A deep mathematical understanding of DNNs

Jiang J.
Data Engineer, Data Scientist
ENSEEIHT Computer Science Engineering Degree
INP Toulouse Dual MSc Research Degree in AI, Big Data and Ops
France

Abstract

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 collection of the important propositions explaining dense neural network (DNN) theories. 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.

1 – Prerequisites and Notations

1.1 – Matrices

Definition – Let $a_{i,j} \in \Re$ for $i \in [\![1,n]\!]$ and $j \in [\![1,m]\!]$. 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

$$\forall i \in [\![1,n]\!] \ , \ \forall j \in [\![1,m]\!] \ , \ A_{i,j} = a_{i,j}$$

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

$$\forall i \in [\![1,n]\!] \ , \ A_{i,:} = \begin{bmatrix} a_{i,1} & a_{i,2} & \cdots & a_{i,n} \end{bmatrix}$$

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

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

Convention – A vector is a matrix with only one row. Thus, the real vector set $\Re^m$ 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 $m \times p$ matrix with

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

Definition – Let $a \in \Re$ 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

$$\forall i \in [\![1,n]\!] \ , \ \forall j \in [\![1,m]\!] \ , \ C_{i,j} = a \times B_{i,j}$$

Notation – Let $a \in \Re^n$ and $b \in \Re^n$. Then the scalar product between two vectors is noted as

$$(a|b) = a \times b^T = b \times a^T = (b|a)$$

1.2 – Function regularity

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

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

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

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

The notation $\zeta(\Re^m)$ means the set of continuous functions from $\Re^m$ to $\Re^m$.

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

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

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

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

The notation $D(\Re^m)$ means the set of derivable functions from $\Re^m$ to $\Re^m$.

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

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

$$\exists(K_i)_{i\in[\![0,n]\!]} \text{ non-empty open subsets such as } \bigcup_{i=0}^{n}\overline{K_i}=K \text{ and } \forall(i,i')\in[\![0,n]\!]^2 \ ,$$

$$i\neq i'\Rightarrow K_i\cap K_{i'}=\emptyset \ ,$$

$$\forall i\in[\![0,n]\!] \ , \ \exists f_i\in D^0(\overline{K_i},\Re^{m'}) \ , \ \forall x\in K_i \ , \ f_i(x)=f(x)$$

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

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

Proposition – Let Ω a non-empty open subset of $\Re^m$. Then

$$D(\Omega,\Re^m)\subset D_{pw}(\Omega,\Re^m)\subset\zeta(\Omega,\Re^m)$$

Proof: TO DO.

Notation – Let $f:\Omega\to\Omega'$ and $g:\Omega'\to\Omega''$. Then the notation $g\circ f$ means the application $g\circ f:\begin{cases}\Omega\to\Omega''\\ x\mapsto g(f(x))\end{cases}$.

Let $(f_i)_{i\in[\![1,n]\!]}$ with $f_i:\Omega_i\to\Omega_{i+1}$ for $i\in[\![1,n]\!]$. Then the notation $\overset{n}{\underset{i=1}{\circ}}f_i$ means the application $\overset{n}{\underset{i=1}{\circ}}f_i:\begin{cases}\Omega_1\to\Omega_{n+1}\\ x\mapsto f_n(f_{n-1}(\dots f_2(f_1(x))))\end{cases}$.

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

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

Proof: TO DO.

1.3 – Function convexity and smoothness

Definition – Let Ω a non-empty convex open subset of $\Re^m$, $\|\cdot\|_m$ a norm on $\Re^m$, and $f:x\mapsto f(x)$ such as $f\in\zeta(\Omega,\Re)$. Then f convex on Ω is equivalent to

$$\forall(y,z)\in\Omega^2,\ \forall t\in[0,1],$$
$$f(t\times y+(1-t)\times z)\le t\times f(y)+(1-t)\times f(z)$$

Proposition – Let Ω a non-empty convex open subset of $\Re^m$, $\|\cdot\|_m$ a norm on $\Re^m$, and $f\in D(\Omega,\Re)$ convex on Ω. Then f convex on Ω is equivalent to

$$\forall(y,z)\in\Omega^2,$$
$$f(y)+\frac{df}{dx}(y)\times(z-y)^T\le f(z)$$

Proof: TO DO.

Definition – Let Ω a non-empty open subset of $\Re^m$, $_m\|\cdot\|$ a norm on $\Re^m$, and $f:x\mapsto f(x)$ such as $f\in D(\Omega,\Re)$. Let $L>0$. Then f $L-smooth$ on Ω is equivalent to

$$\forall(y,z)\in\Omega^2 \quad , \quad _m\|\frac{df}{dx}(y)-\frac{df}{dx}(z)\|\leq L\times_m\|y-z\|$$

Proposition – Let Ω a non-empty convex open subset of $\Re^m$, $_m\|\cdot\|$ a norm on $\Re^m$, and $f:x\mapsto f(x)$ such as $f\in D(\Omega,\Re)$. Then

$$\forall(y,z)\in\Omega^2 \quad ,$$

$$f(y)=f(z)+\int_0^1 \frac{df}{dx}(z+\tau(y-z))(y-z)^T d\tau$$

Proof: TO DO.

<Co-coercivity with L-smooth & convex TODO>

1.3 – Others

Notation – Let $f:\begin{cases}\Omega_1\times\ldots\times\Omega_n\to\Omega'_1\times\ldots\times\Omega'_m\\(x_1,\ldots,x_n)\mapsto f(x_1,\ldots,x_n)\end{cases}$ an application with n parameters and m outputs. Then for $k\in[\![1,n]\!]$ the notation $f(x_1,\ldots,x_{k-1},\cdot,x_{k+1},\ldots,x_n)$ means the application $f(x_1,\ldots,x_{k-1},\cdot,x_{k+1},\ldots,x_n):\begin{cases}\Omega_k\to\Omega'_1\times\ldots\times\Omega'_m\\x_k\mapsto f(x_1,\ldots,x_{k-1},x_k,x_{k+1},\ldots,x_n)\end{cases}$.

Notation – The notation $1_{\Re^+_{\setminus\{0\}}}$ means the $\Re^+_{\setminus\{0\}}$ indicator function on $\Re$.

$$1_{\Re^+_{\setminus\{0\}}}:\begin{cases}\Re^m\to\Re^m\\z\mapsto f(z)\end{cases}$$

2 – Activation functions

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

$$F_{act} : \begin{cases} \Re^m \to \Re^m \\ z \mapsto f(z) \end{cases}$$

is an activation function.

Definition – $ReLU$ is the following vector wise application

$$ReLU : \begin{cases} \Re^m \to \Re^m \\ z \mapsto max(0_{\Re^m}, z) \end{cases}$$

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 $z \in \Re^m$ then

$$\forall j \in [\![1, m]\!] \ , \ ReLU_j(z_j) = max(0, z_j) = ReLU(z)_j$$

$ReLU$ is supposed derivable on every coefficients at 0

$$\forall j \in [\![1, m]\!] \ , \ ReLU_j'(0) = 0$$

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

$$\frac{d\,ReLU}{dz} : \left\{ z \mapsto \begin{bmatrix} 1_{\Re^+_{\setminus\{0\}}}(z_1) & 0 & \cdots & 0 \\ 0 & 1_{\Re^+_{\setminus\{0\}}}(z_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1_{\Re^+_{\setminus\{0\}}}(z_m) \end{bmatrix} \right. \quad \Re^m \to M_{m,m}$$

Proof: TO DO.

Proposition – The following vector wise application is an activation function

$$SoftMax : \begin{cases} \Re^m \to]0,1[^m \\ z \mapsto e^{z_j} \Big/ \sum_{j'=1}^{m} e^{z_{j'}} \end{cases}$$

with e the element-wise exponential operation.

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

Its Jacobian matrix is

$$\frac{dS}{dz} : \begin{cases} \Re^m \to M_{m,m} \\ z \mapsto \frac{dS}{dz}(z) \end{cases}$$

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

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

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 $F_{loss}^{\hat{y}} \in D(\Omega, \Re)$. Then $F_{loss}^{\hat{y}}$ is a loss function is equivalent to the application

$$F_{loss}^{\hat{y}} \circ g : \begin{cases} E \to \Re \\ \epsilon \mapsto (F_{loss}^{\hat{y}} \circ g)(\epsilon) = F_{loss}^{\hat{y}}(\hat{y} + \epsilon) \end{cases}$$

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

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

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

Its Gradient matrix is

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

Proof: TO DO.

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

$$\frac{d(\xi^{\hat{y}} \circ S)}{d\,z}:\begin{cases} \Re^m \to \Re^m \\ z \mapsto S(z) - \hat{y} \end{cases}$$

Proof: TO DO.

4 – Dense layers

Definition – The application

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

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 $L_j:\Re^m \times \Re^m \times \Re \to \Re$ corresponding to the row j of the second matrix component of the dense layer L. Let $y \in \Re^m$ an input vector, $W \in M_{m',m}$ a

weight matrix and $b \in \Re^{m'}$ a bias matrix then

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

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

Let $y \in \Re^m$ an input vector and $b \in \Re^{m'}$ a bias matrix. Then $L_{j'} : \Re^m \times \Re^m \times \Re \to \Re$ is also derivable according the second variable for all $j \in [\![1, m']\!]$.

Its Gradient or Jacobian matrices are

$$\frac{\partial L}{\partial y} : \begin{cases} \Re^m \to M_{m',m} \\ y \mapsto W \end{cases}$$

$$\forall j \in [\![1, m']\!] \quad , \quad \frac{\partial L_j}{\partial w} : \begin{cases} \Re^m \to \Re^m \\ w \mapsto y \end{cases}$$

$$\frac{\partial L}{\partial b} : \begin{cases} \Re^{m'} \to M_{m',m'} \\ b \mapsto I_{m'} \end{cases}$$

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

Proof: TO DO.

Proposition – Let $L : \Re^m \times M_{m',m} \times \Re^{m'} \to \Re^{m'}$ and $ReLU : \Re^{m'} \to \Re^{m'}$ the dense layer and $ReLU$ activation functions. Let $F^{upstream} : \begin{cases} \Re^{m'} \to \Re \\ y' \mapsto F^{upstream}(y') \end{cases}$ such as $F^{upstream} \in D(\Re^{m'}, \Re)$.
Then $F_{upstream} \circ ReLU \circ L(\cdot, W, b) : \Re^m \times M_{m',m} \times \Re^{m'} \to \Re$ is derivable according the first and third variables on $\Re^m$ and $\Re^{m'}$ respectively.

The notation $F_{j'}^{upstream}$ means the application corresponding to the coefficient j' of $F^{upstream}$.
Let $y' \in \Re^{m'}$ then

$$\forall j' \in [\![1, m']\!] \quad , \quad F_{j'}^{upstream}(y'_j) = F^{upstream}(y')_{j'}$$

Let $y \in \Re^m$ an input vector and $b \in \Re^{m'}$ a bias matrix. Then
$F_{j'}^{upstream} \circ ReLU_{j'} \circ L_{j'}(\cdot, w, b_{j'}) : \Re \to \Re$ is also derivable for all $j' \in [\![1, m']\!]$.

Its Gradient matrices are

$$\frac{\partial(F^{upstream} \circ ReLU \circ L(\cdot, W, b))}{\partial y} : \begin{cases} \Re^m \to \Re^m \\ y \mapsto \frac{\partial(F^{upstream} \circ ReLU \circ L(\cdot, W, b))}{\partial y}(y) \end{cases}$$

where $\forall y \in \Re^m$, $\forall j \in [\![1, m]\!]$,

$$\frac{\partial(F^{upstream} \circ ReLU \circ L(\cdot, W, b))}{\partial y}(y)_j = \sum_{j'=1}^{m'} \frac{dF^{upstream}}{dy'}(ReLU(L(y, W, b)))_{j'} \times 1_{\Re_{\setminus\{0\}}^+}(L(y, W, b)_{j'}) \times W_{j',j}$$

with $W \in M_{m',m}$ a weight matrix, $b \in \Re^{m'}$ a bias matrix.

$$\forall j' \in [\![1, m']\!] ,$$
$$\frac{\partial(F_{j'}^{upstream} \circ ReLU_{j'} \circ L_{j'}(\cdot, w, b_{j'}))}{\partial w} : \begin{cases} \Re^m \to \Re^m \\ w \mapsto \frac{\partial(F_{j'}^{upstream} \circ ReLU_{j'} \circ L_{j'}(\cdot, w, b_{j'}))}{\partial w}(w) \end{cases}$$

where $\forall w \in \Re^m$, $\forall j \in [\![1, m]\!]$,

$$\frac{d(F_{j'}^{upstream} \circ ReLU_{j'} \circ L_{j'}(\cdot, w, b_{j'}))}{dw}(w)_j = F_{j'}^{upstream\,\prime}(ReLU_{j'}(L_{j'}(y, w, b_{j'}))) \times 1_{\Re_{\setminus\{0\}}^+}(L_{j'}(y, w, b_{j'})) \times y_j$$

with $y \in \Re^m$ an input matrix, $b \in \Re^{m'}$ a bias matrix.

$$\frac{\partial(F^{upstream} \circ ReLU \circ L(\cdot, W, b))}{\partial b} : \begin{cases} \Re^{m'} \to \Re^{m'} \\ b \mapsto \frac{\partial(F^{upstream} \circ ReLU \circ L(\cdot, W, b))}{\partial b}(b) \end{cases}$$

where $\forall b \in \Re^{m'}$, $\forall j' \in [\![1, m']\!]$,

$$\frac{\partial(F^{upstream} \circ ReLU \circ L(\cdot, W, b))}{\partial b}(b)_{j'} = \frac{dF^{upstream}}{dy'}(ReLU(L(y, W, b)))_{j'} \times 1_{\Re_{\setminus\{0\}}^+}(L(y, W, b)_{j'})$$

with $y \in \Re^m$ 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 $(X^i, \hat{Y}^i) \in \Re^m \times \Re^l$ for $i \in [\![1, n]\!]$. 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 $F_{act}^k \circ L^k(\cdot, W^k, b^k) : \Re^{m_k} \to \Re^{m_{k+1}}$ for $k \in [\![1, p]\!]$ with $W^k \in M_{m_{k+1}, m_k}$ and $b^k \in \Re^{m_{k+1}}$ the L^k weight and bias matrices. Let a training data set $(X^i, \hat{Y}^i) \in \Re^{m_1} \times \Re^{m_{p+1}}$ for $i \in [\![1, n]\!]$. Let $F_{loss}^{\hat{Y}^i} : \Re^{m_{p+1}} \to \Re$ loss functions for $i \in [\![1, n]\!]$ with $(\hat{Y}^i)_{i \in [\![1, n]\!]}$ as ground truth matrices respectively.

Then a neural network is defined as the application $N : \begin{cases} \Re^{m_1} \to \Re^{m_{p+1}} \\ y \mapsto \overset{n}{\underset{k=1}{\circ}} (F_{act}^k \circ L^k(\cdot, W^k, b^k))(y) \end{cases}$.

The optimization problem is $\underset{(W_{1,:}^k, \ldots, W_{m_{k+1},:}^k, b^k)_{k \in [\![1, p]\!]}}{min} \sum_{i=1}^{n} F_{loss}^{\hat{Y}^i}(N(X^i))$ and $F_{loss}^{global} : ((W_{1,:}^k, \ldots, W_{m_{k+1},:}^k, b^k)_{k \in [\![1, p]\!]}) \mapsto \sum_{i=1}^{n} F_{loss}^{\hat{Y}^i}(N(X^i))$ is named the objective function or global loss.

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

Then the following application $N : \Re^{m_1} \to \Re^{m_{p+1}}$ with

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

is a neural network and its objective function is

$$\xi_{loss}^{global}:\left(\left(W_{1,:}^{k},\ldots,W_{m_{k+1},:}^{k},b^{k}\right)_{k\in[\![1,p]\!]}\right)\mapsto\sum_{i=1}^{n}\xi_{loss}^{\hat{Y}^{i}}\left(N\left(X^{i}\right)\right)$$

Let $k\in[\![1,p]\!]$. For all $i\in[\![1,n]\!]$, let

$$y^{downstream(k),X^{i}}=\begin{cases}\underset{l=1}{\overset{k-1}{\circ}}\left(ReLU^{l}\circ L^{l}\left(\cdot,W^{l},b^{l}\right)\right)\left(X^{i}\right) & k\geq 2\\ X^{i} & k=1\end{cases}$$

$$F^{upstream(k),\hat{Y}^{i}}:\begin{cases}\mathfrak{R}^{m_{k+1}}\to\mathfrak{R}\\ y\mapsto\begin{cases}\xi^{\hat{Y}^{i}}\circ S\left(y\right) & k=p\\ \xi^{\hat{Y}^{i}}\circ S\circ L^{p}\left(\cdot,W^{p},b^{p}\right)\left(y\right) & k=p-1\\ \xi^{\hat{Y}^{i}}\circ S\circ L^{p}\left(\cdot,W^{p},b^{p}\right)\circ\underset{l=k+1}{\overset{p-1}{\circ}}\left(ReLU^{l}\circ L^{l}\left(\cdot,W^{l},b^{l}\right)\right)\left(y\right) & k\leq p-2\end{cases}\end{cases}$$

then $\dfrac{dF^{upstream(k),\hat{Y}^{i}}}{dy}:\begin{cases}\mathfrak{R}^{m_{k+1}}\to\mathfrak{R}\\ y\mapsto\begin{cases}S\left(y\right)-\hat{Y}^{i} & k=p\\ \left(S\left(y\right)-\hat{Y}^{i}\right)\times W^{p} & k=p-1\\ \dfrac{\partial\left(F^{upstream(k+1),\hat{Y}^{i}}\circ ReLU^{k+1}\circ L^{k+1}\left(\cdot,W,b\right)\right)}{\partial y}\left(y\right) & k\leq p-2\end{cases}\end{cases}$ where

$\forall k\in[\![1,p-2]\!]$, $\forall y\in\mathfrak{R}^{m_{k+1}}$, $\forall j\in[\![1,m]\!]$,

$$\frac{\partial\left(F^{upstream(k+1),\hat{Y}^{i}}\circ ReLU^{k+1}\circ L^{k+1}\left(\cdot,W^{k+1},b^{k+1}\right)\right)}{\partial y}\left(y\right)_{j}$$

$$=\sum_{j'=1}^{m'}\frac{dF^{upstream(k+1),\hat{Y}^{i}}}{dy'}\left(ReLU^{k+1}\left(L^{k+1}\left(y,W^{k+1},b^{k+1}\right)\right)\right)_{j'}\times 1_{\mathfrak{R}_{\setminus\{0\}}^{+}}\left(L^{k+1}\left(y,W^{k+1},b^{k+1}\right)_{j'}\right)\times W_{j',j}^{k+1}$$

with $W^{k+1}\in M_{m',m}$ a weight matrix, $b^{k+1}\in\mathfrak{R}^{m'}$ a bias matrix.

Let $k=p$. Then ξ_{loss}^{global} Gradient matrices are

$$\forall j'\in[\![1,m_{k+1}]\!]\ ,\ \frac{\partial\xi_{loss}^{global}}{\partial W_{j',:}^{k}}:\begin{cases}\mathfrak{R}^{m_{k}}\to\mathfrak{R}^{m_{k}}\\ w\mapsto\sum_{i=1}^{n}\dfrac{\partial\left(F_{j'}^{upstream(k),\hat{Y}^{i}}\circ L_{j'}^{k}\left(\cdot,w,b_{j'}^{k}\right)\right)}{\partial w}\left(w\right)\end{cases}$$

where $\forall i\in[\![1,n]\!]$, $\forall w\in\mathfrak{R}^{m_{k}}$, $\forall j\in[\![1,m_{k}]\!]$,

$$\frac{\partial\left(F_{j'}^{upstream(k),\hat{Y}^i}\circ L_{j'}^k\left(\cdot,w,b_{j'}^k\right)\right)}{\partial w}(w)_j$$

$=F_{j'}^{upstream(k),\hat{Y}^i\prime}\left(L_{j'}^k\left(y^{downstream(k),X^i},w,b_{j'}^k\right)\right)\times y_j^{downstream(k),X^i}$ with $b^k\in\Re^{m_{k+1}}$ a bias matrix.

$$\frac{\partial\xi_{loss}^{global}}{\partial b^k}:\begin{cases}\Re^{m_{k+1}}\to\Re^{m_{k+1}}\\ b^k\mapsto\sum_{i=1}^n\frac{\partial\left(F^{upstream(k),\hat{Y}^i}\circ L^k\left(\cdot,W^k,b^k\right)\right)}{\partial b^k}\left(b^k\right)\end{cases}$$

where $\forall i\in[\![1,n]\!]$, $\forall b^k\in\Re^{m_k}$, $\forall j'\in[\![1,m_{k+1}]\!]$,

$$\frac{\partial\left(F^{upstream(k),\hat{Y}^i}\circ L^k\left(\cdot,W^k,b^k\right)\right)}{\partial b^k}\left(b^k\right)_{j'}$$

$$=\frac{dF^{upstream(k),\hat{Y}^i}}{dy}\left(L^k\left(y^{downstream(k),X^i},W^k,b^k\right)\right)_{j'}\times 1_{\Re^+_{\setminus\{0\}}}\left(L^k\left(y^{downstream(k),X^i},W^k,b^k\right)_{j'}\right)$$

with $W^k\in M_{m_{k+1},m_k}$ a weight matrix.

Let $k\in[\![1,p-1]\!]$. Then ξ_{loss}^{global} Gradient matrices are

$$\forall j'\in[\![1,m_{k+1}]\!]\ ,\ \frac{\partial\xi_{loss}^{global}}{\partial W_{j',:}^k}:\begin{cases}\Re^{m_k}\to\Re^{m_k}\\ w\mapsto\sum_{i=1}^n\frac{\partial\left(F_{j'}^{upstream(k),\hat{Y}^i}\circ ReLU_{j'}^k\circ L_{j'}^k\left(\cdot,w,b_{j'}^k\right)\right)}{\partial w}(w)\end{cases}$$

where $\forall i\in[\![1,n]\!]$, $\forall w\in\Re^{m_k}$, $\forall j\in[\![1,m_k]\!]$,

$$\frac{\partial\left(F_{j'}^{upstream(k),\hat{Y}^i}\circ ReLU_{j'}^k\circ L_{j'}^k\left(\cdot,w,b_{j'}^k\right)\right)}{\partial w}(w)_j$$

$=F_{j'}^{upstream(k),\hat{Y}^i\prime}\left(ReLU_{j'}^k\left(L_{j'}^k\left(y^{downstream(k),X^i},w,b_{j'}^k\right)\right)\right)\times 1_{\Re^+_{\setminus\{0\}}}\left(L_{j'}^k\left(y^{downstream(k),X^i},w,b_{j'}^k\right)\right)\times y_j^{downstream(k),X^i}$

with $b^k\in\Re^{m_{k+1}}$ a bias matrix.

$$\frac{\partial\xi_{loss}^{global}}{\partial b^k}:\begin{cases}\Re^{m_{k+1}}\to\Re^{m_{k+1}}\\ b^k\mapsto\sum_{i=1}^n\frac{\partial\left(F^{upstream(k),\hat{Y}^i}\circ ReLU^k\circ L^k\left(\cdot,W^k,b^k\right)\right)}{\partial b^k}\left(b^k\right)\end{cases}$$

where $\forall i\in[\![1,n]\!]$, $\forall b^k\in\Re^{m_k}$, $\forall j'\in[\![1,m_{k+1}]\!]$,

$$\frac{\partial\left(F^{upstream(k),\hat{Y}^i}\circ ReLU^k\circ L^k\left(\cdot,W^k,b^k\right)\right)}{\partial b^k}\left(b^k\right)_{j'}$$

$$=\frac{d\,F^{upstream(k),\hat{Y}^i}}{d\,y}\left(ReLU^k\left(L^k\left(y^{downstream(k),X^i},W^k,b^k\right)\right)\right)_{j'}\times 1_{\Re^+_{\setminus\{0\}}}\left(L^k\left(y^{downstream(k),X^i},W^k,b^k\right)_{j'}\right)$$

with $W^k\in M_{m_{k+1},m_k}$ a weight matrix.

Proof: TO DO.

6 – Optimizations

Definition – Let $f\in D\left(\Omega,\Re\right)$ with $\Omega\in\Re^m$. <TODO>

7 – References