KA
SK
Katedra
Architektury
Systemów
Komputerowych

POLITECHNIKA
GDAŃSKA

VoiceLab

Introduction to Machine Learning
Hands on Artificial Neural Networks

Marcin Świniarski

Plan of the attack

- Introduction to **M**achine **L**earning
  - Revolution of AI
    - ❖ Where we are?
    - ❖ Where we are heading?
  - What Machine Learning is?
  - Well known techniques in Machine Learning:
    - ❖ K-Nearest Neighbors(KNN)
    - ❖ Linear Regression
    - ❖ Classification and Regression Trees (CART)
  - How ANN works?
- Hands-on Artificial Neural Networks!
  - Churn Modelling
  - First working Neural Network, which is applicable in business

These people do not exist

Generative Adversarial Networks + NLP

source: https://www.youtube.com/watch?v=HJcdVjkqiW8

Cat or dog?

source: https://www.telegraph.co.uk/news/2016/11/08/is-this-a-cat-or-a-dog-internet-falls-in-love-with-adorable-were/

How to classify this?

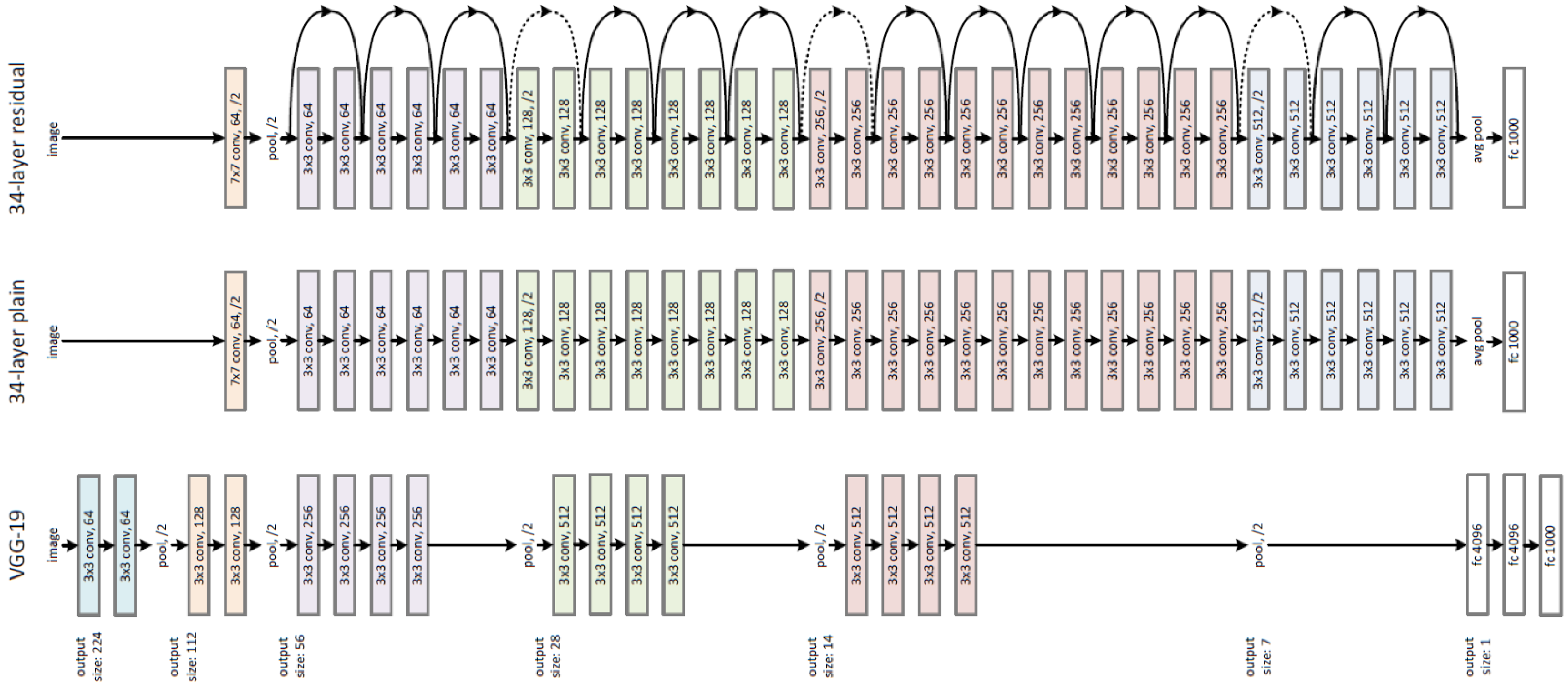
How about this?

$$\sqrt[31]{5193581391931913 01^{192219}} * \int_{-4243}^{1284} e^{-24x} x^{14} dx$$

Why now?

86 milions of neurons

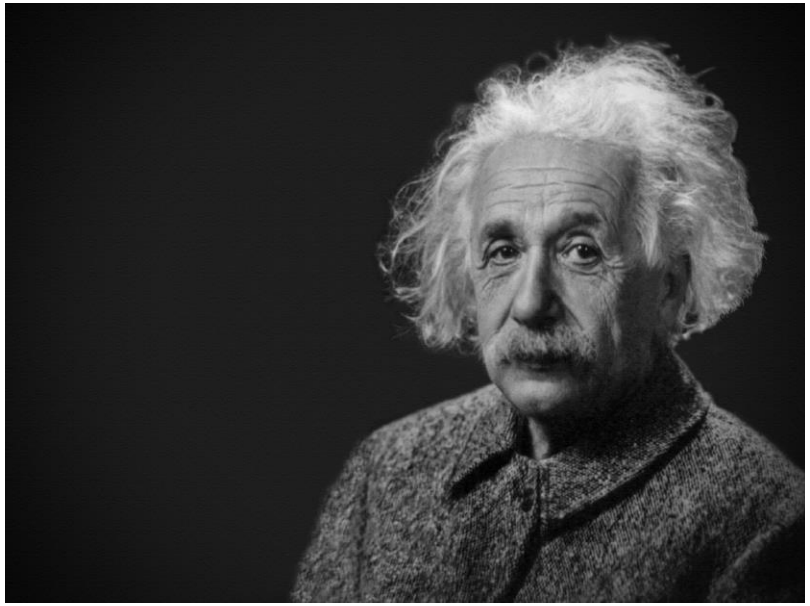
ResNet152 – 60 mln parameters, 152 layers



source: https://arxiv.org/abs/1512.03385

Frog vs Human

VS



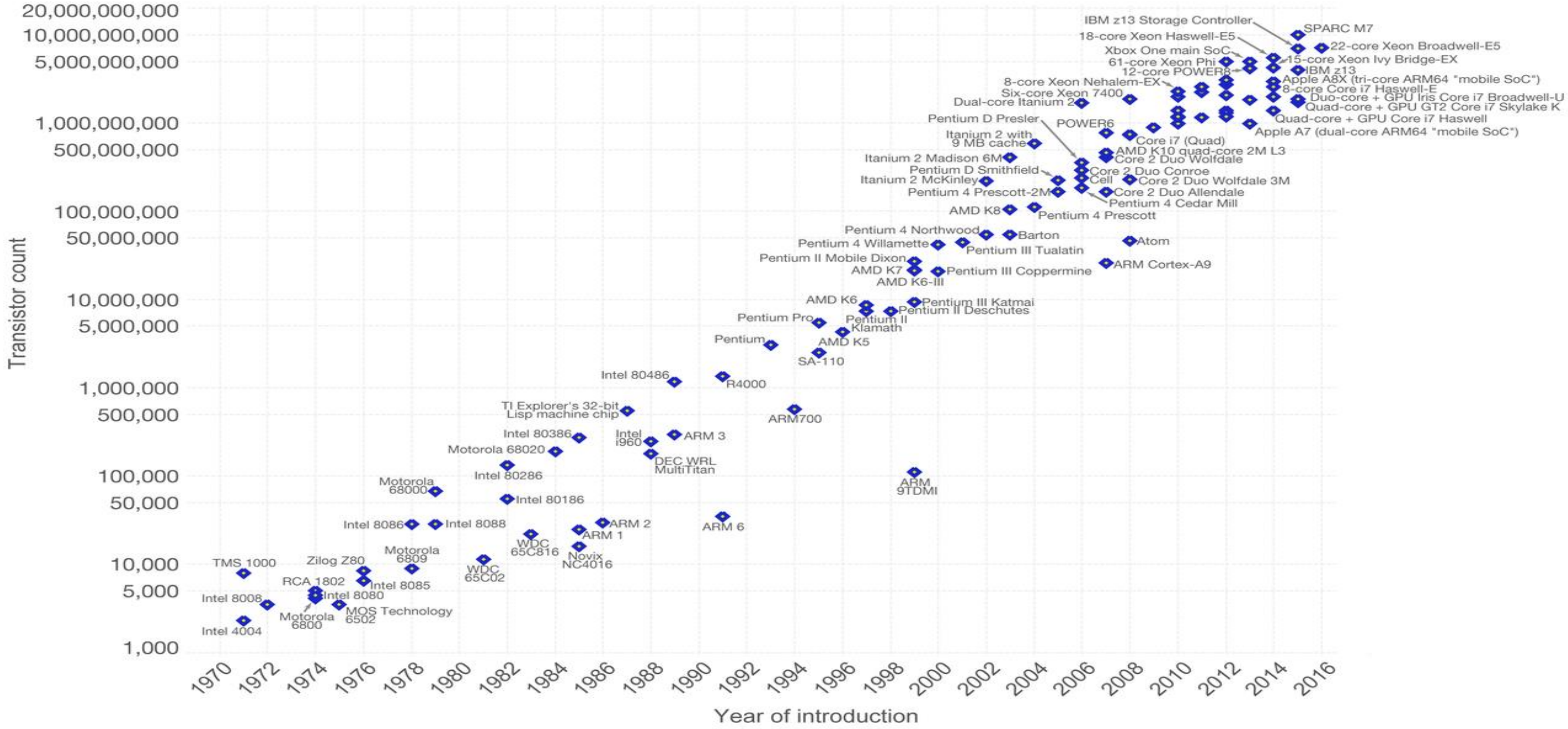
16 mln neurons

86 mln neurons

Moore’s Law – The number of transistors on integrated circuit chips (1971-2016)

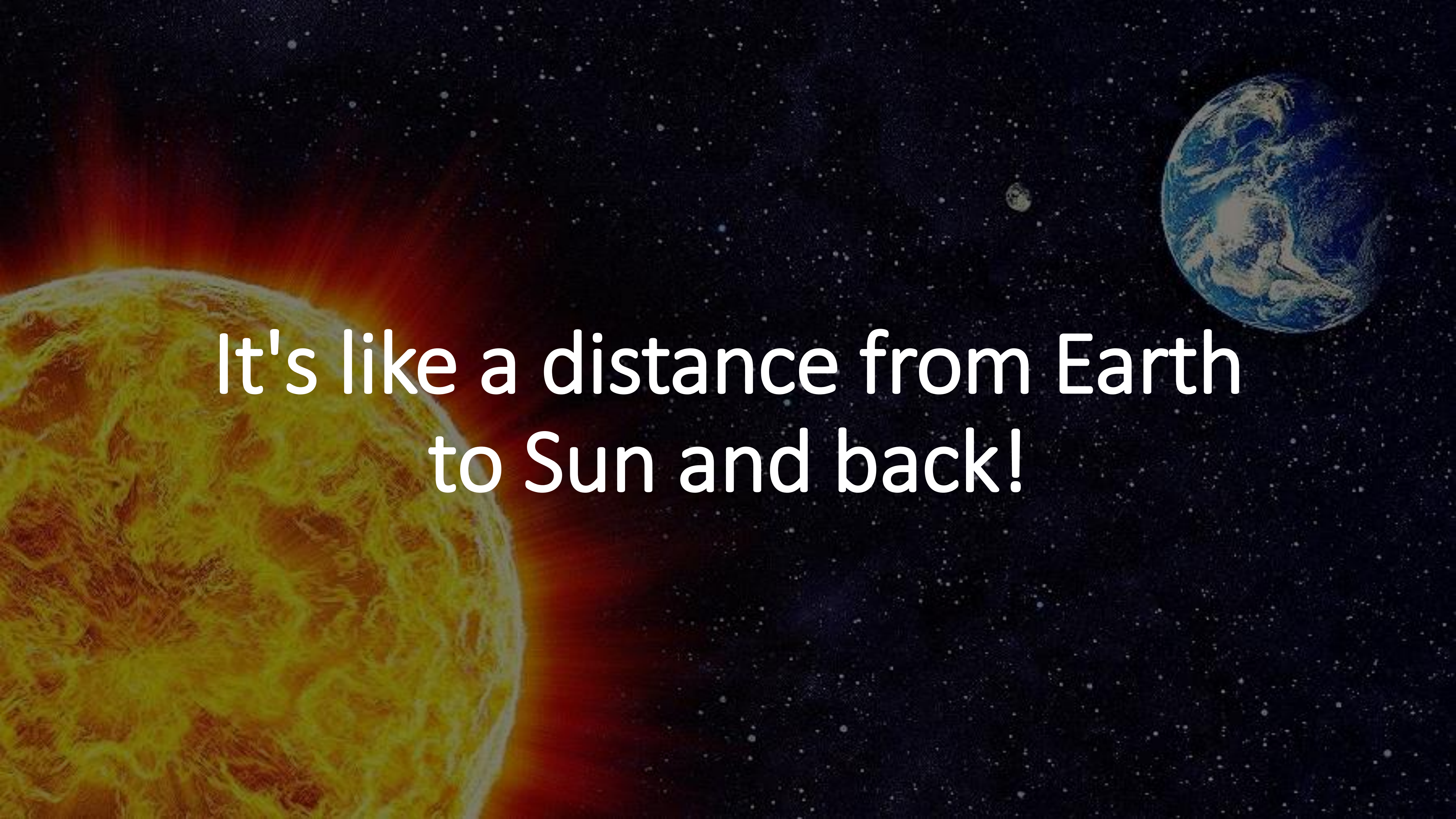
Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.



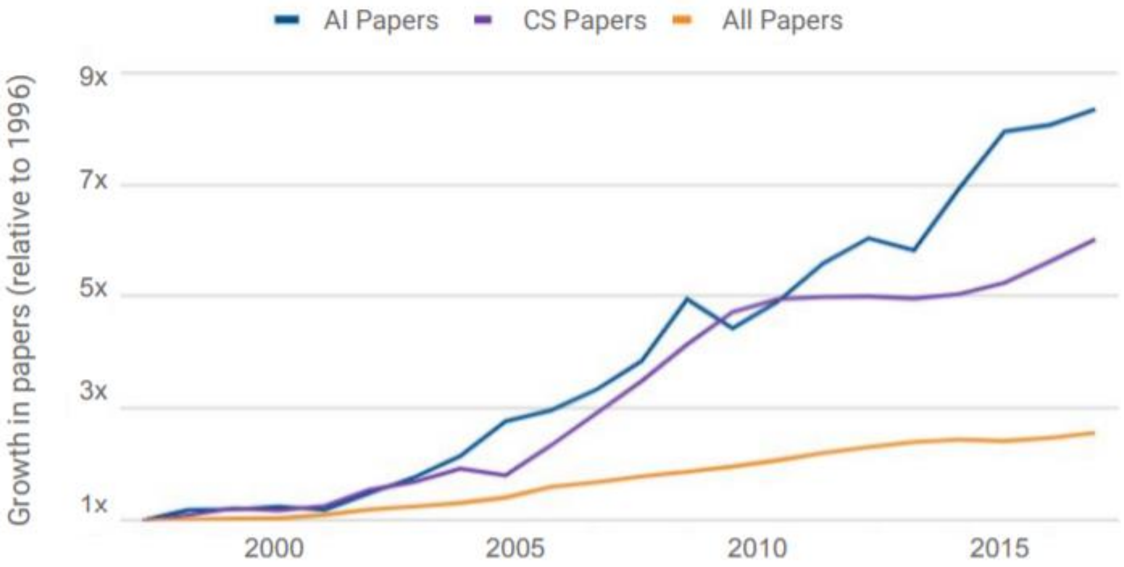
Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

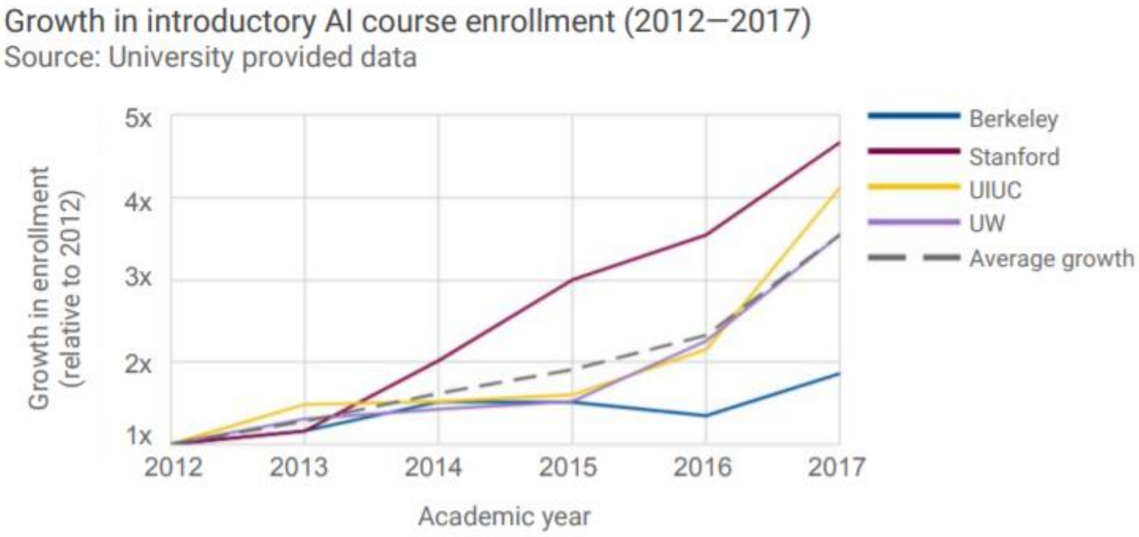
If the efficiency of internal combustion engines increased in accordance with Moore's Law, modern cars would cover a distance of 200 billion kilometers on one tank!


It's like a distance from Earth
to Sun and back!

Supercomputers Today

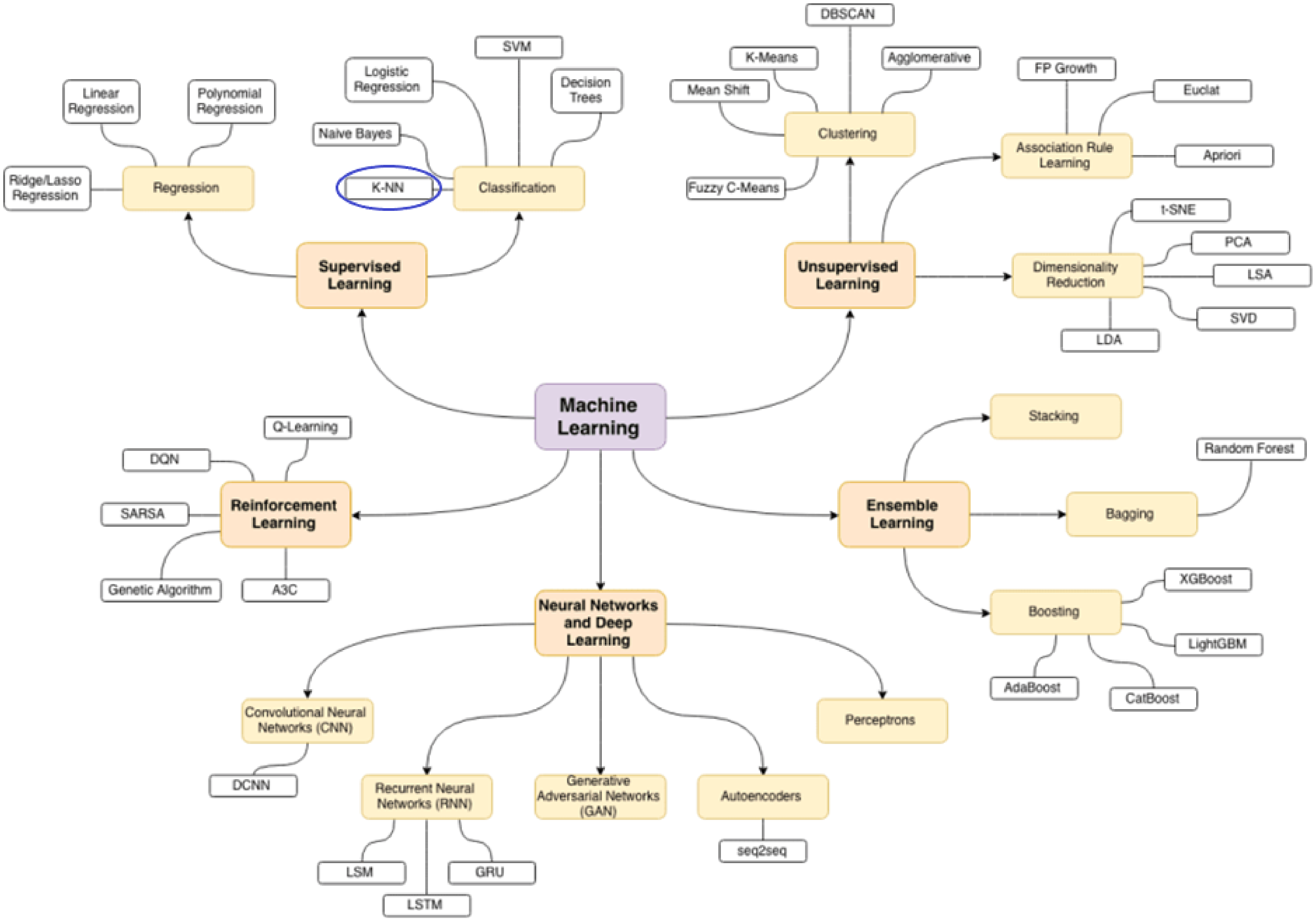
What is the future?





source: https://aiindex.org/

What Machine Learning is?



Machine Learning
Supervised Learning
Regression
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
Classification
Logistic Regression
SVM
Decision Trees
Naive Bayes
K-NN
Unsupervised Learning
Clustering
DBSCAN
K-Means
Agglomerative
Mean Shift
Fuzzy C-Means
Association Rule Learning
FP Growth
Euclat
Apriori
Dimensionality Reduction
t-SNE
PCA
LSA
SVD
LDA
Reinforcement Learning
Q-Learning
DQN
SARSA
Genetic Algorithm
A3C
Ensemble Learning
Stacking
Bagging
Random Forest
Boosting
XGBoost
LightGBM
AdaBoost
CatBoost
Neural Networks and Deep Learning
Convolutional Neural Networks (CNN)
DCNN
Recurrent Neural Networks (RNN)
LSM
LSTM
GRU
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq
Perceptrons

What is Machine Learning?

Meet Bartek, Gradient President!

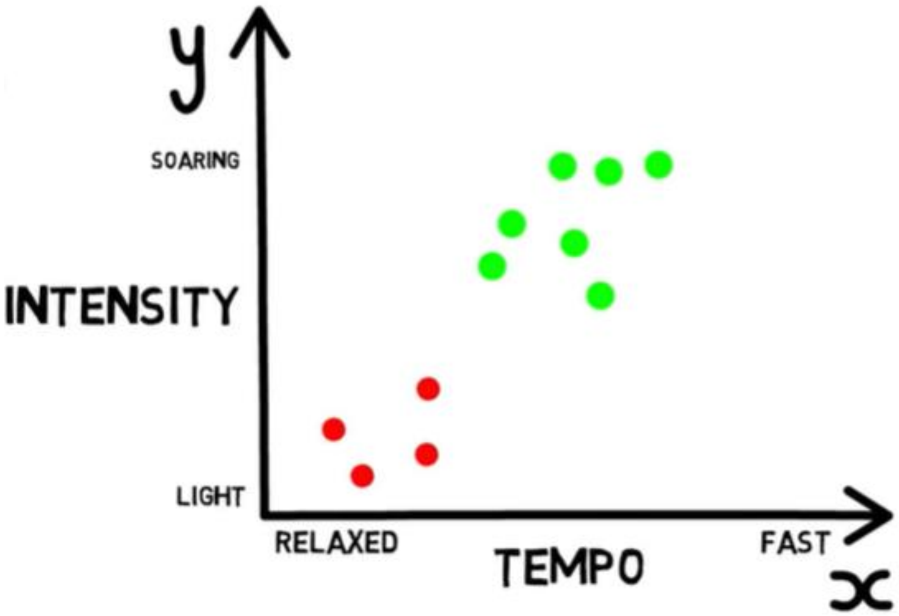
Bartek loves listening to new songs!

He decides whether he likes the song or not on the basis of the song:

- Tempo
- Genre
- Intensity
- Gender of Voice

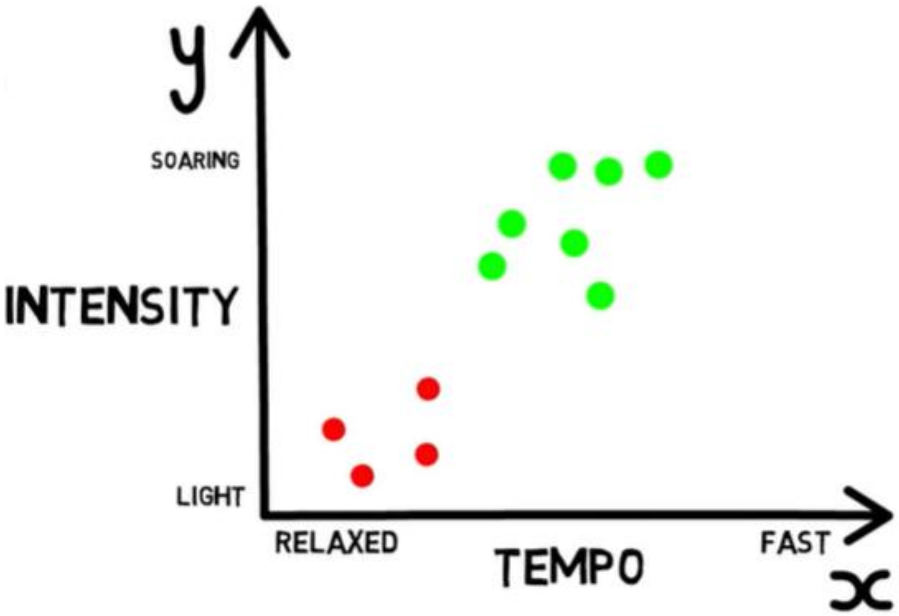
What is Machine Learning?

Let's take data from Bartek!



What is Machine Learning?

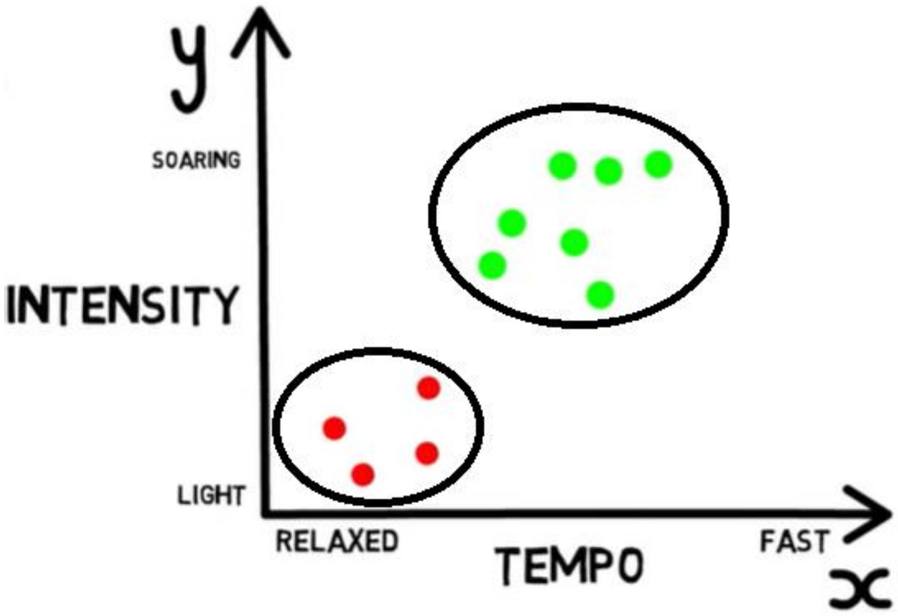
Let's take data from Bartek!



For example:
Bass Astral x Igo

What is Machine Learning?

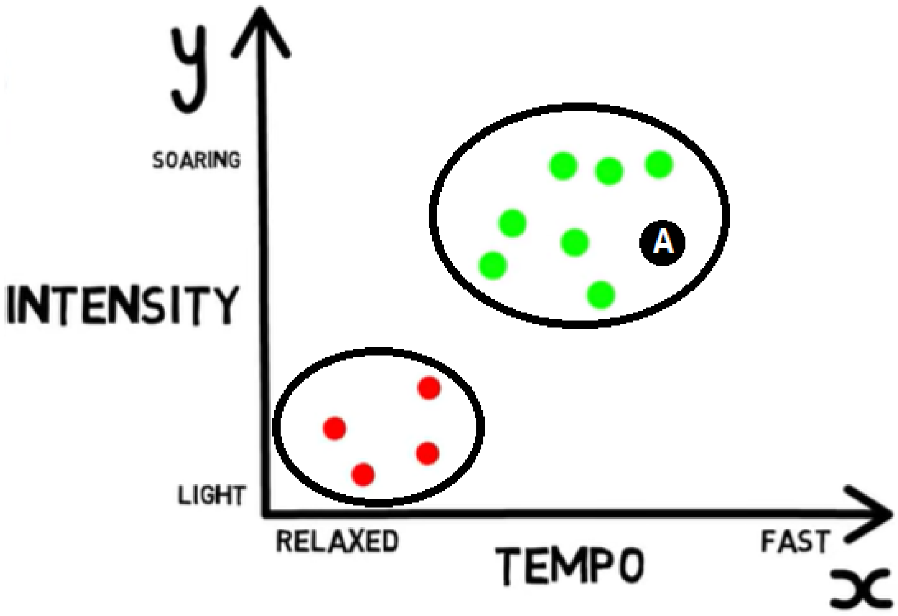
Let's take data from Bartek!



For example:
Bass Astral x Igo

What is Machine Learning?

Bartek found new song(A) on Spotify: "Sex on fire"!

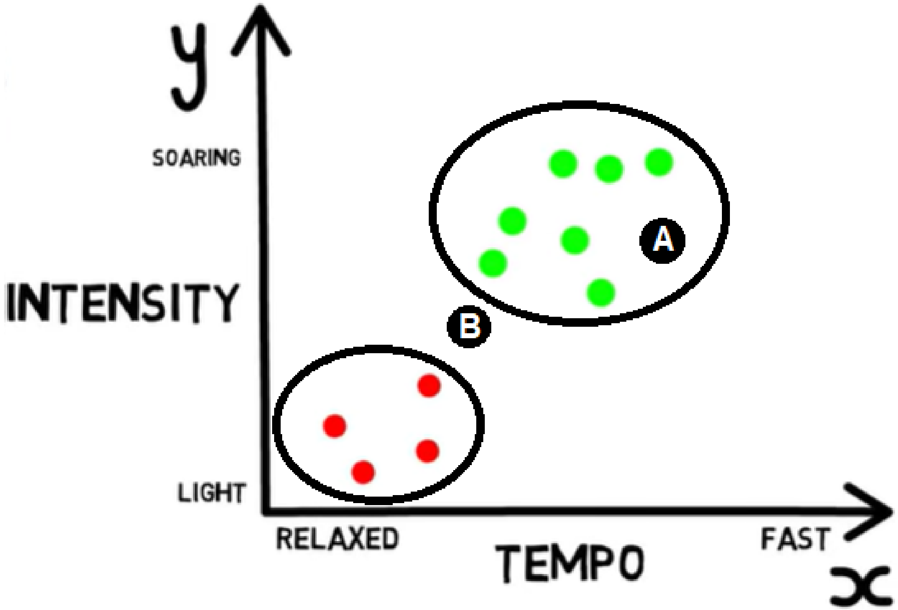


"Sex on fire" has:

- Medium tempo
- Soaring intensity

What is Machine Learning?

Now Bartek is listetning to new song "Better now" by Post Malone(B)



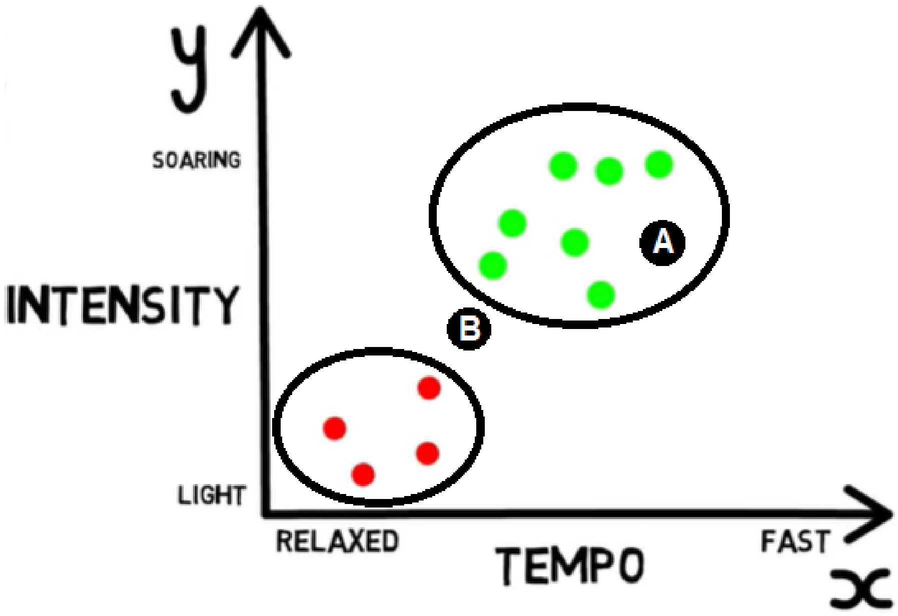
"Song B" has:

- Medium tempo
- Medium intensity

?

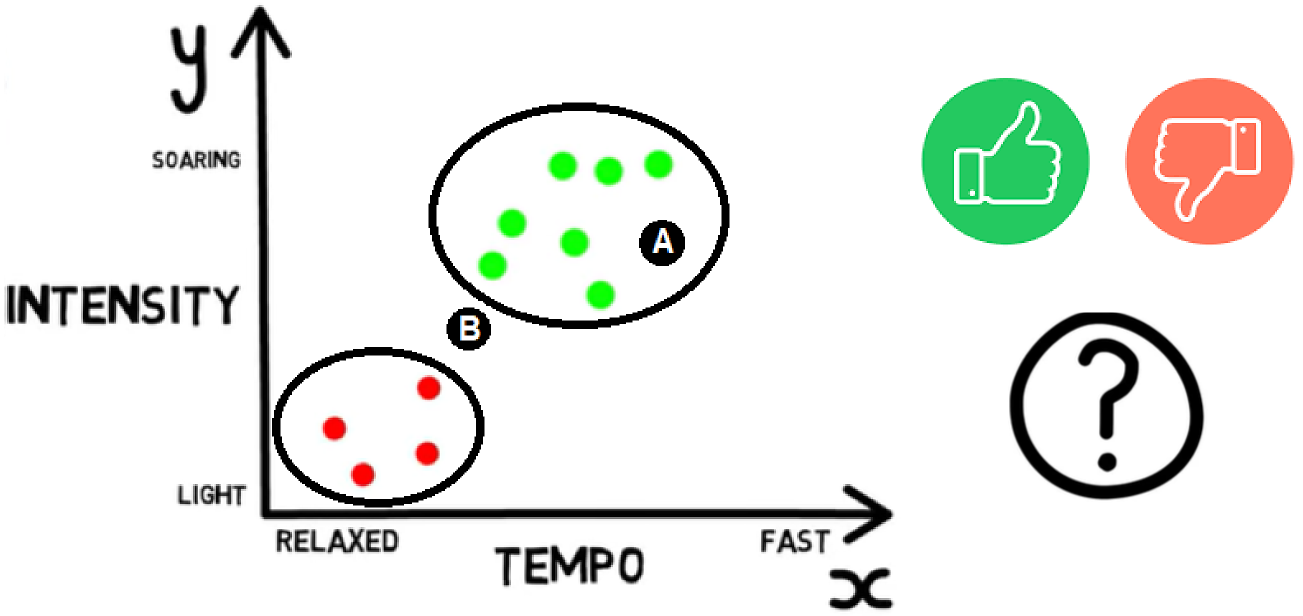
What is Machine Learning?

Will Bartek like this song or not?



What is Machine Learning?

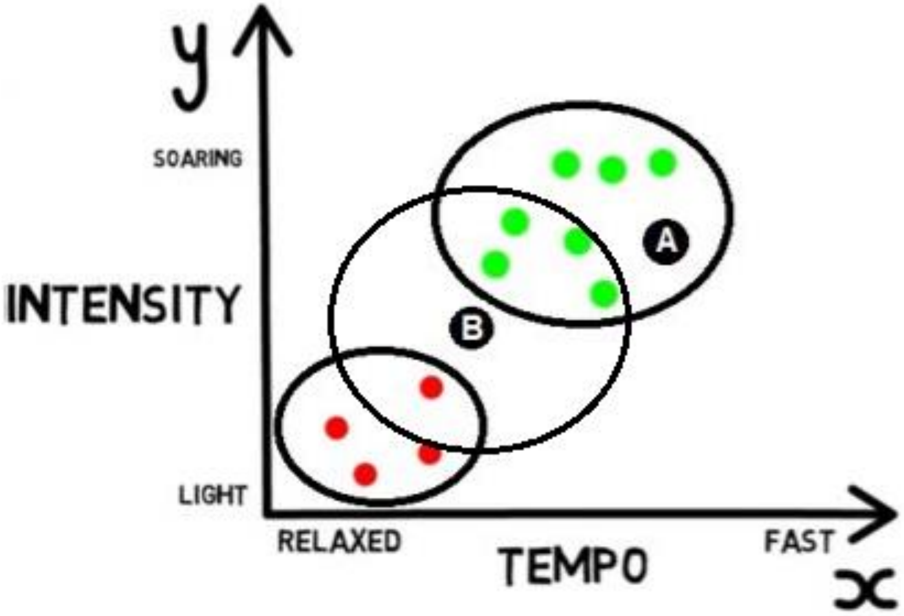
Will Bartek like this song or not?



Thats where Machine Learning comes in!

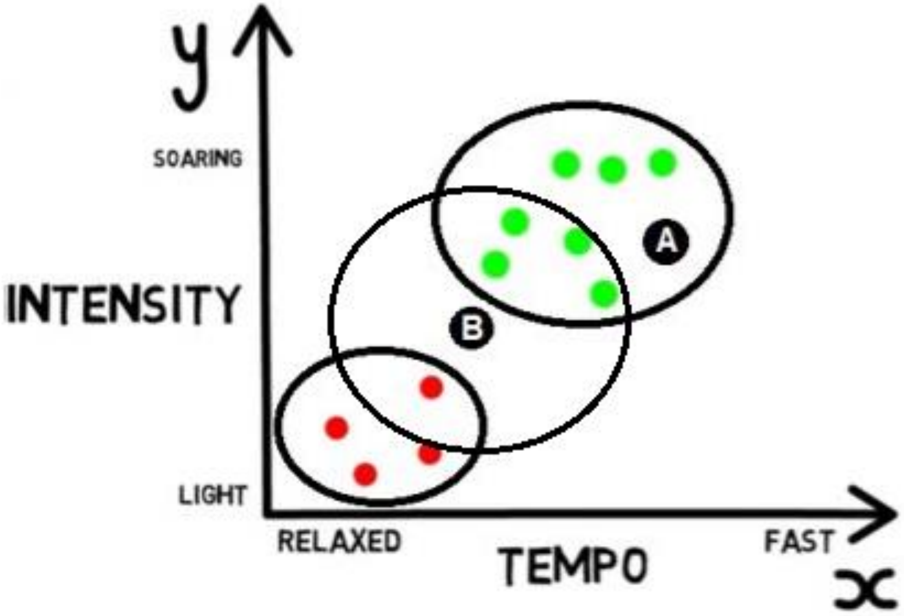
What is Machine Learning?

Bartek will definitely like this song!



K-Nearest Neighbors Algorithm(KNN)

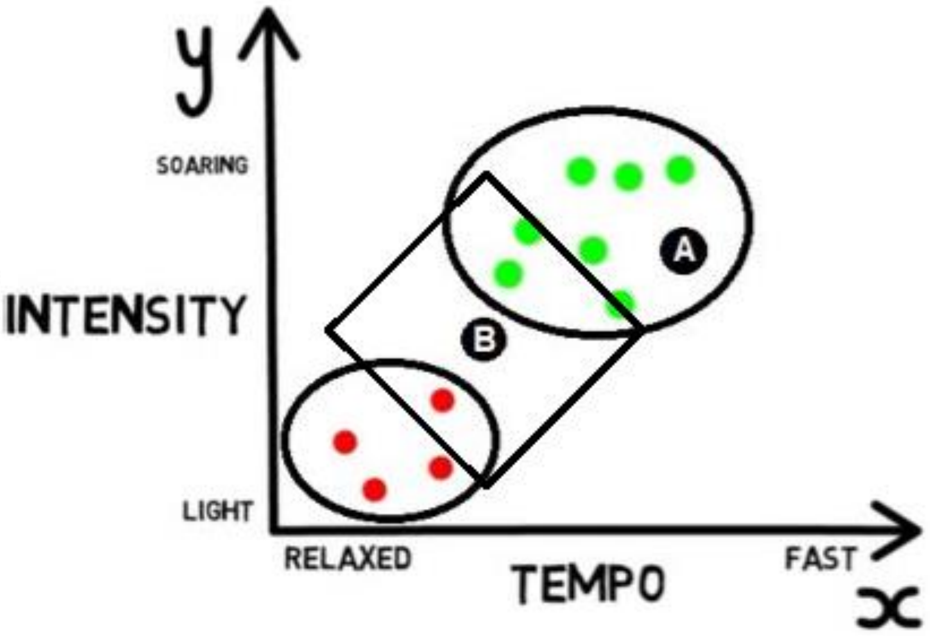
What is Machine Learning?



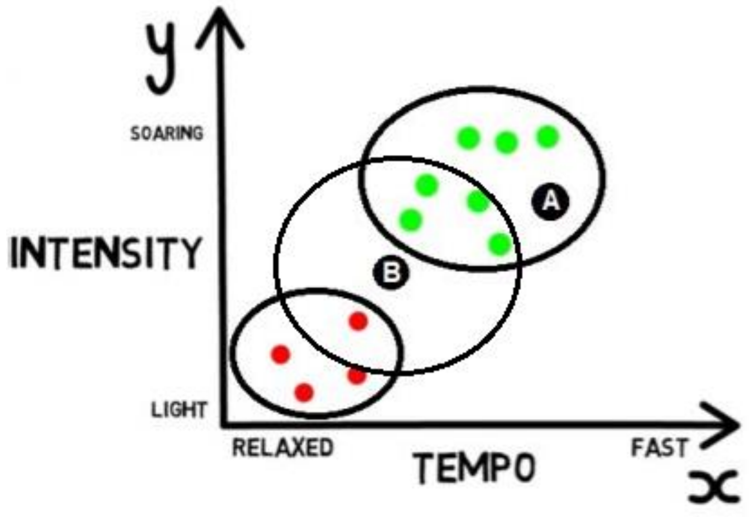
MORE DATA > BETTER MODEL > HIGHER ACCURACY

L1 norm in KNN

Different output of the algorithm!



K-Nearest Neighbors Algorithm(KNN)



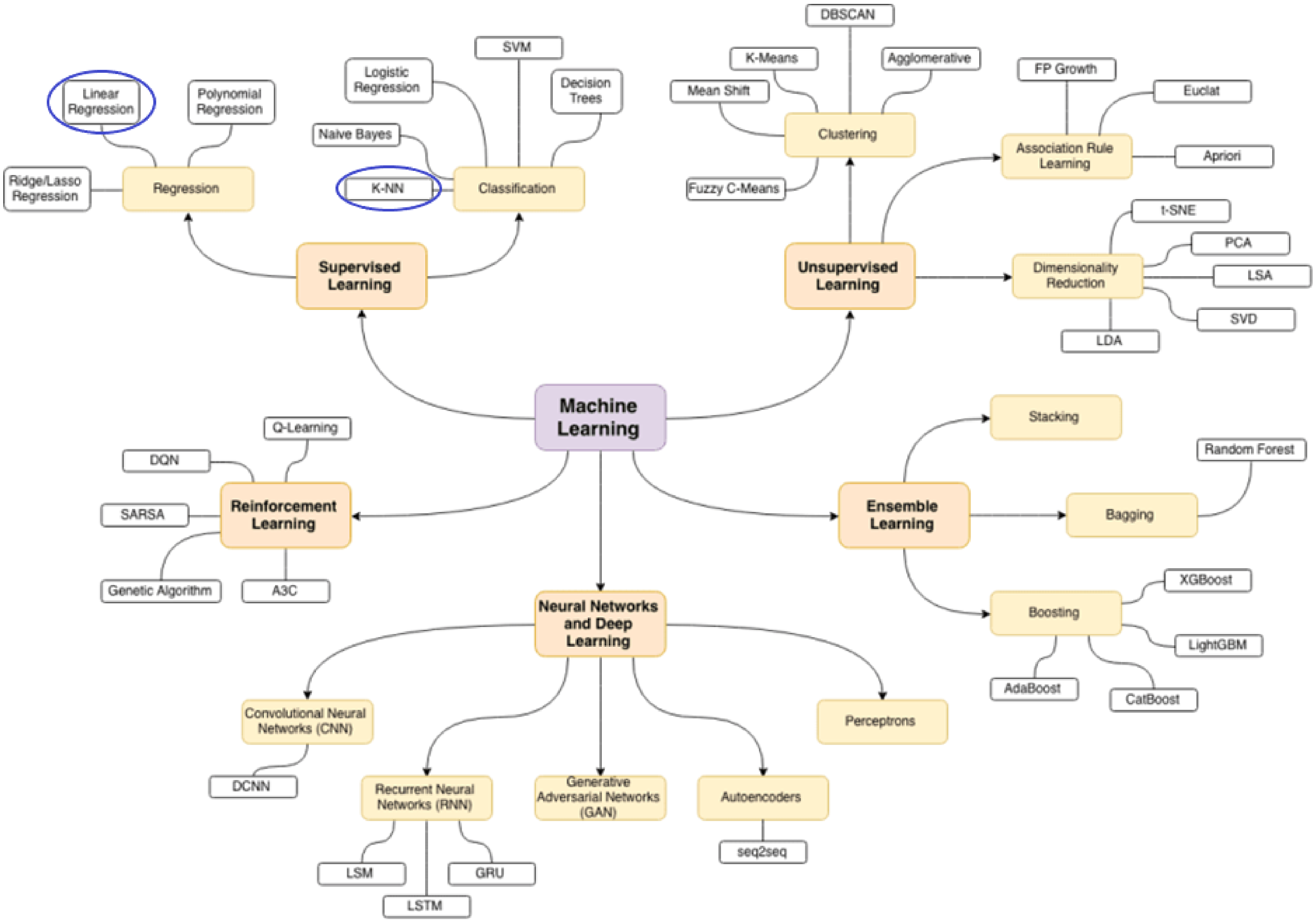
K-nearest neighbors in 3 easy steps:

1. Calculate the distance of a new data point to all other training data points(Euclidean or Manhatan etc.)
2. Then select the K-nearest data points, where K can be any integer
3. Make predictions

KNN in practise - Breast Cancer

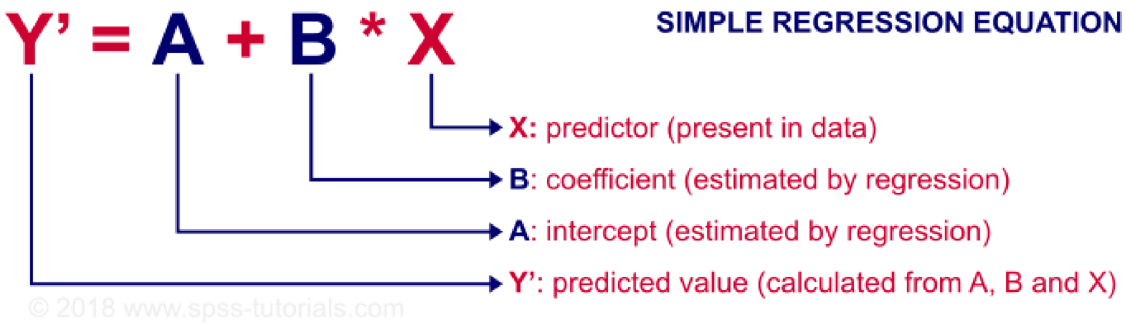
	id	diagnosis	radius_mean	texture_mean	perimeter_mean
	ID number	The diagnosis of breast tissues (M = malignant, B = benign)	mean of distances from center to points on the perimeter	standard deviation of gray-scale values	mean size of the core tumor
	8670 911m	B 63% M 37%	6.98 28.1	9.71 39.3	43.8 189
1	842302	M	17.99	10.38	122.8
2	842517	M	20.57	17.77	132.9
3	84300903	M	19.69	21.25	130
4	84348301	M	11.42	20.38	77.58
5	84358402	M	20.29	14.34	135.1
6	843786	M	12.45	15.7	82.57

Using Breast Cancer Wisconsin Data Set predict if tumor will be maligant or bengin


Machine Learning
Supervised Learning
Regression
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
Classification
Logistic Regression
SVM
Decision Trees
Naive Bayes
K-NN
Unsupervised Learning
Clustering
DBSCAN
K-Means
Agglomerative
Mean Shift
Fuzzy C-Means
Association Rule Learning
FP Growth
Euclat
Apriori
Dimensionality Reduction
t-SNE
PCA
LSA
SVD
LDA
Reinforcement Learning
Q-Learning
DQN
SARSA
Genetic Algorithm
A3C
Ensemble Learning
Stacking
Bagging
Random Forest
Boosting
XGBoost
LightGBM
AdaBoost
CatBoost
Neural Networks and Deep Learning
Convolutional Neural Networks (CNN)
DCNN
Recurrent Neural Networks (RNN)
LSM
LSTM
GRU
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq
Perceptrons

Linear Regression

Simple Linear Regression



Multivariable Linear Regression

$$Y(x_1, x_2, x_3) = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_0$$

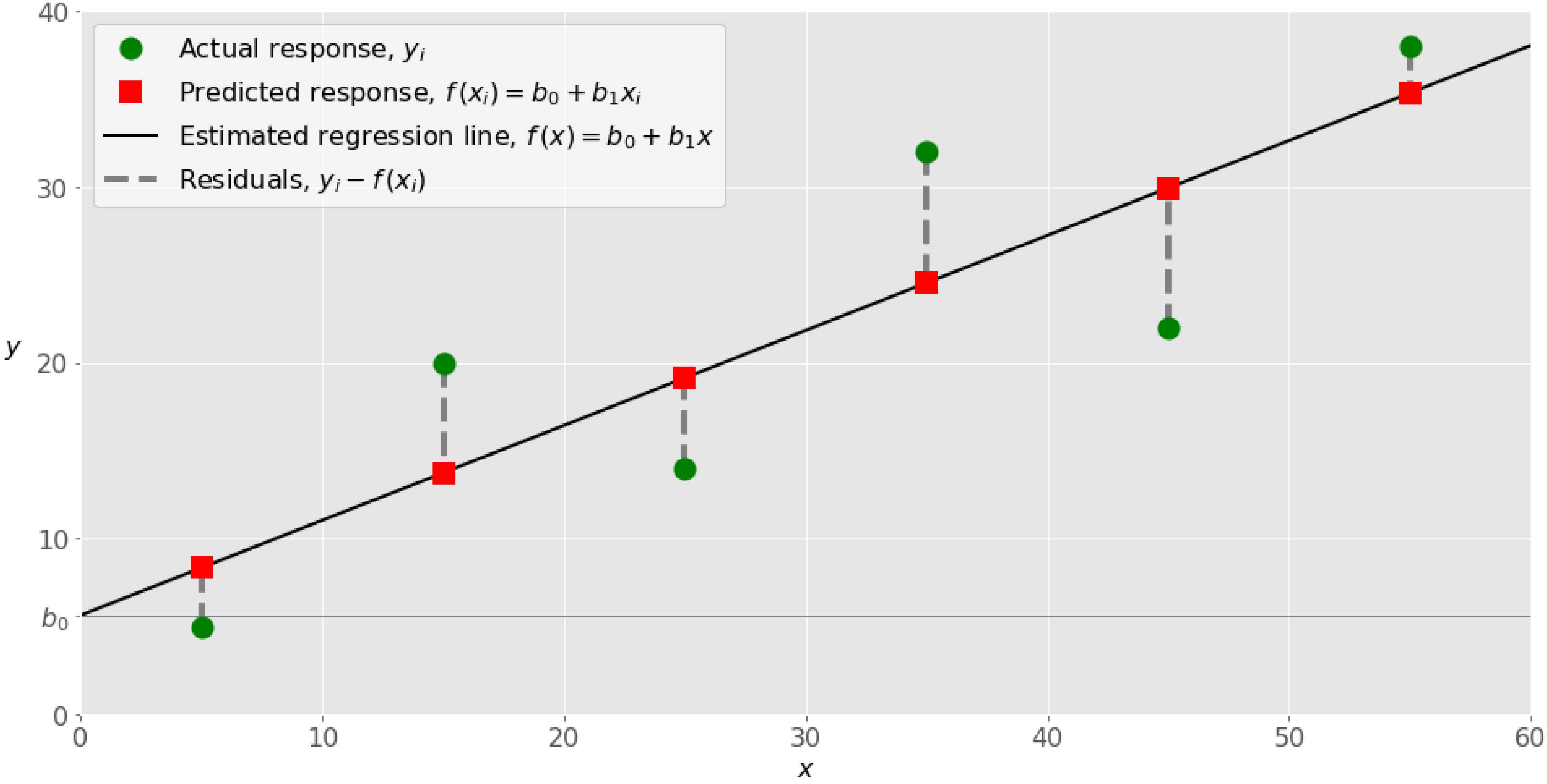
Linear Regression

Mean Squarred Error (MSE) Cost Function

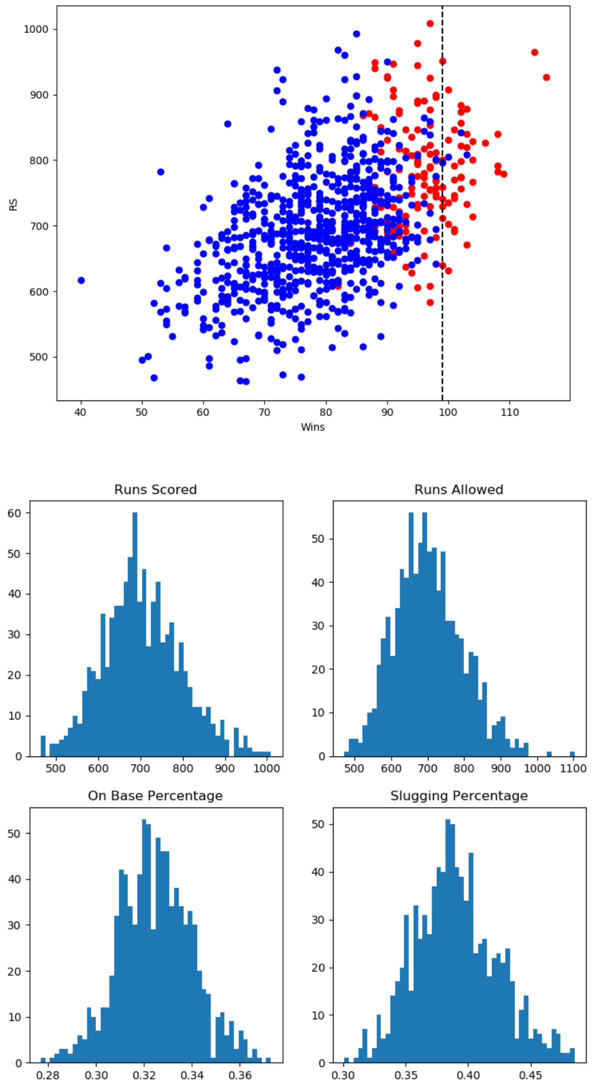
$$MSE = \frac{1}{N}\sum_{i}^{n}(Y_i - y_i)^2$$

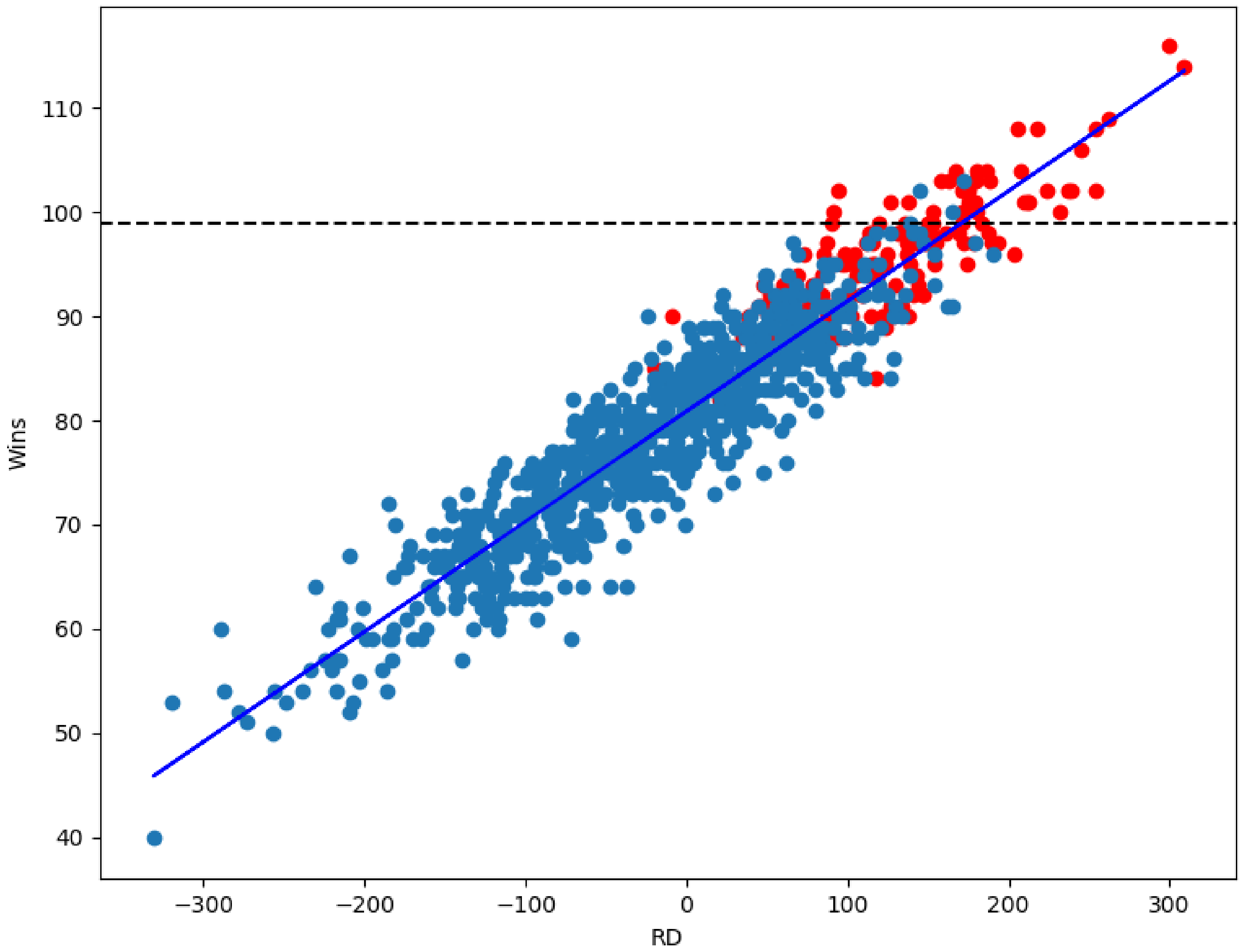
- N - number of points
- Y_i - predicted value
- y_i - actual value

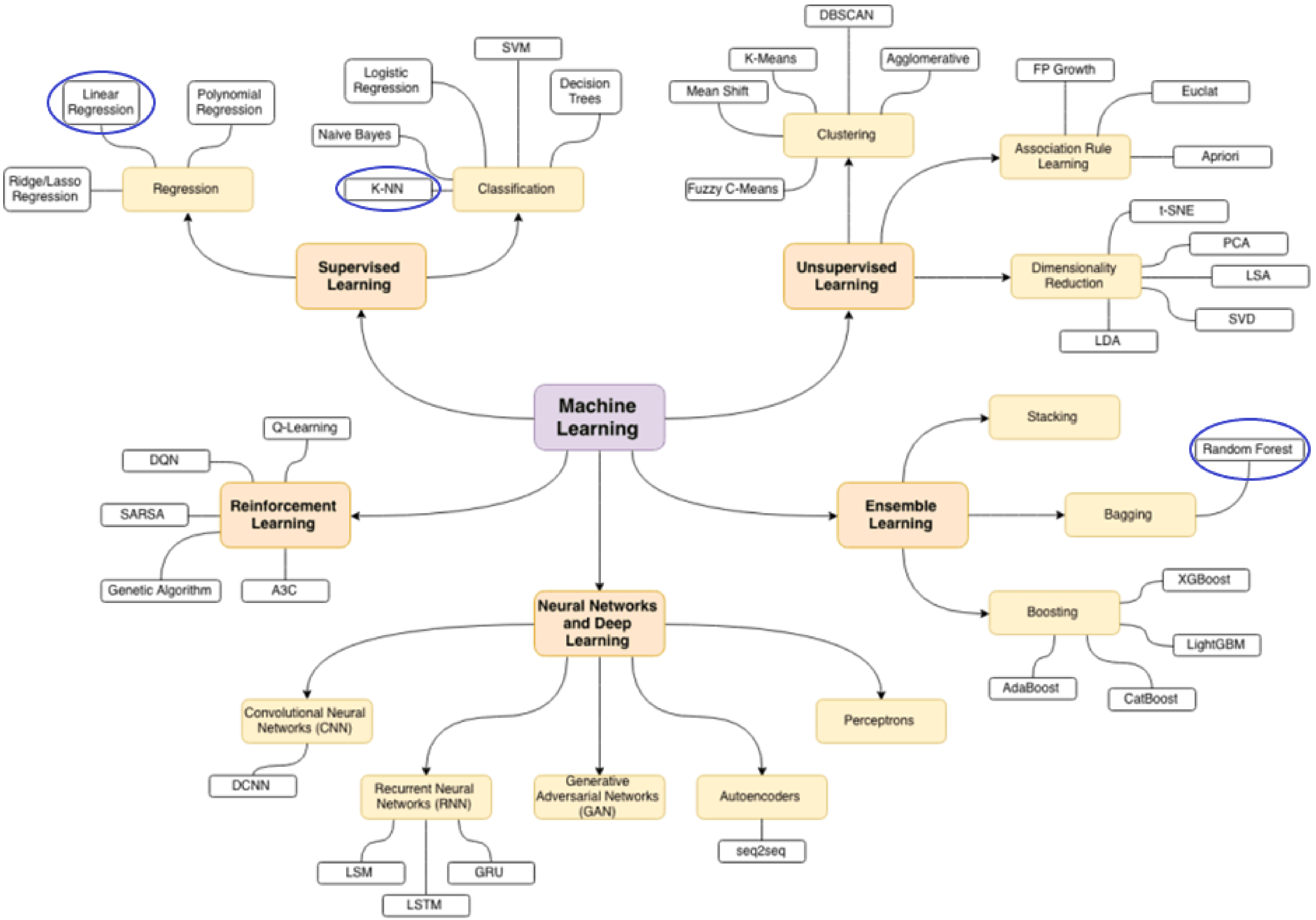
Linear Regression



How Linear Regression broke the history in baseball?

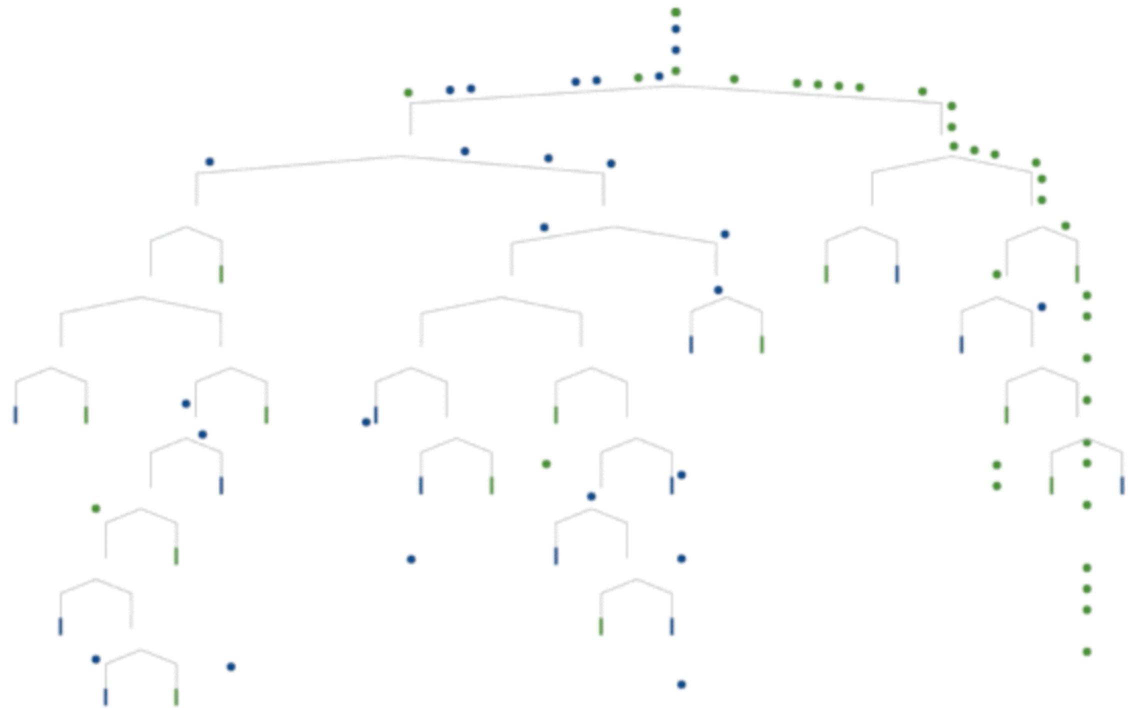






Machine Learning
Supervised Learning
Regression
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
Classification
Logistic Regression
SVM
Decision Trees
Naive Bayes
K-NN
Unsupervised Learning
Clustering
DBSCAN
K-Means
Agglomerative
Mean Shift
Fuzzy C-Means
Association Rule Learning
FP Growth
Euclat
Apriori
Dimensionality Reduction
t-SNE
PCA
LSA
SVD
LDA
Reinforcement Learning
Q-Learning
DQN
SARSA
Genetic Algorithm
A3C
Ensemble Learning
Stacking
Bagging
Random Forest
Boosting
XGBoost
LightGBM
AdaBoost
CatBoost
Neural Networks and Deep Learning
Convolutional Neural Networks (CNN)
DCNN
Recurrent Neural Networks (RNN)
LSM
LSTM
GRU
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq
Perceptrons

Classification and Regression Trees (CARTs)

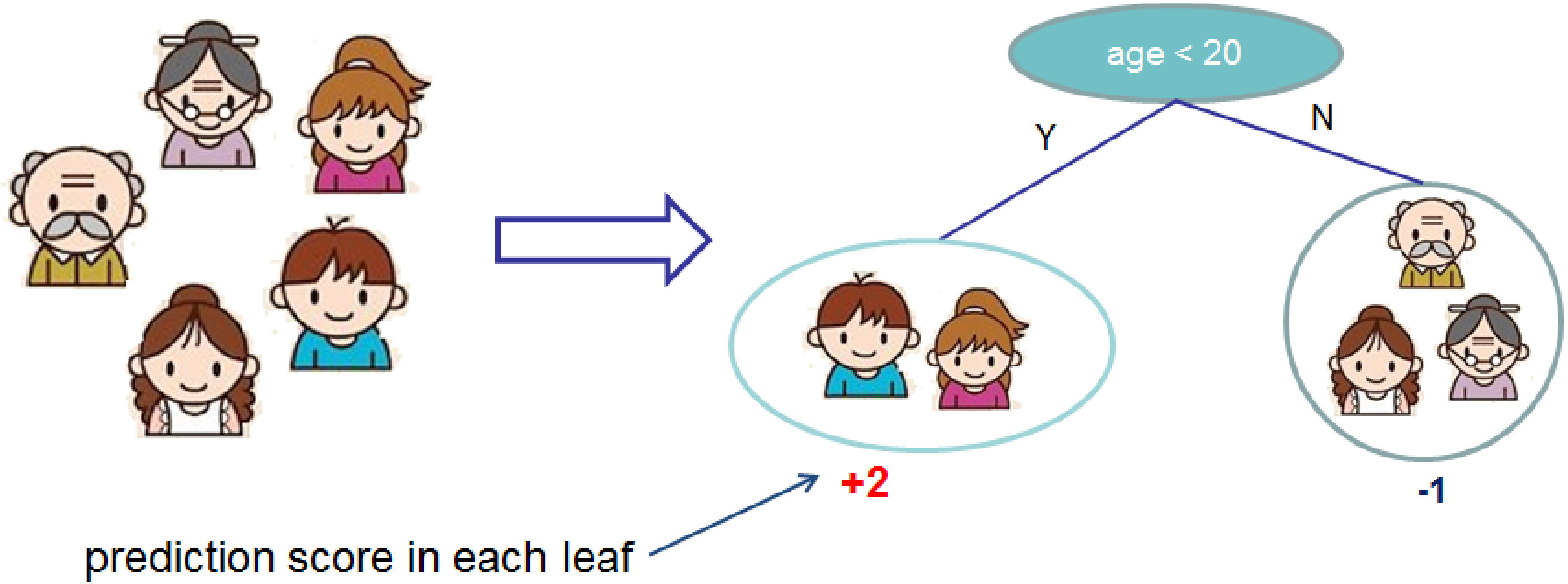


It just asks the series of questions!

Classification and Regression Trees

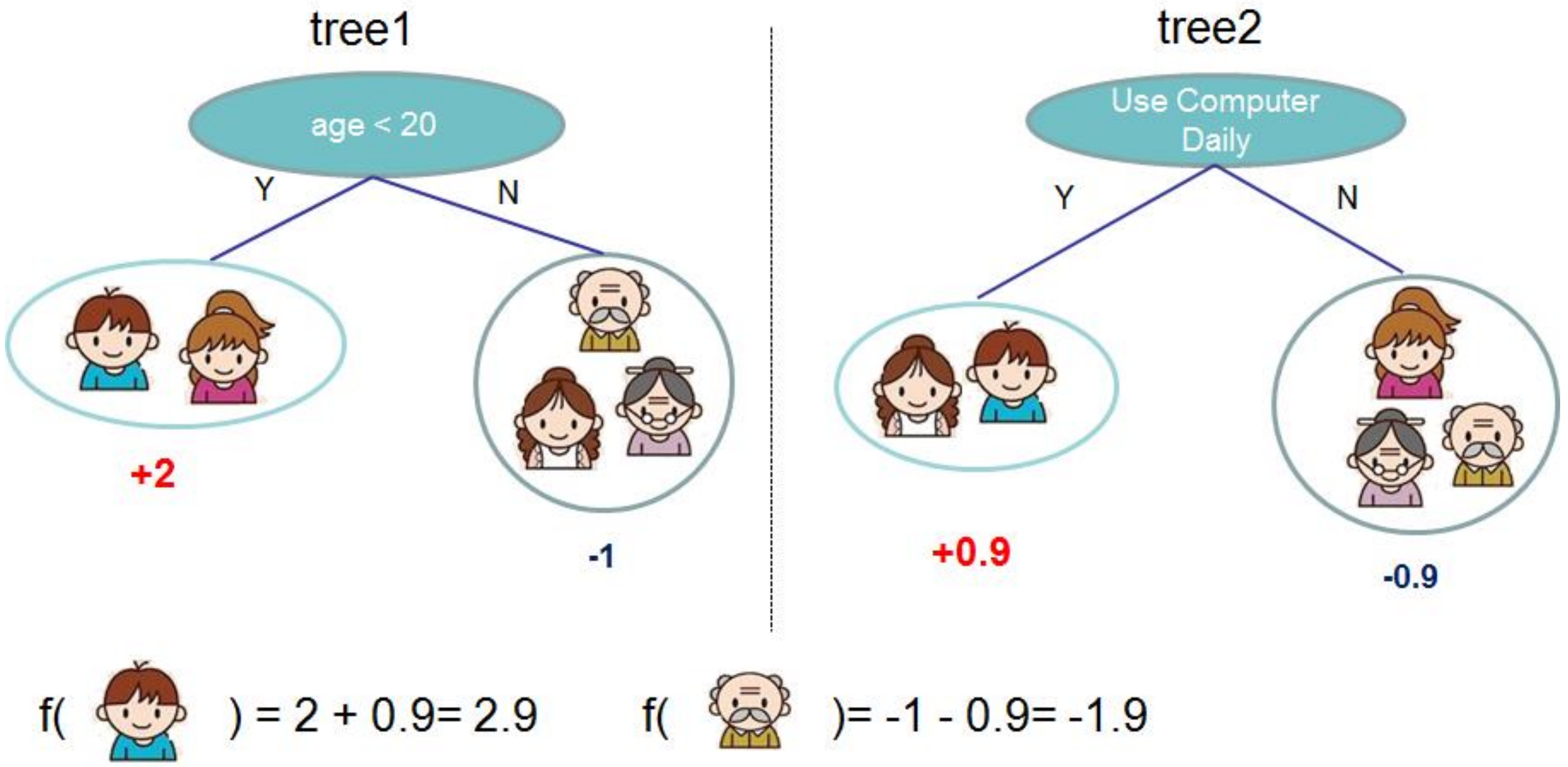
Input: age, gender, occupation, …

Like the computer game X



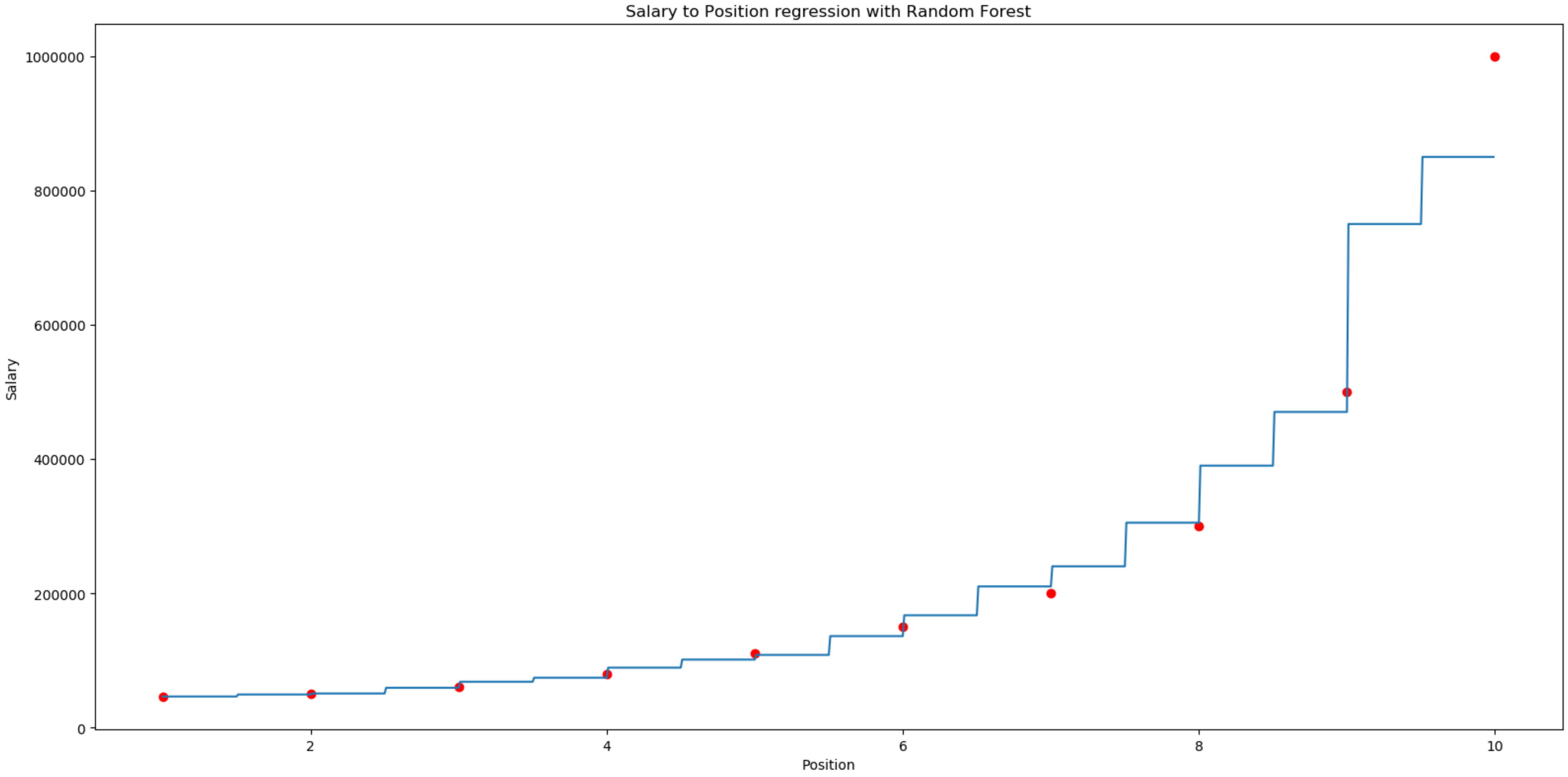
source: https://xgboost.readthedocs.io/en/latest/tutorials/model.html

Classification and Regression Trees

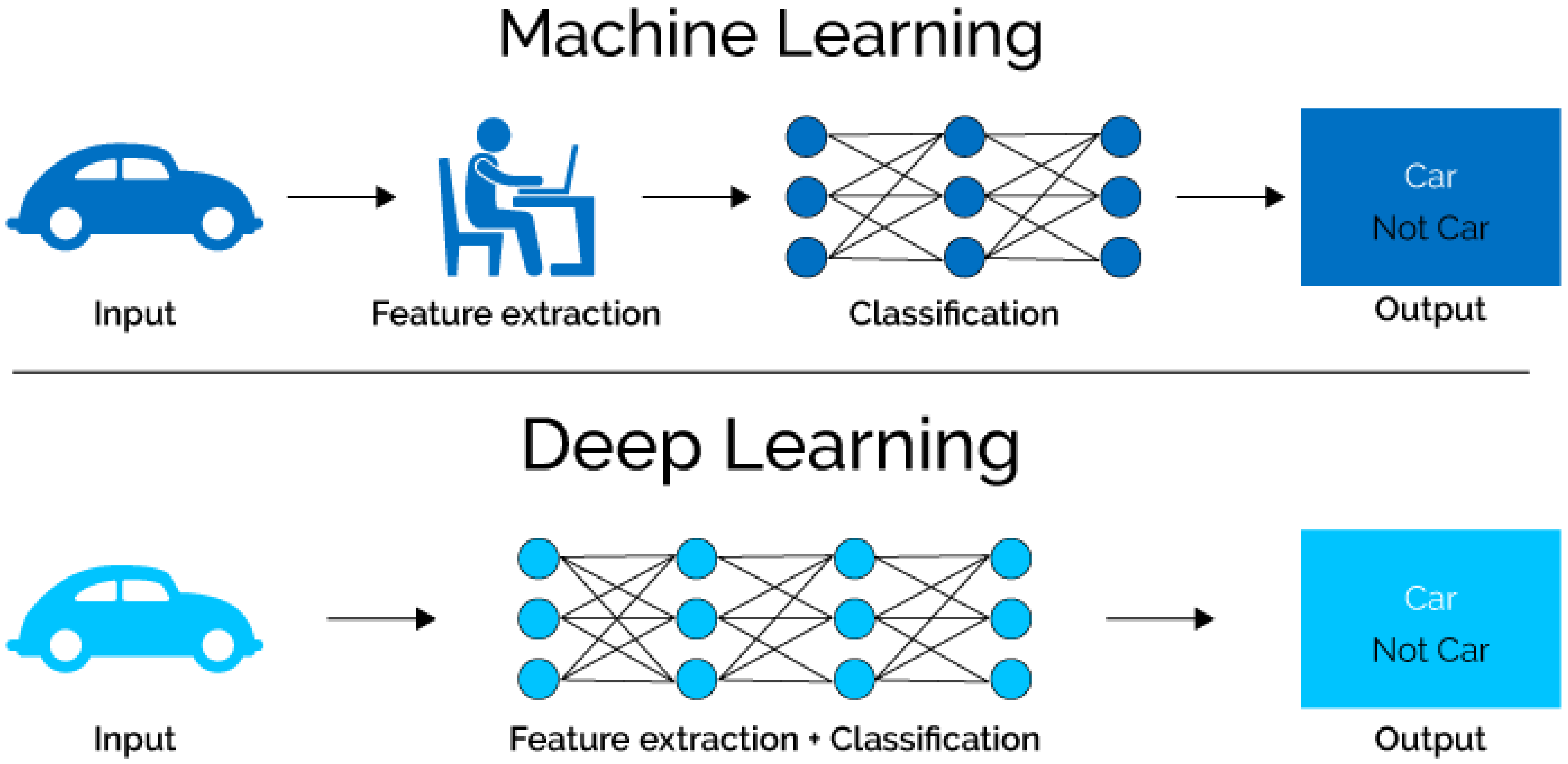


source: https://xgboost.readthedocs.io/en/latest/tutorials/model.html

Random Forest Regression

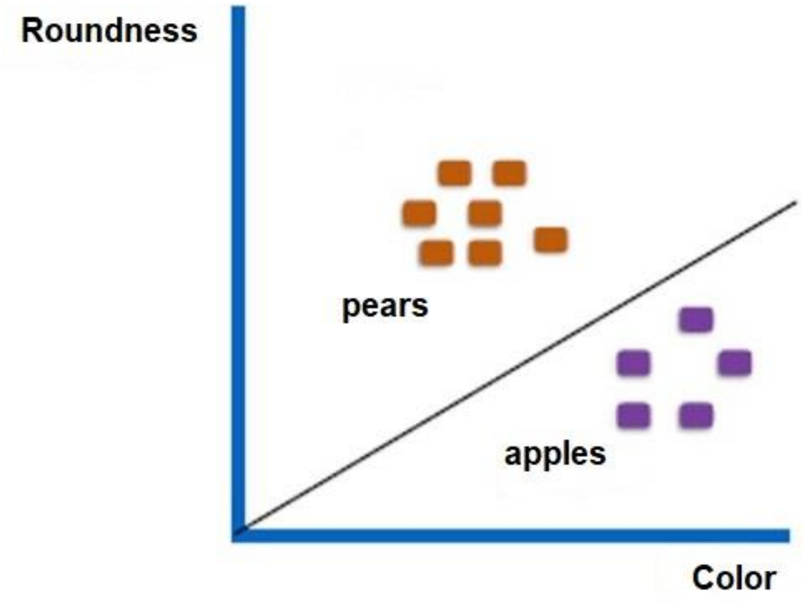


How Artificial Neural Network works?

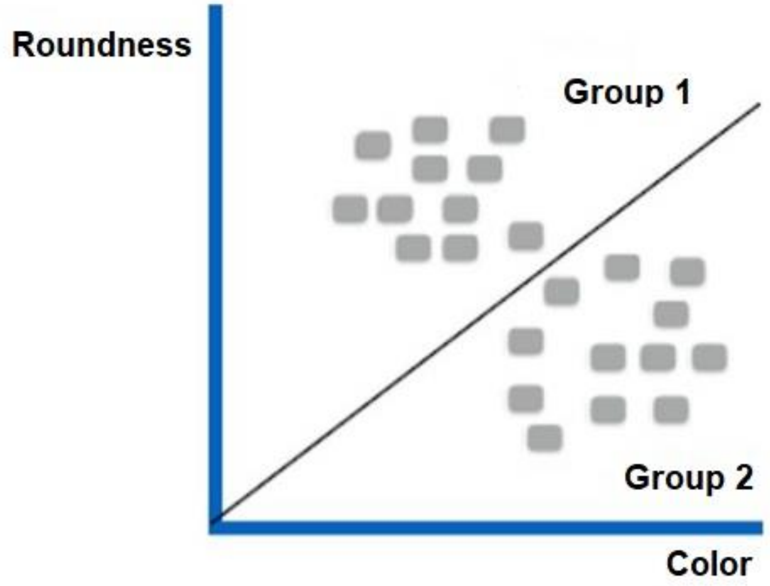


source: https://www.researchgate.net/publication/326834670_Geological_Facies_Prediction_Using_Computed_Tomography_in_a_Machine_Learning_and_Deep_Learning_Environment/figures?lo=1

Types of learning



Supervised Learning

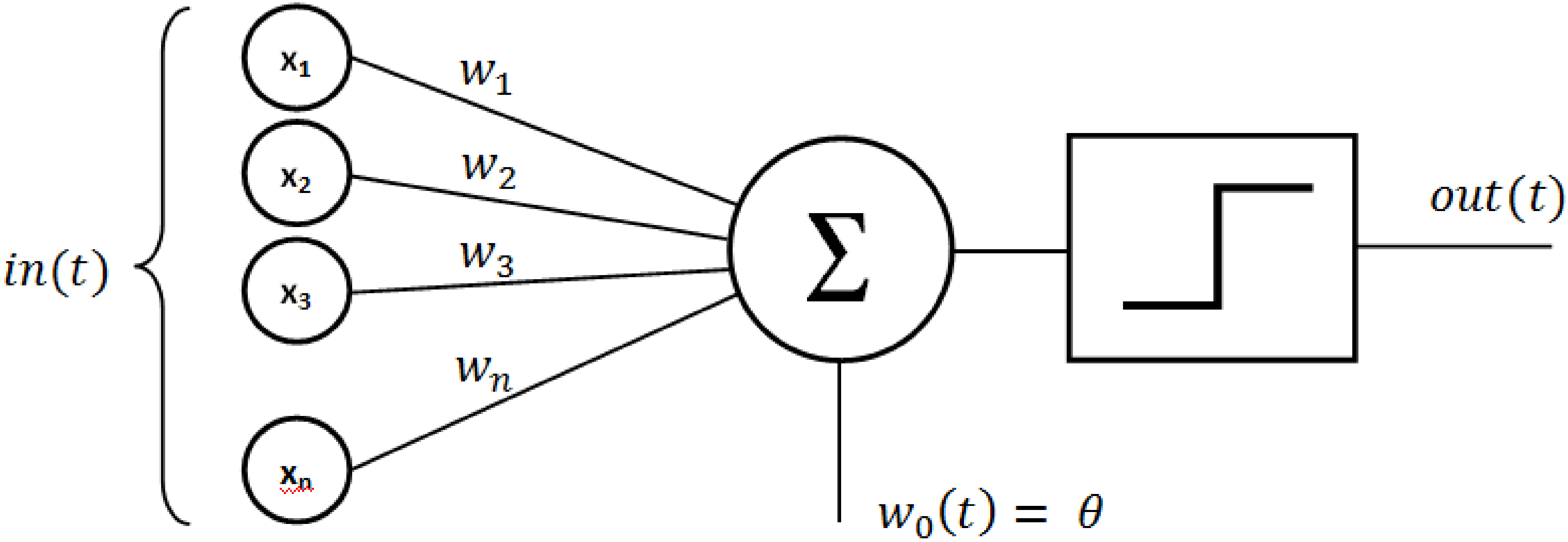


Unsupervised Learning

Reinforcement Learning

https://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html

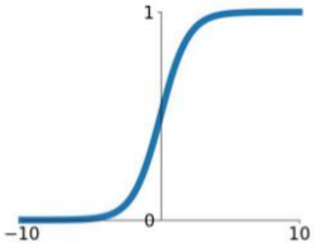
Percepton



Types of activation functions

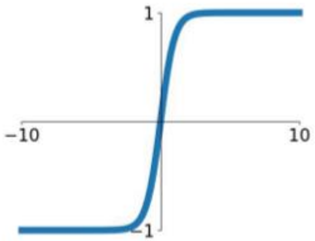
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



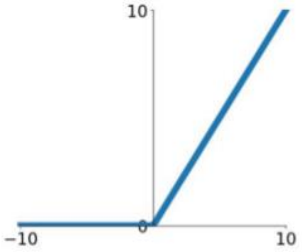
tanh

$\tanh(x)$



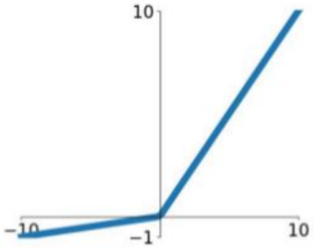
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

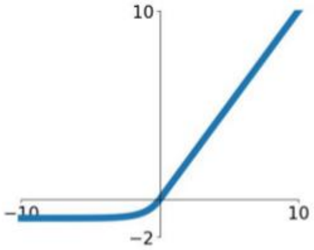


Maxout

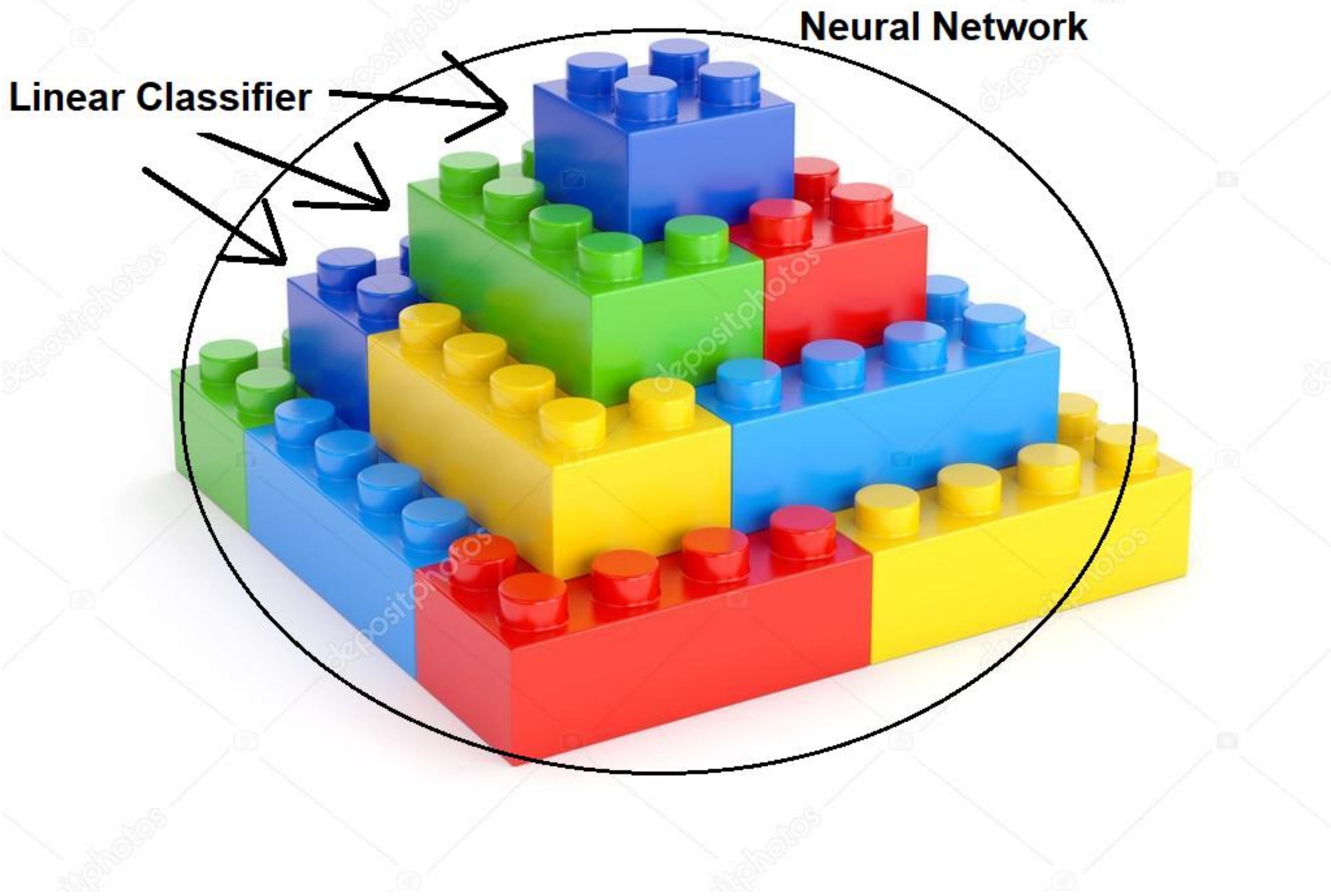
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

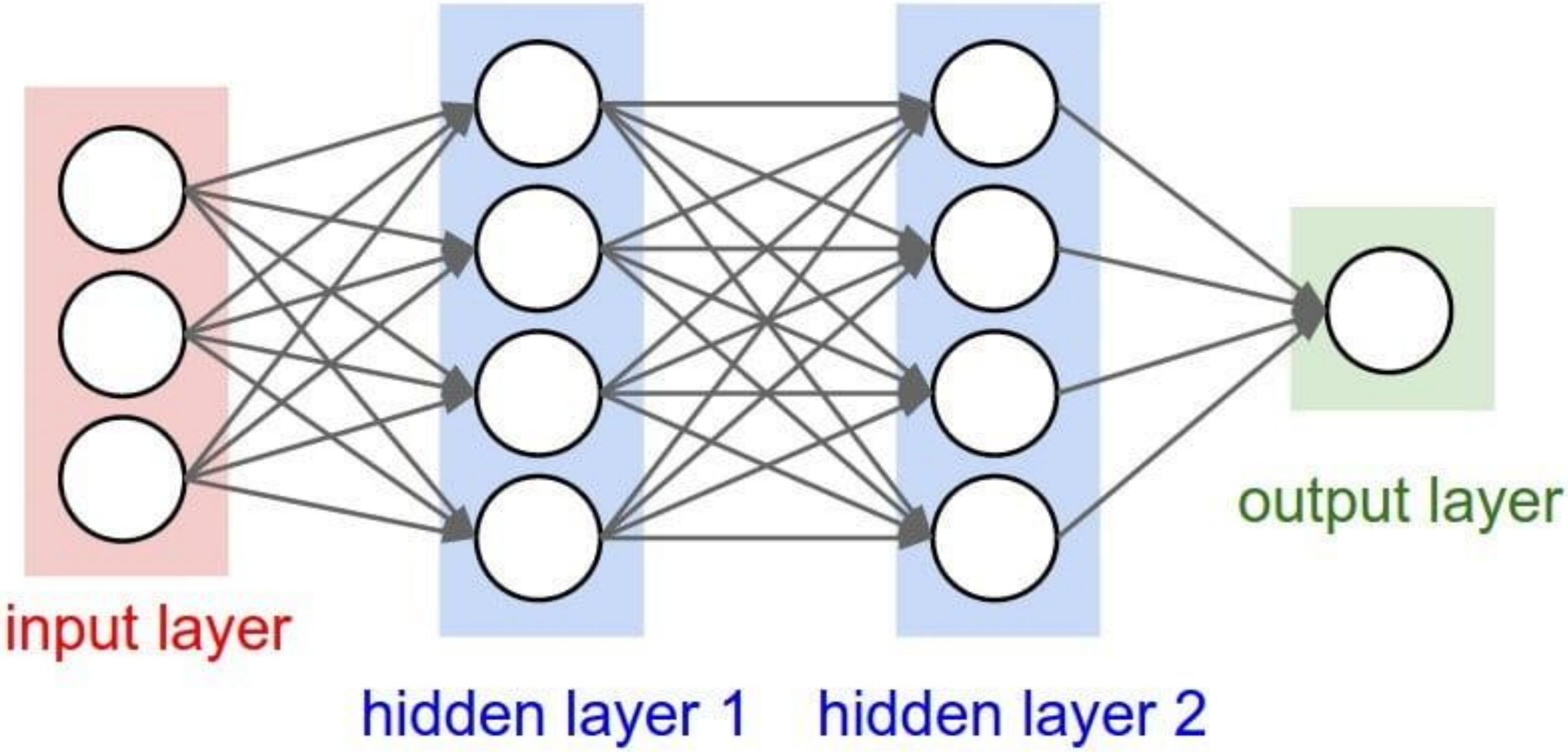
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Neural Network



Artificial Neural Network



Loss function - how good our classifier is

Loss over the dataset is a sum of losses over samples divided by number of samples

$$L = \frac{1}{N} \sum_i L_i$$

- Multiclass SVM Loss

$$L_i = \sum_{j \neq y_i} \max(0, f_j - f_{y_i} + 1)$$

- Cross-entropy Loss

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

How to optimize?

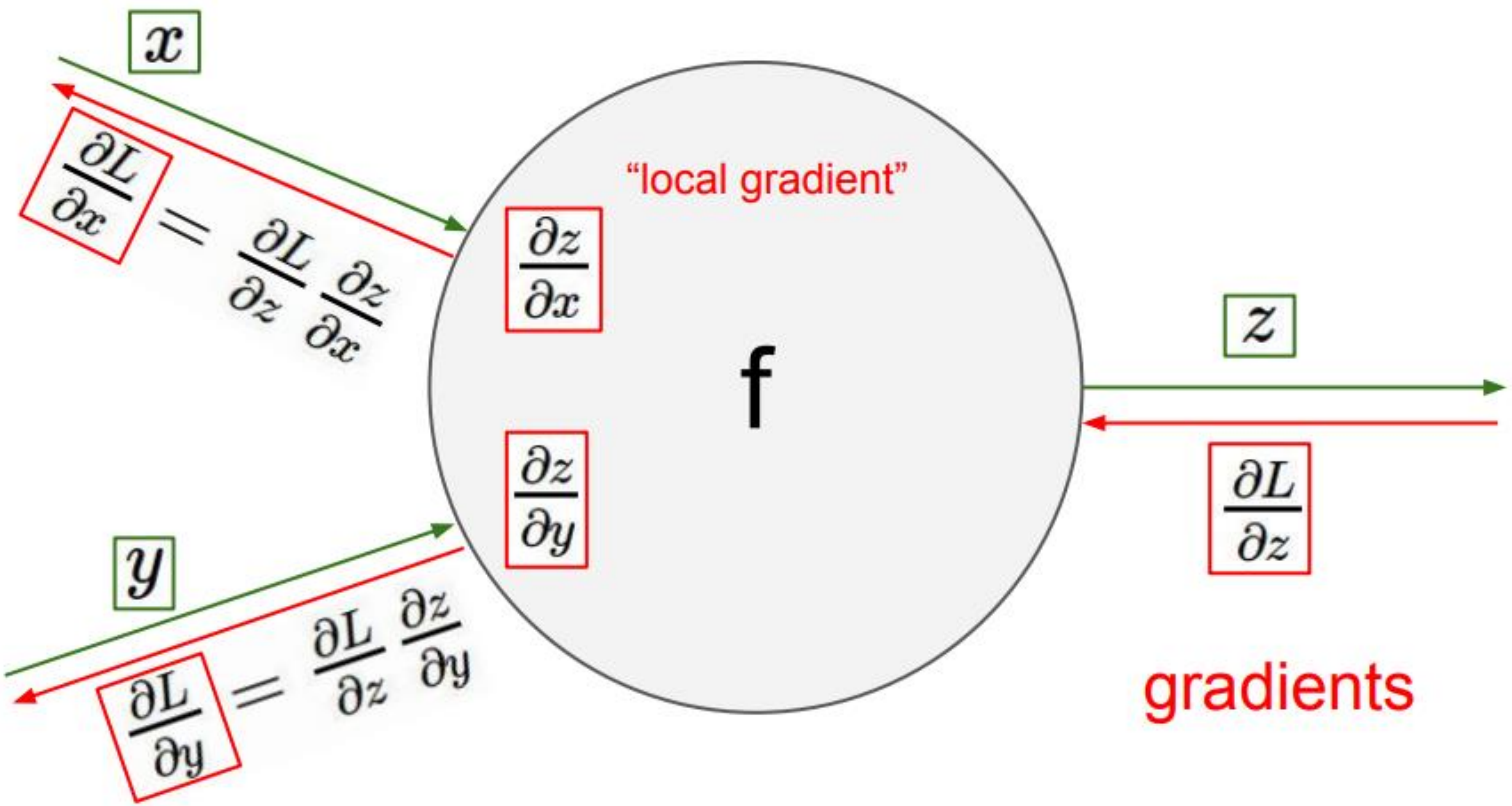
Gradient

$$\frac{df(x)}{dx} = \frac{f(x+h) - f(x)}{h} \quad \text{(bad, do not use)}$$

$$\frac{df(x)}{dx} = \frac{f(x+h) - f(x-h)}{2h} \quad \text{(use instead)}$$

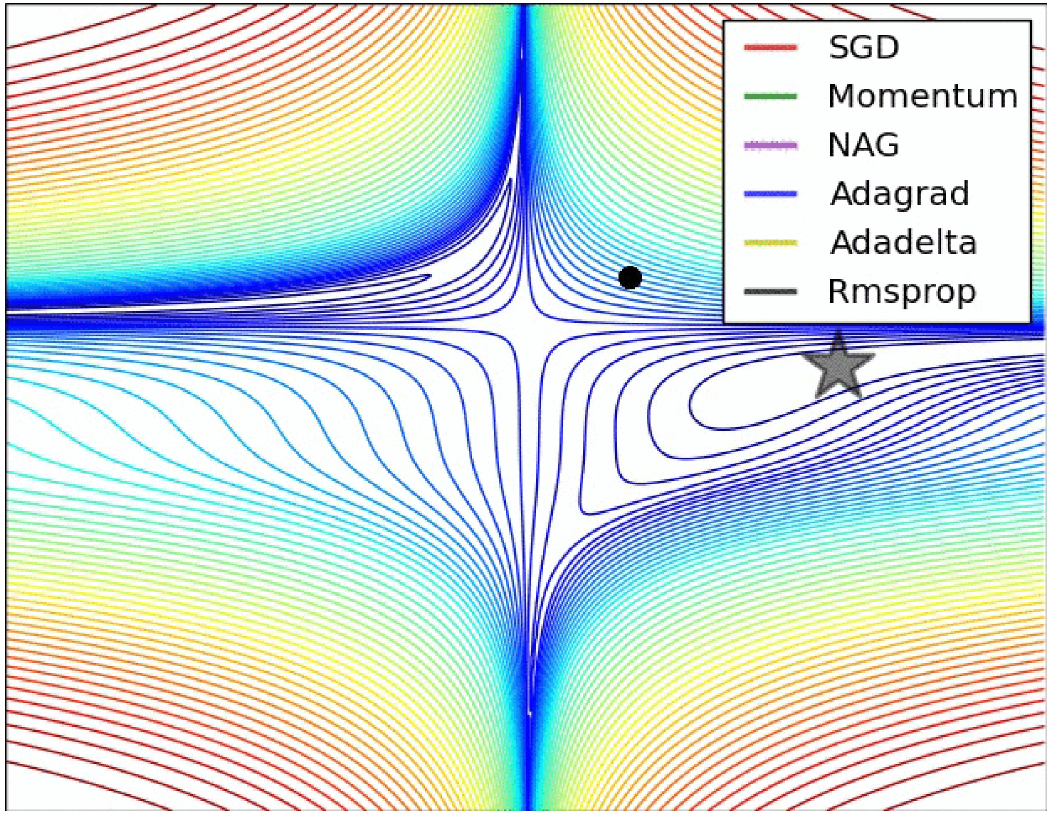
$$\nabla f(p) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(p) \\ \vdots \\ \frac{\partial f}{\partial x_n}(p) \end{bmatrix}$$

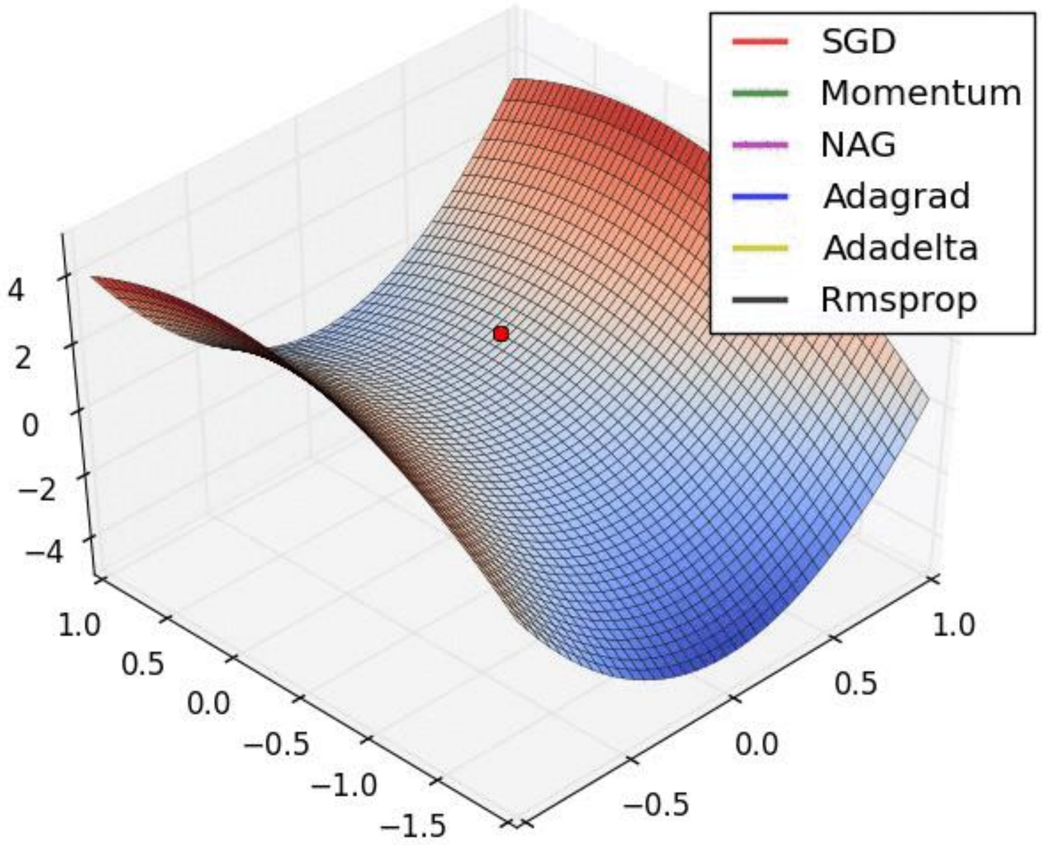
Chain rule



x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
"local gradient"
$\frac{\partial z}{\partial x}$
f
$\frac{\partial z}{\partial y}$
z
$\frac{\partial L}{\partial z}$
y
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$
gradients

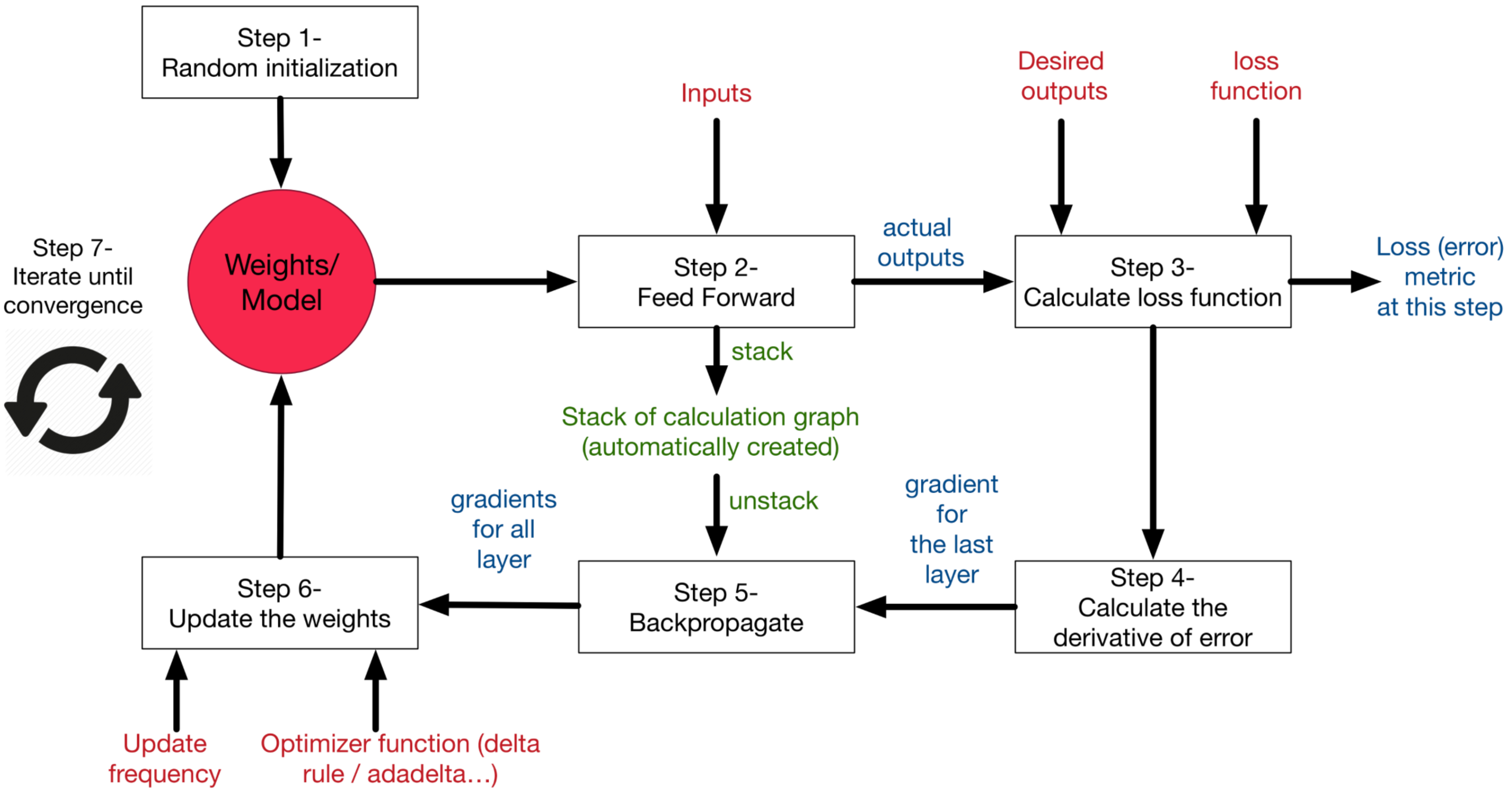
Optimizers



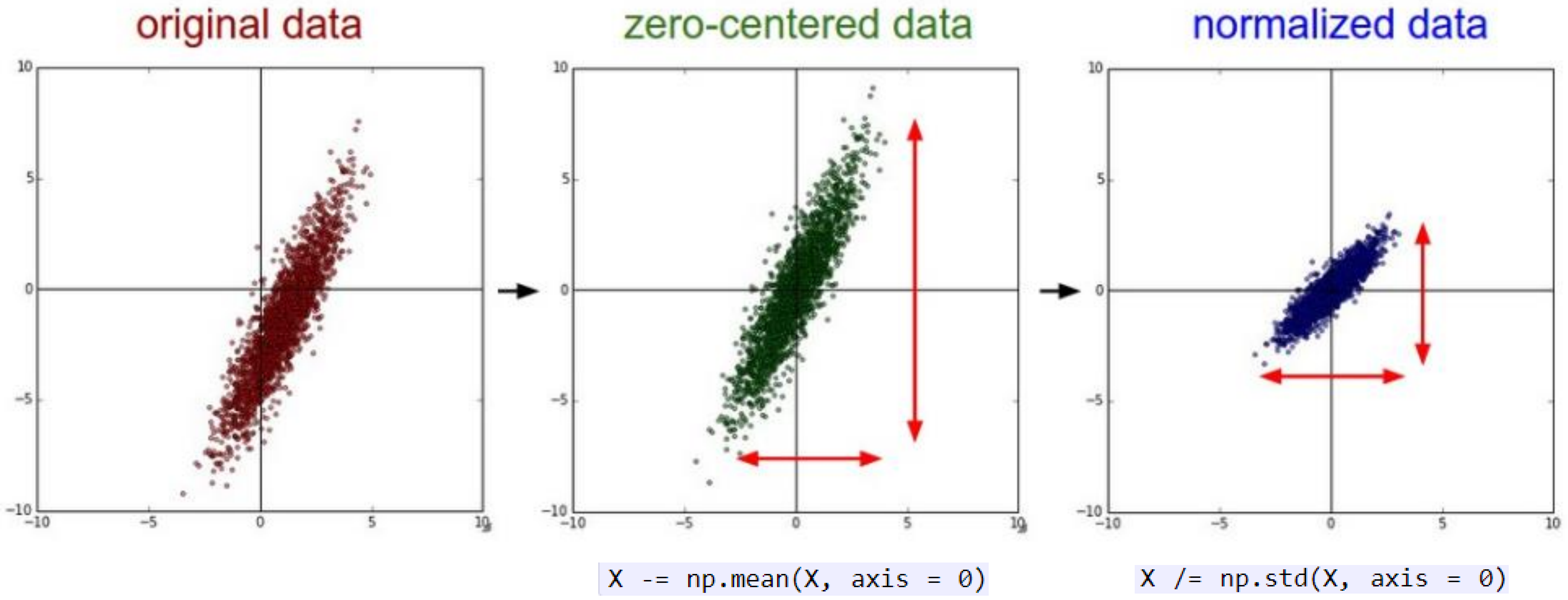


source: http://cs231n.github.io/neural-networks-3/#vis

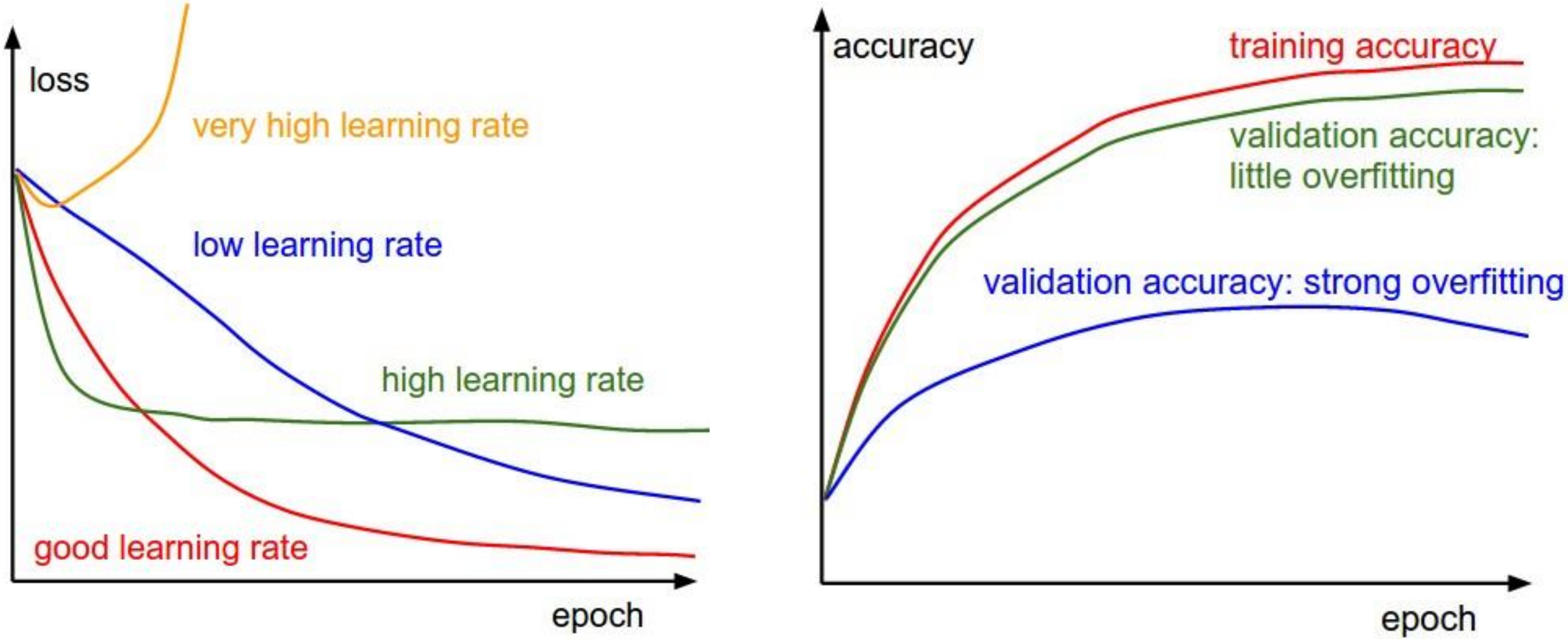
Summary


Step 1-
Random initialization
Inputs
Desired outputs
loss function
Step 7-
Iterate until convergence
Weights/
Model
Step 2-
Feed Forward
actual outputs
Step 3-
Calculate loss function
Loss (error) metric at this step
stack
Stack of calculation graph (automatically created)
unstack
gradients for all layer
gradient for the last layer
Step 6-
Update the weights
Step 5-
Backpropagate
Step 4-
Calculate the derivative of error
Update frequency
Optimizer function (delta rule / adadelta…)

Data preprocessing



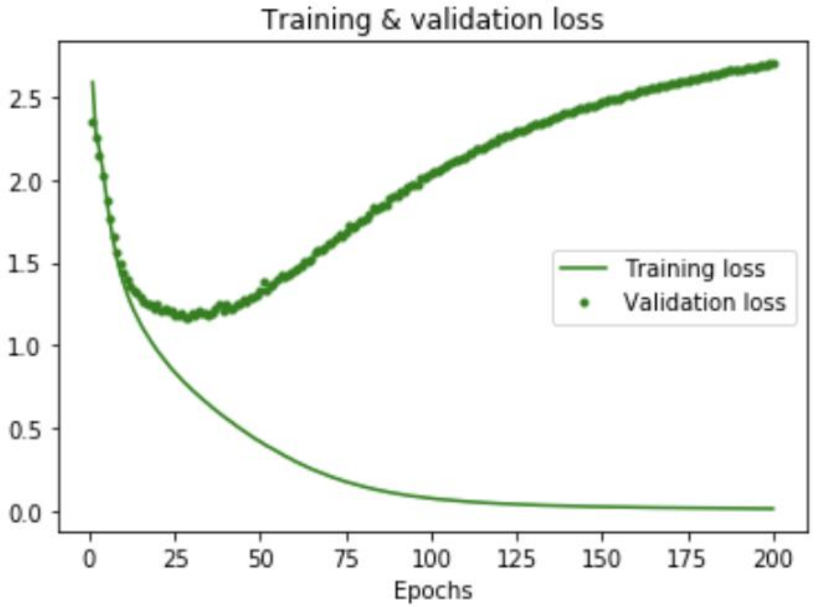
Hyperparameters

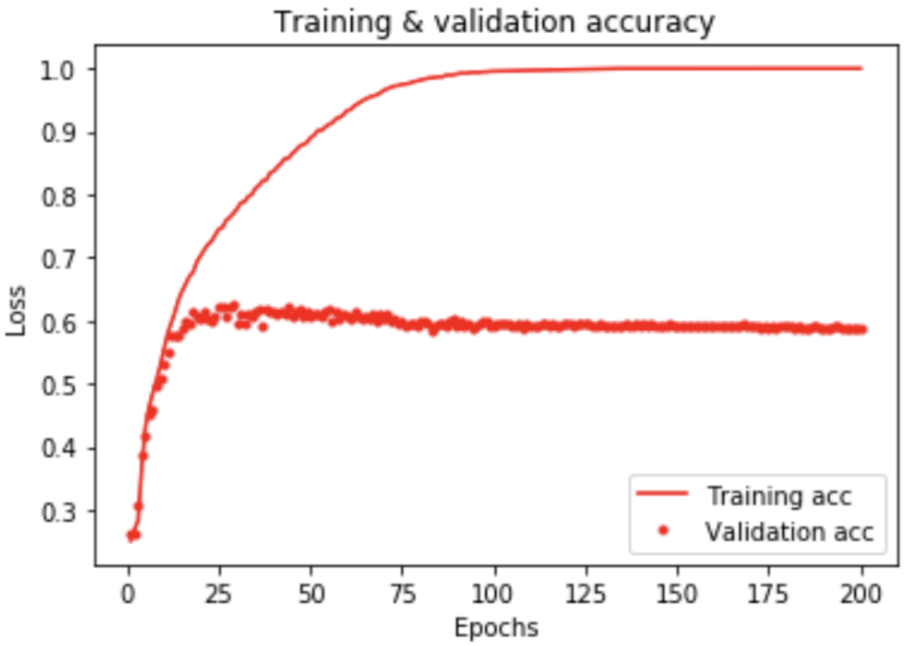


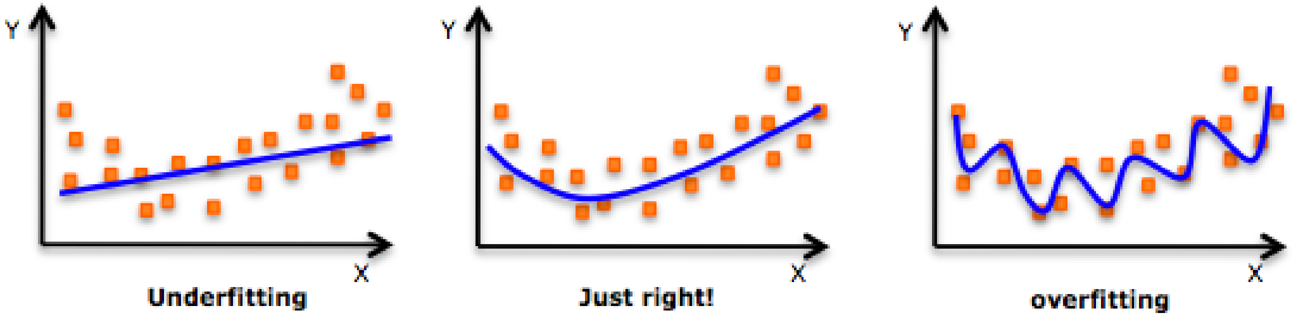
source: http://cs231n.github.io/neural-networks-3/#vis

Overfitting

How do you know your NN is overfitting?

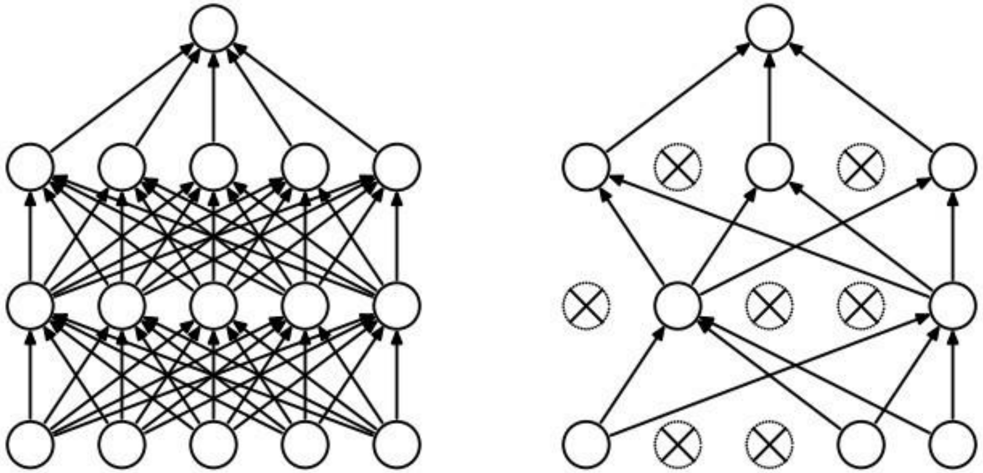






How to prevent your NN from overfitting?

- Early Stopping
- L1 regularization
- L2 regularization
- Adding Dropout Layer



(a) Standard Neural Net (b) After applying dropout.

Regularization

λ = regularization strength (hyperparameter)

$$L = \frac{1}{N}\sum_{i=1}^{N}\sum_{j\neq y_i}\max(0, f(x_i;W)_j - f(x_i;W)_{y_i} + 1) + \boxed{\lambda R(W)}$$

In common use:

L2 regularization $R(W) = \sum_k \sum_l W_{k,l}^2$

L1 regularization $R(W) = \sum_k \sum_l |W_{k,l}|$

How does it look like in code?

- Early Stopping
- L1 regularization
- L2 regularization
- Adding Dropout Layer

```
model= Sequential()
model.add(Dense(100,activation='relu',input_shape=(2000,)))
model.add(Dropout(.3))
model.add(Dense(50,activation='relu'))
model.add(Dropout(.3))
model.add(Dense(19, activation='softmax'))

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
drop_model = model.fit(train_final,
                    label_train_final,
                    epochs=75,
                    batch_size=32,
                    validation_data=(val, label_val))
```

```
model= Sequential()
model.add(Dense(100,activation='relu',kernel_regularizer=regularizers.l2(0.005), input_shape=(2000,)))
model.add(Dense(50,activation='relu',kernel_regularizer=regularizers.l2(0.005)))
model.add(Dense(19,activation='softmax'))
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
L2_reg = model.fit(train_final,
                    label_train_final,
                    epochs=75,
                    batch_size=32,
                    validation_data=(val, label_val))
```

Deep Learning Problems

- Deep learning requires a gigantic learning data set
- Most often it is not possible to transfer the obtained model
- The world is changing. Deep learning implies differently?
- ~~Who is lower? Mr. Marek or his son who is going to the 3rd grade in primary school this year?~~
- Problem in accuracy. We can't trust neural networks 100%

Let's code!

Google Colab(click)

Questions?