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 Ω 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 continuous function is equivalent to

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

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

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 $\Re^m$, ${}_m\|.\|$ a norm on $\Re^m$, ${}_{m'}\|.\|$ a norm on $\Re^{m'}$, and $f:\Omega\to\Re^{m'}$. Then f derivable is equivalent to

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

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

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 $\Re^m$, ${}_m\|.\|$ 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 Ω 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 $a \in \Re$ and $B \in M_{n,m}$. Let the element wise product noted as $a * B$ be

$$C = a * B = B * a$$

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

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

#TODO – Define chain rules

Theorem – Let

3 – Activation functions

Definition – Let $f \in \zeta(\Re^m) \cap D(\Re^m)$. Then the vector wise application

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

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 $z \in \mathfrak{R}^m$ then

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

$ReLU$ is supposed derivable on every coefficients at 0

$$\forall j \in [\![1,m]\!] \quad , \quad \frac{d\,ReLU_j}{d\,z_j}(0) = 0$$

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

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

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

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

with e the element-wise exponential operation.

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

Its Jacobian matrix is

$$\frac{d\,S}{d\,z} : \begin{cases} \mathfrak{R}^m \to M_{m,m} \\ z \mapsto \frac{d\,S}{d\,z}(z) \end{cases}$$

where $\forall z \in \Re^m$, $\forall (j,j') \in \{1,2,...,m\}^2$, $\frac{dS}{dz}(z)_{j,j'} = S(z)_j \times (\delta_{j,j'} - S(z)_{j'})$

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 $f \in \xi(\Omega, \Re) \cap D(\Omega, \Re)$. Then f is a loss function is equivalent to the application

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

is an increasing function according each coefficient with $E \subseteq M_{n,m}$ such as $f \circ g$ 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

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

is a loss function.

The application is commonly named as Categorical cross-entropy loss.

Its Gradient matrix is

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

Proof: TO DO.

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

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

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 corresponding to the coefficient j 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

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

Proposition – Let $L : \Re^m \times M_{m',m} \times \Re^{m'} \to \Re^{m'}$ the 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 $L_j : \Re^m \to \Re$ is also derivable for all coefficient $j \in [\![1, m']\!]$.

Its Jacobian matrices are

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

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

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

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 $F^{upstream}:\Re^{m'}\to\Re$ such as $F^{upstream}\in\xi(\Re^{m'},\Re)\cap D(\Re^{m'},\Re)$. Then $F_{upstream}\circ ReLU\circ L:\Re^m\to\Re^m\times M_{m',m}\times\Re^{m'}\to\Re^{m'}$ is derivable according to 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

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

Let $y\in\Re^m$ an input vector and $b\in\Re^{m'}$ a bias matrix. Then $F_j^{upstream}\circ ReLU_j\circ L_j:\Re^m\to\Re$ is also derivable for all coefficient $j\in[\![1,m']\!]$.

Its Jacobian matrices are

$$\frac{\partial(F^{upstream}\circ ReLU\circ L)}{\partial y}:\begin{cases}\Re^m\to M_{m',m}\\ y\mapsto\begin{bmatrix}\frac{dF^{upstream}}{dy'}(ReLU(L(y,W,b)))_1\times 1_{\Re^+_{\setminus\{0\}}}(L(y,W,b)_1)*W_{1,:}\\ \vdots\\ \frac{dF^{upstream}}{dy'}(ReLU(L(y,W,b)))_{m'}\times 1_{\Re^+_{\setminus\{0\}}}(L(y,W,b)_{m'})*W_{m',:}\end{bmatrix}\end{cases}$$

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

TODO upstream

$$\forall\, j\in[\![1,m']\!]\ ,\ \ \frac{d\left(ReLU_j\circ L_j\right)}{d\,w}:\begin{cases}\Re^m\to\Re^m\\ w\mapsto\left[1_{\Re^+_{\setminus\{0\}}}(L_1(w))\times y_1\ \ \cdots\ \ 1_{\Re^+_{\setminus\{0\}}}(L_m(w))\times y_m\right]\end{cases}$$

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

TODO upstream

$$\frac{\partial\left(ReLU\circ L\right)}{\partial\, b}:\begin{cases}\Re^{m'}\to M_{m',m'}\\ b\mapsto\dfrac{d\,ReLU}{d\,b}\left(L\left(y,W,b\right)\right)\end{cases}$$

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

Proof: TO DO.

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