Lecture 1.6
Machine Learning Fundamentals

Outline

- What is Artificial Intelligence (AI)?
- Types of AI
- What is Machine Learning (ML)?
- Types of ML
- What is Artificial Neural Network (ANN)?
- What is Learning in Neural Network?
- What is Deep Learning?
- Types of Layers
- CNN architectures
- RNN / LSTM / GRU
- Transfer Learning

Learning objectives

- Define artificial intelligence
- Define machine learning
- Define deep learning
- Define neural network and types of layers
- Explain convolution neural network
- Explain types of artificial neural networks

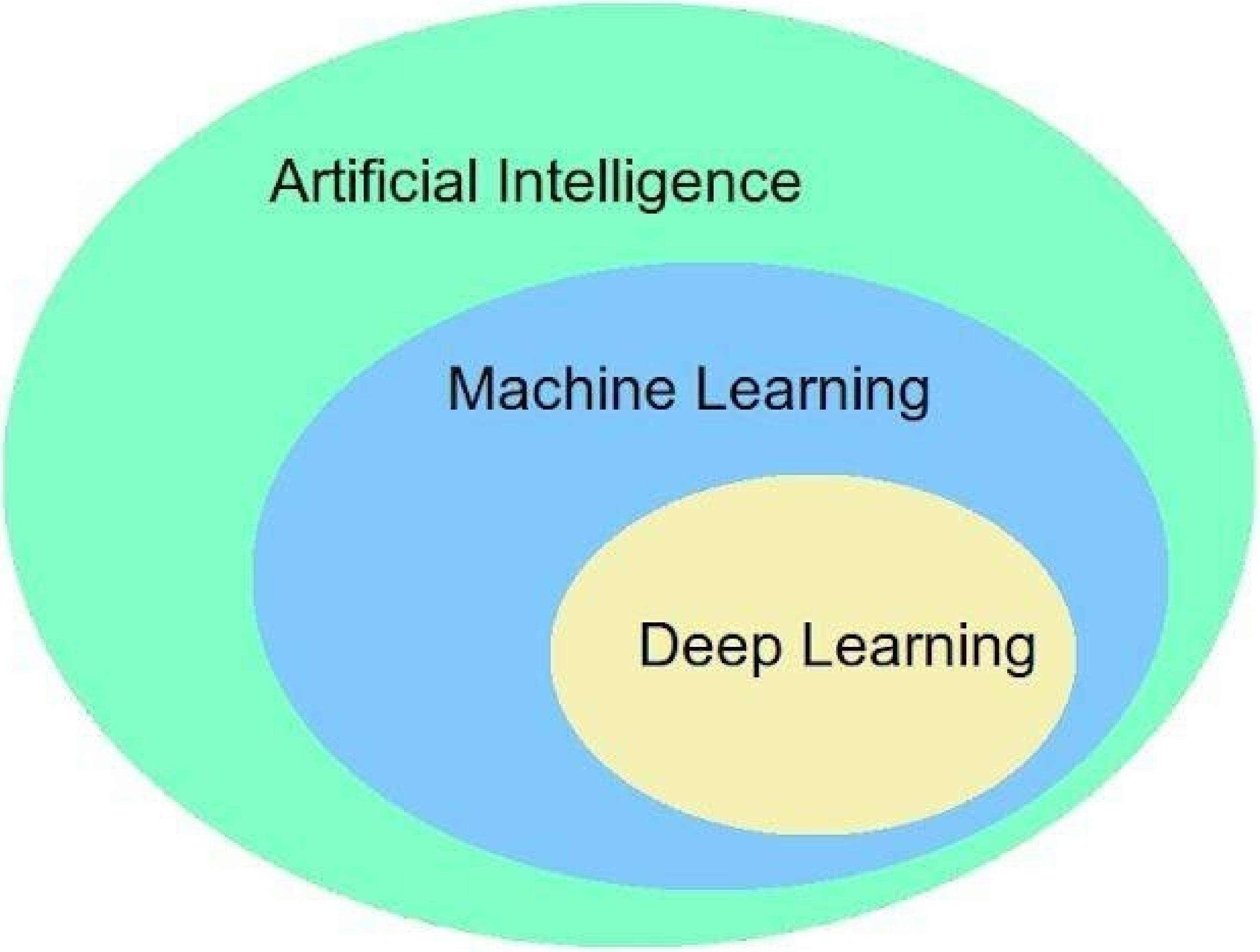
AI – Machine Learning – Deep Learning

1950’s:
First **Artificial Intelligence** concepts

1980’s:
digitalization, faster processors => **Machine Learning**

2000’s:
Large (deep) neural networks => **Deep Learning**

***Data Science** produces **insights**.*
***Machine Learning** produces **predictions**.*
***Artificial Intelligence** produces **actions**.*
David Robinson (@drob)



What is Artificial Intelligence?

- Intelligence exhibited by machines programmed to sense, think and act like humans
- **Humans:** sensory receptors, reasoning, decisions, actions
- **AI** mimics human intelligence and behaviour
    - sensors
    - processors
    - mechanical parts
- AI is now better than us in recognizing images, playing games, medical diagnostics...and driving cars (almost!)
- Future: emotions, sense of humor, ethical decisions

Types of AI

Types of Artificial Intelligence

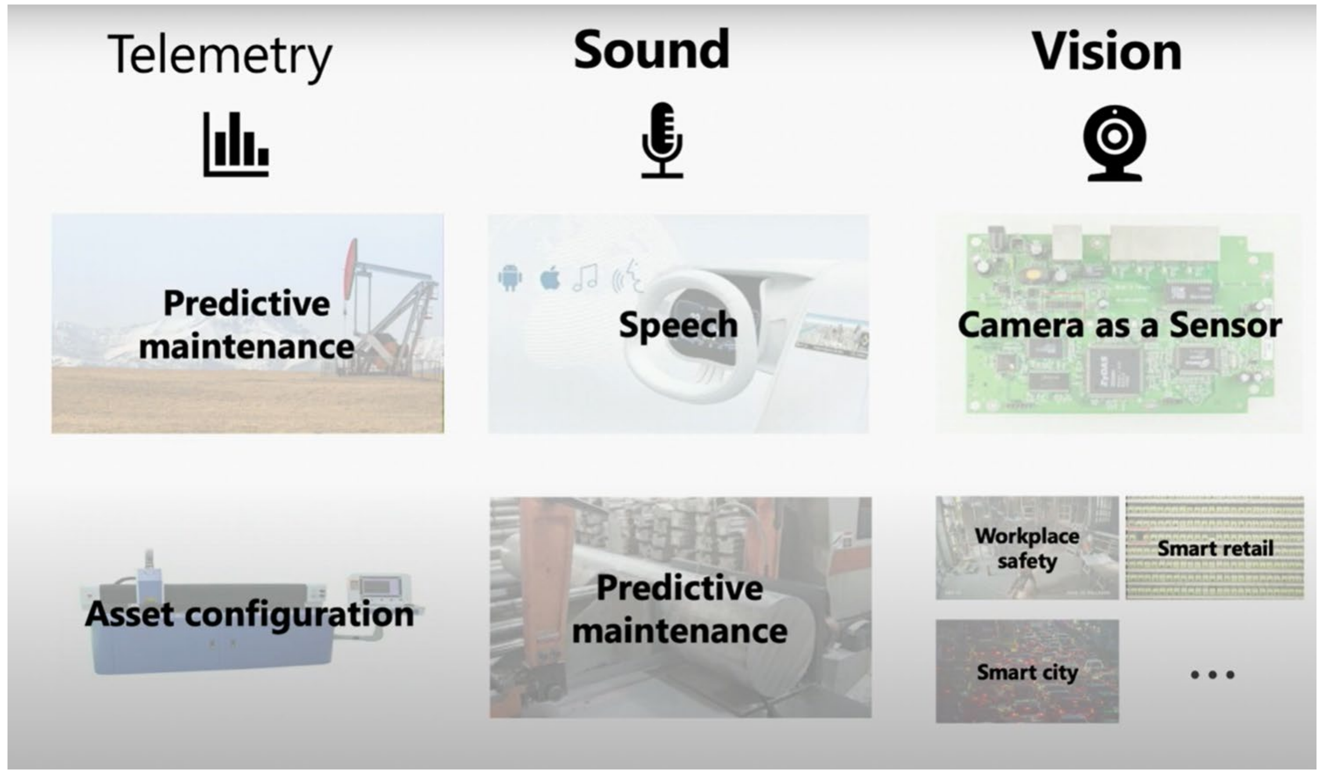
Weak AI (Narrow AI)
Can do only specific tasks like humans or better
Traditional: No learning; algorithm rules and parameters are hand-crafted and fixed
Modern: Learning from experience, machine is trained on data (relies on humans)

Strong AI
Equal to human intelligence
Can solve many types of problems
Can choose the problems it wants to solve
Has Self-awareness; independent - no human intervention needed

Super AI
Superior to human intelligence

Artificial Intelligence at the Edge

- The "Edge" means local (or near local) processing, not in the cloud
- Cloud can still be used for training of AI models
- AI models will then be performed at the edge



What is Machine Learning?

Sub-field of AI that relates to making computers learn from experience.

Requires:

data

- used in learning (training) how to accomplish some task
- Features

algorithm

- improves automatically through experience (training)

Types of Machine Learning

Algorithm requires a **feedback** during training (**learning**) phase.

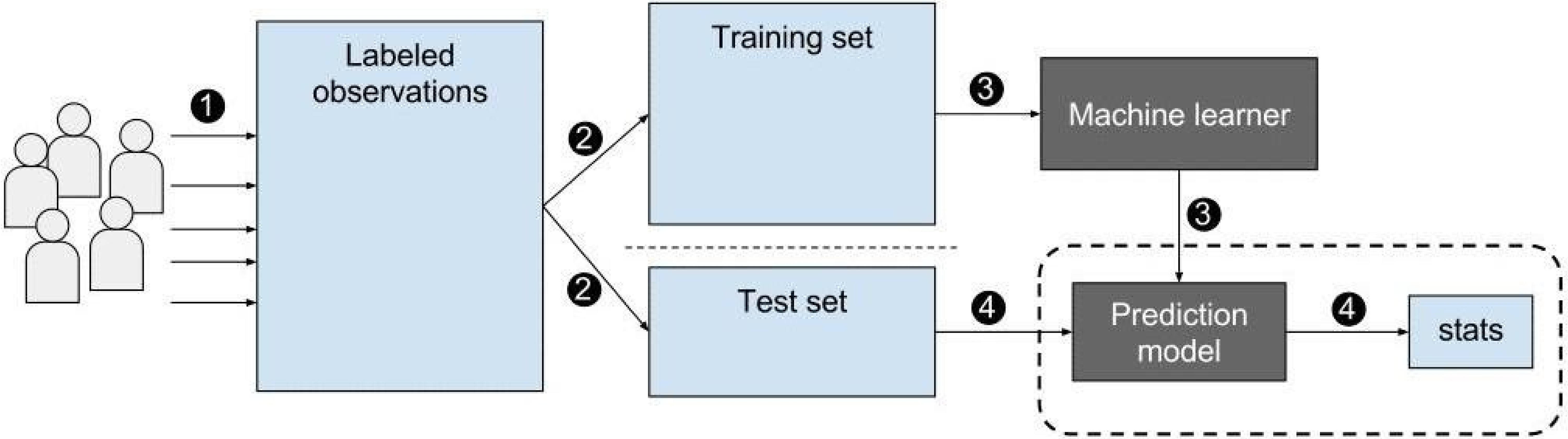
Machine Learning types, by the nature of that feedback:

- Supervised
- Unsupervised
- Semi-supervised
- Reinforcement

Types of ML

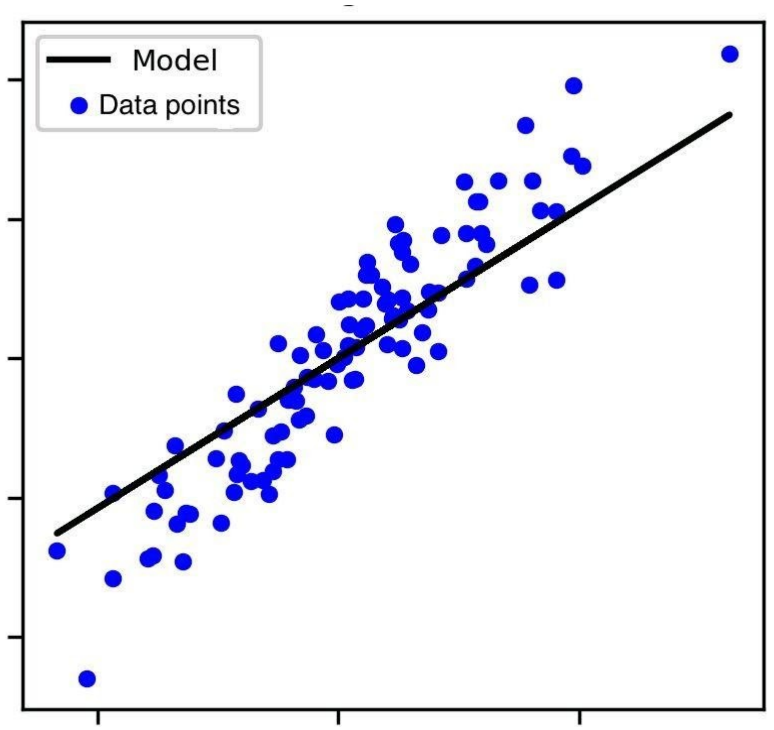
Supervised Machine Learning

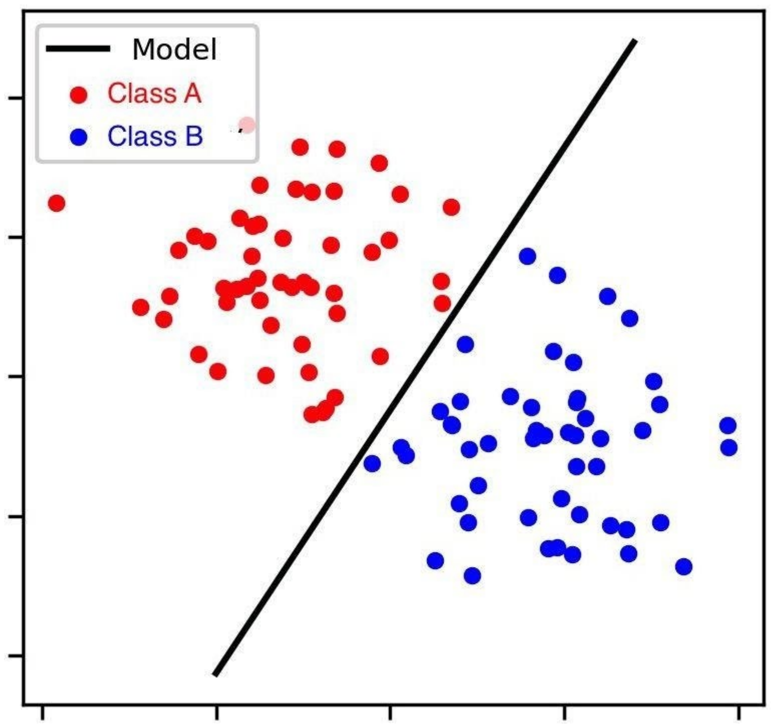
- Training data: labeled (**ground truth** - expected algorithm result; expensive!)
- Training: Algorithm uses labels to evaluate its accuracy on training data.
- Not much training data required. Risk of **overfitting**.



Supervised Machine Learning

2 types of problems it tries to solve:





Regression

- predict numerical (continuous) value
- Linear, Nonlinear Regression

Classification:

- predict categorical (discrete) value
- Naive Bayes Classifier, *Support Vector Machines*, Logistic Regression, ...

Decision Tree, Random Forest, k-NN, Neural Networks, etc...can solve both problems

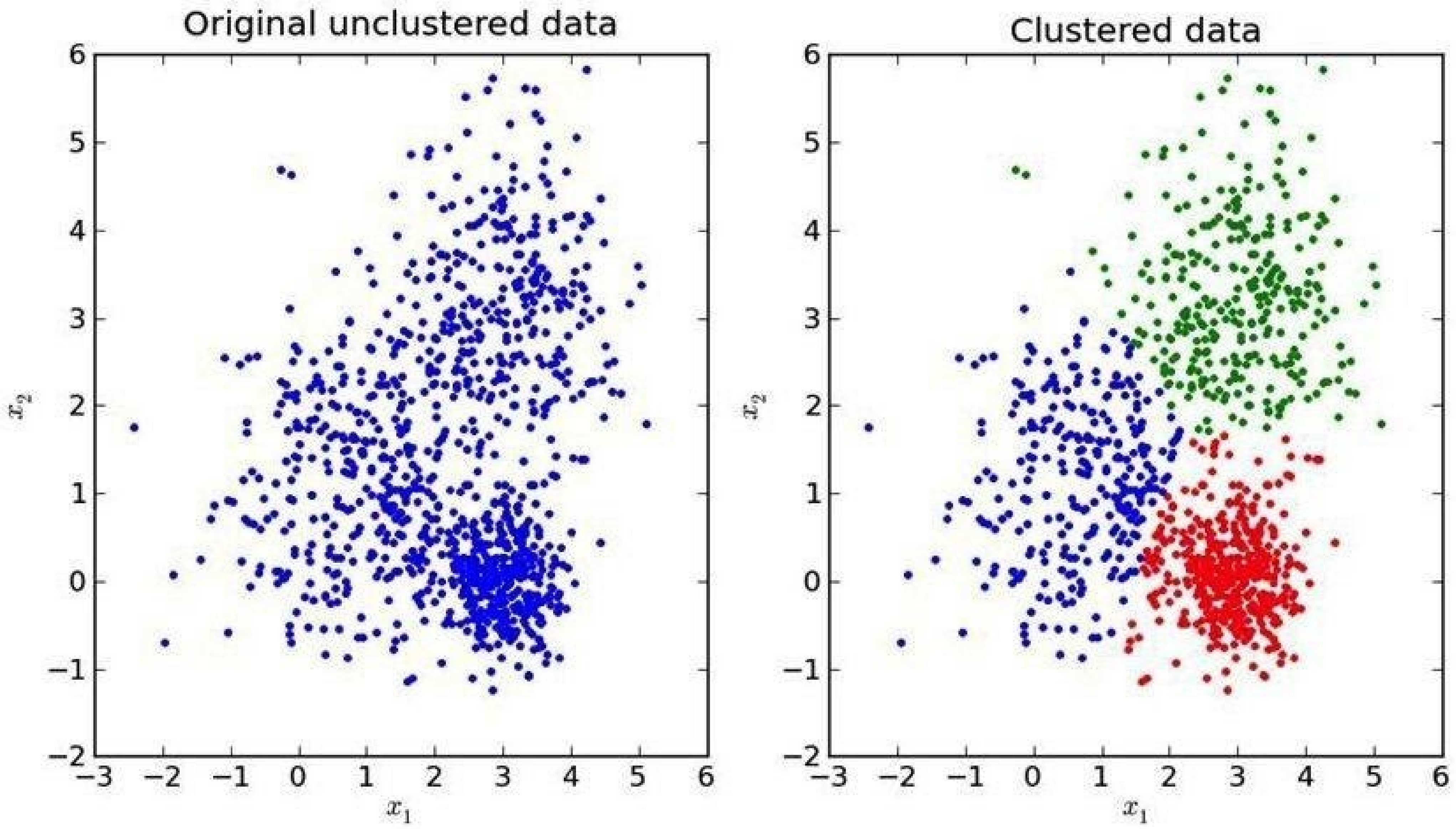
Unsupervised Machine Learning

Training data: Unlabeled data

Training:

- extract **features** and patterns from data itself
- **clustering**: these features used to label and classify the data into clusters

k-Means clustering, ...

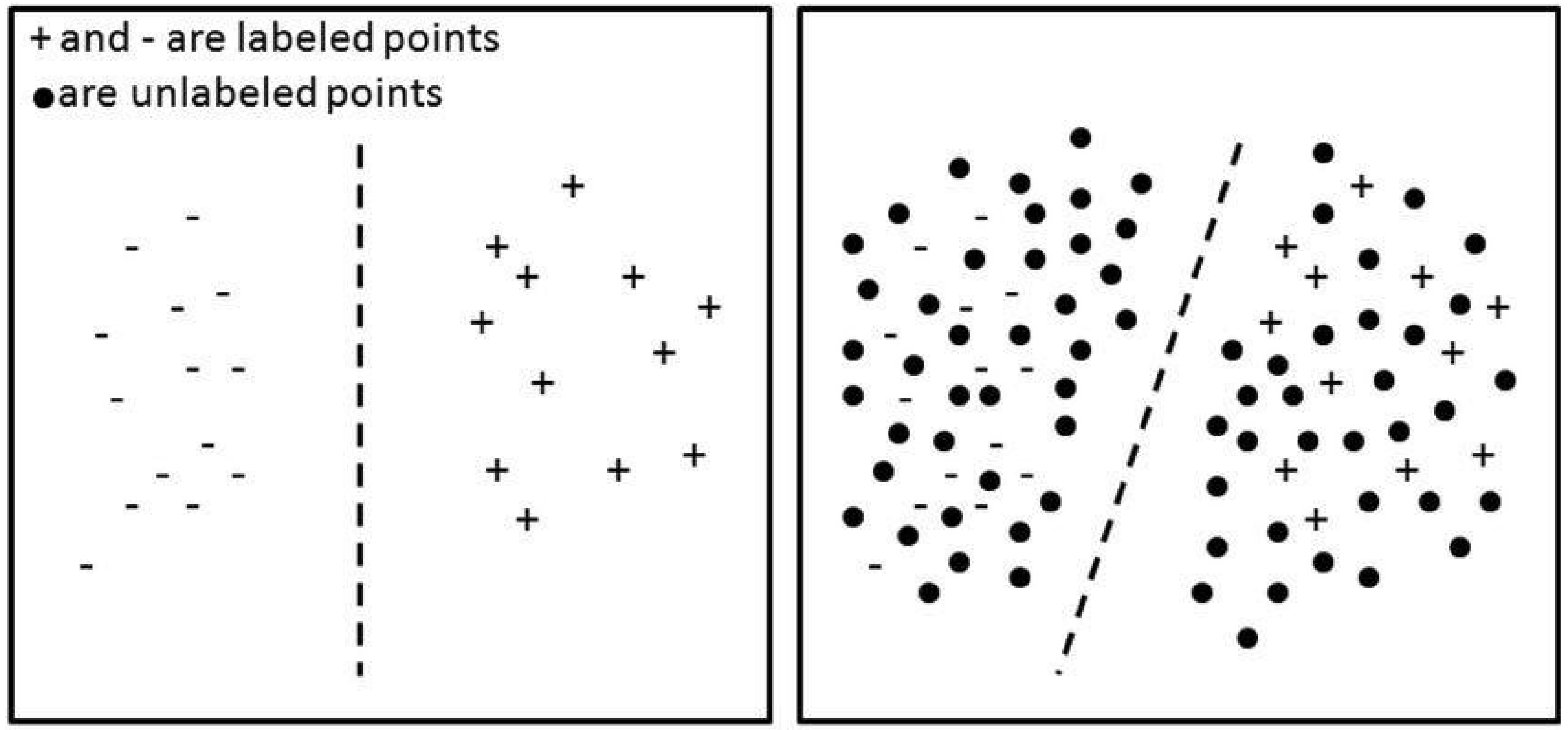


Semi-supervised Machine Learning

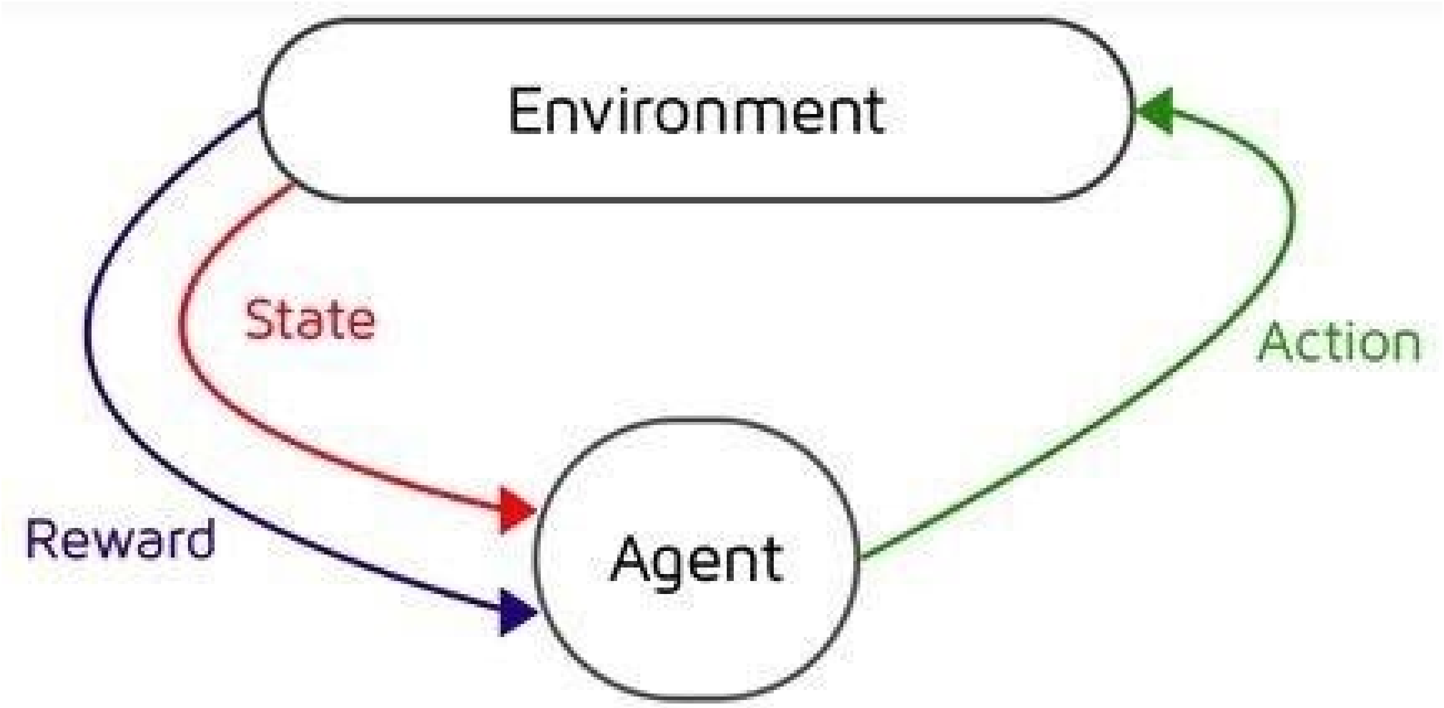
Hybrid learning - between supervised and unsupervised
Solves the problem of having not enough labeled data to train a supervised learning algorithm

Training data: small labeled and large unlabeled data set
Training:

- Train model with labeled data
- Trained model used to predict labels for unlabeled data => **pseudo-labeled data**
- Retrain model with both pseudo- and labeled data sets



Reinforcement Machine Learning



Training data: none

Training:

- Machine trained to make specific decisions
- Machine interacting with its environment
- Trial and error
- **Reward system**: providing feedback when an artificial intelligence agent performs the best action in a particular situation
- Sequence of successful outcomes is reinforced to develop the best solution for a given problem.

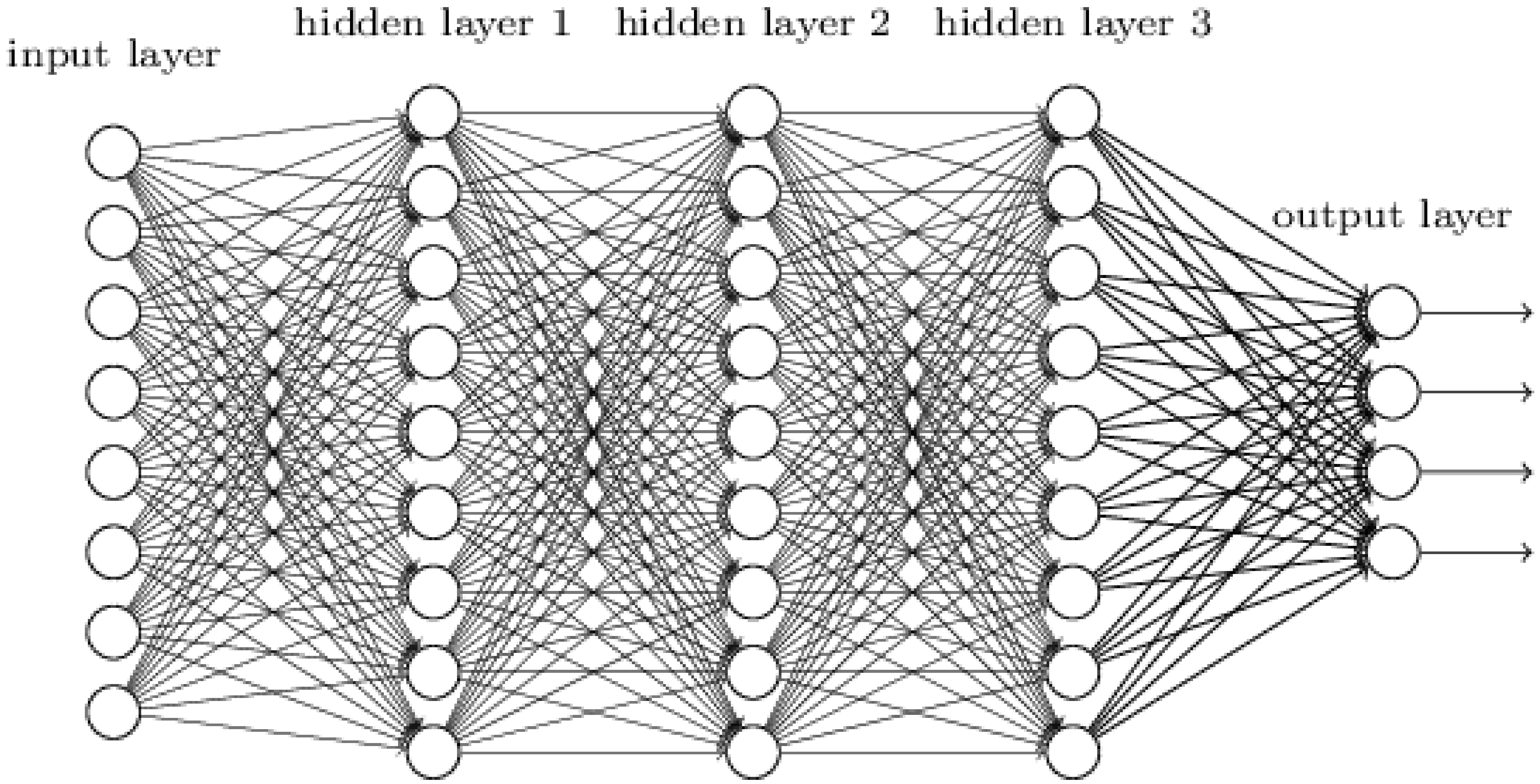
Markov Decision Process

What is Deep Learning?

- **Deep Neural Networks** - more than one hidden layer (vs **Shallow Network**)

- With each new hidden layer system becomes more intelligent; increases capabilities to learn: **new** features and **more complex** features

- Each feature reflects one detail from the input

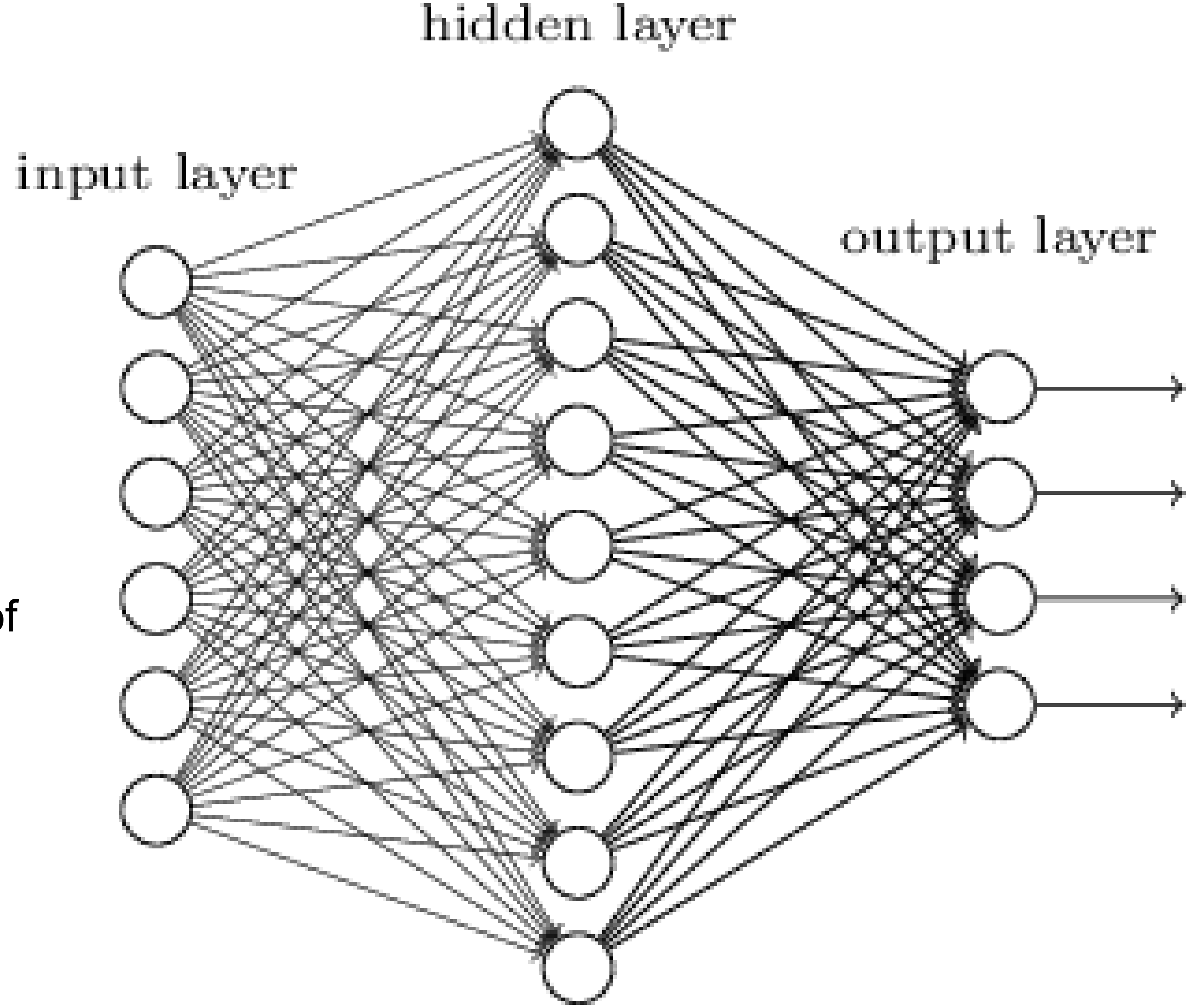


Artificial Neural Network

Collection of simple, trainable, interconnected mathematical units (neurons) that collectively learn complex function

System of interconnected **layers**:

- **input**: numeric representation of the data
- **hidden**: nonlinear function (**activation function**) of the sum of weighted inputs from the previous layer plus bias
- **output**: prediction - set of values (continuous ⇒ model solves **regression** problem; discrete ⇒ **classification**



y = F(w 1 x 1 + w 2 x 2 + ... w N x N + b);
F(x) = max(0, x); (ReLU)
xi - input, wi - weight

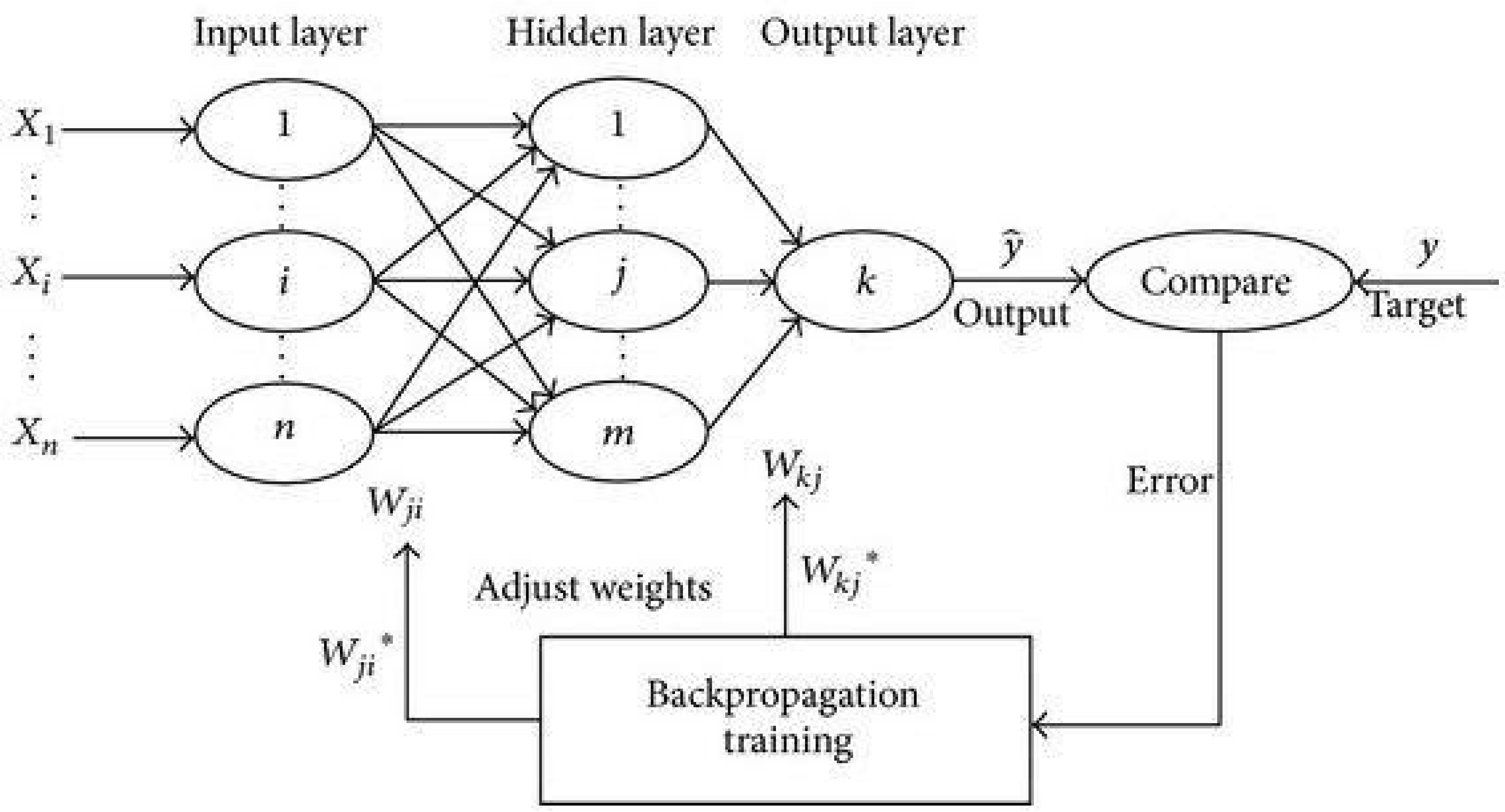
What is Learning in Neural Network?

Learning (training):

- Auto-adjusting model parameters with the each new input sample so the output (**prediction**) gets closer to expected values (**ground truth**)

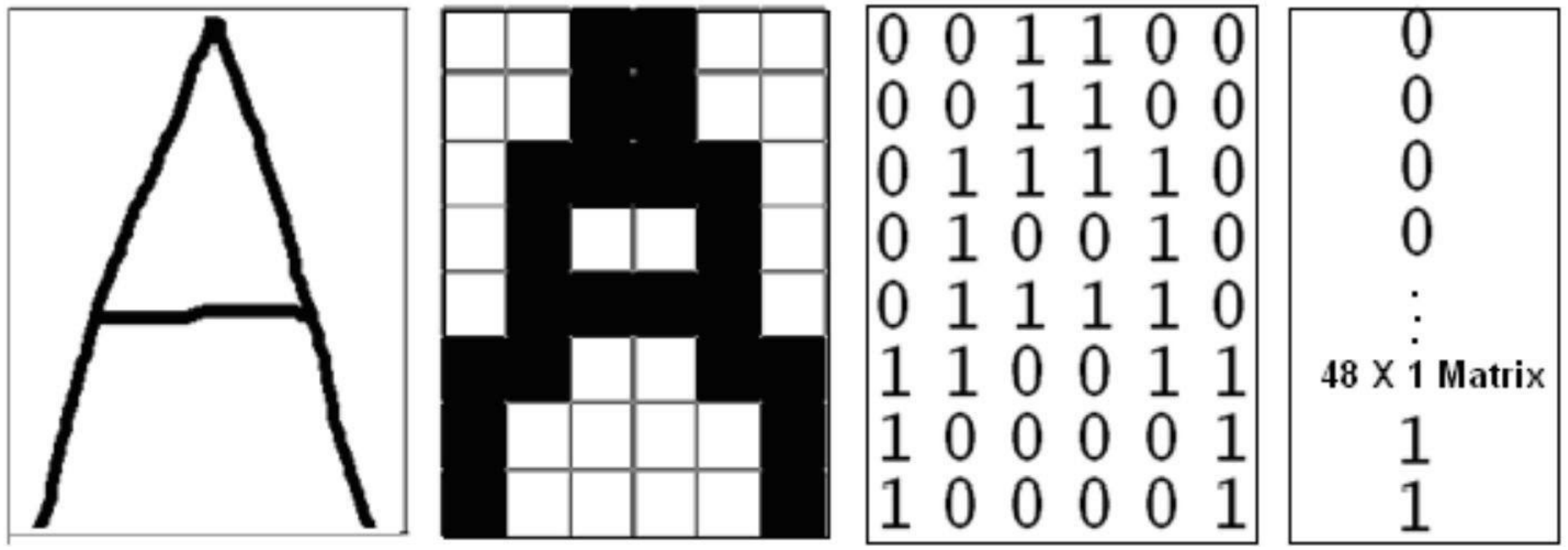
- forward-propagating activations for labeled input and back-propagating

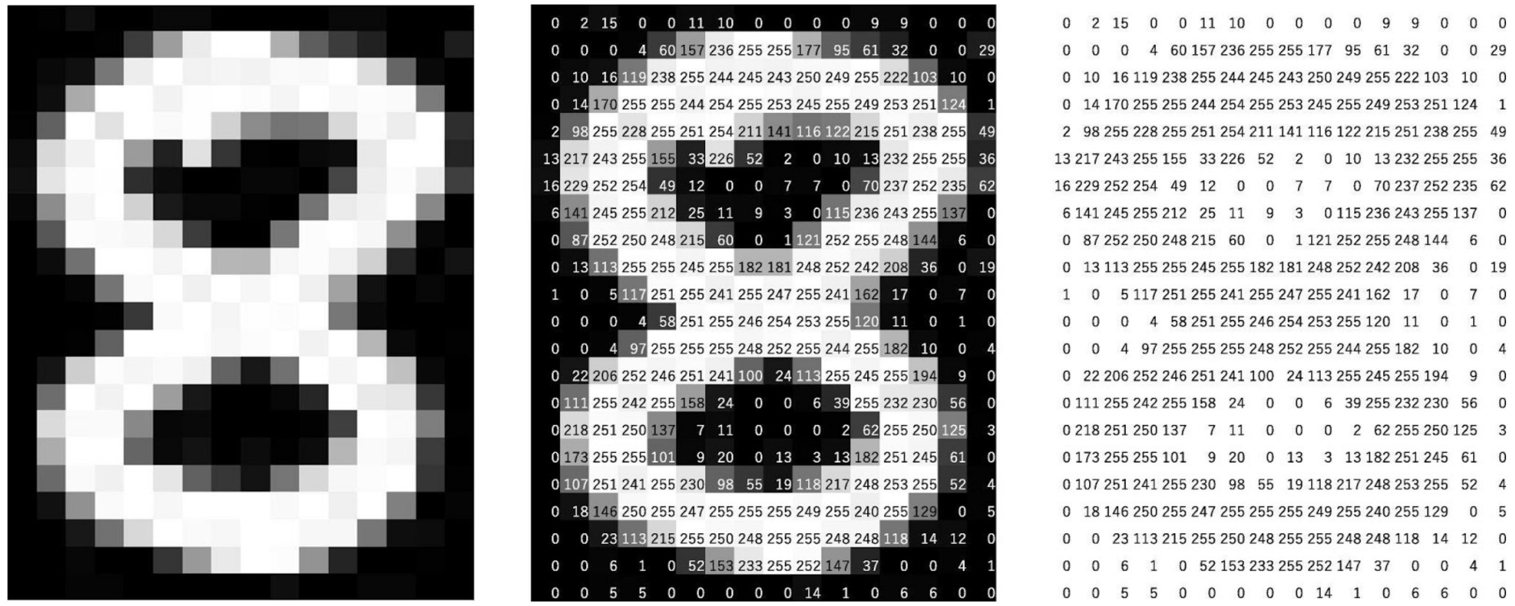
Errors to adjust parameters in each node in order to minimize loss function



Types of Layers

Neural Network Input: Digital Image

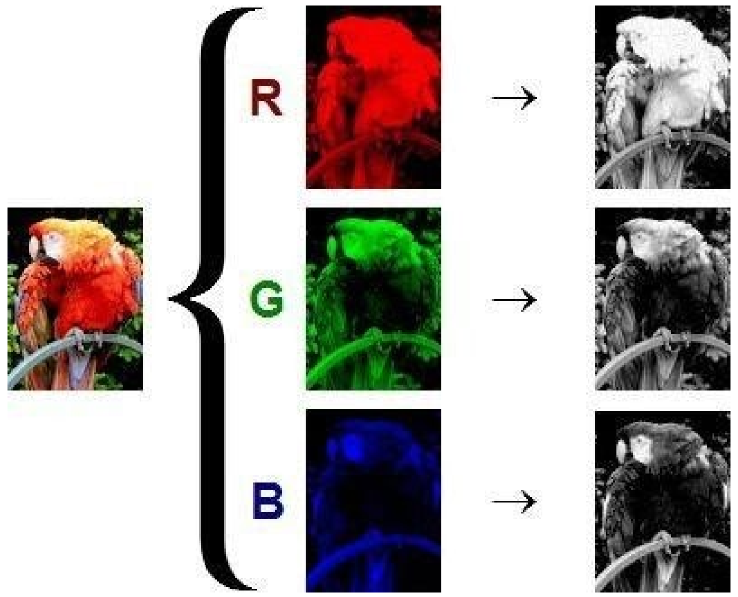


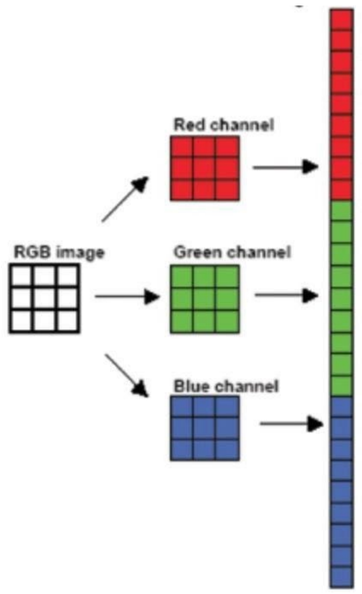


Digital image - 2D set of pixels.

Each **pixel** has numeric value(s) associated to it:

- Monochrome: 0, 1
- Grayscale: 0 (black) - 255 (white)
- Colour: 0 - 255 (in each channel: R, G, B)



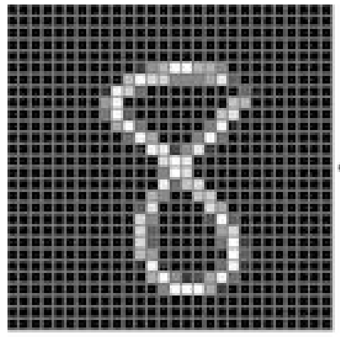


Computer sees an array of numbers.

Fully-connected Input Layer

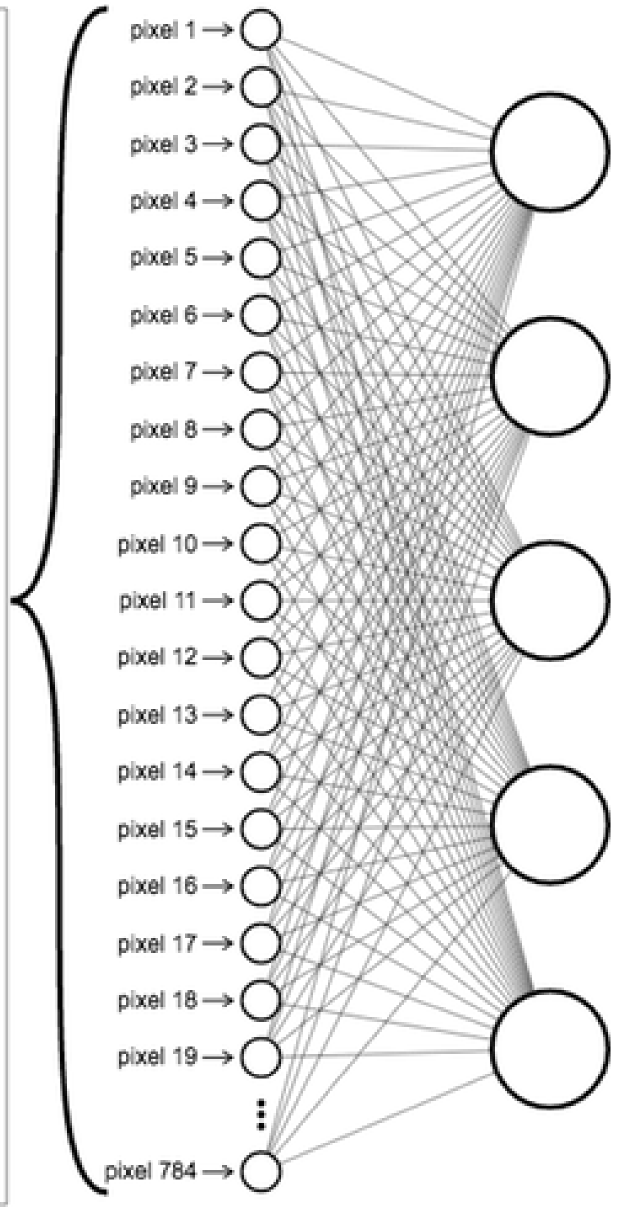
Feeding input to a fully-connected input layer:

2D intensity (or 3D colour) matrix is collapsed into 1D vector



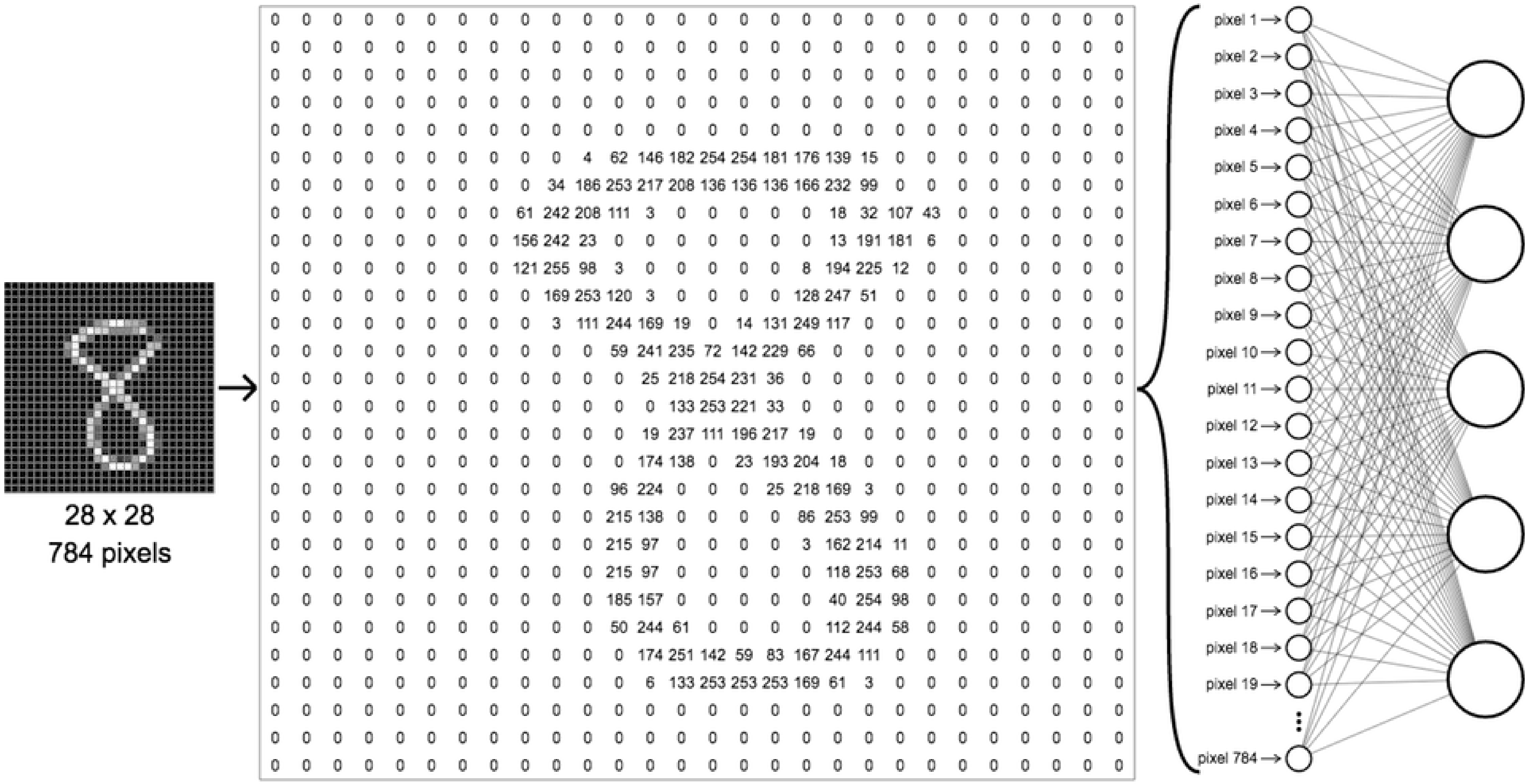
28 x 28
784 pixels

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	4	62	146	182	254	254	181	176	139	15	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	34	186	253	217	208	136	136	136	166	232	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	61	242	208	111	3	0	0	0	0	0	18	32	107	43	0	0	0	0	0	0
0	0	0	0	0	0	0	0	156	242	23	0	0	0	0	0	0	0	13	191	181	6	0	0	0	0	0	0
0	0	0	0	0	0	0	0	121	255	98	3	0	0	0	0	0	8	194	225	12	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	169	253	120	3	0	0	0	0	128	247	51	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	111	244	169	19	0	14	131	249	117	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	59	241	235	72	142	229	66	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	25	218	254	231	36	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	133	253	221	33	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	19	237	111	196	217	19	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	138	0	23	193	204	18	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	96	224	0	0	0	25	218	169	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	138	0	0	0	0	86	253	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	3	162	214	11	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	0	118	253	68	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	185	157	0	0	0	0	0	40	254	98	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	50	244	61	0	0	0	0	112	244	58	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	251	142	59	83	167	244	111	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	6	133	253	253	253	169	61	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Problem with fully-connected input layer

- Explosion of parameters
- Spatial information lost



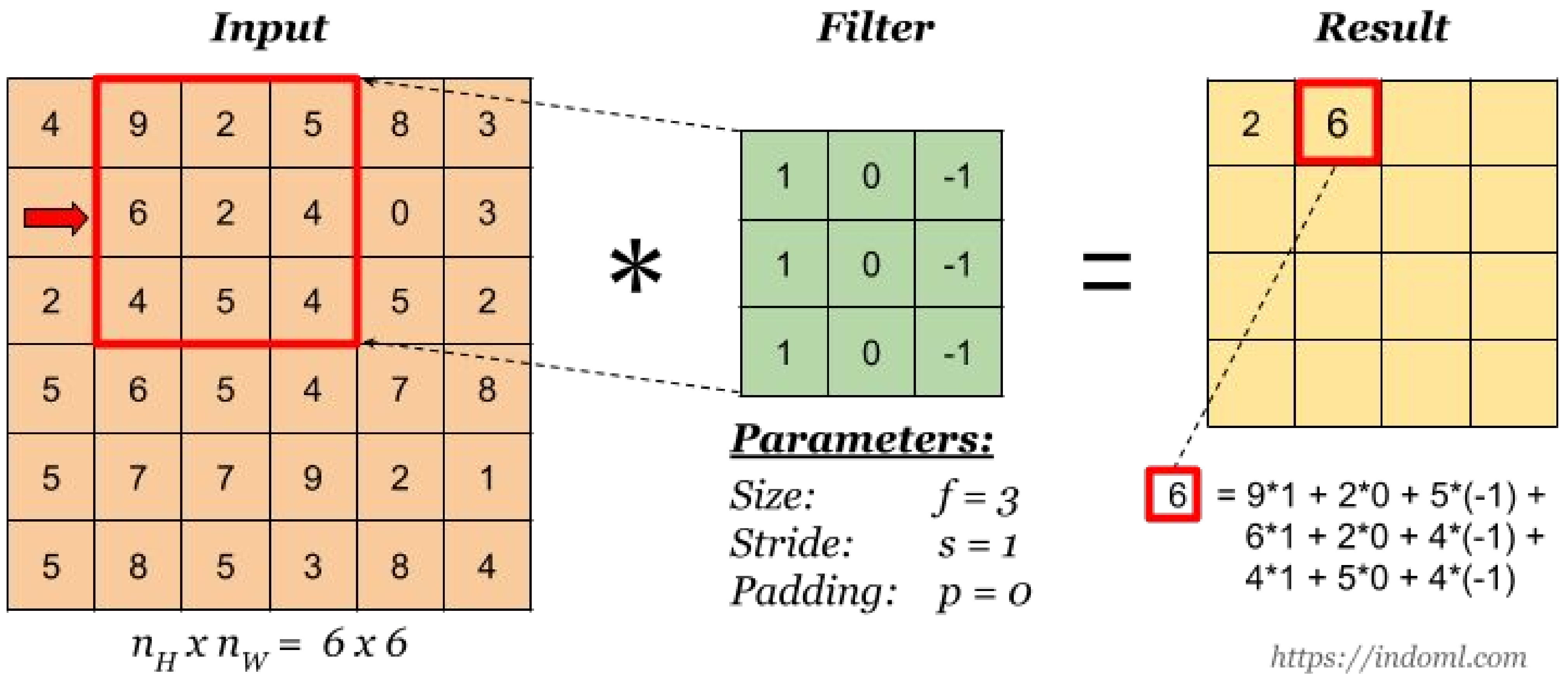
Solution: Convolution

How to preserve that valuable spatial structure of the input?

- Feed a small **patch** of the input image to a single neuron in a hidden layer
- Each neuron in that hidden layer is seeing only a patch in the input image

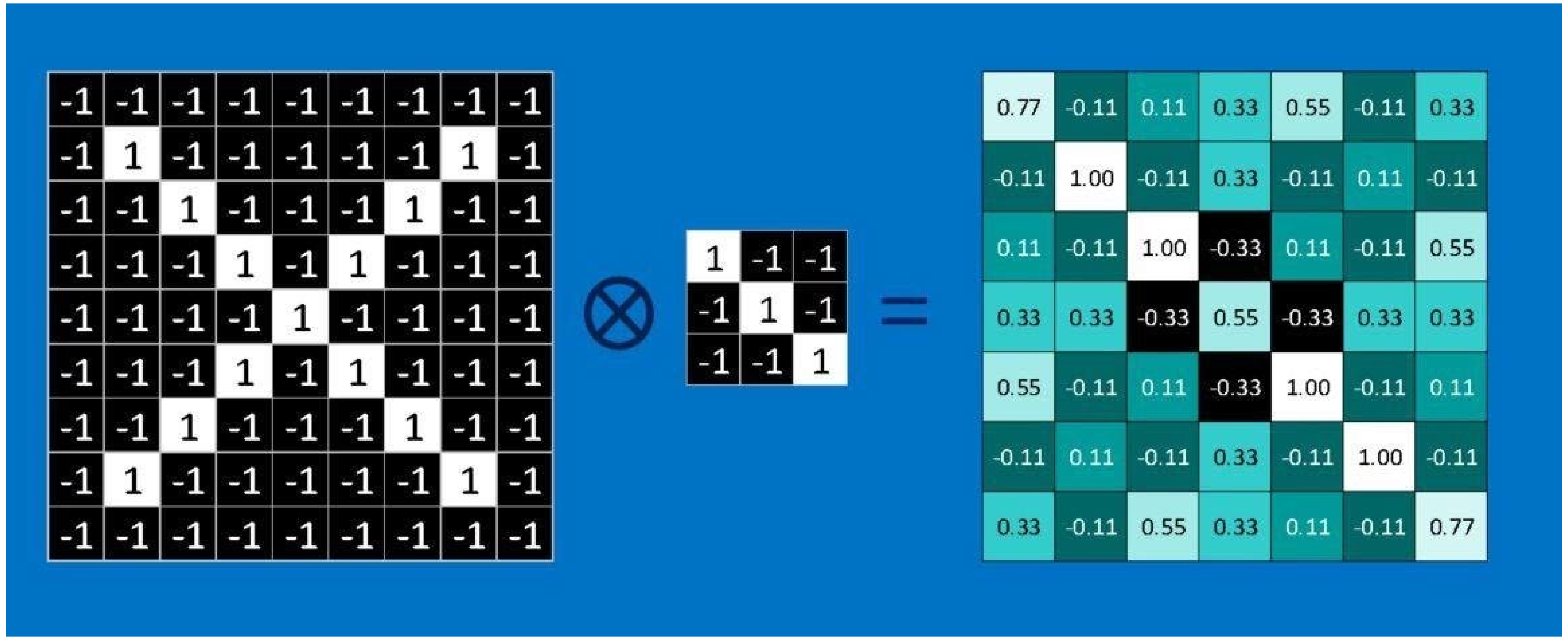
How to extract features? (fundamental task of DNN)

- Use **convolution** as a mapping function of a patch into a neuron



Convolution

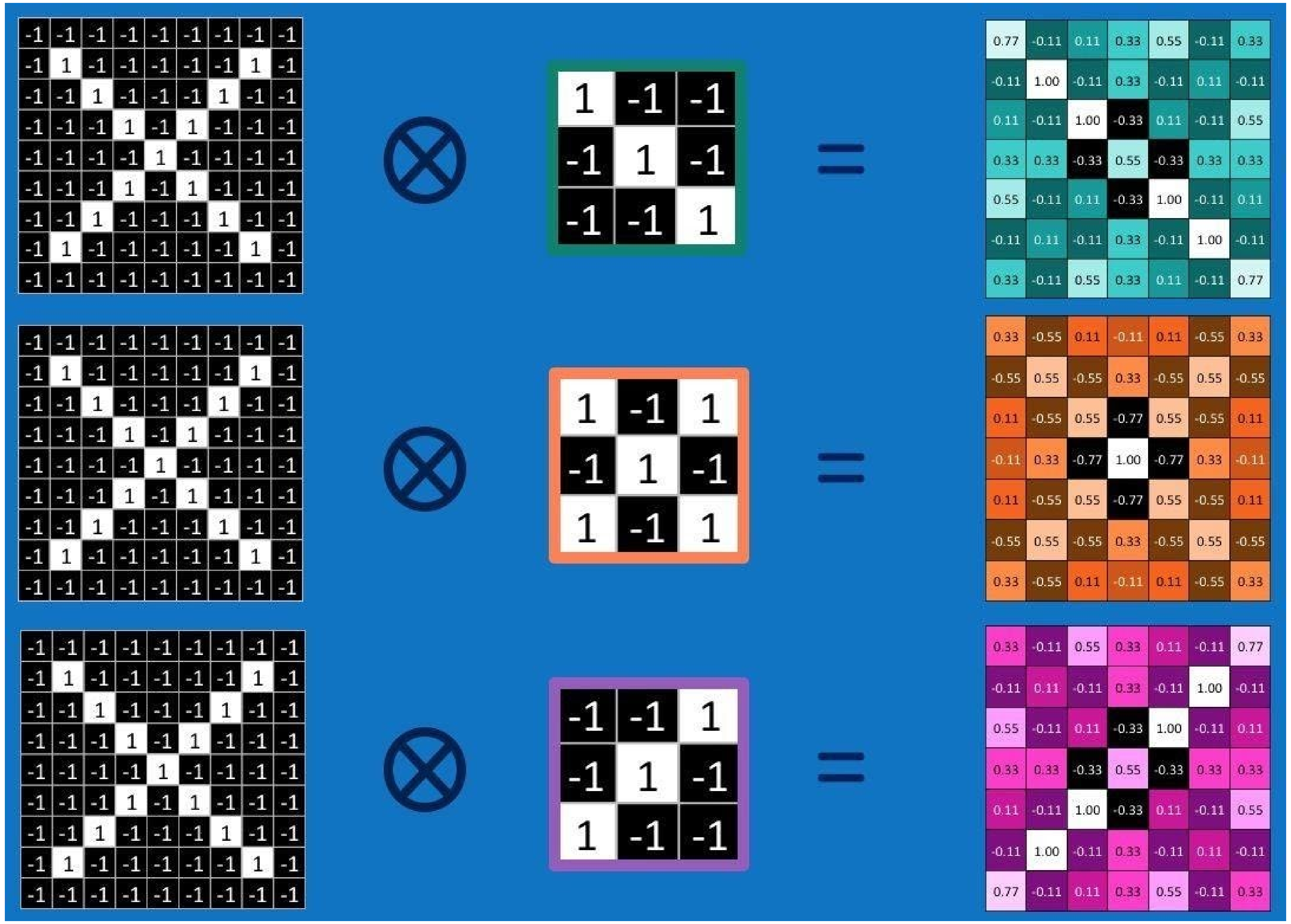
- Measure of similarity between a **filter** (**kernel**, small **feature**) and currently observed **region** in the input image.
- Detects that image contains given feature and creates an output matrix - a **feature map**. Feature map tells where in the image is the activation for this particular filter
- Depth of a filter must match the depth of the input for the filter (e.g. the number of channels).



Feature Maps

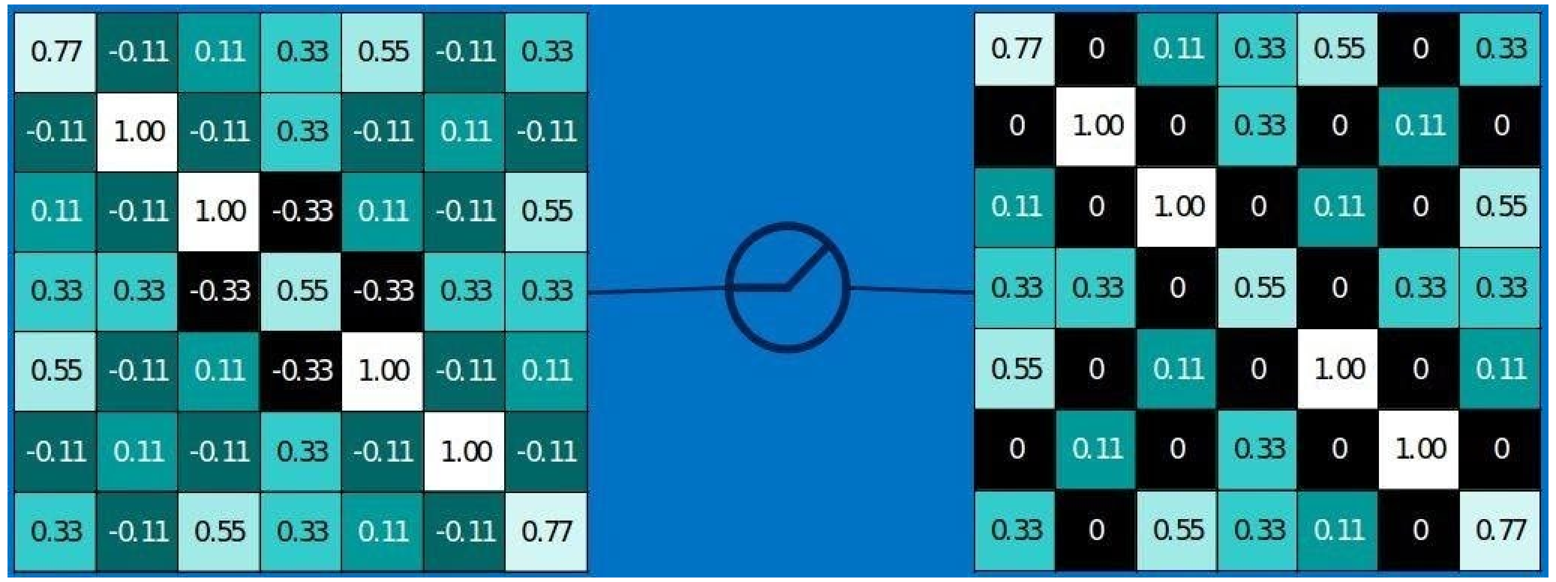
Convolution layer creates a feature map for each filter.

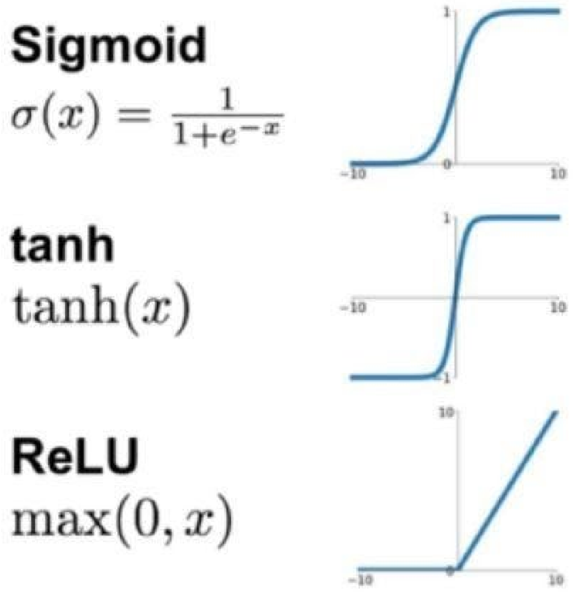
The same pixel in each feature map is connected to the same patch of the input image.



Nonlinearity Layer: Rectified Linear Unit (ReLu)

Activation function: nonlinear function converts **feature map** into the **activation map**



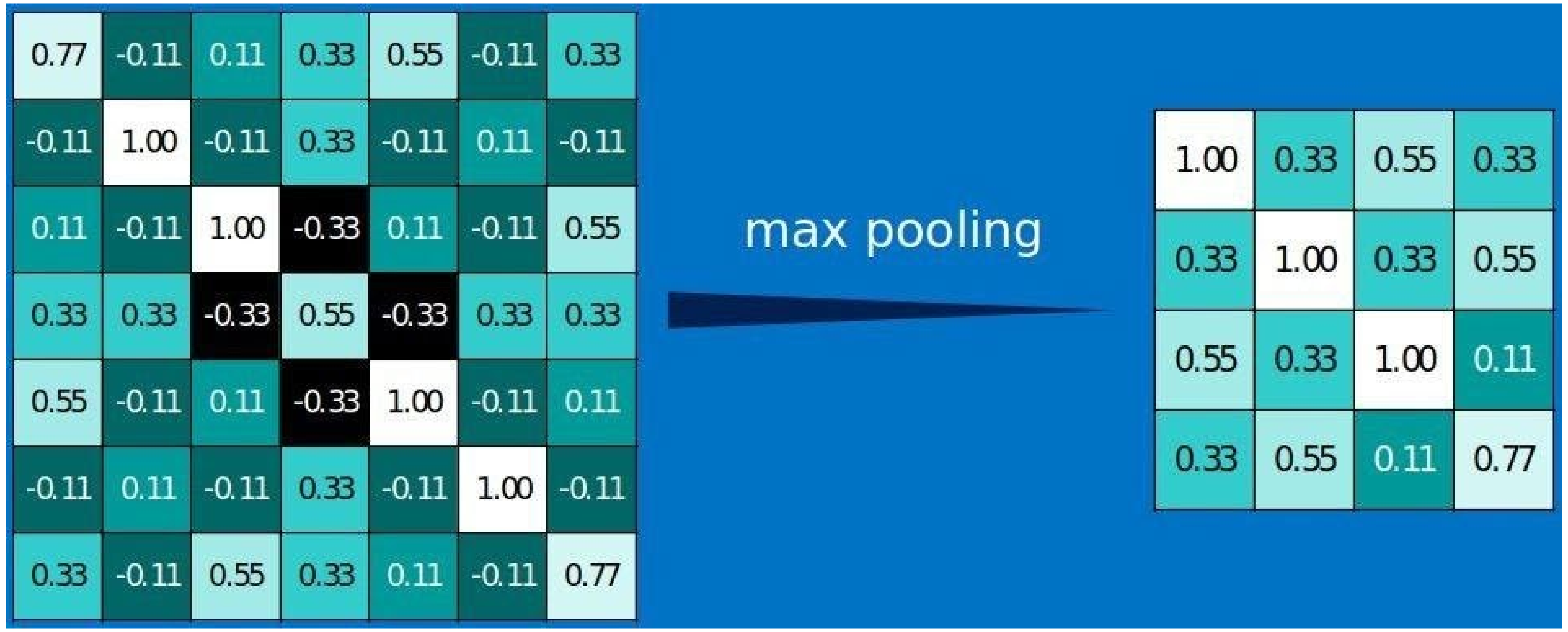


- ReLU makes CNNs perform best.
- CNN data input is usually normalized to range [-1, 1] => improves learning speed and accuracy

Pooling

Reducing the size of the feature stack by downsampling spatial resolution of each feature map =>

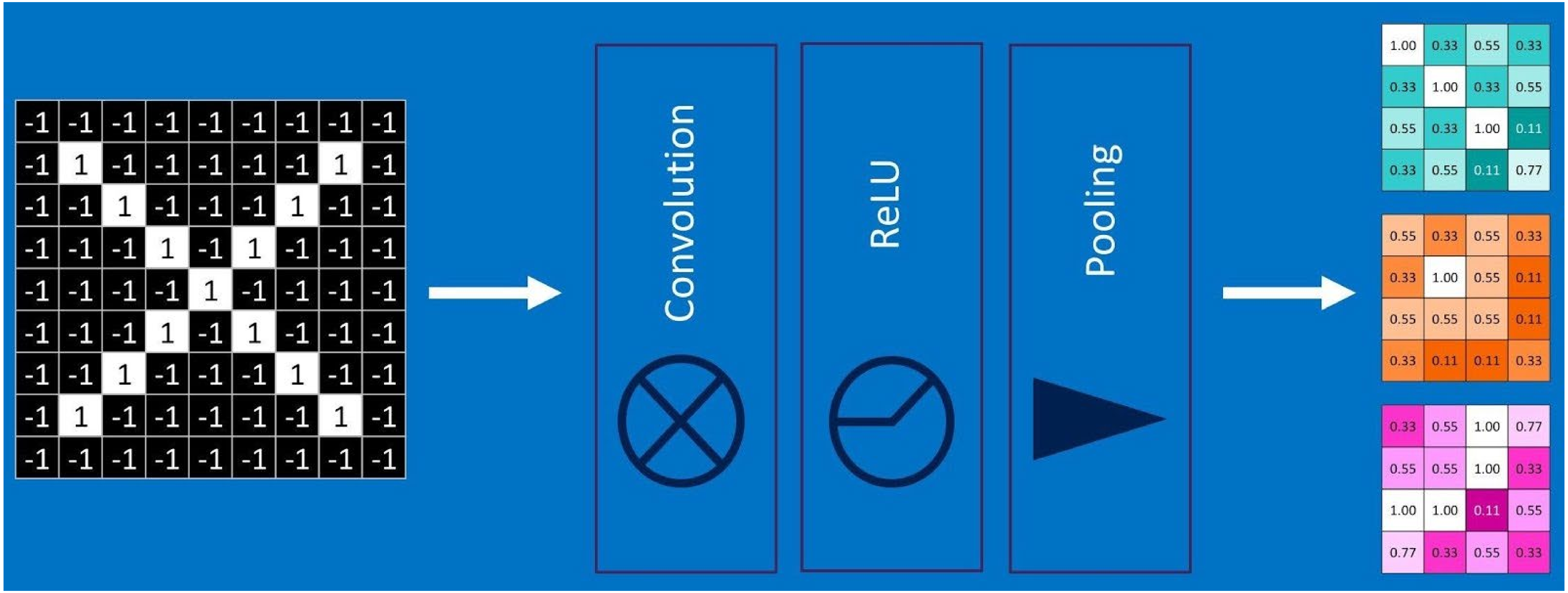
- less data for processing
- features and their spatial information are not lost



Max Pooling:

Slide the window across the image and pick the maximum value at every position.

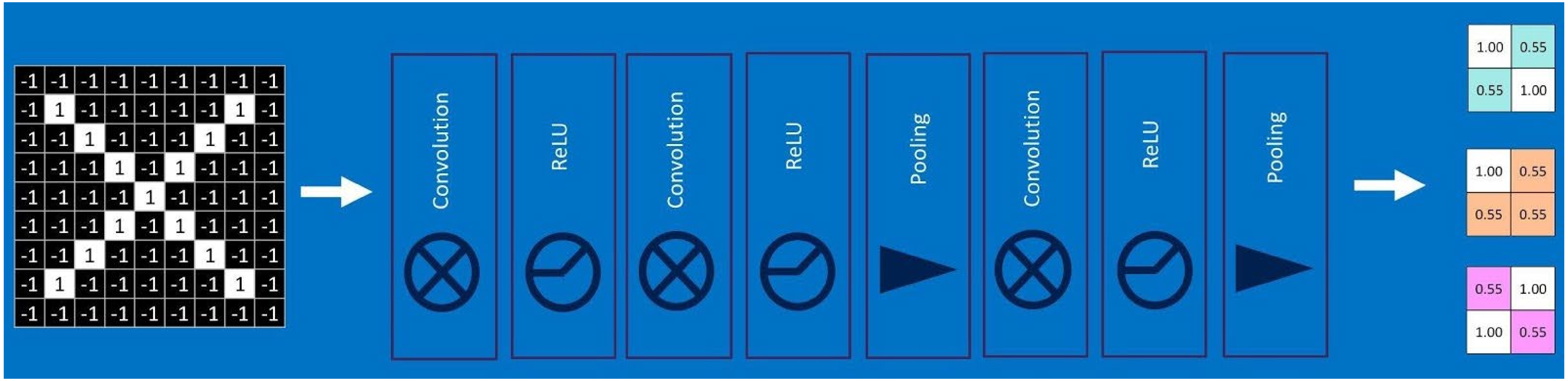
Stack



Deep Stacking

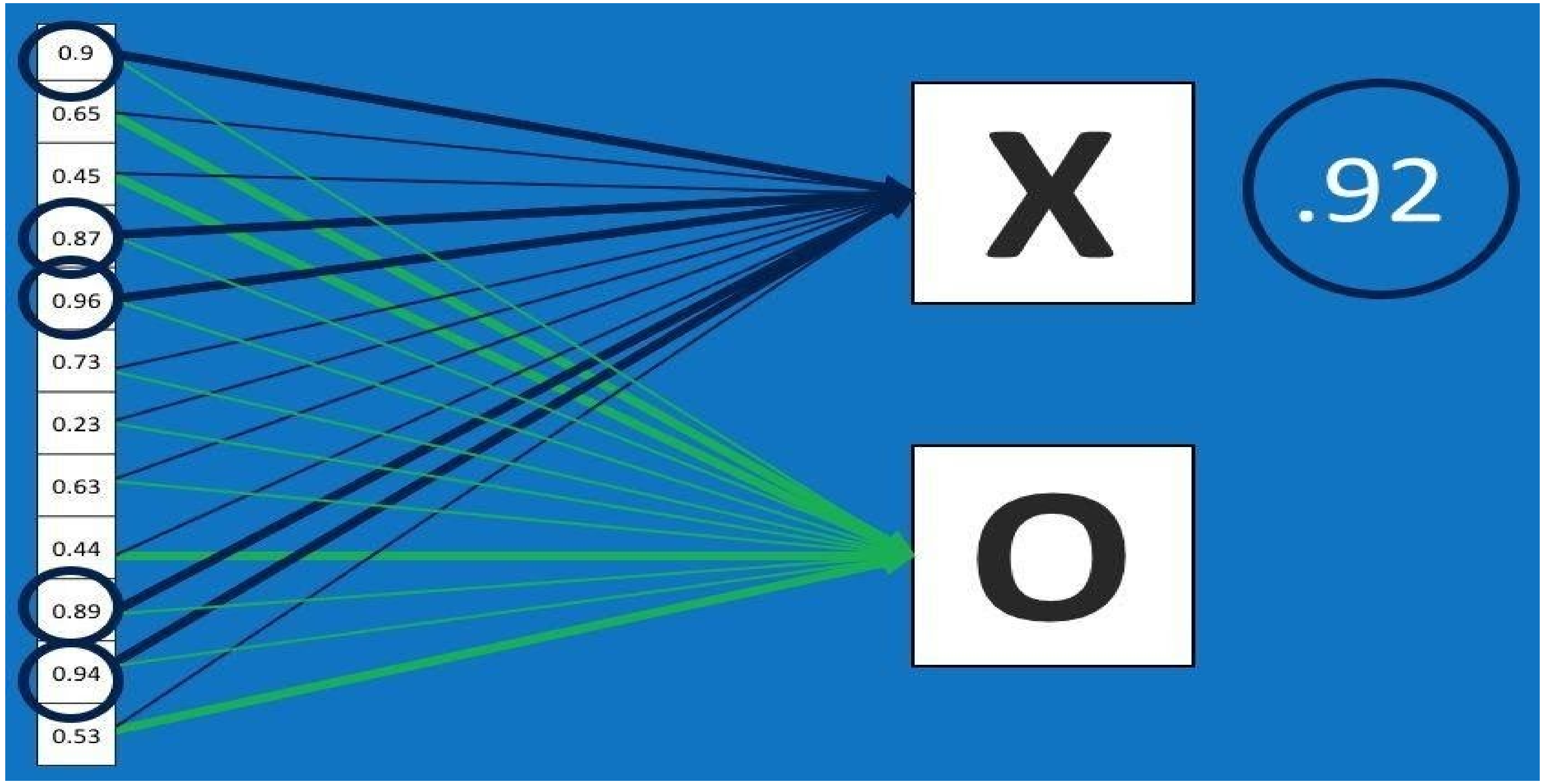
With each new convolutional layer:

- filters become larger and learn larger and more complex features
- feature maps are shrinking and become smaller



Fully-connected layers

They learn non-linear combinations of the high-level features outputted by the convolutional layers.
Features are assign weights which describe their contribution in recognizing particular object.



Softmax Layer

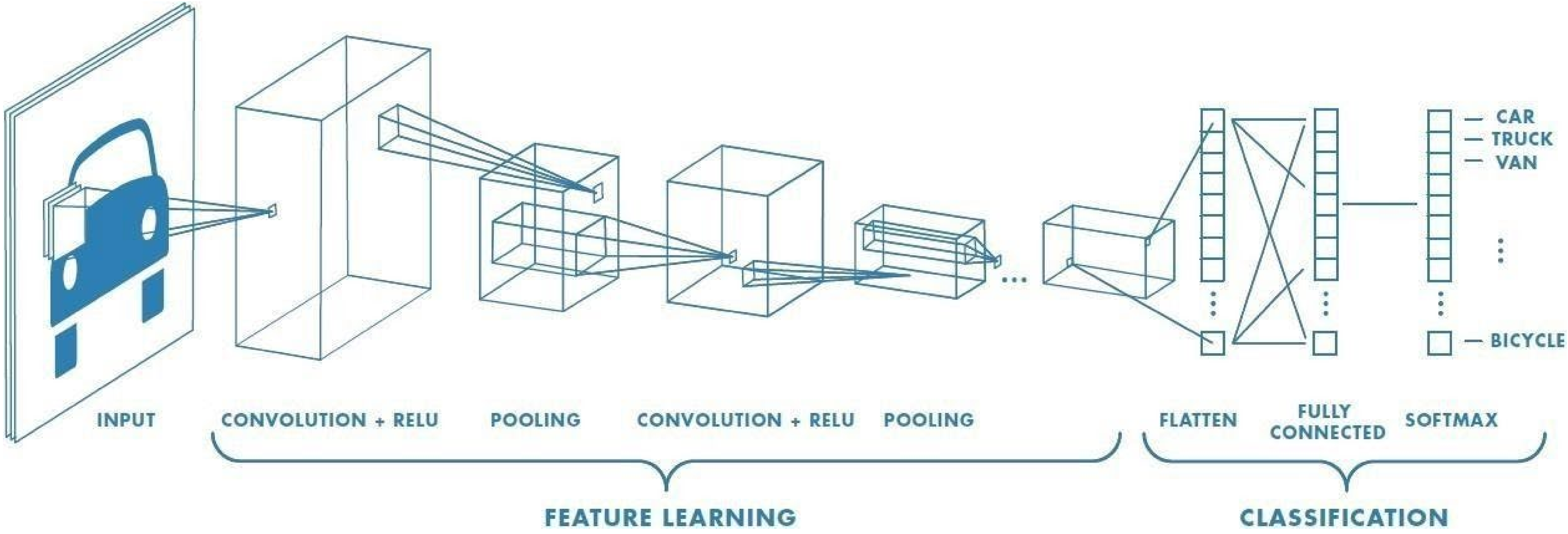
- Softmax Function
- Multiclass logistic regression
- Normalizes the output of a network to a probability distribution over predicted output classes

Last Hidden Layer	Softmax activation function	Predictions

$$\begin{bmatrix} 1.3 \\ 5.1 \\ 2.2 \\ 0.7 \\ 1.1 \end{bmatrix} \longrightarrow \boxed{\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}} \longrightarrow \begin{bmatrix} 0.02 \\ 0.90 \\ 0.05 \\ 0.01 \\ 0.02 \end{bmatrix}$$

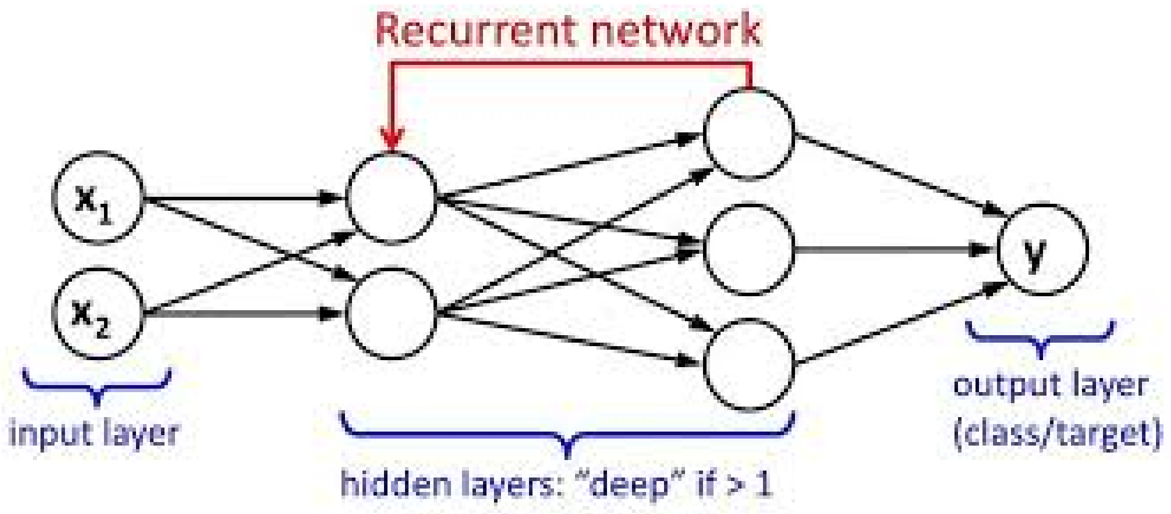
CNN Architectures

The first hidden layer is always a **convolutional layer**.



Recurrent Neural Networks

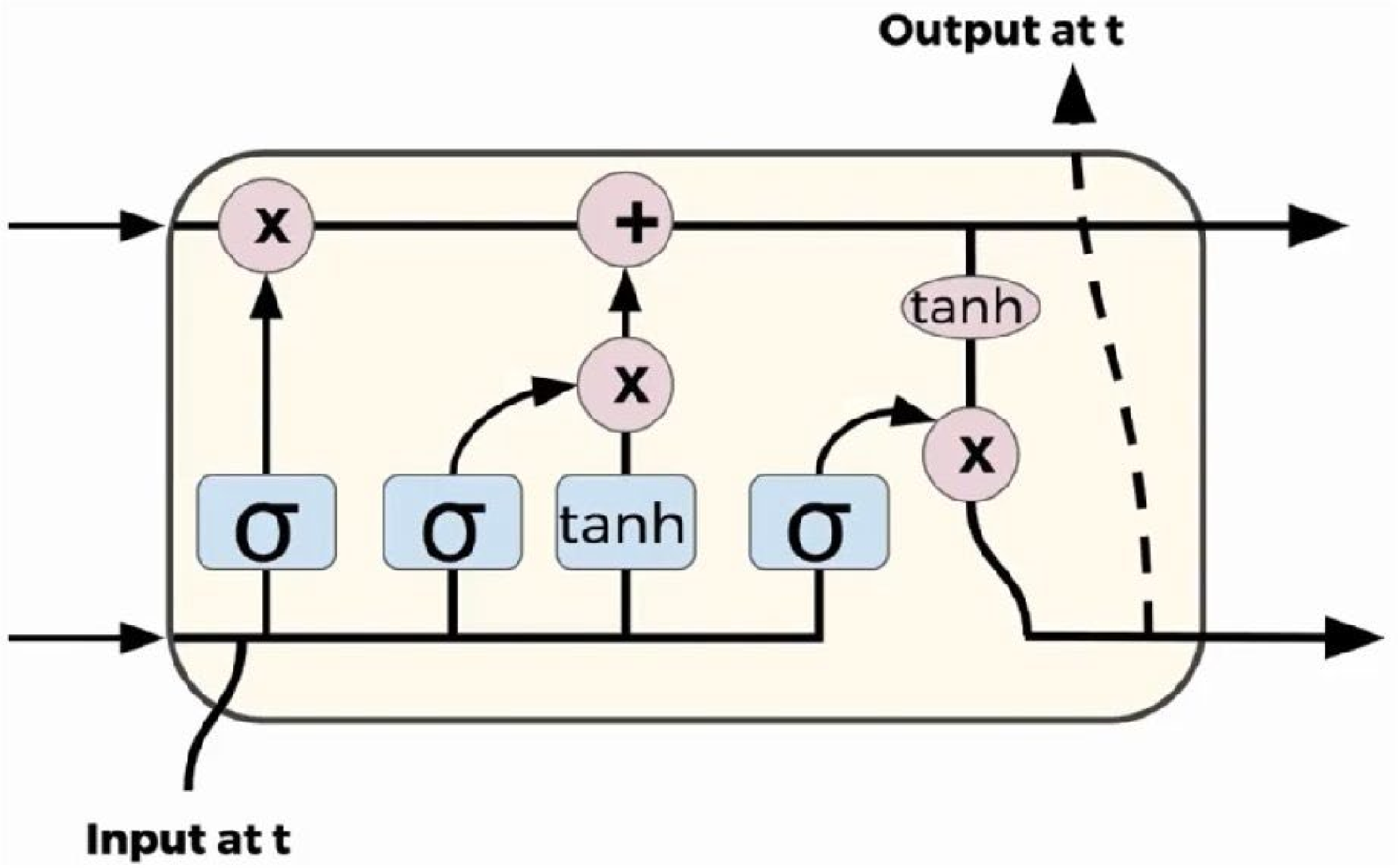
Recurrent neural networks, also known as RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states. They are typically as follows:



- RNN's are good at processing sequence data for prediction.

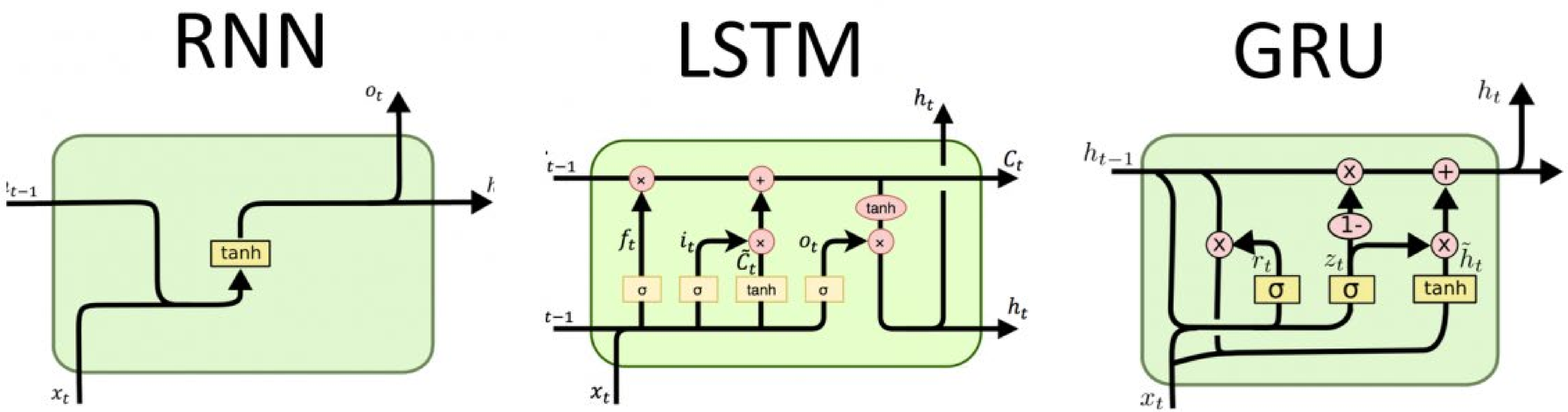
Long short-term memory

Long Short-Term Memory networks, also known as LSTMs, are a type of RNN, capable of learning long-term dependencies in sequence prediction problems.



Gated Recurrent Network

The GRU is the newer generation of Recurrent Neural networks. Just like LSTM, GRU uses gates to control the flow of information.



Transfer Learning

If the classes we want to recognize are not among the labels of the pre-trained model, we can usually **retrain** the pre-trained neural network model to recognize the new classes, by using most of the weights and connections of the pre-trained model.