Introduction to Learning

Types of Learning

1. **Supervised Learning**
2. **Unsupervised Learning**
3. **Reinforcement learning**
4. Others (Zero-shot learning and Transfer learning)

Supervised Learning

- Supervised Learning techniques automatically learn a model of the relationship between a set of **descriptive features** and a **target feature** from a set of historical examples.

Supervised Learning

Labeled Training data

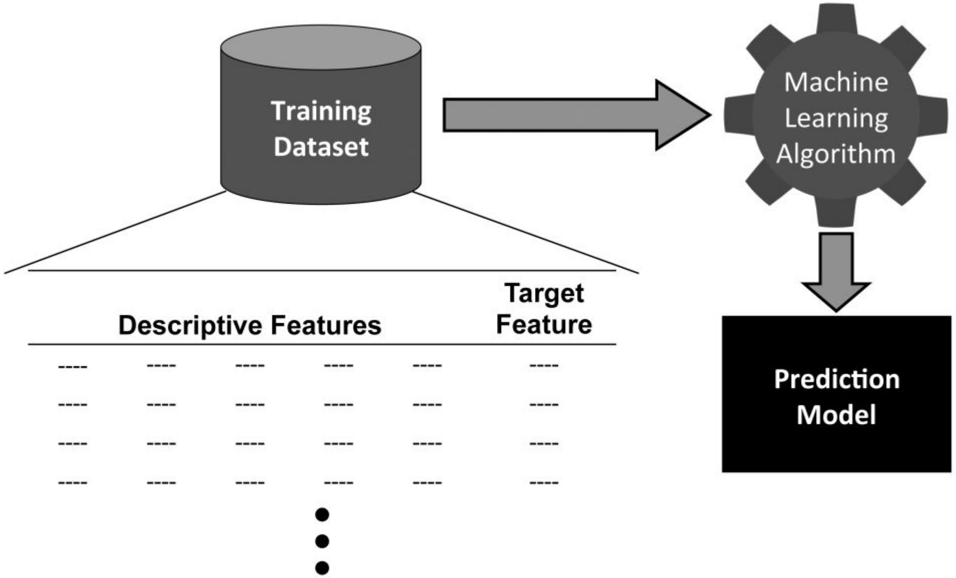
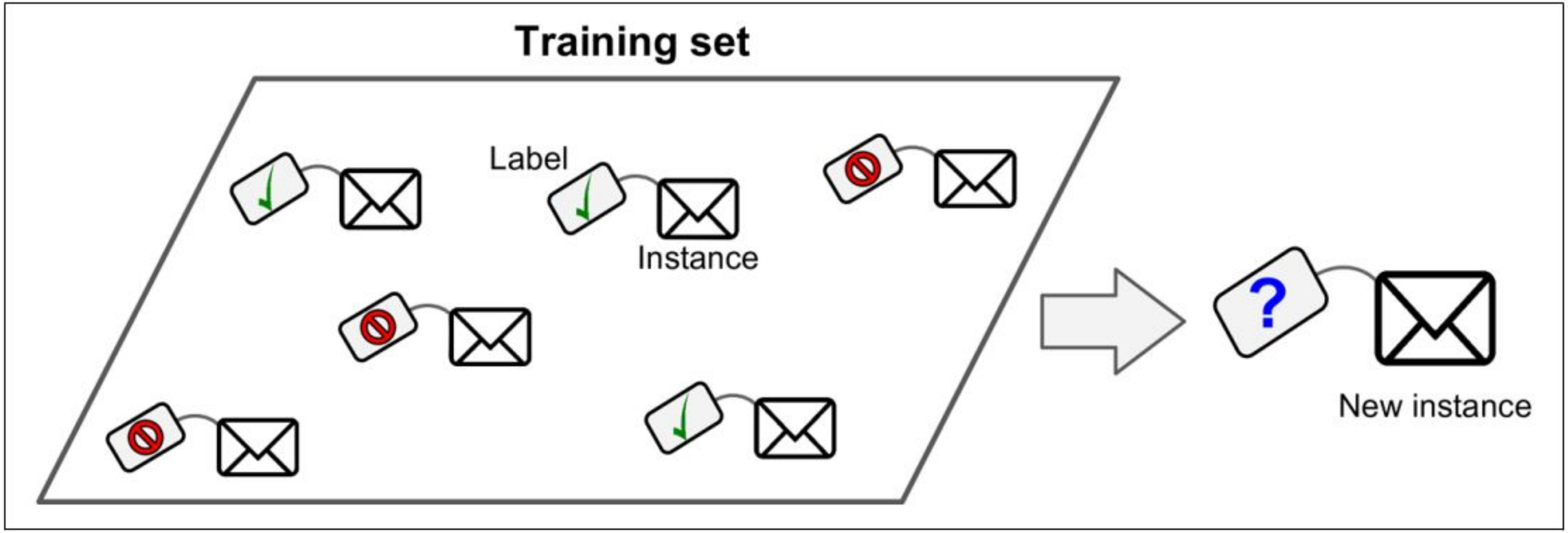


Figure: Using machine learning to induce a prediction model from a training dataset.

Supervised Learning

Figure: Using the model to make predictions for new query instances.

Supervised Learning



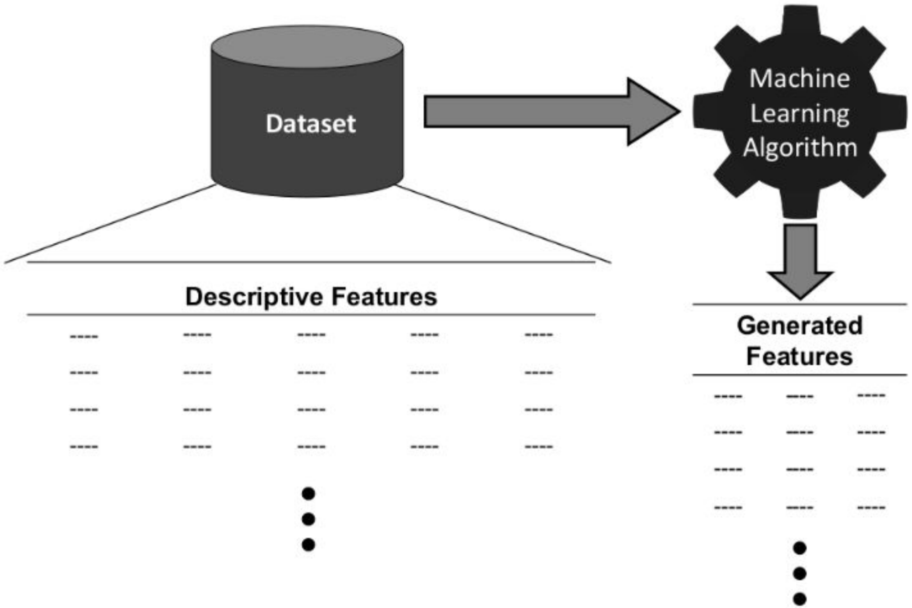
A labeled training set for supervised learning (e.g., spam classification)

Supervised Learning

Figure 1-6. Regression

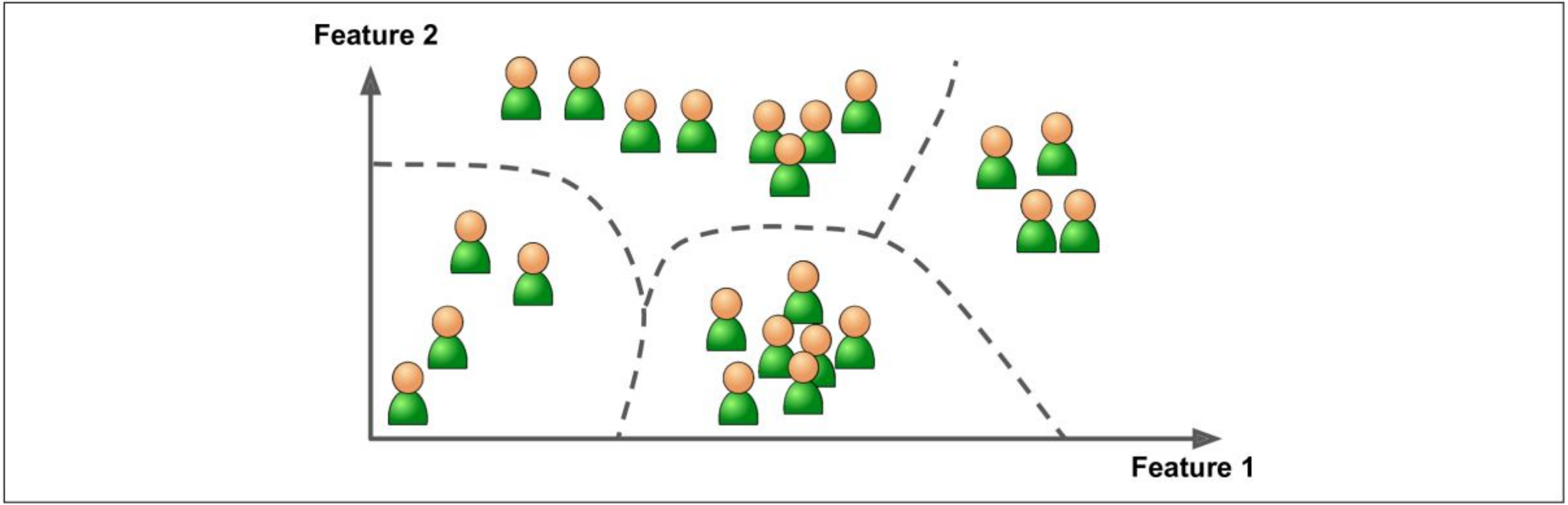
Unsupervised Learning

Unlabeled data



Unsupervised machine learning as a single-step process.

Unsupervised Learning

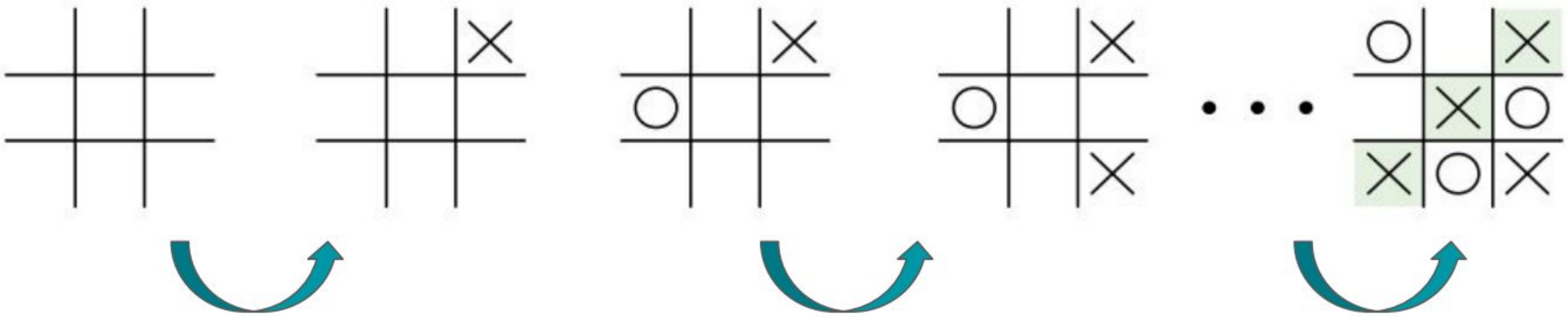


Clustering

Reinforcement learning

- Reinforcement learning (RL) is concerned with solving sequential decision-making problems.
- Many real-world problems - playing video games, sports, driving, robotic control - can be framed in this way.
- For example we can formulate the game of Tic-tac-toe in RL.

Reinforcement learning



sequential decision-making

Tasks (Applications)

1. Regression
2. Classification
3. Dimensionality reduction
4. Clustering
5. Synthesis and sampling
6. Denoising
7. Density estimation
8. Anomaly detection

How Does Machine Learning Work?

How?

- Machine learning algorithms work by searching through a set of possible prediction models for the model that are ***consistent*** with the data.
- Note that a training dataset is only a sample.
- ML is an ill-posed problem.

Table: A simple retail dataset

ID	BBY	ALC	ORG	GRP
1	no	no	no	couple
2	yes	no	yes	family
3	yes	yes	no	family
4	no	no	yes	couple
5	no	yes	yes	single

Table: A full set of potential prediction models before any training data becomes available.

BBY	ALC	ORG	GRP	$\mathbb{M}_1$	$\mathbb{M}_2$	$\mathbb{M}_3$	$\mathbb{M}_4$	$\mathbb{M}_5$	...
no	no	no	?	couple	couple	single	couple	couple	
no	no	yes	?	single	couple	single	couple	couple	
no	yes	no	?	family	family	single	single	single	
no	yes	yes	?	single	single	single	single	single	
yes	no	no	?	couple	couple	family	family	family	...
yes	no	yes	?	couple	family	family	family	family	
yes	yes	no	?	single	family	family	family	family	
yes	yes	yes	?	single	single	family	family	couple	

Table: A sample of the models that are consistent with the training data

BBY	ALC	ORG	GRP	$\mathbb{M}_1$	$\mathbb{M}_2$	$\mathbb{M}_3$	$\mathbb{M}_4$	$\mathbb{M}_5$	...
no	no	no	couple	couple	couple	single	couple	couple	
no	no	yes	couple	single	couple	single	couple	couple	
no	yes	no	?	family	family	single	single	single	
no	yes	yes	single	single	single	single	single	single	
yes	no	no	?	couple	couple	family	family	family	...
yes	no	yes	family	couple	family	family	family	family	
yes	yes	no	family	single	family	family	family	family	
yes	yes	yes	?	single	single	family	family	couple	

Table: A sample of the models that are consistent with the training data

BBY	ALC	ORG	GRP	$\mathbb{M}_1$	$\mathbb{M}_2$	$\mathbb{M}_3$	$\mathbb{M}_4$	$\mathbb{M}_5$	...
no	no	no	couple	couple	couple	single	couple	couple	
no	no	yes	couple	single	couple	single	couple	couple	
no	yes	no	?	family	family	single	single	single	
no	yes	yes	single	single	single	single	single	single	
yes	no	no	?	couple	couple	family	family	family	...
yes	no	yes	family	couple	family	family	family	family	
yes	yes	no	family	single	family	family	family	family	
yes	yes	yes	?	single	single	family	family	couple	

- Notice that there is more than one candidate model left! It is because a single consistent model cannot be found based on a sample training dataset that ML is ill-posed.

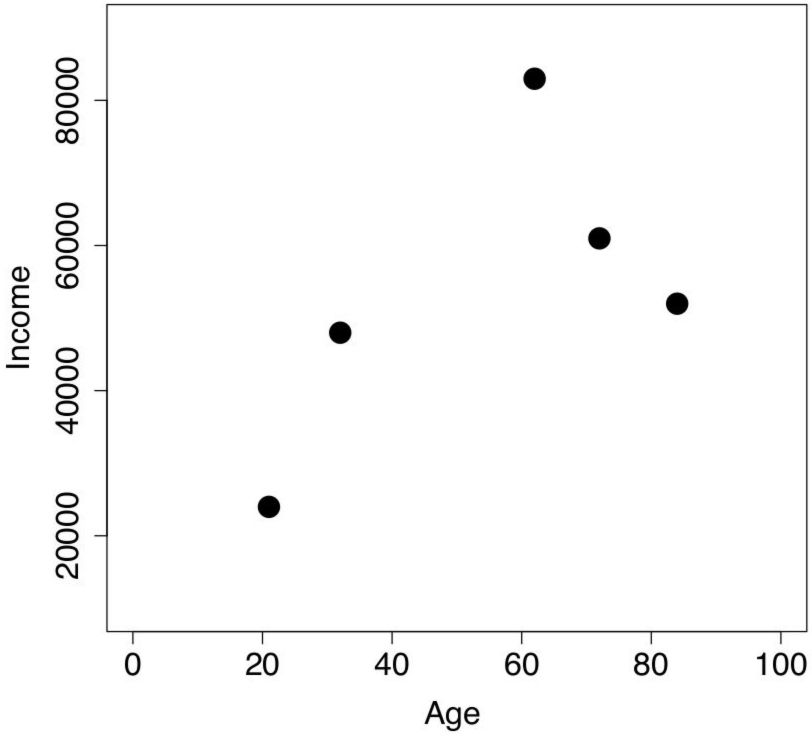
What Can Go Wrong With ML?

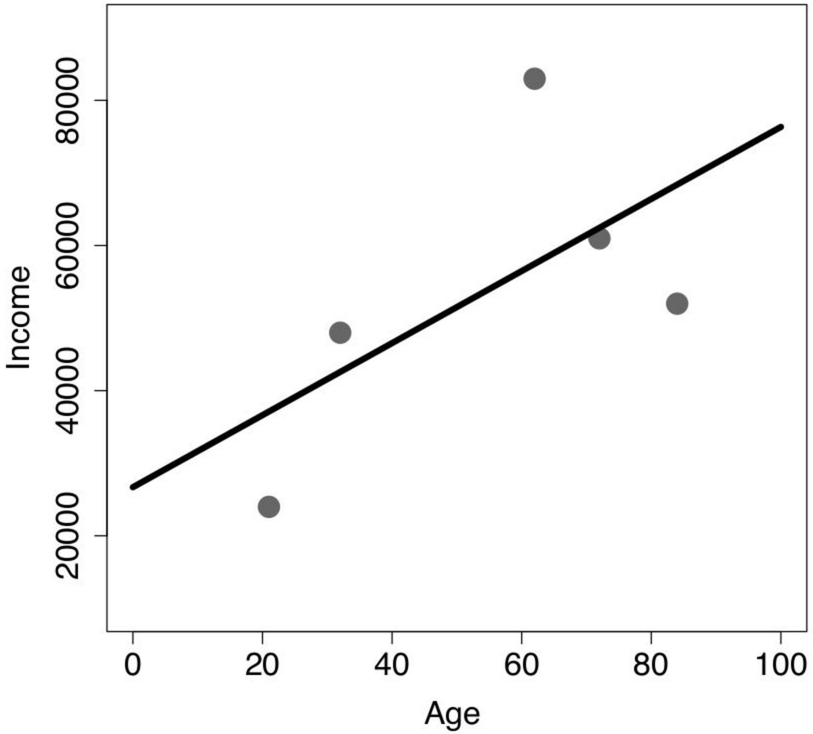
What can go wrong with learning?

- Underfitting.
- Overfitting.
- Wrong inductive bias.
- Sample bias.

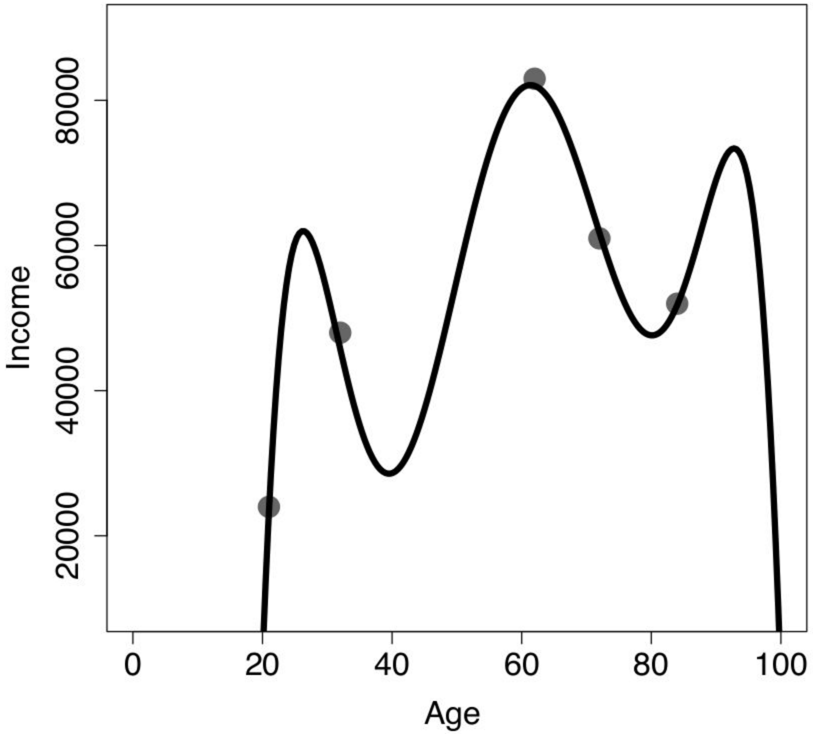
Table: The age-income dataset.

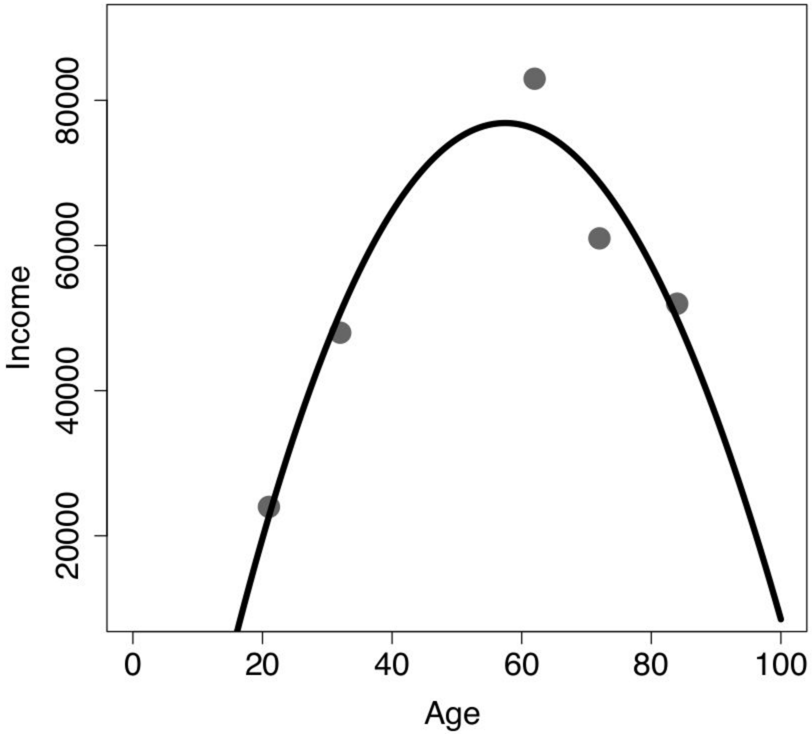
ID	AGE	INCOME
1	21	24,000
2	32	48,000
3	62	83,000
4	72	61,000
5	84	52,000

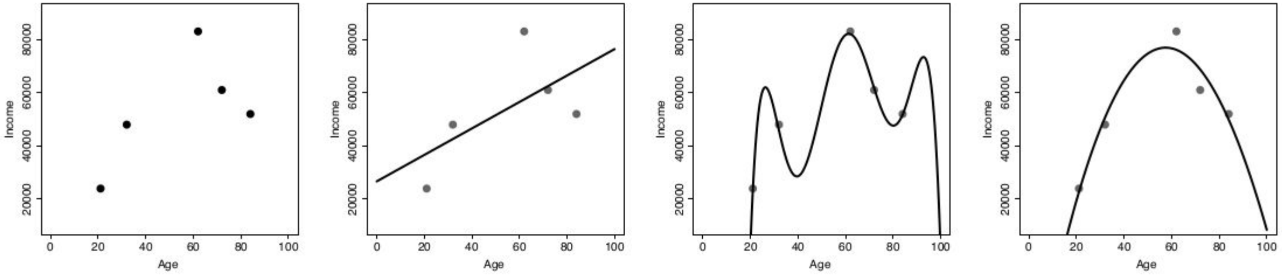

Income
80000
60000
40000
20000
0
20
40
60
80
100
Age



Income
20000
40000
60000
80000
0
20
40
60
80
100
Age


Income
20000
40000
60000
80000
0
20
40
60
80
100
Age


Income
20000
40000
60000
80000
0
20
40
60
80
100
Age



(a) Dataset (b) Underfitting (c) Overfitting (d) Just right

Figure: Striking a balance between overfitting and underfitting when trying to predict age from income.

What can go wrong with learning?

- Underfitting.
- Overfitting.
- Wrong inductive bias.
- Sample bias.

Types of Learning Algorithms

- Information-based learning (Decision Tree, Bagging, Boosting)

Types of Learning Algorithms

- Information-based learning (Decision Tree, Bagging, Boosting)
- Similarity-based learning (Nearest Neighbor Algorithm)

Types of Learning Algorithms

- Information-based learning (Decision Tree, Bagging, Boosting)
- Similarity-based learning (Nearest Neighbor Algorithm)
- Probability-based learning (Naive Bayes Classifier)

Types of Learning Algorithms

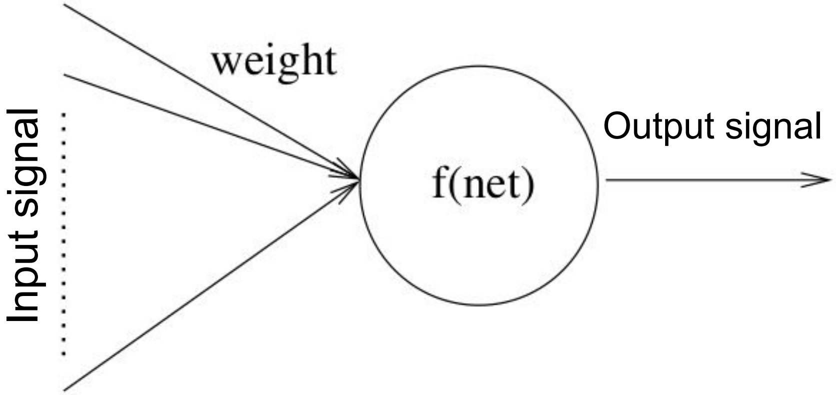
- Information-based learning (Decision Tree, Bagging, Boosting)
- Similarity-based learning (Nearest Neighbor Algorithm)
- Probability-based learning (Naive Bayes Classifier)
- Error-based learning (Simple Linear Regression and Logistic Regression)

Types of Learning Algorithms

- Information-based learning (Decision Tree, Bagging, Boosting)
- Similarity-based learning (Nearest Neighbor Algorithm)
- Probability-based learning (Naive Bayes Classifier)
- Error-based learning (Simple Linear Regression and Logistic Regression)
- Deep Learning (Using Deep Neural Networks for ML Tasks)

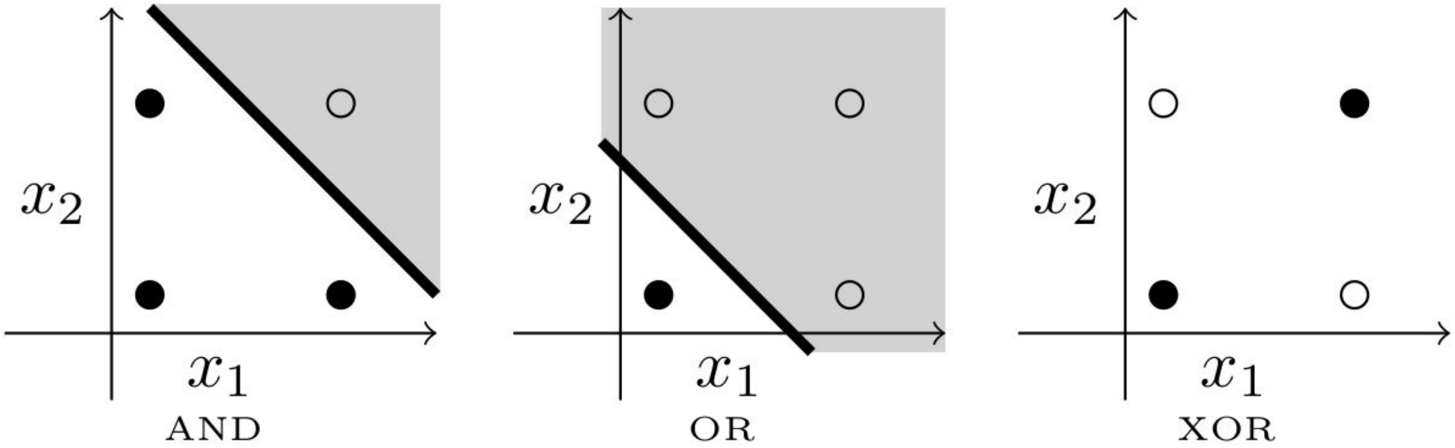
Neural Networks

Artificial Neuron



An Artificial Neuron

Why depth of network is important?

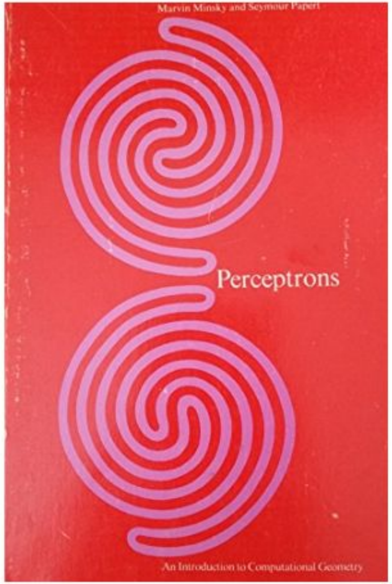


The logical AND and OR functions are linearly separable, but the XOR is not.

The "XOR Affair"

XOR

"[simple] perceptron cannot represent even the XOR function"



M. Minsky

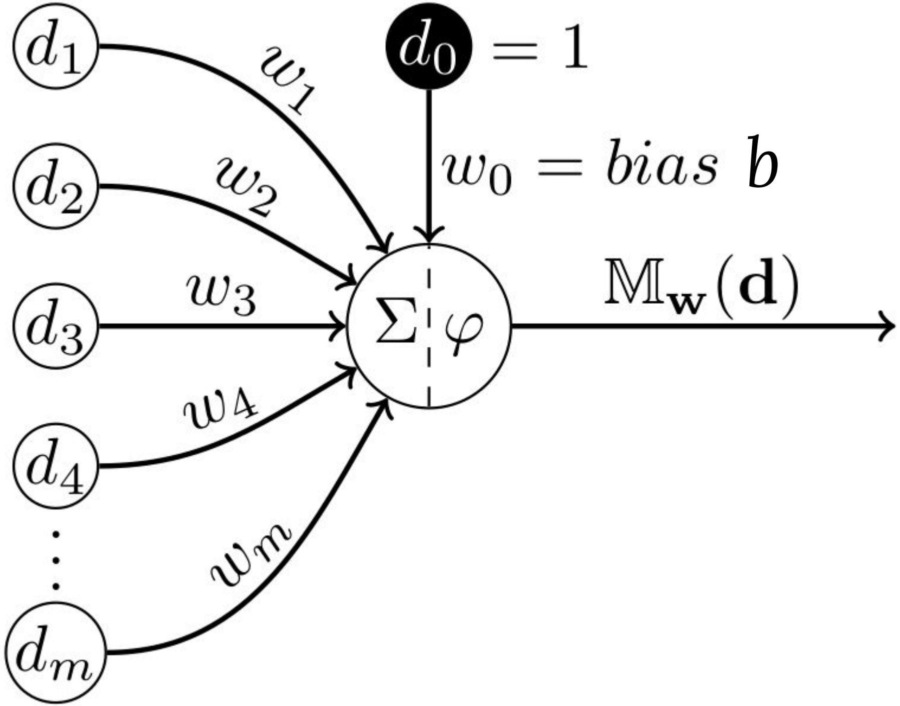
S. Papert

1969

Minsky, Papert 1969

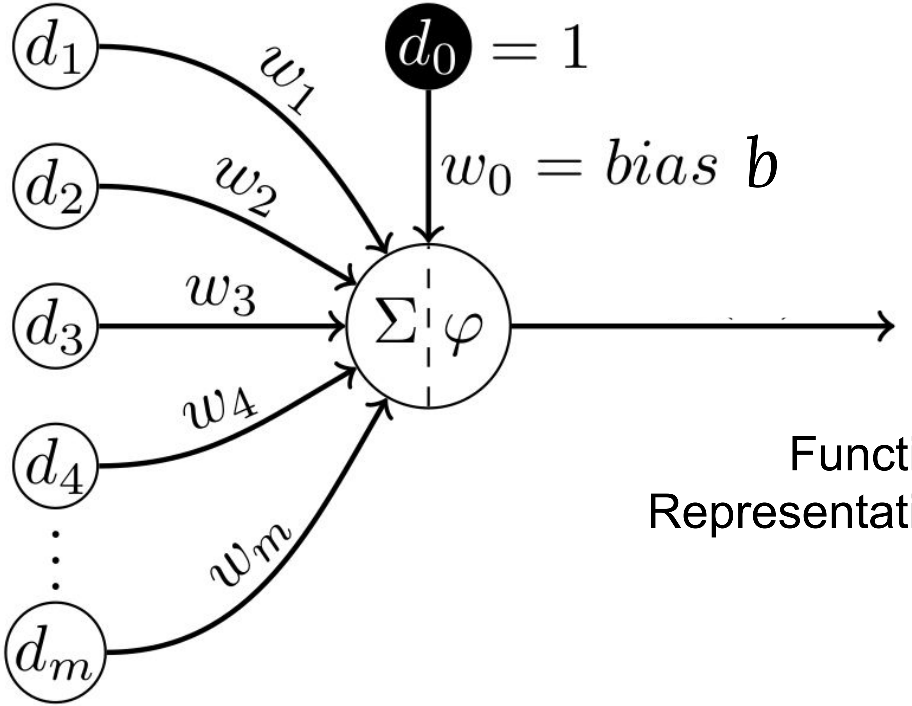

"AI WINTER"

Perceptron



d_1
w_1
d_0 $= 1$
$w_0 = bias\ b$
d_2
w_2
d_3
w_3
Σ
φ
$\mathbb{M}_{\mathbf{w}}(\mathbf{d})$
d_4
w_4
w_m
d_m

Perceptron



$$= \varphi\left([w_0, w_1, \ldots, w_m] \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_m \end{bmatrix} \right)$$

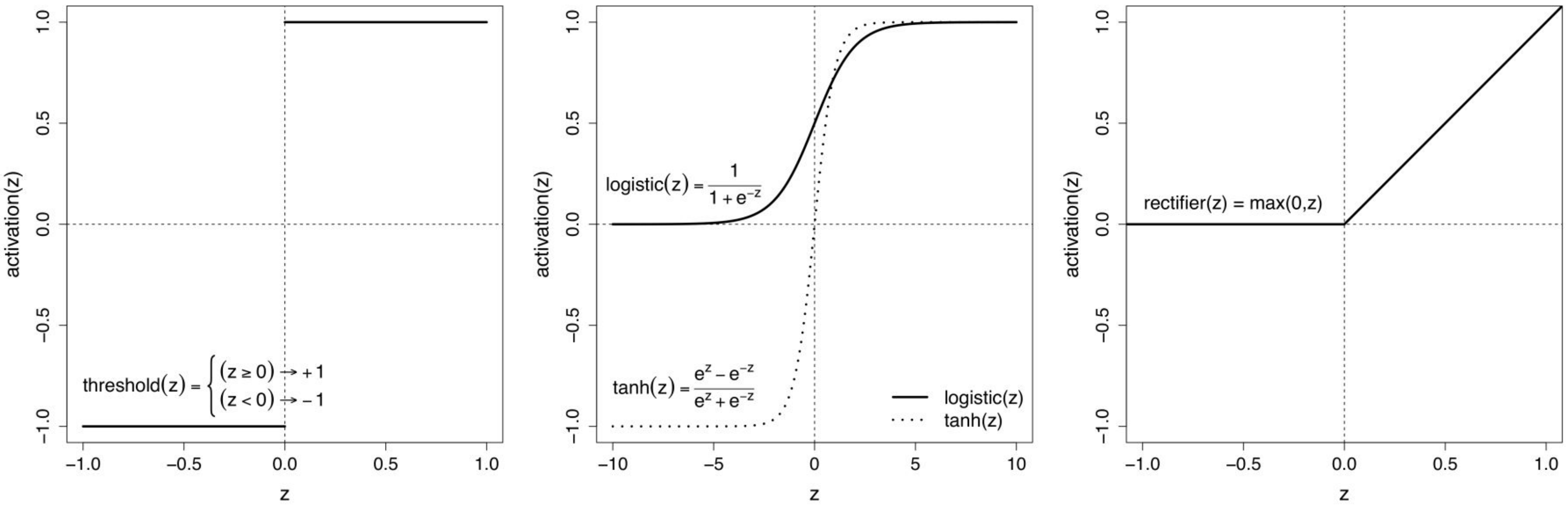
Function Representation

$$= \mathbb{M}(d;\, w, b)$$
$$= \mathbb{M}_{w,b}(d)$$

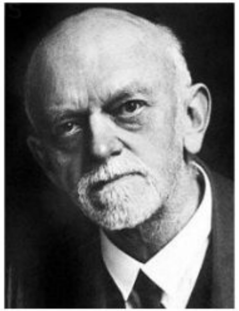
Perceptron

$$
\begin{aligned}
\mathbb{M}_{\mathbf{w}}(\mathbf{d}) &= \varphi\left(\mathbf{w}[0] \times \mathbf{d}[0] + \mathbf{w}[1] \times \mathbf{d}[1] + \cdots + \mathbf{w}[m] \times \mathbf{d}[m]\right) \\
&= \varphi\left(\sum_{i=0}^{m} w_i \times d_i\right) = \varphi\left(\underbrace{\mathbf{w} \cdot \mathbf{d}}_{dot\ product}\right) \\
&= \varphi\left(\underbrace{\mathbf{w}^T \mathbf{d}}_{matrix\ product}\right) = \varphi\left([w_0, w_1, \ldots, w_m] \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_m \end{bmatrix}\right)
\end{aligned}
$$

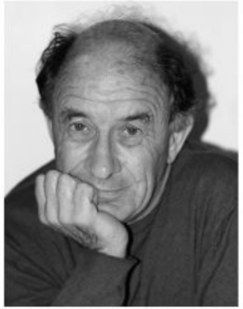
Activation functions φ



Universal Approximation


D. Hilbert


A. Kolmogorov


V. Arnold

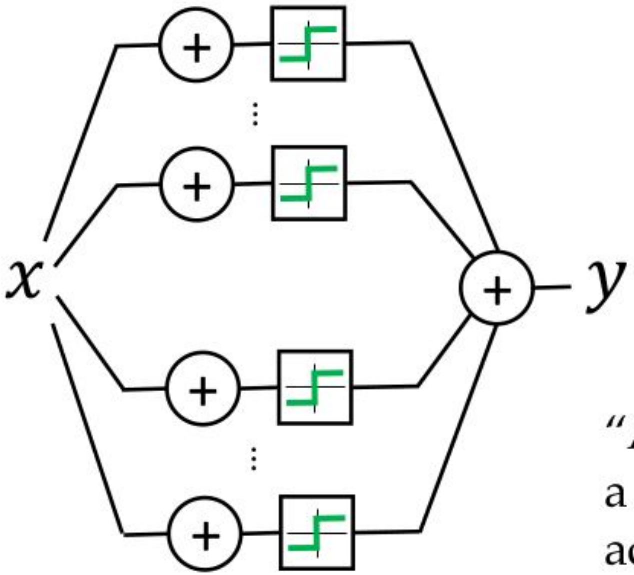

G. Cybenko

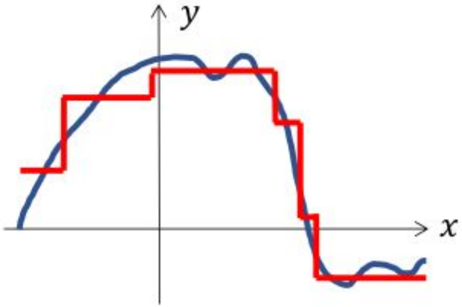

K. Hornik

Results specific to multilayer neural networks

Hilbert 1900; Arnold 1956; Kolmogorov 1957; Cybenko 1989; Hornik 1991

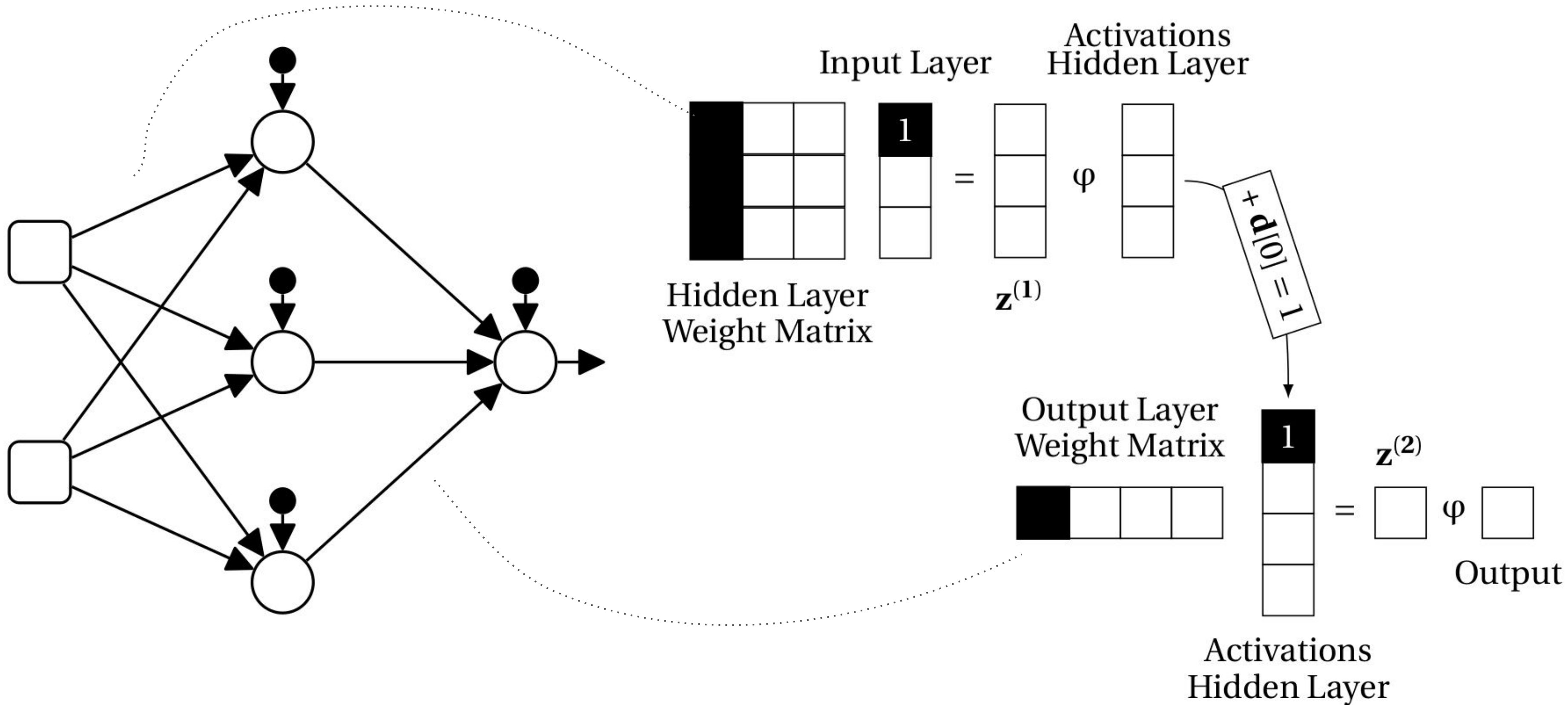
Universal Approximation



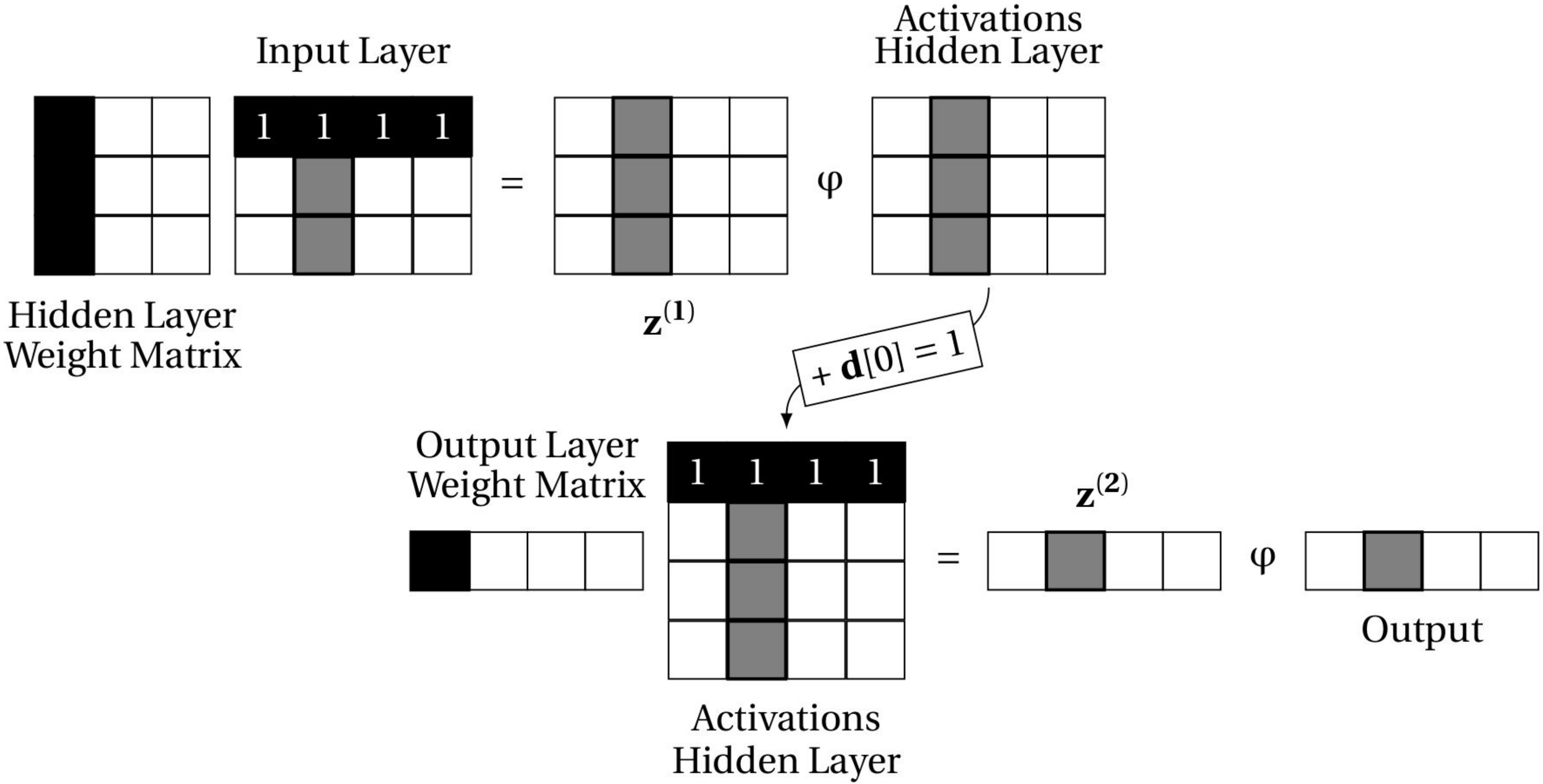


"A 2-layer perceptron can approximate a continuous function to any desired accuracy"

ANN graphical and matrix representations



Batch of examples

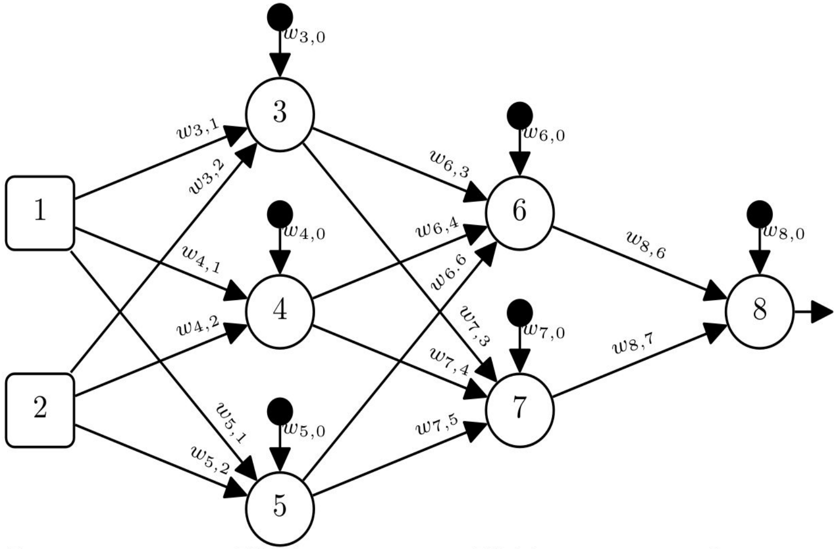


Feedforward artificial neural network (ANN)

where, layer functions are described as:

$$f^l(z) = g^l(W^l z + b^l)$$

- ***l*** is called the layer index
- g^l is called an *activation function*
- parameters W^l(matrix) and b^l (vector)



Function Representation

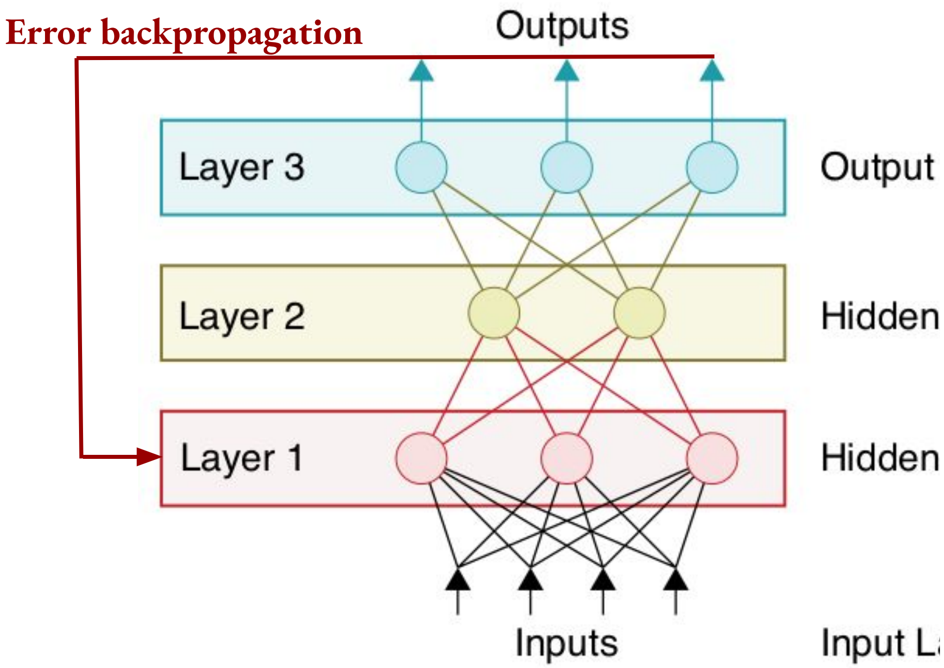
$$\mathbb{M}(d; w, b) = f^3\left(f^2\left(f^1(d)\right)\right)$$

Objective Function

Given N samples as $\{x_i, y_i\}_{i=1}^{N}$,

the objective function is defined as:

$$\min_{w,\, b} \frac{1}{N} \sum_{i=1}^{N} L(f_{w,b}(x_i),\, y_i)$$



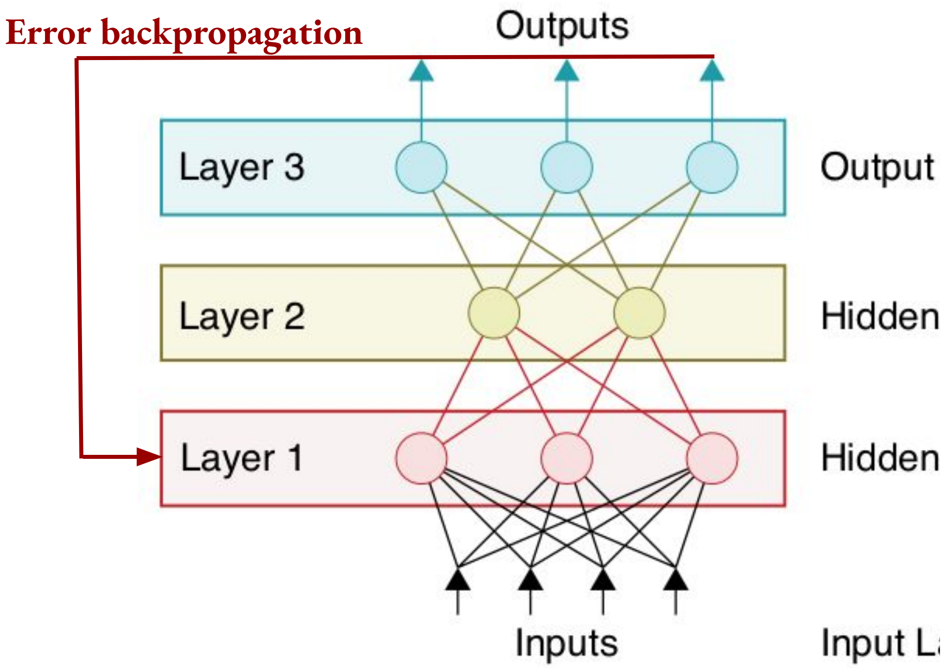
Objective Function

Given N samples as $\{x_i, y_i\}_{i=1}^{N}$,

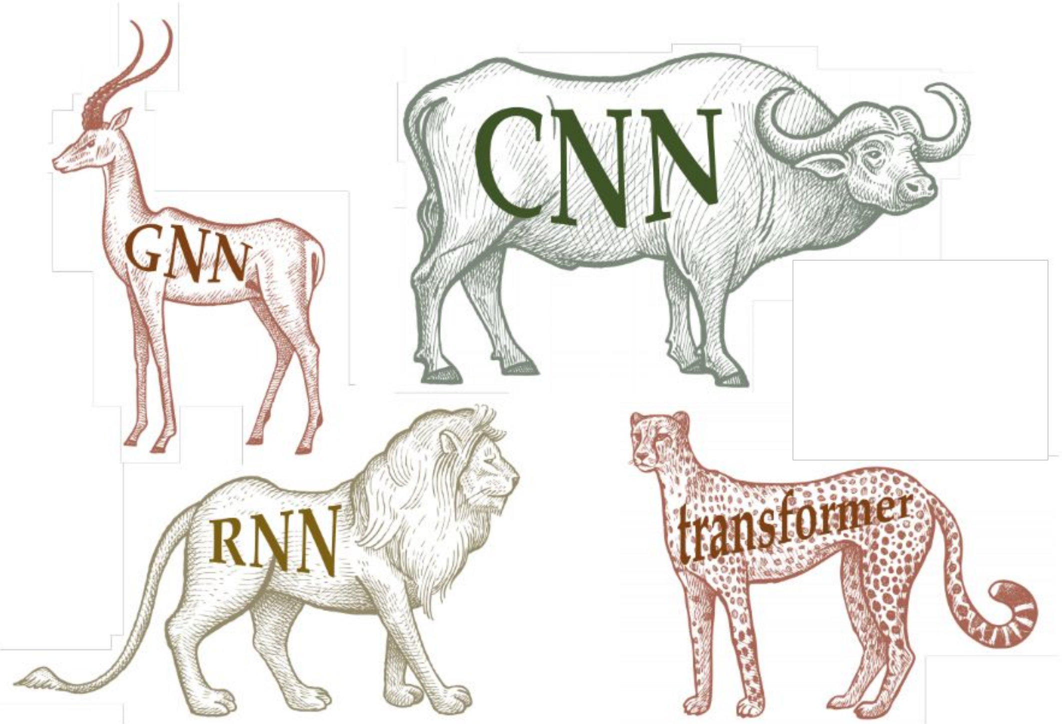
the objective function is defined as:

$$\min_{w,\,b} \frac{1}{N} \sum_{i=1}^{N} L(f_{w,b}(x_i),\, y_i)$$

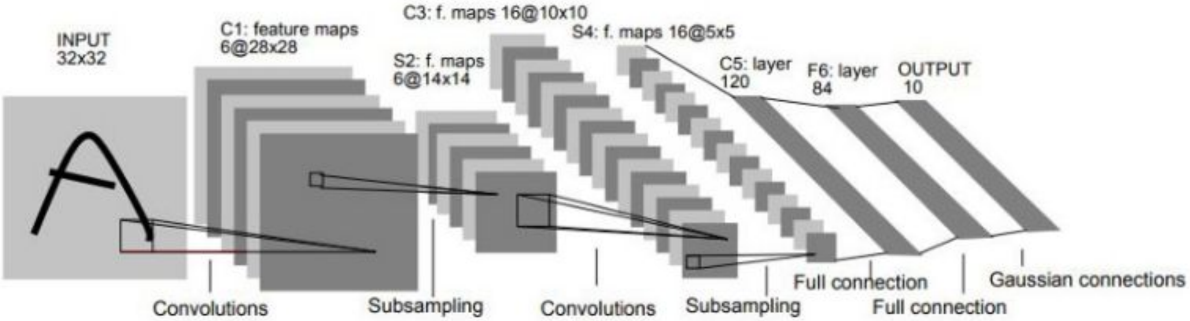
- **Regression**: $f_{w,b}(x_i)$ outputs a scalar
- **Classification**: $f_{w,b}(x_i)$ outputs the class



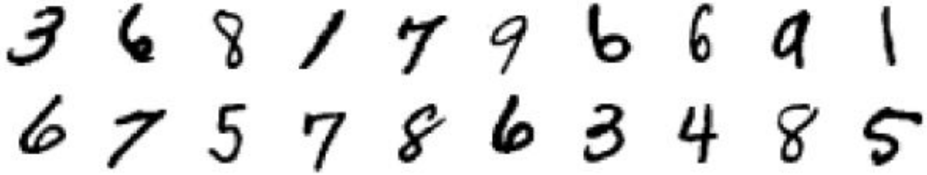
Deep Learning



LeNet



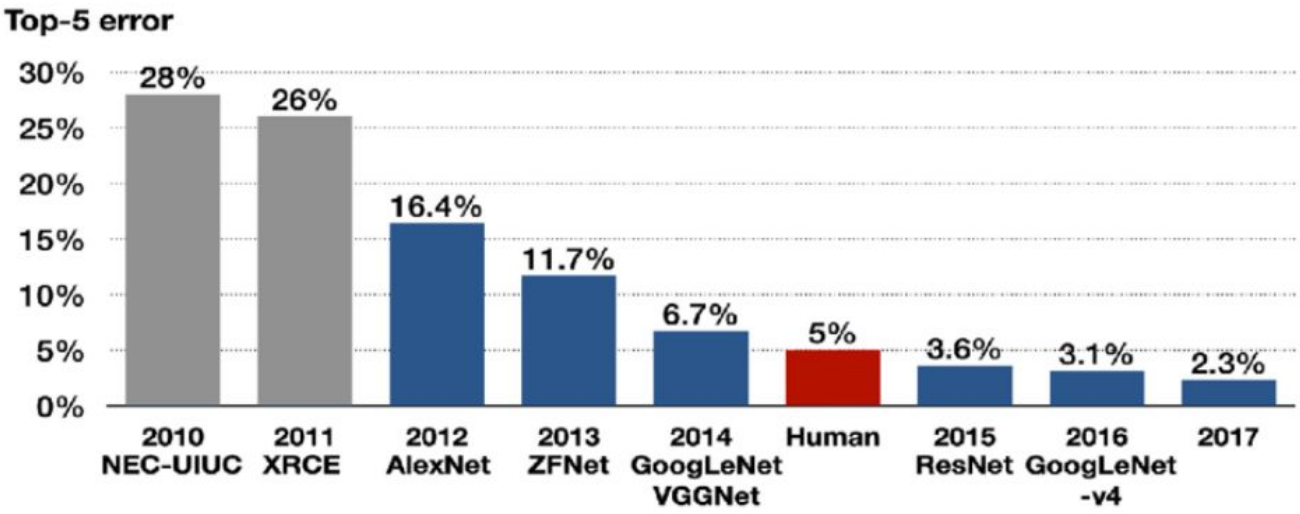
LeNet-5 classical CNN architecture



MNIST digits dataset

Y. LeCun

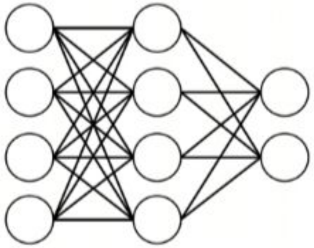
ImageNet



AlexNet beating all "handcrafted" approaches on ImageNet benchmark—the moment of truth for computer vision

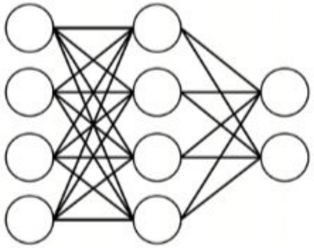
L. Fei-Fei

Deep Learning

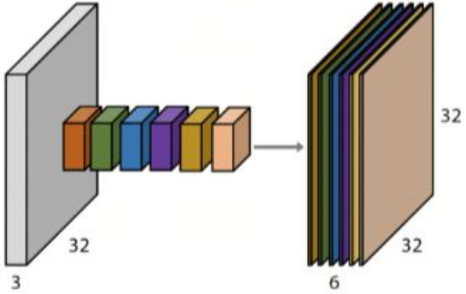


Perceptrons

Deep Learning

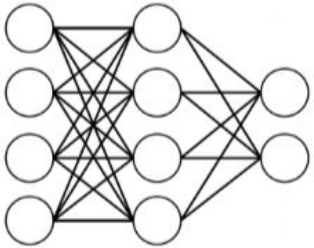


Perceptrons

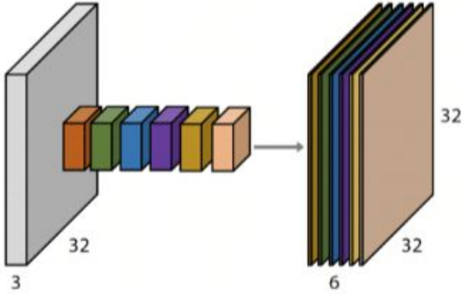


CNNs

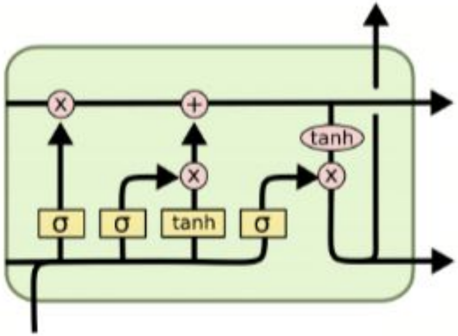
Deep Learning



Perceptrons

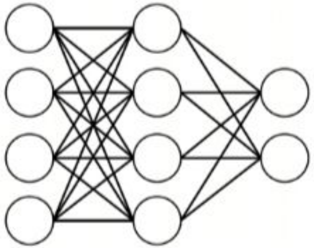


CNNs

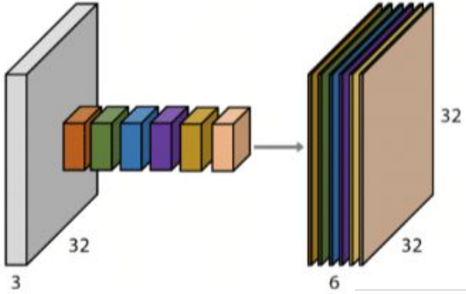


LSTMs

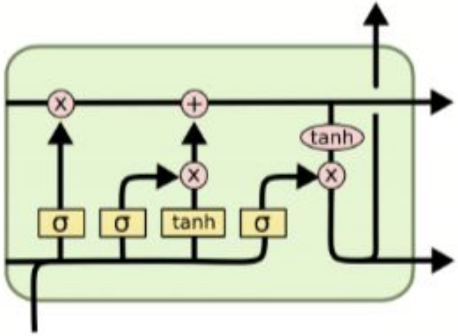
Deep Learning



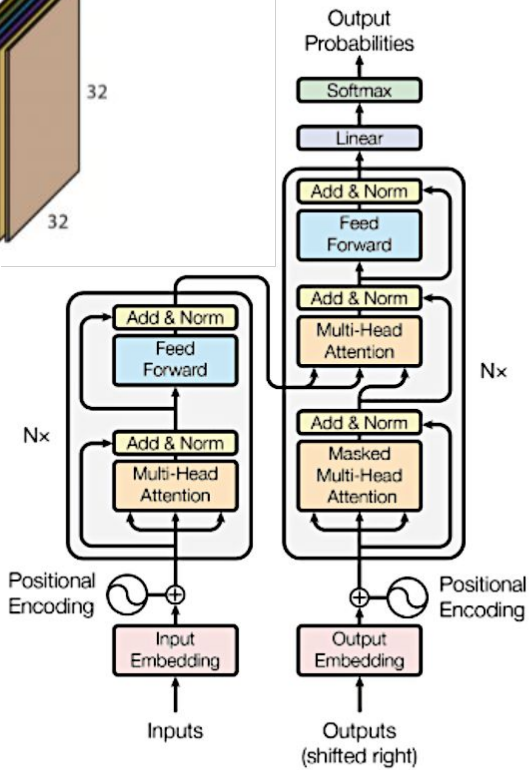
Perceptrons



CNNs

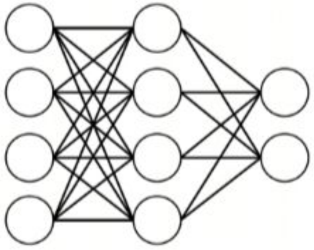


LSTMs

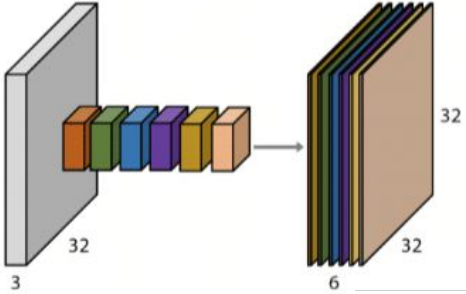


Transformer

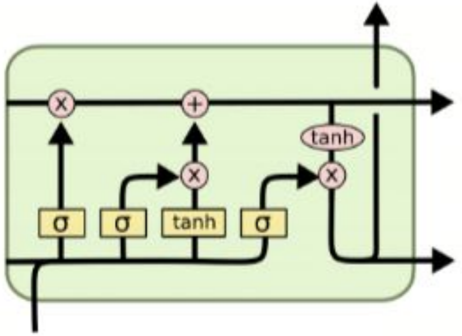
Deep Learning



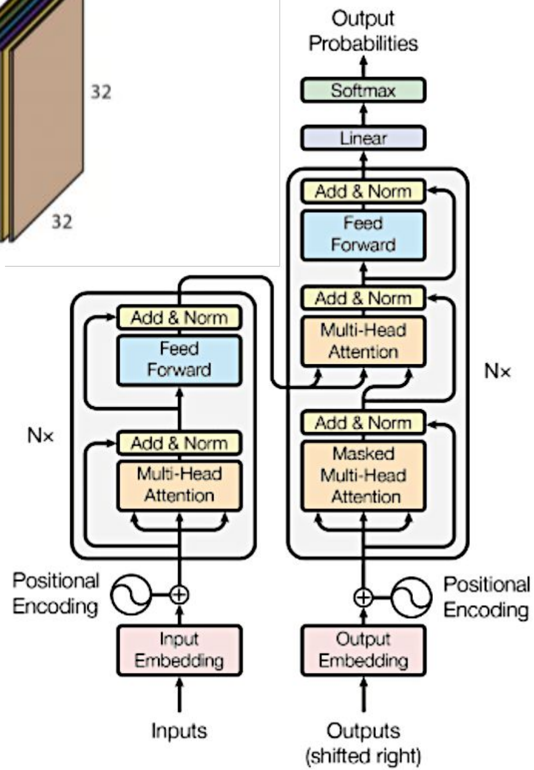
Perceptrons



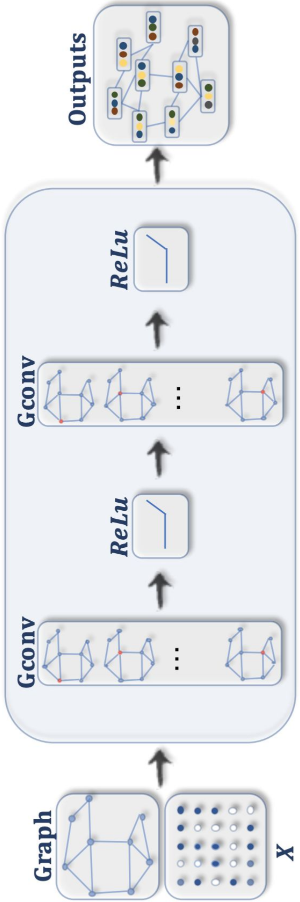
CNNs



LSTMs



Transformer



GNNs