# A mathematical understanding of deep learning

Jiang J.
Data Engineer, Data Scientist
ENSEEIHT 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. These propositions will be mathematically proven. The subject tackled is a multi-class classification problem with – dense layers, *ReLU* and *SoftMax* activation layers, Categorical cross-entropy loss, Stochastic gradient descent optimizer. All the elements below are defined for classifications but can be re-used or easily re-defined to cover regressions.

## 2 – Notation and Nomenclature

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

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

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

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

**Definition** – Let $\Omega$ a non-empty open subset of $\Re^{m}$ , $_{m}\|.\|$ a norm on $\Re^{m}$ , $_{m'}\|.\|$ 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 $\Omega$ to $\Re^{m'}$ .

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

**Definition** – Let $(a_{i,j})_{i\in[\![1,n]\!],j\in[\![1,m]\!]}\in\Re^{n\times m}$ . 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 $a \in \Re^m$ and $b \in \Re^m$ . Let the element wise product noted as $(a|b)$ be

$$c = (a|b) = (b|a)$$

where $c$ is in $\Re^m$ with

$$\forall j \in [\![1,m]\!] \ , \ c_j = a_j \times b_j$$

# 3 – Activation functions

**Definition** – Let $z \in M_{n,m}$ , and $f \in \zeta(\Re^m) \cap D_{pw}(\Re^m)$ . Then the vector wise application

$$\forall i \in [\![1,n]\!] \ , \ f : \begin{cases} \Re^m \rightarrow \Re^m \\ z_{i,:} \mapsto f(z_{i,:}) \end{cases}$$

is an activation function.

**Proposition** – Let $z \in M_{n,m}$ . Then the following vector wise application is an activation function

$$\forall i \in [\![1,n]\!] \ , \ ReLU : \begin{cases} \Re^m \rightarrow \Re^m \\ z_{i,:} \mapsto max(0_{\Re^m}, z_{i,:}) \end{cases}$$

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

Its Jacobian matrix is

$$\forall i \in [\![1,n]\!] \quad , \quad J_{ReLU} : \begin{cases} \Re^m_{\setminus\{0\}} \to M_{m,m} \\ z_{i,:} \mapsto \begin{bmatrix} 1_{\Re^+_{\setminus\{0\}}}(z_{i,1}) & 0 & \cdots & 0 \\ 0 & 1_{\Re^+_{\setminus\{0\}}}(z_{i,2}) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1_{\Re^+_{\setminus\{0\}}}(z_{i,m}) \end{bmatrix} \end{cases}$$

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

Proof: TO DO.

**Proposition** – Let $z \in M_{n,m}$ . Then the following vector wise application is an activation function

$$\forall i \in [\![1,n]\!] \quad , \quad SoftMax : \begin{cases} \Re^m \to \Re^m \\ z_{i,:} \mapsto e^{z_{i,:}} \Big/ \sum_{j'=1}^{m} e^{z_{i,j'}} \end{cases}$$

with $e$ the element-wise exponential operation.

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

Its Jacobian matrix is

$$\forall i \in [\![1,n]\!] \quad , \quad J_S : \begin{cases} \Re^m \to M_{m,m} \\ z_{i,:} \mapsto J_S(z_{i,:}) \end{cases}$$

where $\forall (j, j') \in \{1,2,\dots,m\}^2$ , $J_S(z_{i,:})_{j,j'} = S(z_{i,:})_{i,j} \times (\delta_{j,j'} - S(z_{i,:})_{i,j'})$

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

Proof: TO DO.

# 3 – Categorical cross-entropy loss

**Definition** – Let $\hat{\Omega}$ a closed non empty subset. Let $\Omega$ an open non-empty $\Re$ subset with $\Omega = ]min_{\hat{\Omega}}, max_{\hat{\Omega}}[$ . Let $\hat{y} \in M_{n,m}(\hat{\Omega})$ and $f \in \zeta(\Omega^m, \Re) \cap D_{pw}(\Omega^m, \Re)$ . Let ${}_m\|.\|$ a norm on

$\Re^m$ . Then $f$ is a loss function is equivalent to the application

$$f\circ g:\begin{cases}]0,max_{\hat{\Omega}}-min_{\hat{\Omega}}[^m\rightarrow\Re\\ \epsilon\mapsto(f\circ g)(\epsilon)=f(\hat{y}+\epsilon)\end{cases}$$

is an increasing function according each component.

**Proposition** – Let $\hat{y}\in M_{n,m}(\{0,1\})$ , and $y\in M_{n,m}(]0,1[)$ . Then the application

$$\forall i\in[\![1,n]\!]\ ,\ \xi_{entropy}:\begin{cases}]0,1[^m\rightarrow\Re\\ y_{i,:}\mapsto-\sum_{j=1}^{m}\hat{y_{i,j}}\log(y_{i,j})\end{cases}$$

is a loss function.

Its Gradient matrix is

$$\forall i\in[\![1,n]\!]\ ,\ \nabla_{\xi_{entropy}}:\begin{cases}]0,1[^m\rightarrow\Re^m\\ y_{i,:}\mapsto\begin{bmatrix}\frac{\hat{y_{i,1}}}{y_{i,1}} & \cdots & \frac{\hat{y_{i,m}}}{y_{i,m}}\end{bmatrix}\end{cases}$$

<u>Proof:</u> TO DO.

# 4 – Dense layers

**Definition** – Let $y\in M_{n,m}$ , $W\in M_{m',m}$ and $b\in M_{1,m'}$ . Let $f$ an activation function. Then the application

$$\forall i\in[\![1,n]\!]\ ,\ L_{dense}:\begin{cases}M_{1,m}\times M_{m',m}\times M_{1,m'}\rightarrow M_{1,m'}\\ (y_{i,:},W,b)\mapsto f(y_{i,:}\times W^T+b)\end{cases}$$

defines a dense layer with $y\in M_{n,m}$ named the input vector, $W\in M_{m',m}$ named the weight matrix and $b\in M_{1,m'}$ named the bias matrix.

**Proposition** – Let $y\in M_{n,m}$ the input vector, $W\in M_{m',m}$ the weight matrix and $b\in M_{1,m'}$ the bias matrix. Then the application

$$\forall i\in[\![1,n]\!]\ ,\ L_{dense}:\begin{cases}M_{1,m}\times M_{m',m}\times M_{1,m'}\rightarrow M_{1,m'}\\ (y_{i,:},W,b)\mapsto f(y_{i,:}\times W^T+b)\end{cases}$$

defines a dense layer.

Its Jacobian matrices are

$$\forall i \in [\![1,n]\!] \quad , \quad J_{L_{dense}} : \begin{cases} M_{1,m} \rightarrow M_{m',m} \\ y_{i,:} \mapsto W \end{cases}$$

$$\forall i \in [\![1,n]\!] \quad , \quad J_{L_{dense}} : \begin{cases} M_{1,m'} \rightarrow M_{m',m} \\ b \mapsto \begin{bmatrix} \frac{\hat{y}_{i,1}}{y_{i,1}} & \cdots & \frac{\hat{y}_{i,m}}{y_{i,m}} \end{bmatrix} \end{cases}$$

<u>Proof:</u> TO DO.

**Proposition** – Let $y \in M_{n,m}$ the input vector, $W \in M_{m,m'}$ the weight matrix and $b \in M_{1,m'}$ the bias matrix. Then the sequential operations

$$\forall i \in [\![1,n]\!] \quad , \quad \forall j' \in [\![1,m']\!] \quad , \quad z'_{i,j'} = y_{i,:} \times (W^T)_{:,j'} + b$$

$$\forall i \in [\![1,n]\!] \quad , \quad y'_{i,} = S(z'_{i,:})$$

defines a dense layer.

The gradient matrices are the following

$\forall i \in [\![1,n]\!]$ ,

# 5 – Neural Network

**Definition** – Suppose a data set with $n$ samples. Each sample have $m$ features and a corresponding one-hot encoded label among $l$ possible labels.

Let $X \in M_{n,m}$ , and $Y \in M_{n,l}$ the matrices defining the features and the labels respectively for each sample.

Suppose a neural network with $k$ layers.

# 6 – References