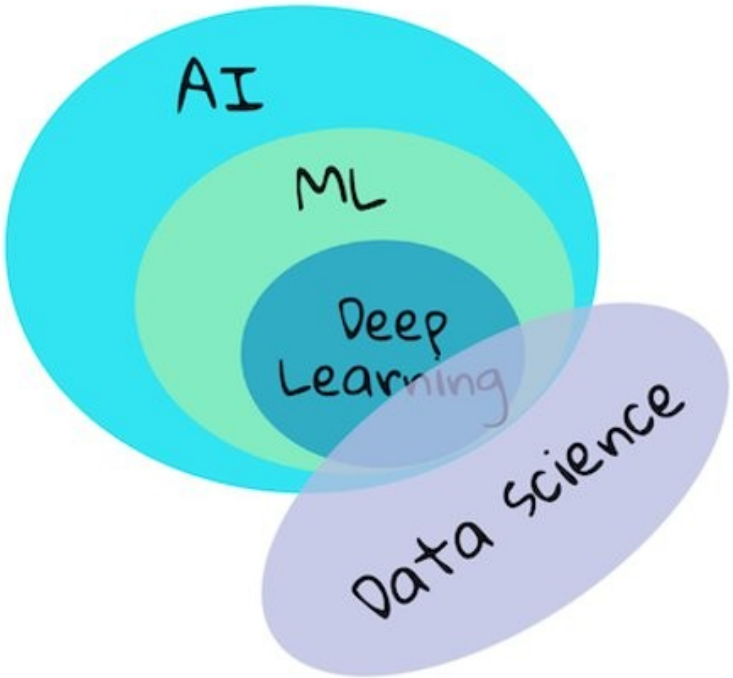
Deep Learning

NASTARAN SHAHPARIAN |SHARCNET | COMPUTE ONTARIO | COMPUTE CANADA |YORK UNIVERSITY

Difference between ML, DL, AI, and Data science

- Artificial Intelligence
  - Any techniques that enables computers to mimic human behaviour
- Machine Learning
  - Ability to learn without explicitly being perf
- Deep learning
  - Extract pattern from data using neural netv



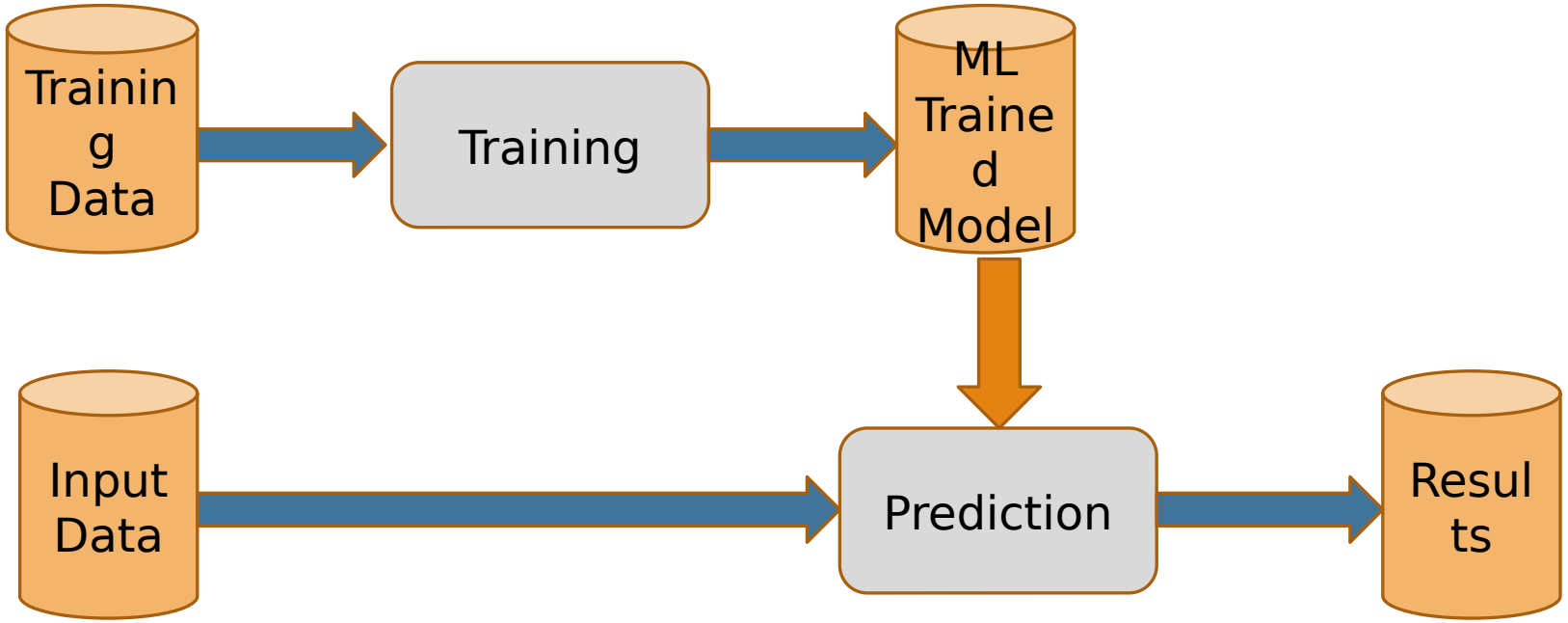
Machine Learning in simple words

- Training a machine learning algorithm on a set of data, allowing it to identify patterns and make predictions or decisions based on that data.
- A type of artificial intelligence that enables machines to learn from experience without being explicitly programmed.
- Has many practical applications, including image and speech recognition, natural language processing, fraud detection, and recommendation systems.

Data is in different forms

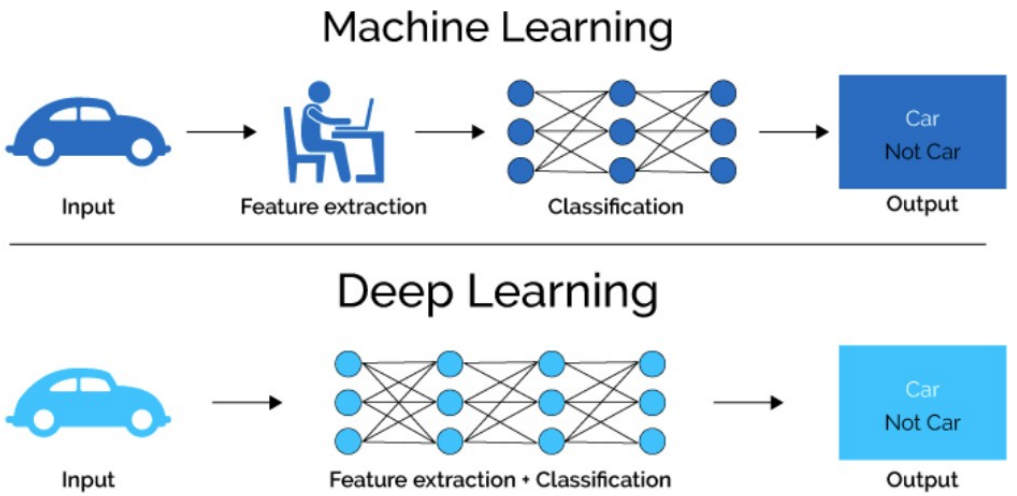
- Numerical data (Marketing Analytics)
- Image data (Face recognition)
- Video data (Object recognition)
- Sound data (Music generation)
- Text data (Sentiment analysis)

ML Workflow


Trainin
g
Data
Training
ML
Traine
d
Model
Input
Data
Prediction
Resul
ts

What is Deep Learning?

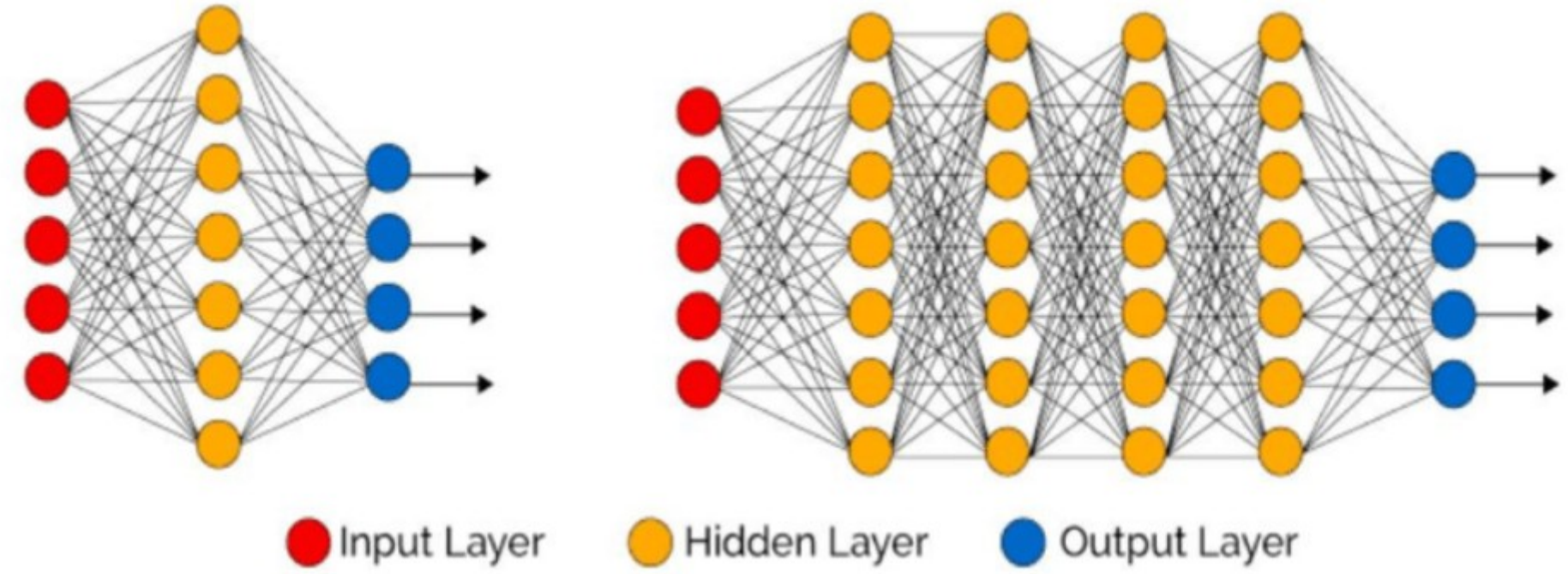
- Deep Learning (DL) is a subset of machine learning
- Multiple layers of nonlinear processing units are used for feature extraction and transformation.
- A computational approach that involves the use of multiple layers of artificial neural networks to model and solve complex problems.



https://medium.com/swlh/ill-tell-you-why-deep-learning-is-so-popular-and-in-demand-5aca72628780

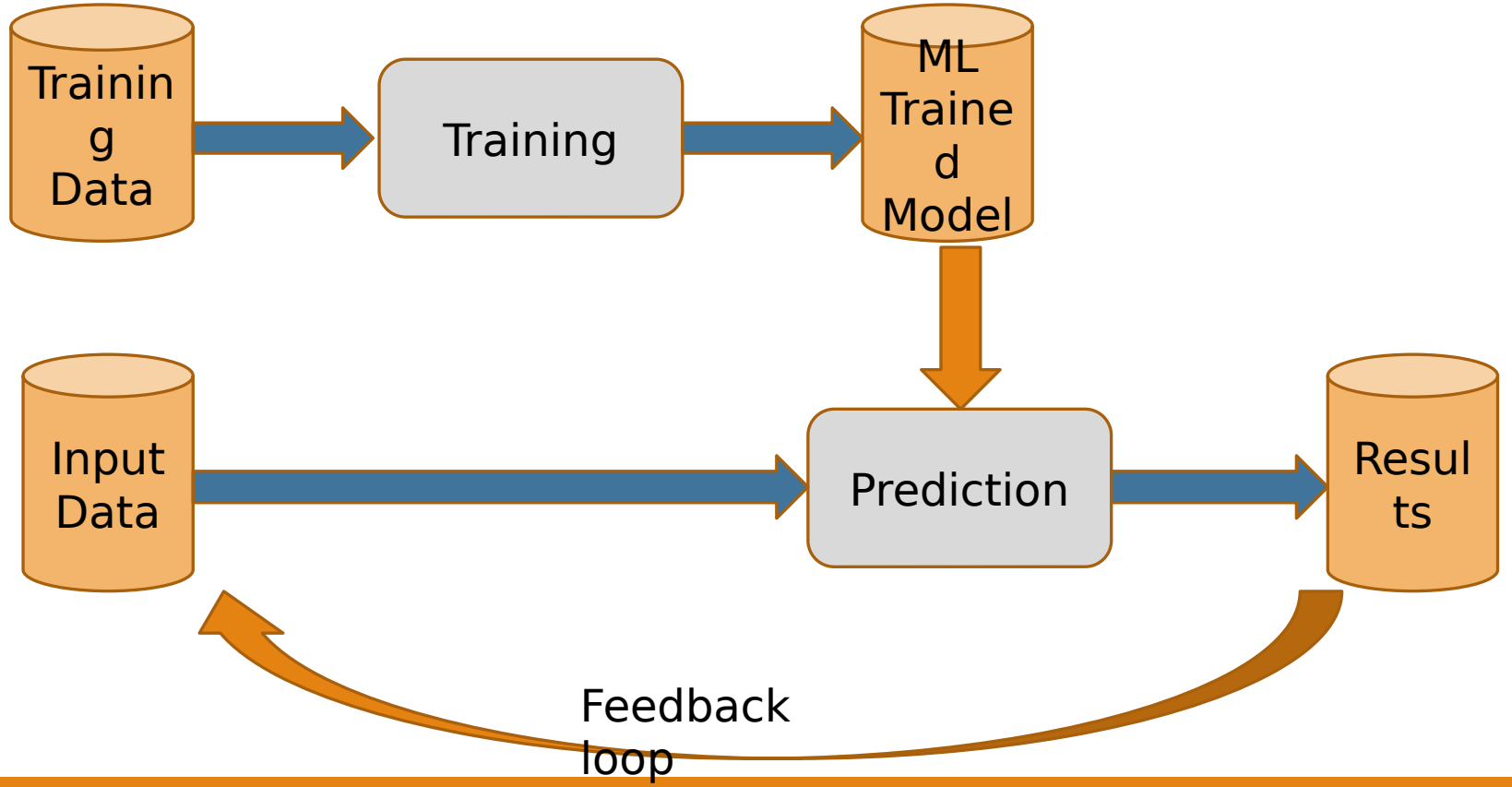
A shallow neural network with single hidden layer

A deep neural network with more than one hidden layer



https://sourcemeridian.com/machine-learning-as-a-tool-to-grow-sales-in-telemarketing-operations/

What is AI?

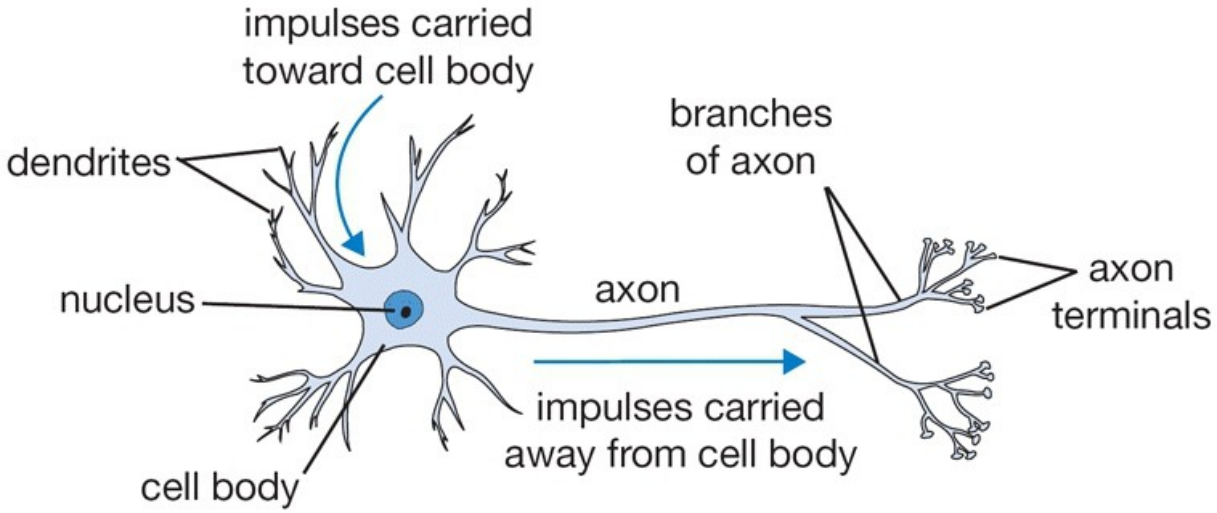

Trainin
g
Data
Training
ML
Traine
d
Model
Input
Data
Prediction
Resul
ts
Feedback
loop

What is Data Science

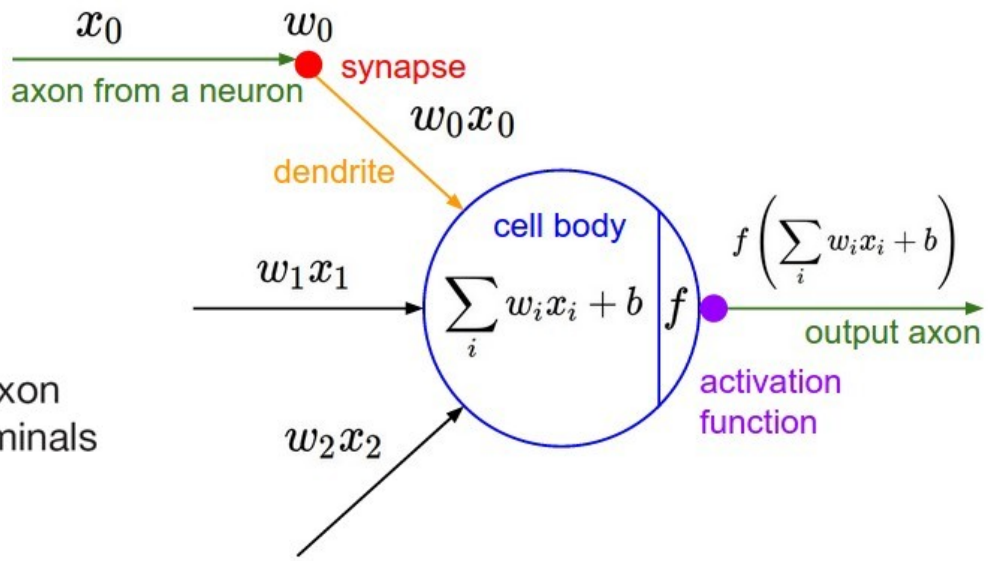
- Data Driven Decision making
- Making sense out of data
- Uncovering the hidden insights and patterns in data
- Using machine learning models, data visualizations and intelligent reports

Neural Network

http://cs231n.github.io/neural-networks-1/



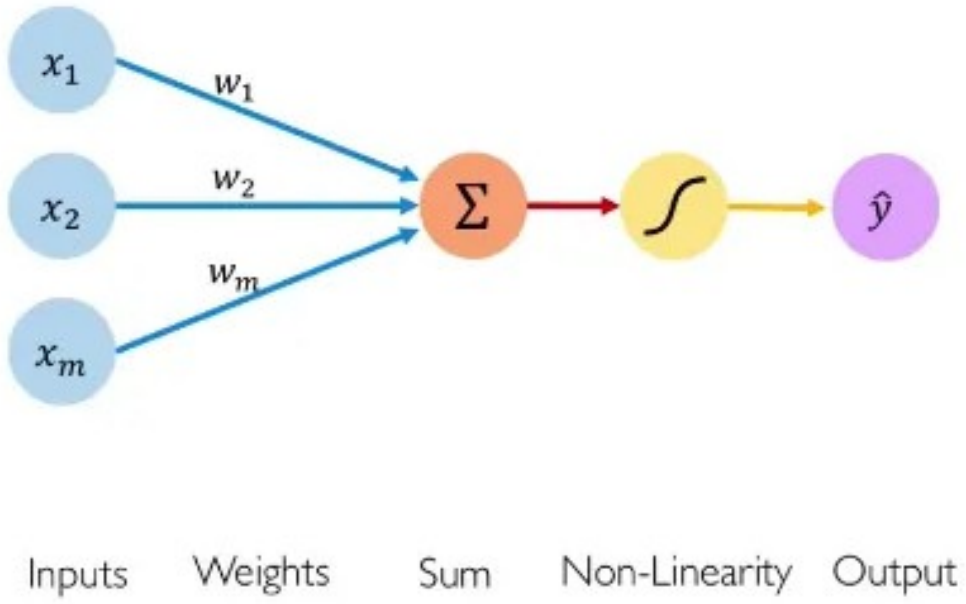
https://www.atrainceu.com/content/3-normal-brain-functions-and-normal-aging

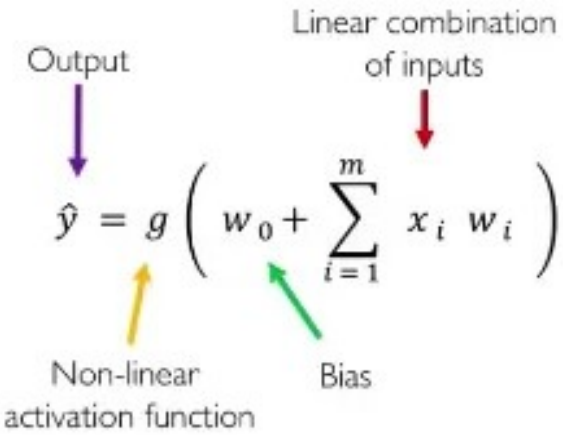


https://bouzouitina-hamdi.medium.com/activation-functions-in-neural-networks-1c1de2c866a

The Perceptron: Forward Propagation

https://medium.com/analytics-vidhya/neural-network-part1-inside-a-single-neuron-fee5e44f1e



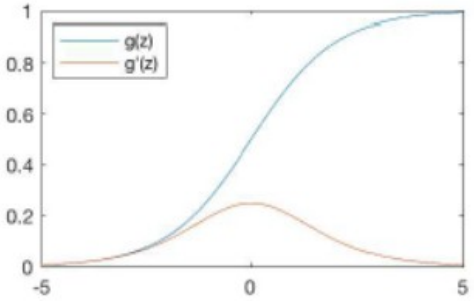


$$\hat{y} = g(w_0 + \boldsymbol{X}^T \boldsymbol{W})$$

where: $\boldsymbol{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$ and $\boldsymbol{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$

Common Activation Functions

Sigmoid Function

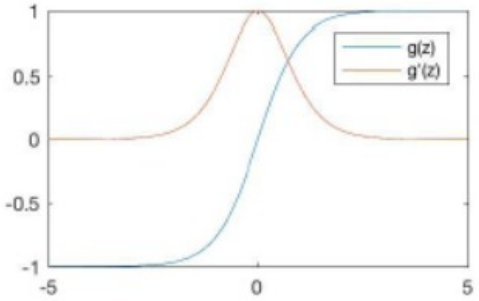


$$g(z) = \frac{1}{1 + e^{-z}}$$

$$g'(z) = g(z)(1 - g(z))$$

tf.nn.sigmoid(z)

Hyperbolic Tangent

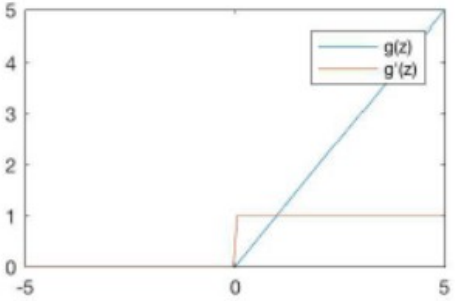


$$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

$$g'(z) = 1 - g(z)^2$$

tf.nn.tanh(z)

Rectified Linear Unit (ReLU)

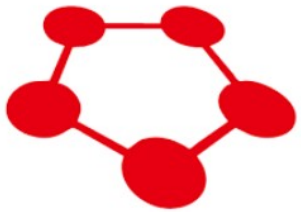


$$g(z) = \max(0, z)$$

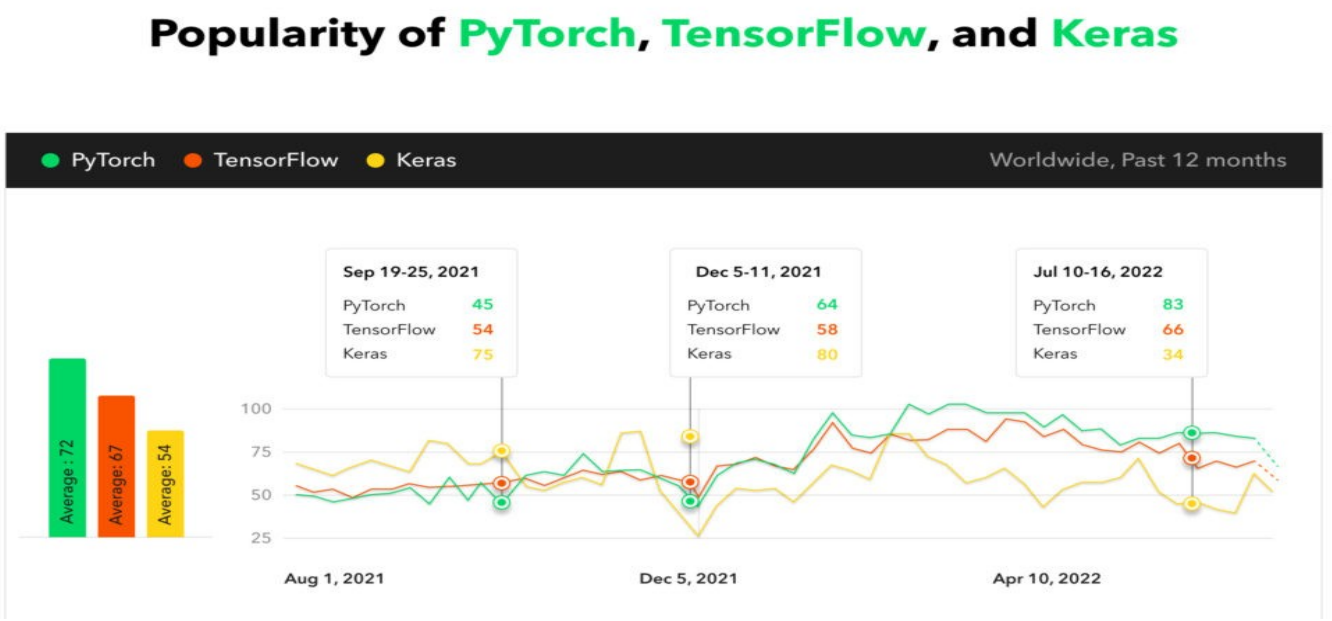
$$g'(z) = \begin{cases} 1, & z > 0 \\ 0, & \text{otherwise} \end{cases}$$

tf.nn.relu(z)

DL Frameworks



DL Frameworks trends



https://www.rapidops.com/blog/tensorflow-pytorch-keras/

DL Frameworks Comparison

TensorFlow vs PyTorch vs Keras

Features	TensorFlow	PyTorch	Keras
Written In	C++, CUDA, Python	Lua	Python
Architecture	Not easy to use	Complex, less readable	Simple, concise, readable
API Level	High and Low	Low	High
Datasets	Large datasets, high-performance	Large datasets, high-performance	Smaller datasets
Debugging	Difficult to conduct debugging	Good debugging capabilities	Simple network, so debugging is not often needed
Does It Have Trained Models?	Yes	Yes	Yes
Popularity	Second most popular	Third most popular	Most Popular
Speed	Fast, high-performance	Fast, high-performance	Slow, low performance

https://www.rapidops.com/blog/tensorflow-pytorch-keras/

ML problems

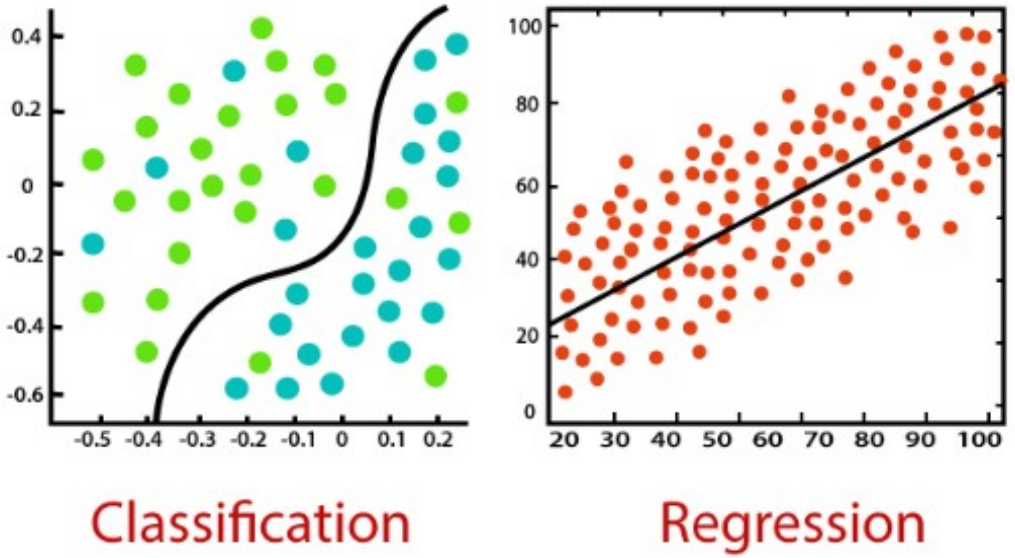
Types of Machine learning

	Supervised	Unsupervised	Reinforcement
Discrete	Classification	Clustering	Rewarding/ punishing behaviour
Continuous	Regression	Dimensionality reduction	Rewarding/ punishing behaviour

Supervised learning

The algorithm is trained on a labeled datasets to predict unseen data

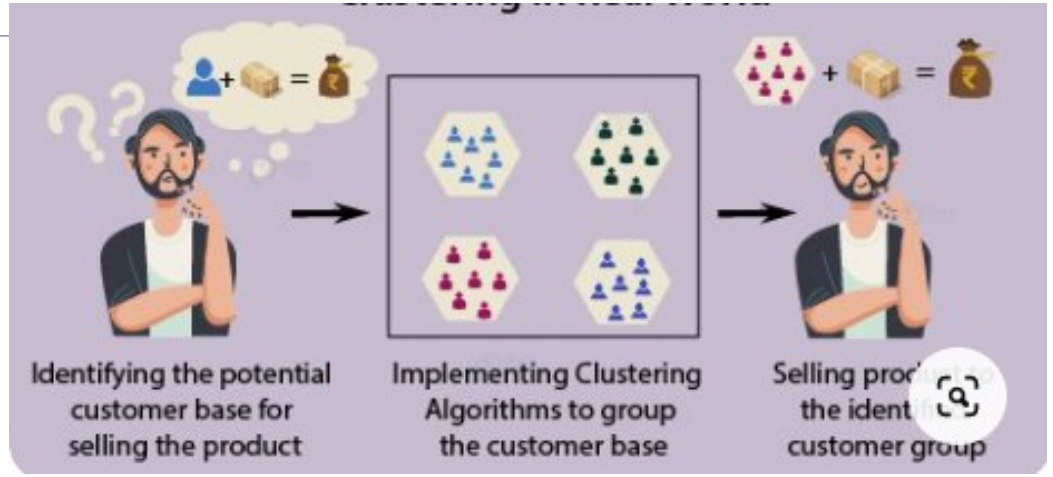
- Regression
  - Predict a continuous output variable. (The price of a house)
- Classification
  - The algorithm learns to predict a categorical output variable (classifying an email as spam or not spam)



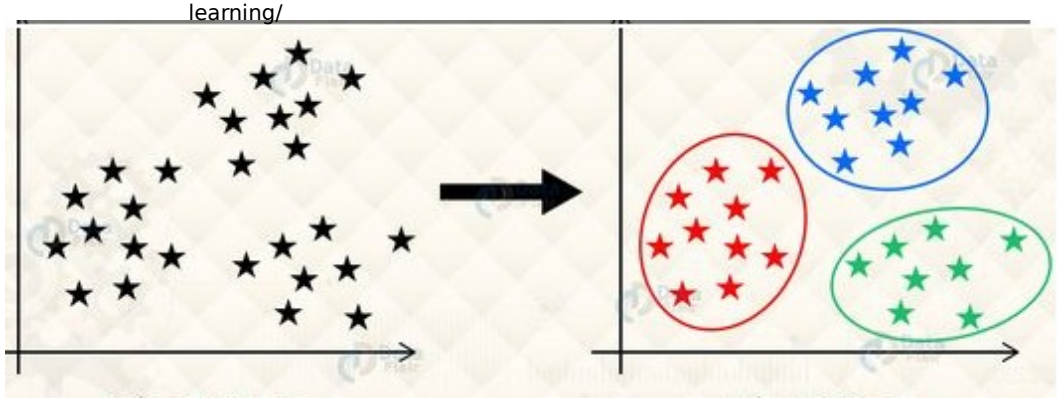
https://www.projectpro.io/article/classificati
regression-in-machine-learning/545

Unsupervised learning

- The algorithm is trained on unlabelled data
- Tasked with finding patterns on its own, without any feedback
  - Clustering
  - Dimensionality reduction
  - Anomaly detection



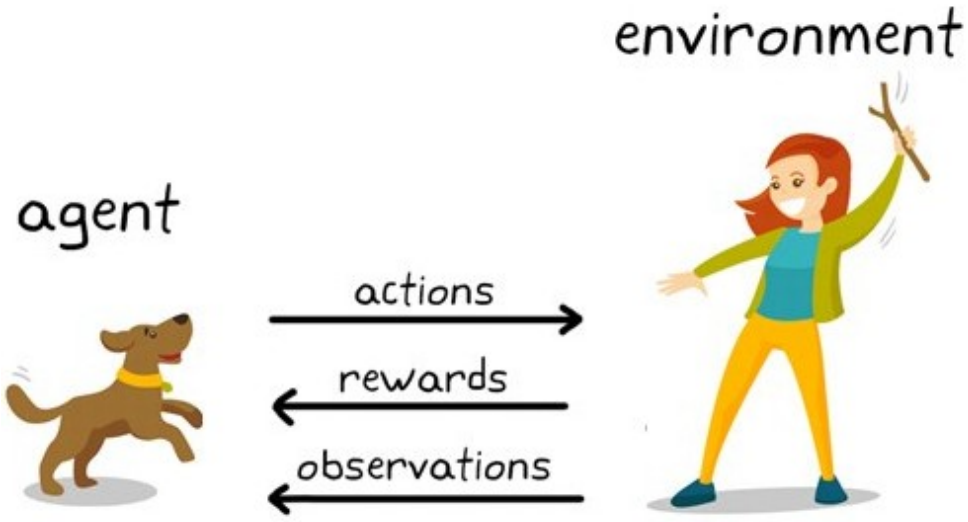
https://data-flair.training/blogs/clustering-in-machine-learning/



https://data-flair.training/blogs/scipy-clustering/

Reinforcement learning

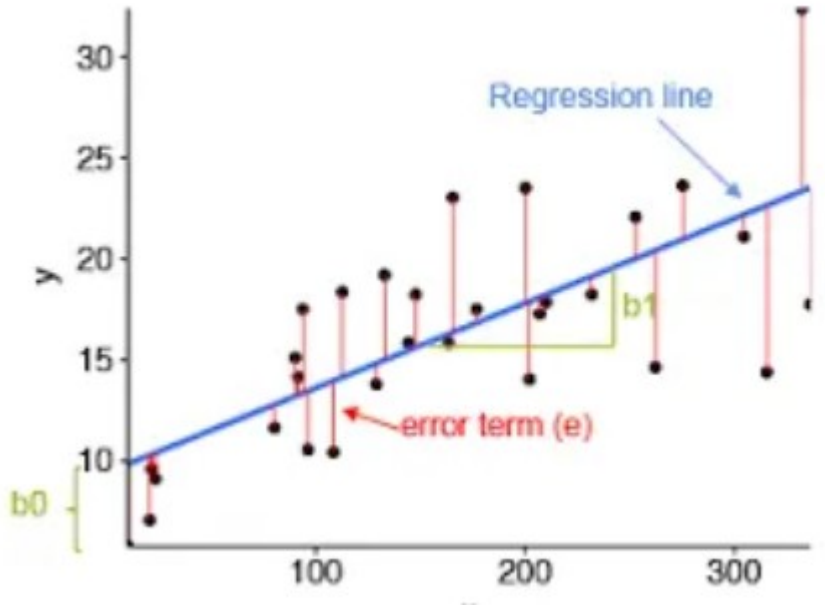
- Rewarding desired/ punishing undesired behaviours
- Able to perceive and interpret its environment, take actions and learn through trial and error



https://medium.com/analytics-vidhya/a-beginners-guide-to-reinforcement-learning-and-its-basic-implementation-from-scratch-2c0b5444cc49

Regression Problem

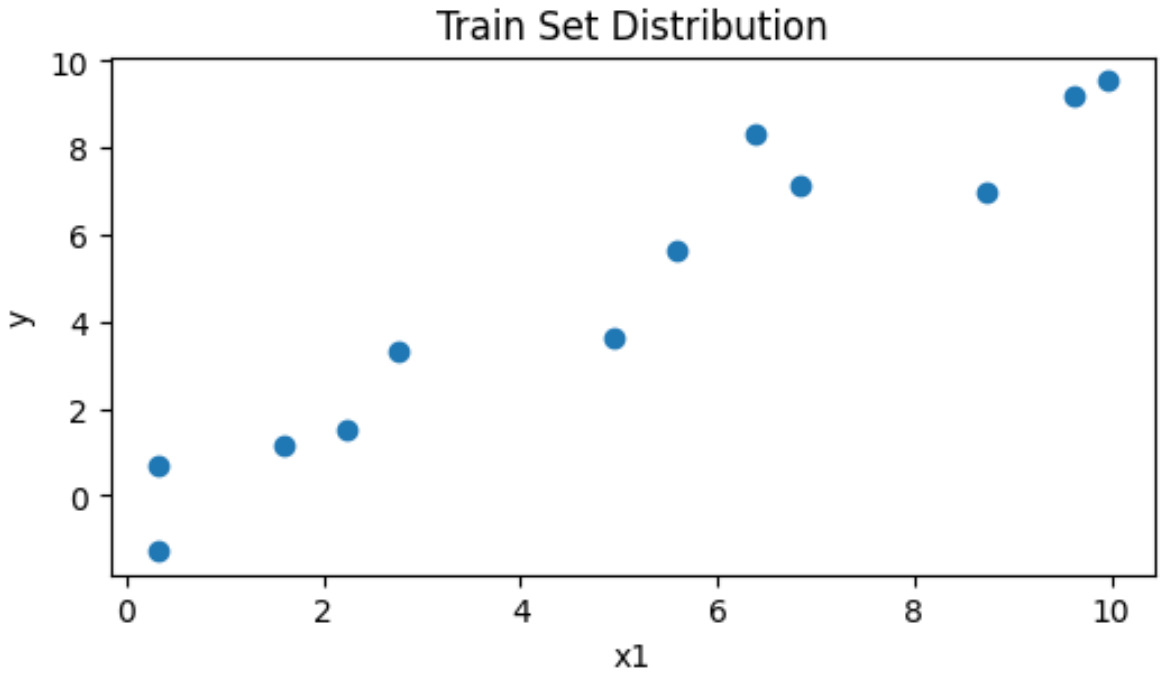
- Find the best-fitting mathematical function that describes the relationship between the variables.
- This line best fits the data and minimizes the sum of the squared differences between the actual values of y and the predicted values of y.



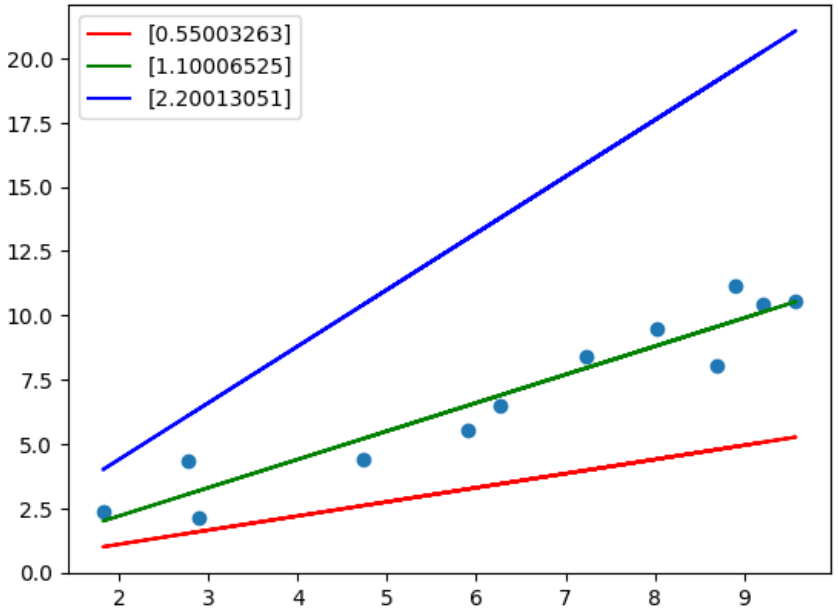
https://data-science-blog.com/blog/2022/05/02/understanding-linear-regression-with-all-statistical-terms/

Input (X,y)

X	y
9.62	5.94
8.74	3.33
2.76	11.5
6.38	8.76
9.95	2.32
5.58	8.25
1.59	7.69
0.32	8.18
4.94	5.64
6.83	8.16
2.23	4.83
0.32	5.41



Cost Function



$$H(x) = Wx+b$$

Which line fits better?

$$H(x) - y$$

Predicted True

Cost Function

Model

$$H(x) = Wx + b$$

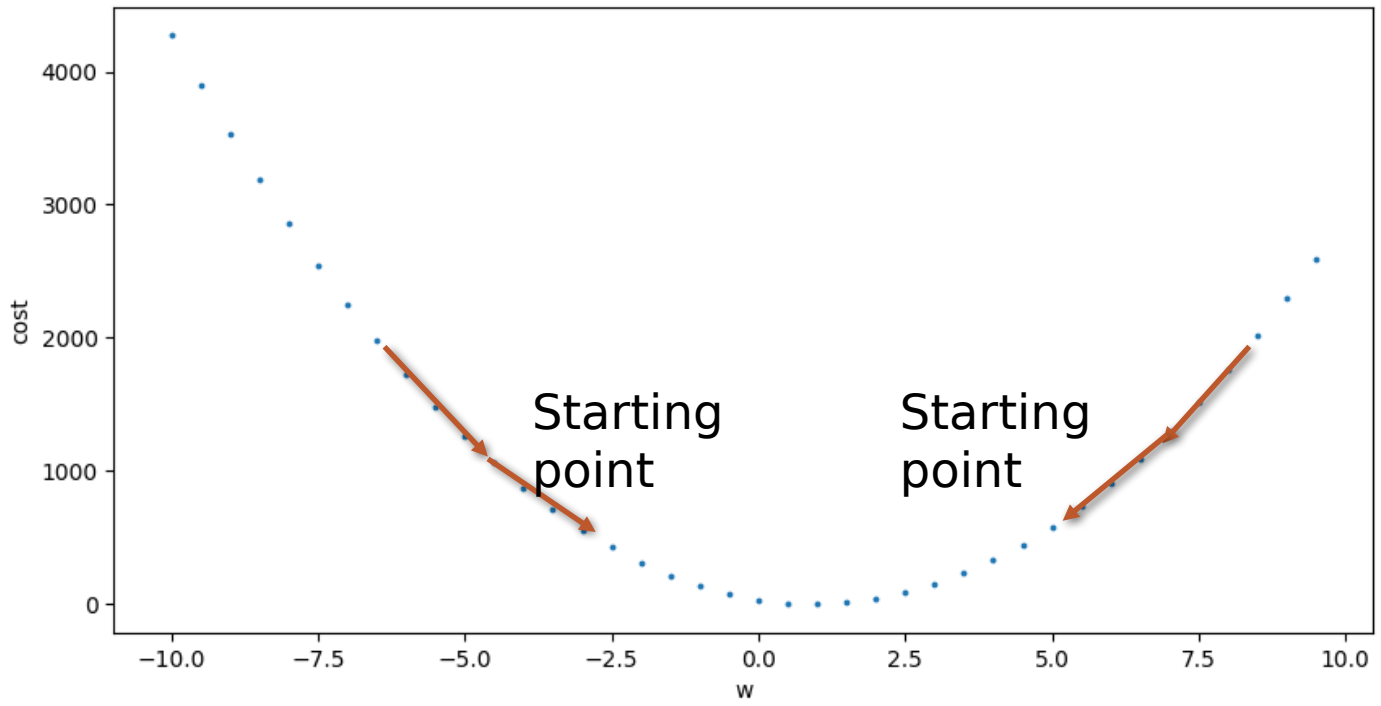
Mean Square Error

$$\text{Cost} = \frac{1}{m}\sum_{i=1}^{m}\left(H(x_i) - y_i\right)^2$$

M : number of data

$$\text{Cost}(W, b) = \frac{1}{m}\sum_{i=1}^{m}\left(\left(Wx_i + b\right) - y_i\right)^2$$ → We want to minimize this equation

Gradient descent algorithm



$$\cos t(W) = \frac{1}{2m}\sum_{i=1}^{m}\left((Wx_i - y_i)\right)^2$$

$$\frac{d}{dW}\cos t(W) = \frac{1}{m}\sum_{i=1}^{m}\left(Wx_i - y_i\right)x_i$$

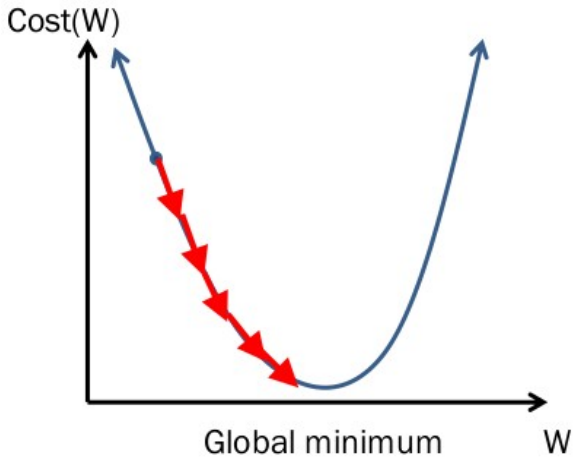
$$W := W - \alpha\frac{d}{dW}\cos t(W)$$

New rate

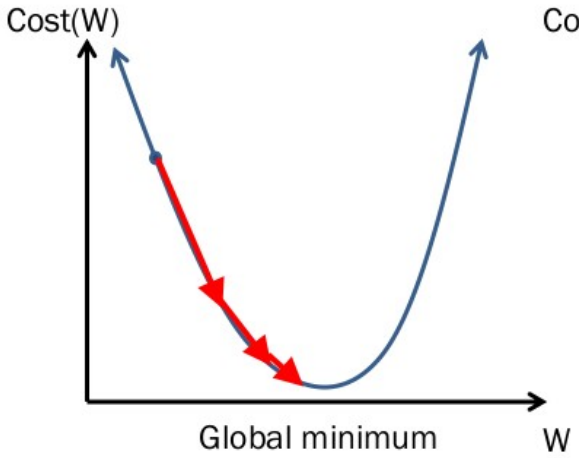
learning

Learning rate

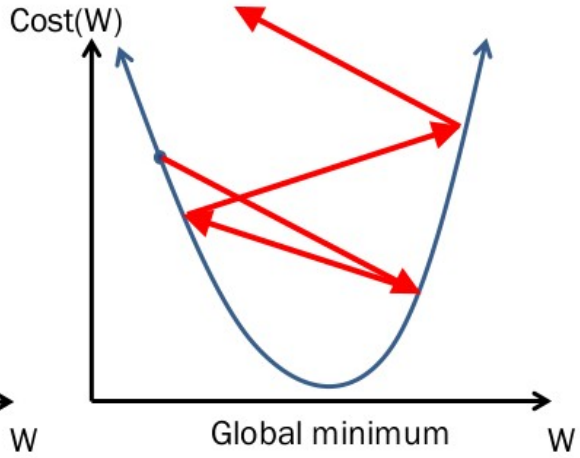
$$W := W - \alpha \frac{d}{dW} \cos t(W)$$



A small learning rate requires time to diverge



An optimal learning rate



Too large learning rate
cause divergence

https://medium.com/swlh/cyclical-learning-ra
the-ultimate-guide-for-setting-learning-rates-f
neural-networks-3104e906f0ae

Regression with different Learning rates