Machine learning and deep learning in evolutionary genetics

Flora Jay, CNRS, LISN
flora.jay@lri.fr @florajay_

+ credits for some slides and tutorials: J Cury, T Sanchez, A Quelin

EvoGenomics.AI

www.evogenomics.ai (sign up for seminar mailing list)

Outline

Part 1

I. Machine Learning: basic concepts and terminology

II. What’s a deep neural network (DNN) ?

Part 2
Deep Learning for population genetics

Opening on applications of unsupervized deep learning to popgen

Hands-on: building/training/re-using ML and DL models with application to population genetics (demography/selection)

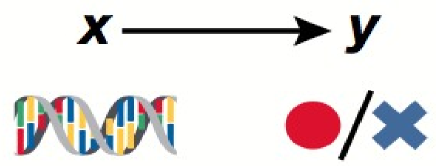
ML: scikit-learn

DL: dnadna https://mlgenetics.gitlab.io/dnadna/

What's machine learning ?

A typical example

TASK:
predict y from x



Angermueller et al Mol Syst Biol. (2016) 12: 878

Data?

- Learning something from **data**
  data = multidimensional object with e.g lots of samples (rows) and lots of variables/predictors/factors/features/markers ...
  (one vector/one matrix/several matrix per sample)

	loc1	loc2	loc3	...
ind1	A/A	C/C	C/G	
ind2	T/A	C/C	G/G	
...				

	Age	Gender	Work	Salary
ind1	55	F	baker	35k
ind2	43	M	..	..
...				

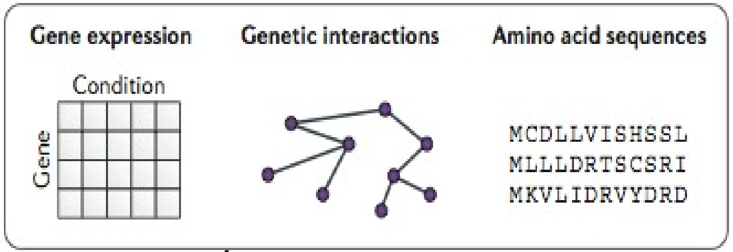
Quantitative and qualitative variables

	loc1	loc2	...	Sport activity	Hours of free time	...	Disease X ?
ind1	A/A	C/C					
ind2	T/A	C/C					
...							

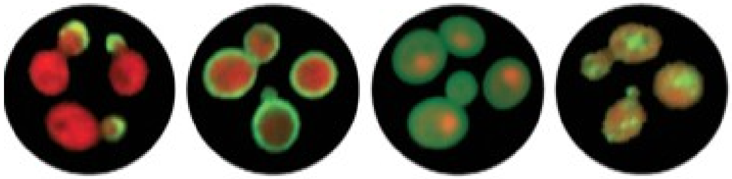
multidimensional and heterogeneous data

Data?

- Learning something from **data**
  data = multidimensional object with e.g lots of samples (rows) and lots of variables/ predictors/factors/features/markers … (one vector/one matrix/several matrix per sample)

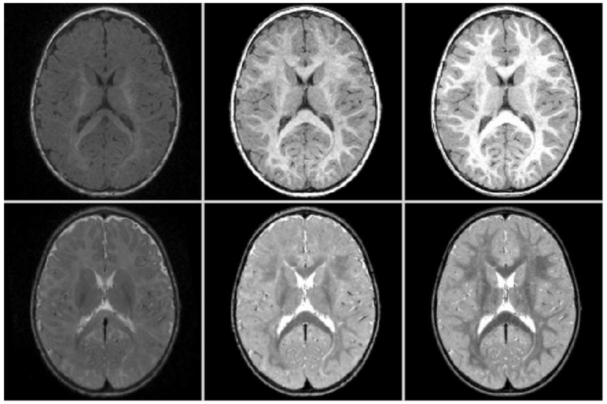


heterogeneous



Images with colors - micrographs of yeast cells expressing GFP-tagged proteins

MRI slices ~3D



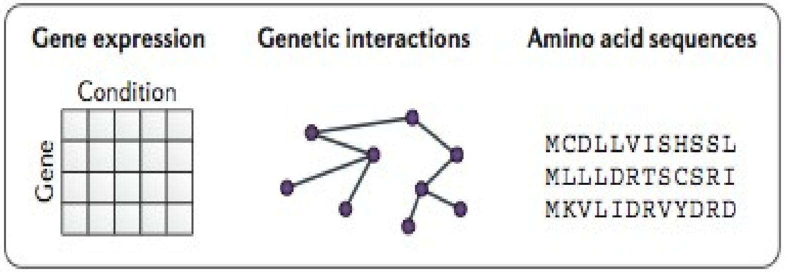
temporal

Labels and learning task

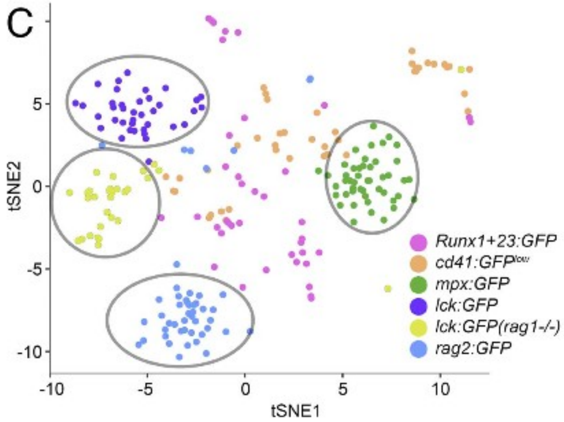
- **Data with or without label**
- **Label: a target class or a target value observed for each sample**
  Data are not always labeled.
  They can also have multiclass labels
  ex : pic of dog/person/car..., price of house, level of cholesterol
- **Task/objective ?**

Learning task?

- Data with or without label
- Label: a target class or a target value observed for each sample
  Data are not always labeled. They can also have multiclass labels
  ex : pic of dog/person/car..., price of house, level of cholesterol
- **Task/objective ?**

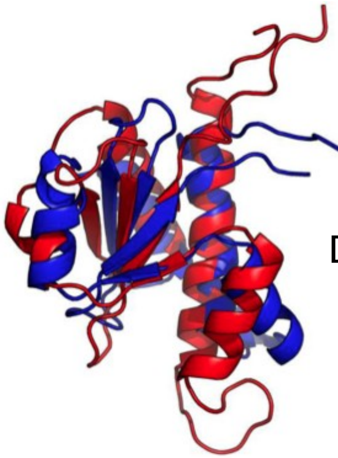


Task = predicting gene function labels



Task = identifying groups (clusters) of eg single-cell (T cells, NK cells ,..) with similar pattern of gene expression

Tang et al JEM 2017



Task = predicting protein 3D structure/contact map from DNA sequences and secondary structure, ... (blue=truth, red=pred)

Unsupervised / Supervised Tasks

- Learning something from **data**
- Either **unsupervised** (no labels) or **supervised** (discrete or continuous labels)

Unsupervised learning

- **Unsupervised = discovering patterns in data without prior knowledge of labels**
  You do NOT have labels, or you do NOT use them

  - ?
  -
  -
  -

Unsupervised learning

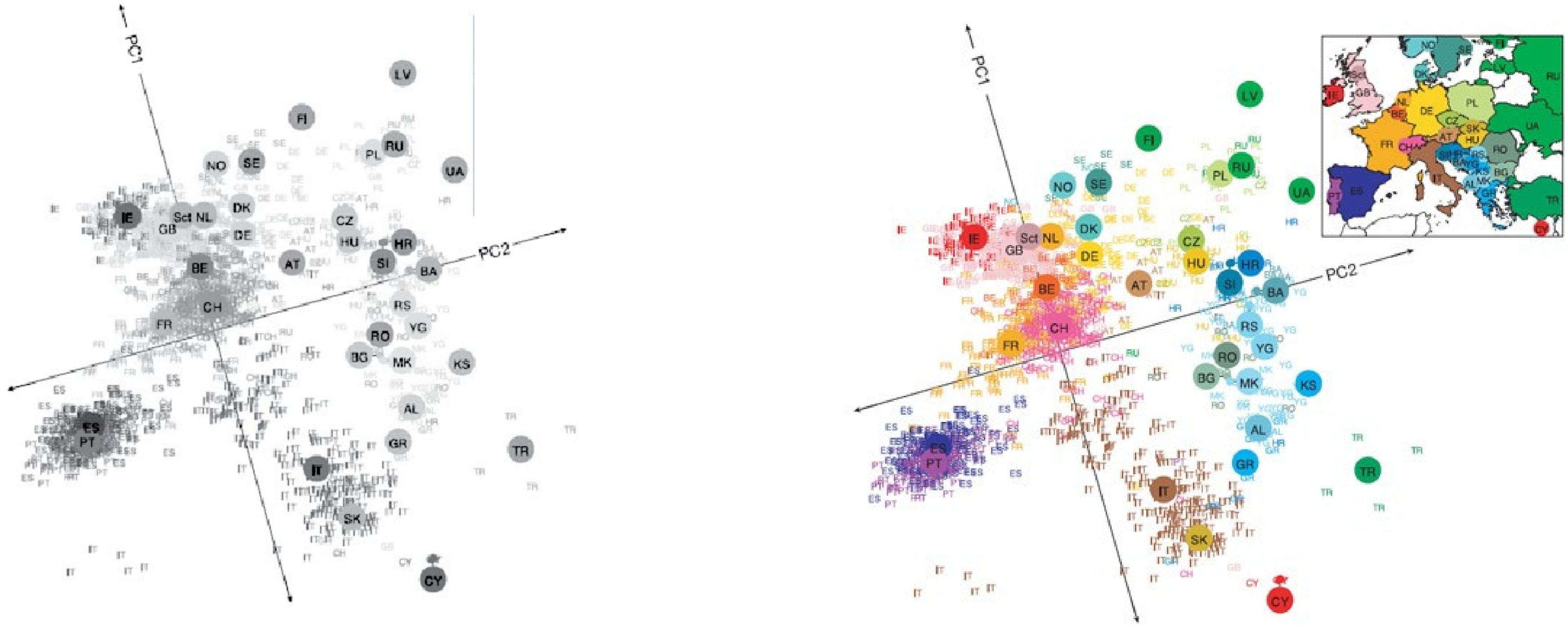
Unsupervised = discovering patterns in data without prior knowledge of labels

You do NOT have labels, or you do NOT use them

- Dimension reduction methods, e.g. PCA, Matrix factorization
- Clustering algorithms, e.g. K-means, hierarchical clustering, …
- Outlier detection (can be then used for filtering, ..)
- ...

Unsupervised learning

- Dimension reduction methods, e.g. PCA, Matrix factorization



PCA to reduce high dimensional genotype data for human populations

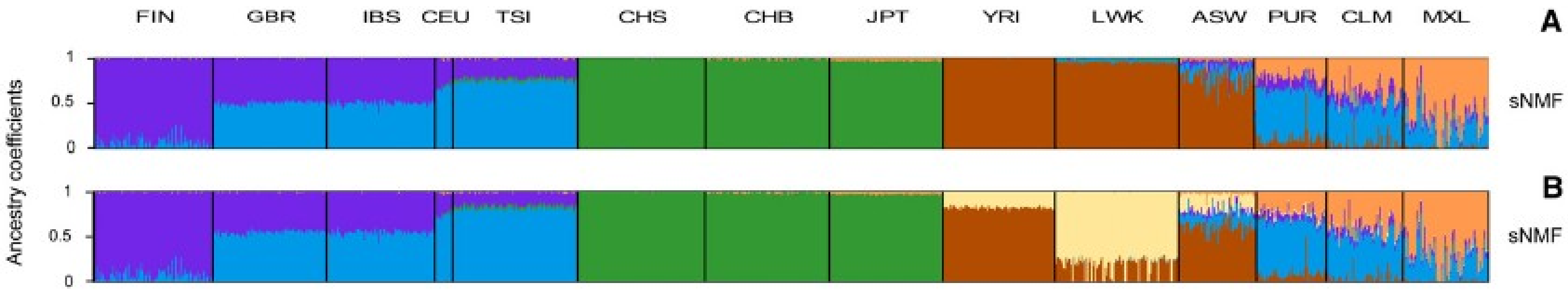
The 1st axis (ie the linear combination of markers) explains the largest part of the variance among samples. The 2nd axis explains the largest part of the remaining variance, and so on..

Novembre et al 2008

Unsupervised learning

- **Clustering** algorithms, e.g. K-means, hierarchical clustering, Non Negative Matrix Factorization…

Clustering into K unpredefined clusters:



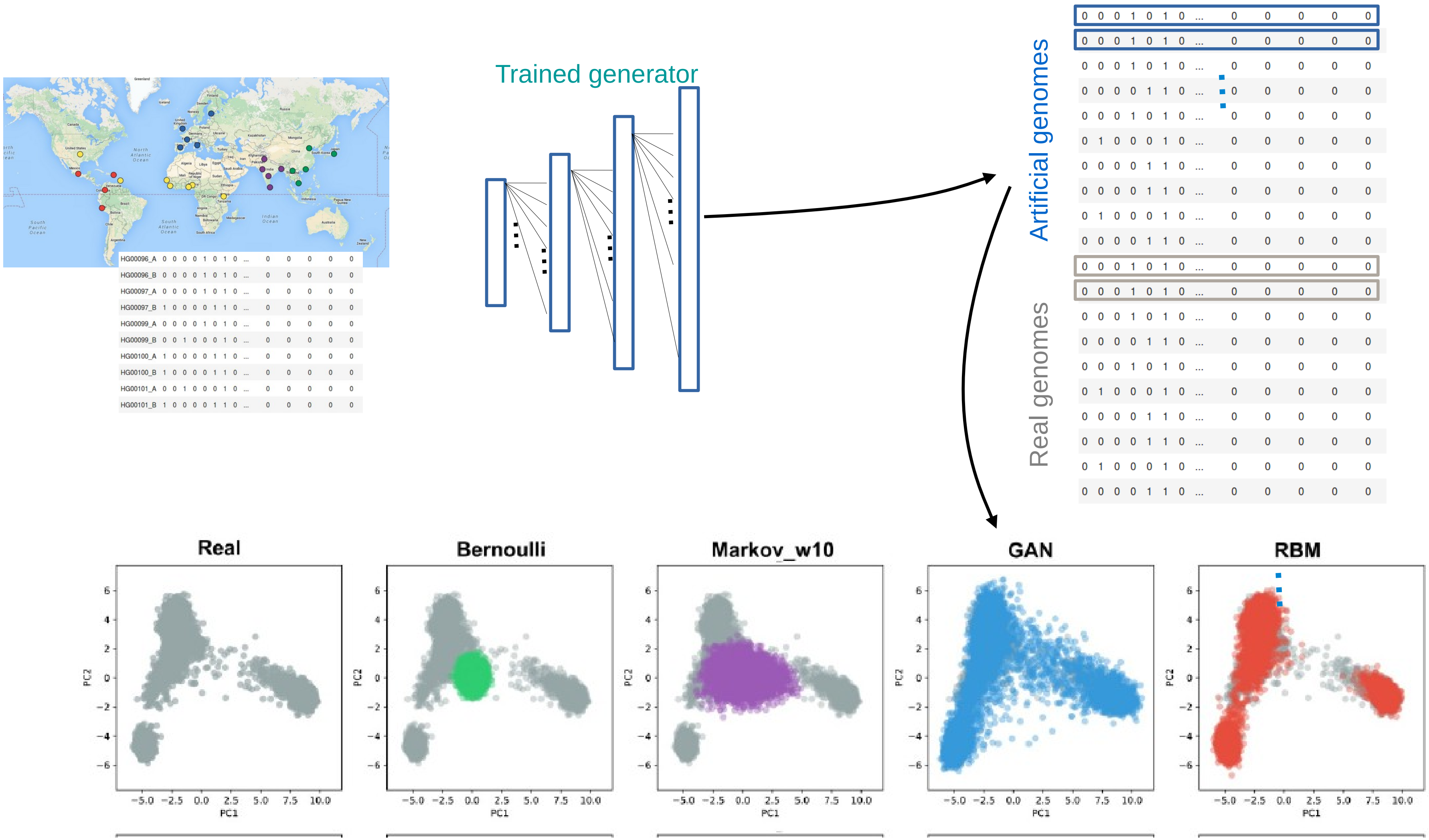
SNMF, a matrix factorization technique, applied to the 1000genomes human dataset goal is similar to STRUCTURE (Pritchard et al 2000)

Frichot et al. Fast and efficient estimation of individual ancestry coefficients. Genetics (2014) 196 (4): 973-983

Engelhardtand Stephens (2010) to understand the links between the classical STRUCTURE algorithm, PCA, Matrix Factorization, etc.

Unsupervised learning

- Generative models: can be used for generation, dimension reduction, exploring latent space Yelmen et al 2021/2023 (GAN, RBM, VAE neural networks) ; Battey et al 2021 (VAE) ; Ausmees et al 2021 (AE – not generative) ; Review coming soon Yelmen and Jay 2023



Yelmen et al 2021 PloS Genetics

Supervised learning

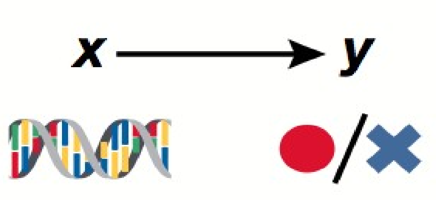
Supervised = Learn a relationship (a general model) linking input data (or features) to observed labels

Can you give examples of supervised tasks in popgen?

Supervised learning

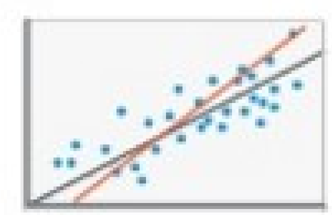
Supervised = Learn a relationship (a general model) linking input data (or features) to observed labels

Classification (predict a class)



Regression (predict a variable)

y



x

What for:

- Predict labels of new unlabeled samples (eg what's on an image?)

- Understand better the relationship between features and the label (eg understand which set of genes allow to predict a disease risk),

- ...

Supervised learning

Supervised = Learn a relationship (a general model) linking input data (or features) to observed labels

Classical pipeline

TASK:
predict y from x

Can you give examples of supervised ML algorithms?

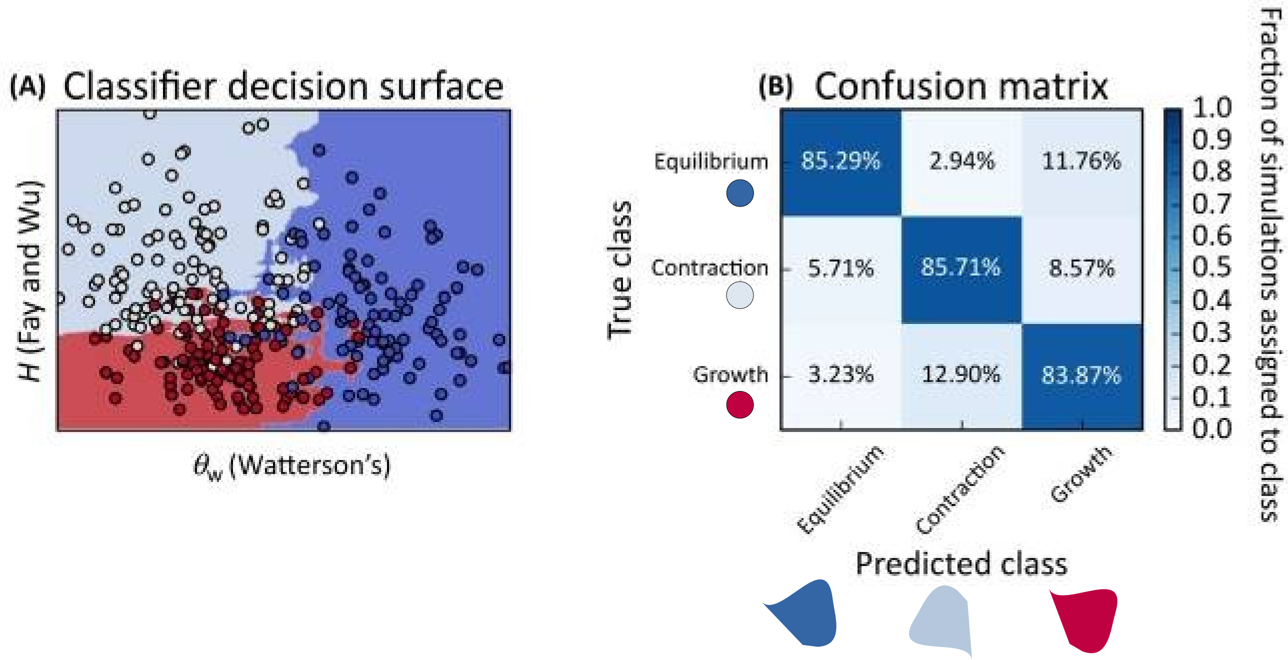
Supervised learning

Supervised = Learn a relationship (a general model) linking input data (or features) to observed labels

- Linear regression, logistic regression, ...
- Random forest
- Support Vector Machine (SVM)
- Predictive Neural Networks
- Some Approximate Bayesian Computation algorithms (ABC-RF, ABC-NN with hight tolerance rate, etc.)
- ...

Supervised learning - tree/forest

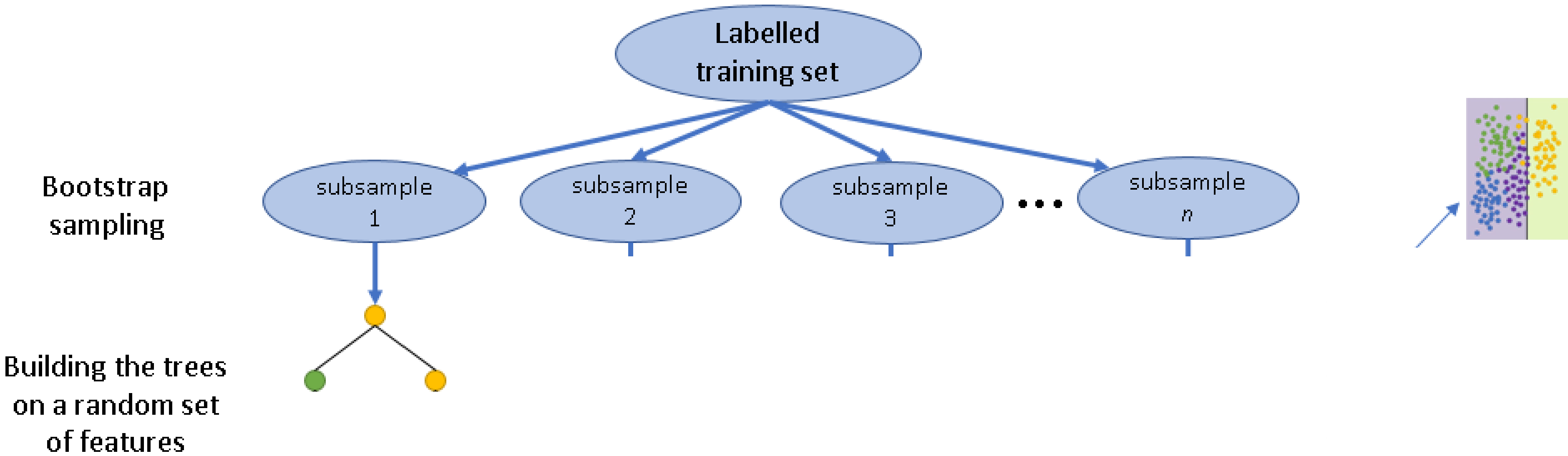
- Random forest / Extra tree classifier require **handcrafted features**



Review on ML methods in population genetic :

Schrider, Daniel R., and Andrew D. Kern. "Supervised machine learning for population genetics: a new paradigm." Trends in Genetics 34.4 (2018): 301-312.

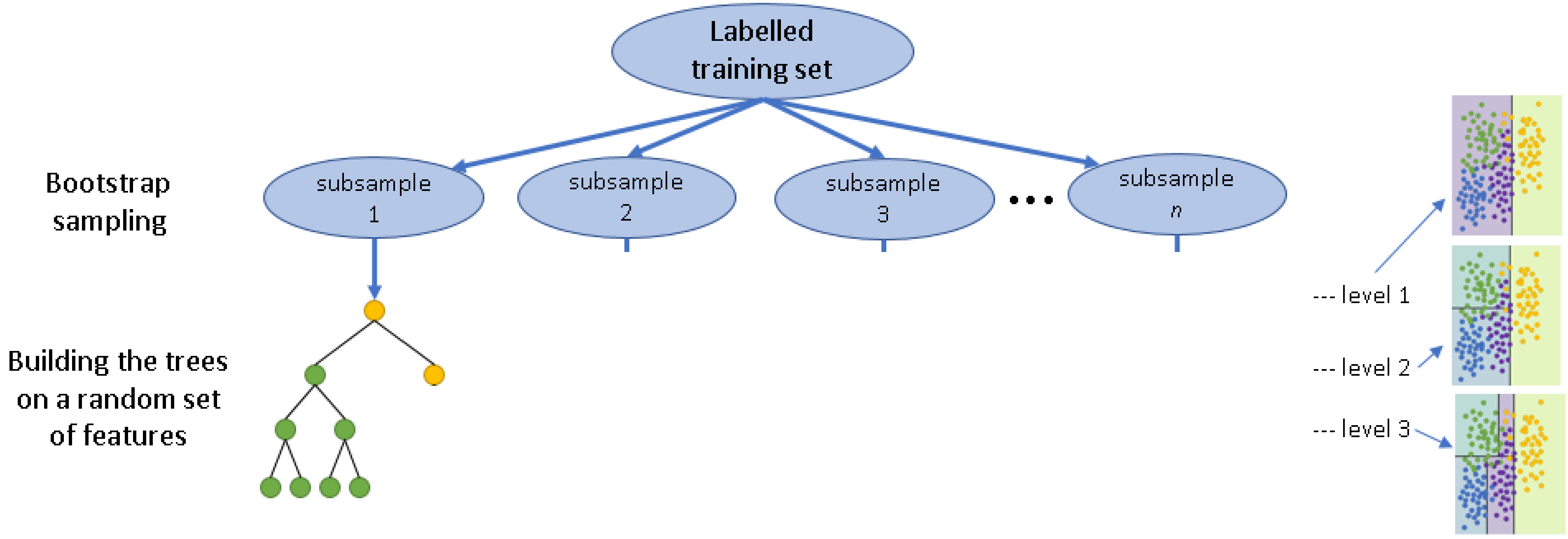
Supervised learning – Random Forest



hypothetical example of partition representation of classification tree across levels

Fig. adapted from https://catalyst.earth/catalyst-system-files/help/concepts/focus_c/oa_classif_intro_rt.html

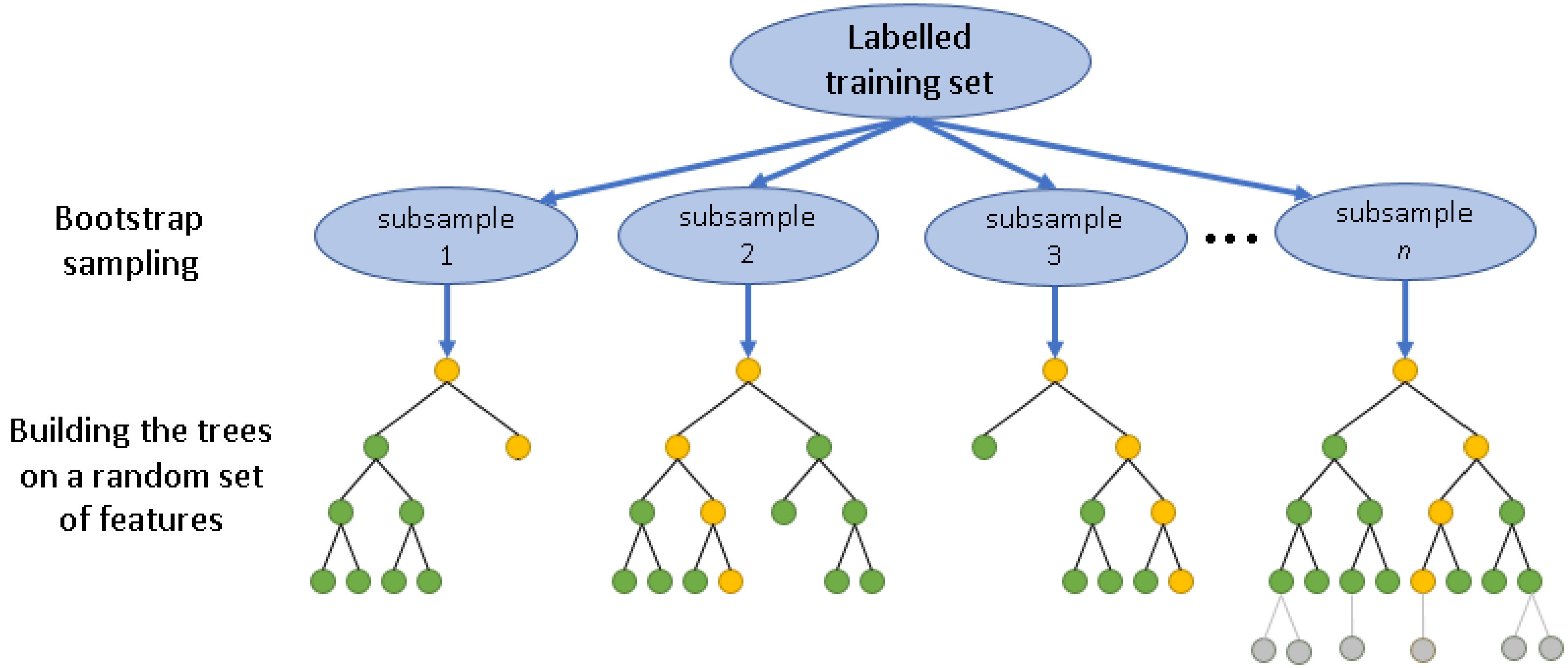
Supervised learning – Random Forest



hypothetical example of partition representation of classification tree across levels

Fig. adapted from https://catalyst.earth/catalyst-system-files/help/concepts/focus_c/oa_classif_intro_rt.html

Supervised learning – Random Forest



Do not pay attention to node colors

Fig. adapted from https://catalyst.earth/catalyst-system-files/help/concepts/focus_c/oa_classif_intro_rt.html

Supervised learning – Random Forest

Let's now follow the yellow path for a new sample through the different trees

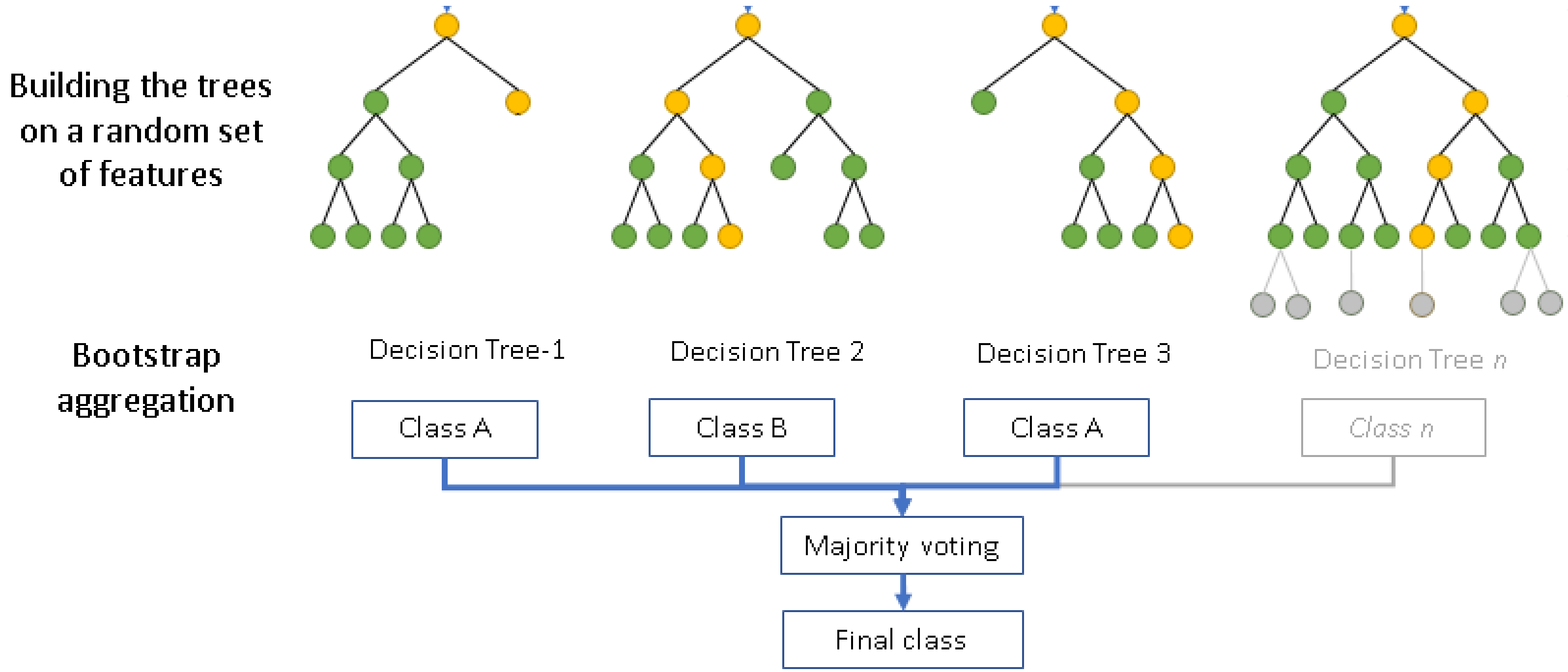
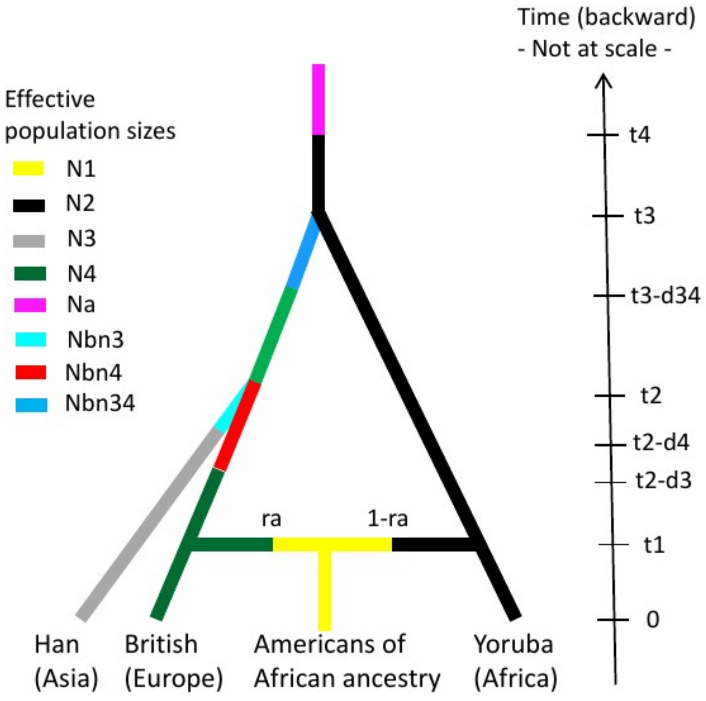


Fig. adapted from https://catalyst.earth/catalyst-system-files/help/concepts/focus_c/oa_classif_intro_rt.html

Supervised learning - example

- ABC - Random forest (Pudlo et al 2016, Raynal et al 2017 PCI evol biol, Raynal et al 2021) requires **handcrafted features** (but scales to very large number of sumstats compare to classical ABC with local regression)

Goal infering ra and N2/Na



Summary statistics

Single population statistics

HP0_i: proportion of monomorphic loci for population i
HM1_i: mean gene diversity across polymorphic loci (Nei, 1987)
HV1_i: variance of gene diversity across polymorphic loci
HMO_i: mean gene diversity across all loci (Nei, 1987)

Two population statistics

FP0_i&j: proportion of loci with null FST distance between the two samples for populations i and j (Weir and Cockerham, 1984)
FM1_i&j: mean across loci of non null FST distances
FV1_i&j: variance across loci of non null FST distances
FMO_i&j: mean across loci of FST distances (Weir and Cockerham, 1984)
NP0_i&j: proportion of 1 loci with null Nei's distance (Nei, 1972)
NM1_i&j: mean across loci of non null Nei's distances
NV1_i&j: variance across loci of non null Nei's distances
NMO_i&j: mean across loci of Nei's distances (Nei, 1972)

Three population statistics

AP0_i_j&k: proportion of loci with null admixture estimate when pop. i comes from an admixture between j and k
AM1_i_j&k: mean across loci of non null admixture estimate
AV1_i_j&k: variance across loci of non null admixture estimated
AMO_i_j&k: mean across all locus admixture estimates (Choisy *et al.*, 2004)

https://evolbiol.peercommunityin.org/articles/rec?id=90

Multiclass Support Vector Machine (SVM)

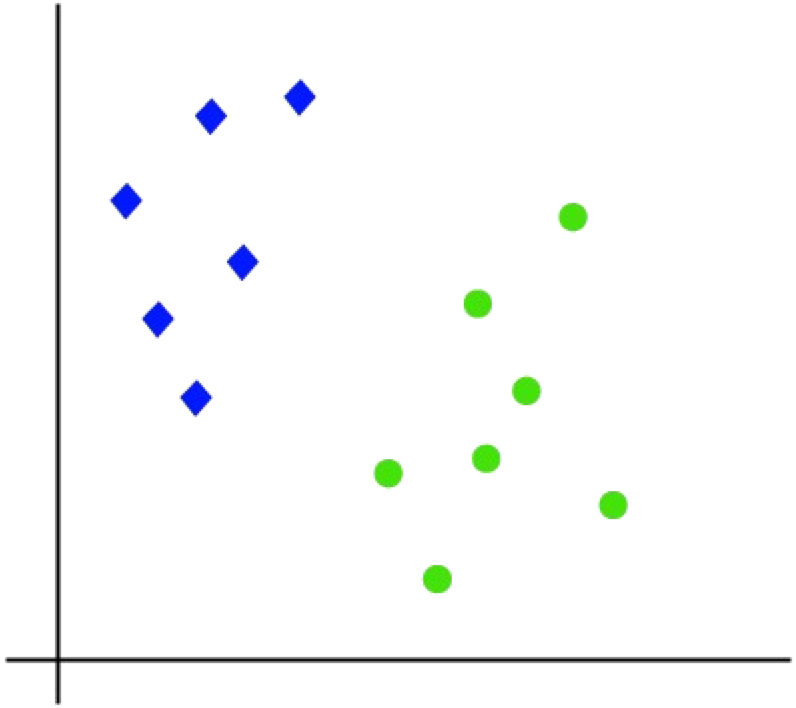
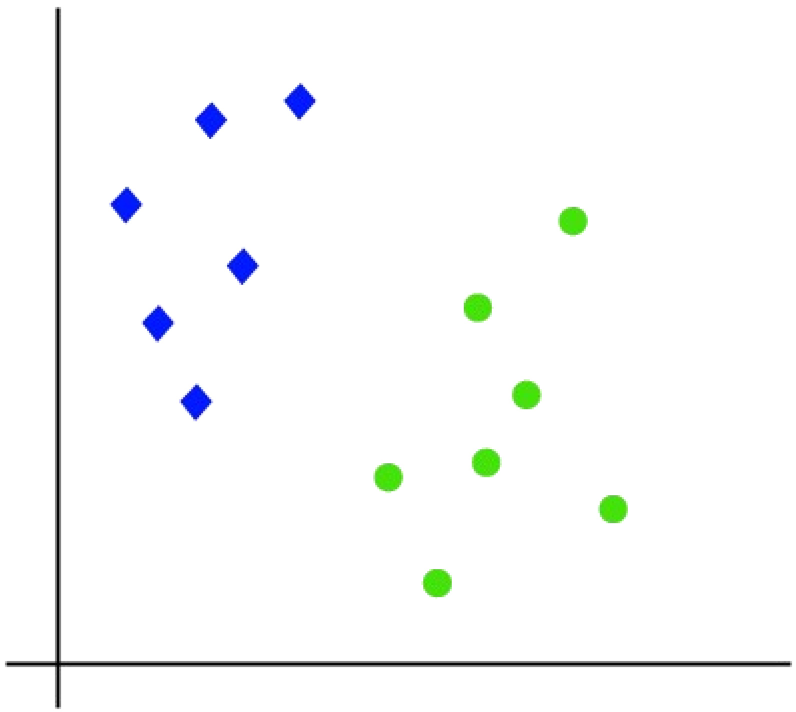


Image from https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

Multiclass Support Vector Machine (SVM)



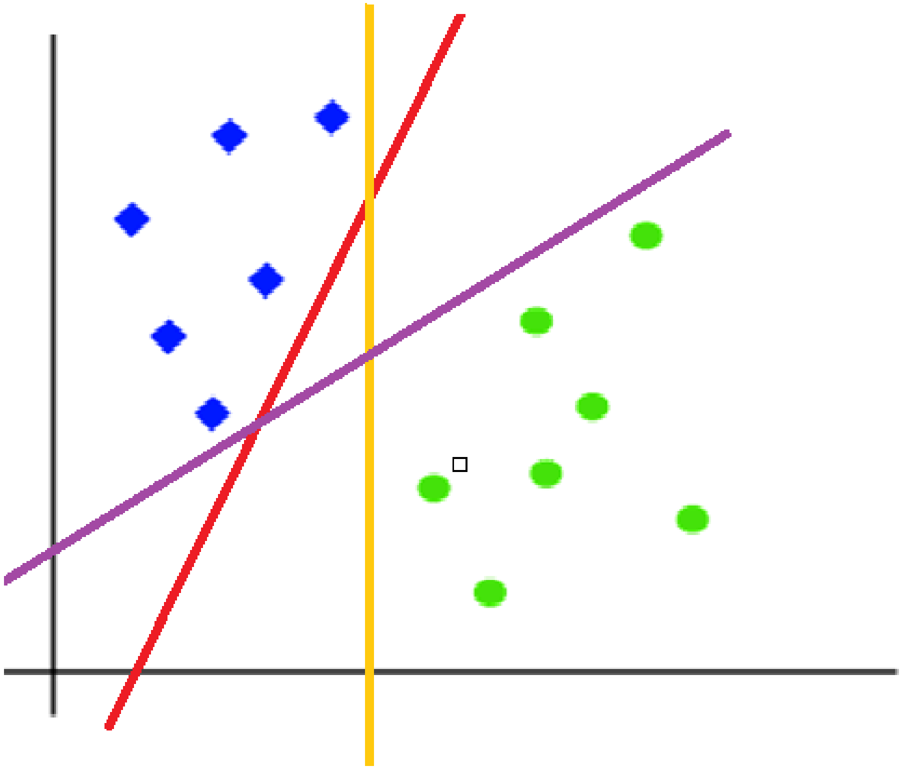
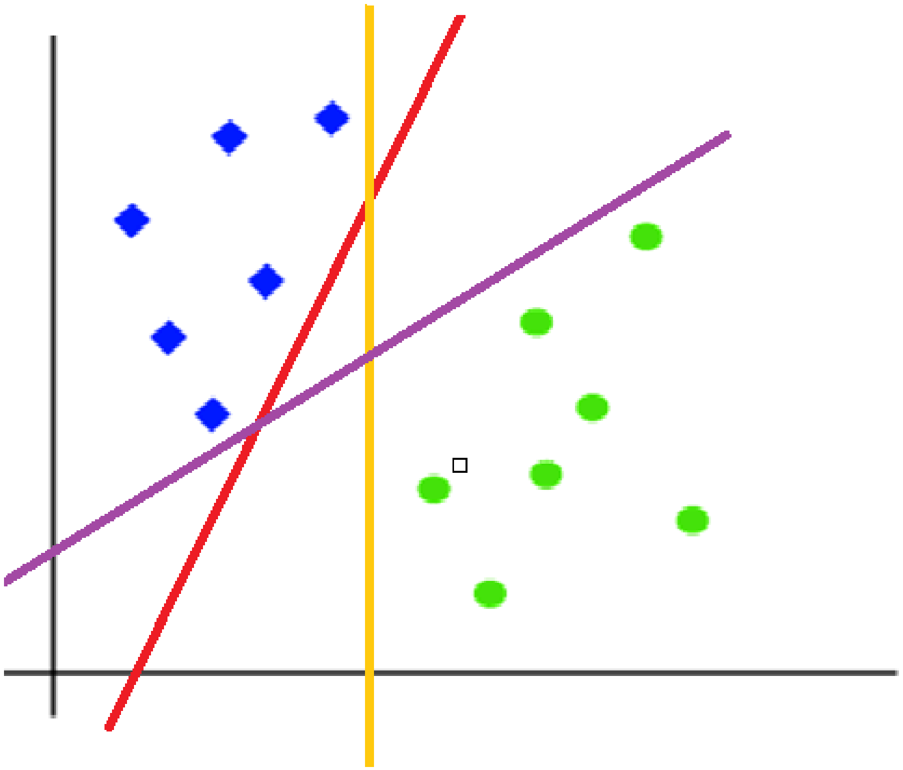


Image from https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

Multiclass Support Vector Machine (SVM)



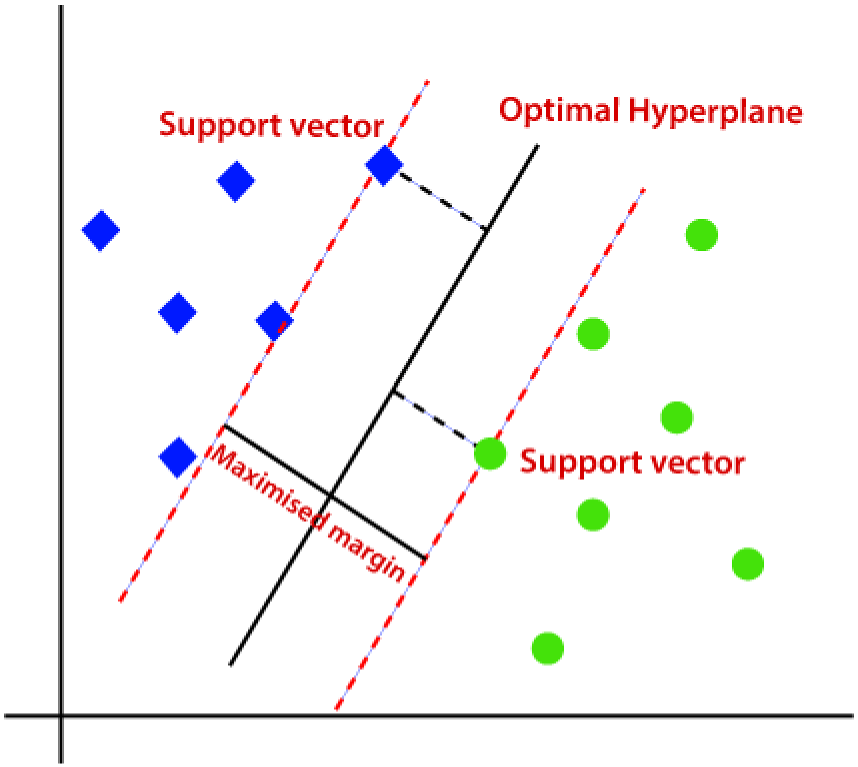


Image from https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

Multiclass Support Vector Machine (SVM) loss

The SVM loss is set so that the SVM "wants" the correct class for each image (y_i) to a have a higher score (s_{y_i}) than the incorrect ones (s_j) by some fixed margin (δ).

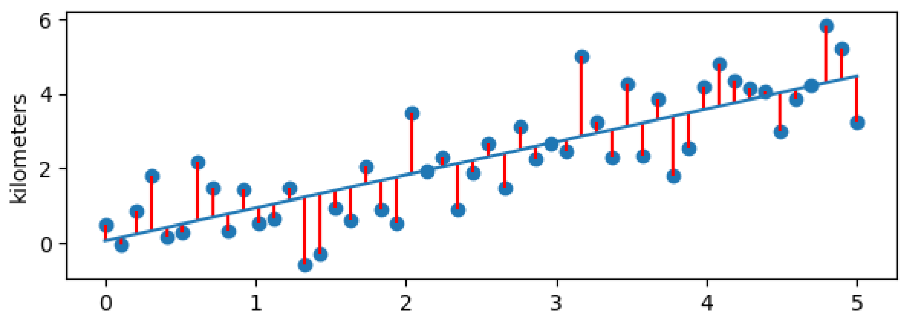
$$L_i = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + \delta)$$

Example:
$s = [13, -7, 11]$,$y_i = 0$,$\delta = 10$
$L_i = ?$

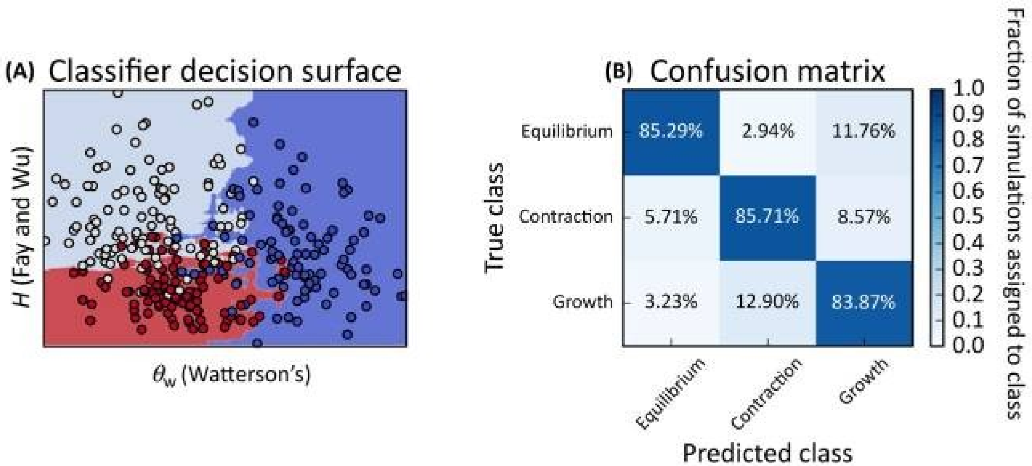
Evaluation of ML and DL methods?

Evaluation of ML and DL methods

- Define a performance measurement : score / loss / prediction error
  Examples :



Residual Mean Squared Error (RMSE)



Sensitivity/Recall (True Positive rate);
Specificity/Precision (True Negative rate);
Misclassification rate; F1-score

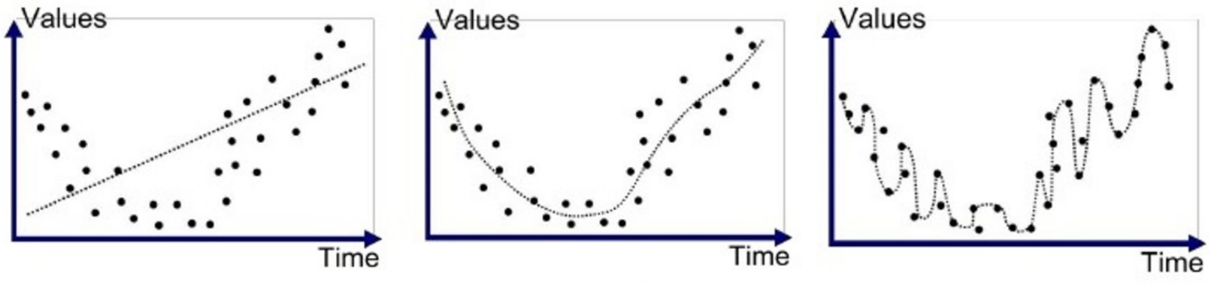
Cross-entropy (CE)

https://stats.stackexchange.com
Schrider & Kern Trends in Genetics 34.4 (2018): 301-312.

Evaluation of ML and DL methods

- Define a performance measurement : score / loss / prediction error
- Should you always use the maximum data available for training a model ?

Underfitting/Overfitting

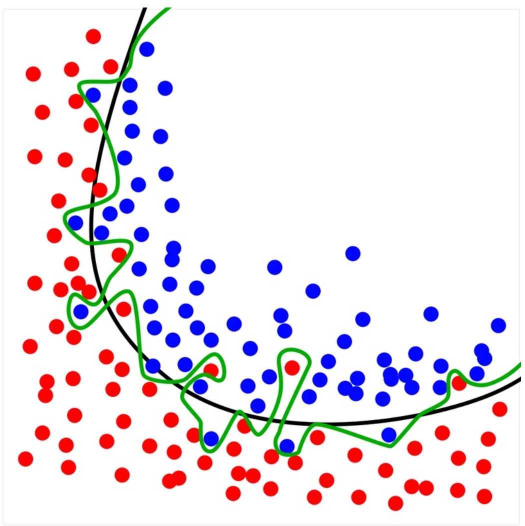


Underfitted

Good Fit/Robust

Overfitted

Part 2



https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76

https://en.wikipedia.org/wiki/Overfitting

Evaluation of ML and DL methods

Should you always use the maximum data available for training a model ? **NO**

split the data train/validation/test OR perform cross-validation

Outline

Part 1

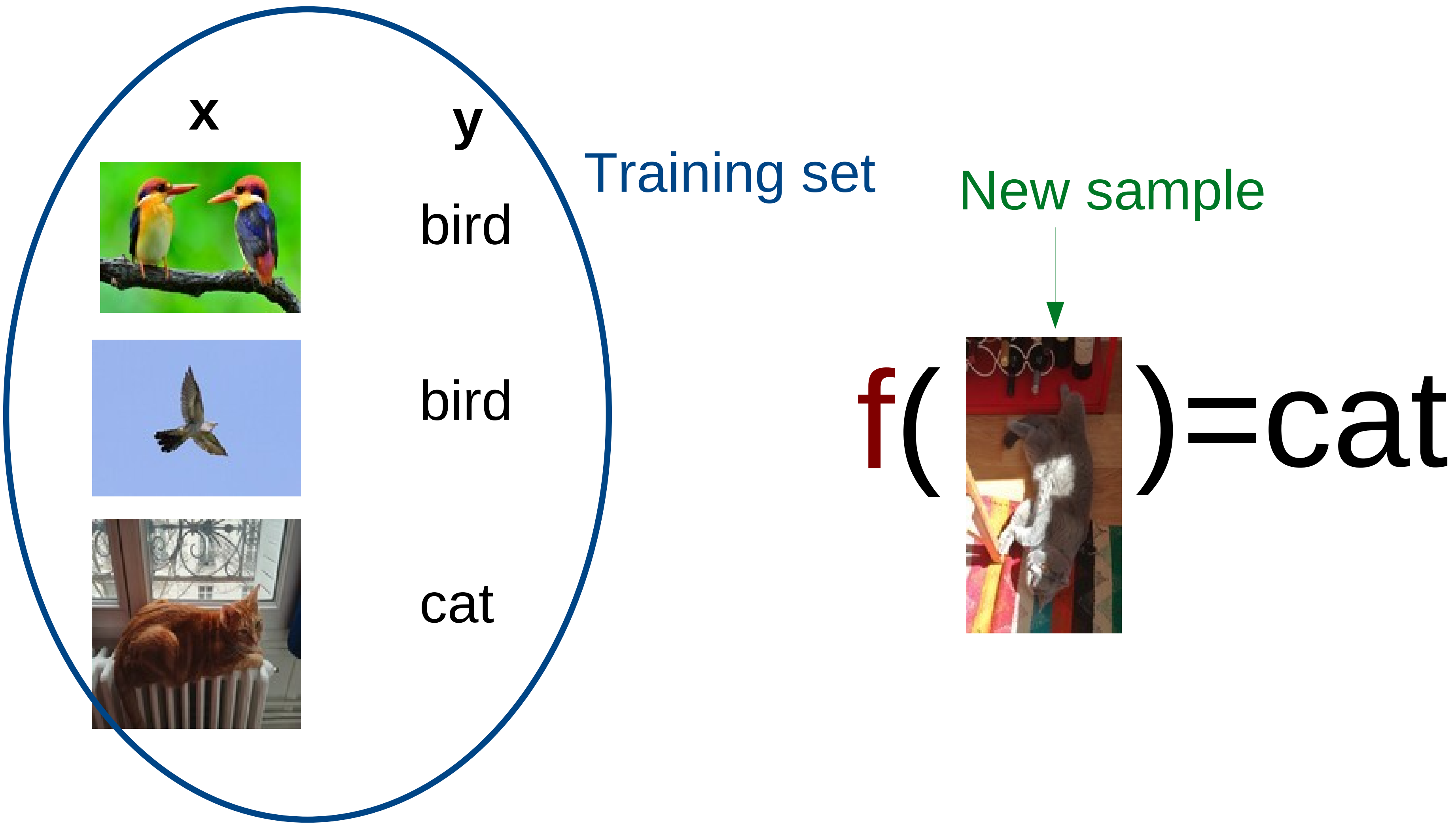
I. Machine Learning: basic concepts and terminology

II. What's a deep neural network (DNN) ?

Part 2
Deep Learning for population genetics

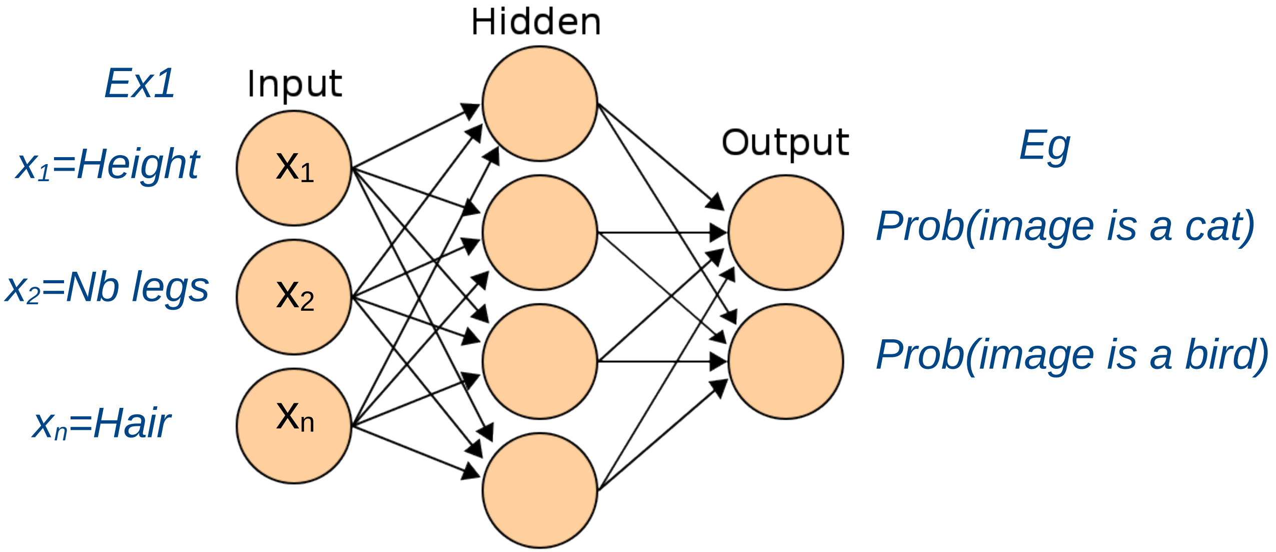
Opening on applications of unsupervized deep learning to popgen

Hands-on: building/training/re-using ML and DL models with application to population genetics (demography/selection)
ML: scikit-learn
DL: dnadna https://mlgenetics.gitlab.io/dnadna/


x
y
bird
bird
cat
Training set
New sample
f( )=cat

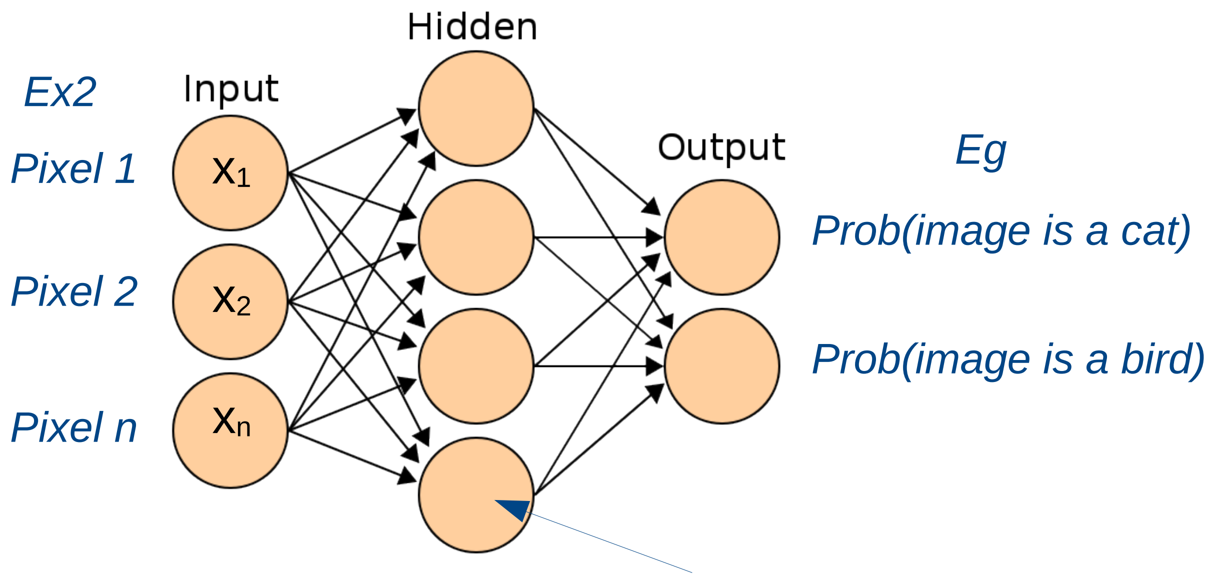
Neural networks

- Multi-Layer Perceptron / Fully connected neural net



Neural networks

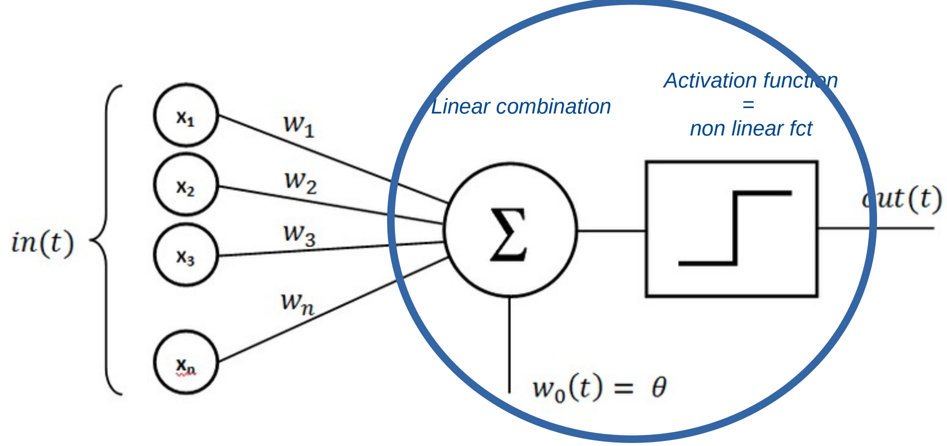
- Multi-Layer Perceptron / Fully connected neural net



NOT handcrafted features!

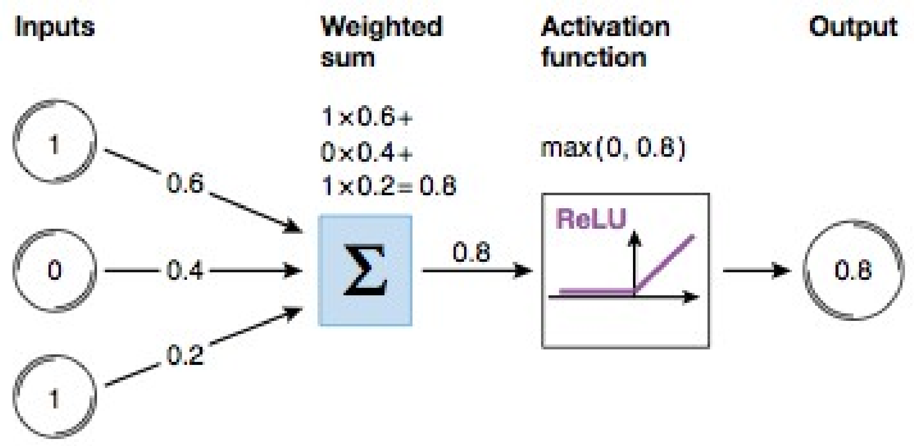
Internal layer(s) : learn an hidden representation from the data (ie learn how to encode the data)

Neural networks

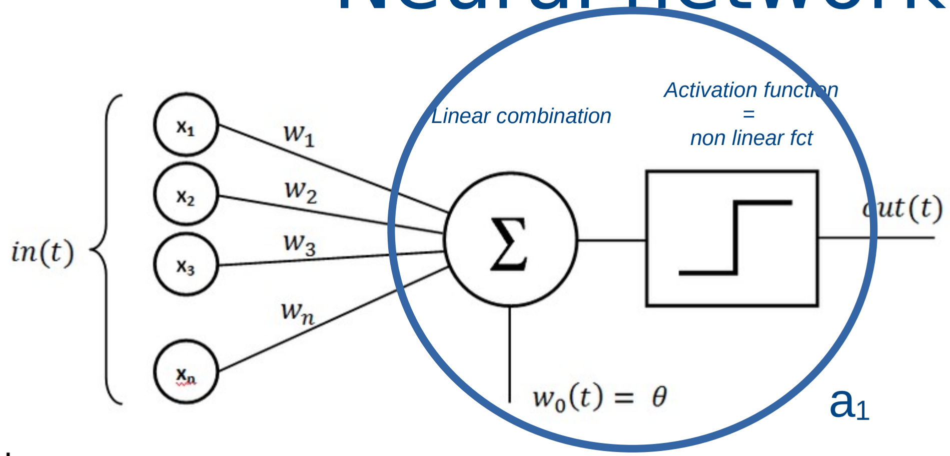


$$out = f\left(w_0 + w_1 x_1 + w_2 x_2 + \ldots\right)$$

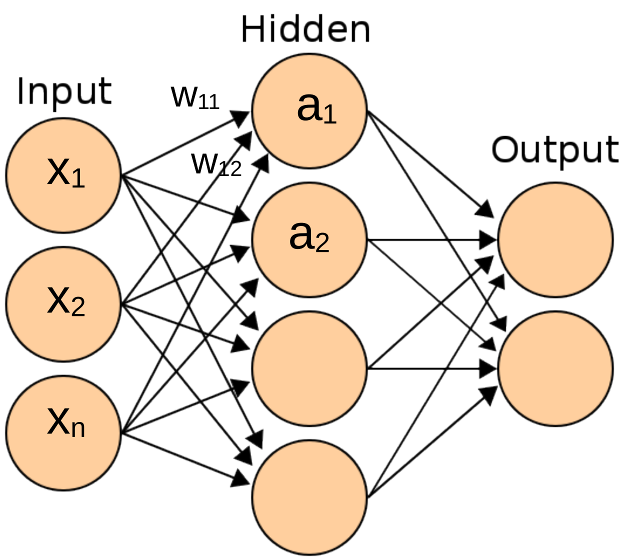
1 neuron = f (∑)
Linear combi and Non Linear activation function f()



Neural networks



$$out = f\left(w_0 + w_1 x_1 + w_2 x_2 + \ldots\right)$$



1 neuron = $f(\sum)$

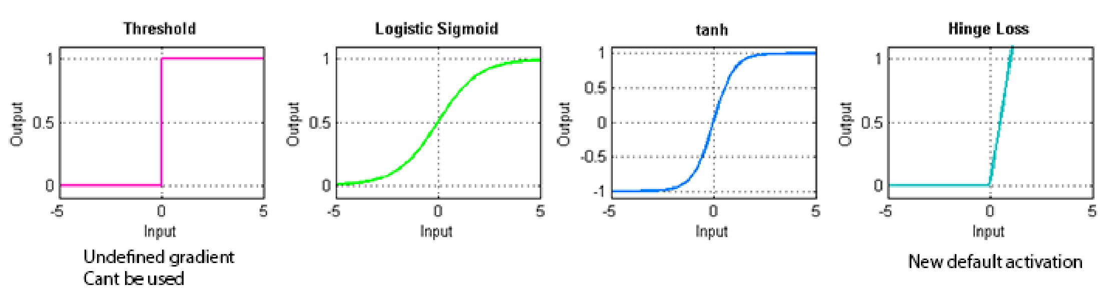
Linear combi and Non Linear activation function f()

$$a_1 = f\left(w_{10} + w_{11} x_1 + w_{12} x_2 + \ldots\right)$$

$$a_2 = f\left(w_{20} + w_{21} x_1 + w_{22} x_2 + \ldots\right)$$

...

Choices for activation functions f



...

Training Neural networks

- Training = estimating all the weights $w_{ij}^{(l)}$ for all layers (l) with the aim of minimizing the loss or cost function
- Loss/cost function: how well your classifier/regressor do ?
→ define a function quantifying how far you are from the truth
eg.

$$loss = \frac{1}{n} \sum_{example\ i=1}^{n} \left(y_i^{predicted} - y_i^{truth} \right)^2$$

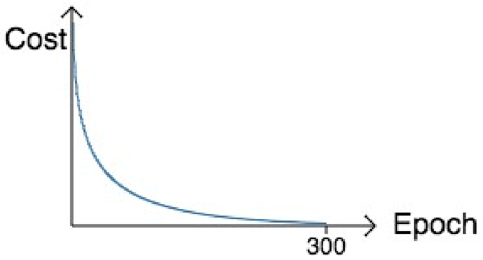
- Training algorithm :

Step 1 Initialize randomly the weights

For each epoch :

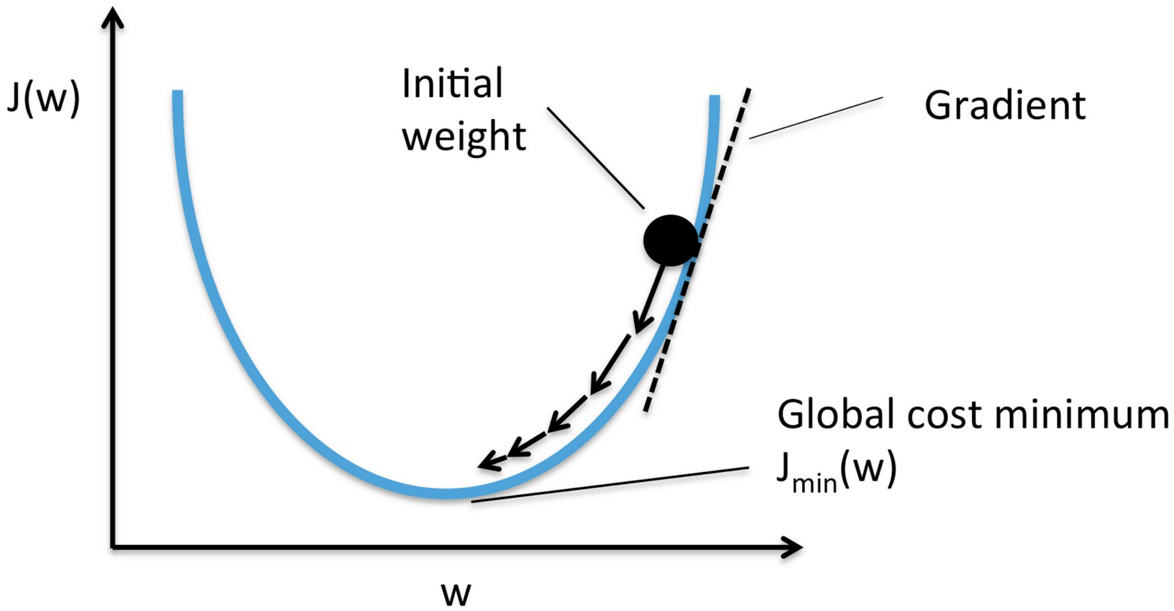
Step 2 Forward pass your examples (eg labeled images of cats and birds) through net to compute predicted values (as previously explained)
Step 3 Compute loss
Step 4 Backward propagation
Step 5 Update weights (gradient descent algorithm)

Repeat from step 2 until a plateau is reached



Training Neural Networks

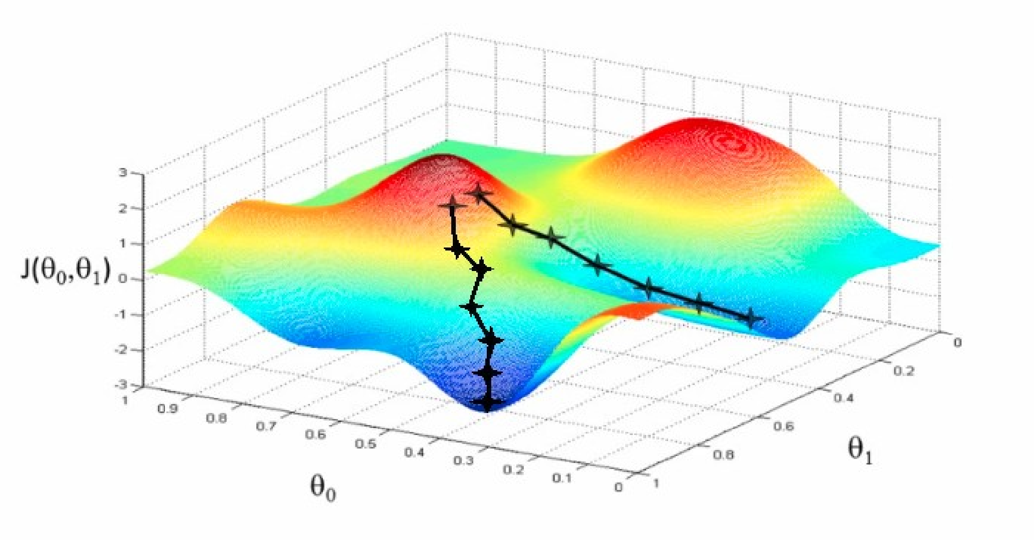
Gradient descent
Parameter w to optimize according to loss J



http://rasbt.github.io/mlxtend/user_guide/general_concepts/gradient-optimization/

Training Neural Networks

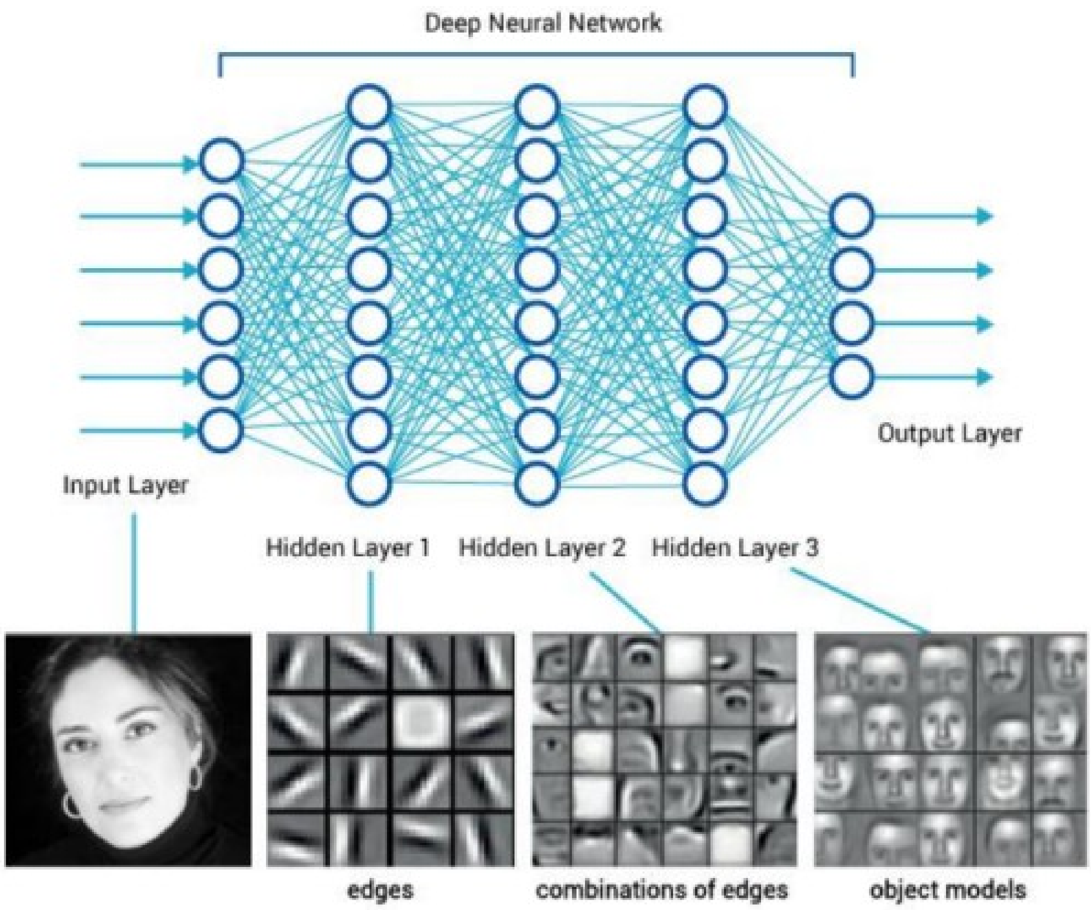
Stochastic gradient descen might reach different local minima



http://rasbt.github.io/mlxtend/user_guide/general_concepts/gradient-optimization/

DL - learning hierachical representations

Deep Learning (DL) = deep neural networks = nnet with multiple layers



DL - learning hierachical representations

- Able to learn a hierarchy of representations with increasing level of abstraction

  Eg. for image :
  pixel → edge → motif → part → full object → combination (eg landscape, scene)

  Eg. for text :
  letter → word → word group –> sentence → story

  ...

- A layer = trainable function that transforms input into features at a certain hierarchy level

Deep learning – When does it work ?

- Lots of data to be able to train the network (BUT see transfer learning)
- Labeled data if it's for a supervised task
- Computational power, in particular GPU
- For vision/image/text data, organized challenges clearly showed the superiority of DL
- Question to ask yourself: in practice does it matter to you to improve the performances by xx% ?

Deep learning hyper-parameters (HP)

- You still have to make decisions about (1) your achitecture (#layers, #nodes per layer, layer type,...) ; (2) the algorithm/optimization hyper-parameters
- Usually done by training numerous networks with numerous HP and keeping the one performing the best. Can be done in a smart way with e.g. bayesian HP optimization. Automatic Deep Learning : active research area

Just a taste of some NN algo hyper-parameters

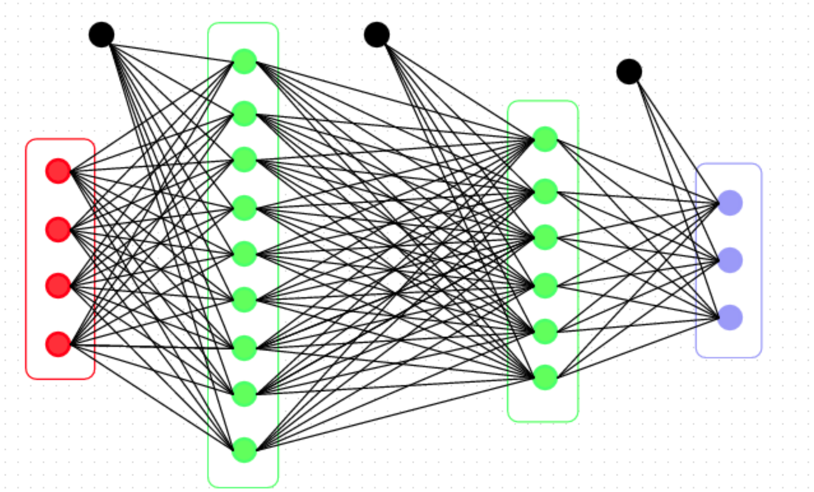


Table 2. Central parameters of a neural network and recommended settings.

Name	Range	Default value
Learning rate	0.1, 0.01, 0.001, 0.0001	0.01
Batch size	64, 128, 256	128
Momentum rate	0.8, 0.9, 0.95	0.9
Weight initialization	Normal, Uniform, Glorot uniform	Glorot uniform
Per-parameter adaptive learning rate methods	RMSprop, Adagrad, Adadelta, Adam	Adam
Batch normalization	Yes, no	Yes
Learning rate decay	None, linear, exponential	Linear (rate 0.5)
Activation function	Sigmoid, Tanh, ReLU, Softmax	ReLU
Dropout rate	0.1, 0.25, 0.5, 0.75	0.5
L1, L2 regularization	0, 0.01, 0.001	