Deep Learning et Réseaux de Neurones

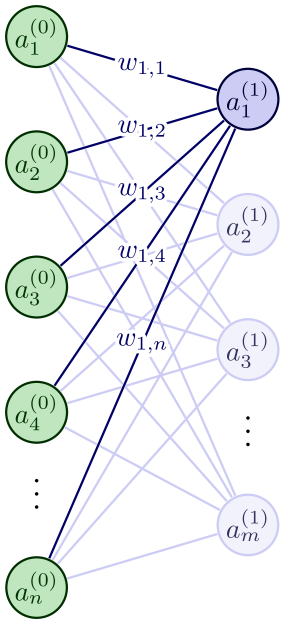
Marie Beurton-Aimar, LE Van Linh

September 17, 2025

Multi-Layers Architecture

Neuron Unit

- A weight
- An activation function
- An output value



$$= \sigma\left(w_{1,1}a_1^{(0)} + w_{1,2}a_2^{(0)} + \ldots + w_{1,n}a_n^{(0)} + b_1^{(0)}\right)$$

$$= \sigma\left(\sum_{i=1}^{n} w_{1,i}a_i^{(0)} + b_1^{(0)}\right)$$

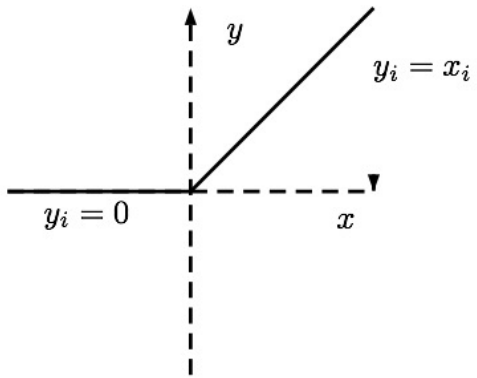
$$\begin{pmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_m^{(1)} \end{pmatrix} = \sigma\left[\begin{pmatrix} w_{1,1} & w_{1,2} & \ldots & w_{1,n} \\ w_{2,1} & w_{2,2} & \ldots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \ldots & w_{m,n} \end{pmatrix}\begin{pmatrix} a_1^{(0)} \\ a_2^{(0)} \\ \vdots \\ a_n^{(0)} \end{pmatrix} + \begin{pmatrix} b_1^{(0)} \\ b_2^{(0)} \\ \vdots \\ b_m^{(0)} \end{pmatrix}\right]$$

$$\mathbf{a}^{(1)} = \sigma\left(\mathbf{W}^{(0)}\mathbf{a}^{(0)} + \mathbf{b}^{(0)}\right)$$

Activation Functions

Rectified Linear Unit

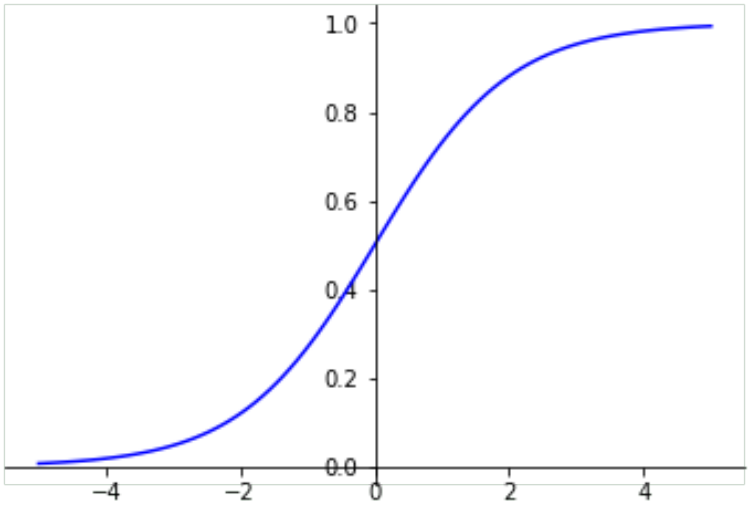
- The most used function in Deep Learning.
- Apply a filter to the layer output by keeping only positive value.
- Relu_Function = max(0,x) - Consequently the negative values are replaced by 0.
- The maximum value can be fixed by user, when output value are greater than the max they are replaced by it. This limits the output value range.



Activation Functions

Sigmoid

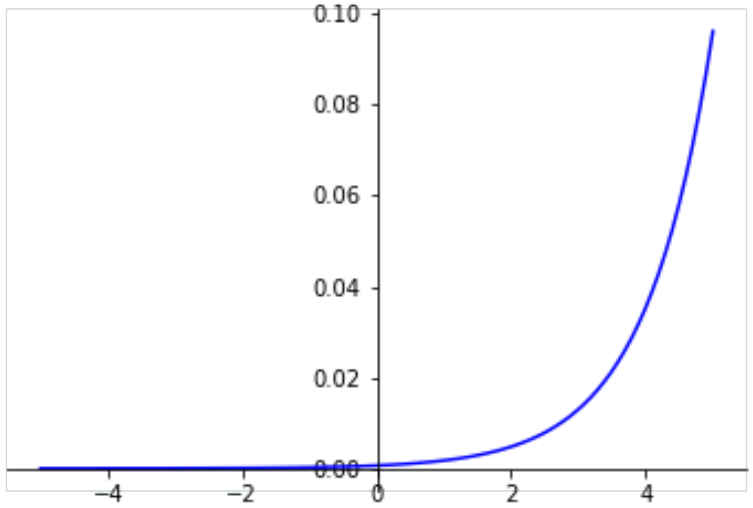
- Used by the last network layer to apply the binary classification task.
- Equation : Sigmoid_function(x) = 1 / (1 + exp(-x))
- Can be view as the probability to belong to one class.
- In the case of a binary classification, the sum of all values is equal to 1.



Activation Functions

Softmax

- Also used to do a multi-class classification task.
- Can be used at the last layer level.
- Each output is between 0 and 1, the sum is equal to 1.
- Softmax_function(x) = exp(x)/sum(exp(x_i))



Activation Functions

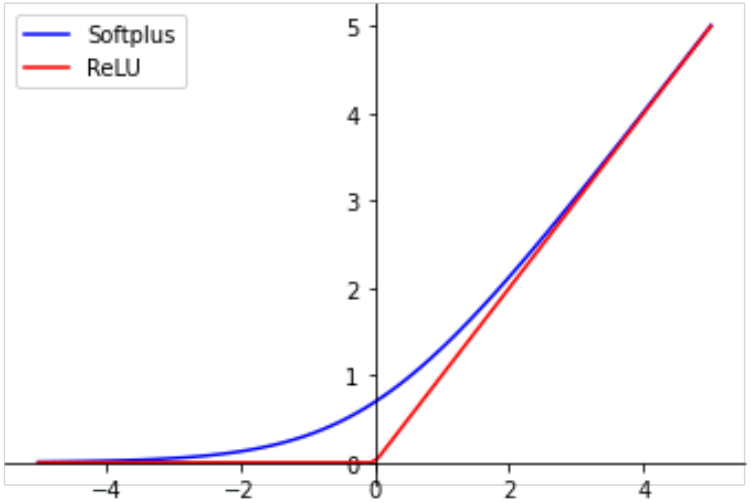
Softmax

- Results can be considered as probability to belong to a class.
- Each value is associated to a class.
- Sigmoid can't be used for the last layer into multi-class classification because in that case the sum is not equal to 0. Softmax has to be used in that case.

Activation Functions

Softplus

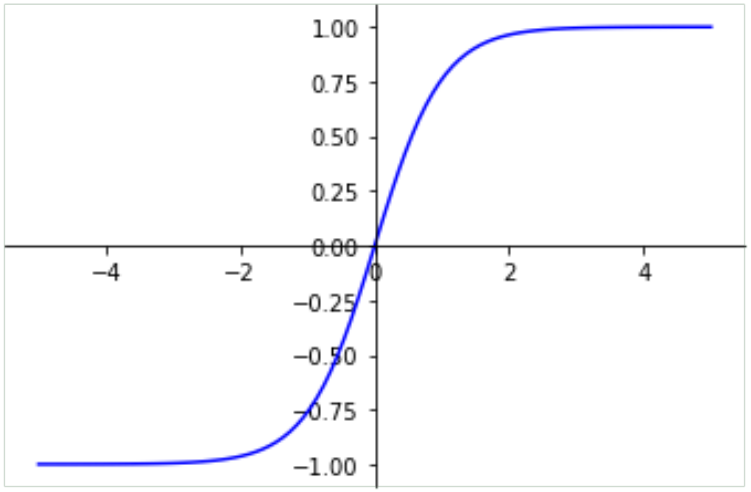
- Soft approximation of ReLU function.
- If the input values are positive, the behavior is the same than ReLU.
- Difference with ReLU: no filter for negative value but values tend toward 0.
- Softplus_function (x) = log(exp(x) + 1)



Activation Functions

tanh

- Normalize the input values.
- Can replace Sigmoid function in binary classification task in the last layer.
- Range result values is -1 to 1, it can't be considered as a probability value .
- tanh_function(x) = ((exp(x) - exp (-x)) / (exp(x) + exp(-x))).

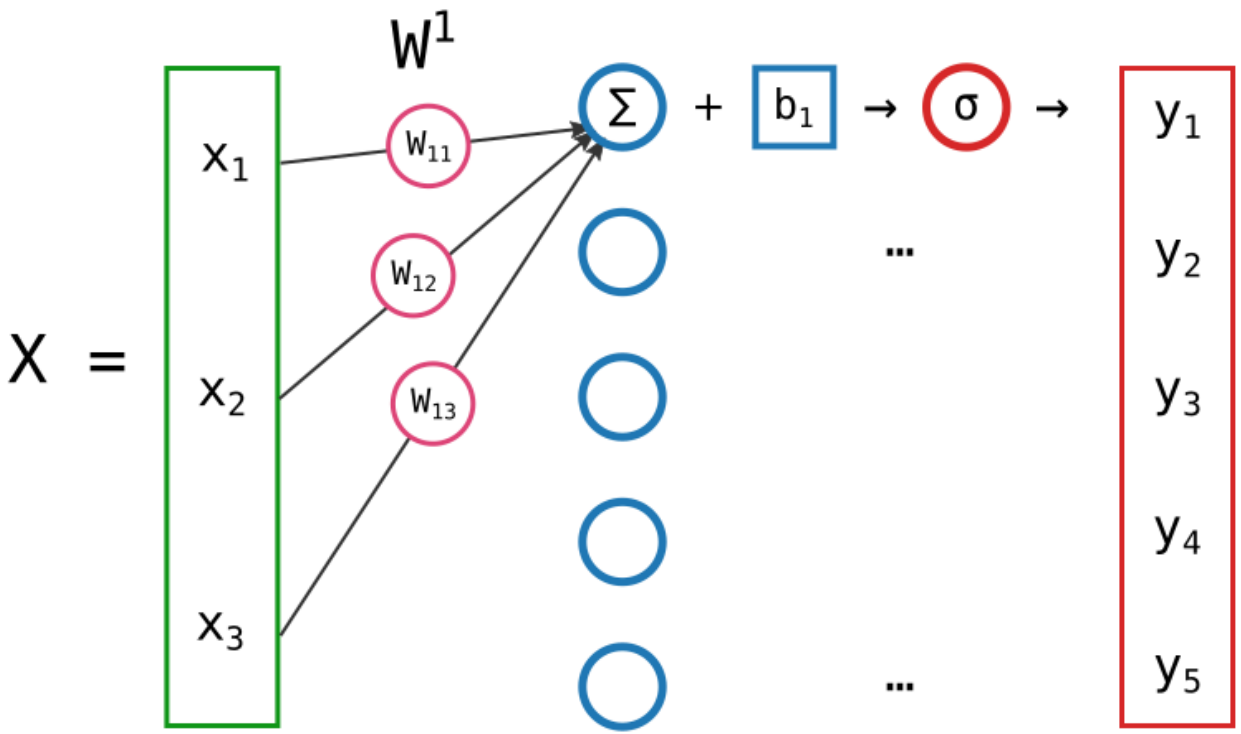


Activation Functions

Resume

- It exists a lot of another activation functions and more it is possible to design a new specific one for a purpose.
- Activation functions belong to the core of deep learning models.
- They apply non-linear transformations to data and allow to modify their spatial representation.
- Knowledge about their behavior is essential in deep learning design. Performances could be undermined by a wrong choice.

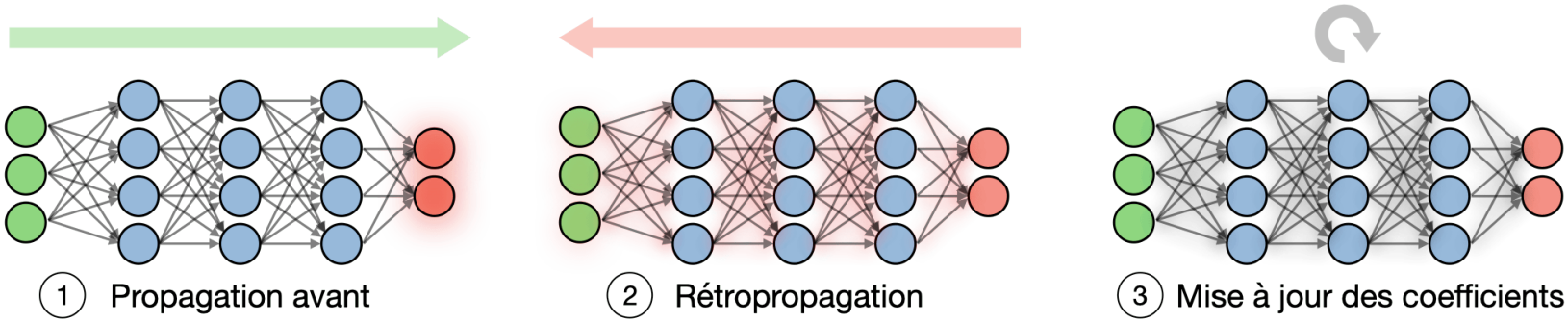
Forward Propagation



Learning Resume

3 steps

1. Take the training data and apply forward propagation
2. Back propagate the loss to obtain the loss gradient of each coefficient.
3. Use the gradient to update the network coefficients.



Backpropagation

Goal

- Is a learning algorithm to improve the outputs.
- Correct the error by using the loss function.
- The loss function mesures the error level at each epoch in order to tell to the next one performances of the current weight values.
- To do it, gradient descent is computed and weights are adjusted to run the next epoch.
- Gradient descent with momentum, with a memory of previous gradient values, can be used to accelerate algorithm convergence.

Backward Propagation

Loss Functions

- Goal : minimize the error rates between prediction and groundtruth.
- Depending on data type, different functions have to be chosen.
- The result is after analyzed by gradient descent algorithm.
- Several optimizations are/can be done.

Break !

Tensor ! What is it ?

- Data structure essential in Deep learning.
- Can be viewed as multi-dimensional matrix.
- In PyTorch they are used for everything: input values, weights and bias, output results.
- All current matrix operations are available for tensors.
- Take care about the exponential explosion with the increase of dimensions.

Loss Functions

Which type ?

- Regression function : to predict continuous results from independant variables.
- Classification function: to provide discret labels corresponding to a class.

Methods

- Regression : mean quadratic error (L2) mean absolute error (MAE). Idea : minimize the square error between predicted and target values. Allows to av
- Classification : Cross-entropy, log

Loss Functions

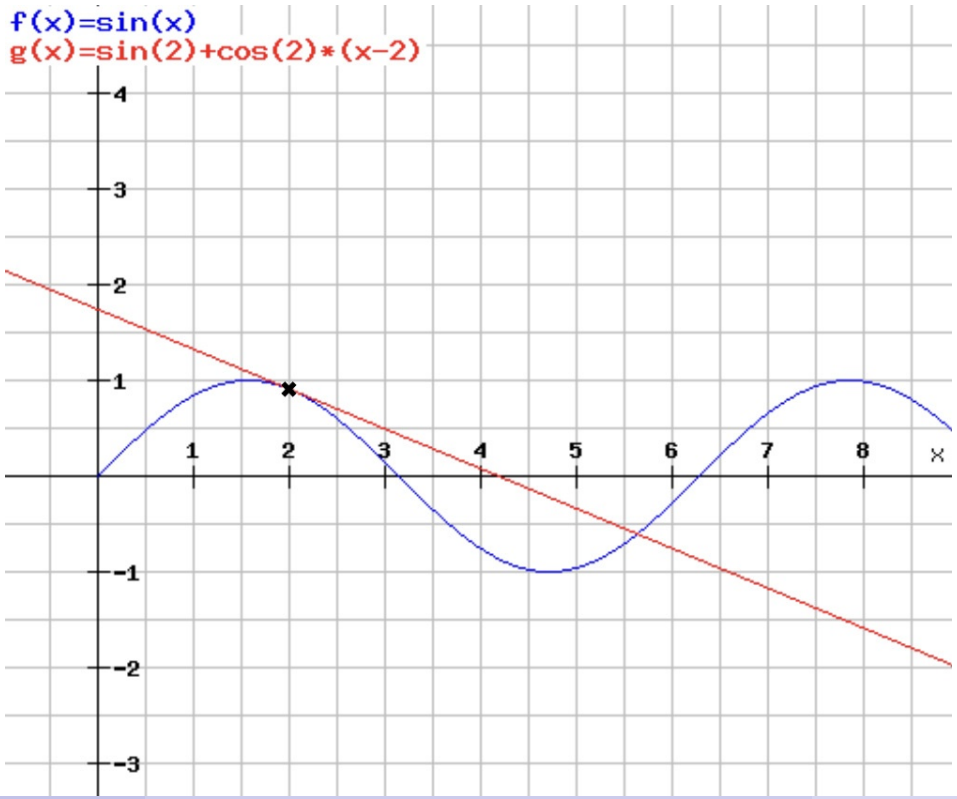
Performance evaluation

- Mean Square Error : Allows to penalize the outlier values.
- Cross entropie/log loss : computing from the difference between the two distributions values.
  Equation : L(y,f(x)) = -[y * log(f(x)) + (1-y)* log(1-f(x))]
  where y is the binary value 0,1 of the label and f(x) the probability to belong to the positive class.
  Penalize the wrong predictions - with a significative difference value from teh positive class.
-

Gradient Descent

Methods

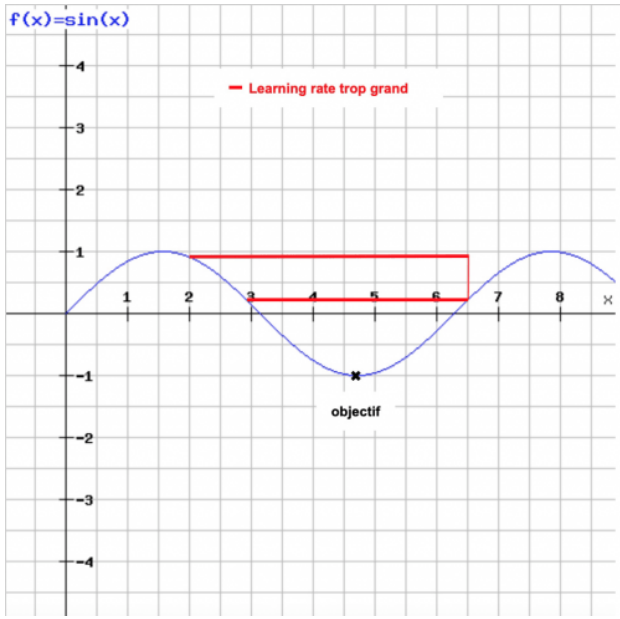
- Could be consider as the derivative function for tensors.
- Used to compute the evolution of f(x).
- Chosing the slope direction.

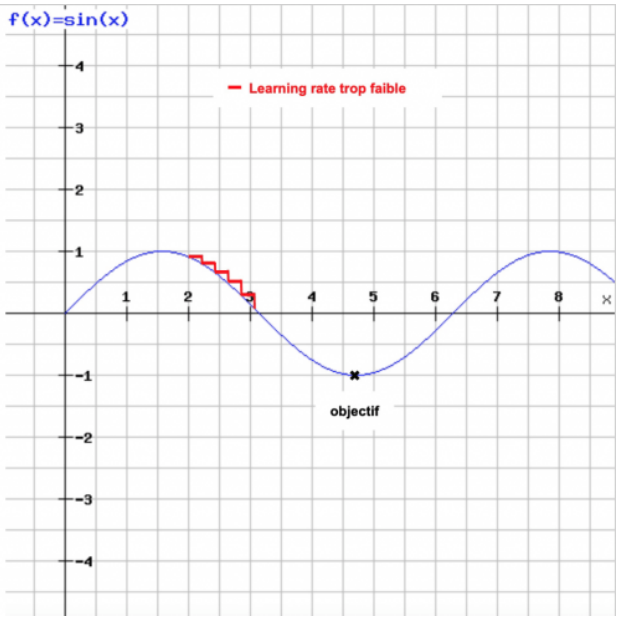


Gradient Descent

Parameters

- Choose the learning rate : distance value to move on the slope.
  - if it is too small the time to compute will be enlarge and you can fall into a local minimum.
  - if it is too large the update can move you to an unreliable place on the slope





Gradient Descent

Parameters

- Choose the iteration number : how many times you move x on the slope.
  - if it is too small the descent will not have enough step to be realized and no optimisation.
  - if it is too large your computing time will be enlarge. In deep learning it is important to keep your time.

Gradient Descent

Backpropagation again !

- Applying the chain rule (https://en.wikipedia.org/wiki/Chain_rule):
- Gradient is computed for all weights and for all layers from the last to the first one.
- The goal is to compute the impact of each parameter and to modify it consequently.
- The loss is minimized through all iterations.

Network Performances

Methods

- Mean Absolute Error : mean sum of residual absolute values.
  To be used to mesure regression results. Several versions of this error are available. Can be used to check over-fitting.
- Accuracy : Acc = number of correct prediction/number of inputs.
  used for classification.
- Precision :
  Prec = number of True Positive/ (number of True Positive + number of False Positive)
- Recall :
  Rec = number of True Positive/ (number of True Positive + number of False Negative)
- F1-score : average between precision and recall
  F1 = (Prec * Rec) / (Prec + Rec)
-

Data Set

Choose the right sample

- Divide the data set into several parts :
  - Test set - exclude 10 % of the whole set and keep it aside.
  - Training set - most often 80% of the rest.
  - Validation set - 20 % that remain.

Usage

- Test set has to be never seen at the training step.
- At each epoch training set is entirely seen. Inputs can be load in several steps depending the batch size.
- At the end of the epoch, the validation set is seen to measure the current network configuration performances.
- When the model is fixed, considered as trained, the test set is load to veryfy the capability to generalize the treatment.