A tentative introduction to Mathematical Aspects of Deep Learning

Jérémie Bigot

Institut de Mathématiques de Bordeaux
Université de Bordeaux

Theory of Machine Learning reading group, LaBRI
May 2018

Main objectives of this talk

Two levels :

- **Global one and very ambitious -** to present some of the current research works towards the understanding of the mathematical aspects of deep learning
- **Local one and (less ?) ambitious -** to gather researchers within the University of Bordeaux (IMB, LaBRI, IMS) with research interests in the understanding of deep learning (machine learning, artificial intelligence,...)

This talk is based on two previous lectures given at IMB in last April :

www.math.u-bordeaux.fr/~jbigot/Site/Enseignement_files/pres_DeepLearning.pdf

www.math.u-bordeaux.fr/~jbigot/Site/Enseignement_files/pres_DeepLearning_article.pdf

Some useful references

Previous talks based on the online (and free) book by Michael Nielsen

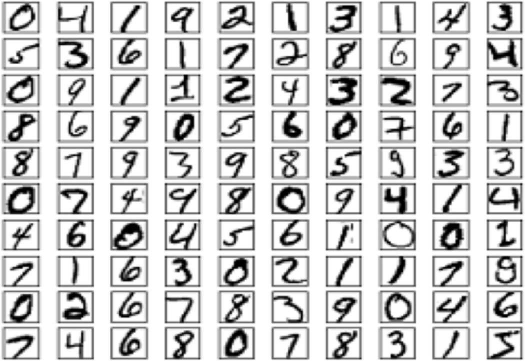
http://neuralnetworksanddeeplearning.com/index.html

with codes in Python :

github.com/mnielsen/neural-networks-and-deep-learning.gi

Some useful references

Illustrative example : MNIST database [1]

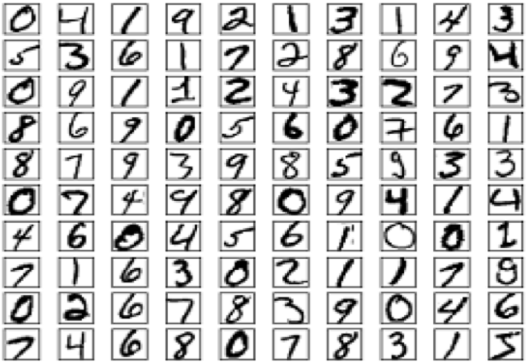


Images of size $d = 28 \times 28 = 784$ pixels, $K = 10$ classes

1. http://neuralnetworksanddeeplearning.com/index.html

Some useful references

Supervised classification by convolutional neural netwroks

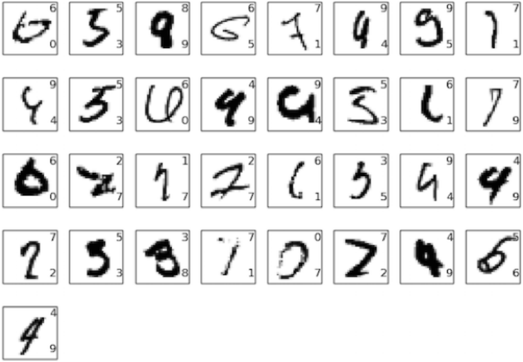


Training set of 50000 images
Correct classification rate $> 99\%$

Test set of 10000 images : Correct classification rate $> 99\%$

Some useful references

33 mis-classified images on the test set [1]



True class : upper-right corner

Predicted class : lower-right corner

1. http://neuralnetworksanddeeplearning.com/index.html

Some useful references

Tutorials and talks online :

- 2017 : “Theories of Deep Learning”, University of Stanford (introduction by David Donoho)

  https://stats385.github.io/

  - Talks on recent theoretical works
  - Links towards various papers (useful for a reading group)

- 2018 : “The Mathematical Theory of Deep Neural Networks”, Institute for Advanced Study - Princeton University

  https://sites.google.com/site/princetondeepmath/

Statistical learning

Generic problem : given a **training set** $(X_i, Y_i)_{1 \leq i \leq n}$ where

- $X_i \in \mathbb{R}^d$
- $Y_i \in \mathbb{R}$ (**regression**) or $Y_i \in \{1; 2; \ldots; K\}$ (**supervised classification**)

One would like to :

- determine a model which links the entry X_i to the output Y_i for all $1 \leq i \leq n$
- for $i_0 \notin \{1, \ldots, n\}$ we want to predict $\hat{Y}_{i_0}$ as an estimation of Y_{i_0} (**not observed**) given the knowledge of X_{i_0} only

The pair(X_{i_0}, Y_{i_0}) is an element of the **test set**.

Remark : in classification, one may also consider that

$$Y_i \in \Sigma_K = \left\{ (p_1, \ldots, p_K) : p_k \geq 0 \text{ et } \sum_{k=1}^{K} p_k = 1 \right\}$$

Choice of a class of models

Definition : a class of model is a set of functions $f_\theta : \mathbb{R}^d \to \mathbb{R}$ (**regression**) or $f_\theta : \mathbb{R}^d \to \Sigma_K$ (**classification**) indexed by a parameter

$$\theta \in \Theta \subset \mathbb{R}^p$$

Learning step : minimization of the empiricial risk

$$\hat{\theta} \in \arg\min M_n(\theta) \quad \text{with} \quad M_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} L(Y_i, f_\theta(X_i))$$

where L is a loss function e.g.

$$L(y,z) = \|y-z\|^2$$

or the cross-entropy in classification i.e.

$$L(y,z) = -\sum_{k=1}^{K} y_k \log(z_k)$$

Prediction : $\hat{Y}_{i_0} = f_{\hat{\theta}}(X_{i_0})$

Some (mathematical) questions [1]

- **Theory of approximation** : what types of functional classes can we approach with parametric functions f_θ for $\theta \in \Theta \subset \mathbb{R}^p$ (here neural networks) given a desired level of accuracy ?

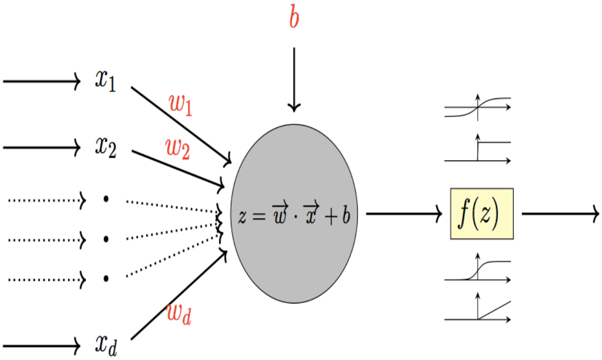
- **Optimization** : how quantifying the performances of stochastic gradient methods (used in the learning step for neural networks) when n and p are very large ?

- **Generalization capacity** of neural networks ? Good performances on the test set and no over-fitting despite a large number of parameters p... Is there some regularization effect ?

1. cf. travaux récents de Tomaso Poggio

Construction of a neural network

Basic neuron : the **Perceptron** model (Rosenblatt, 1957)



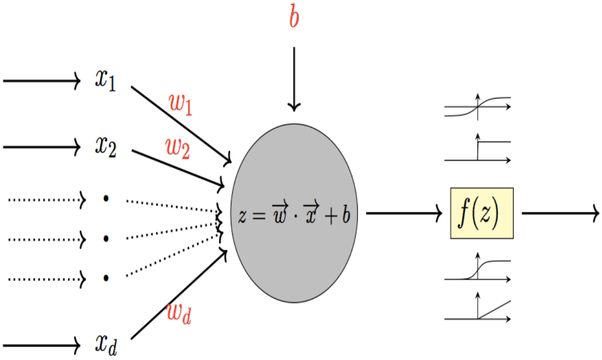
Source : `https://stats385.github.io/`

Linear combination of $x \in \mathbb{R}^d$ with weights $\omega_1, \dots, \omega_d$ and a bias b

Non-linear activation function $f(z) = \sigma(z) = \mathbb{1}_{\{z \geq 0\}}$

Construction of a neural network

Basic neuron : the **Perceptron** model (Rosenblatt, 1957)



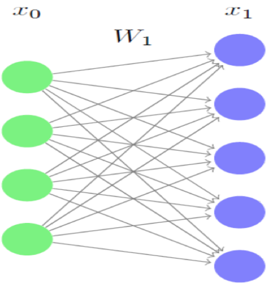
Source : `https://stats385.github.io/`

Linear combination of $x \in \mathbb{R}^d$ with weights $\omega_1, \ldots, \omega_d$ and a bias b

Other choices $\sigma(z) = \frac{1}{1+\exp(-z)}$ (sigmoid) ou $\sigma(z) = \max(0, z)$ (ReLU)

Construction of a neural network

Single layer Perceptron



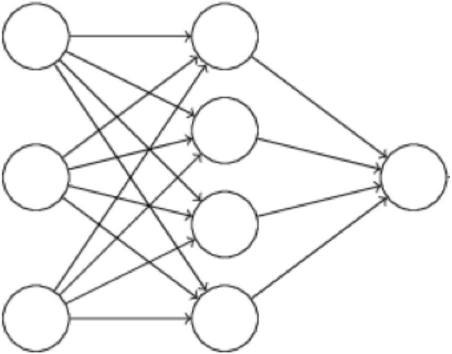
Source : `https://stats385.github.io/`

Simpler formulation : $f_\theta(\mathbf{x}_0) = \sigma_1\left(W_1\mathbf{x}_0 + b_1\right)$, where

- $\sigma_1 : \mathbb{R}^{d_1} \to \mathbb{R}^{d_1}$ is a **non-linear and entry-wise** function
- $W_1 \in \mathbb{R}^{d \times d_1}$ (weights) $b_1 \in \mathbb{R}^{d_1}$ (bias)
- $\theta = (W_1, b_1)$: parameters of the network

Construction of a neural network

Single layer Perceptron - Regression



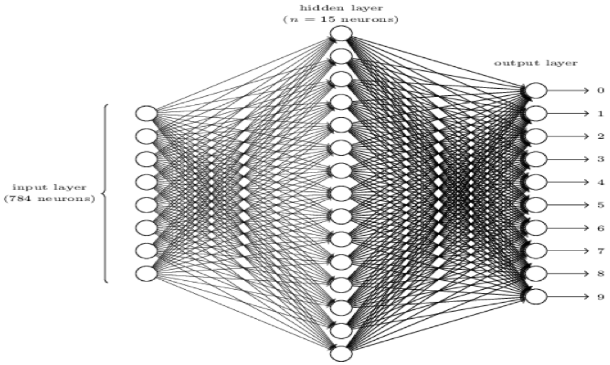
Source : `http://neuralnetworksanddeeplearning.com/index.html`

Simpler formulation : $f_\theta(\mathbf{x}_0) = W_2\sigma_1\left(W_1\mathbf{x}_0 + b_1\right) + b_2,$

with $W_1 \in \mathbb{R}^{d\times d_1}$, $b_1 \in \mathbb{R}^{d_1}$, $W_2 \in \mathbb{R}^{d_1\times 1}$, $b_2 \in \mathbb{R}$ and $\theta = (W_1, b_1, W_2, b_2)$

Construction of a neural network

Single layer Perceptron - Classification



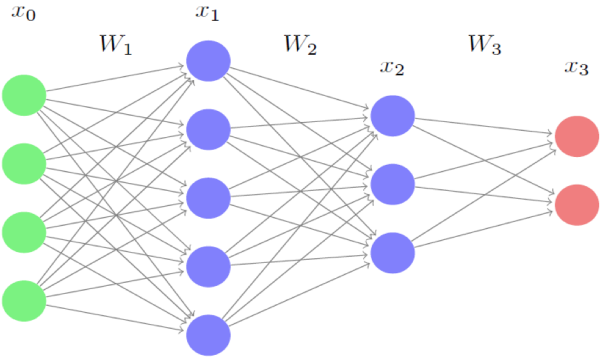
Source : `http://neuralnetworksanddeeplearning.com/index.html`

Simpler formulation : $f_\theta(\mathbf{x}_0) = \sigma_{\text{softmax}}\left(W_2\sigma_1\left(W_1\mathbf{x}_0 + b_1\right) + b_2\right)$,

with $W_1 \in \mathbb{R}^{d\times d_1}$, $b_1 \in \mathbb{R}^{d_1}$, $W_2 \in \mathbb{R}^{d_1\times K}$, $b_2 \in \mathbb{R}^K$, $\theta = (W_1, b_1, W_2, b_2)$

Construction of a neural network

Multi-layer Perceptron



Source : `https://stats385.github.io/`

Simpler formulation : entry $\mathbf{x}_0 \in \mathbb{R}^d$, output $\mathbf{x}_L$, then, for $\ell = 1, \ldots, L$, do $\mathbf{x}_\ell = \sigma_\ell \left(W_\ell \mathbf{x}_{\ell-1} + b_\ell \right)$ with $\sigma_L = Id$ or $\sigma_L = \sigma_{\text{softmax}}$

A theoretical point of view of the generalization capacity of deep neural networks

Overview of the paper :

“Fisher-Rao Metric, Geometry, and Complexity of Neural Networks”, by

Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, James Stokes

Arxiv Pre-print (2018) : arXiv :1711.01530

Neural network with or without biais ?

Mode considered in Liang et al. (2018)

Neural network : $f_\theta : \mathbb{R}^d \to \mathbb{R}^K$ defined by

$$f_\theta(x) = \sigma_{L+1}(\sigma_L(\dots\sigma_2(\sigma_1(x^T W^0)W^1)W^2)\dots)W^L)$$

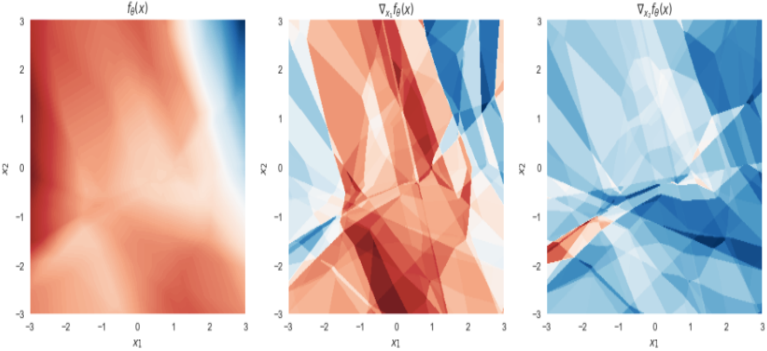
and indexed by the parameter

$$\theta \in \Theta_L \subset \mathbb{R}^p$$

with $\Theta_L = (W^\ell)_{0\leq\ell\leq L}$ and W^ℓ rectangular matrices (**no bias**), and $\sigma_1, \dots, \sigma_{L+1}$ **a priori** non-linear activation functions for each layer.

Neural network with or without biais ?

With bias : dimension of the entry $d = 2$, $L = 3$ hidden layers with 15 RELU neurons in each layer (+ random choice of the weights) - univariate output $(K = 1)$

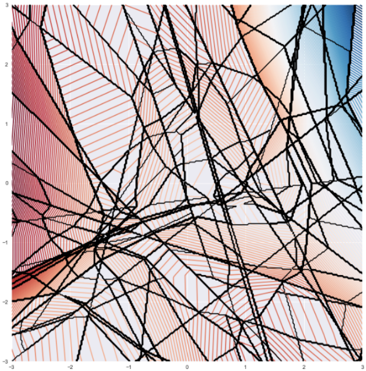


Source :

`http://www.inference.vc/generalization-and-the-fisher-rao-norm-2/`

Neural network with or without biais ?

With bias : dimension of the entry $d = 2$, $L = 3$ hidden layers with 15 RELU neurons in each layer (+ random choice of the weights) - univariate output ($K = 1$) = piece-wise affine function

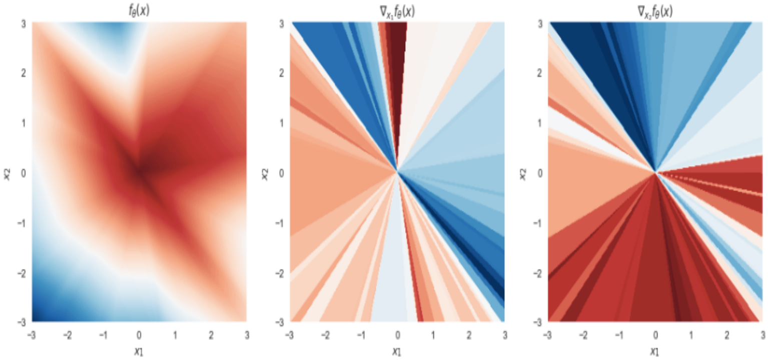


Source :

http://www.inference.vc/generalization-and-the-fisher-rao-norm-2/

Neural network with or without biais ?

With bias : dimension of the entry $d = 2$, $L = 3$ hidden layers with 15 RELU neurons in each layer (+ random choice of the weights) - univariate output ($K = 1$)

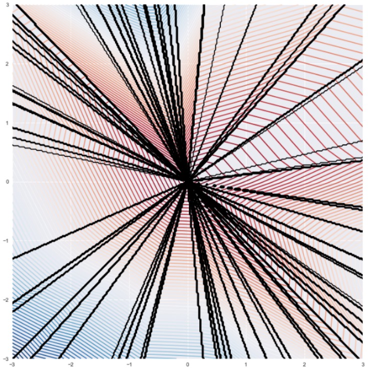


Source :

http://www.inference.vc/generalization-and-the-fisher-rao-norm-2/

Neural network with or without biais ?

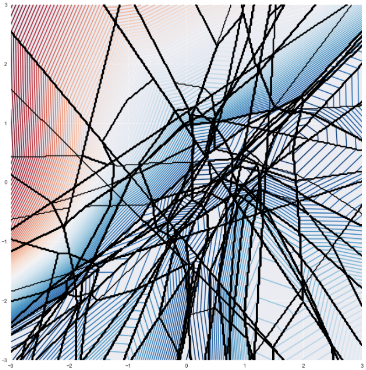
Without bias : dimension of the entry $d = 2$, $L = 3$ hidden layers with 15 RELU neurons in each layer (+ random choice of the weights) - univariate output ($K = 1$) = piece-wise affine function



Source :
`http://www.inference.vc/generalization-and-the-fisher-rao-norm-2/`

Neural network with or without biais ?

Without biais : dimension of the entry $d = 3$, $L = 3$ hidden layers with 15 RELU neurons in each layer (+ random choice of the weights) - univariate output ($K = 1$) - Visualization of $(x_1, x_2) \mapsto f_\theta(x_1, x_2, 1)$



Source :

http://www.inference.vc/generalization-and-the-fisher-rao-norm-2/

Generalization capacity in classification

Learning set : $\mathcal{A}_n = (X_i, Y_i)_{1 \leq i \leq n}$ an iid sequence with

- $X_i \in \mathbb{R}^d$
- $Y_i \in \{-1; +1\}$ (**binary classification**)

Neural network : $f_\theta : \mathbb{R}^d \to \mathbb{R}$ defined by

$$f_\theta(x) = \sigma_{L+1}(\sigma_L(\ldots \sigma_2(\sigma_1(x^T W^0) W^1) W^2) \ldots) W^L)$$

and indexed by the parameter

$$\theta \in \Theta_L \subset \mathbb{R}^p$$

with $\Theta_L = (W^\ell)_{0 \leq \ell \leq L}$ and W^ℓ rectangular matrices (no biais).

Learning step : minimization of the empirical risk

$$\hat{\theta} \in \arg\min_{\theta \in \Theta_L} \frac{1}{n} \sum_{i=1}^{n} L(Y_i, f_\theta(X_i)),$$

Example of a loss function (hinge loss) : $L(y, z) = \max(0, 1 - yz)$

Generalization capacity in classification

Classification rule : define the function

$$\hat{g}(x) = \begin{cases} -1 & \text{if} \quad f_{\hat{\theta}}(x) < 0 \\ +1 & \text{if} \quad f_{\hat{\theta}}(x) \geq 0 \end{cases}$$

Test set $(X, Y) \in \mathbb{R}^d \times \{-1; +1\}$ a random pair with the same distribution than data in the learning set.

“Generalization capacity” : evaluate (by upper bounding) the quantity

$$\mathbb{P}\left(\hat{g}(X) \neq Y | \mathcal{A}_n\right) = \mathbb{E}\left(\mathbb{1}_{\{f_{\hat{\theta}}(X)Y<0\}} | \mathcal{A}_n\right)$$

which is the rate of miss-classification on the test set.

Generalization capacity in classification

To control the generalization capacity of a classification rule f_θ one needs to :

- equip the set of parameters Θ_L with an appropriate norm $\|\cdot\|$
- estime the rate of miss-classification under the constraint that $\|\theta\| \leq \gamma$ with $\gamma > 0$ a given constant which allows to "control the complexity" of the function f_θ.

Main contribution in Liang et al. (2018) : to use the notion of Fisher-Rao norm on Θ_L to evaluate the generalization capacity of neural networks.

Basic principles in statistical learning [1]

Learning set : $\mathcal{A}_n = (X_i, Y_i)_{1 \leq i \leq n}$ iid sequence with

- $X_i \in \mathbb{R}^d$
- $Y_i \in \{-1; +1\}$ (**binary classification**)

Test set : $(X, Y) \in \mathbb{R}^d \times \{-1; +1\}$ a random pair with the same distribution than data in the learning set.

Classification rule : a set of functions $g : \mathbb{R}^d \to \{-1; +1\}$ belonging to some given functional class $\mathcal{C}$

1. Presentation based on the review paper by Boucheron, Bousquet, Lugosi (2005), ESAIM P&S.

Basic principles in statistical learning [1]

Probability of error : $L(g) = \mathbb{P}\left(g(X) \neq Y\right)$

Empirical risk : $L_n(g) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{g(X_i) \neq Y_i\}}$

Classifier : $\hat{g}_n \in \arg\min_{g \in \mathcal{C}} L_n(g)$

Generalization capacity :

$$L(\hat{g}_n) \leq L_n(\hat{g}_n) + \sup_{g \in \mathcal{C}} |L_n(g) - L(g)|,$$

with

$$L(\hat{g}_n) = \mathbb{P}\left(\hat{g}_n(X) \neq Y | \mathcal{A}_n\right)$$

1. Presentation based on the review paper by Boucheron, Bousquet, Lugosi (2005), ESAIM P&S.

Basic principles in statistical learning [1]

Question : how to control the quantity $\sup_{g \in \mathcal{C}} |L_n(g) - L(g)|$?

Using a concentration inequality (bounded difference inequality) leads to

$$\sup_{g \in \mathcal{C}} |L_n(g) - L(g)| \leq \mathbb{E} \sup_{g \in \mathcal{C}} |L_n(g) - L(g)| + \sqrt{\frac{2 \log(1/\delta)}{n}},$$

with probability $1 - \delta$.

Question : how to control the quantity $\mathbb{E}\left(\sup_{g \in \mathcal{C}} |L_n(g) - L(g)|\right)$?

1. Presentation based on the review paper by Boucheron, Bousquet, Lugosi (2005), ESAIM P&S.

Basic principles in statistical learning [1]

Let $\epsilon_1, \ldots, \epsilon_n$ be idd Rademacher random variables i.e. such that

$$\mathbb{P}(\epsilon_i = +1) = \mathbb{P}(\epsilon_i = -1) = 1/2.$$

Then, one has that

$$\mathbb{E} \sup_{g \in \mathcal{C}} |L_n(g) - L(g)| \leq \mathbb{E} R_n(\mathcal{C}),$$

where $R_n(\mathcal{C})$ is the so-called Rademacher complexity of the class $\mathcal{C}$ defined by

$$R_n(\mathcal{C}) = \mathbb{E} \left(\sup_{g \in \mathcal{C}} \frac{1}{n} \left| \sum_{i=1}^{n} \epsilon_i g(X_i) \right| |\mathcal{A}_n \right)$$

1. Presentation based on the review paper by Boucheron, Bousquet, Lugosi (2005), ESAIM P&S.

Basic principles in statistical learning [1]

For a classification rule of the form

$$g(x) = \begin{cases} -1 & \text{if} \quad f_\theta(x) < 0 \\ +1 & \text{if} \quad f_\theta(x) \geq 0 \end{cases}$$

one has to control the Rademacher complexity

$$R_n(B_{\|\cdot\|}(\gamma)) = \mathbb{E}\left(\sup_{\theta \in B_{\|\cdot\|}(\gamma)} \frac{1}{n} \left| \sum_{i=1}^{n} \epsilon_i f_\theta(X_i) \right| |\mathcal{A}_n \right),$$

associated to the set of parameters

$$B_{\|\cdot\|}(\gamma) = \{\theta \in \Theta : \|\theta\| \leq \gamma\}$$

1. Presentation based on the review paper by Boucheron, Bousquet, Lugosi (2005), ESAIM P&S.

Which norm on the set of parameters ?

Neural network : $f_\theta : \mathbb{R}^d \to \mathbb{R}$ defined by

$$f_\theta(x) = \sigma_{L+1}(\sigma_L(\ldots\sigma_2(\sigma_1(x^T W^0)W^1)W^2)\ldots)W^L)$$

and indexed by the parameter

$$\theta \in \Theta_L \subset \mathbb{R}^p$$

with $\Theta_L = (W^\ell)_{0\leq\ell\leq L}$ and W^ℓ rectangular matrices (no biais).

Hypothesis on the activation function :

$$\sigma_\ell(z) = \sigma'_\ell(z)z \quad \text{for all} \quad 1 \leq \ell \leq L$$

Example : identity $\sigma_\ell(z) = z$, or "leaky" RELU $\sigma_\ell(z) = \max\{\alpha z, z\}$ with $0 \leq \alpha < 1$ (no so clear for me...)

Which norm on the set of parameters ?

Main ideas :

- the parameters in a neural network are not identifiable (one may have $f_{\theta_1} = f_{\theta_2}$ with $\theta_1 \neq \theta_2$)
- to construct a norm $\|\cdot\|$ on Θ which satisfies

$$\text{if} \quad f_{\theta_1} = f_{\theta_2} \quad \text{then} \quad \|\theta_1\| = \|\theta_2\|$$

The Fisher-Rao metric

Let $Z \in \mathcal{Z}$ be a random vector with parametric density $p_\theta(z)$ where

$$\theta \in \Theta \text{ open set of } \mathbb{R}^p$$

Hypothesis : regular parametric density model

Fisher information matrix : for any $\theta \in \Theta$

$$I(\theta) = \mathbb{E}\left(\nabla_\theta \log p_\theta(Z) \otimes \nabla_\theta \log p_\theta(Z)\right)$$

i.e. such that

$$\begin{aligned} I_{jk}(\theta) &= \mathbb{E}\left(\frac{\partial \log p_\theta(Z)}{\partial \theta_j}\frac{\partial \log p_\theta(Z)}{\partial \theta_k}\right) \\ &= \int_{\mathcal{Z}} \frac{\partial \log p_\theta(z)}{\partial \theta_j}\frac{\partial \log p_\theta(z)}{\partial \theta_k} p_\theta(z) dz. \end{aligned}$$

The Fisher-Rao metric

Let $Z \in \mathcal{Z}$ be a random vector with parametric density $p_\theta(z)$ where

$$\theta \in \Theta \text{ open set of } \mathbb{R}^p$$

The Fisher information matrix allows to equip the set of parameters Θ with the structure of a Riemannian manifold :

- inner product on the tangent space : $\langle u, v\rangle_\theta = u^T I(\theta) v$ for all $u, v \in T_\theta\Theta$ et $\|u\|_\theta^2 = u^T I(\theta) u$
- induced metric d_{fr} (Fisher-Rao) on Θ is invariant by re-parametrization i.e.

$$d_{\text{fr}}(\theta_1, \theta_2) = d_{\text{fr}}(\phi(\theta_1), \phi(\theta_2))$$

for any diffeomorphism $\phi : \Theta \mapsto \mathbb{R}^p$ (change of parametrization of Θ).

Definition of the Fisher-Rao norm [Liang et al. (2018)]

Definition : the Fisher-Rao pseudo-norm is defined by

$$\|\theta\|_{\mathrm{fr}}^2 := \langle \theta, I(\theta)\theta \rangle$$

where $I(\theta)$ is the matrix (of the type "Fisher information")

$$I(\theta) = \mathbb{E}\left(\nabla_\theta L(Y, f_\theta(X)) \otimes \nabla_\theta L(Y, f_\theta(X))\right)$$

Proposition (Liang et al. (2018))

If $z \mapsto L(y,z)$ is differentiable then (binary classification)

$$\|\theta\|_{\mathrm{fr}}^2 = (L+1)^2 \mathbb{E}\left[\left(\partial_2 L(Y, f_\theta(X))\right)^2 f_\theta(X)^2\right]$$

This implies that if $f_{\theta_1} = f_{\theta_2}$ then $\|\theta_1\|_{\mathrm{fr}} = \|\theta_2\|_{\mathrm{fr}}$.

Definition of the Fisher-Rao norm [Liang et al. (2018)]

Matrix of the type “Fisher information”

$$I(\theta) = \mathbb{E}\left(\nabla_\theta L(Y, f_\theta(X)) \otimes \nabla_\theta L(Y, f_\theta(X))\right)$$

Proposition (Liang et al. (2018))

If $z \mapsto L(y, z)$ is differentiable then (binary classification)

$$\|\theta\|_{\mathrm{fr}}^2 = (L+1)^2 \mathbb{E}\left[\left(\partial_2 L(Y, f_\theta(X))\right)^2 f_\theta(X)^2\right]$$

Key equation for the proof : under the hypothesis that $\sigma_\ell(z) = \sigma_\ell'(z) z$ for all $1 \leq \ell \leq L$, one has that

$$\nabla_\theta f_\theta(X)^T \theta = (L+1) f_\theta(X)$$

Properties of the Fisher-Rao norm

- comparison with other norms :

Proposition (Liang et al. (2018))

For 4 other norms on matrices $\|\cdot\|$ on Θ_L (e.g. the spectral norm), one has that

$$\frac{1}{L+1}\|\theta\|_{\text{fr}} \leq \|\theta\|$$

for any $\theta \in \Theta_L = (W^\ell)_{0\leq\ell\leq L}$

- properties of invariance of f_θ uner re-parametrization of θ...

Control of generalization capacity

Consider the Rademacher complexity

$$R_n(\Theta) = \mathbb{E}_\epsilon \left(\sup_{\theta \in \Theta} \frac{1}{n} \left| \sum_{i=1}^{n} \epsilon_i f_\theta(X_i) \right| \right), \text{ for } \Theta \subset \Theta_L.$$

Assumptions :

- binary classification
- **linear** activation functions $\sigma_\ell(z) = z$ for all $1 \leq \ell \leq L+1$
- the matrix $\mathbb{E}\left[XX^T\right] \in \mathbb{R}^{d \times d}$ is of full rank

Proposition (Liang et al. (2018))

Under these assumptions, one has that

$$\mathbb{E} R_n(B_{\mathrm{fr}}(\gamma)) \leq \gamma \sqrt{\frac{d}{N}}$$

where $B_{\mathrm{fr}}(\gamma) = \left\{ \theta \in \Theta_L : \frac{1}{L+1} \|\theta\|_{\mathrm{fr}} \leq \gamma \right\}$

Control of generalization capacity

Proposition (Liang et al. (2018))

Under these assumptions, one has that

$$\mathbb{E}\mathbb{1}_{\{f_\theta(X)Y<0\}} \leq \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\{f_\theta(X_i)Y_i\leq\alpha\}} + \frac{C}{\alpha}R_n(B_{\mathrm{fr}}(\gamma)) + C\sqrt{\frac{\log(1/\delta)}{n}}$$

for any $\theta \in B_{\mathrm{fr}}(\gamma)$ (with probability $1-\delta$), where $\alpha > 0$ is any margin parameter, and $C > 0$ a universal constant.

Strategy of optimisation

Compute $\hat{\theta}$ by **gradient descent** : (e.g. package `nnet` of R based on BFGS)

$$\hat{\theta}_{j+1} = \hat{\theta}_j - \gamma_j \nabla M_n(\hat{\theta}_j) \quad \text{and} \quad \hat{\theta} = \hat{\theta}_J,$$

for J sufficiently large.

Compute $\hat{\theta}$ by **stochastic gradient descent** :

At each iteration j, random choice of a subset of data $X_{i_1}, \ldots X_{i_q}$ (**batch**) of size $q \ll n$

$$\hat{\theta}_{j+1} = \hat{\theta}_j - \gamma_j \nabla m_q(\hat{\theta}_j) \quad \text{where} \quad m_q(\theta) = \frac{1}{q} \sum_{\ell=1}^{q} L(Y_{i_\ell}, f_\theta(X_{i_\ell}))$$

Gradient descent with respect to a metric

Interpretation of gradient descent : $\hat{\theta}_{j+1} = \hat{\theta}_j - \gamma_j \nabla M_n(\hat{\theta}_j)$ as

$$\hat{\theta}_{j+1} = \arg\min_{\theta \in \mathbb{R}^p} \left\{ M_n(\theta) + \frac{1}{2\gamma_j} \|\theta - \hat{\theta}_j\|^2_{\mathbb{R}^p} \right\}$$

Changing the metric : $\|\theta - \hat{\theta}_j\|^2_{\Sigma} = (\theta - \hat{\theta}_j)^T \Sigma (\theta - \hat{\theta}_j)$ implies that (with Σ a positive-definite matrix)

$$\hat{\theta}_{j+1} = \arg\min_{\theta \in \mathbb{R}^p} \left\{ M_n(\theta) + \frac{1}{2\gamma_j} \|\theta - \hat{\theta}_j\|^2_{\Sigma} \right\}$$

satisfies

$$\hat{\theta}_{j+1} = \hat{\theta}_j - \gamma_j \Sigma^{-1} \nabla M_n(\hat{\theta}_j)$$

Riemannian setting : $\Sigma = \Sigma(\hat{\theta}_j)$

Natural gradient descent

Use the Fisher-Rao norm : $\|\theta\|_{\mathrm{fr}}^2 = \langle \theta, I(\theta)\theta \rangle$ where

$$I(\theta) = \mathbb{E}\left(\nabla_\theta L(Y, f_\theta(X)) \otimes \nabla_\theta L(Y, f_\theta(X))\right)$$

Numerical experiments in Liang et al. (2018) on the computation of $\hat{\theta}$ by **natural gradient descent** (with a stochastic version) :

$$\hat{\theta}_{j+1} = \hat{\theta}_j - \gamma_j \, I(\hat{\theta}_j)^{-1} \nabla M_n(\hat{\theta}_j)$$

Publicité

You are welcome to the next seminar at IMB on the mathematical aspects of Deep Learning !

31 mai 2018 à 11h00 - IMB - Salle de Conférences

Salem Said (IMS)

Presentation of the paper "**Practical Riemannian Neural Networks**" by G. Marceau-Caron & Y. Ollivier