

A gentle introduction to Artificial Intelligence & Machine Learning

DuMAS Department Day

September 22, 2023

Jean-Luc.Charles@mailo.com



An introduction to AI & Machine Learning



Some points of interest

- Preliminaries: the historical way...
 - Unsupervised / Supervised / Reinforcement Machine Learning
 - Societal issues
 - Technical (computing) slides available at the end for those who want to know more

An introduction to AI & Machine Learning



Some points of interest

- Preliminaries: the historical way...
 - Unsupervised / Supervised / Reinforcement Machine Learning
 - Societal issues
 - Technical (computing) slides available at the end for those who want to know more

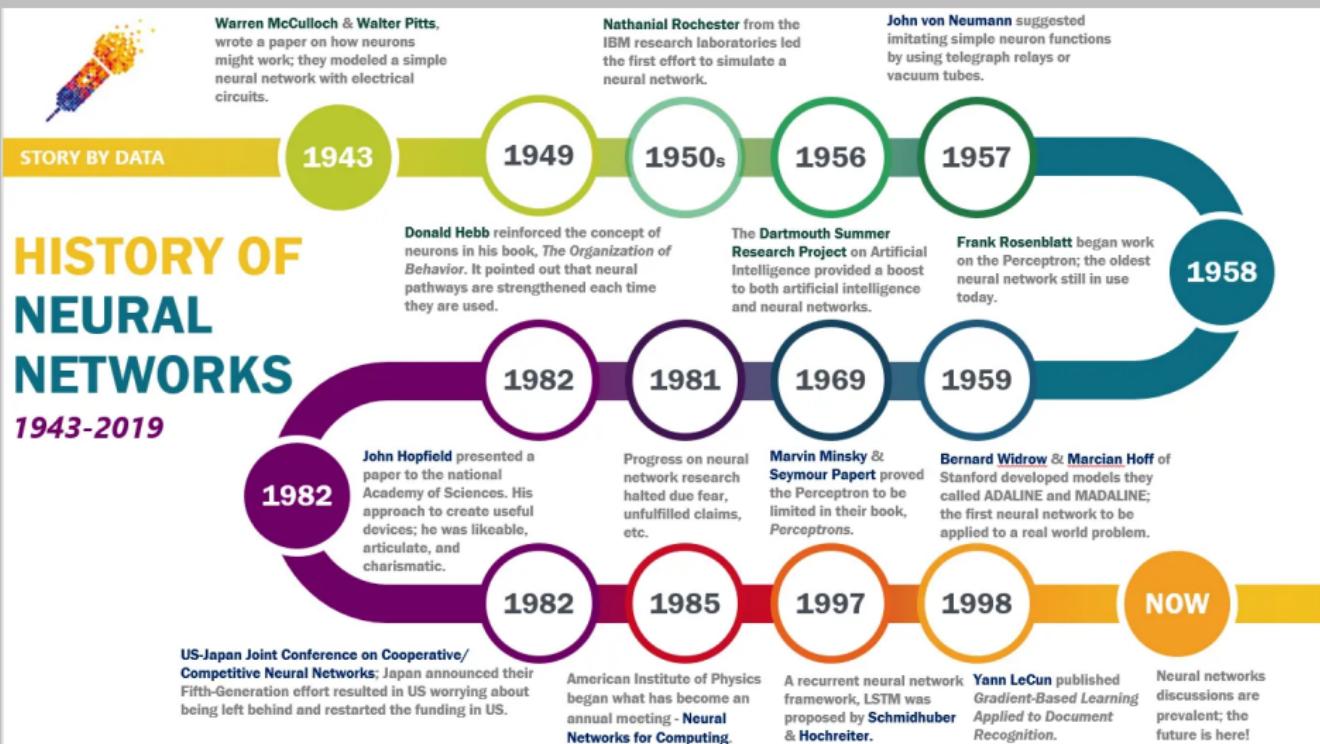


My profile

- I've been teaching Python programming (Scientific & Object Oriented)
 - I started to get interested in ML in 2015
 - I wrote several materials in ML : workshop, project & practical work for different schools (ENSA, ENSEIRB, ENSPIMA, PPU...)

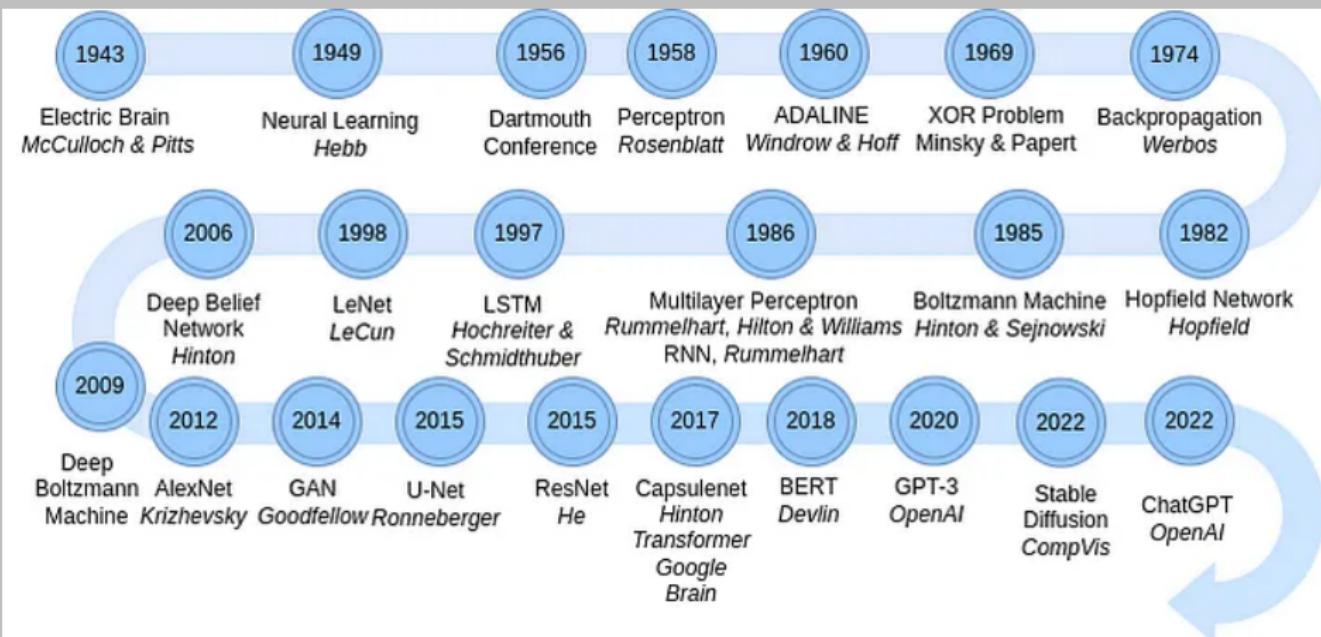
The historical way...

from: Kate Strachni: "Brief History of Neural Networks", medium.com



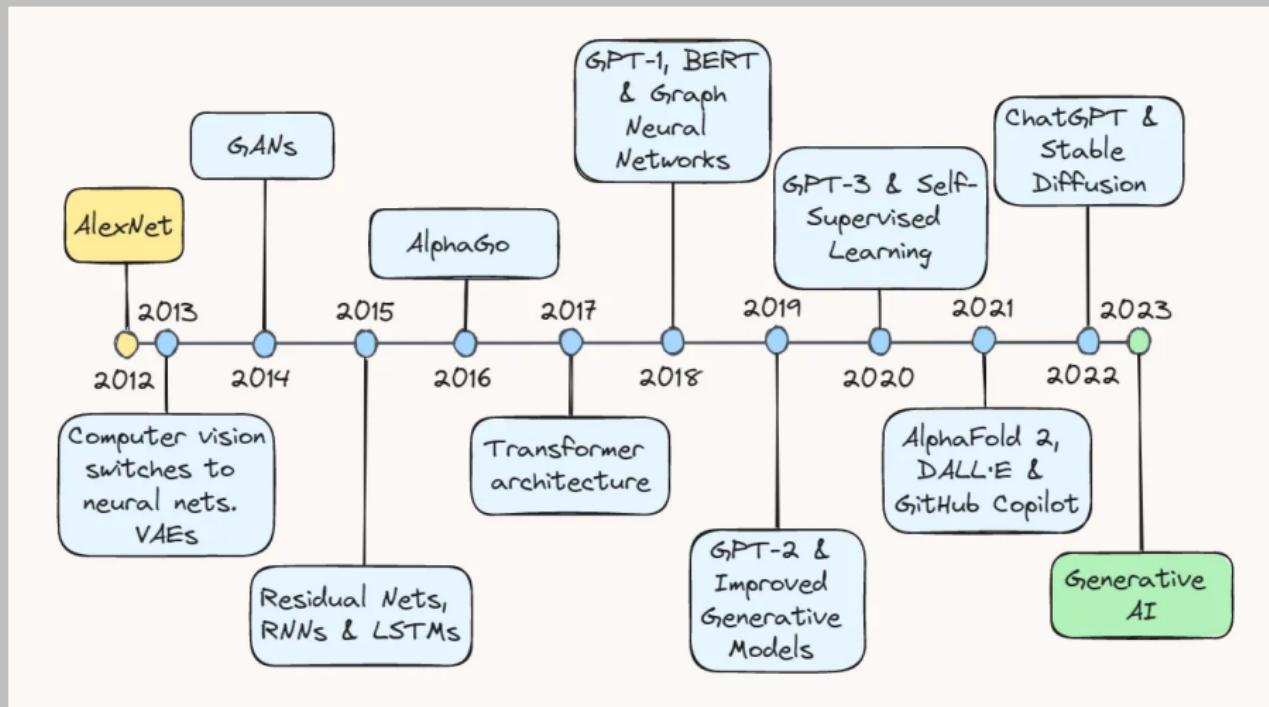
The historical way...

from: Pumalin: "A Brief History of Neural Nets", medium.com



The historical way...

from: Thomas A Dorf: "Ten Years of AI in Review", medium.com



Artificial Intelligence ?



Historically^a *badly chosen term!* Ambiguous current meaning...
Many (contradictory) definitions depending on periods and authors...

^afirst used in 1956 by [John McCarthy](#), researcher at Stanford during the Dartmouth conference

Artificial Intelligence ?



Historically^a *badly chosen term!* Ambiguous current meaning...
Many (contradictory) definitions depending on periods and authors...

^afirst used in 1956 by [John McCarthy](#), researcher at Stanford during the Dartmouth conference

- "...the science of making computers do things that require intelligence when done by humans." [Alan Turing, 1940](#)
- "the field of study that gives computers the ability to learn without being explicitly programmed." [Arthur Samuel, 1960](#)
- "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E ." [Tom Mitchell, 1997](#)
- Notion of *intelligent agent, rational agent*
"...agent that acts in such a way as to reach the best solution or, in an uncertain environment, the best predictable solution."

[Stuart Russel, Peter Norvig, "Intelligence Artificielle" 2015](#)

Artificial Intelligences ?

Strong AI

- Build systems that think exactly the same way that people do.
 - Try also to explain how humans think... **Who are not yet here.**

Weak AI

- Build systems that can behave like humans.
 - The results will tell us nothing about how humans think.
 - **We already are there...** We use it every day!
(anti-spam, facial/voice recognition, language translation...)

Artificial Intelligences ?

Strong AI

- Build systems that think exactly the same way that people do.
- Try also to explain how humans think... **Who are not yet here.**

Weak AI

- Build systems that can behave like humans.
- The results will tell us nothing about how humans think.
- **We already are there...** We use it every day!
(anti-spam, facial/voice recognition, language translation...)

General AI

- AI systems designed for the ability to reason in general.

Narrow AI

- AI systems designed for specific tasks.

Machine Learning: a field of AI

Page from [medium.com/machine-learning-for-humans/...](https://medium.com/machine-learning-for-humans/)

Machine learning \subseteq artificial intelligence

ARTIFICIAL INTELLIGENCE

Design an intelligent agent that perceives its environment and makes decisions to maximize chances of achieving its goal.

Subfields: vision, robotics, machine learning, natural language processing, planning, ...

MACHINE LEARNING

Gives "computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959)

SUPERVISED LEARNING

Classification, regression

UNSUPERVISED LEARNING

Clustering, dimensionality reduction, recommendation

REINFORCEMENT LEARNING

Reward maximization

Branches of Machine Learning

Supervised learning

labeled dataset is used to train algorithms:

- **Classification**

- Images classification
- Objects detection in images
- Speech recognition...

- **Regression**

- Predict a value...

- **Anomaly detection**

- Spam detection
- Manufacturing: finding known (learned) defects
- Weather prediction
- Diseases classification...

...

Branches of Machine Learning

Unsupervised learning

Analyze and cluster **unlabeled datasets**:

- **Clustering & Grouping**

- Data mining, web data grouping, news grouping...
- Market segmentation
- Astronomical data analysis...

- **Anomaly Detection**

- Fraud detection
- Manufacturing: finding defects even new ones
- Monitoring abnormal activity: failure, hacker, fraud...
- Fake account on Internet...

- **Dimensionality reduction**

- Compress data using fewer numbers...

...

Branches of Machine Learning

Deep Reinforcement Learning DRL

An agent (the Neural Network) learns how to drive an environment by maximising a **reward**:

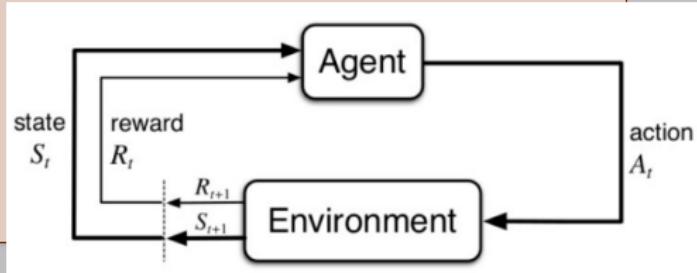
- **Control/command**

- Controlling **robots**, drones, **mecatronic systems**
- Factory optimization
- Financial (stock) trading...

- **Decision making**

- games (video games)
- financial analysis...

...



Various approaches for ML algorithms

Supervised learning:

- Neural Networks
- Bayesian inference
- Random forest
- Decision Tree
- Support Vector Machine (SVM)
- K-Nearest Neighbor
- Linear regression
- Logistic regression
- ...

Unsupervised learning:

- Neural Networks
- Principal Composant Analysis
- Singular Value Decomposition
- K-mean & Prob. clustering
- ...

Reinforcement learning:

- Neural Networks (Q-learning, Actor-Critic, DDPG, PPO...)
- Monte Carlo
- SARSA
- ...

Various approaches for ML algorithms

Supervised learning:

- Neural Networks
- Bayesian inference
- Random forest
- Decision Tree
- Support Vector Machine (SVM)
- K-Nearest Neighbor
- Linear regression
- Logistic regression
- ...

Unsupervised learning:

- Neural Networks
- Principal Composant Analysis
- Singular Value Decomposition
- K-mean & Prob. clustering
- ...

Reinforcement learning:

- Neural Networks
- Monte Carlo
- SARSA
- ...

This introduction deals only with **Artificial Neural Networks**.

Fields & applications of ML

Computer Vision

- Image Classification
- Object Detection
- (Semantic) Segmentation
- Image Generation
Les 10 Meilleurs Générateurs d'Images
- Pose Estimation
- ...



Fields & applications of ML

Computer Vision

- Image Classification
- Object Detection
- (Semantic) Segmentation
- Image Generation
Les 10 Meilleurs Générateurs d'Images
- Pose Estimation
- ...

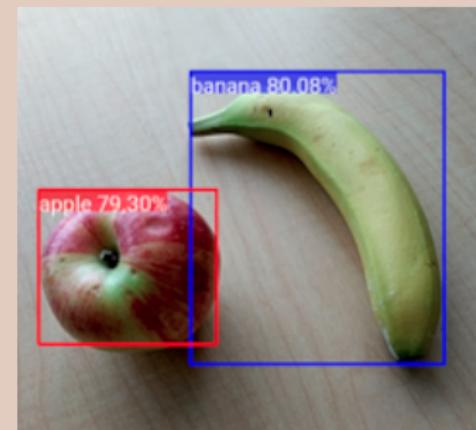


Image credit: [Tensorflow](#)

Fields & applications of ML

Computer Vision

- Image Classification
- Object Detection
- (Semantic) Segmentation
- Image Generation
Les 10 Meilleurs Générateurs d'Images
- Pose Estimation
- ...



Fields & applications of ML

Computer Vision

- Image Classification
- Object Detection
- (Semantic) Segmentation
- Image Generation
[Les 10 Meilleurs Générateurs d'Images](#)
- Pose Estimation
- ...



Image credit: [stylegan](#)

Fields & applications of ML

Computer Vision

- Image Classification
- Object Detection
- (Semantic) Segmentation
- Image Generation
[Les 10 Meilleurs Générateurs d'Images](#)
- Pose Estimation
- ...

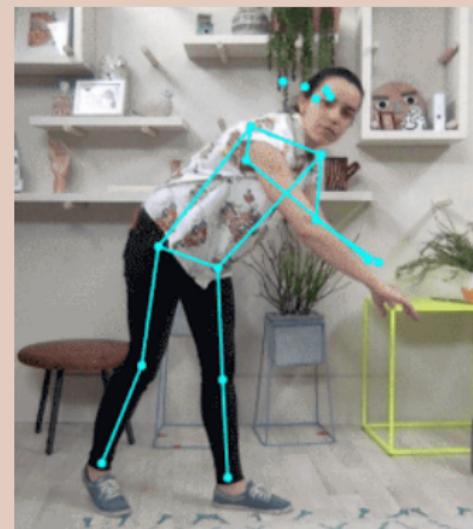


Image credit: [Tensorflow-Pose Estimation](#)

Fields & applications of ML

Natural Language Processing: NLP

- Natural Language Understanding (NLU)
- Natural Language Generation (NLG)
- Speech recognition / Speech Synthesis (Text To Speech)
- Machine Translation (languages)
- Virtual agents and ChatBots
- Optical character recognition (OCR)
- ...

Neural Network Architectures

Many NN architectures for many applications (non-exhaustive list):

- **Feed Forward**: the simplest architecture made of successive layers of neurones, with *Feed Forward* and *Back Propagation* algorithms.
- **Convolutional** (CNN): Mostly used for analyzing and classifying images.
- **Recurrent** (RNN): Used for time series, like the Long Short-Term Memory (LSTM) algorithm.
- **Transformers** : Recently used for Natural Language Processing and then for image classification.
- **Auto Encoder** (AEN): Dimensionality reduction, Feature extraction, Denoising of data/images, Inputting missing data.
- **Generative Adversarial** (GAN): to generate text, images, music...
- **Large Language Model** (LLM): read text, sound, write books, images, speak, make music ...ChatGPT

[Synthetic Graphical chart: [from Saul Dobilas on Medium](#)]

Societal issues: Explainability

Explainability is fast becoming a top priority in research, where it is often abbreviated as

xAI	Explainable Artificial Intelligence
iML	Interpretable Machine learning

Explainability

- **Unexplainability** of the results computed by Neural Networks still constitutes an obstacle to their dissemination today.
- Deep learning with Neural Networks is often denigrated as a **Black box** by scientists with a Cartesian approach.
- Even ML developpers have difficulties to explain by which way a NN decision has been computed

Societal issues: Decision-making, Certification

Decision-making

- An increasing amount of **decisions** in sensible fields (criminal justice, health, insurance, defense...) are being ceded to ML algorithms to the detriment of human control, raising concern for loss of fairness and equitability.
- Decision-making algorithms rely inevitably on assumptions (such as the quality of data the algorithm is trained with)

Societal issues: Decision-making, Certification

Decision-making

- An increasing amount of **decisions** in sensible fields (criminal justice, health, insurance, defense...) are being ceded to ML algorithms to the detriment of human control, raising concern for loss of fairness and equitability.
- Decision-making algorithms rely inevitably on assumptions (such as the quality of data the algorithm is trained with)

Evaluation, Certification

- The evaluation and the certification of AI systems is still a major issue for their integration in the industry
- Formal certification of ML algorithms is still a subject of research today
- Example: LNE Process certification for AI

Societal issues: Trustworthy AI



On 8 April 2019, the European High-Level Expert Group on AI (AI HLEG)¹ presented [Ethics Guidelines for Trustworthy Artificial Intelligence](#)

According to the Guidelines, trustworthy AI should be:

- Lawful respecting all applicable laws and regulations
- Ethical respecting ethical principles and values
- Robust both from a technical perspective while taking into account its social environment

Based on the feedback received, the AI HLEG presented the final **Assessment List for Trustworthy AI (ALTAI)** in July 2020.

¹Groupe d'Experts indépendants de Haut Niveau en Intelligence Artificielle (GEHN IA)

Societal issues: Trustworthy AI

ALTAI

Assessment List for Trustworthy AI

- Does the AI system potentially negatively discriminate against people...?
- Does the AI system respect the rights of the child?
- Does the AI system protect personal data relating to individuals in line with GDPR?
- Does the AI system respect the freedom of expression and information and/or freedom of assembly and association?



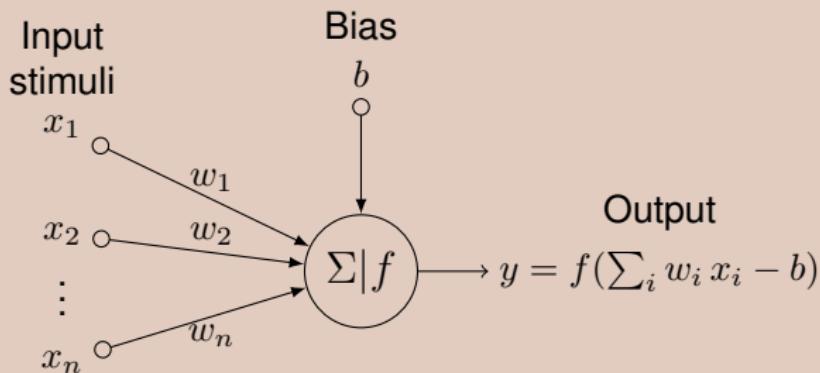
Societal issues

Some references

- The Alan Turing Institute
[Understanding artificial intelligence ethics and safety](#)
- Nature
[Ethical principles in machine learning and artificial intelligence: cases from the field and possible ways forward](#)
- Rapport Villani (2018)
[Donner un sens à l'intelligence artificielle : pour une stratégie nationale et européenne](#)

Computing aspects

The Artificial neuron model

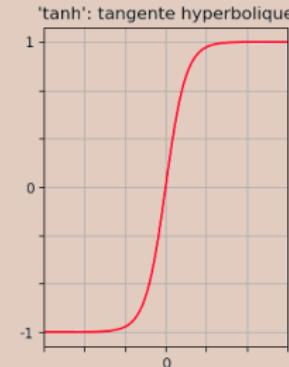


An **artificial neuron**:

- receives the input stimuli $(x_i)_{i=1..n}$ with **weights** (w_i)
- computes the **weighted sum** of the input $\sum_i w_i x_i - b$
- outputs its activation $f(\sum_i w_i x_i - b)$, computed with a non-linear **activation function** f .

Computing aspects

Common activation functions

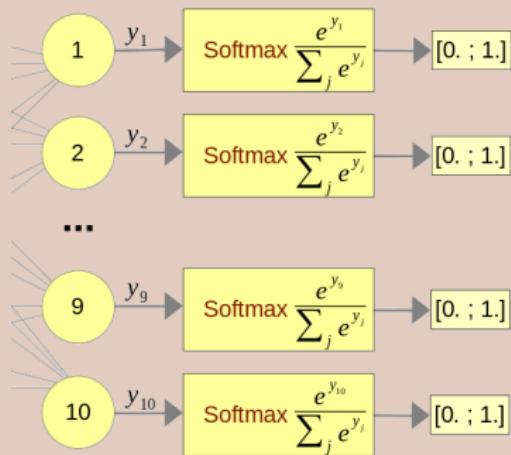


- Introduces a non-linear behavior.
- Sets the range of the neuron output: $[-1, 1]$, $[0, 1]$, $[0, \infty[$...
- The bias b sets the activation threshold of the neuron.

Computing aspects

- intermediate layers \leadsto **relu** promotes network learning ².
- **softmax** always used for in the last layer for **Classifying**.

Example: activation function *softmax* for the last layer, 10 classes



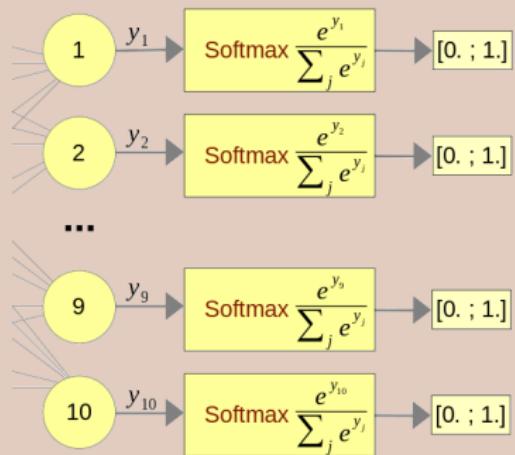
- The activation of neuron k is $Y_k = e^{y_k} / \sum_i e^{y_i}$ with $y_k = \sum_i \omega_{ix_i} - b$ calculated by the neuron k .
 - The neurons outputs are interpreted as **probabilities** in the interval [0,1].
- \leadsto the label of the neuron with the highest probability is the network response

² avoids the *vanishing gradient* that appears in *back propagation*

Computing aspects

- intermediate layers \leadsto **relu** promotes network learning ².
- **softmax** always used for in the last layer for **Classifying**.

Example: activation function *softmax* for the last layer, 10 classes



- The activation of neuron k is $Y_k = e^{y_k} / \sum_i e^{y_i}$ with $y_k = \sum_i \omega_{ix_i} - b$ calculated by the neuron k .
- The neurons outputs are interpreted as **probabilities** in the interval $[0,1]$.
 - \leadsto the label of the neuron with the highest probability is the network response

² avoids the *vanishing gradient* that appears in *back propagation*

Computing aspects [Classification]

One-hot coding of labels

Goal: transform the image labels to match the network output (vector of probabilities):

- Image labels: ordered set **integers**.
- Network output: **vector of float** in the interval [0;1] calculated by the *softmax* activation of the output neurons.
- *One-hot* coding of an ordered set \mathcal{L} of N labels:
 - each label value is coded as a vector with N components all zero except one (equal to 1),
 - the rank of the 1 in the vector gives the value of the label.

Computing aspects [Classification]

One-hot coding of labels

Goal: transform the image labels to match the network output (vector of probabilities):

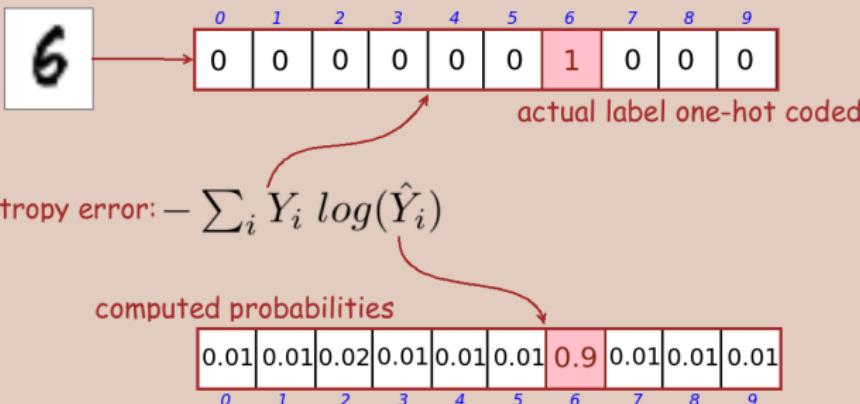
- Image labels: ordered set **integers**.
- Network output: **vector of float** in the interval [0;1] calculated by the *softmax* activation of the output neurons.

Label	One-hot vector	
0	[1 0 0 0 0 0 0 0 0]	an ordered set \mathcal{L} of N labels: is coded as a vector with N components all zero
1	[0 1 0 0 0 0 0 0 0]	to 1), in the vector gives the value of the label.
2	[0 0 1 0 0 0 0 0 0]	
3	[0 0 0 1 0 0 0 0 0]	
4	[0 0 0 0 1 0 0 0 0]	
5	[0 0 0 0 0 1 0 0 0]	The <i>one-hot</i> encoding of the labels '0' to '9'
6	[0 0 0 0 0 0 1 0 0]	gives a vector with 10 components.
7	[0 0 0 0 0 0 0 1 0]	

Computing aspects [Classification]

Error function: *Cross entropy error*

- The image analysed by the network \leadsto vector \hat{Y} of float to be compared to the vector Y of the *hot-one* encoding of the true image label.
- The error (loss) function *cross entropy* is adapted to *one-hot* coding: $e(Y, \hat{Y}) = -\sum_i Y_i \log(\hat{Y}_i)$



Computing aspects

Optimization and *Back Propagation*

- During the training an optimization algorithm computes the gradient of the error function with respect to the network weights.

Computing aspects

Optimization and *Back Propagation*

- During the training an optimization algorithm computes the gradient of the error function with respect to the network weights.
- The *Back Propagation* algorithm **modifies** the weights of the network layer by layer thanks to the gradient of the error function, iterating from the last layer to the first layer.

Computing aspects

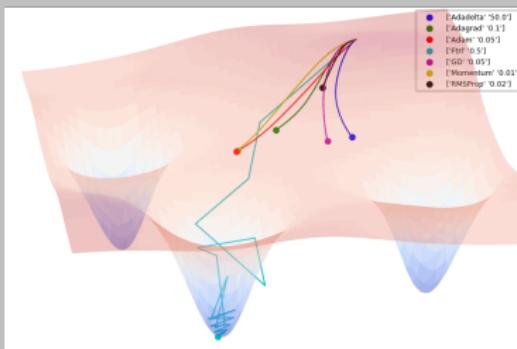
Optimization and *Back Propagation*

- During the training an optimization algorithm computes the gradient of the error function with respect to the network weights.
- The *Back Propagation* algorithm **modifies** the weights of the network layer by layer thanks to the gradient of the error function, iterating from the last layer to the first layer.
- Examples of optimization algorithms:
 - Gradient Descent (*Gradient Descent (GD)*)
 - Stochastic Gradient Descent (*Stochastic Gradient Descent (SGD)*)
 - *Adam* (improved version of gradient descent)...

The module `tf.keras.optimizers` provides Python implementation of several optimization algorithms.

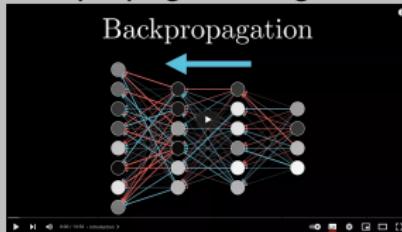
Computing aspects

Visualization of iterations of gradient descent algorithms for a very-simple loss function with only 2 variables:



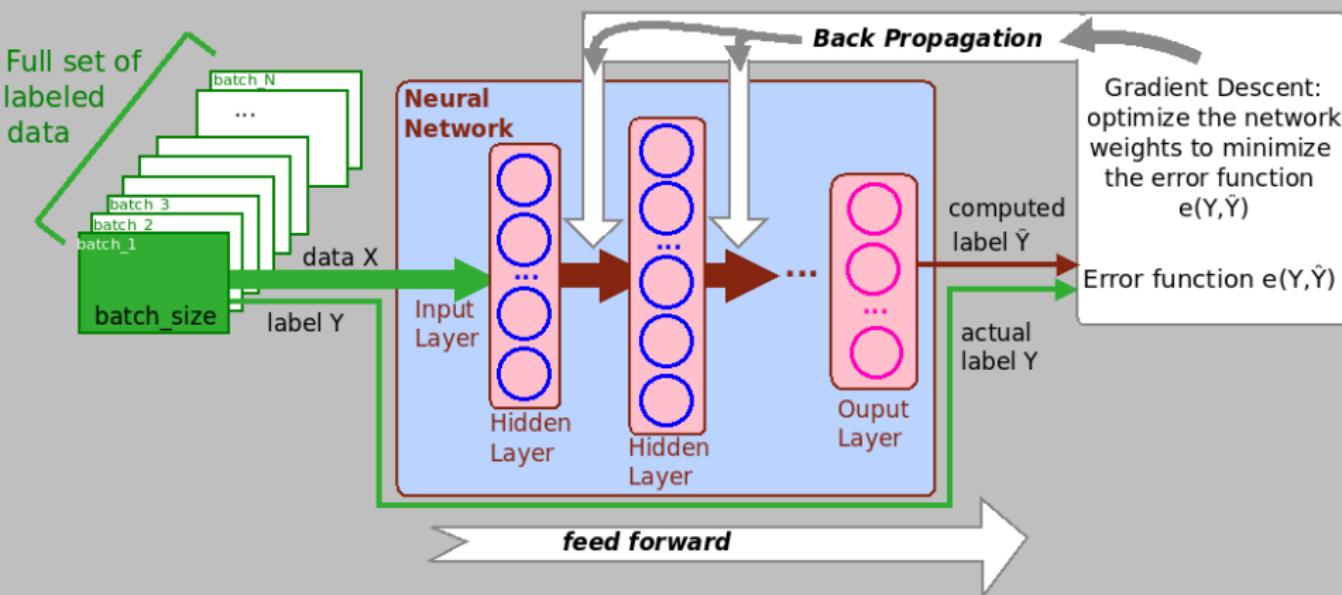
(source: github.com/Jaewan-Yun/optimizer-visualization)

Video explaining the *back propagation* algorithm:



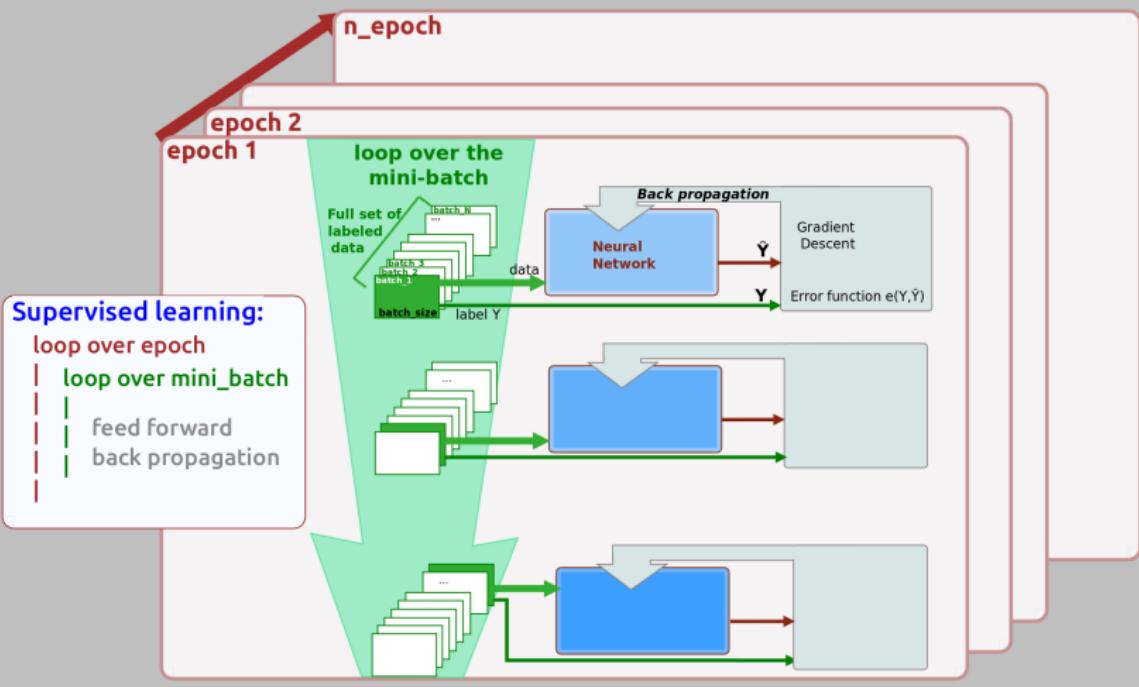
Computing aspects

Supervised learning : Feed Forward and Back Propagation



- The dataset is split into (mini) **batches** of size **batch_size**
- After each *feed forward* the *Back Propagation* algorithm modifies the weights neurons to minimize the error e .

Computing aspects



- Training with the whole dataset is repeated n_{epoch} times,
 - The network state at the end of epoch n becomes the initial state for epoch $n+1$

Why use PVE to develop ML programs with Python

PVE Advantages

- To create a dedicated environment (disk tree) with fixed version of the Python interpreter and sensitive modules (like tensorflow)
- Easy to create and destroy as many times as you want
- To Protect your projects against operating system updates or hazardous manipulations...
~ you can load/update modules within a PVE without breaking modules compatibility for the other projects
- Each Python project should have its own PVE...

Disadvantages

- ? (just do it)