Deep Learning and Applications UEC630

Lecture – 01 (Introduction)

By

Dr. Ashu

Course structure

Syllabus

Introduction: Introduction to Deep Learning, Deep Supervised Learning, Deep Supervised Learning**.**

Deep Learning Fundamentals: Optimization Methods and Regularization: Gradient Descent, Stochastic Gradient Descent (SGD), Regularization; Neural Network Fundamentals, Convolutional Neural Networks: History of Convolutional Networks, Convolutional Networks and Computer Vision, Audio and Other Domains, Structural Prediction and Natural Language Processing

Learning with Memory: Recurrent Neural Network Basics, Advanced Recurrent Neural Networks, Sequential Data Modeling, Embedding Methods for NLP: Unsupervised and Supervised Embeddings, Embedding Methods for NLP: Embeddings for Multi-relational Data, Deep Natural Language Processing

Applications: LeNet: Recognizing Handwritten Digits, MiniVGGNet: Going Deeper with CNNs.

Evaluation elements	Weightage
MST	30%
EST	40%
Sessional (lab component + quiz)	30% (15% + 15%)

Sessionals:

1. Lab Component (15%):

Lab assignments

or

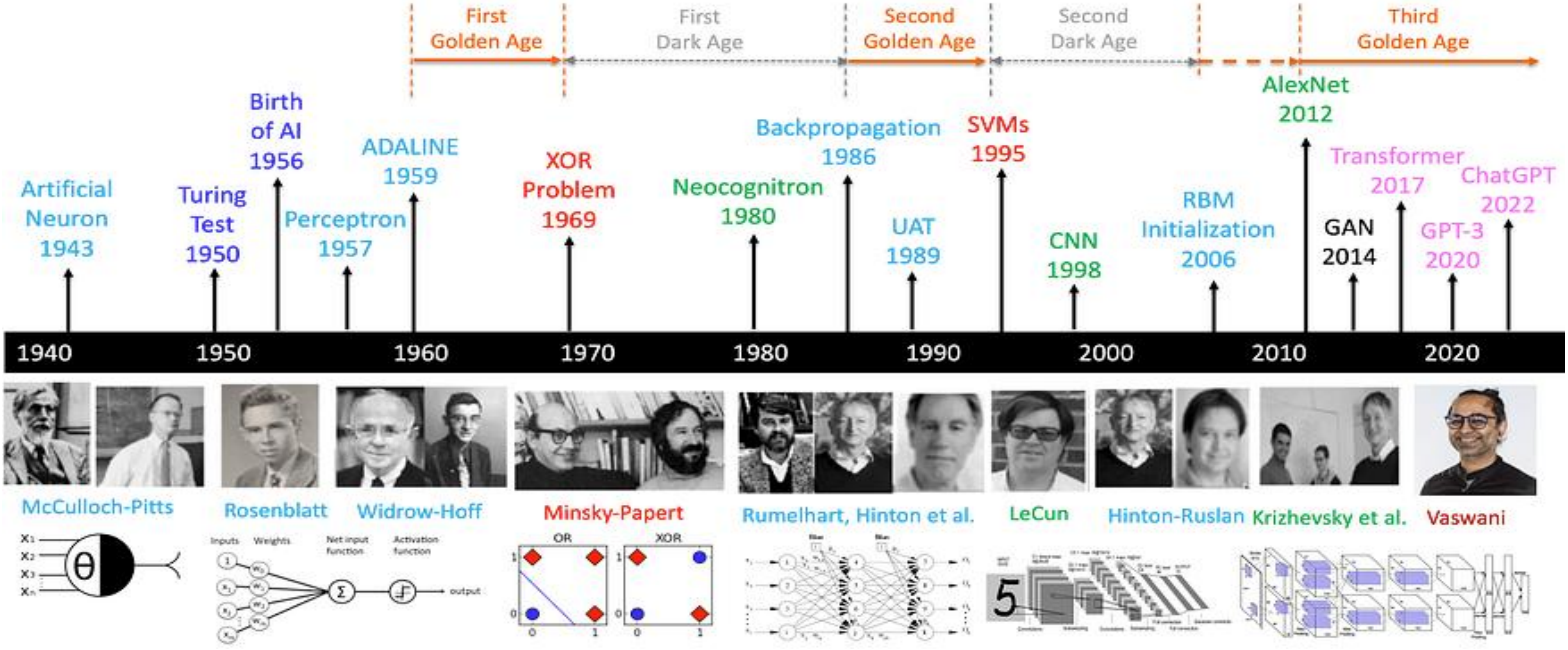
Mini deep learning projects

Both components include regular assignment submission and viva/presentation towards the end of semester.

2. Quiz after MST (Syllabus upto MST) (15%)

Introduction to Deep Learning

A Brief History of AI with Deep Learning



Deep Learning

- https://www.youtube.com/watch?v=AmUC4m6w1wo
- study material\Barack Obama _ Intro to Deep Learning _ MIT 6.S191 (online-video-cutter.com).mp4

Deep Learning

2020

…creating this 2 minute video required…

2 hours of professional audio

50 hours of HD video

Static, pre-defined script

Over $15K USD of compute

Deep Learning

2020

…creating this 2 minute video required…

2 hours of professional audio

50 hours of HD video

Static, pre-defined script

Over $15K USD of compute

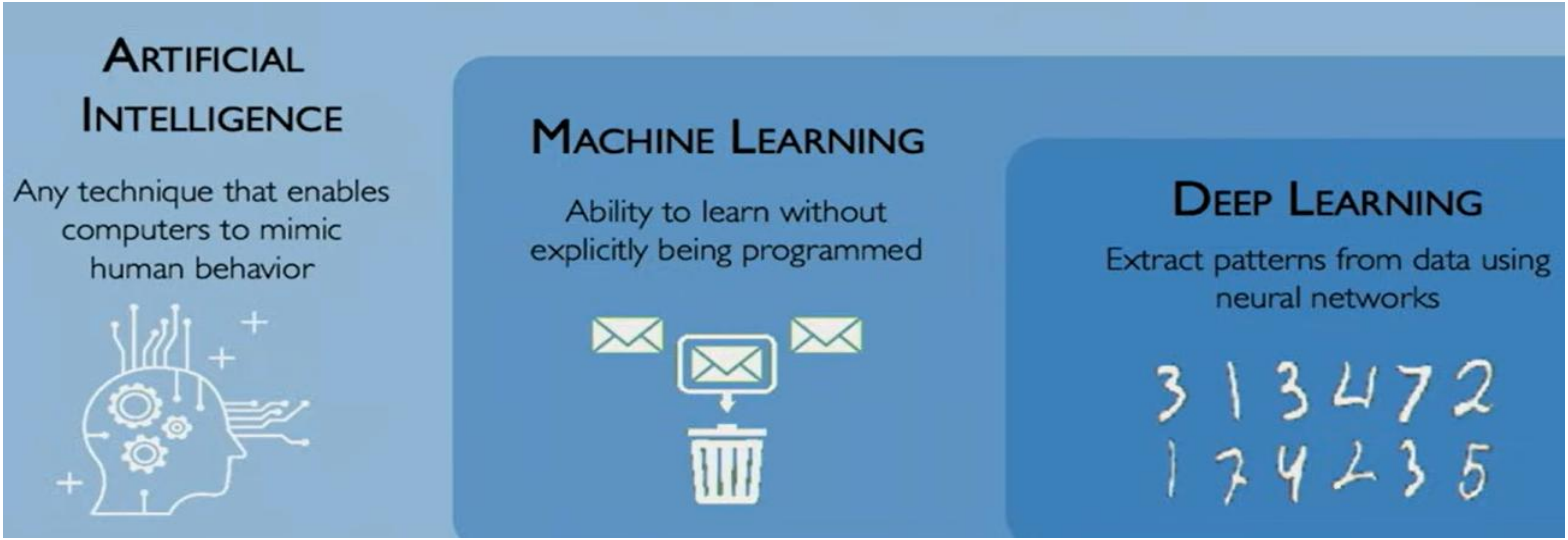
2025

…fast forward a few years…

?

source:developer.nvidia.com/deep--learning--courses

Deep Learning



Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL)

AI is intelligence demonstrated by machines, as opposed to natural intelligence displayed by animals including humans.

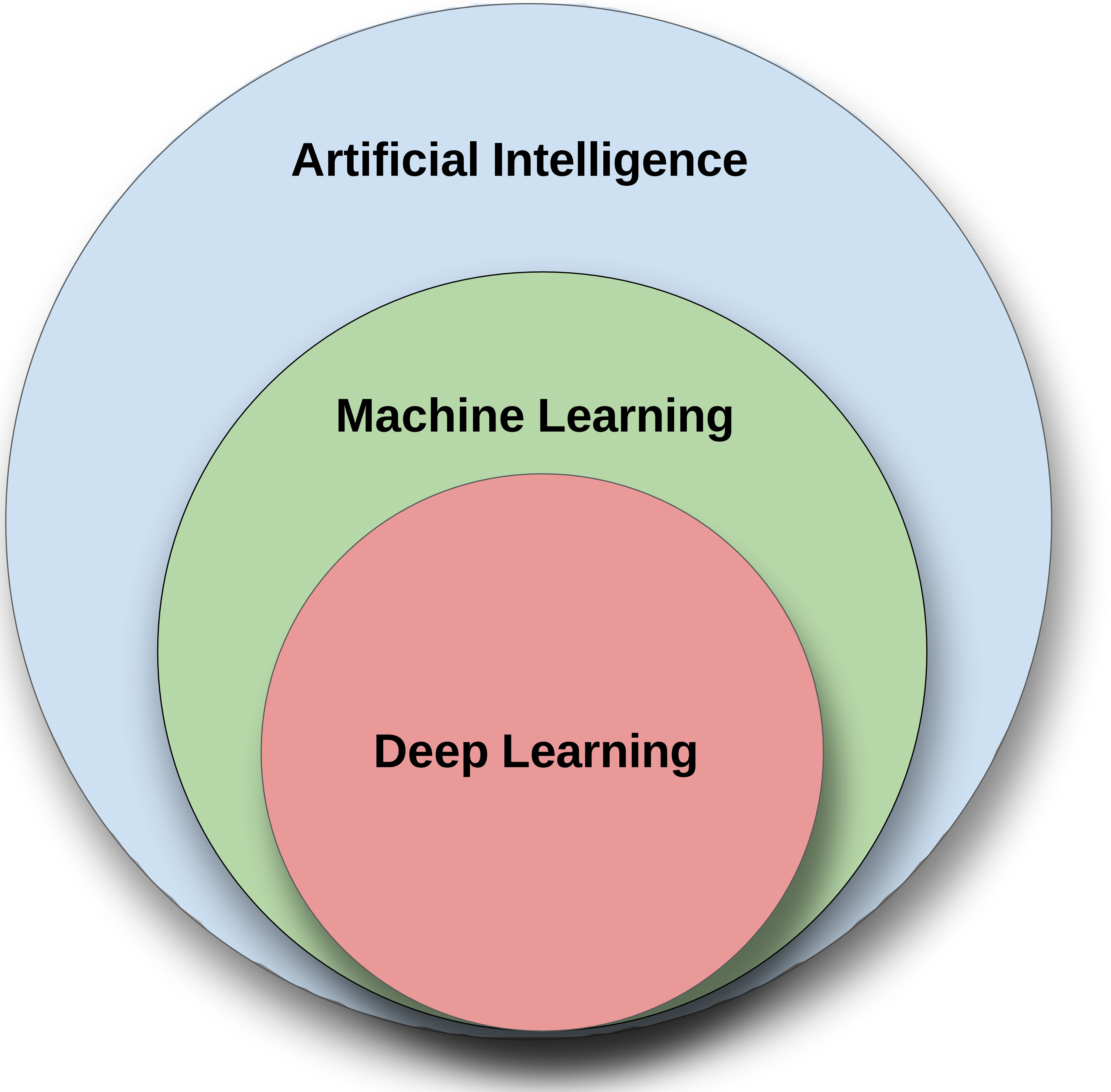
ML is the study of computer algorithms that can improve automatically through experience and using data. It is seen as a part of artificial intelligence.

DL is part of a broader family of machine learning methods based on artificial neural networks.

(Source: Wikipedia)

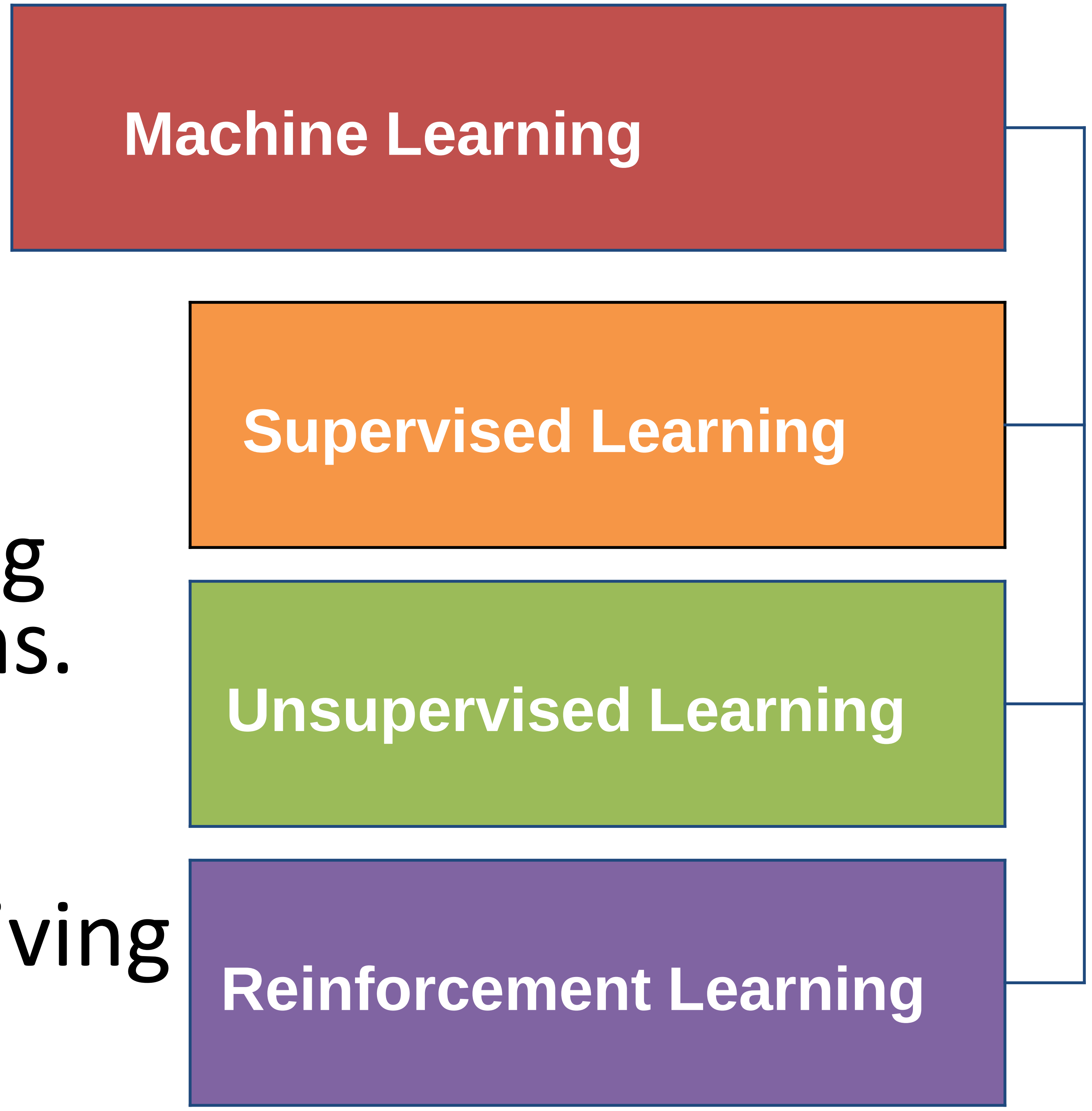
Relationship of AI, ML and DL

- **Artificial Intelligence (AI)** is anything about man-made intelligence exhibited by machines.
- **Machine Learning (ML)** is an approach to achieve **AI**.
- **Deep Learning (DL)** is one technique to implement **ML**.



Types of ML Algorithms

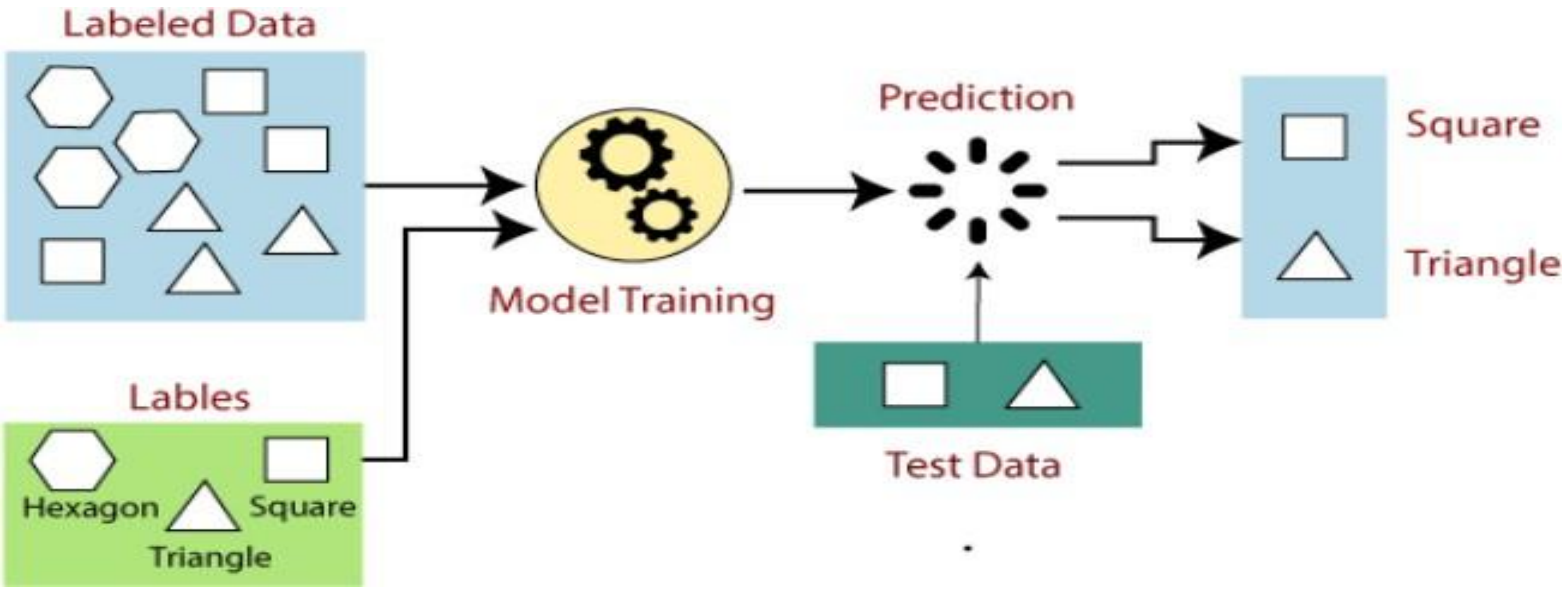
- **Supervised Learning**
  - trained with labeled data; including regression and classification problems
- **Unsupervised Learning**
  - trained with unlabeled data; clustering and association rule learning problems.
- **Reinforcement Learning**
  - no training data; stochastic Markov decision process; robotics and self-driving cars.



Strategy / Learning Algorithm

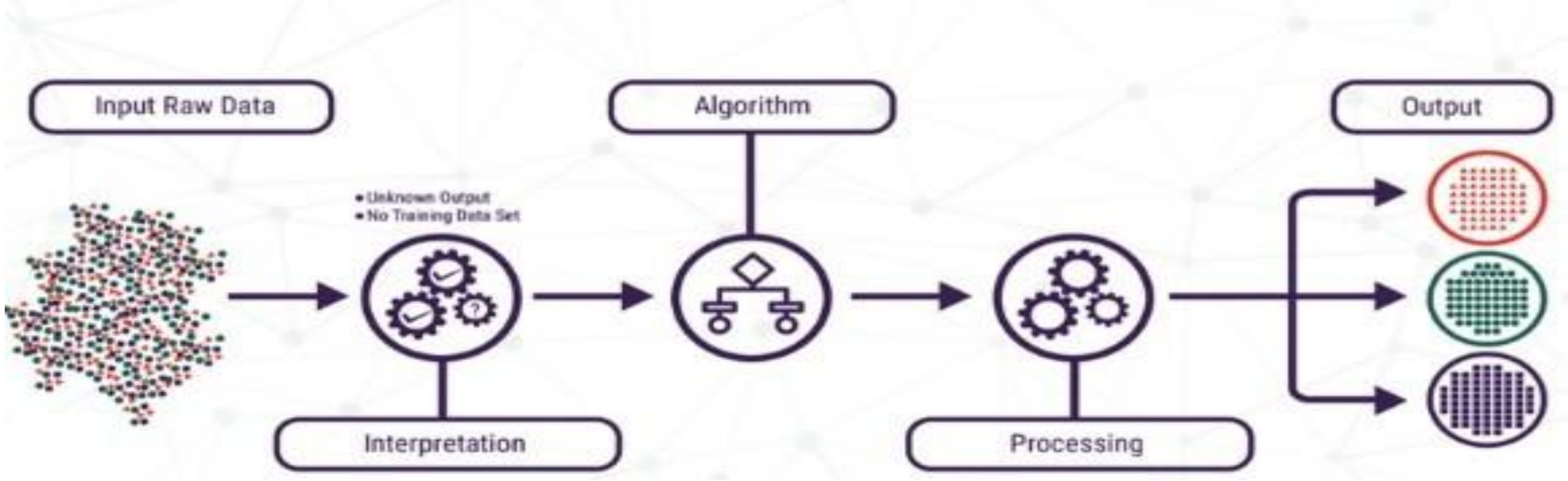
Supervised Learning

- Learning is performed by presenting pattern with target
- During learning, produced output is compared with the desired output
  - The difference between both output is used to modify learning weights according to the learning algorithm
- Recognizing hand-written digits, pattern recognition and etc.
- Neural Network models: perceptron, feed-forward, radial basis function, support vector machine.

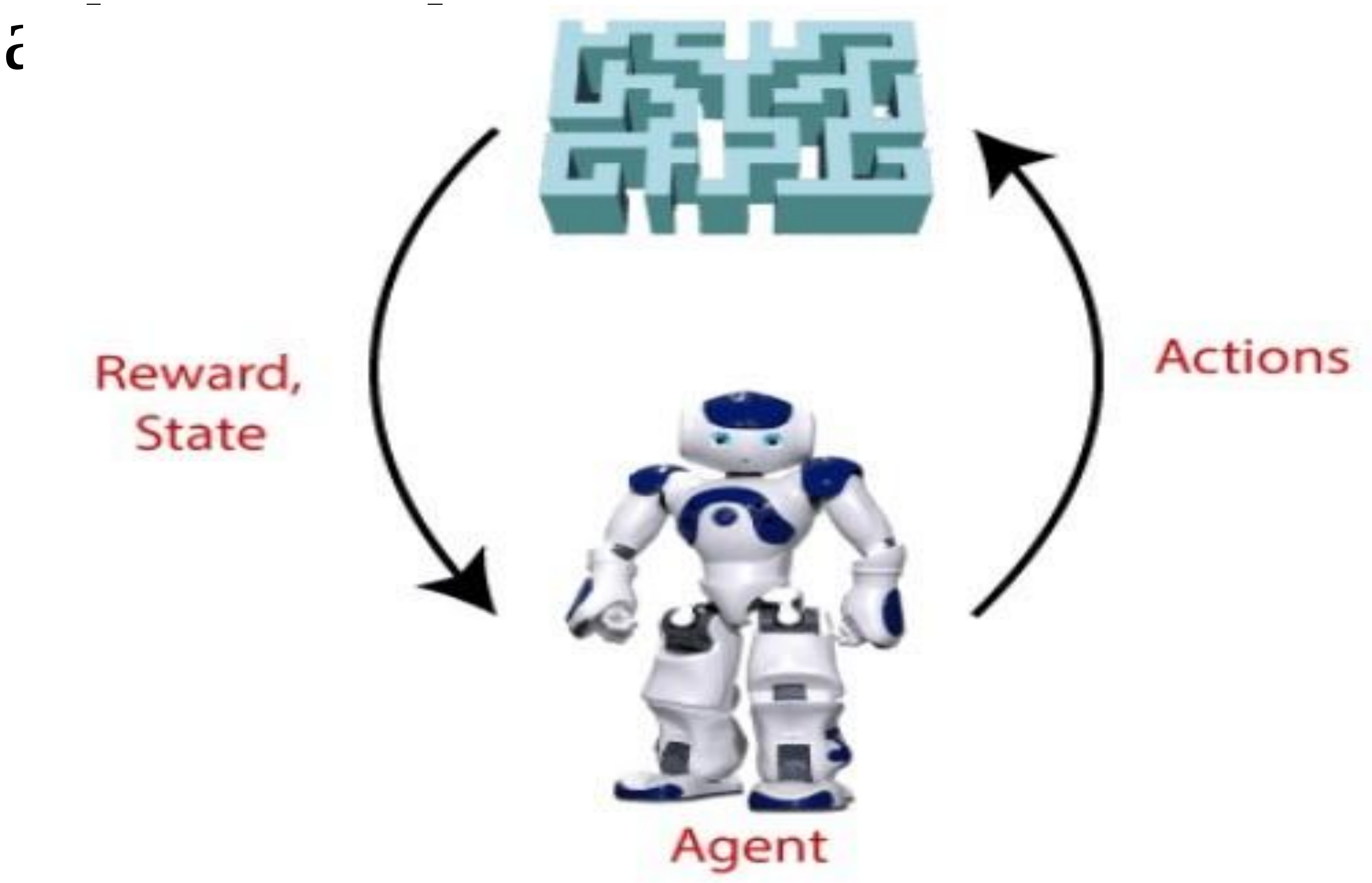


Unsupervised Learning

- Targets are not provided
- Appropriate for clustering task
  - Find similar groups of documents in the web, content addressable memory, clustering.
- Neural Network models: Kohonen, self organizing maps, Hopfield networks.



Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedba



Why Deep Learning?

- Limitations of traditional machine learning algorithms
  - not good at handling high dimensional data.
  - difficult to do feature extraction and object recognition.

- Advantages of deep learning
  - DL is computationally expensive, but it is capable of handling high dimensional data.
  - feature extraction is done automatically.

Machine Learning

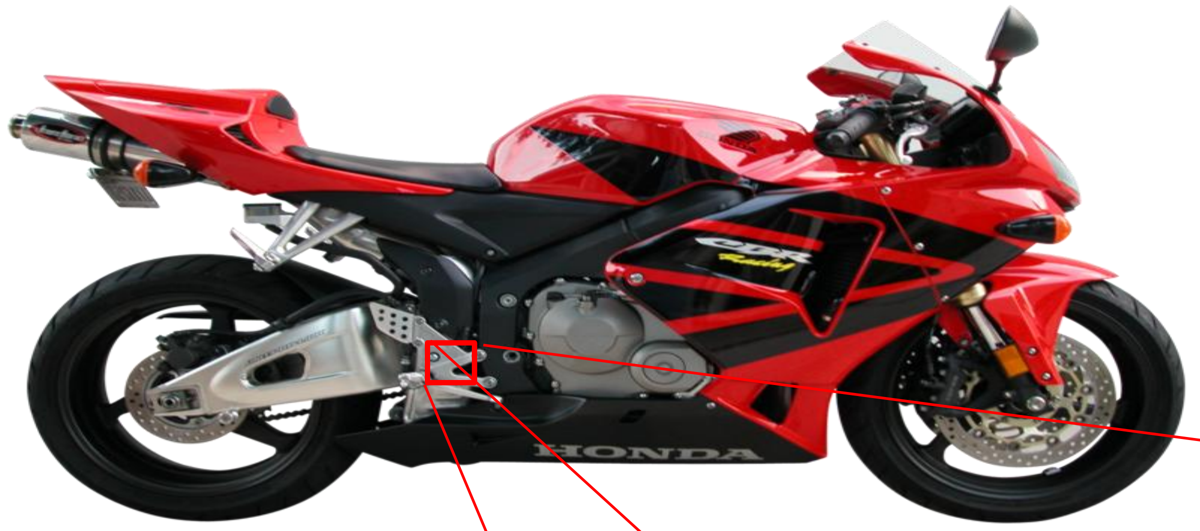
Input: X

Output: Y

Label"motorcycle"

Why is it hard?

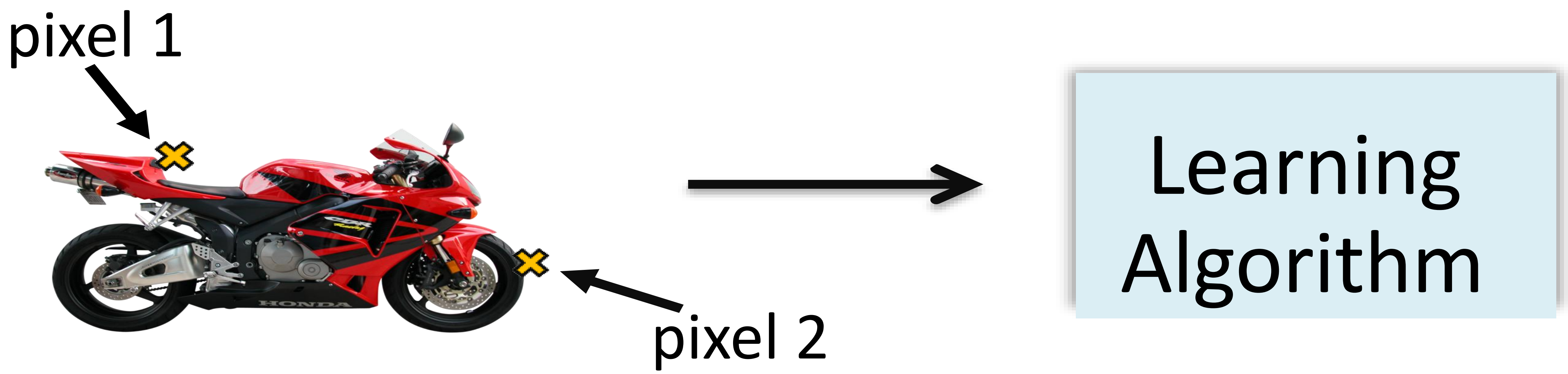
You see this

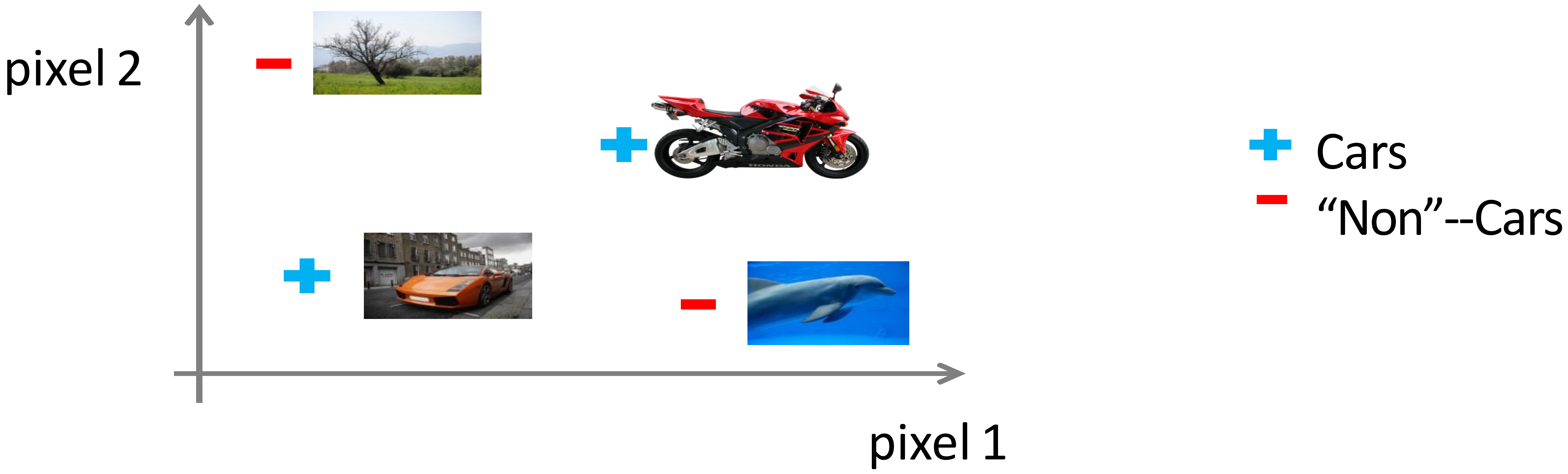


But the camera sees this:

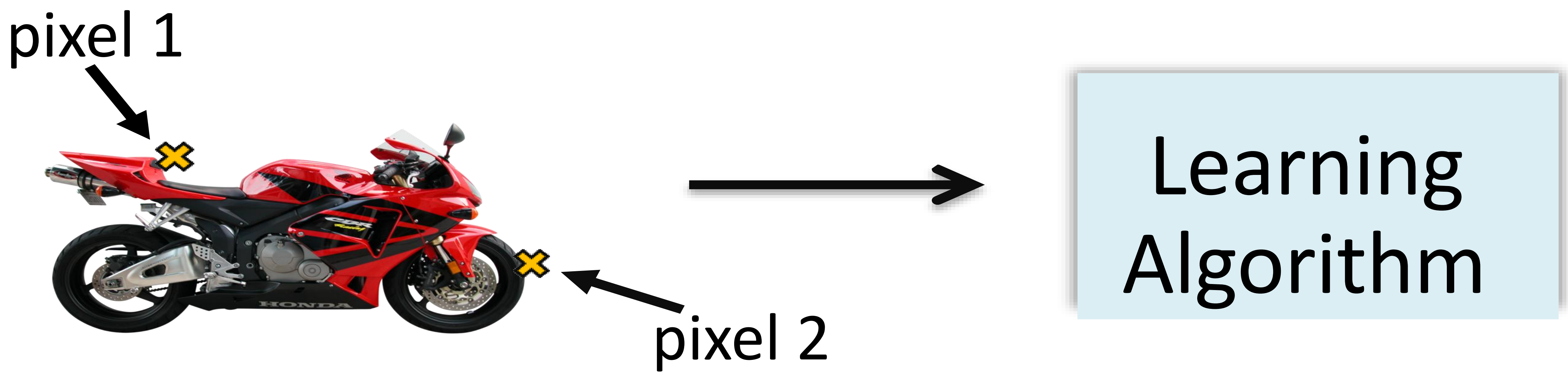
194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

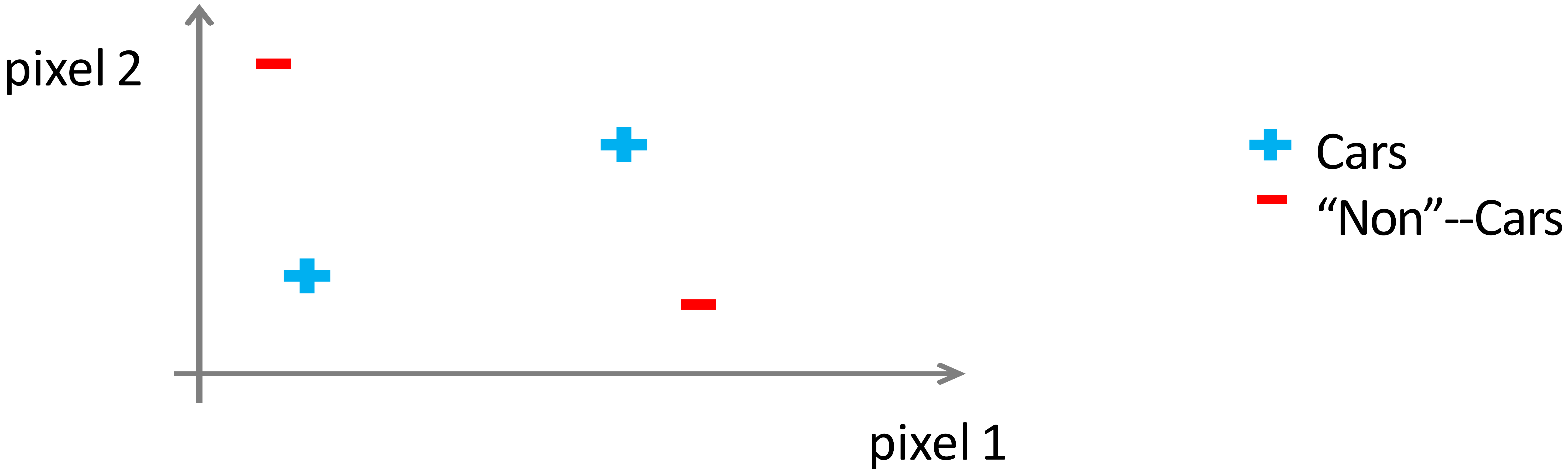
Raw Image Representation





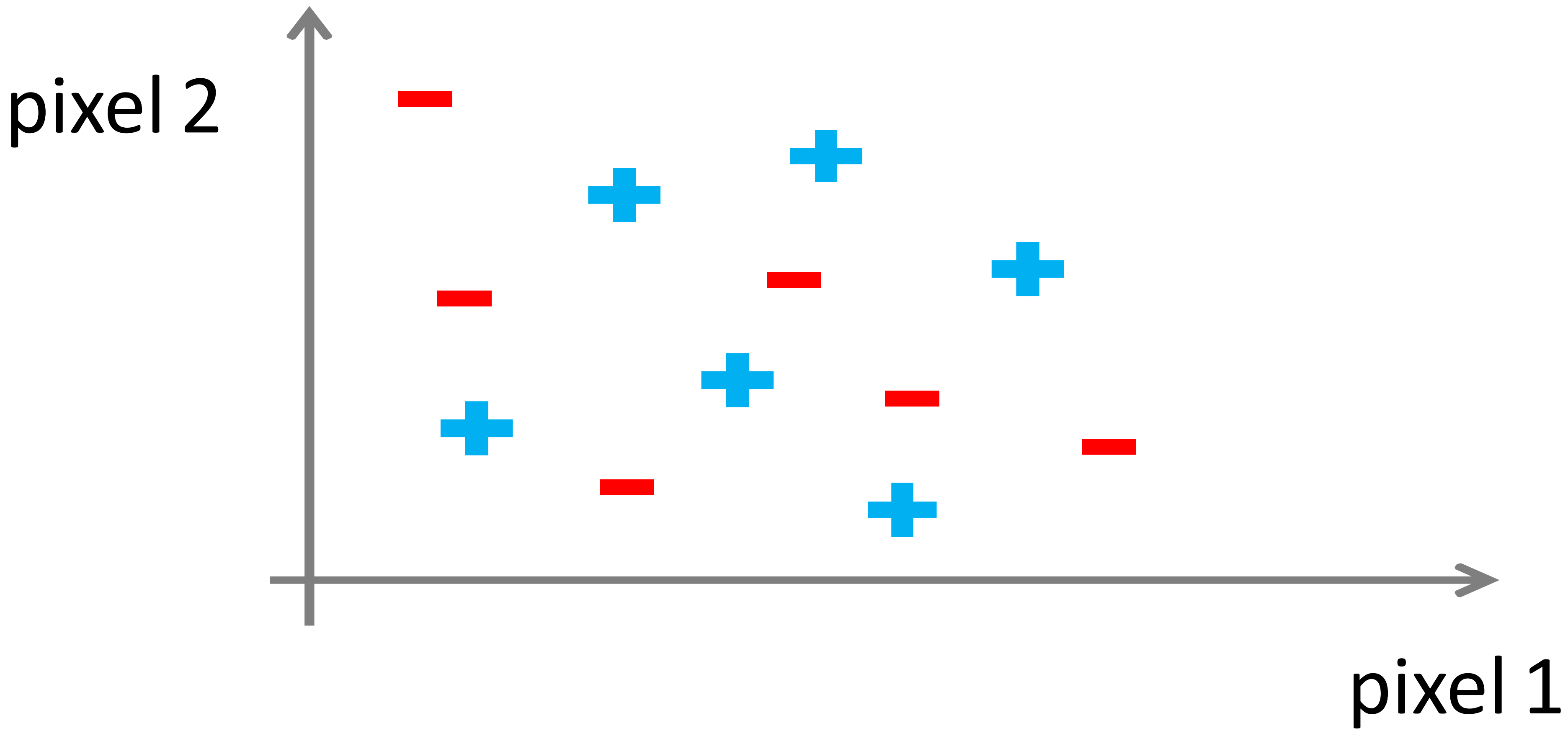
Raw Image Representation





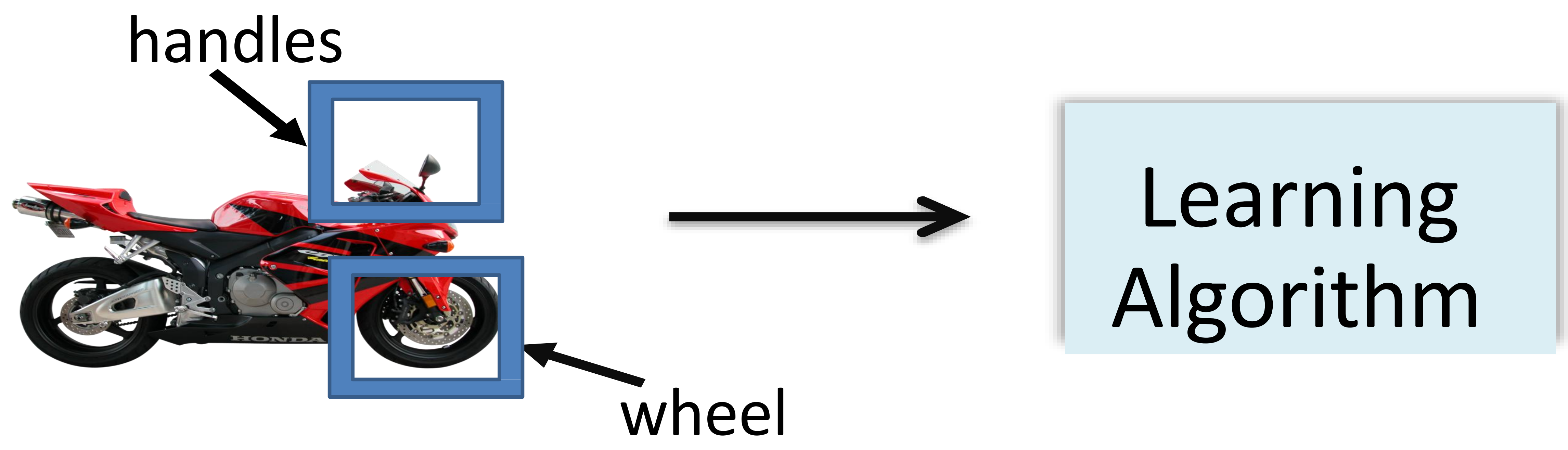
Raw Image Representation

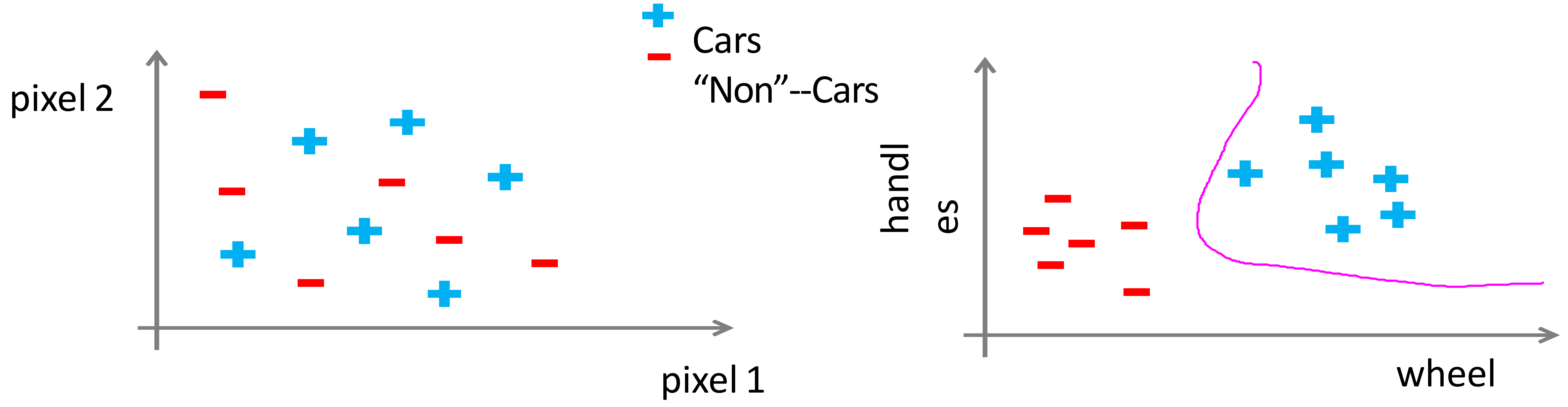
Learning Algorithm



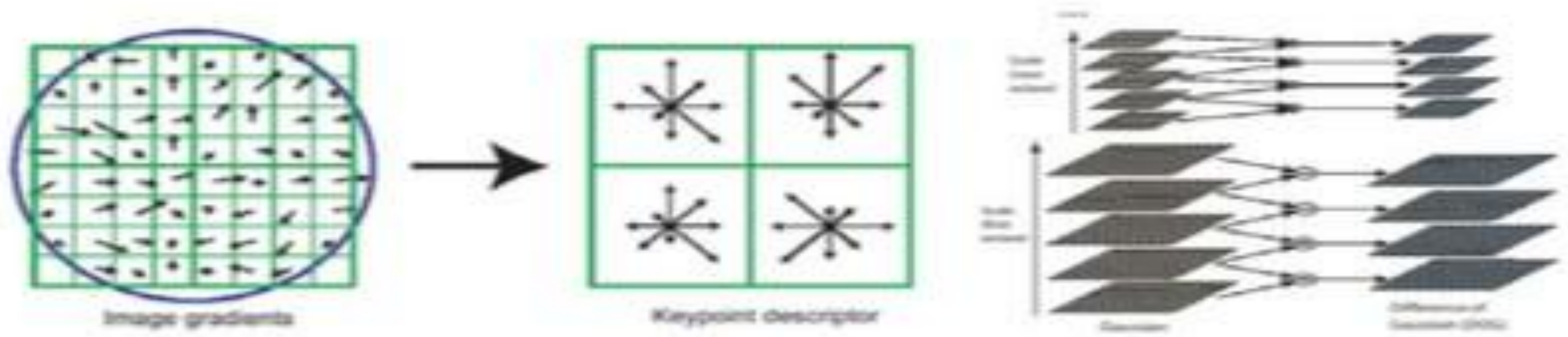
Cars
"Non"--Cars

Better Feature Representation?

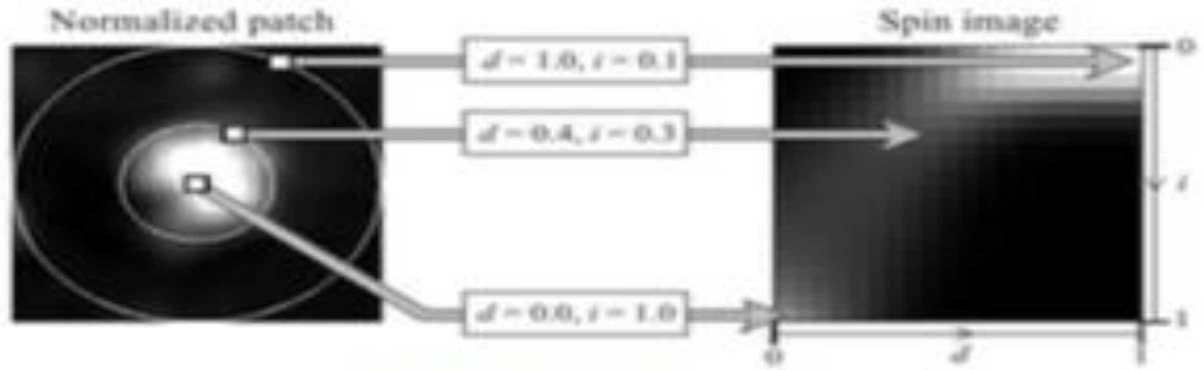




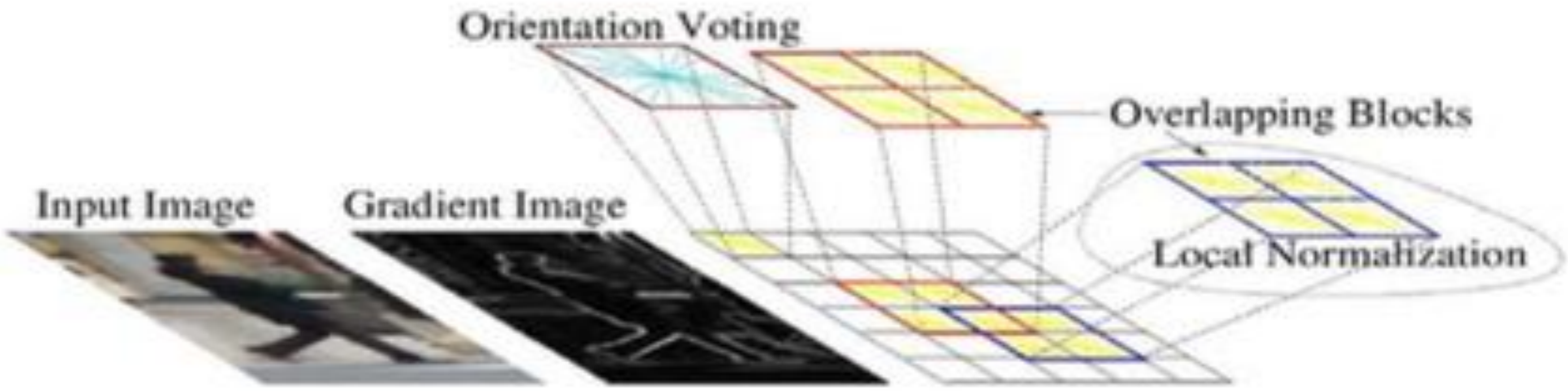
Feature Representations



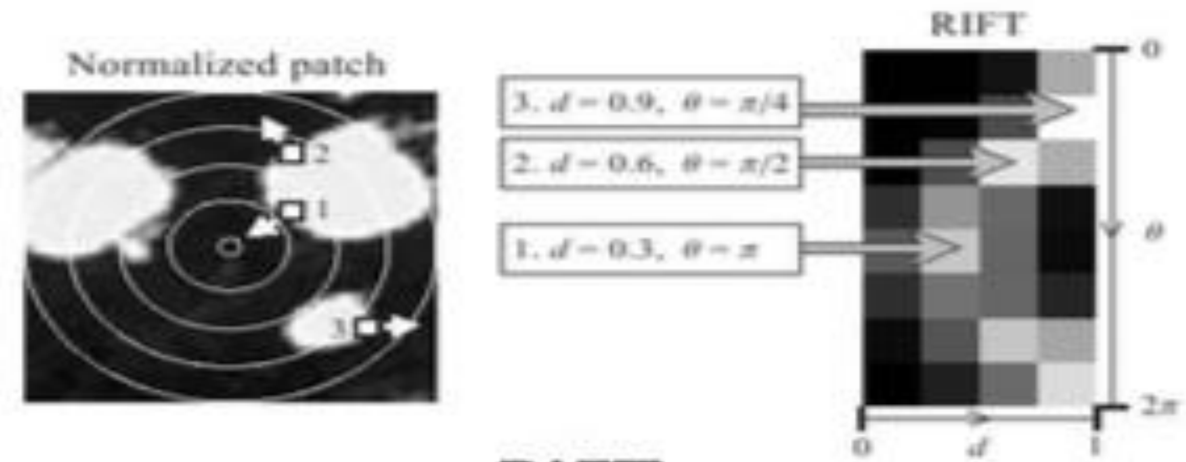
SIFT



Spin image

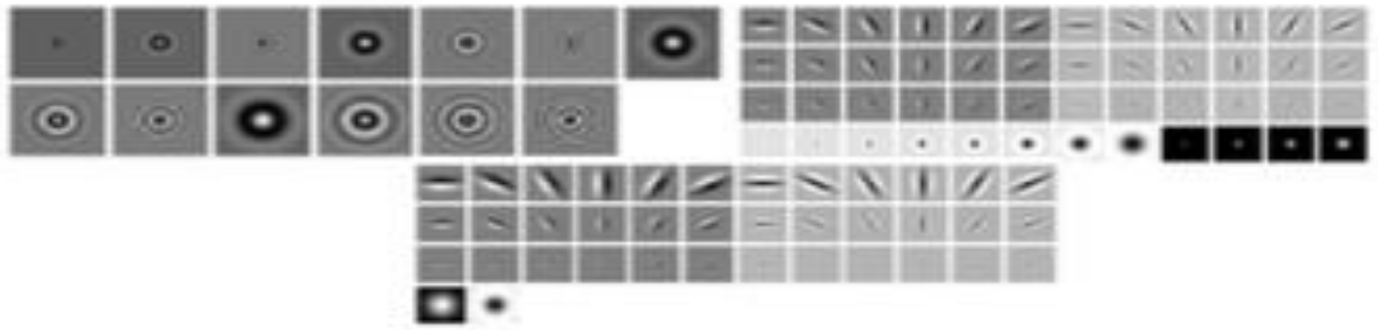


HoG

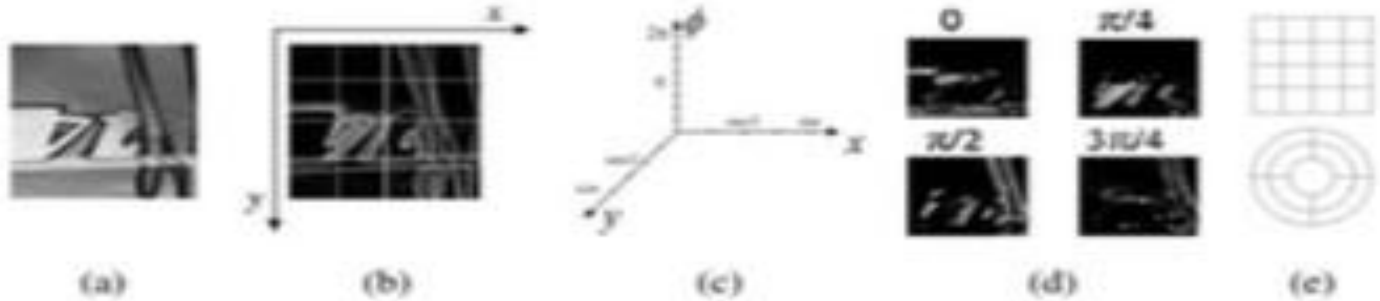


RIFT

Expert Knowledge!



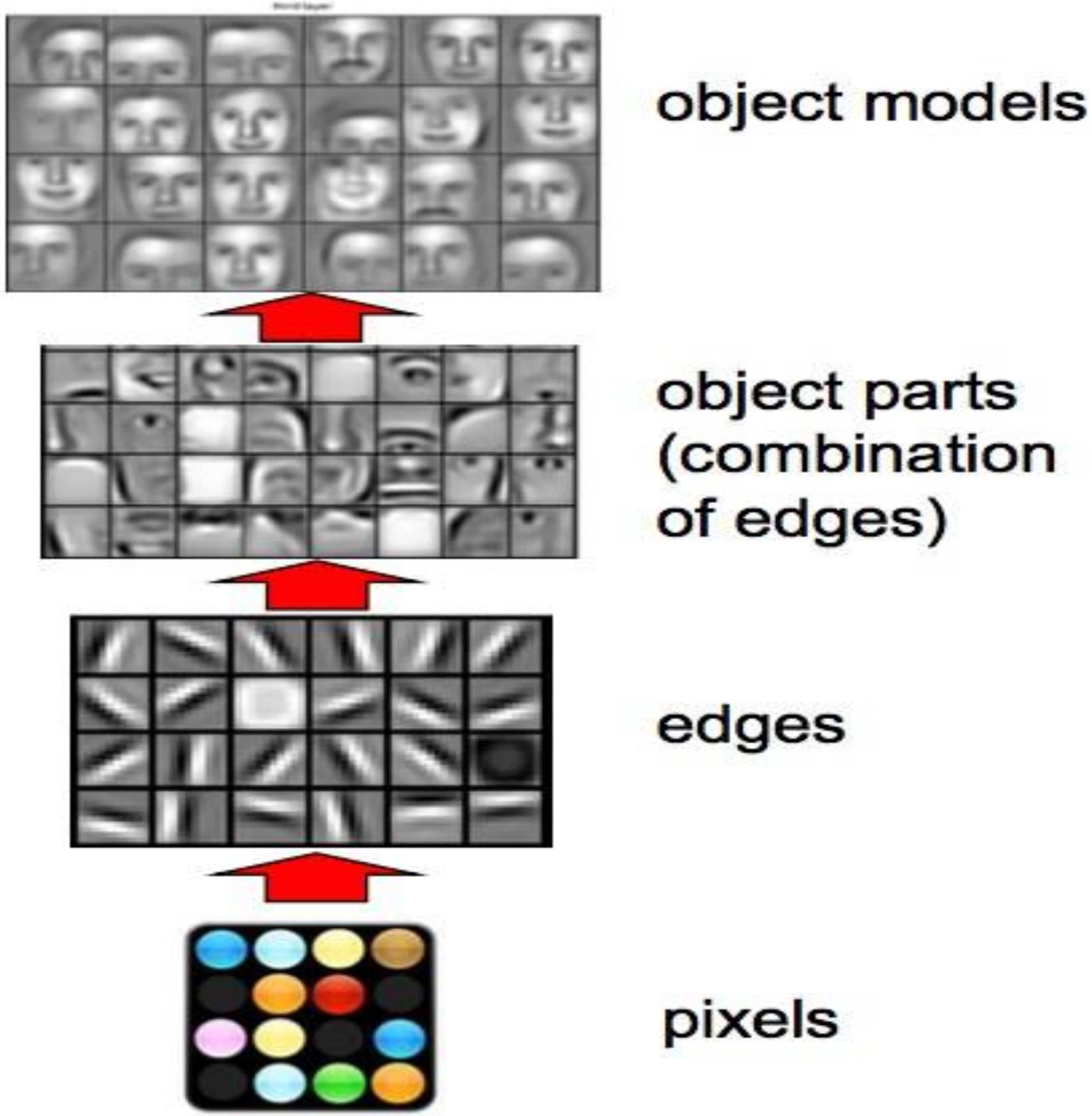
Textons



GLOH

Source: feature representations in computer vision(Honglak lee)

Deep Learning: learn representations!



Source:Lee et.al. ICML2009

So, 1. **what exactly is deep learning** ?

And, 2. **why is it generally better** than other methods on image, speech and certain other types of data?

The short answers

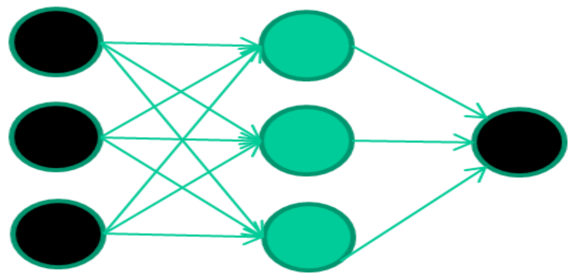
1. **'Deep Learning' means** using a **neural network** with **several layers of nodes** between input and output

2. **the series of layers between input & output do feature identification and processing in a series of stages, just as our brains seem to.**

Sources: https://www.macs.hw.ac.uk/~dwcorne/Teaching/**introdl.ppt** hmmm… OK, but:

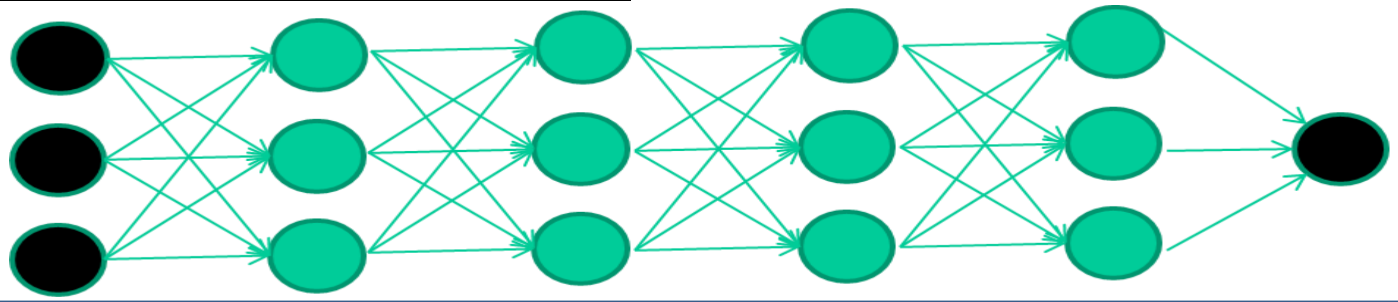
3. **multilayer neural networks have been around for 25 years.   What's actually new?**

we have always had good algorithms for learning the weights in networks with 1 hidden layer



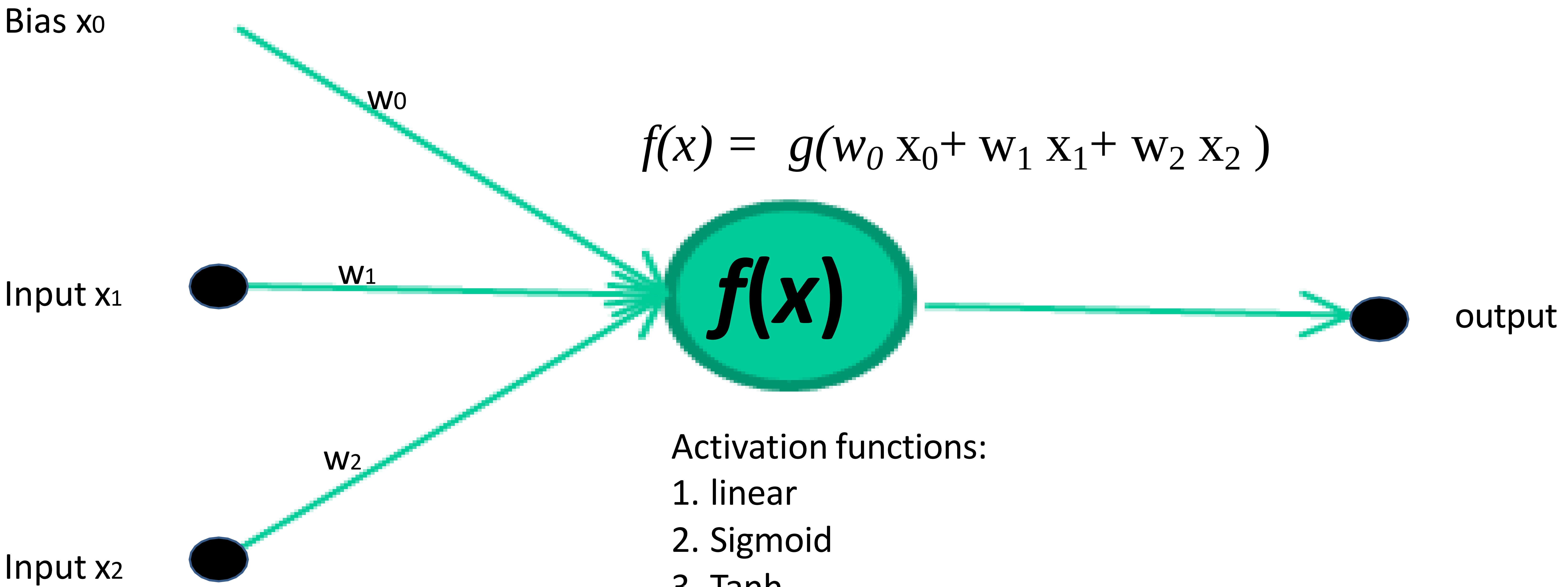
but these algorithms are not good at learning the weights for networks with more hidden layers

what's new is: **<u>algorithms for training many--later networks</u>**



https://www.macs.hw.ac.uk/~dwcorne/Teaching/introdl.ppt

Single Unit, Input, weights, activation function, output



$$f(x) = g(w_0\, x_0 + w_1\, x_1 + w_2\, x_2\,)$$

Activation functions:

1. linear
2. Sigmoid
3. Tanh
4. Relu
5. Softmax

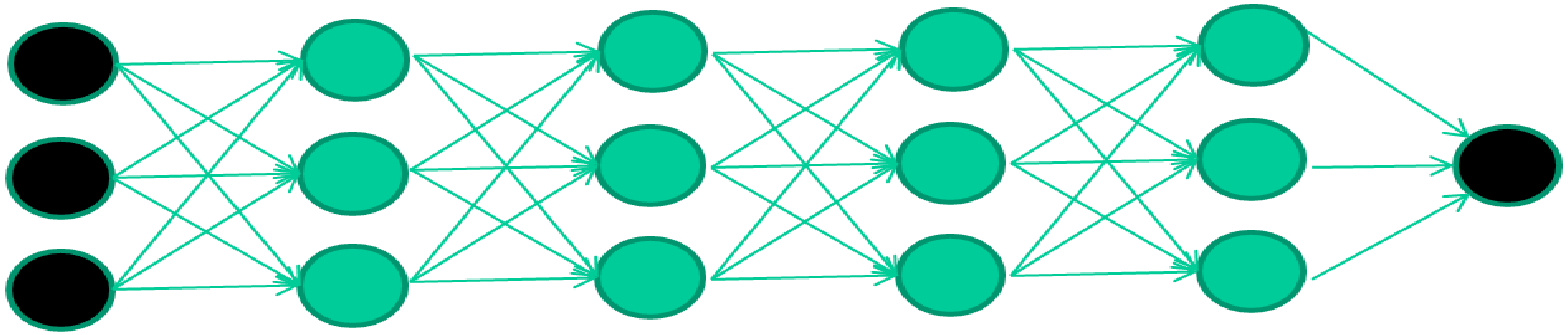
etc.

https://www.macs.hw.ac.uk/~dwcorne/Teaching/**introdl.ppt**

A dataset

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

Train the deep neural network

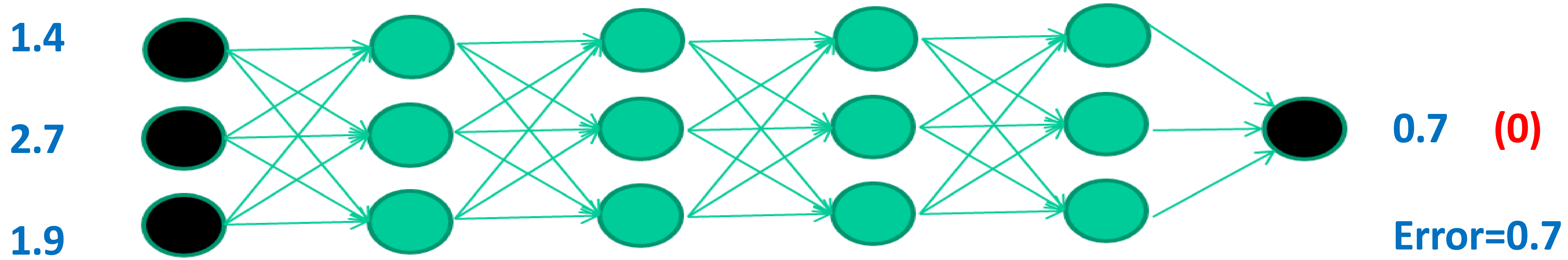


https://www.macs.hw.ac.uk/~dwcorne/Teaching/introdl.ppt

A dataset

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

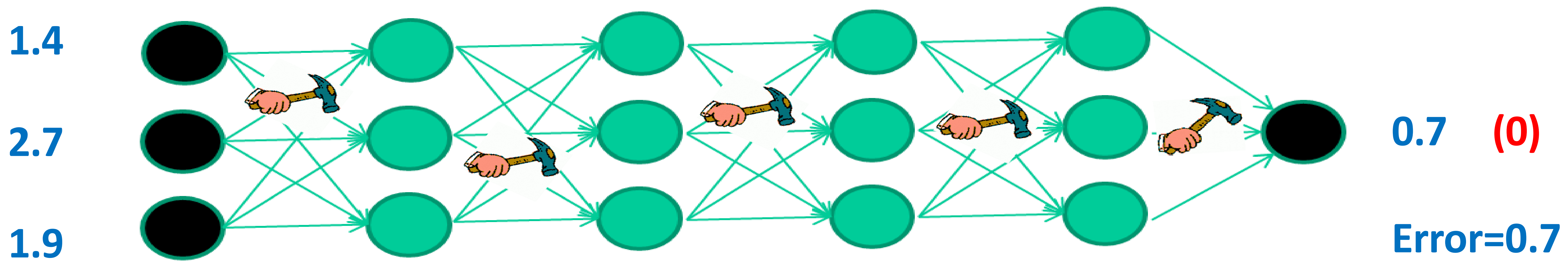
Initialize with random weights



Compare with the target output

https://www.macs.hw.ac.uk/~dwcorne/Teaching/introdl.ppt

Adjust weights based on error



Repeat this thousands, maybe millions of times – each time taking a random training instance, and making slight weight adjustments

Algorithms for weight adjustment are designed to make changes that will reduce the error

https://www.macs.hw.ac.uk/~dwcorne/Teaching/introdl.ppt

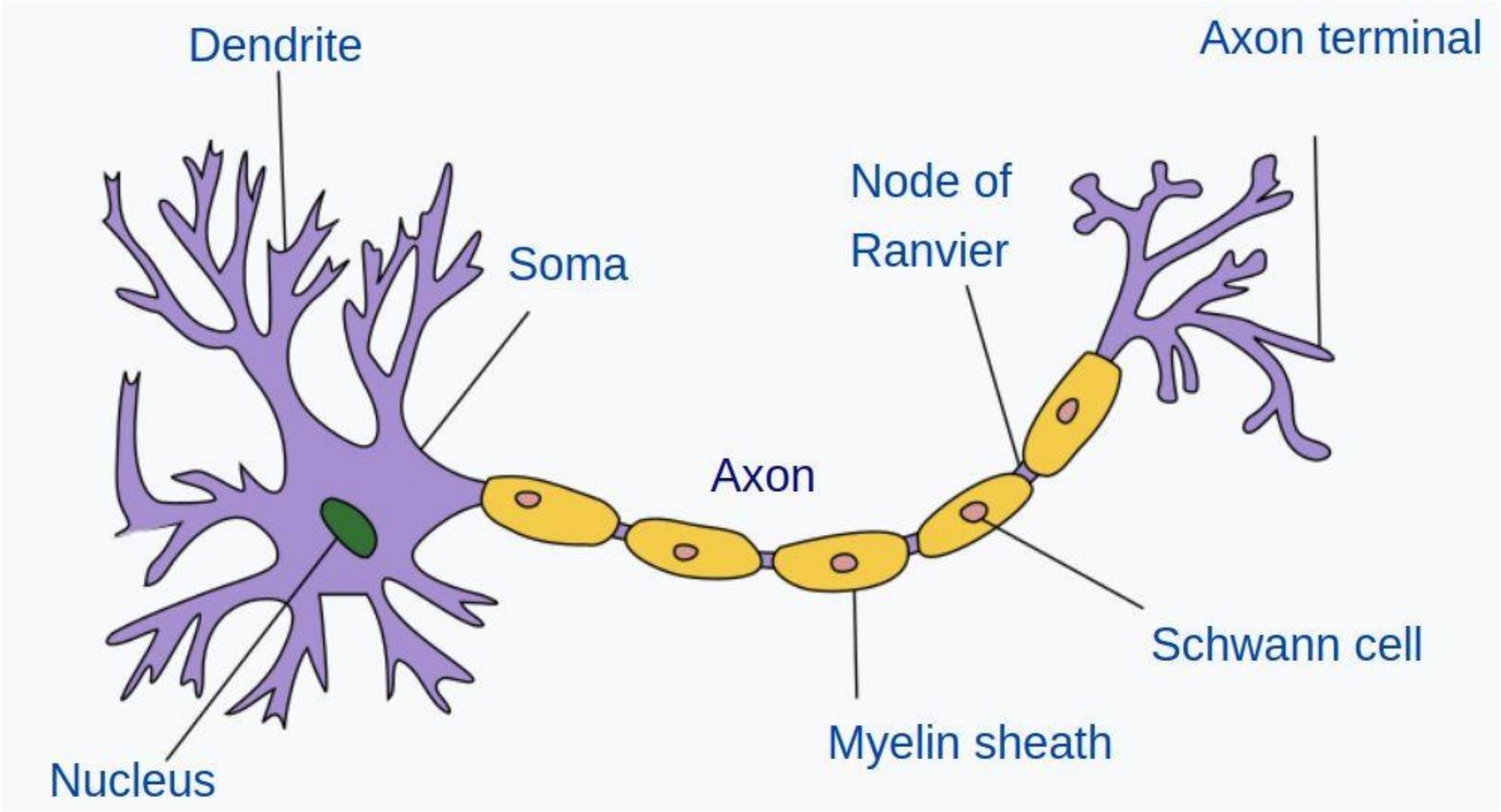
What is Deep Learning?

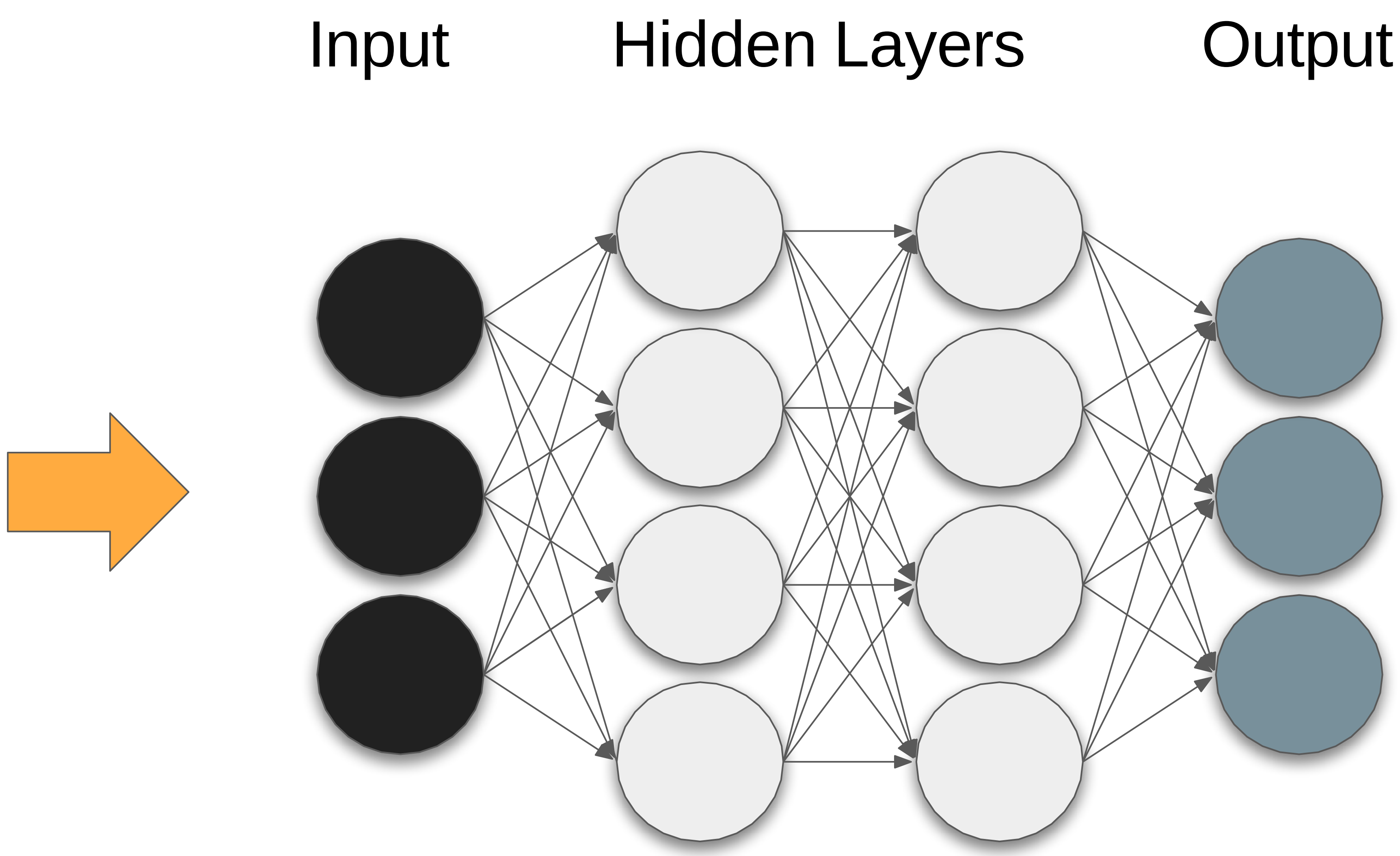
Deep learning is a class of machine learning algorithms that:

- use a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input.
- learn in supervised (e.g., classification) and/or unsupervised (e.g., pattern analysis) manners.
- learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.

(Source: Wikipedia)

Artificial Neural Network

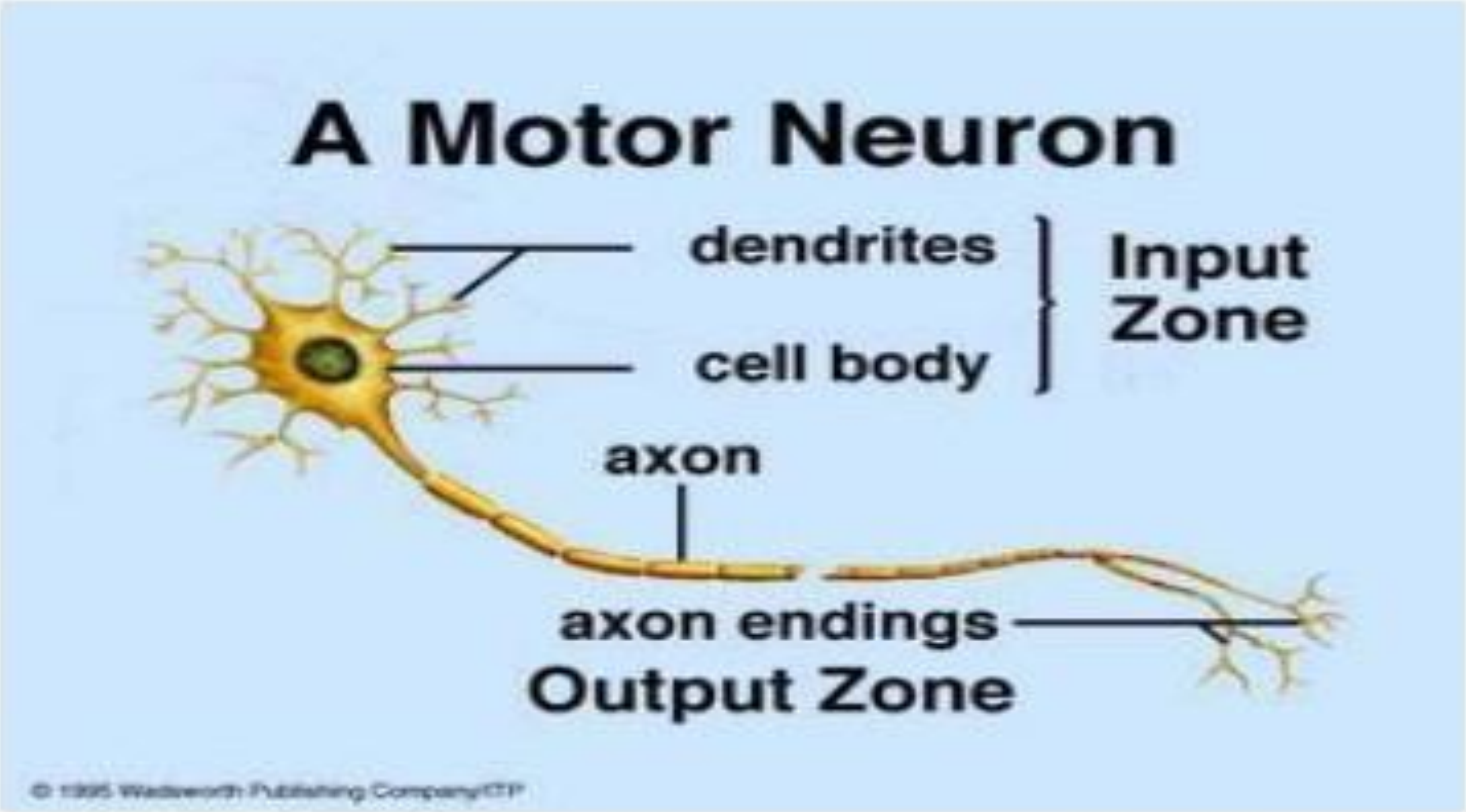




(Image Credit: Wikipedia)

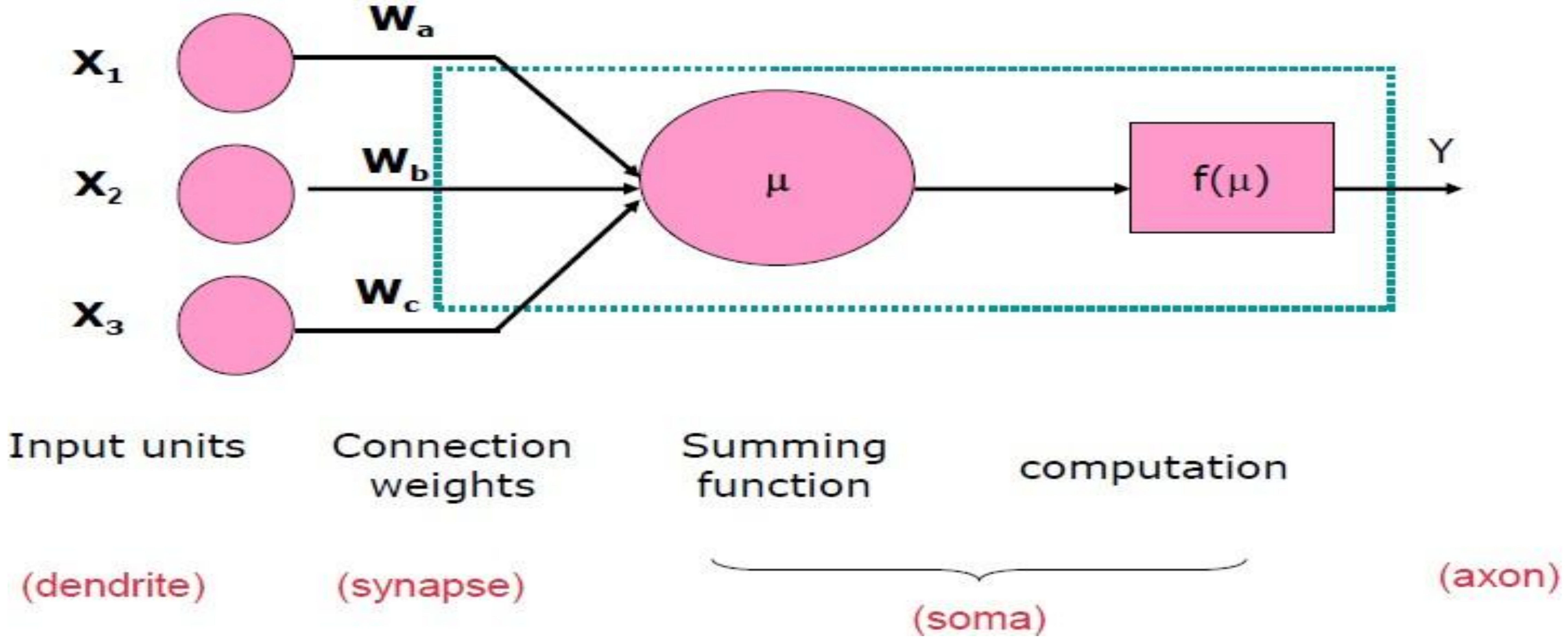
Biological Neural Networks

- A biological neuron has three types of main components; dendrites, soma (or cell body) and axon.
- Dendrites receives signals from other neurons.

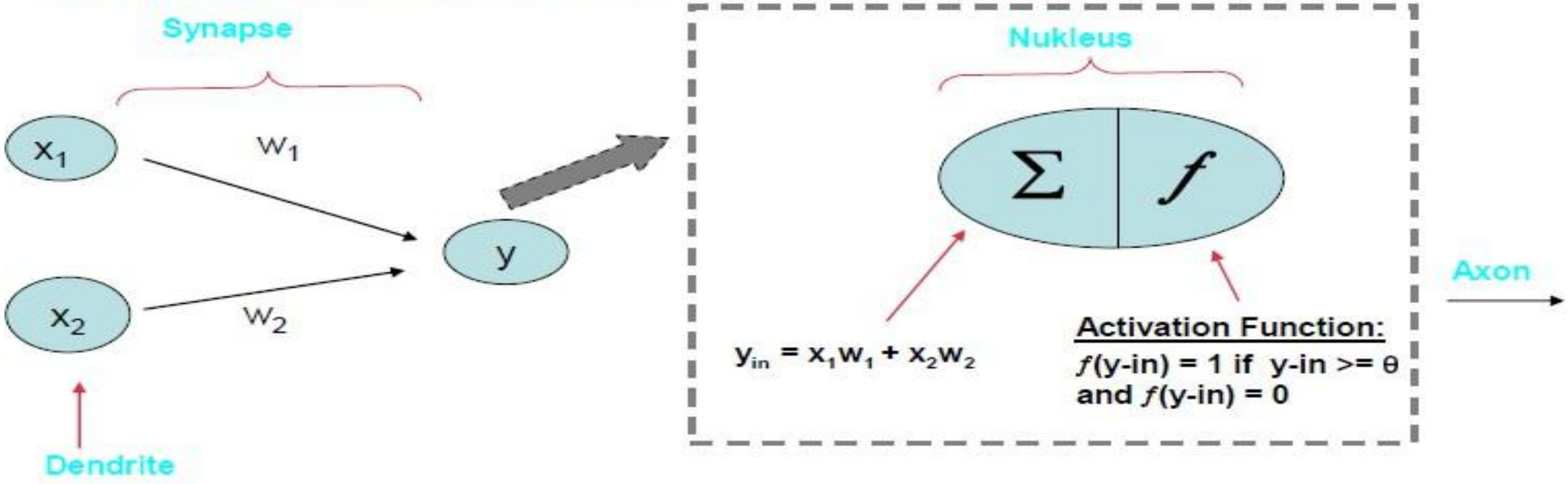


- The soma, sums the incoming signals. When sufficient input is received, the cell fires; that is it transmit a signal over its axon to other cells.

Model Of A Neuron



Artificial Neural Network

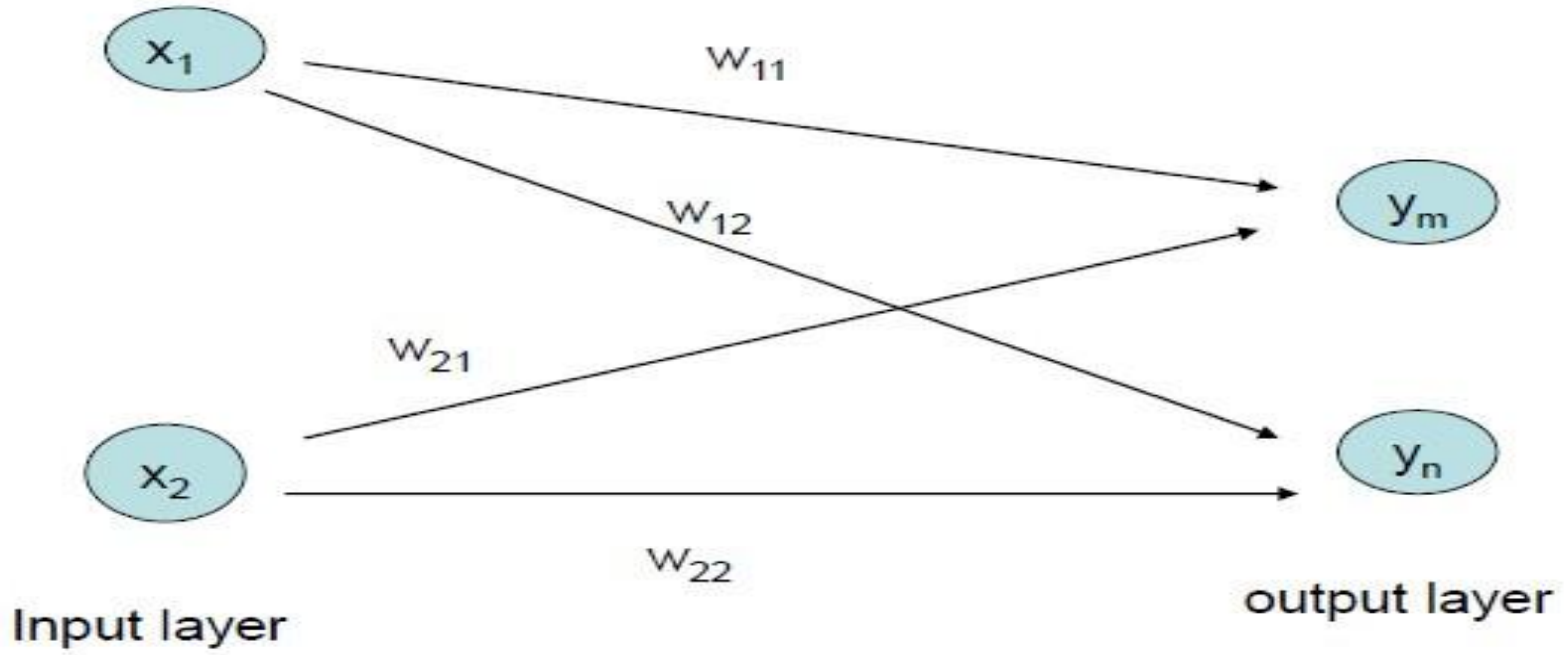


-A neuron receives input, determines the strength or the weight of the input, calculates the total weighted input, and compares the total weighted with a value (threshold)

-The value is in the range of 0 and 1

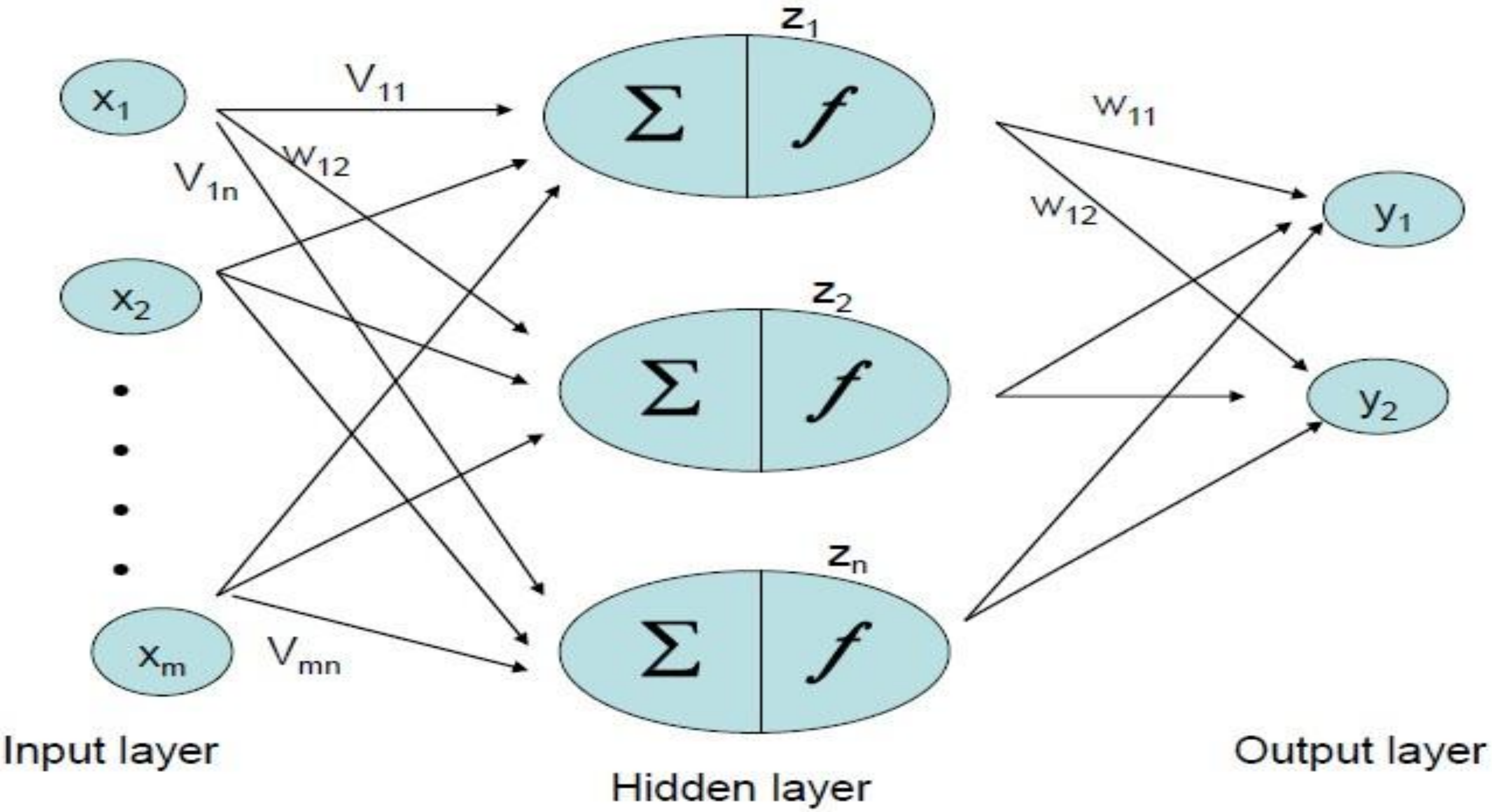
- If the total weighted input greater than or equal the threshold value, the neuron will produce the output, and if the total weighted input less than the threshold value, no output will be produced

Single Layer Feedforward NN



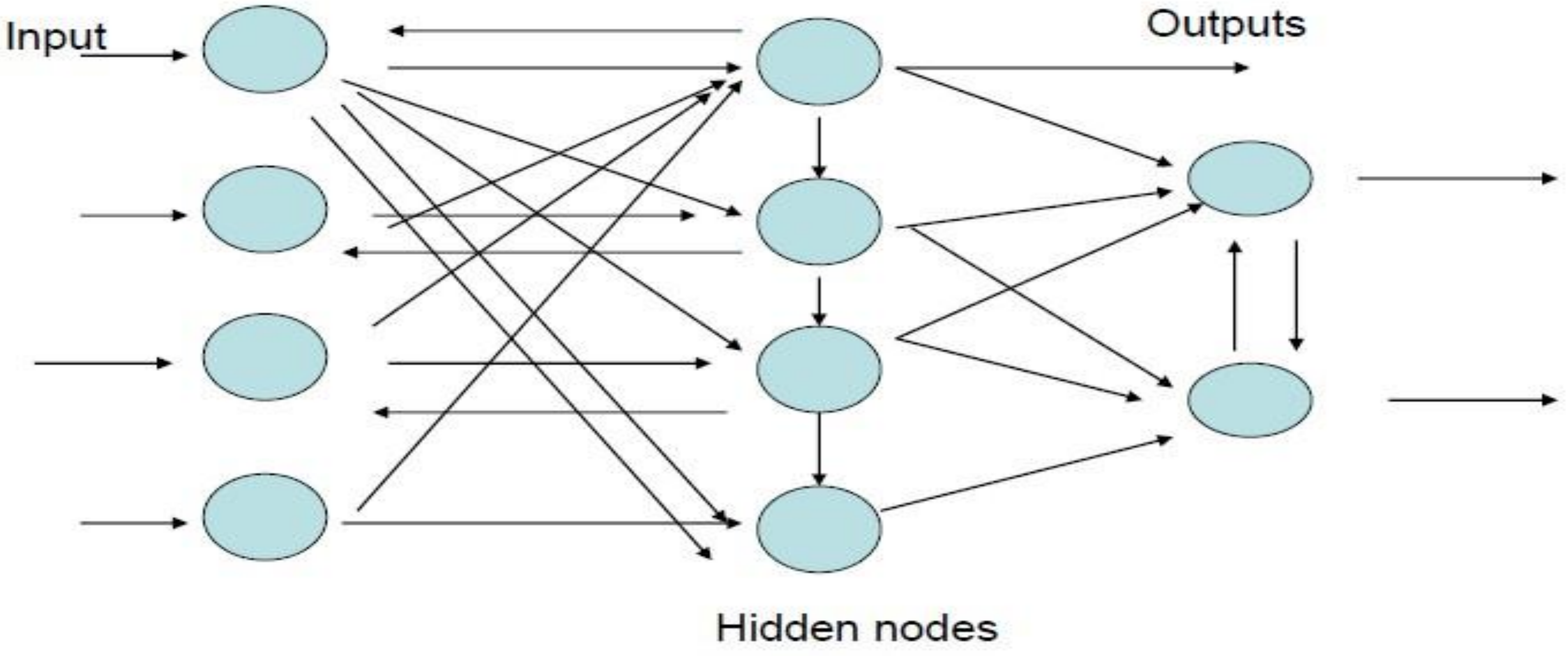
Contoh: **ADALINE, AM, Hopfield, LVQ, Perceptron, SOFM**

Multilayer Neural Network



Contoh: **CCN, GRNN, MADALINE, MLFF with BP, Neocognitron, RBF, RCE**

Recurrent NN



Contoh: **ART, BAM, BSB, Boltzman Machine, Cauchy Machine, Hopfield, RNN**

Inputs and Outputs

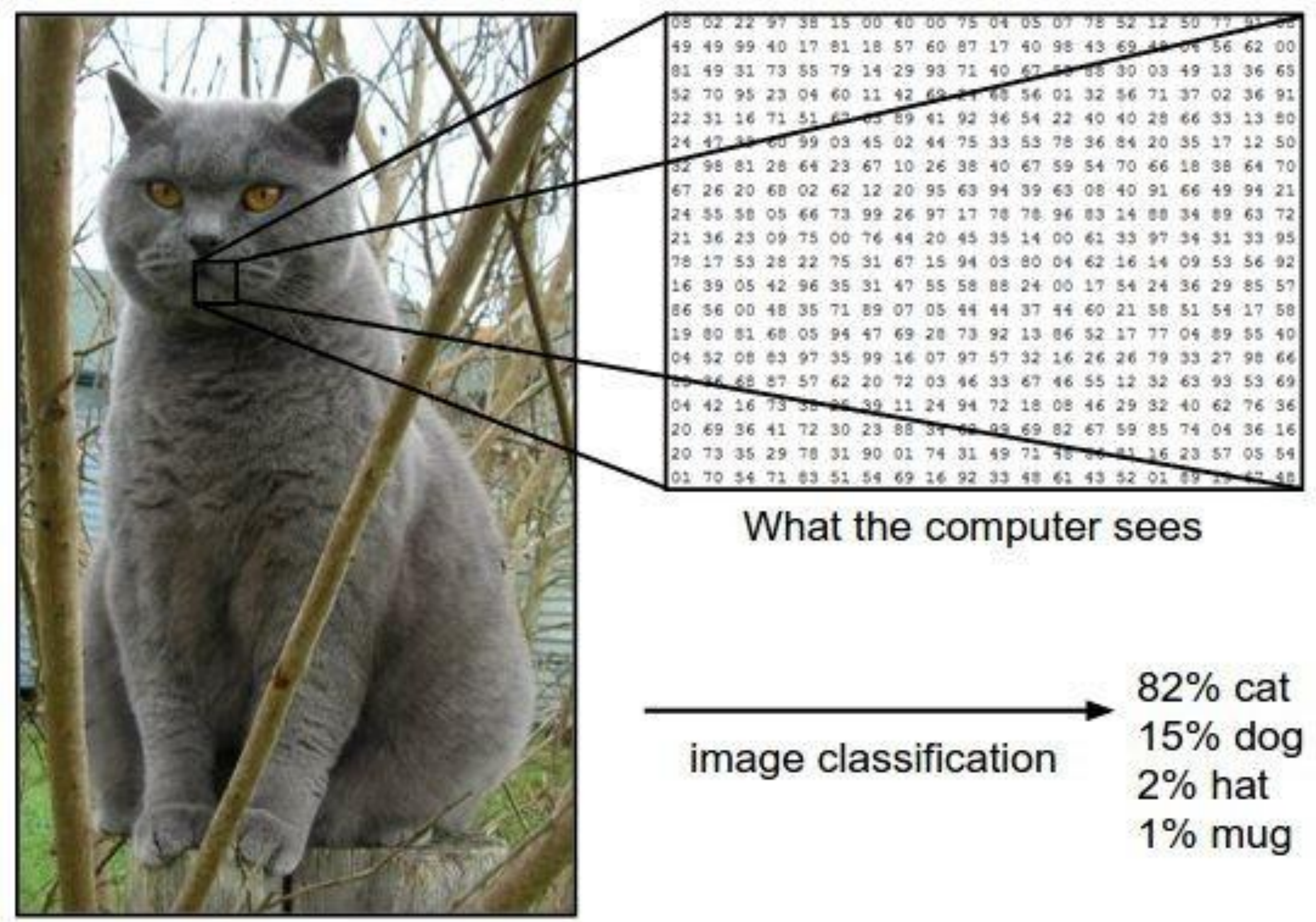
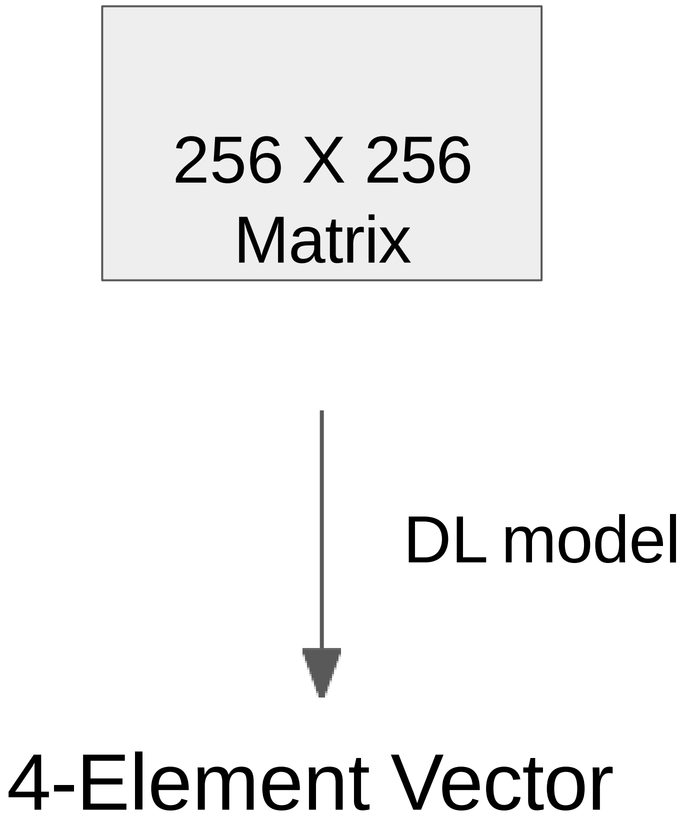
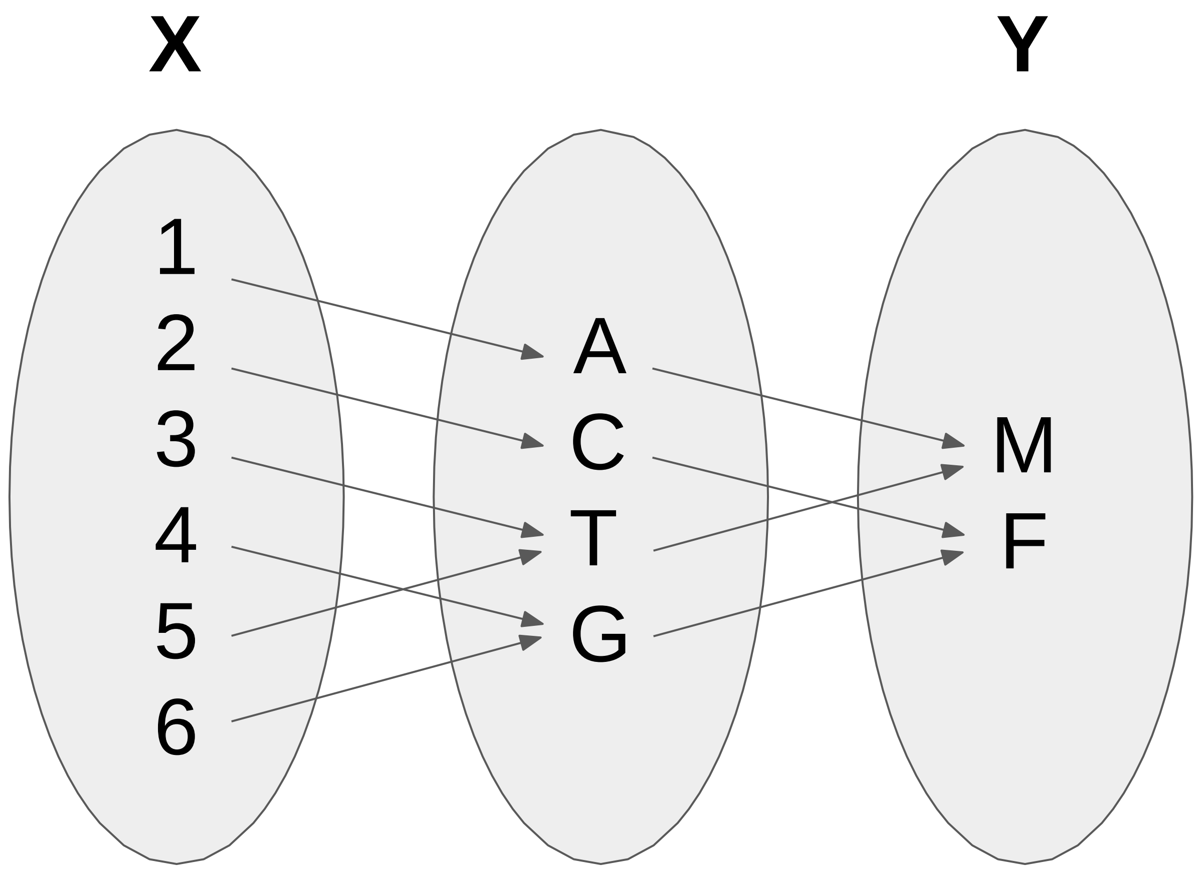


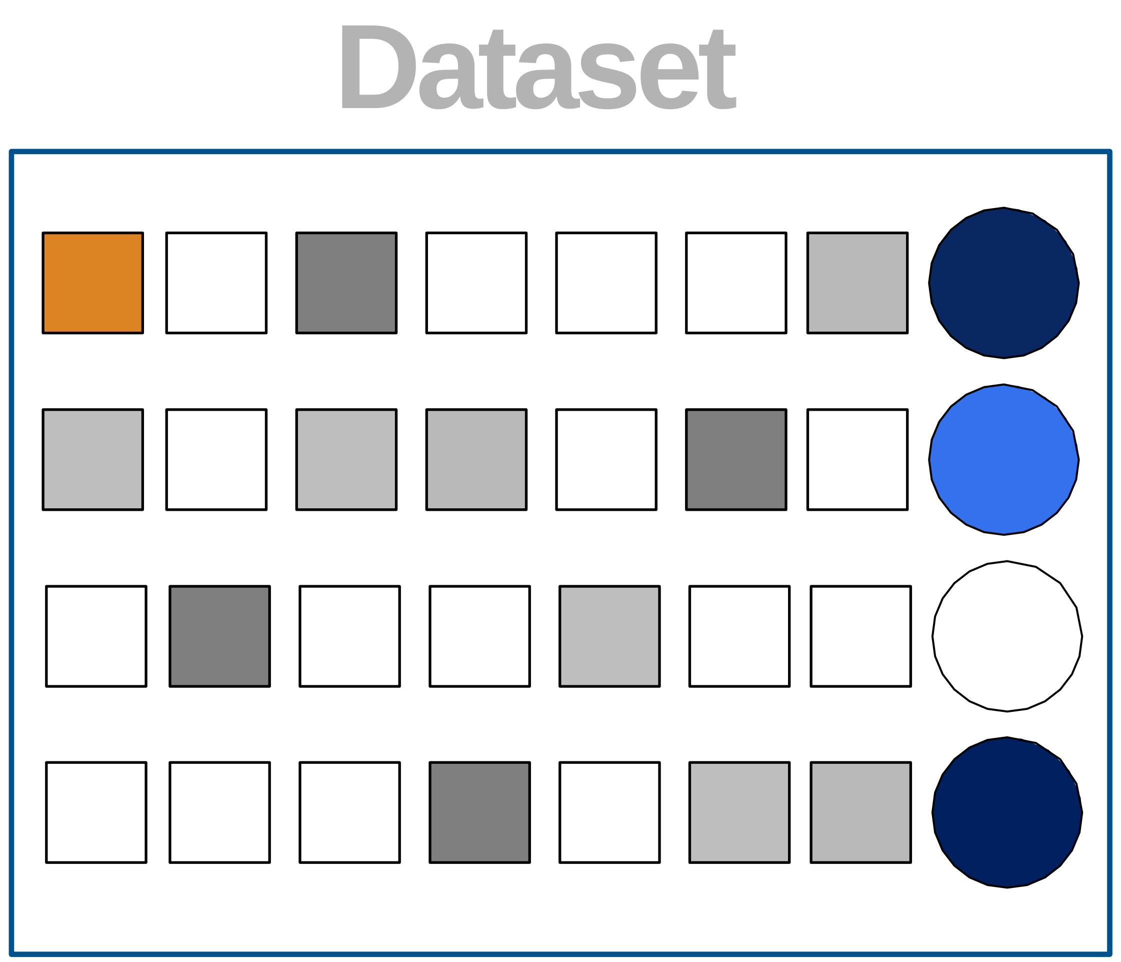
Image from the Stanford CS231 Course

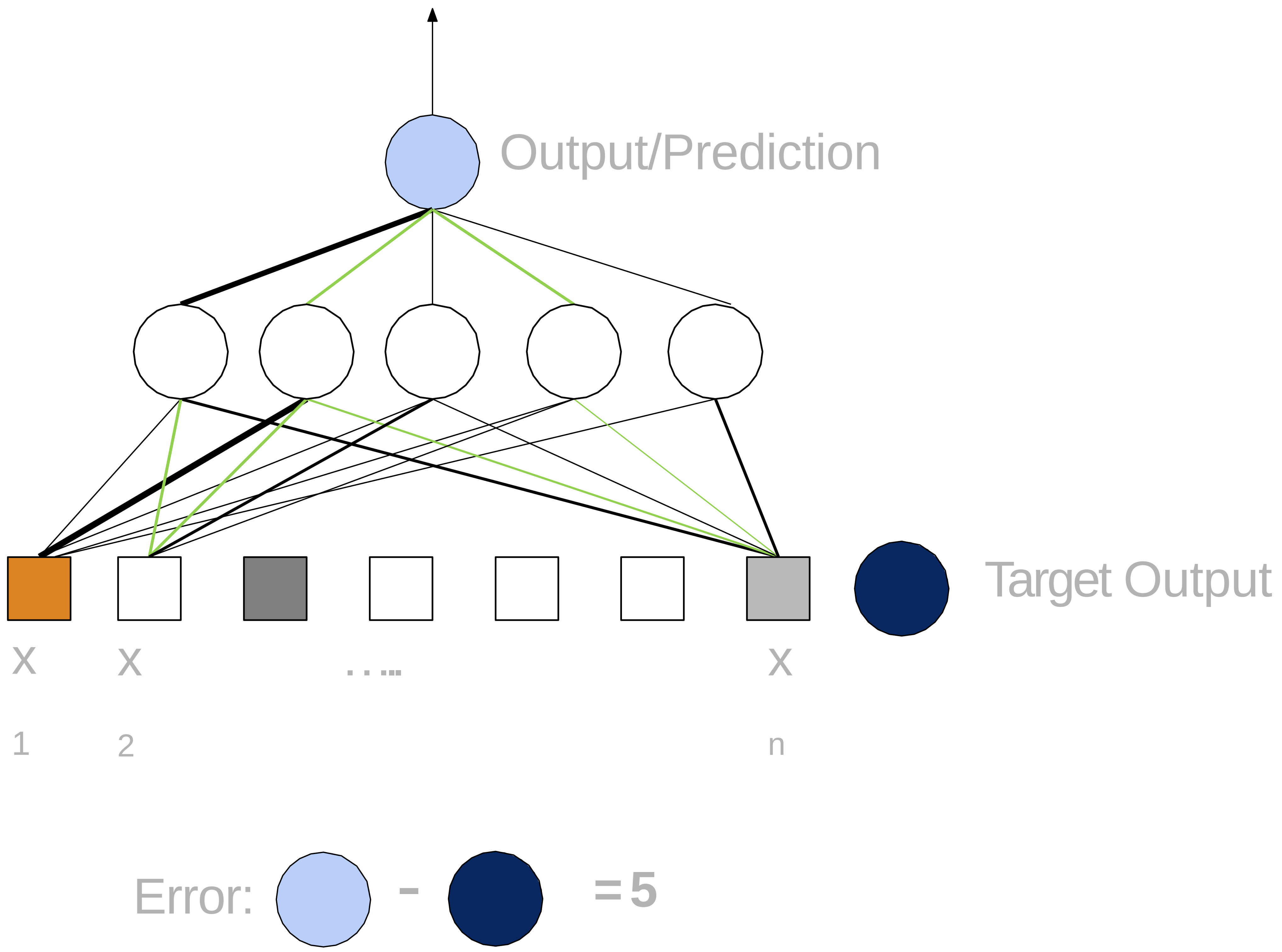




With deep learning, we are searching for a **surjective** (or **onto**) function **f** from a set **X** to a set **Y**.

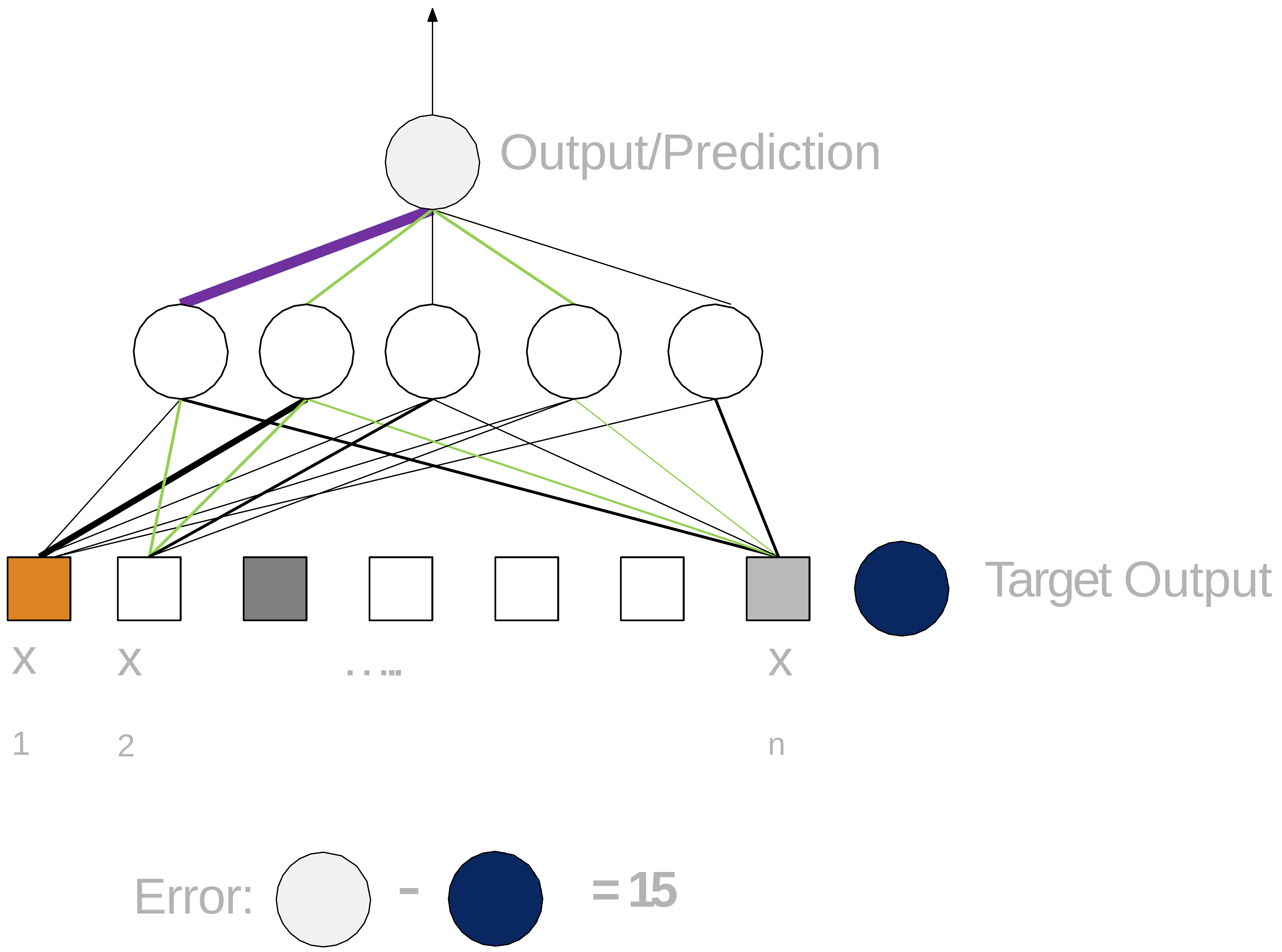
Learning Principle





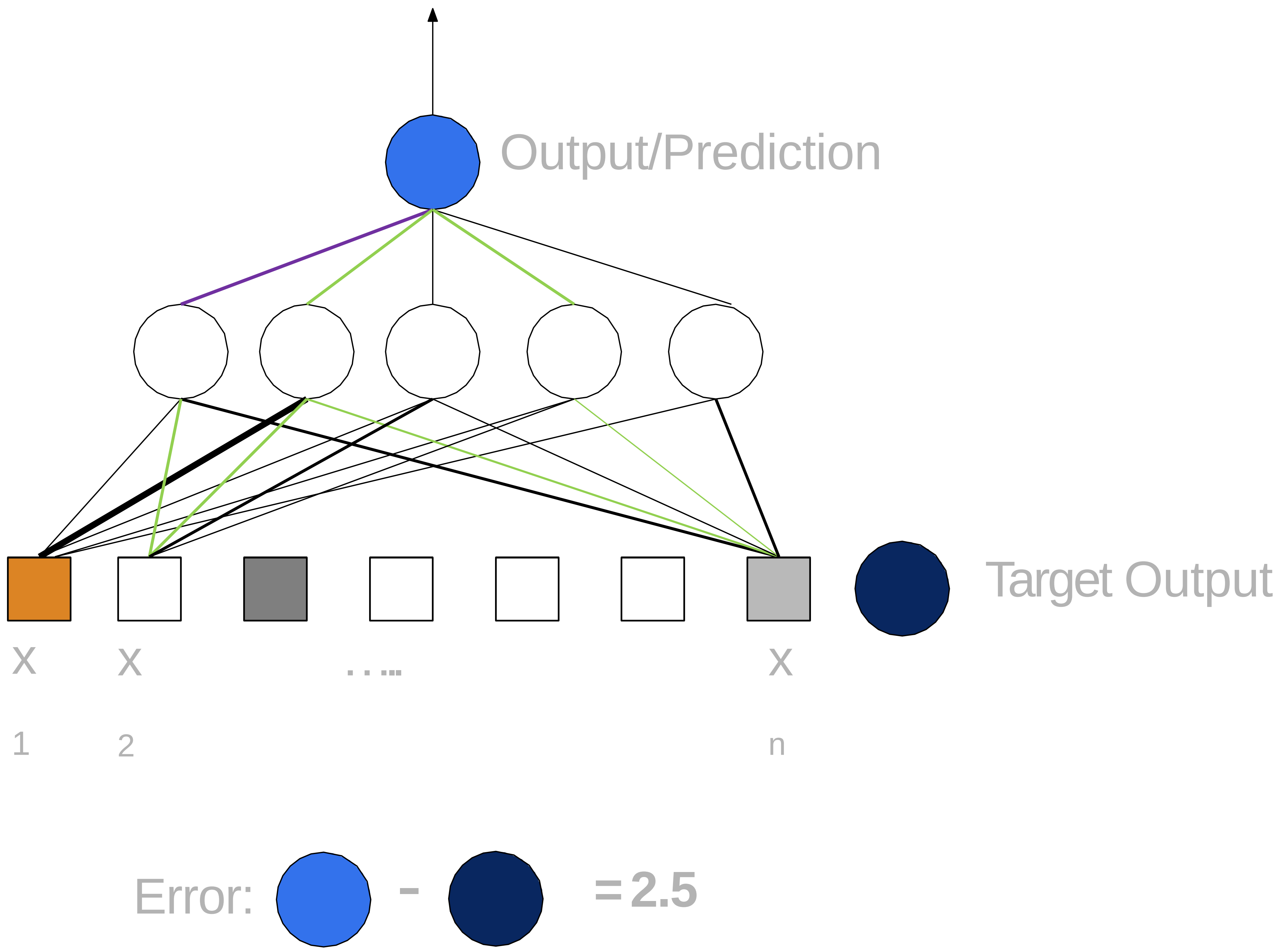
(Image Credit: NVIDIA Deep Learning Institute)

Learning Principle



(Image Credit: NVIDIA Deep Learning Institute)

Learning Principle



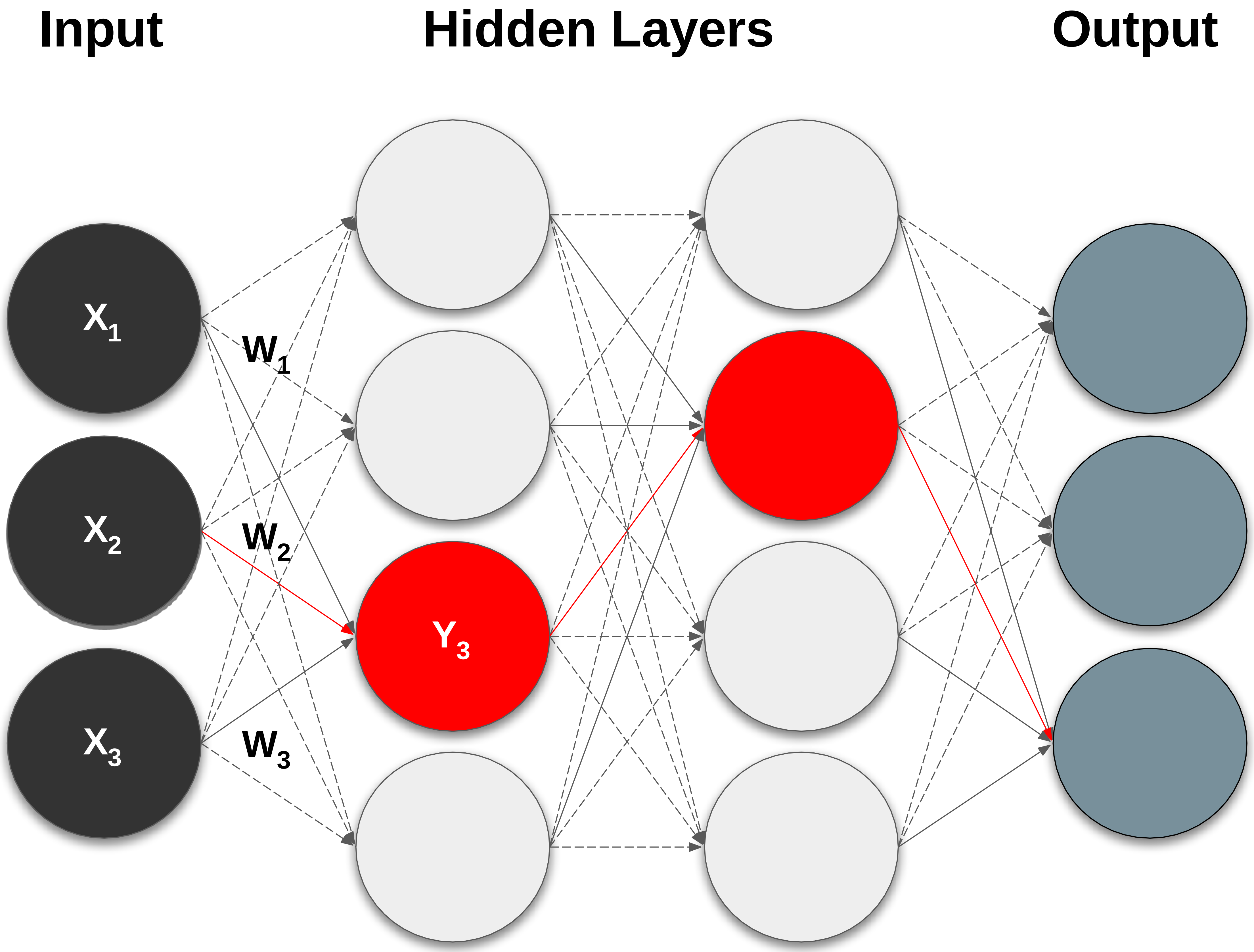
(Image Credit: NVIDIA Deep Learning Institute)

Supervised Deep Learning with Neural Networks

From one layer to the next

$$Y_j = f\left(\sum_i W_i X_i + b_i\right)$$

f is the activation function, W_i is the weight, and b_i is the bias.



Training - Minimizing the Loss

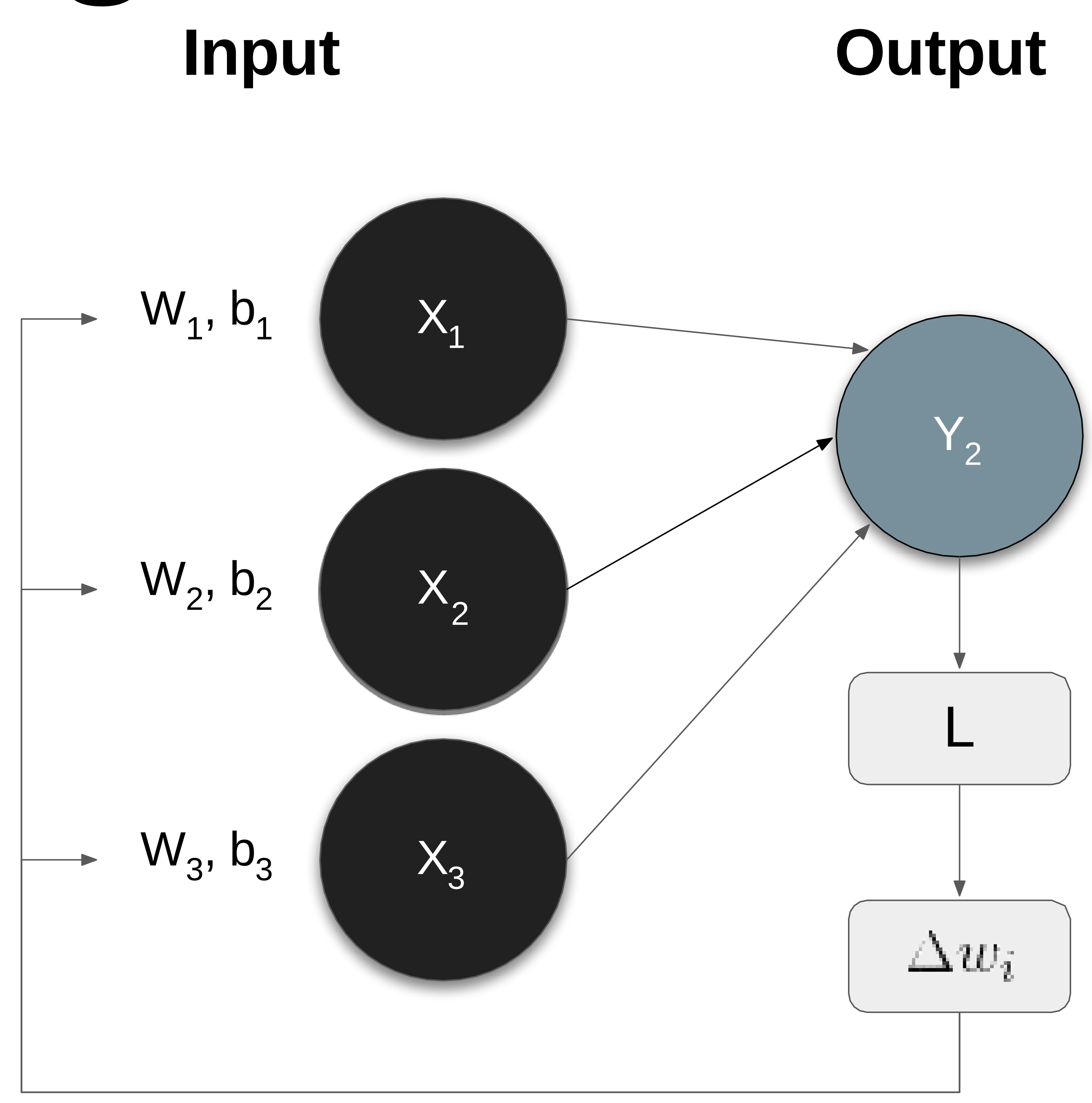
The loss function with regard to weights and biases can be defined as

$$L(\mathbf{w}, \mathbf{b}) = \frac{1}{2} \sum_i (\mathbf{Y}(\mathbf{X}, \mathbf{w}, \mathbf{b}) - \mathbf{Y}'(\mathbf{X}, \mathbf{w}, \mathbf{b}))^2$$

The weight update is computed by moving a step to the opposite direction of the cost gradient.

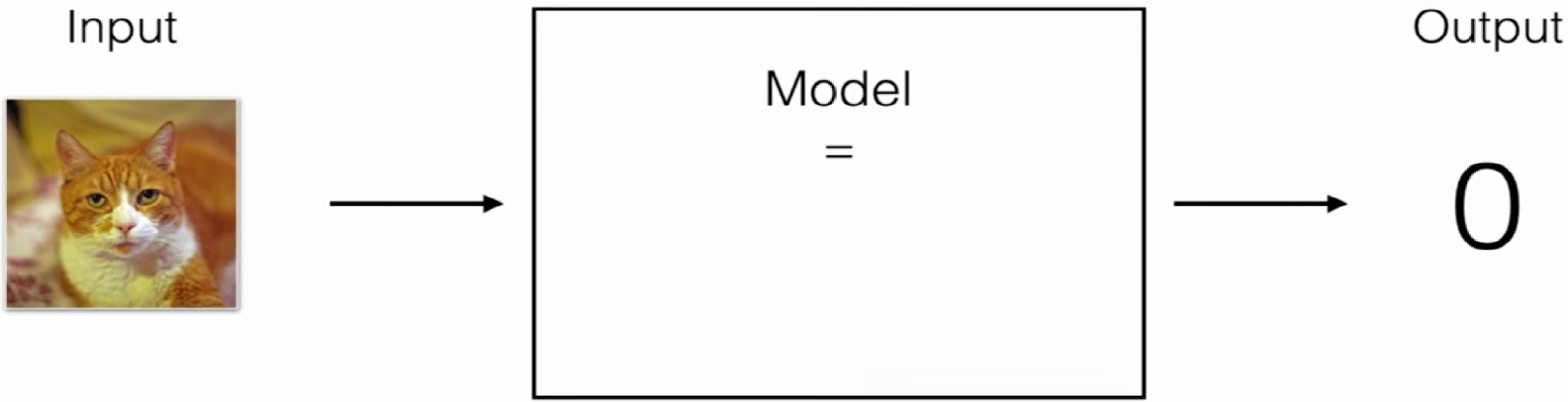
$$\Delta w_i = -\alpha \frac{\partial L}{\partial w_i}$$

Iterate until L stops decreasing.



Deep Learning

Learning Process



Deep Learning

Learning Process

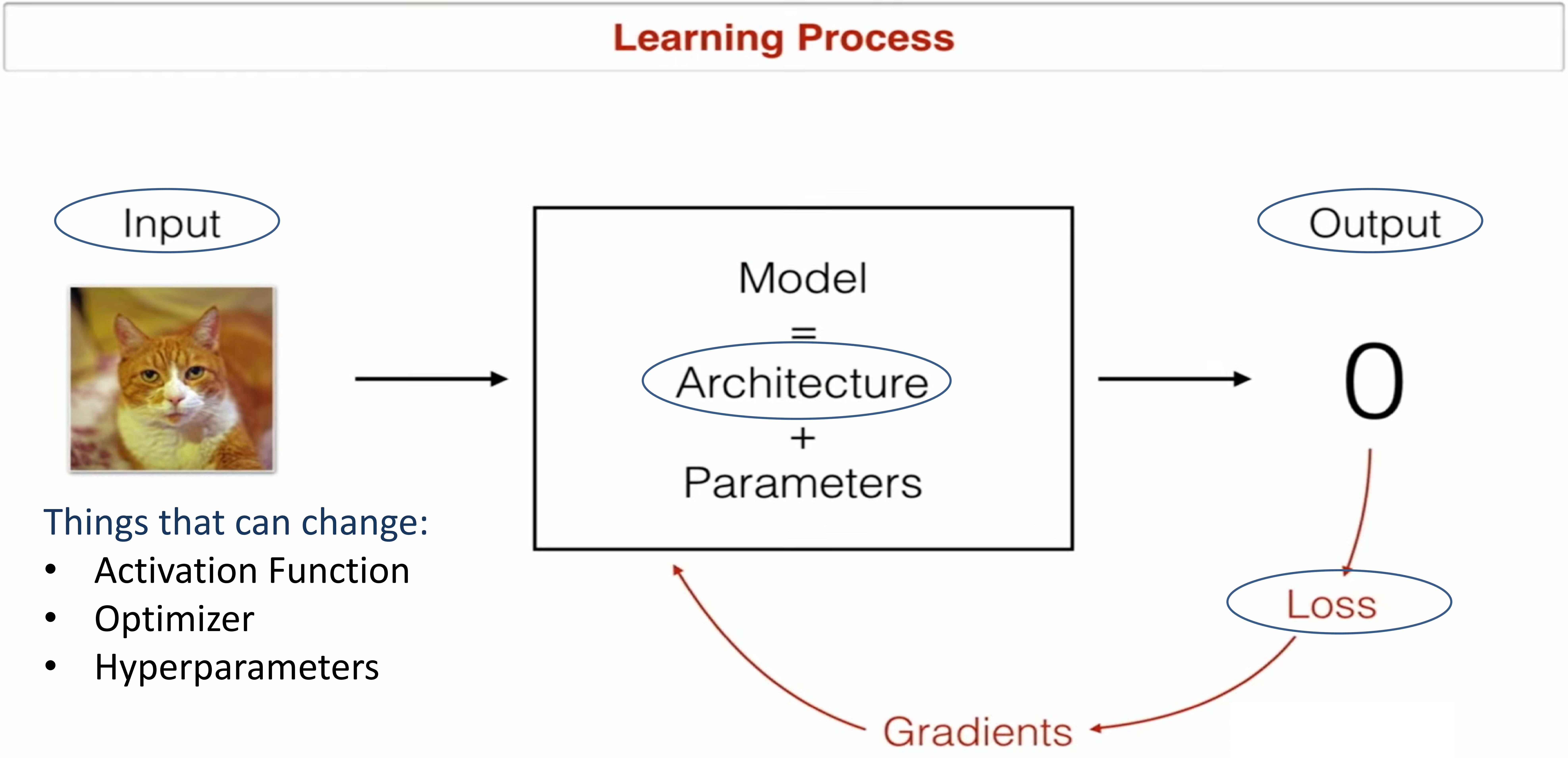
Input

Model
=
Architecture
+
Parameters

Output

0

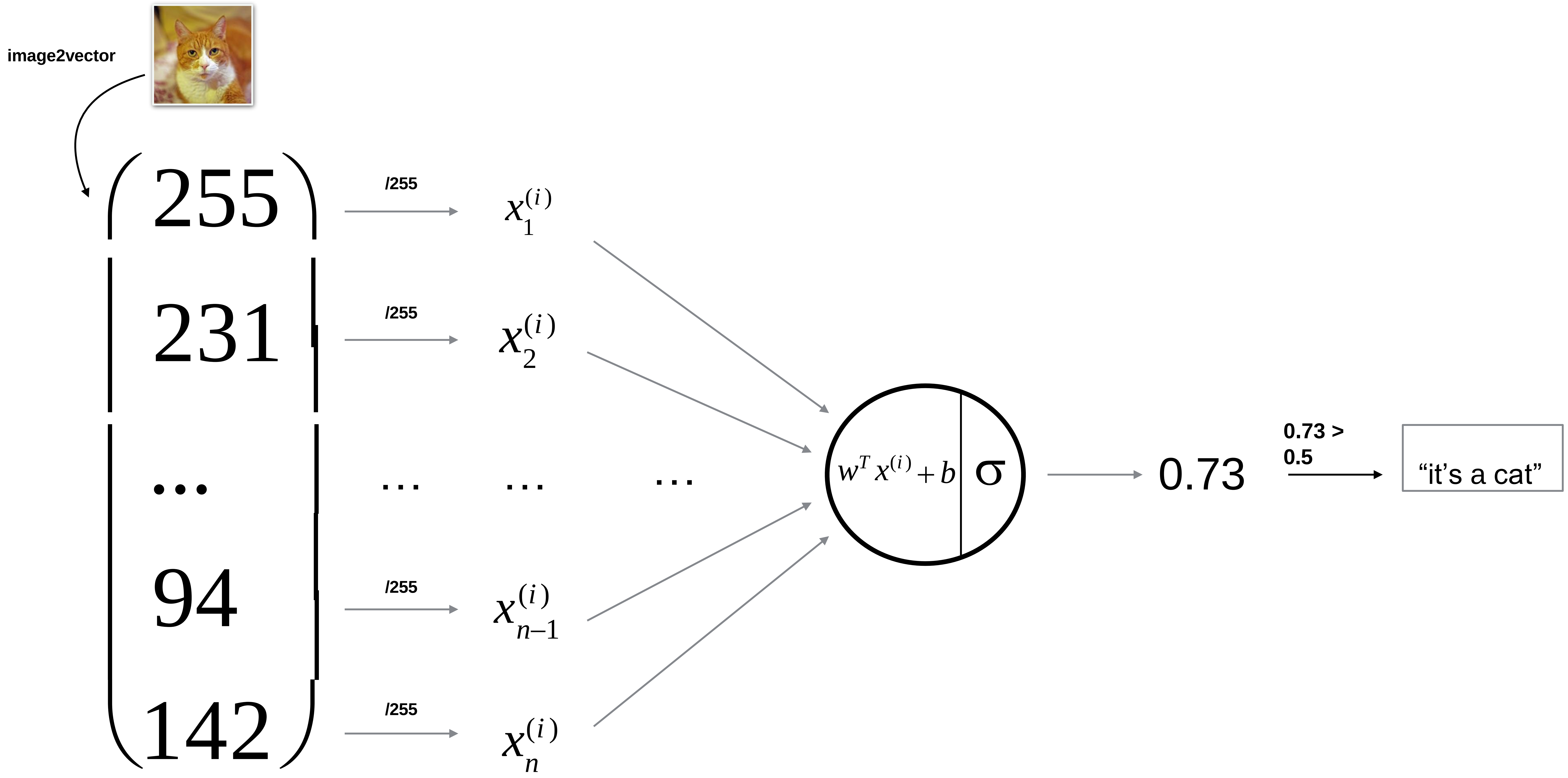
Deep Learning



Things that can change:

- Activation Function
- Optimizer
- Hyperparameters

Logistic Regression as a Neural Network



Multi-class

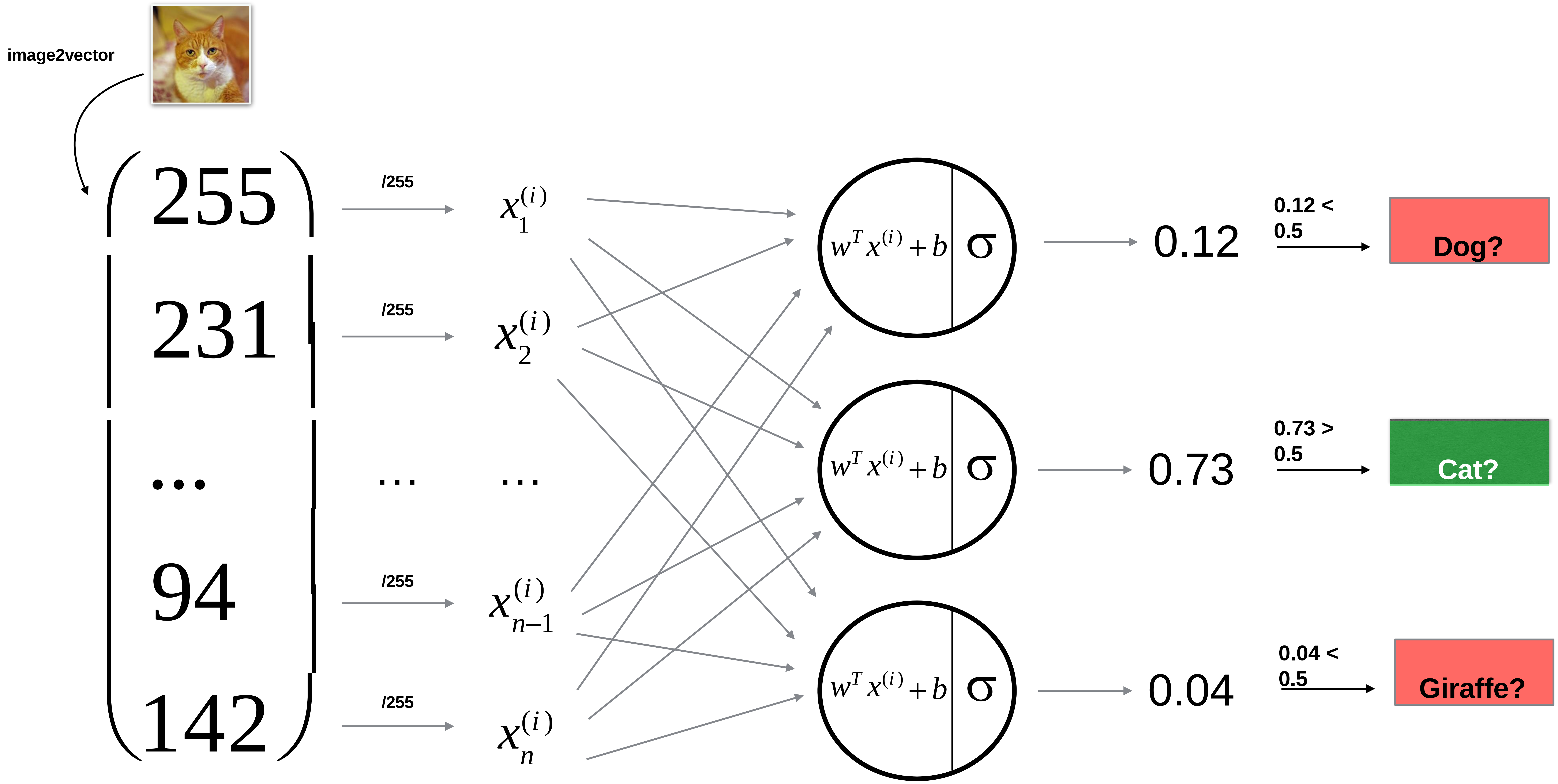
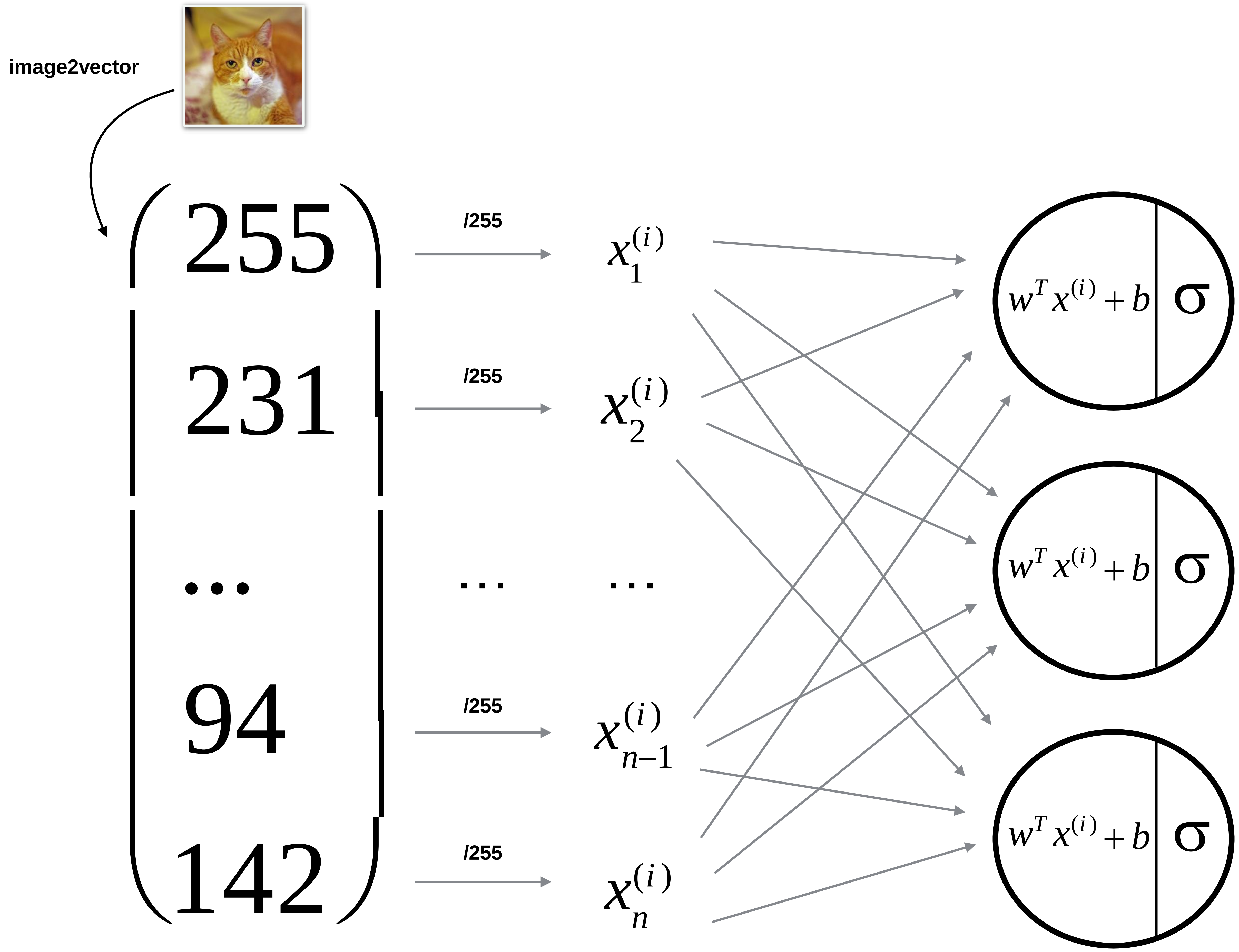
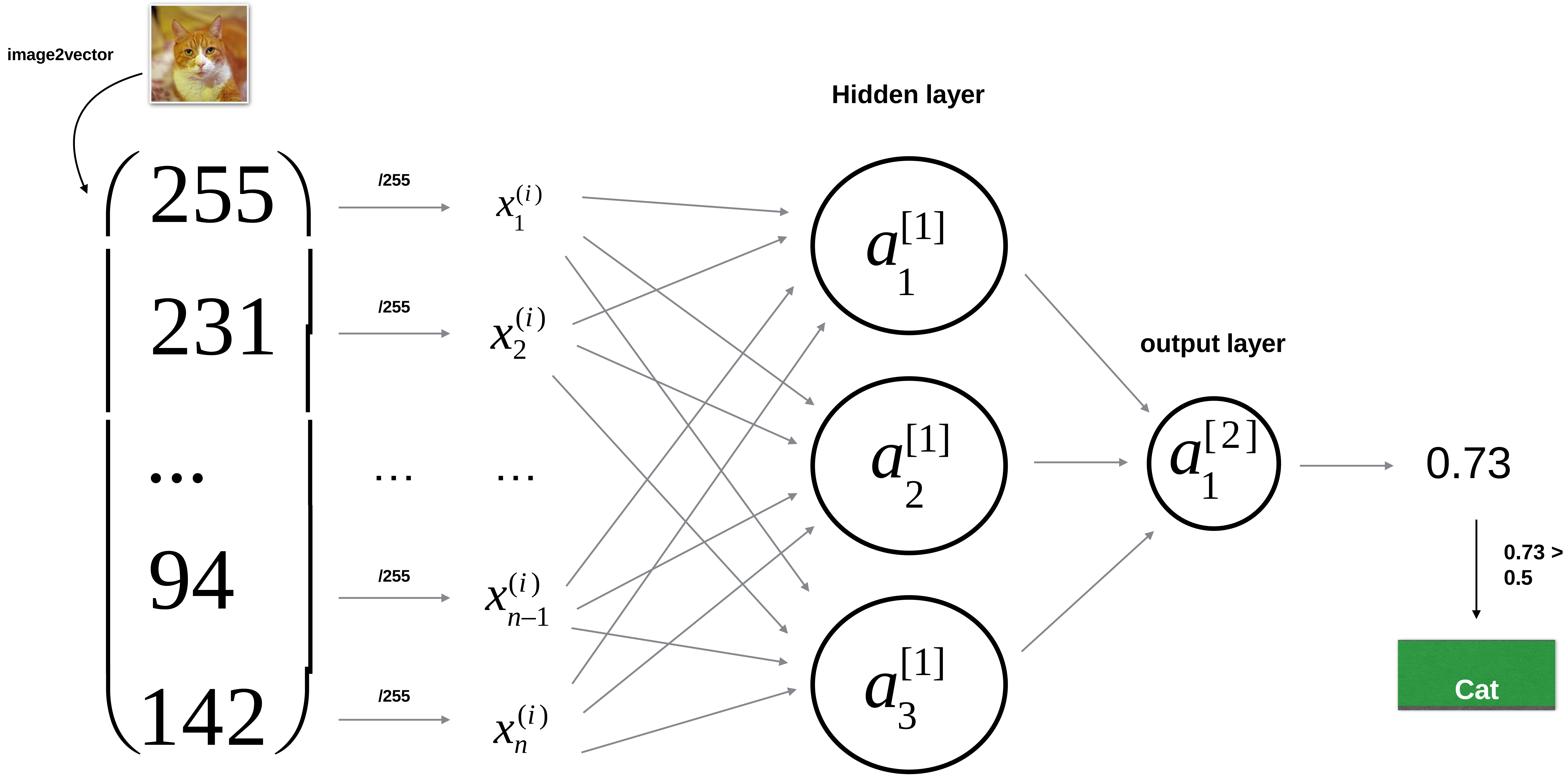


image2vector
255
231
...
94
142
/255
/255
/255
/255
$x_1^{(i)}$
$x_2^{(i)}$
...
...
$x_{n-1}^{(i)}$
$x_n^{(i)}$
$w^T x^{(i)} + b$ | σ
$w^T x^{(i)} + b$ | σ
$w^T x^{(i)} + b$ | σ
0.12
0.12 < 0.5
Dog?
0.73
0.73 > 0.5
Cat?
0.04
0.04 < 0.5
Giraffe?

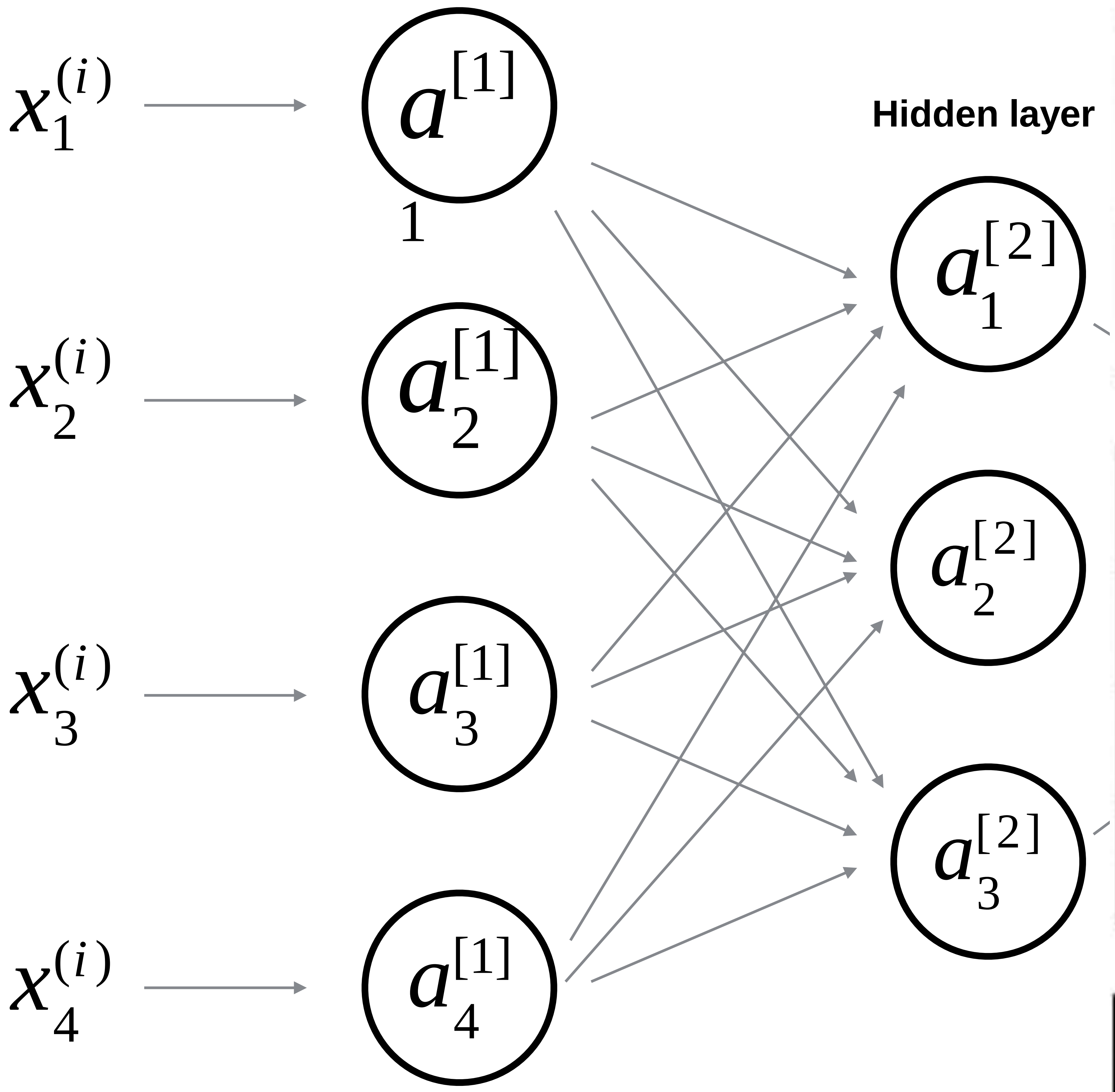
Neural Network (Multi-class)

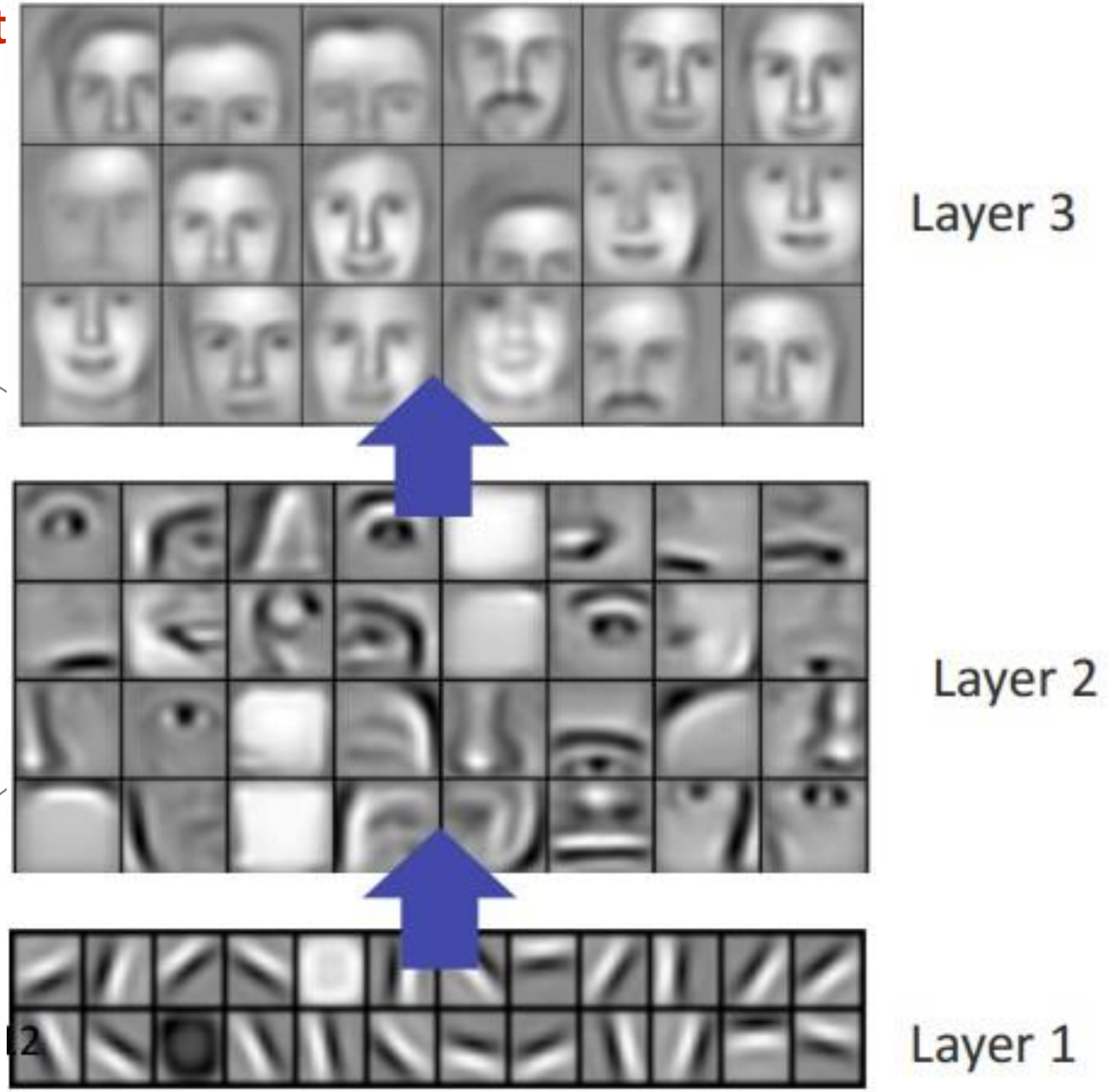


Neural Network (1 hidden layer)



Deeper net





Technique called “encoding”

Summary from this learning

- A **model** is defined by its **architecture** and its **parameters**.
- The labelling strategy matters to successfully train your models. For example, if you're training a 3-class (dog, cat, giraffe) classifier under the constraint of one animal per picture, you might use **one-hot vectors** to label your data.
- We introduced a set of **notations** to differentiate indices for neurons, layers and examples.
- In deep learning, **feature learning** replaces **feature engineering**.

Case study A: Day ’n’ Night classification

Goal: Given an image, classify as taken “during the day” (0) or “during the night” (1)

1. **Data?** 10,000 images Split? Bias?

2. **Input?**  Resolution? (64, 64, 3)

3. **Output?** y = 0 or y = 1 Last Activation? sigmoid

4. **Architecture ?** A shallow CNN should do the job pretty well

5. **Loss?** $$L = -[y\log(\hat{y}) + (1-y)\log(1-\hat{y})]$$

Case study B: Face Verification

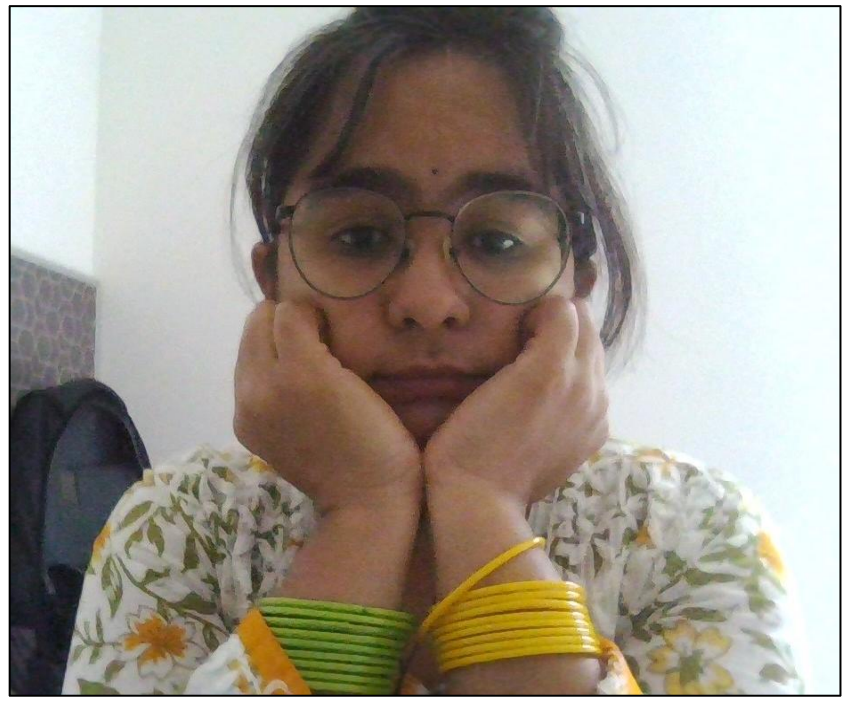
Goal: A school wants to use Face Verification for validating student IDs in facilities (dinning halls, gym, pool …)

1. **Data?**

Picture of every student labelled with their name

Ashu

2. **Input?**



Resolution?

(412, 412, 3)

3. **Output?**

$y = 1$ (it’s you)

or

$y = 0$ (it’s not you)

Case study B: Face Verification

Goal: A school wants to use Face Verification for validating student IDs in facilities (dinning halls, gym, pool …)

4. What architecture?

Simple solution:

database image

compute distance pixel per pixel

if less than threshold then y=1

input image

Issues:

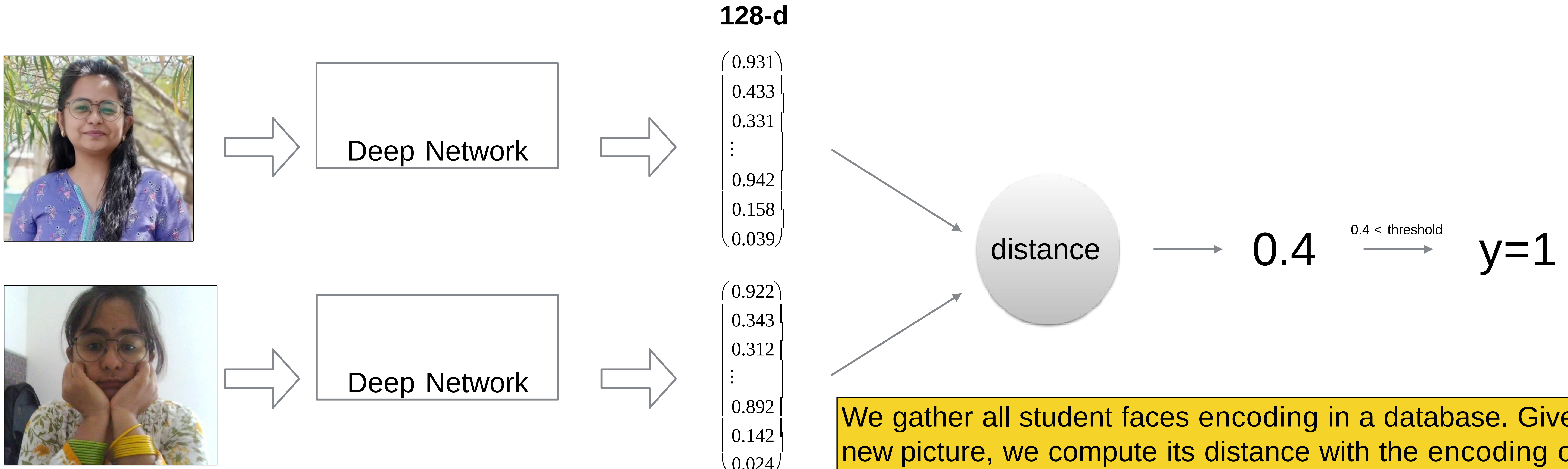
- Background lighting differences
- A person can wear make-up, grow a beard…
- ID photo can be outdated

Case study B: Face Verification

Goal: A school wants to use Face Verification for validating student IDs in facilities (dinning halls, gym, pool …)

4. What architecture?

Our solution: encode information about a picture in a vector



We gather all student faces encoding in a database. Given a new picture, we compute its distance with the encoding of card holder

Case study B: Face Verification

Goal: A school wants to use Face Verification for validating student IDs in facilities (dinning hall, gym, pool …)

4. Loss? Training?

We need more data so that our model understands how to encode:
Use public face datasets

So let's generate triplets:

anchor positive negative

minimize encoding distance

maximize encoding distance

What we really want:

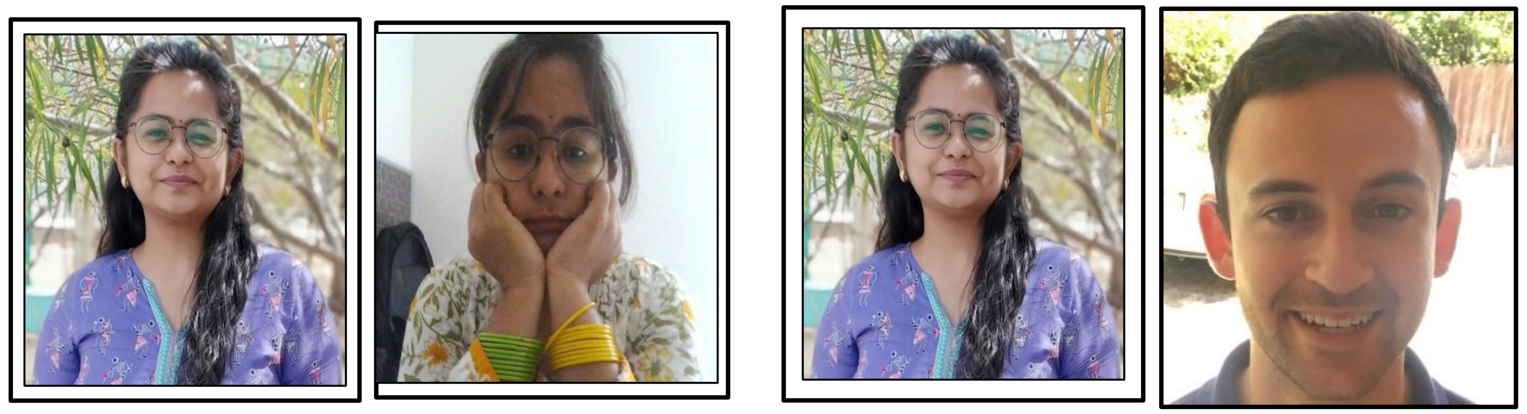
similar encoding

different encoding

[Schroff et al (2015): FaceNet: A Unified Embedding for Face Recognition and Clustering]

Case study B: Face Verification

What we really want:



similar encoding

different encoding

So let's generate triplets:

anchor

positive

negative

minimize encoding distance

maximize encoding distance

Which loss should you minimize?

$$L = \|Enc(A) - Enc(P)\|_2^2 - \|Enc(A) - Enc(N)\|_2^2$$

$$L = \|Enc(A) - Enc(N)\|_2^2 - \|Enc(A) - Enc(P)\|_2^2$$

$$L = \|Enc(P) - Enc(N)\|_2^2 - \|Enc(P) - Enc(A)\|_2^2$$

A B C

[Schroff et al (2015): FaceNet: A Unified Embedding for Face Recognition and Clustering]

Case study B: Face Verification

Input

anchor

positive

negative

Model
=
Architecture
+
Parameters

Output

$$\begin{pmatrix} 0.13 \\ 0.42 \\ .. \\ 0.10 \\ 0.31 \\ 0.73 \\ .. \\ 0.43 \\ 0.33 \end{pmatrix} \quad \begin{pmatrix} 0.01 \\ 0.54 \\ .. \\ 0.45 \\ 0.11 \\ 0.49 \\ .. \\ 0.12 \\ 0.01 \end{pmatrix} \quad \begin{pmatrix} 0.95 \\ 0.45 \\ .. \\ 0.20 \\ 0.41 \\ 0.89 \\ .. \\ 0.31 \\ 0.34 \end{pmatrix}$$

Enc(A) Enc(P) Enc(N)

Loss

$$L = \|Enc(A) - Enc(P)\|_2^2 - \|Enc(A) - Enc(N)\|_2^2 + \alpha$$

Gradients

[Schroff et al (2015): FaceNet: A Unified Embedding for Face Recognition and Clustering]