EI320A(3) 深度學習使用 Python

Instructors

Tipajin Thaipisutikul (t.greentip@gmail.com)

Prof. Huang-Chia Shih (hcshih@Saturn.yzu.edu.tw)

Week	Date	Content	Note	Total
1	2/26	Welcome to the course	Homework (1)	1
2	3/5	Crash Course of Python, NumPy, Pandas, and Matplotlib	In class hands-on (4)	5
3	3/12	Get to know about Data, ML: Classification Models	In class hands-on (5)	10
4	3/19	ML: Regression Models	In class hands-on (5)	15
5	3/26	ML: Clustering/Apriori Models	In class hands-on (5)	20
6	4/2	Holiday		
7	4/9	Introduction to Deep Learning (ANN)		
8	4/16	ANN Labs, Introduction to Convolutional Neural Network (CNN)	In class hands-on (10)	30
9	4/23	Convolutional Neural Network (CNN) & CNN Labs	In class hands-on (5)	35
10	4/30	Introduction to Recurrent Neural Network (RNN)	In class hands-on (5)	40
11	5/7	Recurrent Neural Network (RNN) & RNN Labs	In class hands-on (5)	45
12	5/14	Wrap Up all ANN, CNN, RNN Project Proposal Presentation	Proposal Presentation (10)	55
13	5/21	Generative Adversarial Network (GAN)	In class hands-on (5)	60
14	5/28	Reinforcement Learning (RL)	In class hands-on (5)	65
15	6/4	NLP & S2S & Attention Neural Network	In class hands-on (5)	70
16	6/11	N/A	In class hands-on (5)	75
17	6/18	Final Project Presentation	Final Presentation (30)	105

3 different types of machine learning

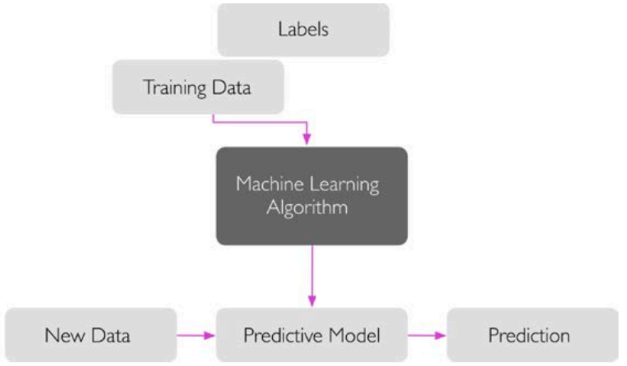
ARTIFICIAL INTELLIGENCE

Science thae enables computers to mimic human imtelligence.

Subfields: Machine Learning, robotics, and computer vision

MACHINE LEARNING

Subset of AI that enable maches to improve at tasks with experience

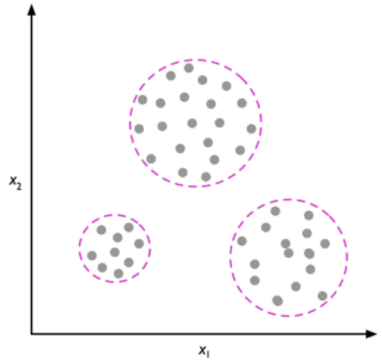


Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

Unsupervised Learning

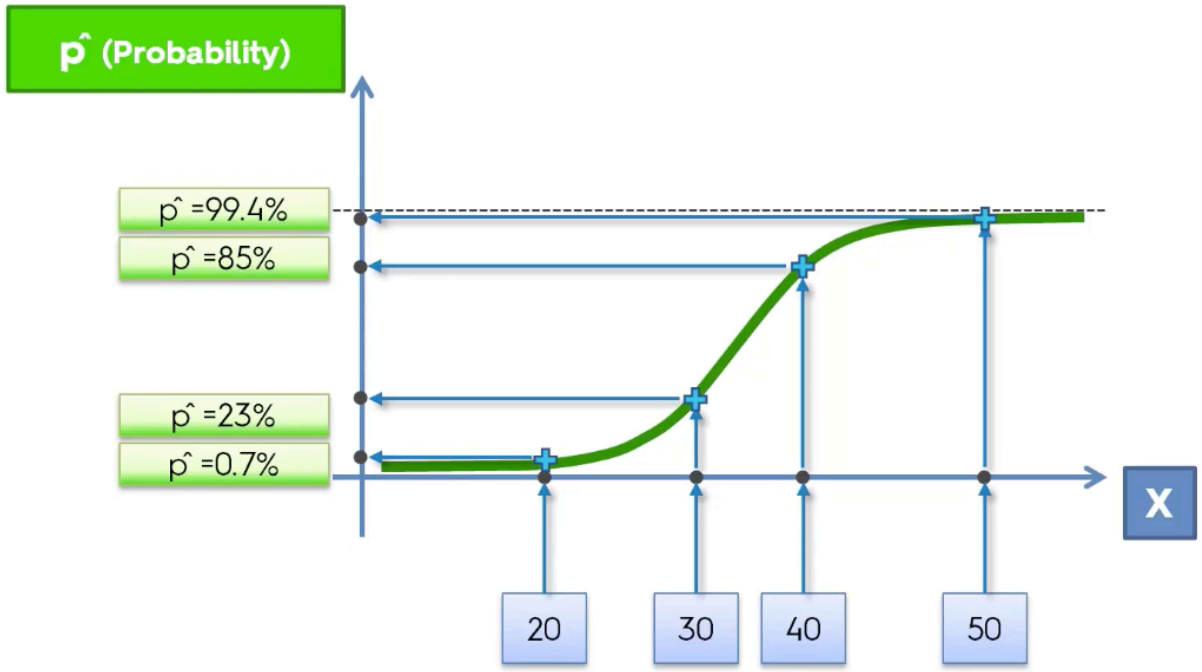
- No labels
- No feedback
- Find hidden structure in data

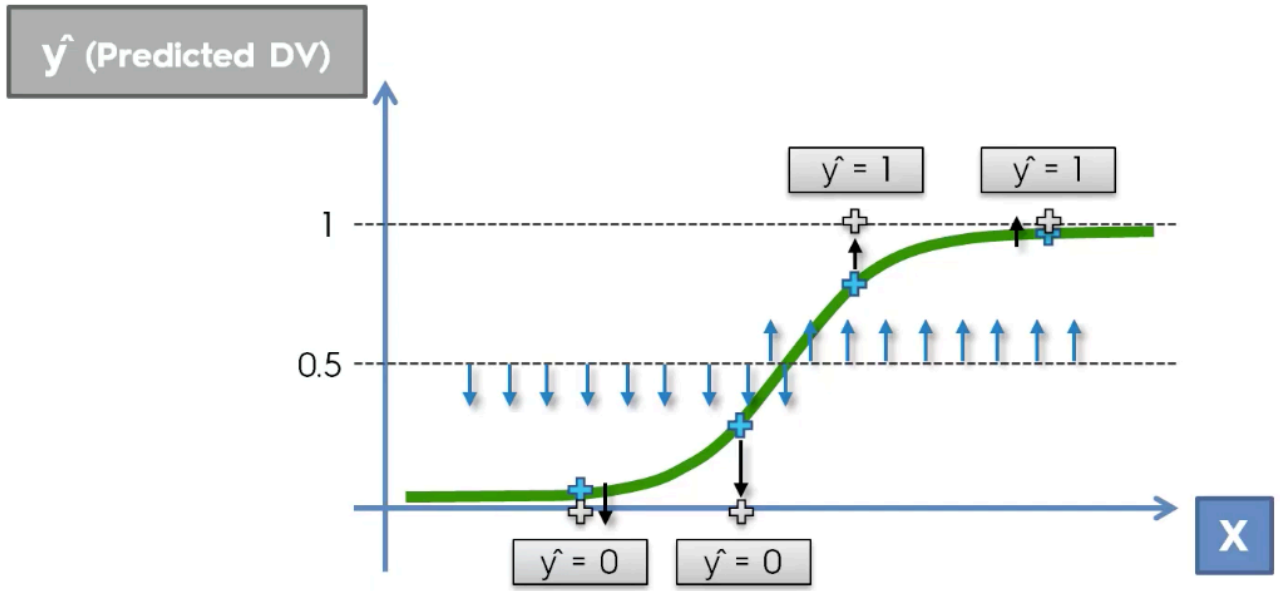


Reinforcement Learning

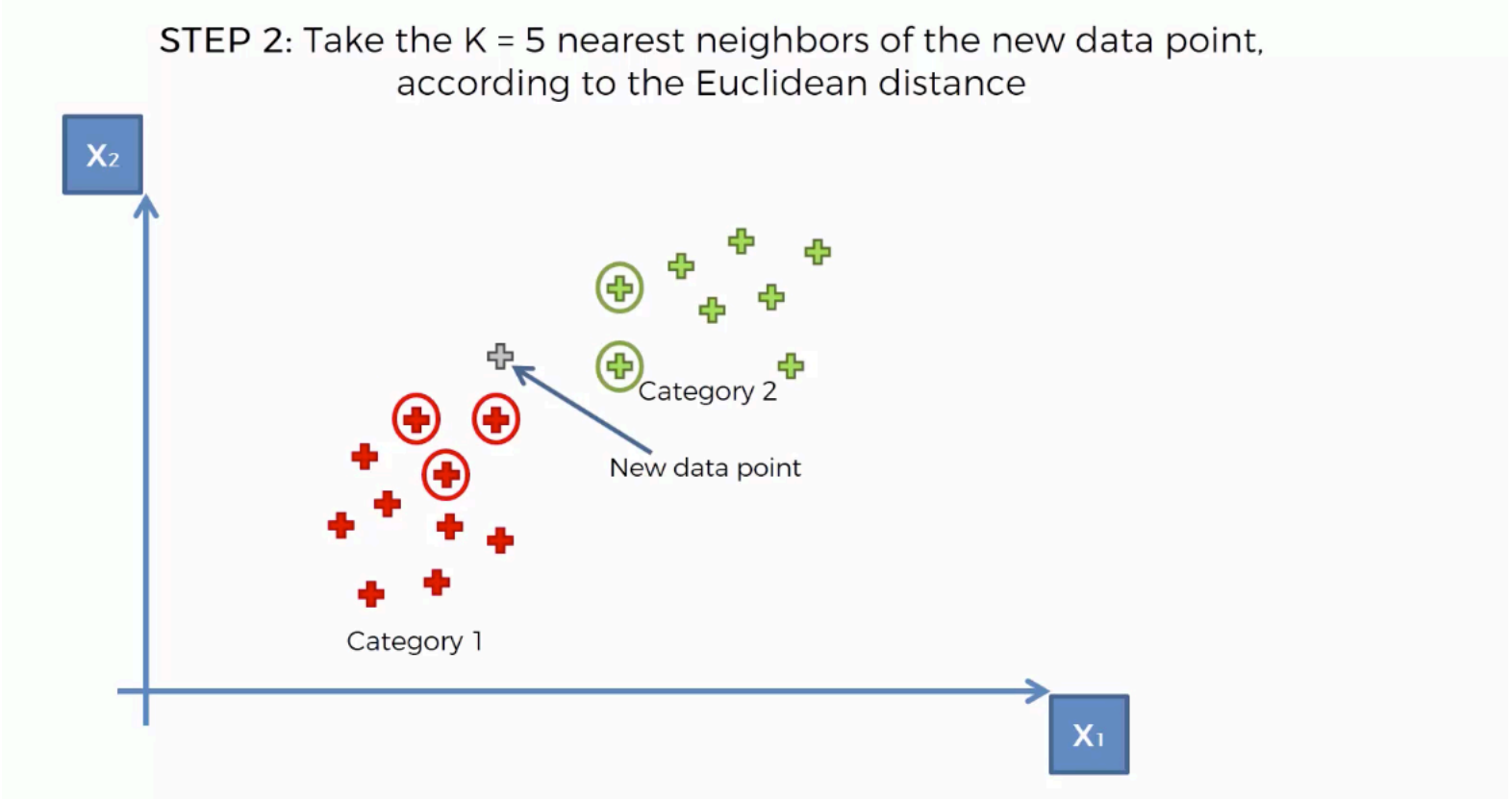
- Decision process
- Reward system
- Learn series of actions

1. Logistic Regression Classification

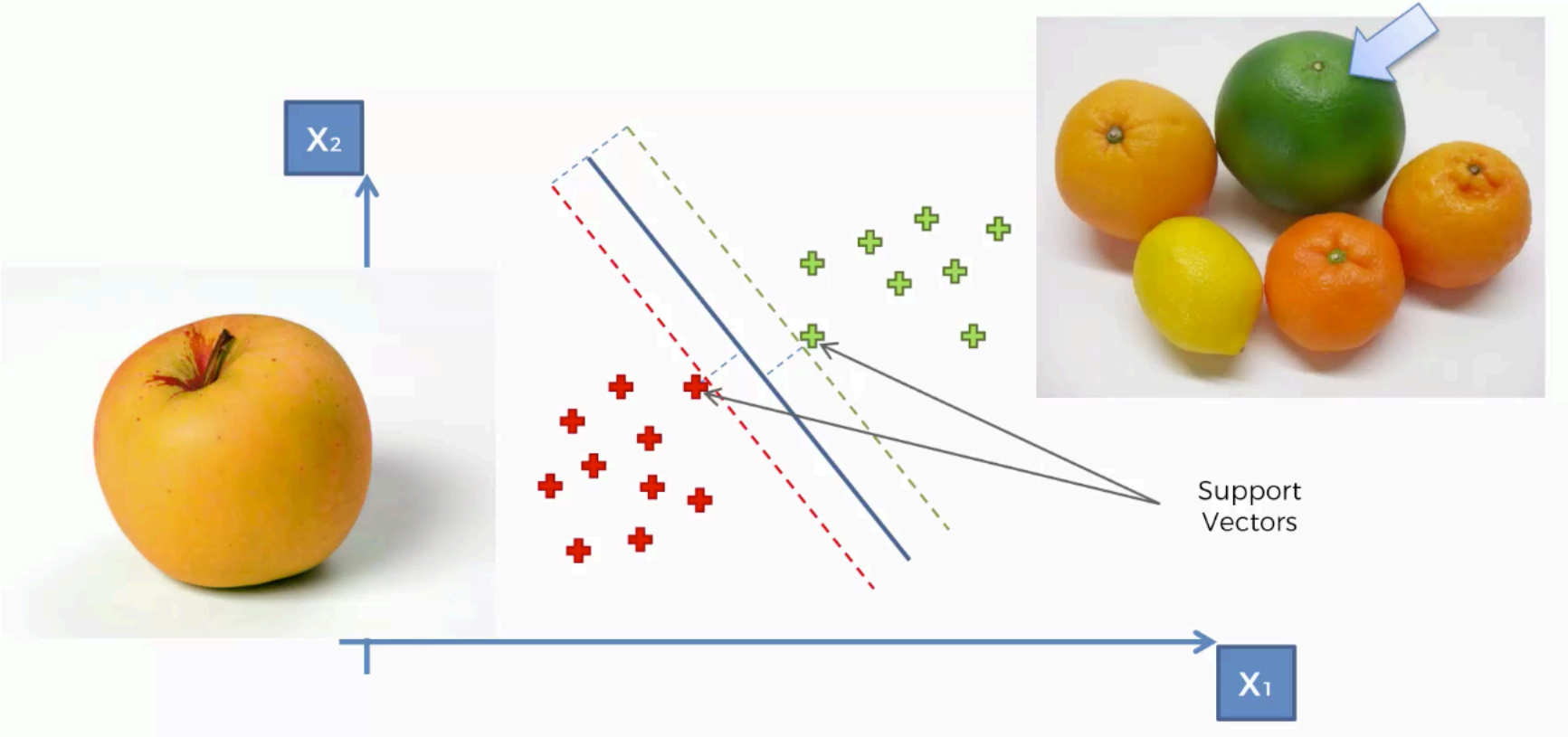




2. K-Nearest Neighbors (K-NN)



3. Support Vector Machine (SVM)



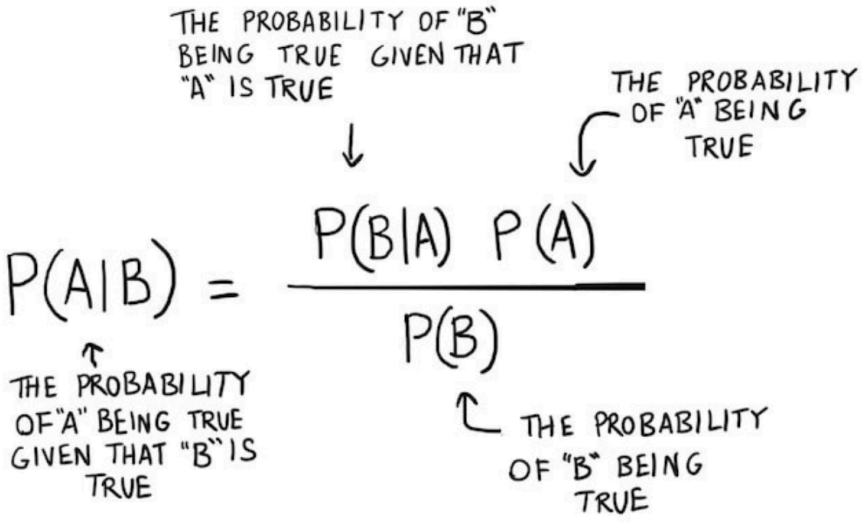
4. Naïve Bayes

A Brief look on Bayes Theorem :

Bayes Theorem helps us to find the probability of a hypothesis given our prior knowledge.

As per wikipedia,In probability theory and statistics, **Bayes' theorem** (alternatively **Bayes' law** or **Bayes' rule**, also written as **Bayes's theorem**) describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

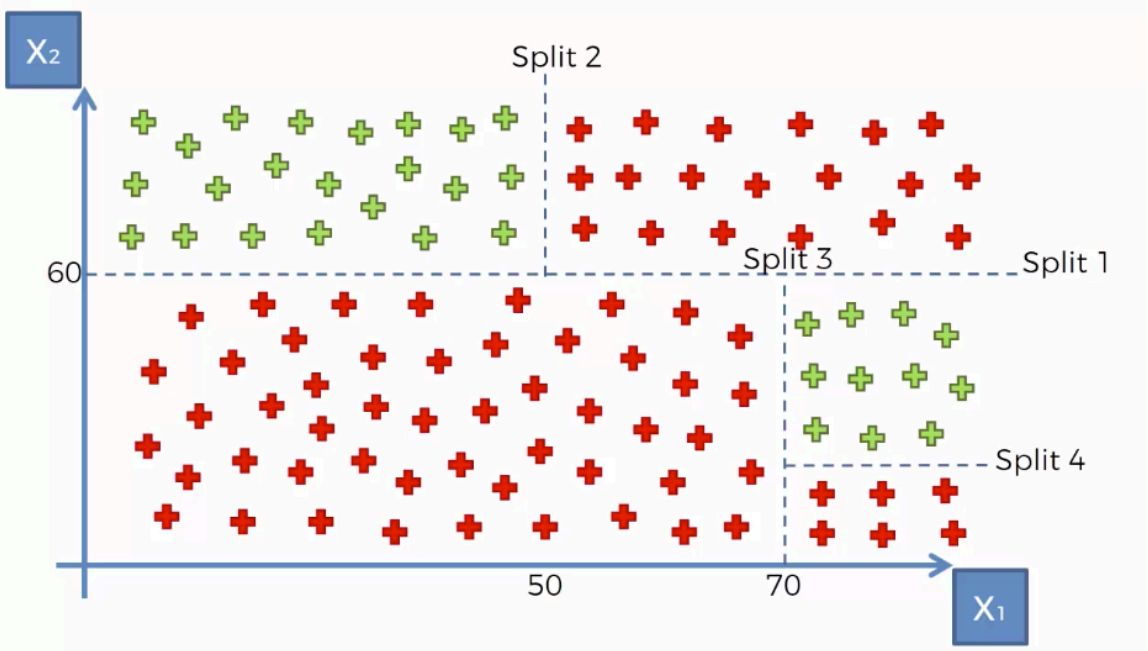
Lets look at the equation for Bayes Theorem,

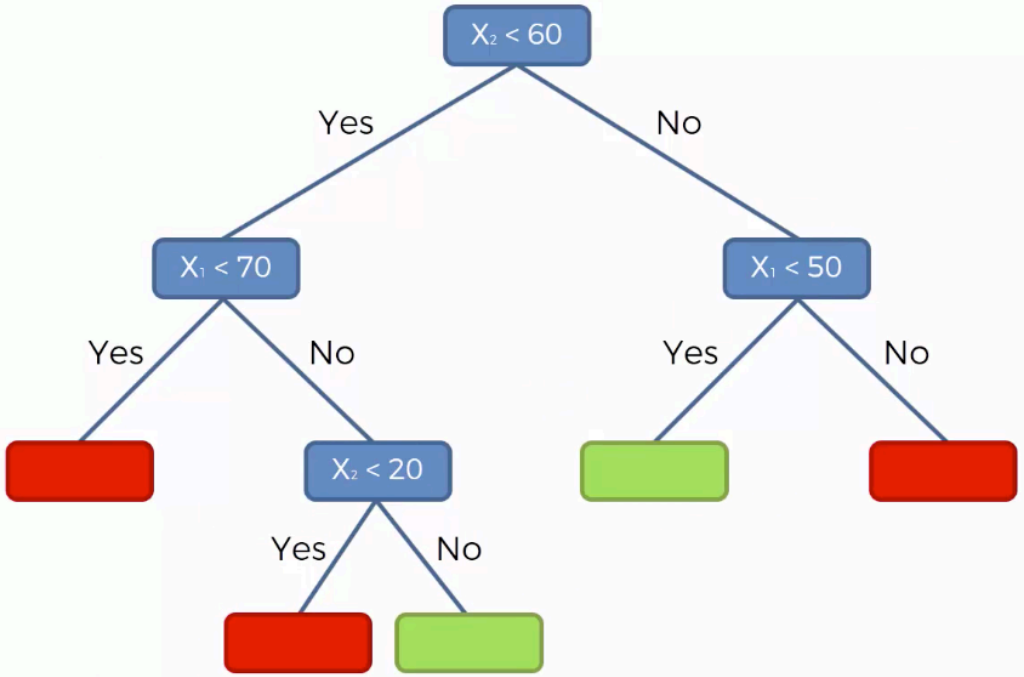


Where,

- **P(A|B)** is the probability of hypothesis A given the data B. This is called the **posterior probability.**
- **P(B|A)** is the probability of data B given that the hypothesis A was true.
- **P(A)** is the probability of hypothesis A being true (regardless of the data). This is called the **prior probability of A**.
- **P(B)** is the probability of the data (regardless of the hypothesis).

5. Decision Tree Classification





6. Random Forest Classification

Ensemble Learning

STEP 1: Pick at random K data points from the Training set.

STEP 2: Build the Decision Tree associated to these K data points.

STEP 3: Choose the number Ntree of trees you want to build and repeat STEPS 1 & 2

STEP 4: For a new data point, make each one of your Ntree trees predict the category to which the data points belongs, and assign the new data point to the category that wins the majority vote.

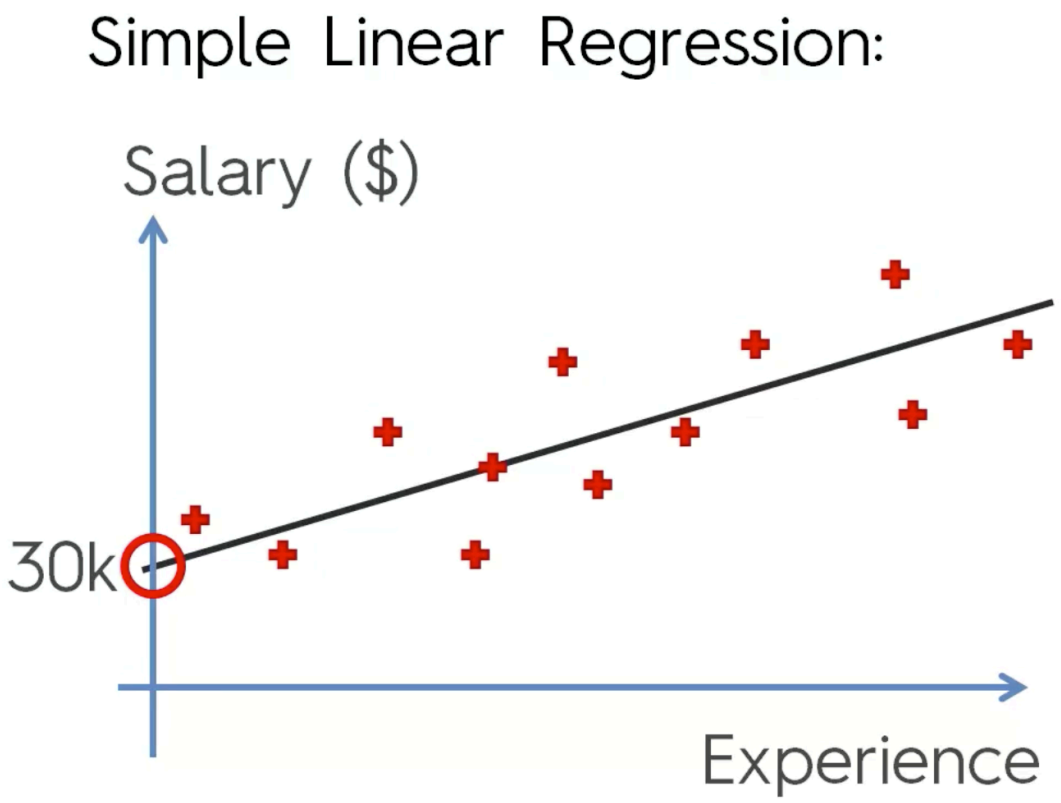
Classification Metrics

2. Confusion Matrix (Precision, Recall, F-Measure)

	Predicted class POSITIVE (spam ✉)	**Predicted class NEGATIVE (normal 📧)**	
Actual class POSITIVE (spam ✉)	TRUE POSITIVE (TP) ✉ ✉ **320**	FALSE NEGATIVE (FN) ✉ 📧 **43**	*Recall* $= \frac{TP}{TP + FN}$ $= \frac{320}{320 + 43} = 0.882$
Actual class NEGATIVE (normal ✉)	FALSE POSITIVE (FP) ✉ ✉ **20**	TRUE NEGATIVE (TN) ✉ 📧 **538**	
	Precision $= \frac{TP}{TP + FP}$ $= \frac{320}{320 + 20} = 0.941$		

Regression Algorithms

Simple Linear Regression



$$y = b_0 + b_1 * x$$

Salary = b_0 + b_1 *Experience

Regression Metrics

1. Mean Absolute Error
2. Mean Squared Error

Mean Absolute Error (MAE)

This is simply the average of the absolute difference between the target value and the value predicted by the model. Not preferred in cases where outliers are prominent.

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

Mean absolute error. Image by the author.

MAE does not penalize large errors.

Mean Squared Error (MSE)

The most common metric for regression tasks is MSE. It has a convex shape. It is the average of the squared difference between the predicted and actual value. Since it is differentiable and has a convex shape, it is easier to optimize.

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

Mean squared error. Image by the author.

MSE penalizes large errors.

3 different types of machine learning

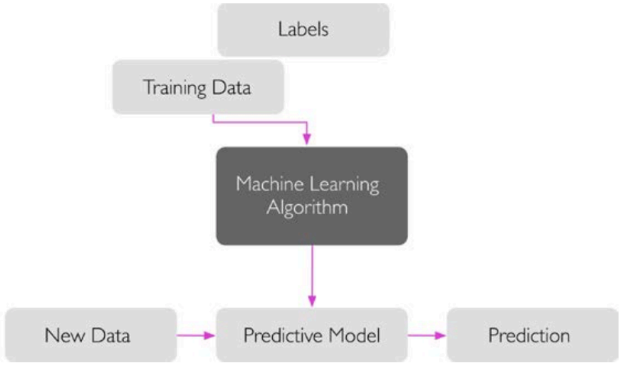
ARTIFICIAL INTELLIGENCE

Science thae enables computers to mimic human imtelligence.

Subfields: Machine Learning, robotics, and computer vision

MACHINE LEARNING

Subset of AI that enable maches to improve at tasks with experience

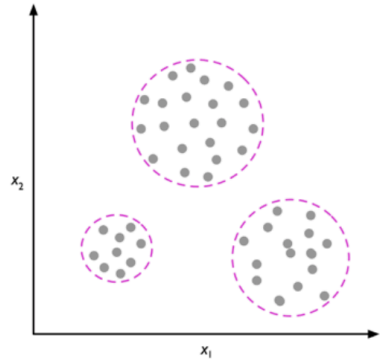


Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

Unsupervised Learning

- No labels
- No feedback
- Find hidden structure in data

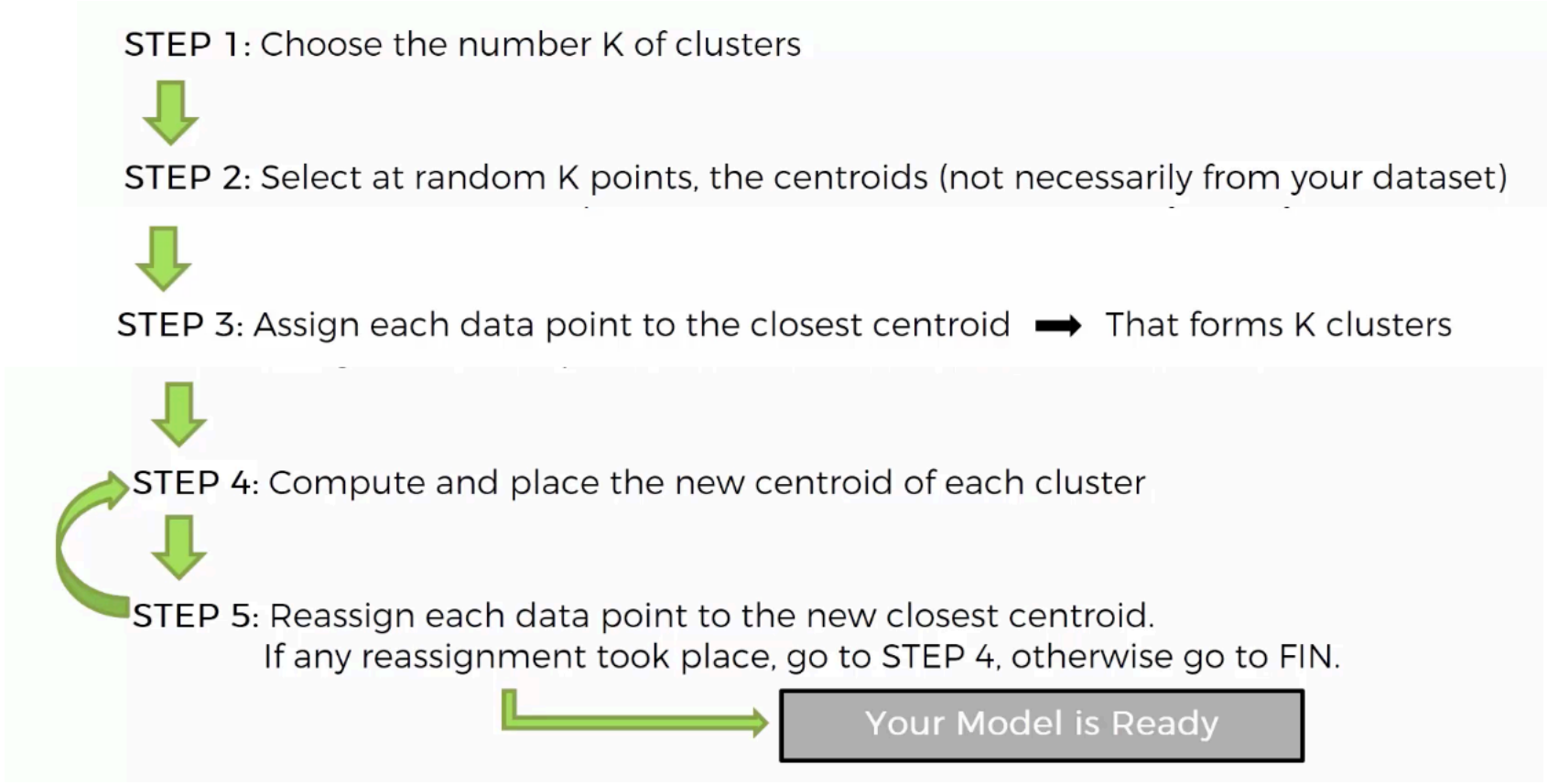


Reinforcement Learning

- Decision process
- Reward system
- Learn series of actions

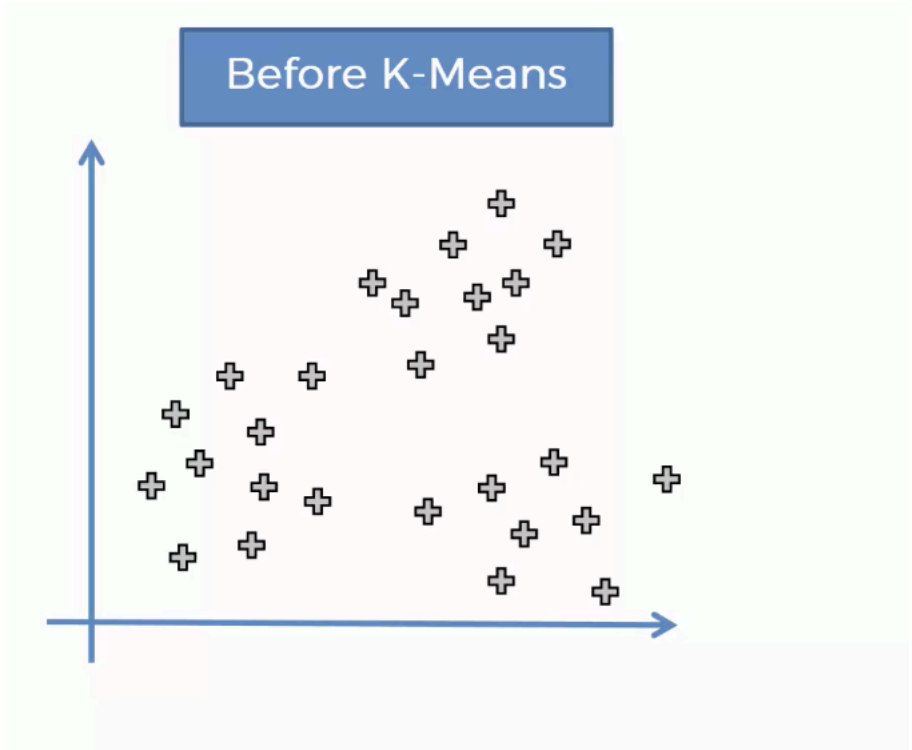
K-Means Clustering

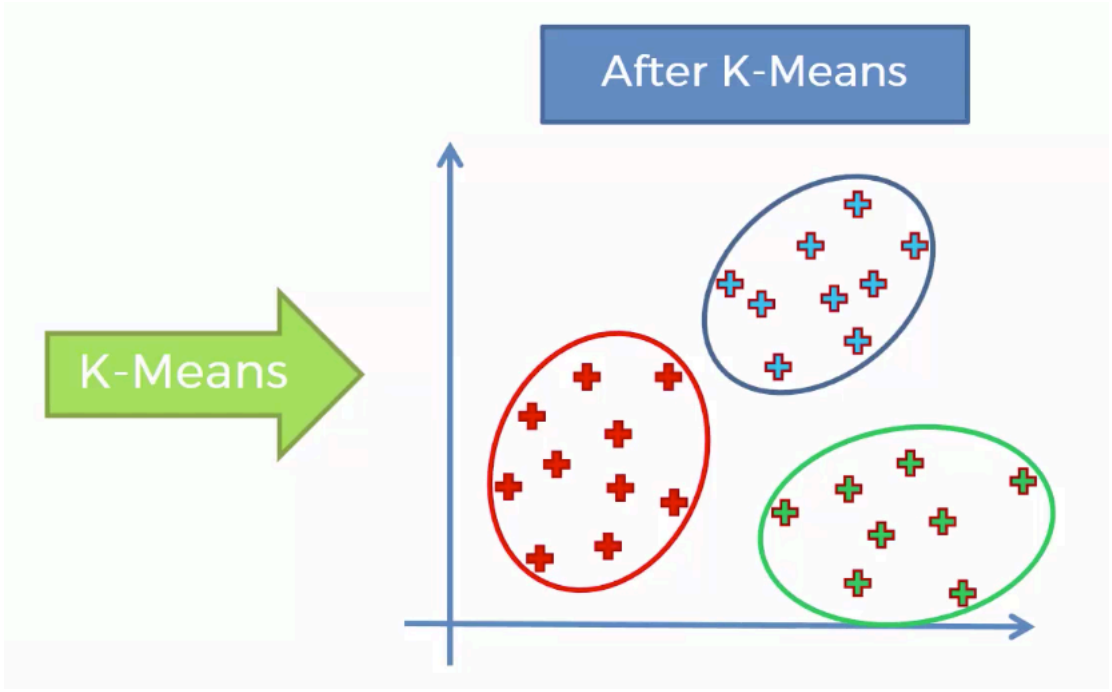
Intuition (How did it do that?)



K-Means Clustering

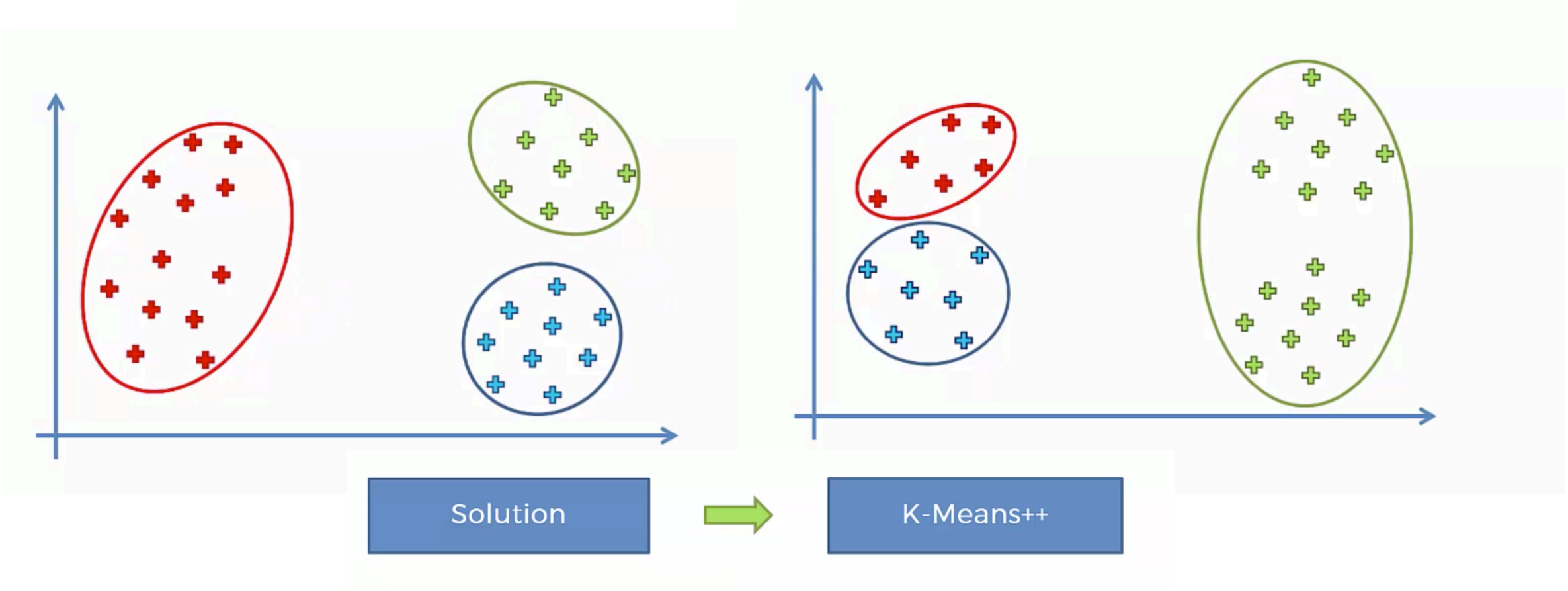
Intuition





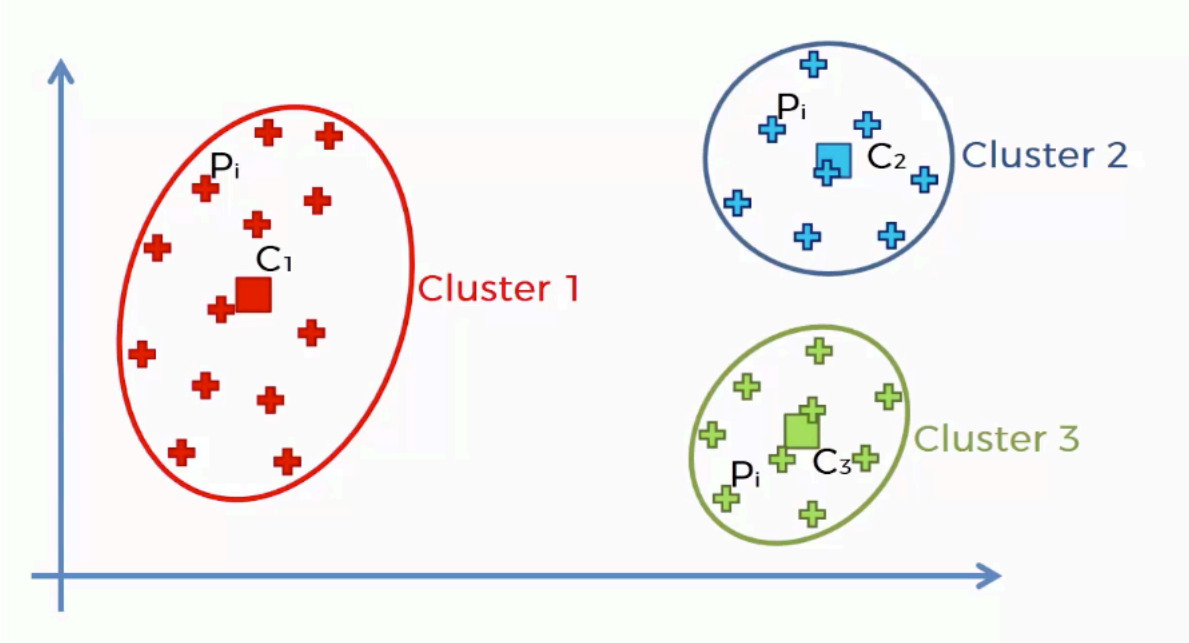
K-Means Clustering

2. Random Initialization Trap



K-Means Clustering

3. Selecting The Number Of Clusters

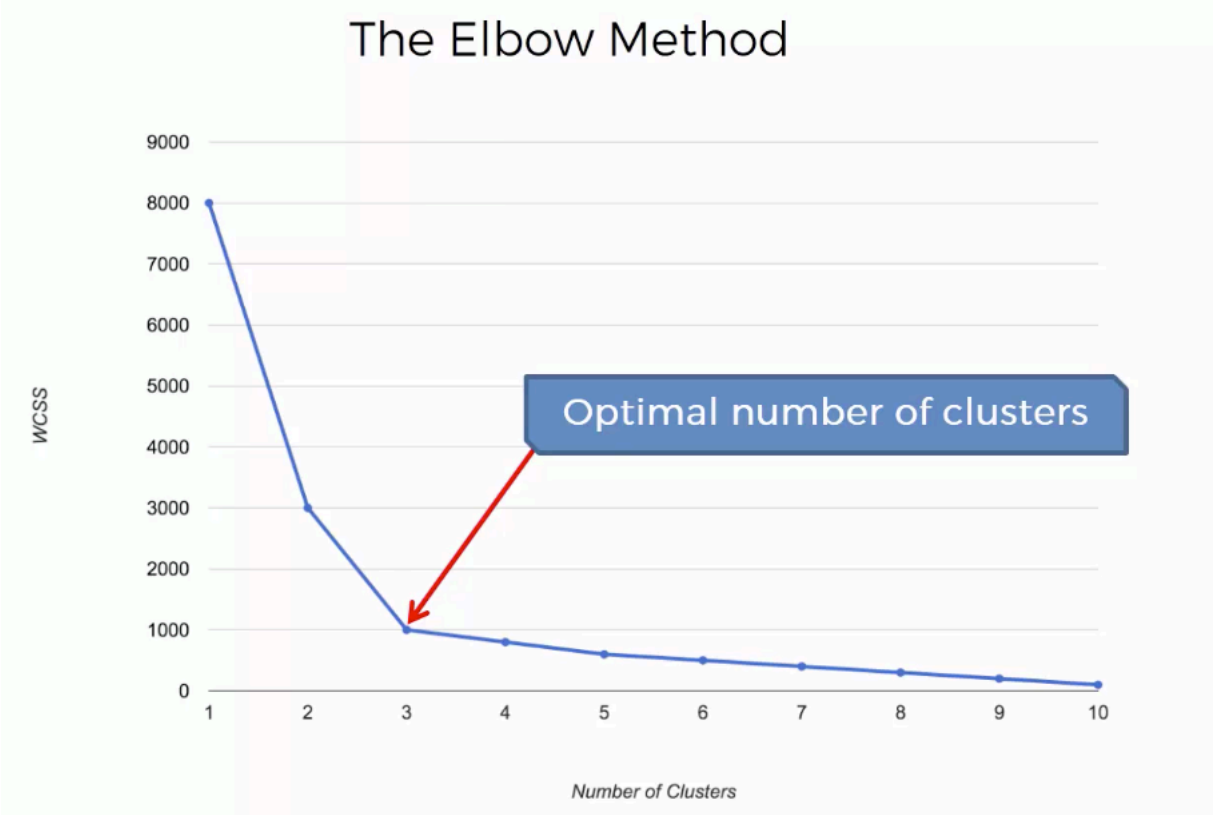


Within-Cluster-Sum-of-Squares (**WCSS**)

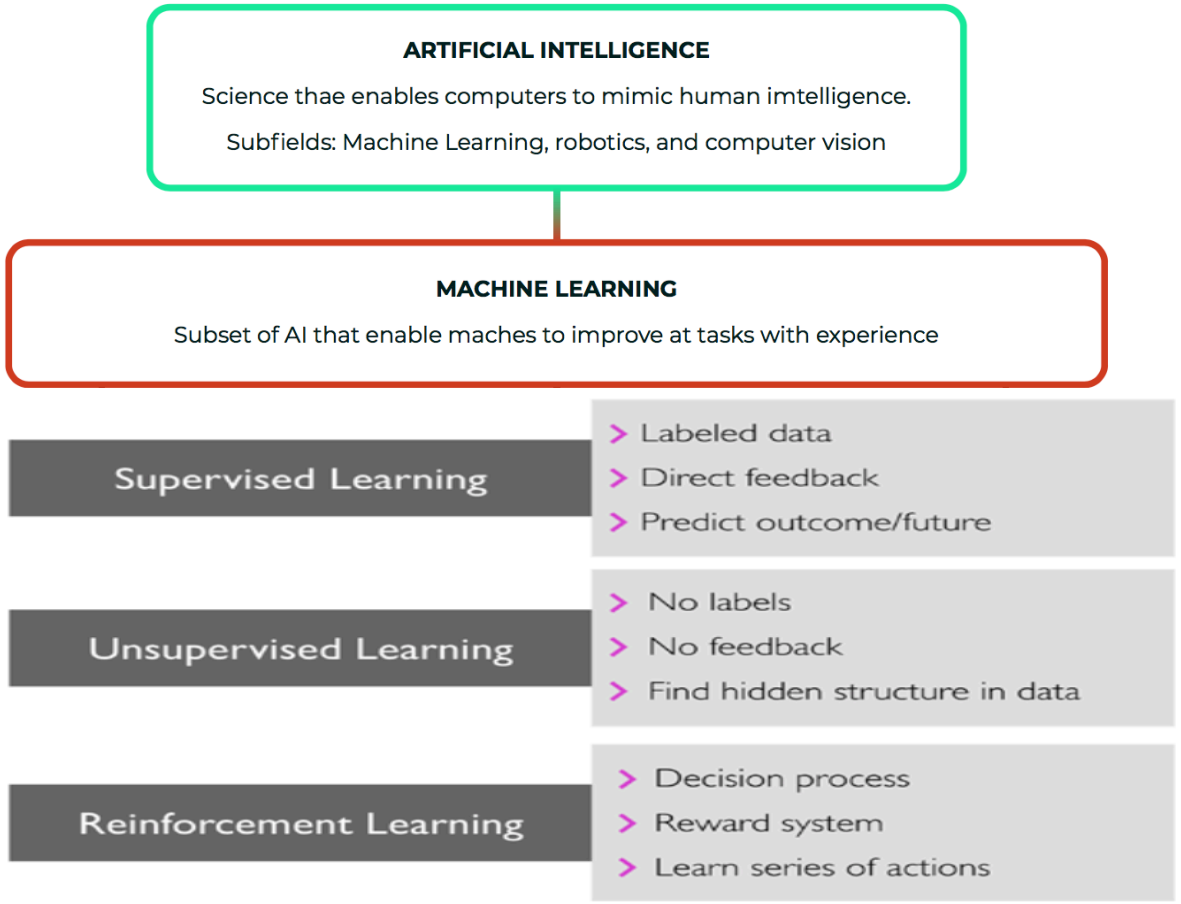
$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

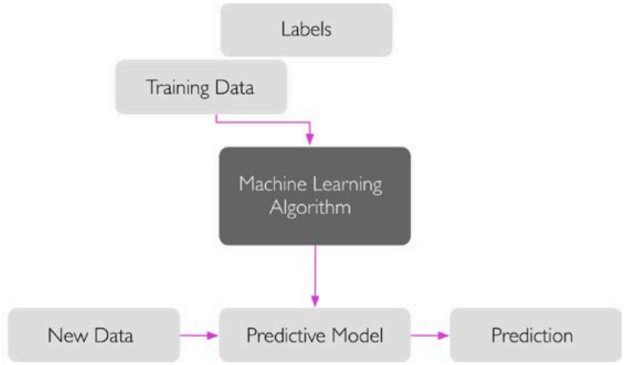
K-Means Clustering

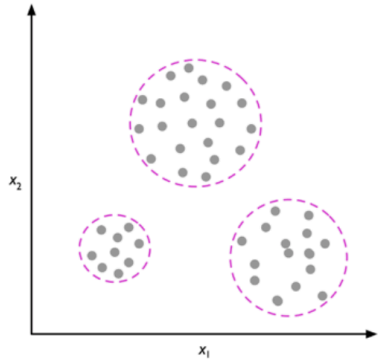
3. Selecting The Number Of Clusters



3 different types of machine learning

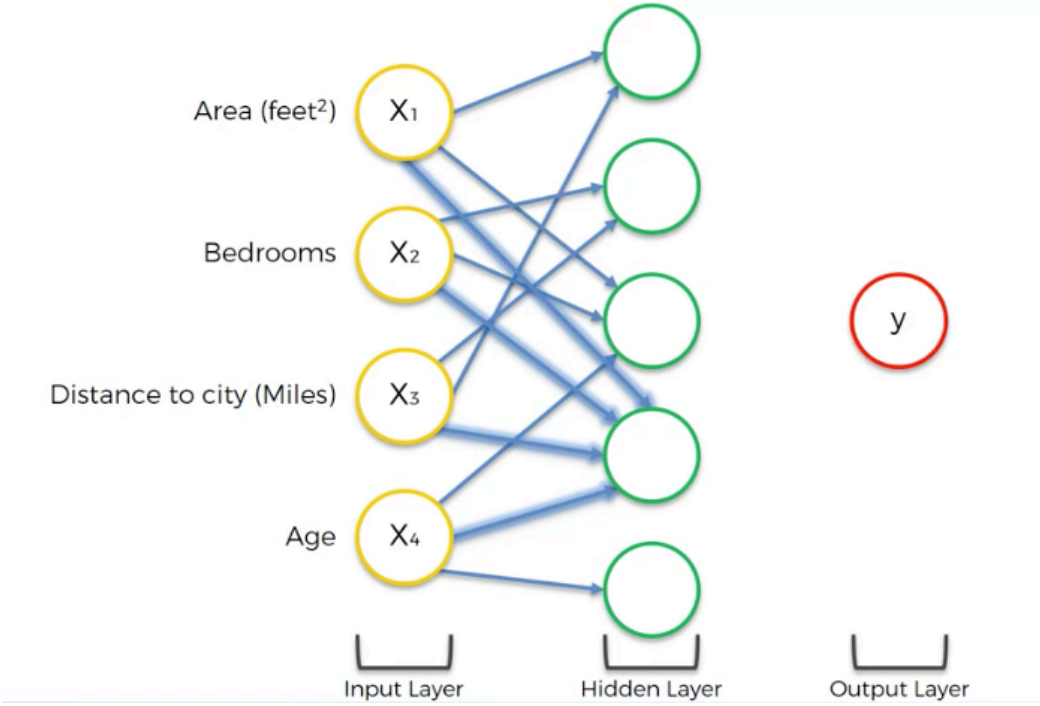




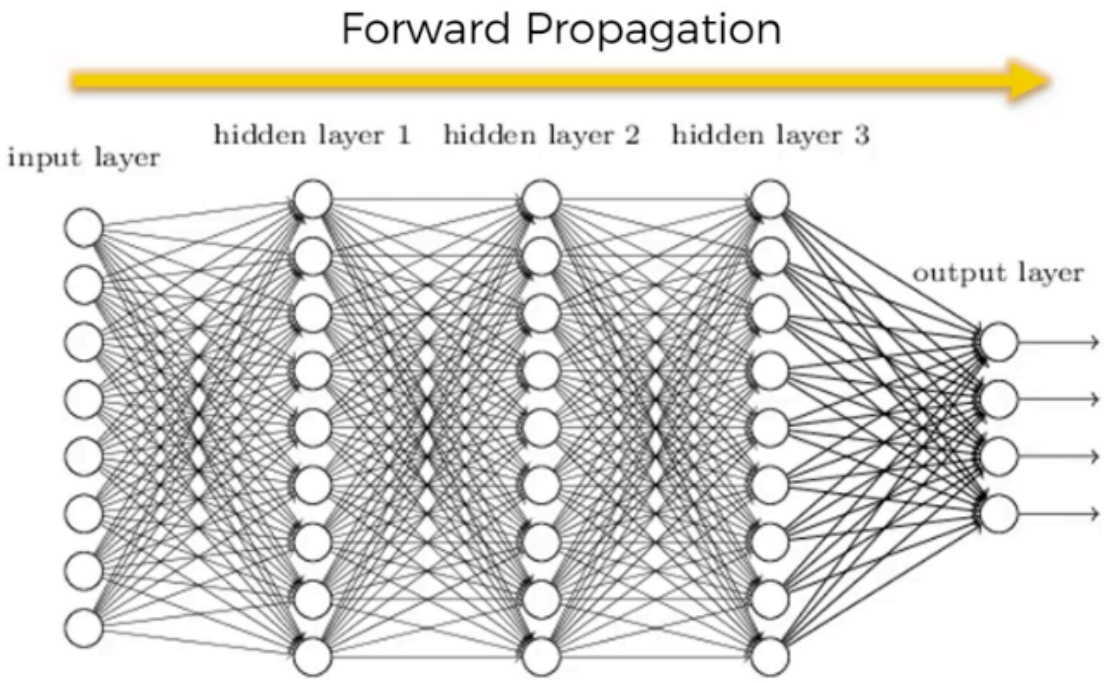


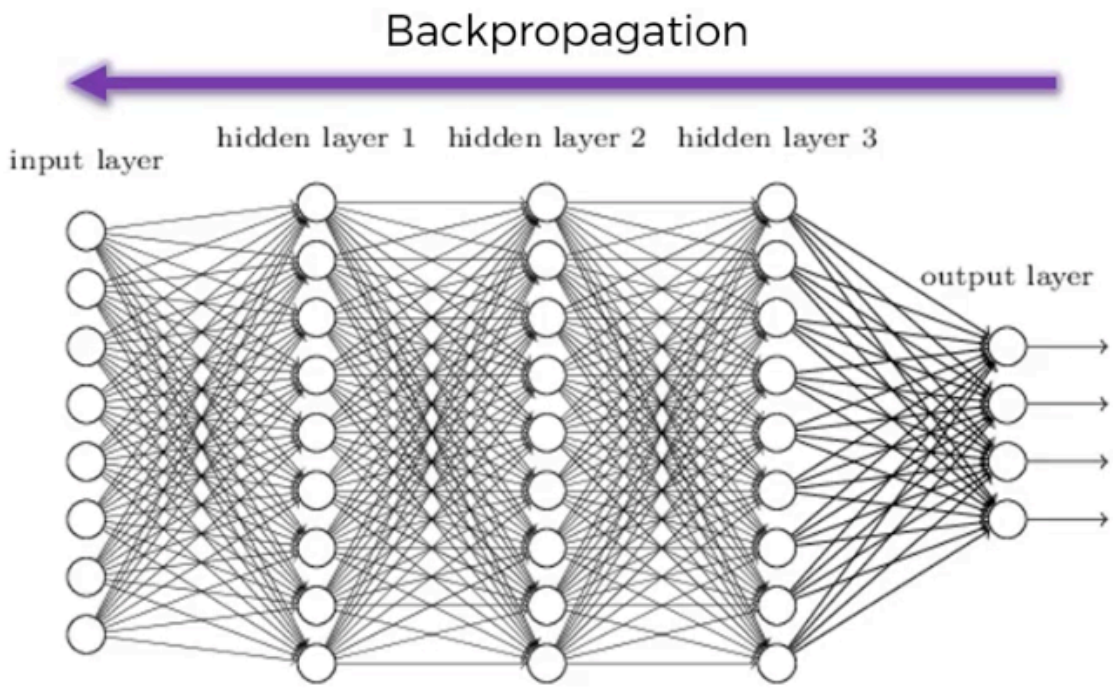
ANN, CNN, RNN

How do Neural Networks work?



Back propagation

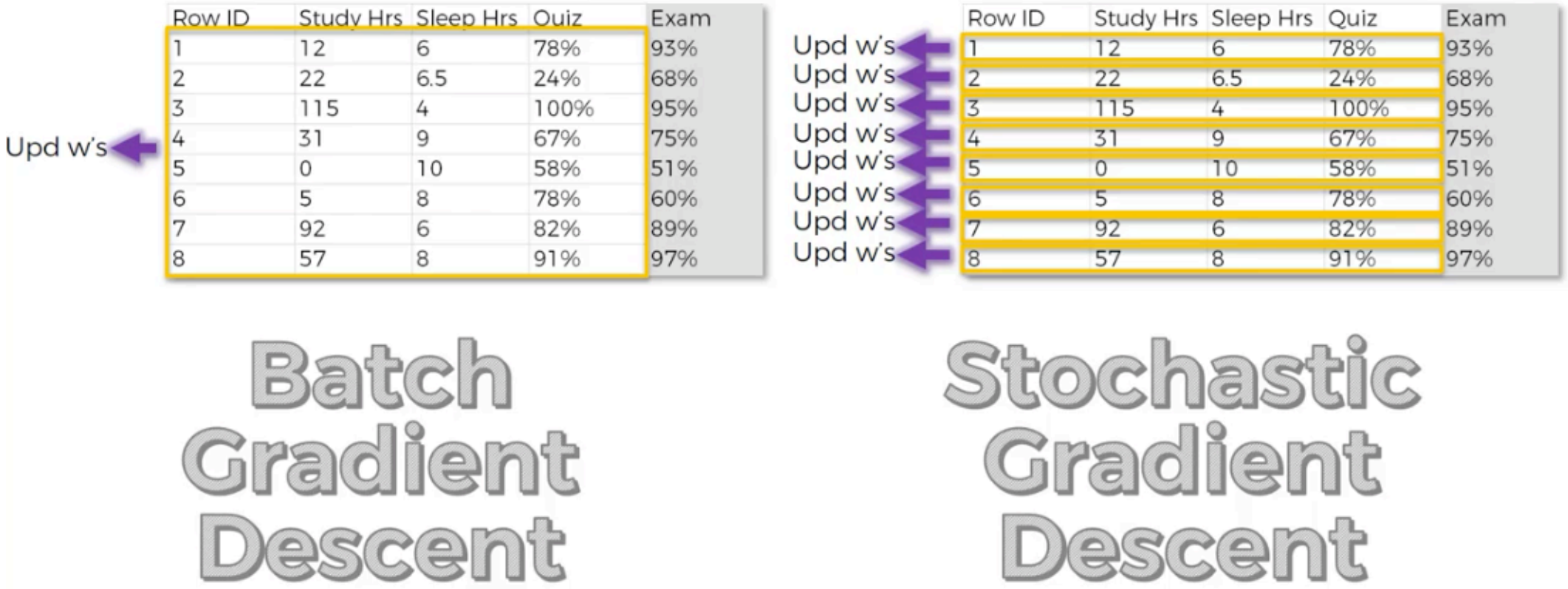




https://keras.io/api/models/model_training_apis/

Stochastic Gradient Descent

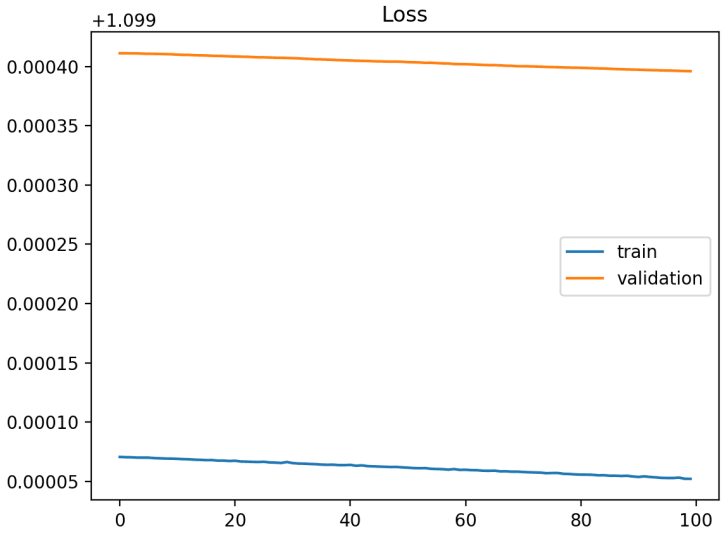
- **Batch GD** is optimistic method but may have local minimum problem.
- **SGD** = we can random pick the row -> High fluctuation -> More likely to find global minimum -> Faster cz they don't need to load the whole data into memory
- **Mini-batch GD** = combine both methods

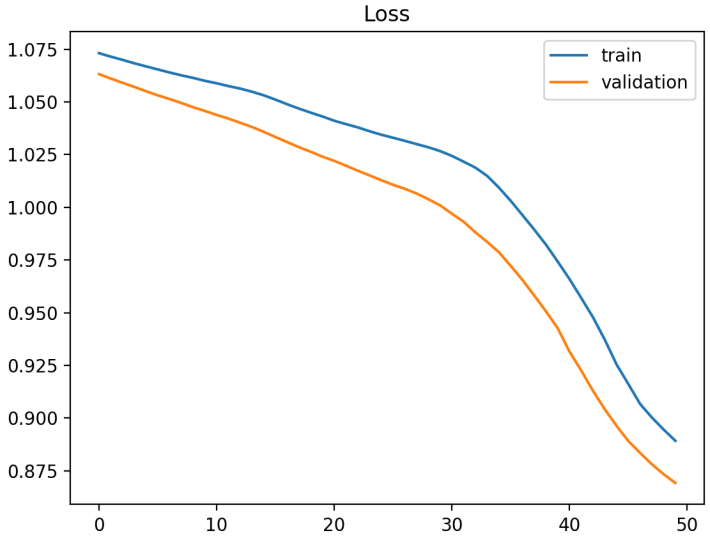


How to use Learning Curves to Diagnose DL Model Performance

Underfit Learning Curves

- A plot of learning curves shows underfitting if:
  1. The training loss remains flat regardless of training.
  2. The training loss continues to decrease until the end of training.





How to use Learning Curves to Diagnose DL Model Performance

Overfit Learning Curves

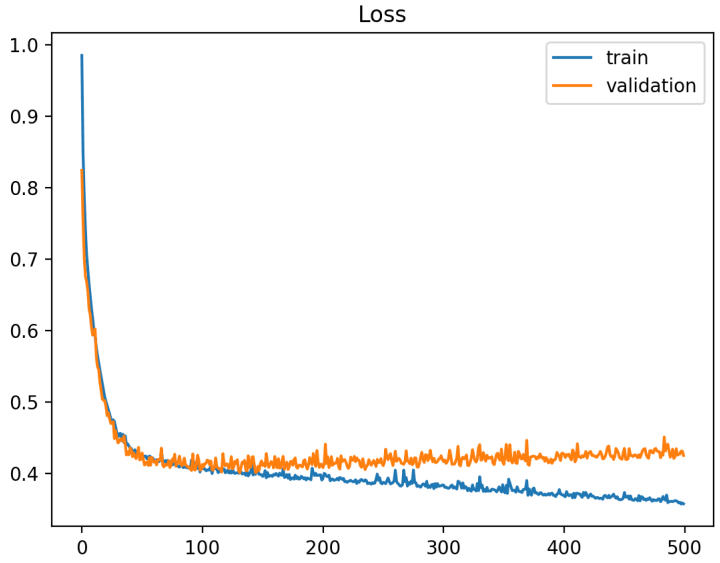
Overfitting refers to a model that has learned the training dataset too well, including the statistical noise or random fluctuations in the training dataset.

> This often occurs if the model has more capacity than is required for the problem, and, in turn, too much flexibility. It can also occur if the model is trained for too long.

A plot of learning curves shows overfitting if:

- The plot of training loss continues to decrease with experience.
- The plot of validation loss decreases to a point and begins increasing again.

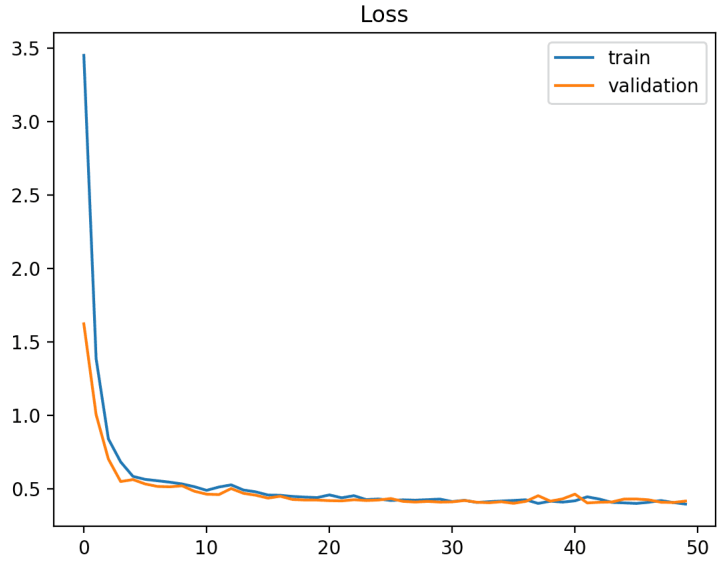
The inflection point in validation loss may be the point at which training could be halted as experience after that point shows the dynamics of overfitting.



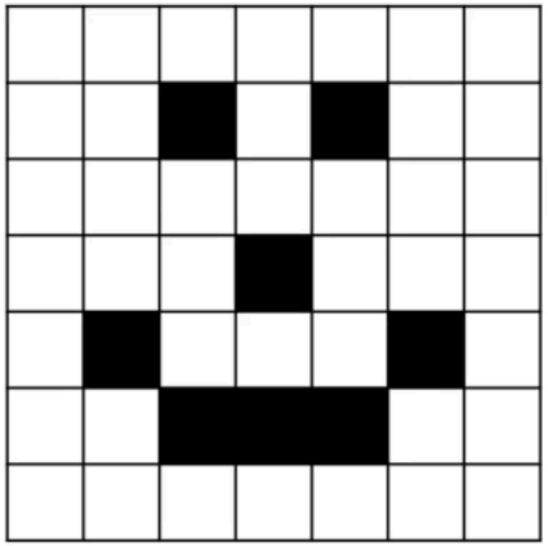
How to use Learning Curves to Diagnose DL Model Performance

Good Fit Learning Curves

- A plot of learning curves shows a good fit if:
  - The plot of training loss decreases to a point of stability.
  - The plot of validation loss decreases to a point of stability and has a small gap with the training loss.
- **Continued training of a good fit will likely lead to an overfit.**



Convolutional Neural Networks



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

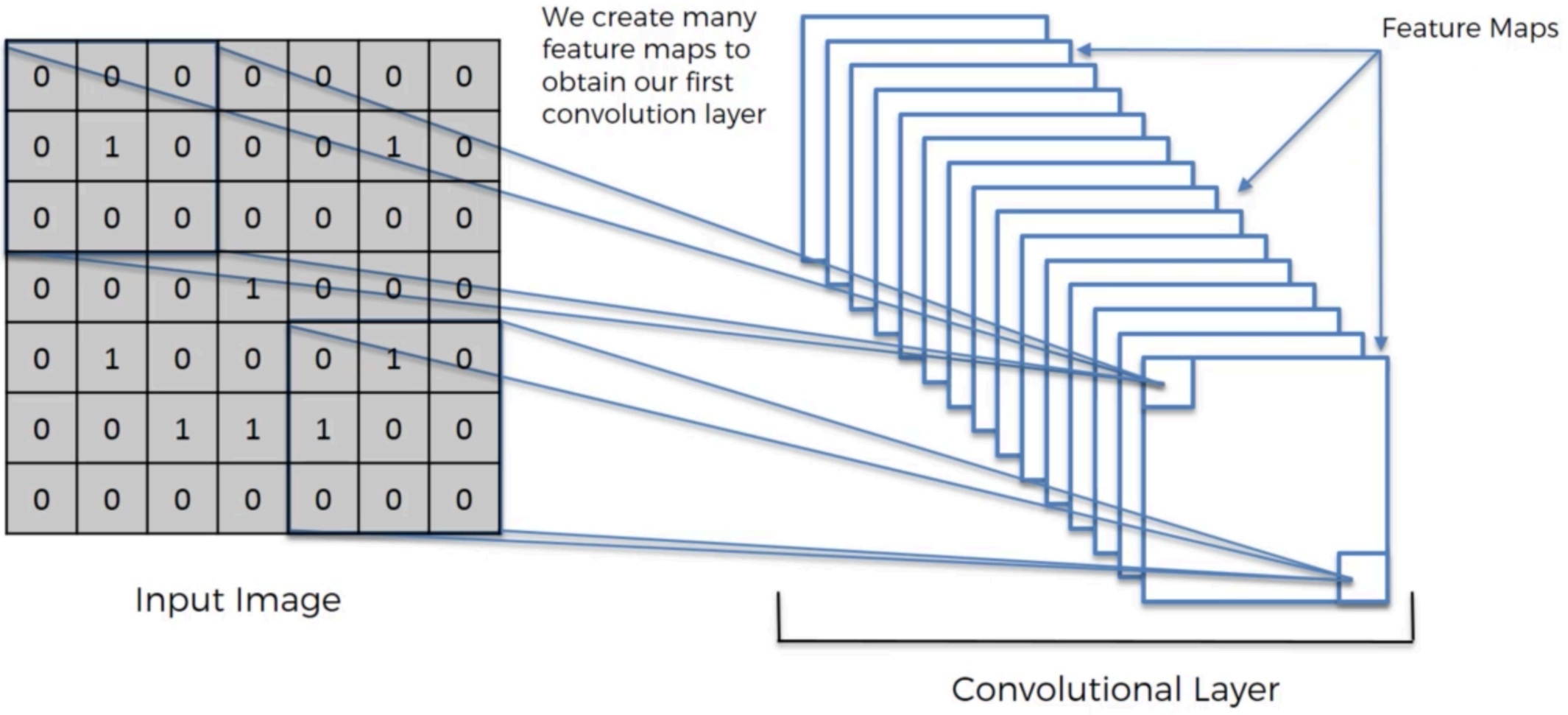
STEP 1: Convolution

STEP 2: Max Pooling

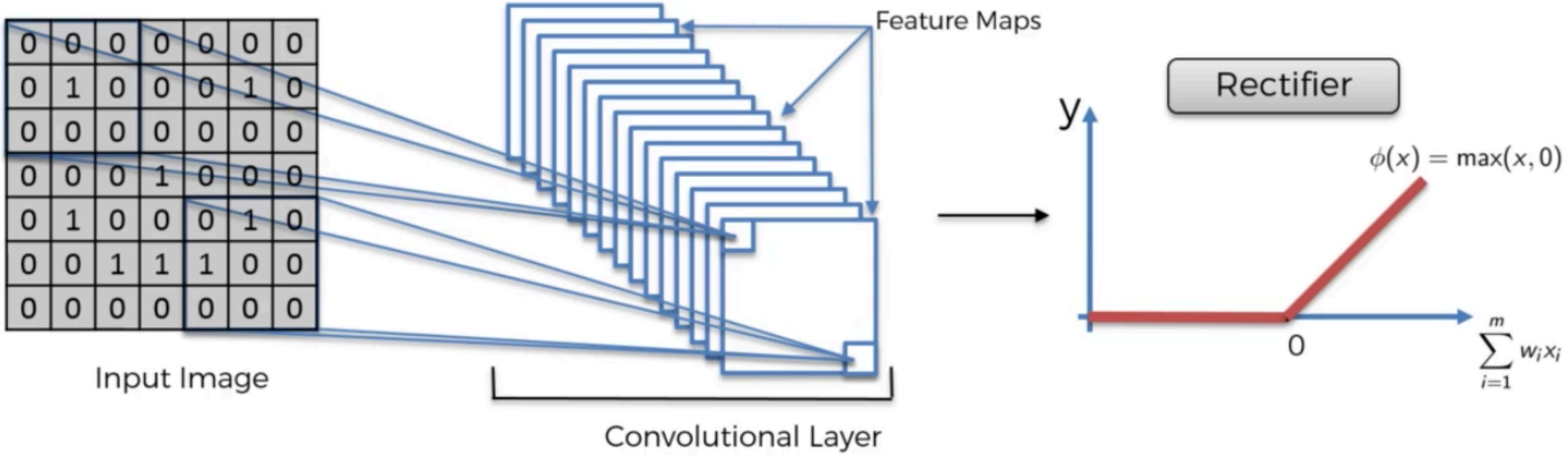
STEP 3: Flattening

STEP 4: Full Connection

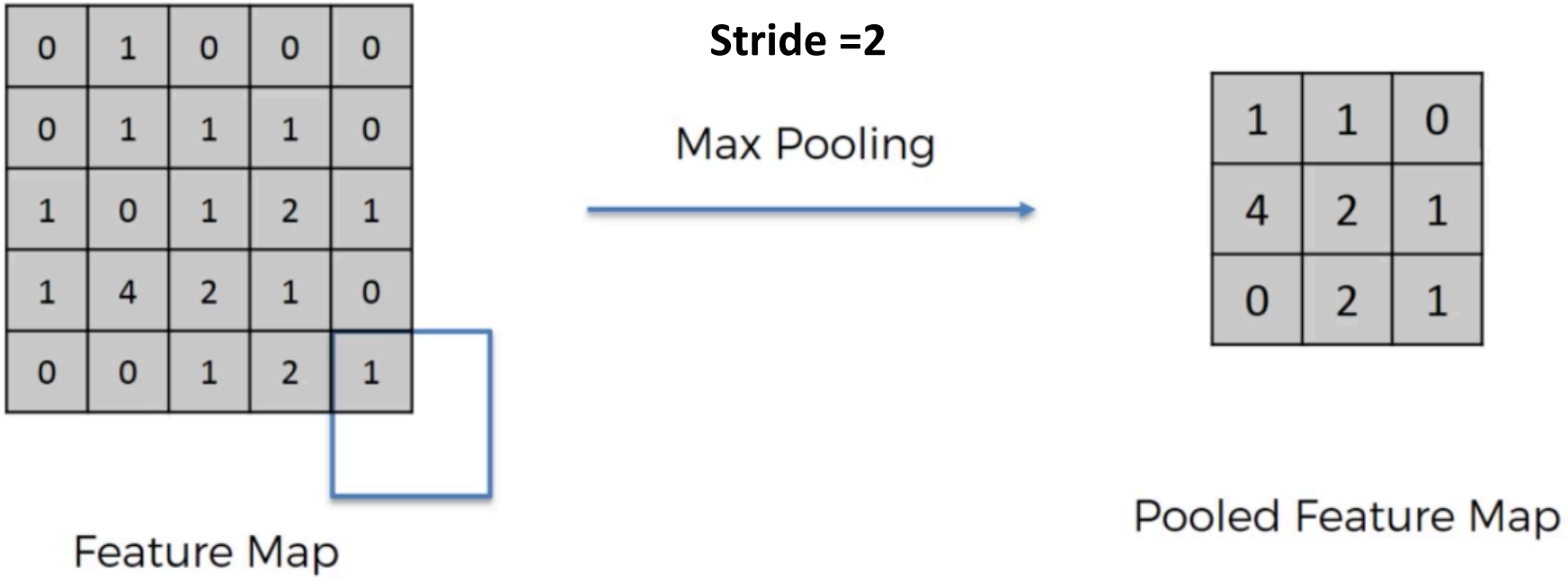
Step 1 - Convolutional Operation



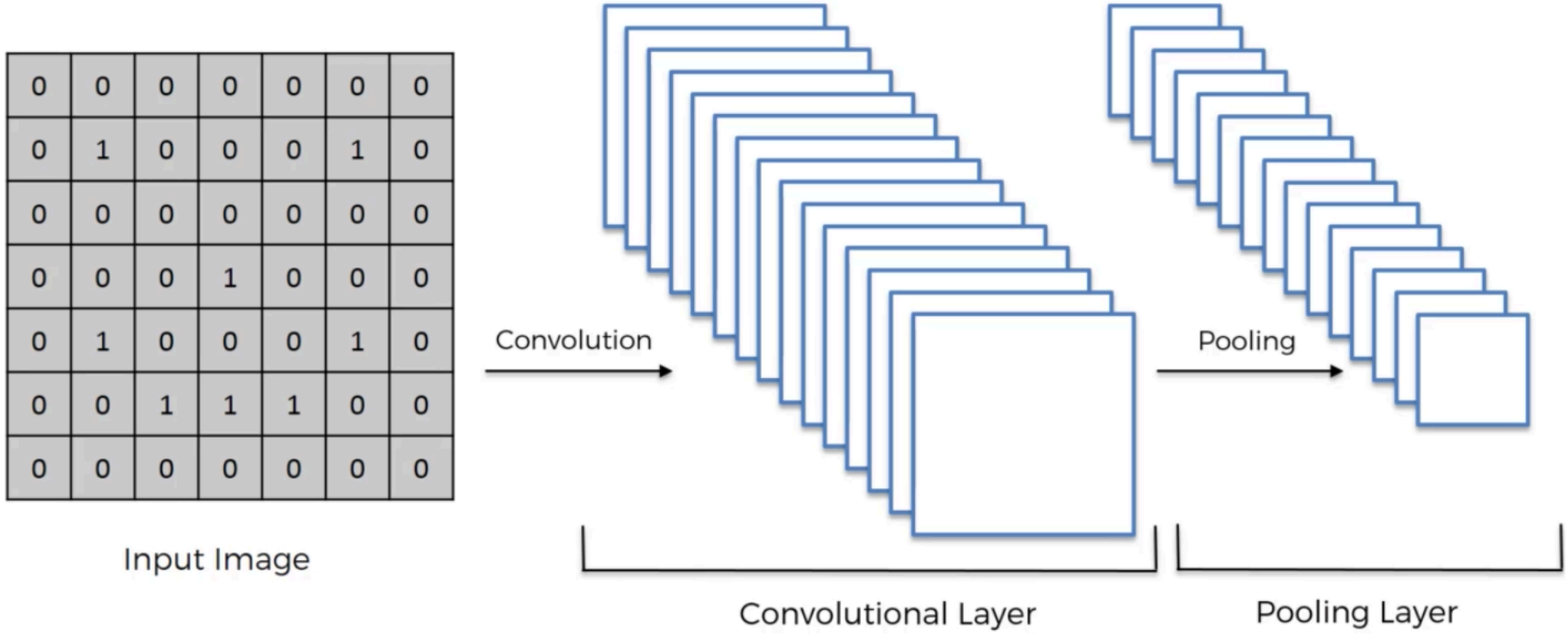
Step 1(b) – ReLu Layer



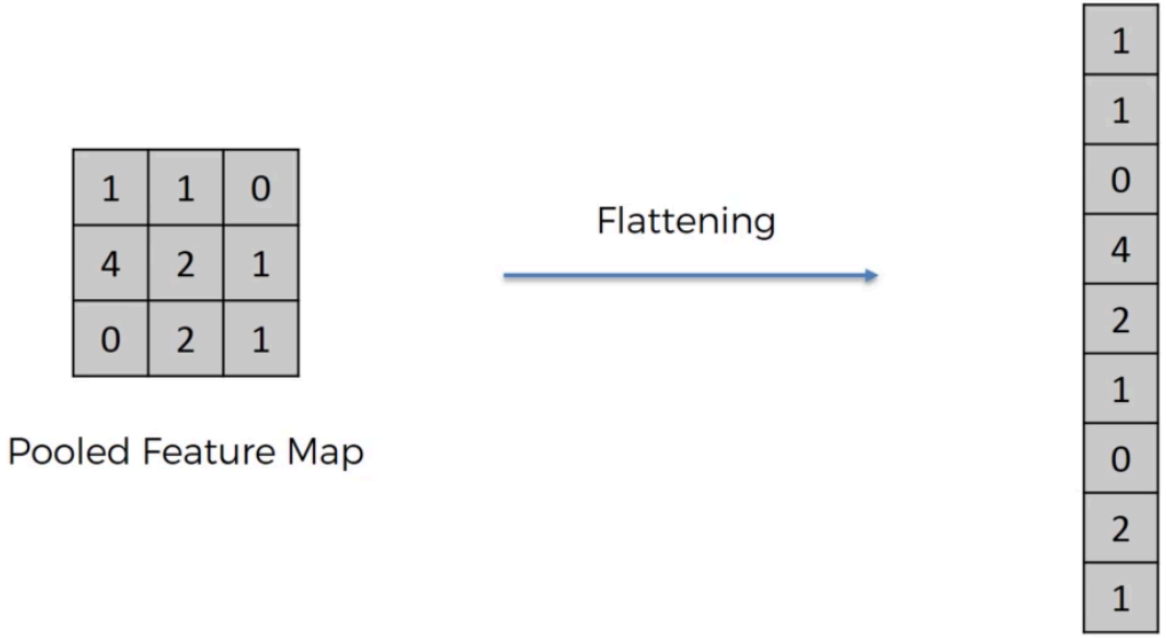
Step 2 – Max Pooling



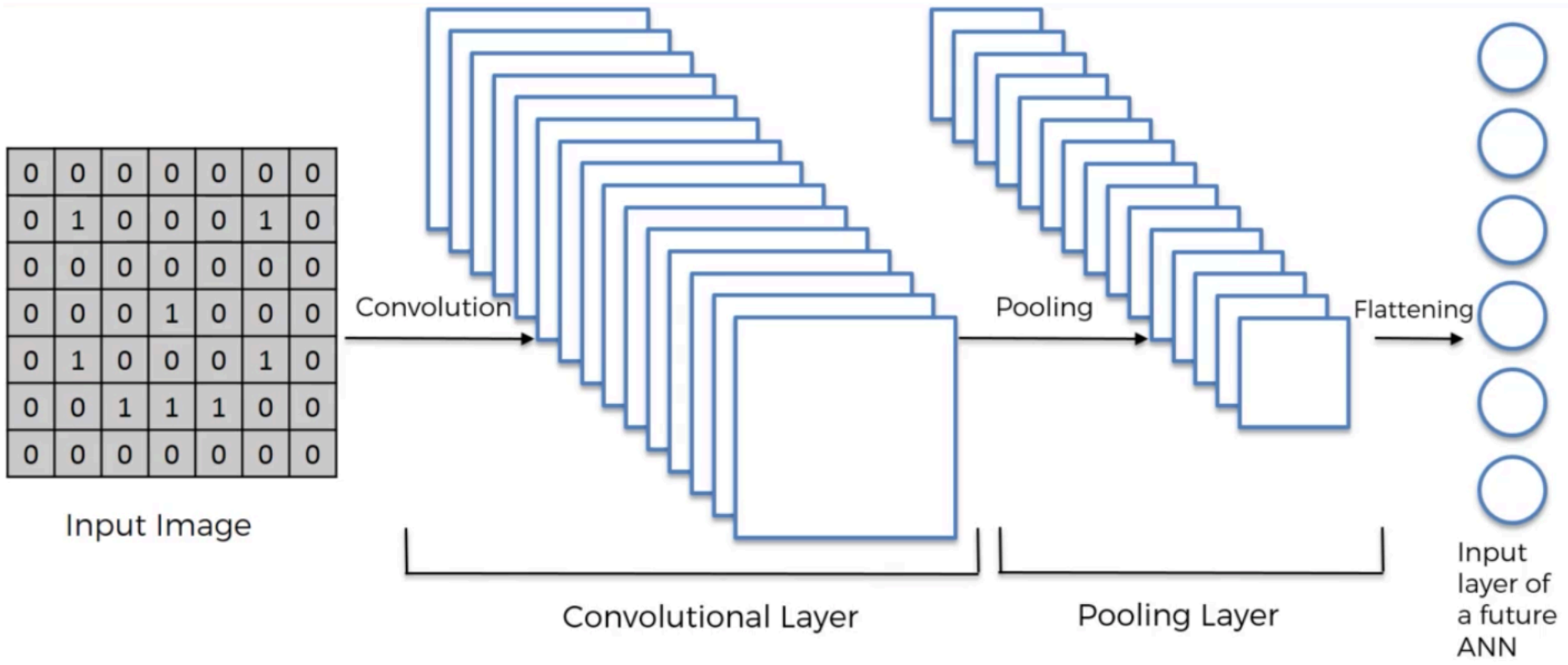
Step 2 – Max Pooling



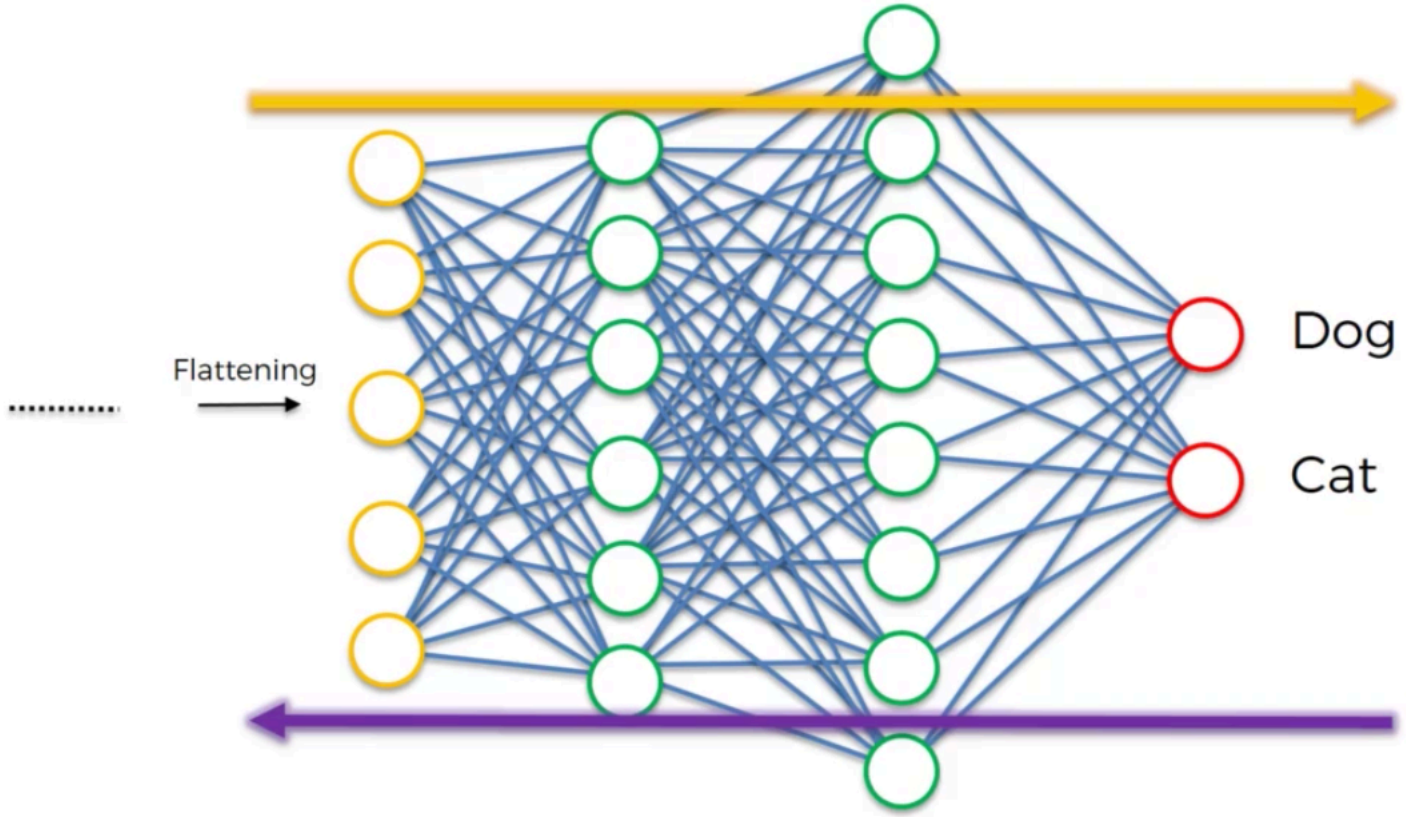
Step 3 - Flattening



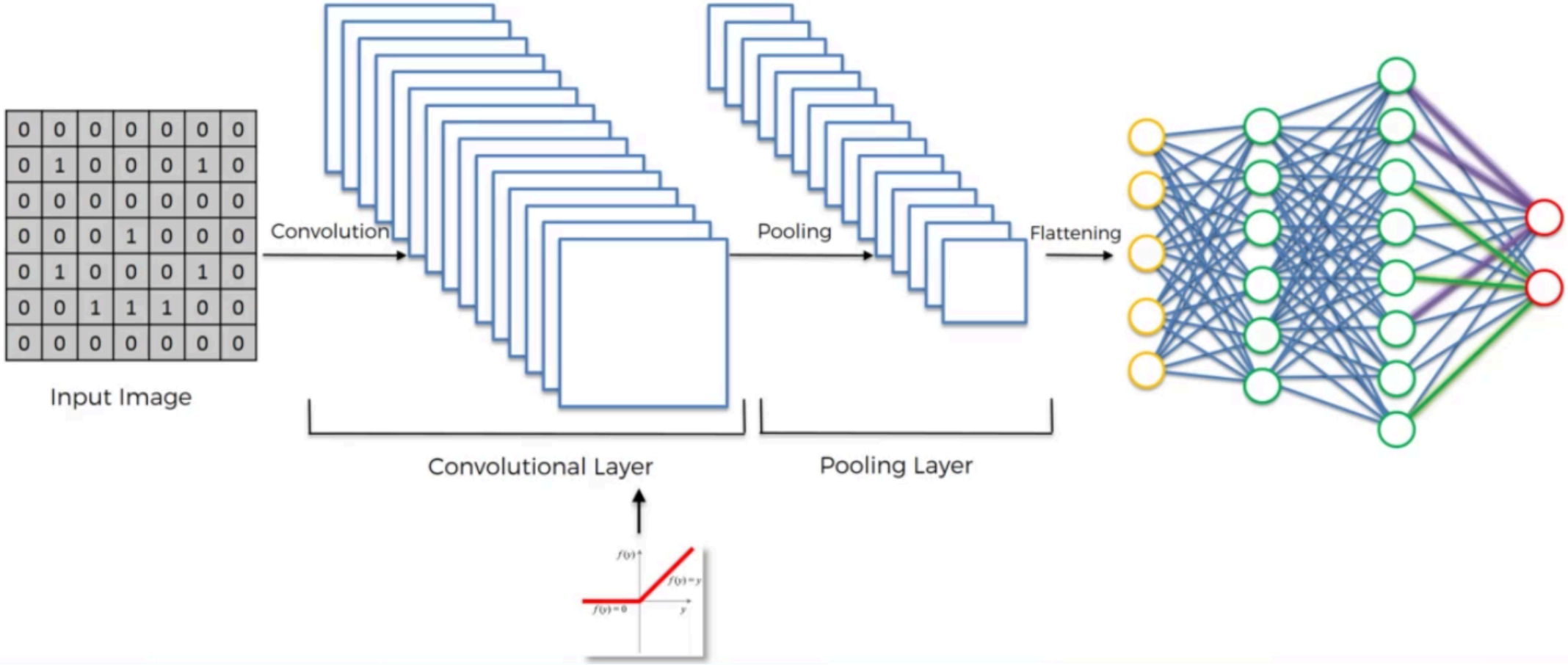
Step 3 - Flattening



Step 4 – Full Connection

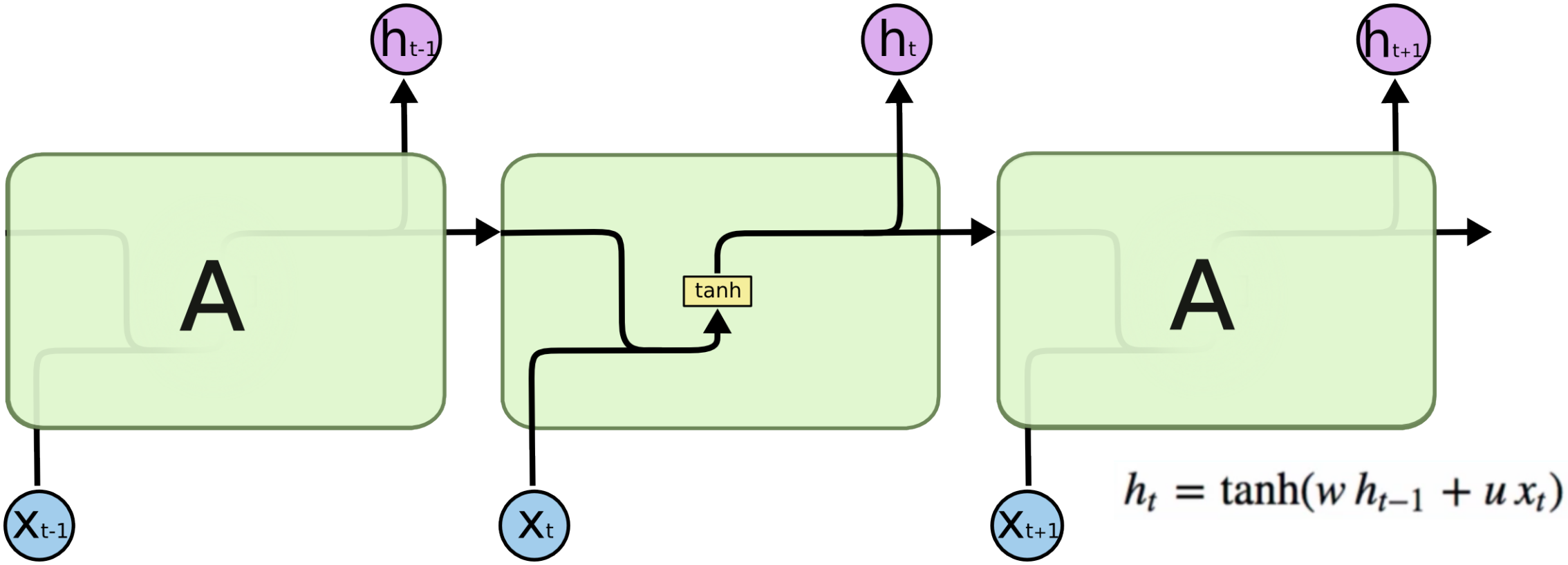


Summary

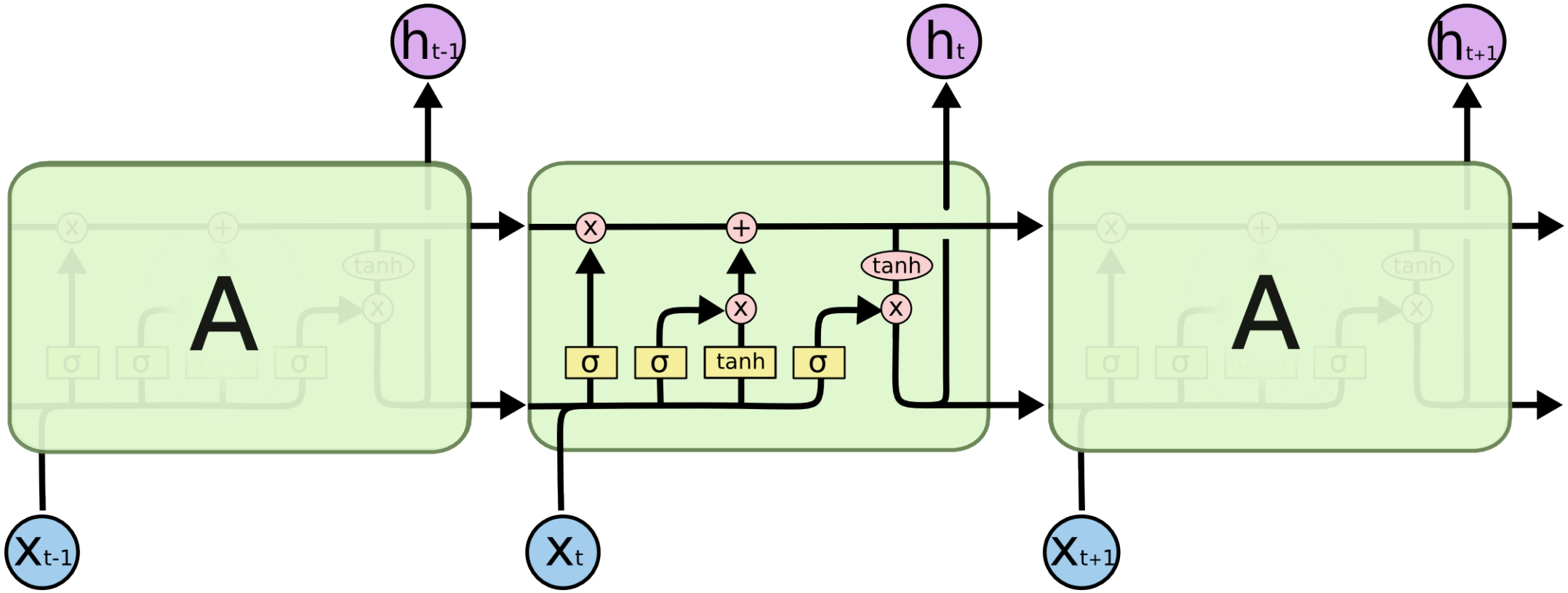

0 0 0 0 0 0 0
0 1 0 0 0 1 0
0 0 0 0 0 0 0
0 0 0 1 0 0 0
0 1 0 0 0 1 0
0 0 1 1 1 0 0
0 0 0 0 0 0 0
Input Image
Convolution
Pooling
Flattening
Convolutional Layer
Pooling Layer

Recurrent Neural Network (RNN)

Ref: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

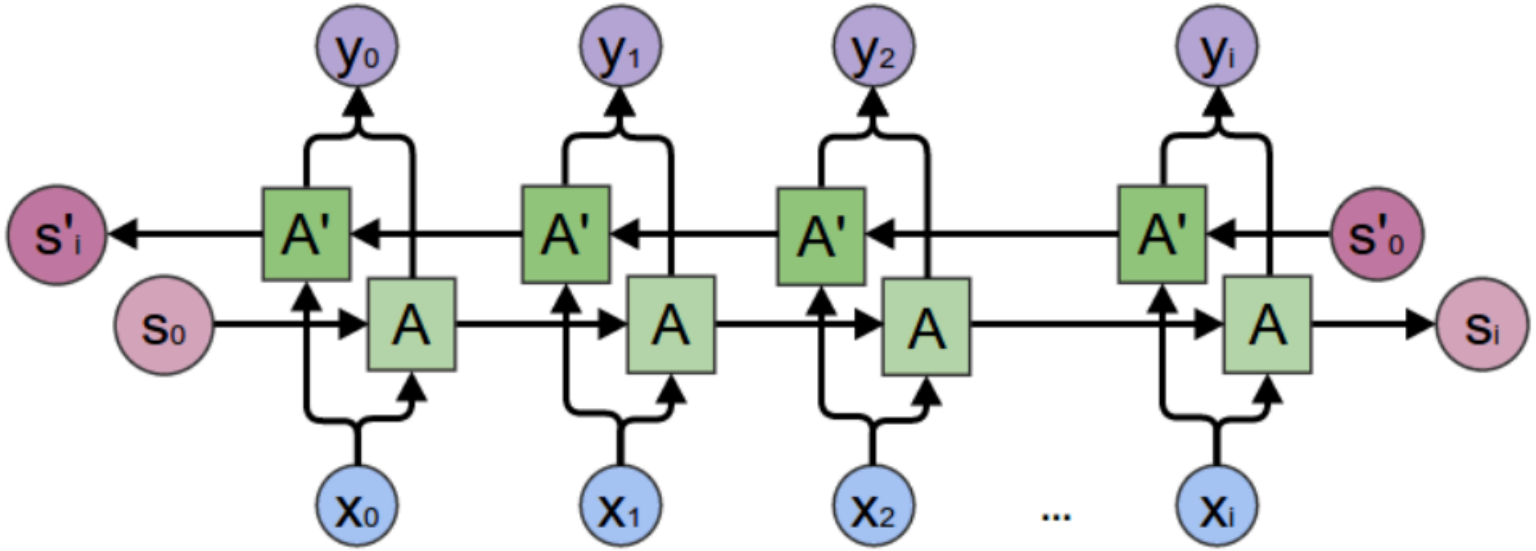


Long-Short Term Memory (LSTM)



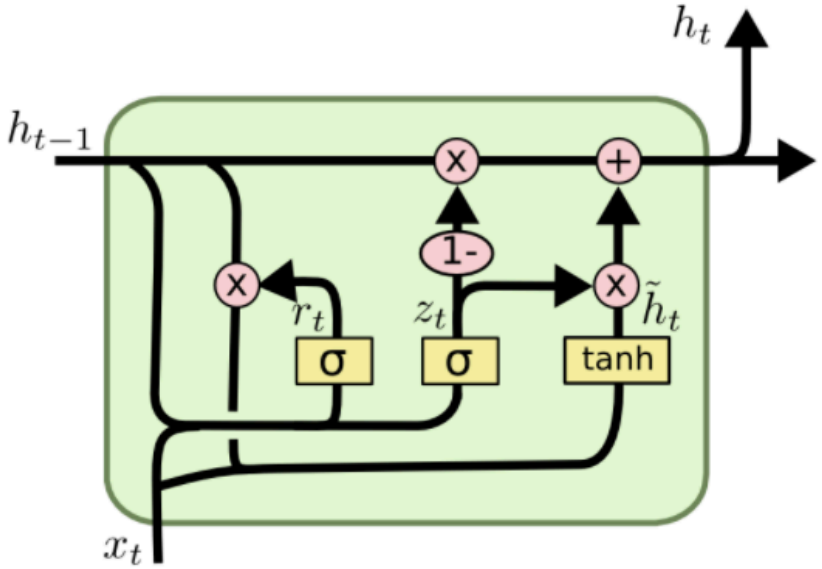
How to develop Bi-directional LSTMs?

- To enable straight (past) and reverse traversal of input (future), Bidirectional RNNs, or BRNNs, are used.
- A BRNN is a combination of 2 RNNs - **one RNN moves forward, beginning from the start of the data sequ**ence, and the **other, moves backward, beginning from the end of the data sequence**.
- The network blocks in a BRNN can either be simple RNNs, GRUs, or LSTMs.



Gated Recurrent Unit (GRU)

1. A slightly more dramatic variation on the LSTM is the Gated Recurrent Unit, or GRU.
2. It **combines the forget and input gates into a single "update gate." (z_t)**
3. It also **merges the cell state and hidden state**, and makes some other changes.
4. The resulting model is simpler than standard LSTM models, and has been growing increasingly popular.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$