Data Mining

Final Review

https://data-mining.github.io/winter-2026/

CS 453/553 – Winter 2026
Yu Wang, Ph.D.
Assistant Professor
Computer Science
University of Oregon

Neural Network Architecture

1. Dimension: input/output based on concrete application

2. Physical Meaning of Neural Network: WX, describe and draw the picture justifying the physical meaning

3. Why we need to have WX, WX+B, nonlinearity function

4. **Linear Regression**
    1. **Problem Formulation**
    2. **Understand Loss Function and how optimize the loss will lead to the desired effect we want**
    3. **Theoretical/Computational Optimization**
    4. **Local/Global Optimal?**

5. **Classification**
    1. **Binary Classification Loss, understand why optimizing the loss lead to the desired effect we want**
    2. **Multi-Class Classification Loss, understand why optimizing the loss lead to the desired effect we want**

Graph Mining

1. How to represent a graph-structured data typically in ML?
    1. Feature Matrix
    2. Graph Adjacency Matrix, Graph Adjacency List, Pros/Cons?

2. **Mathematically represent message-passing**
    1. **Understand the physical meaning**
    2. **Can compute concrete example**

3. Differences between Label Propagation, MLP and GNN in node classification
    1. Key assumption in label propagation and GNN
    2. Difference between label propagation and GNN

4. Mathematical Format of GNN

5. Differences between Heuristic Method, MLP, and GNN in link prediction
    1. Common Neighbor/Jaccard Similarity

Image Mining

1. How to represent an Image-structured data in computer?

2. **Problem of MLP for Image Classification**
    1. **Able to work for Image Translation? If not, why?**
    2. **Able to work for Image Rotation? If not, why?**

3. **Invariance and Equivariance, can you define and identify whether a specific operator is Invariance or Equivariance?**

4. CNN Architecture
    1. Kernal Convolution Operation
        1. Stride
        2. Padding
        3. Dimension Computation after applying Kernal Convolution
    2. Pooling (Max/Mean/…)

5. Explanation
    1. Principal of Grad-CAM
    2. Principal of Using Mask to Explain

Language Mining

1. Language Mining Application and how to represent an Language-structured data in computer?

2. **General Formulation of Next Token Generation**
    1. **Notation**
    2. **Conditional Distribution**
    3. Distribution Factorization

3. N-Gram Language Model
    1. Core Technique (Frequency Computing)
    2. Pros/Cons

4. RNN
    1. **Token Embedding, Word2Vec**
    2. **How to query token embedding given token ID?**
    3. **Architecture of RNN and Physical Meaning of Each Component**
    4. How we want to aggregate the output to do downstream task
        1. Last one?
        2. Every layer output?
    5. Problem with RNN: Gradient Vanish and Gradient Explosion

Transformer and LLM

1. Motivation of Transformation

2. **Attention:**
    1. **Motivation of Attention**
    2. **Physical Meaning of Query, Key and Value**
    3. **Master Every Detail and I will let you physical compute the forward process of Attention**
    4. **Multi-Head**
    5. **Causal Attention: Which Task?**
    6. **Cross Attention: Which Task?**
    7. **Self Attention: Which Task?**

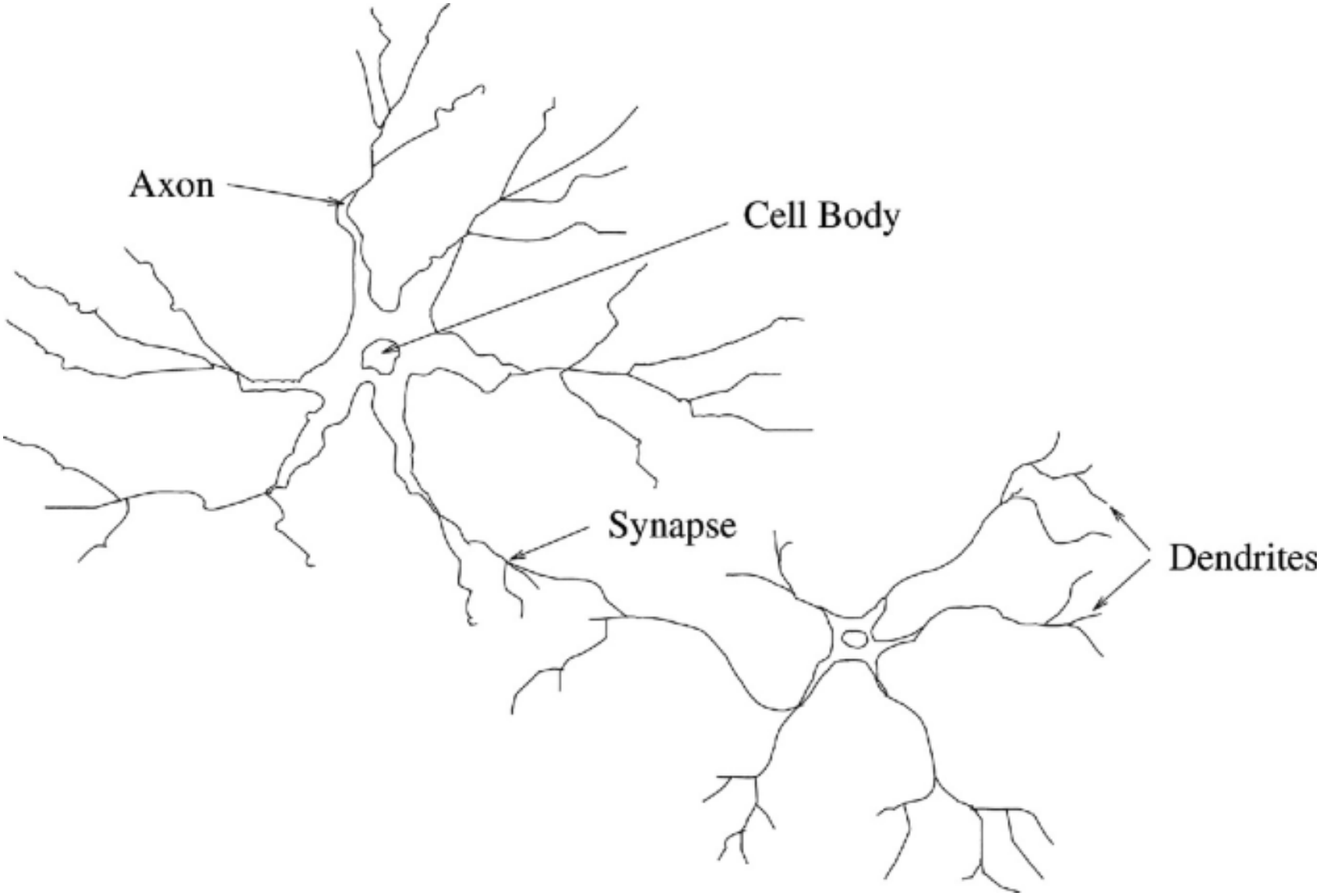
3. Positional Encoding

4. **What is auto-regressive generation?**

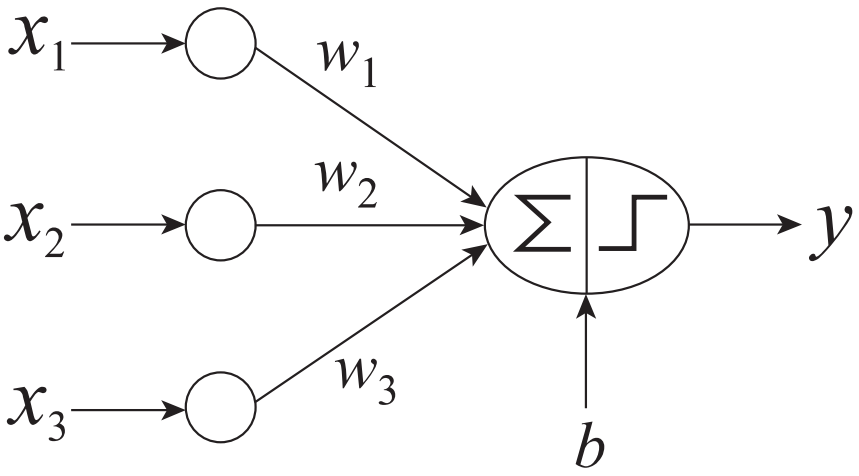
5. **What is beam search?**

6. How to train LLM, what is the loss? Physical Meaning

Biology Perceptron



Biology Perceptron



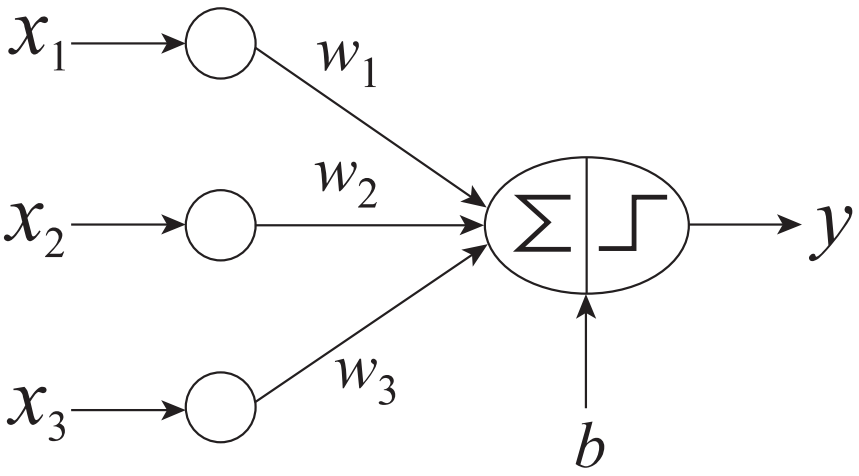
$$y = \sigma(w^{\mathrm{T}}x + b)$$

$$y = \sigma(\boldsymbol{w}^{\mathrm{T}}x + b) \qquad \textbf{Linear Transformation}$$

$$y = \sigma(w^{\mathrm{T}}x + \boldsymbol{b}) \qquad \textbf{Bias}$$

$$y = \boldsymbol{\sigma}(w^{\mathrm{T}}x + b) \qquad \textbf{Nonlinear Activation}$$

Biology Perceptron



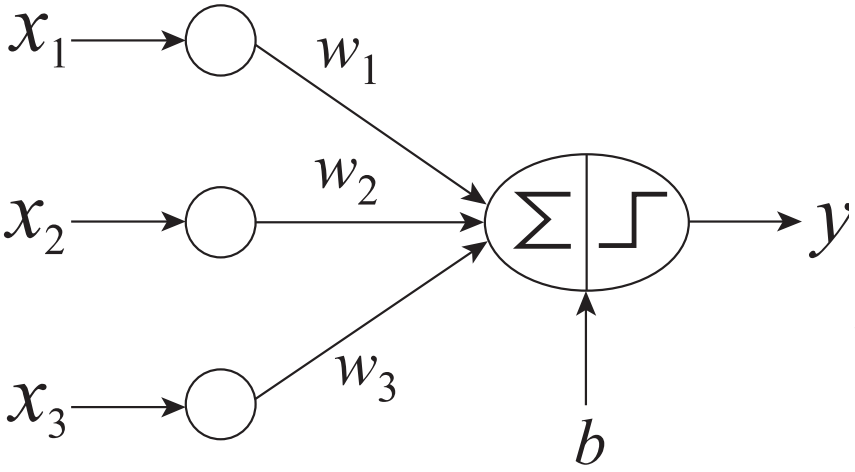
$$y = \sigma(w^{\mathrm{T}}x + b)$$

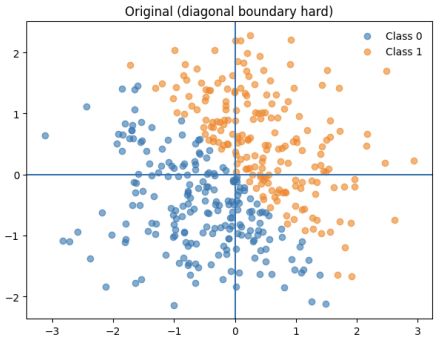
$$y = \sigma(\boldsymbol{w}^{\mathrm{T}}x + b) \qquad \textbf{Linear Transformation}$$

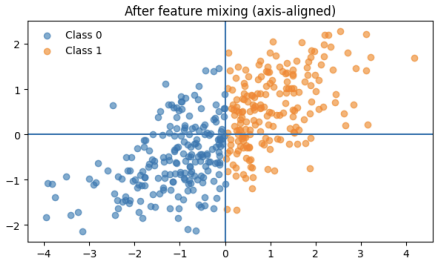
$$y = \sigma(w^{\mathrm{T}}x + \boldsymbol{b}) \qquad \textbf{Bias}$$

$$y = \boldsymbol{\sigma}(w^{\mathrm{T}}x + b) \qquad \textbf{Nonlinear Activation}$$

Biology Perceptron





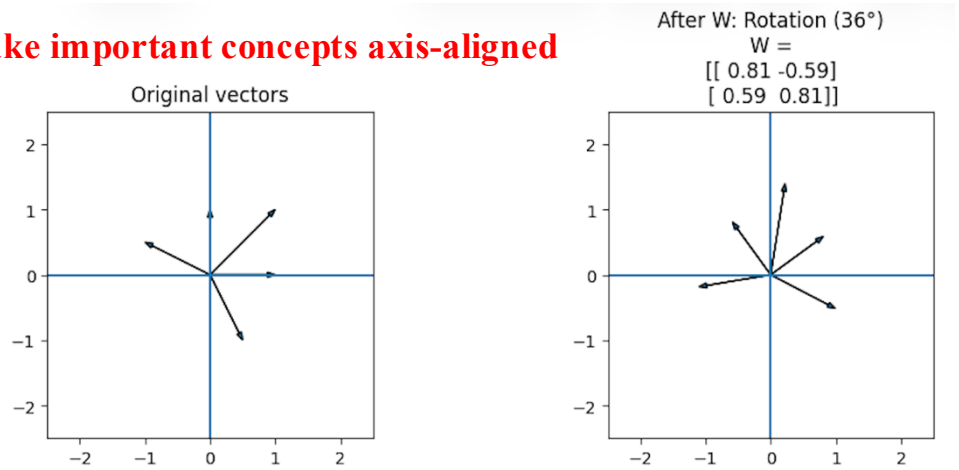


$$y = \sigma(\boldsymbol{w}^{\mathrm{T}} x + b)$$

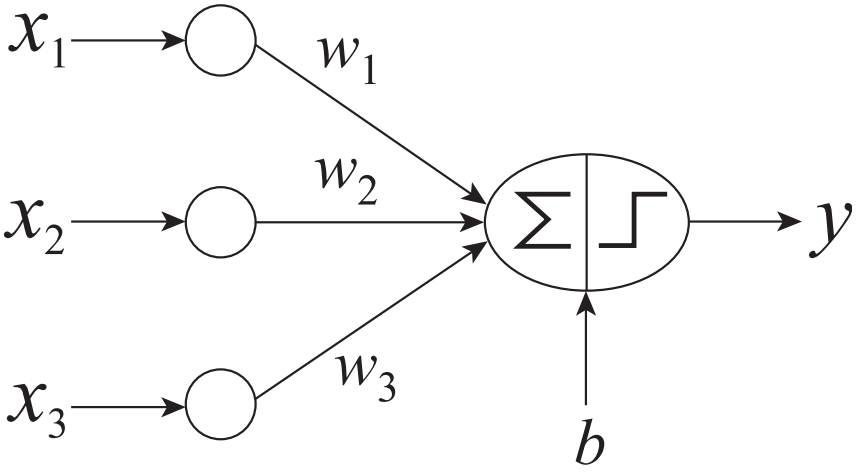
- Rotates the input space
- Scales directions differently
- Mixes features

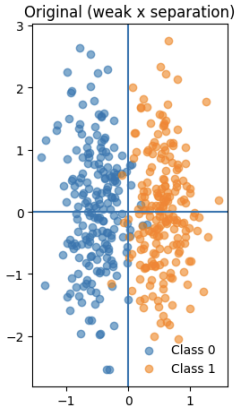
Linear Transformation

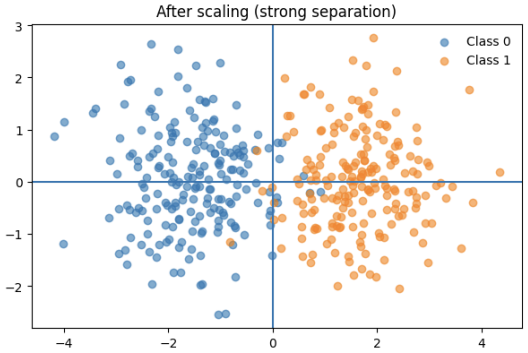
Make important concepts axis-aligned



Biology Perceptron





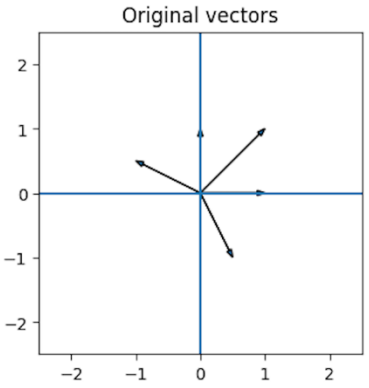


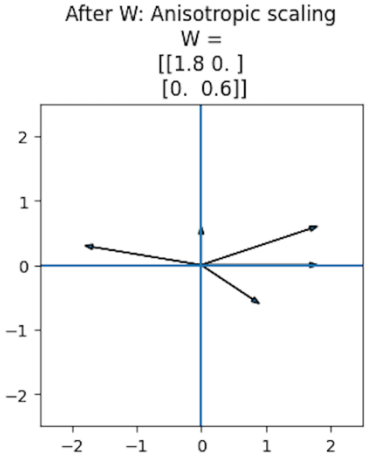
$$y = \sigma(\boldsymbol{w}^{\mathrm{T}} x + b)$$

Linear Transformation

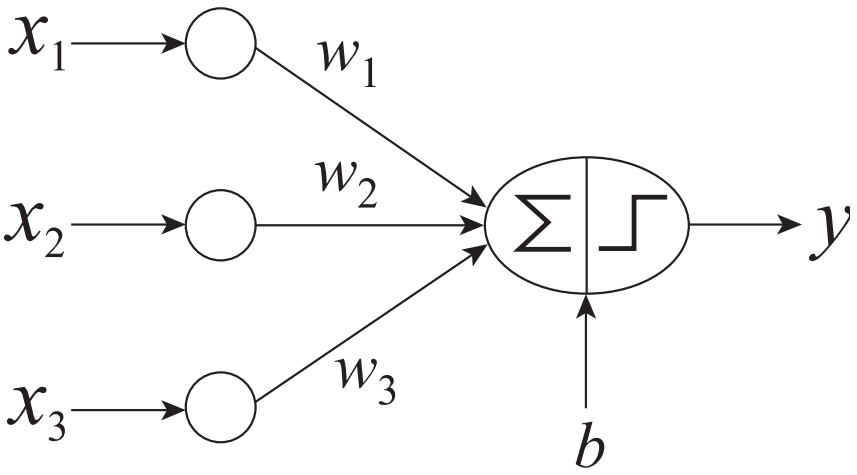
Assigning importance weights to features
Controlling sensitivity of neurons

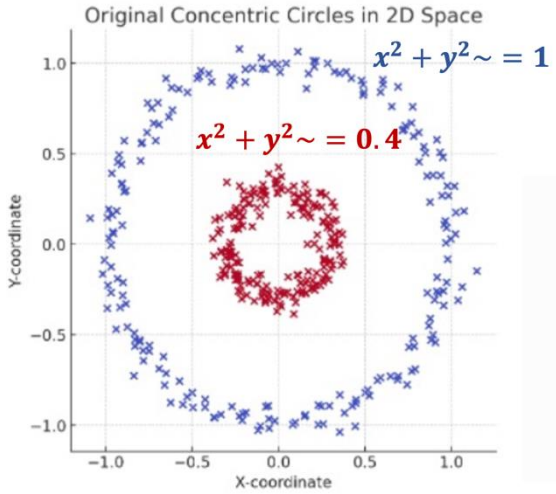
- Rotates the input space
- Scales directions differently
- Mixes features

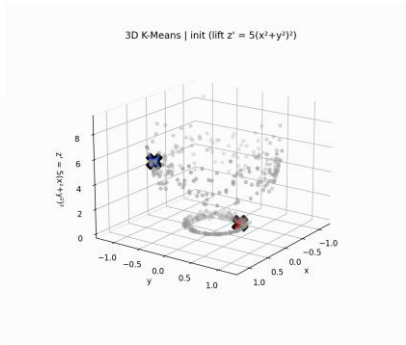




Biology Perceptron





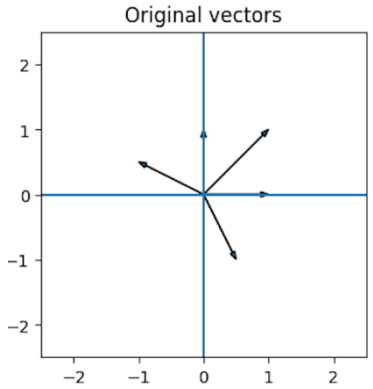


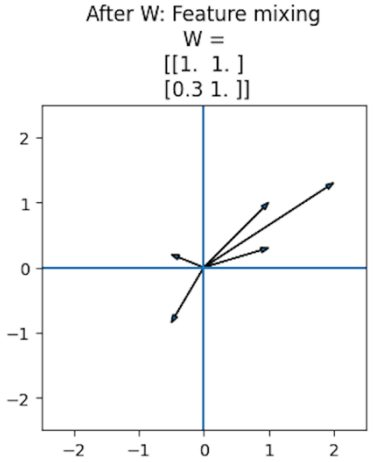
$$y = \sigma(\boldsymbol{w}^{\mathrm{T}} x + b)$$

- Rotates the input space
- Scales directions differently
- Mixes features

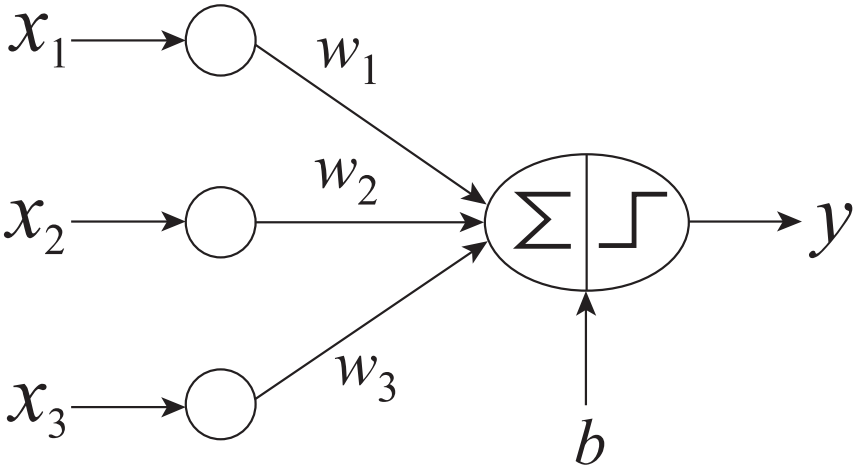
Linear Transformation

neurons don't look at one raw feature
they respond to combinations





Biology Perceptron

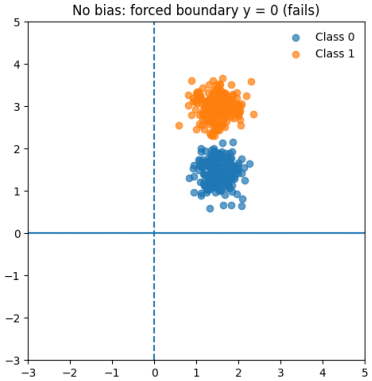


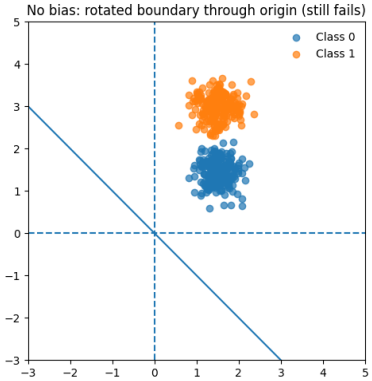
$$y = \sigma(w^{\mathrm{T}}x + \boldsymbol{b})$$ **Bias** $$y = w^{\mathrm{T}}x = 0$$

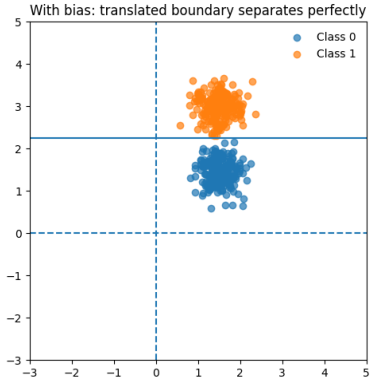
$$\mathrm{ReLU}(z) = \max(0, z)$$

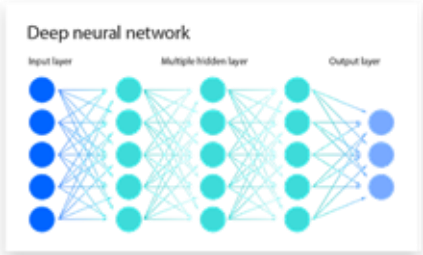
$$\mathbb{E}[z] = 0$$

$$\mathbb{E}[\mathrm{ReLU}(z)] > 0$$

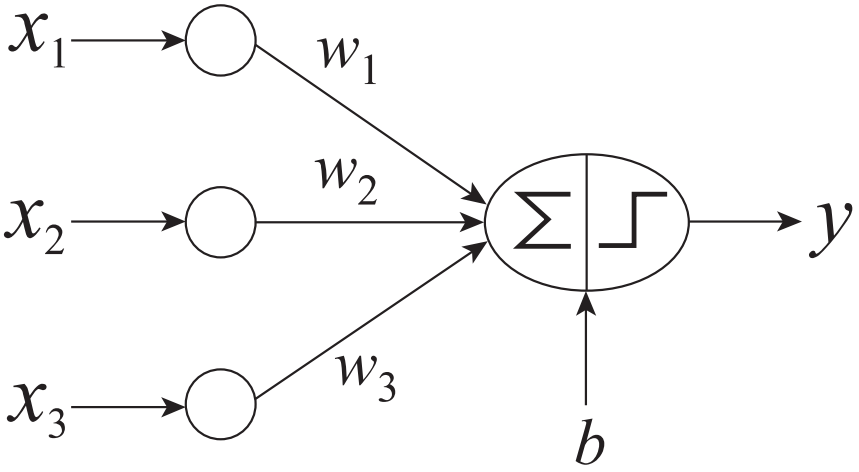




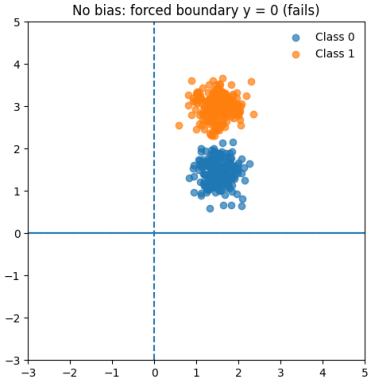


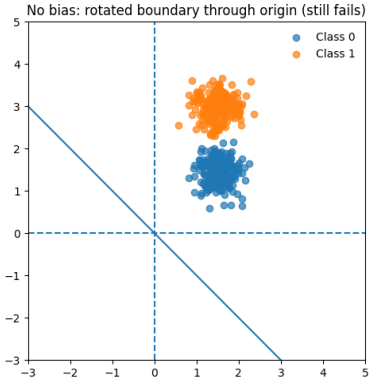


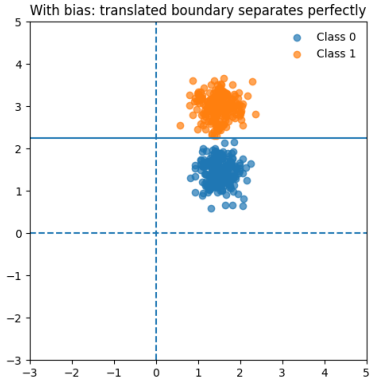
Biology Perceptron

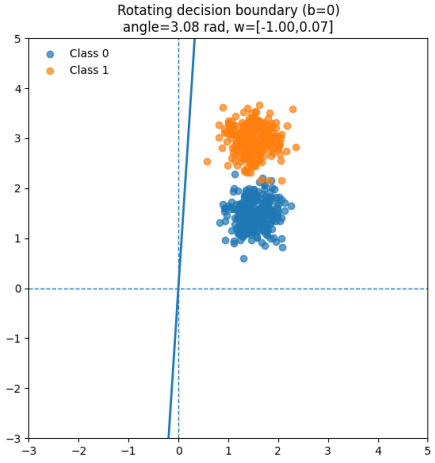


$$y = \sigma(w^{\mathrm{T}}x + \boldsymbol{b})$$ **Bias** $$y = w^{\mathrm{T}}x = 0$$

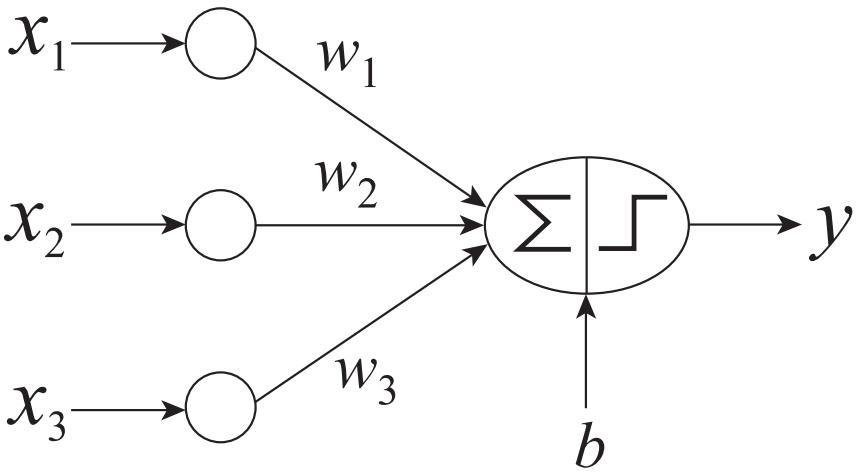








Biology Perceptron



$$y = \sigma(w^{\mathrm{T}}x + \boldsymbol{b})$$

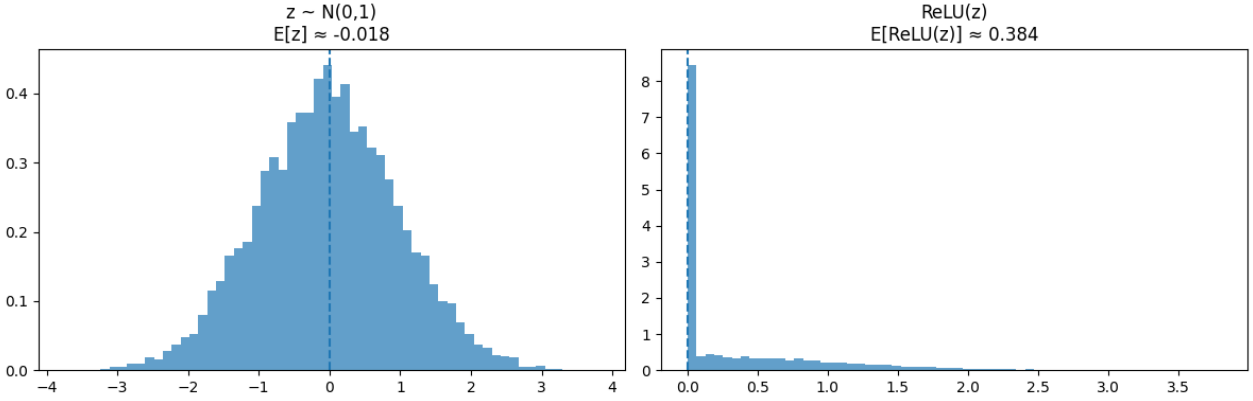
Bias

$$y = w^{\mathrm{T}}x = 0$$

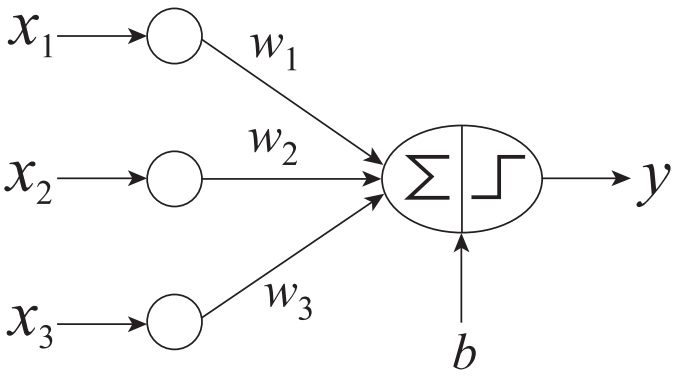
$$\mathrm{ReLU}(z) = \max(0, z)$$

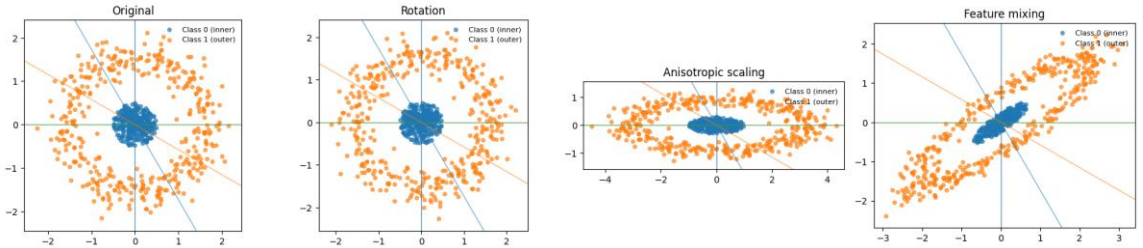
$$\mathbb{E}[z] = 0$$

$$\mathbb{E}[\mathrm{ReLU}(z)] > 0$$



Biology Perceptron





$$y = \sigma(w^{\mathrm{T}}x + b)$$

Nonlinear Activation

$$W_3(W_2(W_1x + b_1) + b_2) + b_3 = \tilde{W}x + \tilde{b}$$

Any neural network composed only of linear transformations and bias is equivalent to a single linear model.

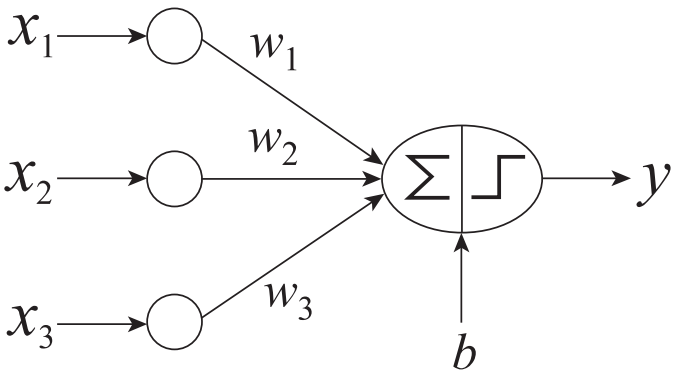
Linear + bias can:

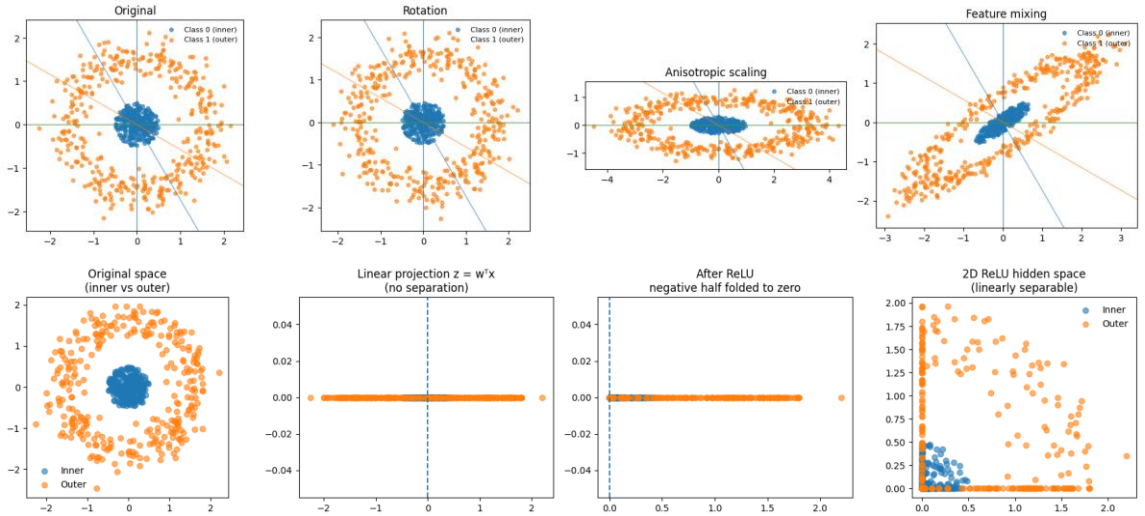
- Rotate space
- Stretch space
- Shift boundaries

But it **cannot bend space**.

- Lines → lines
- Planes → planes
- Half-spaces → half-spaces

Biology Perceptron

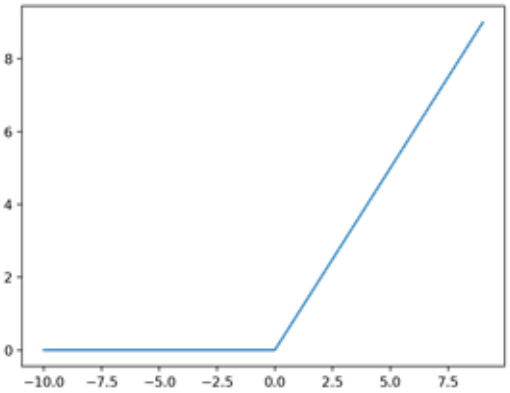




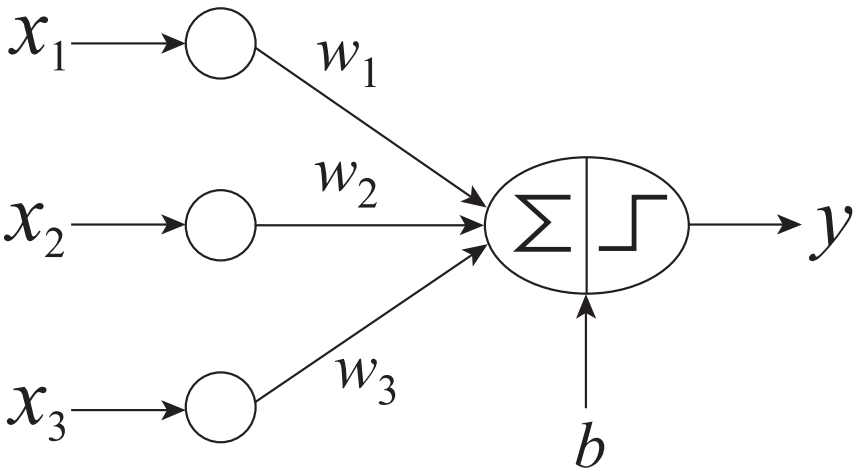
$$y = \sigma(w^{\mathrm{T}}x + b)$$ **Nonlinear Activation**

$$W_3(W_2(W_1x + b_1) + b_2) + b_3 = \tilde{W}x + \tilde{b}$$

Any neural network composed only of linear transformations and bias is equivalent to a single linear model.



Biology Perceptron



$$y = w^{\mathrm{T}}x + b$$

W takes the entire input space and rotates, stretches, compresses, shears, and possibly projects it into a new space.

How much does x align with pattern w?

Price = 2*Bed + 2*Living + 1*Bath

ML = 2*Agent + 2*RL

Musk = 1*pixel(0,0)+2*pixel(0, 1) + …

One Layer Neural Network for Linear Regression

- Data - $\left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$
- Regression – Find f that minimizes our uncertainty about y given x

$$y = f(x)$$

- Minimizing Mean Squared Error = Minimizing Negative Log-Likelihood

$$\underset{f}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y^{(i)} - f\left(x^{(i)}\right)\right)^2$$

Binary Classification

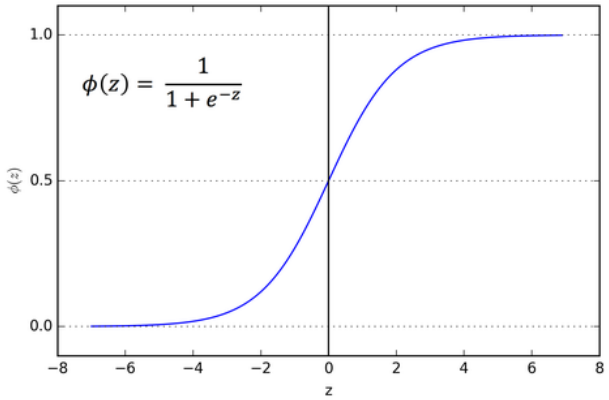
Data | **Model** | **Loss** | **Optimization**

Used for **binary classification** where $y_i \in \{0, 1\}$ and $\hat{y}_i = \sigma(z_i) \in (0, 1)$.

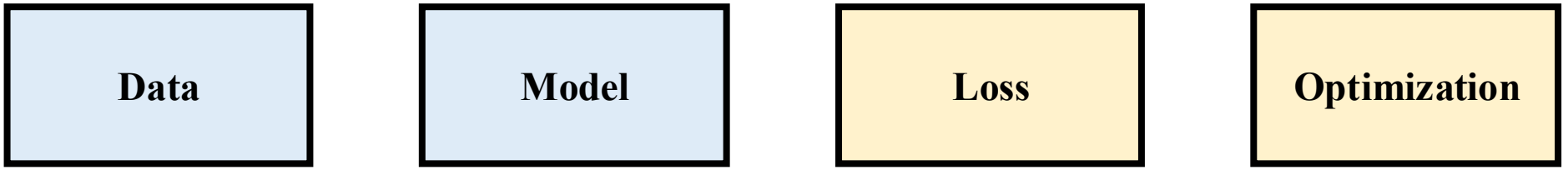
$$\mathcal{L}_{\text{BCE}} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

Vector form:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{n}\left(\mathbf{y}^T \log \hat{\mathbf{y}} + (1 - \mathbf{y})^T \log(1 - \hat{\mathbf{y}})\right)$$



Multi-Class Classification

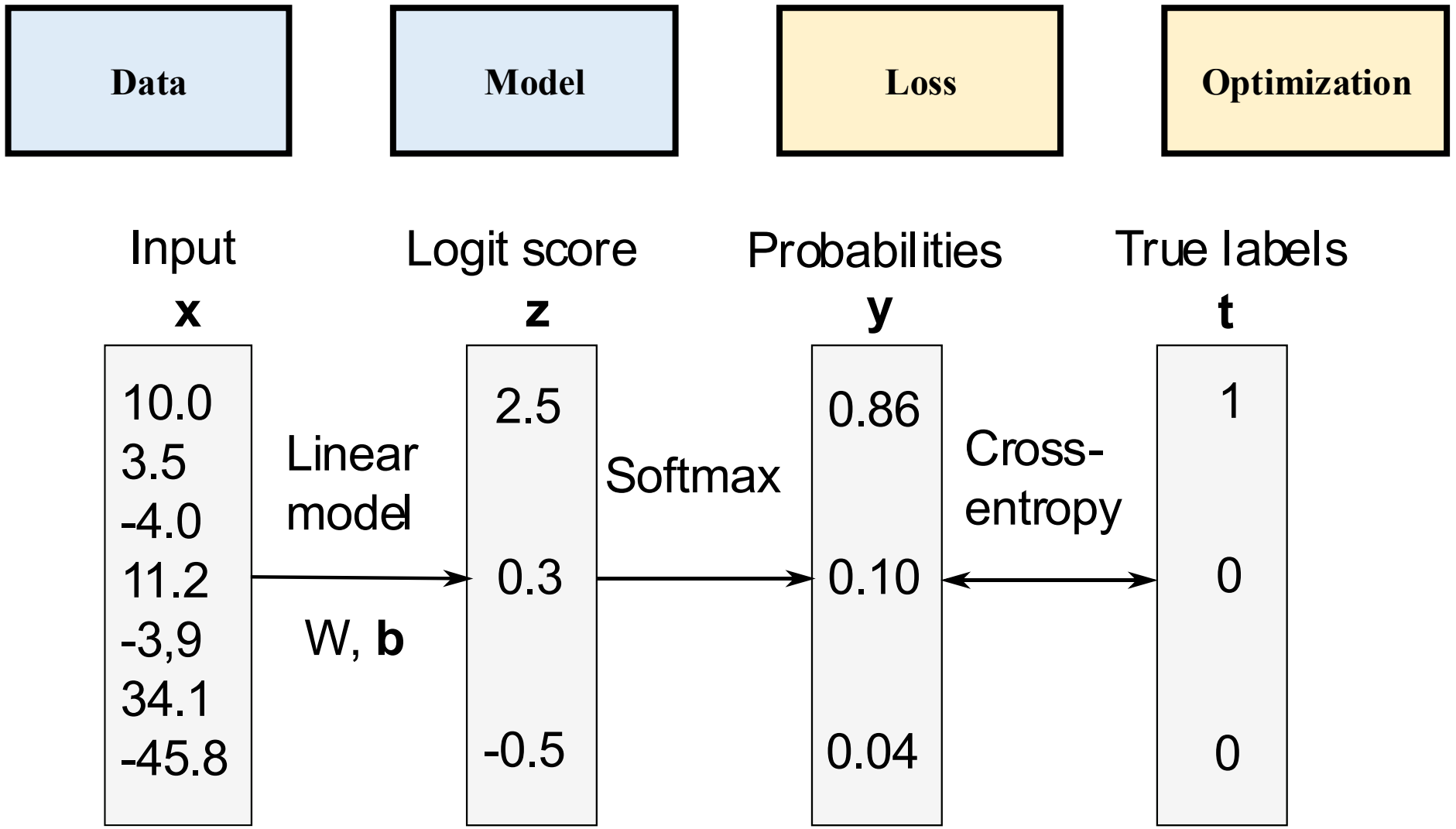


Multi-Class Classification

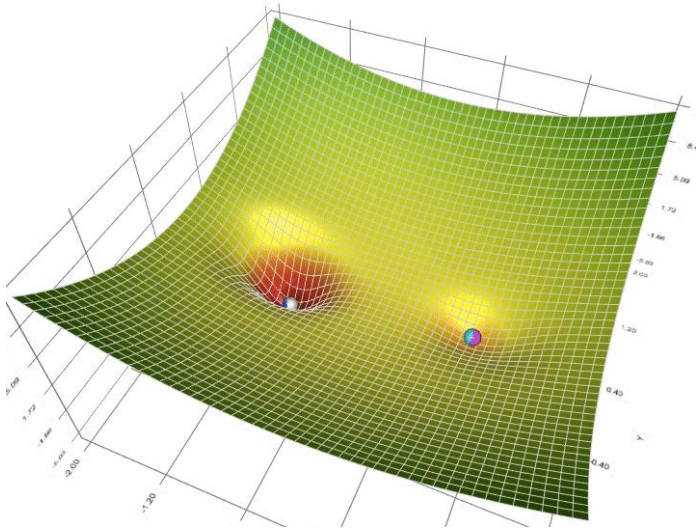
$$\mathcal{L}_{\text{CE}} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} y_{ic}\log(\hat{y}_{ic})$$

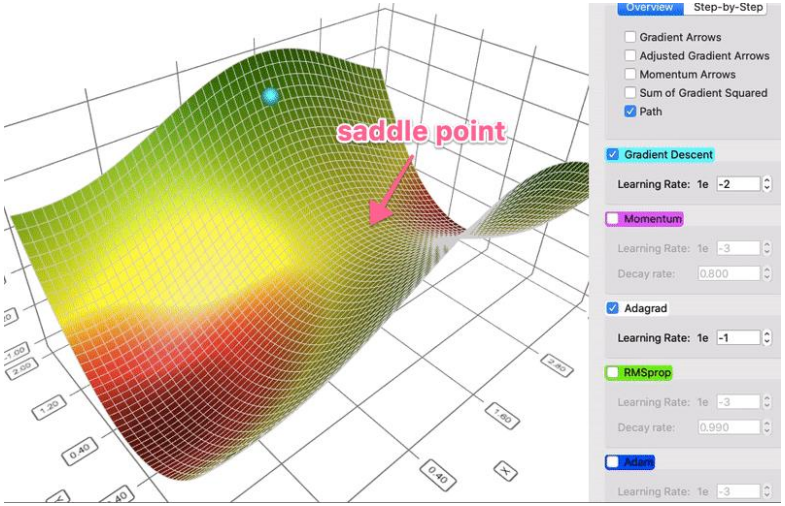
$$\hat{y}_{ic} = \frac{e^{z_{ic}}}{\sum_{k=1}^{C} e^{z_{ik}}}$$

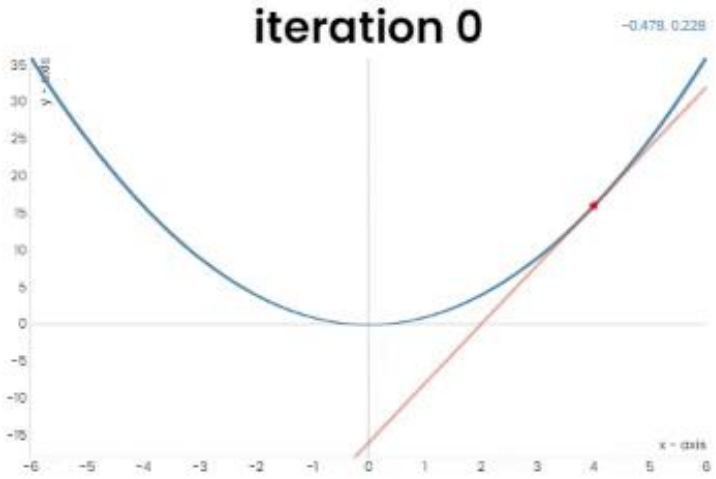
Multi-Class Classification



Local/Global Optimal







Graph Mining

1. How to represent a graph-structured data typically in ML?
    1. Feature Matrix
    2. Graph Adjacency Matrix, Graph Adjacency List, Pros/Cons?

2. **Mathematically represent message-passing**
    1. **Understand the physical meaning**
    2. **Can compute concrete example**

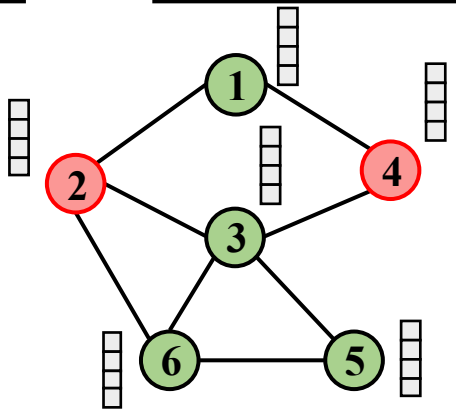
3. Differences between Label Propagation, MLP and GNN in node classification
    1. Key assumption in label propagation and GNN
    2. Difference between label propagation and GNN

4. Mathematical Format of GNN

5. Differences between Heuristic Method, MLP, and GNN in link prediction
    1. Common Neighbor/Jaccard Similarity

Graph Representation

Data | **Model** | **Loss** | **Optimization**




```
from torch_geometric.datasets import Planetoid
```

Dataset

```
dataset = Planetoid(root='/tmp/Cora', name='Cora')
```

```
dataset[0]
```

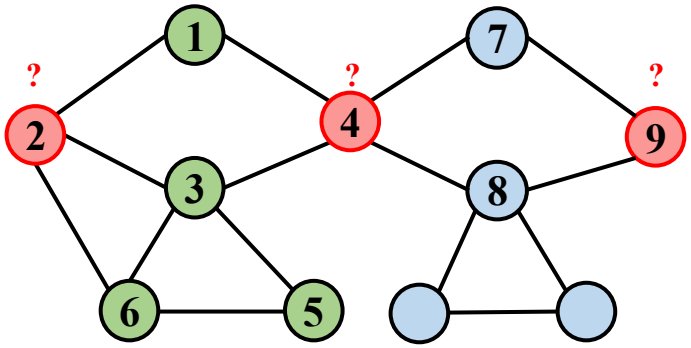
```
Data(x=[2708, 1433], edge_index=[2, 10556], y=[2708], train_mask=[2708], val_mask=[2708], test_mask=[2708])
```

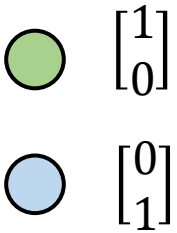
Feature + **Label** = **MLP**

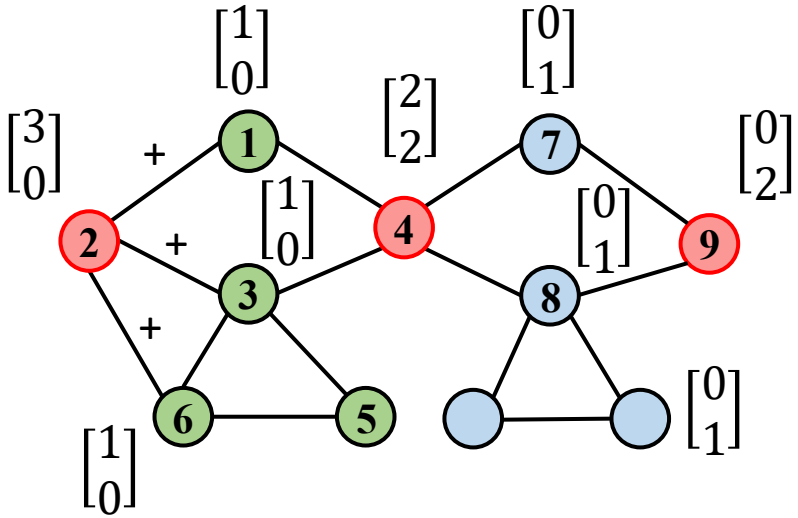
Graph + **Label** = **Label Propagation**

Feature + **Label** + **Graph** = **GNN**

Node Classification – Label Propagation







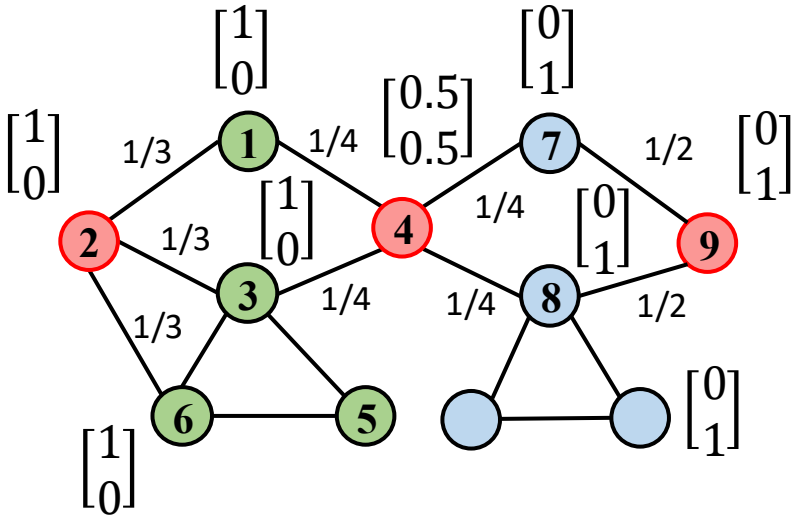


Image Mining

1. How to represent an Image-structured data in computer?

2. **Problem of MLP for Image Classification**
    1. **Able to work for Image Translation? If not, why?**
    2. **Able to work for Image Rotation? If not, why?**

3. **Invariance and Equivariance, can you define and identify whether a specific operator is Invariance or Equivariance?**

4. CNN Architecture
    1. Kernal Convolution Operation
        1. Stride
        2. Padding
        3. Dimension Computation after applying Kernal Convolution
    2. Pooling (Max/Mean/…)

5. Explanation
    1. Principal of Grad-CAM
    2. Principal of Using Mask to Explain

Image

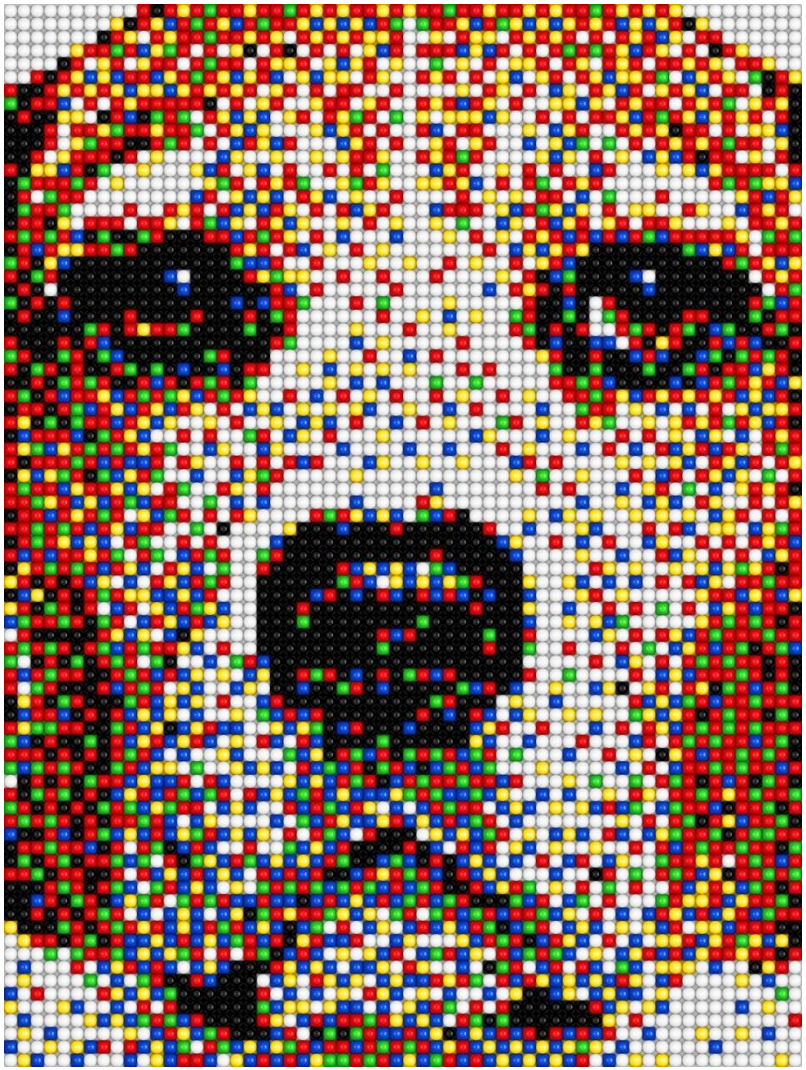
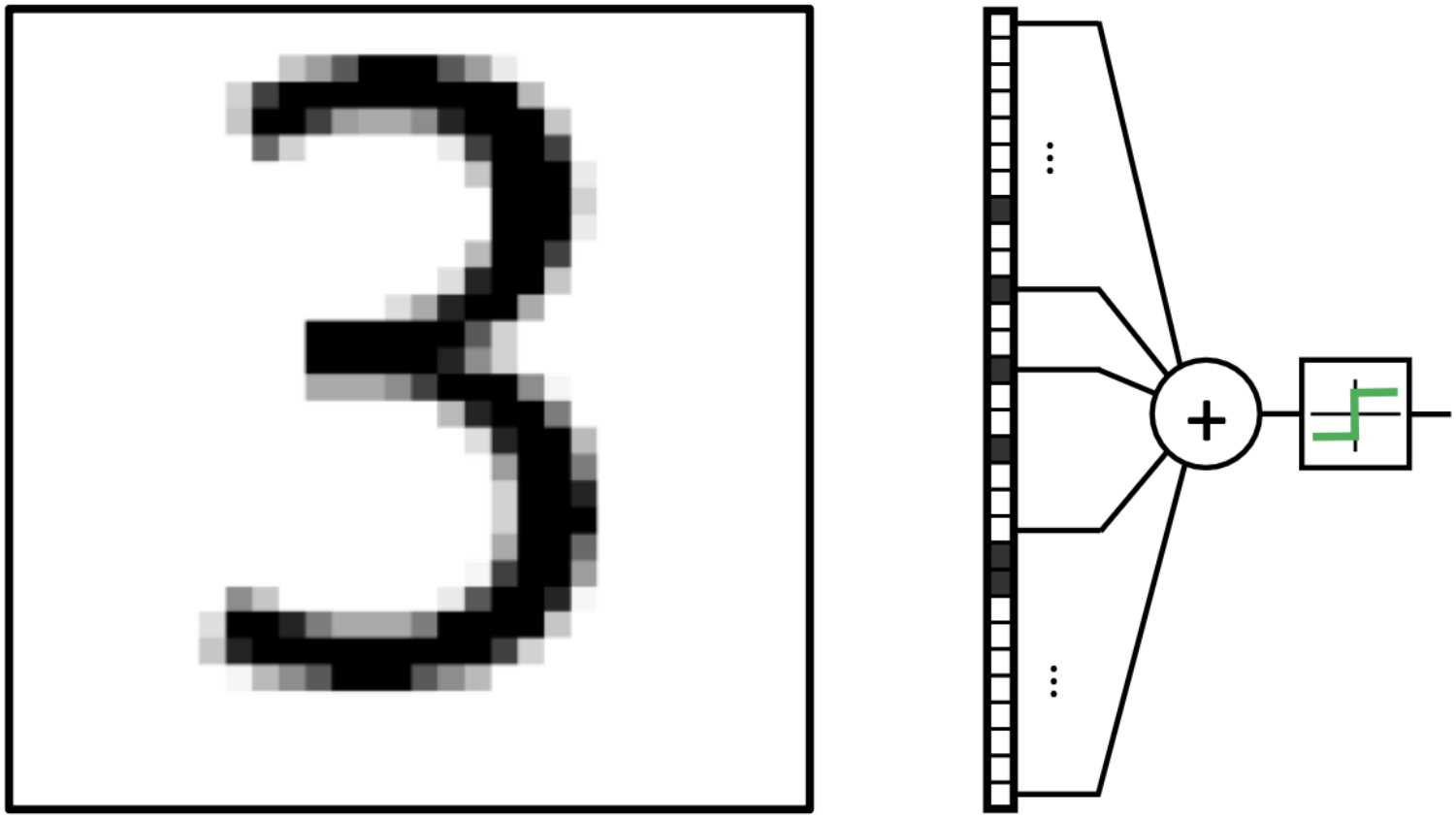


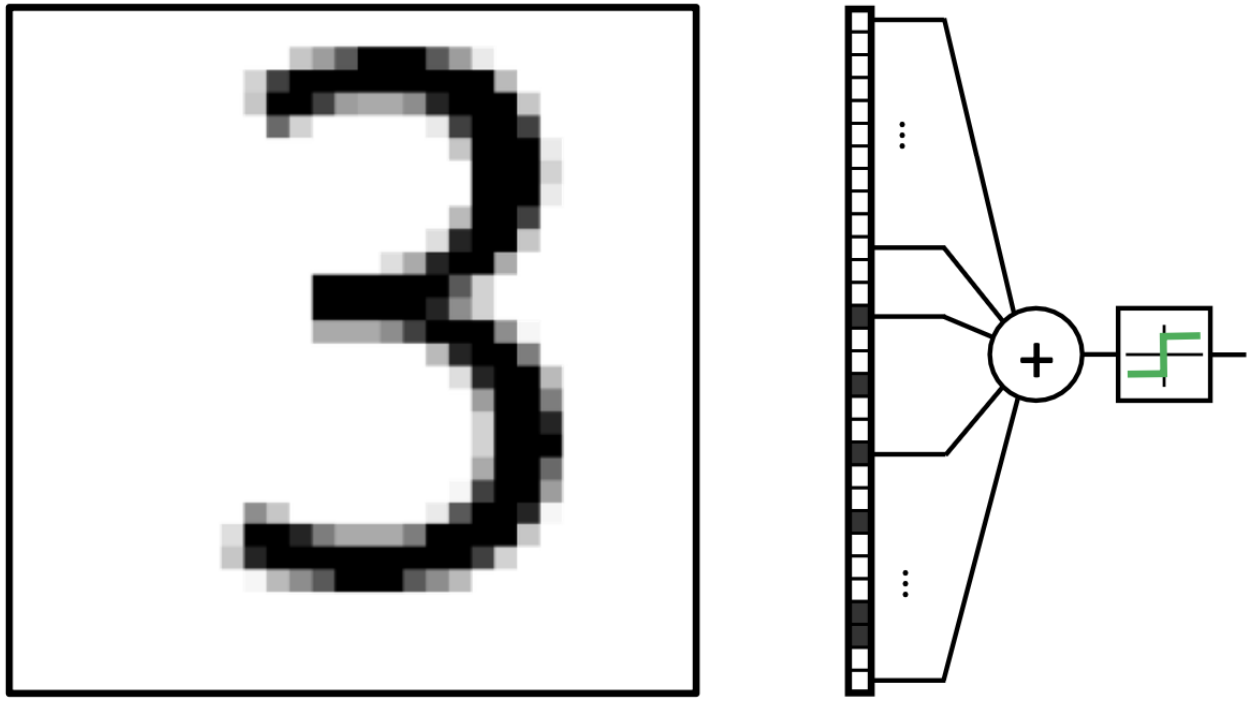
Image Mining



input image

input vector

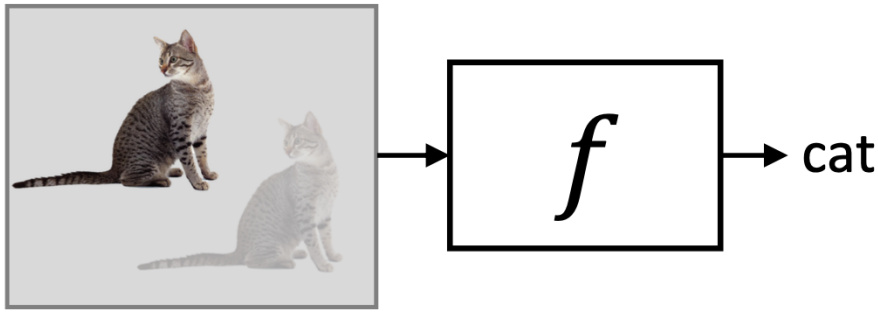
Image Mining



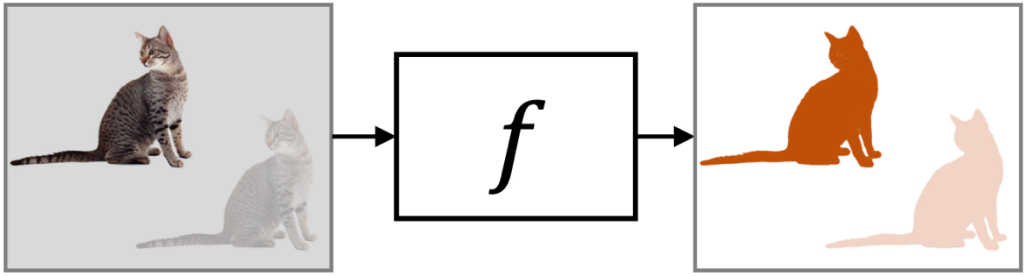
input image input vector

must learn shift invariance from data!

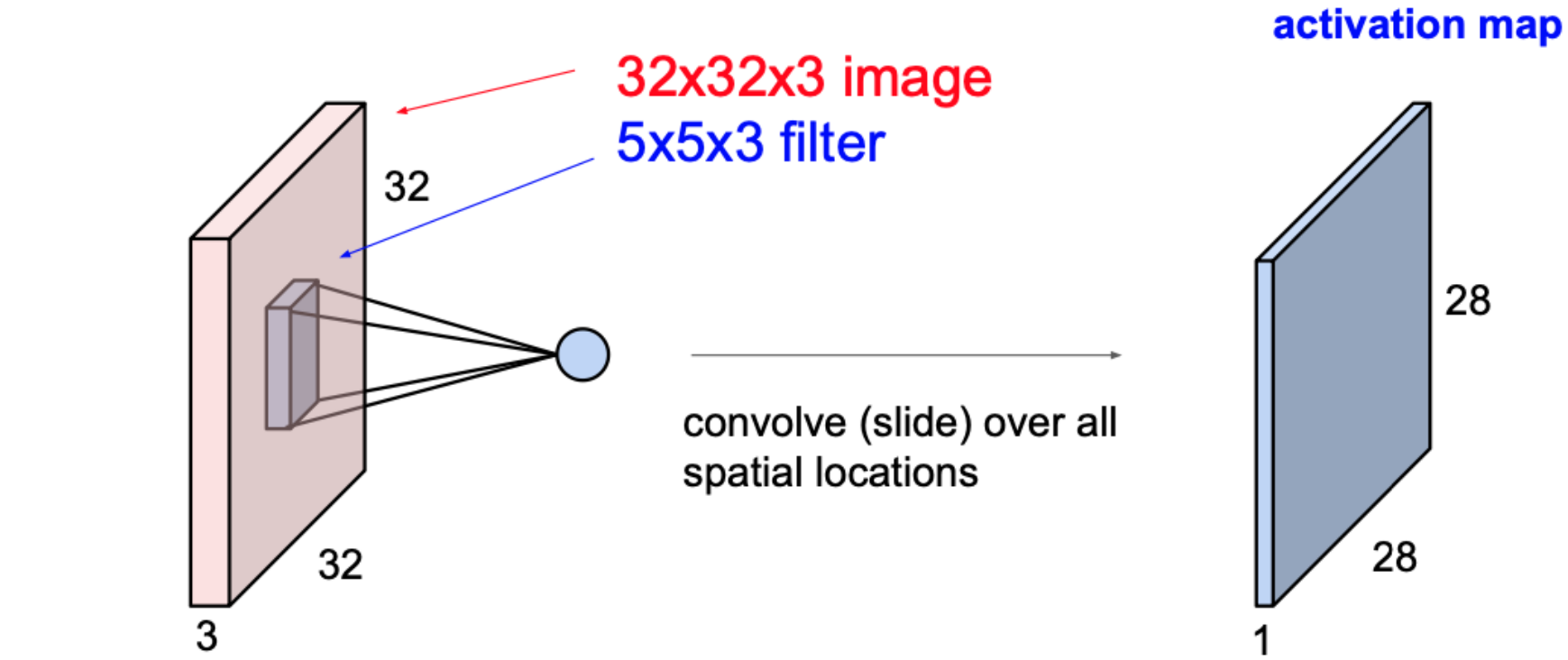
$\mathfrak{G}$-invariance $f(\rho(\mathrm{g})x) = f(x)$



$\mathfrak{G}$-equivariance $f(\rho(\mathrm{g})x) = \rho(\mathrm{g})f(x)$

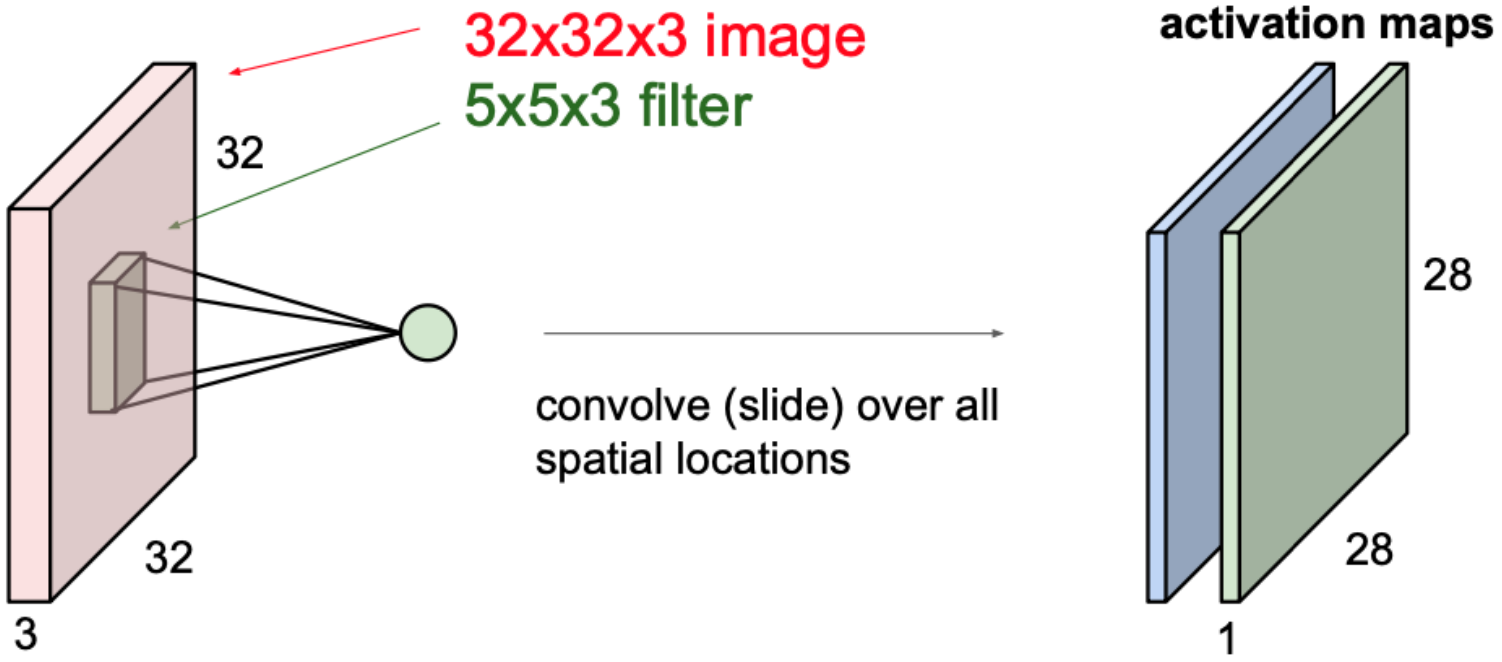


Convolution Layer



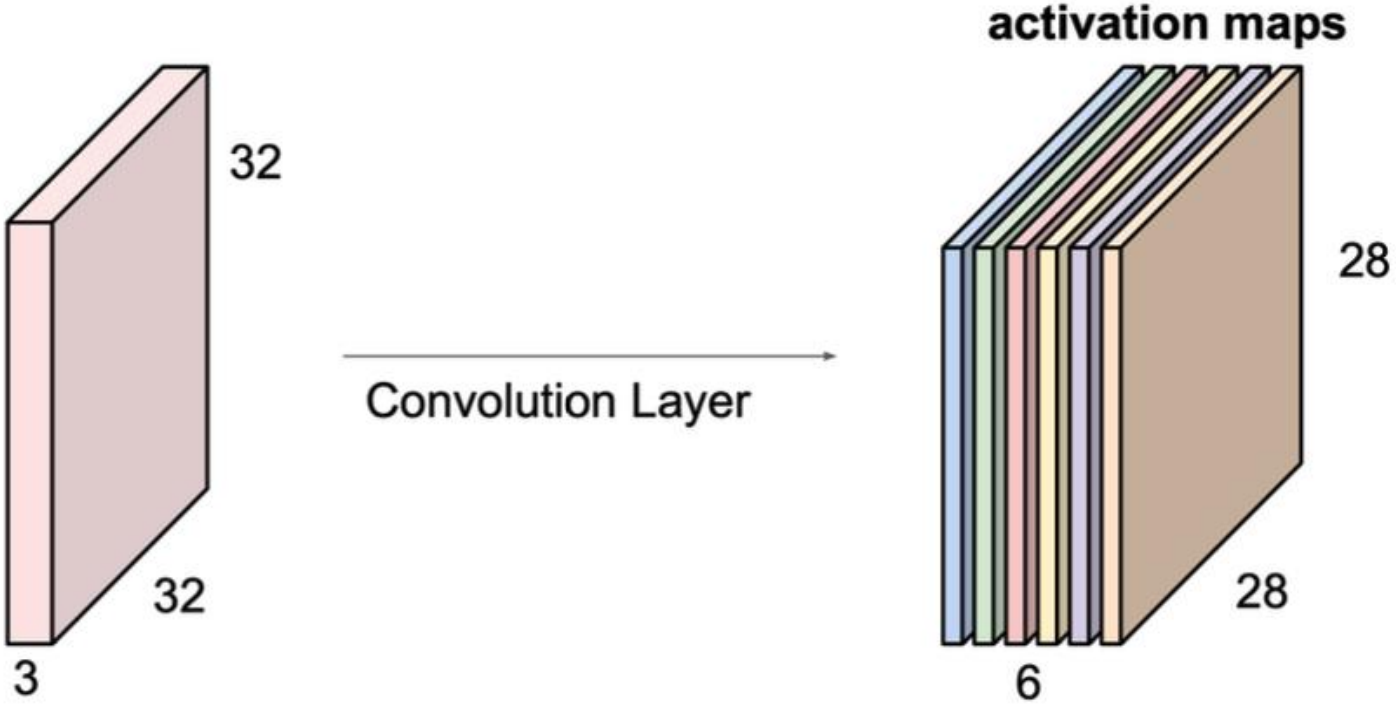
Convolution Layer

consider a second, green filter



CNN for Image Classification

For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a "new image" of size 28x28x6!

CNN for Image Classification

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

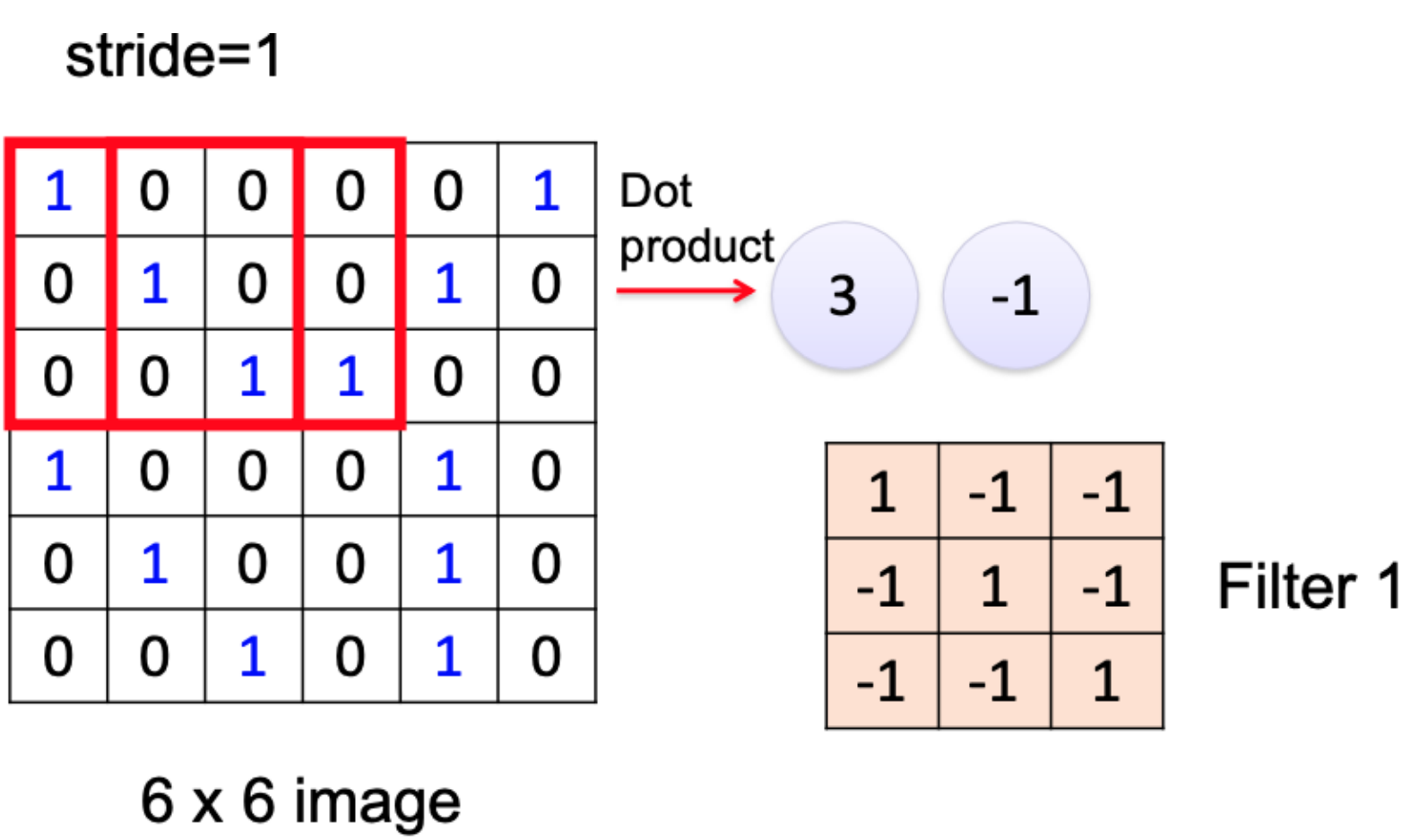
-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

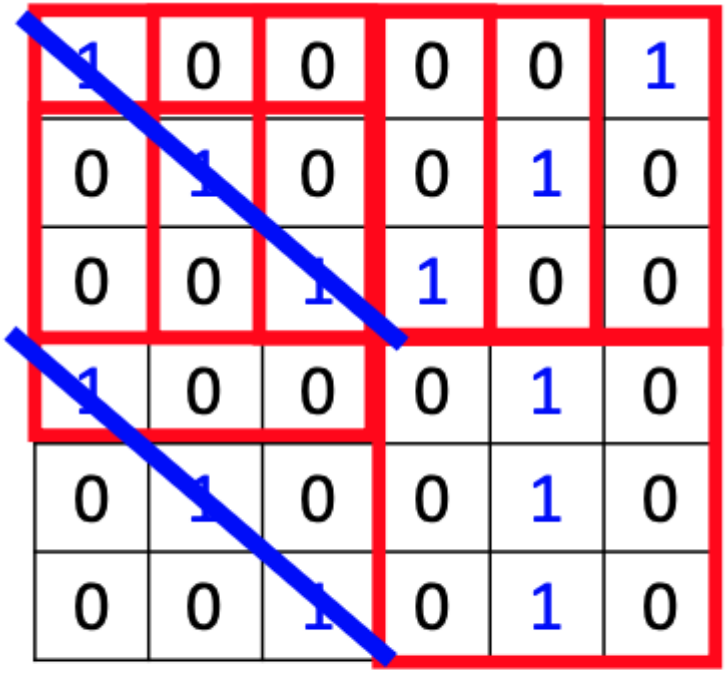
⋮ ⋮

Each filter detects a small pattern (3 x 3).

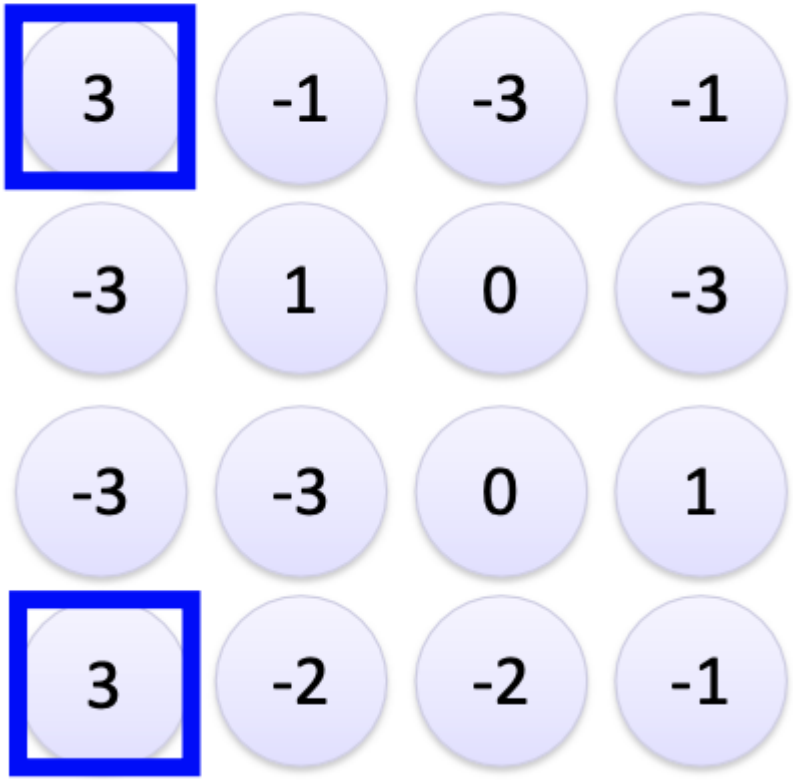
CNN for Image Classification



CNN for Image Classification



6 x 6 image



CNN for Image Classification

bird

bird

1	-1	-1
-1	1	-1
-1	-1	1

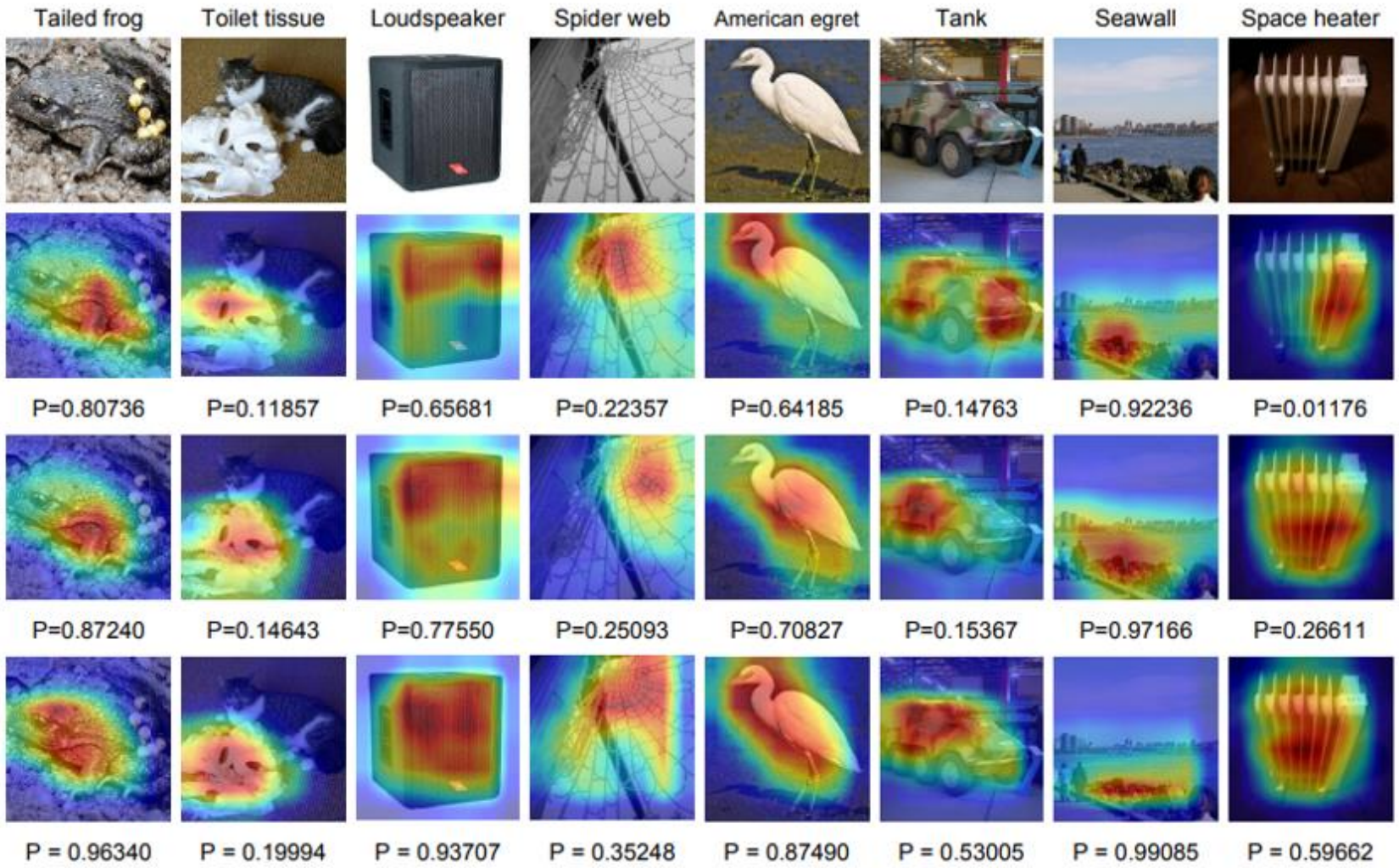
Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

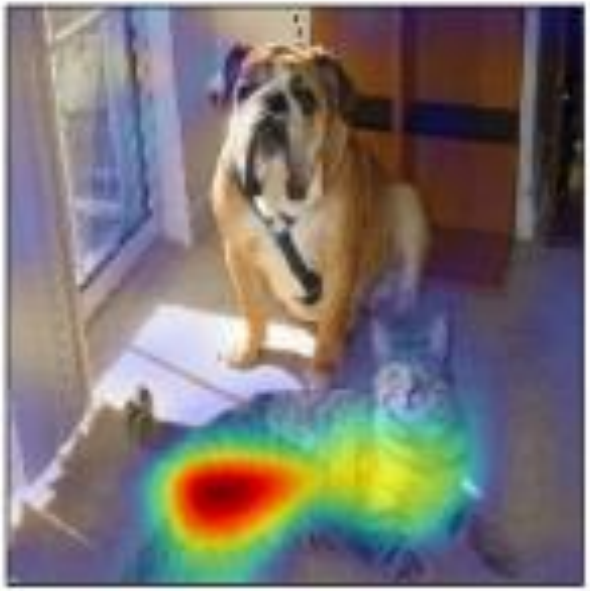
3	-1	-3	-1	-1	-1	-1	-1
-3	1	0	-3	-1	-1	-2	1
-3	-3	0	1	-1	-1	-2	1
3	-2	-2	-1	-1	0	-4	3

Explanation



Explanation

Original Image



Grad-CAM 'Cat'

Grad-CAM 'Dog'

$$y = f_\theta(x) \qquad \frac{\partial y}{\partial x}$$

How much change of each input of unit will change the output

Language Mining

1. Language Mining Application and how to represent an Language-structured data in computer?

2. **General Formulation of Next Token Generation**
    1. **Notation**
    2. **Conditional Distribution**
    3. Distribution Factorization

3. N-Gram Language Model
    1. Core Technique (Frequency Computing)
    2. Pros/Cons

4. RNN
    1. **Token Embedding, Word2Vec**
    2. **How to query token embedding given token ID?**
    3. **Architecture of RNN and Physical Meaning of Each Component**
    4. How we want to aggregate the output to do downstream task
        1. Last one?
        2. Every layer output?
    5. Problem with RNN: Gradient Vanish and Gradient Explosion

Transformer and LLM

1. Motivation of Transformation

2. **Attention:**
    1. **Motivation of Attention**
    2. **Physical Meaning of Query, Key and Value**
    3. **Master Every Detail and I will let you physical compute the forward process of Attention**
    4. **Multi-Head**
    5. **Causal Attention: Which Task?**
    6. **Cross Attention: Which Task?**
    7. **Self Attention: Which Task?**

3. Positional Encoding

4. **What is auto-regressive generation?**

5. **What is beam search?**

6. How to train LLM, what is the loss? Physical Meaning

Decomposing using the Chain Rule

$$p(\boldsymbol{X} = \boldsymbol{x}) = \begin{pmatrix} p(X_1 = x_1) \\ \cdot\, p(X_2 = x_2 \mid X_1 = x_1) \\ \cdot\, p(X_3 = x_3 \mid \boldsymbol{X}_{1:2} = \boldsymbol{x}_{1:2}) \\ \vdots \\ \cdot\, p(X_N = \bigcirc \mid \boldsymbol{X}_{1:N-1} = \boldsymbol{x}_{1:N-1}) \end{pmatrix}$$

$$= \prod_{i=1}^{N} p(X_i = x_i \mid \boldsymbol{X}_{1:i-1} = \boldsymbol{x}_{1:i-1})$$

Example:

$\boldsymbol{x}$ = (*I, like, this, movie, …*)

$$p(\boldsymbol{x}) = \cdots p_\theta(like \mid I)\, p_\theta(this \mid I, like) \cdots$$

Predict each word based on the "history"

Unigram Model: Empty History

$$p(\boldsymbol{X} = \boldsymbol{x}) = \prod_{i=1}^{N} p(X_i = x_i \mid \boldsymbol{X}_{1:i-1} = \boldsymbol{x}_{1:i-1})$$

$$\overset{\text{assumption}}{=} \prod_{i=1}^{N} p(X_i = x_i; \boldsymbol{\theta}) = \prod_{i=1}^{N} \theta_{x_i}$$

Multinomial distribution

Maximum likelihood estimate: for every $v \in \mathcal{V}$,

$$\theta_v^* = \frac{\sum_{i=1}^{N} \mathbf{1}\{x_i = v\}}{N}$$

$$= \frac{\text{count}_{\boldsymbol{x}}(v)}{N}$$

Example

The probability of

Presidents tell lies .

is:

$$p(X_1 = \text{Presidents}) \cdot p(X_2 = \text{tell}) \cdot p(X_3 = \text{lies}) \cdot p(X_4 = .) \cdot p(X_5 = \bigcirc)$$

In unigram model notation:

$$\theta_{\text{Presidents}} \cdot \theta_{\text{tell}} \cdot \theta_{\text{lies}} \cdot \theta_{.} \cdot \theta_{\bigcirc}$$

Using the maximum likelihood estimate for $\boldsymbol{\theta}$, we could calculate:

$$\frac{\text{count}_{\boldsymbol{x}}(\text{Presidents})}{N} \cdot \frac{\text{count}_{\boldsymbol{x}}(\text{tell})}{N} \cdots \frac{\text{count}_{\boldsymbol{x}}(\bigcirc)}{N}$$

Unigram Models: Assessment

Pros:

- Easy to understand
- Cheap
- Good enough for information retrieval (maybe)

Cons:

- Fixed, known vocabulary assumption
- "Bag of words" assumption is linguistically inaccurate
  - $p(\text{the the the the}) \gg p(\text{I want ice cream})$

"One Hot" Vectors

Let $\mathbf{e}_i \in \mathbb{R}^V$ be the ith column of the identity matrix $\mathbf{I}$.

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}; \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix}; \quad \ldots; \quad \mathbf{e}_V = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$\mathbf{e}_i$ is the "one hot" vector for the ith word in $\mathcal{V}$.

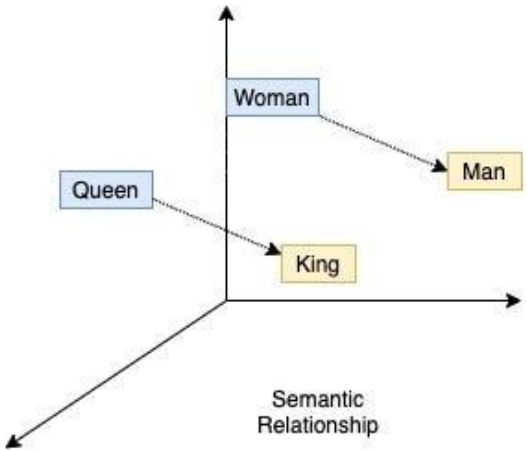
A neural language model starts by "looking up" each word by multiplying its one hot vector by a matrix $\underset{V \times d}{\mathbf{M}}$; $\mathbf{e}_v^\top \mathbf{M} = \mathbf{m}_v$, the "embedding" of v.

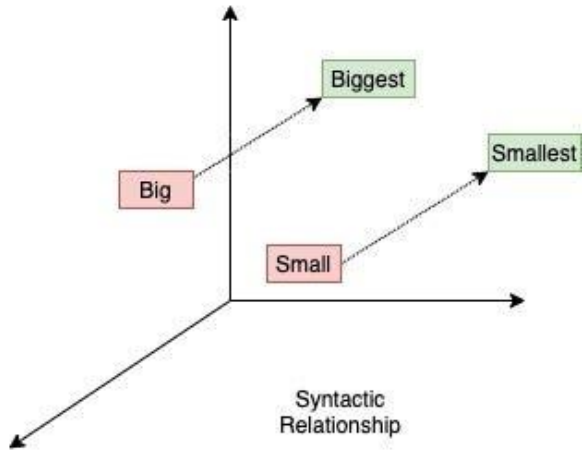
$\mathbf{M}$ becomes part of the parameters ($\boldsymbol{\theta}$).

Sequences of Word Vectors

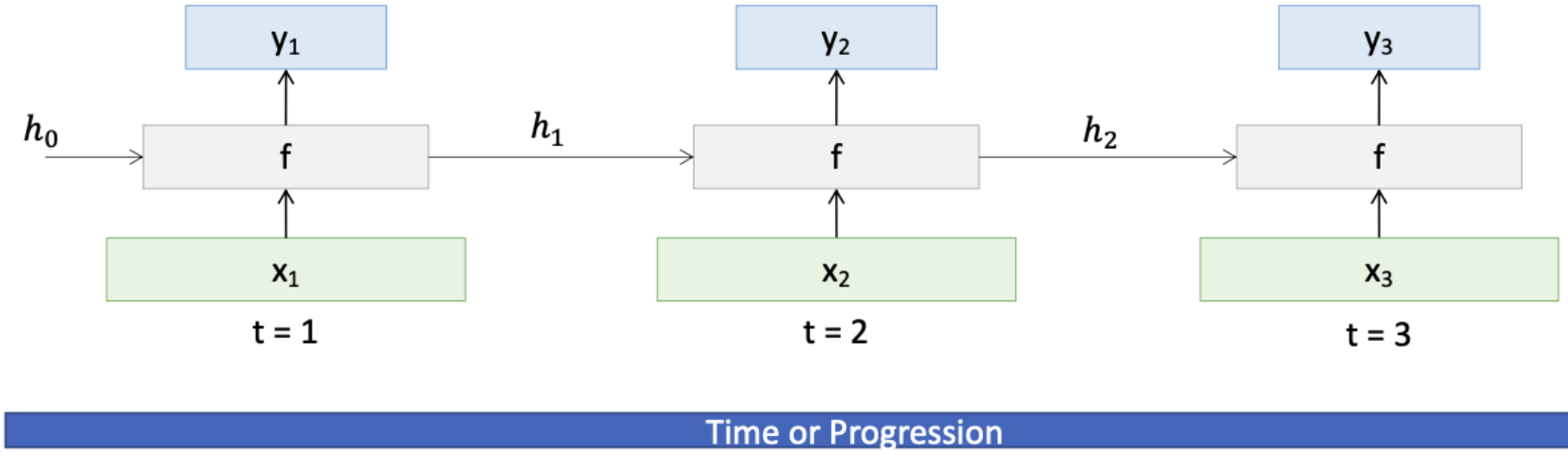
Given a word sequence $\langle v_1, v_2, \ldots, v_k \rangle$, we transform it into a sequence of word vectors,

$$\mathbf{m}_{v_1}, \mathbf{m}_{v_2}, \ldots, \mathbf{m}_{v_k}$$

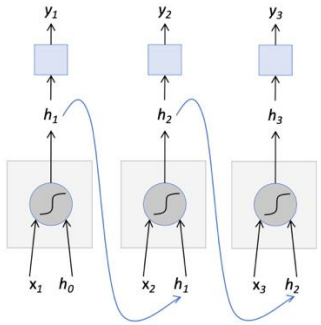




What if we have sequences of variable lengths, like sentences or videos that we would like to analyze?



How would you setup the model (token embedding, MLP layer)



- $h_t = \sigma(W^T \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix})$
- $y_t = F(h_t)$

Language Mining – Transformer - Attention

Think of YouTube.

First you enter your ***Query*** *in the search bar. Then your* ***Query*** *is compared against a set of* ***Keys*** *(in this case, video titles, tags and descriptions etc. within the YouTube database). After this, YouTube proceeds to retrieve the videos that best match your* ***Query****. These video results are referred to as* ***Values****.*

Key Value Query

I would like to taste the local food in France.

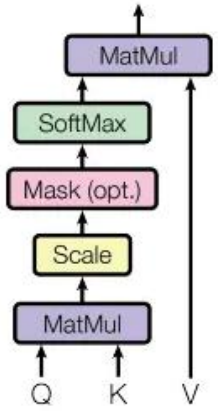
I would like to taste the local food in France.

$$Q = XW_Q \qquad K = XW_K \qquad V = XW_V$$

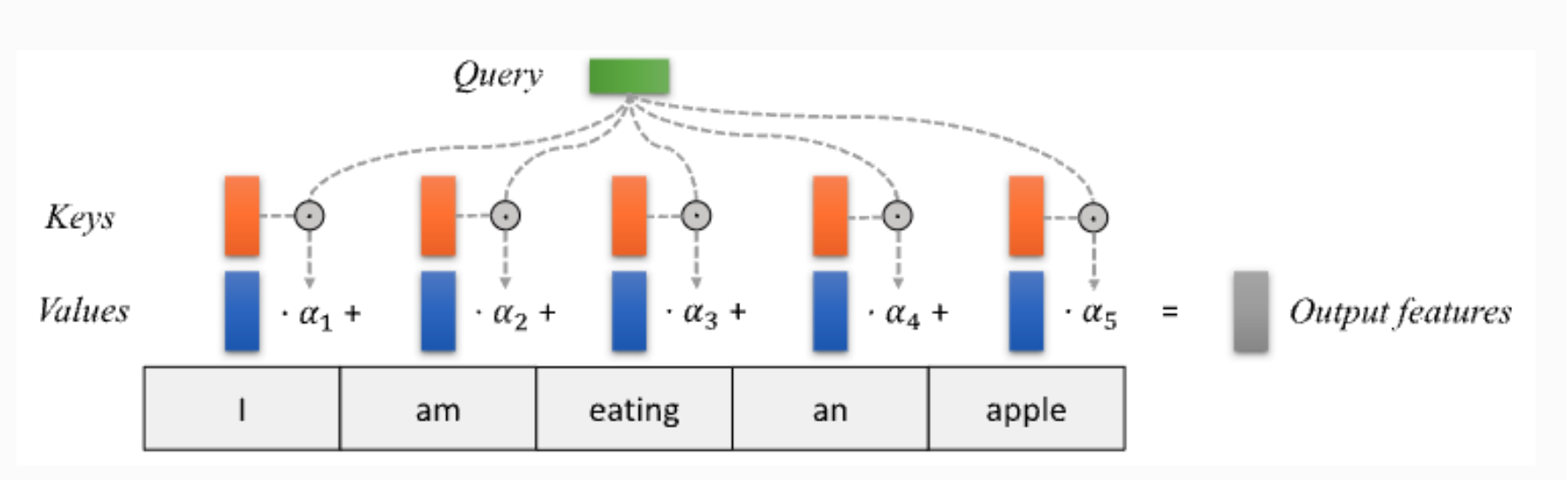
$$Q \cdot K = QK^T =$$

	I	would	like	to	taste	the	local	food	in	France
I	h									
would		h								
like			h							
to				h						
taste					h			h		
the						h				
local							h			h
food					h			h		
in									h	
France							h			h

$$softmax\left(\frac{Q_i K^T}{\sqrt{d_k}}\right) = \frac{e^{\left(\frac{Q_i K^T}{\sqrt{d_k}}\right)}}{\sum_{j=1}^{d_k} e^{\left(\frac{Q_j K^T}{\sqrt{d_k}}\right)}}$$



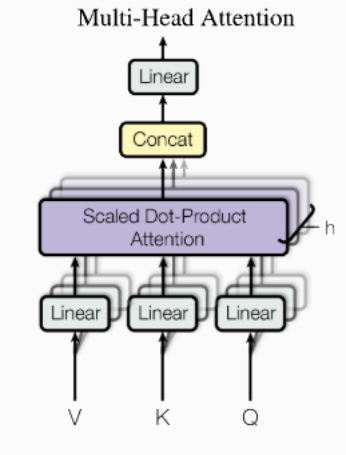
Language Mining – Transformer - Attention



$$\alpha_i = \frac{\exp\left(f_{attn}\left(\text{key}_i, \text{query}\right)\right)}{\sum_j \exp\left(f_{attn}\left(\text{key}_j, \text{query}\right)\right)}, \quad \text{out} = \sum_i \alpha_i \cdot \text{value}_i$$

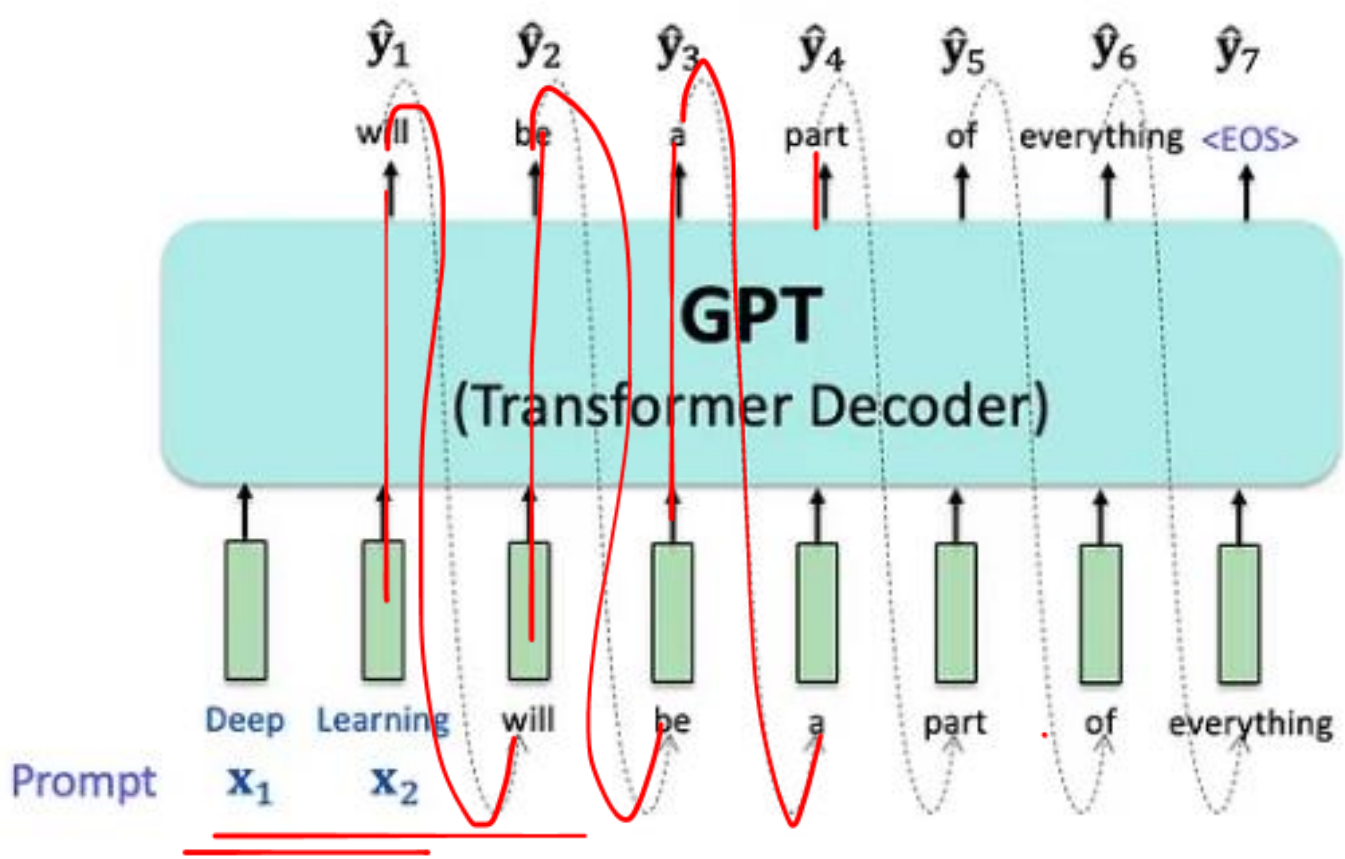
The scaled dot product attention allows a network to attend over a sequence. However, often there are multiple different aspects a sequence element wants to attend to, and a single weighted average is not a good option for it. This is why we extend the attention mechanisms to multiple heads

$$\text{Multihead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$
$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



$$W_{1...h}^Q \in \mathbb{R}^{D \times d_k}, W_{1...h}^K \in \mathbb{R}^{D \times d_k}, W_{1...h}^V \in \mathbb{R}^{D \times d_v}, \text{and } W^O \in \mathbb{R}^{h \cdot d_v \times d_{out}}$$

Language Mining – LLM

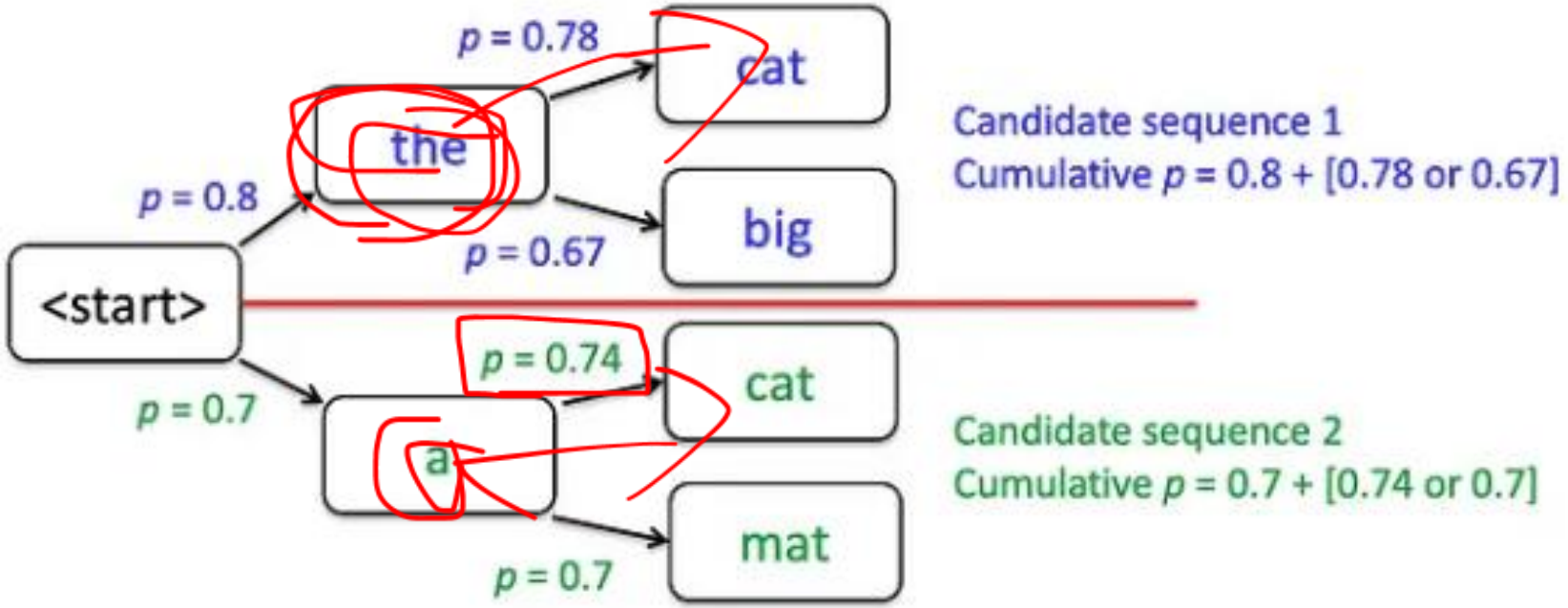


$$\hat{\mathbf{y}}_i = \arg\max_{\hat{\mathbf{y}}} P(\hat{\mathbf{y}}|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m, \hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_{i-1})$$

Greedy Decoding

Greedy decoding is computationally efficient and easy to implement.
It does not explore alternative paths that might lead to more globally optimal sequences.

Language Mining – LLM



https://medium.com/@lmpo/mastering-llms-a-guide-to-decoding-algorithms-c90a48fd167b

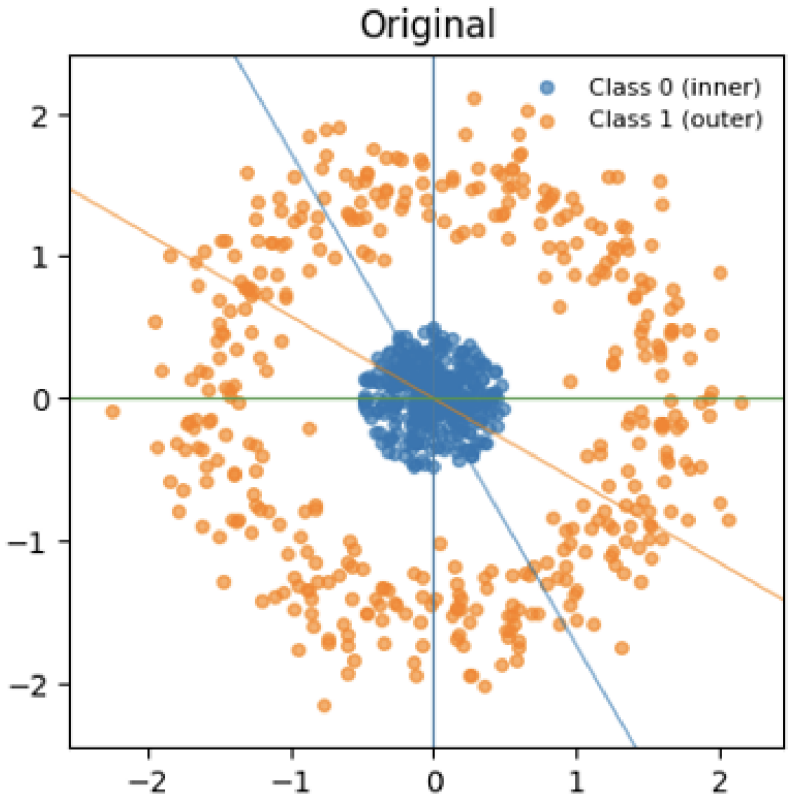


Figure 1

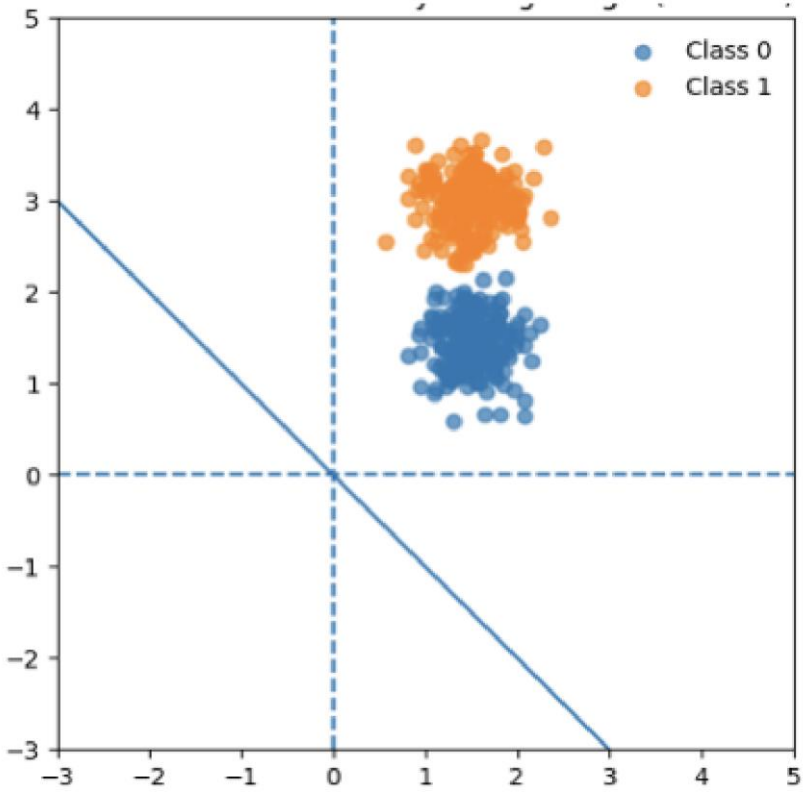


Figure 2