On Convex Neural Networks

Mathematical Foundations of Data Science

Eloïse Berthier

December 6, 2018

1/2 page: What problem(s) is studied ? Why is it relevant ? What solution(s) is proposed ? Which contributions (theory, numerics, etc) ?

1 Introduction

While the use of neural networks has dramatically increased the performance of some recognition systems, there still lacks a clear mathematical understanding of this success. One of the key issues is that optimizing neural networks is a non-convex problem, hence training algorithms may not return a global minimum.

Yet in practice, training large neural networks often results in satisfying solutions, which are empirically considered *not too far* from a global minimum. Thus it suggests that there may be some implicit assumptions on the data or on the model that are usually verified and could explain this regularity, somehow making the problem *more convex* than at first sight.

There have been several attempts to study this phenomenon, heading in different directions. Here are some examples.

Some rely on convex relaxations of the original optimization problem. [Zhang et al., 2016] define a convex relaxation for training convolutional neural networks. They show that in the case of a two-layer CNN, the generalization error of the solution to the convex relaxed problem converges to that of the best possible two-layer CNN.

In [Haeffele and Vidal, 2017] are studied conditions under which the optimization landscape for the non-convex optimization problem is such that all critical points are either global minimizers or saddle points. [Li et al., 2018] provide a tool to visualize the loss surface along random directions near the optimal parameters.

There is also a series of some very recent contributions on the problem: [Du et al., 2018b], [Du et al., 2018a] and [Zou et al., 2018]. They rely on the analysis of the dynamics of the prediction space rather than of the parameter space, the former being governed by the spectral property of a Gram matrix. If the data is *not degenerate*, this Gram matrix's least eigenvalue is lower bounded, and randomly initialized gradient descent linearly converges to a global optimum.

In the infinite-width limit, [Jacot et al., 2018] relate the evolution of a neural network function to a kernel that they call the *neural tangent kernel*. Convergence to a global minimum can then be related to the positive-definiteness of the limiting kernel, and which is the case when the data are on a sphere.

In this report, we concentrate on another approach called *convex neural networks*. The core idea was introduced by [Bengio et al., 2006] and focuses on one hidden-layer neural networks. Instead of optimizing the weights of a network with a fixed number of hidden units, we could consider the equivalent problem of optimizing on the set of all the possible hidden unit functions. The training problem is convex with respect to the parameters of the model (which are now only the ouput weights).

Of course, while this approach solves the non-convexity issue, it introduces some other difficulties. These are extensively studied in [Bach, 2017], on which we will mainly focus in this report. A complete analysis of the generalization performance of convex neural networks is derived, in particular showing that high-dimensional non-linear variable selection may be achieved, without any strong assumption regarding the data. Yet this approach requires to solve a convex problem in infinite dimension and is only possible if the non-convex subproblem of adding a new hidden unit can be solved efficiently. Finding a polynomial-time algorithm solving this subproblem induced by the conditional gradient method is still an open question.

[Chizat and Bach, 2018] focus on solving the same convex problem, but avoid the potentially NP-hard problem of optimizing with conditional gradient. Instead, they discretize the unknown measure as a mixture of particles. This approach is inspired by optimal transport and brings strong asymptotical global optimality guarantees for gradient flow optimization. It also provides simple numerical experiments.

Looking at all these different approaches, we may notice some recurrent patterns in the assumptions that are commonly made:

- *one hidden-layer models*: most of the articles focus on simple neural network models, that already include a wide variety of classical supervised learning configurations, but do not directly cover deep neural networks.
- *over parametrization*: all the good generalization properties are derived in the case of an over parametrized model, yet the key question is *how much* over parametrization is needed.
- *positive homogeneity*: most of the results are achieved in the case of positively homogeneous activation functions. It could explain the experimental advantage of using ReLU functions over others in deep learning.
- *proper initialization*: several articles rely on the conservation of an initial regularity through the optimizing process, for instance using gradient flows, and so emphasize the role of the initialization.

2 Toward a Convex Problem

2.1 General layout

We consider single hidden-layer neural networks. It defines a class of prediction functions on $\mathbb{R}^d$ parametrized as: $f_{\eta,b,v}(x) = \sum_{j=1}^{k} \eta_j \sigma(v_j^T x + b_j)$, where σ is a fixed activation function, k is the number of hidden units, v_j and b_j are the weight and bias of layer j and $(\eta_j)_{j=1,\dots,k}$ the weights of the output layer.

We can eliminate the b_j by integrating it into v_j and adding a one on top of x.

We will mainly consider a fixed non-decreasing and positively homogeneous activation function of some integer degree, i.e. $\sigma(u) = (u)_+^\alpha$, for some $\alpha \in \mathbb{N}$ (it includes ReLU and hard-thresolding functions). We will also later consider sigmoid activation functions in the numerical experiments.

2.2 From non-convexity to convexity

For a fixed value of k, training a neural network is an empirical risk minimization problem: $\min_{\eta,v} \frac{1}{n}\sum_{i=1}^{n} \ell(f_{\eta,v}(x_i), y_i) + \Omega(\eta, v)$, where ℓ is a smooth convex loss function (convex in its first argument) and Ω a convex regularization function, and (x_i, y_i) the observations in $\mathcal{X} \times \mathbb{R}$. This is not a convex problem because of the non-convexity of the prediction function.

Now consider the set $\mathcal{F}_\mathcal{V}$ of all the possible hidden unit functions $\varphi_v : \mathcal{X} \to \mathbb{R}$, with $\mathcal{X}$ any measurable space. We will call it the set of basis functions. It is parametrized by the compact topological space $\mathcal{V}$ of all possible hidden unit weight vectors. We assume that for all $x \in \mathcal{X}$, the functions $v \mapsto \varphi_v(x)$ are continuous. To represent any affine function, $\mathcal{V}$ has dimension $d + 1$ for inputs in $\mathcal{X}$ of dimension d.

Let $\mathcal{W}$ be the Hilbert space of functions from $\mathcal{F}_\mathcal{V}$ to $\mathbb{R}$, with $\cdot$ an inner product. For any $x \in \mathcal{X}$, we define $h(x)$ as the function that maps any $\varphi_v \in \mathcal{F}_\mathcal{V}$ to $\varphi_v(x)$. $h(x)$ is an element of $\mathcal{W}$. $h(x)$ can be seen as the vector of activations of the hidden units when we observe x as input. An element η of $\mathcal{W}$ can also be understood as the output weights vector in a neural network. We extend the definition of Ω to be a convex regularization functional from $\mathcal{W}$ to $\mathbb{R}$.

We can define the following problem:

$$\min_{\eta \in \mathcal{W}} \mathcal{C}(\eta) = \frac{1}{n}\sum_{i=1}^{n} \ell(\eta \cdot h(x_i), y_i) + \Omega(\eta) \tag{1}$$

Lemma 2.1 *Problem (1) is a convex minimization problem.*

The proof is straightforward: any Hilbert space $\mathcal{W}$ is convex, $\ell(\eta \cdot \varphi(x_i), y_i)$ is convex in η and by additivity the objective function also is.

This problem formulation is from [Bengio et al., 2006]. It is a bit hard to handle as it introduces many notations that are not all explicit. Still it provides an intuitive idea of the problem we want to define: considering all the possible hidden units we could pick, how to combine them to make a good prediction, that is, how to choose the output weight η to minimize $\mathcal{C}$.

In this problem the penalization term is crucial. Unlike the initial optimization problem where the number of hidden units was fixed, it is now part of the optimization problem. Without any regularization, the solution could have an arbitrary large number (possibly infinite) of non-zero variables η_j. We will later see that a proper regularization can ensure a finite number of selected units, and so a finite model.

$\mathcal{W}$ is the space of the output weights, and since it has infinite dimension, it is handled as a space of functions. What makes it is a bit entangled is that eventually $\mathcal{W}$ is defined as a space of functions (output layer) of functions (hidden layer) of $x \in \mathcal{X}$, but it can also be seen as a simple vector (in potentially infinite dimension). The vector of activations of the hidden layer for a fixed input x is also an element of $\mathcal{W}$ because it has the same shape as η. Notice that due to these notations, this formulation could hardly be generalized to a greater number of layers.

For the rest of this report, we will focus on the more general problem formulation in [Bach, 2017], which defines two spaces $\mathcal{F}_1$ and $\mathcal{F}_2$, for which we do not explicitely define an output Hilbert space $\mathcal{W}$. In fact, we do not actually manipulate the parameters of the neural network, but we will rather directly optimize on end-to-end neural network fonctions $f \in \mathcal{F}_1$ or $\mathcal{F}_2$. To take the penalty into account, we will have to define a notion of norm on these spaces, and discrete Radon measures will now play the role of the output weights η.

2.3 Variation norm

To properly define $\mathcal{F}_1$, we have to introduce Radon measures.

Definition 2.1 *Real-valued Radon measures are continuous linear forms on the space of continuous functions from $\mathcal{V}$ to $\mathbb{R}$, equipped with the uniform norm.*

The total variation norm $|\mu|(\mathcal{V})$ of a Radon measure μ is equal to the supremum of $\int_{\mathcal{V}} g(v)\mathrm{d}\mu(v)$, over all continuous functions g with values in $[-1, 1]$. When μ has a density with respect to a fixed probability measure τ, $d\mu(v) = p(v)d\tau(v)$, then this is the L^1 norm of this density:

$$|\mu|(\mathcal{V}) = \int_{\mathcal{V}} |p(v)|\mathrm{d}\tau(v)$$

When the measure is discrete, that is $\mu = \sum_{j=1}^{k} \eta_j \delta_{v_j}$ the total variation of μ is the ℓ^1-norm of η:

$$|\mu|(\mathcal{V}) = \sum_{j=1}^{k} |\eta_j|$$

Definition 2.2 *$\mathcal{F}_1$ is the space of functions f that can be written as $f(x) = \int_{\mathcal{V}} \varphi_v(x) \mathrm{d}\mu(v)$, where μ is a signed Radon measure on $\mathcal{V}$ with finite total variation.*

Definition 2.3 *The variation norm of $f \in \mathcal{F}_1$ with respect to $\mathcal{V}$ is the infimum of $|\mu|(\mathcal{V})$ over all decompositions of f as $f = \int_{\mathcal{V}} \varphi_v \mathrm{d}\mu(v)$. It is a norm γ_1 on $\mathcal{F}_1$.*

If we assume in this definition that μ must have a density with respect to τ with full support on $\mathcal{V}$, then $\gamma_1(f)$ is the infimum of $|\mu|(\mathcal{V}) = \int_{\mathcal{V}} |p(v)| \mathrm{d}\tau(v)$ over all integrable functions p such that $f(x) = \int_{\mathcal{V}} p(v)\varphi_v(x) \mathrm{d}\tau(v)$. This defines the same norm $\gamma_1(f)$ because all Radon measures are limits of measures with densities.

If f has a finite number of neurons, $f(x) = \sum_{j=1}^{k} \eta_j \varphi_{v_j}(x)$, the infimum is attained when μ is a mixture of k Diracs at v_j with weights η_j, and then $\gamma_1(f) = ||\eta||_1$. The variation norm of f controls the ℓ^1 norm of the output weights, so it could be used as a penalization term. We will rather express it as a constraint over the space of admissible f, which is equivalent up to a constant.

2.4 Problem formulation in $\mathcal{F}_1$

We define the convex problem of minimizing a functional J on functions restricted to $\hat{\mathcal{X}}$ a subset of $\mathcal{X}$, that is:

$$\min_{f_{|\hat{\mathcal{X}}} \in \mathbb{R}^{\hat{\mathcal{X}}} \ s.t. \ \gamma_{1|\hat{\mathcal{X}}}(f_{|\hat{\mathcal{X}}}) \leq \delta} J(f_{|\hat{\mathcal{X}}}) \tag{2}$$

where $\gamma_{1|\hat{\mathcal{X}}}(f_{|\hat{\mathcal{X}}})$ is the infimum of the total variation of a measure over the decompositions defined for $f_{|\hat{\mathcal{X}}}$ instead of f.

This subset will typically be a finite set of observations $x_1, ..., x_n$ among a potentially infinite set $\mathcal{X}$. With this penalization, the empirical risk minimization problem is well represented: J measures the quality of the fit to the observations, and the constraint controls the complexity of the model. Note that solving this optimization problem provides an optimal measure μ and a decomposition of $f_{|\hat{\mathcal{X}}}$, and thus also gives an expression for f by extending this decomposition to any $x \in \mathcal{X}$.

The following result highlights the role of controling the variation norm.

Theorem 2.2 [From Carathéodory's theorem for cones] *If $\hat{\mathcal{X}}$ has only n elements, the solution f to problem (2) may be decomposed into at most n functions φ_v, that is, μ is supported by at most n points of $\mathcal{V}$ in the expression:*

$$f(x) = \int_{\mathcal{V}} \varphi_v(x) \mathrm{d}\mu(v)$$

However, we do not know in advance which will be these n functions, whereas if we consider the representer theorem for RKHS, the functions are known to be kernels centered in the observations.

2.5 Problem formulation in $\mathcal{F}_2$

Following the previous results, we may define another space of functions $\mathcal{F}_2$ that has a reproducing property.

We previously noticed that in the definition of the variation norm, the infimum over measures could also be taken over densities with respect to a base measure, hence we could minimize $|\mu|(\mathcal{V}) = \int_{\mathcal{V}} |p(v)| \mathrm{d}\tau(v)$ over all integrable functions p such that $f(x) = \int_{\mathcal{V}} p(v)\varphi_v(x)\mathrm{d}\tau(v)$. Instead of taking the L^1 norm of the densities, we may consider the infimum of their L^2 norm $\int_{\mathcal{V}} |p(v)|^2 \mathrm{d}\tau(v)$ over all decompositions of f. It defines a squared norm γ_2^2.

Proposition 2.1 *If $\mathcal{V}$ is compact, the space $\mathcal{F}_2$ of functions with finite γ_2 norm is a reproducing kernel Hilbert space (RKHS), with positive definite kernel $k(x,y) = \int_{\mathcal{V}} \varphi_v(x)\varphi_v(y)\mathrm{d}\tau(v)$.*

The empirical risk minimization problem in $\mathcal{F}_2$ can be solved by sampling m i.i.d hidden units $v_1, ..., v_m$ and learning a function of the form $\frac{1}{m}\sum_{j=1}^{m} \eta_j \varphi_{v_j}(x)$ with a penalization on the ℓ^2 norm of η. When m tends to infinity, the approximate kernel $\hat{k}(x,y) = \sum_{j=1}^{m} \varphi_{v_j}(x)\varphi_{v_j}(y)$ tends to $k(x,y)$.

[Le Roux and Bengio, 2007] provide explicit formulas for the kernel with the hard-thresholding activation fonction, and similars results can be derived for positive homogeneous functions of degree 1 and 2. All the optimization can then be performed in finite dimension, whereas this is not possible in $\mathcal{F}_1$.

3 Theoretical Analysis of errors

nbr of units / sample complexity / curse of dimensionality

Sample complexity is exponential in dimension if only assume lispchitz continuous (curse of dimensionality). Adding some structure: gets polynomial. We would like 1 HL Neural networks to adapt to these structures. Generalization properties: adaptivity to structure and variable selection, assuming f* has a simpler expression.

Problem 4: Empirical risk minimization (cf video) min 1/n sum l(f(xi), yi) st gamma1(f)<delta or gamma2(f)

3.1 Errors decomposition

3 errors decomposition.

approximation + estimation + optimization. Leave the last one for part 4.

3.2 Generalization error

Generalization error = approximation error + estimation error

1) estimation: rademacher analysis for uniform deviations (...) O(delta/sqrt n)

2) approximation: gamm1 <= gamma2 donc F1 inclus dans F2

smoothness asumptions -> appartenance a F1/F2. gamma1 and gamma2 exist under smoothness asumptions Approximations par fonctions avec gamma < delta, bound depend de d. proof with spherical harmonics.

Adaptivity: can we do better when subspace s, replace d par s? -> the difference between F1 and F2. approximation errors for lipschitz continuous gives bounds. Function of d. Only adaptive for F1 and not F2. Reason: F2 inclus dans F1. Too small, lack of adaptivity. Only learning smooth functions, allow singularity with l1. l2 is too small. l1 performs selection on hidden units, induces sparsity. Not l2. For instance cannot put diracs in l2 (à préciser).

Conclusion: F1 is good, F2 not enough

4 An Optimization Problem

Optimizing in F1 is hard...

4.1 Frank Wolfe / Learning from basis functions

Frank Wolfe / conditional gradient.

Barron: one way to provide a finite decomposition of functions in F1 with error epsilon and control on the number of neurons depending on gamma1. -> Using FW.

Bail des zonotopes donne une meilleure borne sur le nombre de neurones ?

Link to NP hard problems.
Open problems.

4.2 The Particle Gradient Flow Approach

A more general result, can be applied to this problem. Tackle an optimization problem and provide global convergence guarantees.

5 Numerical Experiments

Main body (10 pages) : Presentation of the method(s). Theoretical guarantees. Numerics.

6 Conclusion and perspectives

Conclusion and perspective (1 page) Summary of the result obtained: pros and cons (limitation, problems, error in the articles, etc) Possible improvement/extension

References

[Bach, 2017] Bach, F. (2017). Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53.

[Bengio et al., 2006] Bengio, Y., Roux, N. L., Vincent, P., Delalleau, O., and Marcotte, P. (2006). Convex neural networks. In *Advances in neural information processing systems*, pages 123–130.

[Chizat and Bach, 2018] Chizat, L. and Bach, F. (2018). On the global convergence of gradient descent for over-parameterized models using optimal transport. *arXiv preprint arXiv:1805.09545*.

[Du et al., 2018a] Du, S. S., Lee, J. D., Li, H., Wang, L., and Zhai, X. (2018a). Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*.

[Du et al., 2018b] Du, S. S., Zhai, X., Poczos, B., and Singh, A. (2018b). Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*.

[Haeffele and Vidal, 2017] Haeffele, B. D. and Vidal, R. (2017). Global optimality in neural network training.

[Jacot et al., 2018] Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*.

[Le Roux and Bengio, 2007] Le Roux, N. and Bengio, Y. (2007). Continuous neural networks. In *Artificial Intelligence and Statistics*, pages 404–411.

[Li et al., 2018] Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018). Visualizing the loss landscape of neural nets. In *Neural Information Processing Systems*.

[Zhang et al., 2016] Zhang, Y., Liang, P., and Wainwright, M. J. (2016). Convexified convolutional neural networks. *arXiv preprint arXiv:1609.01000*.

[Zou et al., 2018] Zou, D., Cao, Y., Zhou, D., and Gu, Q. (2018). Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*.

(a) Average return during the gradient ascent (b) Mean parameter during the gradient ascent