Machine Learning & Deep Learning Concepts

Intelligent Systems Lab (ISL)

PHAROS
THE GREEK AI FACTORY

ARISTOTLE UNIVERSITY OF THESSALONIKI

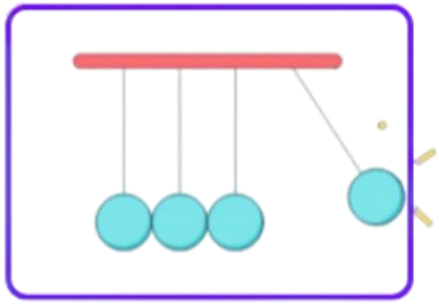
IS_LAB

Introduction to Modeling

Approaches

- Known Physical Laws
- Few Data

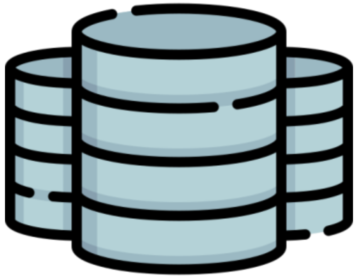
(1)



Differential Equations

- Unknown/Complex Laws
- Large Volumes of Data

(2)



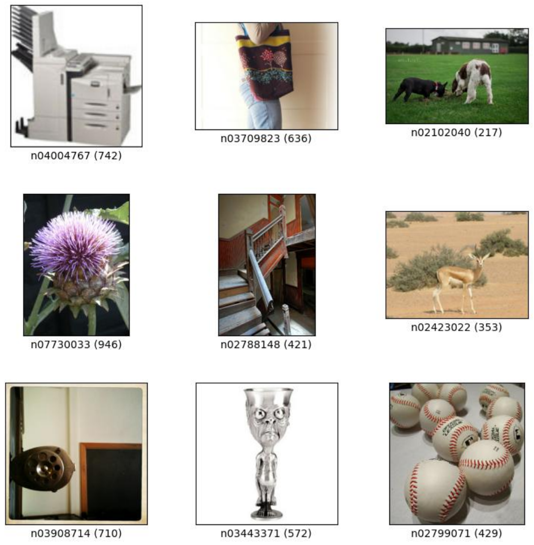
Machine Learning

Modeling Data

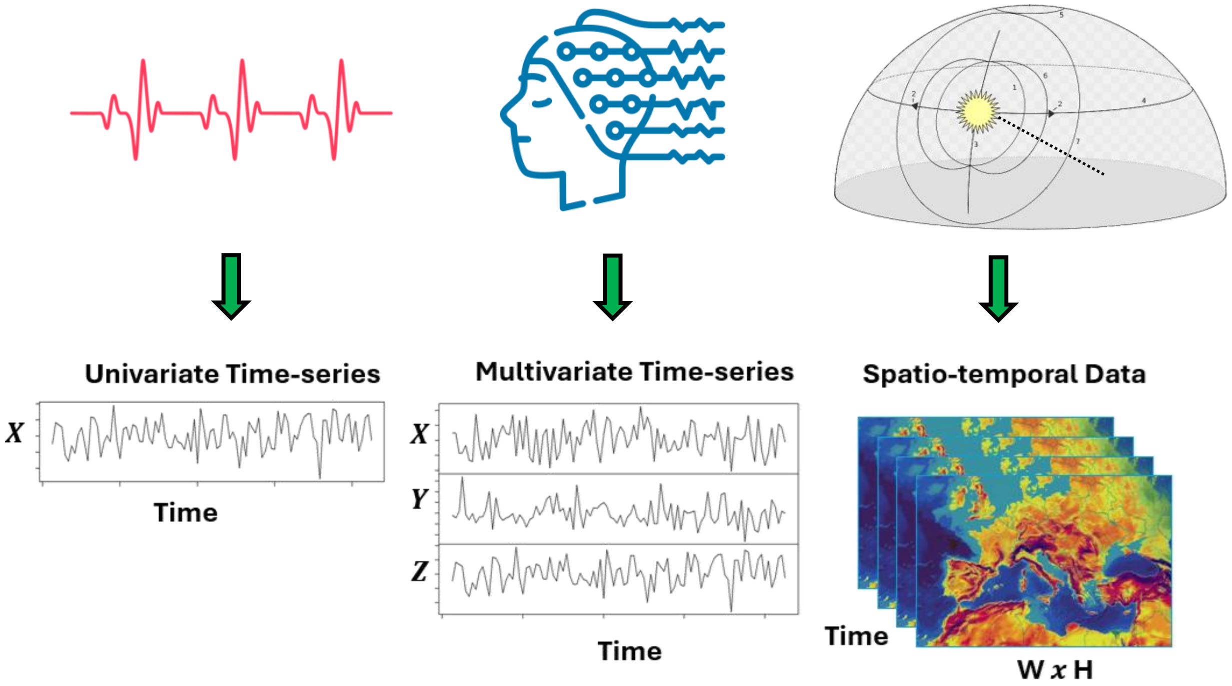
Tabular

Patients		
Name	Gender	Age
Bill	Male	21
Miko	Female	15
Juan	Other	45

Visual

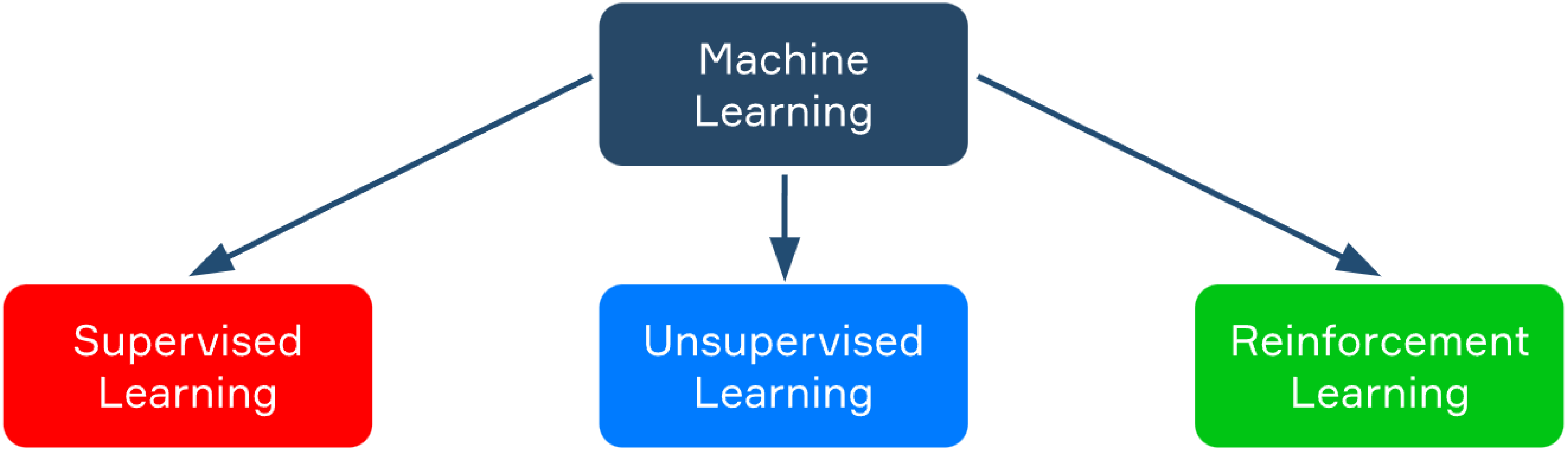


Sequences



Machine Learning

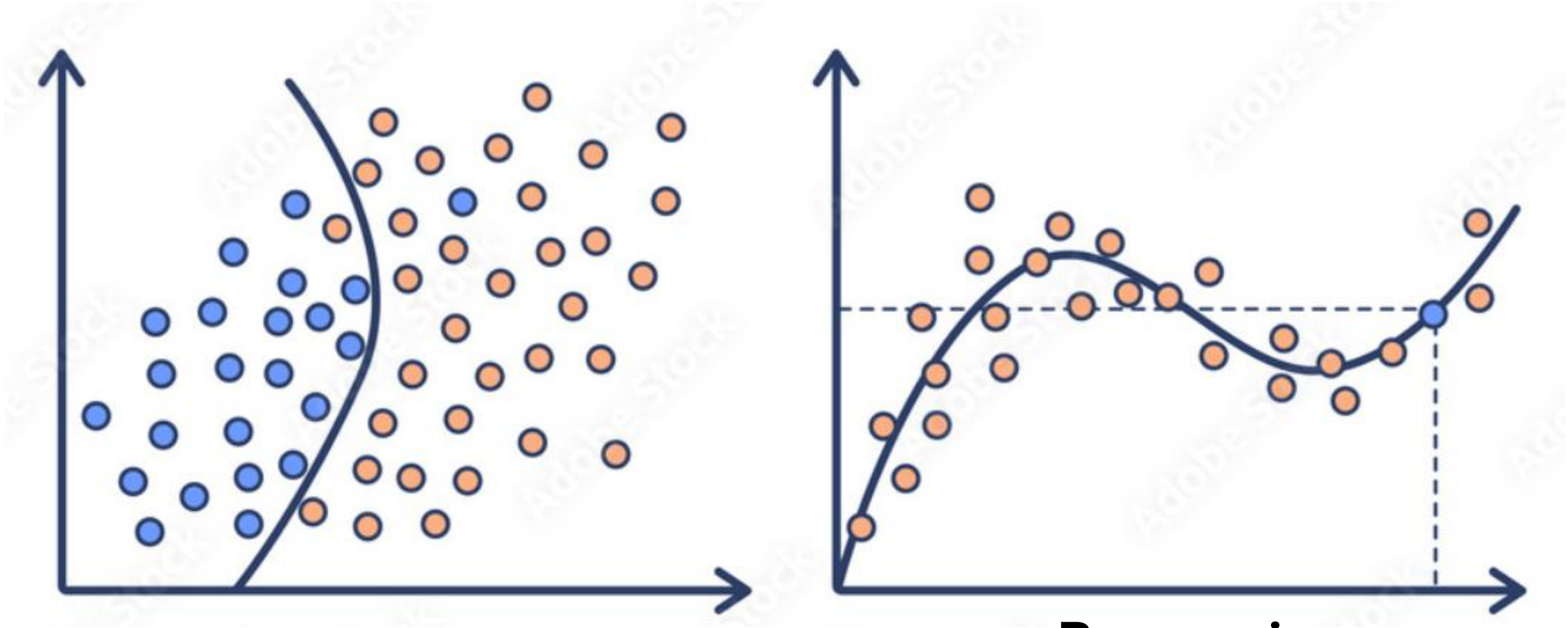
Algorithms that learn from data without explicit programming



Supervised Learning

Algorithms that feed on (input, target) data and try to find a mapping:

$$X:\ inputs \rightarrow Y:\ targets$$

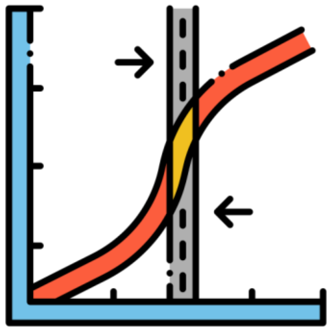


Classification **Regression**

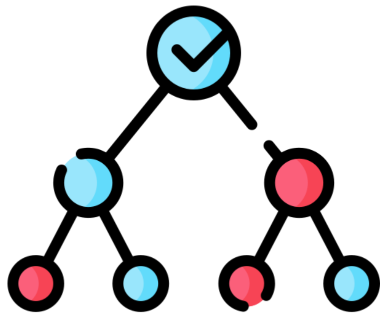
Supervised Learning Algorithms

Linear Regression

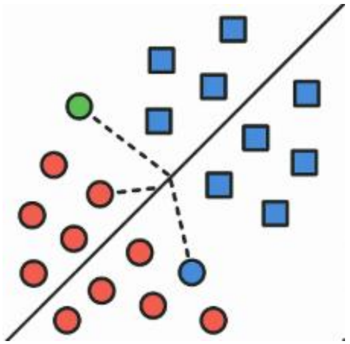
Logistic Regression



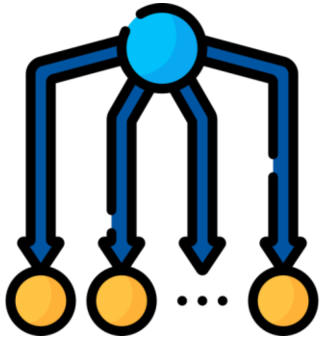
Decision Tree



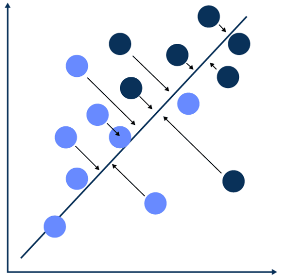
K-Nearest Neighbors



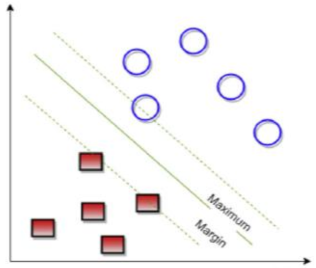
Naïve Bayes



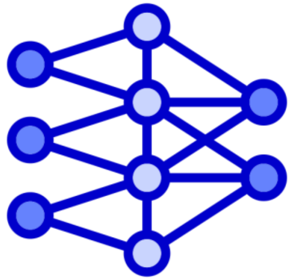
Linear Discriminant Analysis



Support Vector Machines

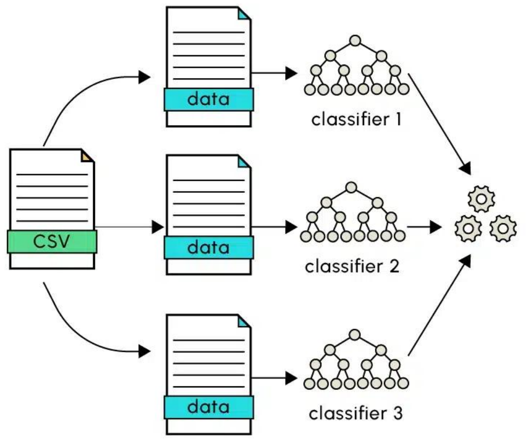


Artificial Neural Networks



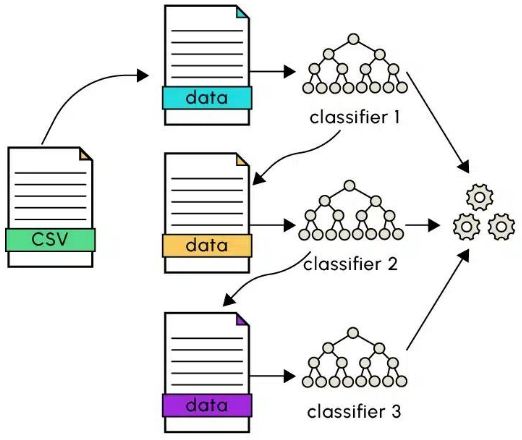
Ensemble Techniques

Bagging



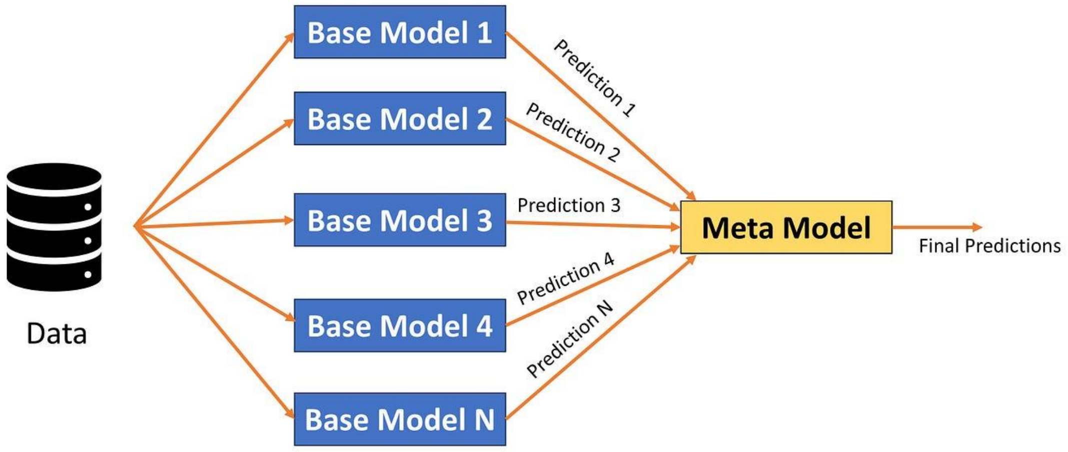
Parallel

Boosting

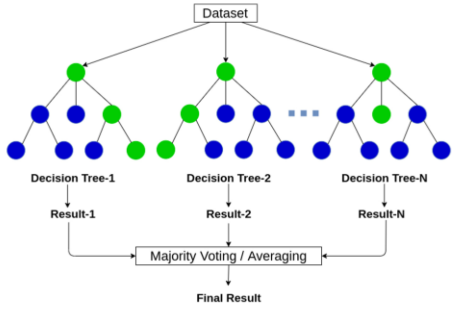


Sequential

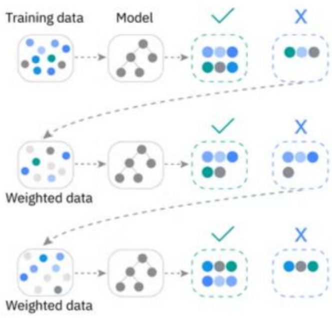
Stacking



Random Forest



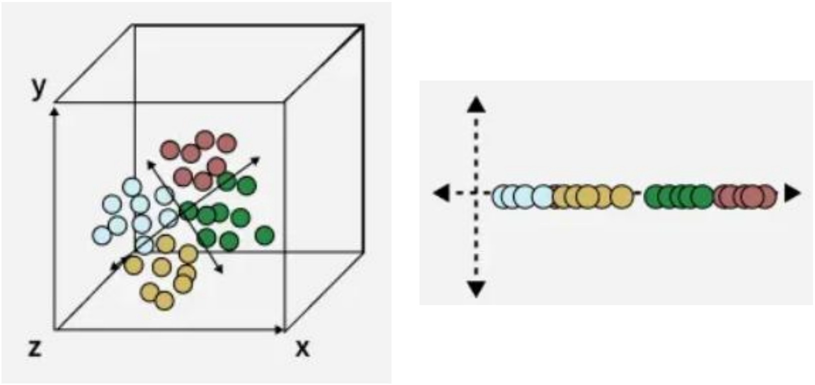
XGBoost



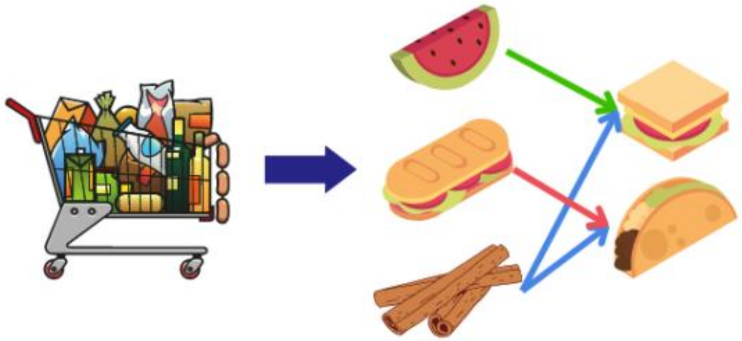
Unsupervised Learning

Algorithms that feed on input data and try to extract patterns and correlations

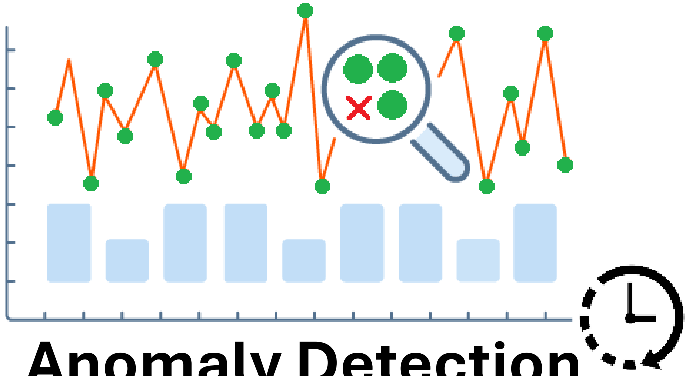
Clustering



Dimensionality Reduction



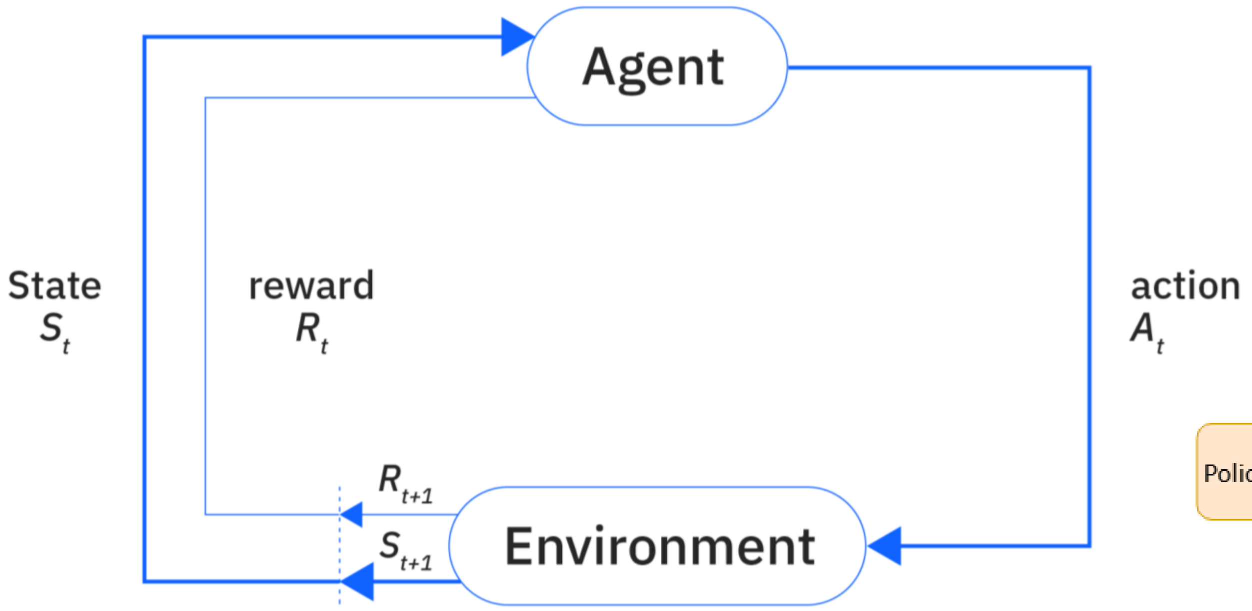
Association Rules

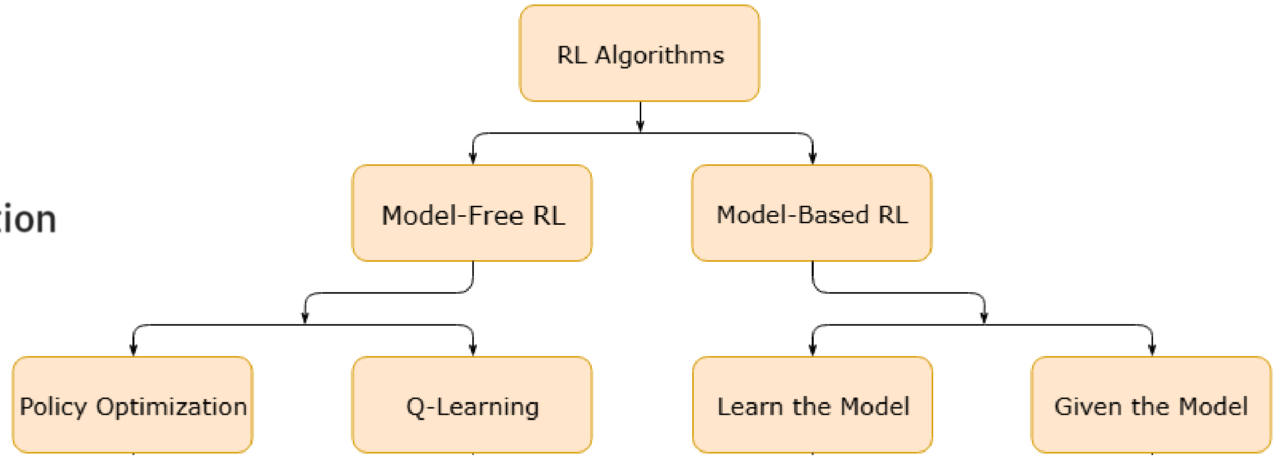


Anomaly Detection

Reinforcement Learning

Agents that learn optimal policies that solve a task through trial-and-error, by interacting with an environment



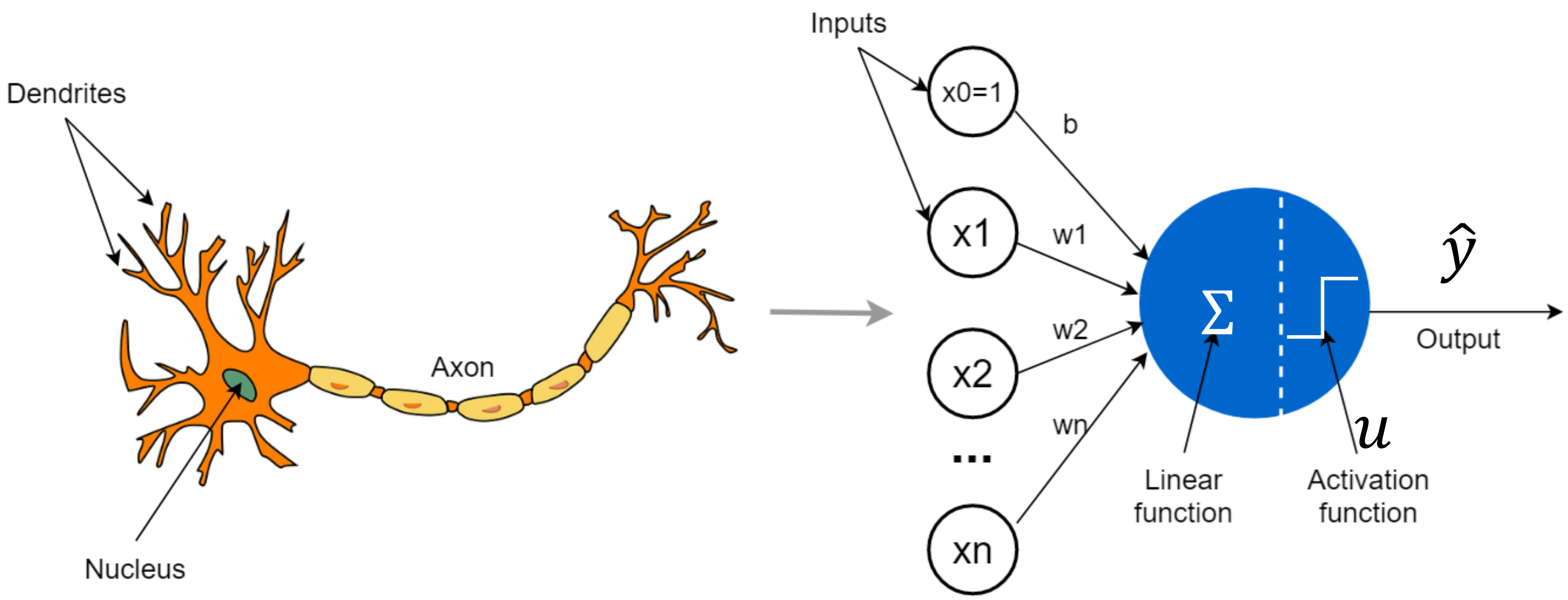


Goal: maximize cumulative returns $G_t = R_{t+1} + R_{t+2} + \cdots$

Why Neural Networks

- ✓ **Non-linear** relationships
- ✓ **Large volumes of data**
- ✓ **Many features**
- ✓ Easy to **Adapt**
- ✓ Impressive **performance**
- ✓ NNs → **Supervised, Unsupervised, Reinforcement Learning**
- ✓ **Universal Approximation Theorem**

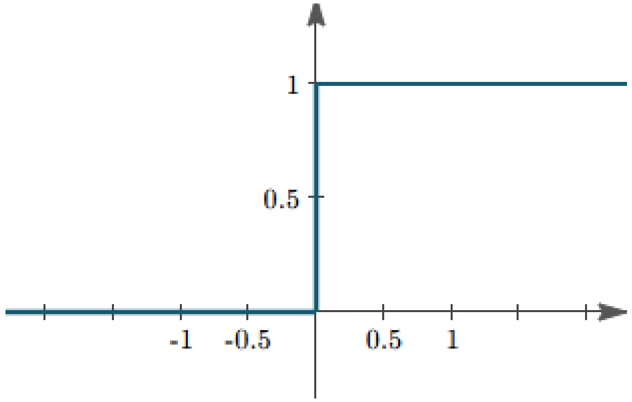
Perceptron



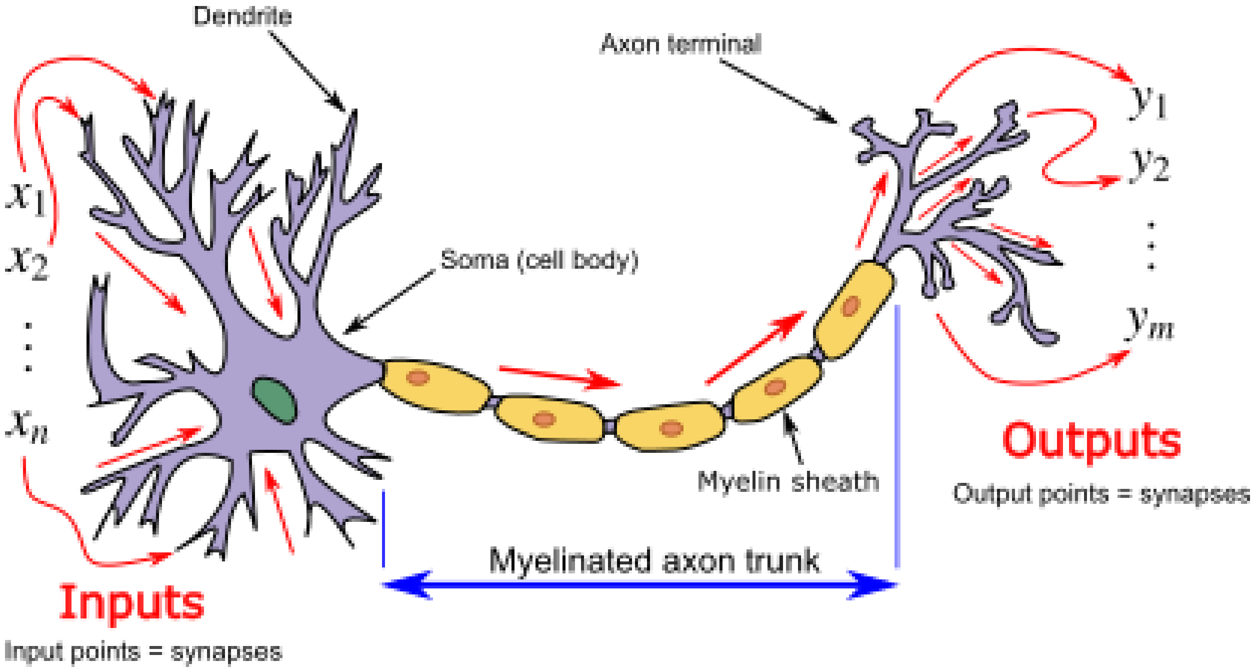
Frank Rosenblatt, 1957

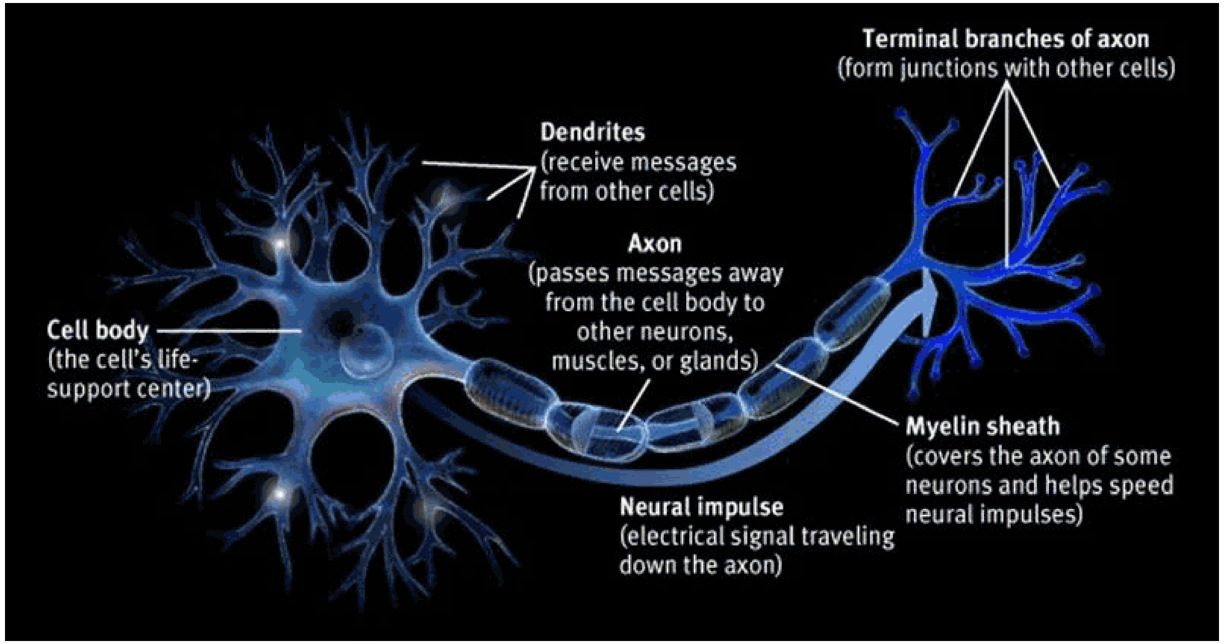
$$\hat{y} = u(\underbrace{w_1x_1 + w_2x_2 + \cdots w_nx_n + w_0}_{\textbf{Linear Model}}) = u\left(\sum_{i=1}^{n} w_ix_i + w_0\right)$$

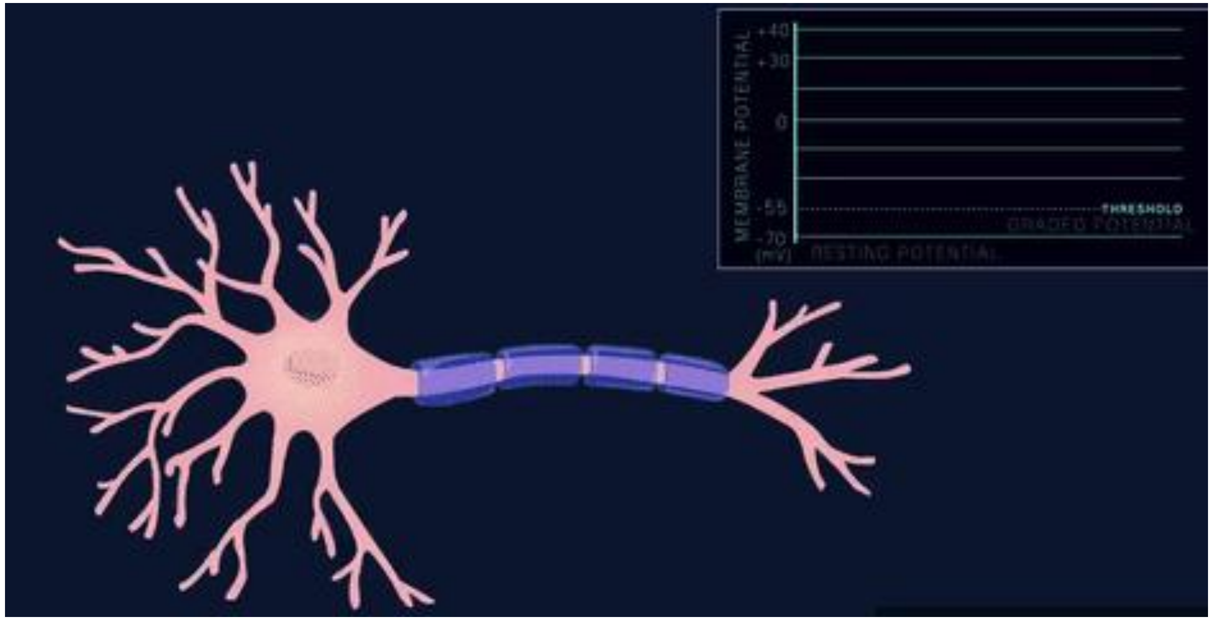
Step Function



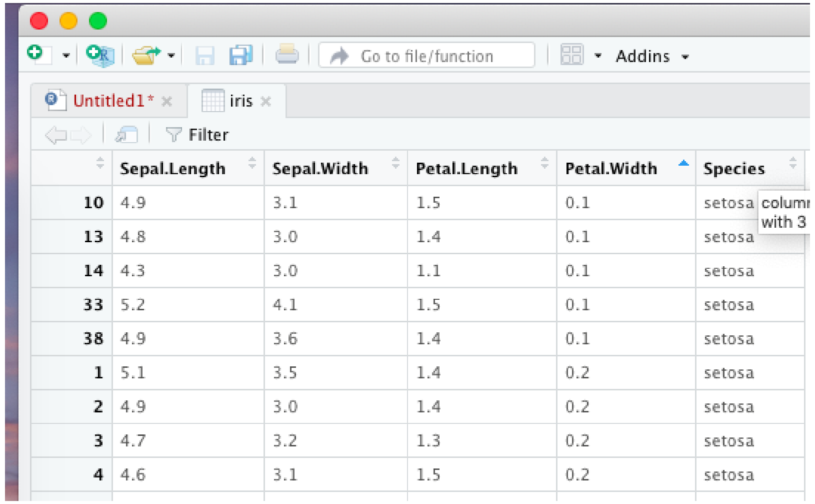
Biological Neuron





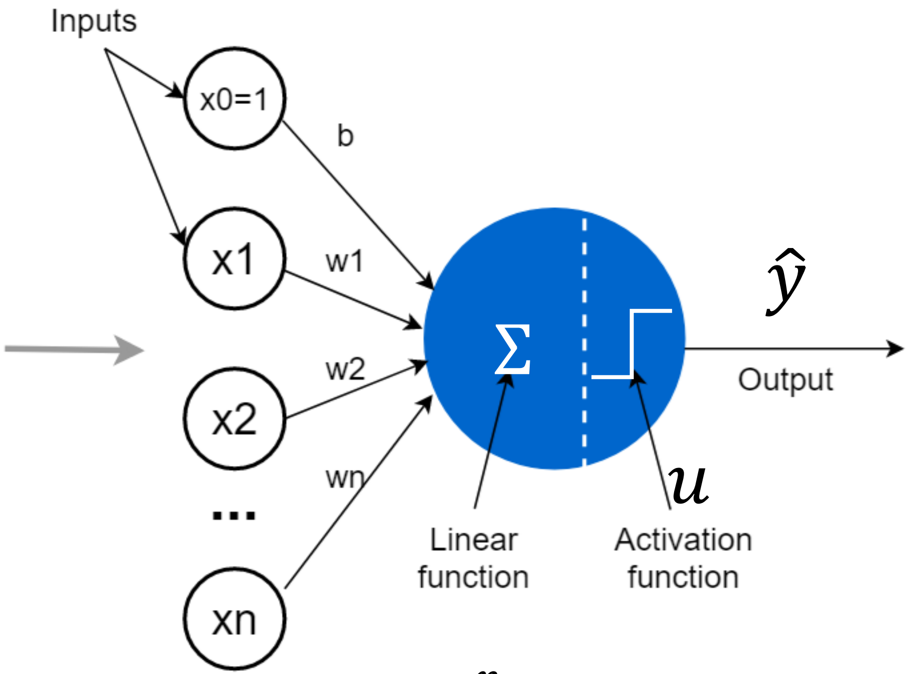


Perceptron Example



	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
10	4.9	3.1	1.5	0.1	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
33	5.2	4.1	1.5	0.1	setosa
38	4.9	3.6	1.4	0.1	setosa
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa

$\boldsymbol{x_1}$ $\boldsymbol{x_2}$ $\boldsymbol{x_3}$ $\boldsymbol{x_4}$ $\boldsymbol{y}$



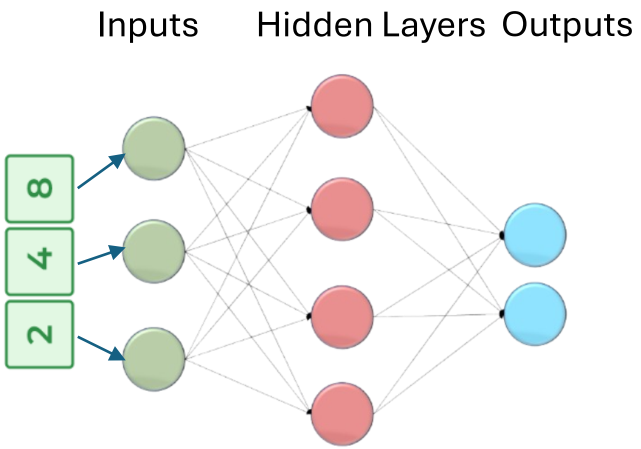
Minimize $\boldsymbol{E(\hat{y}, y)}$

$$\hat{y} = u(\underbrace{w_1 x_1 + w_2 x_2 + \cdots w_n x_n + w_0}_{\textbf{Linear Model}}) = u\left(\sum_{i=1}^{n} w_i x_i + w_0\right)$$

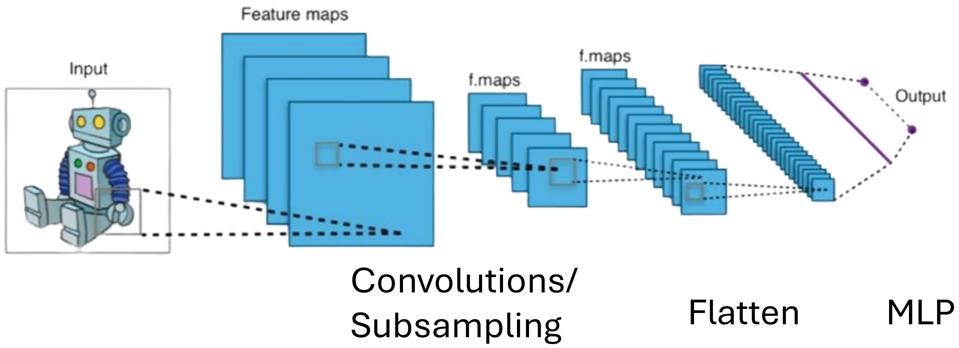
Step Function

Deep Neural Network (DNN)

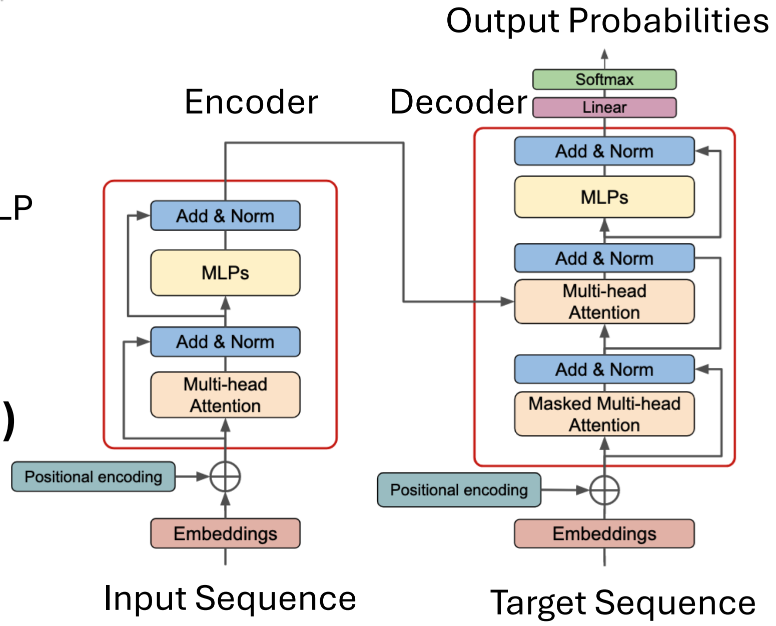
Multi-Layer Perceptron (MLP)



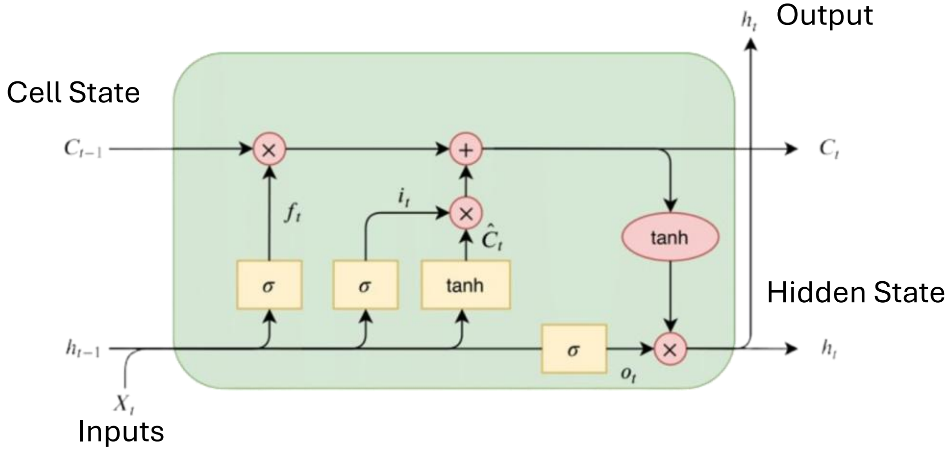
Convolutional Neural Network (CNN)



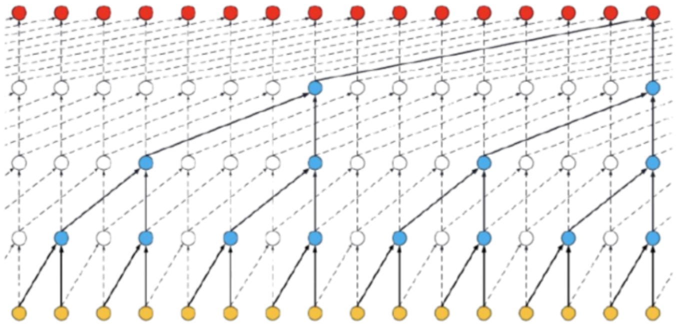
Transformer



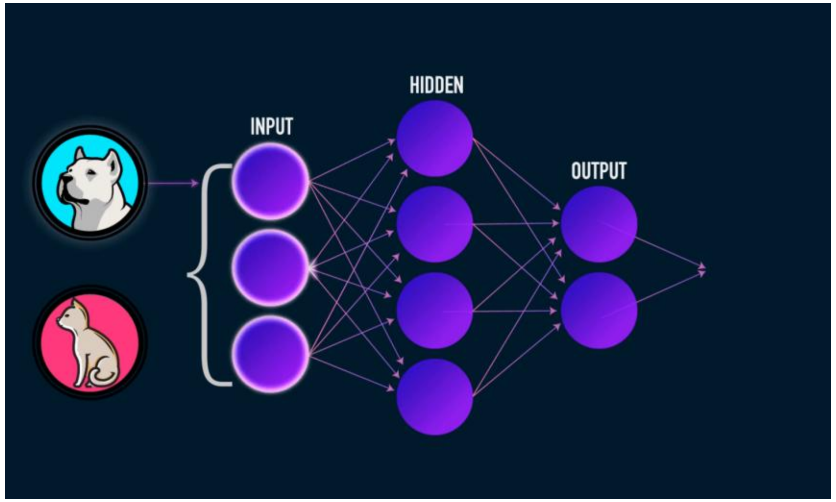
Long Short-Term Memory (LSTM)



Temporal Convolution Network (TCN)



Multi-Layer Perceptron (MLP)



Neural Networks with at least one hidden layer.

- **Layers**:
  - ➢ Input Layer
  - ➢ Hidden Layers
  - ➢ Output Layer
- Each layer is also called “*Dense*” or “*Linear*”.
  - ➢ Input layer → Hidden Layers → Output Layer
  - ➢ Each neuron applies a transformations and passes it to the next layer

➢ **Shallow Networks**: Maximum 2 hidden layers.

- **Deep Neural Networks**: More than 2 hidden.

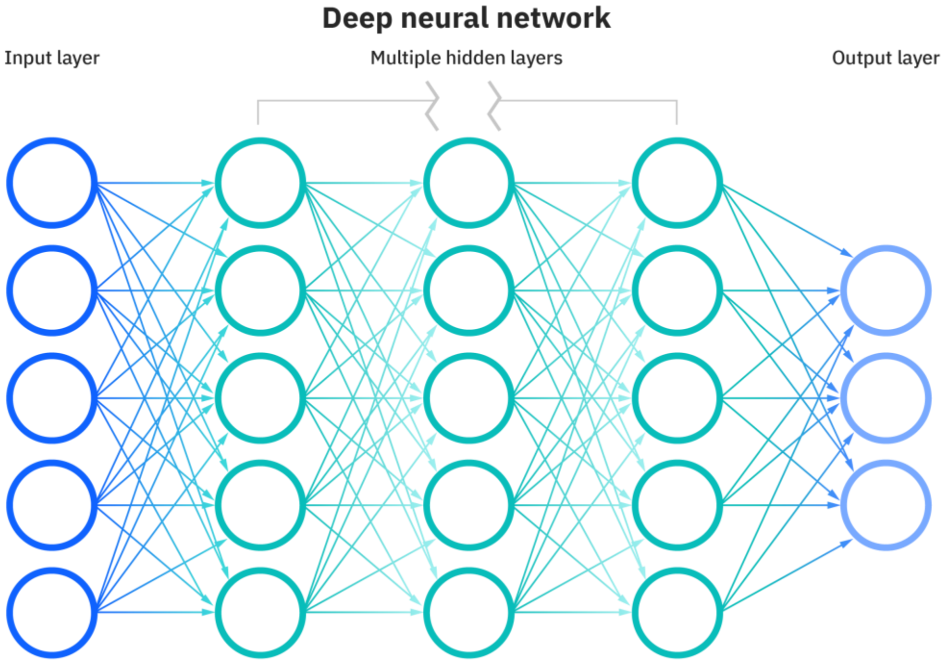
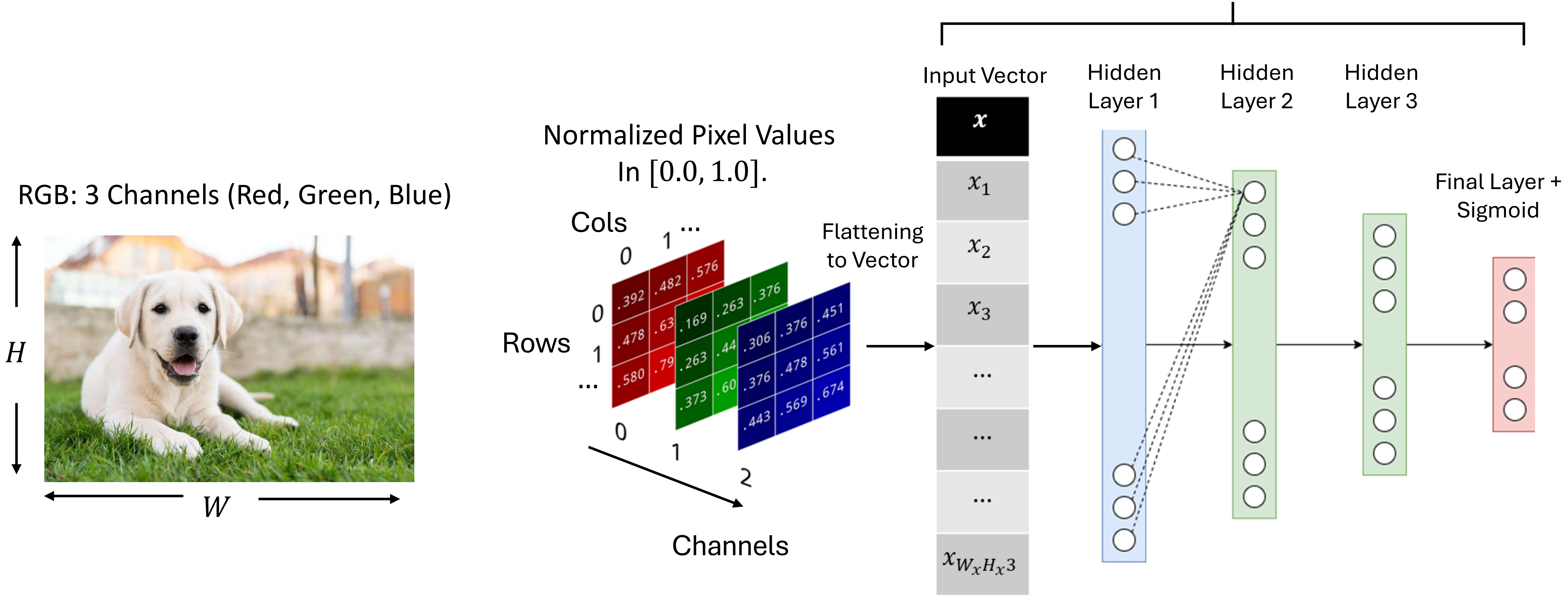
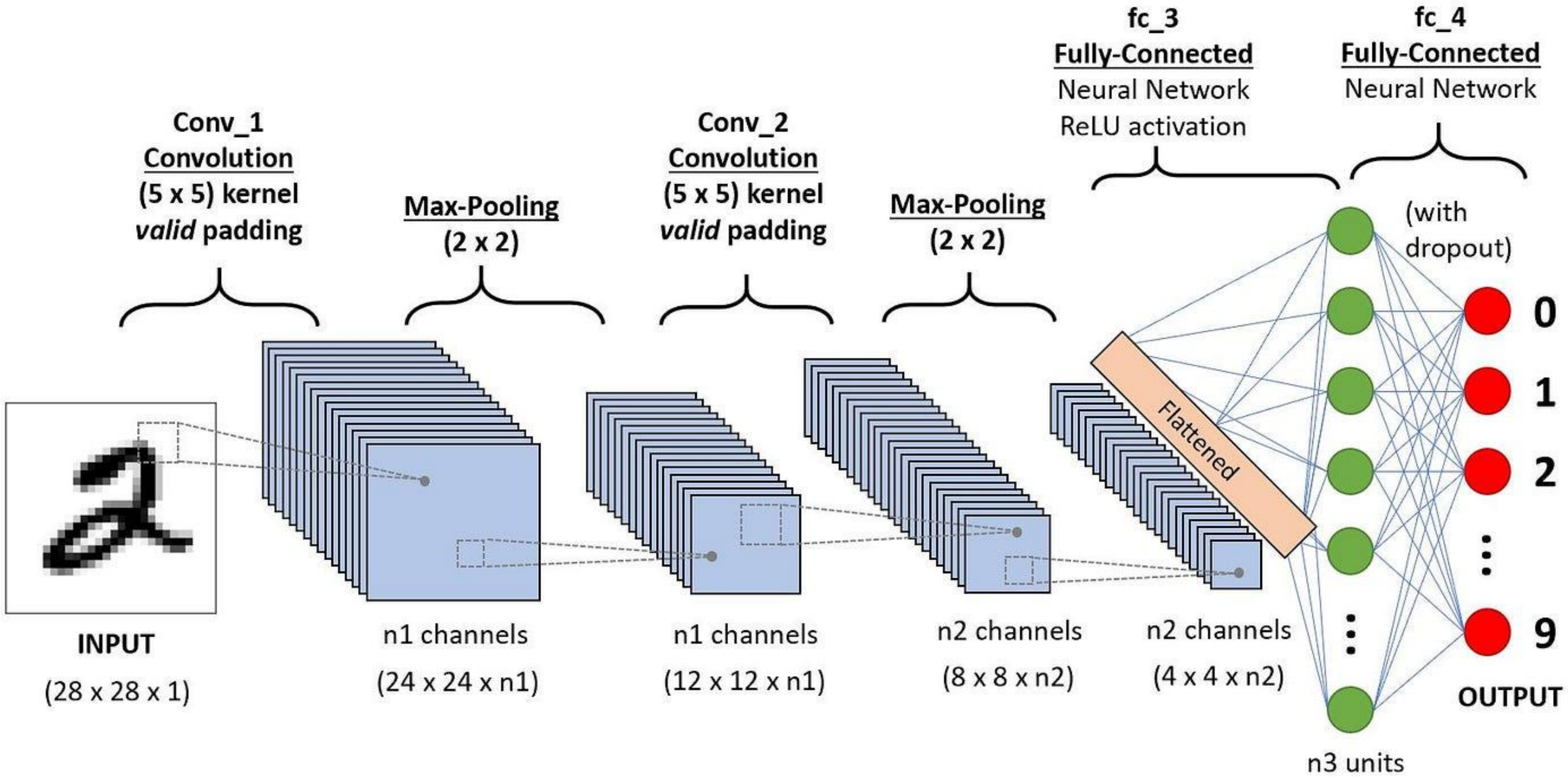


Image Classification with MLPs

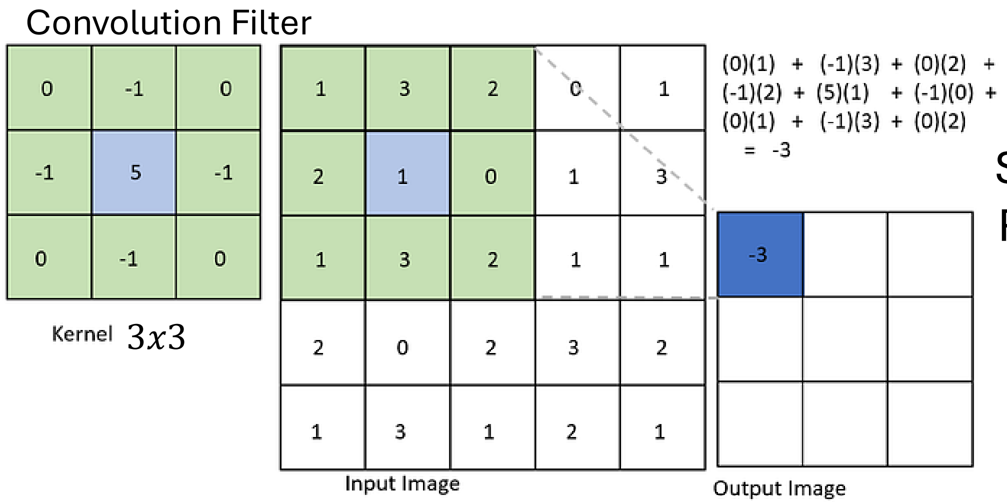


Convolutional Neural Network (CNN)

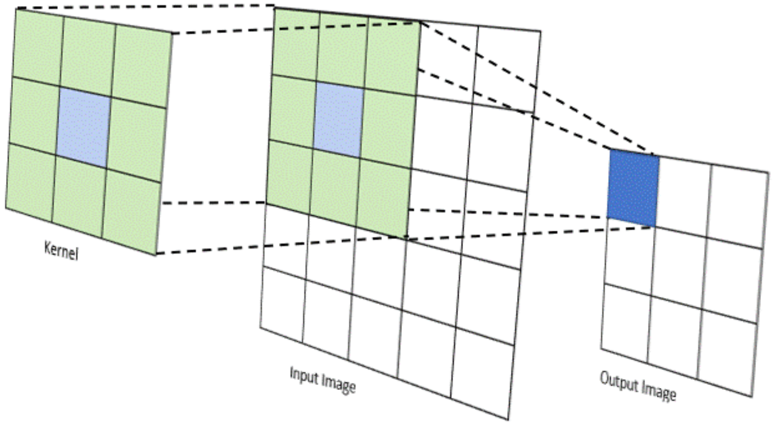
- Used for Image Classification, Recognition, Detection
- Each layer applies convolutional filters, which are learnt by the Neural Network

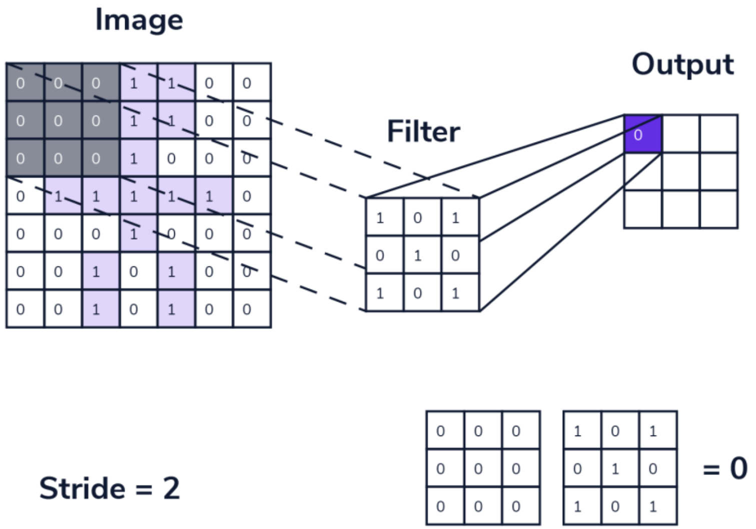


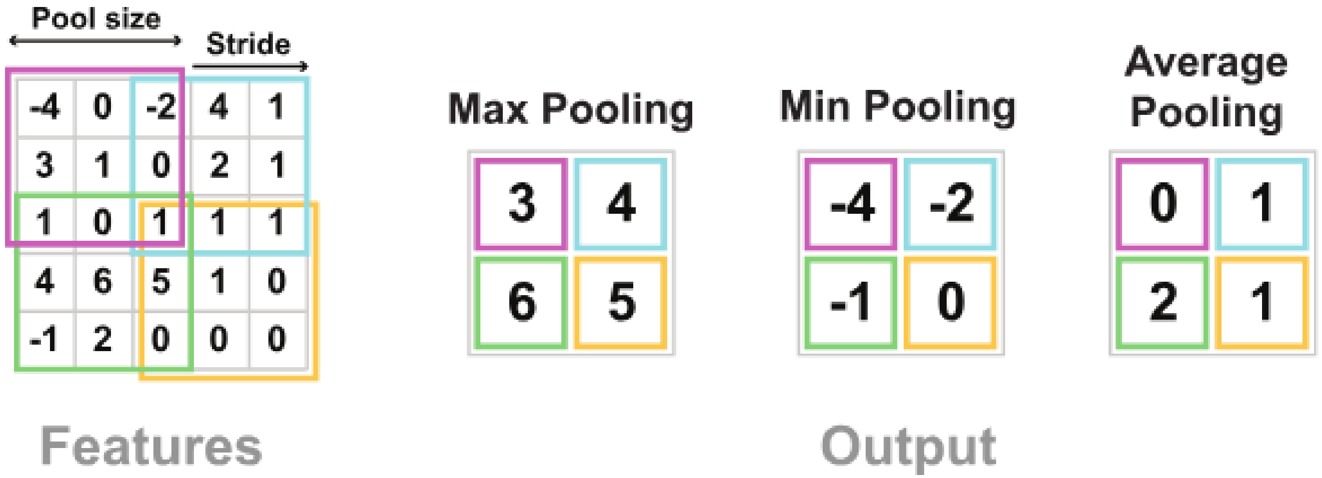
CNN Operations



Slide & Repeat

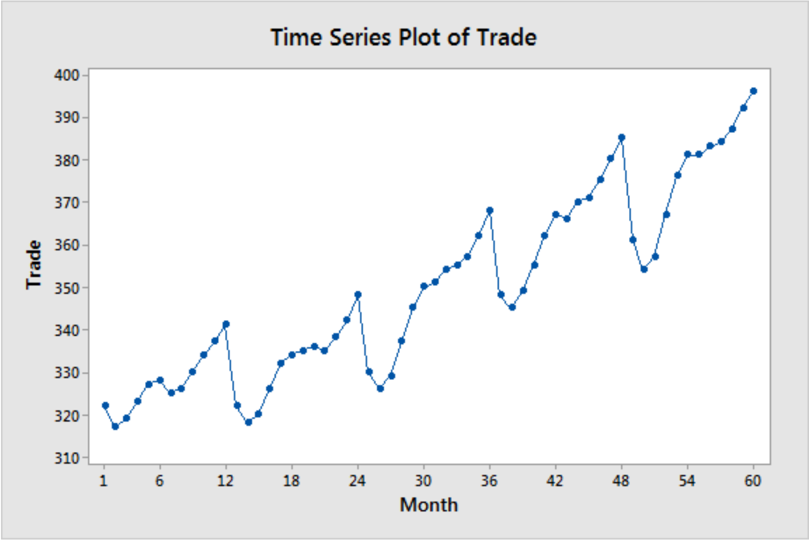


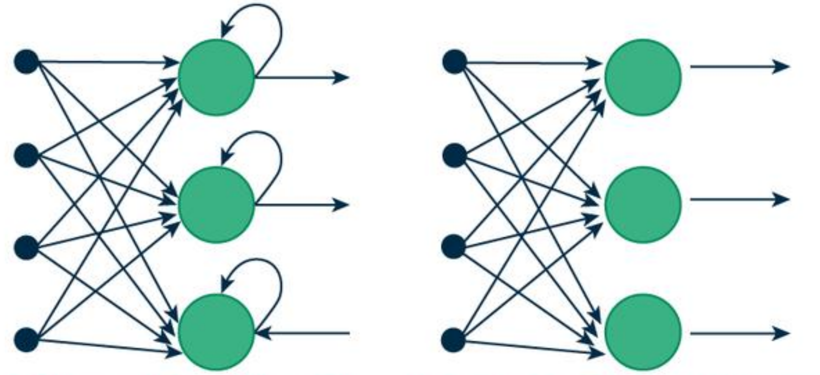




Recurrent Neural Network (RNN)

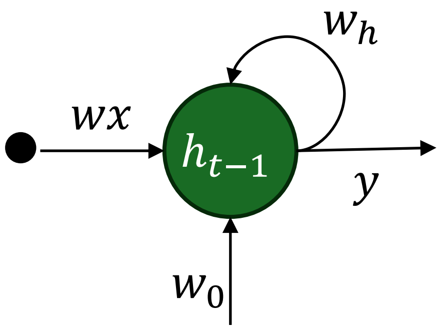
- Model sequential data, such as timeseries $y_1, y_2, y_3 \dots, y_t$.
  - ➢ MLP considers that y_i input is independent.
  - ➢ Instead, RNNs keep a memory weight w_h that remembers the previous input that the neuron received.
- Then, it uses w_h to compute its hidden state h_t
- h_t is updated every time a value is passed to the neuron





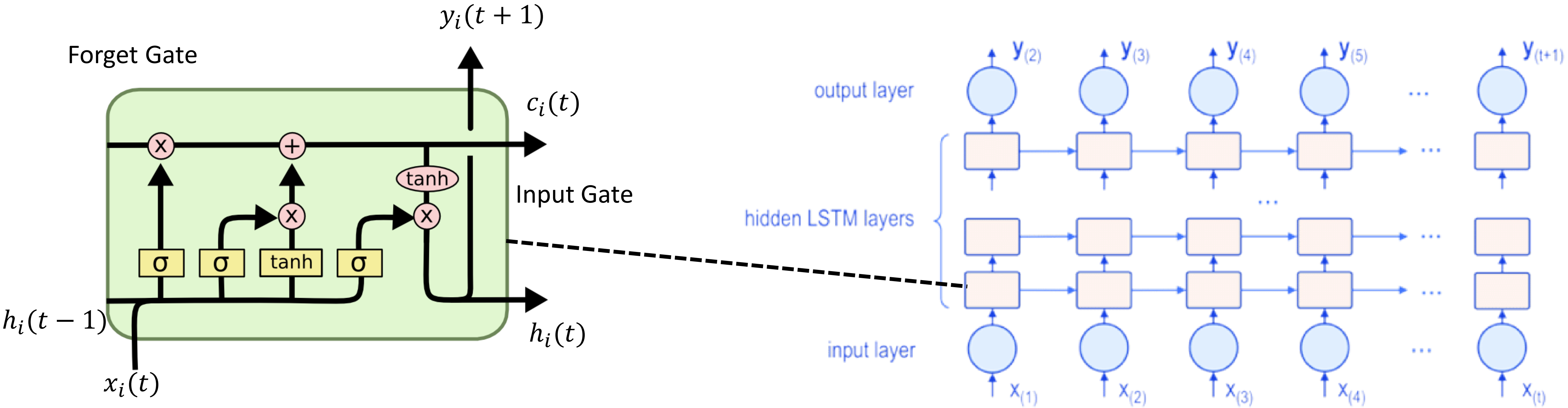
(a) Recurrent Neural Network (b) Feed-Forward Neural Network

Recurrent (RNN) Cell



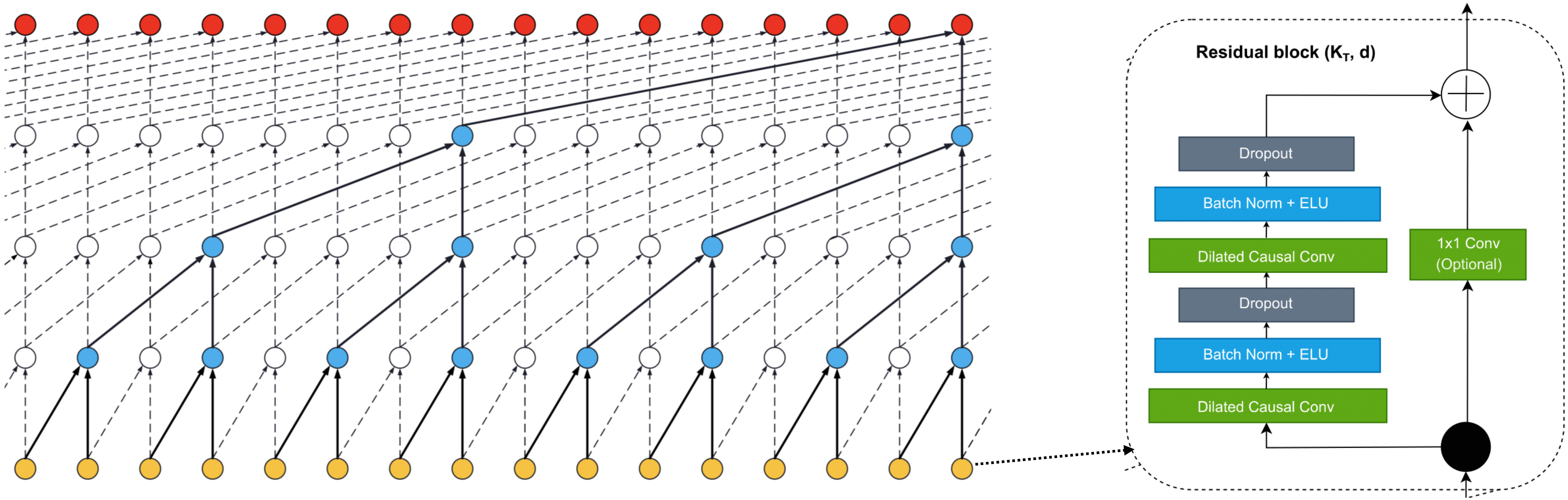
$$\boldsymbol{h_t = wx + hw_h}$$
$$\boldsymbol{y = h_t w_t + w_0}$$

Long Short-Term Memory (LSTM)



- Improved RNN architecture, which:
  - Passes each cell's (neuron's) state to the other neurons.
  - Includes gate mechanisms for the input and its memory which bound the values, and thus mitigating the Exploding Gradient problem.

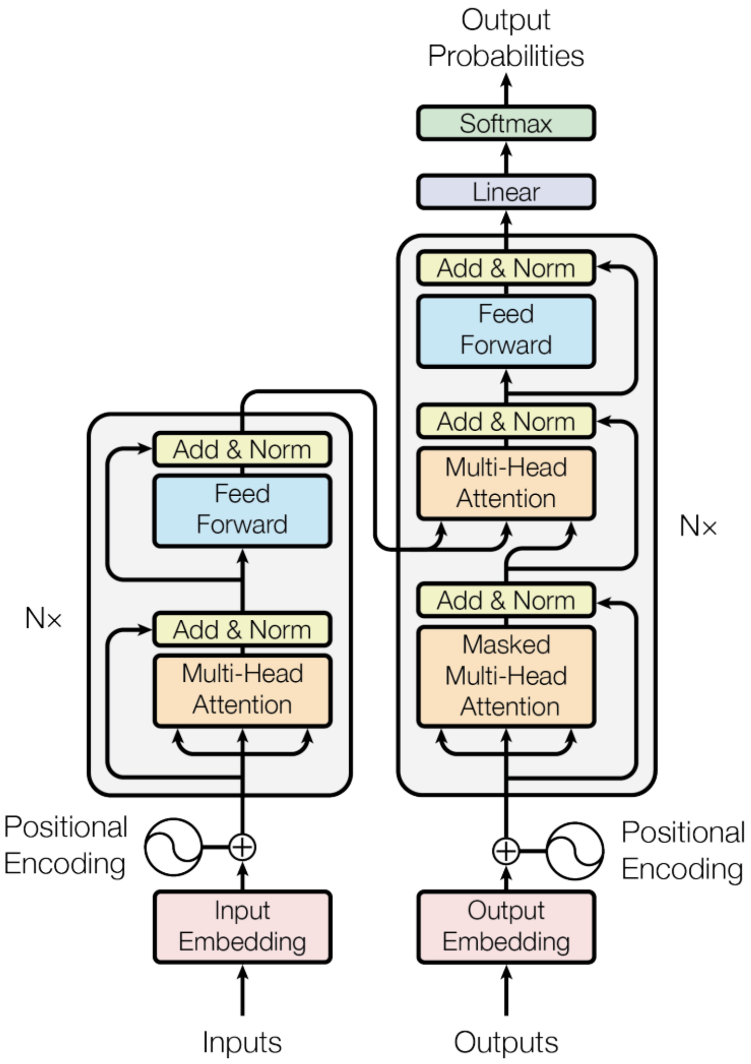
Temporal Convolution Network (TCN)



Transformers

BERT

Encoder

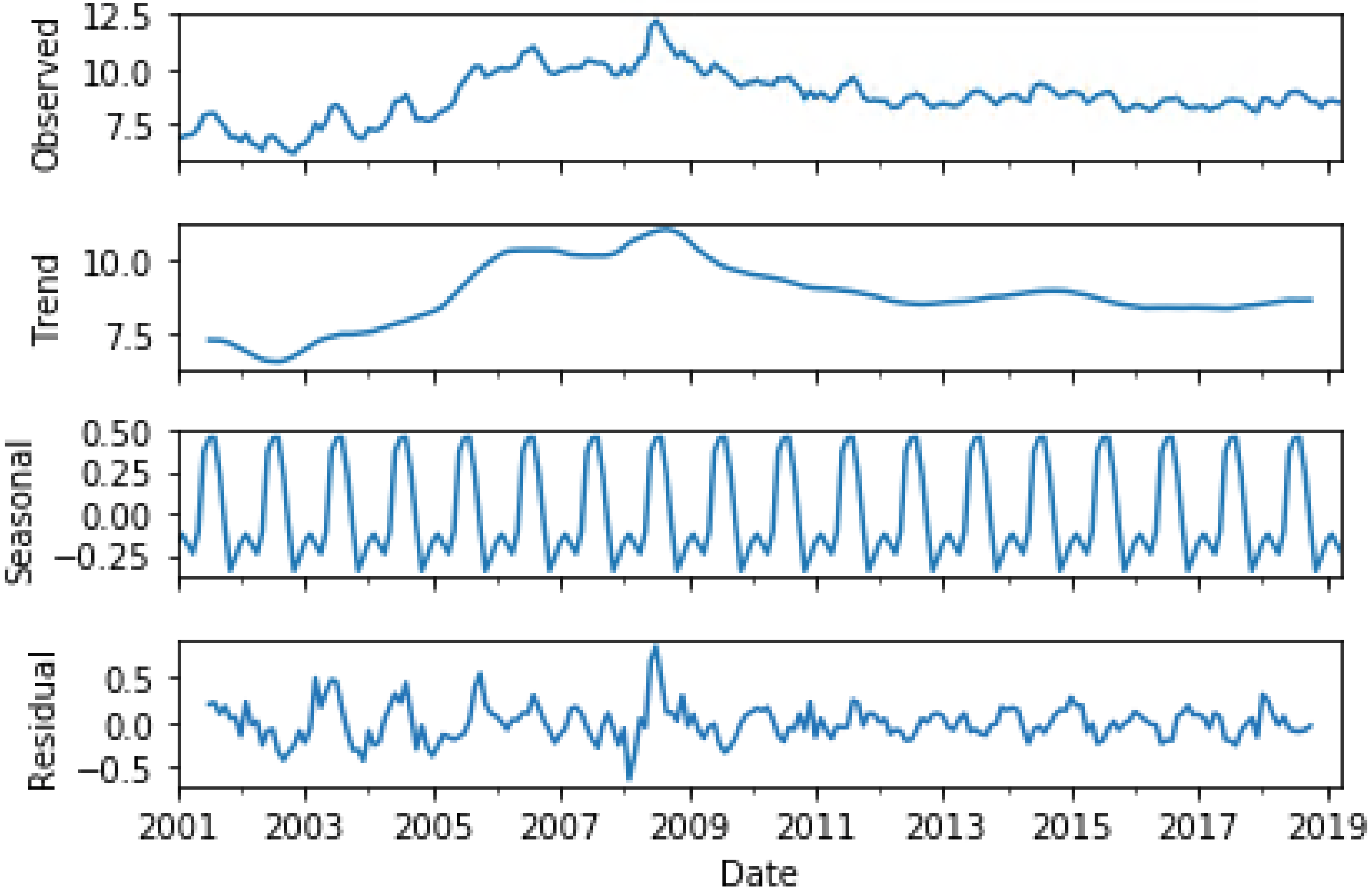


GPT

Decoder

Timeseries

Sequence of observations ordered by time



T_t

S_t

I_t

Additive Model

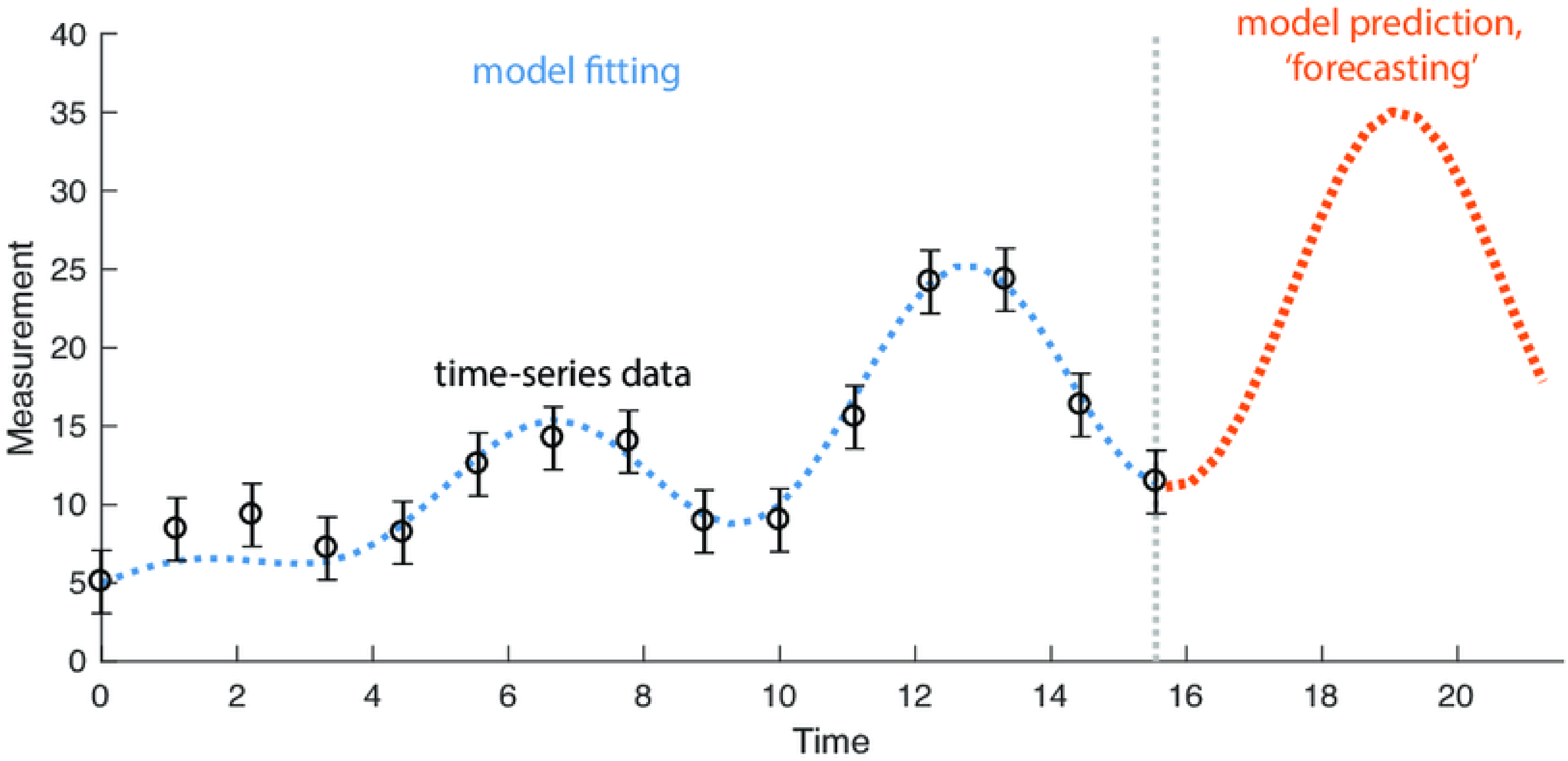
$$y_t = T_t + S_t + I_t$$

Multiplicative Model

$$y_t = T_t \times S_t \times I_t$$

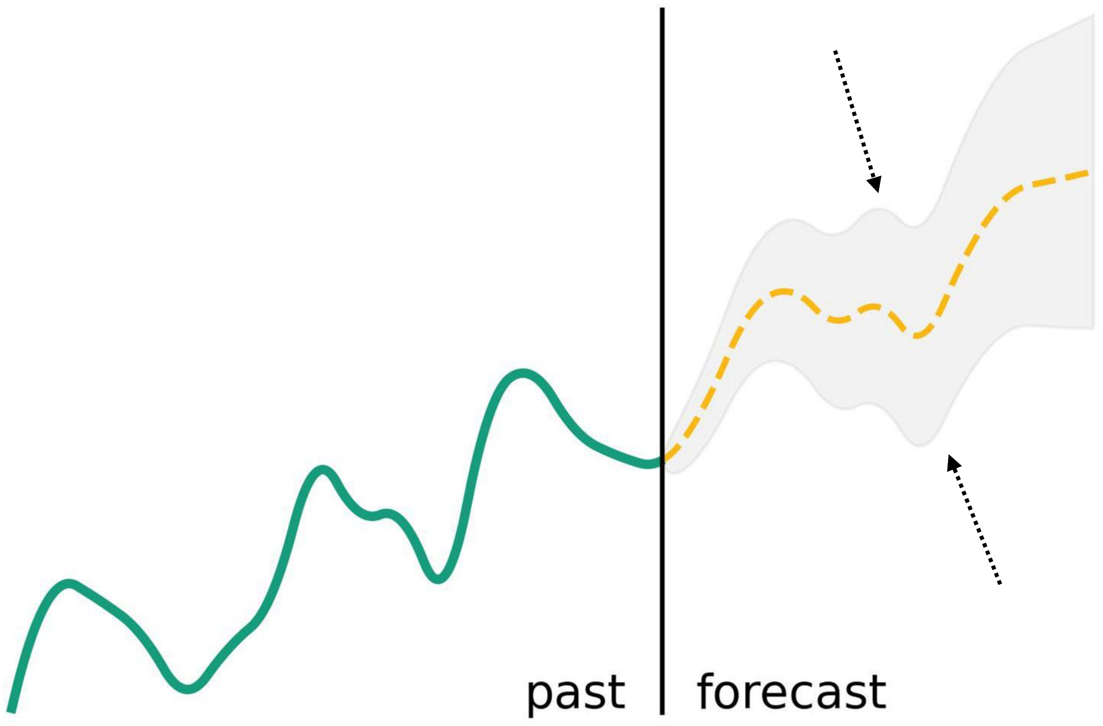
Timeseries Forecasting

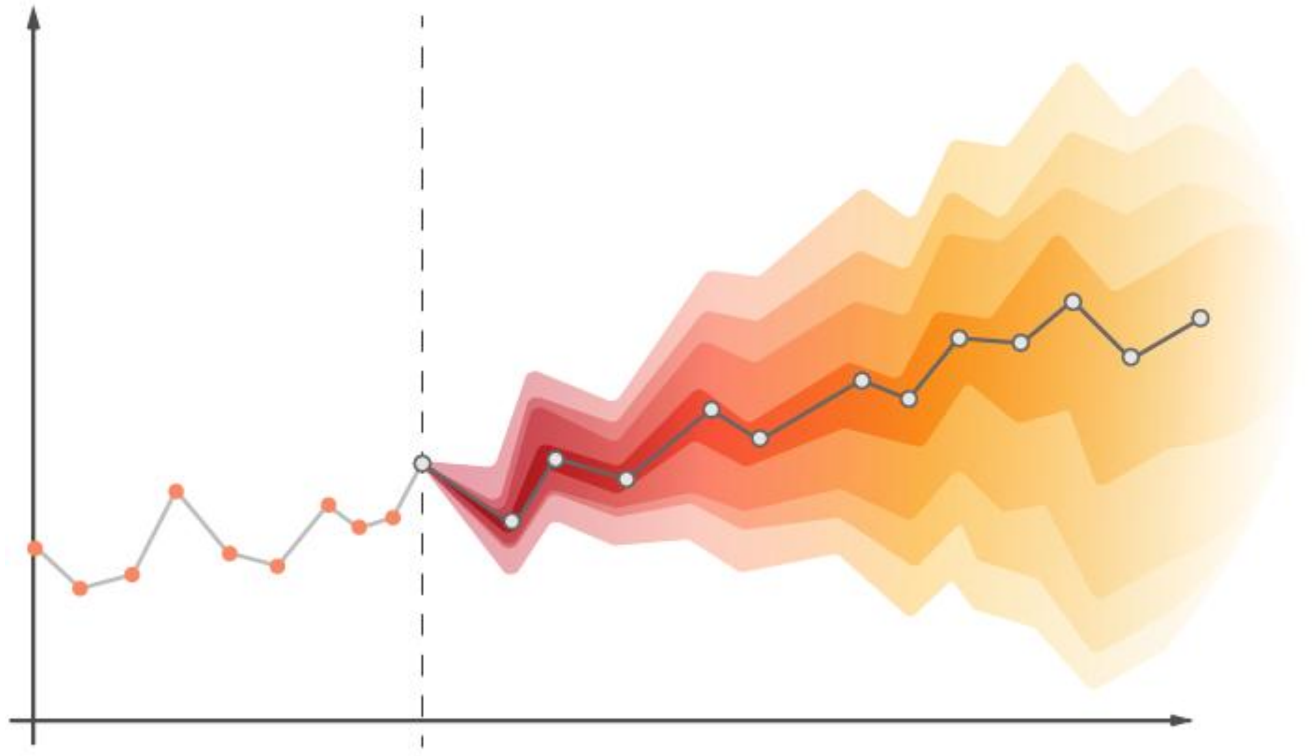
Goal: given past observations $y_{t-T}, y_{t-T+1}, \dots, y_t$, predict the values of the next observations $y_{t+1}, y_{t+2}, \dots$



Probabilistic Forecasting

Goal: given past observations $y_{t-T}, y_{t-T+1}, \dots, y_t$, predict the quantiles of future observations



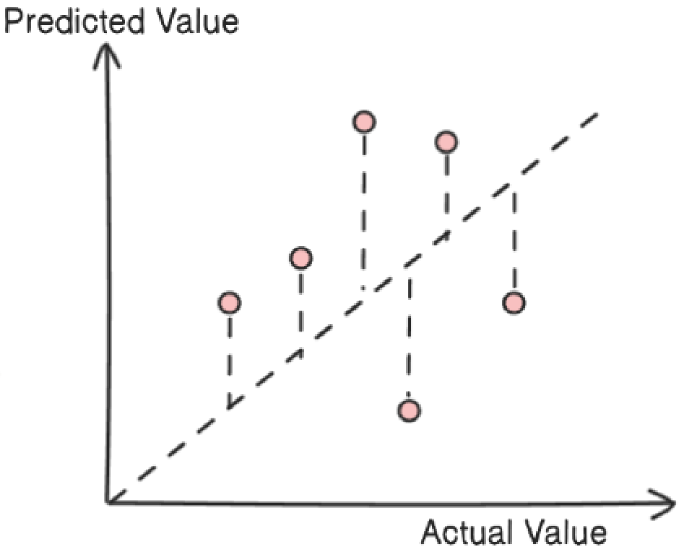


Regression Metrics

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$



$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right|$$

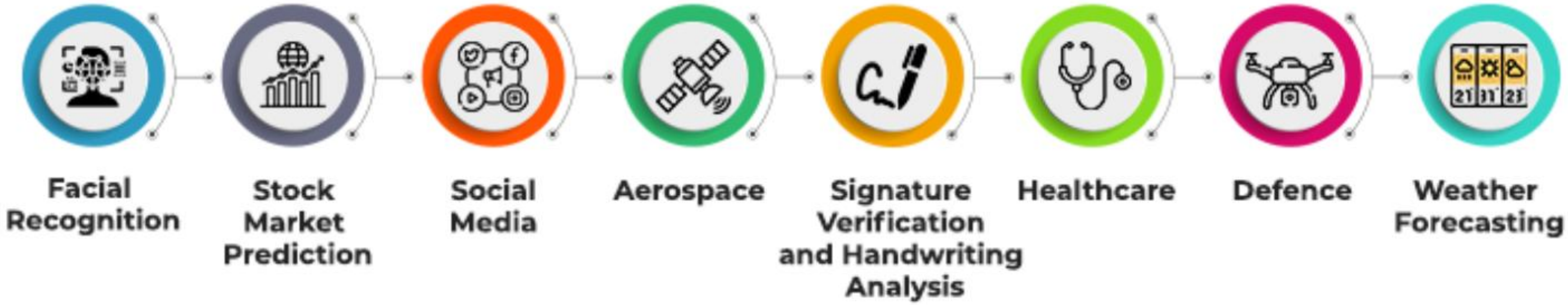
$$WMAPE = \frac{\sum_{t=1}^{t=n}(w_t|y' - y|)}{\sum_{t=1}^{t=n}(w_t|y|)}$$

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

Conclusions

- Deep Neural Networks achieve exception performance, but require large volumes of data
- In comparison to statistics, work well under noisy data and datasets with many features
- **Applications**:



Machine Learning & Deep Learning Concepts

Intelligent Systems Lab (ISL)

PHAROS
THE GREEK AI FACTORY

ARISTOTLE UNIVERSITY OF THESSALONIKI