Machine Learning and Deep Learning

Machine Learning and Deep Learning

What is learning?

- *"Learning denotes changes in a system that ... enable a system to do the same task … more efficiently the next time."* - **Herbert Simon**
- *"Learning is constructing or modifying representations of what is being experienced."* - **Ryszard Michalski**
- *"Learning is making useful changes in our minds."* - **Marvin Minsky**

"Machine learning refers to a system capable of the autonomous acquisition and integration of knowledge."

Architecture of a Learning System

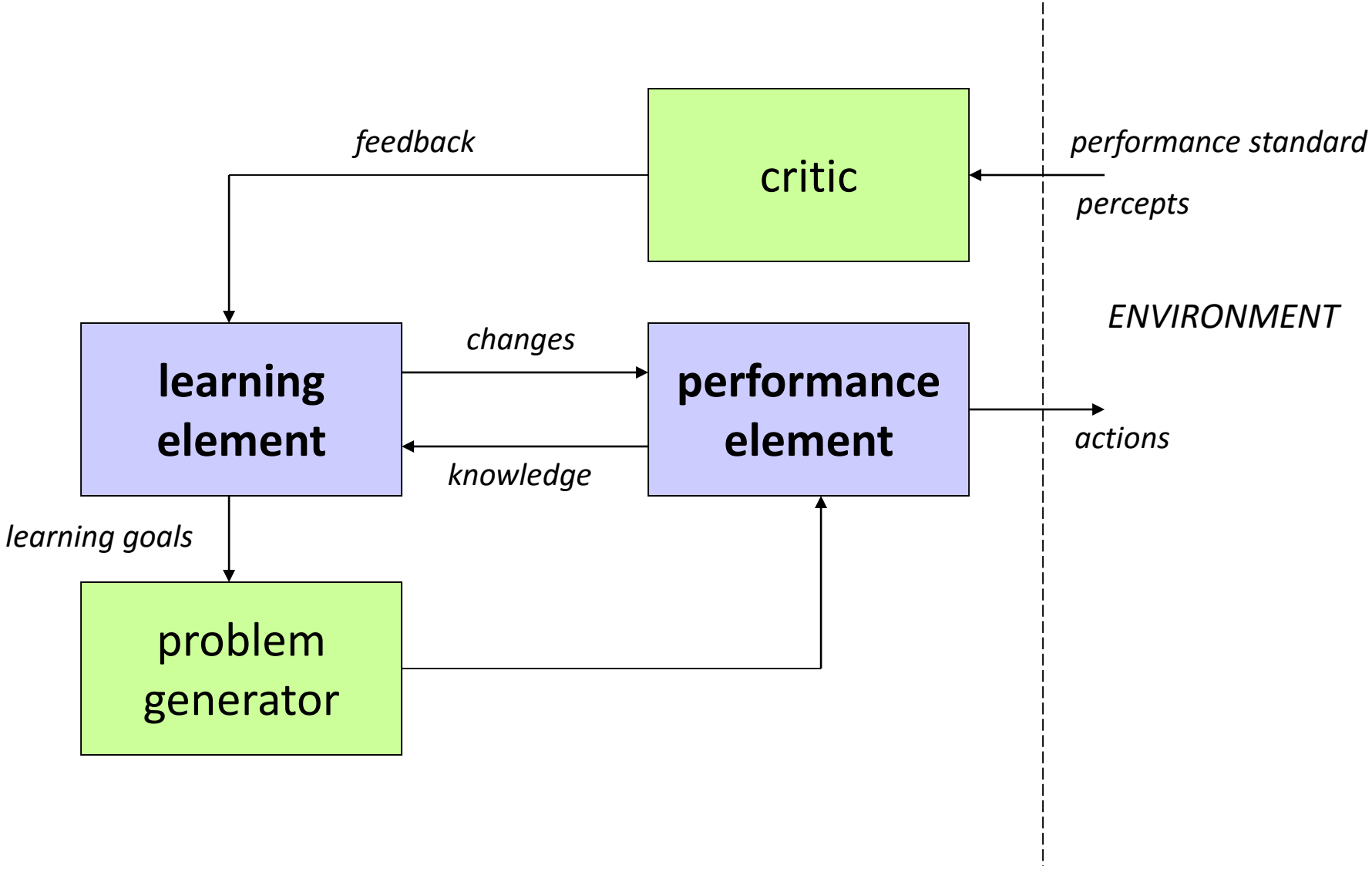
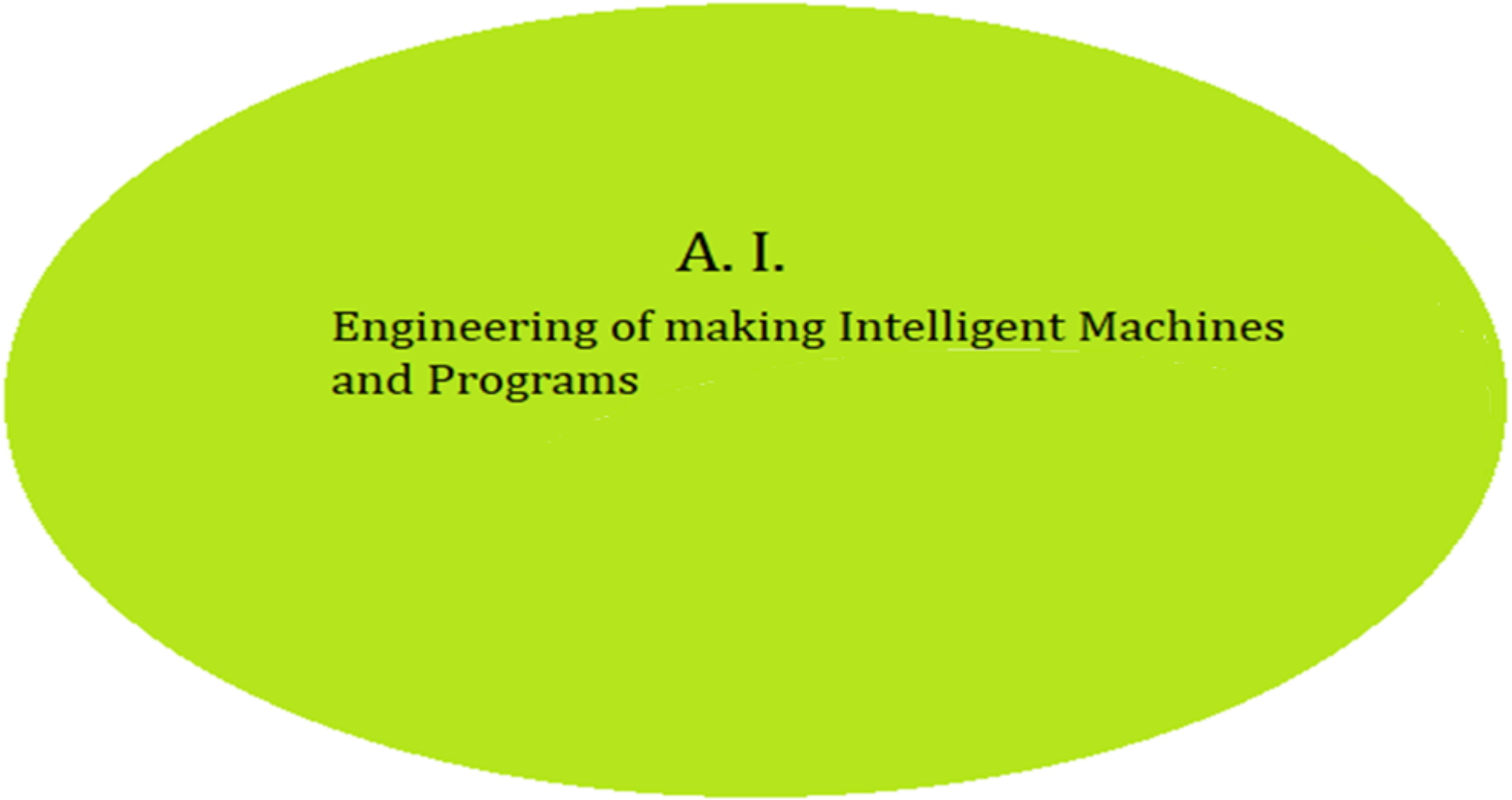


Image Source: Machine Learning: Overview, SRI International

Dimensions of Learning Systems

- type of feedback
  - supervised (labeled examples)
  - unsupervised (unlabeled examples)
  - reinforcement (reward)
- representation
  - attribute-based (feature vector)
  - relational (first-order logic)
- use of knowledge
  - empirical (knowledge-free)
  - analytical (knowledge-guided)

Artificial Intelligence



What is Artificial Intelligence ?

	HUMAN	RATIONAL
THOUGHT	Systems that think like humans	Systems that think rationally
BEHAVIOUR	Systems that act like humans	Systems that act rationally

Systems that act like humans: Turing Test

- “The art of creating machines that perform functions that require intelligence when performed by people.” – **Kurzweil**

- “The study of how to make computers do things at which, at the moment, people are better.” - **Rich and Knight**

Systems that act like humans

- These cognitive tasks include:
  - *Natural language processing*
    - for communication with human
  - *Knowledge representation*
    - to store information effectively & efficiently
  - *Automated reasoning*
    - to retrieve & answer questions using the stored information
  - *Machine learning*
    - to adapt to new circumstances

The total Turing Test

- Includes two more issues:
  - *Computer vision*
    - to perceive objects (seeing)
  - *Robotics*
    - to move objects (acting)

What is Artificial Intelligence ?

	HUMAN	RATIONAL
THOUGHT	Systems that think like humans	Systems that think rationally
BEHAVIOUR	Systems that act like humans	Systems that act rationally

Systems that think like humans: cognitive modeling

- Humans as observed from ‘inside’
- How do we know how humans think?
  - Introspection vs. psychological experiments
- Cognitive Science
- “The exciting new effort to make computers think … machines with *minds* in the full and literal sense” - **Haugeland**
- “[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning …” **- Bellman**

What is Artificial Intelligence ?

	HUMAN	RATIONAL
THOUGHT	Systems that think like humans	Systems that think rationally
BEHAVIOUR	Systems that act like humans	Systems that act rationally

Systems that think ‘rationally’ "laws of thought"

- Humans are not always ‘rational’
- Rational - defined in terms of logic?
- Logic can’t express everything (e.g. uncertainty)
- Logical approach is often not feasible in terms of computation time (needs ‘guidance’)
- “The study of mental facilities through the use of computational models” - **Charniak and McDermott**
- “The study of the computations that make it possible to perceive, reason, and act” - **Winston**

What is Artificial Intelligence ?

	HUMAN	RATIONAL
THOUGHT	**Systems that think like humans**	**Systems that think rationally**
BEHAVIOUR	**Systems that act like humans**	**Systems that act rationally**

Systems that act rationally: "Rational agent"

- Rational behavior: doing the right thing
- The right thing: that which is expected to maximize goal achievement, given the available information
- Giving answers to questions is 'acting'.
- I don't care whether a system:
  - replicates human thought processes
  - makes the same decisions as humans
  - uses purely logical reasoning

Rational agents

- An agent is an entity that perceives and acts
- Abstractly, an agent is a function from percept histories to actions:

$$[f: P^* \rightarrow A]$$

- For any given class of environments and tasks, we seek the agent (or class of agents) with the best performance

- Caveat: computational limitations make perfect rationality unachievable

  → design best program for given machine resources

- Artificial
  - Produced by human art or effort, rather than originating naturally.
- “Intelligence is the ability to acquire knowledge and use it" - **Pigford and Baur**

- **So AI was defined as:**
  - AI is the study of ideas that enable computers to be intelligent.
  - AI is the part of computer science concerned with design of computer systems that exhibit human intelligence **(From the Concise Oxford Dictionary)**

The main topics in AI

Artificial intelligence can be considered under a number of headings:

- Search (includes Game Playing).
- Representing Knowledge and Reasoning with it.
- Planning.
- Machine Learning.
- Natural language processing.
- Expert Systems.
- Interacting with the Environment
  (e.g. Vision, Speech recognition, Robotics)

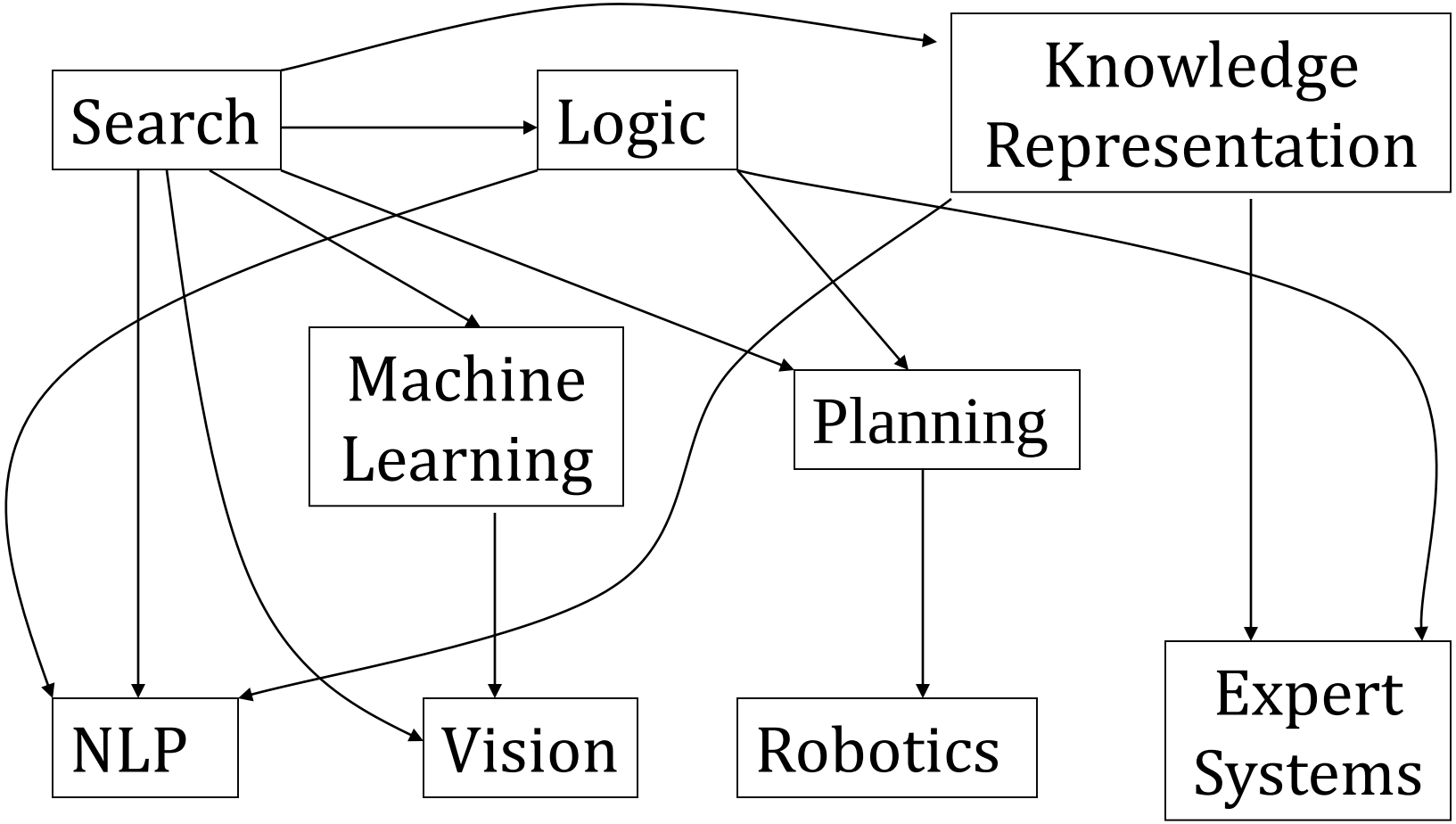
Some Advantages of Artificial Intelligence

- more powerful and more useful computers
- new and improved interfaces
- solving new problems
- better handling of information
- relieves information overload
- conversion of information into knowledge

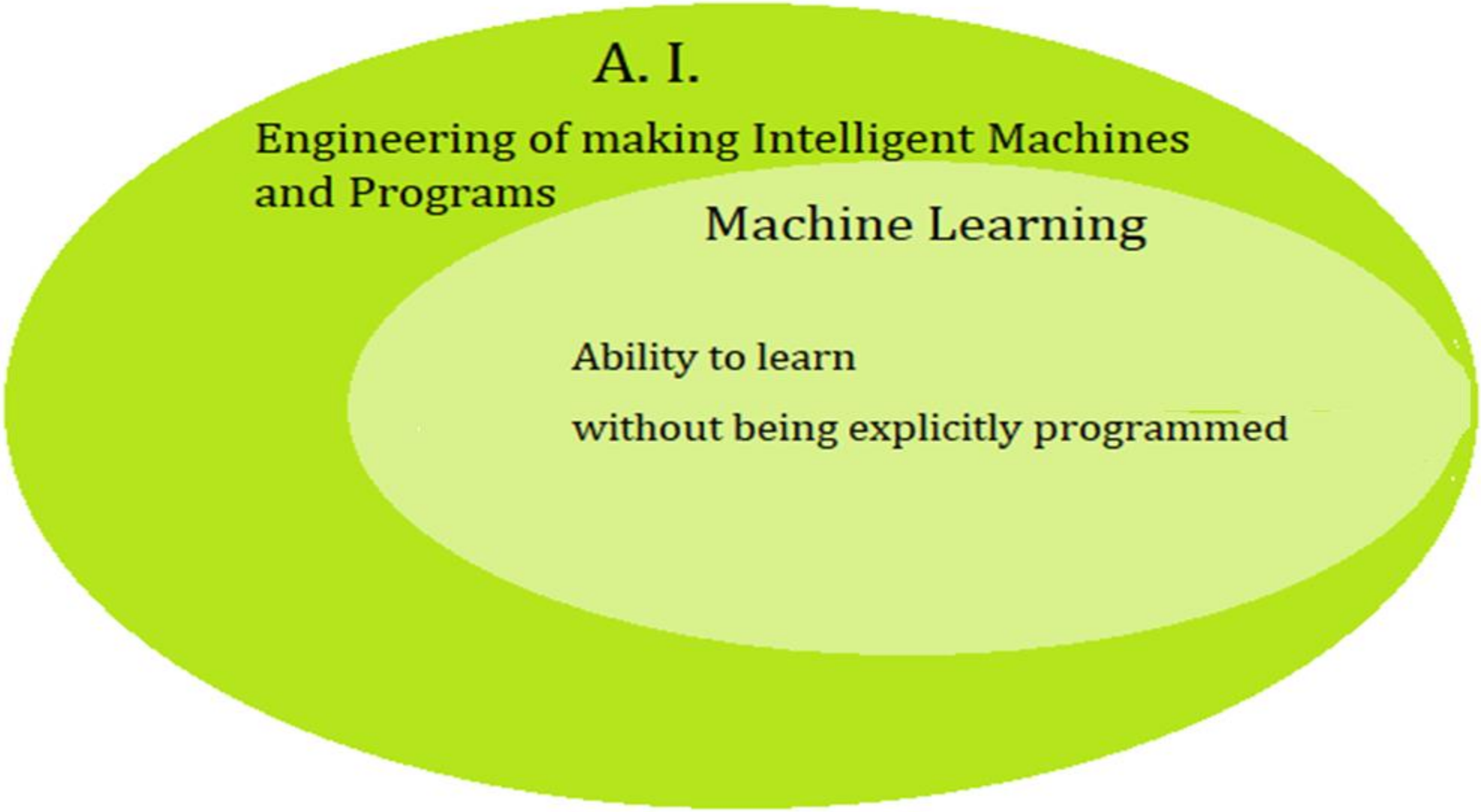
The Disadvantages

- increased costs
- difficulty with software development - slow and expensive
- few experienced programmers
- few practical products have reached the market as yet.

Areas of AI


Search
Logic
Knowledge Representation
Machine Learning
Planning
NLP
Vision
Robotics
Expert Systems

Machine Learning



Machine Learning – A definition

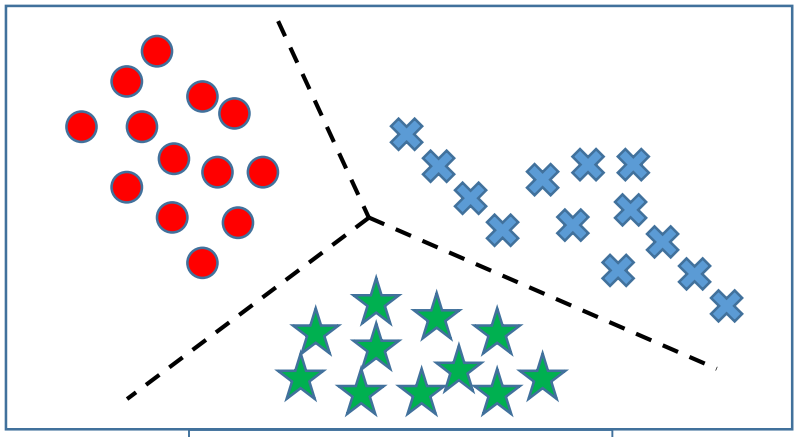
"A computer program is said to ***learn from experience E*** *with respect to* ***some class of tasks T*** *and* ***performance measure P*** *if its performance at tasks in T, as measured by P, improves with experience E "* - **Tom Mitchell**

Why Machine Learning?

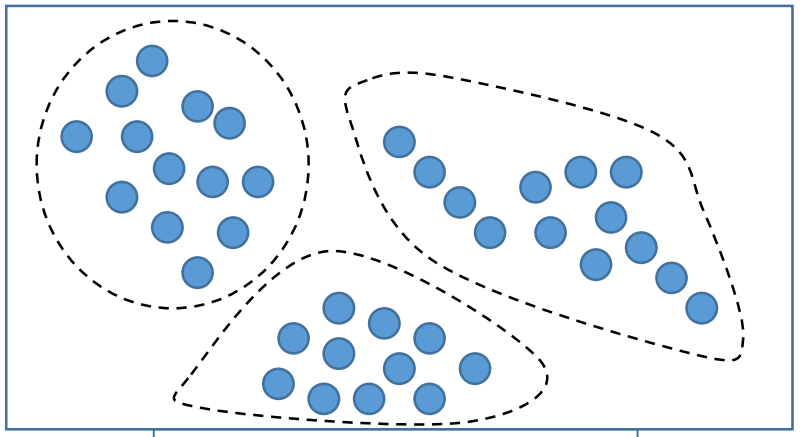
- **No human experts**
  - industrial/manufacturing control
  - mass spectrometer analysis, drug design, astronomic discovery
- **Black-box human expertise**
  - face/handwriting/speech recognition
  - driving a car, flying a plane
- **Rapidly changing phenomena**
  - credit scoring, financial modeling
  - diagnosis, fraud detection
- **Need for customization/personalization**
  - personalized news reader
  - movie/book recommendation

Learning Methods

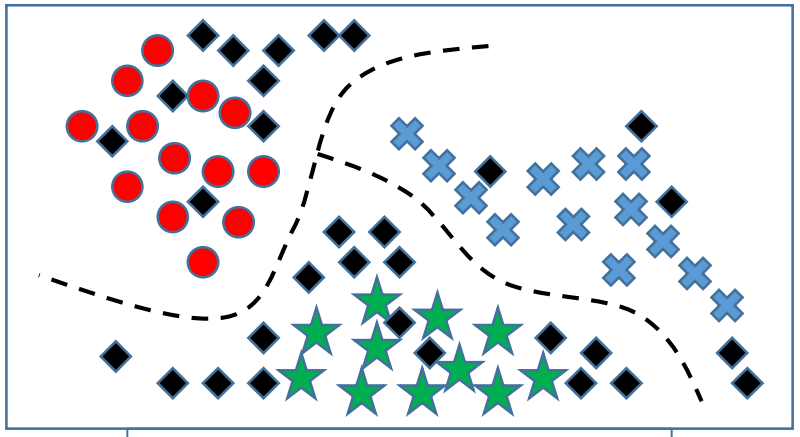
- **Supervised learning** ($\{x_n \in R^d, y_n \in R\}_{n=1}^N$)
  - Prediction
  - Classification (discrete labels), Regression (real values)
- **Unsupervised learning** ($\{x_n \in R^d\}_{n=1}^N$)
  - Clustering
  - Probability distribution estimation
  - Finding association (in features)
  - Dimension reduction
- **Semi-supervised learning**
- **Reinforcement learning**
  - Decision making (robot, chess machine)



Supervised learning



Unsupervised learning



Semi-supervised learning

Image Source: Yi-Fan Chang, An Overview of Machine Learning

Supervised learning

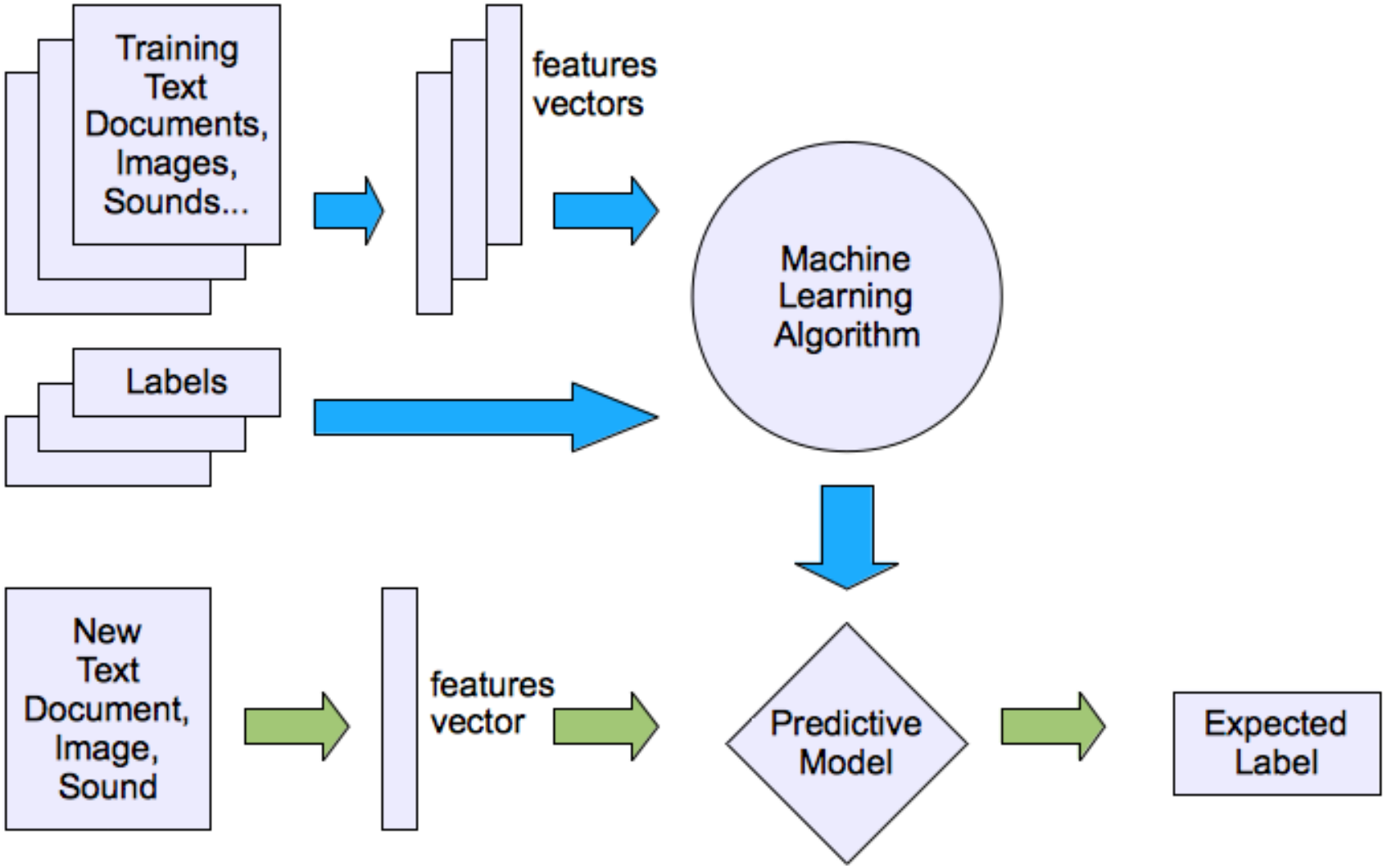
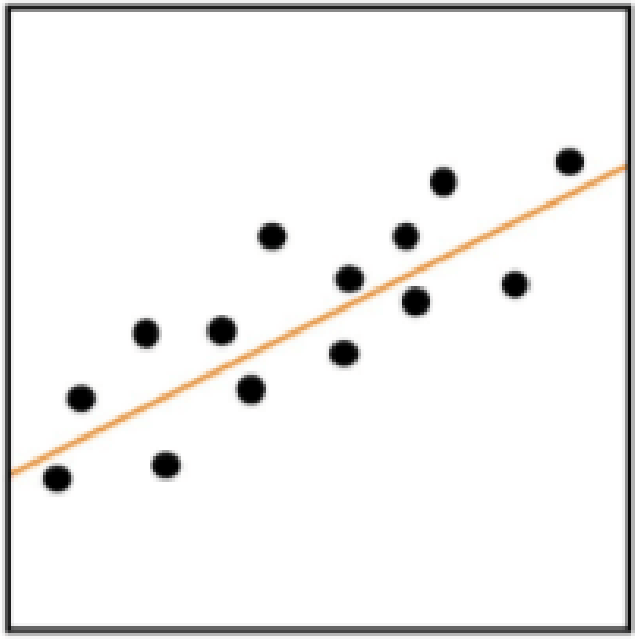


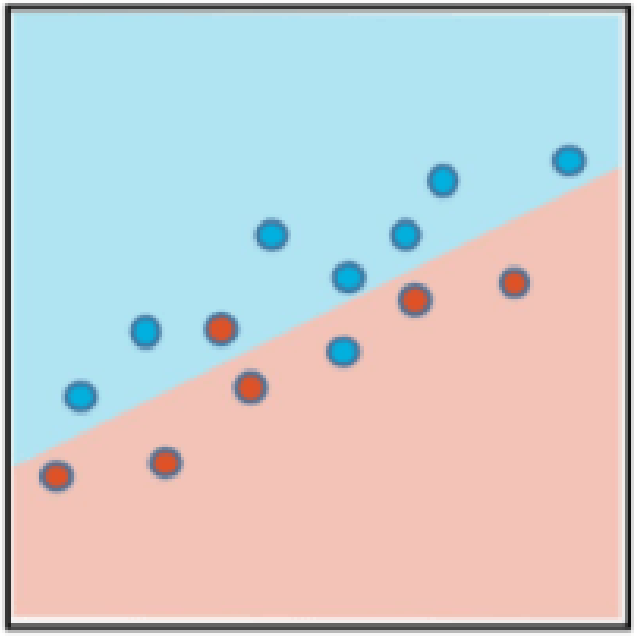
Image Source: Yi-Fan Chang, An Overview of Machine Learning

Classification and Regression

- $x \rightarrow f(x) \rightarrow y \in \{-1, 1\}$; Binary Classification
- $x \rightarrow f(x) \rightarrow y \in \mathbb{R}$; Regression



a) Regression



b) Classification

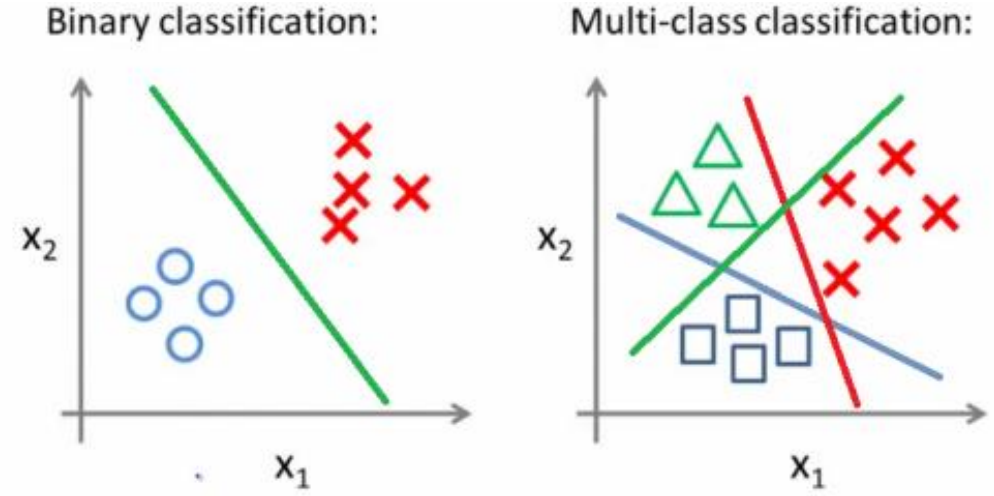
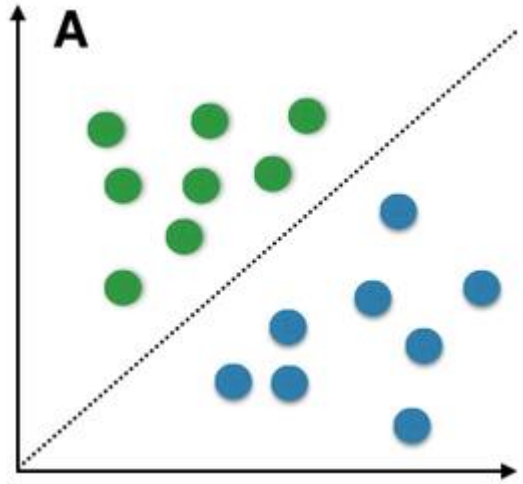


Image Source:
https://utkuufuk.github.io/2018/06/03/one-vs-all-classification/
https://www.edvancer.in/foundations-of-data-science-made-simple-part-2/

Linear and Non Linear Classification

- Linear Classification

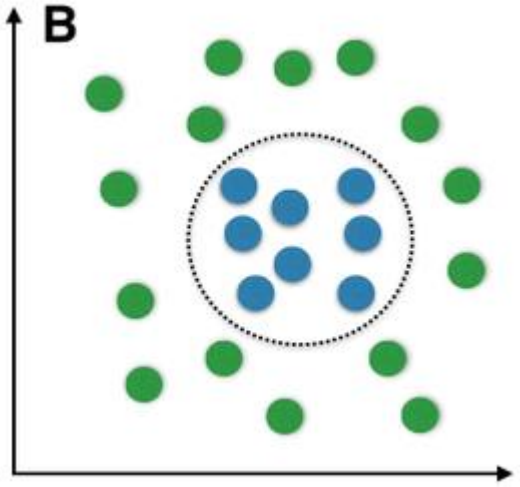
Find a **straight line** to separate (classify) **green** from **blue**



Examples of linear classifier are:
Linear Discriminant Classifier, Naive Bayes, Logistic Regression, Perceptron, SVM (with linear kernel)

- Non-Linear Classification

There is **no way** that a **straight line** can be drawn to discriminate the two classes



Examples of non-linear classifiers are:
Quadratic Discriminant Classifier, Multi-layer Perceptron (most deep networks), Decision Trees, Random Forest, K-Nearest Neighbour

The Classification framework

- Apply a prediction function to a feature representation of the image to get the desired output:

f() = “apple”

f() = “tomato”

f() = “cow”

The Classification framework

$$y = f(\mathbf{x})$$

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Slide credit: L. Lazebnik

Supervised learning categories and techniques

- **Linear classifier** (numerical functions)
- **Parametric** (Probabilistic functions)
  - Naïve Bayes, Gaussian discriminant analysis (GDA), Hidden Markov models (HMM), Probabilistic graphical models
- **Non-parametric** (Instance-based functions)
  - *K*-nearest neighbors, Kernel regression, Kernel density estimation, Local regression
- **Non-metric** (Symbolic functions)
  - Classification and regression tree (CART), decision tree
- **Aggregation**
  - Bagging (bootstrap + aggregation), Adaboost, Random forest

Applications of Supervised learning :

- Aka Pattern recognition
- Face recognition: Pose, lighting, occlusion (glasses, beard), make-up, hair style
- Character recognition: Different handwriting styles.
- Speech recognition: Temporal dependency.
  - Use of a dictionary or the syntax of the language.
  - Sensor fusion: Combine multiple modalities; eg, visual (lip image) and acoustic for speech
- Medical diagnosis: From symptoms to illnesses
- Web Advertising: Predict if a user clicks on an ad on the Internet.

Unsupervised learning

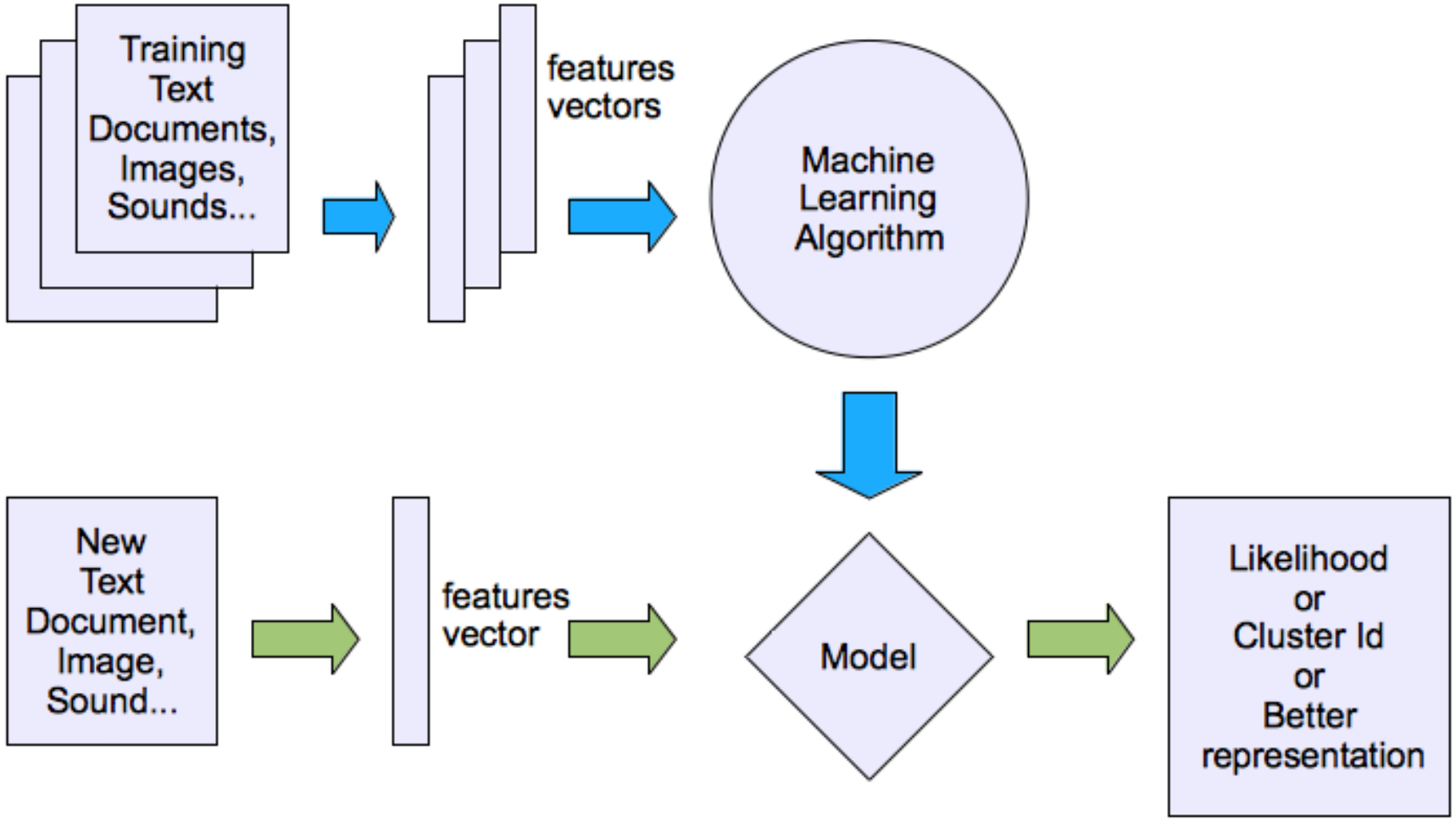


Image Source: Yi-Fan Chang, An Overview of Machine Learning

Unsupervised learning categories and techniques

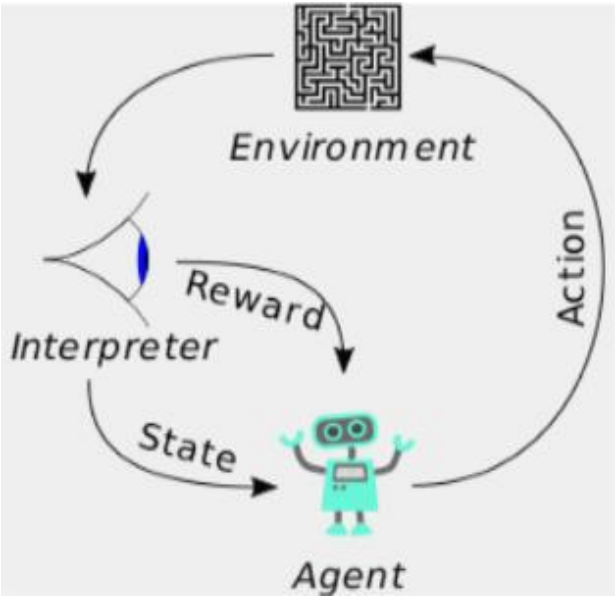
- **Clustering**
  - K-means clustering
  - Spectral clustering
- **Density Estimation**
  - Gaussian mixture model (GMM)
  - Graphical models
- **Dimensionality reduction**
  - Principal component analysis (PCA)
  - Factor analysis

Applications of unsupervised learning:

- **Clustering:** Grouping similar instances
- **Other applications:** Summarization, Association Analysis
- **Example applications:**
  - Customer segmentation in CRM
  - Image compression: Color quantization
  - Bioinformatics: Learning motifs

Reinforcement Learning

- A computer program interacts with a dynamic environment in which it must perform a certain goal. The program is provided feedback in terms of rewards and punishments as it navigates its problem space.
  - Policies: what actions should an agent take in a particular situation
  - Utility estimation: how good is a state (→used by policy)
- No supervised output but delayed reward
- Credit assignment problem (what was responsible for the outcome)
- Applications:
  - Game playing
  - Robot in a maze
  - Multiple agents, partial observability, ...



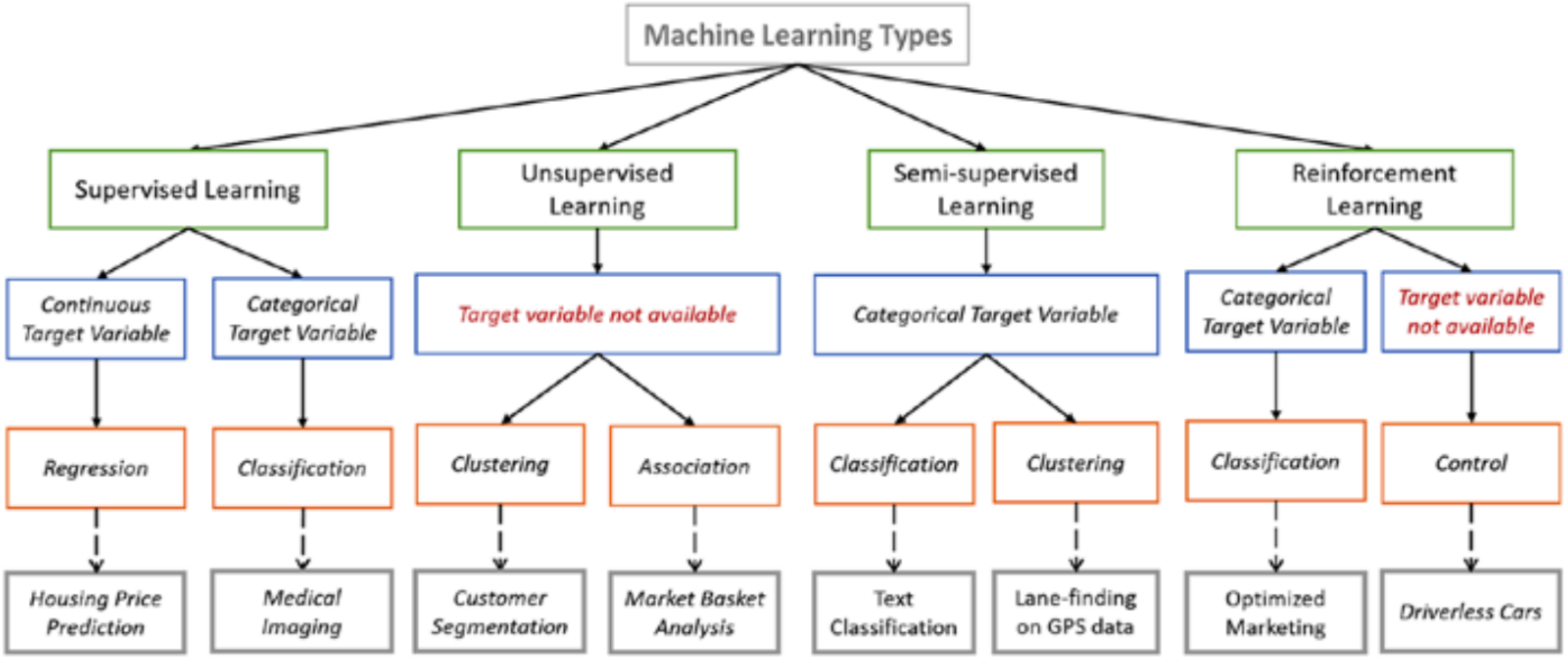
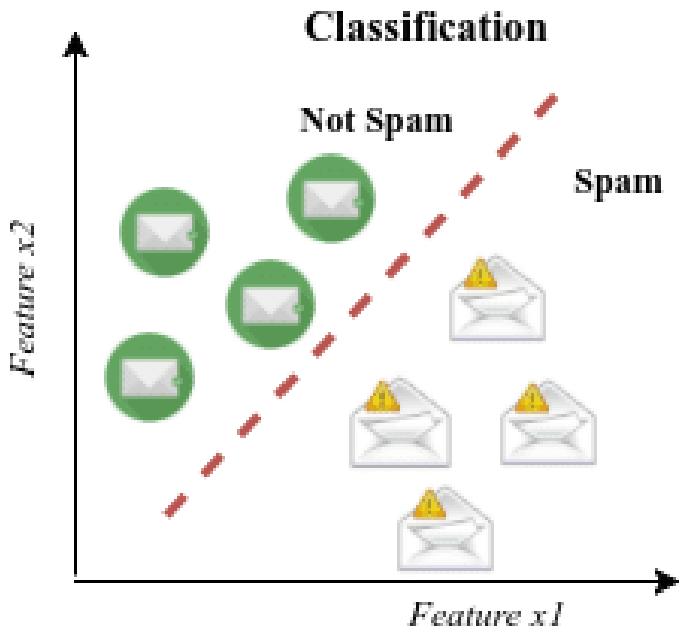
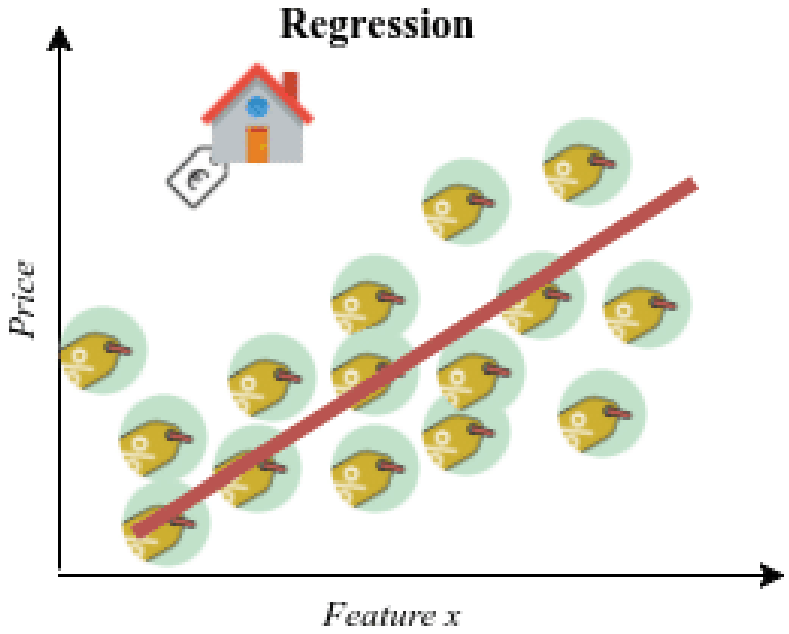
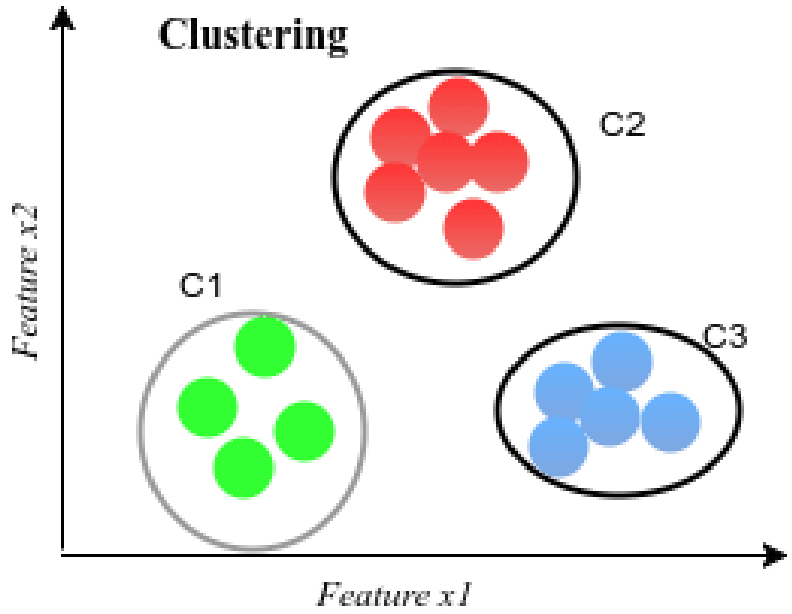


Image Source: https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861







Examples of real-life problems in the context of supervised and unsupervised learning tasks: Spam filtering as a classification task and House price estimation as a regression task are part of supervised learning; Clustering is part of unsupervised learning in which customers are grouped into three different categories based on their purchasing behavior.

Image Source: DEEP GENERATIVE MODELS FOR SYNTHETIC RETINAL IMAGE GENERATION, Lappeenranta University of Technology, July 2017

The Machine Learning Life Cycle

1. **Define Project Objectives**
   - ☐ Specify business problem
   - ☐ Acquire subject matter expertise
   - ☐ Define unit of analysis and prediction target
   - ☐ Prioritize modeling criteria
   - ☐ Consider risks and success criteria
   - ☐ Decide whether to continue
2. **Acquire & Explore Data**
   - ☐ Find appropriate data
   - ☐ Merge data into single table
   - ☐ Conduct exploratory data analysis
   - ☐ Find and remove any target leakage
   - ☐ Feature engineering
3. **Model Data**
   - ☐ Variable selection
   - ☐ Build candidate models
   - ☐ Model validation and selection
4. **Interpret & Communicate**
   - ☐ Interpret model
   - ☐ Communicate model insights
5. **Implement, Document & Maintain**
   - ☐ Set up batch or API prediction system
   - ☐ Document modeling process for reproducibility
   - ☐ Create model monitoring and maintenance plan

Image Source: https://www.datarobot.com/wiki/machine-learning-life-cycle/

Machine Learning Applications

- Recognizing patterns:
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns:
  - Generating images or motion sequences
- Recognizing anomalies:
  - Unusual sequences of credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant or unusual sound in your car engine.
- Prediction:
  - Future stock prices or currency exchange rates

Some web-based examples of machine learning

- The web contains a lot of data. Tasks with very big datasets often use machine learning
  - especially if the data is noisy or non-stationary.
- Spam filtering, fraud detection:
  - The enemy adapts so we must adapt too.
- Recommendation systems:
  - Lots of noisy data. Million dollar prize!
- Information retrieval:
  - Find documents or images with similar content.
- Data Visualization:
  - Display a huge database in a revealing way.

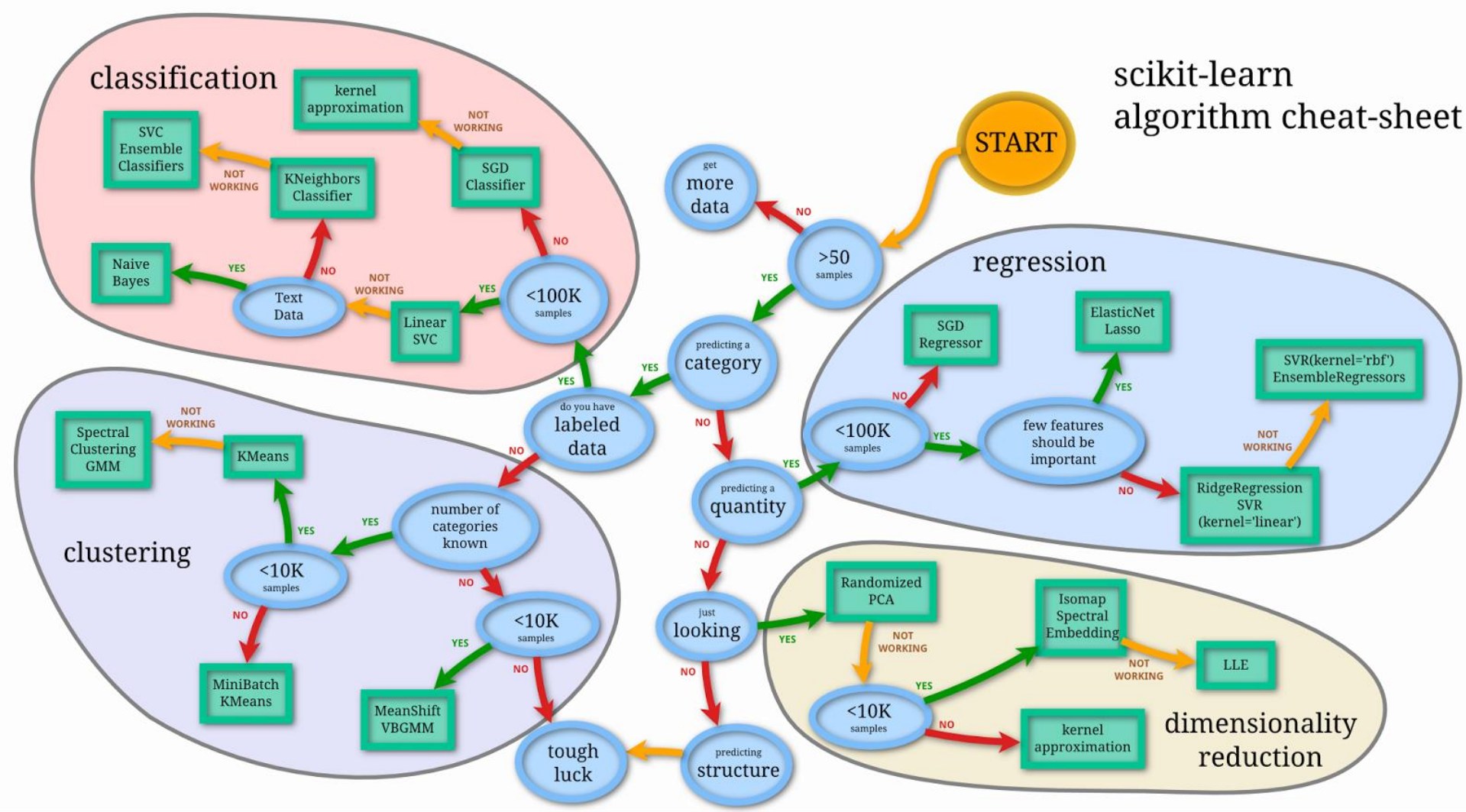
Python Libraries for Machine Learning

Machine Learning	Visualization	Mathematics for Data Science	Data Manipulation & Analysis
TensorFlow	matplotlib	SciPy	pandas
scikit-learn	Bokeh	NumPy	Scrapy
PYTORCH	plotly	Natural Language ToolKit	
Keras	Seaborn	StatsModels Statistics in Python	
XGBoost CatBoost LightGBM	pydot	spaCy	
dist-keras elephas spark-deep-learning		gensim	
eli5			

Scikit-Learn - (formerly **scikits.learn**)

It features various **classification, regression and clustering** algorithms including **support vector machines, random forests, gradient boosting, k-means and DBSCAN**, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Scikit-Learn



TensorFlow

TensorFlow™ is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.

Keras

- Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation.
- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.

OpenCV

- OpenCV is the leading open source library for computer vision, image processing and machine learning, and now features GPU acceleration for real-time operation.
- OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing.

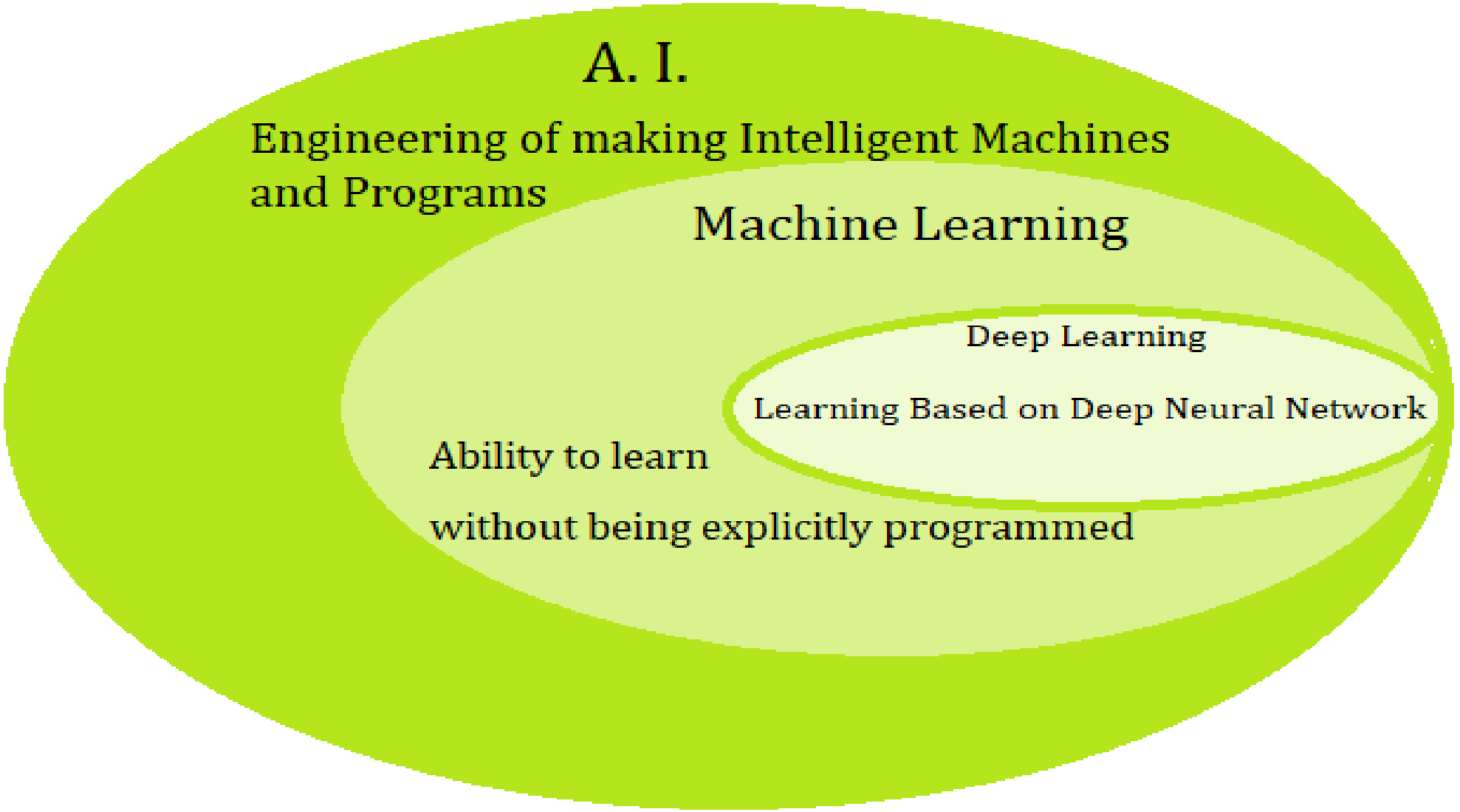
Research issues in Machine Learning

- Handling large amount of data
- Training Time is more (Hours or Days)
- Labeling of data or labeling of Images & Videos
- Processing Performance for Training and Testing (Validating)
- Accessing Reference Model – Transfer Learning
- Optimizing Hyper parameters
- Rapid and Optimized deployment
- Curse of Modularity
- Class Imbalance
- Feature Engineering
- Variance and Bias
- Applications specific training – Less Generalization

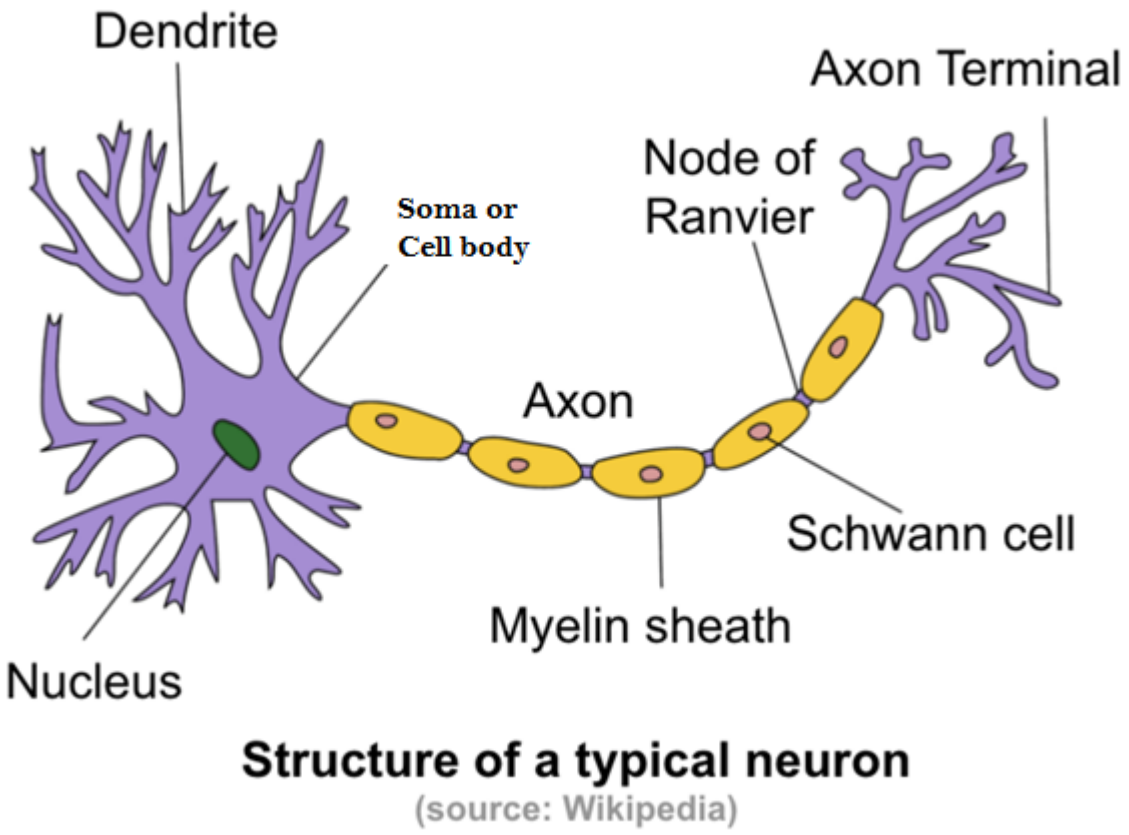
Growth of Machine Learning – Current State

- Machine learning is preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - Computational biology
- This trend is accelerating – Deep Learning…
  - Improved machine learning algorithms
  - Improved data capture, networking, faster computers
  - Software too complex to write by hand
  - New sensors / IO devices
  - Demand for self-customization to user, environment

Deep Learning



Biological Inspiration



Structure of a typical neuron

(source: Wikipedia)

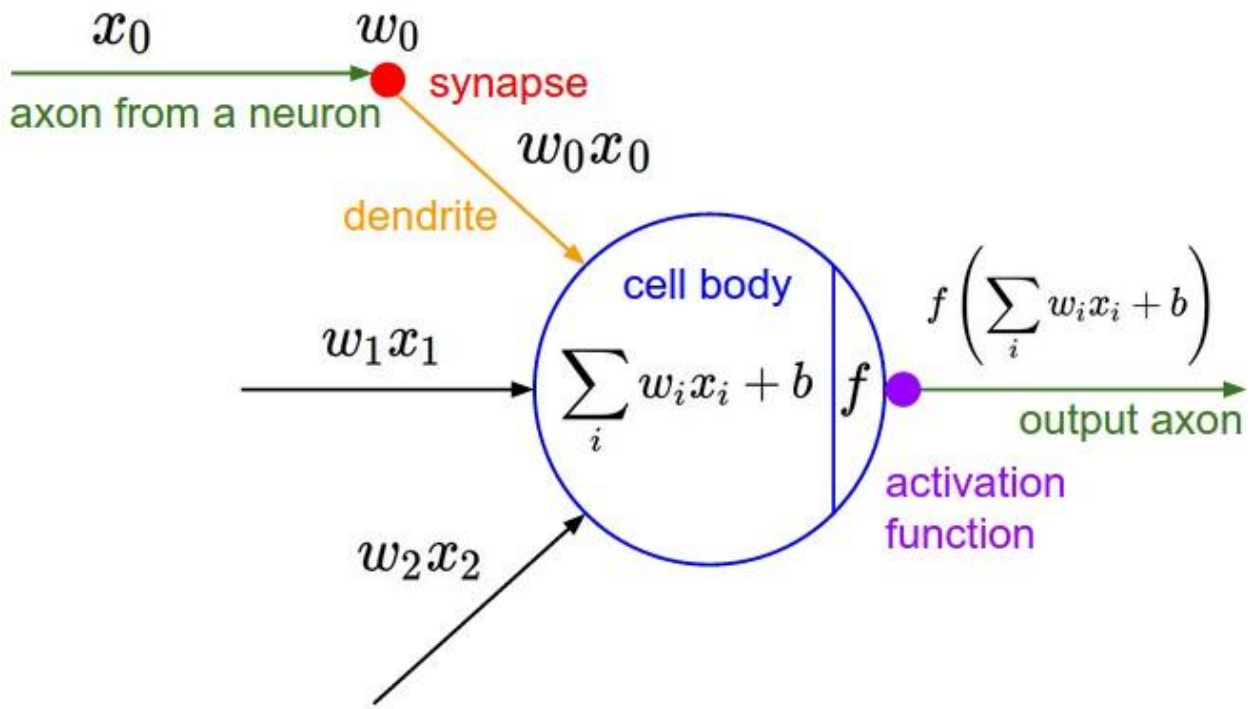


Image Source: http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/

Deep Learning – A Definition

Deep Learning is a subfield of Machine Learning which uses computational models, with **hierarchical architectures composed by multiple processing layers**, to learn representations of complex data such as images, sound and text.

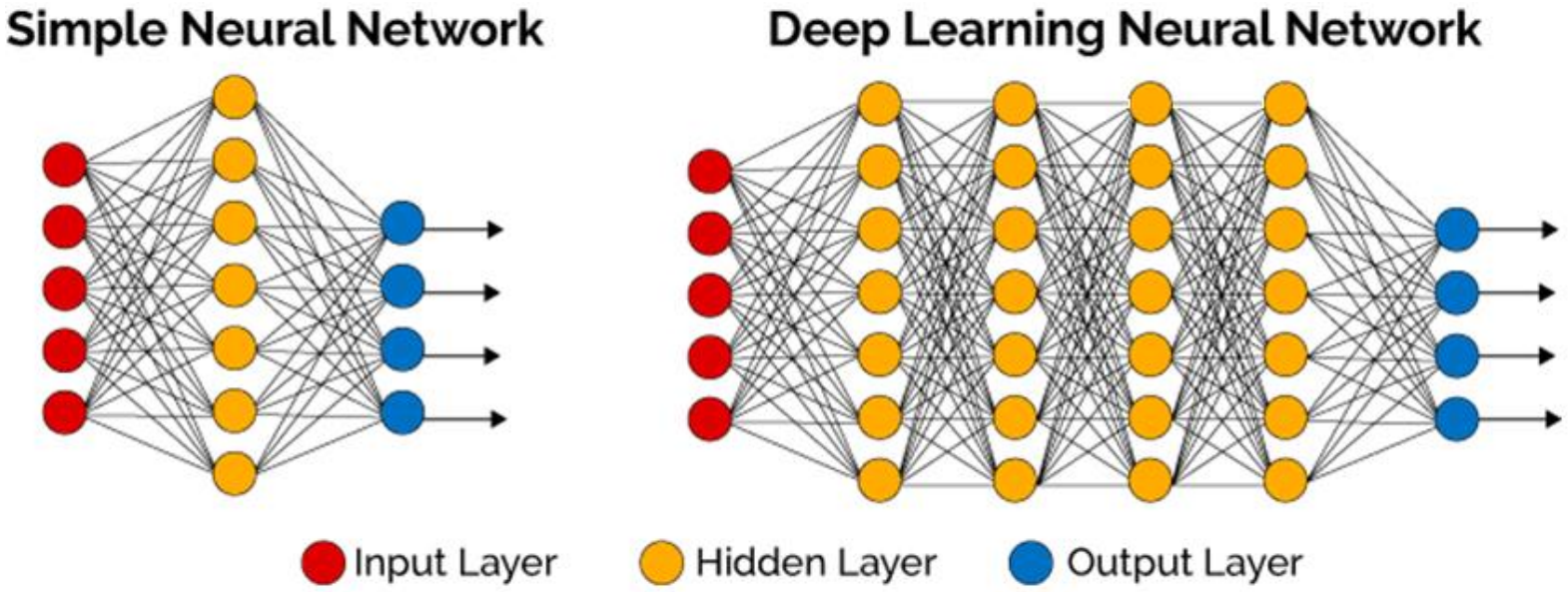


Image Source: https://www.linkedin.com/pulse/intuition-deep-neural-networks-abhishek-parbhakar

A bit of History

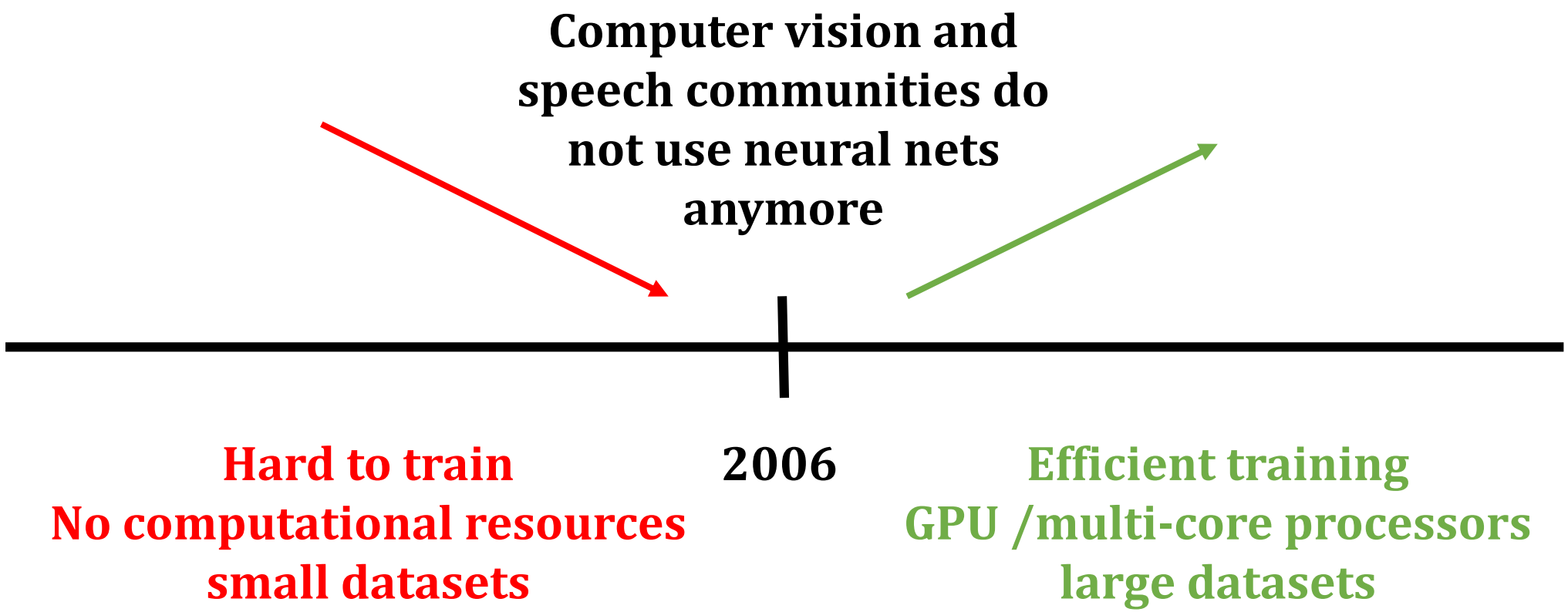
1965
Alexey Ivakhnenko

First works on
Deep neural networks

1986
Geoffrey Hinton

backpropagation
algorithm

A bit of History



A bit of History

2011
Microsoft

breakthrough
in speech recognition

2012
Geoffrey Hinton
(University of Toronto)

breakthrough
in computer vision

Researchers of Deep Learning

Andrew Ng

Fei Fei Li

Yann LeCun

Yoshua Bengio

Ian Goodfellow

Andrej Karpathy

Jeff Dean

Nitish Srivastava

Dr Ketan Kotecha

Difference between ML and DL

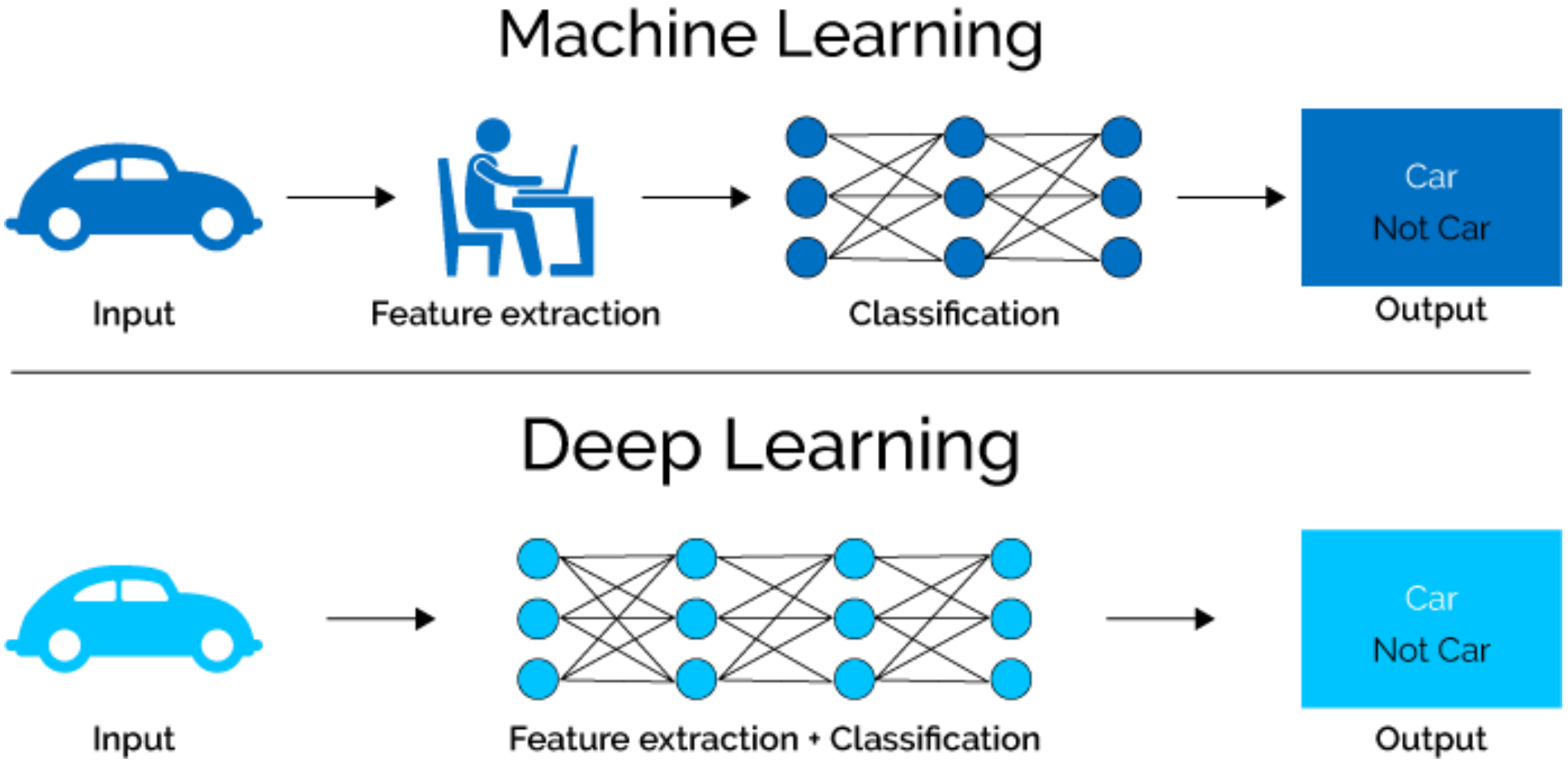


Image Source: https://www.xenonstack.com/blog/log-analytics-with-deep-learning-and-machine-learning

Data dependencies

ML

- Traditional machine learning algorithms with their handcrafted rules succeed in this scenario.

DL

- When the data is small, deep learning algorithms don't perform that well. This is because deep learning algorithms need a large amount of data to understand it perfectly.

Hardware dependencies

ML

- Traditional machine learning algorithms can work on low-end machines.

DL

- Deep learning algorithms heavily depend on high-end machines because deep learning algorithms inherently do a large amount of matrix multiplication operations. These operations can be efficiently optimized using a GPU because GPU is built for this purpose.

Feature engineering

ML

- Most of the applied features need to be identified by an expert and then hand-coded as per the domain and data type.
- For example, features can be pixel values, shape, textures, position and orientation. The performance of most of the Machine Learning algorithm depends on how accurately the features are identified and extracted.

DL

- Deep learning algorithms try to learn high-level features from data.
- deep learning reduces the task of developing new feature extractor for every problem. Like, Convolutional NN will try to learn low-level features such as edges and lines in early layers then parts of faces of people and then high-level representation of a face.

Problem Solving approach

ML

- Traditional machine learning is generally recommended to break the problem down into different parts, solve them individually and combine them to get the result.
- For object detection problem, you would divide the problem into two steps, object detection and object recognition.

DL

- Deep learning in contrast advocates to solve the problem end-to-end.
- Image would be passed to network as an input, and it would give out the location along with the name of object. (i.e. YOLO net)

Execution time

ML

- Machine learning comparatively takes much less time to train, ranging from a few seconds to a few hours.

DL

- A deep learning algorithm takes a long time to train. This is because there are so many parameters in a deep learning algorithm that training them takes longer than usual.
- State of the art deep learning algorithm ResNet takes about two weeks to train completely from scratch.

Interpretability

ML

- Easy to interpret the reasoning behind it.

DL

- No one interprets what happens in a node of DL model.

Let's take an example. Suppose we use deep learning to give automated scoring to essays. The performance it gives in scoring is quite excellent and is near human performance. But there's is an issue. It does not reveal why it has given that score. Indeed mathematically you can find out which nodes of a deep neural network were activated, but we don't know what there neurons were supposed to model and what these layers of neurons were doing collectively. So we fail to interpret the results.

On the other hand, machine learning algorithms like decision trees give us crisp rules as to why it chose what it chose, so it is particularly easy to interpret the reasoning behind it. Therefore, algorithms like decision trees and linear/logistic regression are primarily used in industry for interpretability.

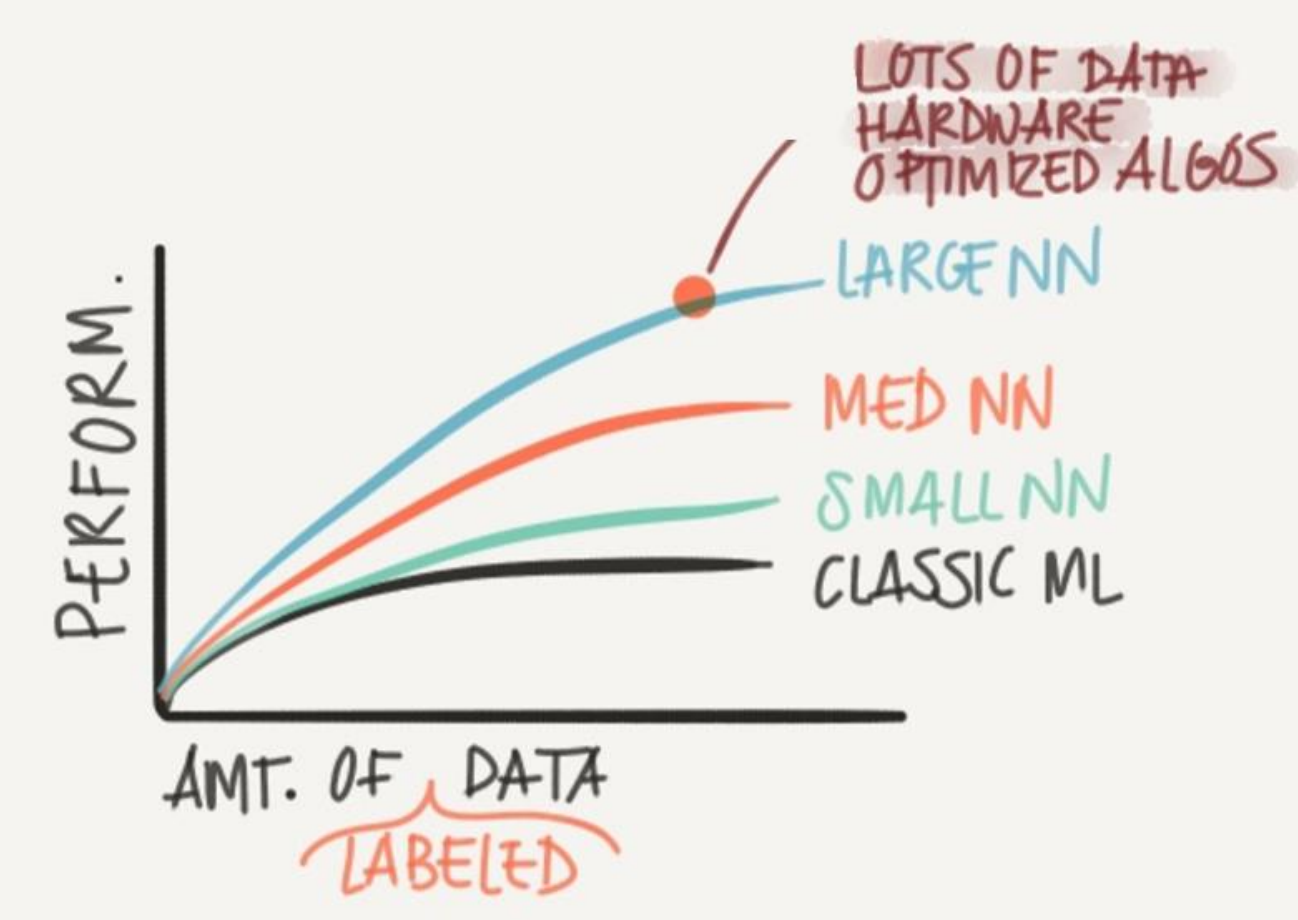
Why Deep Learning taking off?

1. Lots of Data – Structured and

   Unstructured

2. Computation Power – GPU

3. Efficient Algorithms



Why deep representations?

- Deep NN makes relations with data from simpler to complex.
- In each layer it tries to make a relations between the previous layer.
- Face recognition application:
  - Image ==> Edges ==> Face parts ==> Faces ==> desired face
- Audio recognition application:
  - Audio ==> Low level sound features ==> Phonemes ==> Words ==> Sentences
- Neural Researchers thinks that deep neural networks thinks like brains (Simple ==> Complex)

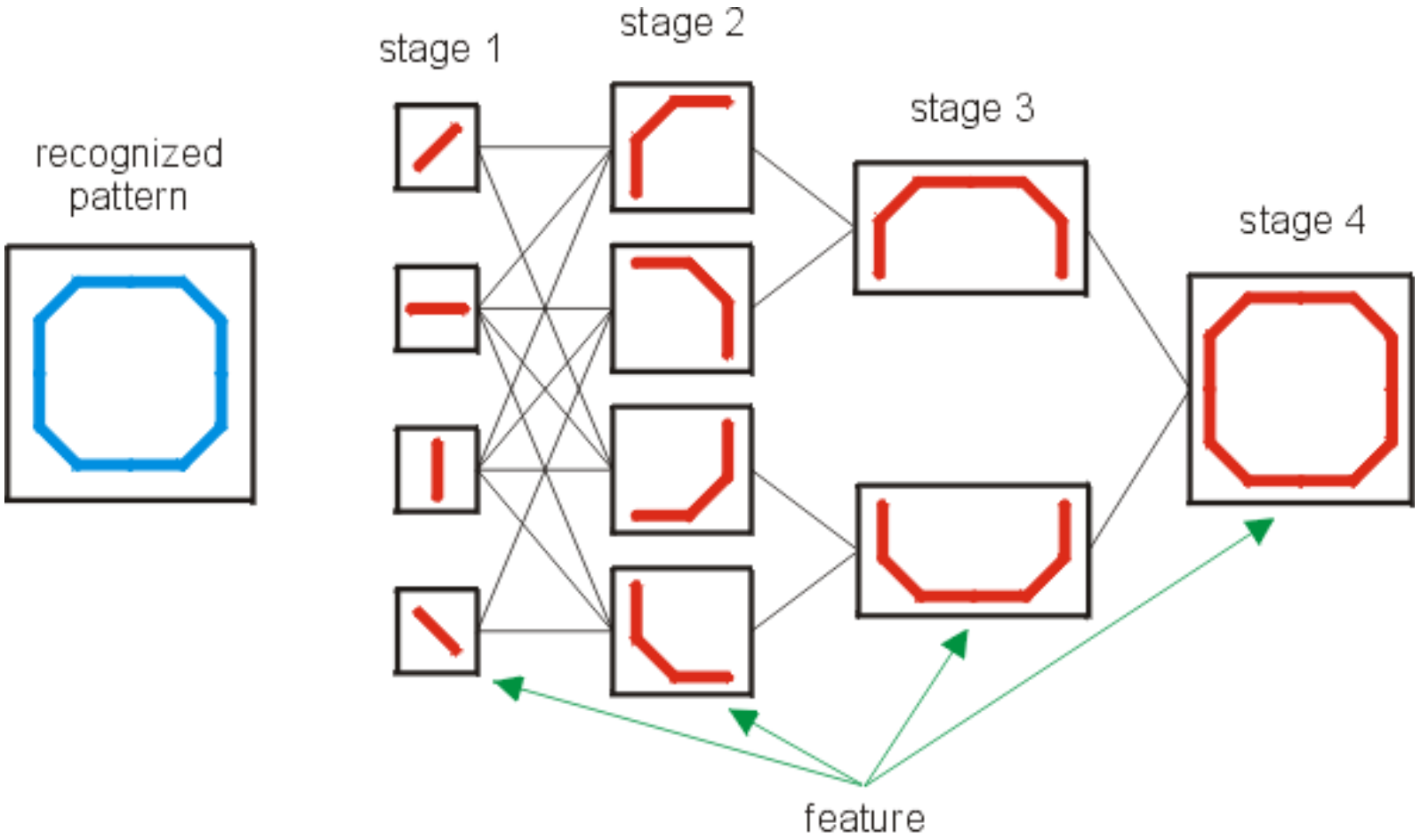
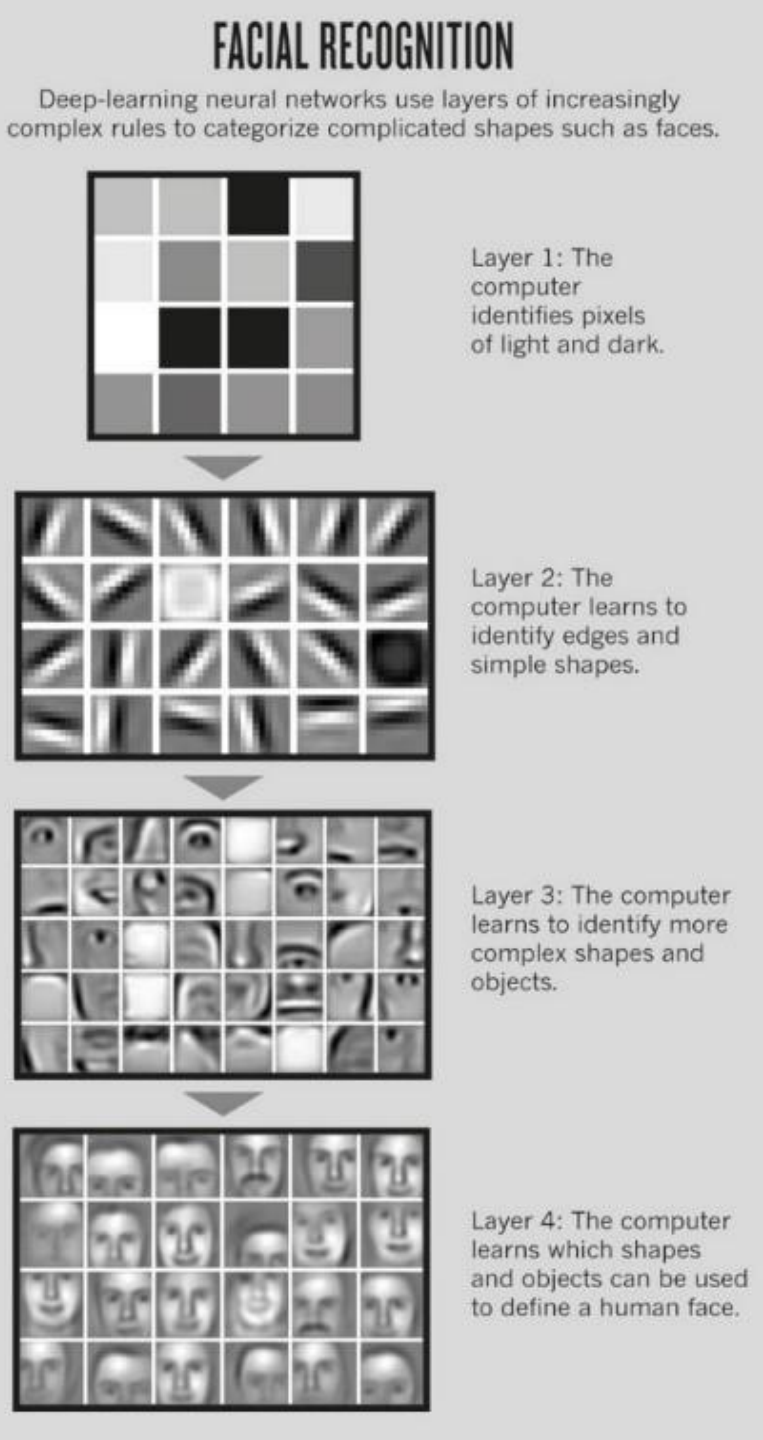


Image Source: https://www.kiv.zcu.cz/studies/predmety/uir/NS/Neocognitron/en/hierarch-det.html



SWIMMING

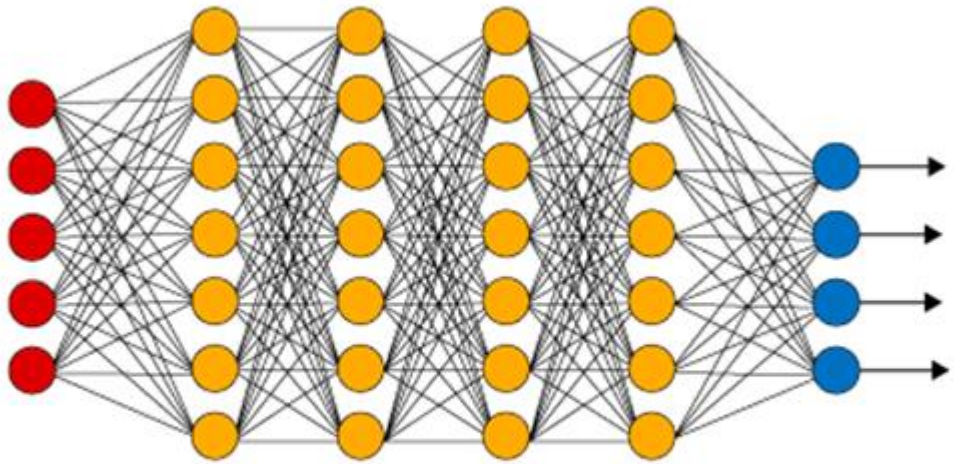
Move Around on the Surface of a Deep Swimming Pool

Binary Classification

1: CAT
0: NOT CAT

model(X_train, Y_train, X_test, Y_test, num_iterations = 2000, learning_rate = 0.5, print_cost = False)

Deep Learning Neural Network



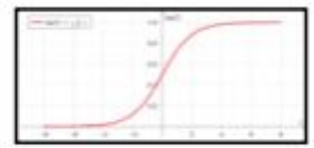
Input Layer Hidden Layer Output Layer

$w\ x + b$ | σ

0.73

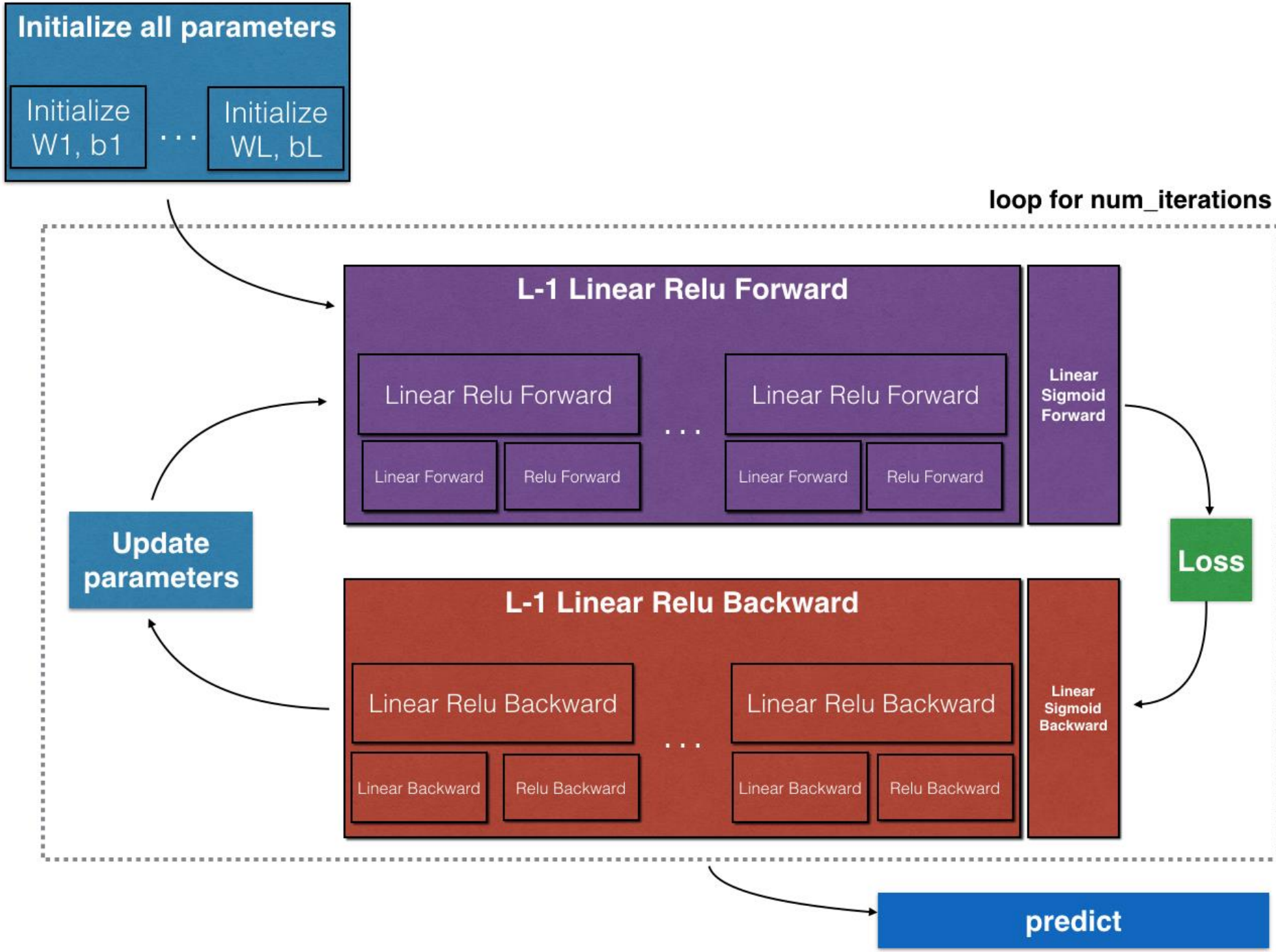
0.73 > 0.5

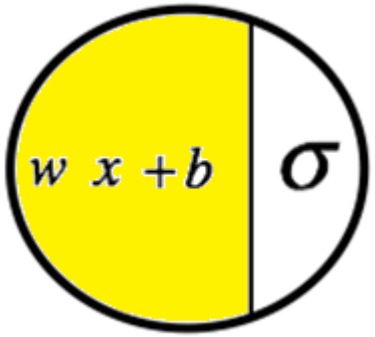
"it's a cat"



Algorithm:

- Define the model structure (such as number of input and outputs)
- Initialize the model's parameters.
- Loop.
  - Calculate current loss (forward propagation)
  - Calculate current gradient (backward propagation)
  - Update parameters (gradient descent)
    - Repeat Until it Converges


Initialize all parameters
Initialize W1, b1
. . .
Initialize WL, bL
loop for num_iterations
L-1 Linear Relu Forward
Linear Relu Forward
Linear Relu Forward
. . .
Linear Forward
Relu Forward
Linear Forward
Relu Forward
Linear Sigmoid Forward
Loss
Update parameters
L-1 Linear Relu Backward
Linear Relu Backward
Linear Relu Backward
. . .
Linear Backward
Relu Backward
Linear Backward
Relu Backward
Linear Sigmoid Backward
predict



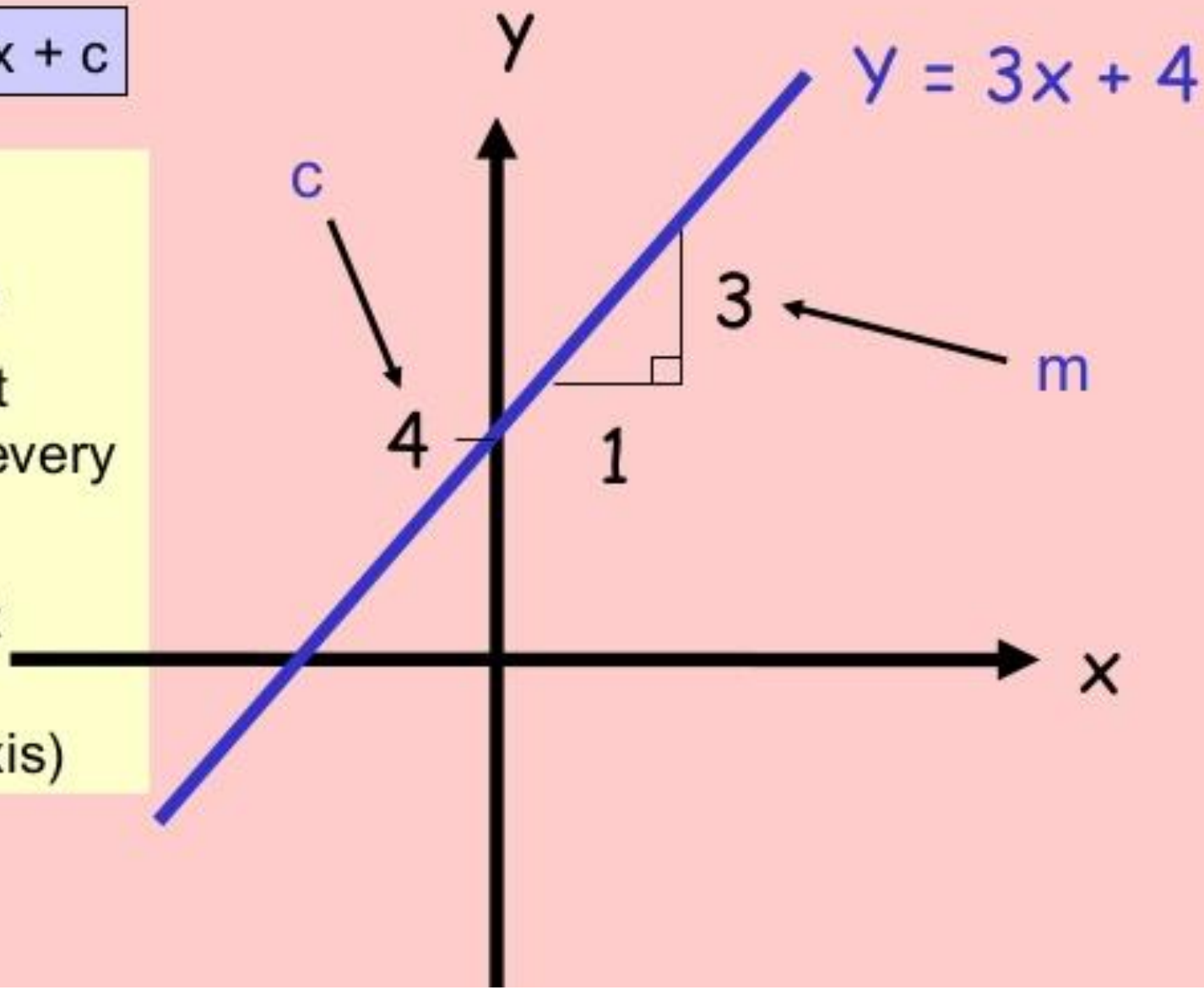
Graphs of y = mx + c

In the equation:

y = mx + c

m = the gradient (how far up for every one along)

c = the intercept (where the line crosses the y axis)



EXAMPLE 1 Use the two points shown in Figure 8-9 to find the slope of the line passing through the points with coordinates $(-2, 4)$ and $(3, -4)$.

Solution We can let $(x_1, y_1) = (-2, 4)$ and $(x_2, y_2) = (3, -4)$. Then

$$m = \frac{\text{change in } y}{\text{change in } x}$$

$$= \frac{y_2 - y_1}{x_2 - x_1}$$

$$= \frac{-4 - 4}{3 - (-2)}$$

Substitute -4 for y_2, 4 for y_1, 3 for x_2, and -2 for x_1.

$$= \frac{-8}{5}$$

$$= -\frac{8}{5}$$

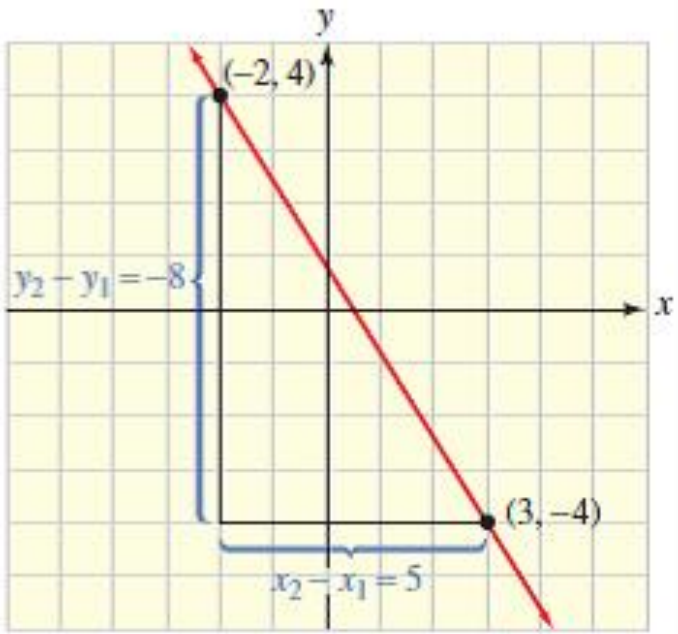


Figure 8-9

The slope of the line is $-\frac{8}{5}$. We would obtain the same result if we let $(x_1, y_1) = (3, -4)$ and $(x_2, y_2) = (-2, 4)$.

y = mx + c

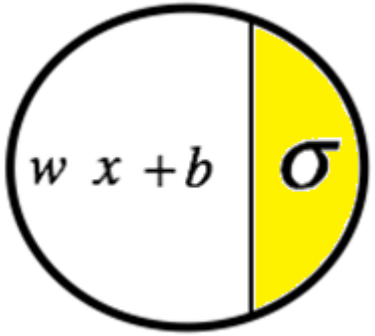
y - y1 = m(x - x1)

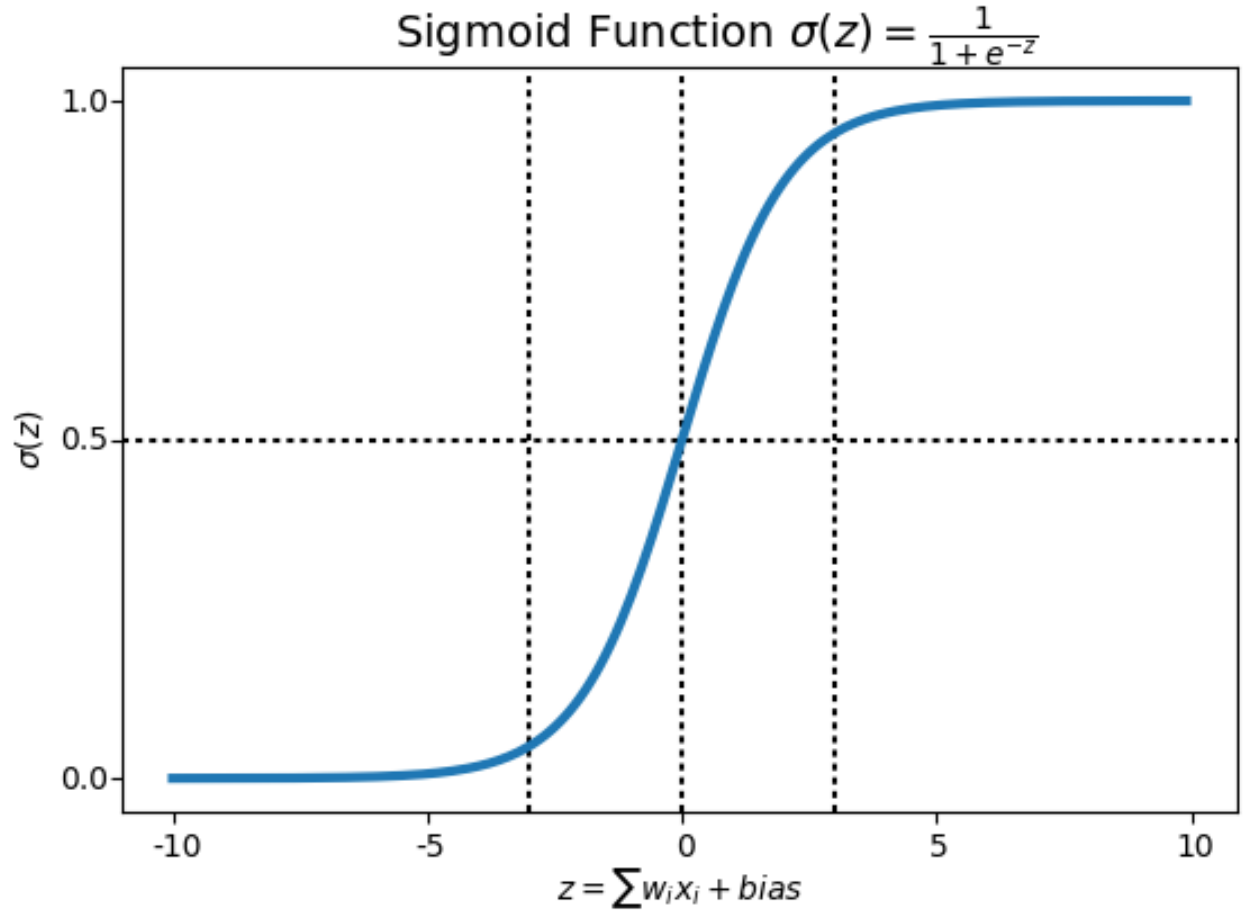
y - 4 = (-1.6)(x - (-2))
y - 4 = -1.6x - 3.2
y = -1.6x - 3.2 + 4

y = -1.6x + 0.8

y = -1.6(3) + 0.8
y= -4

Activation Function





Logistic Regression

$z = b + a_1x_1 + a_2x_2 + a_3x_3$
$p = 1.0 / (1.0 + e^{-z})$

Ex:

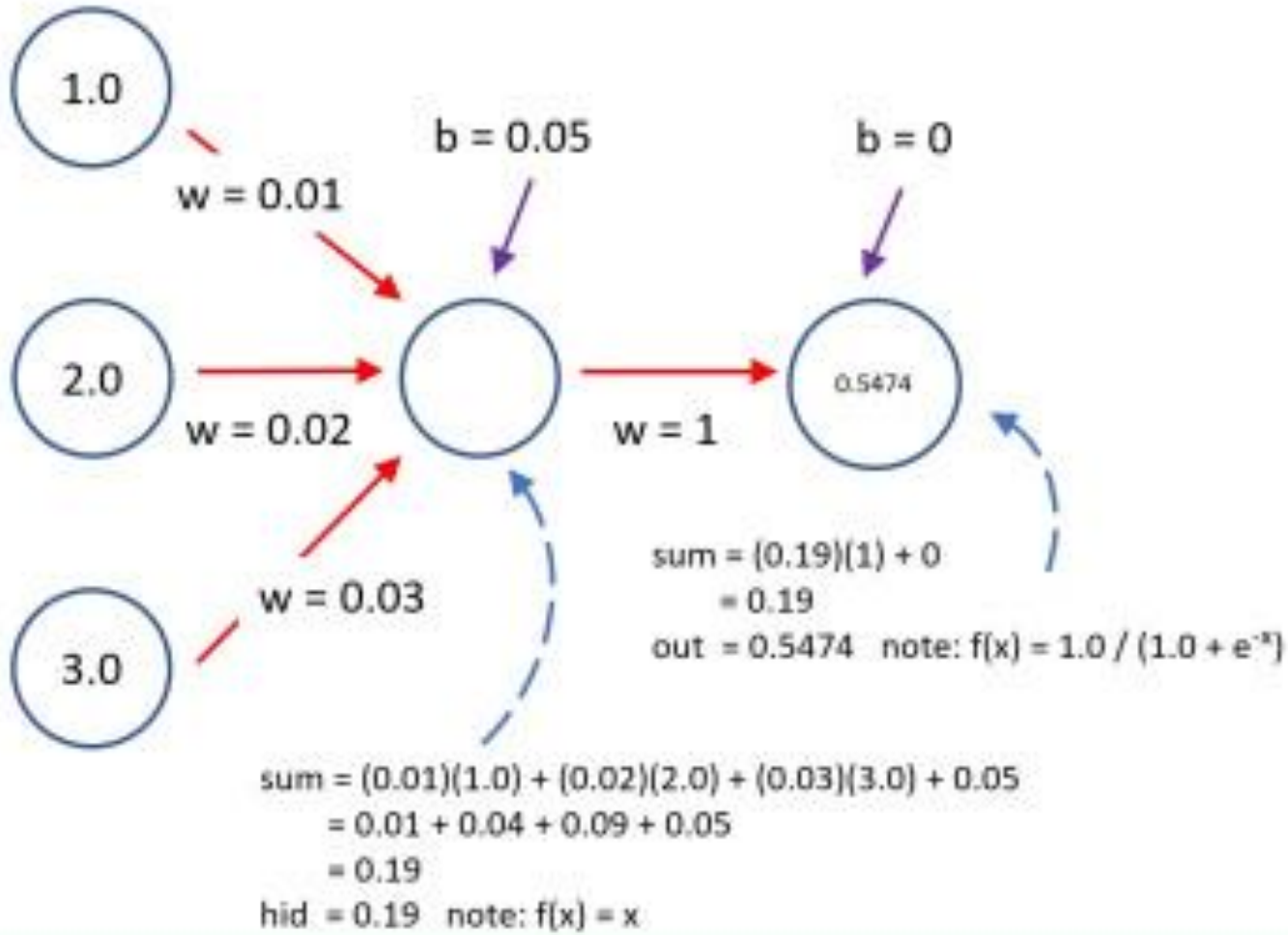
$w_1 = 1.0$ $a_1 = 0.01$
$w_2 = 2.0$ $a_2 = 0.02$
$w_3 = 3.0$ $a_3 = 0.03$
$b = 0.05$

$z = (0.05) + (0.01)(1.0) + (0.02)(2.0) + (0.03)(3.0)$
$= 0.05 + 0.01 + 0.04 + 0.09$
$= 0.19$

$p = 1.0 / (1.0 + e^{-0.19})$
$= 0.5474$ (predicted class = 1)

Neural Network

single hidden layer, identity activation f(x) = x
single output node, logistic sigmoid activation f(x) = $1 / (1 + e^{-x})$



1. Forward Propagation Equation

$$z^{(i)} = w^T x^{(i)} + b$$

$$\hat{y}^{(i)} = a^{(i)} = sigmoid(z^{(i)})$$

Loss $$\mathrm{L}(a^{(i)}, y^{(i)}) = -y^{(i)}\log(a^{(i)}) - (1 - y^{(i)})\log(1 - a^{(i)})$$

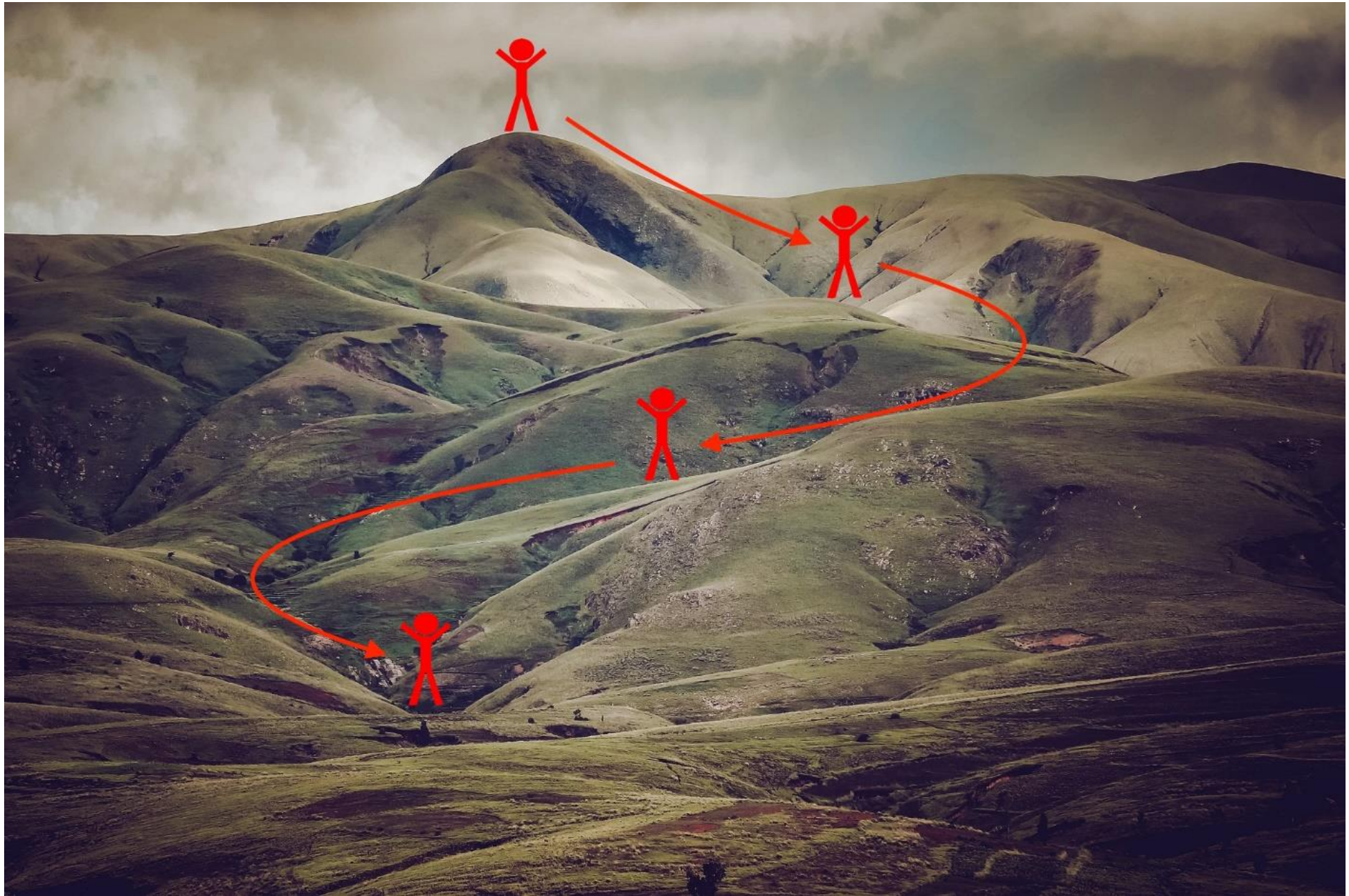
Cost $$J = \frac{1}{m}\sum_{i=1}^{m} \mathrm{L}(a^{(i)}, y^{(i)})$$

2. Backward Propagation

$$\frac{\partial J}{\partial w} = \frac{1}{m} X (A - Y)^T$$

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} (a^{(i)} - y^{(i)})$$

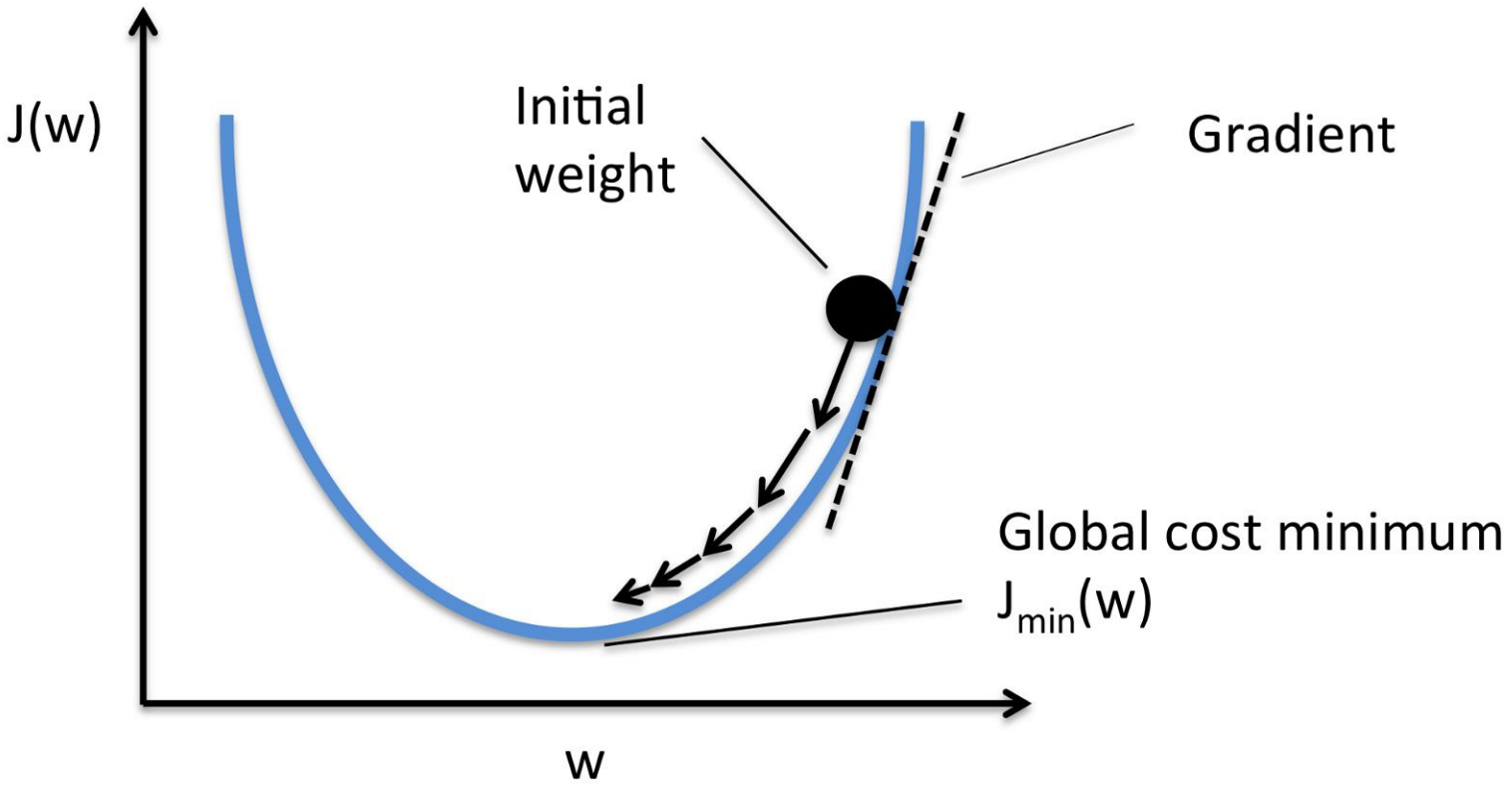
Gradient Descent

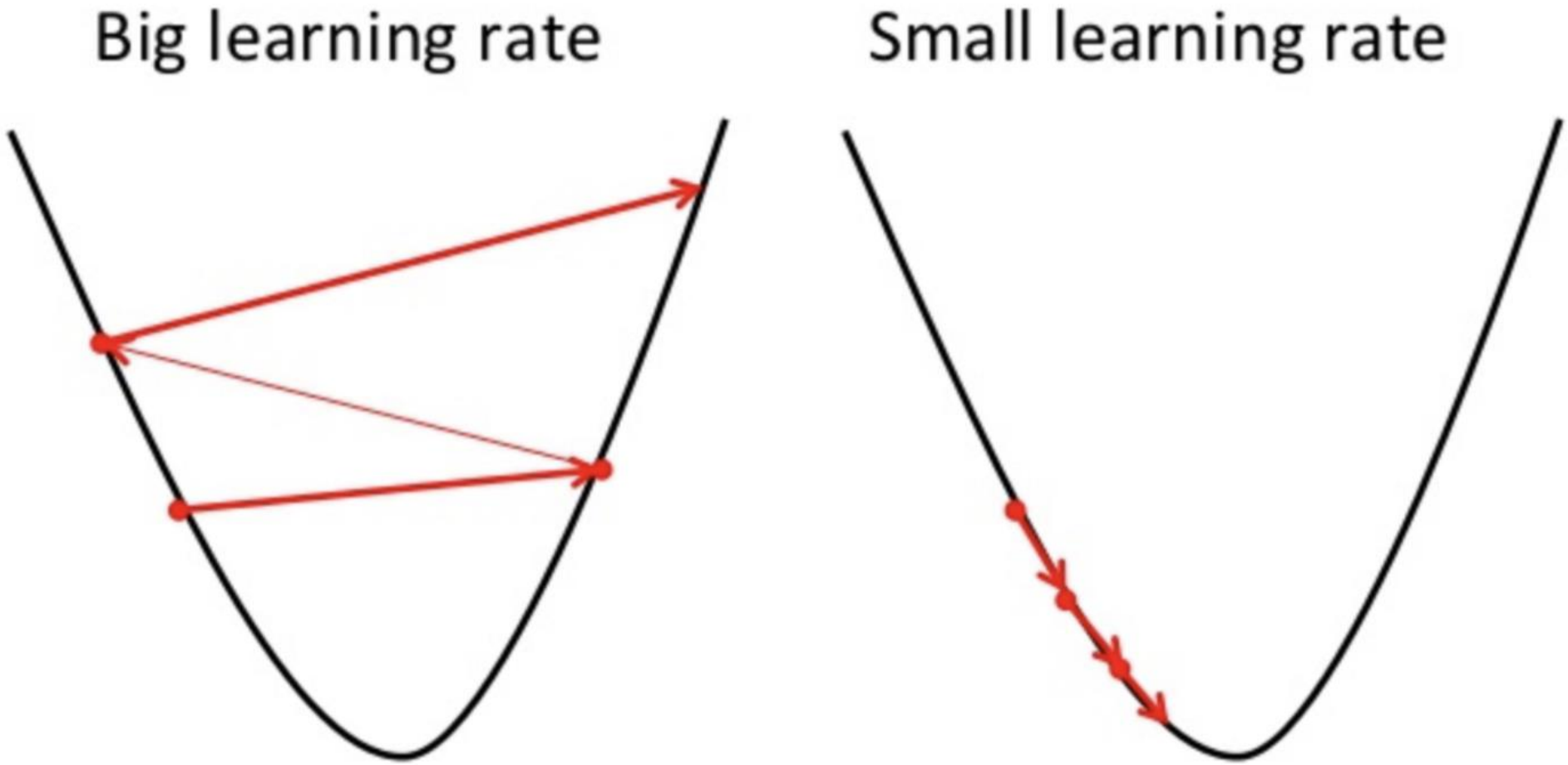


3. Update parameters (gradient descent)

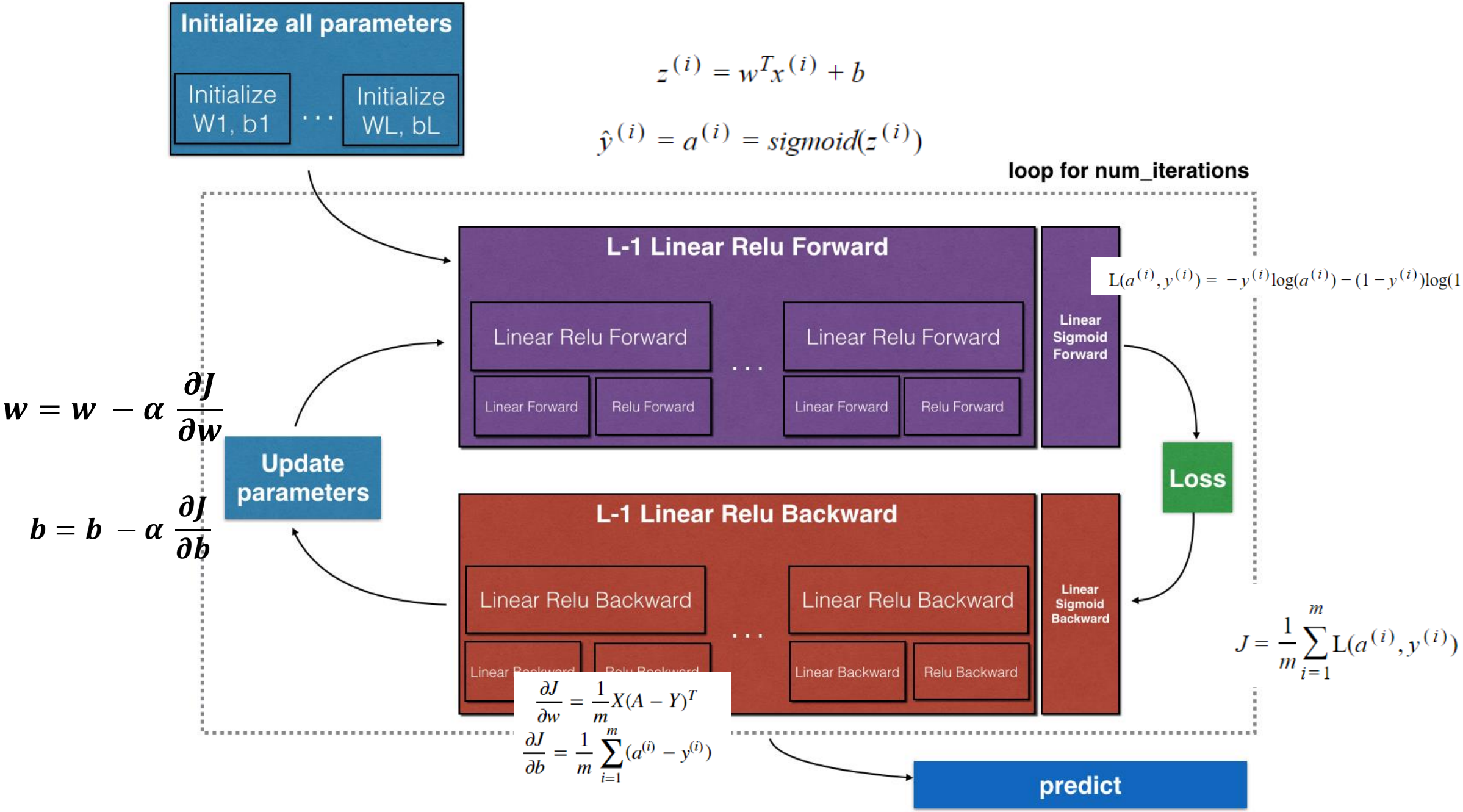
$$w = w - \alpha \frac{\partial J}{\partial w}$$

$$b = b - \alpha \frac{\partial J}{\partial b}$$





$$\alpha = 0.1 \; or \; 0.01 \; or \; 0.001 \; or \; 0.0001 \; or \; less(Some \; times)$$



model(X_train, Y_train, X_test, Y_test, num_iterations = 2000, learning_rate = 0.5, print_cost = False)

Input

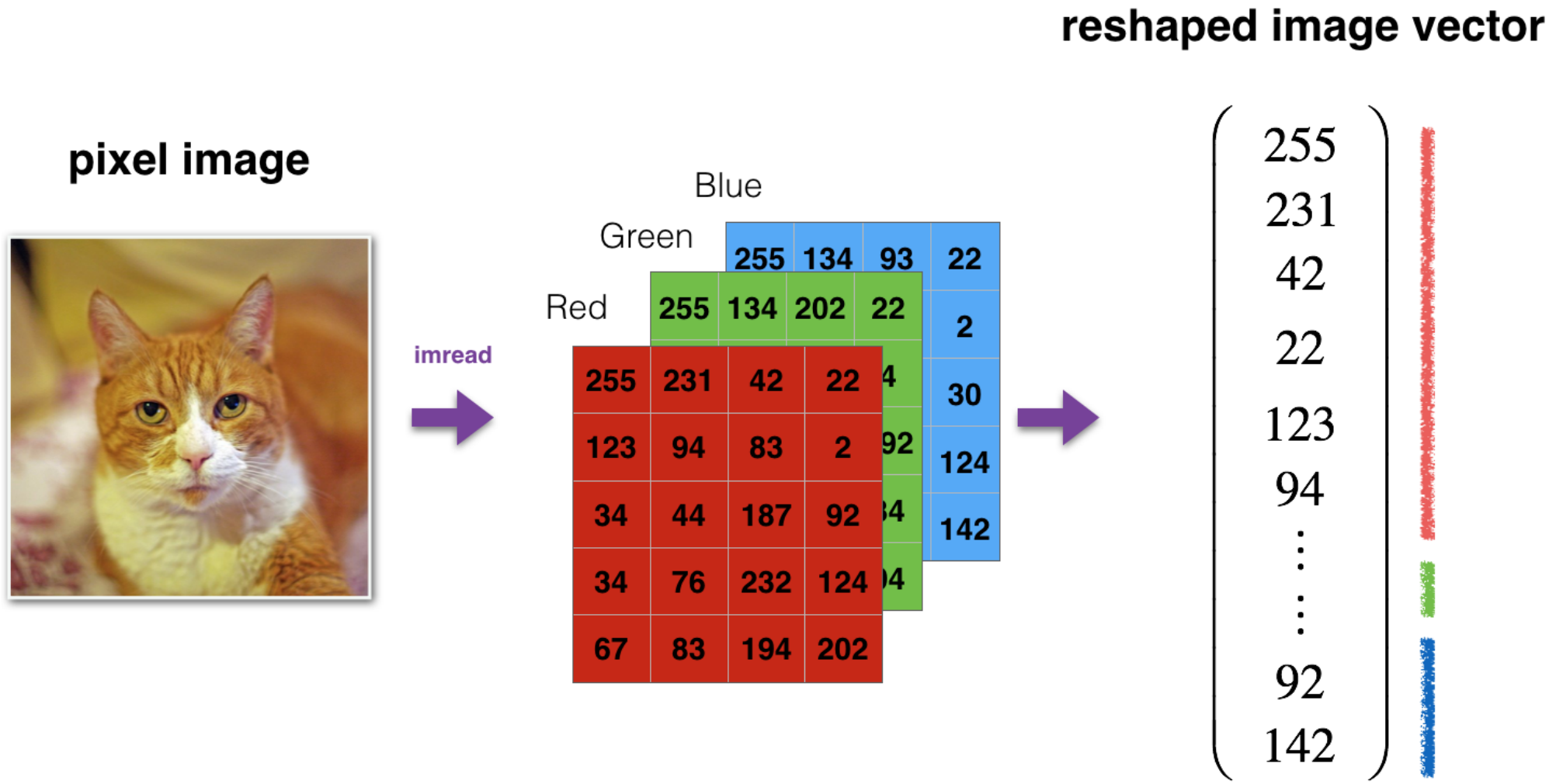

pixel image
imread
Blue
Green
Red
255 134 93 22
255 134 202 22
2
30
124
142
255 231 42 22
123 94 83 2
34 44 187 92
34 76 232 124
67 83 194 202
reshaped image vector
255
231
42
22
123
94
92
142

image2vector

$$\begin{pmatrix} 255 \\ 231 \\ \dots \\ 94 \\ 142 \end{pmatrix}$$

/255 /255 /255 /255

$x_0^{(i)}$ $x_1^{(i)}$ $\dots$ $x_{12286}^{(i)}$ $x_{12287}^{(i)}$

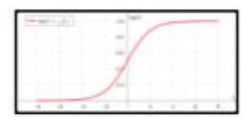
w_0 w_1 $\dots$ w_{12286} w_{12287}

$w^T x^{(i)} + b$ | σ

0.73

0.73 > 0.5

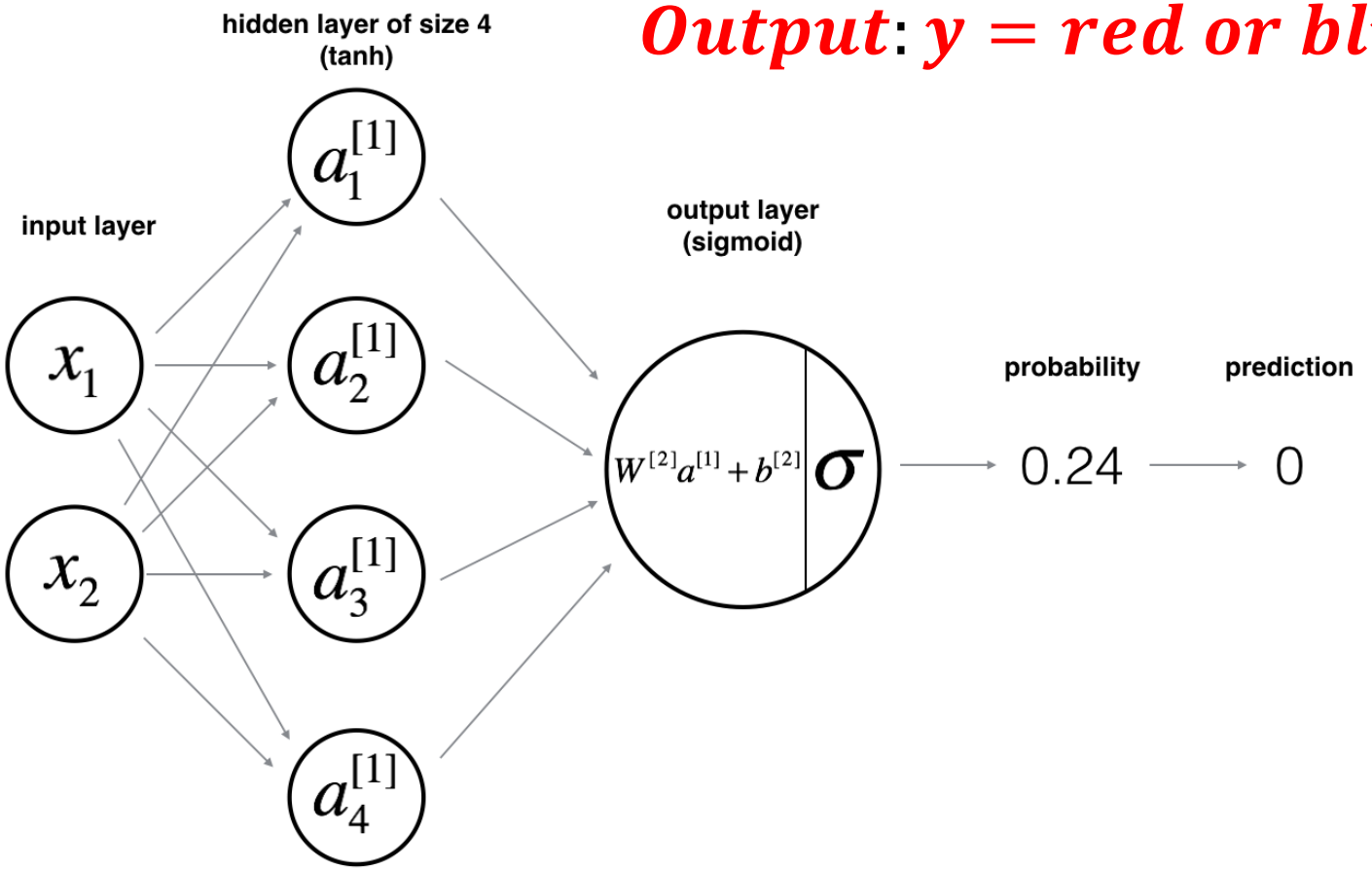
"it's a cat"

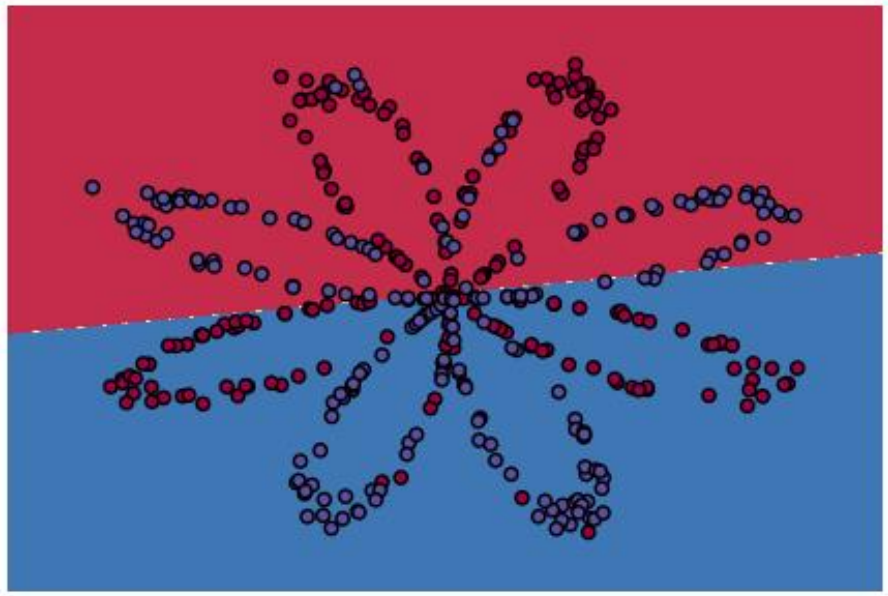


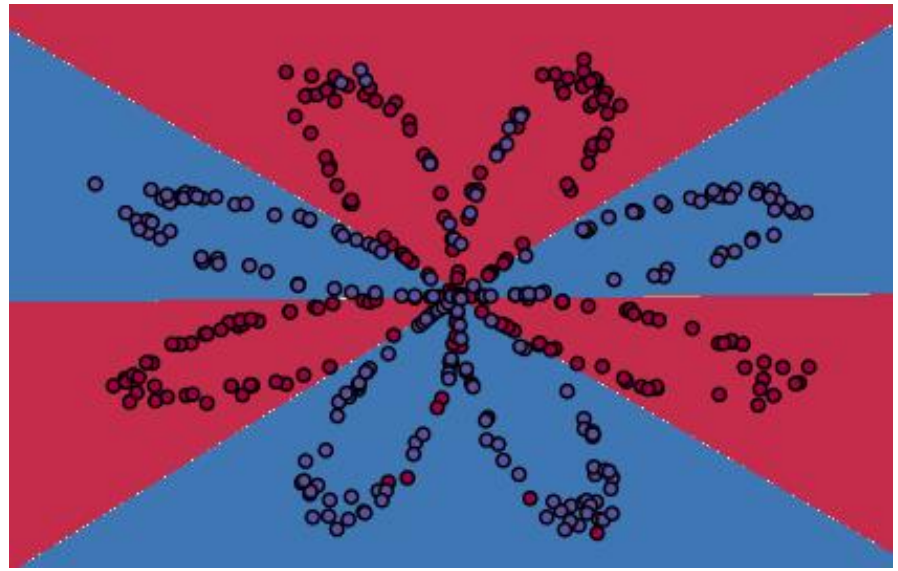
2 layer NN (1 HL + 1 OL)

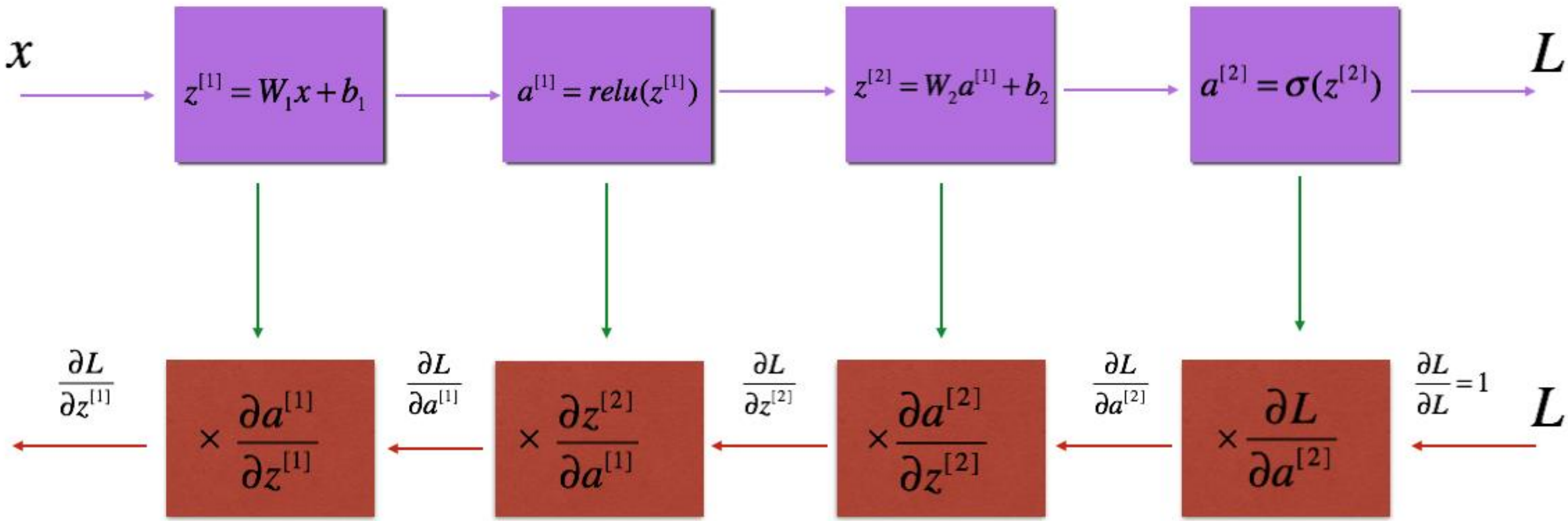
$$Input: x1 = x$$
$$x2 = y$$

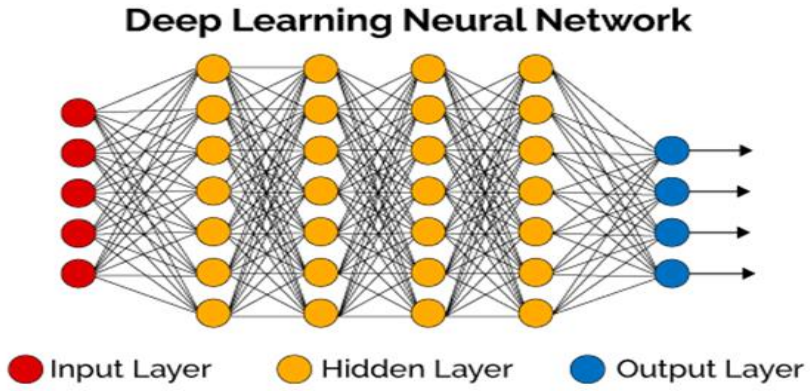
$$Output: y = red\ or\ blue$$








x
$z^{[1]} = W_1 x + b_1$
$a^{[1]} = relu(z^{[1]})$
$z^{[2]} = W_2 a^{[1]} + b_2$
$a^{[2]} = \sigma(z^{[2]})$
L
$\frac{\partial L}{\partial L} = 1$
L
$\times \frac{\partial L}{\partial a^{[2]}}$
$\frac{\partial L}{\partial a^{[2]}}$
$\times \frac{\partial a^{[2]}}{\partial z^{[2]}}$
$\frac{\partial L}{\partial z^{[2]}}$
$\times \frac{\partial z^{[2]}}{\partial a^{[1]}}$
$\frac{\partial L}{\partial a^{[1]}}$
$\times \frac{\partial a^{[1]}}{\partial z^{[1]}}$
$\frac{\partial L}{\partial z^{[1]}}$



CLASSIC ML

100 - 10000 SAMPLES

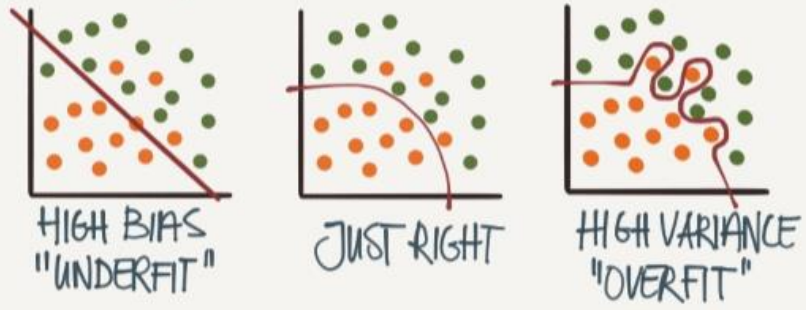
60% 20% 20%

DEEP LEARNING

1M SAMPLES

98% 1% 1%

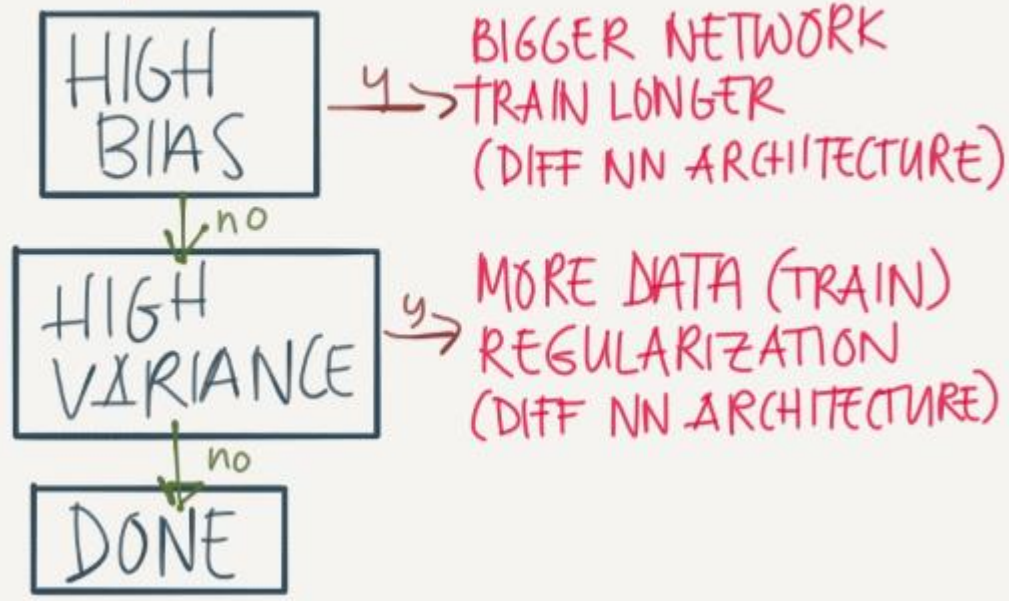
BIAS/VARIANCE



	ERROR			
TRAIN	1%	15%	15%	0.5%
TEST	11%	16%	30%	1%
	HIGH VARIANCE	HIGH BIAS	HIGH BIAS & VARIANCE	LOW BIAS & VARIANCE

ASSUMING HUMANS GET 0% ERROR

THE ML RECIPE



Parameter and Hyperparameter

Parameter - Weight and Bias

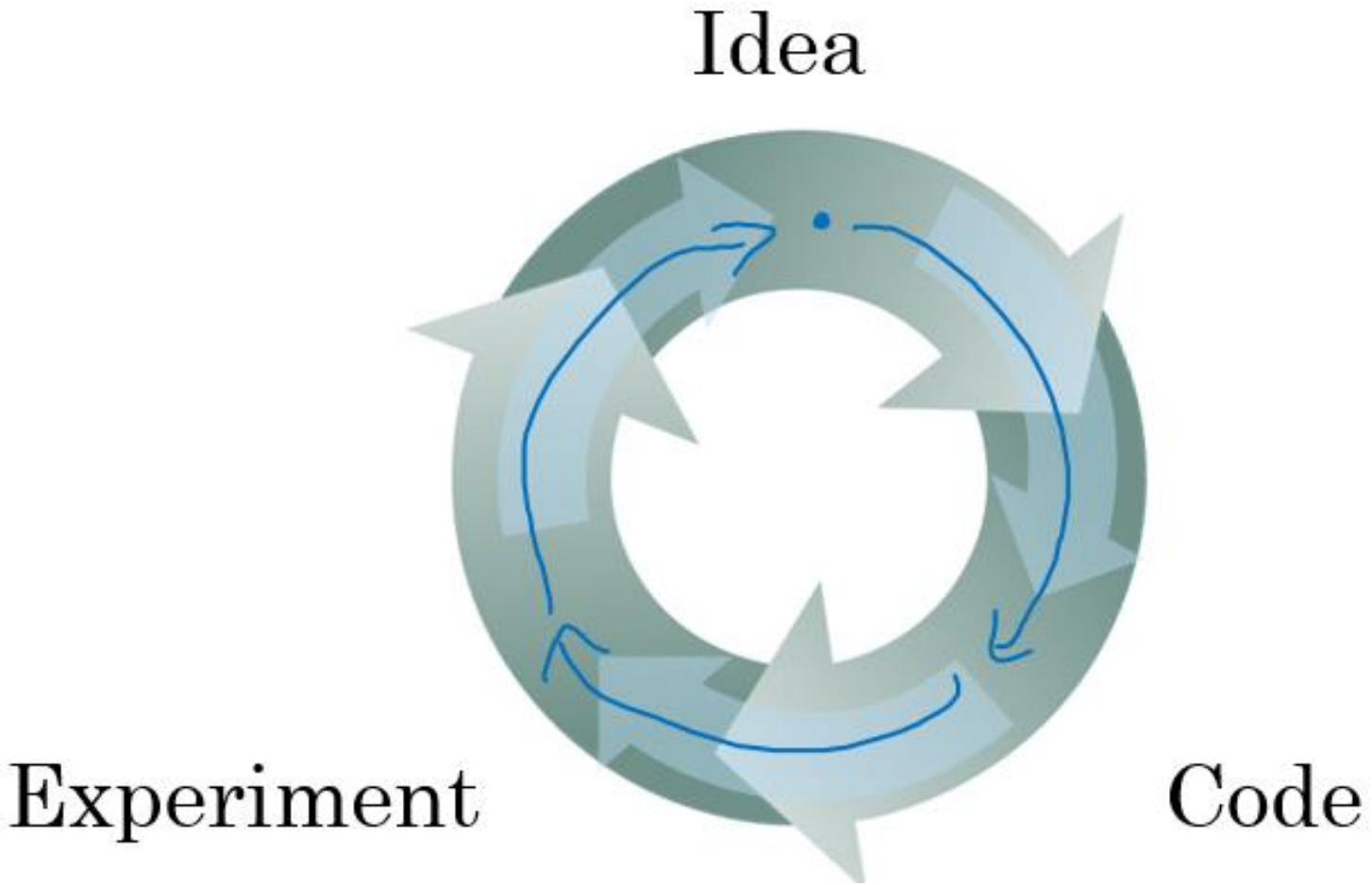
Hyperparameter (parameters that control the algorithm):

1) No of hidden layers - Try and error
2) No of neurons in each hidden layer - Try and error
3) Learning rate. (The most important parameter) – 0.1, 0.01, 0.001, 0.0001, .00001
4) Activation functions – Sigmoid, Relu, Leaky Relu, Tanh
5) Number of iteration – Try and error
6) Epoch – Try and error
7) Batch size – 4, 8, 16, 32, 64, 128… (Power of 2)
8) Regularization – Dropout, L1, L2
9) Regularization Rate and Dropout lembda
10) Normalizing inputs – Min-max, mean, Z-Score
11) Weights and Bias Initialization – Zero, Random, He
12) optimization algorithm – SGD, ADAM, RMSProp
13) Learning rate decay
14) momentum – 0.9

Iterations, Epoch and Batch Size

- **batch size** = the number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.
- number of **iterations** = number of passes, each pass using [batch size] number of examples. To be clear, one pass = one forward pass + one backward pass (we do not count the forward pass and backward pass as two different passes).
- one **epoch** = one forward pass and one backward pass of *all* the training examples.
- **Example**: if you have 5000 training examples, and your batch size is 32, then it will take 157 iterations to complete 1 epoch.
- **5000/32 = 156.25 =157 iterations (156 x 32 + 1 x 8)**
- **If you set epoch = 20, 20 x 157 = 3140 iterations**

Empirical Process



Deep Learning Architectures

1. CNN (Convolutional Neural Network)

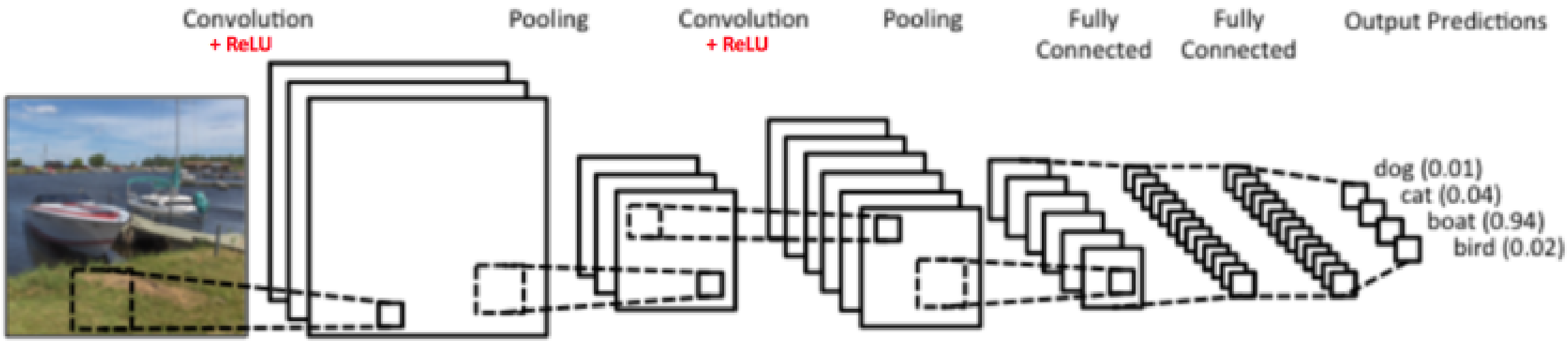
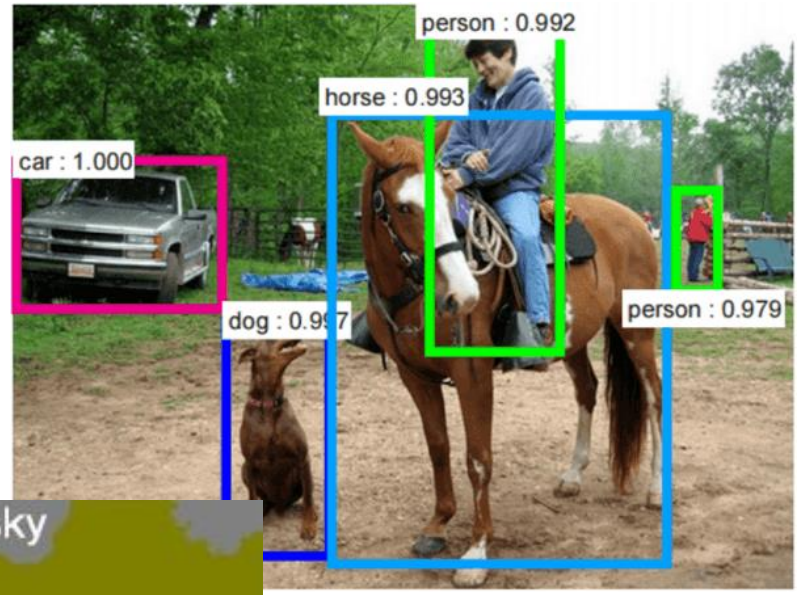
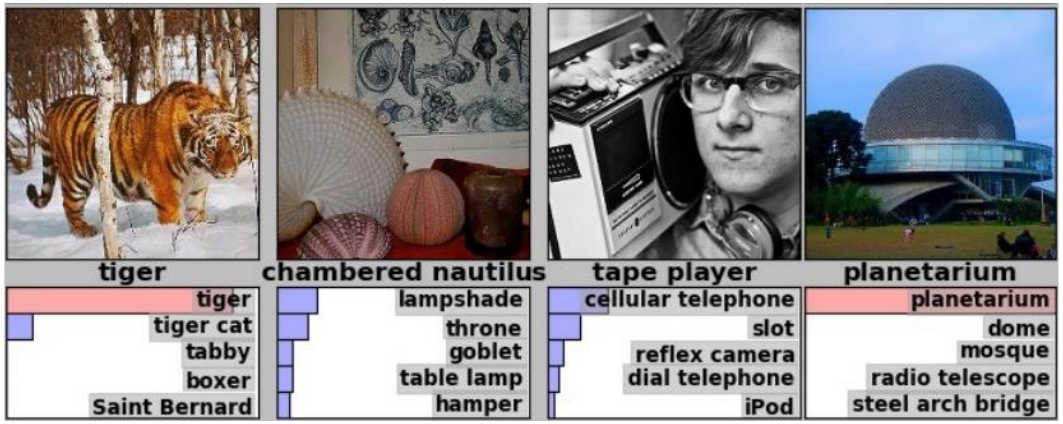
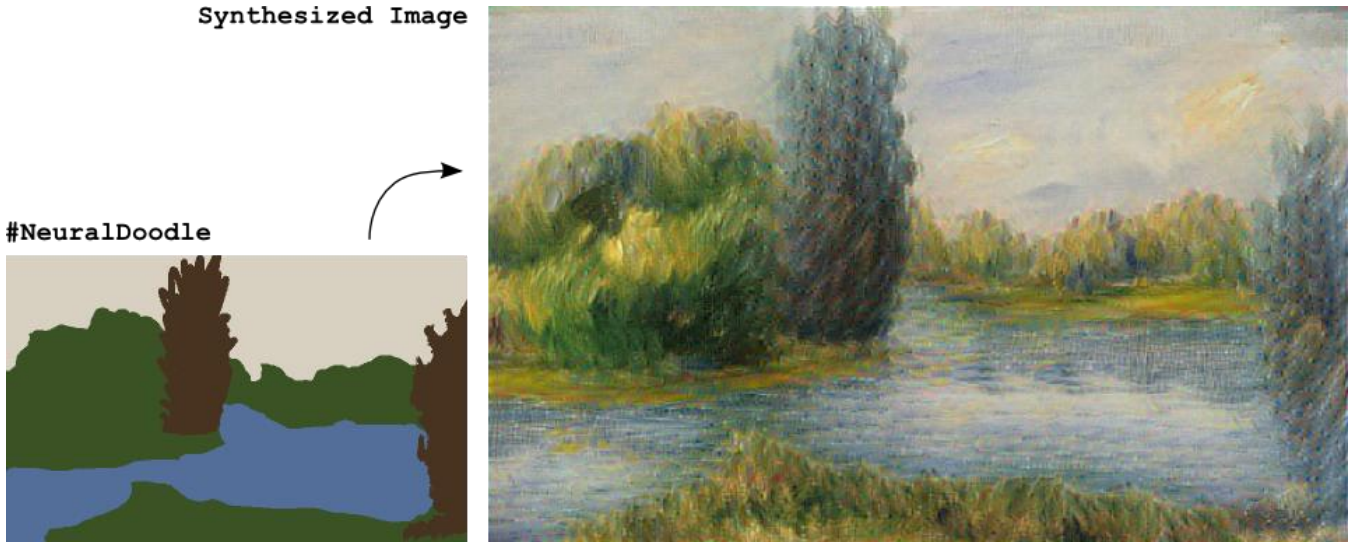


Image Source: https://www.clarifai.com/technology

Application of CNN

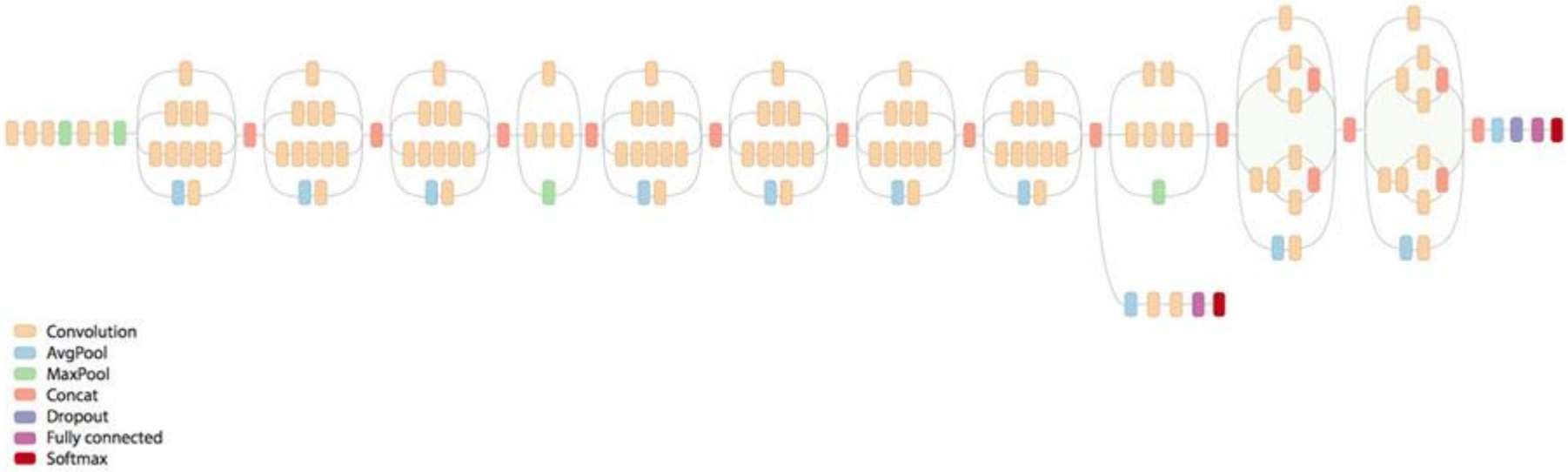


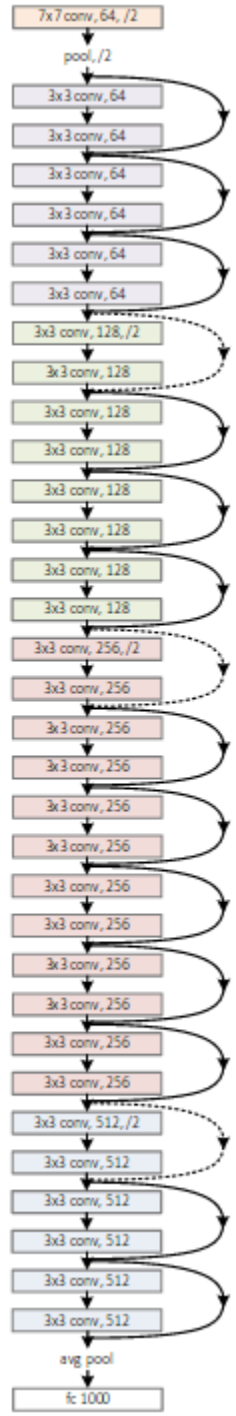




CNN Networks

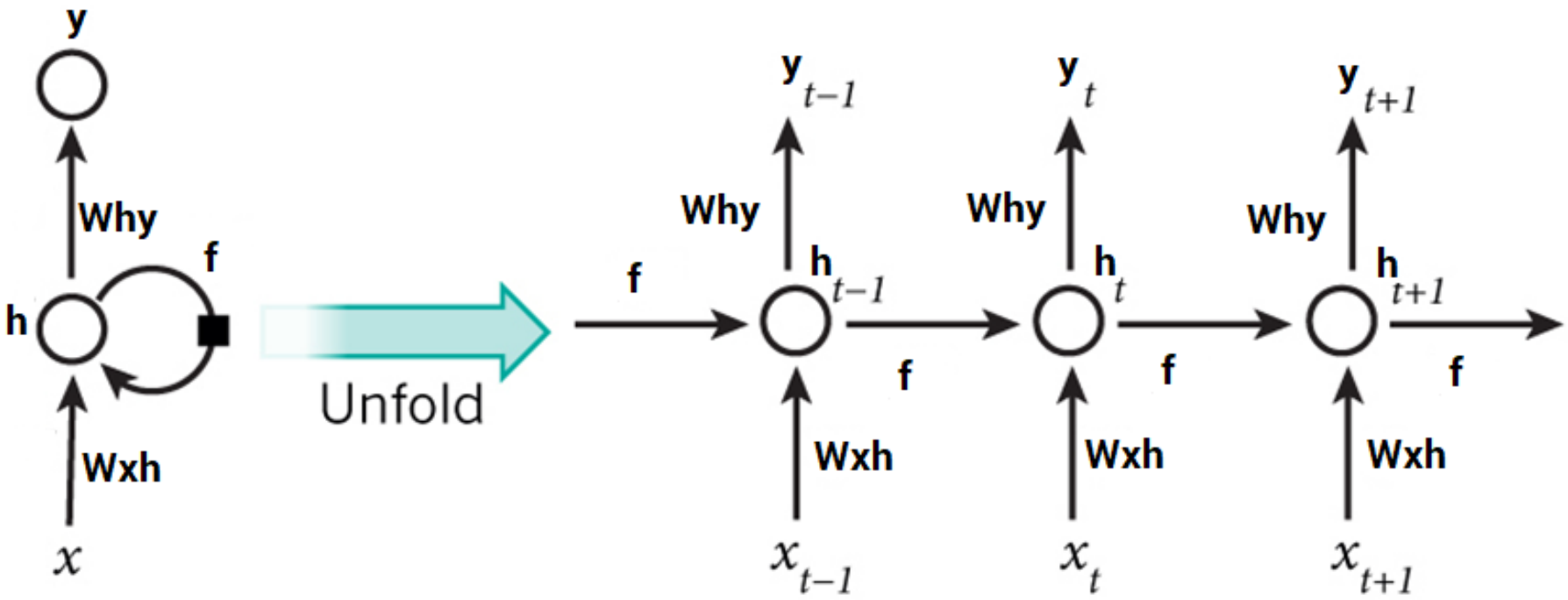
Google LeNet





ResNet

2. RNN (recurrent neural network)



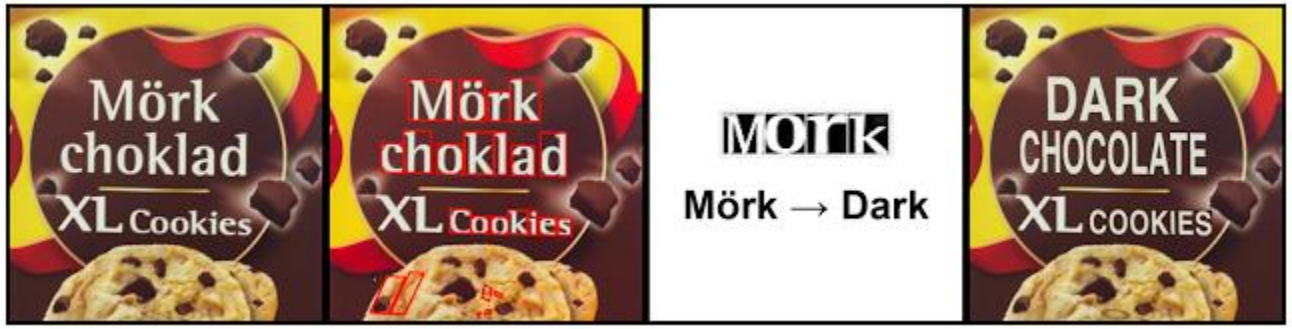
Application of RNN

1. **Automatically Adding Sounds To Silent Movies**
2. **Automatic Text Generation**

Machine learning Mastery

Madine Learning Mastery

Hachnne Learning Mastery

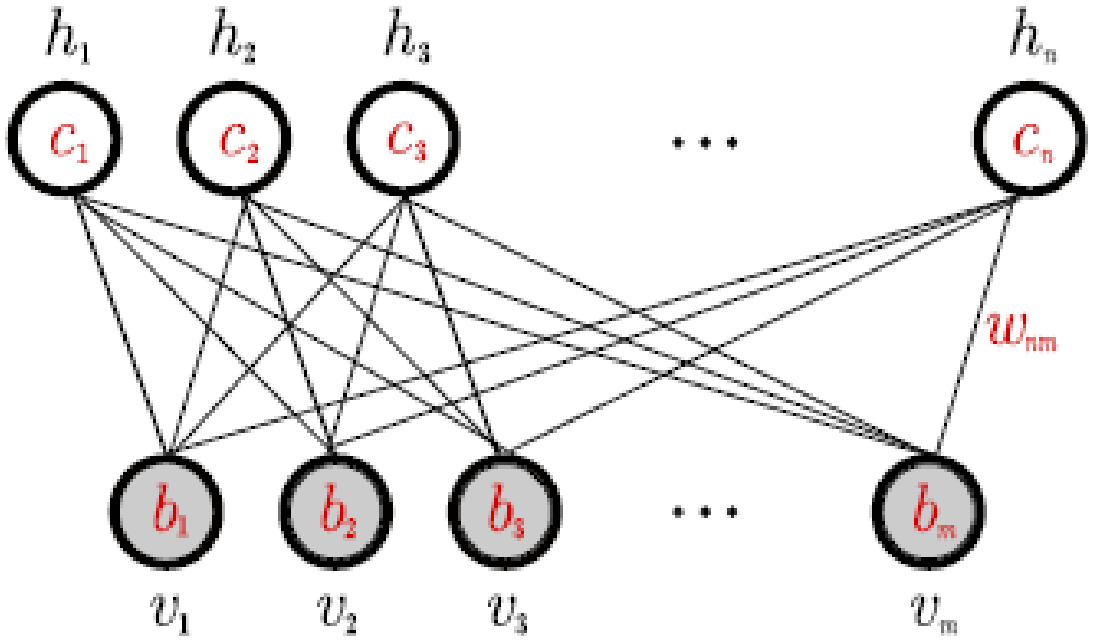


"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

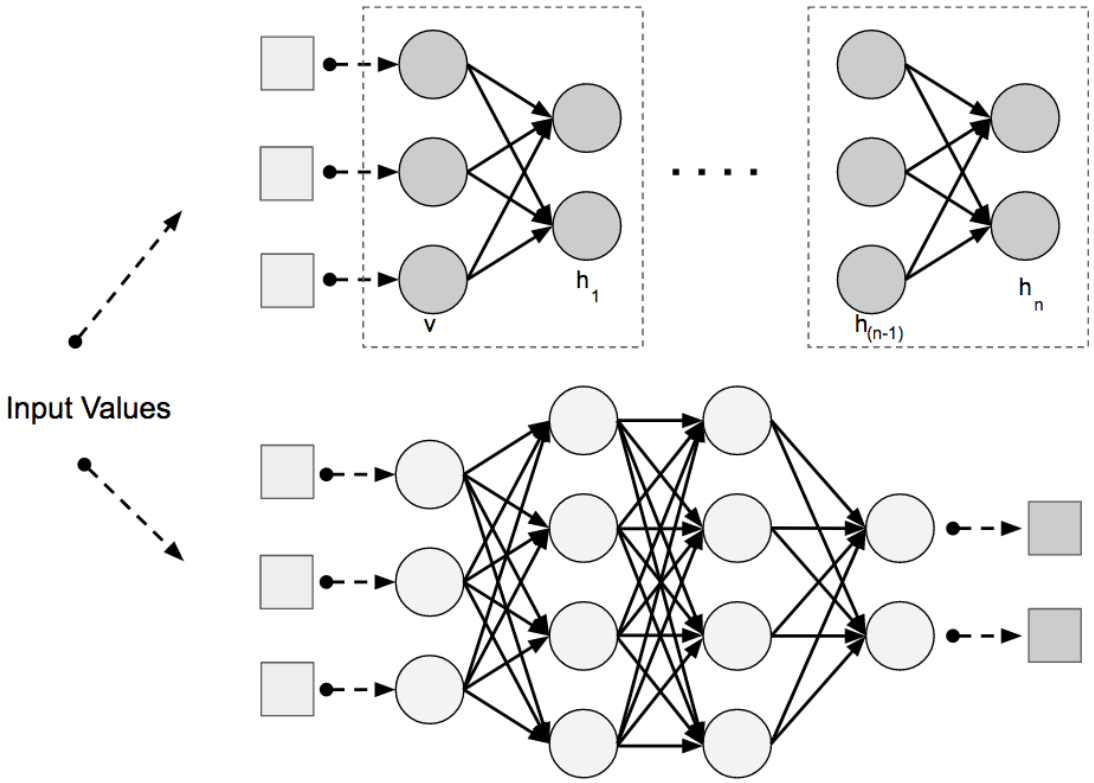
3. RBM (Restricted Boltzmann Machine)



Applications of RBM:

- Feature extraction
- Labelling of dataset
- Construct Deep Belief Networks

4. Deep Belief Network



1. Unsupervised pre-learning provides a good initialization of the network

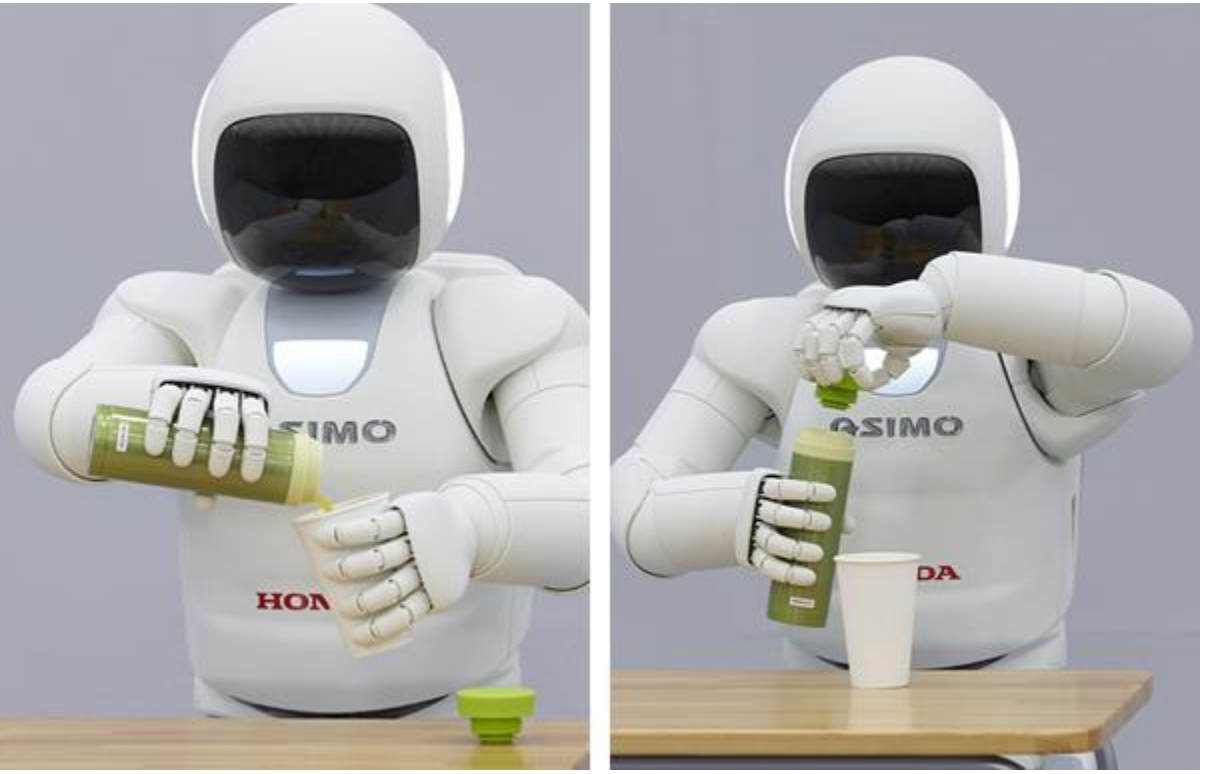
2. Supervised fine-tuning

Application:

- Image/ Face Recognition
- Video Sequence recognition
- Motion-capture data

NASA is using DBN to classify high-resolution, highly diverse satellite imagery.

General Applications of Deep Learning:



HOW UBER'S FIRST SELF-DRIVING CAR WORKS

Top mounted **LiDAR** beams 1.4 million laser points per second to create a 3D map of the car's surroundings.

There are **20 cameras** looking for braking vehicles, pedestrians, and other obstacles.

A **colored camera** puts LiDAR map into color so the car can see traffic light changes.

Antennae on the roof rack let the car position itself via GPS.

LiDAR modules on the front, rear, and sides help detect obstacles in blind spots.

A cooling system in the car makes sure everything runs without overheating.

SOURCE: Uber

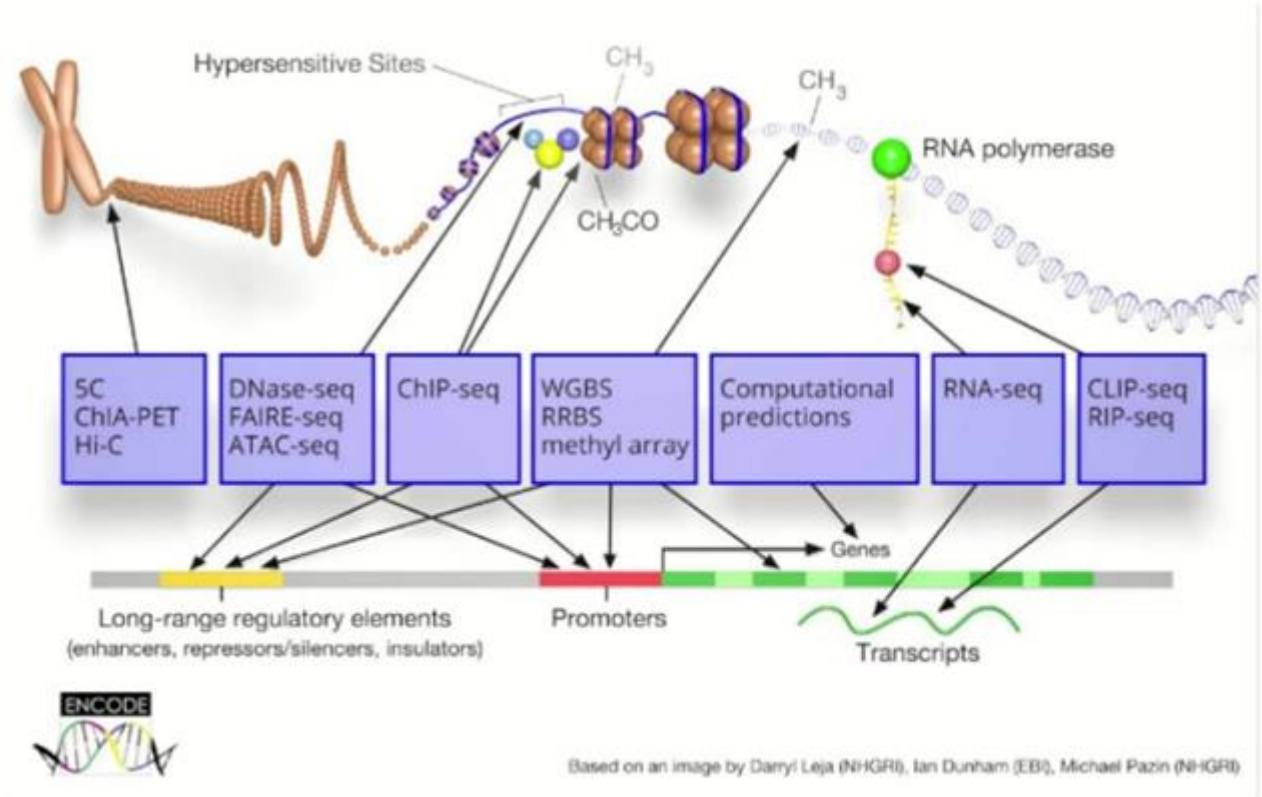
BUSINESS INSIDER

Deep Learning on Genomics Regulatory

DragoNN

The dragonn package implements deep neural networks (DNNs) for regulatory genomics, methods for DNN interpretation, and provides tutorials showcasing dragonn models using sequence simulations.

Anshul Kundaje



Challenges in Computer Vision

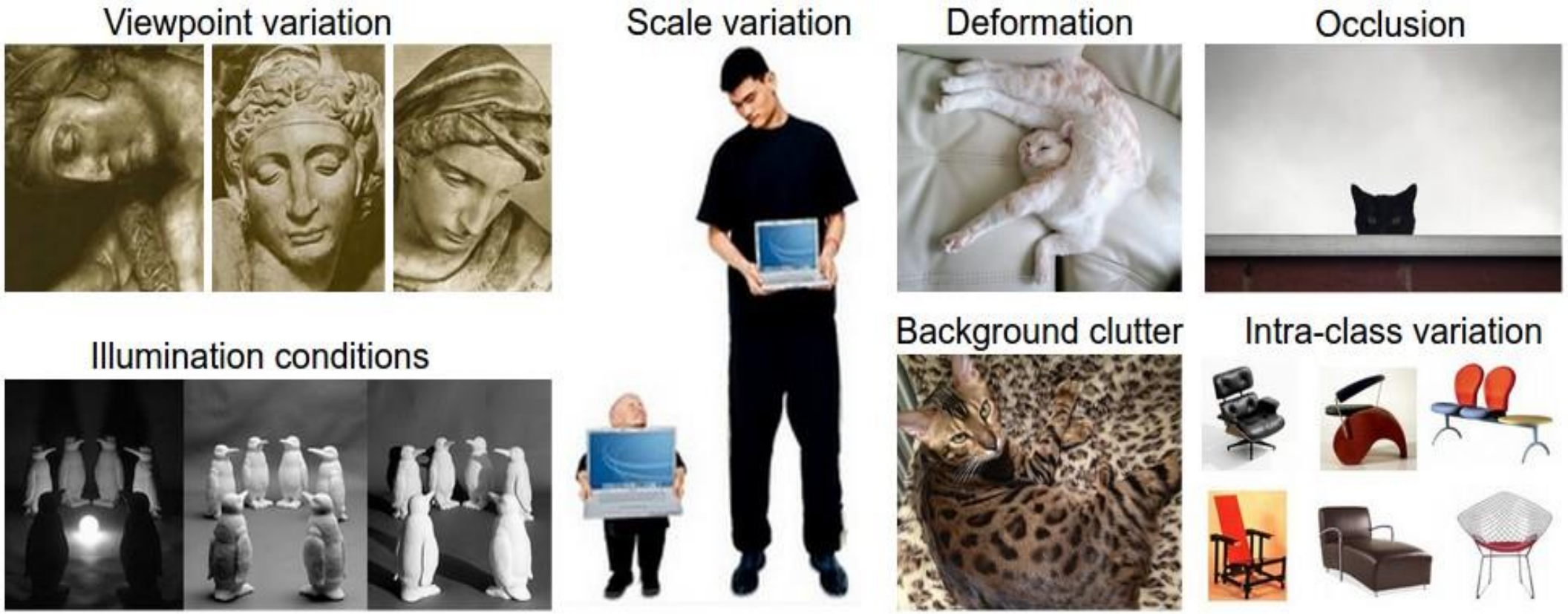


Image Source: http://cs231n.github.io/classification/

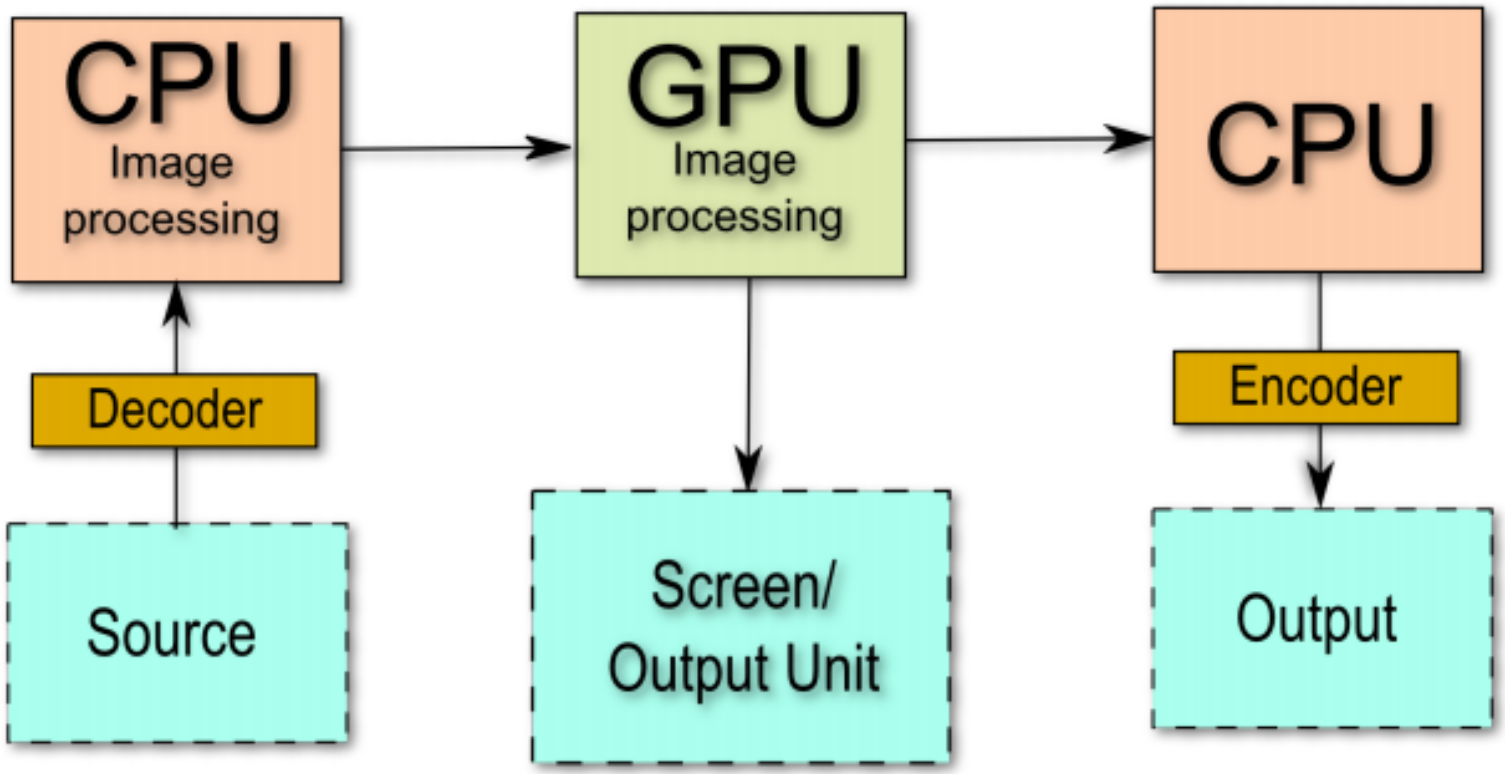
Practical issues with neural networks

- Data augmentation
- Data normalization
- Architecture optimization
- Loss function
- Weight initialization
- Learning rate and its decay schedule
- Overfitting: regularization, early stopping
- Momentum

Research issues in Deep Learning

- Huge amount of data
  - Overfitting in Neural Networks
  - Hyperparameter Optimization
- High-performance Hardware Requirement
- Essentially – It is a Blackbox
- Lack of Flexibility, Scalability and Multitasking
- Generative Models – learn and then generate new images
- Video Processing and Analytics
- Deep Learning in Embedded System
- Generalization of algorithms – (Partial Solution: Transfer Learning)
- CPU to GPU Data Transfer Time

A typical workflow of images/video processing



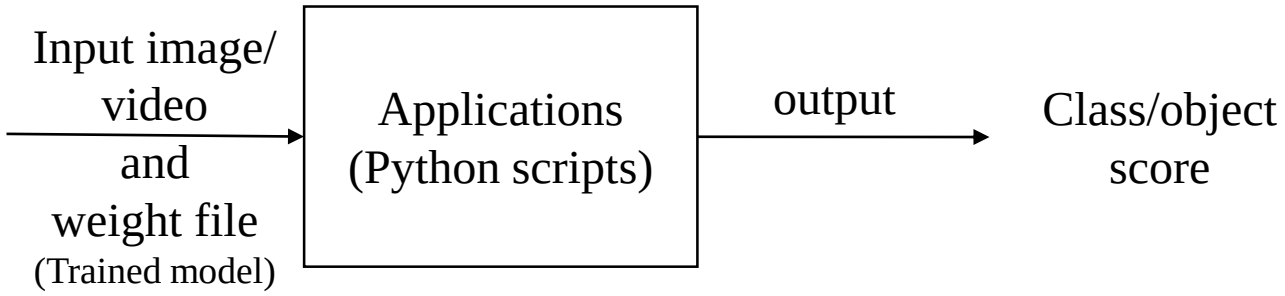
Deep Learning Library

Deep Learning Cloud Services

Comparison of Deep Learning Library

Software	Creator	Software license	Open source	Platform	Written in	Interface	CUDA support
TensorFlow	Google Brainteam	Apache 2.0	Yes	Linux, Mac OS, Windows	C++,Python	Python, (C/C++public API only for executing graphs)	Yes
Keras	François Chollet	MIT license	Yes	Linux, Mac OS, Windows	Python	Python	Yes
Deeplearning4j	Adam Gibson	Apache 2.0	Yes	Linux, Mac OS, Windows, Android (Cross-platform)	C, C++	Java, Scala	Yes
Caffe	Yangqing Jia	BSD License	Yes	Linux, Mac OS, Windows	C++	Python, Matlab	Yes
Theano	Université de Montréal	BSD License	Yes	Cross-platform	Python	Python	Yes
Torch	Ronan Collobert,Kavukcuoglu, Clement	BSD License	Yes	Linux, Mac OS, Windows, Android, iOS	C, Lua	Lua, C, utility library for C++/OpenCL	Yes
Microsoft Cognitive Toolkit - CNTK	Microsoft Research	MIT license	Yes	Windows, Linux	C++	Python, C++, Command line, BrainScript(.NET on roadmap)	Yes

Schematic Diagram

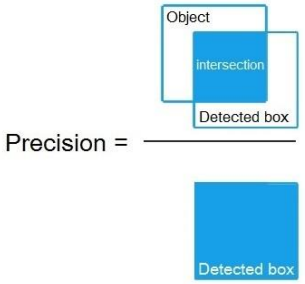


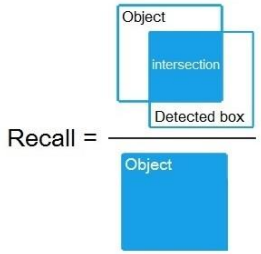
Evaluation Parameters

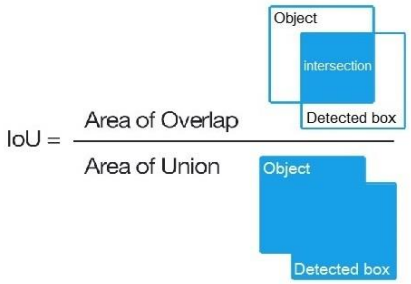
		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

- Accuracy – Confusion matrix
- Precision
- Recall
- F1 Score (try to maximize - to create a balanced classification model)
- mAP – IOU (For Object Detection)
- Loss/Cost – Minimize the difference between actual and predicated
- Entropy

$$recall = \frac{true\ positives}{true\ positives\ + false\ negatives} \qquad precision = \frac{true\ positives}{true\ positives + false\ positives}$$







$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

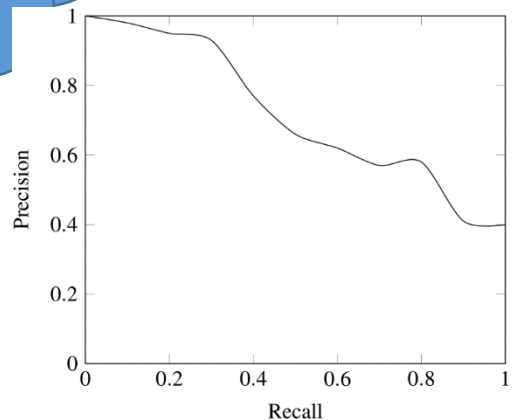
Dear, Do you remember all wedding anniversary surprises from me?

They are celebrating 6th wedding anniversary.

If you can recall all **5** gifts correctly, then, your recall ratio is **1.0** (**100%**). If you can recall 4 events correctly, your recall ratio is **0.8** (**80%**).

After some days, again she asked same question……

Most Dangerous Question!!!



For example, you answers 15 times, 10 times are correct and 5 times are wrong. This means you can recall all events but it's not so precise.

That means: (15 answers: 10 correct answers, 5 wrong answers), you get **100%** recall but your precision is only **66.67%** (10 / 15).

Facts of Deep Learning

- Majority research work of Deep Learning published in only reputed conference like CVPR, ECCV, ICCV not in journals.
- Cold war between image processing and deep learning researcher, because of the nature of deep leaning – It is BlackBox. No one is knowing "How it works from inside?"
- Very hard to publish paper in reputed journal. It gets publish only strong survey paper or hybrid branch paper like bioinformatics.
- It follows very old methods. It becomes popular due to huge amount of data and processing power.
- Deep Learning is empirical process. It takes time to get expertise.

Online DEMO

- https://playground.tensorflow.org

NVIDIA Titan Xp

(Grant by Nvidia Graphics Pvt. Ltd.)

❖ Configuration

- GPU Architecture – Pascal
- NVIDIA Cuda Cores – 3840
- Memory Size – 12 GB GDDR5X
- Memory Speed - 11.4 Gbps
- Nvidia Cuda Compute Capblity – 6.1
- VR Ready

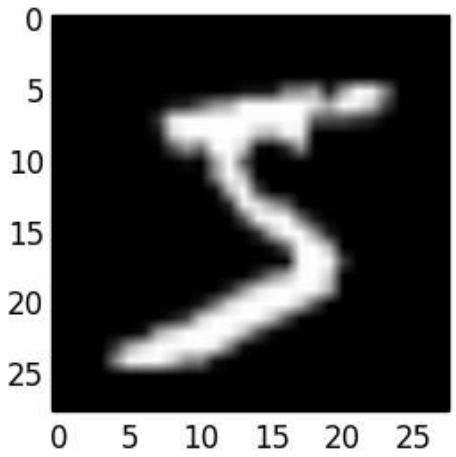
Work Station (Donated by CHARUSAT)

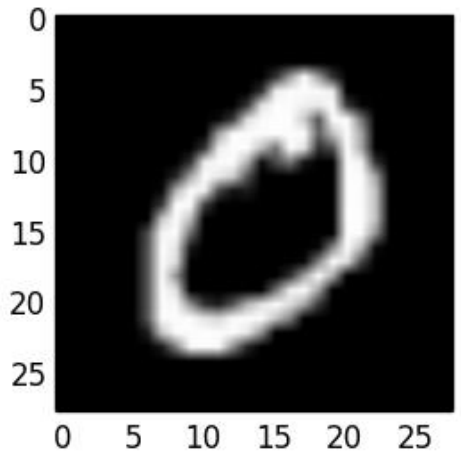
❖ **Configuration**

- 32 GB RAM
- Core i7 8th Generation CPU
- 2 TB HDD
- ASRock Mother Board Z series
- Cooler Master Cabinet
- 19’’ FHD LED
- Logitech Keyboard and Mouse

MNIST Dataset

- The MNIST database of **handwritten digits**, available from this page, has a **training set of 60,000** examples, and a **test set of 10,000** examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.
- Each Image Size – 28 x 28
- Color Channel – 1 (Black and White)





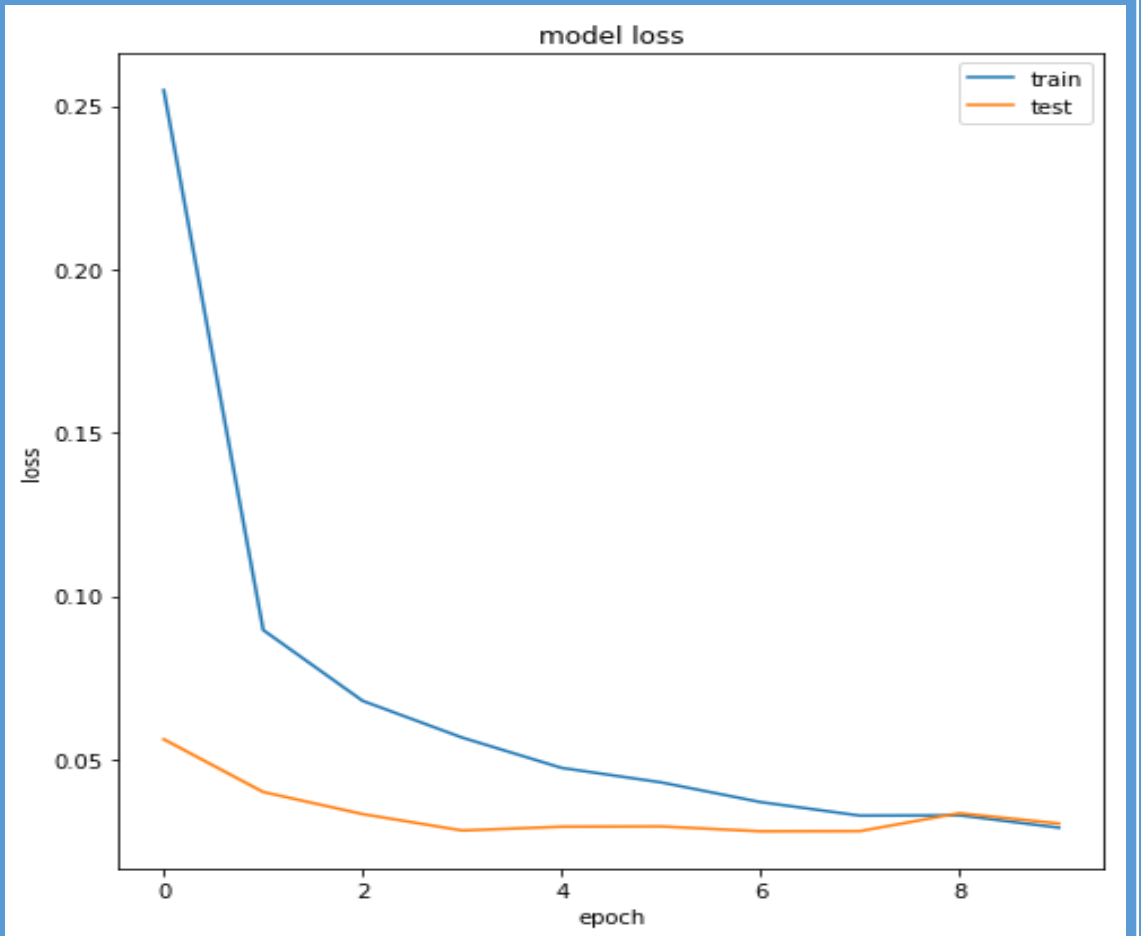
CNN Model

Implementation using Keras and Tensorflow as backend

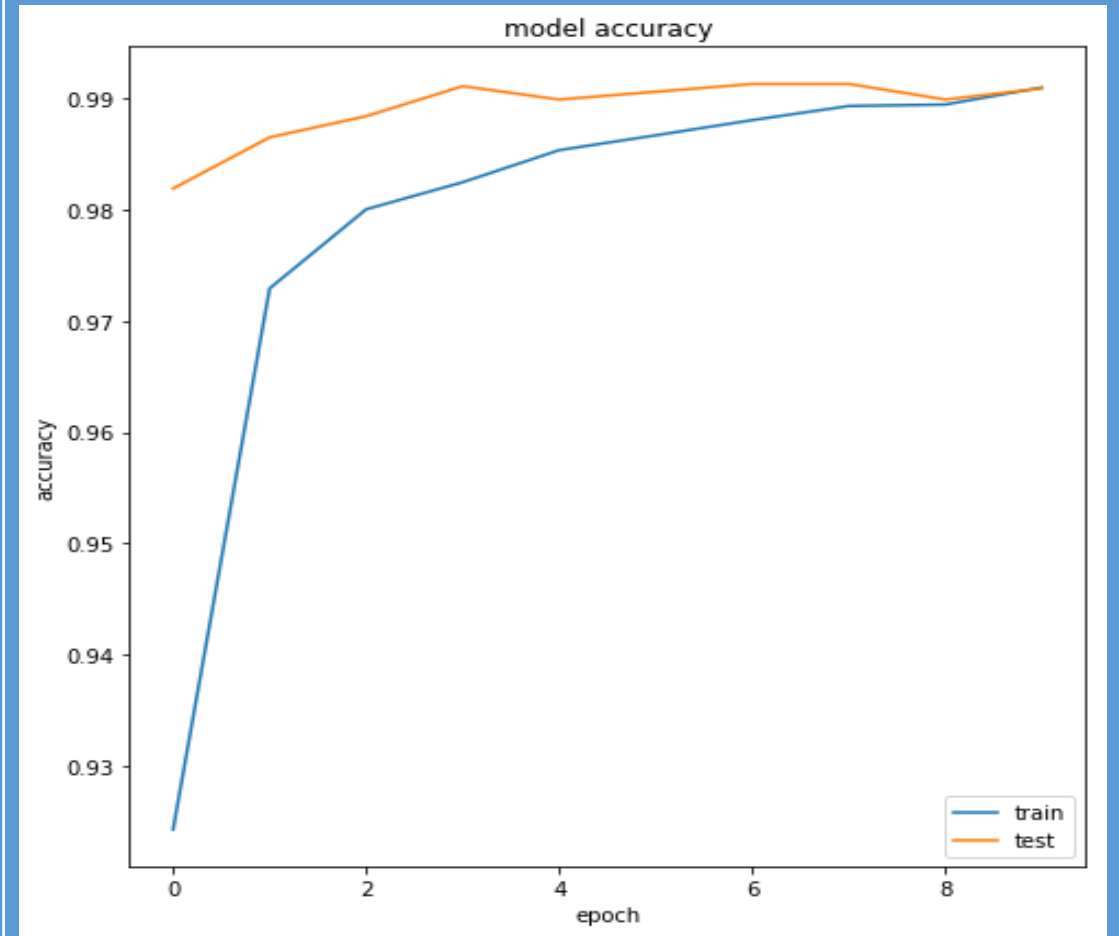
Layer (type)	Output Shape	Param #
conv2d_17 (Conv2D)	(None, 26, 26, 32)	320
conv2d_18 (Conv2D)	(None, 24, 24, 64)	18496
max_pooling2d_9 (MaxPooling)	(None, 12, 12, 64)	0
dropout_17 (Dropout)	(None, 12, 12, 64)	0
flatten_9 (Flatten)	(None, 9216)	0
dense_17 (Dense)	(None, 128)	1179776
dropout_18 (Dropout)	(None, 128)	0
dense_18 (Dense)	(None, 10)	1290

Total params: 1,199,882
Trainable params: 1,199,882
Non-trainable params: 0

Results in CPU of CNN Model

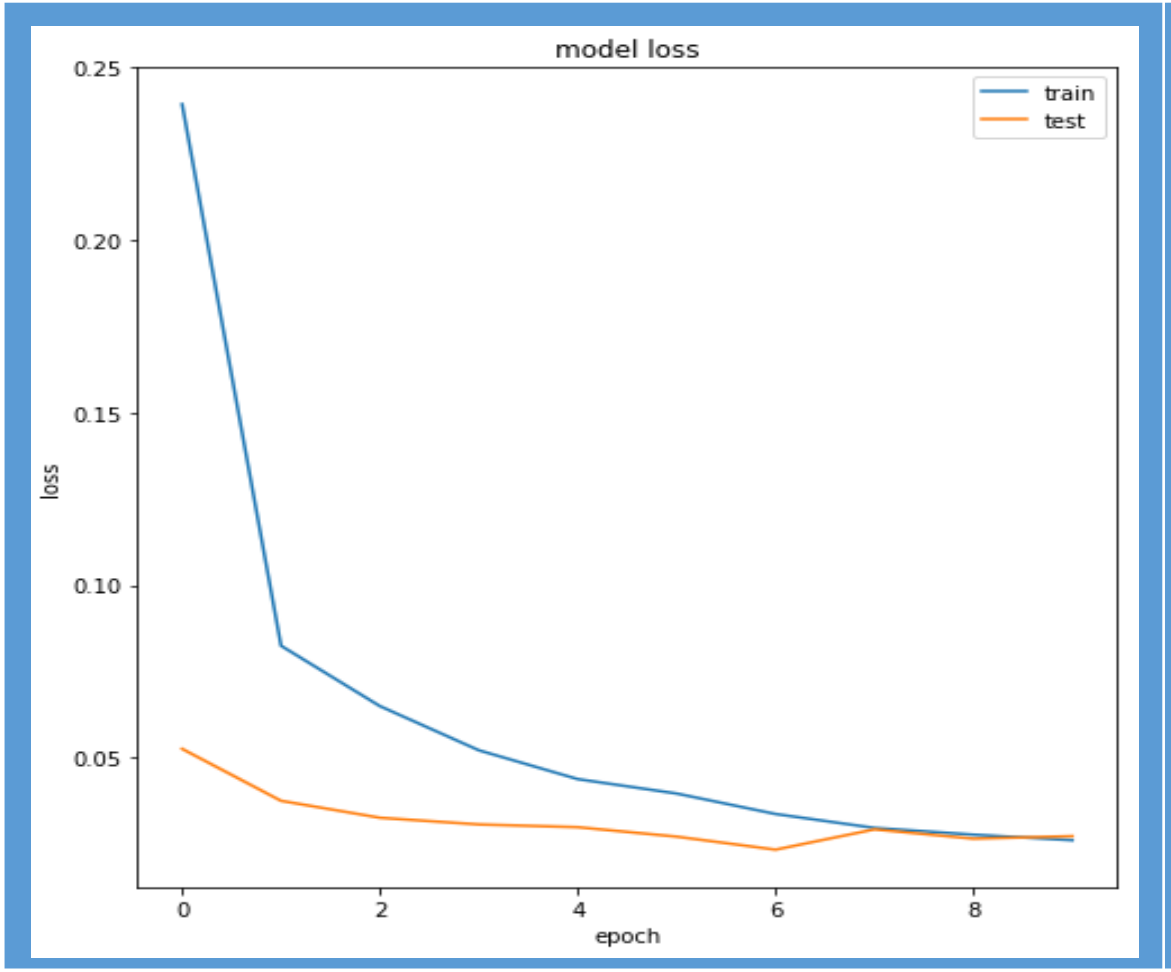


Model Loss vs. Epoch in CPU

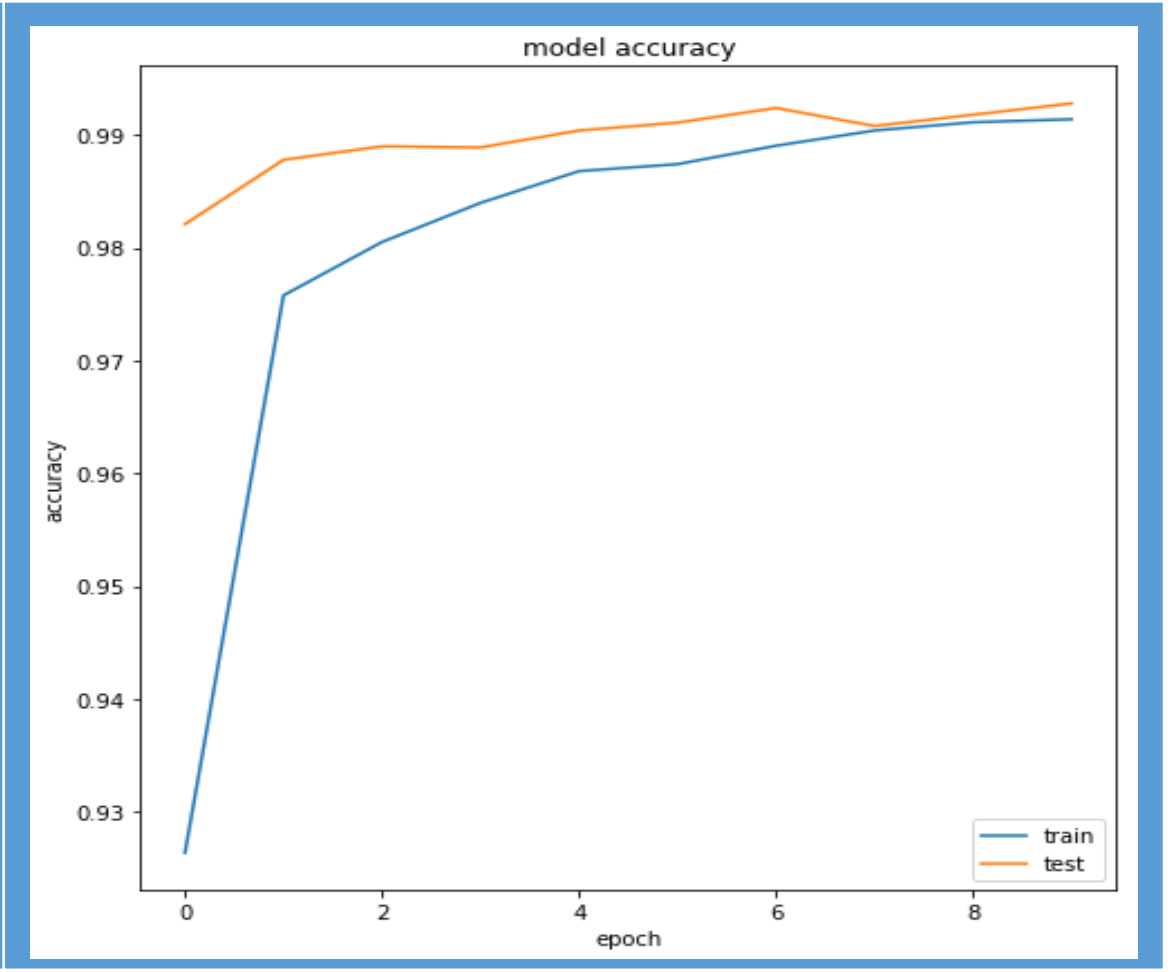


Model Accuracy vs. Epoch in CPU

Results in GPU of CNN Model

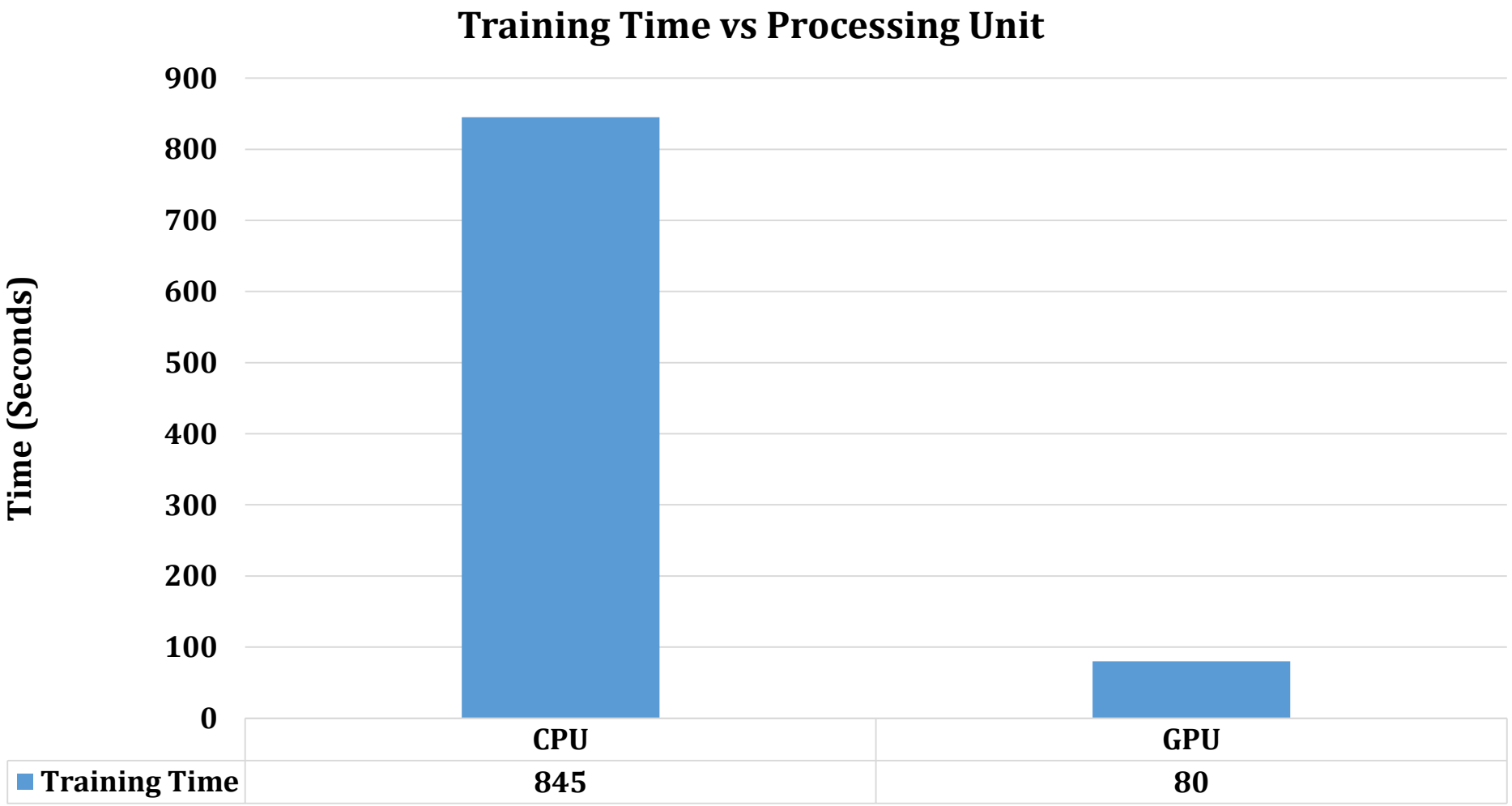


Model Loss vs. Epoch in GPU



Model Accuracy vs. Epoch in GPU

Comparison of training time CNN model in CPU and GPU

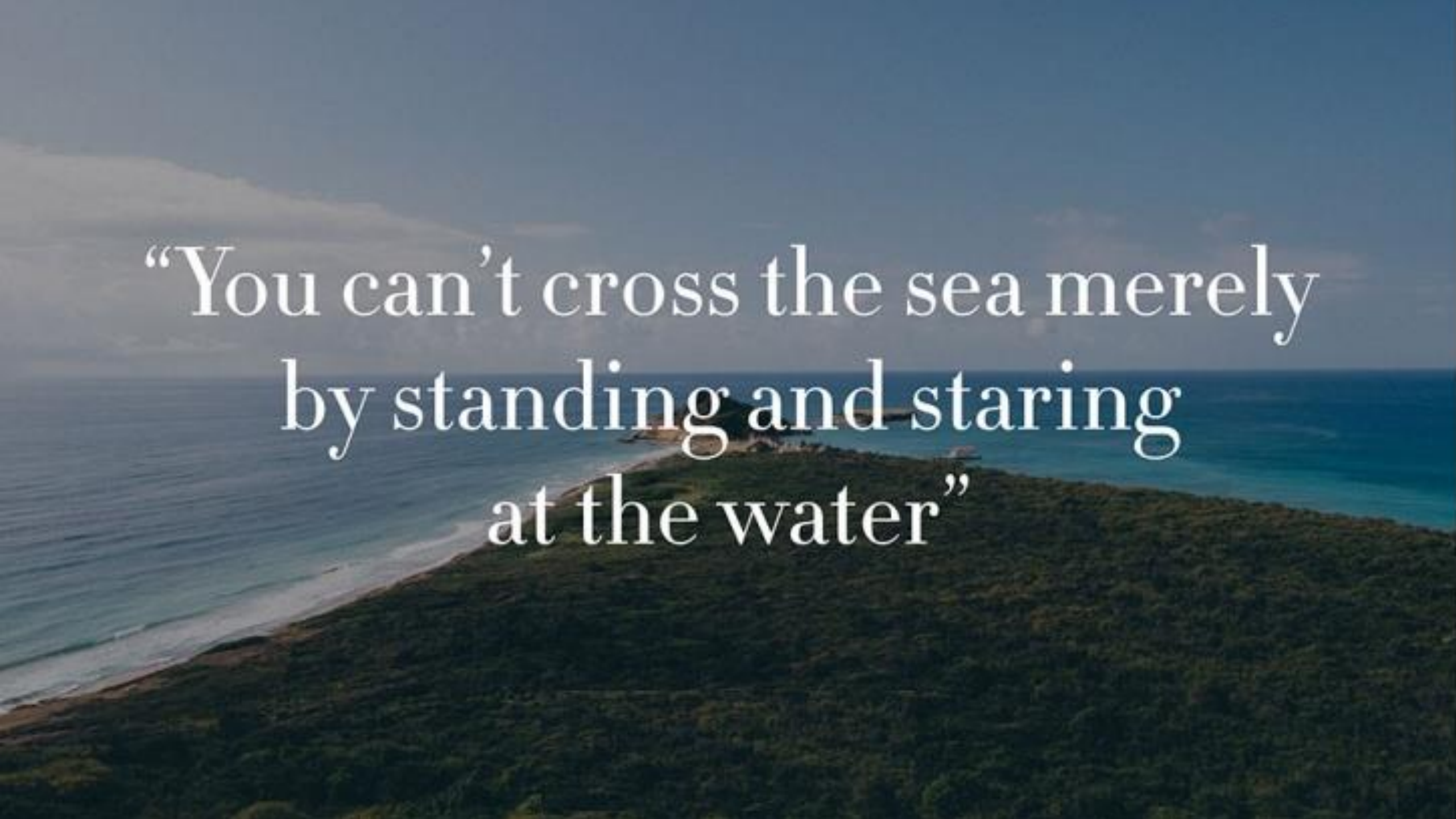


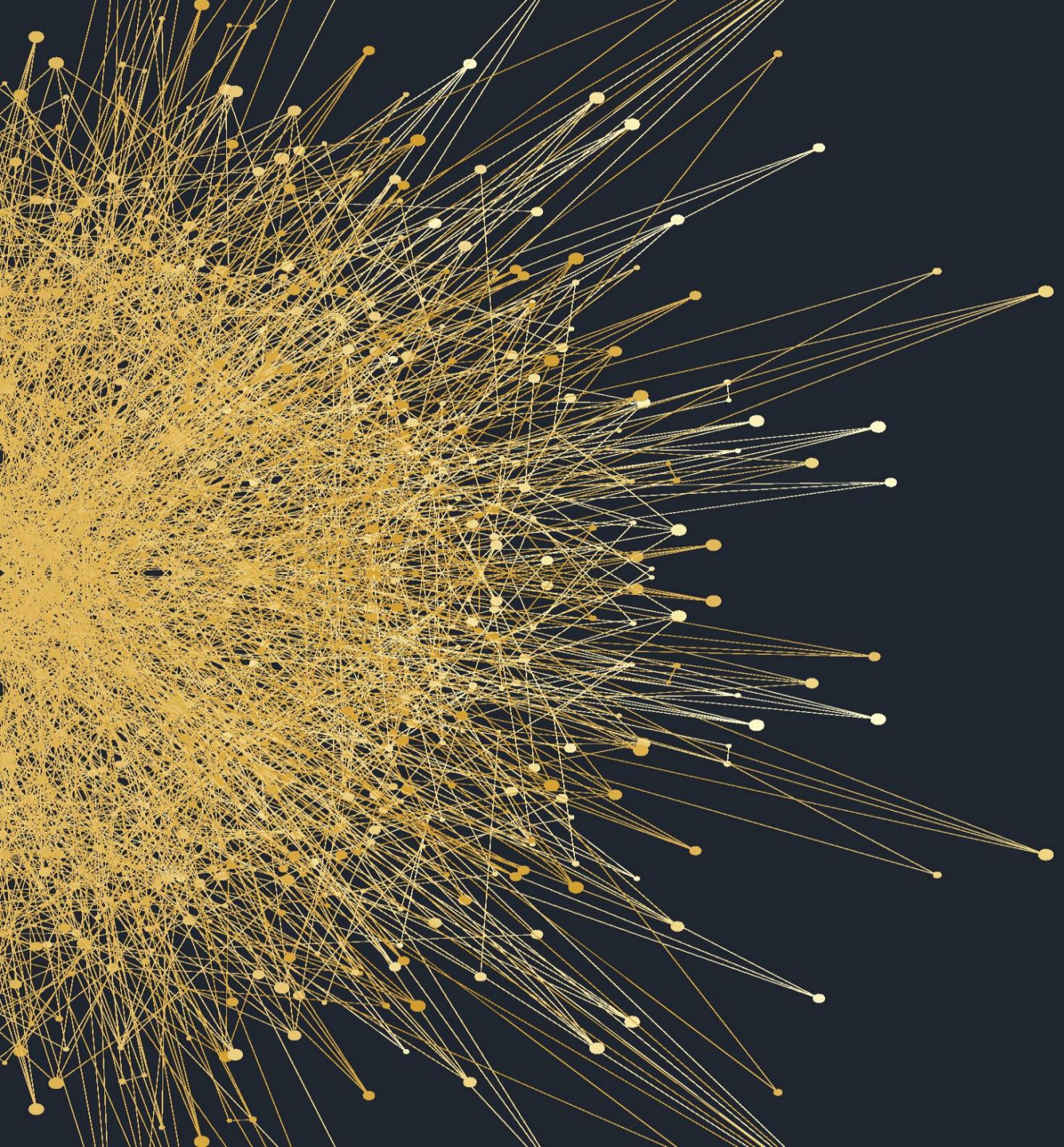
ML/DL Conferences/Journals

- **CVPR** - Conference on Computer Vision and Pattern Recognition
- **NIPS** - Neural Information processing systems
- **ICML** - International conference on machine learning
- **ECCV** - European Conference on Computer Vision
- **ICCV** - International Conference on Computer Vision
- **ECML** - European Conference on Machine Learning
- **IJCAI** - International Joint Conference on Artificial Intelligence
- **JML** - Journal of Machine Learning
- IEEE Transactions on Neural Networks
- IEEE Transactions on Pattern Analysis and Machine Intelligence
- Neural Networks
- Artificial Intelligence
- Pattern Recognition
- Expert Systems with Applications
- IEEE Transactions on Knowledge and Data Engineering
- Neural Computing and Applications
- IEEE Transactions on Audio, Speech and Language Processing
- International Journal of Neural Systems

Reading Materials

- **Text Book:** https://www.deeplearningbook.org/
- https://www.coursera.org/learn/machine-learning
- deeplearning.ai
- http://cs231n.stanford.edu/
- https://github.com/floodsung/Deep-Learning-Papers-Reading-Roadmap
- https://github.com/terryum/awesome-deep-learning-papers


“You can’t cross the sea merely
by standing and staring
at the water”



Thank you!

Any Question..?