Bioinformatics carpentry utilizing Galaxy: An Introduction to Machine Learning

Maximilian Hillemanns

de.NBI Training – 15th April 2021

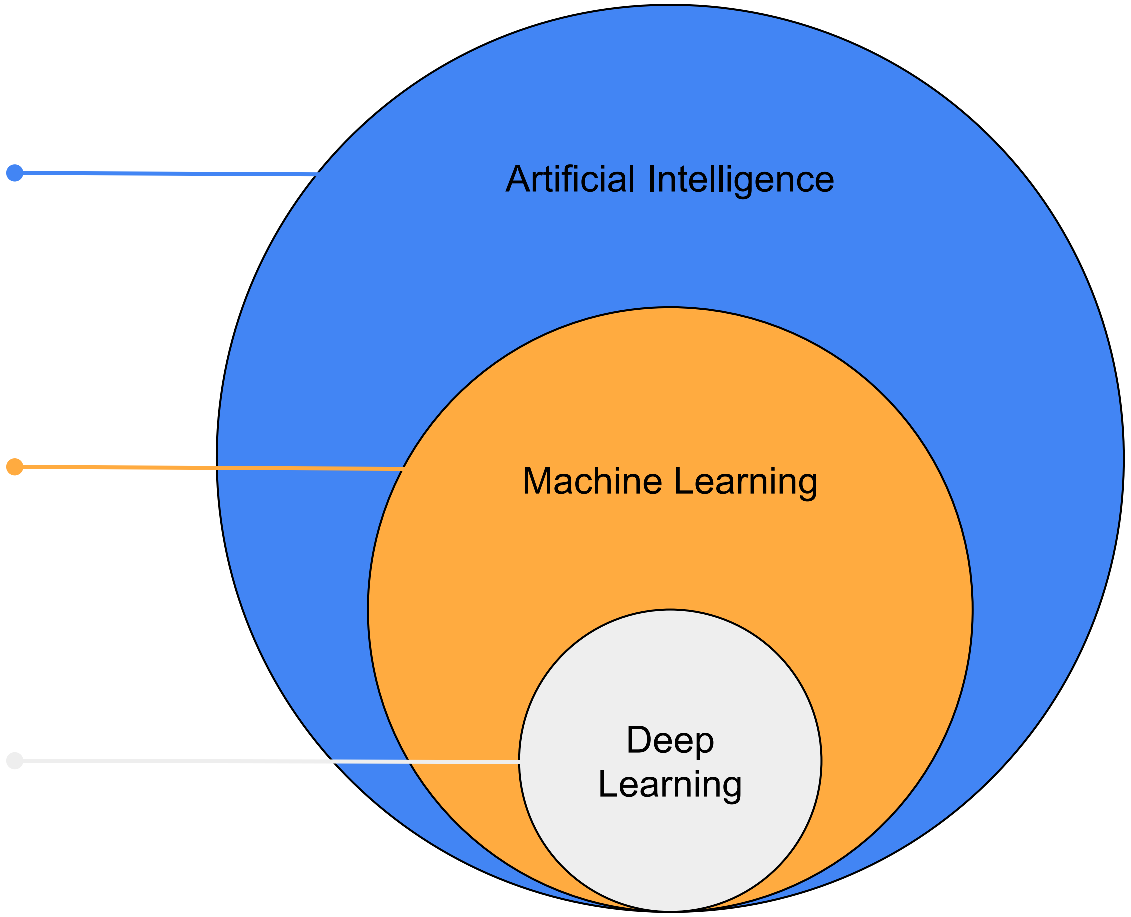
www.sbi.uni-rostock.de

What is Machine Learning?

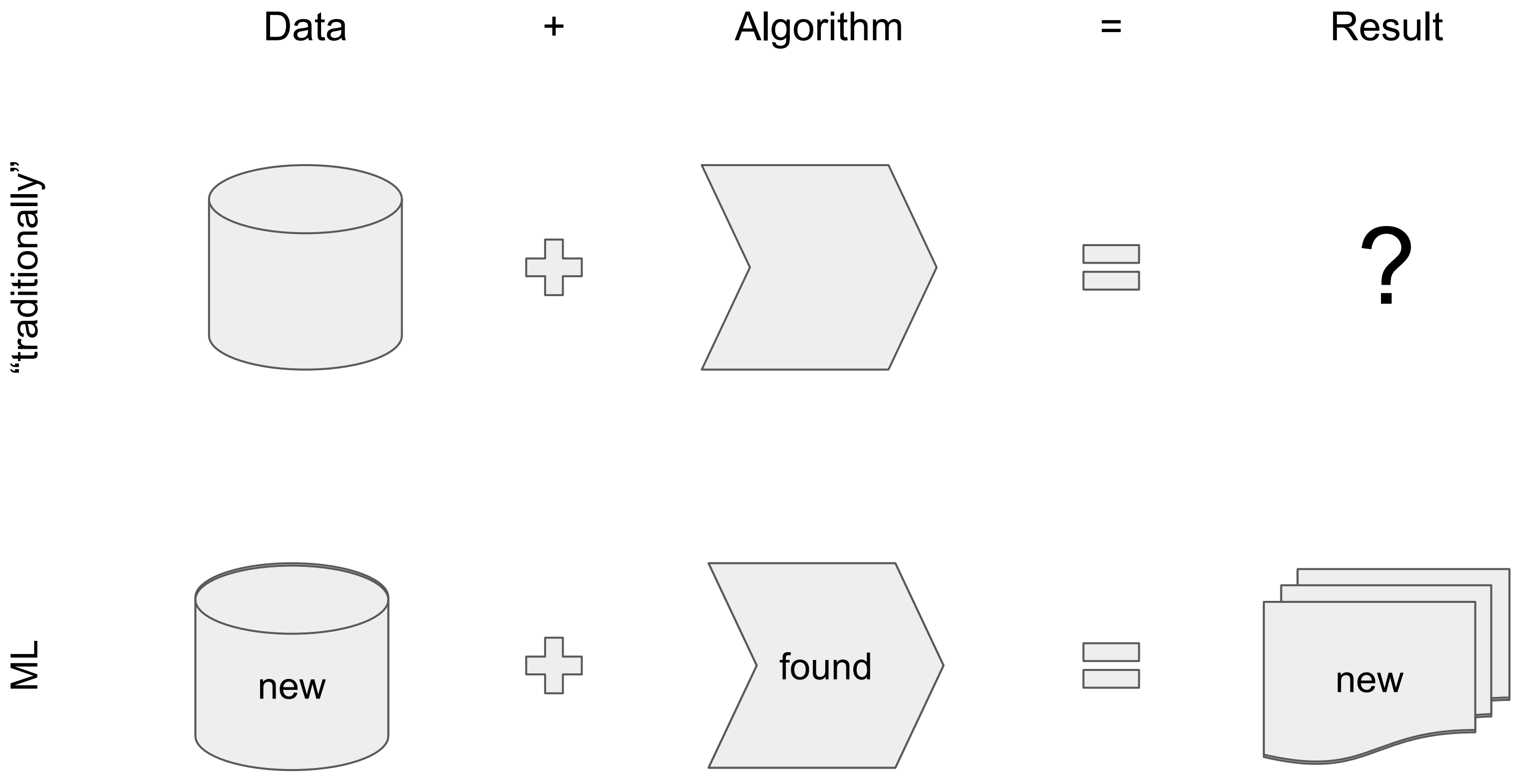
Techniques which enable machines to mimic human behaviour

Techniques which make machines learn by experience

Techniques which use Deep Neural Networks (DNNs)



What is Machine Learning?



ML and supervision

supervised:

- all data is labeled
- algorithms learns to predict output from input
- classification, regression

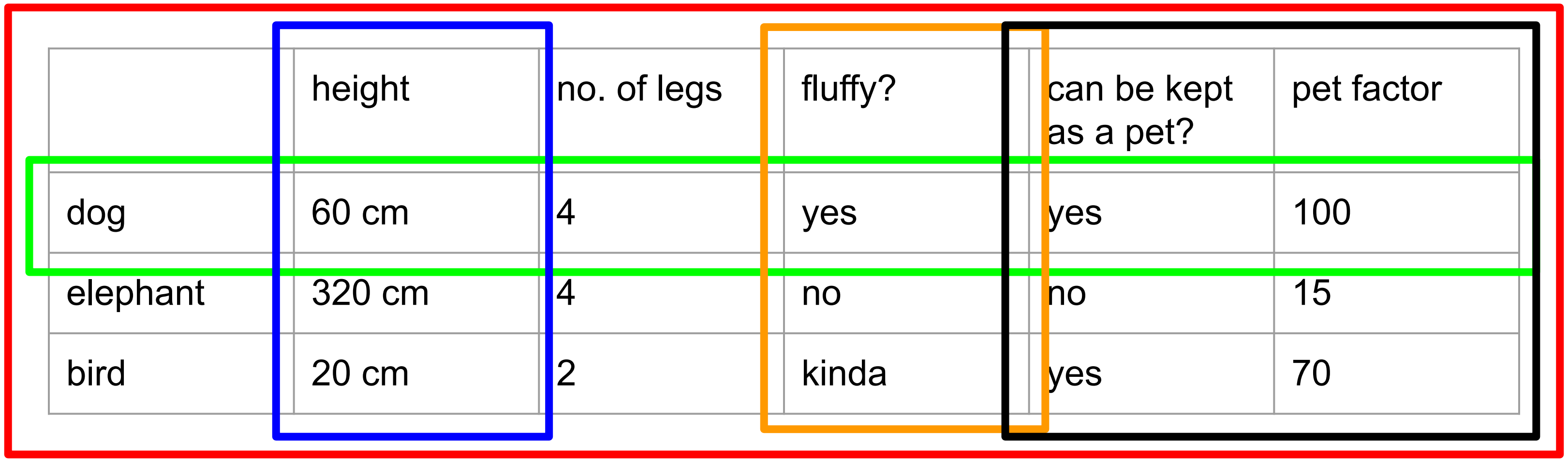
unsupervised:

- all data is unlabeled
- algorithms learns to represent structure in the input
- clustering

semi-supervised:

- some data is labeled
- real-life scenarios
- mixture of (un)supervised methods

ML terms

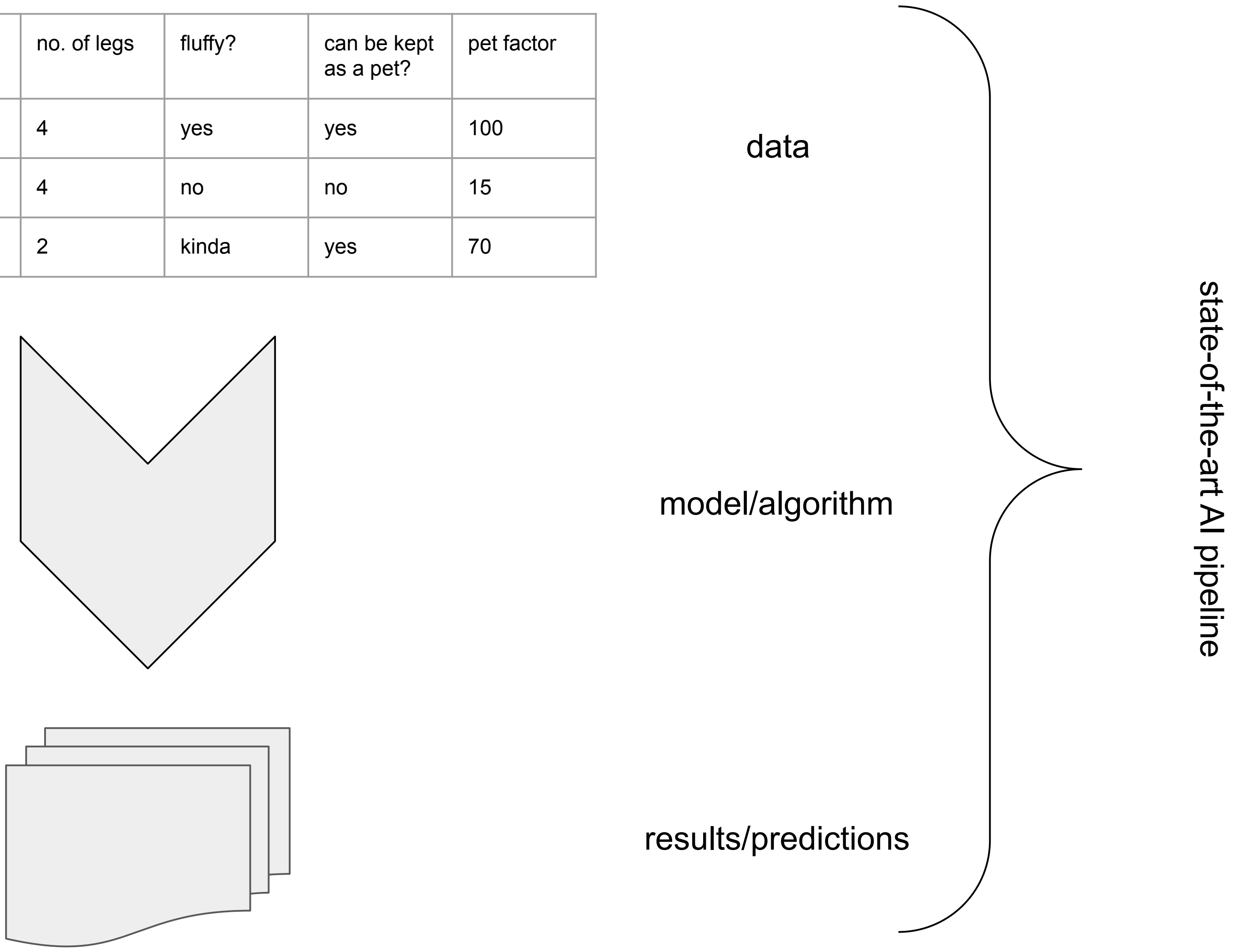


	height	no. of legs	fluffy?	can be kept as a pet?	pet factor
dog	60 cm	4	yes	yes	100
elephant	320 cm	4	no	no	15
bird	20 cm	2	kinda	yes	70

- data
- data point
- numerical feature
- categorical feature
- target value/label/ground truth (categorical for classification, numerical for regression)

ML terms

	height	no. of legs	fluffy?	can be kept as a pet?	pet factor
dog	60 cm	4	yes	yes	100
elephant	320 cm	4	no	no	15
bird	20 cm	2	kinda	yes	70



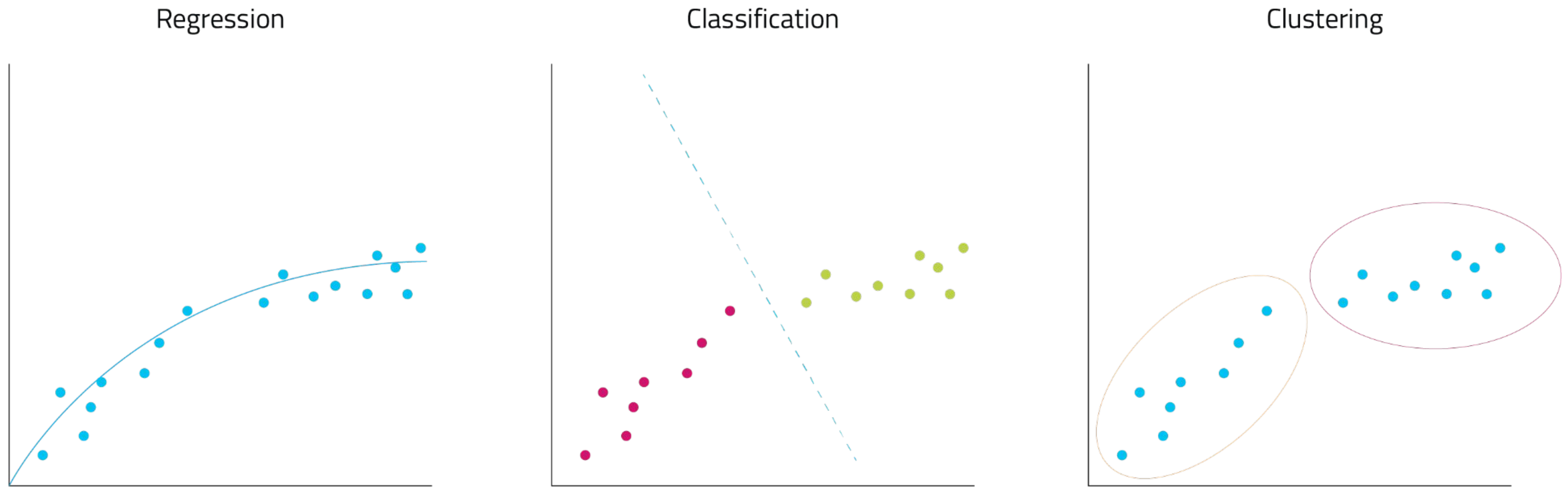
ML terms

- training data: data used to train and validate the model
- testing data: data used to test and evaluate the model
- loss (function): the error of our model, commonly defined as the difference between the ground truth and the predicted output
- optimizer: the way the loss is minimized
- (hyper)parameters/weights: the details of our model

supervised training principle:

- create a model (random/zero-weights/…)
- plug a training example into the model and create an prediction
- compare the prediction to the ground truth
- change the model according to this comparison
- repeat until a certain criteria is met (accuracy/number of iterations/…)

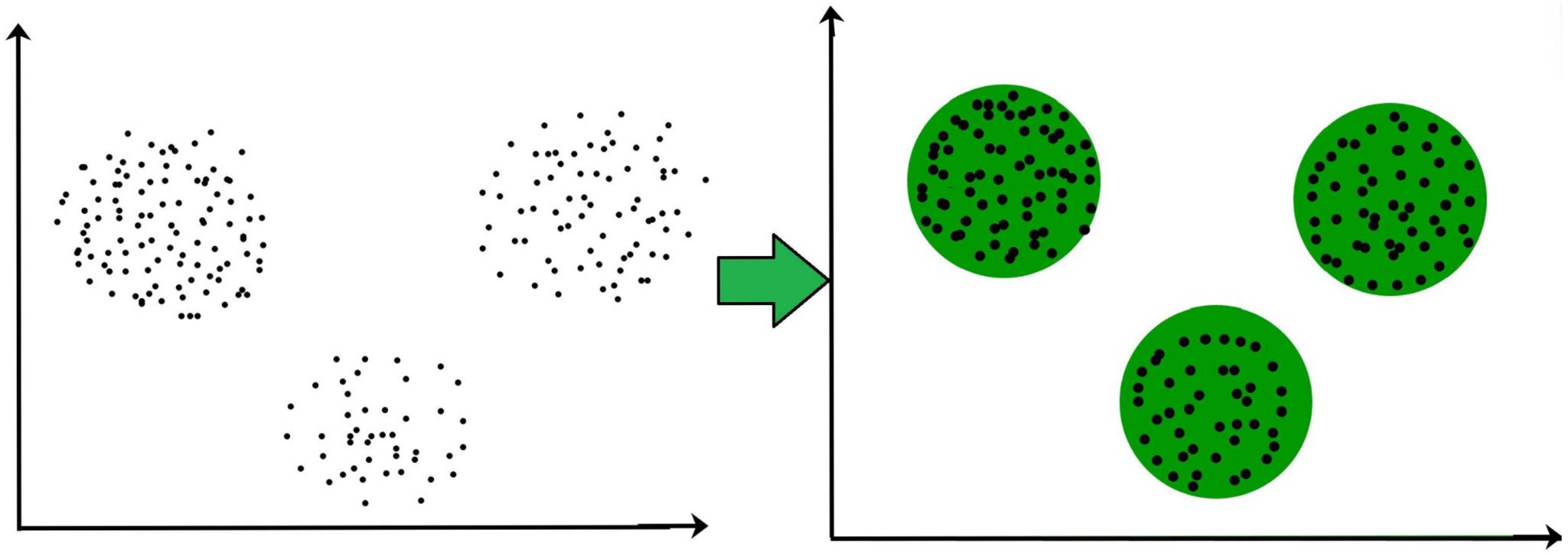
What is Machine Learning?



(+ Reinforcement Learning)

What is Clustering?

Clustering is the grouping of data points

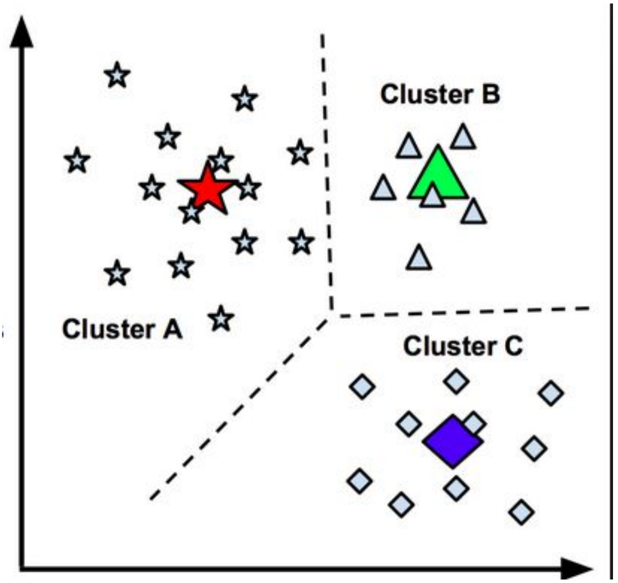


… to find patterns in the data.
… to make classifications easy.
… to see how the data is distributed.
...

https://tutorialspoint.dev/image/merge3cluster.jpg

k means clustering

1. Select *k*, the number of clusters and a distance metric, for example *euclidean*
2. Randomly initialize *k* cluster centers
   - completely random OR
   - on random data points
3. Calculate the distance between each data point and cluster center
4. Assign a data point to the nearest cluster center
5. Move the cluster centers to the mean of its assigned data points
6. repeat steps 3-5 until the clusters do not change anymore or another criterion is met (number of iterations, etc.)



DEMO:
https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

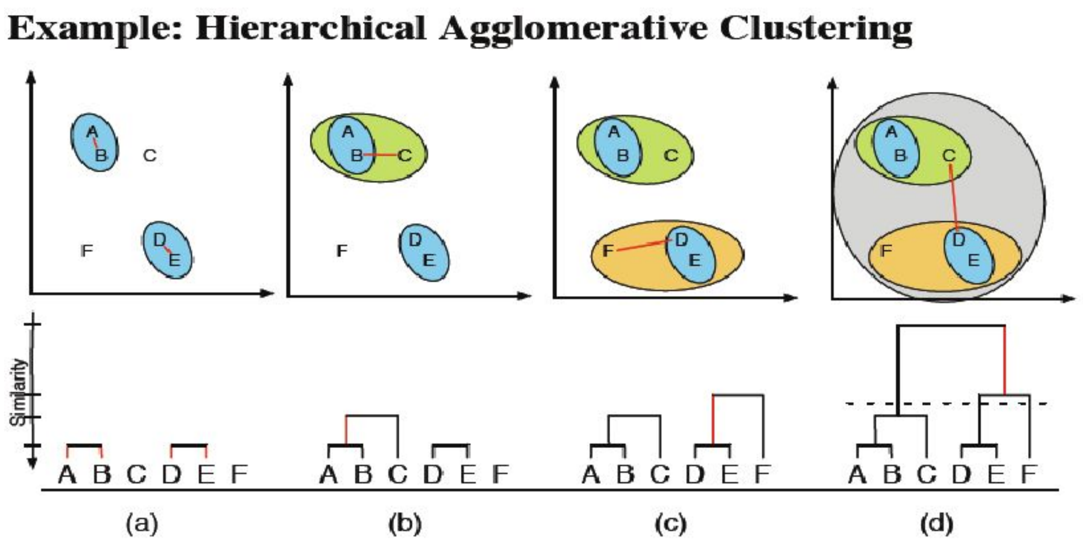
Advantages	**Disadvantages**
Fast	Selection of k
Easy to compute	not consistent

https://cdn-images-1.medium.com/max/1600/1*fG8u8nV7qR91wDyFDEEV-g.png

Hierarchical clustering

1. Start by treating each data point as its own cluster
2. Combine the two clusters with the smallest distance between them
   - i.e. average euclidean distance between data points in the first and data points in the second cluster
3. Repeat step 2 until only one cluster containing all the data points is left
4. Choose the number of clusters best fitting to your data

DEMO: https://live.yworks.com/demos/analysis/clustering/index.html



Advantages	Disadvantages
no prior knowledge of data set needed	slow
not sensitive to the used metric	
number of clusters can be arbitrarily chosen after computation	

https://www.researchgate.net/profile/Carsten-Walther/publication/273456906/figure/fig3/AS:294866065084419@1447312956501/Example-of-hierarchical-clustering-clusters-are-consecutively-merged-with-the-most.png

DBSCAN

Density **B**ased **S**patial **C**lustering of **A**pplications with **N**oise

1. Define a neighborhood ε and a number of points *minPoints*
2. Start by "visiting" a random data point
3. If there are at least minPoints in the ε neighborhood of this data point, it is marked as part of a cluster, otherwise as noise
    - it is marked as "visited" nonetheless
4. If a data point is part of a cluster, all data points in the ε neighborhood are also added to this cluster
5. Then the first data point that has just been added to the cluster is visited and steps 3 and 4 are repeated
6. Steps 3-5 are repeated until all data points in the cluster are determined
7. Then, a data point outside of this cluster is visited and the process is repeated until every point has been visited

DEMO: https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

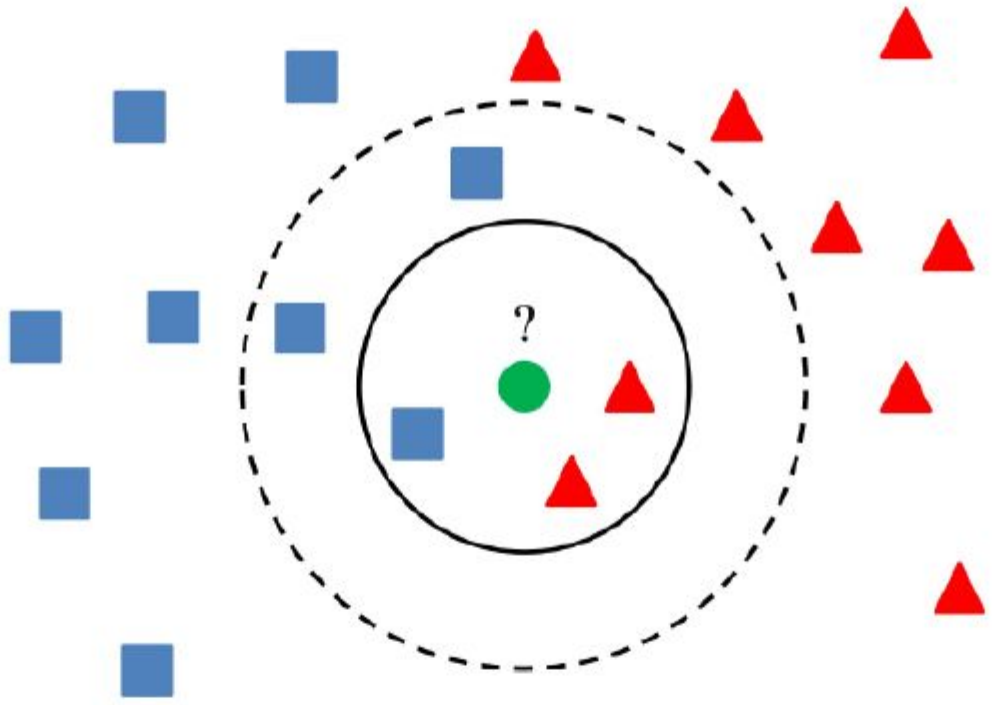
Advantages	**Disadvantages**
clusters of arbitrarily size and shape can be found	poor performance on clusters with varying density
incorporation of noise	algorithm parameters are not trivial to choose (esp. in higher dimensions)
does not require a pre-set number of clusters	

Clustering

DEMO

k Nearest Neighbors

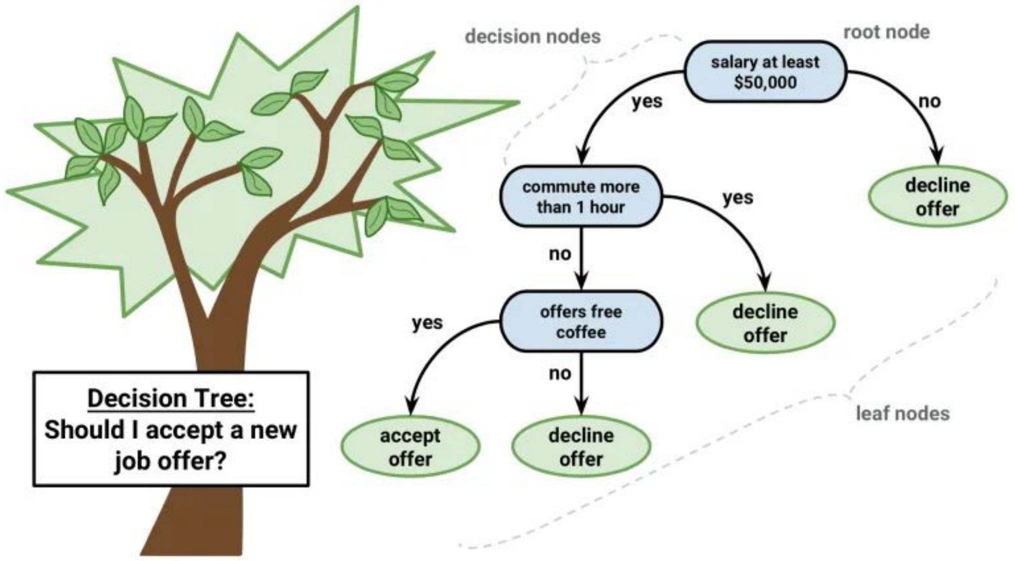
- the simplest classification algorithm not called “guessing”
- needs just one parameter: *k*, the number of nearest neighbors in distance metric of choice
- a point is automatically assigned to the class, which is more prominent in the neighborhood



https://www.researchgate.net/figure/K-nearest-neighbor-algorithm-illustration-The-green-circle-is-the-sample-which-is-to-be_fig14_267953942

Decision Tree

- relatively simple classifier
  - regression is possible, but not feasible
- works well for categorical data
- to obtain a prediction
  - start at the root node
  - follow the decision or internal nodes
  - end up at a leaf node

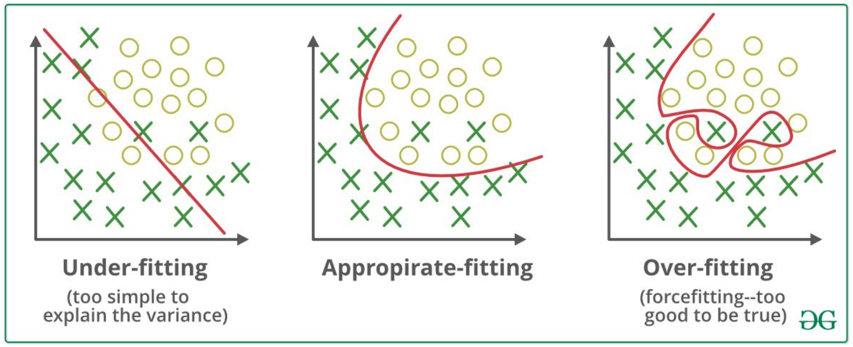


Finally, after years of search I found a real tree

https://dataaspirant.com/how-decision-tree-algorithm-works/

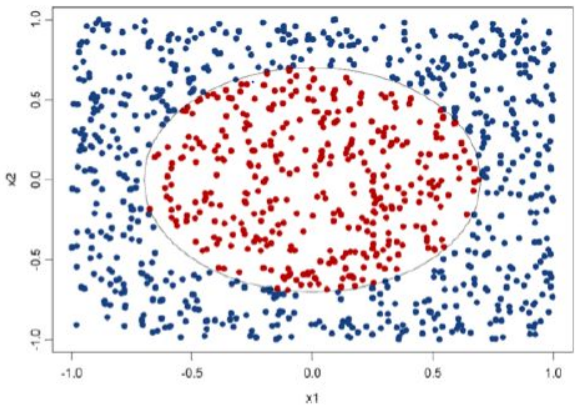
Decision Tree

- key to good decision trees: know the worth of the features
- the feature with the highest information gain goes into the root node and so on
- this does not change anything about the classification itself, but the tree is smaller/simpler
    - Occam’s Razor (prefer the simplest hypothesis that fits the data)
- what happens if there a tons of features? **OVERFITTING**
    - especially if some of them do not have a real benefit for the prediction
- big problem in machine learning
- the model loses its generalization ability because it practically memorizes the training data
- poor performance on new data

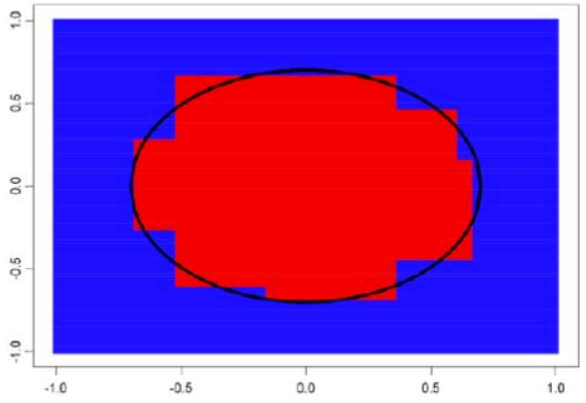

Under-fitting
(too simple to explain the variance)
Appropirate-fitting
Over-fitting
(forcefitting--too good to be true)
GG

Random Forest

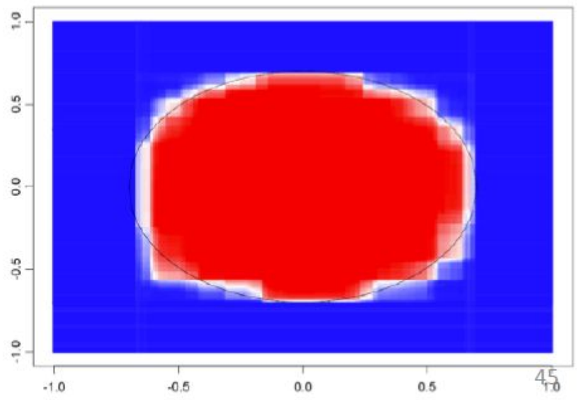
- consists of multiple decision trees
- principle:
  - grow several uncorrelated trees
  - take into account the classification of all trees (aggregation)
- but how to make the trees uncorrelated?
  - bootstrapping: create new subsamples of the data by leaving out data points/features and/or duplicating some
  - grow one decision tree on each subsample
  - nice side effect: out of bootstrap (OOB) data points can be used to determine the OOB error, which is a good indicator for the quality of a Random Forest
- this technique is called bagging (bootstrapping + aggregation)
- Random Forests can also be used for regression



1 tree



100 trees



Linear Regression

- very simple regression model
- implies a linear relationship between data points and target value
- surprisingly, often a good estimate

find a linear representation of the type

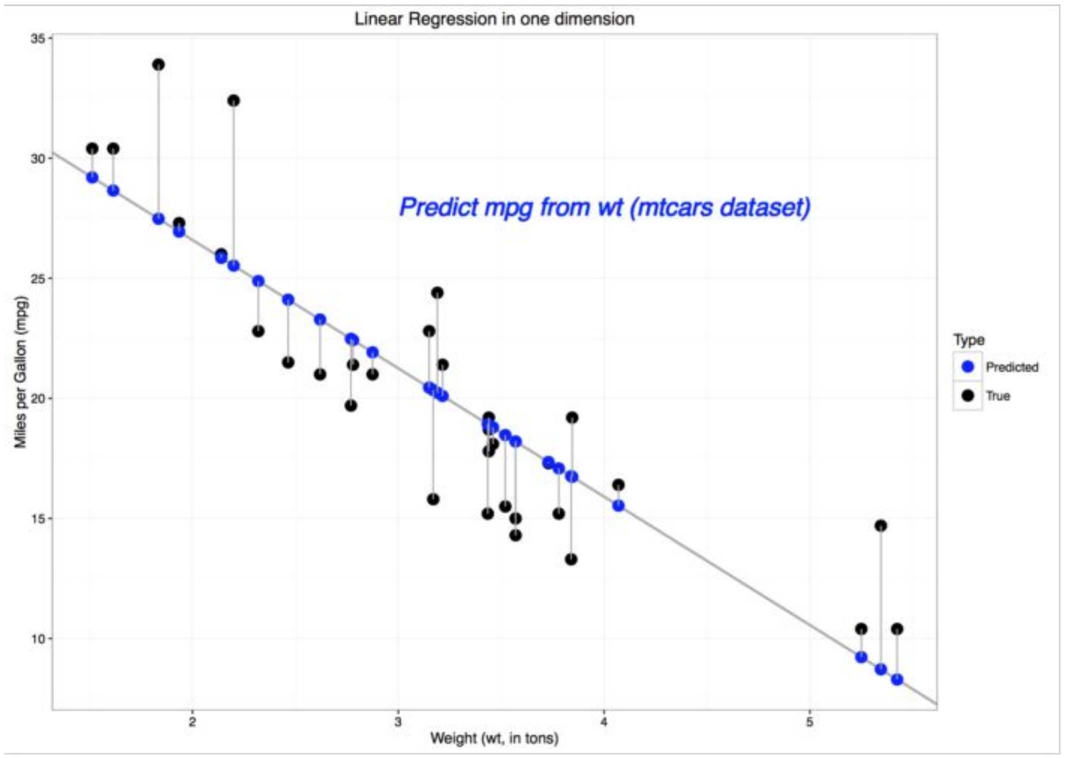
$$y = c_0 + c_1 x_1 + c_2 x_2 + \dots$$

or

$$y = Xc$$

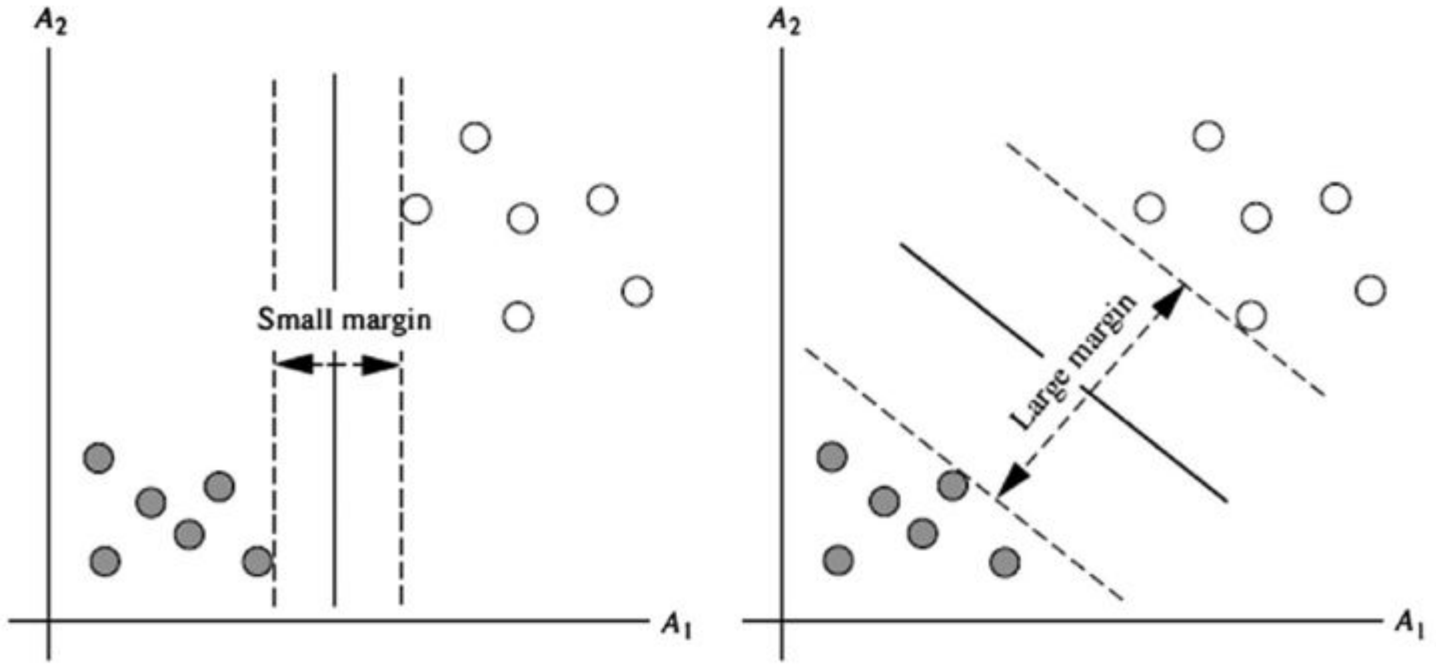
where *X* is our data points matrix, *y* is the output and *c* are the parameters we search that minimizes the function

$$f(c) = ||Xc-y||^2$$

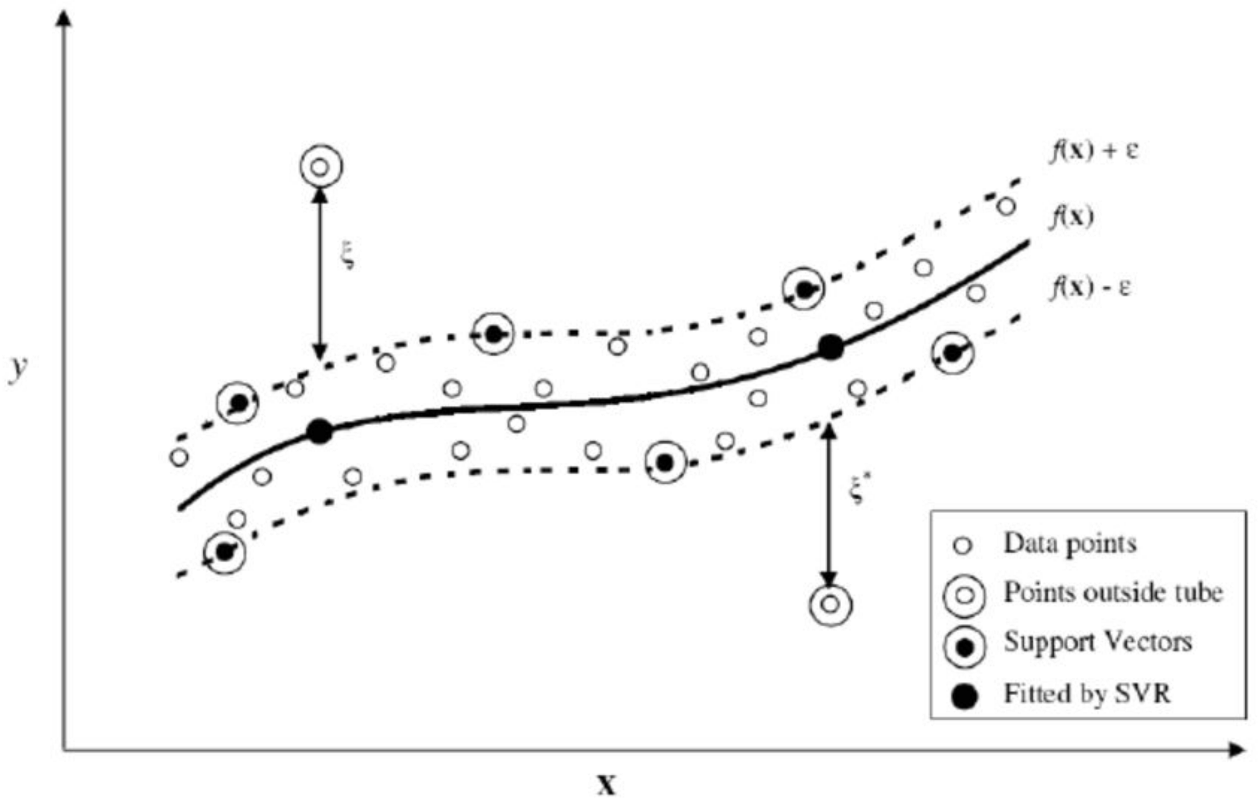


Support Vector Machines

- find a classification/regression curve, which maximizes the margin between the curve and the nearest (or farthest) data points
    - these data points are the so called support vector
- SVMs project the data into a higher dimension (kernel trick)
    - n data points are linearly separable in n-1 dimensions

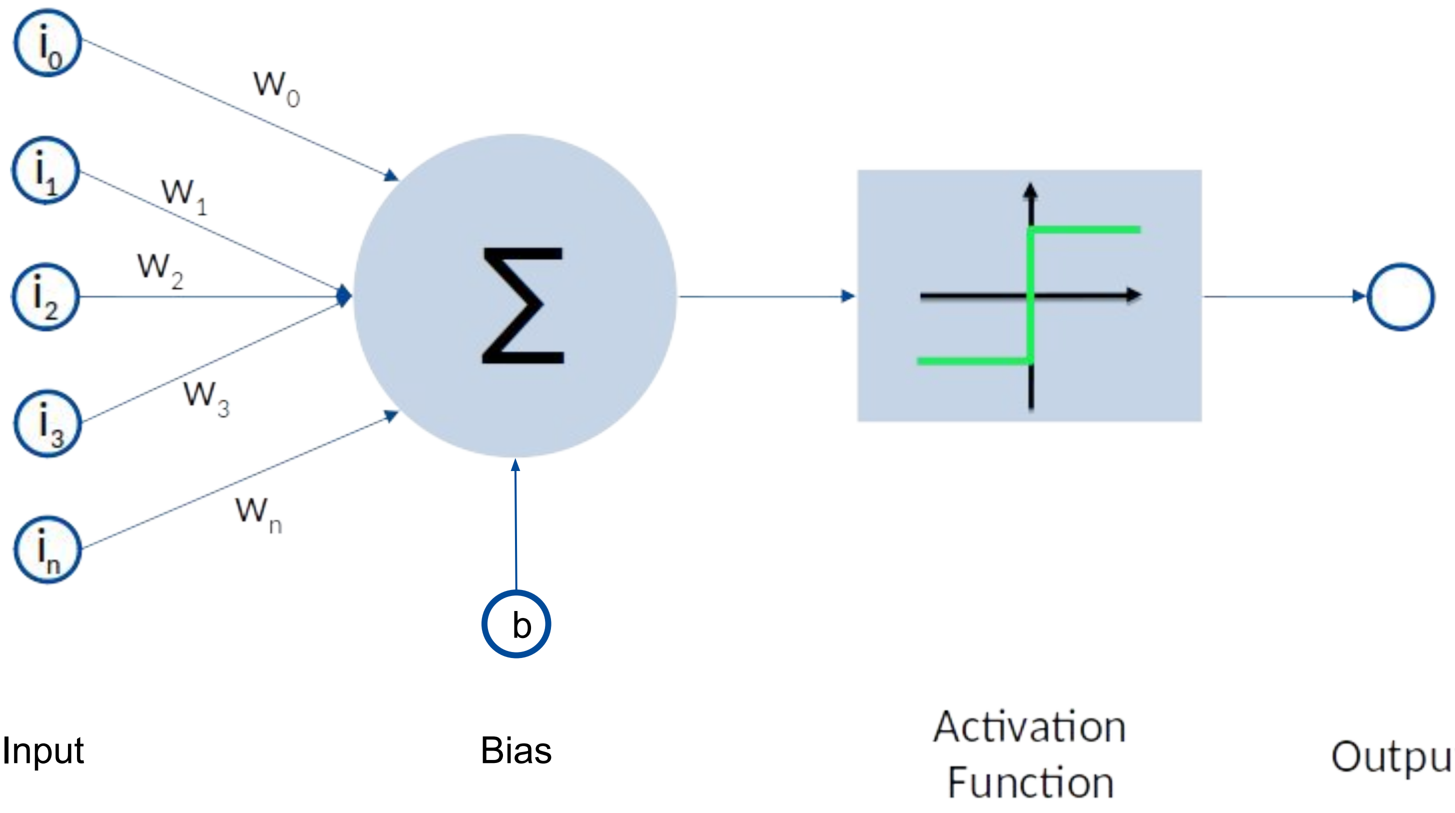


Support Vector Machines

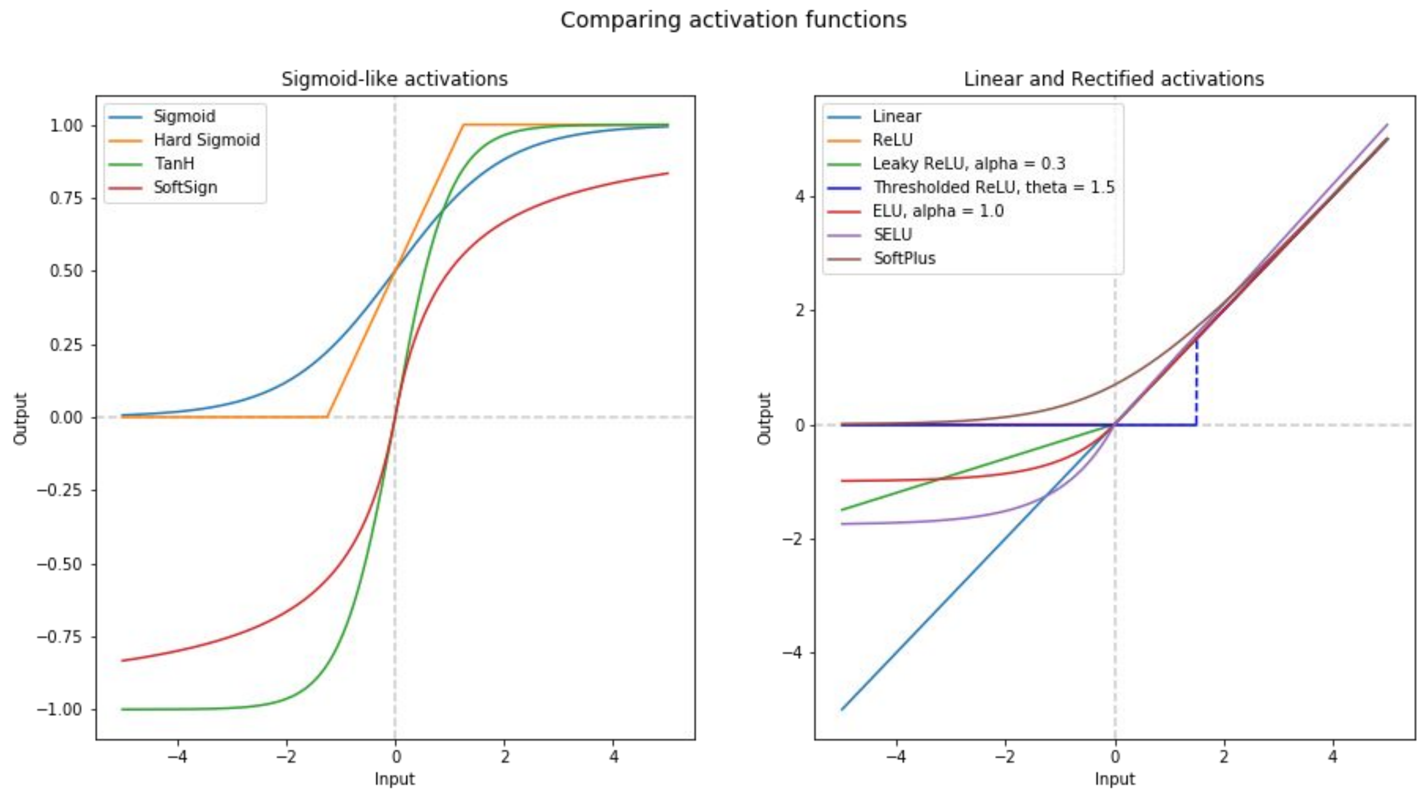


https://www.researchgate.net/figure/A-schematic-diagram-of-the-support-vector-regression-using-e-sensitive-loss-functio
n_fig1_26543790

Deep Learning - The artificial neuron



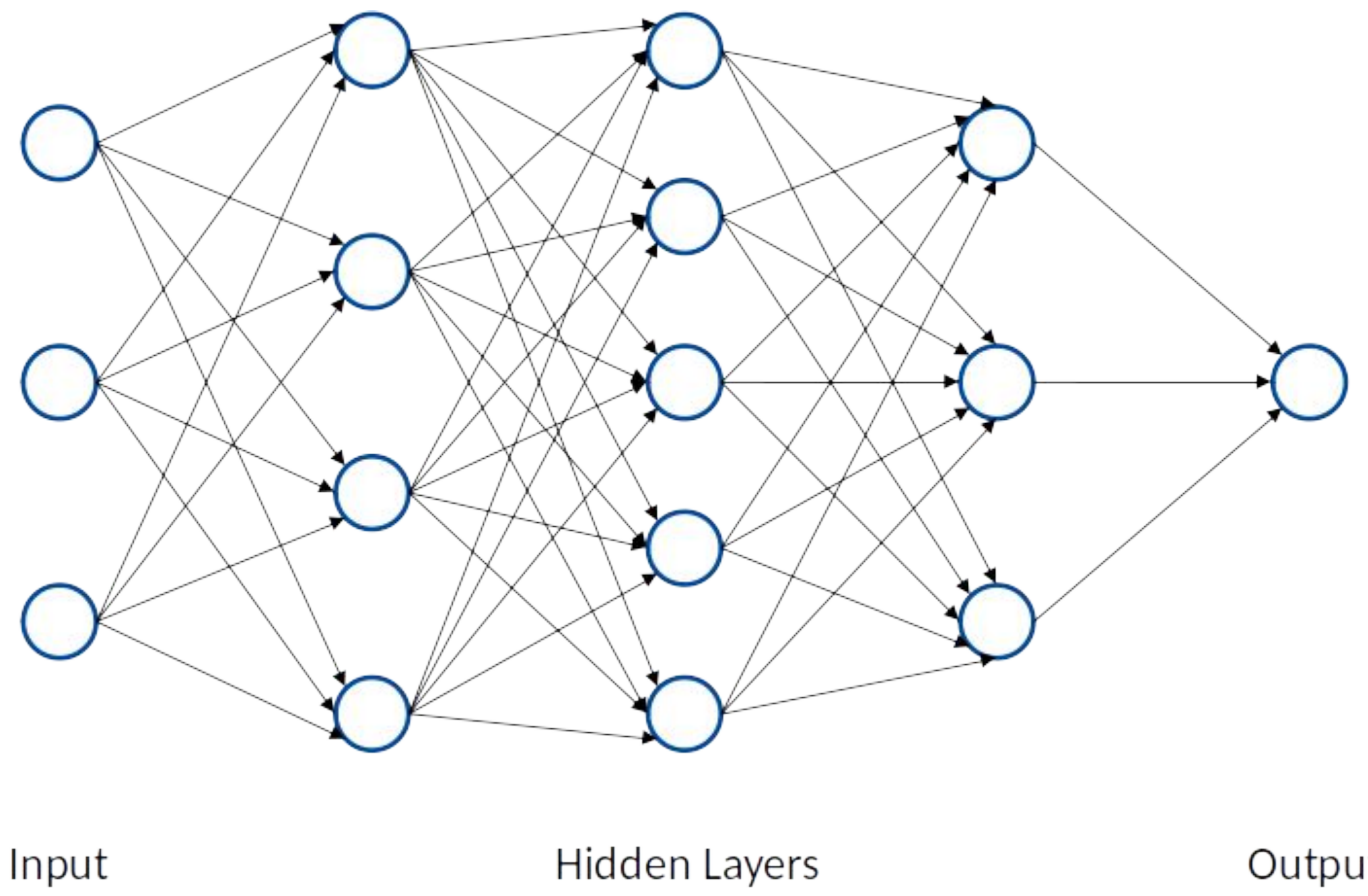
Activation Functions



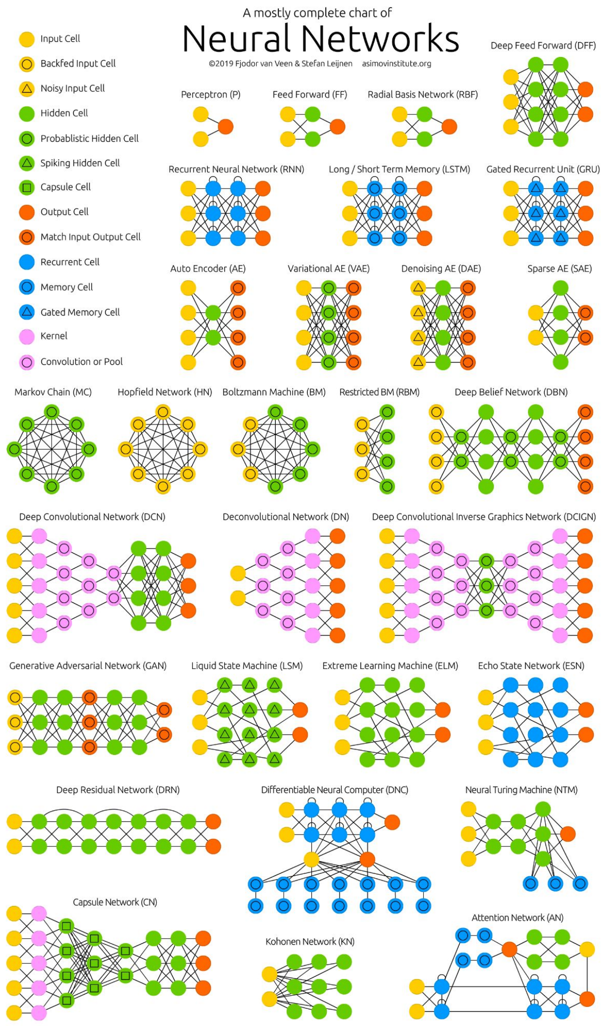
- each one has their own use case
  - **linear** for regression output
  - **ReLU** for image classification
  - **softmax** for classification output
  - ...

https://towardsdatascience.com/deep-study-of-a-not-very-deep-neural-network-part-2-activation-functions-fd9bd8d406fc

Neural networks

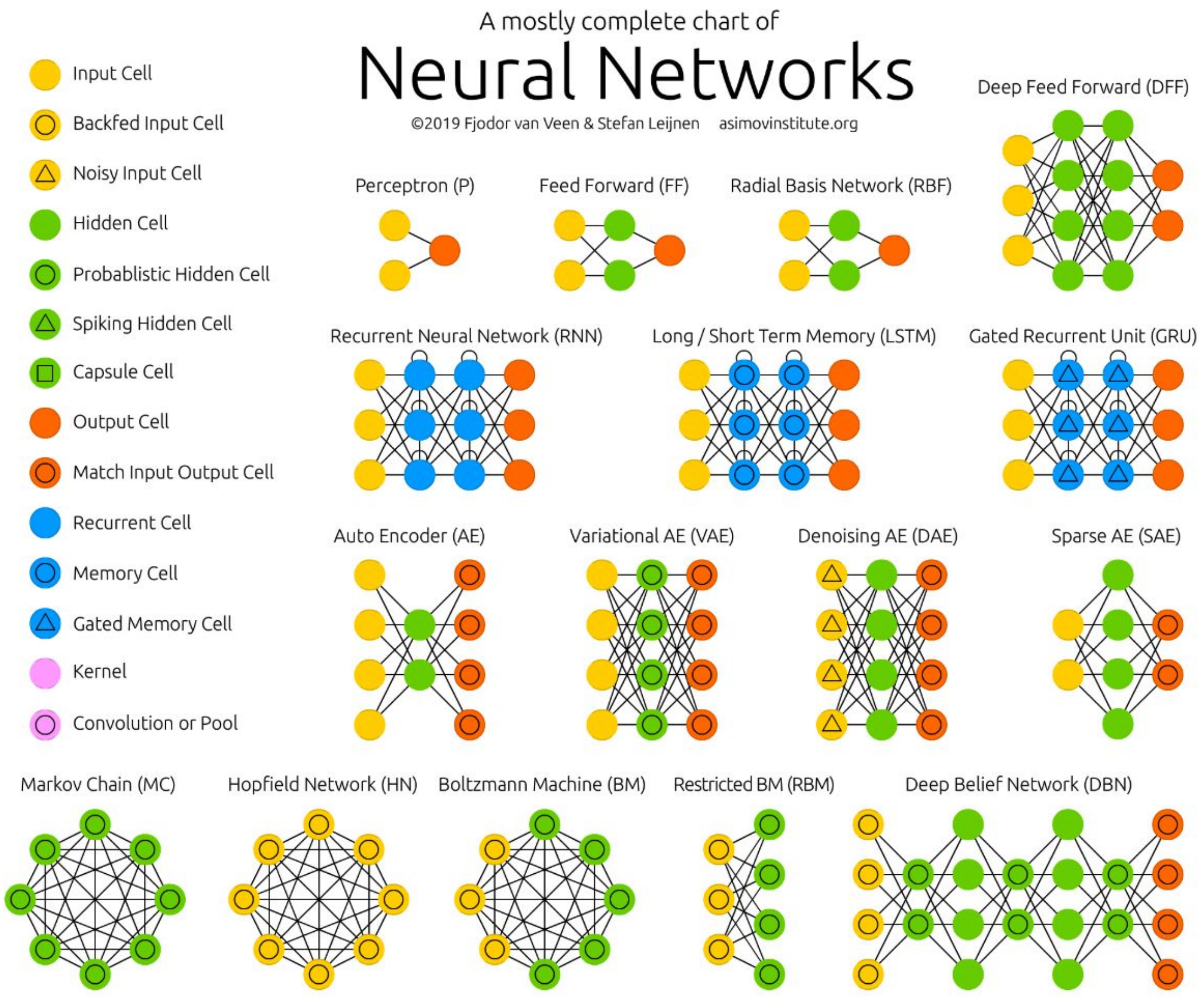


The Neural Network Zoo


A mostly complete chart of
Neural Networks
©2019 Fjodor van Veen & Stefan Leijnen asimovinstitute.org
Input Cell
Backfed Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Capsule Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Gated Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Differentiable Neural Computer (DNC)
Neural Turing Machine (NTM)
Capsule Network (CN)
Kohonen Network (KN)
Attention Network (AN)

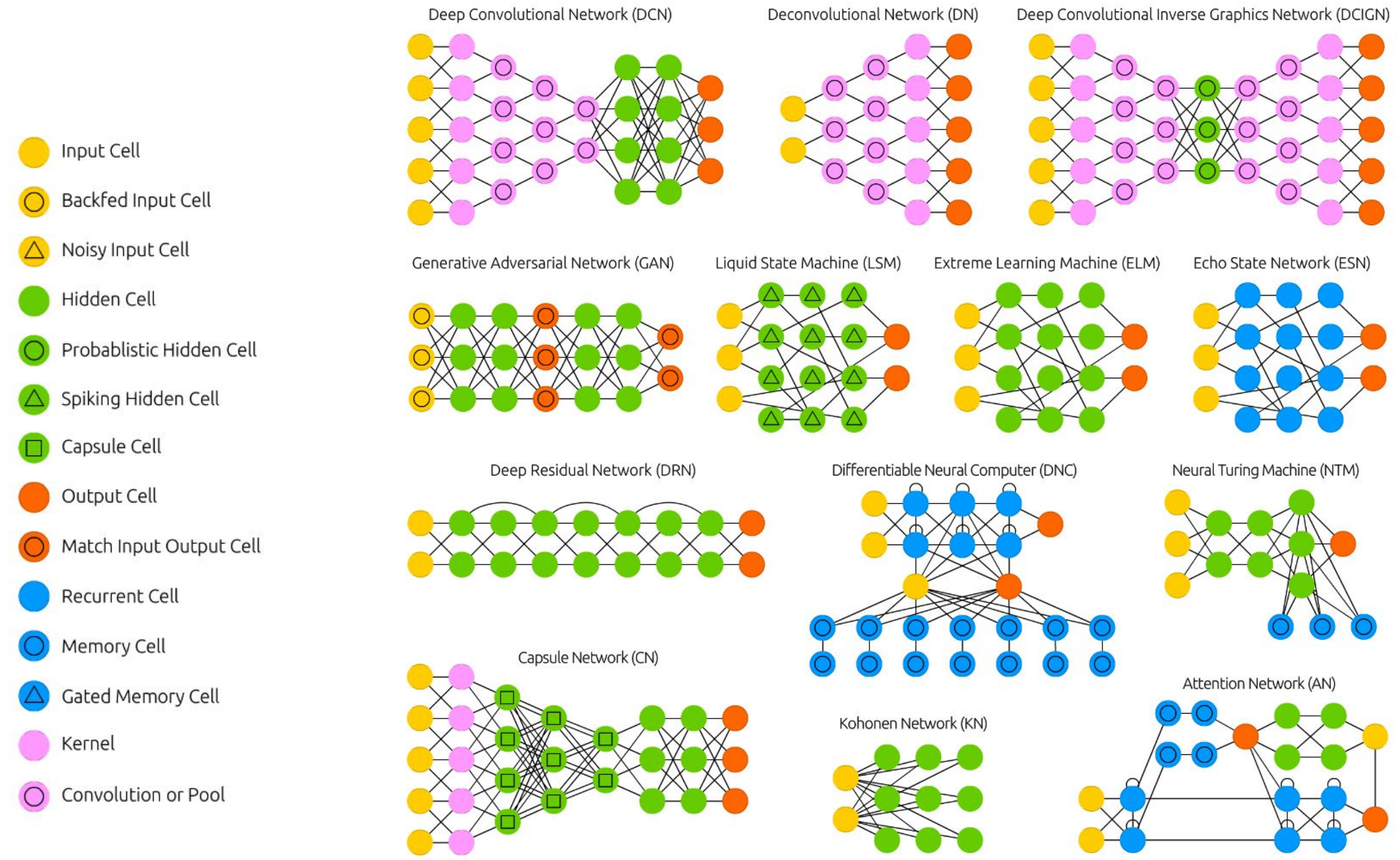
https://www.asimovinstitute.org/neural-network-zoo/

The Neural Network Zoo



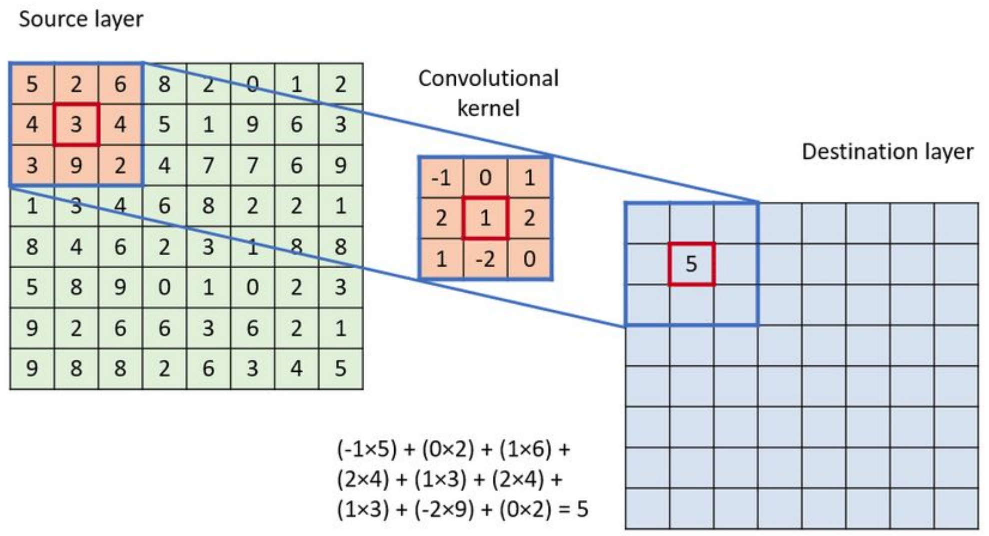
https://www.asimovinstitute.org/neural-network-zoo/

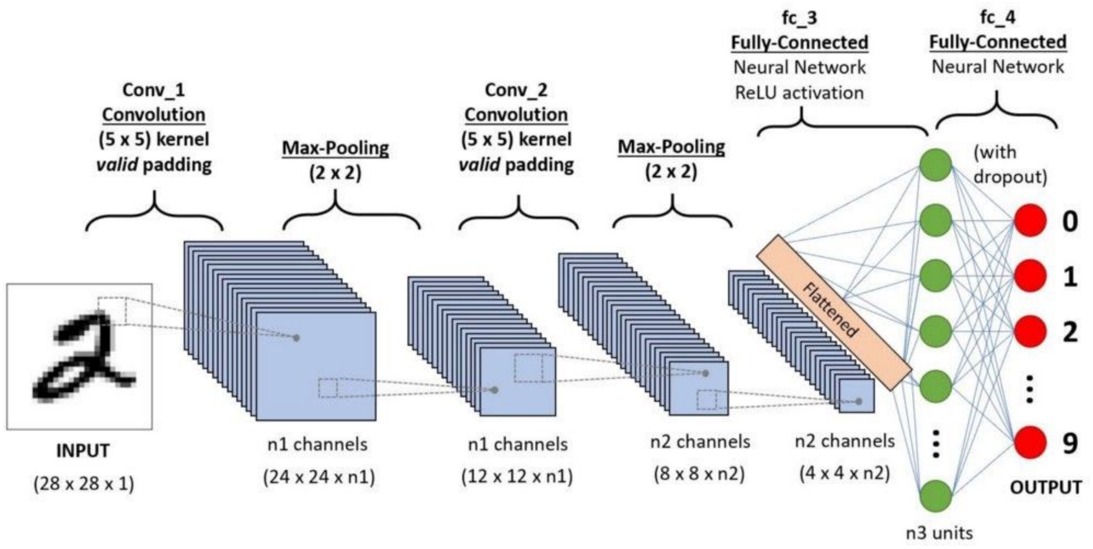
The Neural Network Zoo



https://www.asimovinstitute.org/neural-network-zoo/

Convolutional Neural Networks (CNNs)





https://www.researchgate.net/figure/Schematic-illustration-of-a-convolutional-operation-The-convolutional-kernel-shifts-over_fig2_332190148

https://towardsdatascience.com/a-simple-guide-to-convolutional-neural-networks-751789e7bd88

Machine Learning in Galaxy

DEMO

Should I use Machine Learning?

The definite answer: mhm, maybe, depends on...

… your data

… you, the user

… your machine

The User & the Machine

What you should bring to the table:

- (some) experience does not hurt
- good understanding of your data
- an open mind
  - for new algorithms
  - to see your data from a new perspective
- time

What your machine should have:

- enough computational power
  - especially RAM is important
- GPU (server) is the optimum

The data

- machine readable
  - tables in proper formatting
  - images as matrices
  - words as vectors
- uniform
  - no missing values (or at least have a plan on how to handle them)
  - numerical and categorical values don’t mix that well
- reproducible
  - your algorithm will probably be very narrow
  - new data has to look exactly like the training data
- decent size
  - rule of thumb: the more complex the data and the algorithm, the more data is needed
  - also: enough data to justify a ML approach
- good representation
  - make sure you have enough examples of each class (oversampling or data augmentation if needed)

How to do Machine Learning

- check your data
    - benefit of ML (new insights, unbiased decisions, …)
    - is it suitable for ML?
- make it suitable for ML
- identify the problem you want to tackle with ML
- choose a model
    - make an educated guess
    - when in doubt, choose the simpler model
- train your algorithm
- evaluate your model
- if it performs well on the first try
    - it never does
- if not
    - fine tune the hyperparameters of your model or choose another model
    - repeat
- if you went through some trial and error and the model looks good
    - deploy on some new data
    - add AI researcher on your LinkedIn profile
- make your code (+ data) accessible

I hope you enjoyed the training, now it's time to test it out yourself :)

The hands-on session is available at github

Any questions?

E-mail me: maximilian.hillemanns@uni-rostock.de

Universität Rostock

Traditio et Innovatio

SYSTEMS BIOLOGY BIOINFORMATICS ROSTOCK

Galaxy Training Network

GERMAN NETWORK FOR BIOINFORMATICS INFRASTRUCTURE