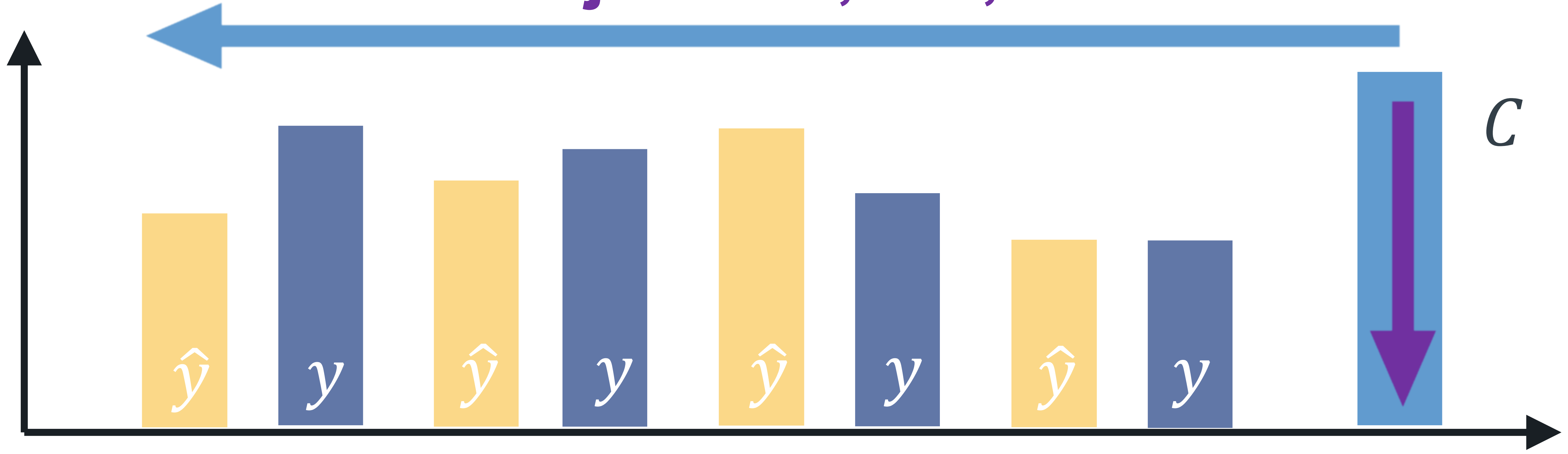
➢Normal Gradient Descent



Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	90%
2	11	8	77%	88%
3	8	8	80%	75%
4	6	12	70%	68%

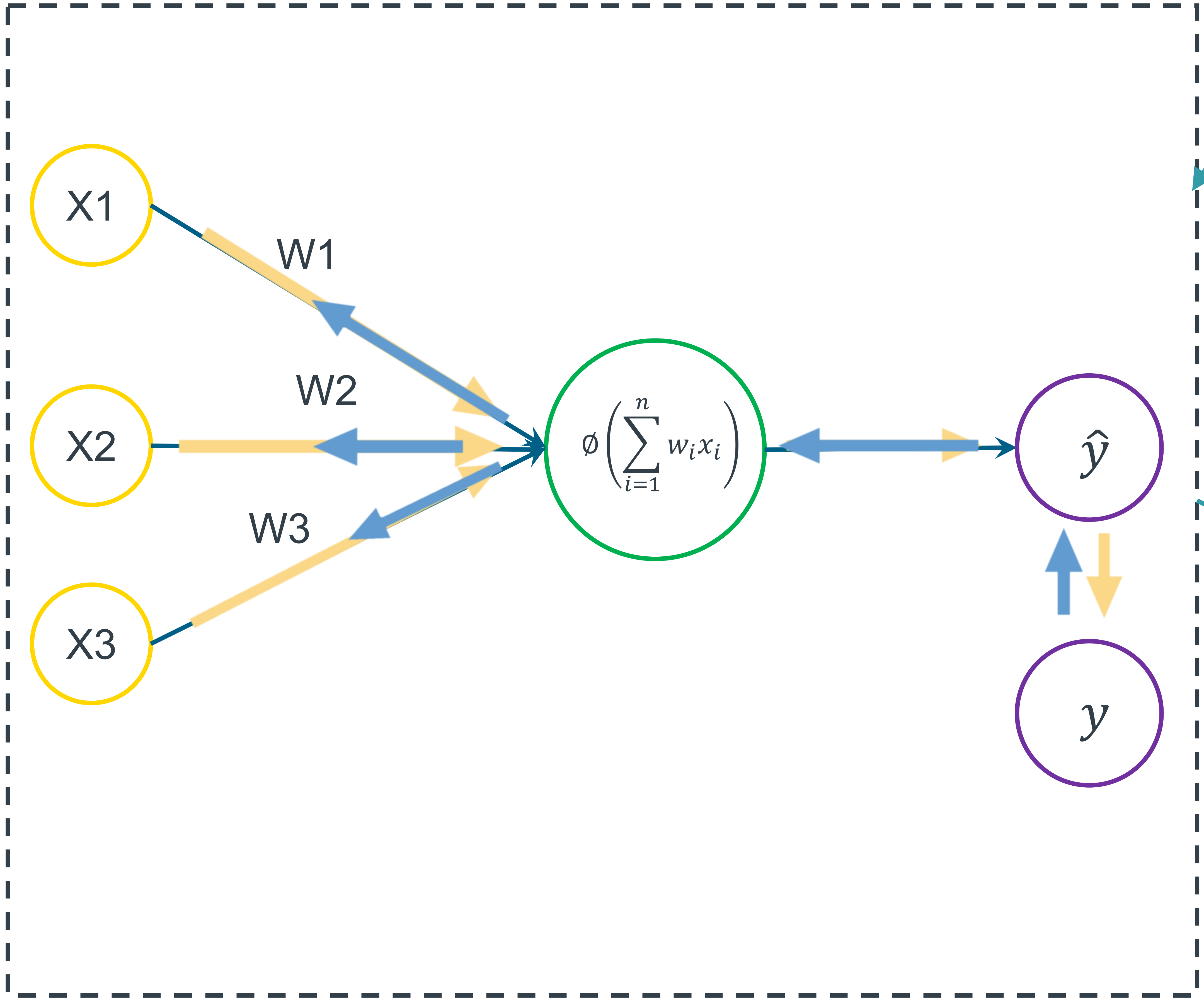
Cost Function: $C = \sum \frac{1}{2}(\hat{y} - y)^2$

Adjust W1, W2, W3



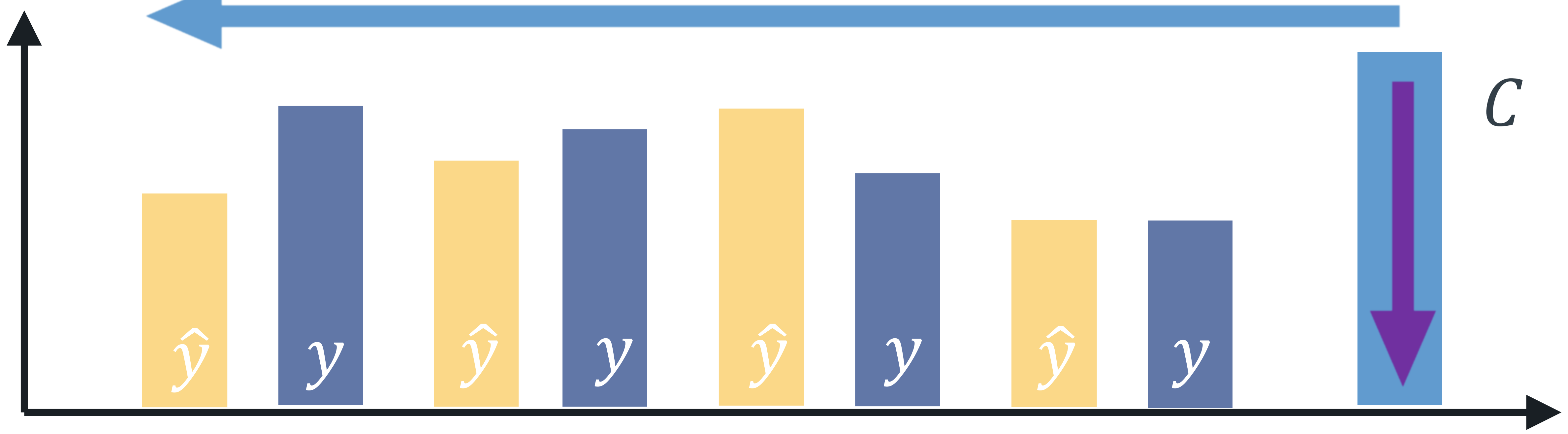
➢Stochastic Gradient Descent

Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	90%
2	11	8	77%	88%
3	8	8	80%	75%
4	6	12	70%	68%

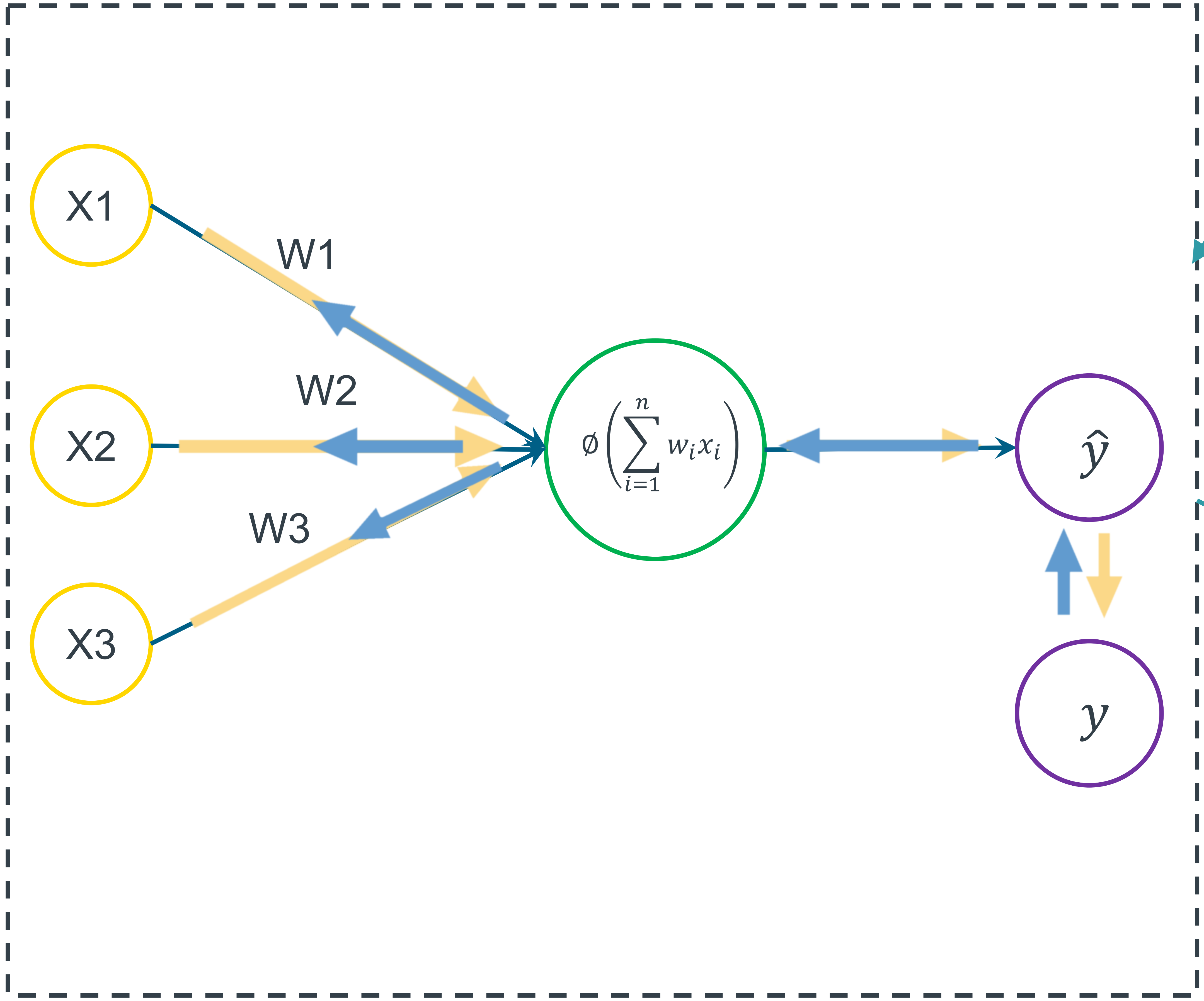


Cost Function: $C = \sum \frac{1}{2}(\hat{y} - y)^2$

Adjust W1, W2, W3



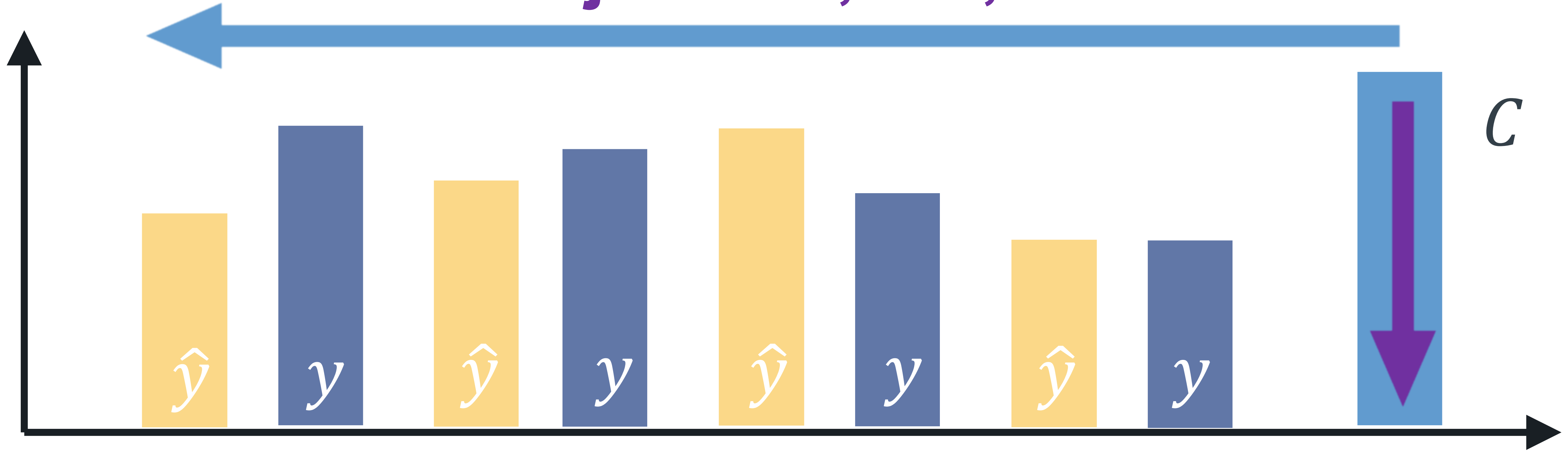
➢Stochastic Gradient Descent



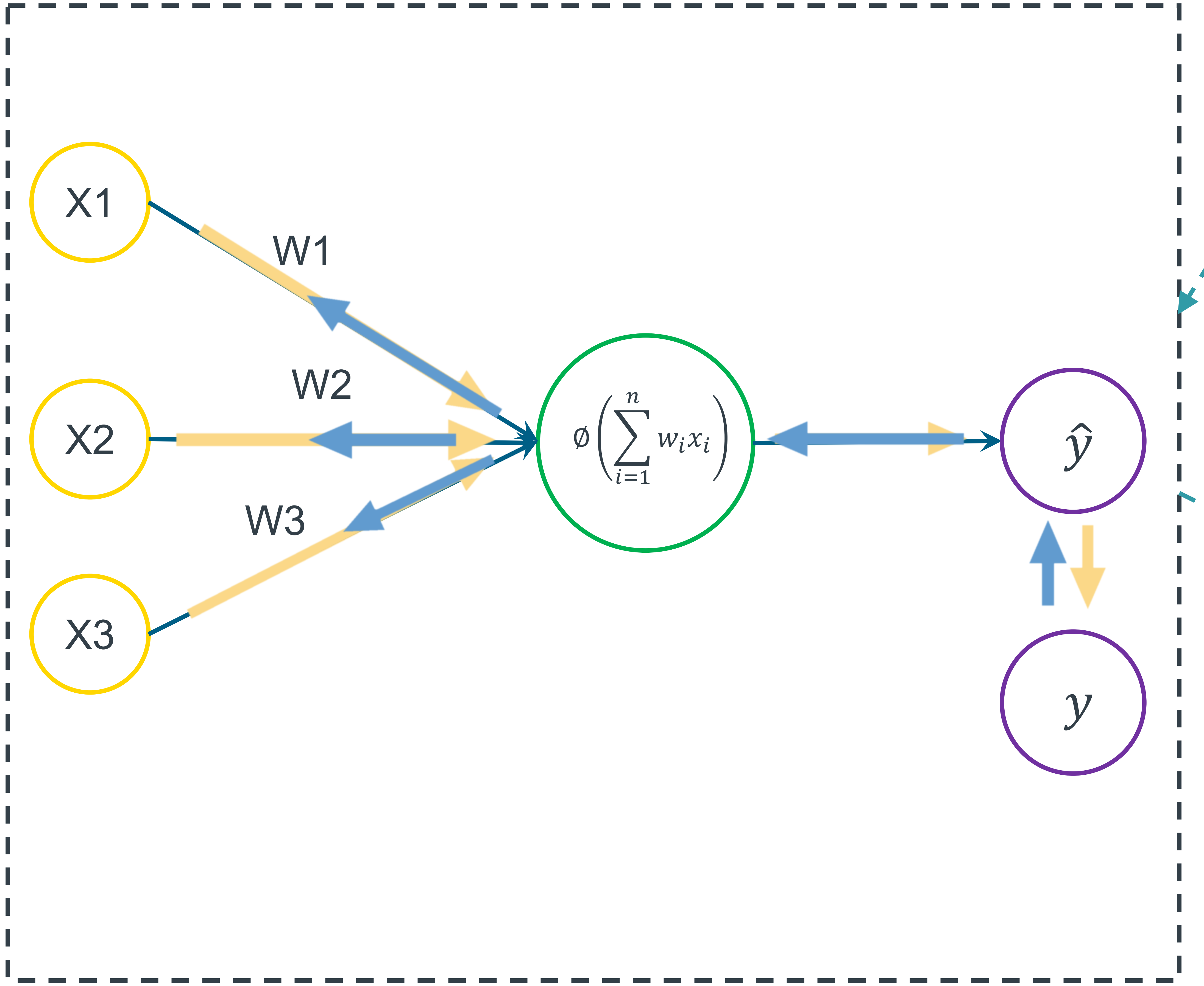
Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	90%
2	11	8	77%	88%
3	8	8	80%	75%
4	6	12	70%	68%

Cost Function: $C = \sum \frac{1}{2}(\hat{y} - y)^2$

Adjust W1, W2, W3



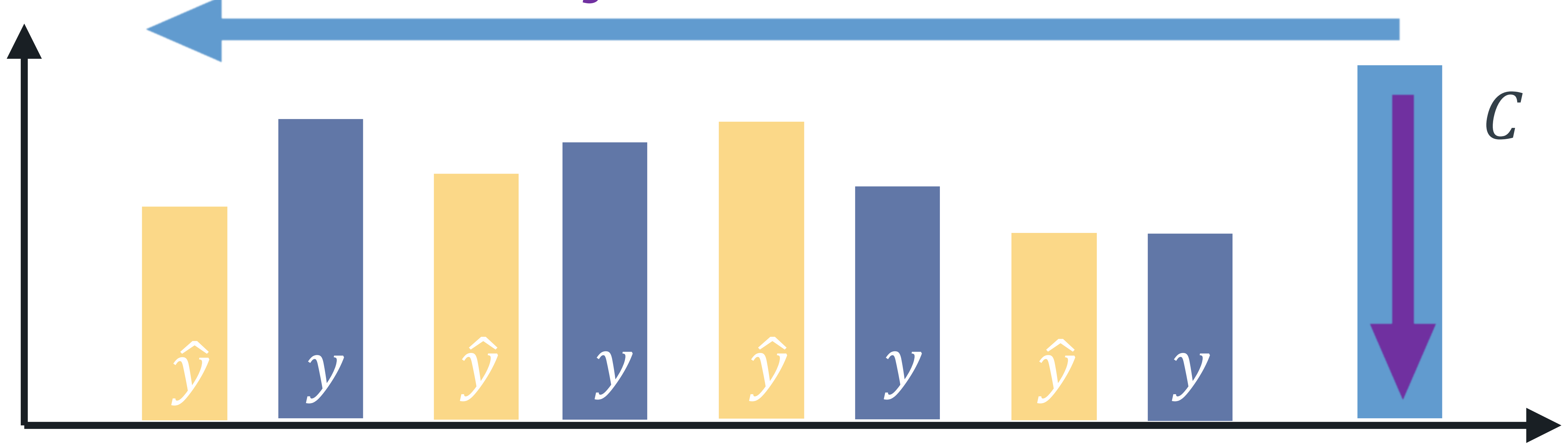
➢Stochastic Gradient Descent



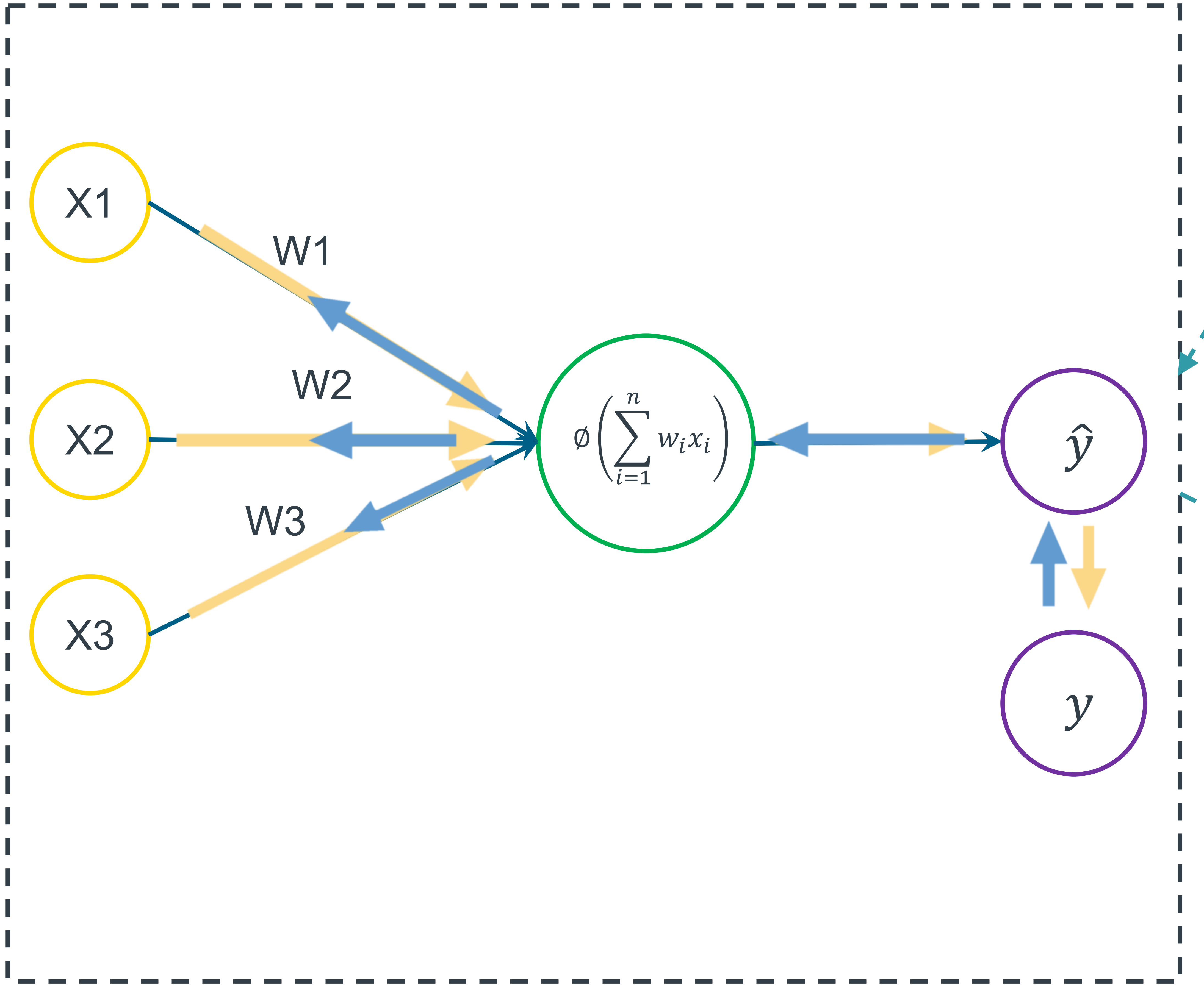
Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	90%
2	11	8	77%	88%
3	8	8	80%	75%
4	6	12	70%	68%

Cost Function: $C = \sum \frac{1}{2}(\hat{y} - y)^2$

Adjust W1, W2, W3



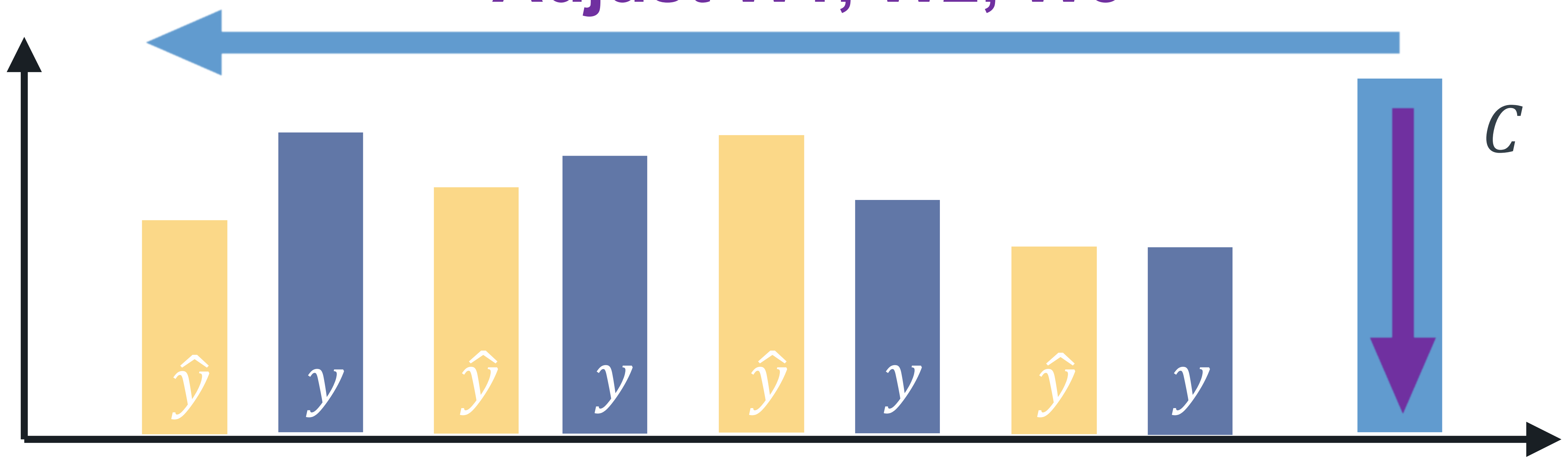
➢Stochastic Gradient Descent



Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	90%
2	11	8	77%	88%
3	8	8	80%	75%
4	6	12	70%	68%

Cost Function: $C = \sum \frac{1}{2}(\hat{y} - y)^2$

Adjust W1, W2, W3



Upd W's

Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	90%
2	11	8	77%	88%
3	8	8	80%	75%
4	6	12	70%	68%

Normal (Batch) Gradient Descent

✓Easy to flow to the local minimum

✓It has lower efficiency

✓The main advantage of GD is that it is a deterministic algorithm, i.e., each time you have the same initial weights, you will get the same process to update your weights.

	Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
Upd W's	1	12	7	79%	90%
Upd W's	2	11	8	77%	88%
Upd W's	3	8	8	80%	75%
Upd W's	4	6	12	70%	68%

Stochastic Gradient Descent

✓Row-by-row running makes SGD has much higher fluctuation and thus more likely to find the global minimum

✓It's a faster algorithm than GD

✓SGD cannot ensure the same update of weights even if the initial weights are the same.

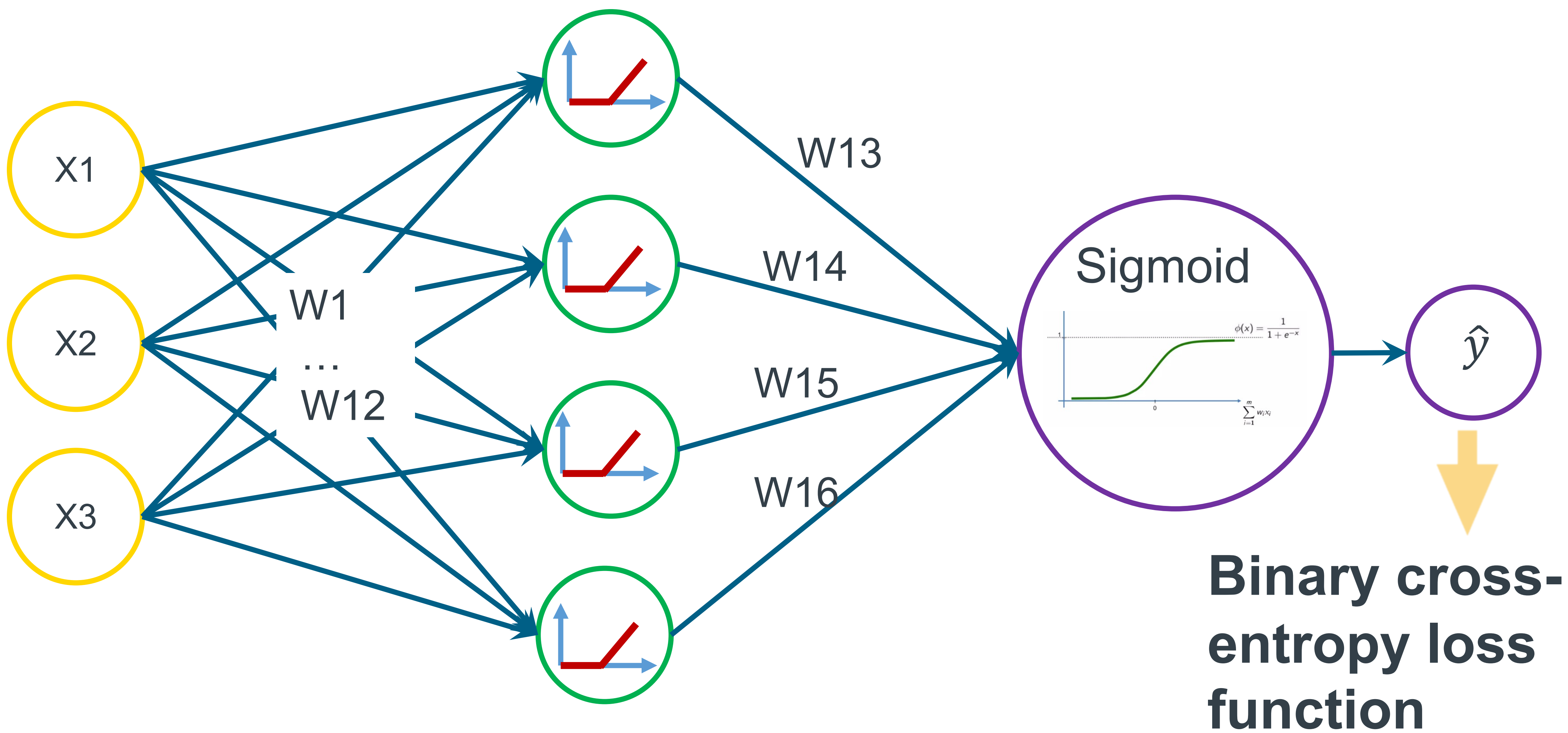
Some Basic Concepts of Neural Network

Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	1
2	11	8	77%	1
…	…	…	…	…
80	6	12	70%	0
81	11	8	69%	0
82	8	10	85%	1
…	…	…	…	…
100	6	6	66%	0

Training (rows 1–80)

Testing (rows 81–100)

1: Pass 0: Not Pass



Binary cross-entropy loss function

Exam (Actual)	Log	Corrected probability	Predicted probability (p_i)
1	-0.04	0.92	0.92
1	-0.36	0.44	0.44
0	-0.40	0.40	0.60
…	…	…	…
0	-0.11	0.78	0.22

Trained Model

$$\text{Log loss} = \frac{1}{N}\sum_{i=1}^{N} -\left(y_i * \log(p_i) + (1-y_i) * \log(1-p_i)\right)$$

$$Loss = -\frac{1}{80}(-0.04 - 0.36 - 0.40 - \cdots - 0.11)$$

Row ID	Study Hours	Sleep Hours	Quiz	Exam (Actual)
1	12	7	79%	1
2	11	8	77%	1
…	…	…	…	…
80	6	12	70%	0
81	11	8	69%	0
82	8	10	85%	1
…	…	…	…	…
100	6	6	66%	0

Training (rows 1–80)

Testing (rows 81–100)

1: Pass 0: Not Pass

Trained Model

Row ID	Exam (Actual)	Predicted Probability for Pass
81	0	0.48
82	1	0.88
…	…	…
100	0	0.53

Predictive Performance Evaluation

➢ **Confusion matrix**

Determine a probability threshold (typically 0.5)

Actual \ Predicted	Not Pass	Pass
Not Pass	True Negative (TN)	False Positive (FP)
Pass	False Negative (FN)	True Positive (TP)

$$Precision = \frac{TP}{TP + FP}$$

$$True\ Positive\ Rate\ (TPR) = \frac{TP}{TP + FN}$$

$$Recall = TPR = \frac{TP}{TP + FN}$$

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN}$$

$$F1\ Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

(harmonic mean of precision and recall, range from 0 to 1 where 1 is the best)

Predictive Performance Evaluation

- **Confusion matrix**

Determine a probability threshold (typically 0.5)

Actual \ Predicted	Not Pass	Pass
Not Pass	True Negative (TN)	False Positive (FP)
Pass	False Negative (FN)	True Positive (TP)

$$Precision = \frac{TP}{TP + FP}$$

$$True\ Positive\ Rate\ (TPR) = \frac{TP}{TP + FN}$$

$$Recall = TPR = \frac{TP}{TP + FN}$$

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN}$$

$$F1\ Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

(harmonic mean of precision and recall, range from 0 to 1 where 1 is the best)

Row ID	Exam (Actual)	Predicted Probability for Pass
81	0	0.48
82	1	0.88
83	1	0.49
84	0	0.34
85	1	0.43
86	0	0.2
87	0	0.1
88	1	0.65
89	1	0.9
90	1	0.9

Predictive Performance Evaluation

➢ **Confusion matrix**

Determine a probability threshold (typically 0.5)

		Predicted: Not Pass	Predicted: Pass
Actual	Not Pass	True Negative (TN)	False Positive (FP)
	Pass	False Negative (FN)	True Positive (TP)

$$Precision = \frac{TP}{TP + FP}$$

$$True\ Positive\ Rate\ (TPR) = \frac{TP}{TP + FN}$$

$$Recall = TPR = \frac{TP}{TP + FN}$$

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN}$$

$$F1\ Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

(harmonic mean of precision and recall, range from 0 to 1 where 1 is the best)

Row ID	Exam (Actual)	Predicted Probability for Pass	
81	0	0.48	✓
82	1	0.88	✓
83	1	0.39	✗
84	0	0.15	✓
85	1	0.43	✗
86	0	0.21	✓
87	0	0.71	✗
88	1	0.65	✓
89	1	0.9	✓
90	1	0.9	✓

		Predicted: Not Pass	Predicted: Pass
Actual	Not Pass	TN=3	FP=1
	Pass	FN=2	TP=4

$$Precision = \frac{4}{5} \qquad TPR = Recall = \frac{2}{3}$$

$$FPR = \frac{1}{4} \qquad F1\ Score = \frac{8}{11}$$

Predictive Performance Evaluation

➢ Confusion matrix

Determine a probability threshold (typically 0.5)

		Predicted: Not Pass	Predicted: Pass
Actual	Not Pass	True Negative (TN)	False Positive (FP)
Actual	Pass	False Negative (FN)	True Positive (TP)

$$Precision = \frac{TP}{TP + FP}$$

$$True\ Positive\ Rate\ (TPR) = \frac{TP}{TP + FN}$$

$$Recall = TPR = \frac{TP}{TP + FN}$$

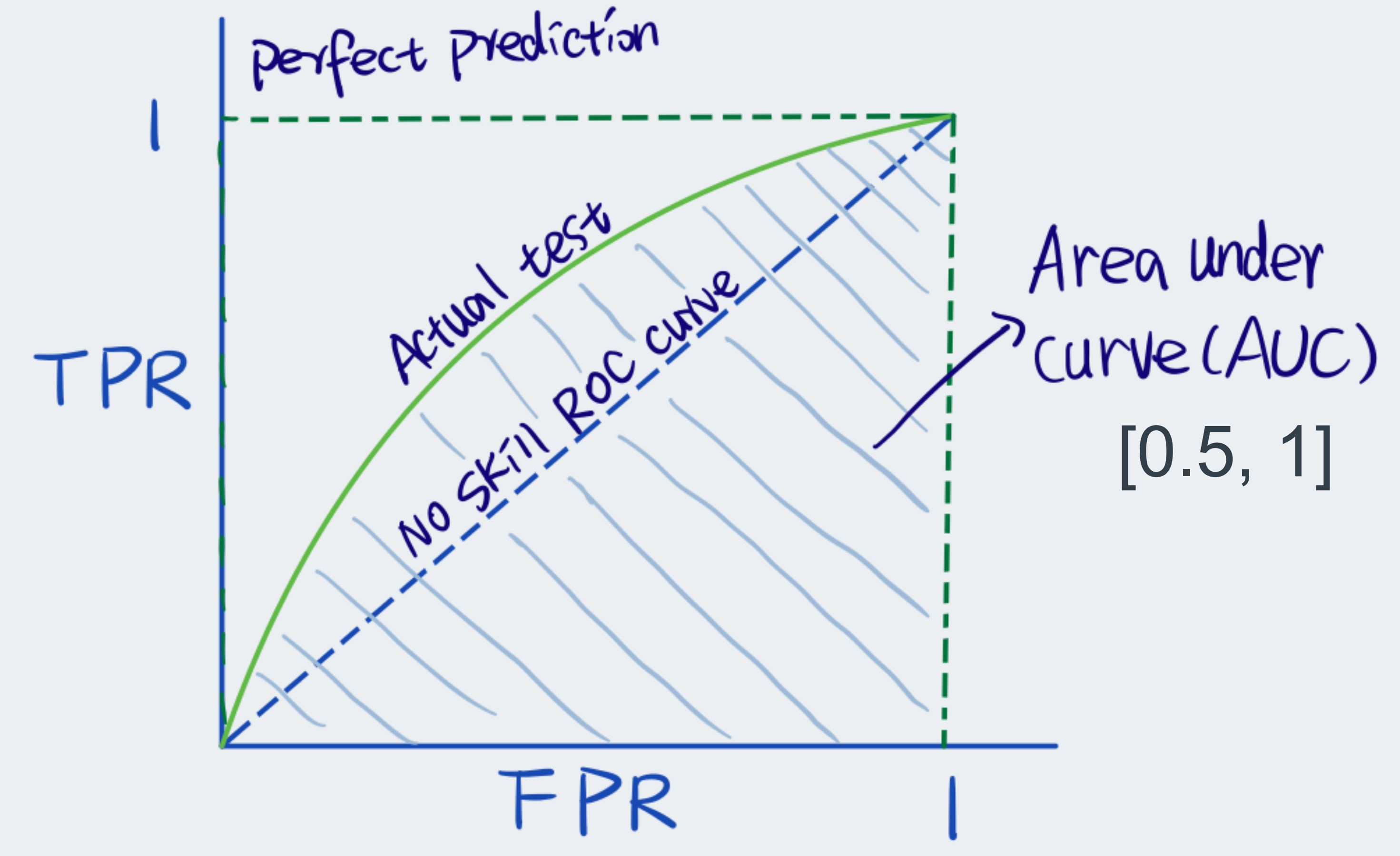
$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN}$$

$$F1\ Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

(harmonic mean of precision and recall, range from 0 to 1 where 1 is the best)

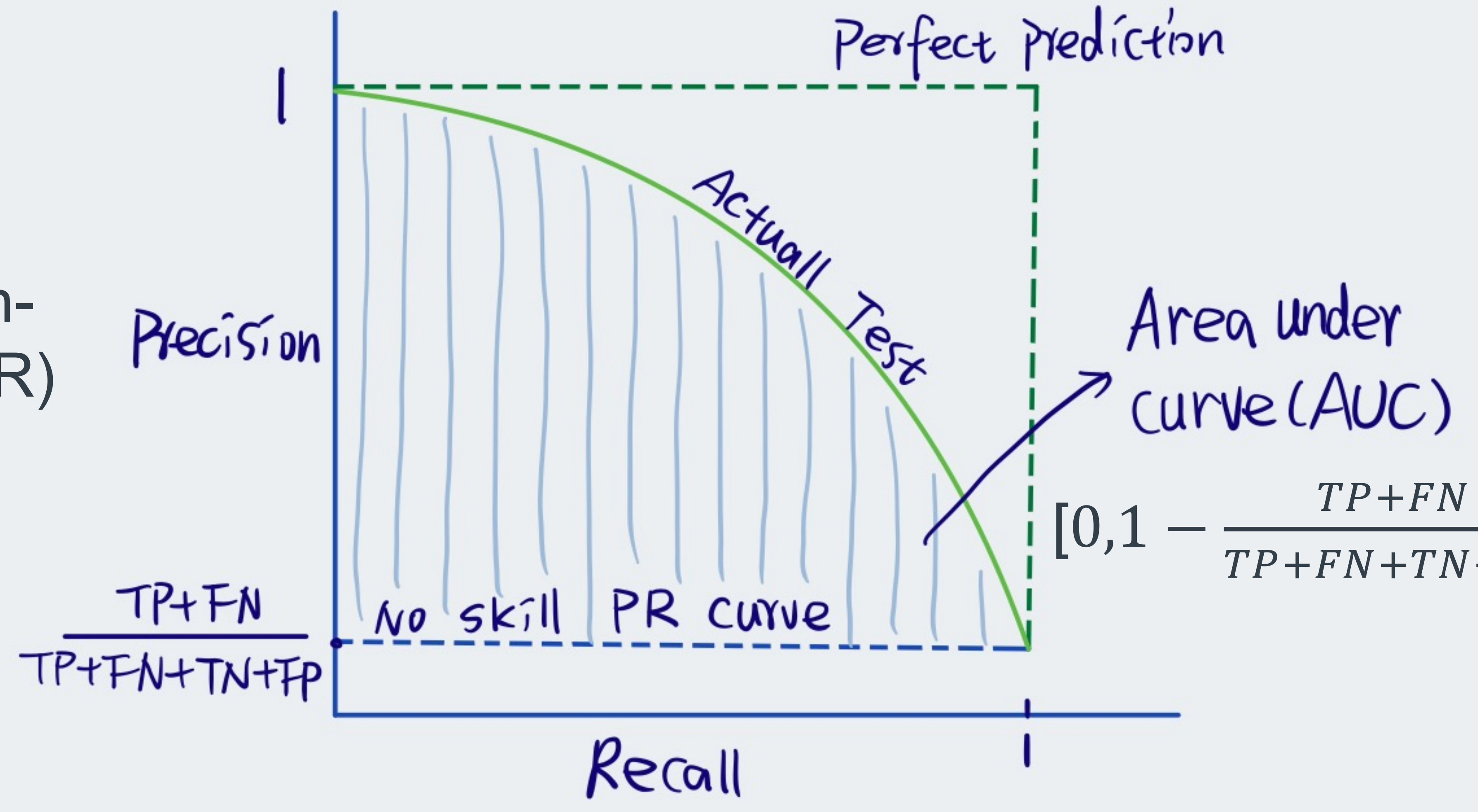
➢ Aggregated measurement

Receiver Operating Characteristic (ROC) curve



[0.5, 1]

Precision-Recall (PR) curve

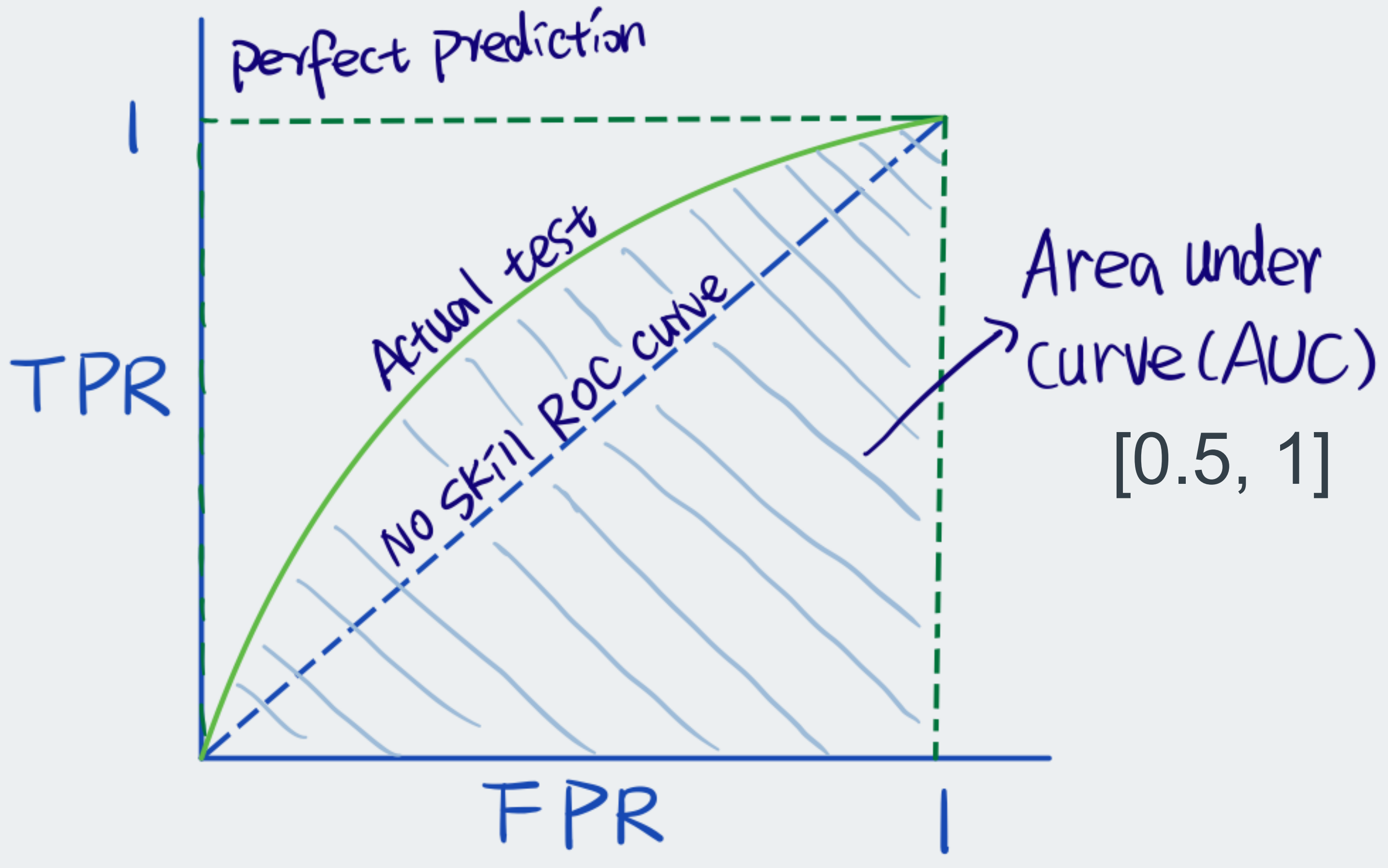


$[0,1 - \frac{TP+FN}{TP+FN+TN+FP}]$

Predictive Performance Evaluation

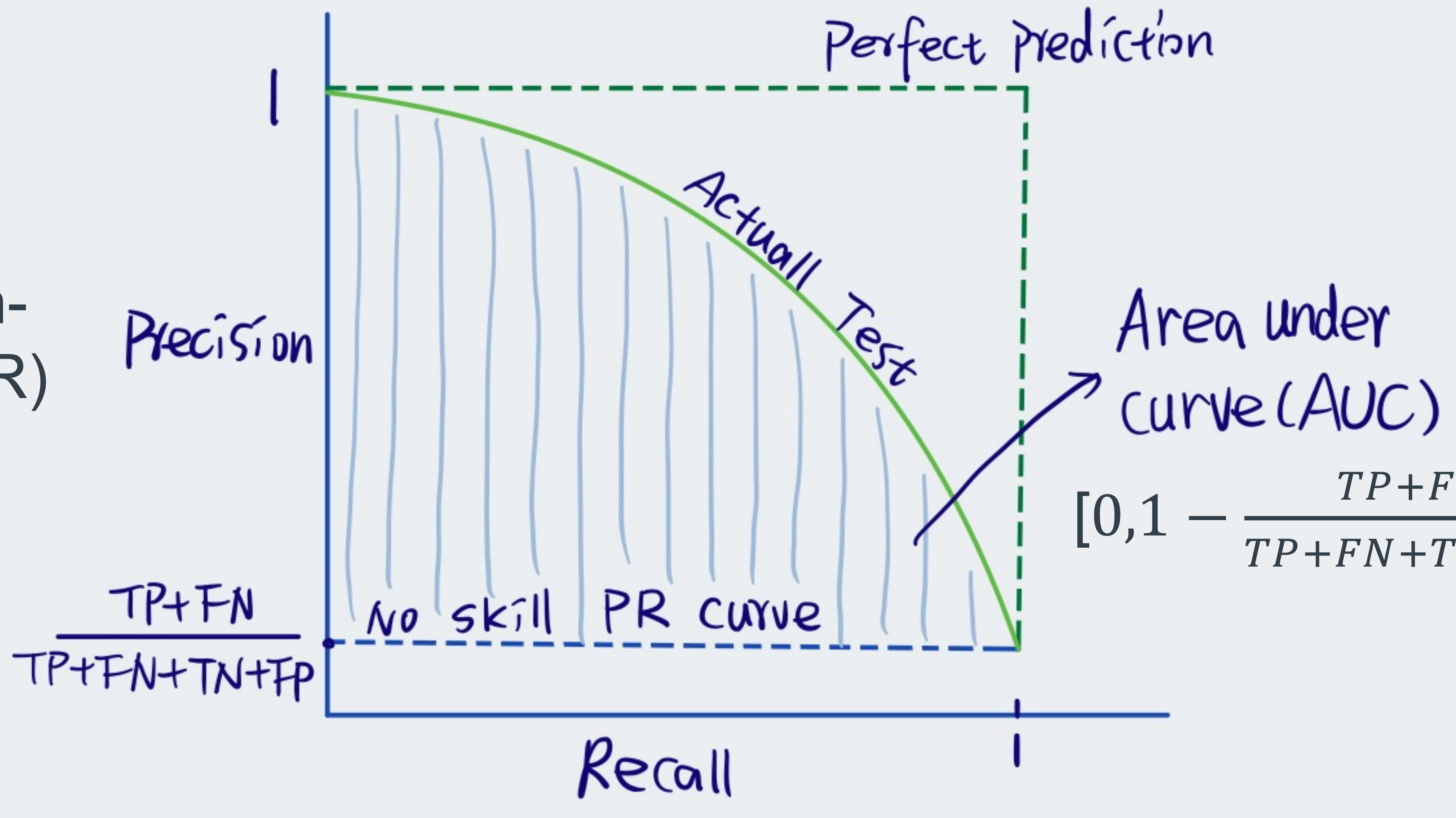
➢ **Aggregated measurement**

Receiver Operating Characteristic (ROC) curve



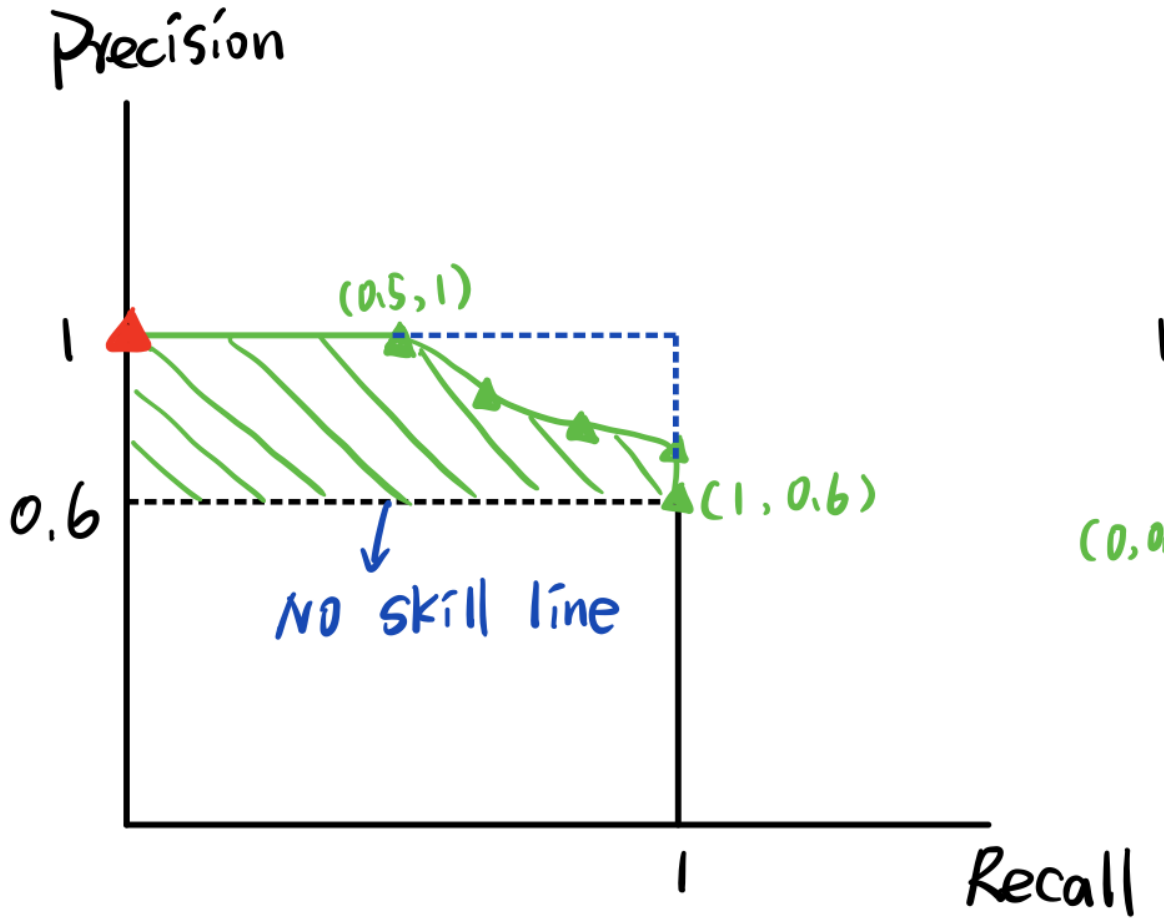
[0.5, 1]

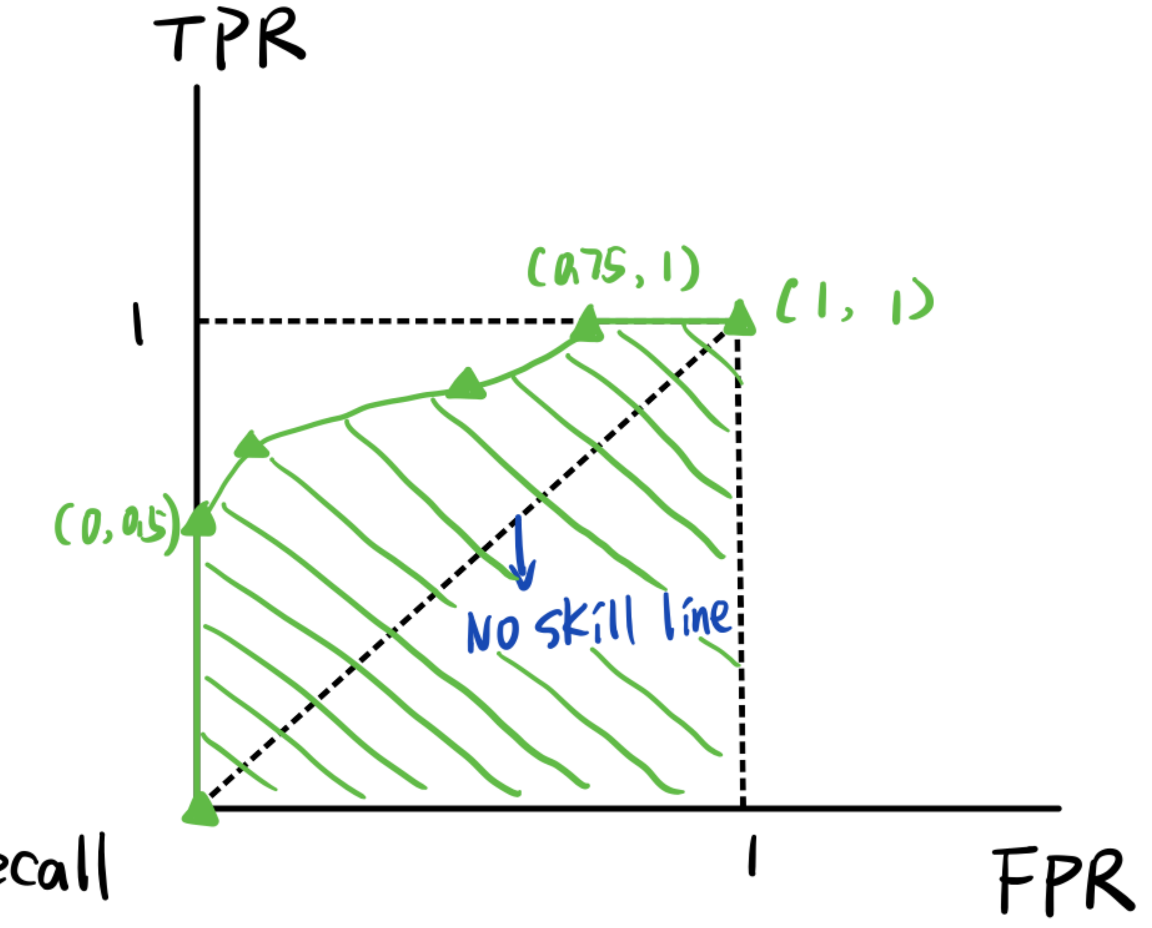
Precision-Recall (PR) curve



$$\left[0,1-\frac{TP+FN}{TP+FN+TN+FP}\right]$$

Row ID	Exam (Actual)	Predicted Probability for Pass	0	0.2	0.4	0.6	0.8	1
81	0	0.48	1	1	1	0	0	0
82	1	0.88	1	1	1	1	1	0
83	1	0.39	1	1	0	0	0	0
84	0	0.15	1	0	0	0	0	0
85	1	0.43	1	1	1	0	0	0
86	0	0.21	1	1	0	0	0	0
87	0	0.71	1	1	1	1	0	0
88	1	0.65	1	1	1	1	0	0
89	1	0.9	1	1	1	1	1	0
90	1	0.9	1	1	1	1	1	0
TPR			1	1	0.83	0.67	0.5	0
FPR			1	0.75	0.5	0.25	0	0
Precision			0.6	0.67	0.71	0.8	1	-
Recall			1	1	0.83	0.67	0.5	0





Outline

- **Section 1. Some Basic Concepts of Neural Network**
  - Neuron
  - Activation function
  - How do neural networks work
  - Artificial Neural Network (ANN) example
- **Section 2. Graph Neural Network (GNN)**
  - Graph data and graph tasks
  - How do GNNs work
  - GraphSAGE
- **Section 3. Application: Modeling Shared Mobility System Using GNN**

- Image as graph

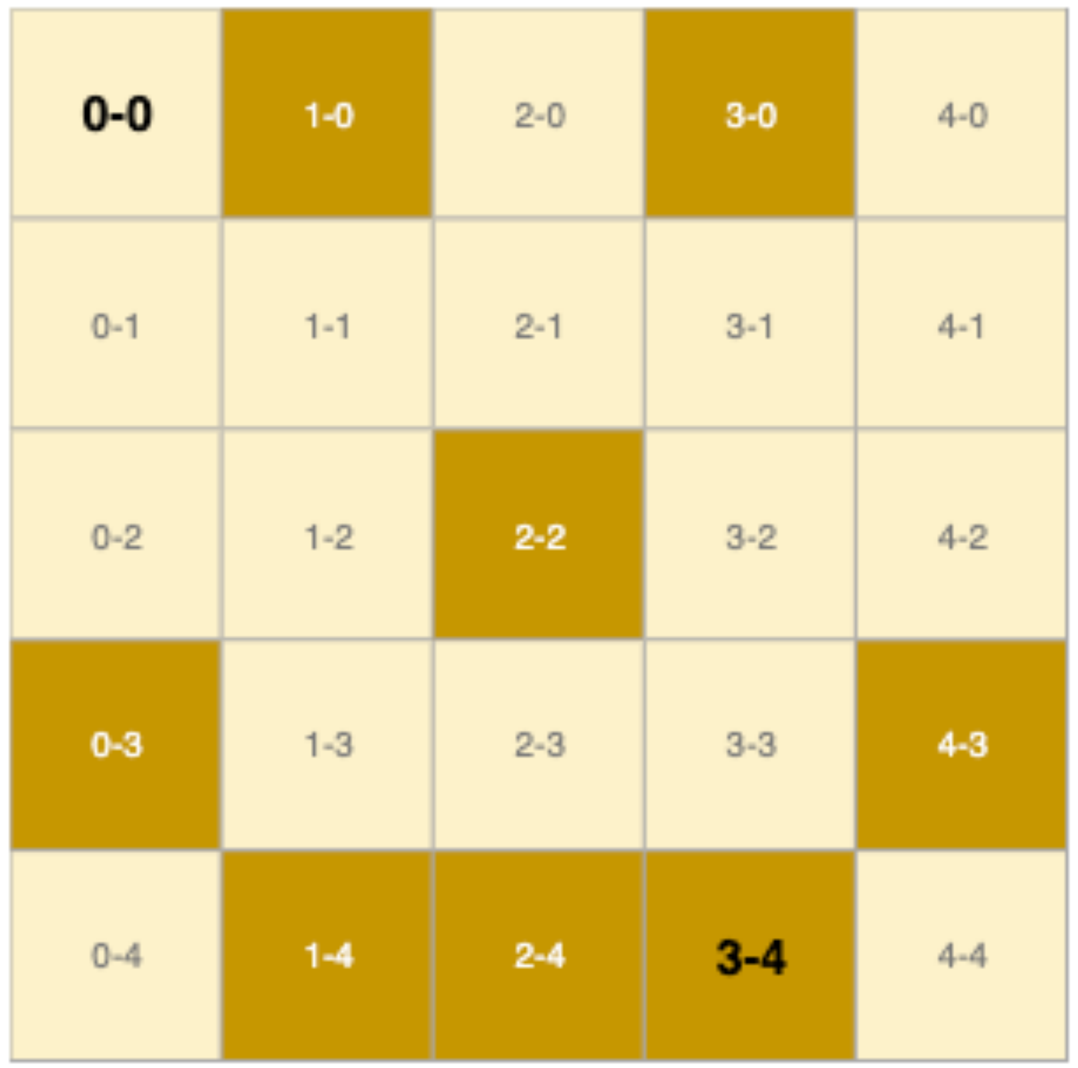
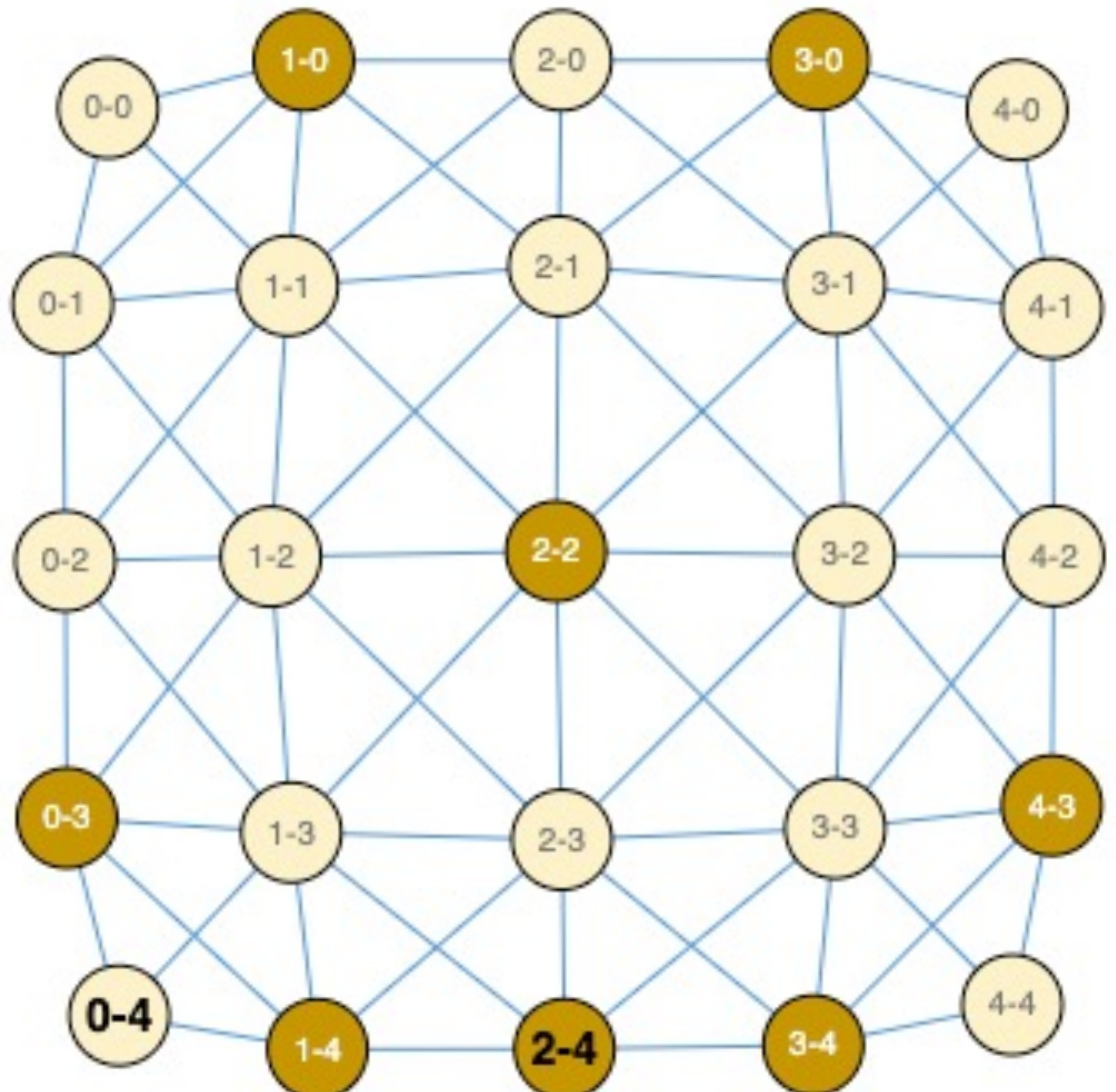
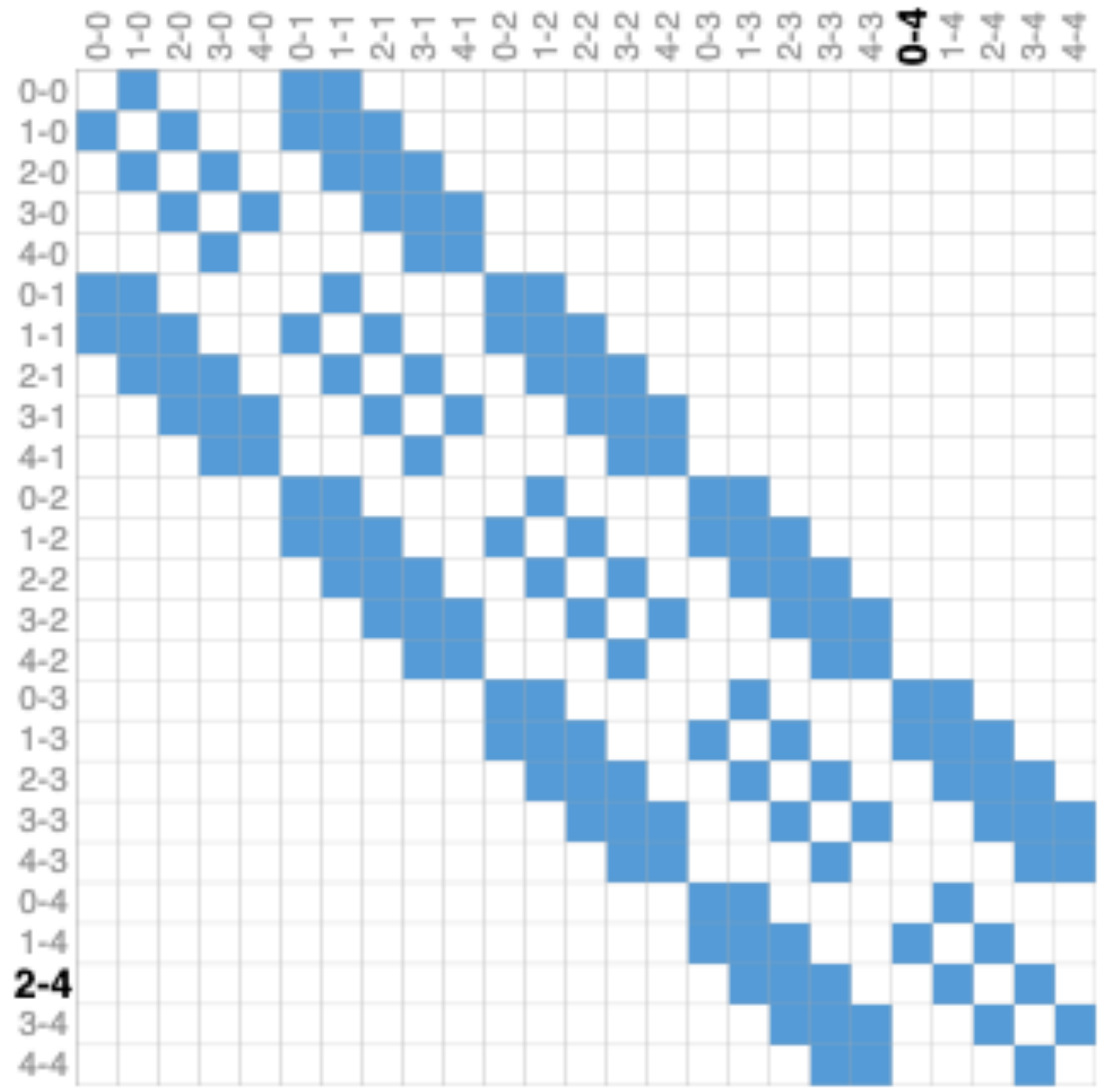


Image Pixels



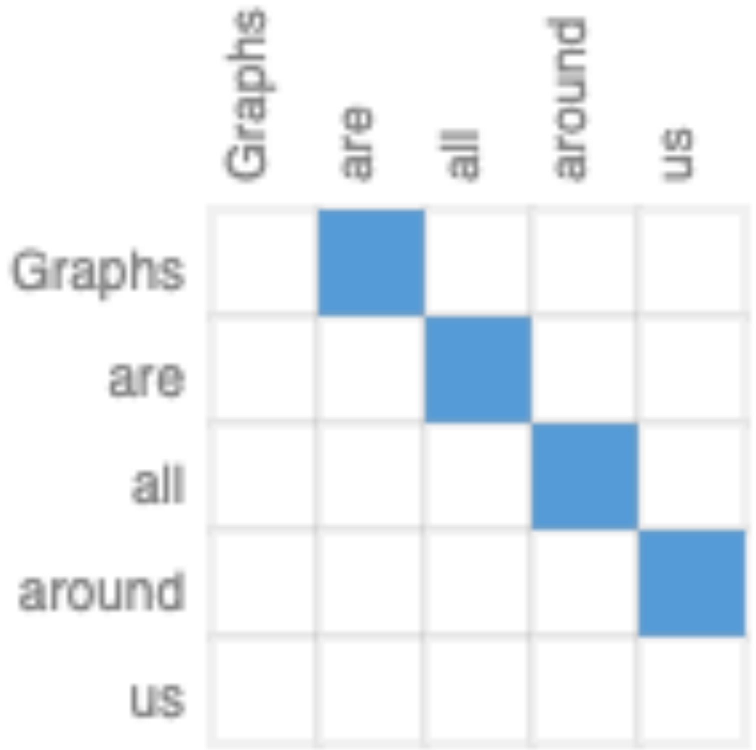
Graph



Adjacency Matrix

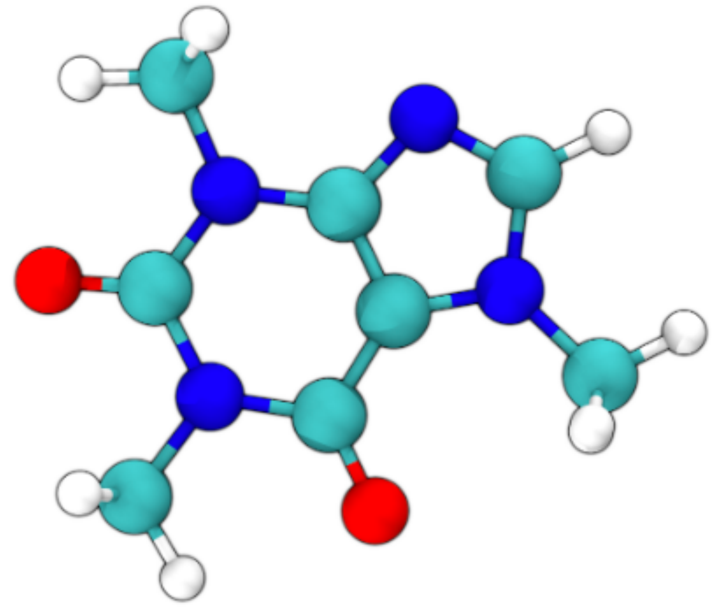
- Text as graph

Graphs are all around us

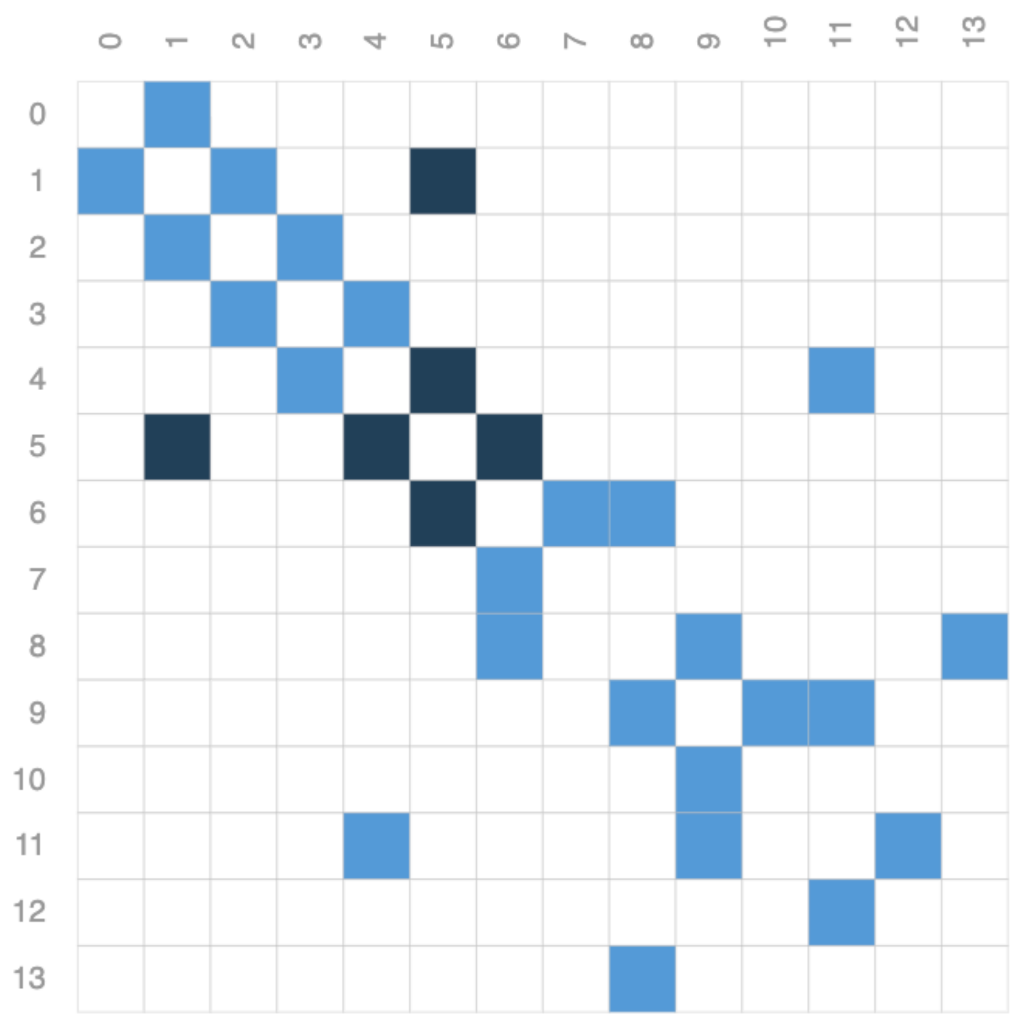


➢ Graph-valued data in the real world

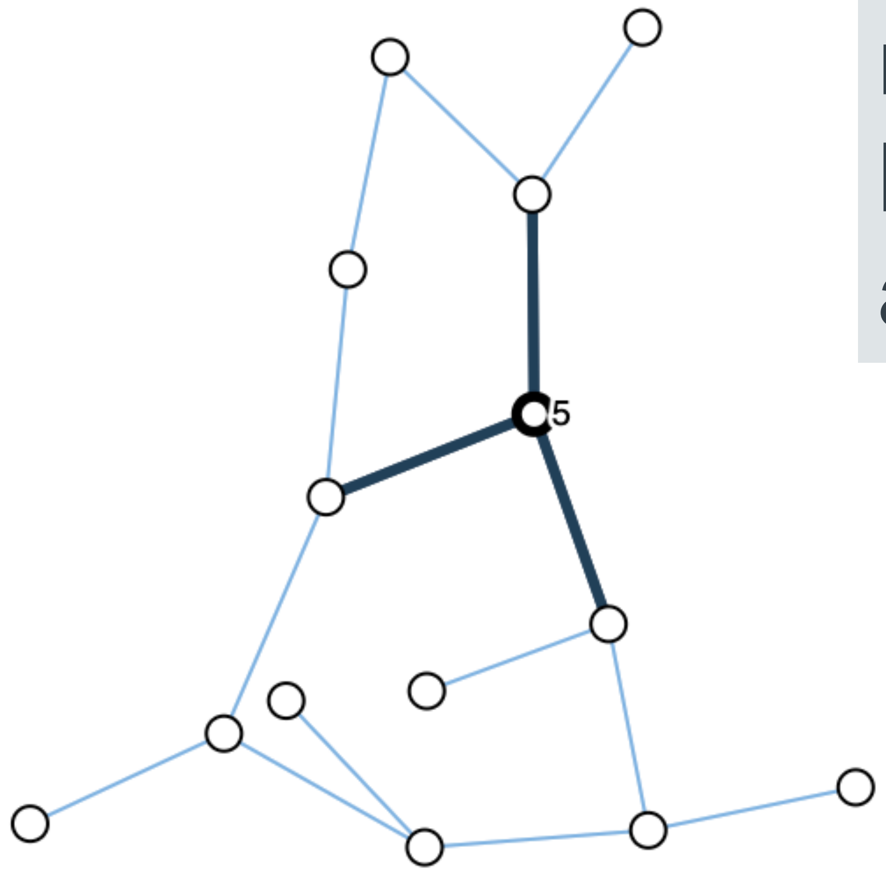
Unlike image and text data, complex networks do not have identical adjacency matrices



3d representation of the Caffeine molecule

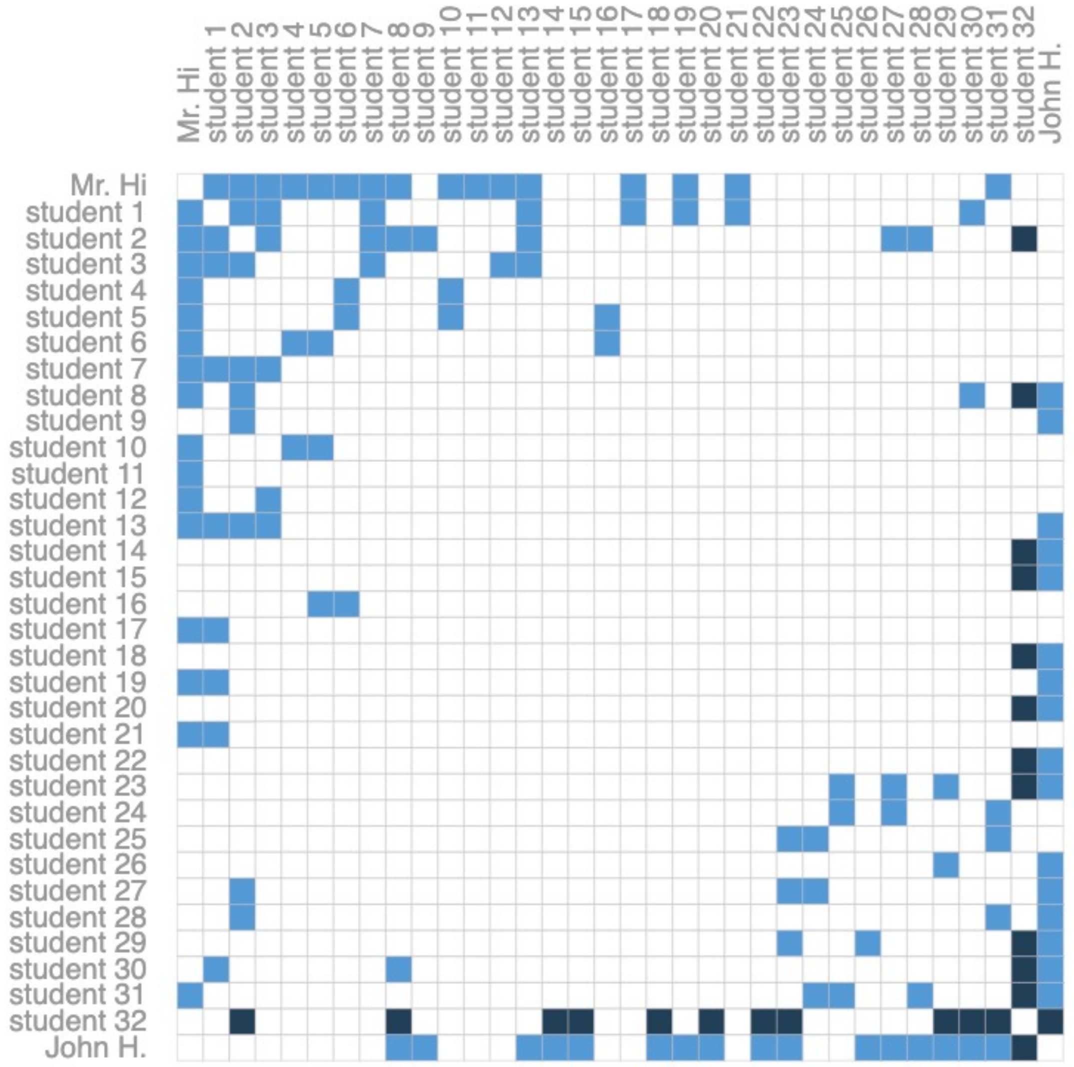


Adjacency matrix of the bonds in the molecule

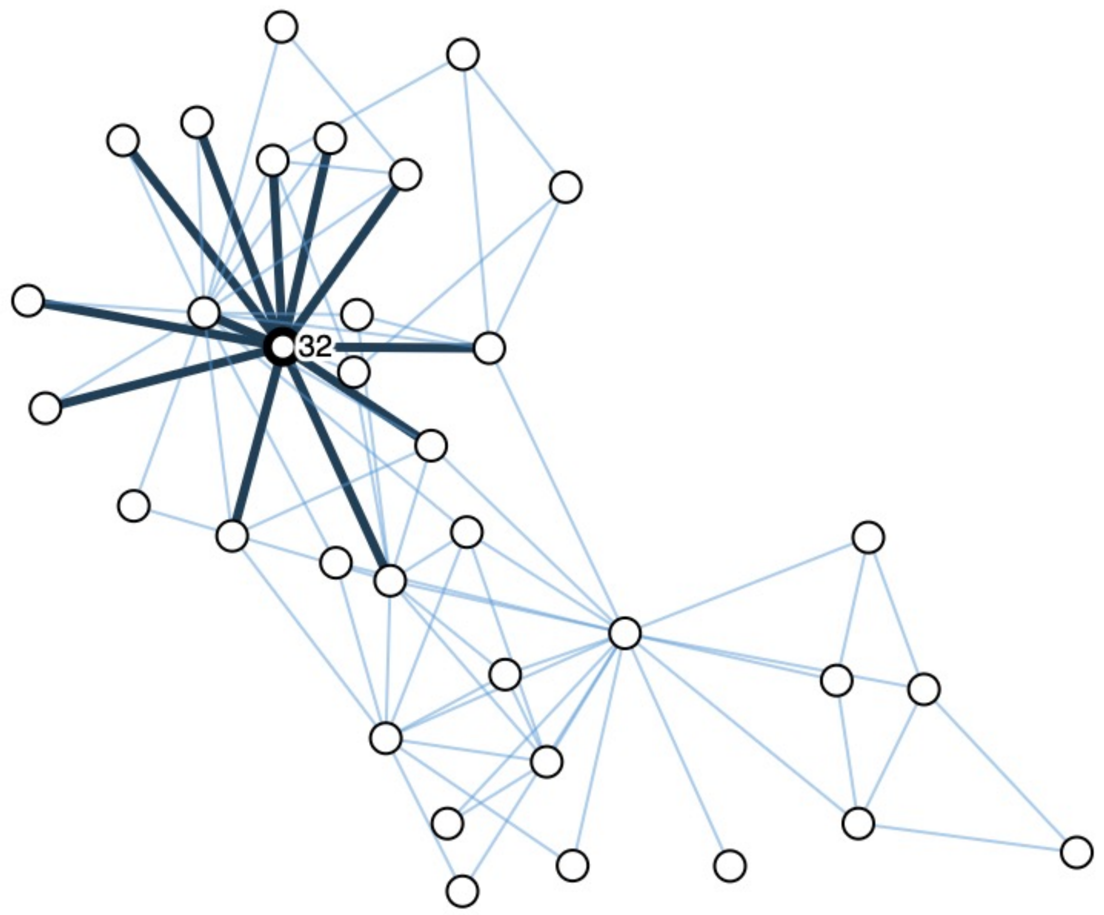


Graph representation of the molecule

Image of karate tournament



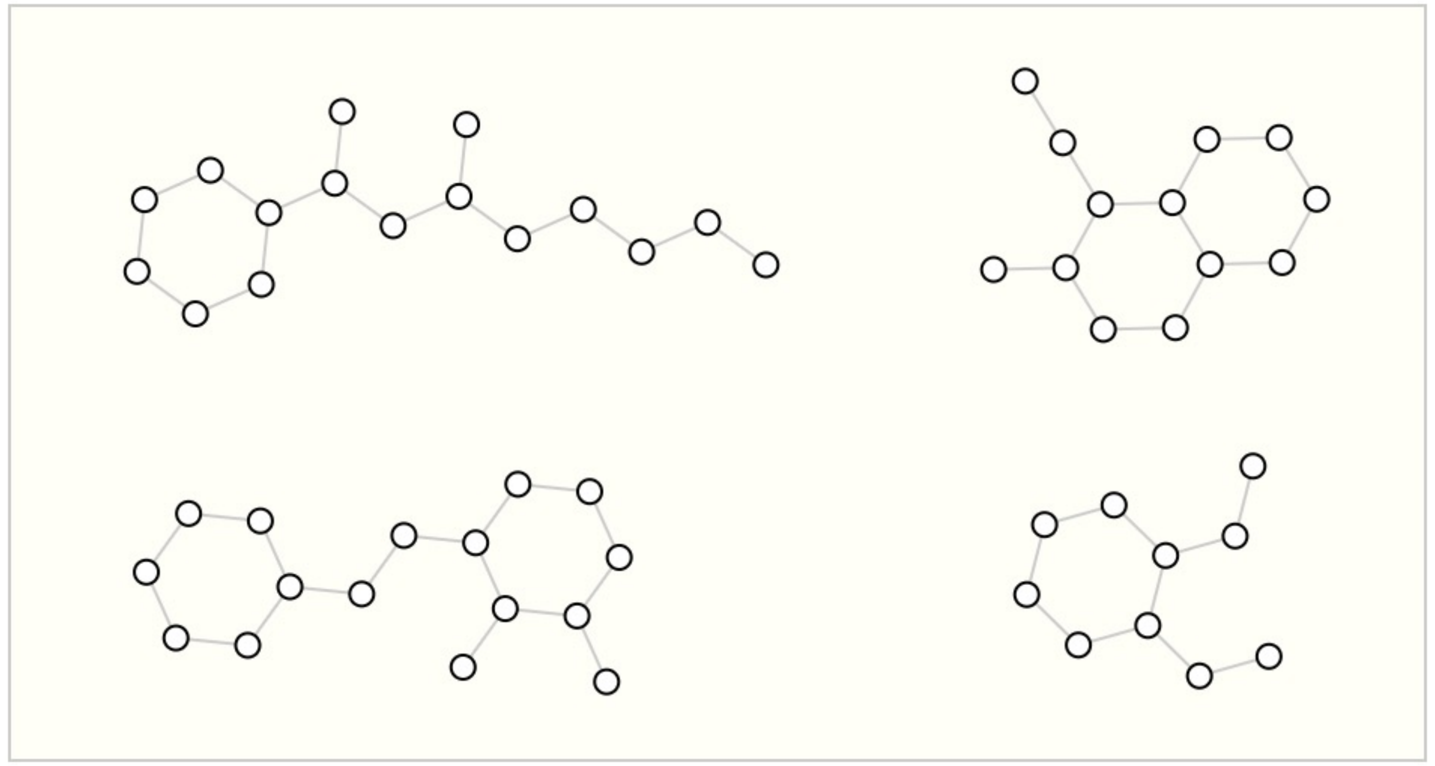
Adjacency matrix of the interaction between people in a karate club



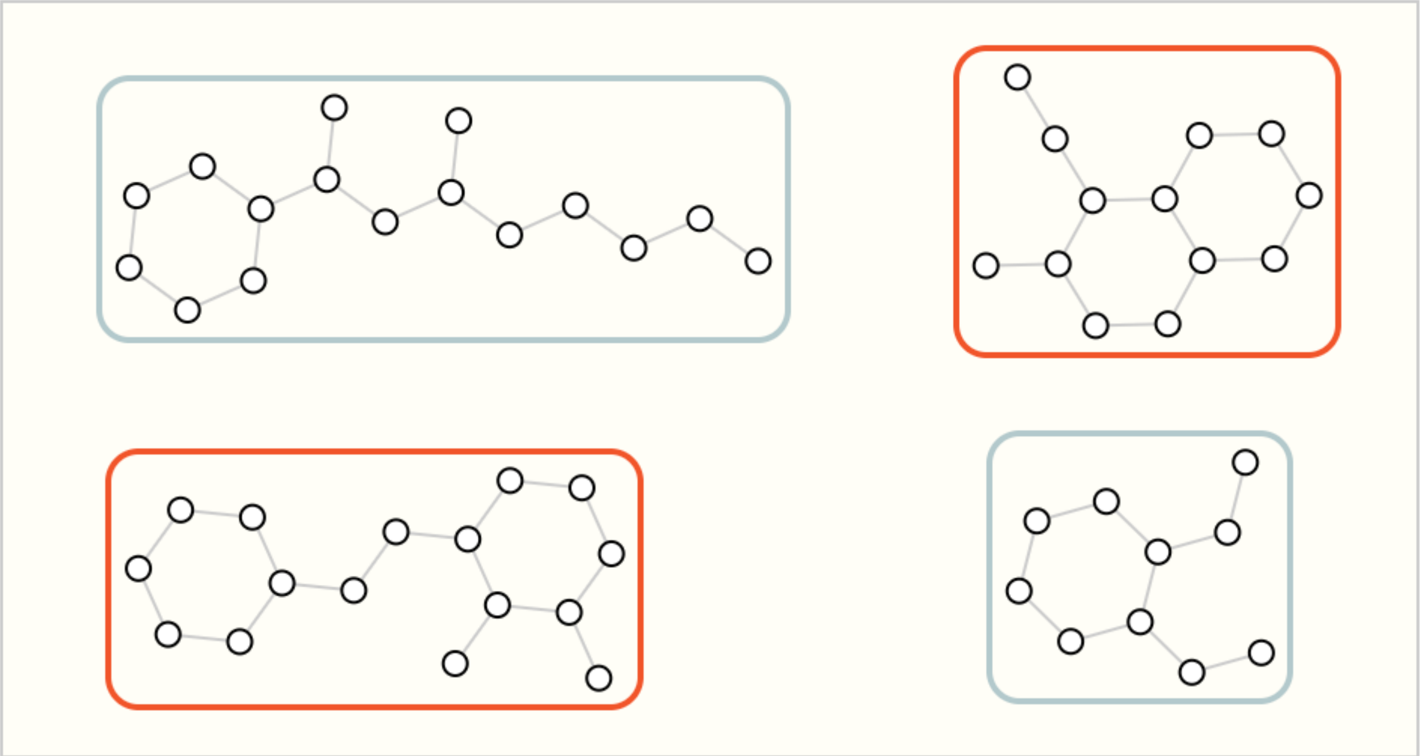
Graph representation of these interactions

➢ Graph-based tasks

Graph-level classification

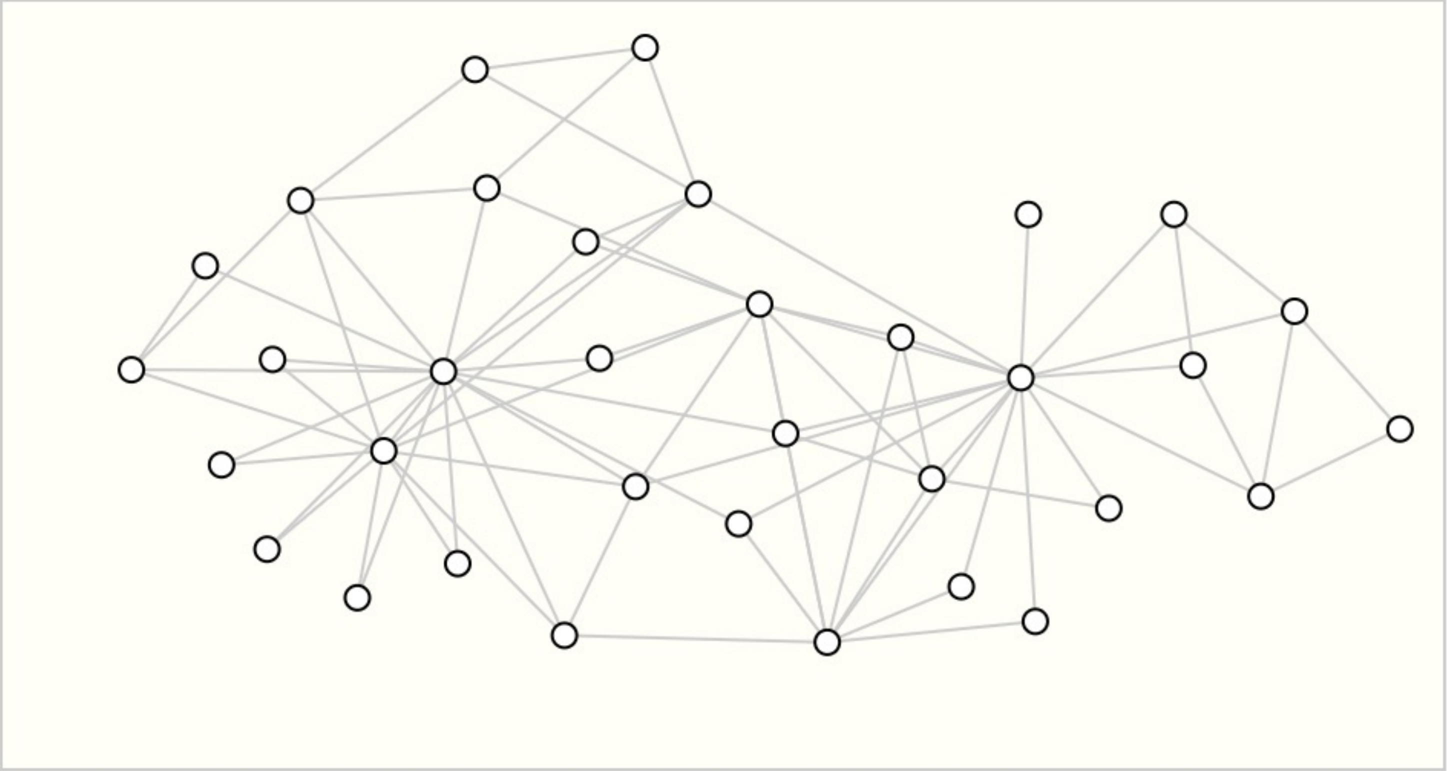


Input: graphs

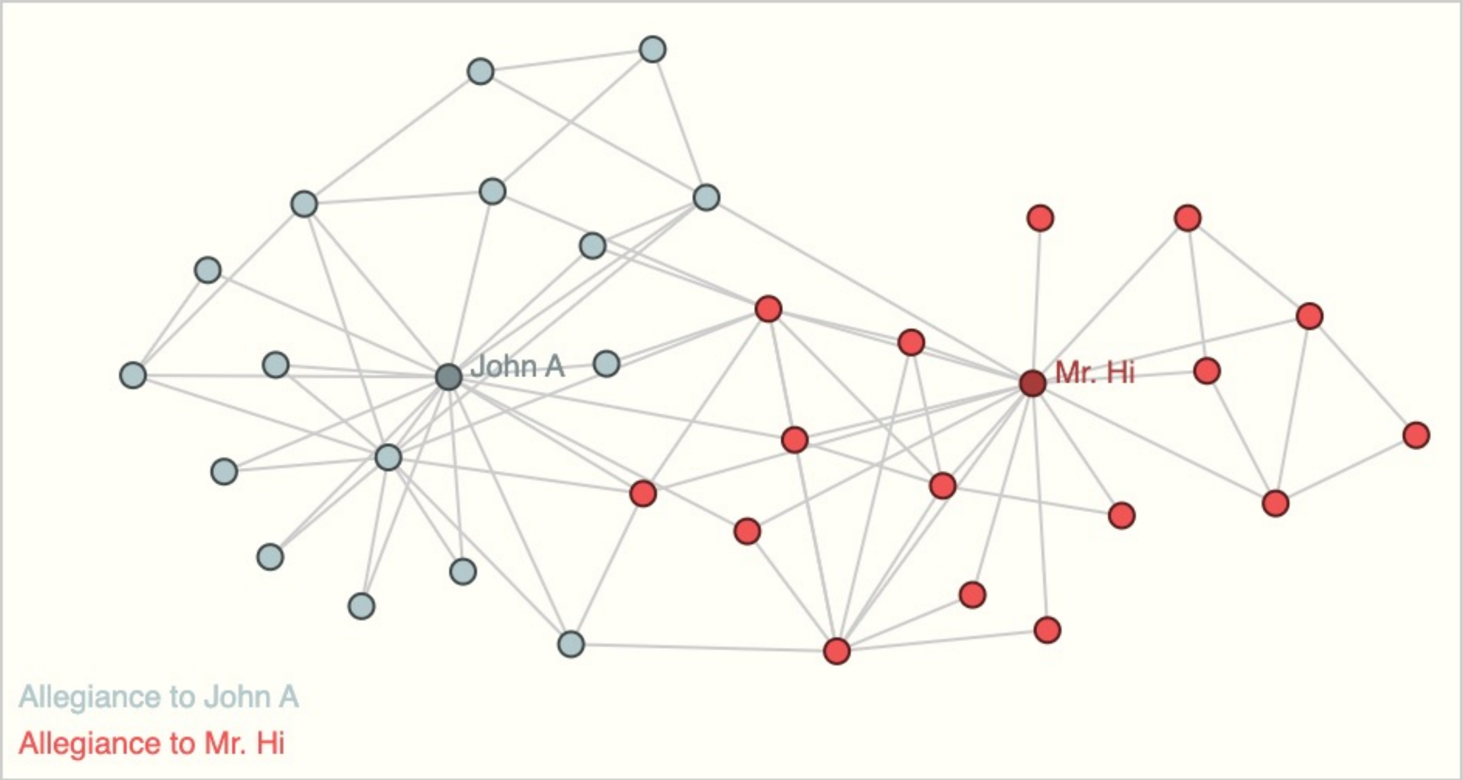


Output: labels for each graph
(e.g., graph with/without two rings)

Node-level clustering

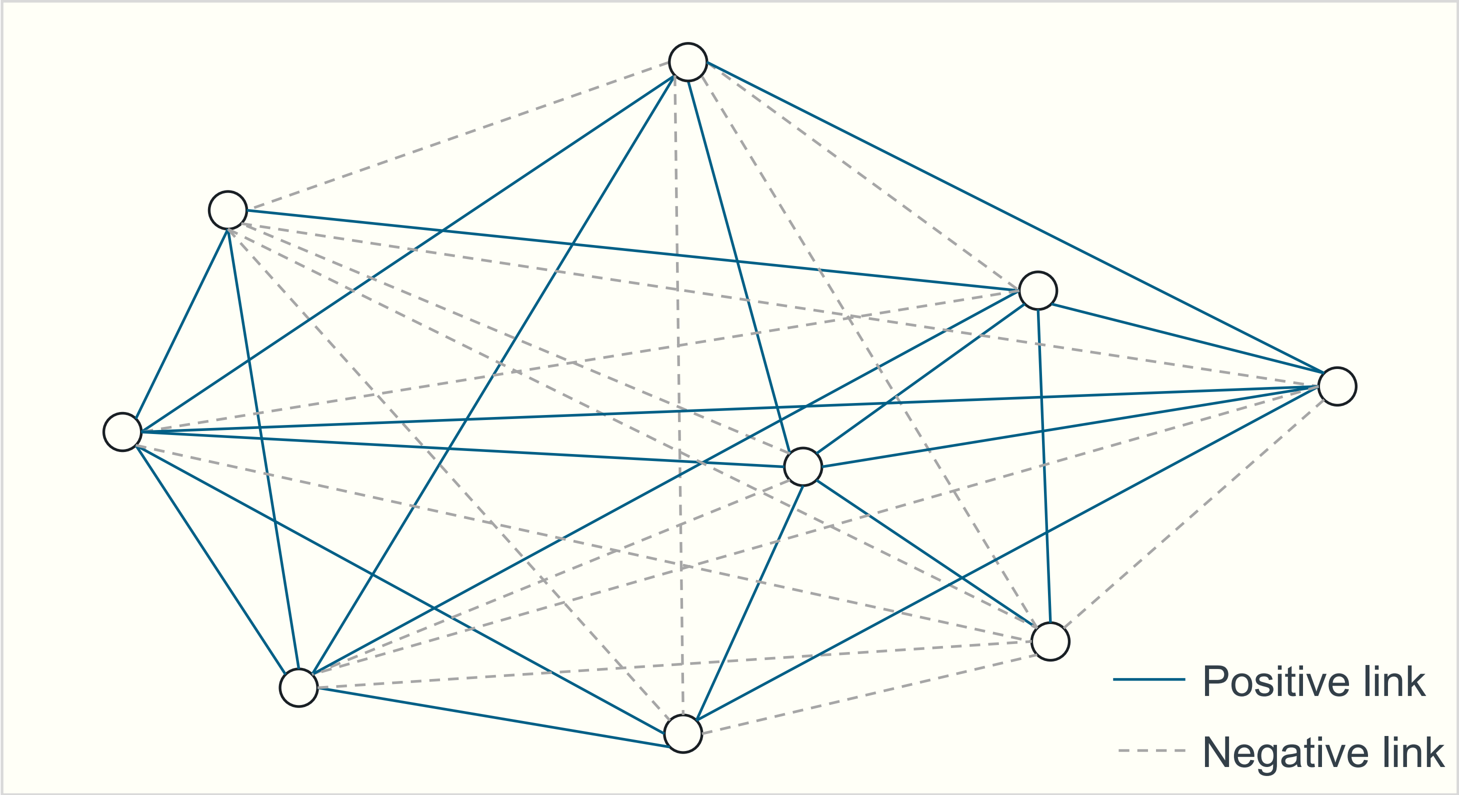


Input: graph with unlabeled nodes

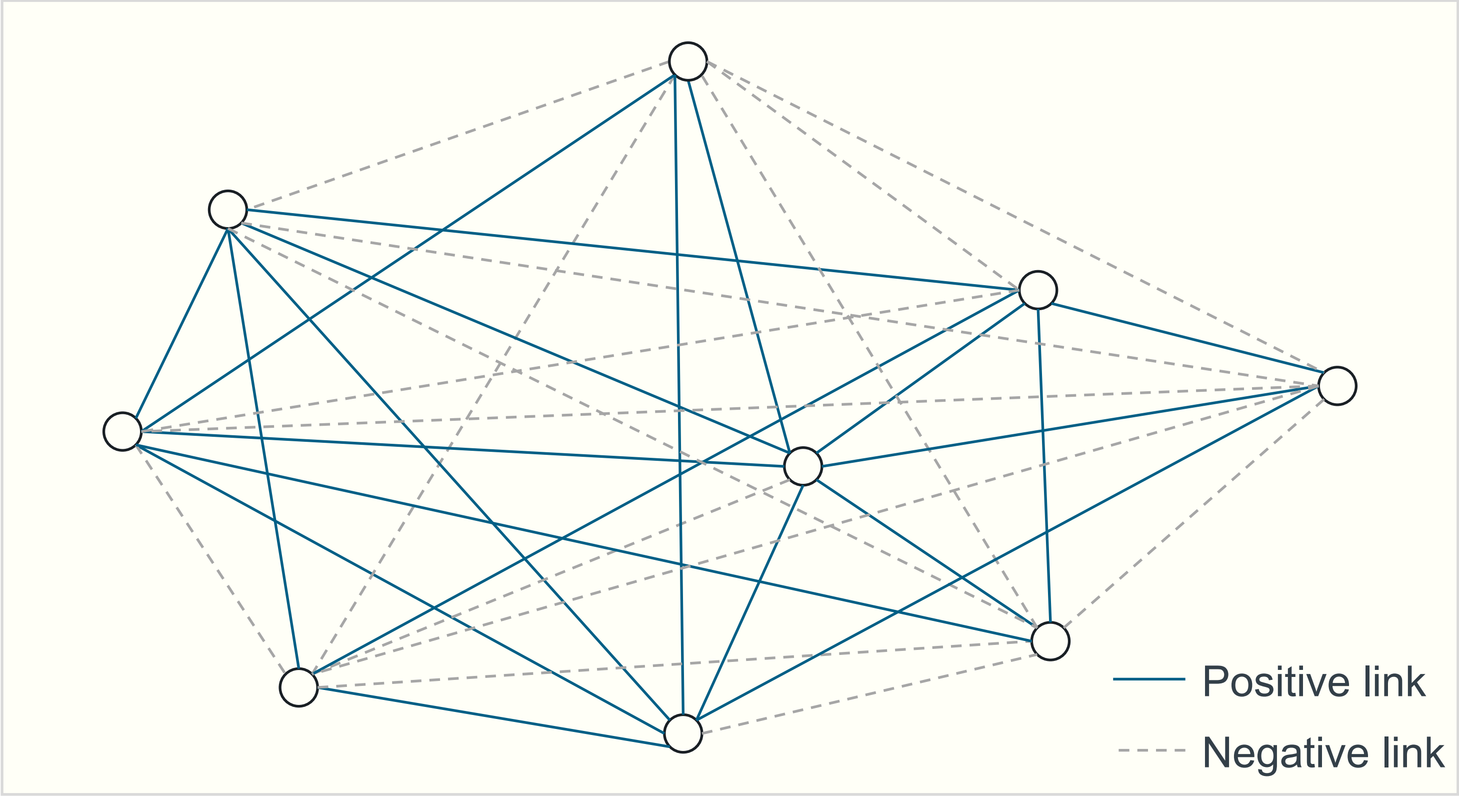


Output: graph with labeled nodes

Edge-level Prediction



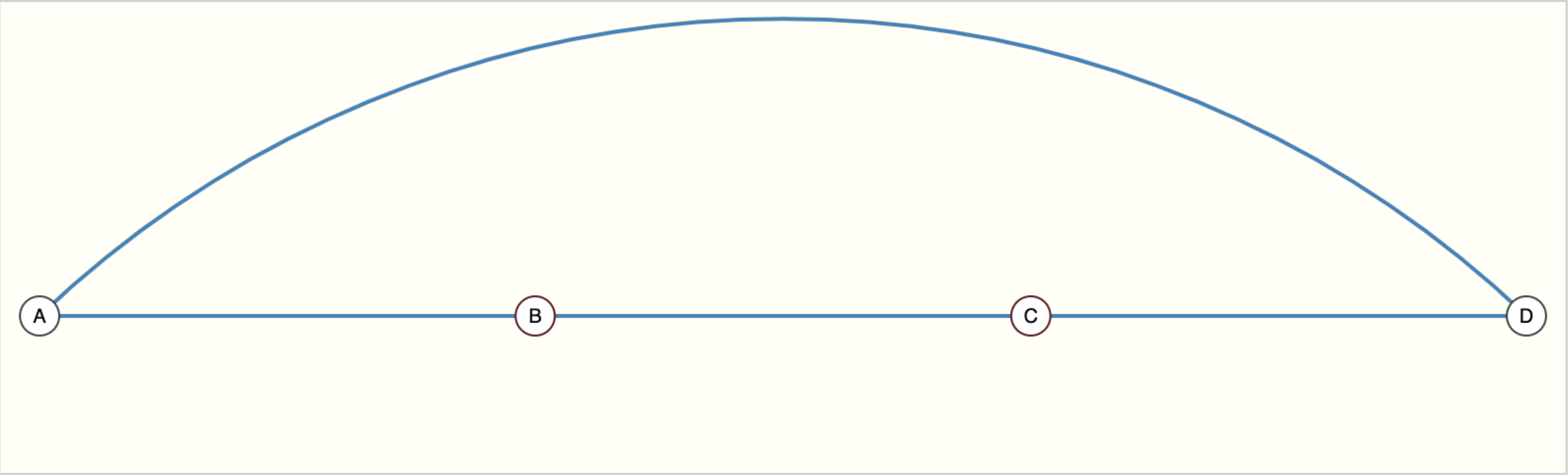
Input: period one graph

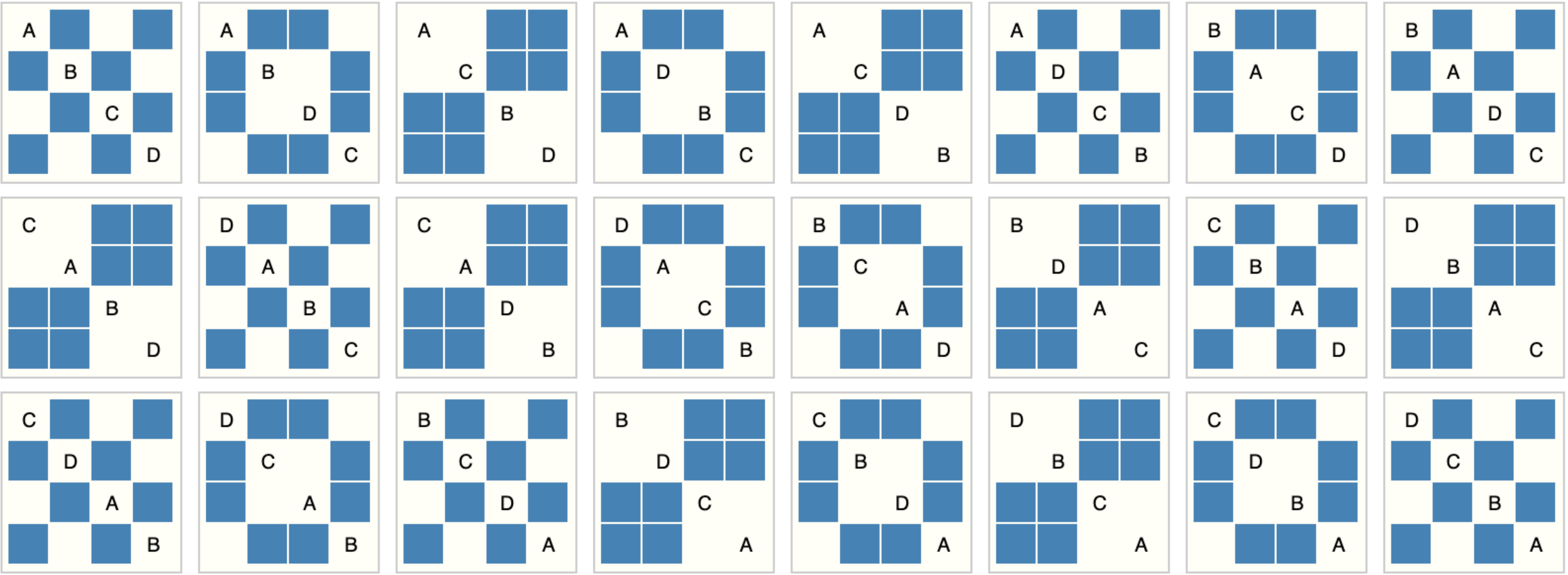


Output: period two graph

➢ The challenges of graph-based deep learning

- Different types of graph information need different approaches to represent and be compatible with neural networks
- The representation of a graph's connectivity is especially complicated
- Drawbacks of adjacency matrix representation:
  - ✓ Space-inefficient ($O(n_{nodes}^2)$)
  - ✓ One graph connection can be encoded by different adjacency matrices

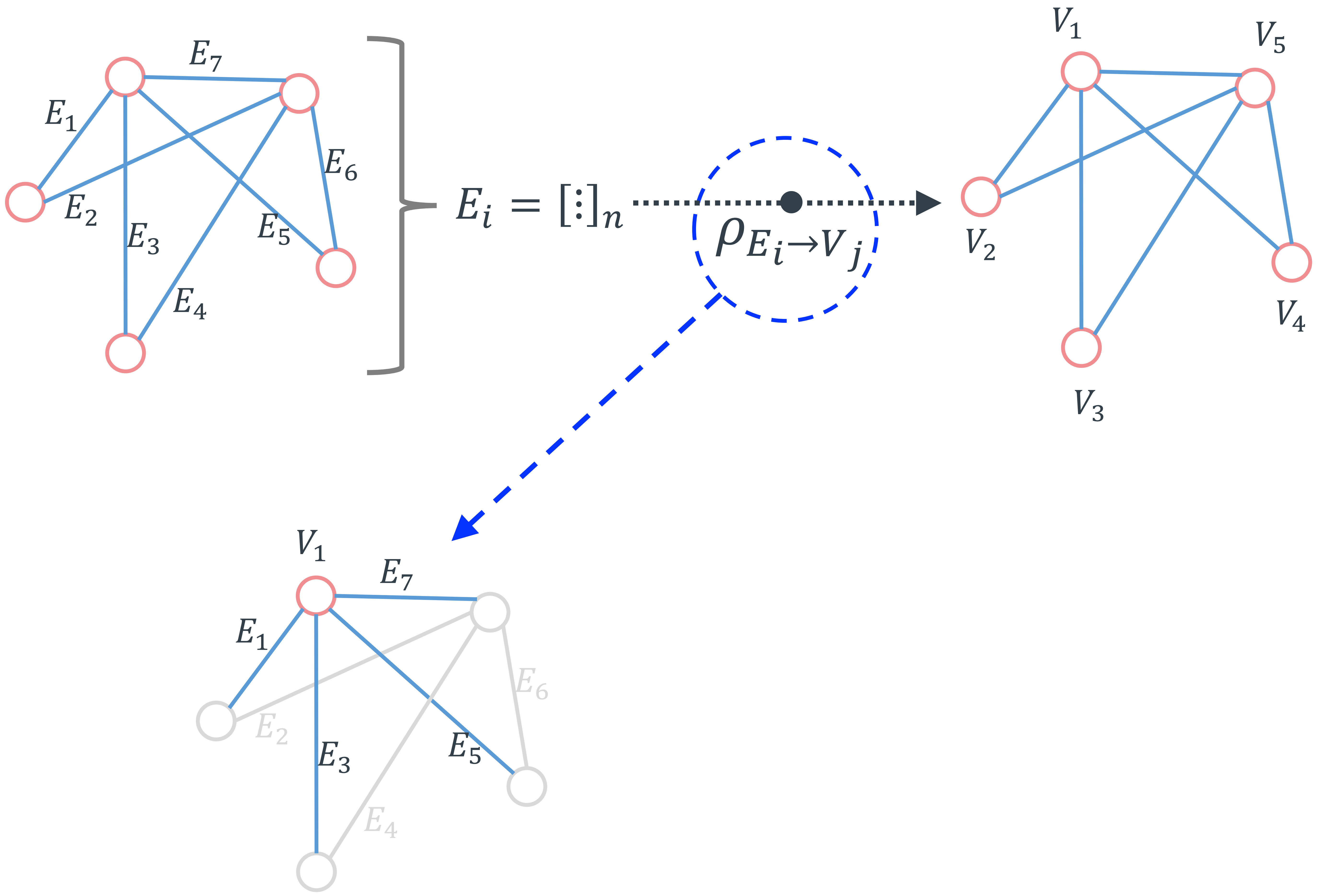




- Using adjacency list for the representation of graph connection is more memory-efficient ($O(n_{edges})$)

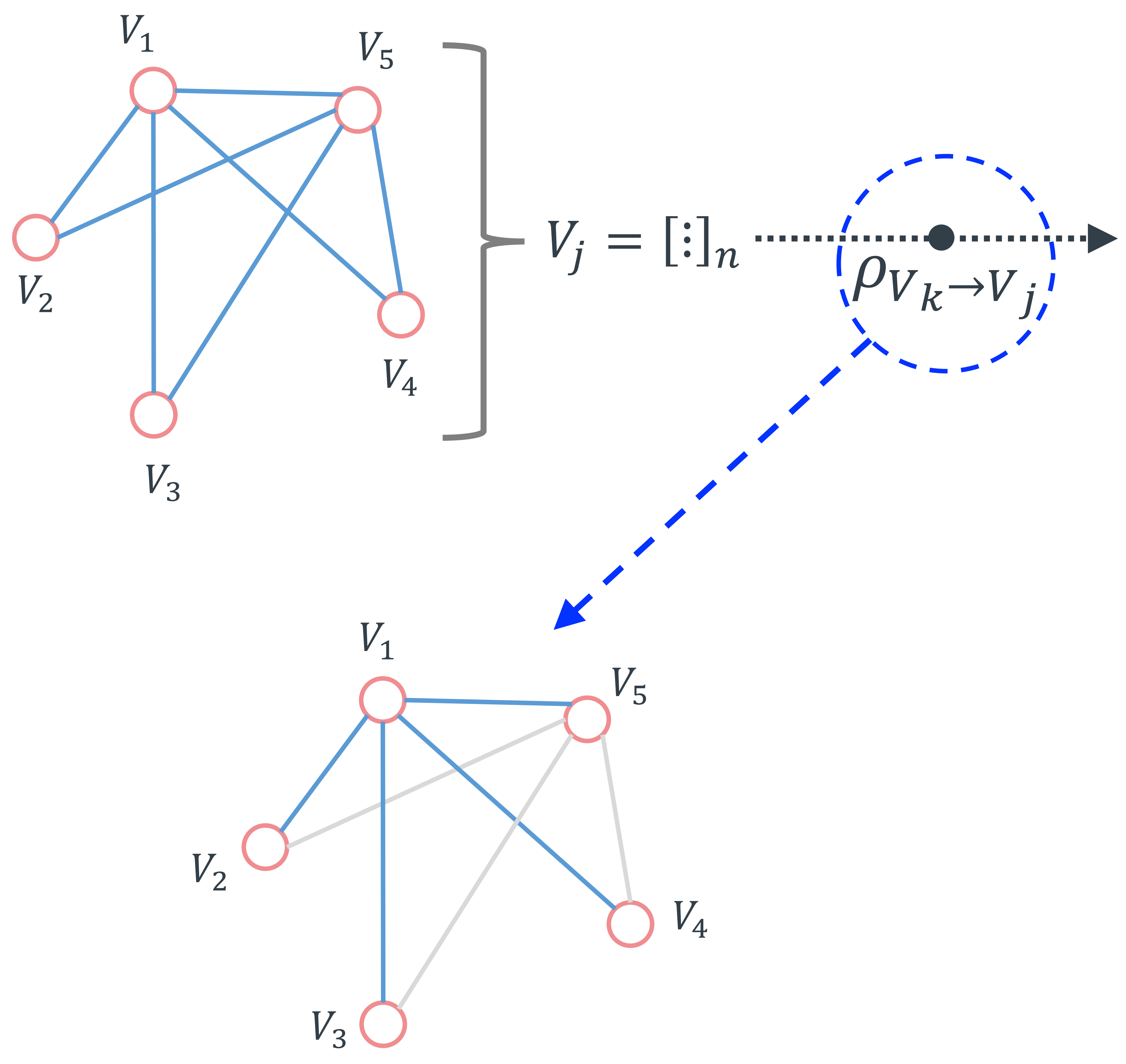
Adjacency List	
A	B
B	C
C	D
A	D

➢ Case 1: Edge to node embedding

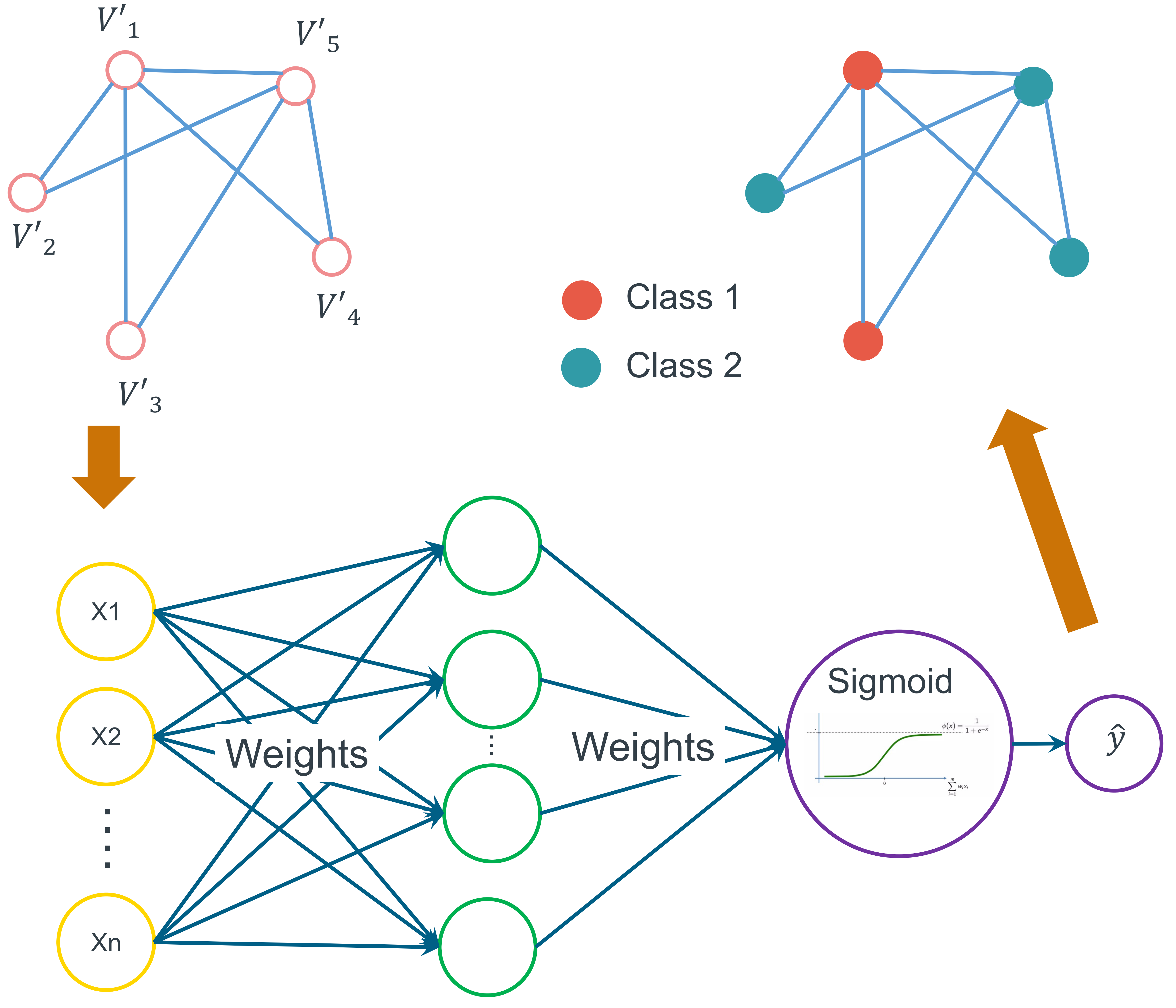


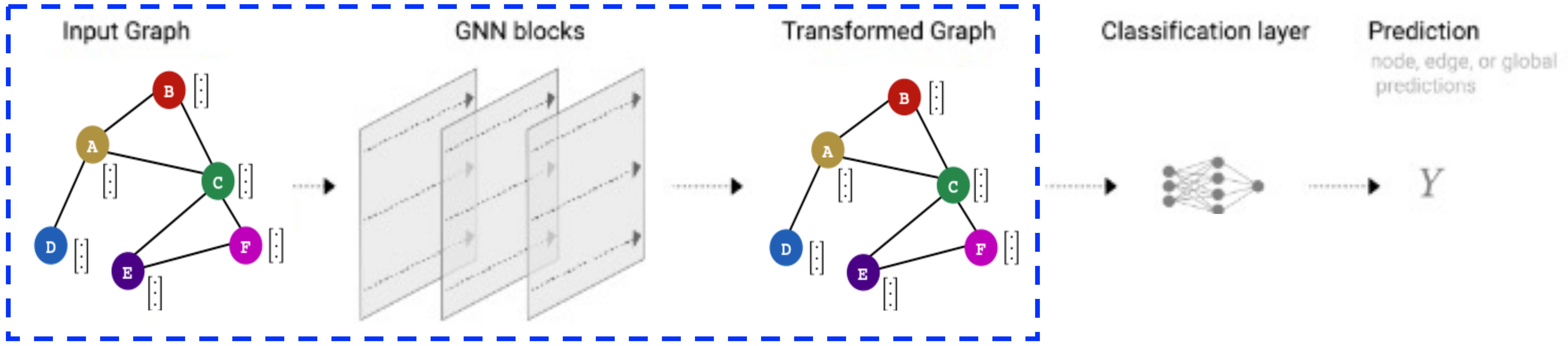
$V_1 = E_1 + E_3 + E_5 + E_7 = [\vdots]_n$

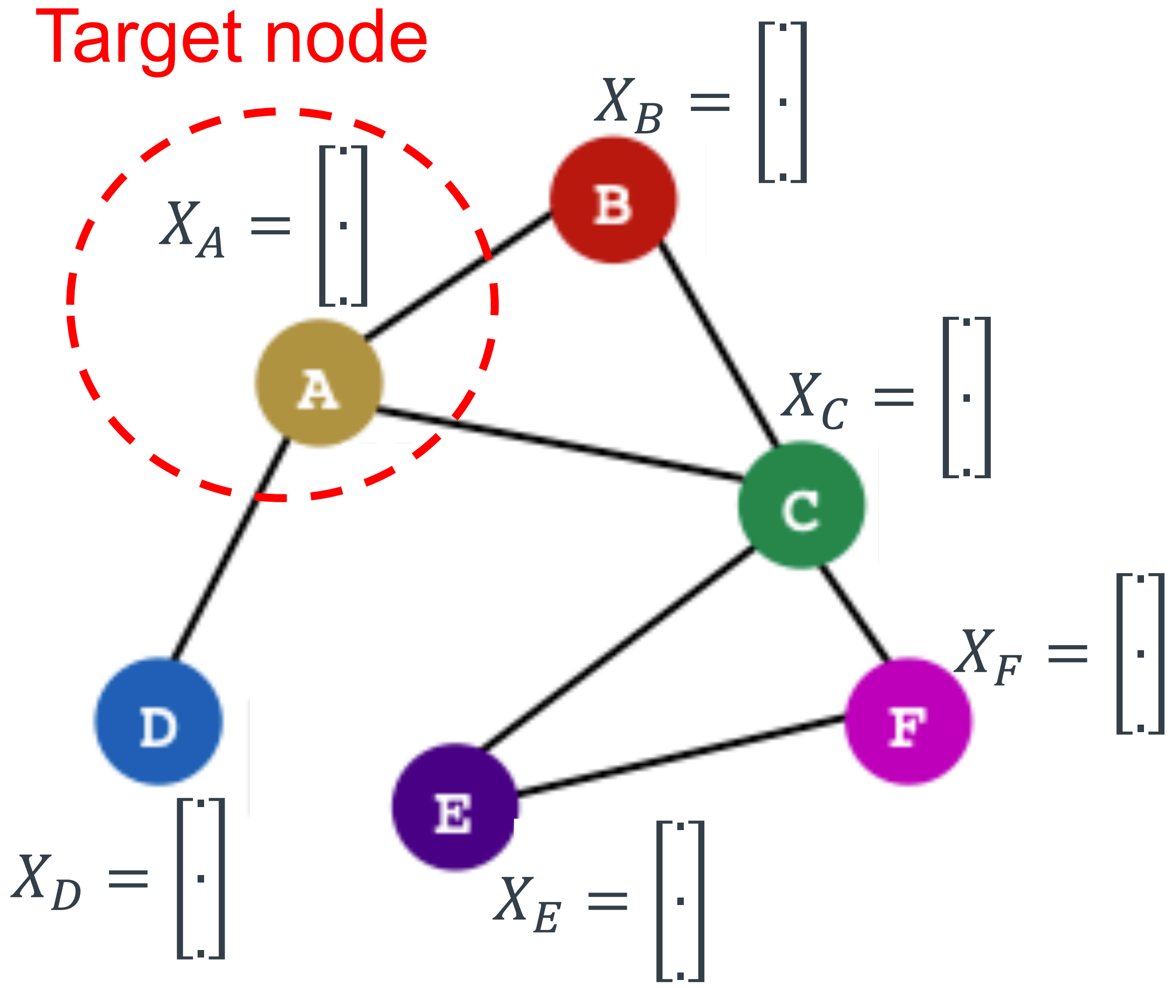
➢ Case 2: Neighborhood aggregation to node embedding

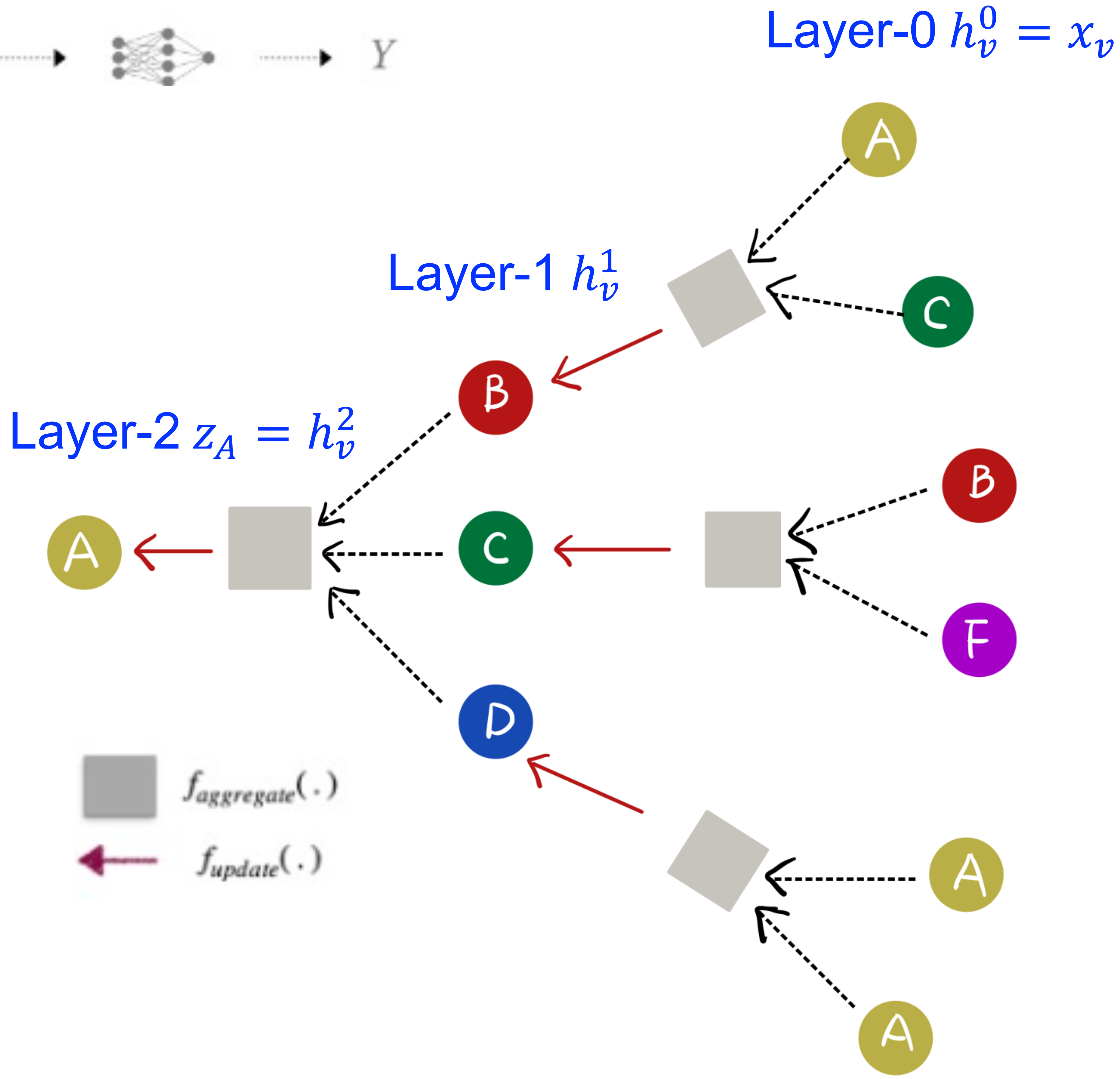


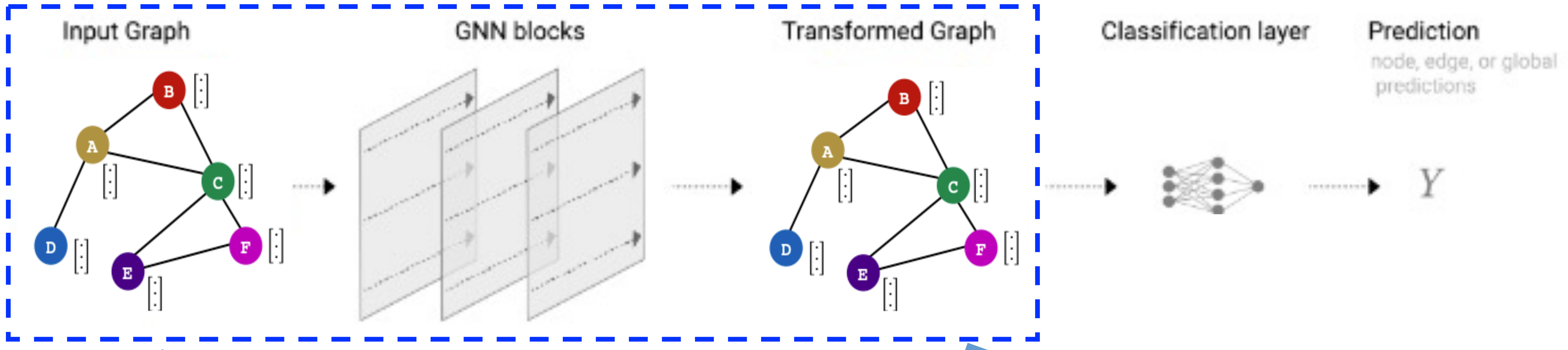
$$V'_1 = \frac{1}{5}[V_1 + V_2 + V_3 + V_4 + V_5] = [\vdots]_n$$

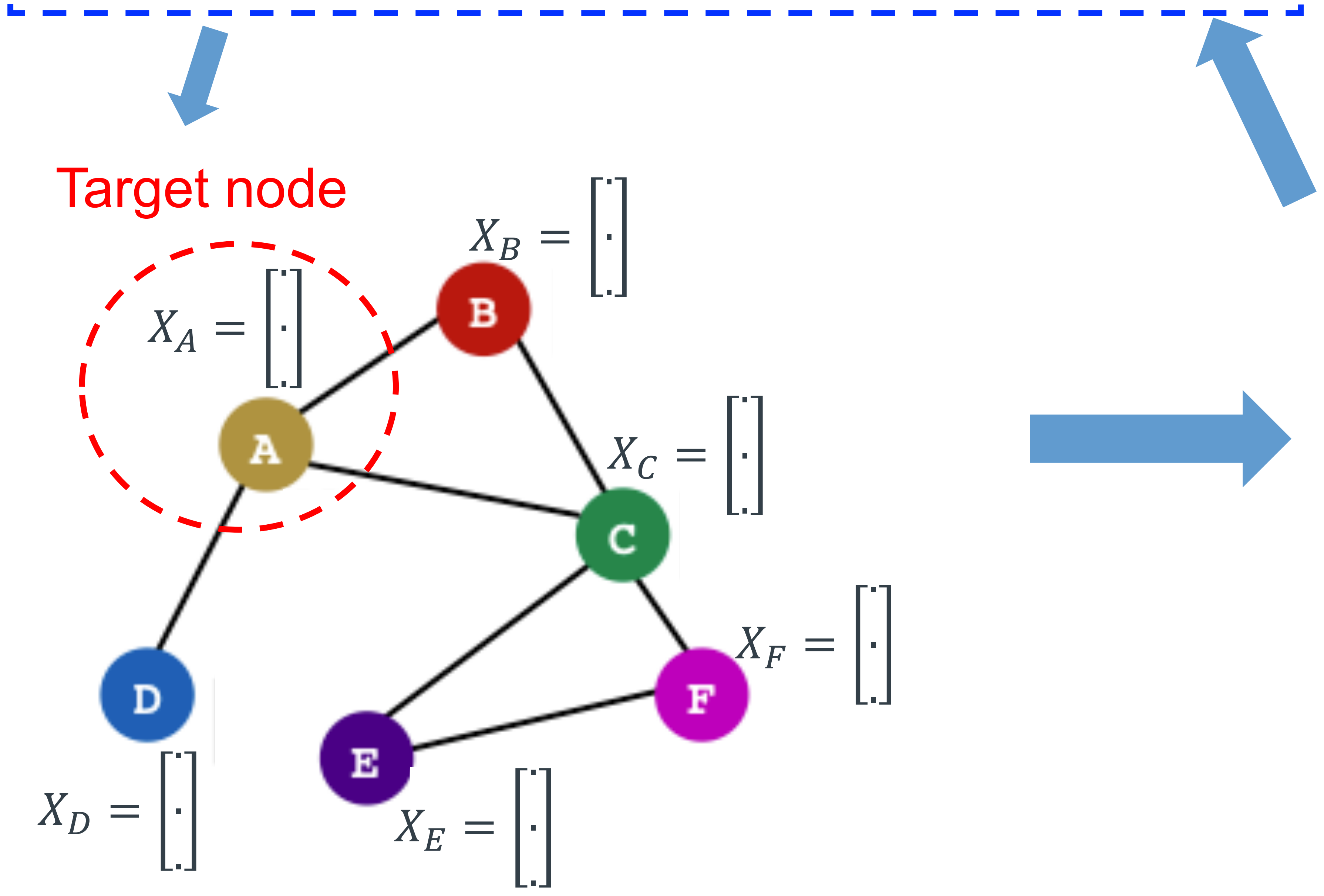


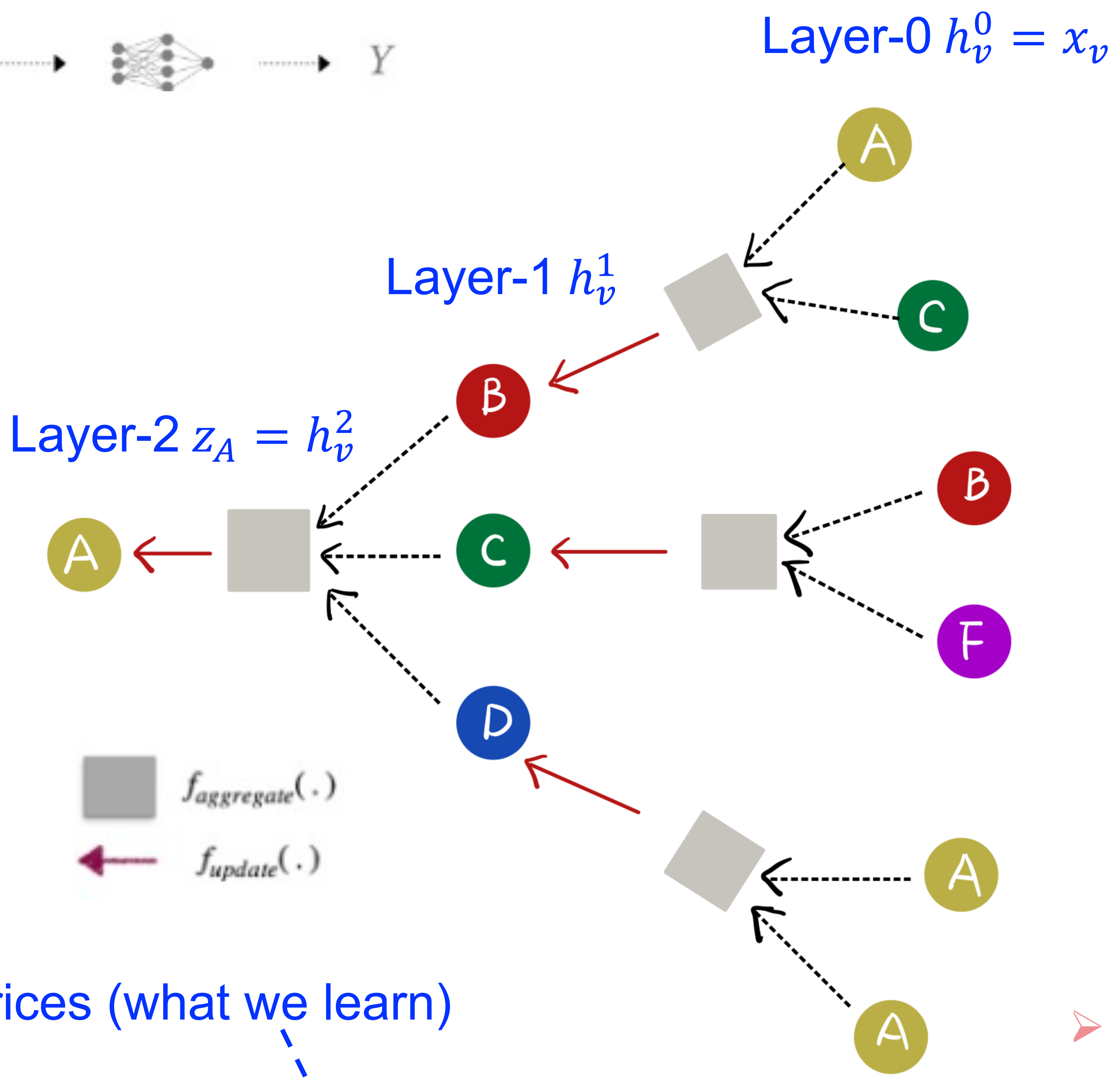

Input Graph
GNN blocks
Transformed Graph
Classification layer
Prediction
node, edge, or global predictions
Y


Target node
$X_A =$
$X_B =$
$X_C =$
$X_D =$
$X_E =$
$X_F =$
A
B
C
D
E
F


Layer-0 $h_v^0 = x_v$
Layer-1 h_v^1
Layer-2 $z_A = h_v^2$
$f_{aggregate}(.)$
$f_{update}(.)$
A
B
C
D
F







Activation function (e.g., ReLU)

Trainable weight matrices (what we learn)

$$h_v^k = \sigma\left(\left[W_k \cdot AGG\left(h_u^{k-1}, \forall u \in N(v)\right), B_k h_v^{k-1}\right]\right)$$

Aggregation of neighbor's previous layer embeddings

Previous layer embedding of target node

- One type of AGG:

$$\mathrm{AGG} = \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|}$$

Activation function (e.g., ReLU)

Aggregation of neighbor's previous layer embeddings

Previous layer embedding of target node

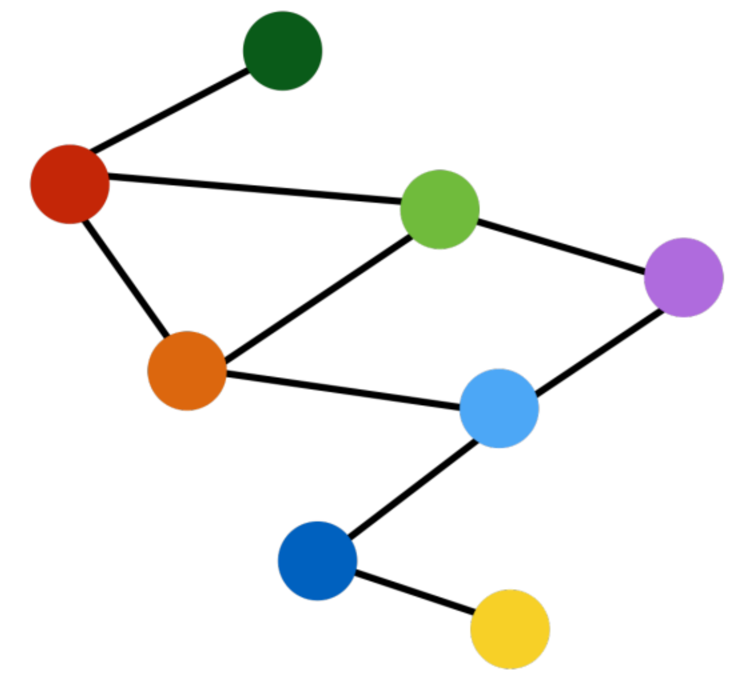
$$h_v^k = \sigma\left(\left[\boxed{W_k} \cdot AGG\left(h_u^{k-1}, \forall u \in N(v)\right), \boxed{B_k} h_v^{k-1}\right]\right)$$

Trainable weight matrices (what we learn)

Inductive capability:

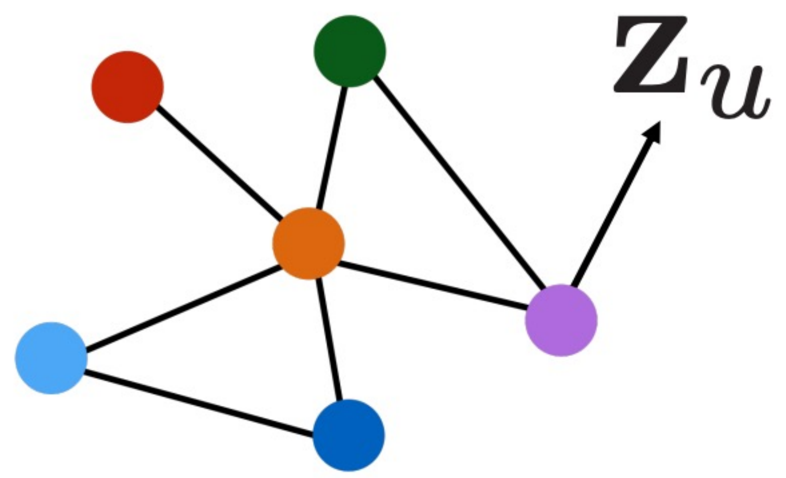
Once obtain the trained weight matrices, they are shared for all nodes

➢ Example: train on protein interaction graph from model organism A and generate embeddings on newly collected data about organism B, which shares similar features with A.



Train on model graph A

Obtain the trained weight matrices: W_k, B_k



Generate embeddings for new graph B

Outline

- Section 1. Some Basic Concepts of Neural Network
  - Neuron
  - Activation function
  - How do neural networks work
  - Artificial Neural Network (ANN) example
- Section 2. Graph Neural Network (GNN)
  - Graph data and graph tasks
  - How do GNNs work
  - GraphSAGE
- **Section 3. Application: Modeling Shared Mobility System Using GNN**

ASME® 2022 IDETC-CIE
International Design Engineering Technical Conferences & Computers and Information in Engineering Conference

ST. LOUIS UNION STATIOM HOTEL,
ST. LOUIS, MISSOURI

CONFERENCE: Aug 14–17, 2022
EXHIBITION: Aug 15–17, 2022

IDETC-CIE 2022-90694

Travel Links Prediction In Shared Mobility Networks Using Graph Neural Network Models

Yinshuang Xiao[1] | Faez Ahmed[2] | Zhenghui Sha[1,*]
yinshuangxiao@utexas.edu | faez@mit.edu | zsha@austin.utexas.edu

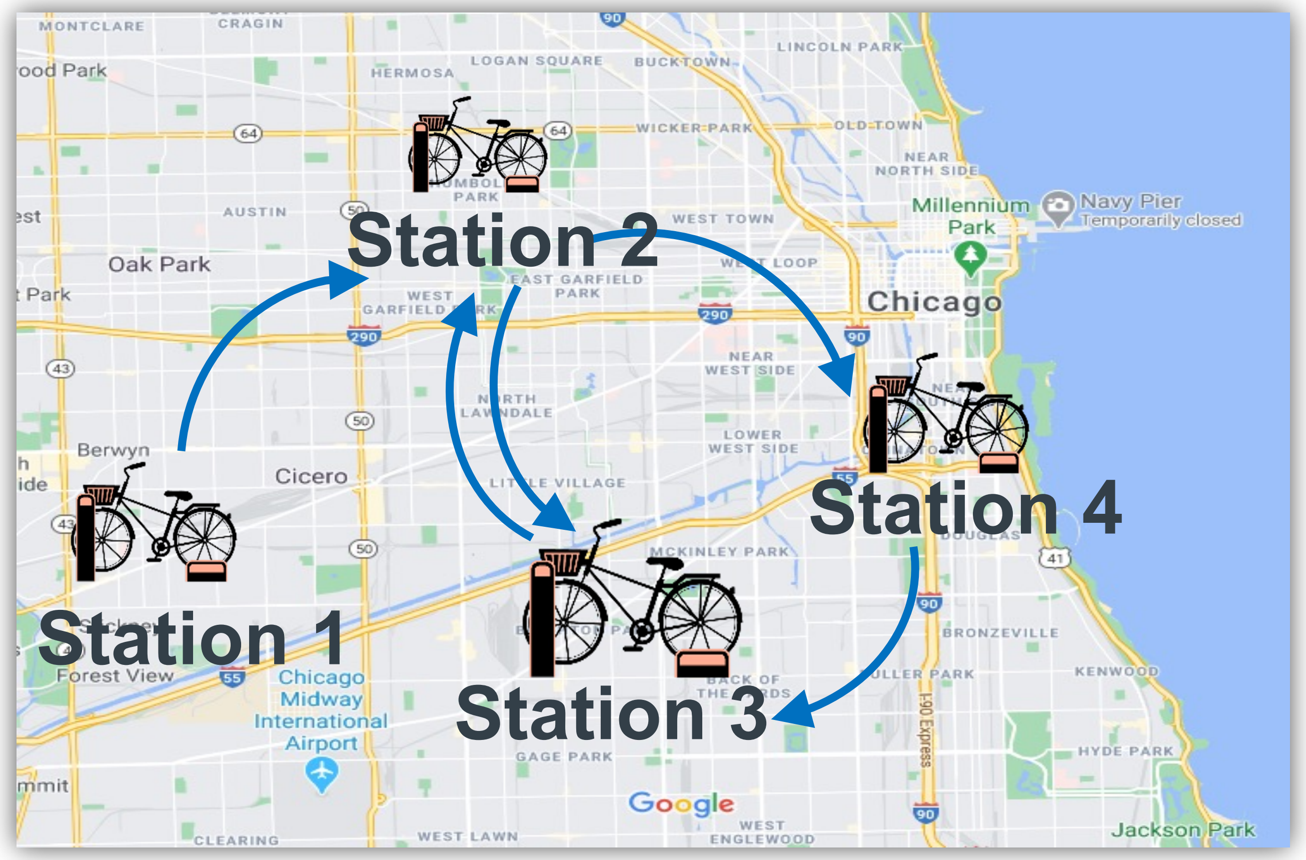
IDETC–CIE 2022, August 14–15, 2022, St. Louis, Missouri

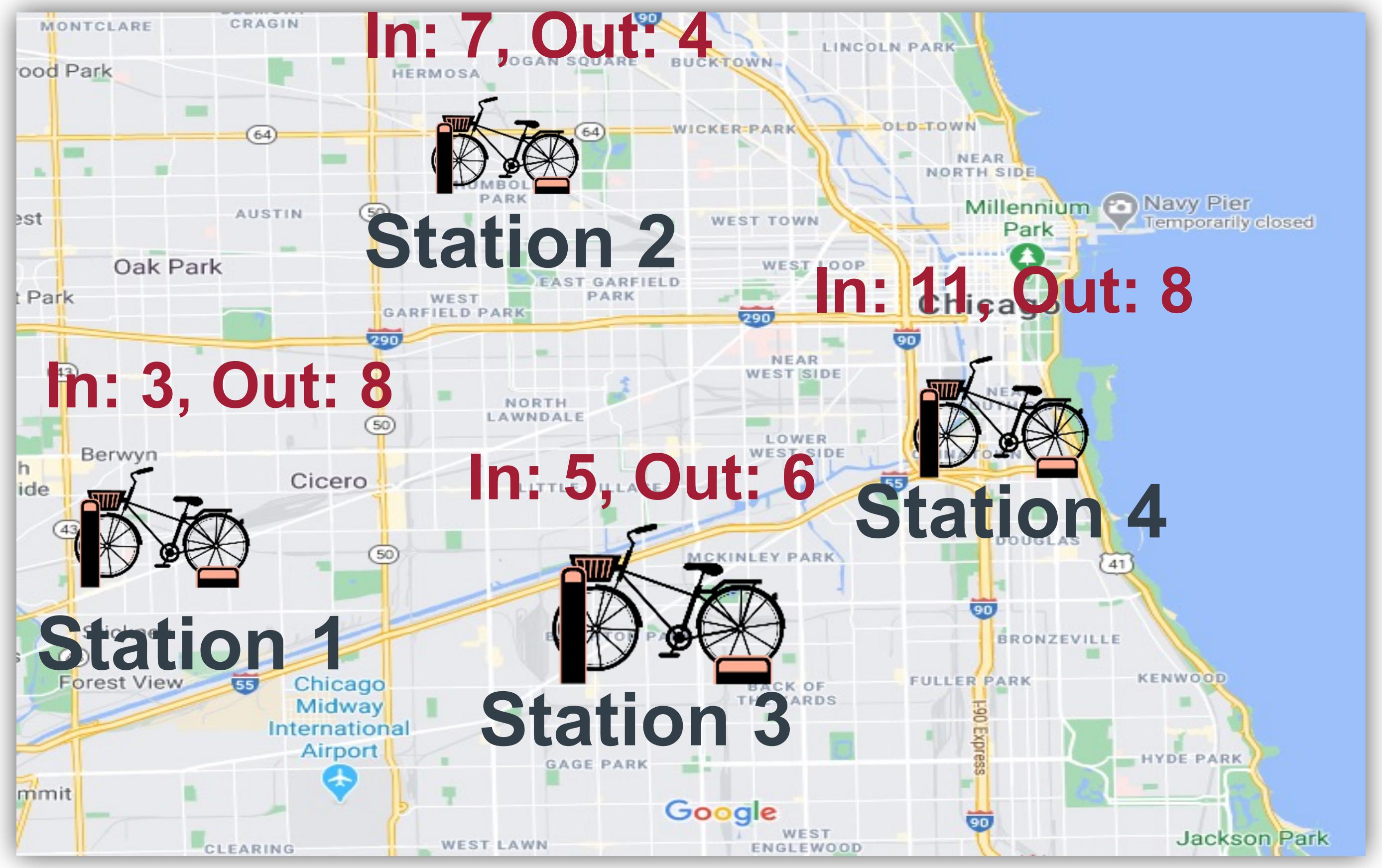
1 Walker Department of Mechanical Engineering, The University of Texas at Austin, USA
2 Department of Mechanical Engineering, Massachusetts Institute of Technology, USA
* Corresponding Author

TEXAS The University of Texas at Austin | SIDI LAB System Integration & Design Informatics Laboratory

MIT DECODE DESIGN COMPUTATION & DIGITAL ENGINEERING LAB

Background and Motivation





Factors Influencing Travel Behavior

- Time factors, e.g., peak hours, holidays
- Climate effects
- Spatial dependencies among serving stations
- Surrounding Point of Interests (POIs)

Shared Mobility System Predictive Model

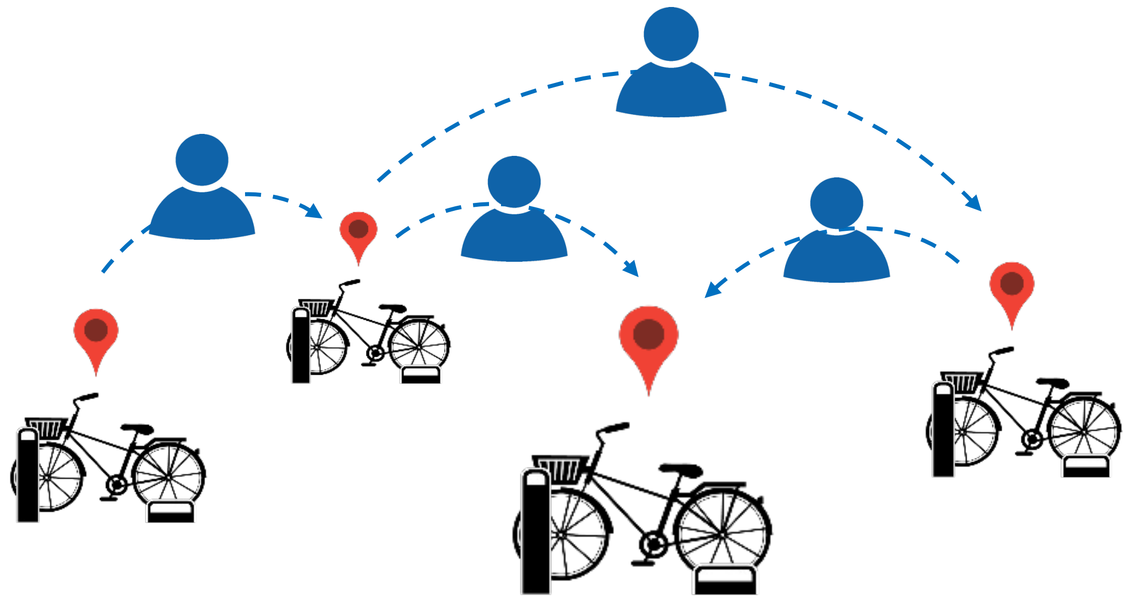
- Help test system design approach
- Forecast the usage of system capacity
- Guide decision-making on system operation

Limitation of Existing Study

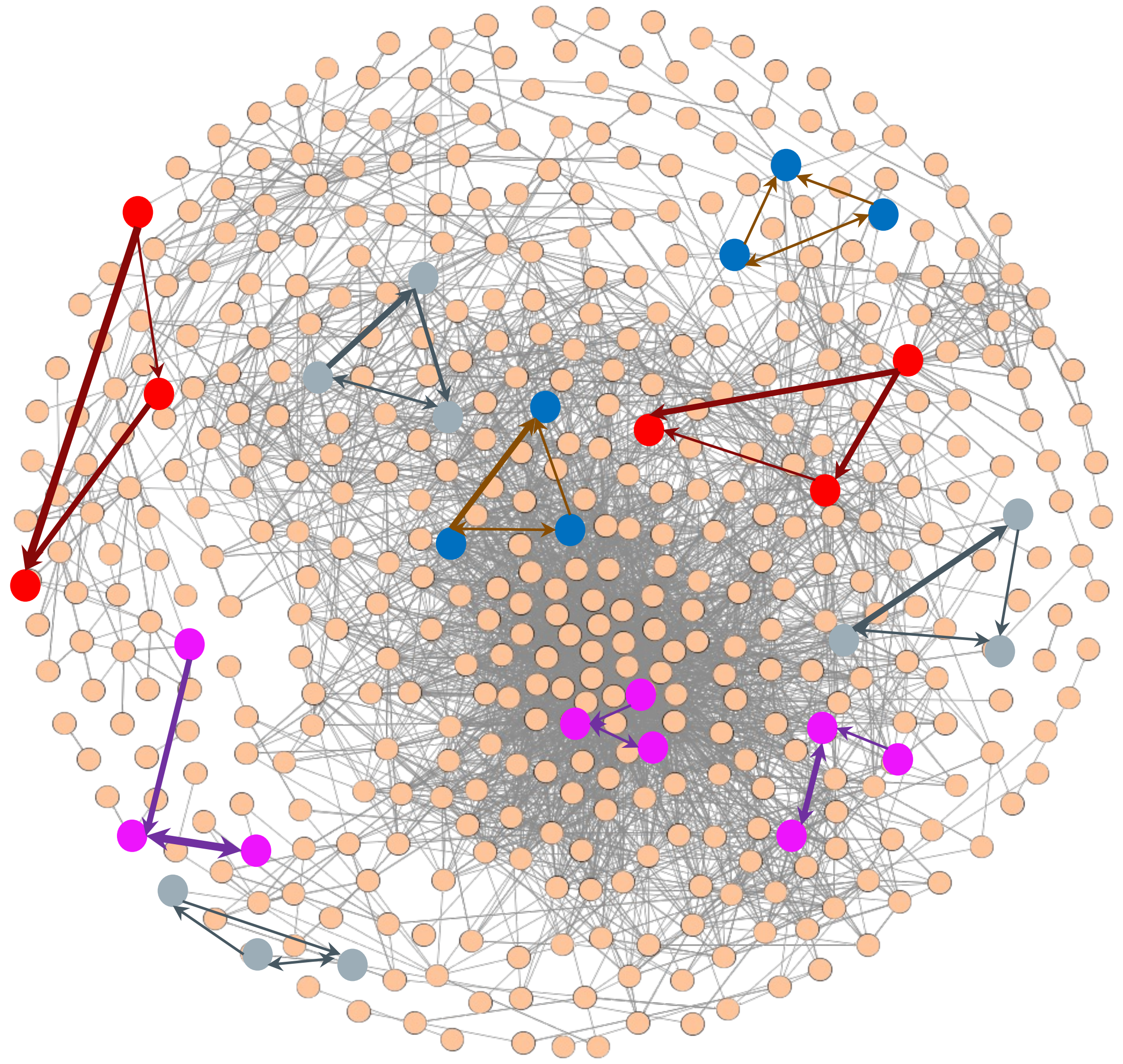
- Only predict station-level rental and return demands but do not tell where the return comes from and the rental goes to.

Background and Motivation

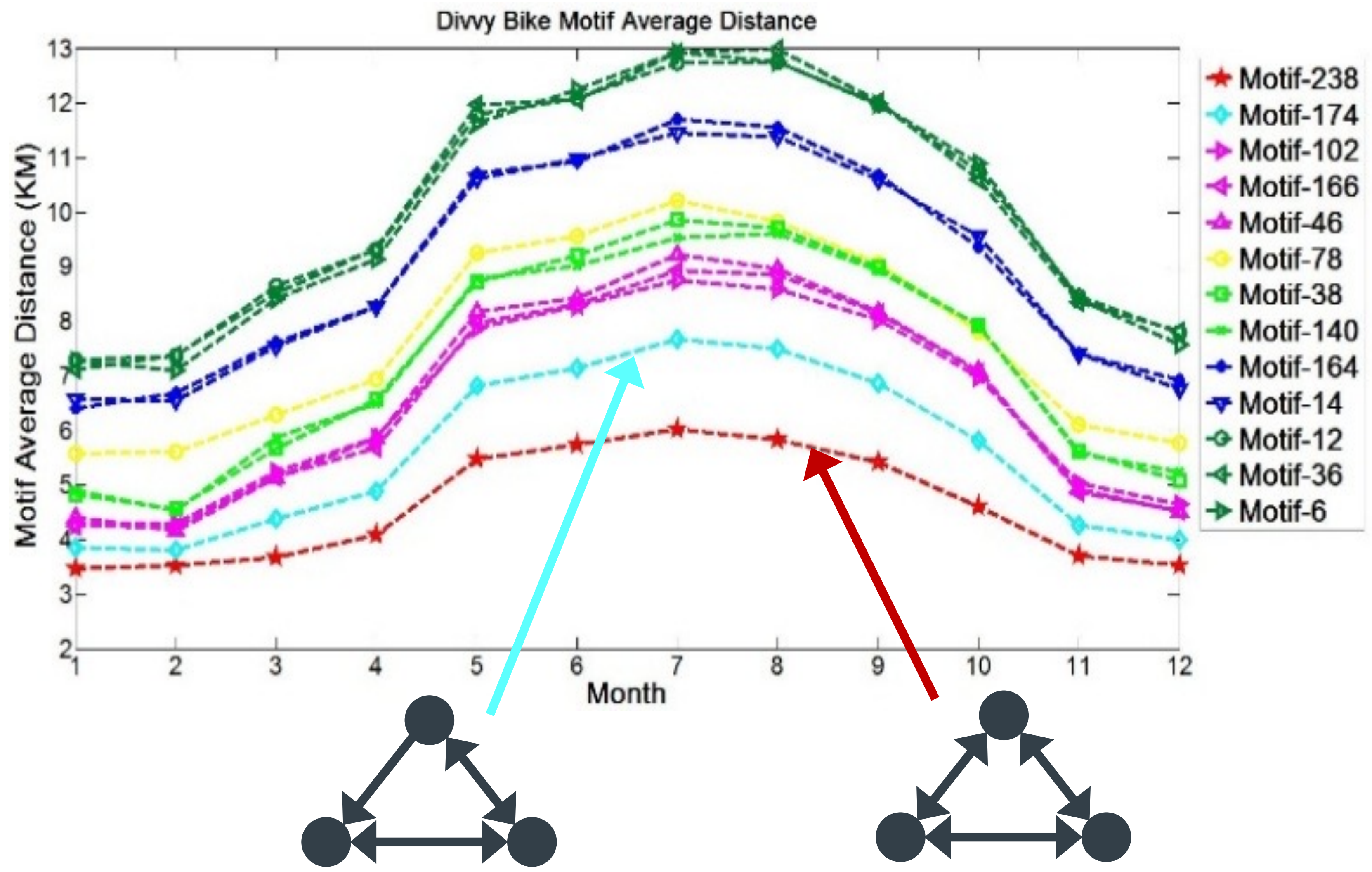
Shared Mobility System



Shared Mobility Network and Significant Local Service Systems



Hierarchical Average Distance Distributions of Different Local Service Systems



Understanding Generated From Previous Works[1]

- The Complex network is a powerful tool to represent shared mobility systems and user behaviors.
- Significant local service systems are identified based on network motif theory.
- The local service systems show typical characteristics such as obvious hierarchical average geographical distance distribution which is correlated to the local system structures.

[1] Xiao, Y., and Sha, Z., 2020. "Towards engineering com- plex socio-technical systems using network motifs: A case study on bike-sharing systems". In International Design Engineering Technical Conferences and Computers and In- formation in Engineering Conference, Vol. 84003, American Society of Mechanical Engineers, p. V11AT11A045.

Research Question and Objective

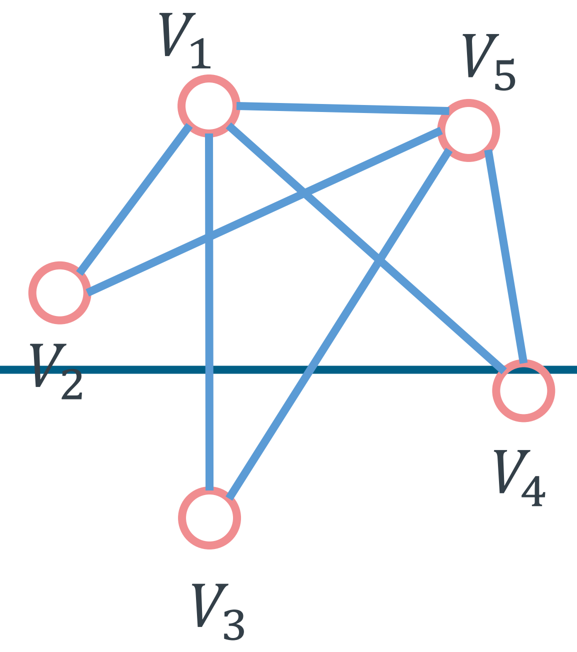
Research Objective

The existences of trips occurring from one station to another in a period of time.

Develop a network-based approach to predict travel demand between stations in shared mobility systems based on local network information.

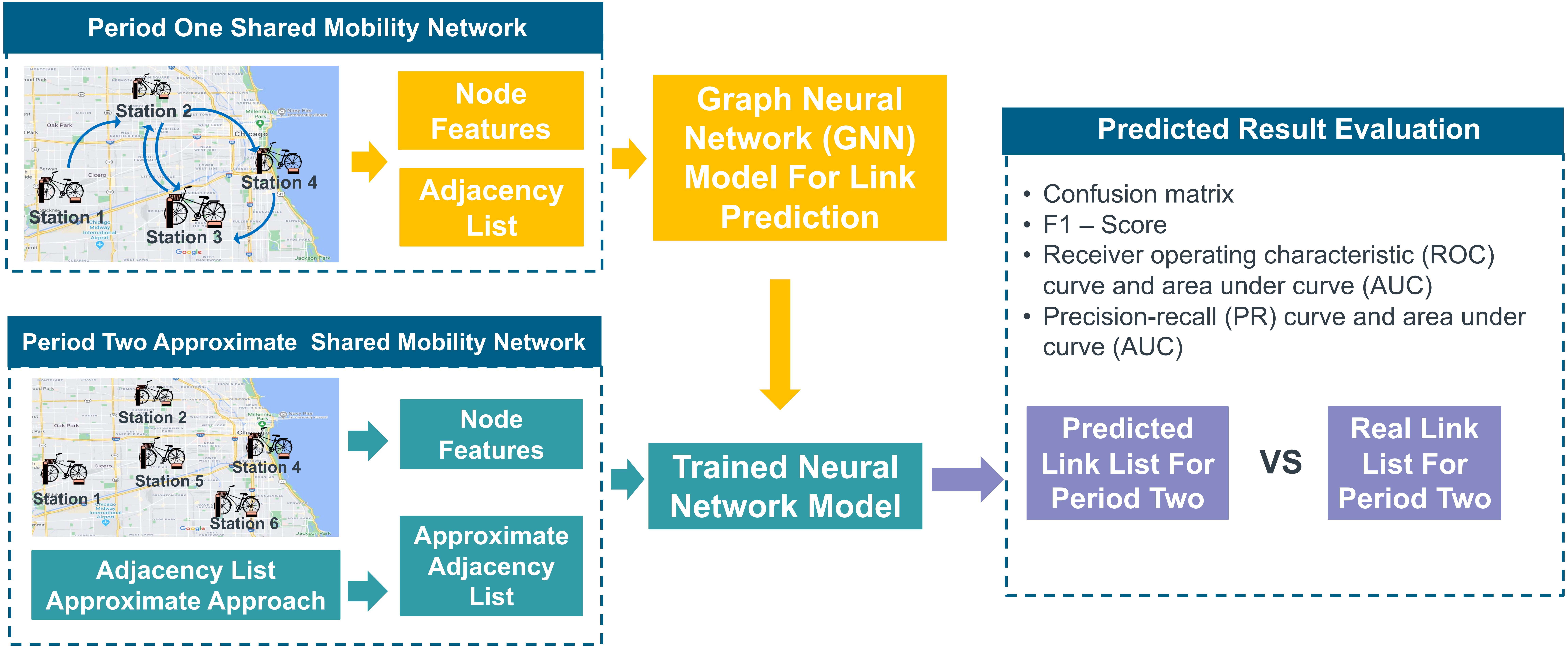
Research Questions

E.g., the local network information of V2 means the information of V1 and V5



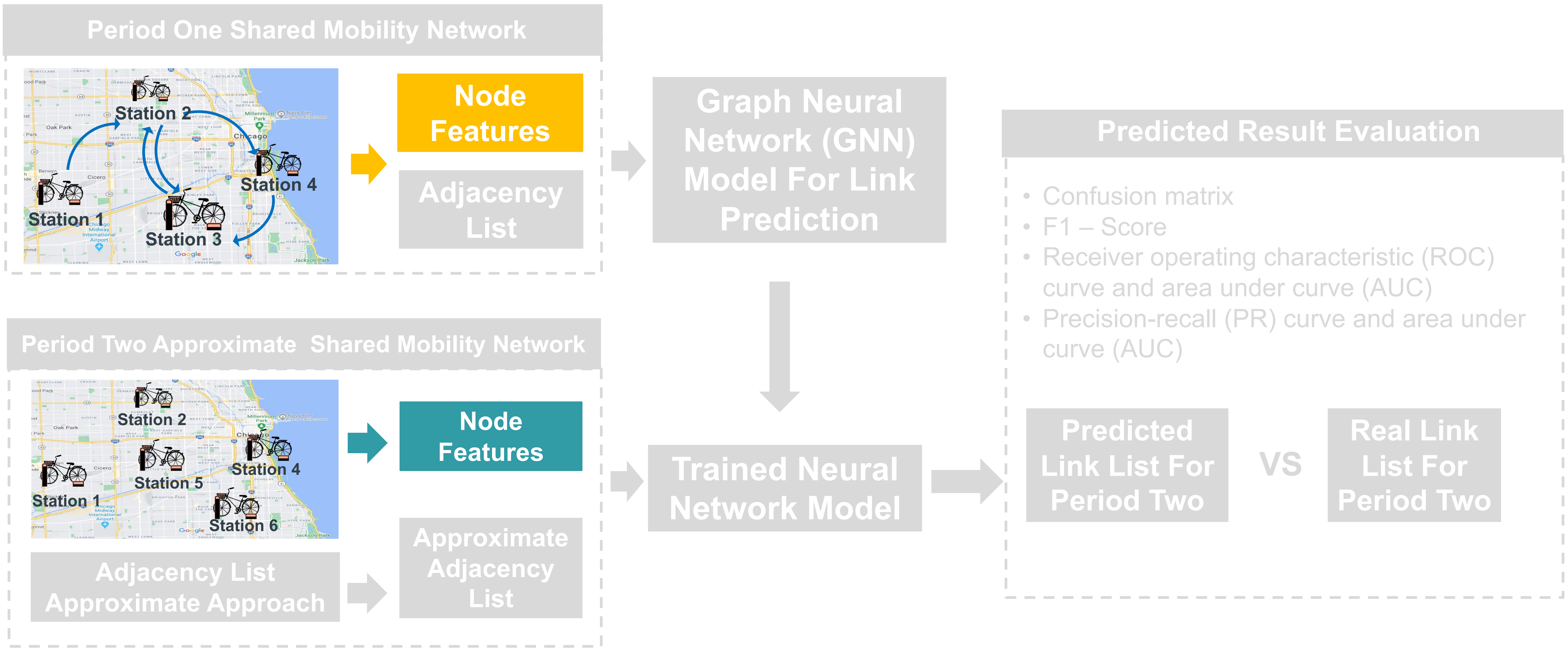
Whether and to what extent does the local network information (e.g., structure and node features) play a role in the formation of a shared mobility network?

Research Approach



Period One: Month i in year Y; **Period Two**: Month i in year $Y+1$, (i=1,…,12)

Research Approach



Period One: Month i in year Y; **Period Two**: Month i in year $Y+1$, (i=1,…,12)

Node Features

Station attributes determined by system designers

Latitude	Longitude	Dock #	Financial #	Education #	Recreational & Tourism #	Residential #	Sustenance #	Healthcare #	Transportation #

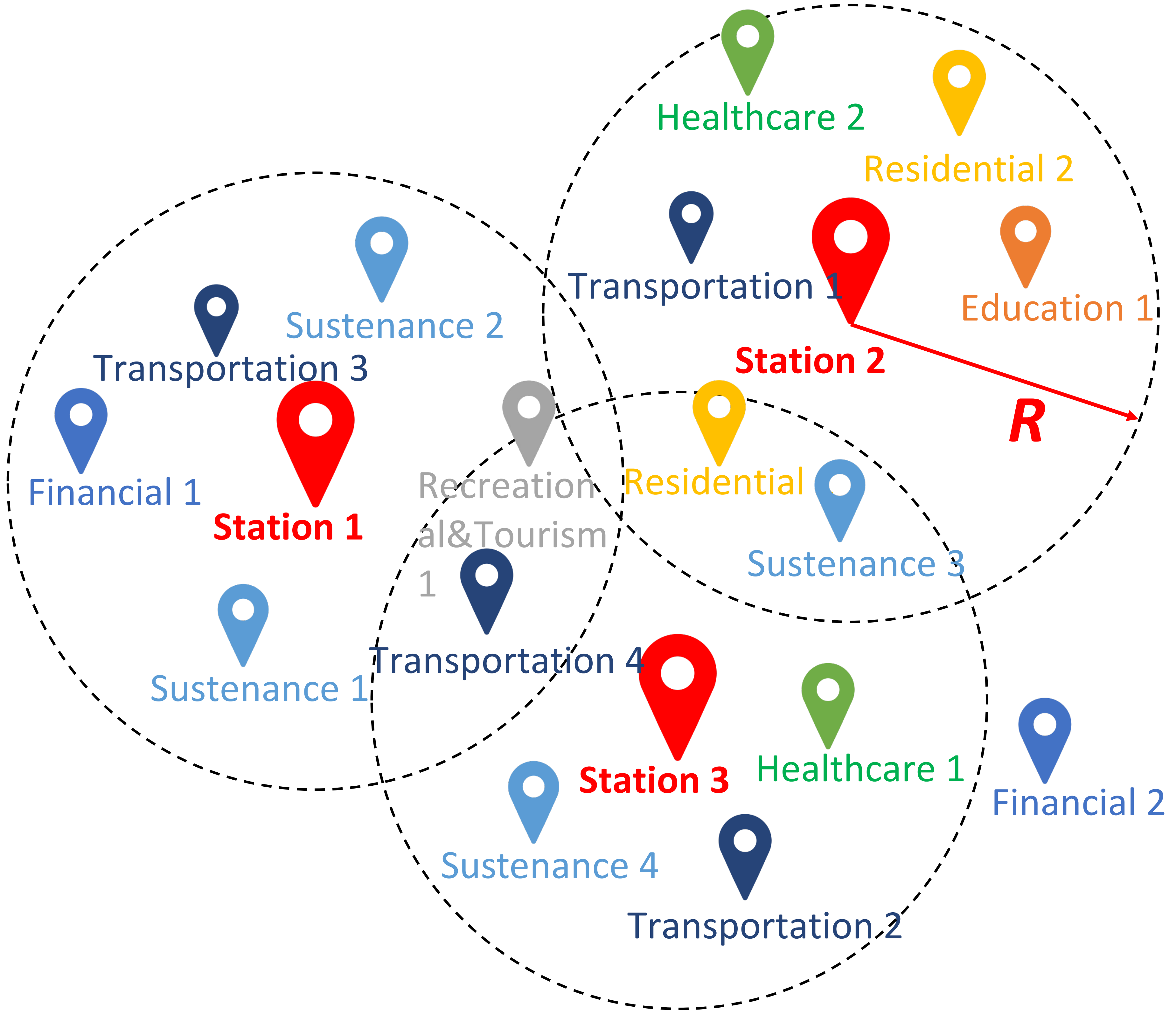
[2] Liu, J., Sun, L., Li, Q., Ming, J., Liu, Y., and Xiong, H., 2017. "Functional zone based hierarchical demand pre- diction for bike system expansion". In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 957–966.

Node Features

Station attributes determined by system designers

Latitude	Longitude	Dock #	Financial #	Education #	Recreational & Tourism #	Residential #	Sustenance #	Healthcare #	Transportation #

Station surrounding point of interests (POIs)



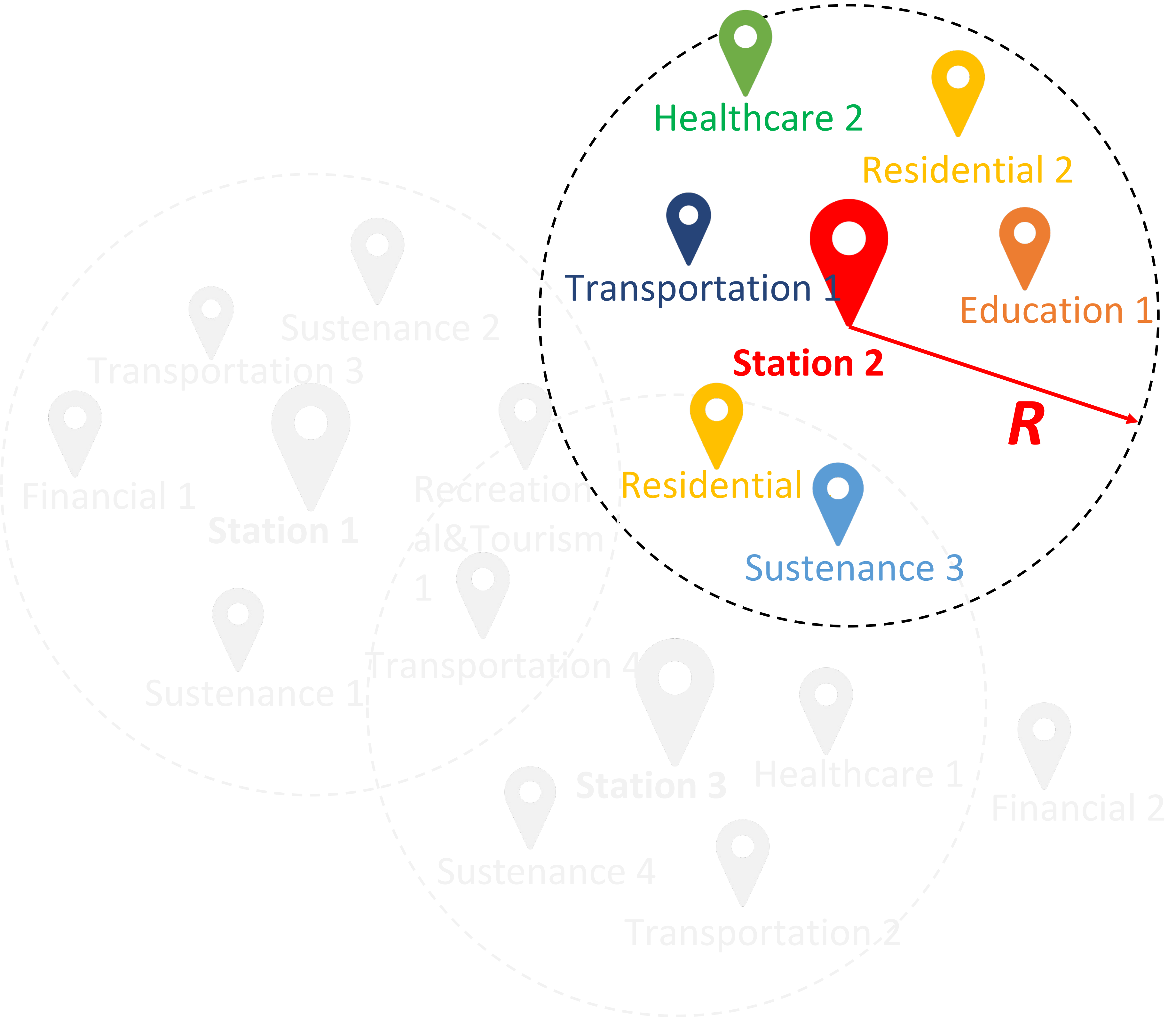
[2] Liu, J., Sun, L., Li, Q., Ming, J., Liu, Y., and Xiong, H., 2017. "Functional zone based hierarchical demand pre- diction for bike system expansion". In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 957–966.

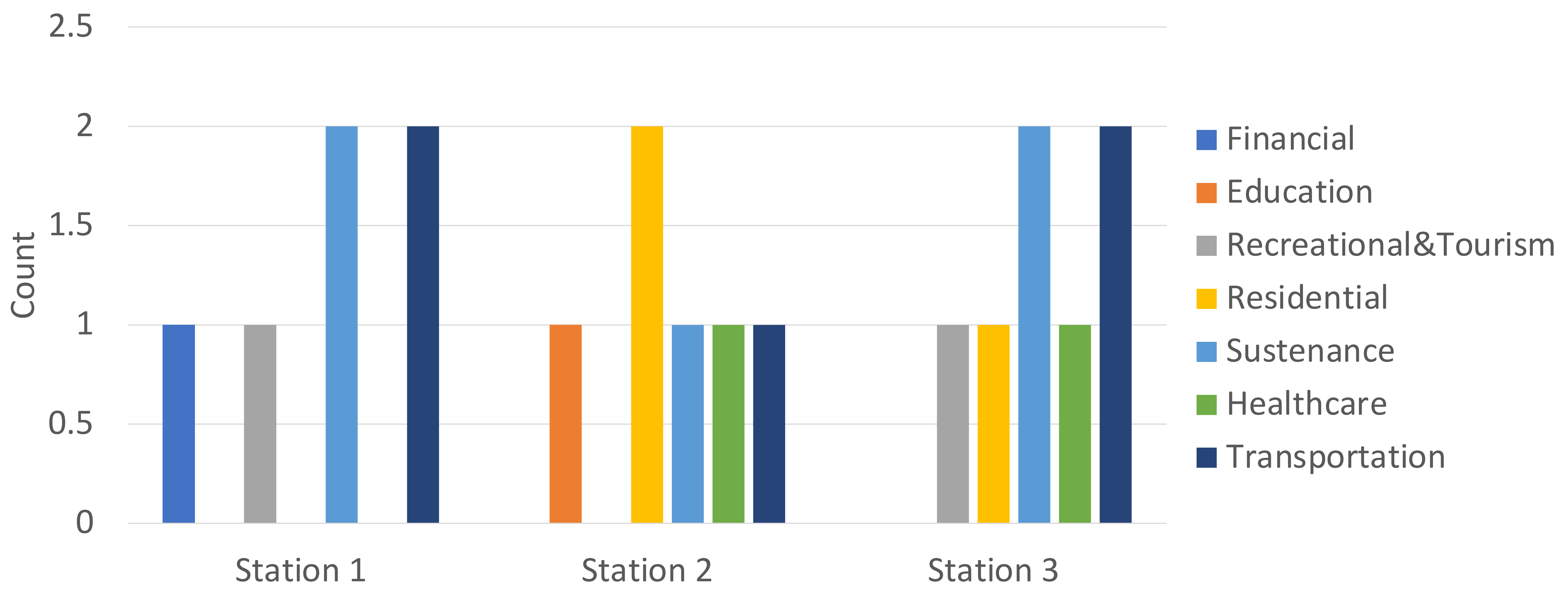
Node Features

Station attributes determined by system designers

Latitude	**Longitude**	**Dock #**	**Financial #**	**Education #**	**Recreational & Tourism #**	**Residential #**	**Sustenance #**	**Healthcare #**	**Transportation #**

Station surrounding point of interests (POIs)

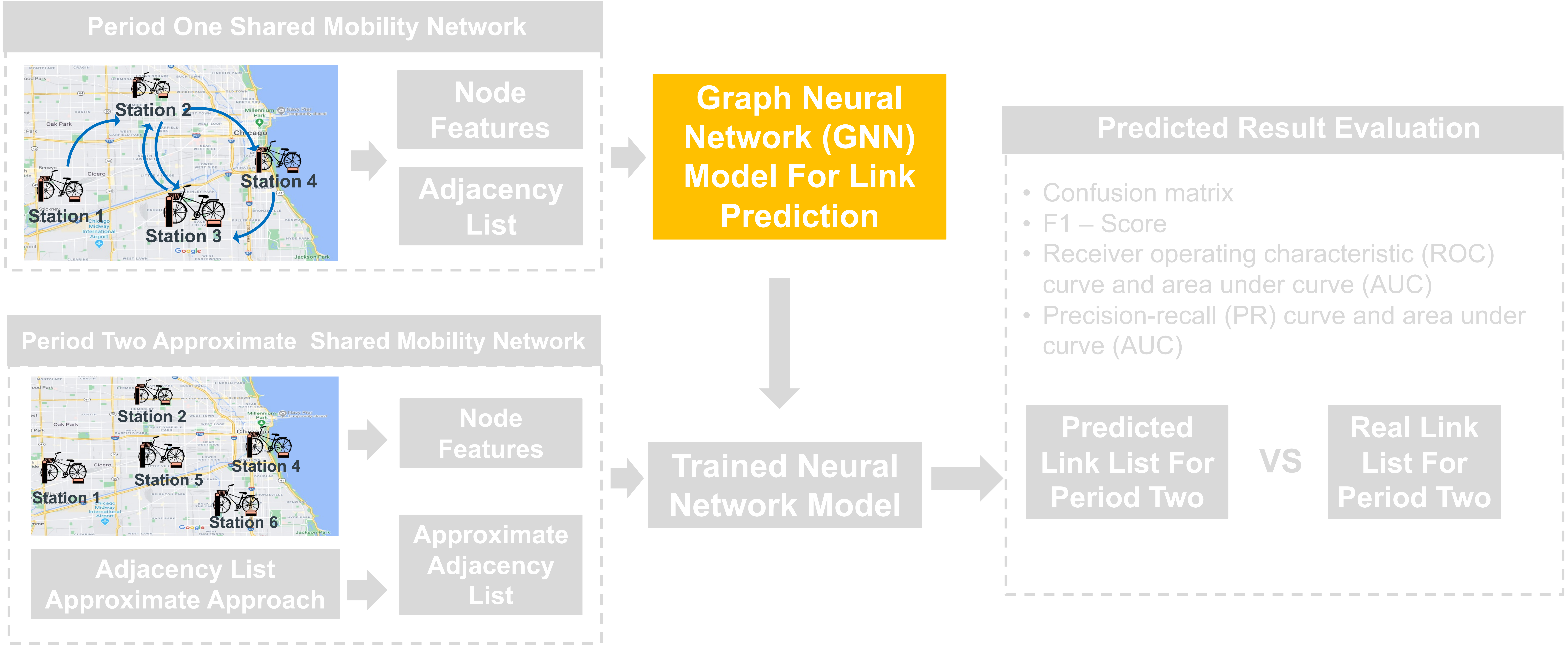




R: referring to existing work[2], a person would prefer to ride when the origin-to-destination distance is 1.5 ~ 2.7 miles. Therefore, **1.5 miles** is the longest distance that people are willing to commute by walking.

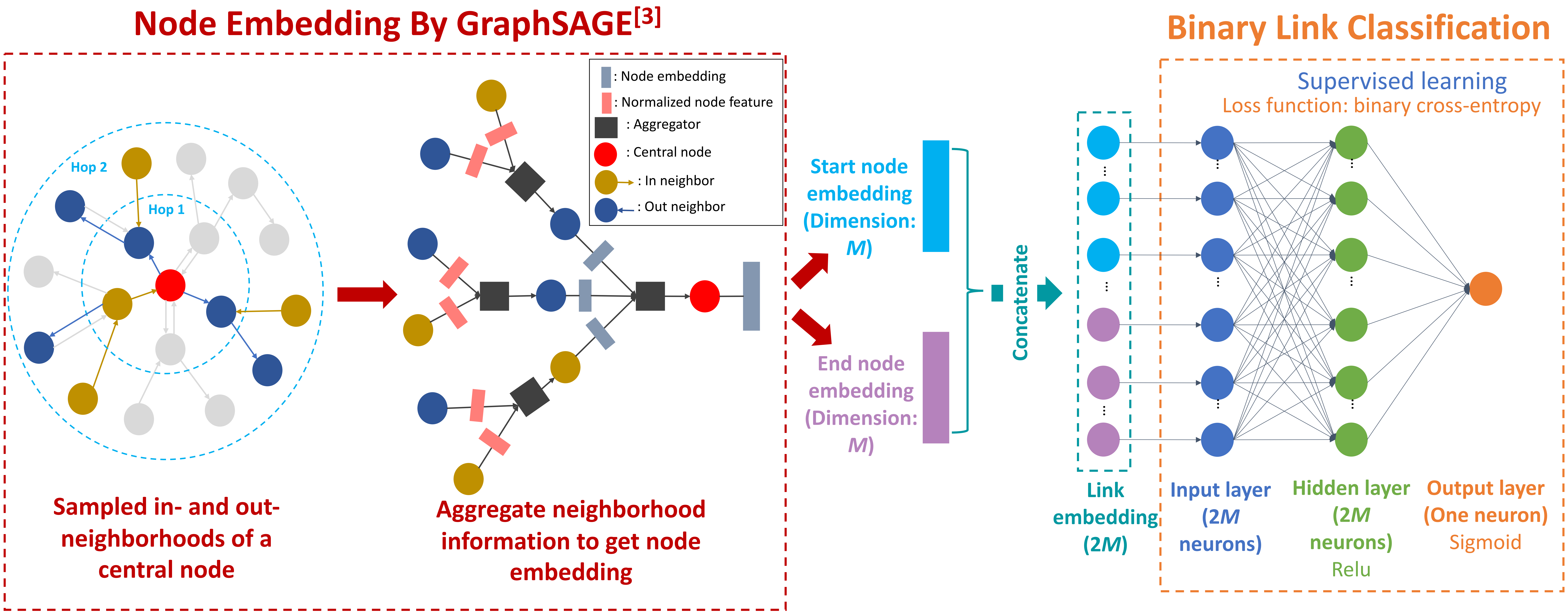
[2] Liu, J., Sun, L., Li, Q., Ming, J., Liu, Y., and Xiong, H., 2017. "Functional zone based hierarchical demand pre- diction for bike system expansion". In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 957–966.

Research Approach



Period One: Month i in year Y; **Period Two**: Month i in year $Y+1$, (i=1,…,12)

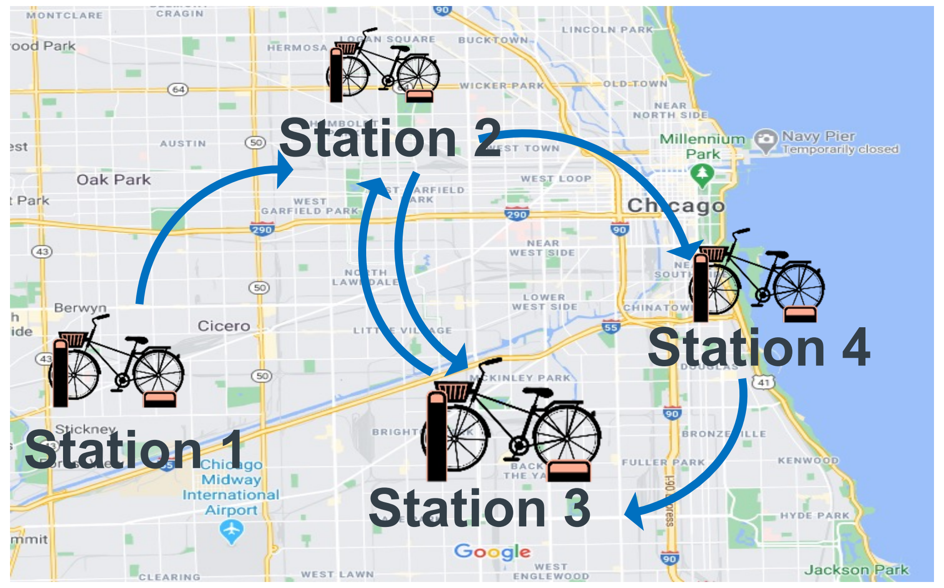
GNN Model For Link Prediction



[3] Hamilton, W. L., Ying, R., and Leskovec, J., 2017. "Inductive representation learning on large graphs". In Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 1025–1035.

Research Approach

Period One Shared Mobility Network

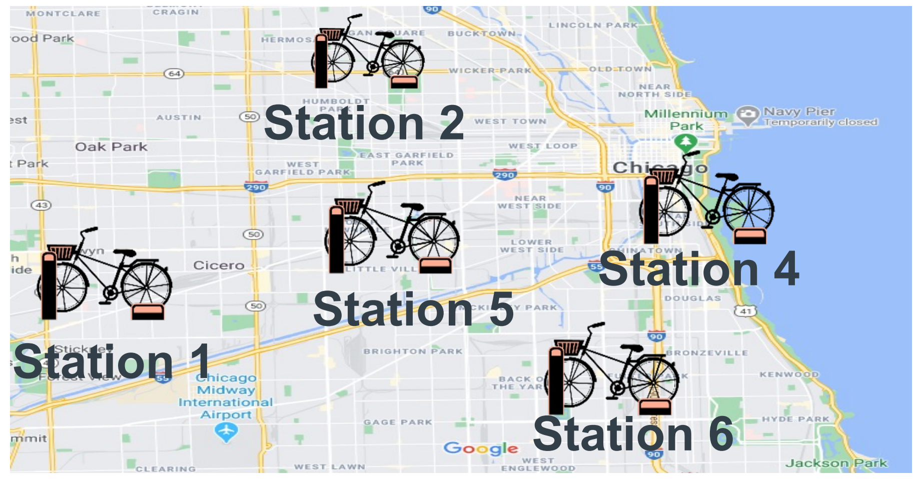


Node Features

Adjacency List

Graph Neural Network (GNN) Model For Link Prediction

Period Two Approximate Shared Mobility Network



Node Features

Adjacency List Approximate Approach

Approximate Adjacency List

Trained Neural Network Model

Predicted Result Evaluation

- Confusion matrix
- F1 – Score
- Receiver operating characteristic (ROC) curve and area under curve (AUC)
- Precision-recall (PR) curve and area under curve (AUC)

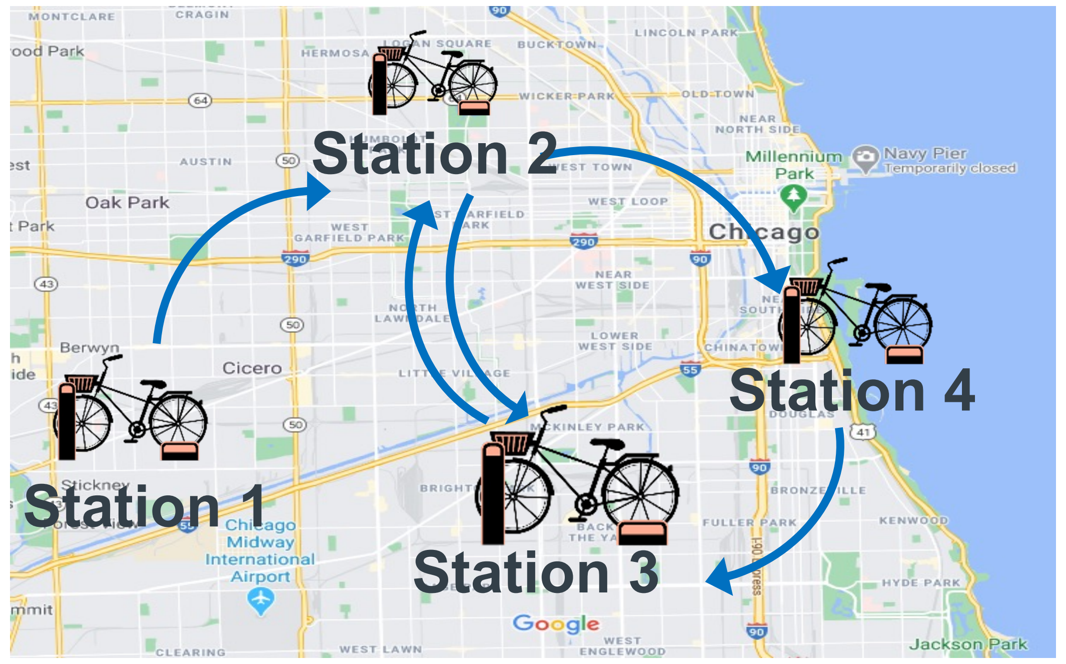
Predicted Link List For Period Two VS Real Link List For Period Two

Period One: Month i in year Y; **Period Two**: Month i in year $Y+1$, (i=1,…,12)

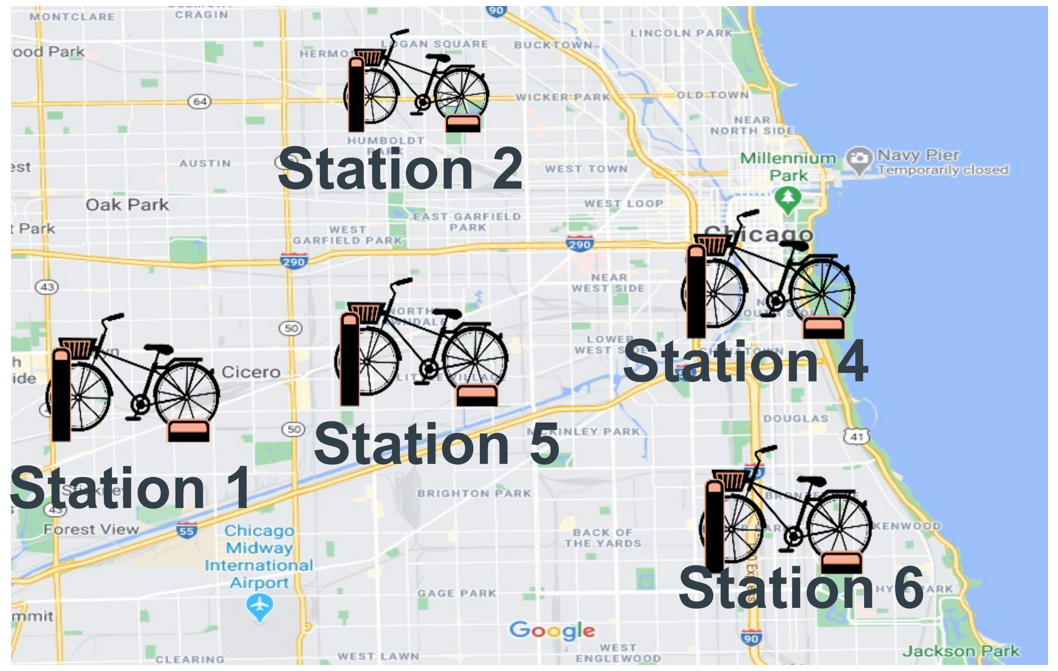
Adjacency List Approximate Approach

➢Approach 1: Modified Period One Mobility Network

[

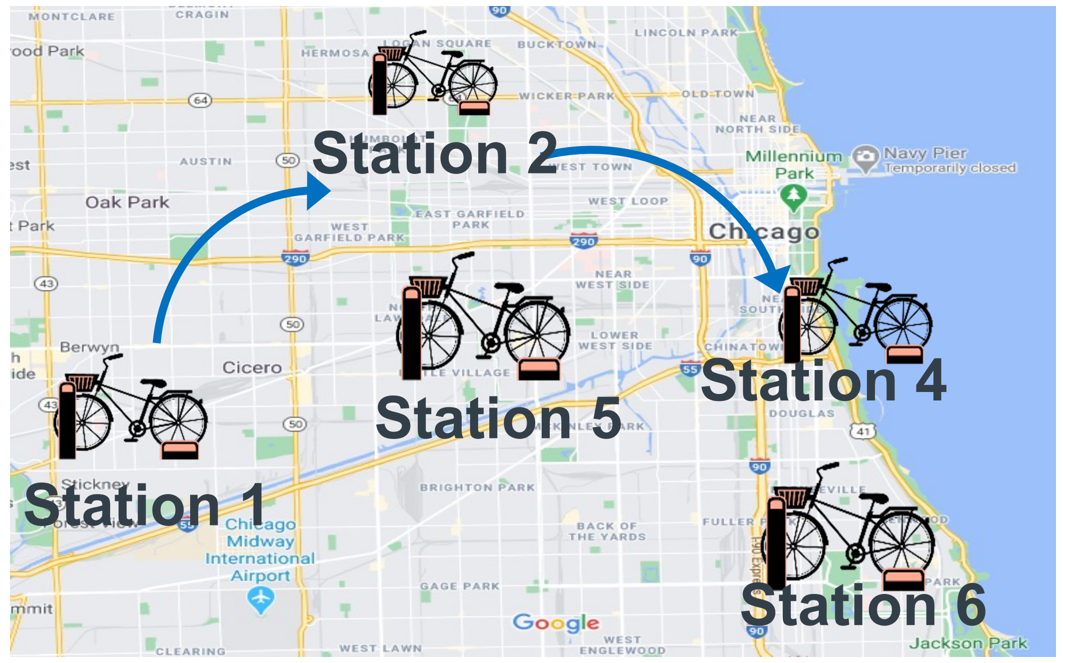


Period One Mobility Network



Period Two Mobility Stations

] =



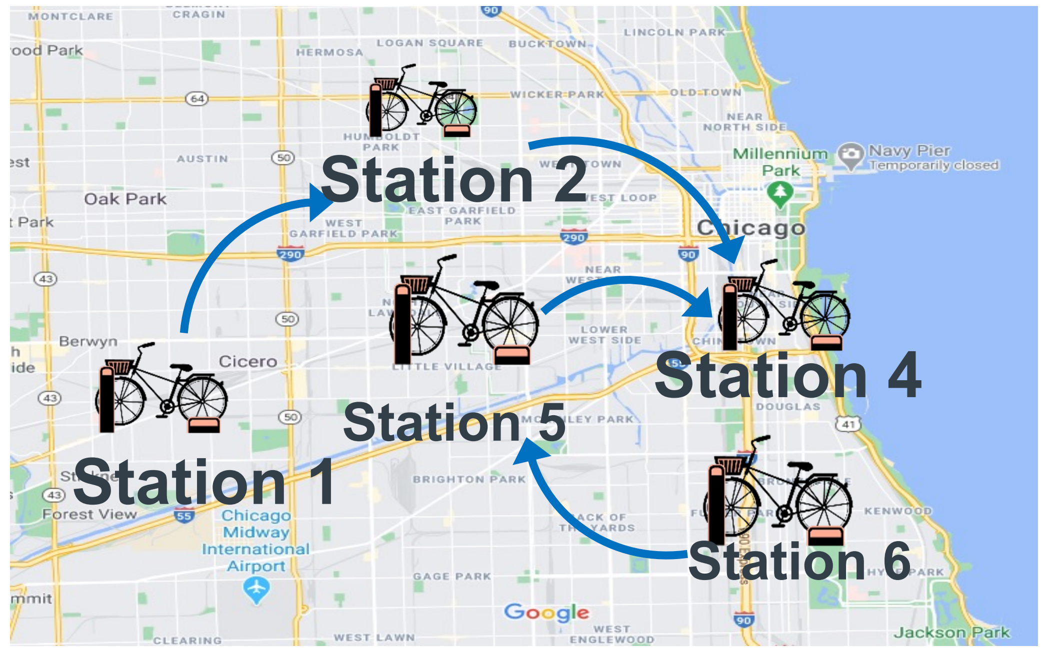
Modified Period One Mobility Network

➢Approach 2: ANN-Based Approximate Period Two Mobility Network

Artificial Neural Network (ANN) Model For Link Prediction

Approximate Period Two Mobility Network

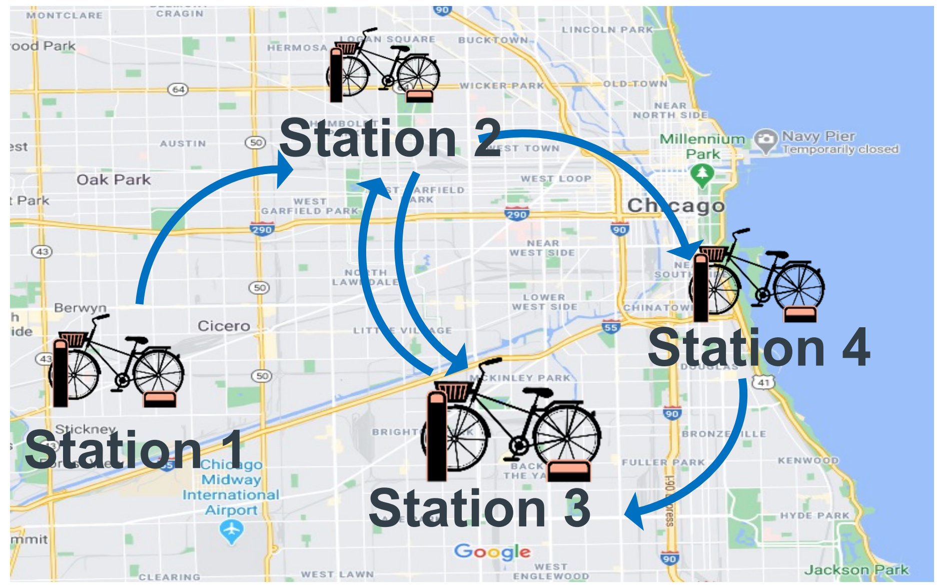
➢Approach 3 (Ground Truth): Real Period Two Mobility Network



Real Period Two Mobility Network

Baseline

Period One Shared Mobility Network



Node Features

Artificial Neural Network (ANN) Model For Link Prediction

Period Two Approximate Shared Mobility Network

Node Features

Trained Neural Network Model

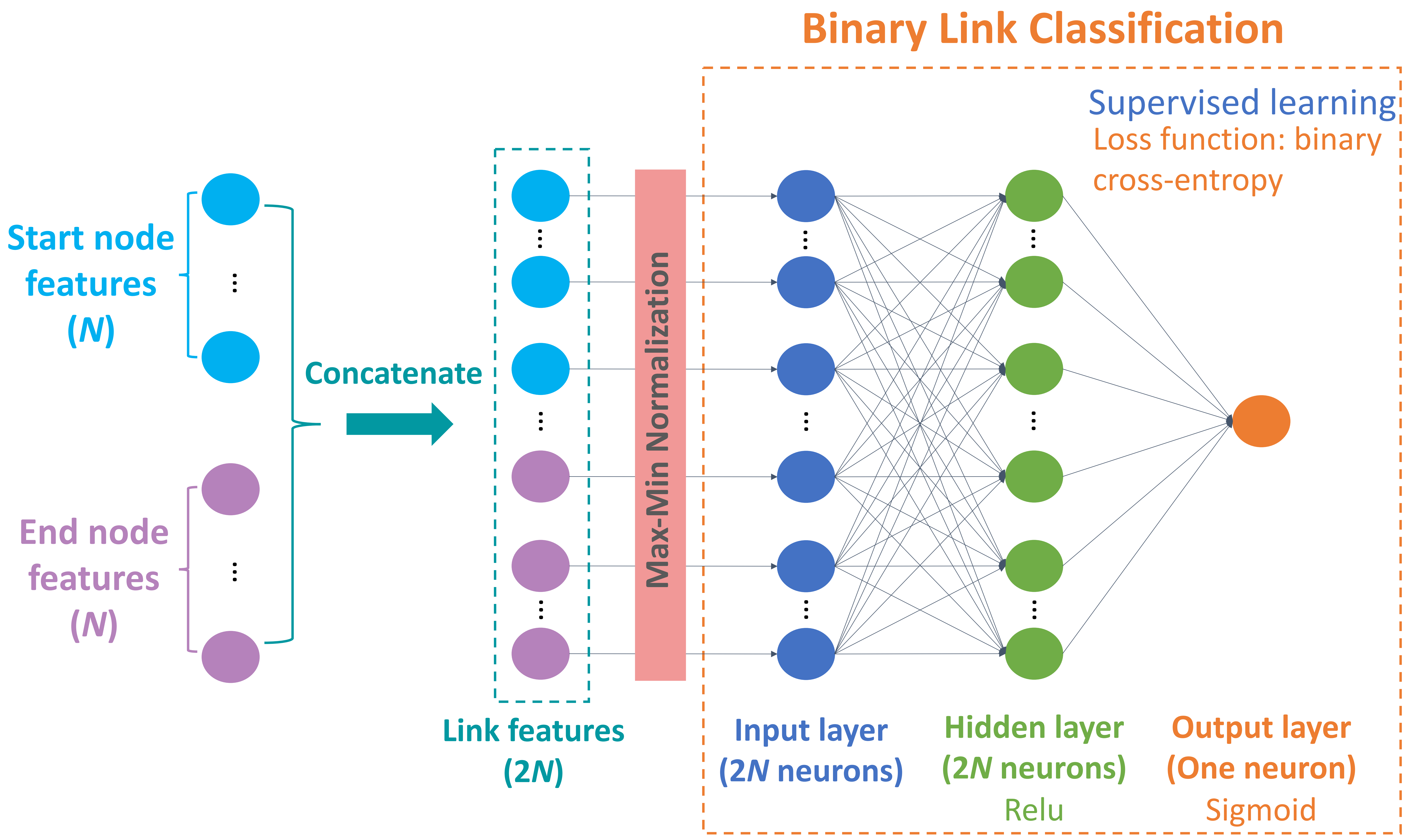
Predicted Result Evaluation

- Confusion matrix
- F1 – Score
- Receiver operating characteristic (ROC) curve and area under curve (AUC)
- Precision-recall (PR) curve and area under curve (AUC)

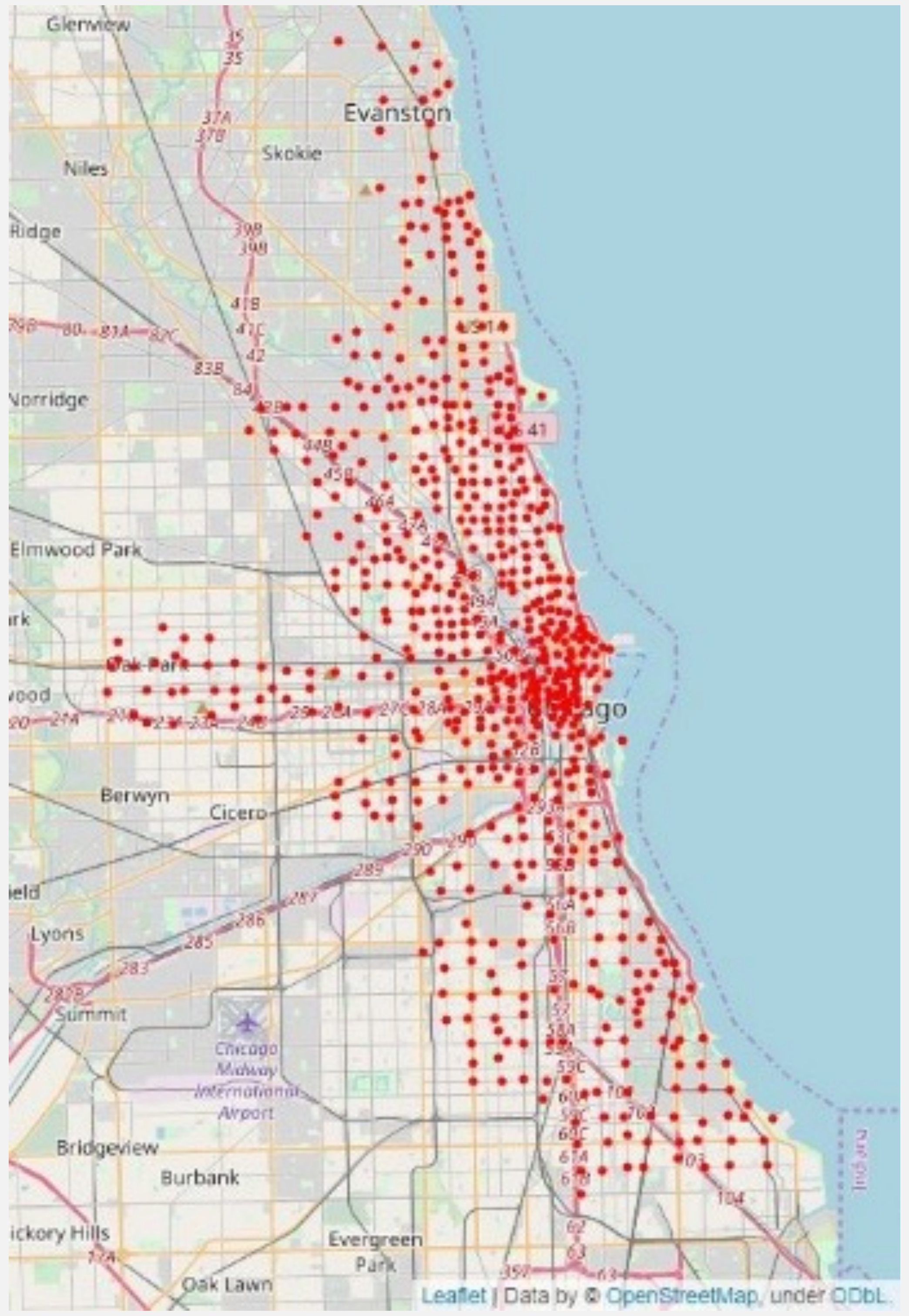
Predicted Link List For Period Two **VS** **Real Link List For Period Two**

Period One: Month i in year Y; **Period Two**: Month i in year $Y+1$, (i=1,…,12)

Baseline: ANN Link Prediction Model



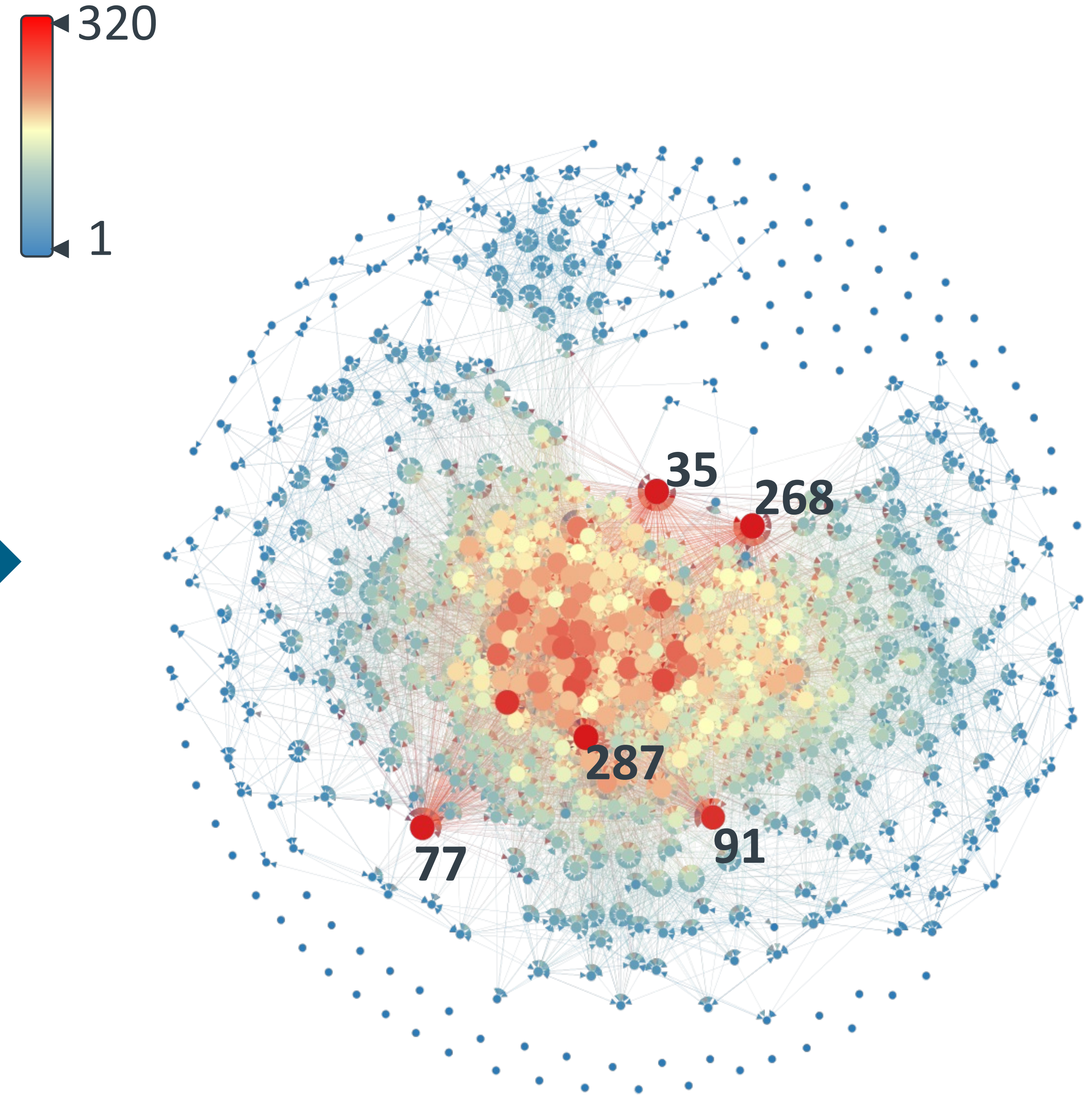
Case Study



- **System Name:** Divvy Bike
- **Location:** Chicago
- **Date of Operation Began:** June 2013

Period One Shared Mobility Network

# of Connections



May 2016 Binary Directed Trip Network
(# of Nodes: 535, # of Edges: 21221)

Period Two Shared Mobility Network

?

May 2017 Binary Directed Trip Network
(# of Nodes: 582, **# of Edges: ?**)

Data Preparation and Experiment Settings

➢Data preparation for GNN-based link prediction

- May 2016 trip network (node features, adjacency matrix)
- Total # of positive links in May 2016 + an equal # of sampled negative links (70% for training, 30% for validation)

Training Input

- May 2017 approximate trip network (node features, approximate adjacency matrix)

Predicting Input

- All of the positive and negative links in May 2017

Evaluation Input

➢Data preparation for ANN-based link prediction

- May 2016 node features
- Total # of positive links in May 2016 + an equal # of sampled negative links (70% for training, 30% for validation)

Training Input

- May 2017 node features

Predicting Input

- All of the positive and negative links in May 2017

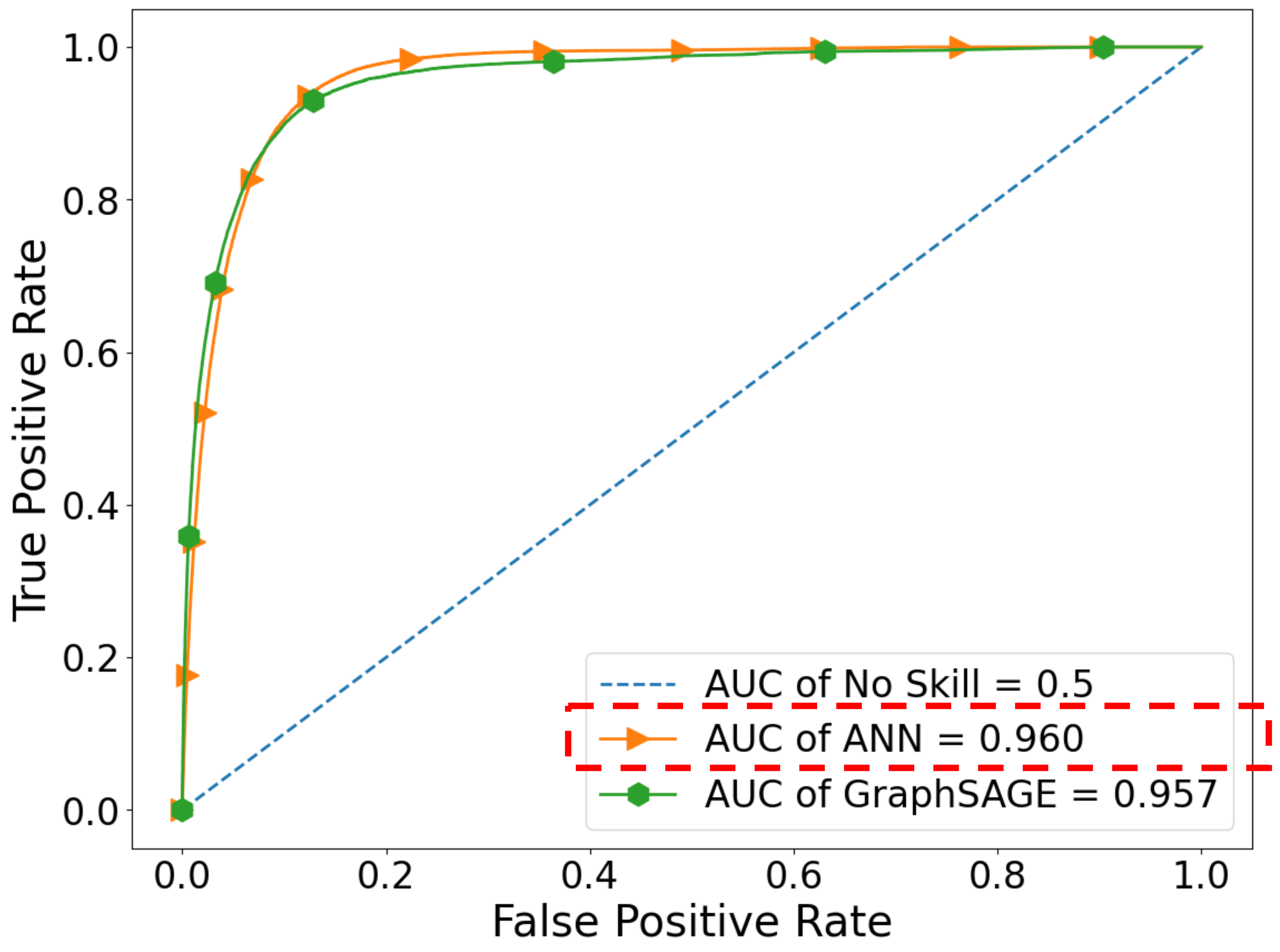
Evaluation Input

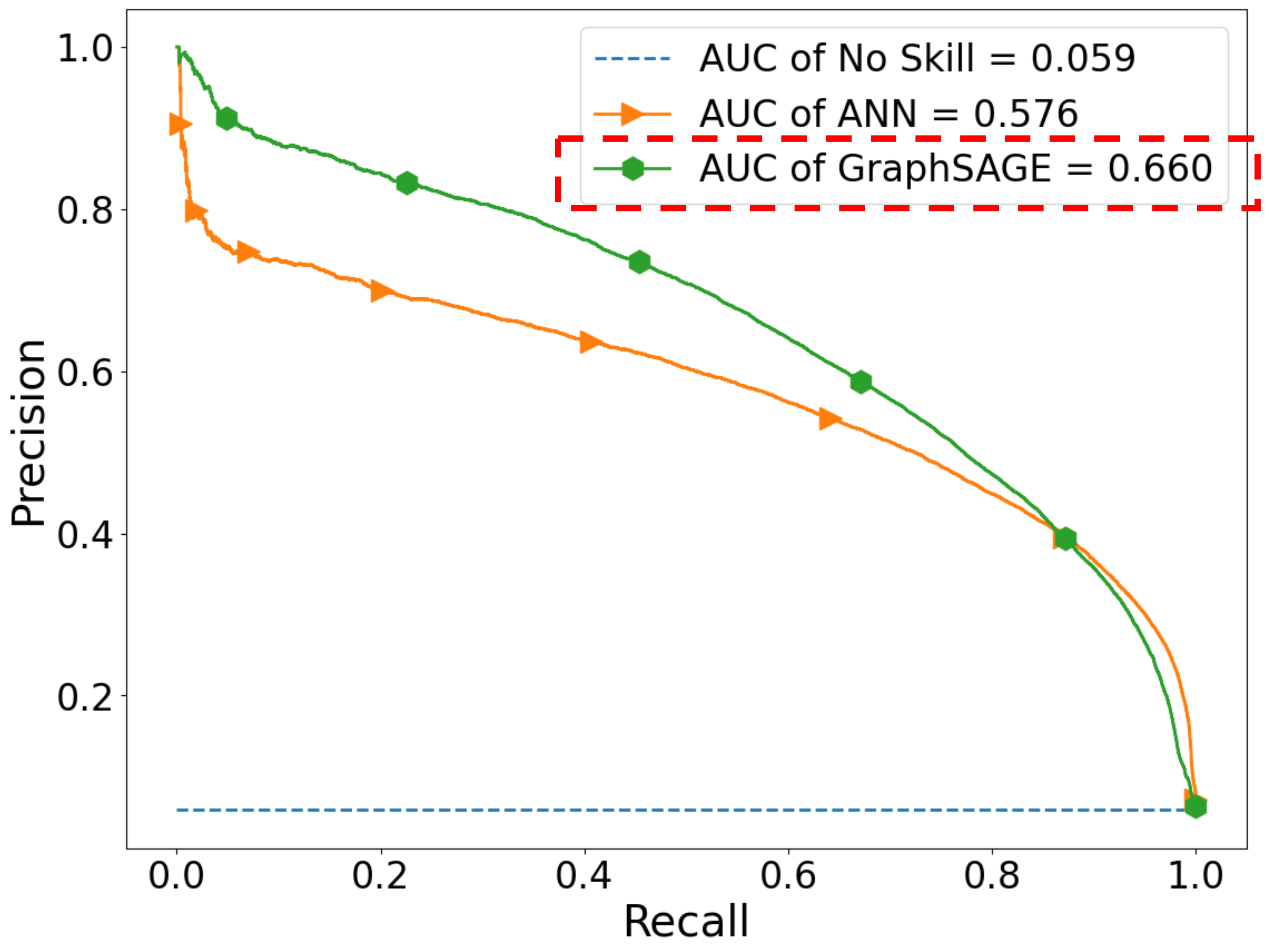
➢Experiment Parameter Settings

Setting Items	Model Applied	Value
Neighbor search depth	GraphSAGE	2
# of sampled in- and out-neighbors in two hops		10
Node embedding size		30
Input and hidden layer size for GraphSAGE		60
Input and hidden layer size for ANN	ANN	20
Minibatch size	GraphSAGE & ANN	192
Epoch		500
Learning rate		4e-4
Dropout		0

Result: GNN (GraphSAGE) VS ANN

Confusion Matrix (Probability threshold = 0.5)		**ANN Link Prediction** 0	**ANN Link Prediction** 1	**GNN (GraphSAGE) Link Prediction** 0	**GNN (GraphSAGE) Link Prediction** 1
Actual Class	0	278859 (TNR 87.61%)	39426 (FPR 12.39%)	278930 (TNR 87.64%)	39355 (FPR 12.36%)
Actual Class	1	1262 (FNR 6.36%)	18595 (TPR 93.64%)	1475 (FNR 7.43%)	18382 (TPR 92.57%)
F1-Score		0.478		0.474	

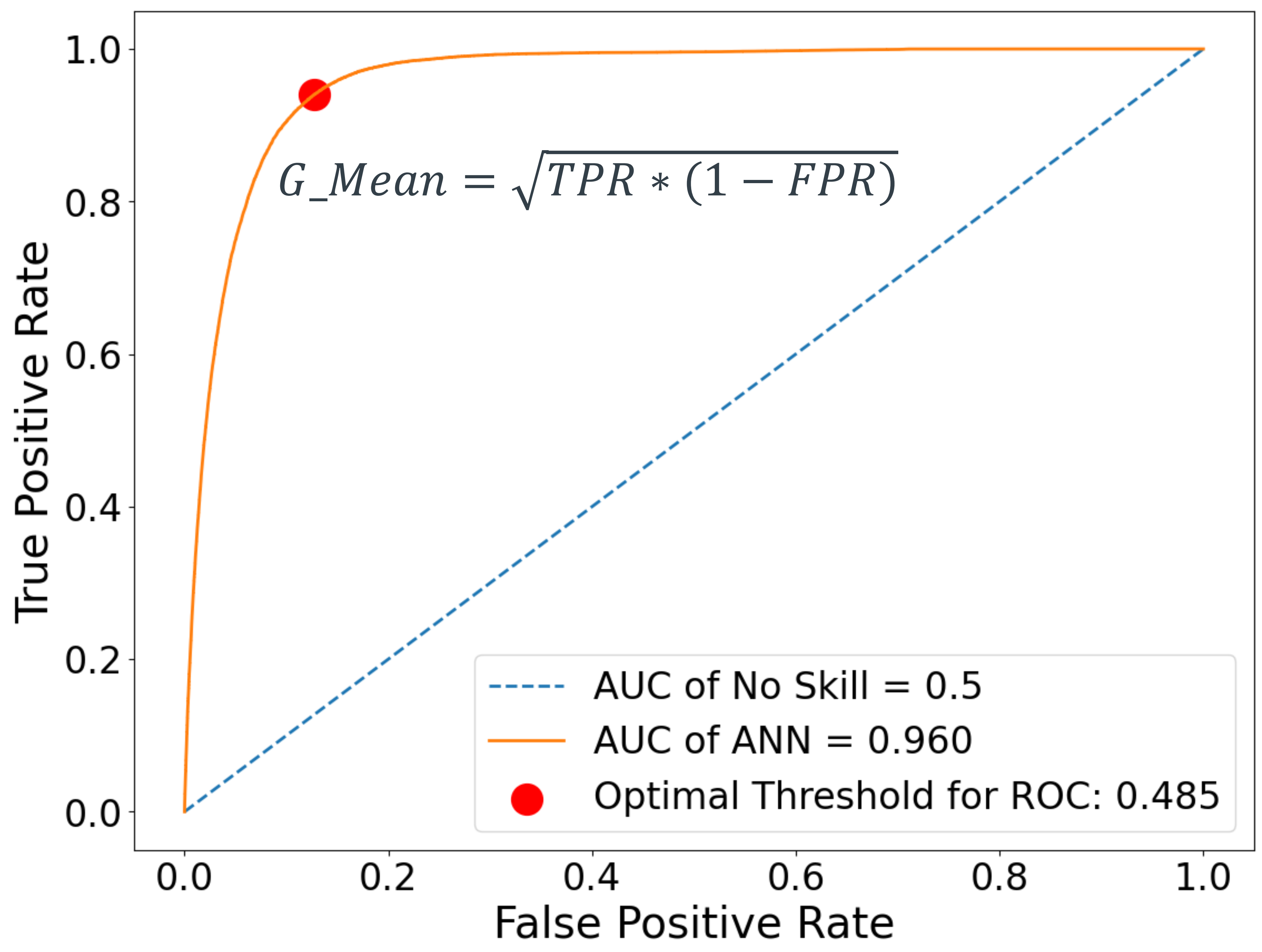


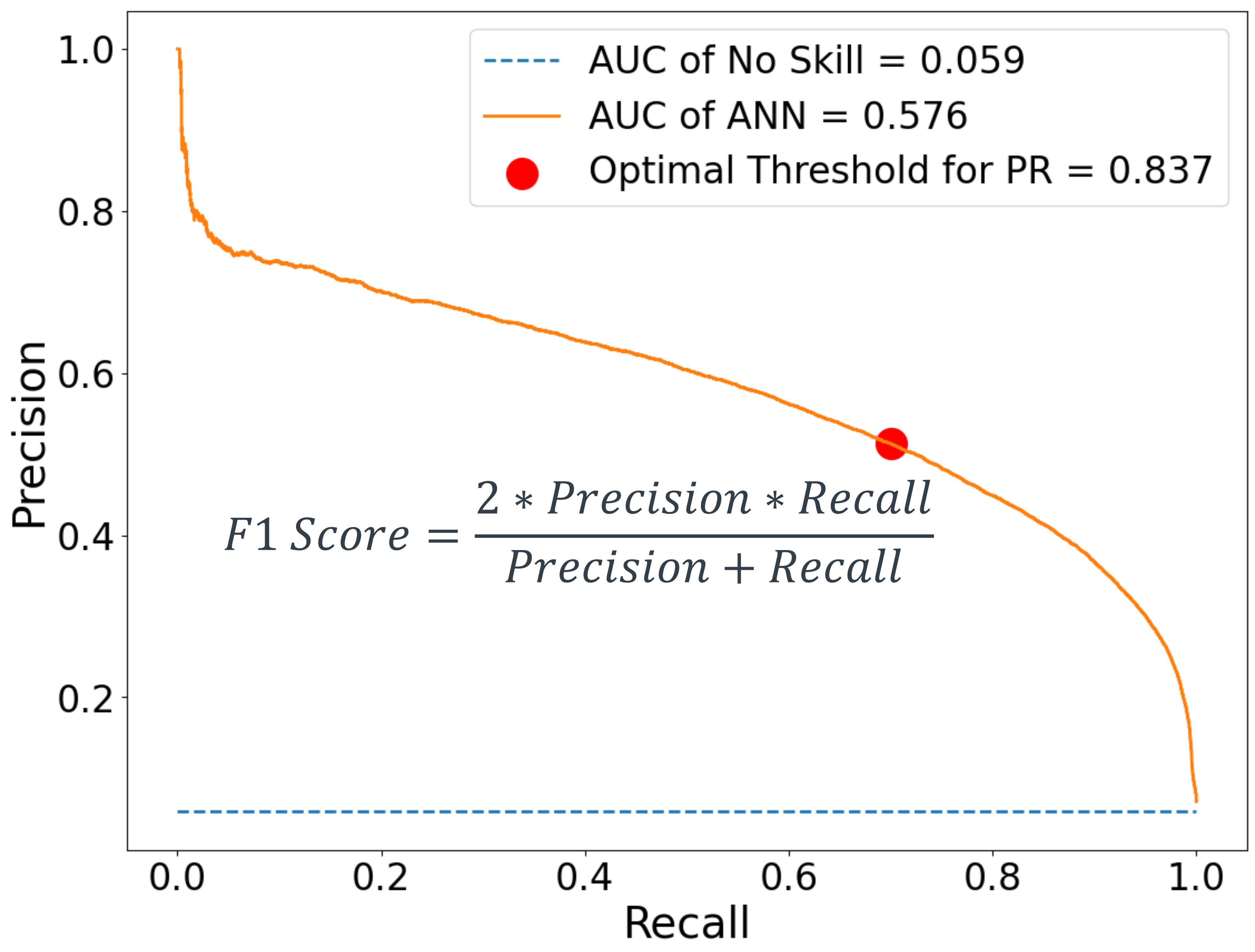


Findings

- When taking 0.5 as the probability threshold, GNN and ANN share a similar predictive power in both positive and negative categories.
- Similar F1-Score and ROC AUC (difference less than 0.005) further indicate both models' identical predictive power in both positive and negative categories.
- Higher PR AUC of GraphSAGE implies that neighborhood information can enhance the predictive power of the minority class (positive links) at an aggregated level.

Approximate Adjacency Lists By ANN





Optimal threshold for ROC: a balance between true positive and false positive rates.

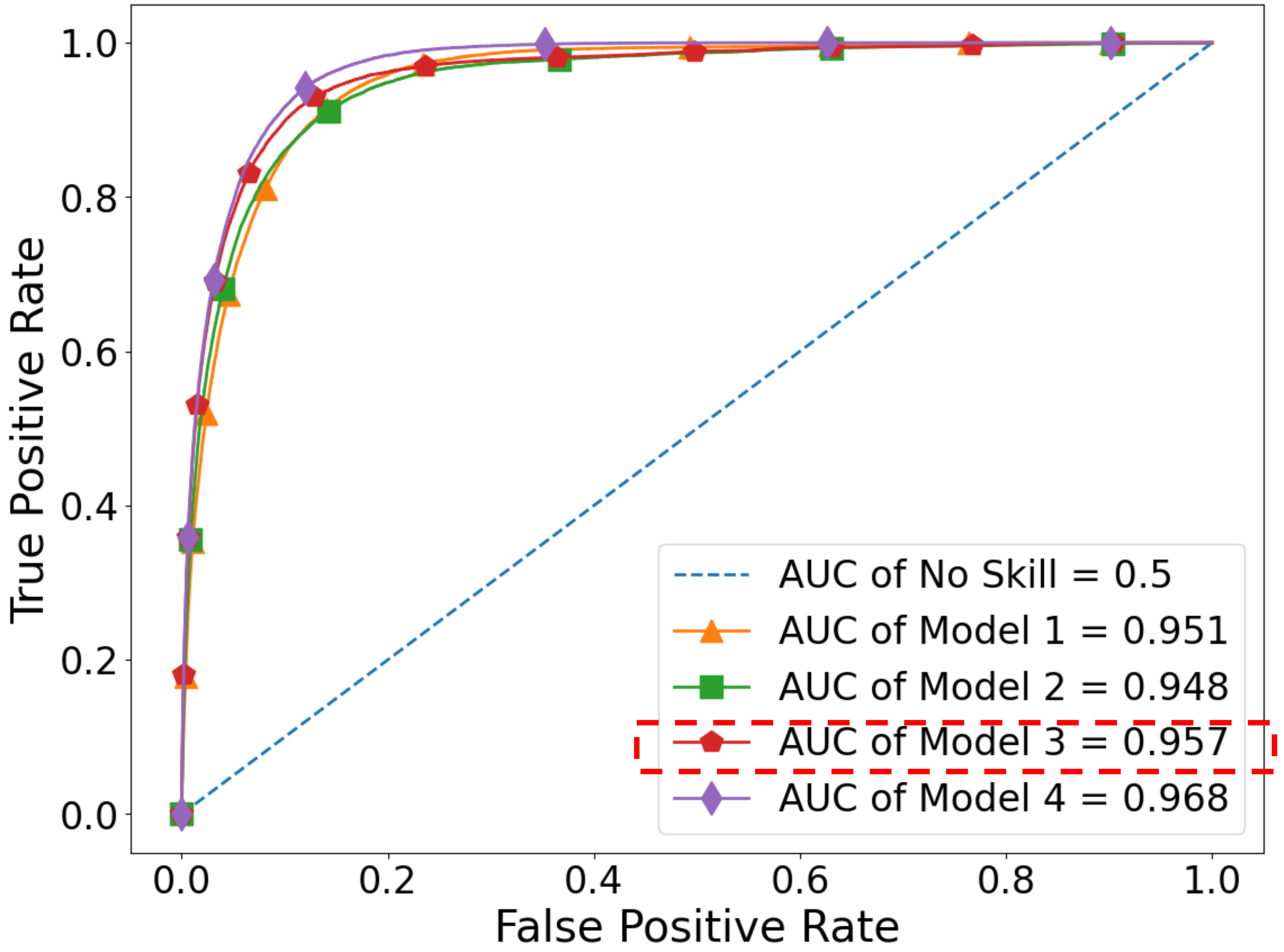
Optimal threshold for ROC: a balance between precision and recall.

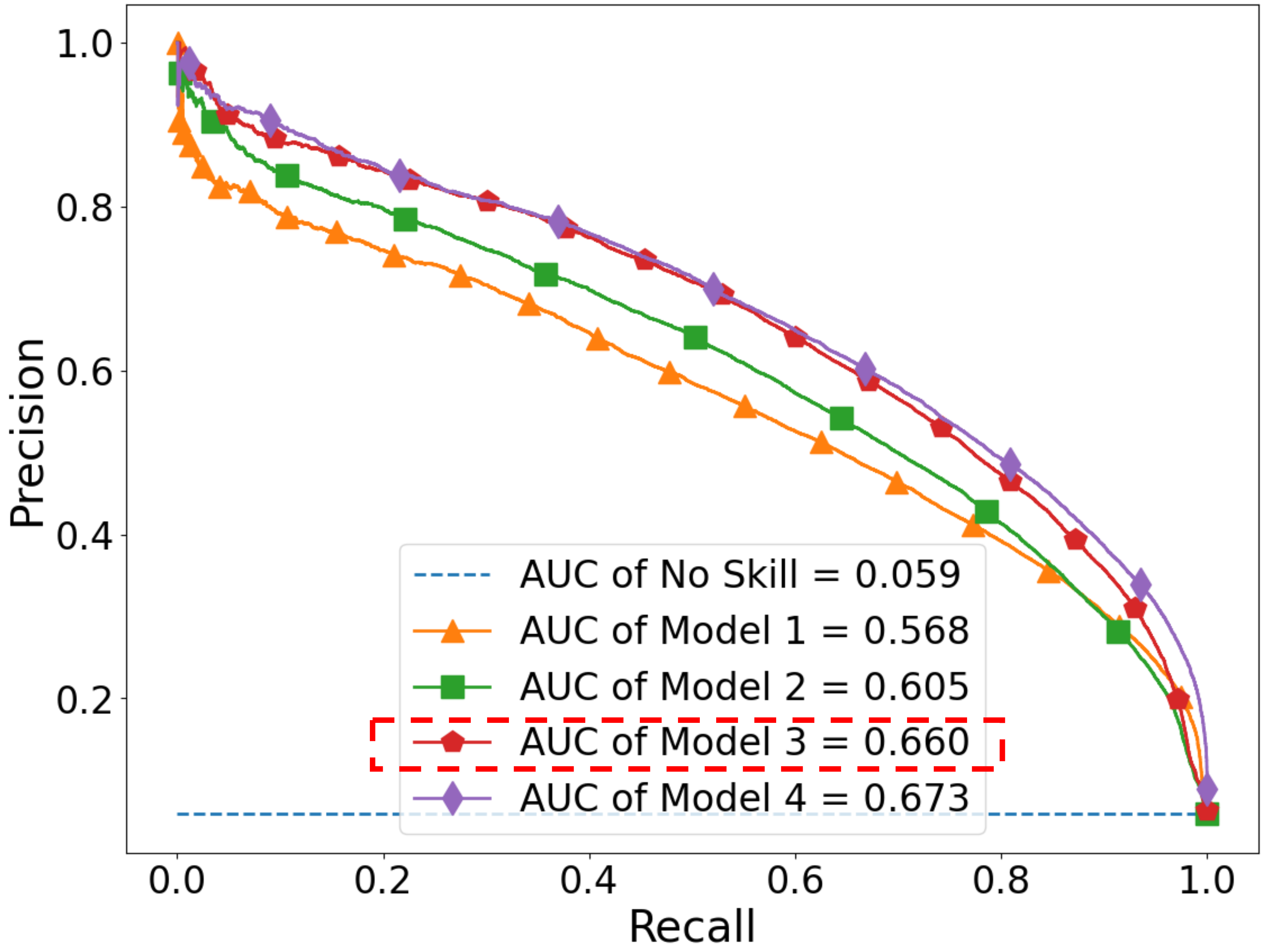
Result: GNN (GraphSAGE) With Different Approximate Adjacency Lists

Confusion Matrix (Probability threshold = 0.5)		**Model 1: ANN-Based Approximate Approach** (With Optimal Threshold for ROC)		**Model 2: ANN-Based Approximate Approach** (With Optimal Threshold for PR)	
		0	1	0	1
Actual Class	0	262343 (TNR 82.42%)	55942 (FPR 17.58%)	**282863 (TNR 88.87%)**	35422 (FPR 11.13%)
	1	1475 (FNR 5.52%)	18382 (TPR 94.48%)	2474 (FNR 12.46%)	17383 (TPR 87.54%)
F1-Score		0.397		**0.478**	

Confusion Matrix (Probability threshold = 0.5)		**Model 3: Modified May 2016 Trip Network**		**Model 4: Real May 2017 Trip Network** (Ground Truth)	
		0	1	0	1
Actual Class	0	278930 (TNR 87.64%)	39355 (FPR 12.36%)	278203 (TNR 87.41%)	40082 (FPR 12.59%)
	1	1475 (FNR 7.43%)	18382 (TPR 92.57%)	1042 (FNR 5.25%)	**18815 (TPR 94.75%)**
F1-Score		0.474		**0.478**	

Result: GNN (GraphSAGE) With Different Approximate Adjacency Lists





Findings

Model 3 shows higher ROC AUC and PR AUC than Model 1 and Model 2 indicating using the modified Period One network to approximate the neighbor information in the Period Two network is adequate for link prediction.

Summary

- We proposed a complex network-based approach based on GNN to predict travel demand between stations in shared mobility systems.
- By comparing to ANN, we revealed the importance of network neighboring information in travel demand prediction of shared mobility system.
- We tested different adjacency list approximate approaches and figured out taking previous year’s network structure to approximate next year’s node embedding can generate the best link prediction results for shared mobility system.

Referred Resources

- Section 1:
  - Neural Network: https://www.slideshare.net/KirillEremenko/deep-learning-az-artificial-neural-networks-ann-module-1
  - Activation Function: https://www.v7labs.com/blog/neural-networks-activation-functions
- Section 2:
  - A Gentle Introduction to Graph Neural Networks

    https://distill.pub/2021/gnn-intro/
  - Stanford CS224W: Machine Learning with Graphs

    http://snap.stanford.edu/class/cs224w-2019/
- Section 3: Travel Links Prediction In Shared Mobility Networks Using Graph Neural Network Models

  https://sidilab.files.wordpress.com/2022/05/bss_prediction_idetc_2022_final.pdf

THANK YOU.