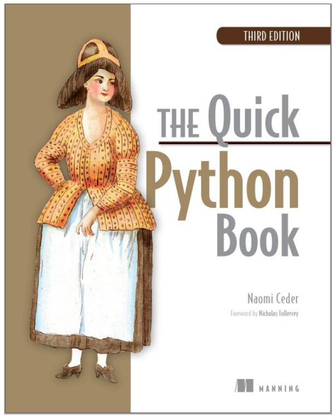
Introduction to PyTorch

25 September 2022

Department of Mathematics, Chiang Mai University

Why Python?

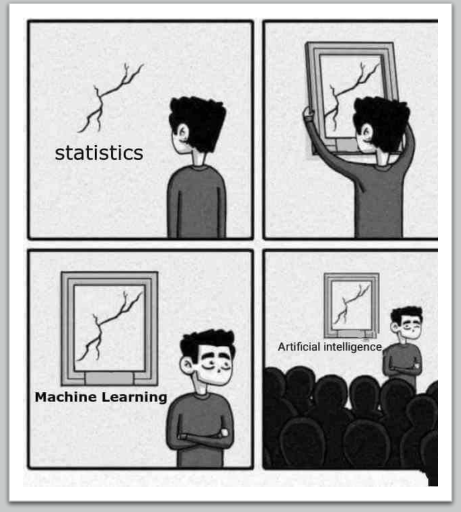


- Python is a modern programming language developed by Guido van Rossum in the 1990s
- *What Python does well*
  - *Python is easy to use*
  - *Python is expressive*
  - *Python is readable*
  - *Python is complete—"batteries included"*
  - *Python is cross-platform*
  - *Python is free*
- *What Python doesn't do as well*
  - *Python isn't the fastest language*
  - *Python doesn't have the most libraries*
  - *Python doesn't check variable types at compile time*
  - *Python doesn't have much mobile support*
  - *Python doesn't use multiple processors well*

Notepad++ : https://notepad-plus-plus.org/downloads/

Visual Studio Code : https://code.visualstudio.com/

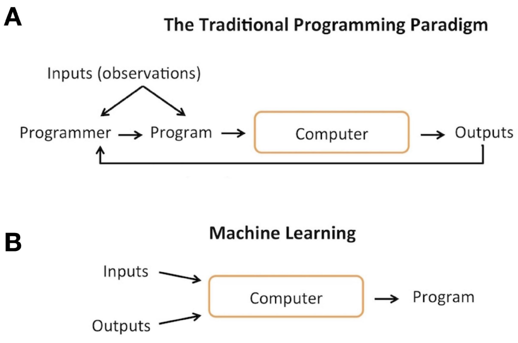
PC
ANACONDA
CO
SPYDER
jupyter



What is Machine Learning?

- Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.
- https://en.wikipedia.org/wiki/Machine_learning

Comparison between traditional programming (A) and machine learning (B)



A

The Traditional Programming Paradigm

Inputs (observations)

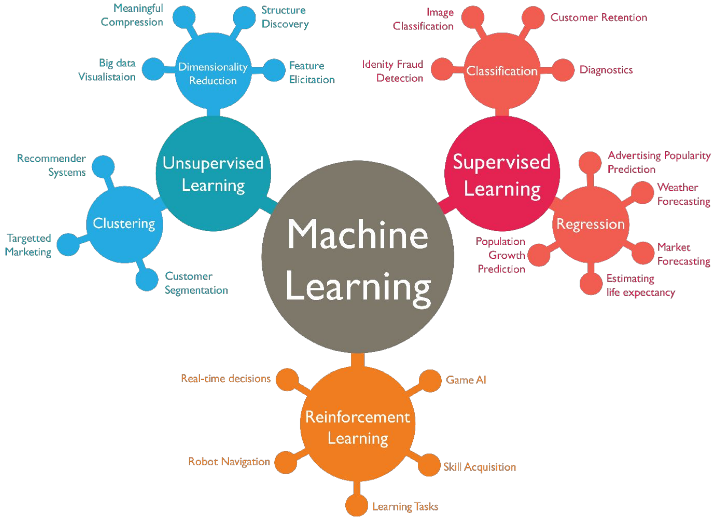
Programmer → Program → Computer → Outputs

B

Machine Learning

Inputs → Computer → Program

Outputs → Computer



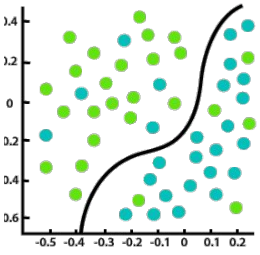
Machine Learning

Unsupervised Learning
- Dimensionality Reduction: Meaningful Compression; Structure Discovery; Big data Visualistaion; Feature Elicitation
- Clustering: Recommender Systems; Targetted Marketing; Customer Segmentation

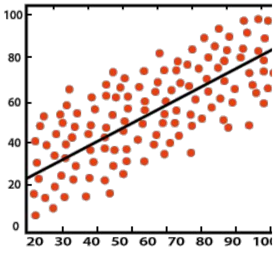
Supervised Learning
- Classification: Image Classification; Customer Retention; Idenity Fraud Detection; Diagnostics
- Regression: Advertising Popularity Prediction; Weather Forecasting; Population Growth Prediction; Market Forecasting; Estimating life expectancy

Reinforcement Learning
- Real-time decisions; Game AI; Robot Navigation; Skill Acquisition; Learning Tasks

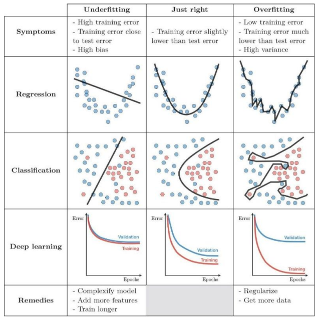
Regression vs Classification



Classification



Regression



	Underfitting	Just right	Overfitting
Symptoms	- High training error - Training error close to test error - High bias	- Training error slightly lower than test error	- Low training error - Training error much lower than test error - High variance
Regression			
Classification			
Deep learning			
Remedies	- Complexify model - Add more features - Train longer		- Regularize - Get more data

Introduction to Pytorch

Neural Networks in brief

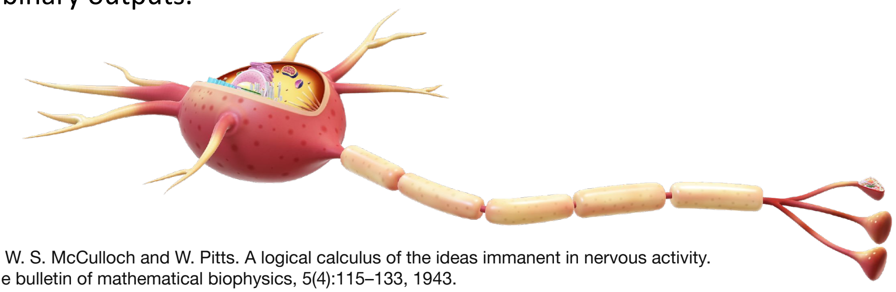
PyTorch package : Deep Learning Tools

Pytorch Linear Regression

scikit learn PyTorch TensorFlow Keras

1943 – The first mathematical model of a neural network

ANNs began with Warren McCulloch and Walter Pitts [1] who drew an analogy between biological neurons and simple logic gates with binary outputs.

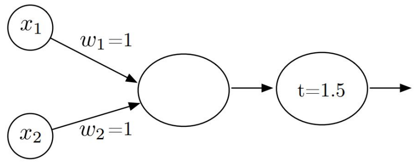


[1] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4):115–133, 1943.

1943 – The first mathematical model of a neural network

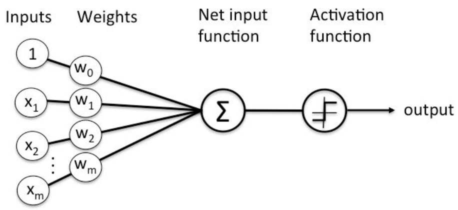
ANNs began with Warren McCulloch and Walter Pitts [1] who drew an analogy between biological neurons and simple logic gates with binary outputs.

x_1	x_2	Out
0	0	0
0	1	0
1	0	0
1	1	1

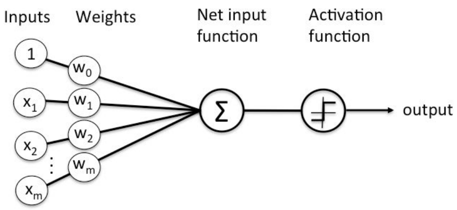


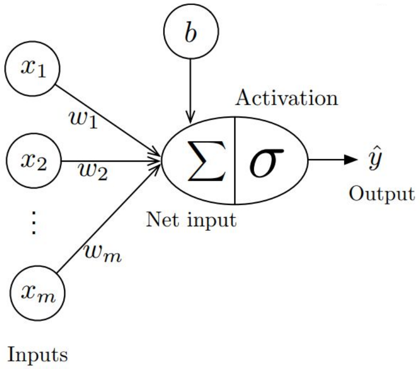
[1] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4):115–133, 1943.

Neural Networks Mathematical Model



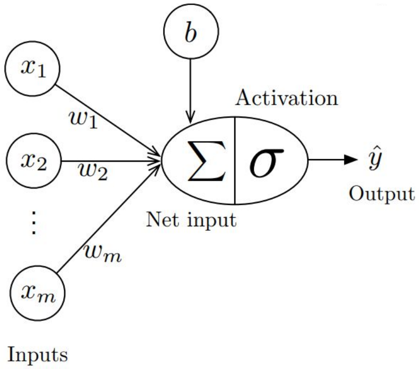
Neural Networks Mathematical Model



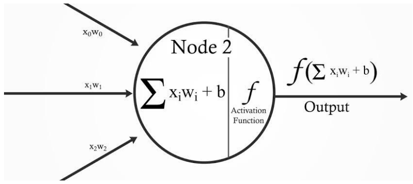


Linear Algebra inside NNs model

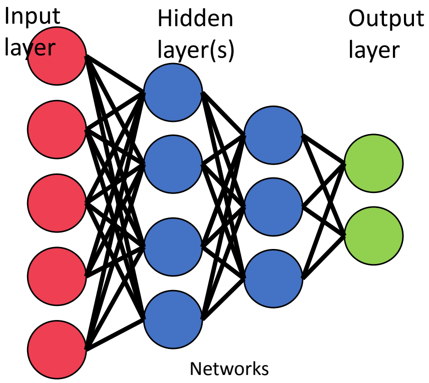
- Output $\hat{y} \coloneqq \sigma(z)$ where $z = x^T\boldsymbol{w} + b, \boldsymbol{w} \in \mathbb{R}^{m\times 1}$
- For 1 example : $x \in \mathbb{R}^{m\times 1}$,
  - $z = x^T\boldsymbol{w} + b$
  - $\hat{y} = \sigma(z)$
- For n examples : $X \in \mathbb{R}^{n\times m}$
  - $X\boldsymbol{w} + b = \begin{bmatrix} (x^{[1]})^T\boldsymbol{w} + b \\ \vdots \\ (x^{[n]})^T\boldsymbol{w} + b \end{bmatrix} = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} = \boldsymbol{z} \in \mathbb{R}^{n\times 1}$
  - $\hat{y} = \begin{bmatrix} \sigma(z_1) \\ \vdots \\ \sigma(z_n) \end{bmatrix} = \sigma(\boldsymbol{z})$



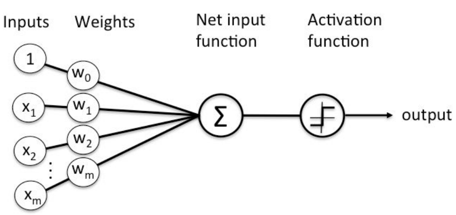
Artificial Neural Networks



Neural

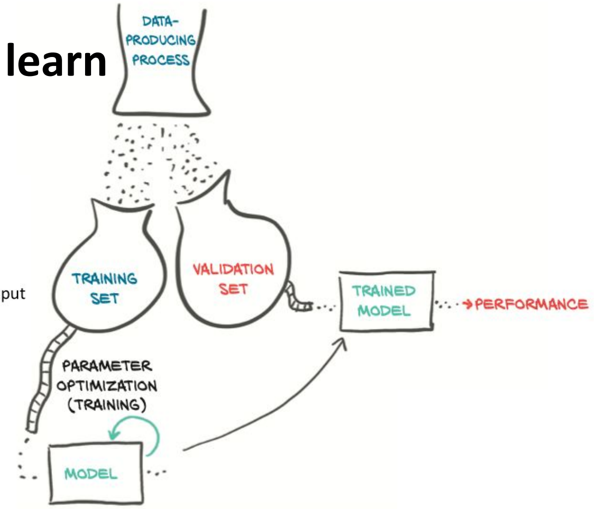


How the networks learn

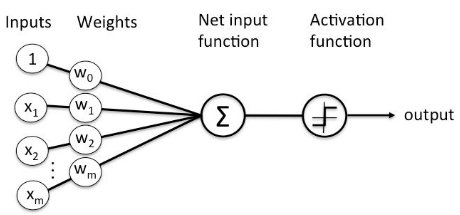


Inputs Weights Net input function Activation function

1 w_0 x_1 w_1 x_2 w_2 x_m w_m Σ output

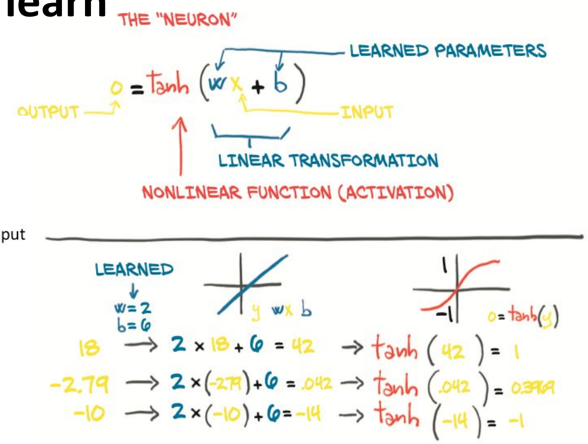


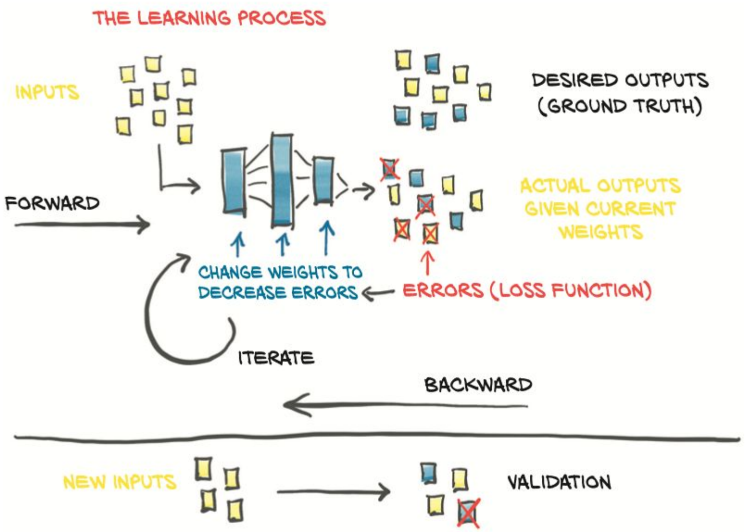
How the networks learn

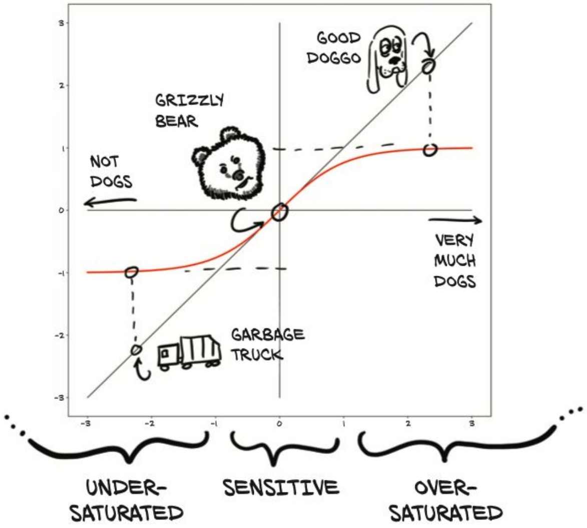


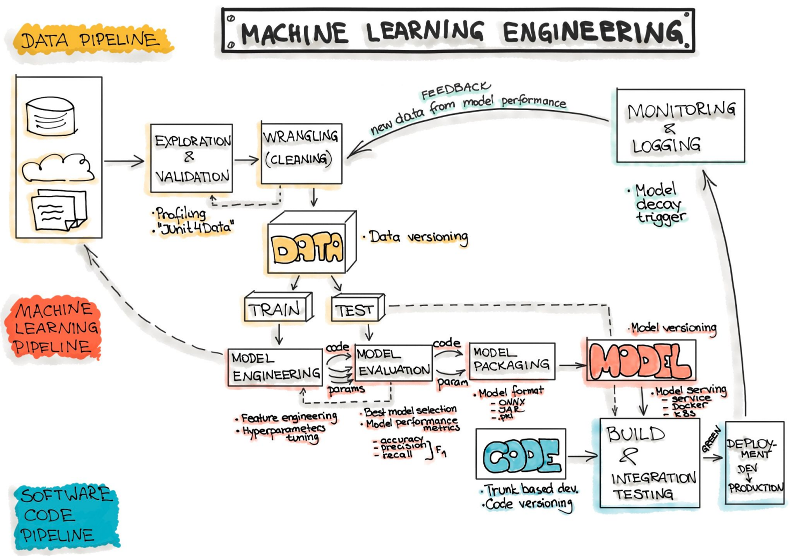
Inputs Weights Net input function Activation function

1 w_0 x_1 w_1 x_2 w_2 x_m w_m Σ output


THE "NEURON"
LEARNED PARAMETERS
o = tanh (w x + b)
OUTPUT
INPUT
LINEAR TRANSFORMATION
NONLINEAR FUNCTION (ACTIVATION)
LEARNED
w = 2
b = 6
y wx b
1
-1
o = tanh(y)
18 → 2 × 18 + 6 = 42 → tanh (42) = 1
-2.79 → 2 × (-2.79) + 6 = .042 → tanh (.042) = 0.3969
-10 → 2 × (-10) + 6 = -14 → tanh (-14) = -1


THE LEARNING PROCESS
INPUTS
DESIRED OUTPUTS (GROUND TRUTH)
FORWARD
ACTUAL OUTPUTS GIVEN CURRENT WEIGHTS
CHANGE WEIGHTS TO DECREASE ERRORS
ERRORS (LOSS FUNCTION)
ITERATE
BACKWARD
NEW INPUTS
VALIDATION

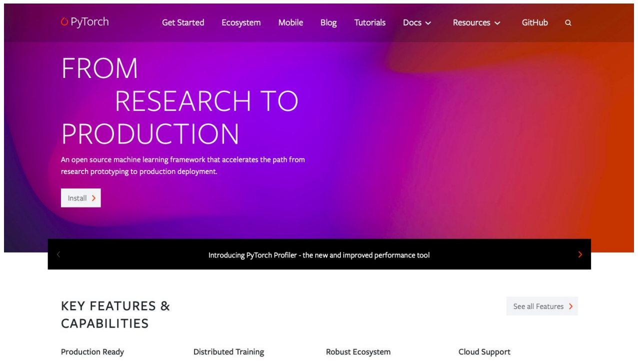




Introduction to PyTorch

PyTorch

- torch.tensor
- torch.autograd
- torch.nn
- torch.optim



PyTorch

FROM
RESEARCH TO
PRODUCTION

An open source machine learning framework that accelerates the path from research prototyping to production deployment.

Introducing PyTorch Profiler - the new and improved performance tool

KEY FEATURES & CAPABILITIES

Production Ready

Distributed Training

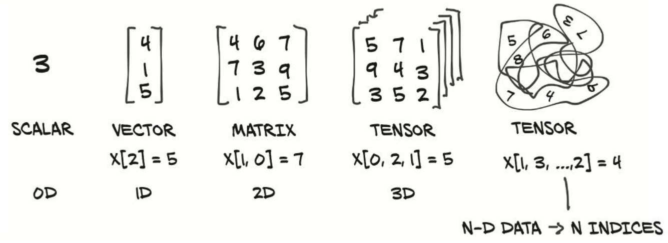
Robust Ecosystem

Cloud Support

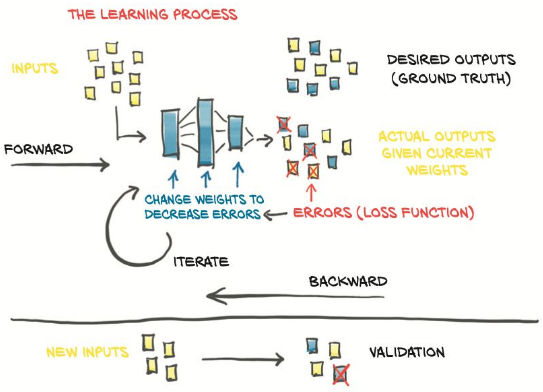
https://pytorch.org

torch.tensor

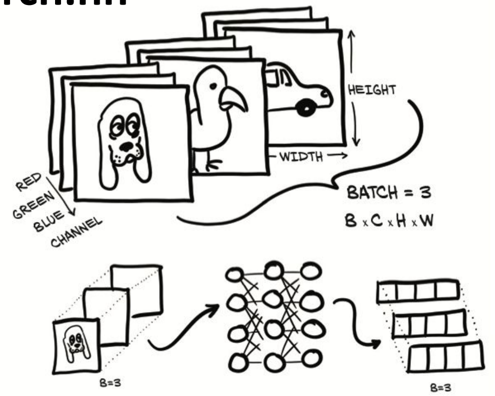
- A torch.tensor is a multi-dimensional matrix containing elements of a single data type.
- Tensors are similar to NumPy's ndattarys, with the addition being that Tensors can also be used on a GPU to accelerate computing.

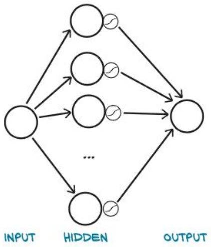


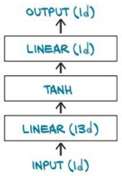
torch.autograd


THE LEARNING PROCESS
INPUTS
DESIRED OUTPUTS (GROUND TRUTH)
FORWARD
ACTUAL OUTPUTS GIVEN CURRENT WEIGHTS
CHANGE WEIGHTS TO DECREASE ERRORS
ERRORS (LOSS FUNCTION)
ITERATE
BACKWARD
NEW INPUTS
VALIDATION

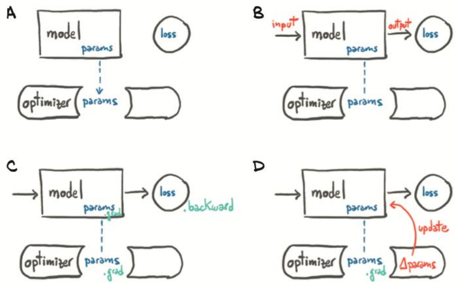
torch.nn

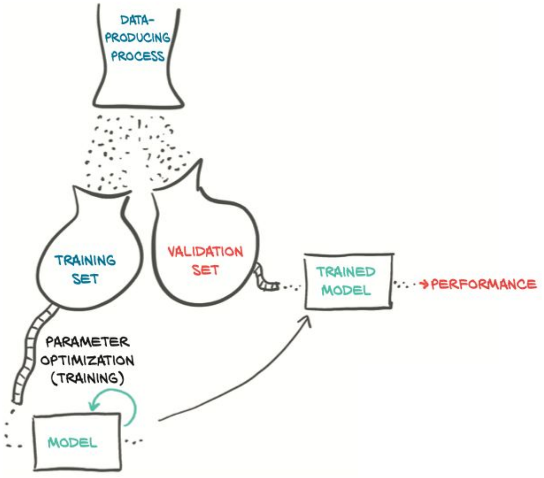






torch.optim

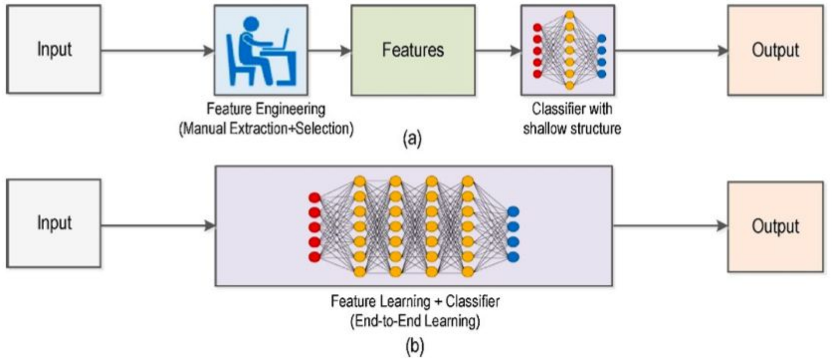




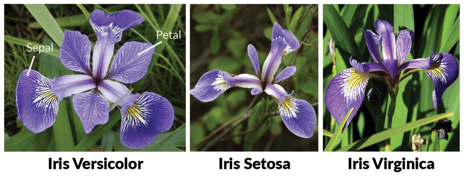
[Review] Gradient Descent

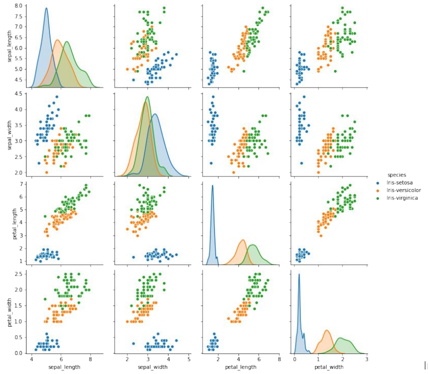
Colab:
https://colab.research.google.com/drive/1Qpbsy_kQlXMyGnaT_ApO-_AvQZdGYpEE?usp=sharing

Traditional computer vision workflow vs Deep learning workflow

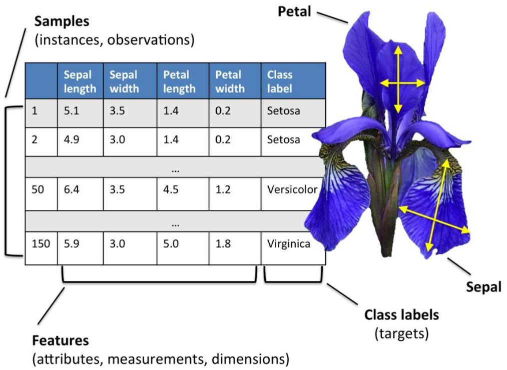


Paper source : https://arxiv.org/pdf/1910.13796.pdf

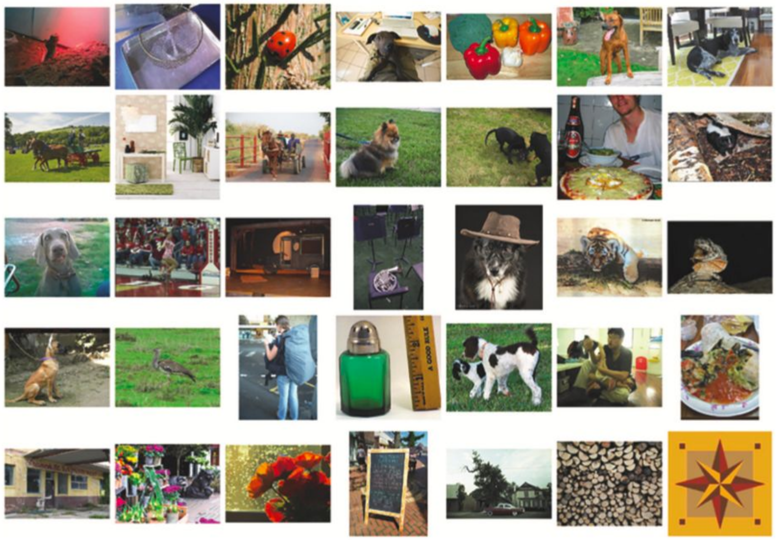




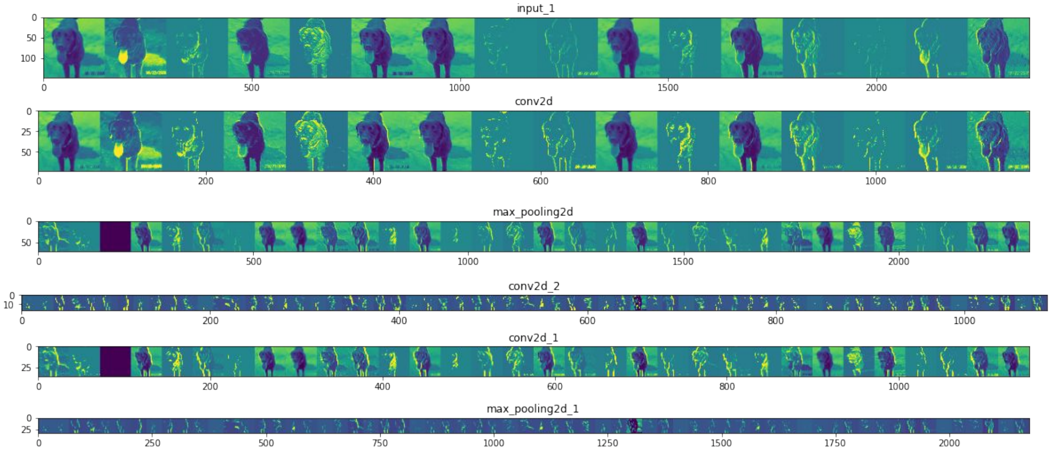
Iris dataset


Petal
Samples
(instances, observations)
Sepal
Class labels
(targets)
Features
(attributes, measurements, dimensions)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica



Machines try to see a dog as...


input_1
conv2d
max_pooling2d
conv2d_2
conv2d_1
max_pooling2d_1