

A deep mathematical understanding of DNNs

Jiang J.

Data Engineer, Data Scientist

ENSEEIH Computer Science Engineering Degree, INP Toulouse Dual MSc Research Degree in AI / Big Data / Ops
France

1 – Introduction

Frameworks such as TensorFlow or PyTorch make deep learning developments easy. They have made this field wide spread for every enthusiast. Implementations only needs an instinctive understanding of deep learning. The proper math aspect is little by little forgotten.

The objective is to do a summary of the important propositions of dense neural networks (DNN). These propositions will be mathematically proven. The subject used as reference is a multi-class classification problem with – dense layers, *ReLU* and *SoftMax* activation layers, Categorical cross-entropy loss, and Stochastic gradient descent optimizer. But all the elements below can be easily re-used or re-defined to cover regressions.

2 – Notation and Nomenclature

Definition – Let Ω a non-empty open subset of \mathcal{R}^m , $\|\cdot\|_m$ a norm on \mathcal{R}^m , $\|\cdot\|_{m'}$ a norm on $\mathcal{R}^{m'}$, and $f: \Omega \rightarrow \mathcal{R}^{m'}$. Then f continuous function is equivalent to

$$\forall a \in \Omega,$$

$$\forall \epsilon > 0, \exists \eta > 0, \forall x \in \Omega, \|x - a\|_m \leq \eta \Rightarrow \|f(x) - f(a)\|_{m'} \leq \epsilon$$

The notation $\mathcal{C}(\Omega, \mathcal{R}^{m'})$ means the set of continuous functions from Ω to $\mathcal{R}^{m'}$.

The notation $\mathcal{C}(\mathcal{R}^m)$ means the set of continuous functions from \mathcal{R}^m to $\mathcal{R}^{m'}$.

Definition – Let Ω a non-empty open subset of \mathcal{R}^m , $\|\cdot\|_m$ a norm on \mathcal{R}^m , $\|\cdot\|_{m'}$ a norm on $\mathcal{R}^{m'}$, and $f: \Omega \rightarrow \mathcal{R}^{m'}$. Then f derivable is equivalent to

$$\forall a \in \Omega, \exists f'(a) \in \mathcal{R}^{m'},$$

$$\forall \epsilon > 0, \exists \eta > 0, \forall x \in \Omega, \|x - a\|_m \leq \eta \Rightarrow \left\| \frac{f(x) - f(a)}{\|x - a\|_m} - f'(a) \right\|_{m'} \leq \epsilon$$

The notation $D(\Omega, \mathfrak{R}^{m'})$ means the set of derivable functions from Ω to $\mathfrak{R}^{m'}$.

The notation $D(\mathfrak{R}^m)$ means the set of derivable functions from \mathfrak{R}^m to \mathfrak{R}^m .

Definition – Let Ω a non-empty open subset of \mathfrak{R}^m , $\|\cdot\|_m$ a norm on \mathfrak{R}^m , $\|\cdot\|_{m'}$ a norm on $\mathfrak{R}^{m'}$, and $f: \Omega \rightarrow \mathfrak{R}^{m'}$. Then f piece-wise derivable is equivalent to

$\forall K \subset \Omega$ such as K compact and bounded,

$$\exists (K_i)_{i \in \llbracket 0, n \rrbracket} \text{ non-empty open subsets such as } \bigcup_{i=0}^n \overline{K_i} = K \text{ and } \forall (i, i') \in \llbracket 0, n \rrbracket^2, \\ i \neq i' \Rightarrow K_i \cap K_{i'} = \emptyset,$$

$$\forall i \in \llbracket 0, n \rrbracket, \exists f_i \in D^0(\overline{K_i}, \mathfrak{R}^{m'}), \forall x \in K_i, f_i(x) = f(x)$$

The notation $D_{pw}(\Omega, \mathfrak{R}^{m'})$ means the set of piece-wise derivable functions from Ω to $\mathfrak{R}^{m'}$.

The notation $D_{pw}(\mathfrak{R}^m)$ means the set of piece-wise derivable functions from \mathfrak{R}^m to \mathfrak{R}^m .

Theorem – $D(\mathfrak{R}^m) \subset D_{pw}(\mathfrak{R}^m) \subset \zeta(\mathfrak{R}^m)$

Proof: TO DO.

Definition – Let $a_{i,j} \in \mathfrak{R}$ for $i \in \llbracket 1, n \rrbracket$ and $j \in \llbracket 1, m \rrbracket$. Then the ordered rectangular array

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m} \end{bmatrix}$$

is a real matrix of dimension $n \times m$.

The following notations are considered

$$\forall i \in \llbracket 1, n \rrbracket, \forall j \in \llbracket 1, m \rrbracket, A_{i,j} = a_{i,j}$$

$$\forall j \in \llbracket 1, m \rrbracket, A_{:,j} = \begin{bmatrix} a_{1,j} \\ a_{2,j} \\ \vdots \\ a_{n,j} \end{bmatrix}$$

$$\forall i \in \llbracket 1, n \rrbracket, A_{i,:} = [a_{i,1} \ a_{i,2} \ \cdots \ a_{i,n}]$$

The notation $M_{n,m}$ means the matrix set of dimension $n \times m$ with coefficients in \mathfrak{R} .

The notation $M_{n,m}(E)$ means the matrix set of dimension $n \times m$ with coefficients in $E \subseteq \mathfrak{R}$.

Convention – A vector is a matrix with only one row. Thus, the real vector set \mathfrak{R}^m is equivalent to $M_{1,m}$.

Notation – The matrix transpose operation will be noted as A^T .

Definition – Let $A \in M_{n,m}$, and $B \in M_{m,p}$, and let the product noted $A \times B$ or AB be

$$C = A \times B = AB$$

where C is a $n \times p$ matrix with

$$\forall i \in \llbracket 1, n \rrbracket, \forall j \in \llbracket 1, p \rrbracket, C_{i,j} = \sum_{k=1}^m A_{i,k} \times B_{k,j}$$

Definition – Let $a \in \mathfrak{R}$ and $B \in M_{n,m}$. Let the scalar wise product noted as $a \times B$ be

$$C = a \times B = B \times a$$

where C is in $M_{n,m}$ with

$$\forall i \in \llbracket 1, n \rrbracket, \forall j \in \llbracket 1, m \rrbracket, C_{i,j} = a \times B_{i,j}$$

Theorem – Let U and V non-empty open subsets of \mathfrak{R}^n and \mathfrak{R}^m . Let $f: \begin{cases} U \rightarrow V \\ x \mapsto f(x) \end{cases}$ and $g: \begin{cases} V \rightarrow \mathfrak{R}^p \\ y \mapsto g(y) \end{cases}$ such as $f \in D(U, V)$ and $g \in D(V, \mathfrak{R}^p)$. Then $g \circ f: \begin{cases} U \rightarrow \mathfrak{R}^p \\ x \mapsto g(f(x)) \end{cases}$ is in $D(U, \mathfrak{R}^p)$ and its Jacobian is

$$\frac{d(g \circ f)}{dx}: \begin{cases} U \rightarrow M_{p,n} \\ x \mapsto \frac{dg}{dy}(f(x)) \times \frac{df}{dx}(x) \end{cases}$$

Proof: TO DO.

3 – Activation functions

Definition – Let $F_{act} \in D(\mathfrak{R}^m)$. Then the vector wise application

$$F_{act} : \begin{cases} \mathfrak{R}^m \rightarrow \mathfrak{R}^m \\ z \mapsto f(z) \end{cases}$$

is an activation function.

Definition – $ReLU$ is the following vector wise application

$$ReLU : \begin{cases} \mathfrak{R}^m \rightarrow \mathfrak{R}^m \\ z \mapsto \max(0, z) \end{cases}$$

with \max the element-wise maximum operation between two vectors.

Hypothesis – The notation $ReLU_j$ means the application corresponding to the coefficient j of the function $ReLU$. Let $z \in \mathfrak{R}^m$ then

$$\forall j \in \llbracket 1, m \rrbracket, \quad ReLU_j(z_j) = \max(0, z_j) = ReLU(z)_j$$

$ReLU$ is supposed derivable on every coefficients at 0

$$\forall j \in \llbracket 1, m \rrbracket, \quad ReLU_j'(0) = 0$$

Proposition – $ReLU$ is an activation function. Its Jacobian matrix is

$$\frac{d ReLU}{dz} : \begin{cases} \mathfrak{R}^m \rightarrow M_{m,m} \\ z \mapsto \begin{bmatrix} 1_{\mathfrak{R}_{\setminus\{0\}}^+}(z_1) & 0 & \cdots & 0 \\ 0 & 1_{\mathfrak{R}_{\setminus\{0\}}^+}(z_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1_{\mathfrak{R}_{\setminus\{0\}}^+}(z_m) \end{bmatrix} \end{cases}$$

with $1_{\mathfrak{R}_{\setminus\{0\}}^+}$ the $\mathfrak{R}_{\setminus\{0\}}^+$ indicator function on \mathfrak{R} .

Proof: TO DO.

Proposition – The following vector wise application is an activation function

$$\text{SoftMax} : \begin{cases} \mathfrak{R}^m \rightarrow]0,1[^m \\ z \mapsto \frac{e^{z_j}}{\sum_{j'=1}^m e^{z_{j'}}} \end{cases}$$

with e the element-wise exponential operation.

The *SoftMax* function will be denoted as S for simplicity.

Its Jacobian matrix is

$$\frac{dS}{dz} : \begin{cases} \mathfrak{R}^m \rightarrow M_{m,m} \\ z \mapsto \frac{dS}{dz}(z) \end{cases}$$

where $\forall z \in \mathfrak{R}^m$, $\forall (j, j') \in \{1, 2, \dots, m\}^2$,

$$\frac{dS}{dz}(z)_{j,j'} = S(z)_j \times (\delta_{j,j'} - S(z)_{j'})$$

with $\delta_{j,j'}$ the Kronecker delta.

Proof: TO DO.

3 – Loss

Definition – Let $\hat{\Omega} \in M_{n,m}$ and $\Omega \subseteq M_{n,m}$ non empty subsets. Let $\hat{y} \in \hat{\Omega}$ and $F_{\text{loss}}^{\hat{y}} \in D(\Omega, \mathfrak{R})$. Then $F_{\text{loss}}^{\hat{y}}$ is a loss function is equivalent to the application

$$F_{\text{loss}}^{\hat{y}} \circ g : \begin{cases} E \rightarrow \mathfrak{R} \\ \epsilon \mapsto (F_{\text{loss}}^{\hat{y}} \circ g)(\epsilon) = F_{\text{loss}}^{\hat{y}}(\hat{y} + \epsilon) \end{cases}$$

is an increasing function according each coefficient with $E \subseteq M_{n,m}$ such as $F_{\text{loss}}^{\hat{y}} \circ g$ is always defined.

The \hat{y} matrix is named the ground truth.

Proposition – Let $\hat{y} \in \{0,1\}^m$ a ground truth matrix. Then the application

$$\xi^{\hat{y}}: \begin{cases}]0,1[^m \rightarrow \mathbb{R} \\ y \mapsto -\sum_{j=1}^m \hat{y}_j \log(y_j) \end{cases}$$

is a loss function. The application is named Categorical cross-entropy loss.

Its Gradient matrix is

$$\frac{d\xi^{\hat{y}}}{dy}: \begin{cases}]0,1[^m \rightarrow \mathbb{R}^m \\ y \mapsto -\begin{bmatrix} \hat{y}_1 & \dots & \hat{y}_m \\ y_1 & & y_m \end{bmatrix} \end{cases}$$

Proof: TO DO.

Proposition – Let $\hat{y} \in]0,1[^m$ a ground truth matrix. Let $S: \mathbb{R}^m \rightarrow]0,1[^m$ and $\xi^{\hat{y}}:]0,1[^m \rightarrow \mathbb{R}$ the *SoftMax* activation and Categorical cross-entropy loss functions. Then $\xi^{\hat{y}} \circ S: \mathbb{R}^m \rightarrow \mathbb{R}$ is derivable on \mathbb{R}^m and its Gradient matrix is

$$\frac{d(\xi^{\hat{y}} \circ S)}{dz}: \begin{cases} \mathbb{R}^m \rightarrow \mathbb{R}^m \\ z \mapsto S(z) - \hat{y} \end{cases}$$

Proof: TO DO.

4 – Dense layers

Definition – The application

$$L: \begin{cases} \mathbb{R}^m \times M_{m',m} \times \mathbb{R}^{m'} \rightarrow \mathbb{R}^{m'} \\ (y, W, b) \mapsto y \times W^T + b \end{cases}$$

defines a dense layer with y named the input vector, W named the weight matrix and b named the bias matrix.

The notation L_j means the application $L_j: \mathbb{R}^m \rightarrow \mathbb{R}^{m'}$ corresponding to the coefficient j of the dense layer L . Let $y \in \mathbb{R}^m$ an input vector, $W \in M_{m',m}$ a weight matrix and $b \in \mathbb{R}^{m'}$ a bias

matrix then

$$\forall j \in \llbracket 1, m' \rrbracket, \quad L_j(W_{j,:}) = y \times (W_{j,:})^T + b_j = L(y, W, b)_j$$

Proposition – Let $L: \mathfrak{R}^m \times M_{m',m} \times \mathfrak{R}^{m'} \rightarrow \mathfrak{R}^{m'}$ the dense layer function. Then L is derivable according the first and third variables on \mathfrak{R}^m and $\mathfrak{R}^{m'}$ respectively.

Let $y \in \mathfrak{R}^m$ an input vector and $b \in \mathfrak{R}^{m'}$ a bias matrix. Then $L_{j,:}: \mathfrak{R}^m \rightarrow \mathfrak{R}$ is also derivable for all coefficient $j \in \llbracket 1, m' \rrbracket$.

Its Jacobian matrices are

$$\begin{aligned} \frac{\partial L}{\partial y} &: \begin{cases} \mathfrak{R}^m \rightarrow M_{m',m} \\ y \mapsto W \end{cases} \\ \forall j \in \llbracket 1, m' \rrbracket, \quad \frac{dL_j}{dw} &: \begin{cases} \mathfrak{R}^m \rightarrow \mathfrak{R}^m \\ w \mapsto y \end{cases} \\ \frac{\partial L}{\partial b} &: \begin{cases} \mathfrak{R}^{m'} \rightarrow M_{m',m'} \\ b \mapsto I_{m'} \end{cases} \end{aligned}$$

with $I_{m'}$ the identity matrix of size $m' \times m'$.

Proof: TO DO.

Proposition – Let $L: \mathfrak{R}^m \times M_{m',m} \times \mathfrak{R}^{m'} \rightarrow \mathfrak{R}^{m'}$ and $ReLU: \mathfrak{R}^{m'} \rightarrow \mathfrak{R}^{m'}$ the dense layer and

$ReLU$ activation functions. Let $F^{upstream}: \begin{cases} \mathfrak{R}^{m'} \rightarrow \mathfrak{R} \\ y' \mapsto F^{upstream}(y') \end{cases}$ such as $F^{upstream} \in D(\mathfrak{R}^{m'}, \mathfrak{R})$.

Then $F^{upstream} \circ ReLU \circ L: \mathfrak{R}^m \times M_{m',m} \times \mathfrak{R}^{m'} \rightarrow \mathfrak{R}$ is derivable according the first and third variables on \mathfrak{R}^m and $\mathfrak{R}^{m'}$ respectively.

The notation $F_{j'}^{upstream}$ means the application corresponding to the coefficient j' of $F^{upstream}$.

Let $y' \in \mathfrak{R}^{m'}$ then

$$\forall j' \in \llbracket 1, m' \rrbracket, \quad F_{j'}^{upstream}(y'_j) = F^{upstream}(y')_j,$$

Let $y \in \mathfrak{R}^m$ an input vector and $b \in \mathfrak{R}^{m'}$ a bias matrix. Then $F_{j'}^{upstream} \circ ReLU_{j'} \circ L_{j'} : \mathfrak{R} \rightarrow \mathfrak{R}$ is also derivable for all coefficient $j' \in \llbracket 1, m' \rrbracket$.

Its Gradient matrices are

$$\frac{\partial (F_{j'}^{upstream} \circ ReLU \circ L(\cdot, W, b))}{\partial y} : \left\{ y \mapsto \frac{\partial (F_{j'}^{upstream} \circ ReLU \circ L(\cdot, W, b))}{\partial y}(y) \right.$$

where $\forall y \in \mathfrak{R}^m$, $\forall j \in \llbracket 1, m \rrbracket$,

$$\frac{\partial (F_{j'}^{upstream} \circ ReLU \circ L(\cdot, W, b))}{\partial y}(y)_j = \sum_{j'=1}^{m'} \frac{d F_{j'}^{upstream}}{d y'}(ReLU(L(y, W, b)))_{j'} \times 1_{\mathfrak{R}_{\setminus \{0\}}^+}(L(y, W, b)_{j'}) \times W_{j', j}$$

with $W \in M_{m', m}$ a weight matrix, $b \in \mathfrak{R}^{m'}$ a bias matrix and $1_{\mathfrak{R}_{\setminus \{0\}}^+}$ the $\mathfrak{R}_{\setminus \{0\}}^+$ indicator function on \mathfrak{R} .

$$\forall j' \in \llbracket 1, m' \rrbracket, \quad \frac{d (F_{j'}^{upstream} \circ ReLU_{j'} \circ L_{j'})}{d w} : \left\{ w \mapsto \frac{d (F_{j'}^{upstream} \circ ReLU_{j'} \circ L_{j'})}{d w}(w) \right.$$

where $\forall w \in \mathfrak{R}^m$, $\forall j \in \llbracket 1, m \rrbracket$,

$$\frac{d (F_{j'}^{upstream} \circ ReLU_{j'} \circ L_{j'})}{d w}(w)_j = F_{j'}^{upstream}{}'(ReLU_{j'}(L_{j'}(w))) \times 1_{\mathfrak{R}_{\setminus \{0\}}^+}(L_{j'}(w)) \times y_j$$

with $y \in \mathfrak{R}^m$ an input matrix, $b \in \mathfrak{R}^{m'}$ a bias matrix and $1_{\mathfrak{R}_{\setminus \{0\}}^+}$ the $\mathfrak{R}_{\setminus \{0\}}^+$ indicator function on \mathfrak{R} .

$$\frac{\partial (F_{j'}^{upstream} \circ ReLU \circ L(y, W, \cdot))}{\partial b} : \left\{ b \mapsto \frac{\partial (F_{j'}^{upstream} \circ ReLU \circ L(y, W, \cdot))}{\partial b}(b) \right.$$

where $\forall b \in \mathfrak{R}^{m'}$, $\forall j' \in \llbracket 1, m' \rrbracket$,

$$\frac{\partial (F_{j'}^{upstream} \circ ReLU \circ L(y, W, \cdot))}{\partial b}(b)_{j'} = \frac{d F_{j'}^{upstream}}{d y'}(ReLU(L(y, W, b)))_{j'} \times 1_{\mathfrak{R}_{\setminus \{0\}}^+}(L(y, W, b)_{j'})$$

with $y \in \mathfrak{R}^m$ a weight matrix and $W \in M_{m', m}$ a weight matrix.

Proof: TO DO.

5 – Neural Networks

Definition – A training data set is defined as couples of vectors $(X^i, \hat{Y}^i) \in \mathfrak{R}^m \times \mathfrak{R}^l$ for $i \in \llbracket 1, n \rrbracket$. The X^i are named input or feature matrices and the \hat{Y}^i target or label matrices.

Definition – Let p dense layers with activation functions $F_{act}^k \circ L^k(\cdot, W^k, b^k): \mathfrak{R}^{m_k} \rightarrow \mathfrak{R}^{m_{k+1}}$ for $k \in \llbracket 1, p \rrbracket$ with $W^k \in M_{m_{k+1}, m_k}$ and $b^k \in \mathfrak{R}^{m_{k+1}}$ the L^k weight and bias matrices. Let a training data set $(X^i, \hat{Y}^i) \in \mathfrak{R}^{m_1} \times \mathfrak{R}^{m_{p+1}}$ for $i \in \llbracket 1, n \rrbracket$. Let $F_{loss}^{\hat{Y}^i}: \mathfrak{R}^{m_{p+1}} \rightarrow \mathfrak{R}$ loss functions for $i \in \llbracket 1, n \rrbracket$ with $(\hat{Y}^i)_{i \in \llbracket 1, n \rrbracket}$ as ground truth matrices respectively.

Then a neural network is defined as the application $N: \mathfrak{R}^{m_1} \rightarrow \mathfrak{R}^{m_{p+1}}$ such as

$$N = F_{act}^p \circ L^p(\cdot, W^p, b^p) \circ \dots \circ F_{act}^1 \circ L^1(\cdot, W^1, b^1).$$

The optimization problem is $\min_{(W_{1,:}^k, \dots, W_{m_{k+1},:}^k, b^k)_{k \in \llbracket 1, p \rrbracket}} \sum_{i=1}^n F_{loss}^{\hat{Y}^i}(N(X^i))$ and

$F_{loss}^{global}: ((W_{1,:}^k, \dots, W_{m_{k+1},:}^k, b^k)_{k \in \llbracket 1, p \rrbracket}) \mapsto \sum_{i=1}^n F_{loss}^{\hat{Y}^i}(N(X^i))$ is named the objective function or global loss.

Theorem – Let p dense layers with activation functions – $ReLU^k \circ L^k(\cdot, W^k, b^k): \mathfrak{R}^{m_k} \rightarrow \mathfrak{R}^{m_{k+1}}$ for $k \in \llbracket 1, p-1 \rrbracket$ and $S \circ L^p(\cdot, W^p, b^p): \mathfrak{R}^{m_p} \rightarrow \mathfrak{R}^{m_{p+1}}$. $W^k \in M_{m_{k+1}, m_k}$ and $b^k \in \mathfrak{R}^{m_{k+1}}$ are defined as the L^k weight and bias matrices. Let a training data set $(X^i, \hat{Y}^i) \in \mathfrak{R}^{m_1} \times \mathfrak{R}^{m_{p+1}}$ for $i \in \llbracket 1, n \rrbracket$. Let $\xi^{\hat{Y}^i}: \mathfrak{R}^{m_{p+1}} \rightarrow \mathfrak{R}$ Categorical cross-entropy losses for $i \in \llbracket 1, n \rrbracket$ with $(\hat{Y}^i)_{i \in \llbracket 1, n \rrbracket}$ as ground truth matrices respectively.

Then the following application $N: \mathfrak{R}^{m_1} \rightarrow \mathfrak{R}^{m_{p+1}}$ with

$$N = S \circ L^p(\cdot, W^p, b^p) \circ ReLU^{p-1} \circ L^{p-1}(\cdot, W^{p-1}, b^{p-1}) \circ \dots \circ ReLU^1 \circ L^1(\cdot, W^1, b^1)$$

is a neural network and its objective function is

$$\xi_{loss}^{global} : ((W_{1,:}^k, \dots, W_{m_{k+1},:}^k, b^k)_{k \in \llbracket 1, p \rrbracket}) \mapsto \sum_{i=1}^n \xi_{loss}^{\hat{Y}^i}(N(X^i))$$

Let $k \in \llbracket 2, p-2 \rrbracket$. Let

$$y^{downstream(k), X^i} = (ReLU^{k-1} \circ L^{k-1}(\cdot, W^{k-1}, b^{k-1}) \circ \dots \circ ReLU^1 \circ L^1(\cdot, W^1, b^1))(X^i)$$

$$F^{upstream(k), \hat{Y}^i} : \begin{cases} \mathfrak{R}^{m_{k+1}} \rightarrow \mathfrak{R} \\ y \mapsto F^{upstream(k), \hat{Y}^i}(y) \end{cases} \quad \text{with}$$

$$F^{upstream(k), \hat{Y}^i} = \xi^{\hat{Y}^i} \circ S \circ L^p(\cdot, W^p, b^p) \circ ReLU^{p-1} \circ L^{p-1}(\cdot, W^{p-1}, b^{p-1}) \circ \dots \circ ReLU^{k+1} \circ L^{k+1}(\cdot, W^{k+1}, b^{k+1})$$

Then ξ_{loss}^{global} Gradient matrices are

$$\forall j' \in \llbracket 1, m_{k+1} \rrbracket, \quad \frac{\partial \xi_{loss}^{global}}{\partial W_{j',:}^k} : \begin{cases} \mathfrak{R}^{m_k} \rightarrow \mathfrak{R}^{m_k} \\ w \mapsto \sum_{i=1}^n \frac{d(F_{j'}^{upstream(k), \hat{Y}^i} \circ ReLU_{j'}^k \circ L_{j'}^k)}{dw}(w) \end{cases}$$

where $\forall i \in \llbracket 1, n \rrbracket$, $\forall w \in \mathfrak{R}^{m_k}$, $\forall j \in \llbracket 1, m_k \rrbracket$,

$$\frac{d(F_{j'}^{upstream(k), \hat{Y}^i} \circ ReLU_{j'}^k \circ L_{j'}^k)}{dw}(w)_j = F_{j'}^{upstream(k), \hat{Y}^i}(ReLU_{j'}^k(L_{j'}^k(w))) \times 1_{\mathfrak{R}_{\setminus \{0\}}^+}(L_{j'}^k(w)) \times y_j^{downstream(k), X^i}$$

with $\mathfrak{R}_{\setminus \{0\}}^+$ indicator function on \mathfrak{R} .

$$\frac{\partial \xi_{loss}^{global}}{\partial b^k} : \begin{cases} \mathfrak{R}^{m_{k+1}} \rightarrow \mathfrak{R}^{m_{k+1}} \\ b^k \mapsto \sum_{i=1}^n \frac{\partial (F^{upstream(k), \hat{Y}^i} \circ ReLU^k \circ L^k(y^{downstream(k), X^i}, W^k, \cdot))}{\partial b^k}(b^k) \end{cases}$$

where $\forall i \in \llbracket 1, n \rrbracket$, $\forall b^k \in \mathfrak{R}^{m_k}$, $\forall j \in \llbracket 1, m_{k+1} \rrbracket$,

$$\frac{\partial (F^{upstream(k), \hat{Y}^i} \circ ReLU^k \circ L^k(y^{downstream(k), X^i}, W^k, \cdot))}{\partial b^k}(b^k)_j$$

$$= \frac{d F^{\text{upstream}(k), \hat{Y}^i}}{d y^k} (\text{ReLU}^k(L^k(y^{\text{downstream}(k), X^i}, W^k, b^k)))_j \times 1_{\mathfrak{R}^+_{\setminus \{0\}}} (L^k(y^{\text{downstream}(k), X^i}, W^k, b^k)_j)$$

Proof: TO DO.

6 – Optimizations

7 – References