Image Recognition

Xccelerate - Data Science Immersive

Agenda

What is Deep Learning?

Structure of a Artificial Neural Network

The Activation Function

Back Propagation concept

Evaluate, Improve and Tune your ANN

History of AI / Deep Learning

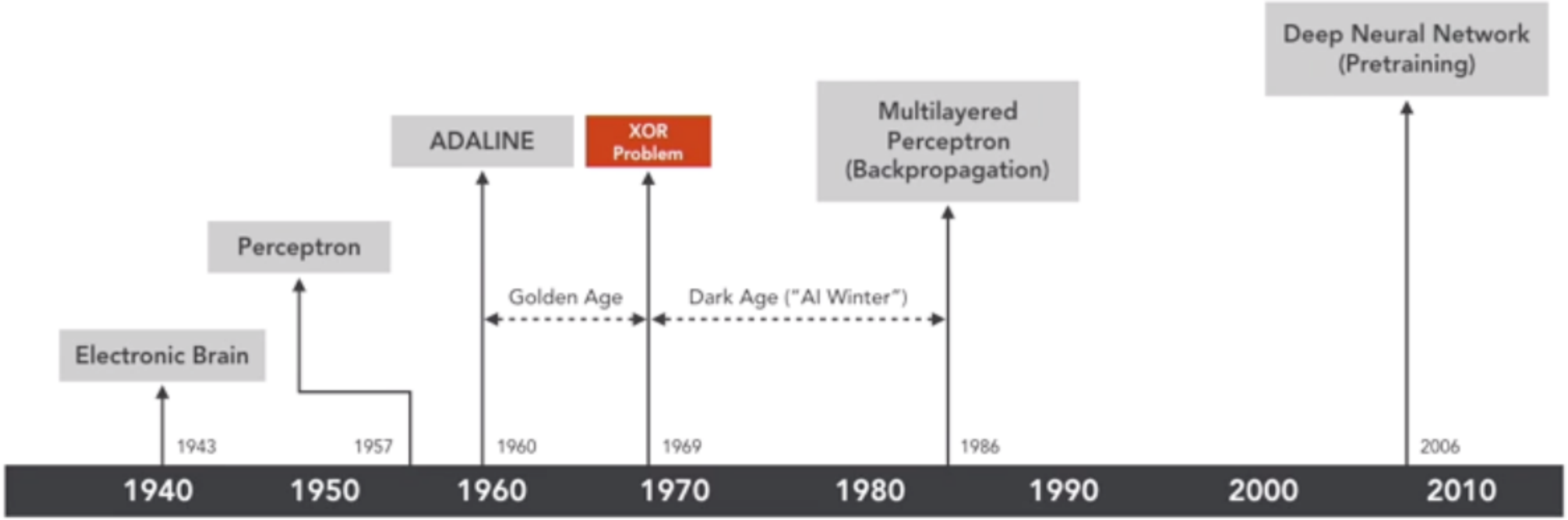
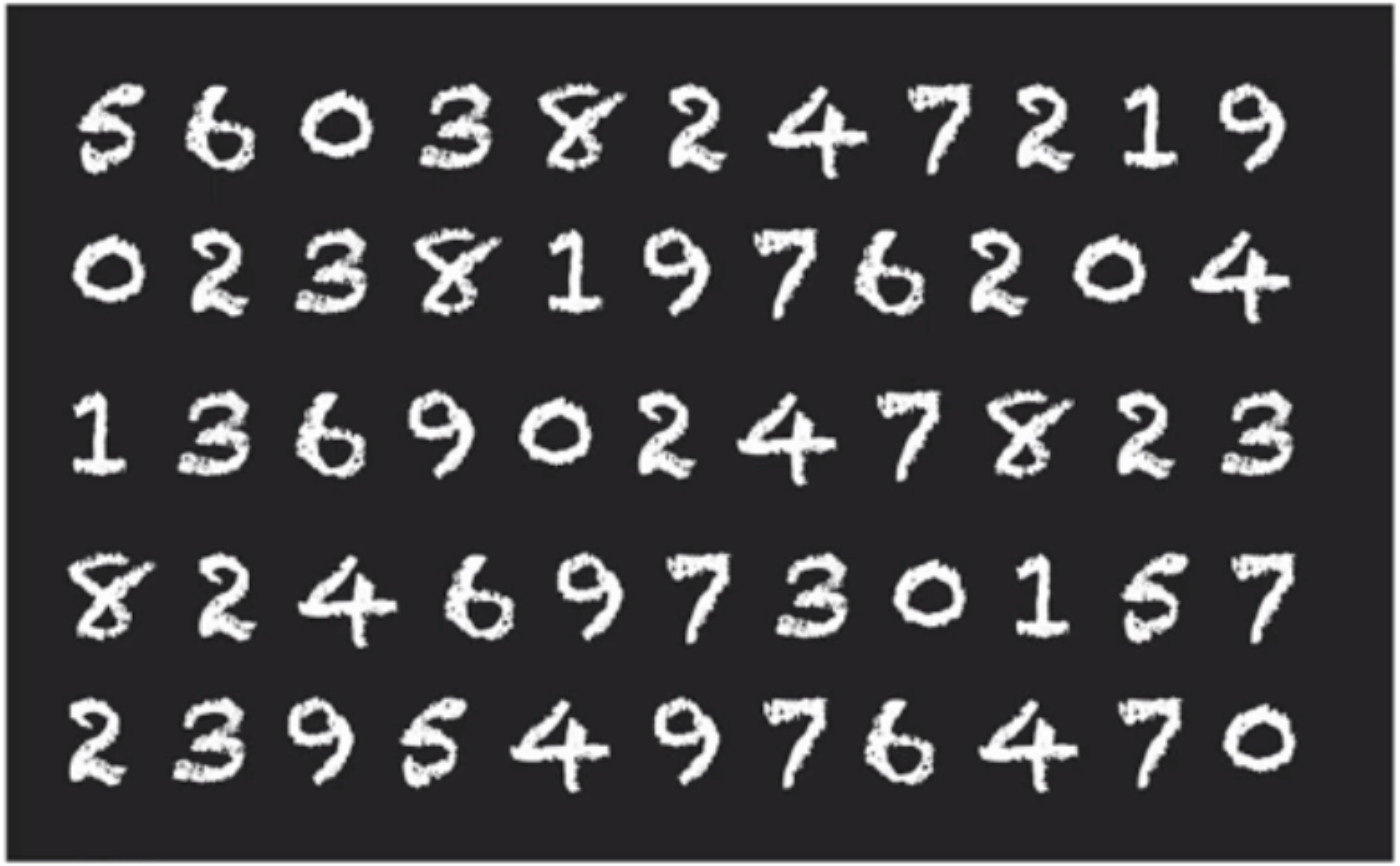
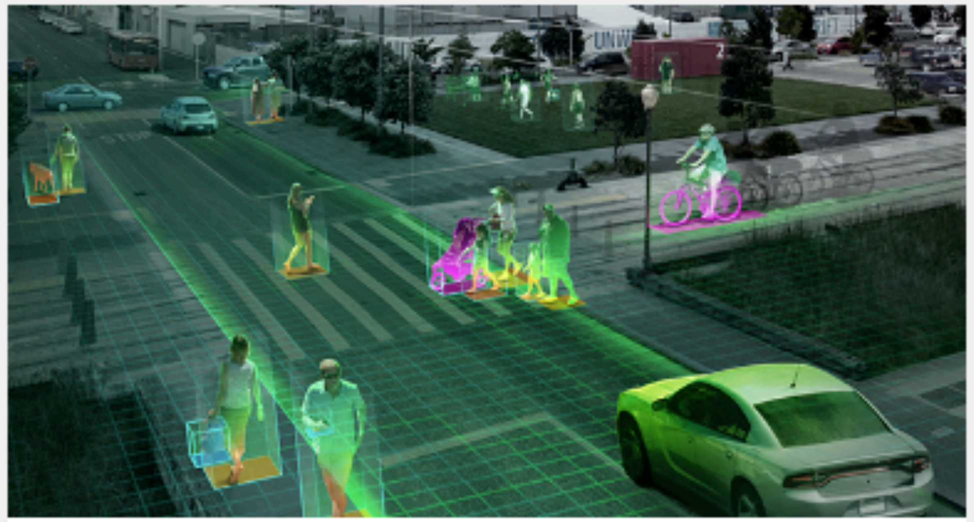


Image recognition

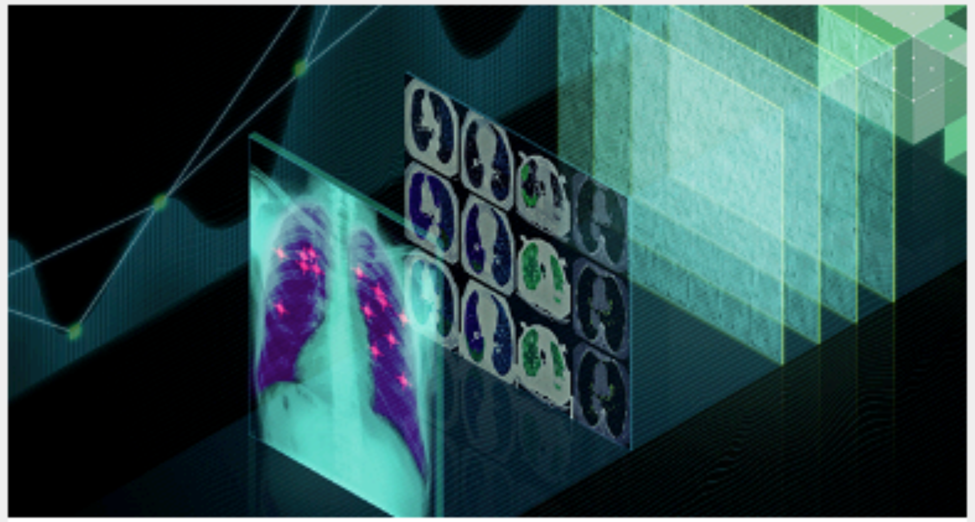


DEEP LEARNING IS FUELING ALL AREAS OF BUSINESS



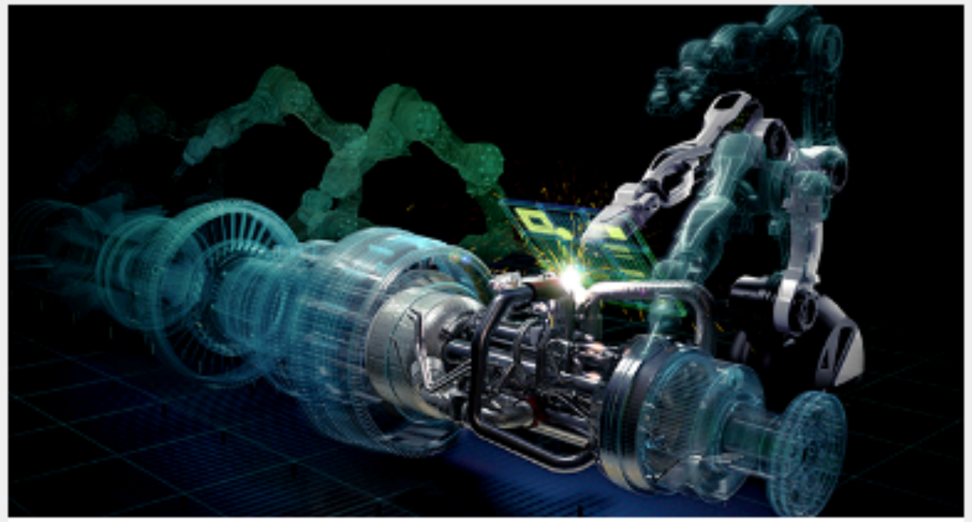
AI Cities

AI for Public Good

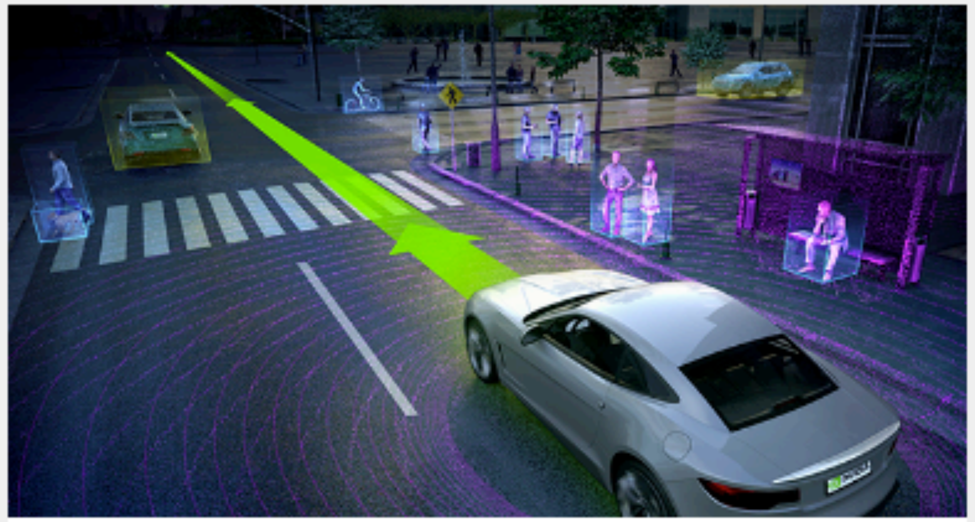


Healthcare

Retail



Robotics

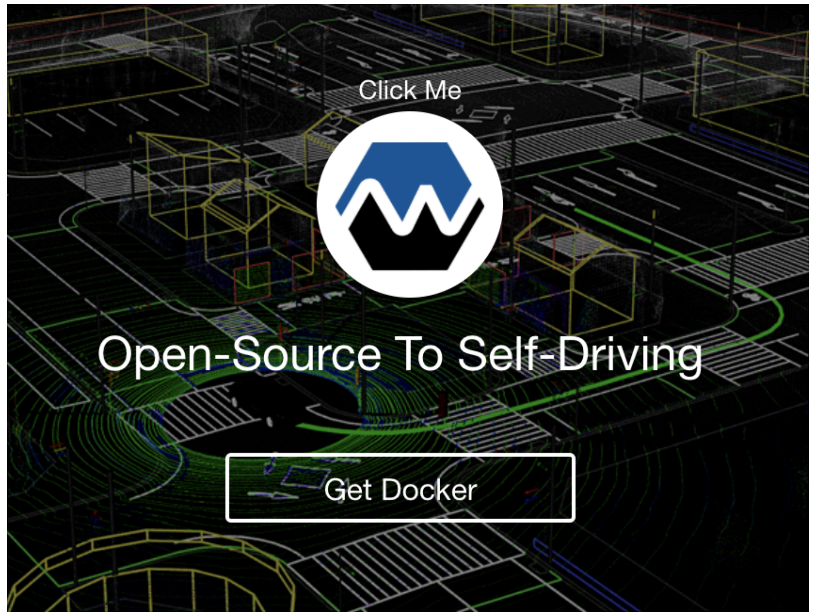


Self-Driving Cars

Self-driving cars

Question:

What kind of tasks (object detection for instance) are involved in self-driving cars?

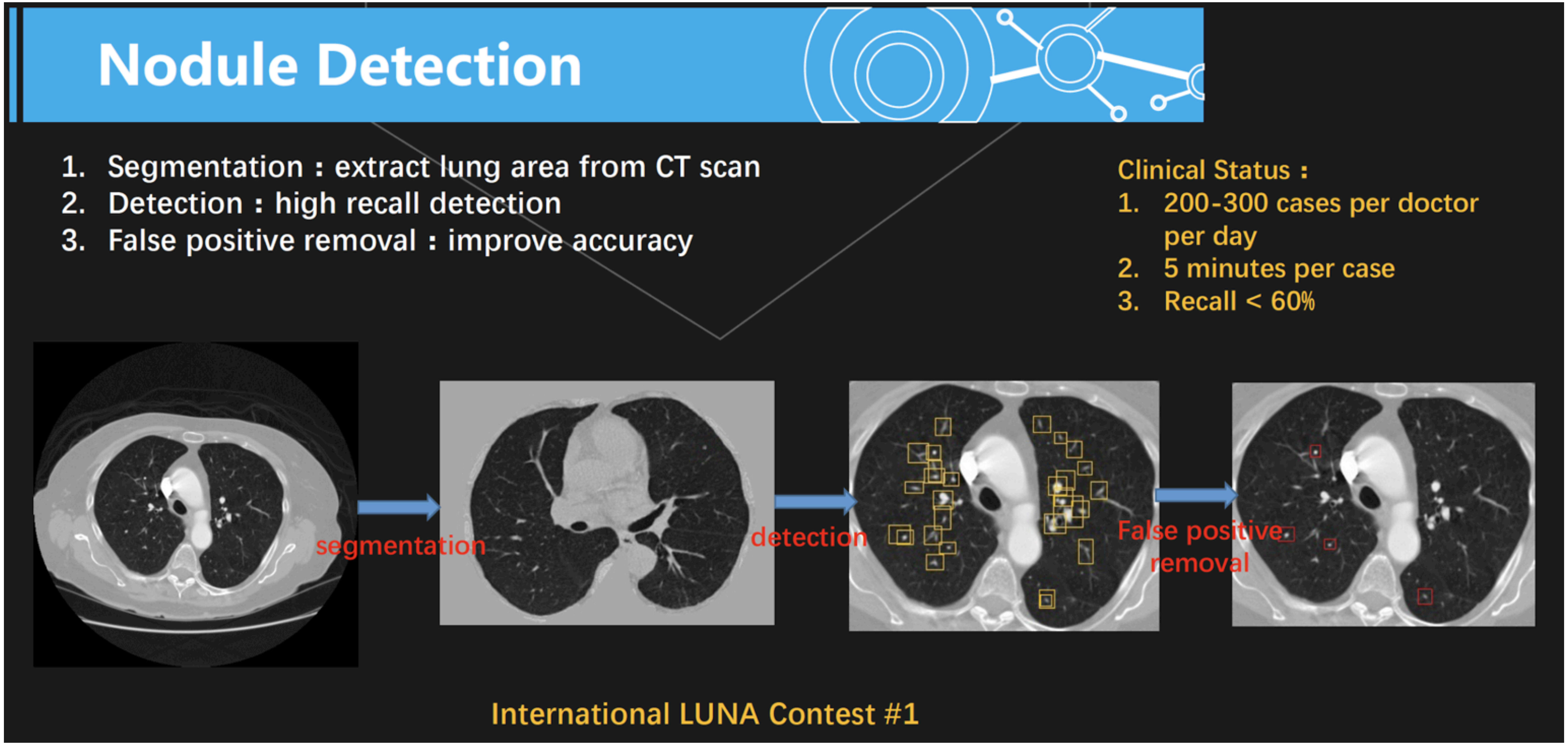


Source: https://www.autoware.ai/

Deep learning in the wild:

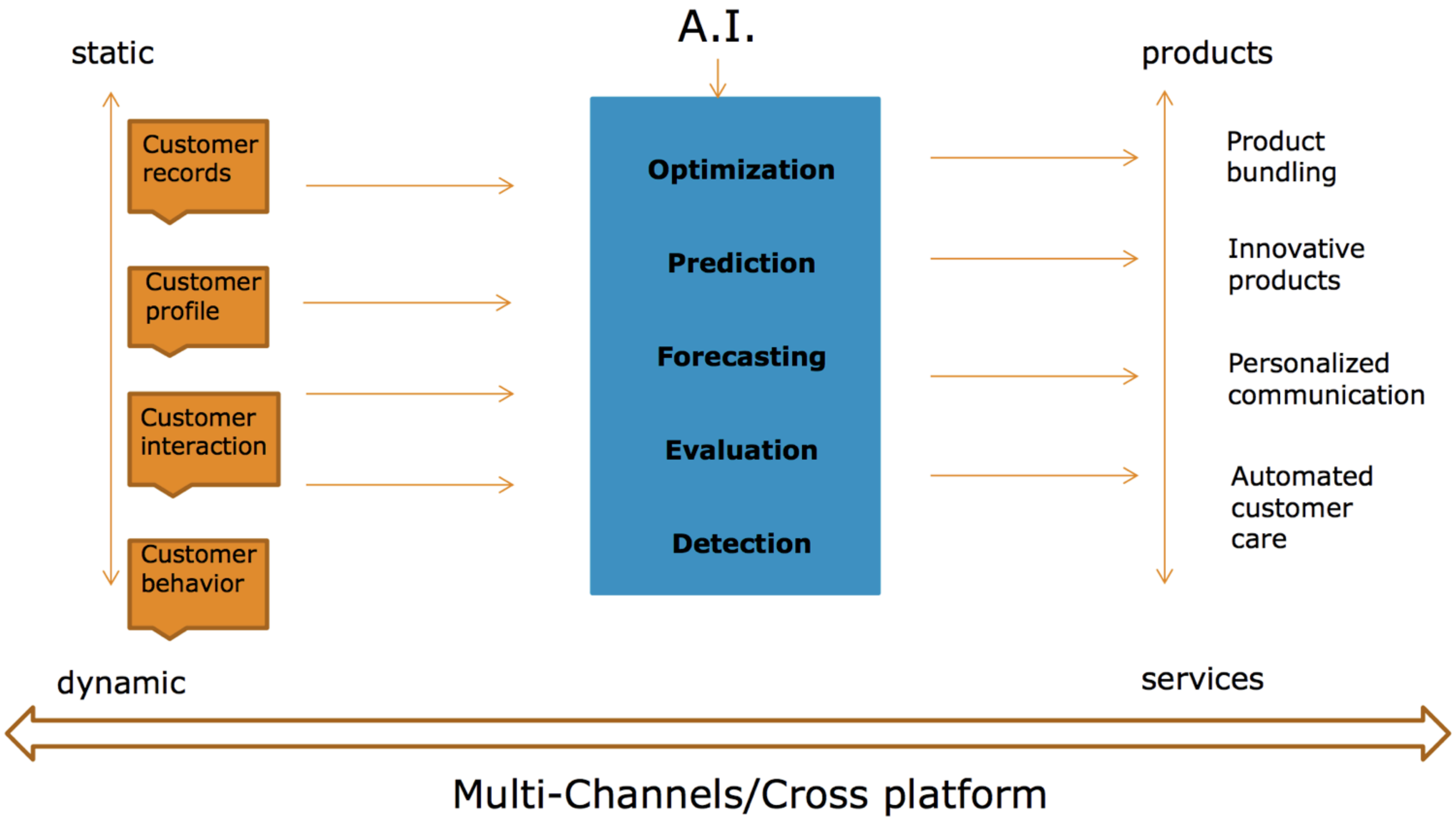
Healthcare

At GTC, Taking the Pulse of How AI is Transforming Healthcare



Deep learning in the wild:

AI in Finance



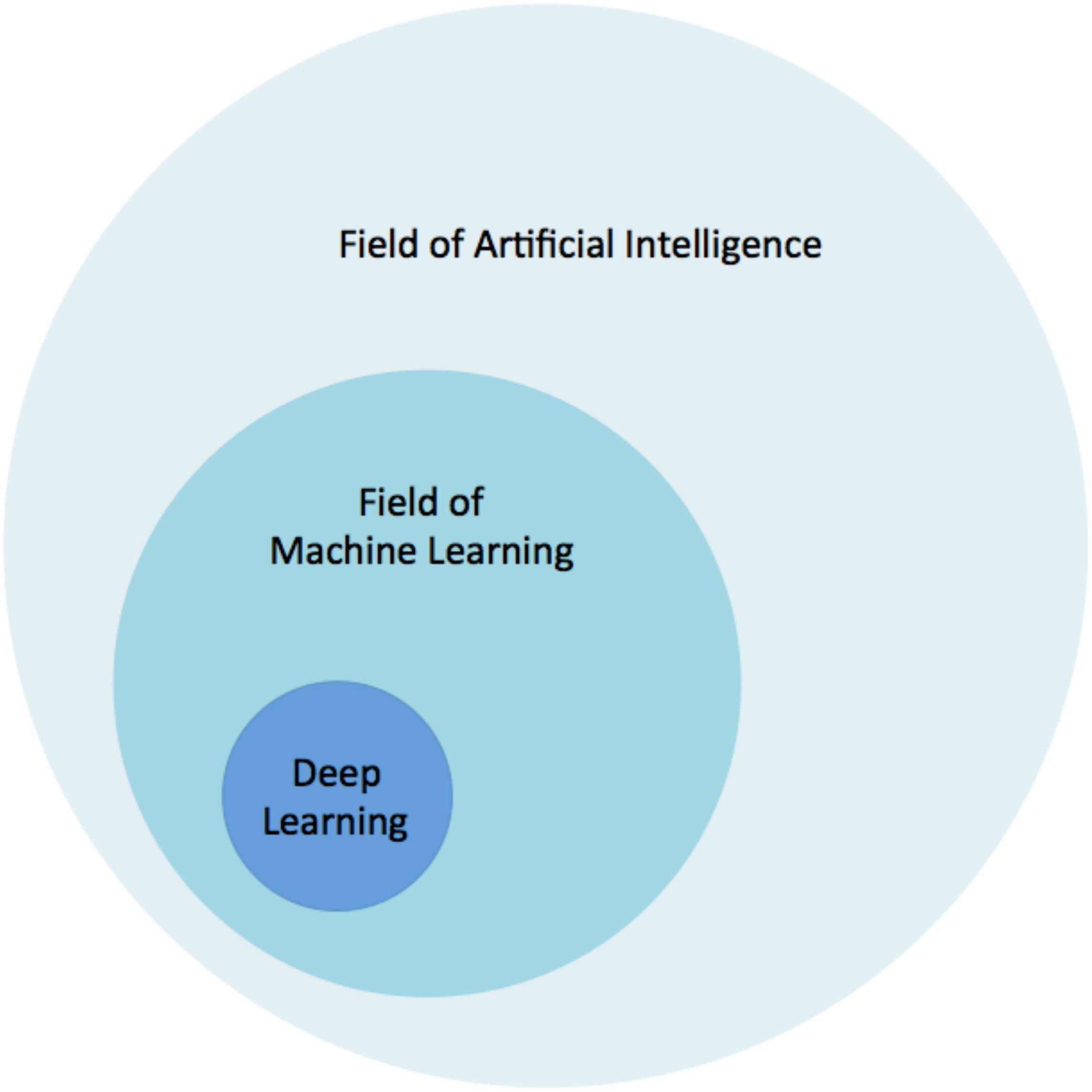
Introduction

Deep learning is a specific subset of Machine Learning, which is a specific subset of Artificial Intelligence. For individual definitions:

+ Artificial Intelligence is the broad mandate of creating machines that can think intelligently

+ Machine Learning *is one way of doing that*, by using algorithms to glean insights from data (see our gentle introduction here)

+ Deep Learning *is one way of doing that*, using a specific algorithm called a **Neural Network** (convolutional/recurrent neural networks, capsules, echo state network, etc).



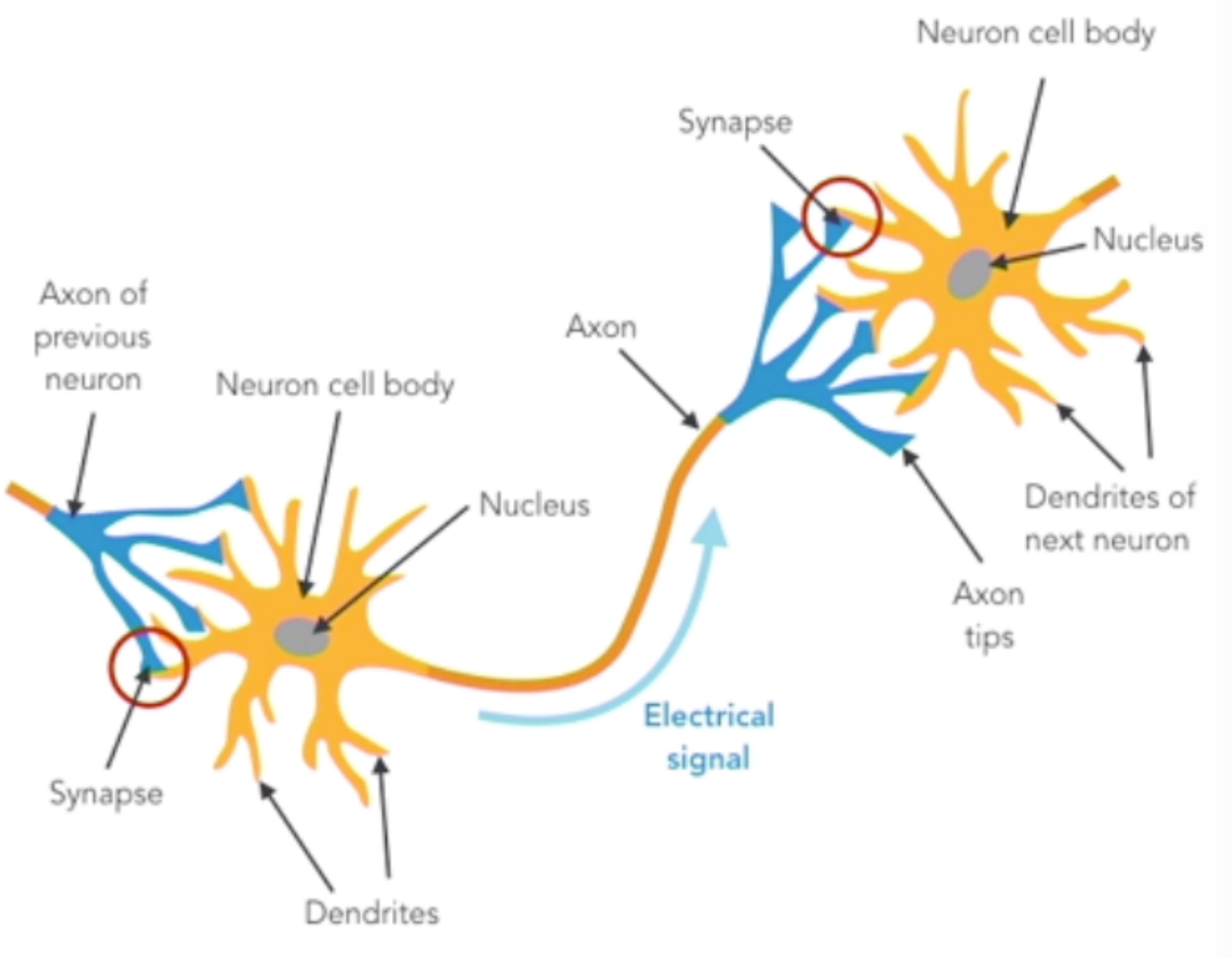
Biological Neurons

Neuron is the basic building block of the brain.

A specialised cell designed to transmit

Neuron + Axon(transmitters) and Denrites (receivers).

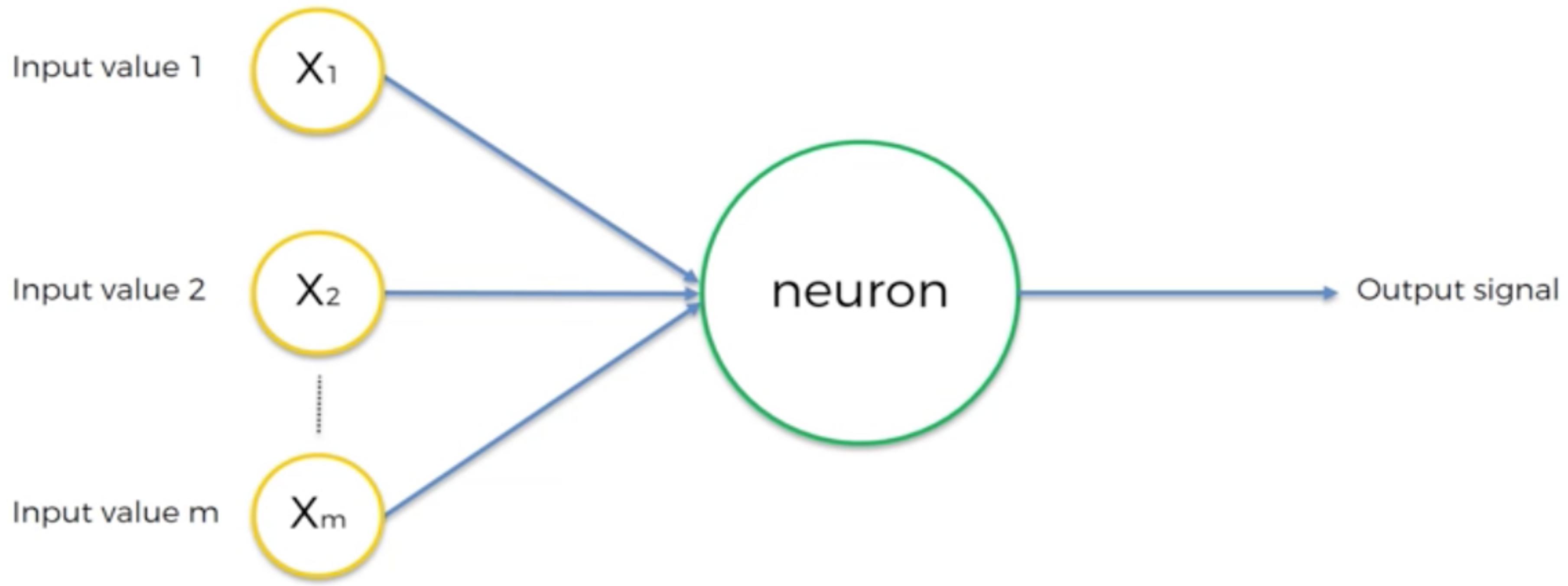
The signal or connection is called synapse.



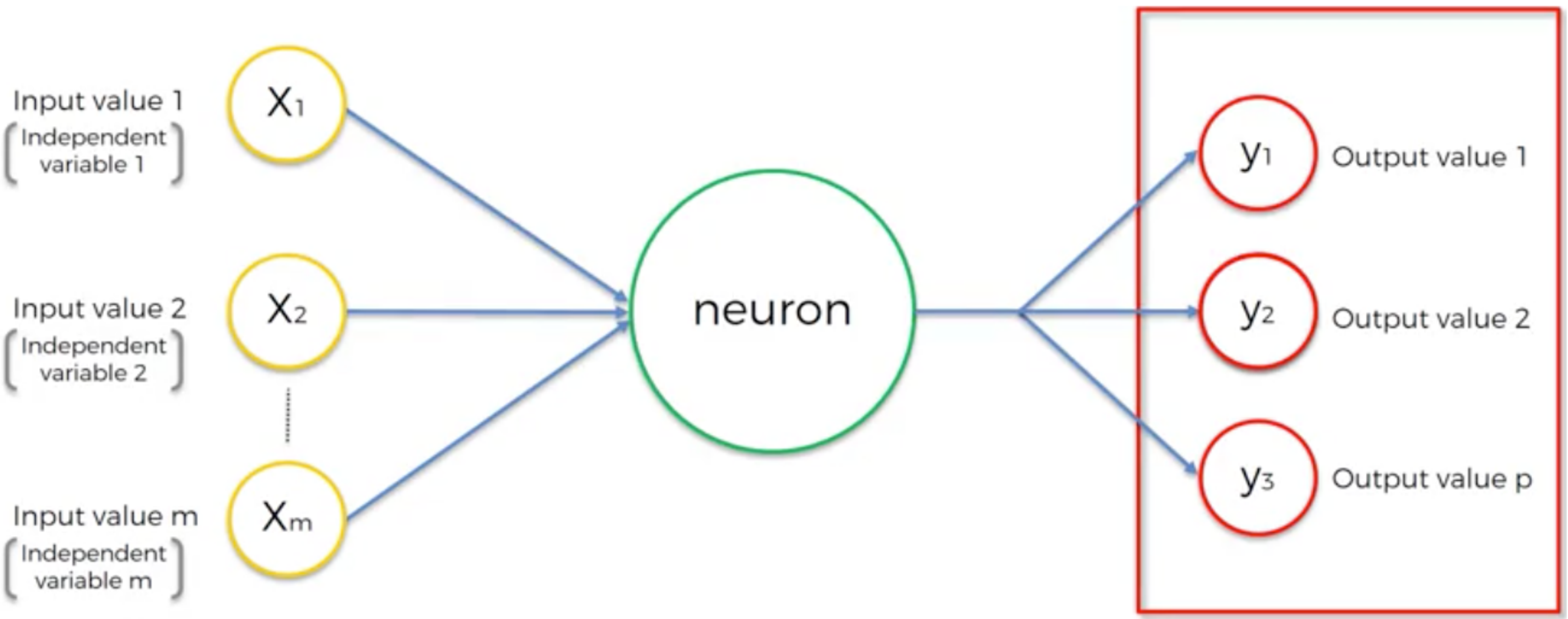
Artificial Neuron –

Inspired by the structure of the cerebral cortex.

1957 – Proposed by Rosenblat



Artificial Neuron



ACCELERATE

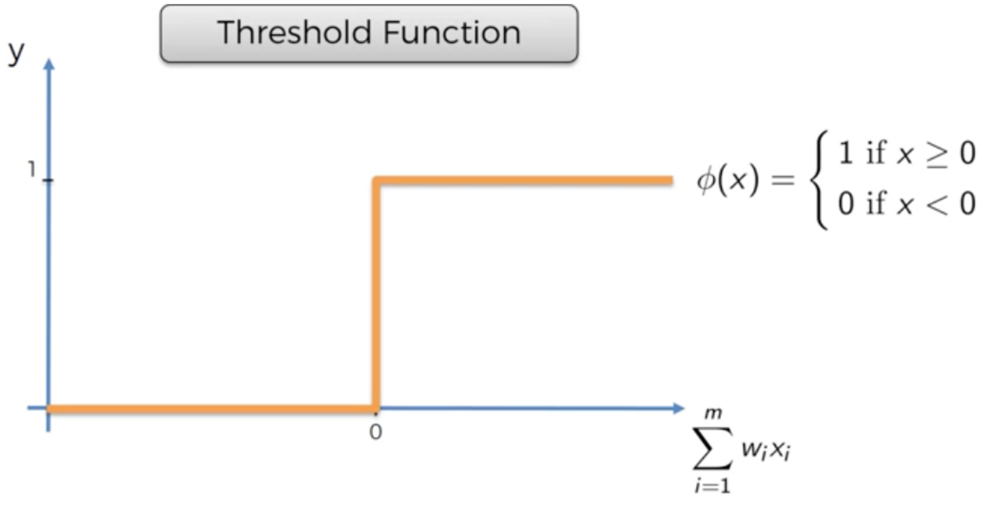
Activation functions

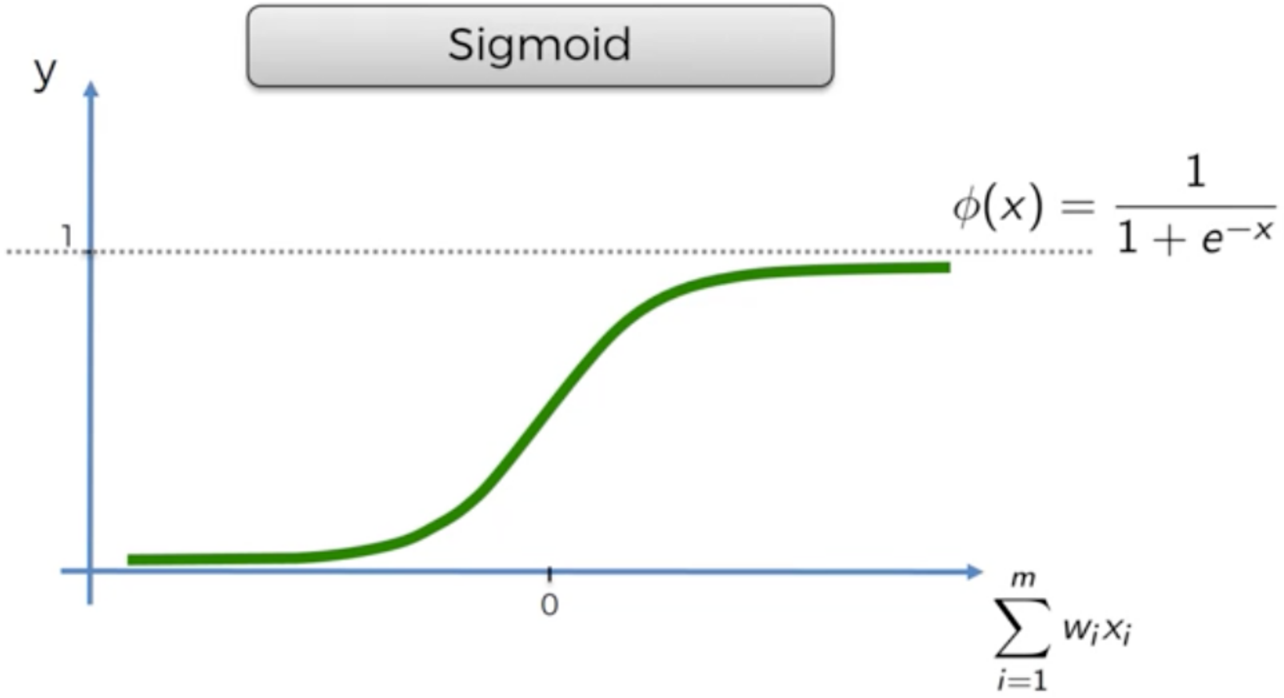
Activation functions transform the weighted sum of inputs that goes into the artificial neurons. These functions should be non-linear to encode complex patterns of the data

Activation functions

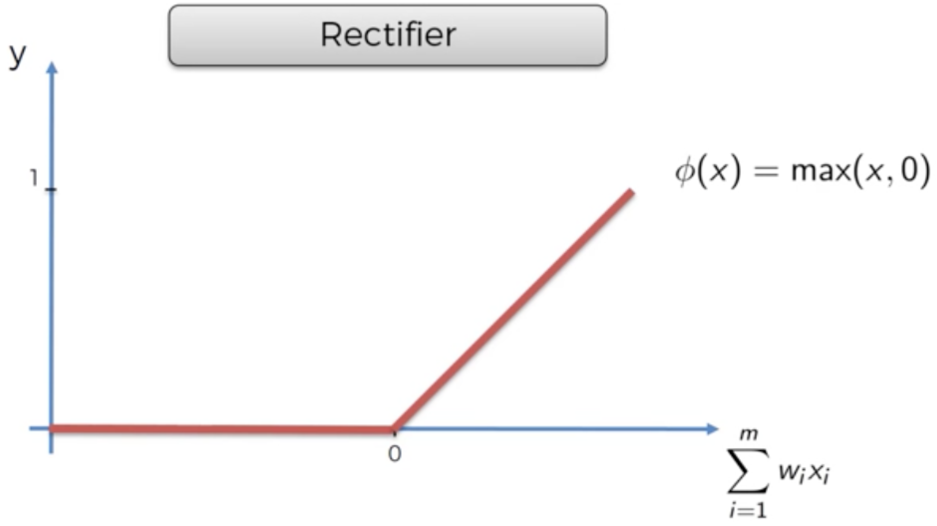
The sigmoid function -

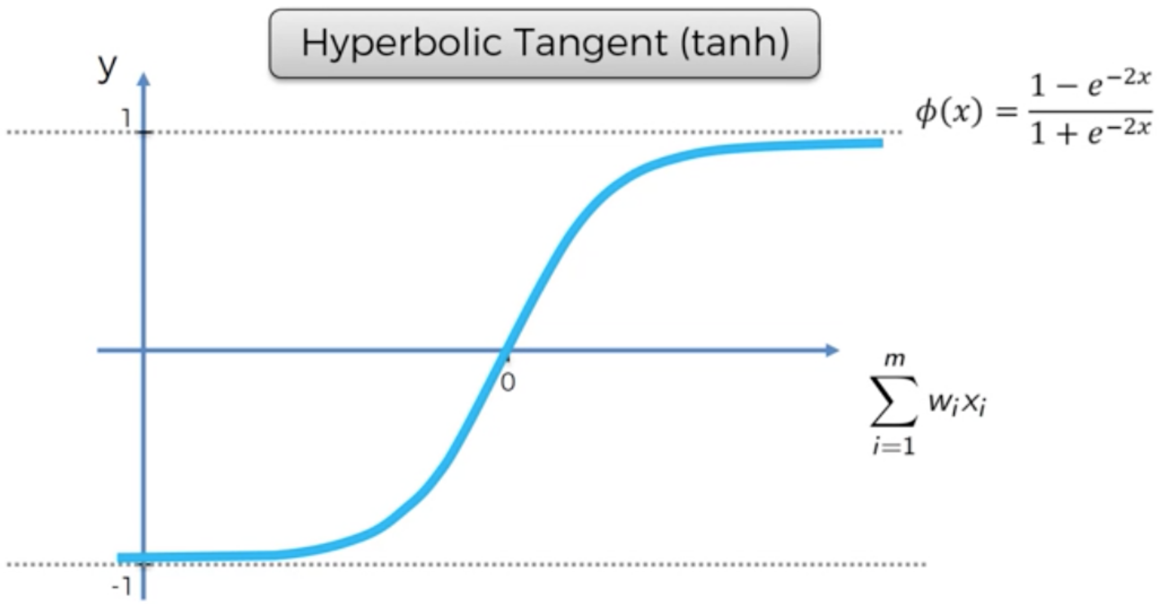
Threshold function





Rectifier – (RELU

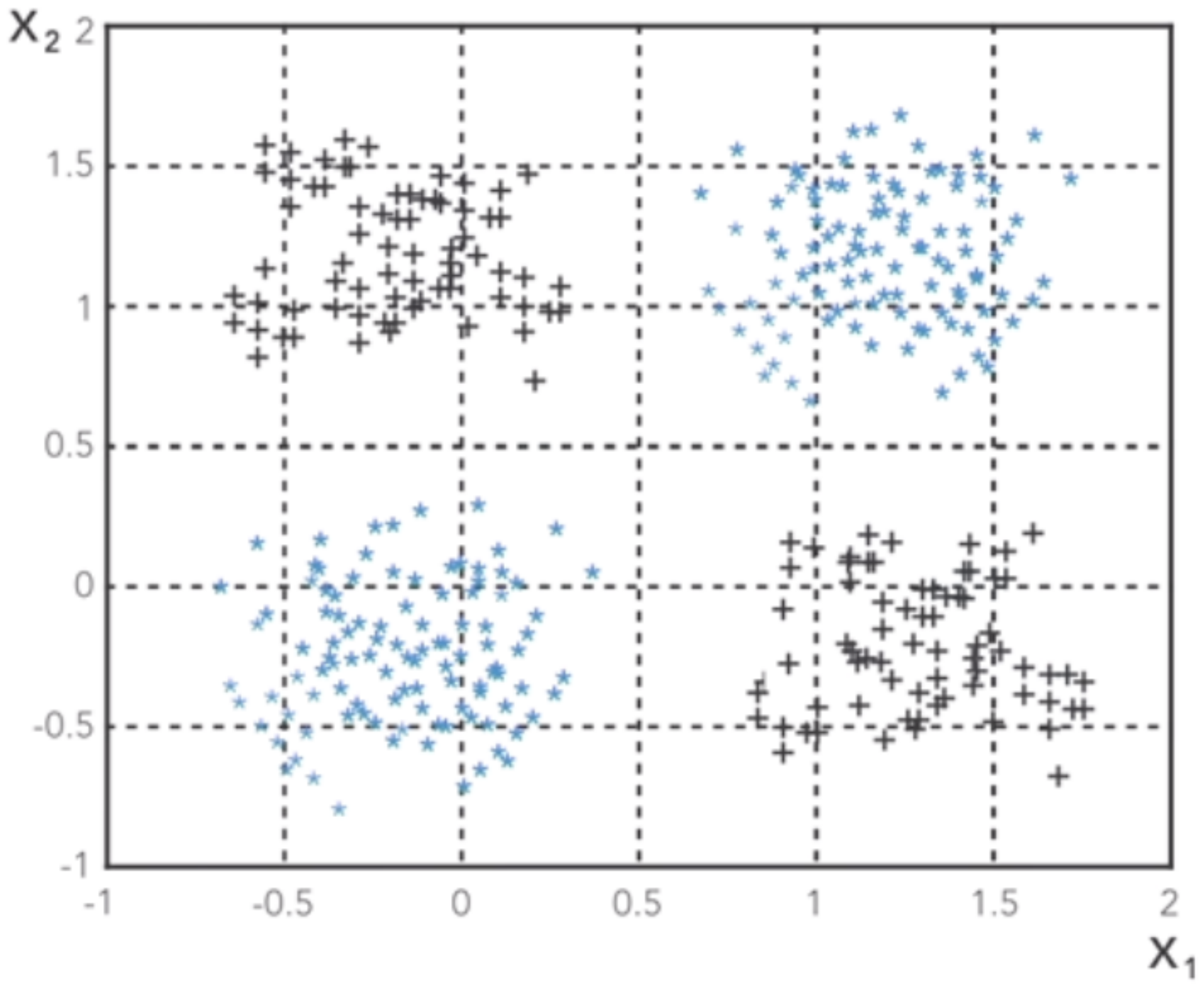




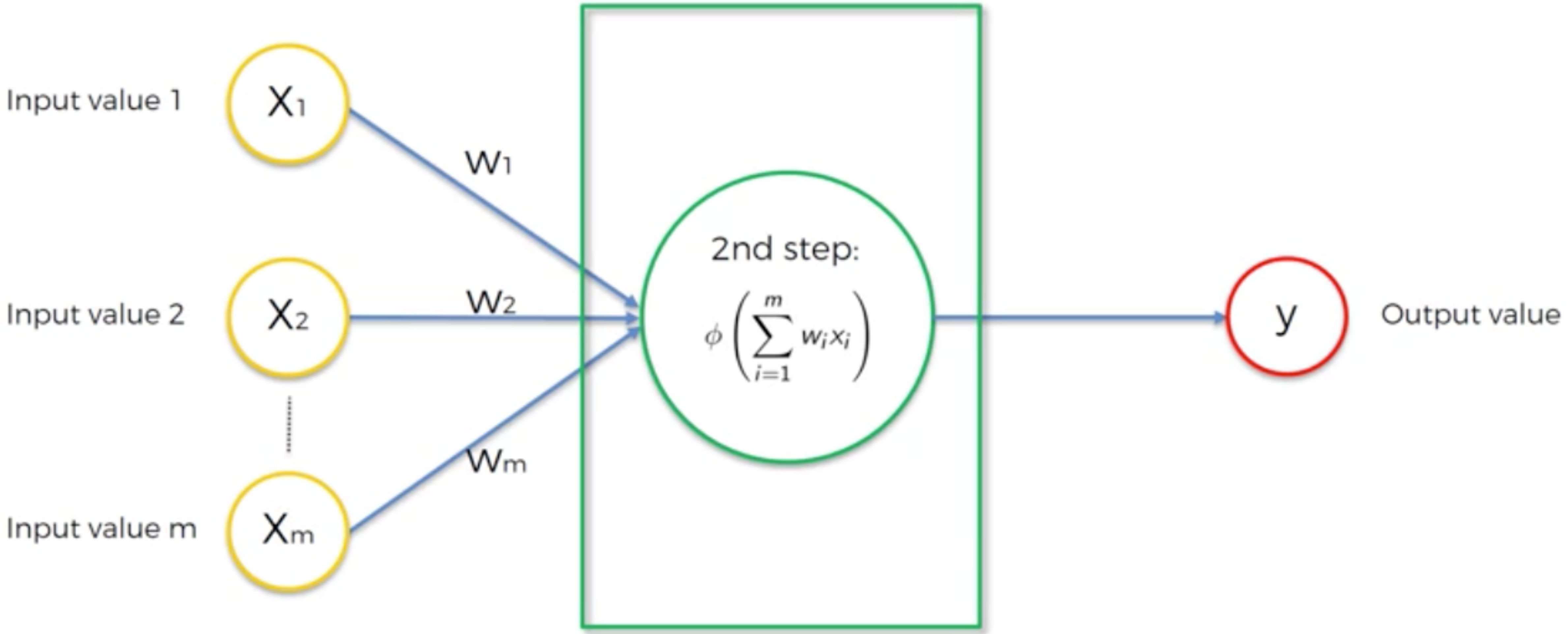
Deep sparse rectifier NN : Xavier Glorot (2011)

TanH – Hyperbolic tangent

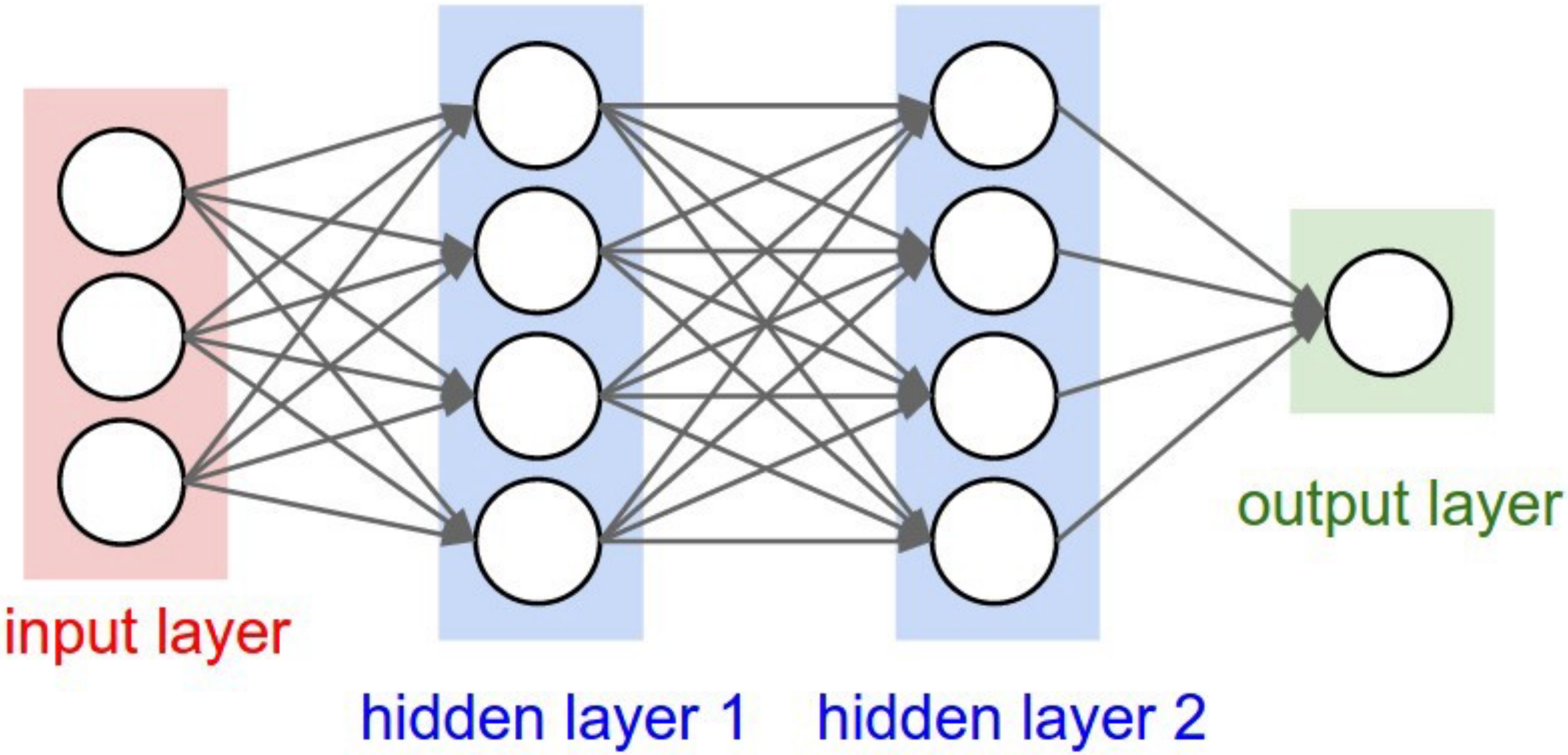
Multi layers and non-linearity help separate data classes like these



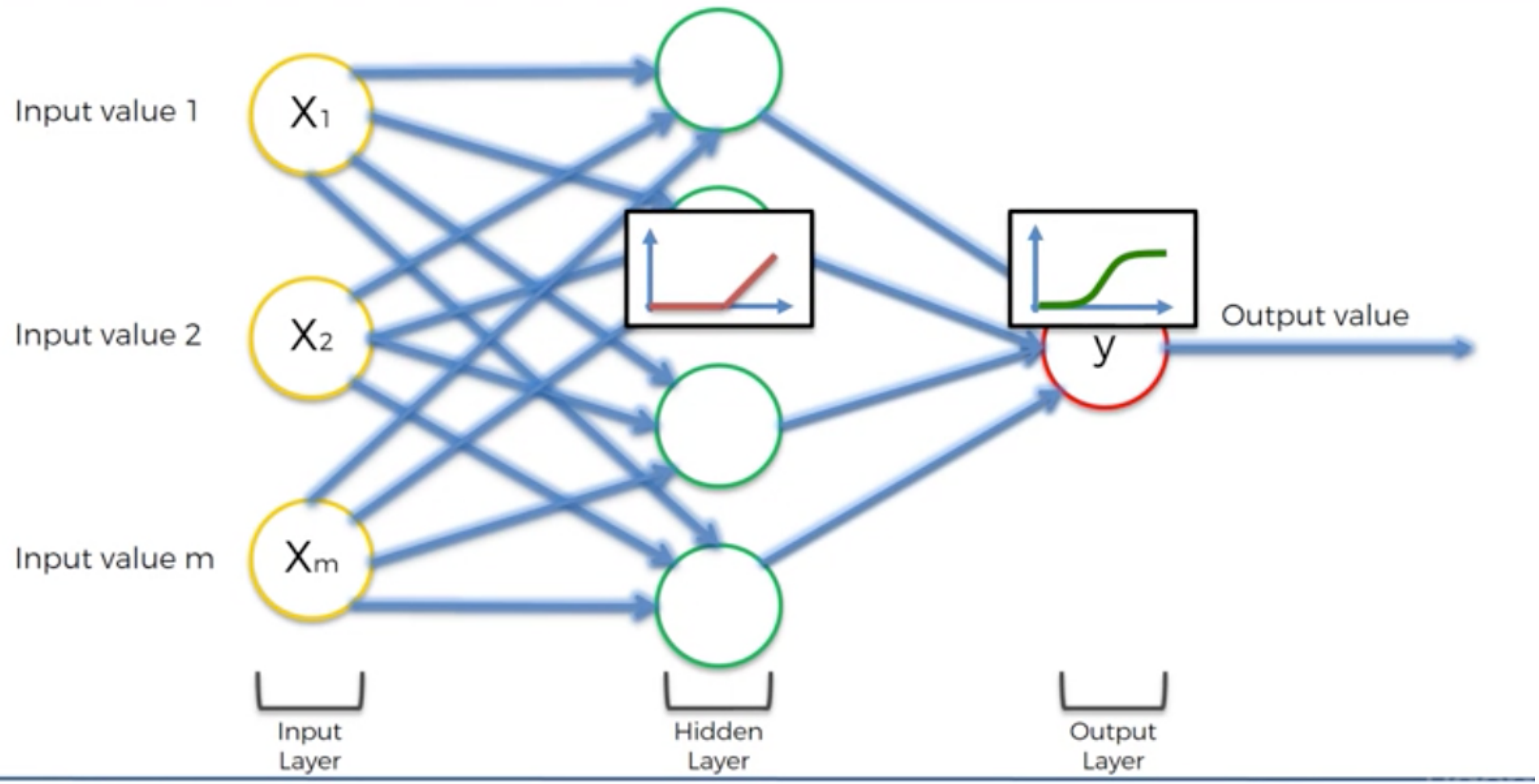
Artificial Neural network with single layer – Perceptron



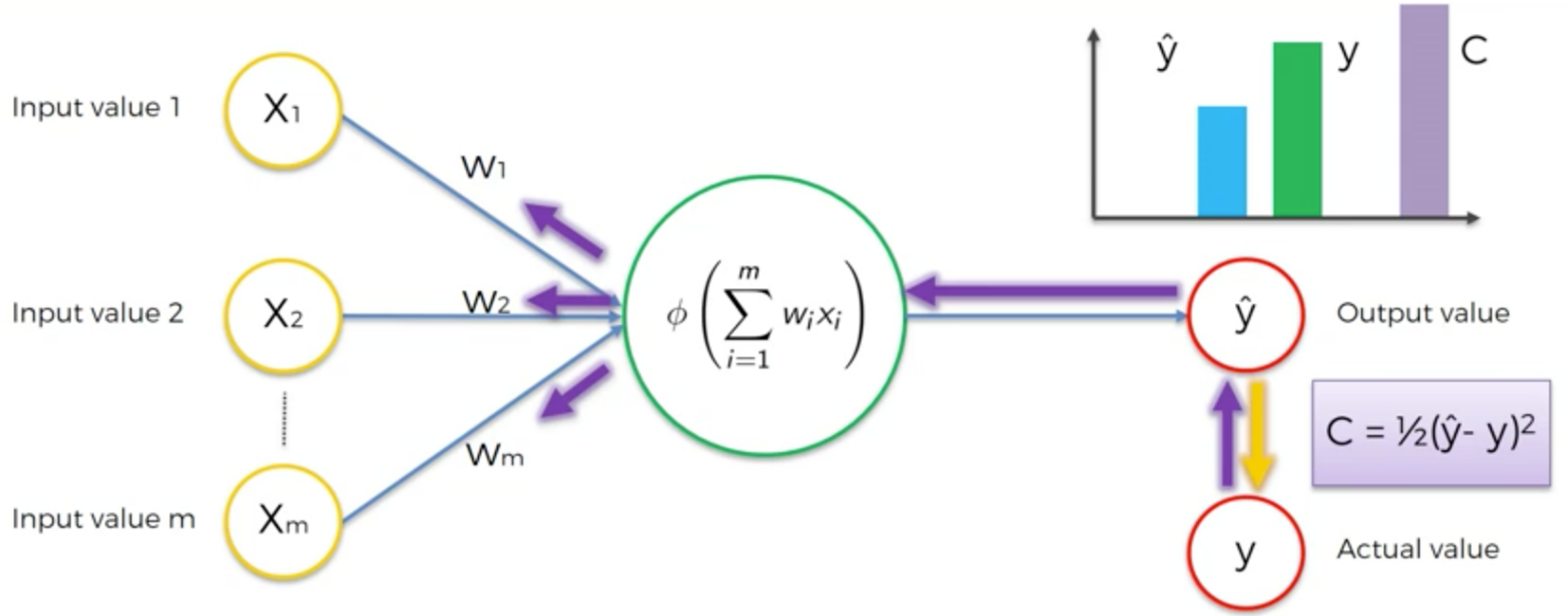
Sample Artificial Neural Network



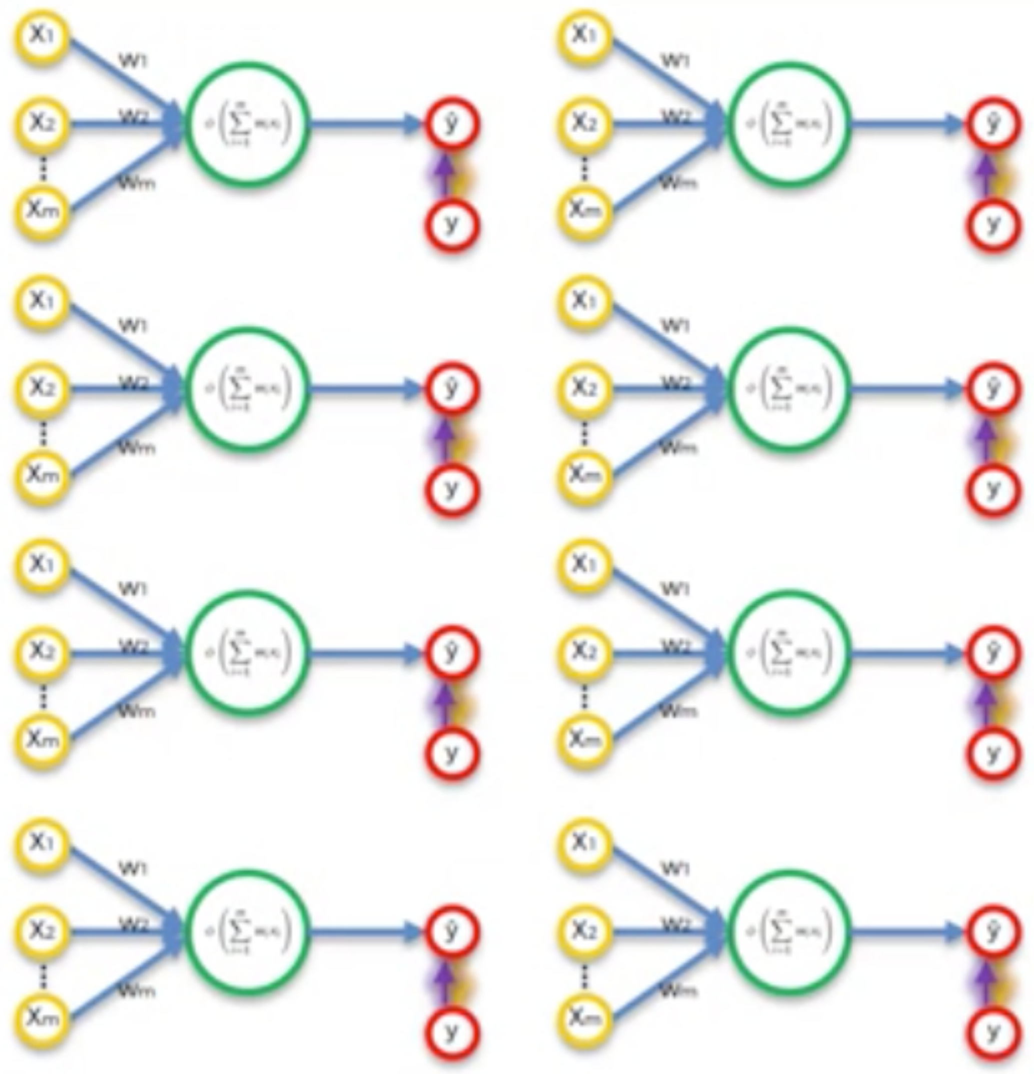
Diff Activation functions at diff layers can be applied –



How do Neural networks Learn?

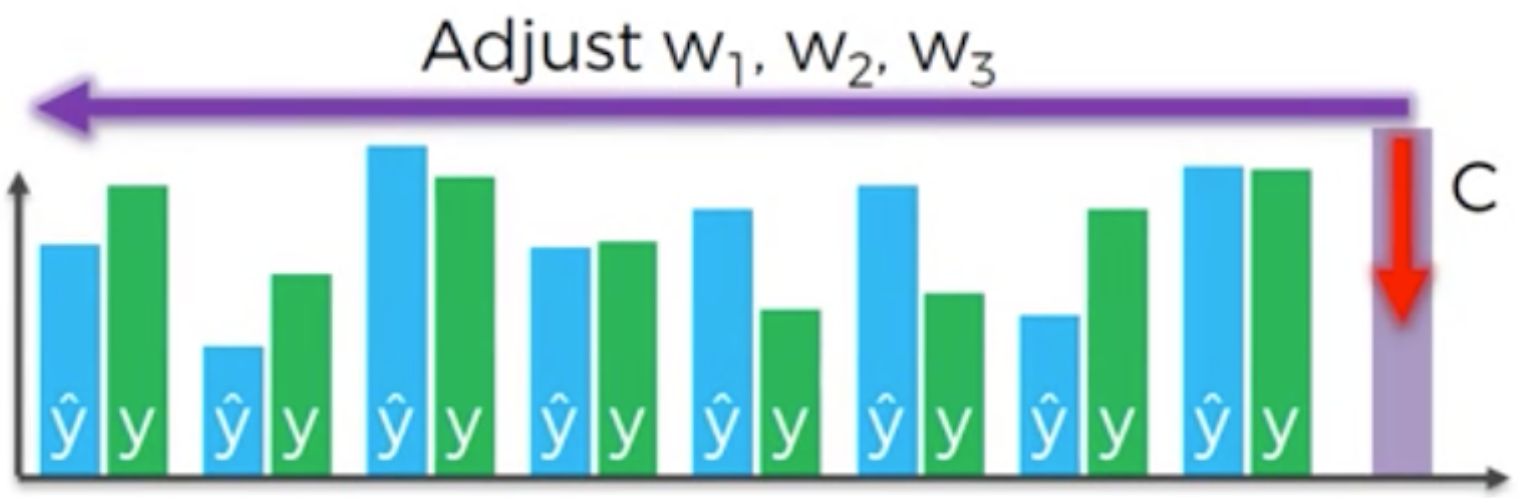


How do Neural networks Learn?

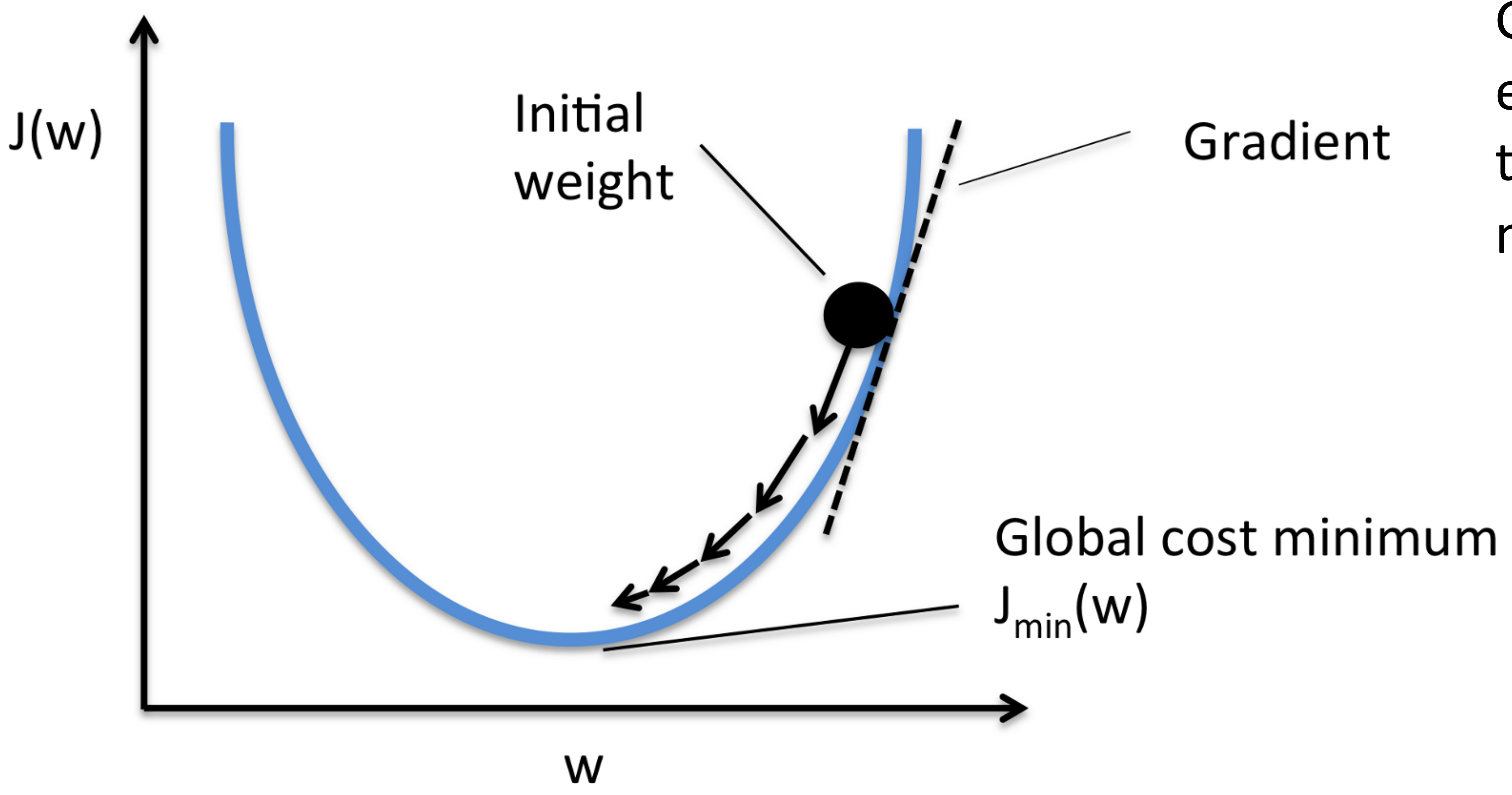


Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

$$C = \sum \tfrac{1}{2}(\hat{y} - y)^2$$

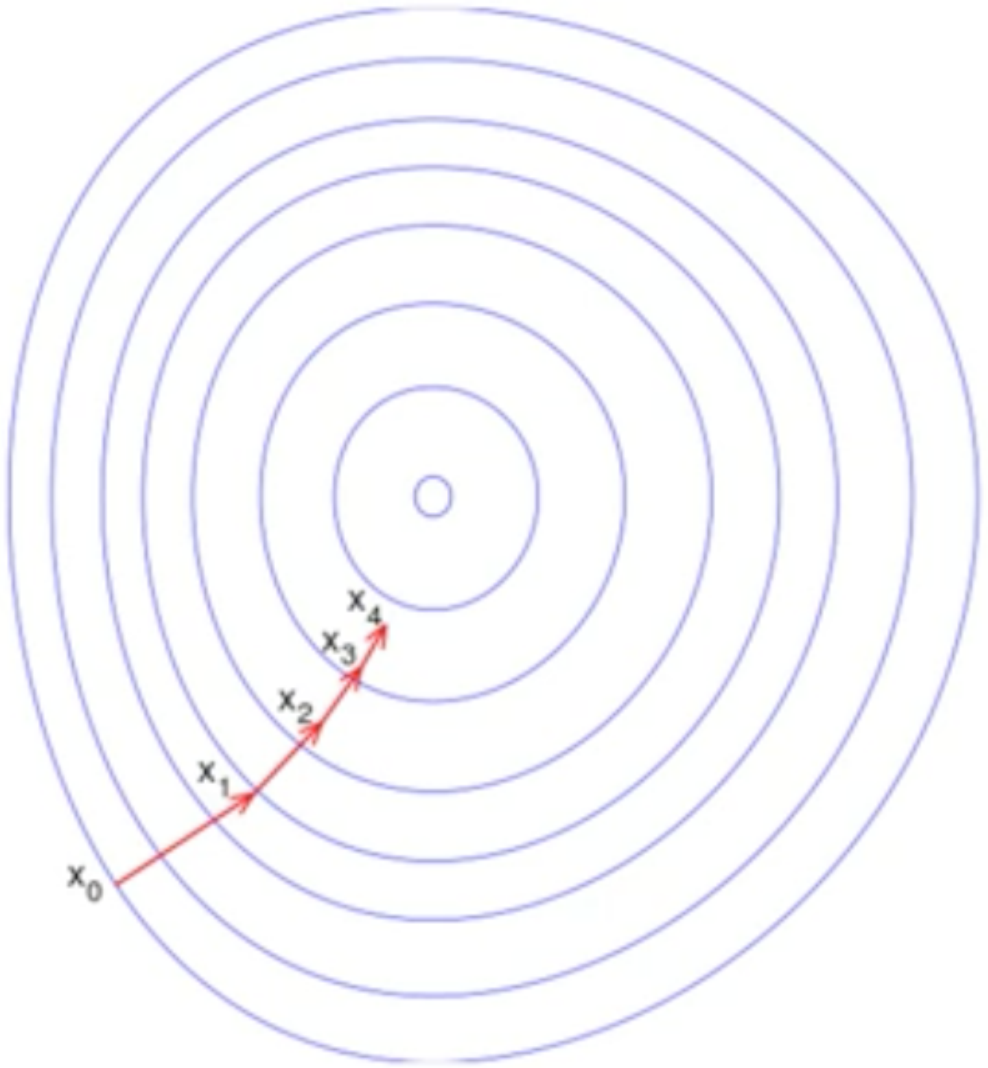


Gradient Descent



Gradient descent is an efficient method for solving the optmisation problem of min the cost function..

Gradient Descent – 2D space



Gradient Descent

Batch gradient descent will adjust weights after fill batch.

Schotastic gradient descent will adjust weights after each row.
SGD helps you avoid the local minimum - can handle much better fluctuations. Most likely to find the global minimal than the local minima.

Mini-batch small batches of data for handling - in between.

https://iamtrask.github.io/2015/07/27/python-network-part2/
http://neuralnetworksanddeeplearning.com/

Back Propagation

Step 1 - Randomise the weights to small numbers - close to 0
Step 2 - insert one observation as input each feature in one input node
Step 3 - forward prop from left to right . Propogate activations until you predict.
Step 4 - Compare predicted result to actual result. Measure error
Step 5 - Back propagation to right to left. Error is back propogated. Update weights according to how much they are responsible for error. Learning rates decided how much we updates the rates
Step 6 - repeat steps 1-5 after each observation (SGD) or after a batch of observations. (batch learning)
When whole training set passed through an ANN - it makes an epoch.
Redo more epochs

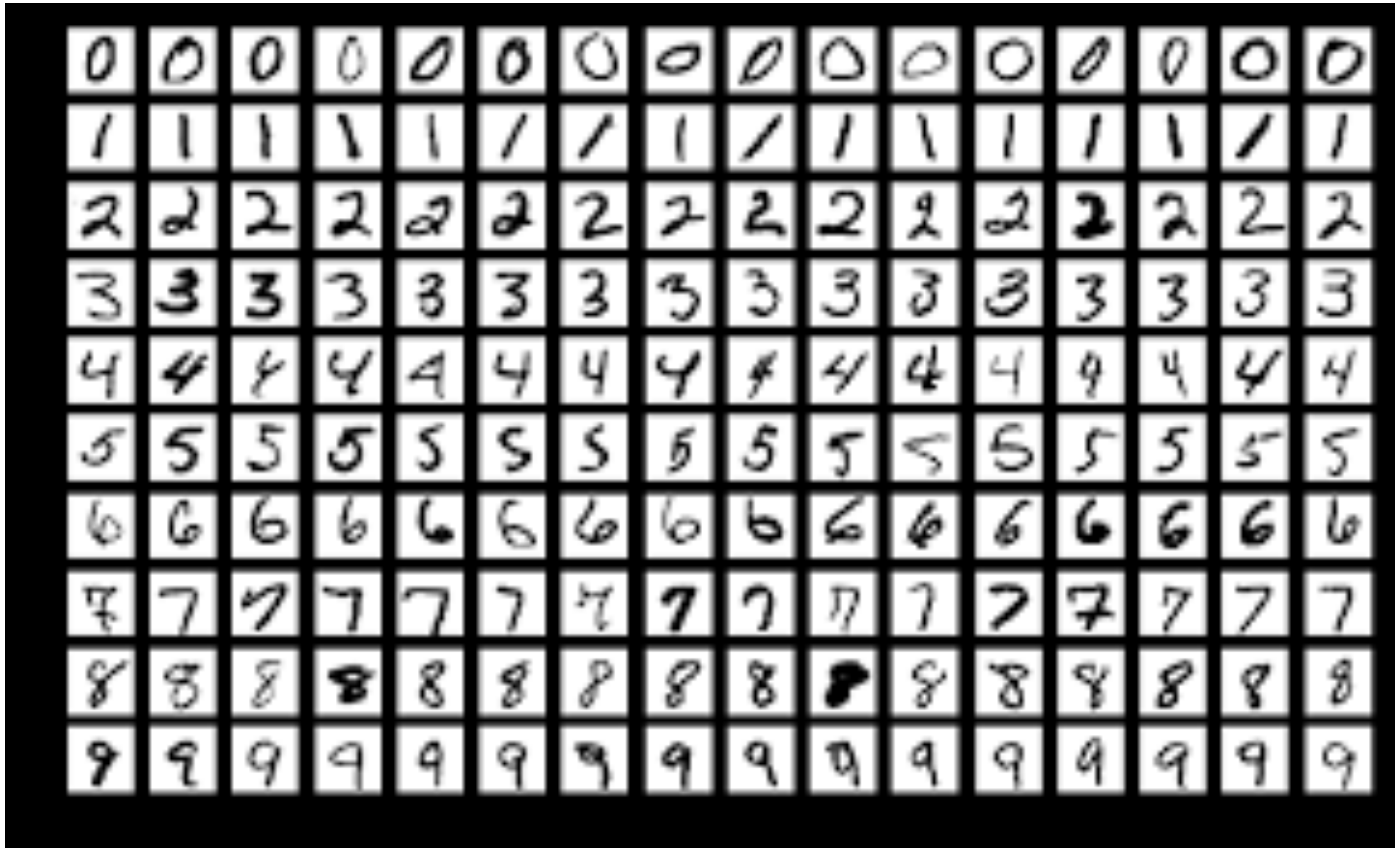
Tensorflow Visualisation – putting it all together

Neural Network in Keras

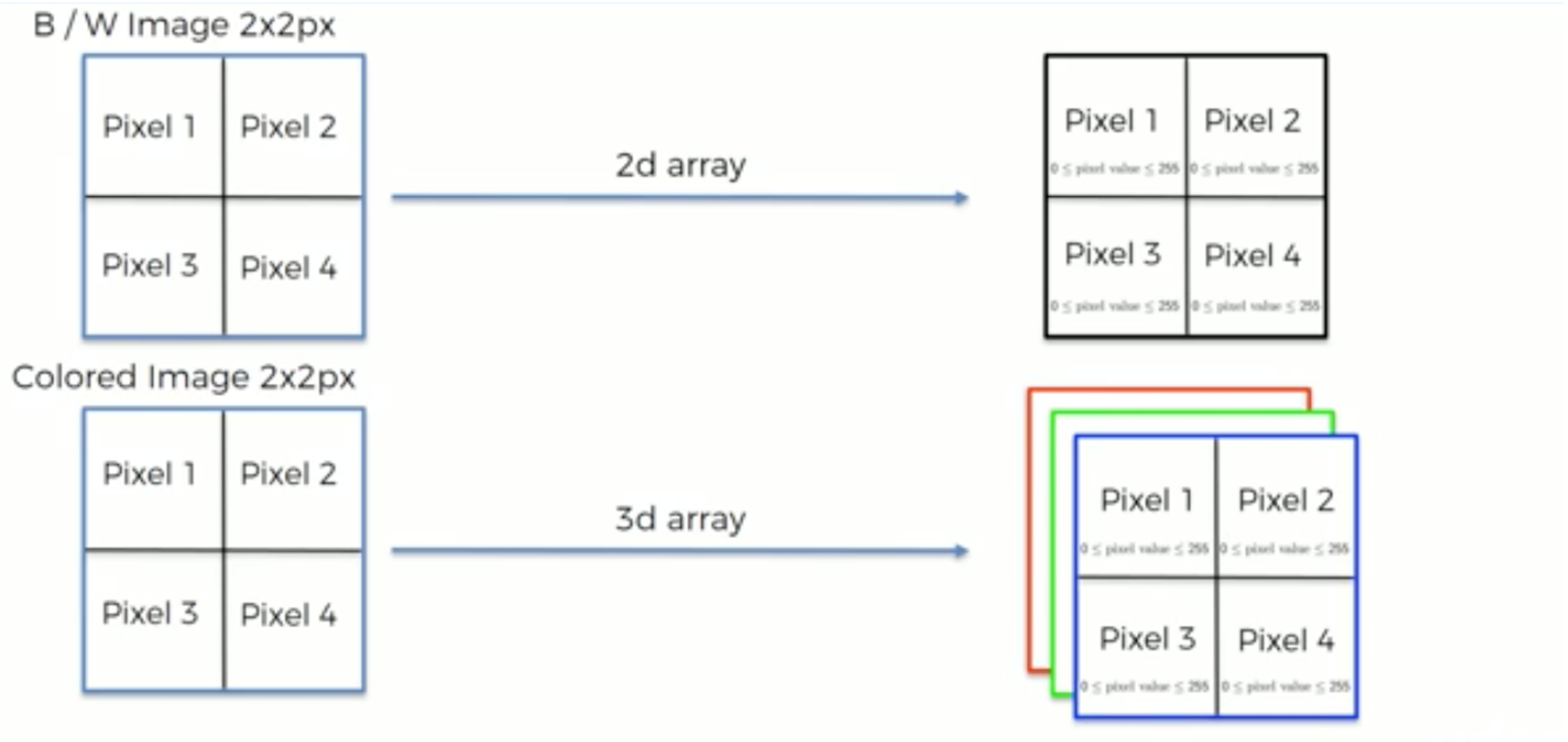
Keras (www.keras.io)
Popular library for deep learning – works with Tensorflow / Theano and CNTK backends

```
# create model
model = Sequential()
model.add(Dense(12, input_dim=8, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X, Y, epochs=150, batch_size=10)
```

MNIST database



What is an Image?



Use Keras and MNIST dataset to directly

Use 2 dense layers of 512

One output layer with 10 output classes.

Train the network and analyse the results.

Challenges with Images

Images can be larger in size,

Neural networks don’t scale well with larger images

Too many parameters and computing power

Small color image = 200*200*3 = 120,000 weights

Obviously we need something special to deal with images

Challenges with Images

- What makes one image different from the other
- Areas in image within a cluster may provide feature importance
- Object of interest could be anywhere within the whole image
- Obviously we need something special to deal with images!

Convolutional Neural Networks

What are convolutional neural networks?

Convolution

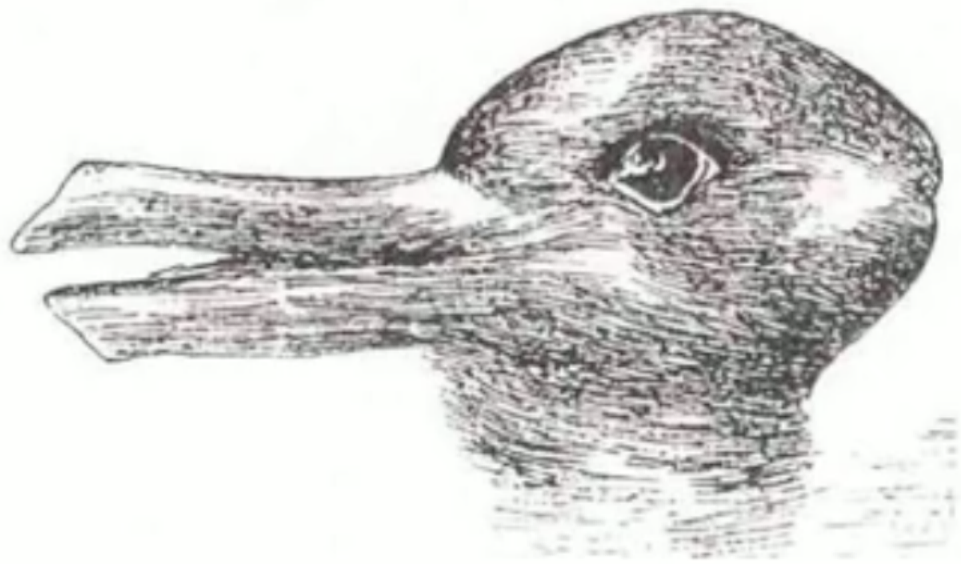
- Apply RELU.

Pooling

Flattening

Full connection

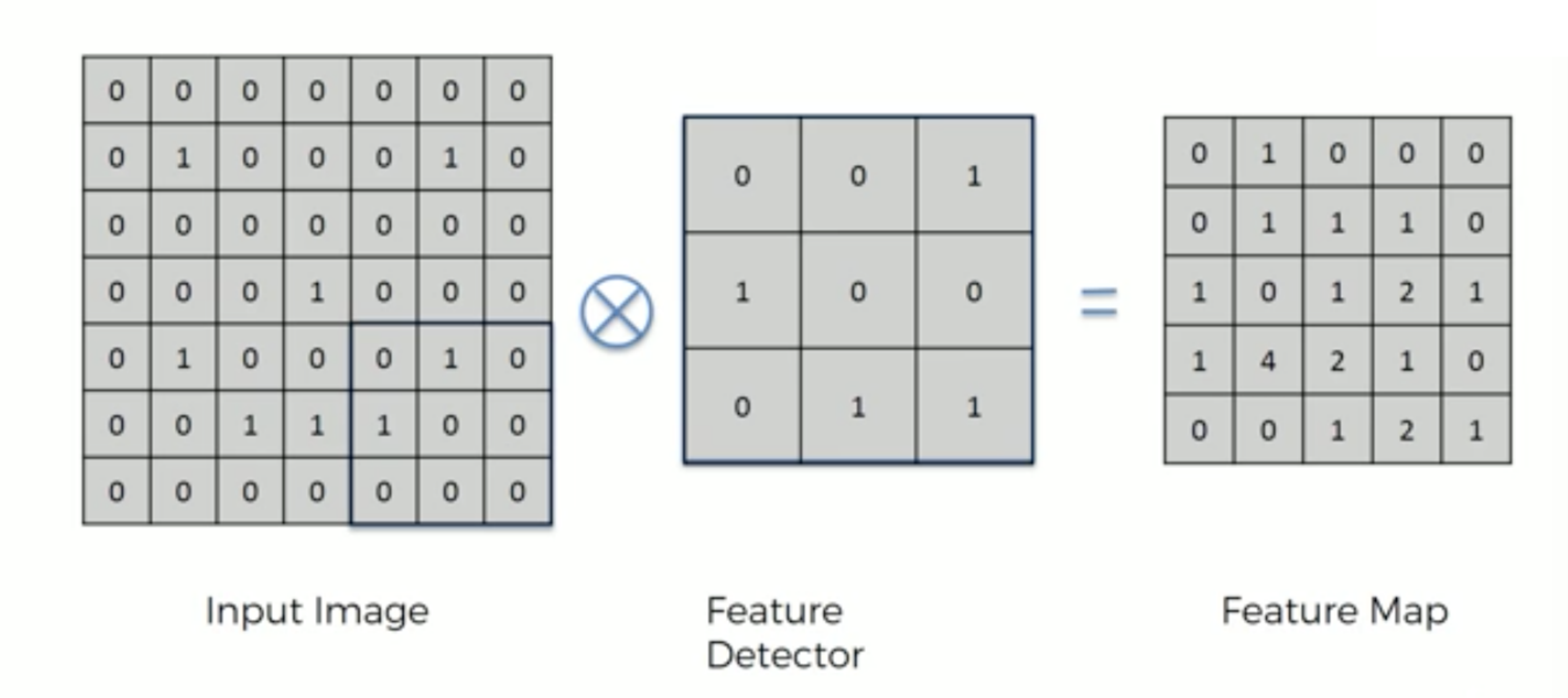
Our brain deals with features



Convolution

Yann Lecun – Pioneered CNN (works at facebook)

$$(f * g)(t) \stackrel{\mathrm{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$



Feature detectors (kernels or fiilter)

Strides

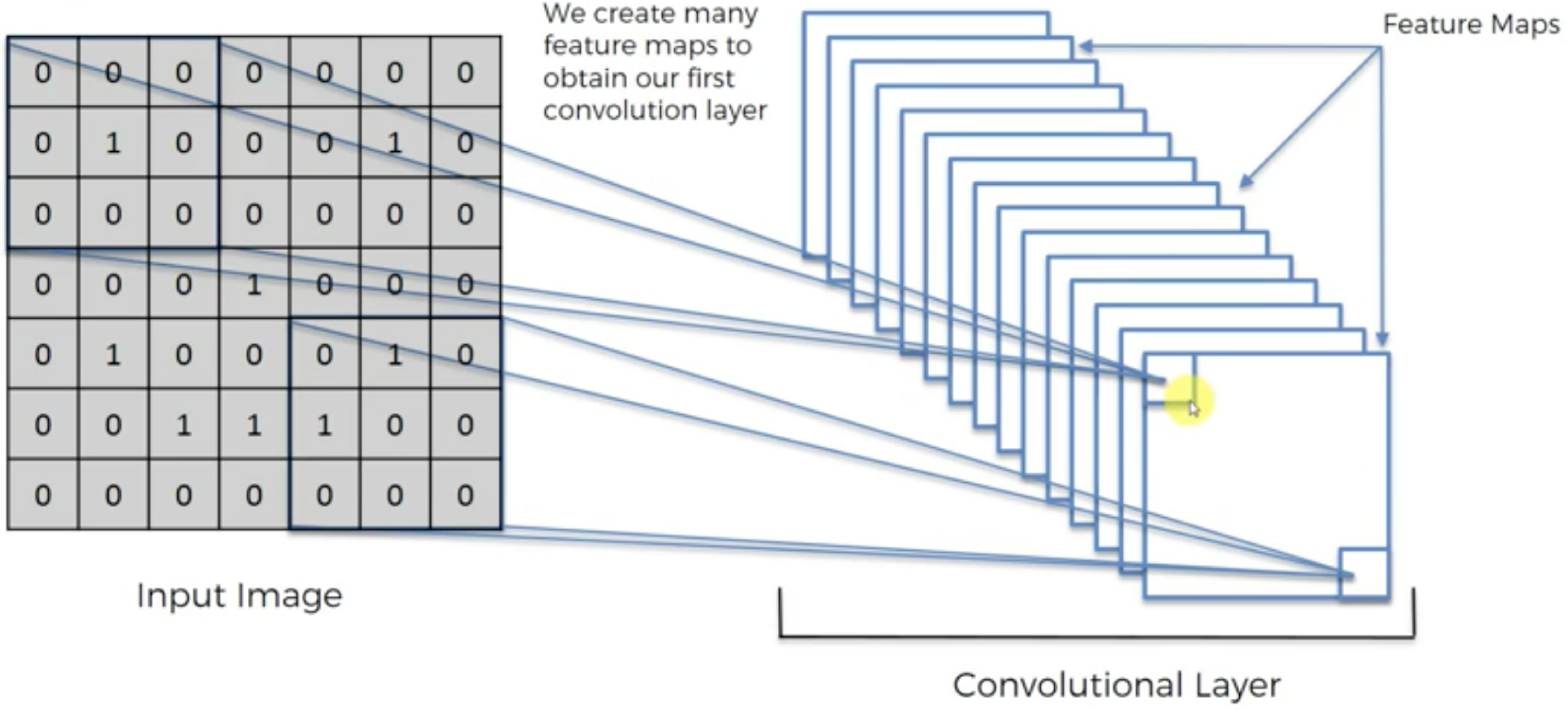
Feature maps helps us preserve the important parts of the image

VIDEO Andrew Ng.

Jianxin Wu –Paper Intro to convolution neural networks.

Convolution – detect features

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

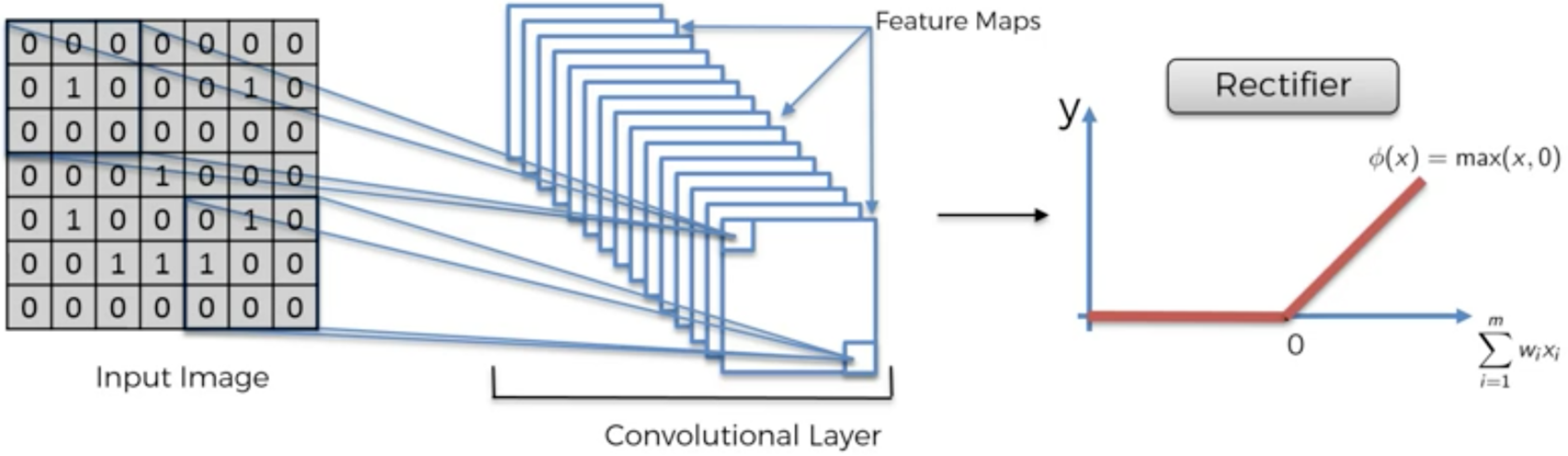


https://docs.gimp.org/2.8/en/plug-in-convmatrix.html

Jianxin Wu –Paper Intro to convulational neural networks.

Convolution – Apply RELU

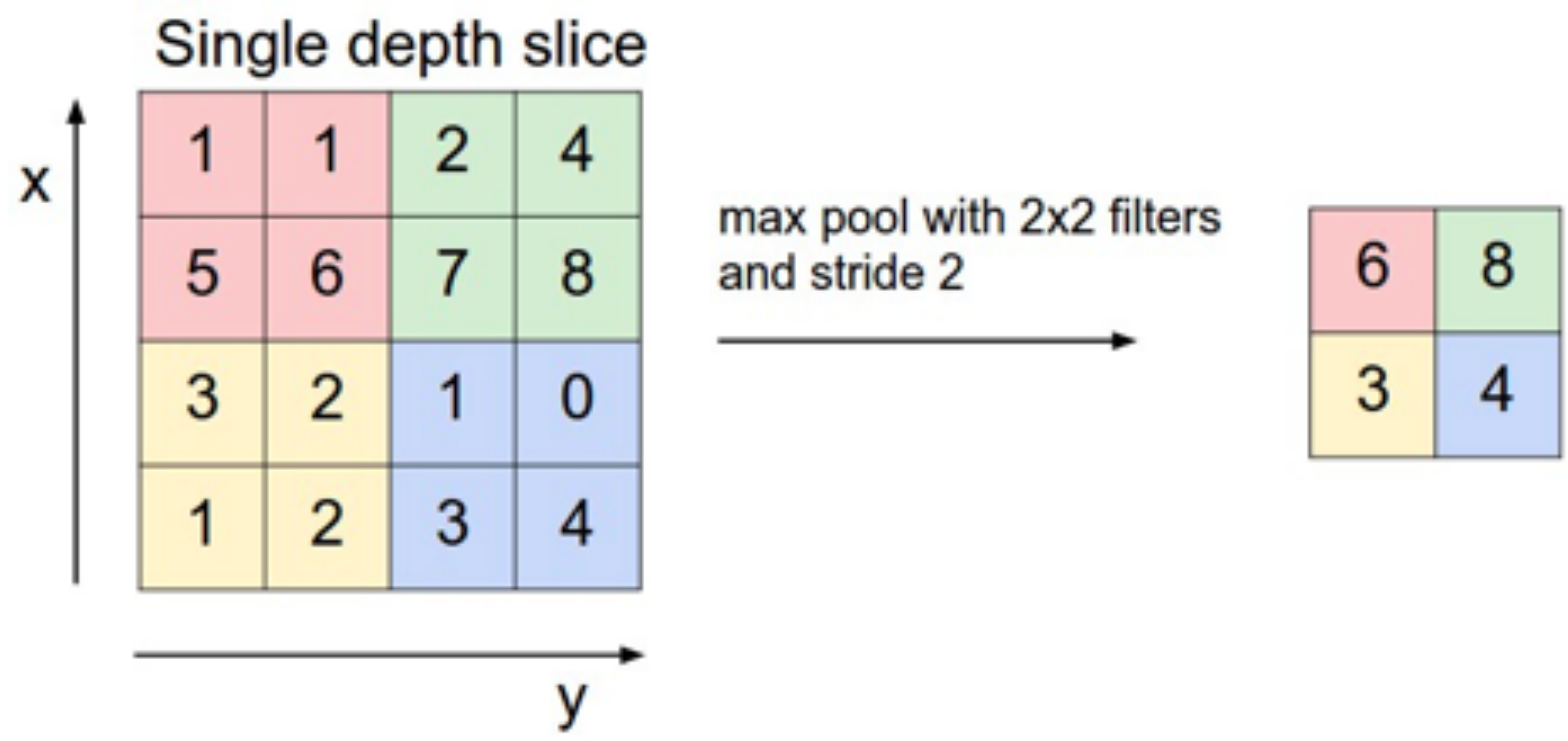
$$(f * g)(t) \stackrel{\text{def}}{=} \int^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$



Apply RELU because it increase non-linearity because images themselves are non-linear. We need to break up the linearity.

Jianxin Wu –Paper Intro to convulational neural networks.

(max) Pooling



Function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network.

Pooling layer operates on each feature map independently.

(max) Pooling (aka downsampling)

Function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network.

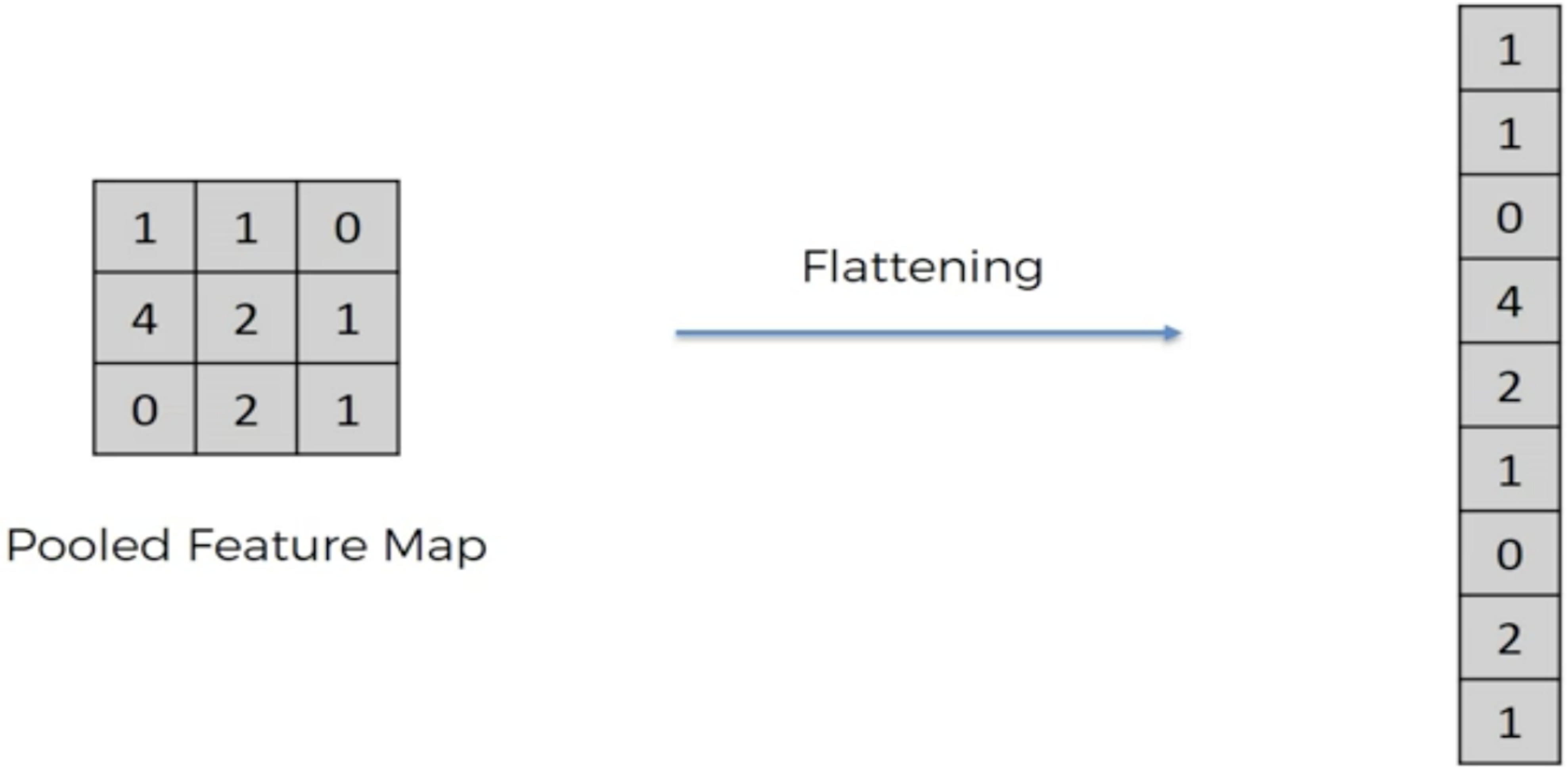
Pooling layer operates on each feature map independently.

Benefits:

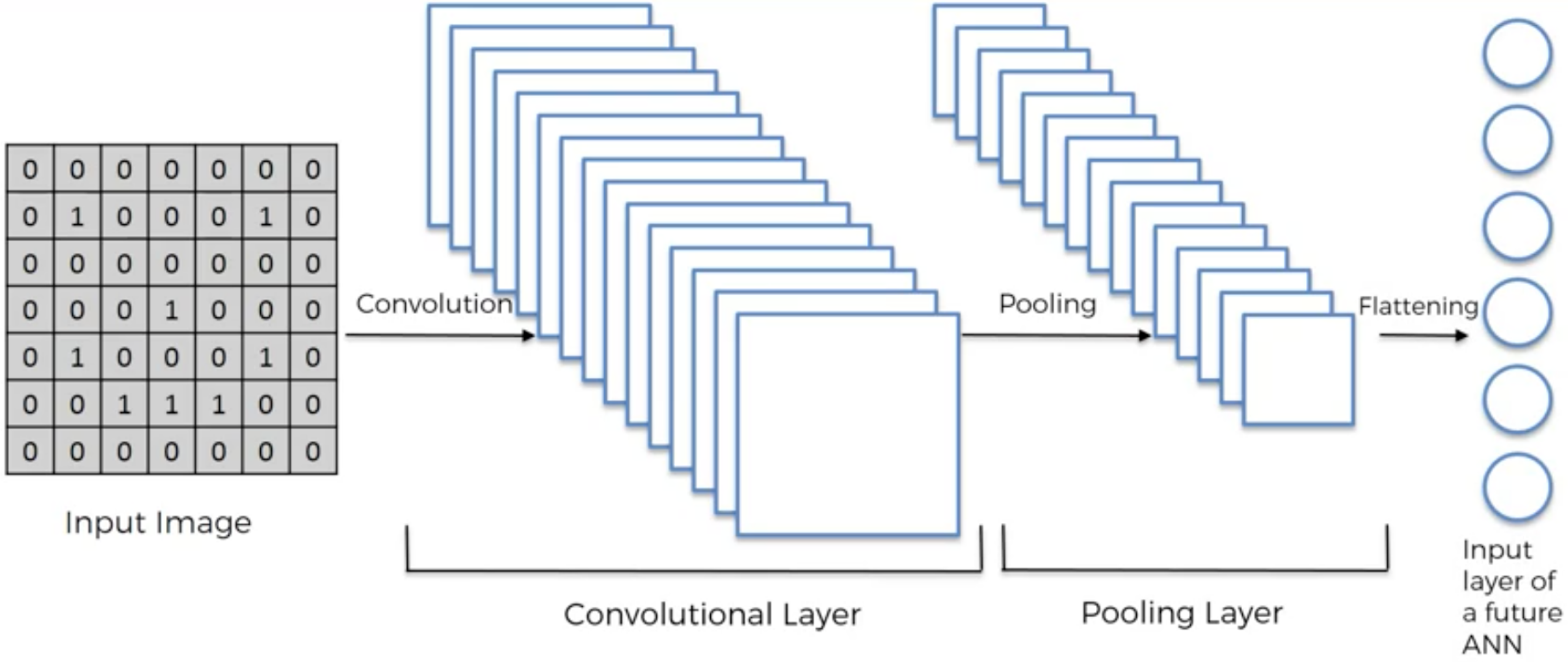
Still being able to preserve features and spatial variance
Reducing the size
Reducing the number of parameters.- thereby preventing overfitting

Dominic Sherer – evaluation of pooling white paper.

Flattening

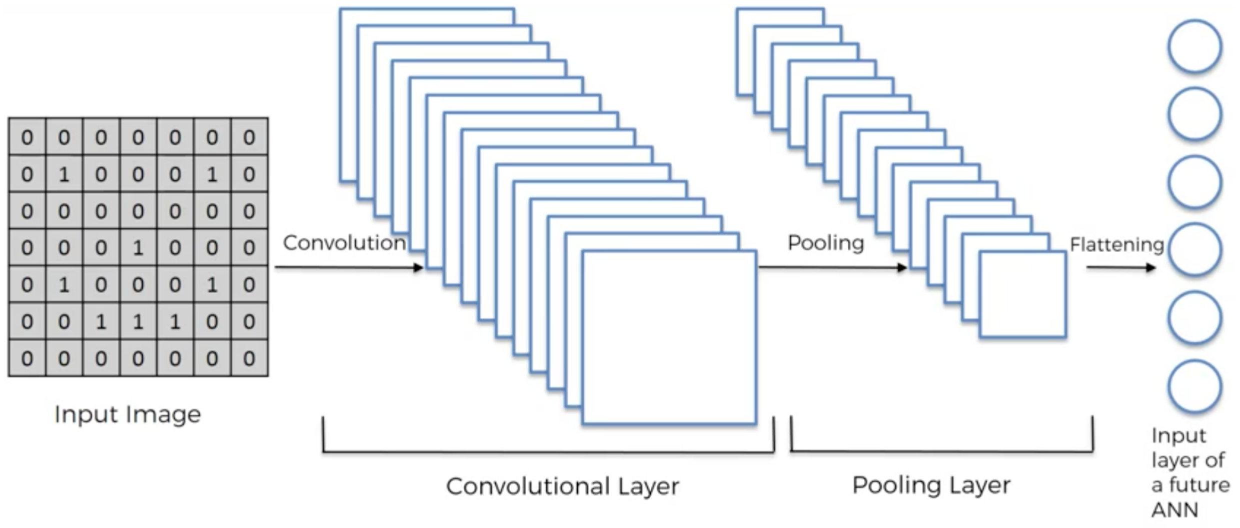


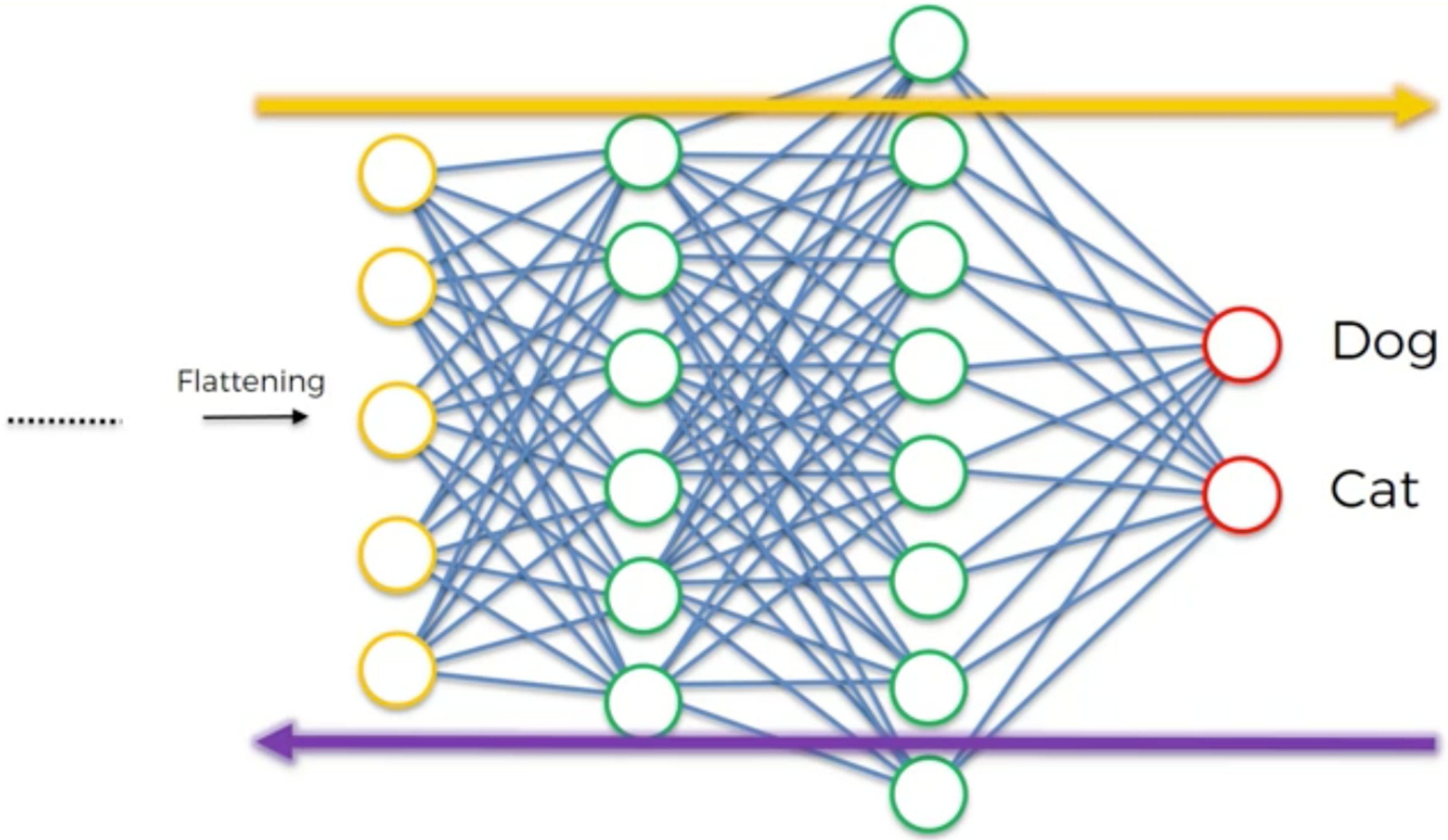
So far:



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
Convolutional Layer
Pooling
Pooling Layer
Flattening
Input layer of a future ANN

CNN - Connected to a Neural Network





Still being able to preserve features and spatial variance
Reducing the size
Reducing the number of parameters.- thereby preventing overfitting.

CNN - Connected to a Neural Network

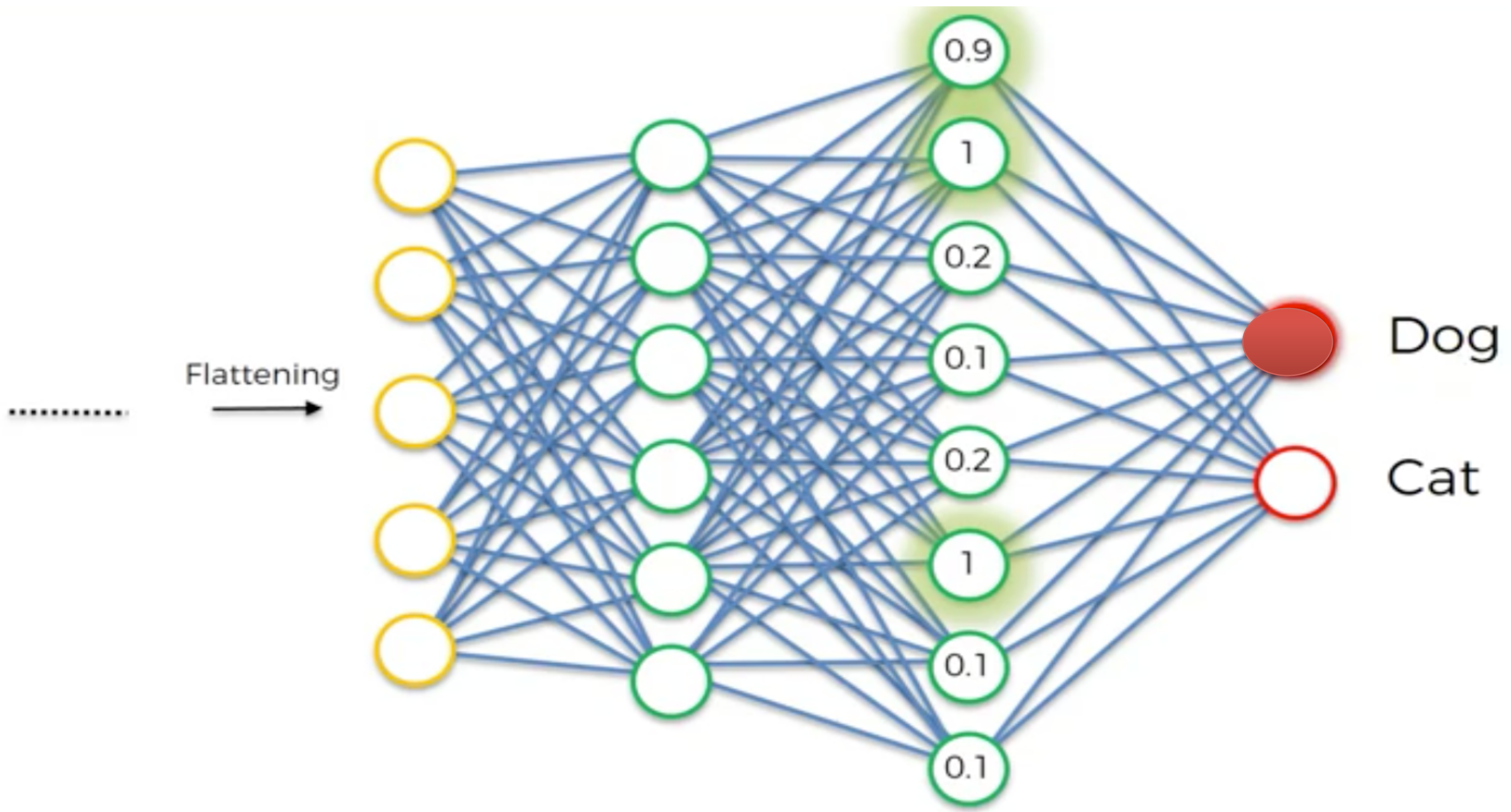


Image recognition in Action

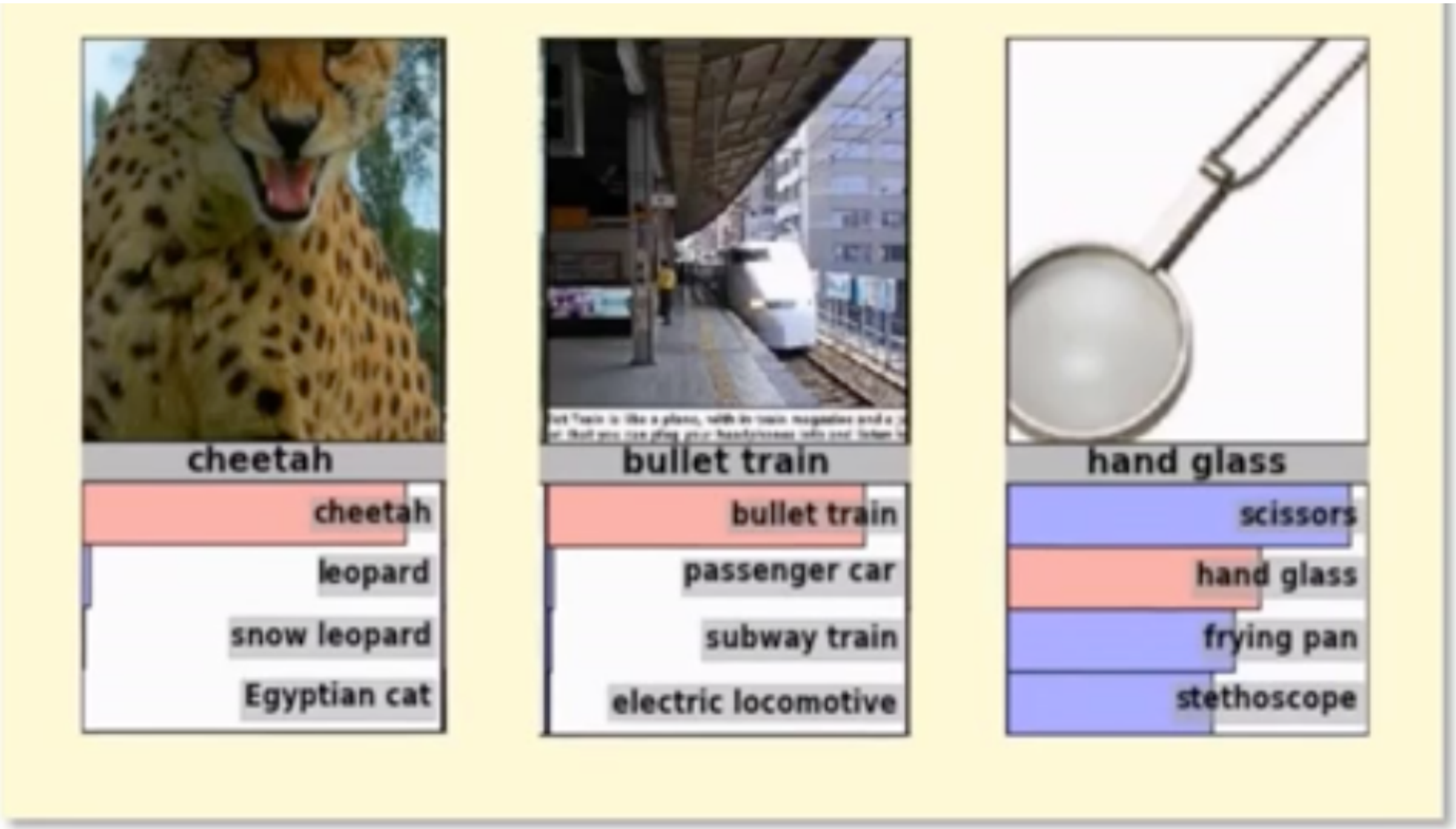
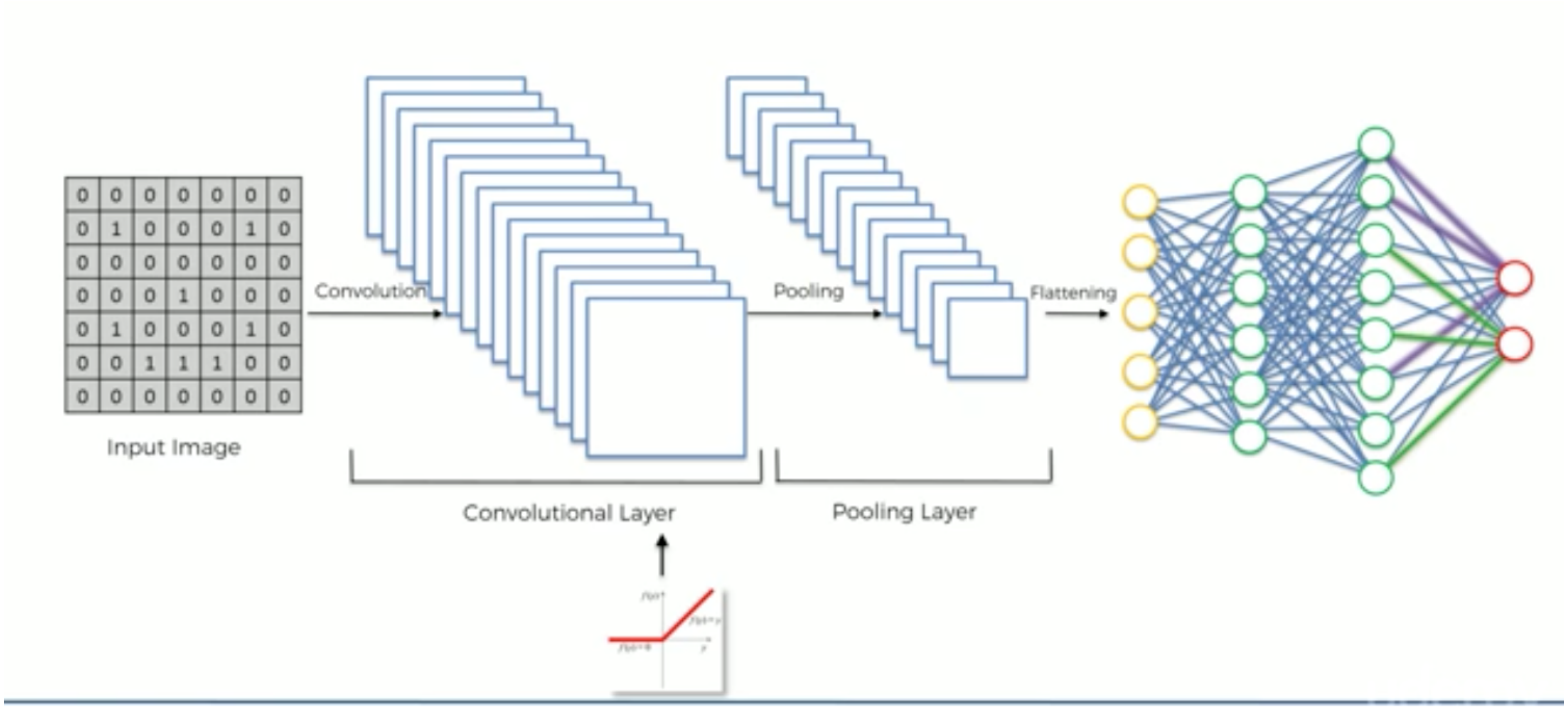


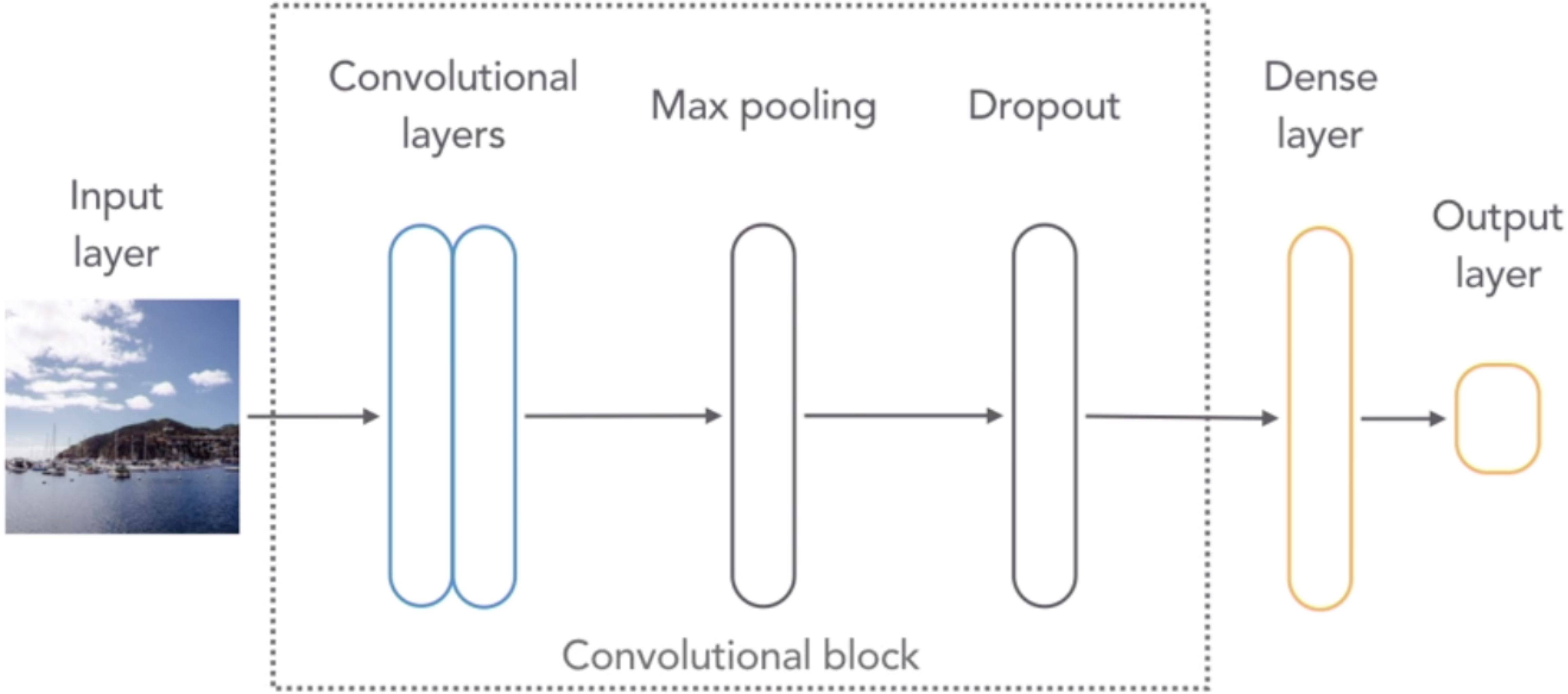
Image Source: a talk by Geoffrey Hinton

Finally



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

Another representation:



Summary of steps

Input image :to which we applied mutiple different feature detectors.
On top of layer we apply RELU to remove any linearity.

Then we apply pooling layer for each feature map
Main purpose of pooling layer to have spatial invariance , reduces size, helps with avoiding any overfitting etc. pooling preserves main features. We use max pooling.

Then we flattened pooling data into one layer of values and apply into a fully connected ANN.

Final layer provides the voting which predicts the class based on backpropogation.
Even feature detectors are adjusted in the gradient descent process.
Recognise images and classify them.