"Mathematics of Deep Learning" Review: Insights & Analysis

Juan Carrascal
B.S. in Mathematics
Universidad Nacional de Colombia
jdcarrascali@unal.edu.co

María Perpiñán
Major in Statistics
Universidad Nacional de Colombia
maperpinanb@unal.edu.co

Edward Soto
B.S. in Mathematics
Universidad Nacional de Colombia
edsotom@unal.edu.co

Mayra Vega
M.S. in Applied Mathematics
Universidad Nacional de Colombia
mvegan@unal.edu.co

I. Introduction

The chosen article reviews recent work that aims to provide a mathematical justification for several empirical properties of deep networks, such as global optimality, geometric stability, and invariance of the learned representations.
The structure of the article is:

- The problem of training deep networks and conditions for global optimality.
- The invariance and stability properties of CNN.
- Structural properties of deep networks.
- Information-theoretic properties of deep representations.

II. Global Optimality in Deep Learning

In this section, we study the problem of learning the parameters $W = \{W^l\}_{l=1}^L$ of a deep network from N training examples (X, Y). The problem of learning the network weights W is formulated as the following optimization problem

$$\min_{\{W^l\}_{l=1}^L} \mathcal{L}(Y, \Phi(X, W^1, \ldots, W^L)) + \lambda\Theta(W^1, \ldots, W^L), \tag{1}$$

where $\mathcal{L}(Y, \Phi)$ is the loss function and Θ is the regularization term.

A. The challenge of non-convexity in neural network training

For most deep networks, $\mathcal{L}(Y, \Phi)$ is a convex function, however the map $\Phi(X, Y)$ is not due to the product of the W^l variables and the non-linearities of the activation functions. This can become a problem for the following reasons:

- The optimization algorithms (gradient descent, stochastic gradient descent, back-propagation, among others) we use to solve (1) only converge when the function to be minimized/maximized is convex, in cases where the function is not, there is no guarantee of convergence for these methods.
- For non-convex functions, the distribution of their critical points may include saddle points, constant points, global and local minima, global and local maxima as shown in Fig. 1. Although ideally we want to reach points like (b) and (d), we may be reaching points like (h).

Although the non-convexity problem has not been solved, certain important facts have been proved experimentally or theoretically that help us to understand the behavior of this type of functions in high dimensions. Assuming certain properties in the data, in the weights W and in the distribution of the output of the network activation functions, in [1] they prove the following:

- For large-size networks, most local minima are equivalent and yield similar performance on a test set. This implies that as long as the necessary conditions are met, no matter at which random point (W) the network is initialized, a good[1] local minimum is always reached.
- The probability of finding a bad (high value) local minimum is non-zero for small-size networks and decreases quickly with network size.

At the same time, they state that we should not be interested in finding the global minimum as this would cause the network to overfit the training data.

B. Global optimality for positively homogeneous networks

In the vast majority of cases, theoretical optimality conditions are obtained by making assumptions on the distribution of data and/or network weights. It would be better to be able to obtain such results without these assumptions since they usually cannot be guaranteed in real life data. In [2], [3] they prove some important results just making some assumptions on

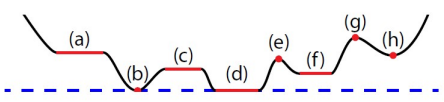

Fig. 1. Example of critical points of a non-convex function (shown in red). (a,c) Plateaus. (b,d) Global minima. (e,g) Local maxima. (f,h) Local minima.

[1]By good we mean that the local minimum is always close to the global minimum.

the activation and regularization functions. If Φ and Θ are the sum of positively homogeneous functions of the same degree, we have that:

- Saddle points and plateaus are the only critical points that one needs to be concerned with due to the fact that for networks of sufficient size, local minima that require one to climb the objective surface to escape from, such as (f) and (h) in Fig. 1, are guaranteed not to exist.
- Any local minimizer of the non-convex optimization problem

$$\min_{\{W^l\}_{l=1}^{L}} \mathcal{L}(Y, \Phi_r(X, W^1, \ldots, W^L)) + \lambda \sum_{i=1}^{r} \Theta(W^1, \ldots, W^L), \tag{2}$$

  such that $(W_{i_0}^1, \ldots, W_{i_0}^L) = (0, \ldots, 0)$ for some $i_0 \in \{1, \ldots, r\}$ is a global minimizer of (2). Moreover, $X = \Phi_r(W^1, \ldots, W^L)$ is a global minimizer of (1).

The latter theorem could provide an explanation as to why networks with ReLU or max-pooling activation functions perform so well, since both are positive homogeneous functions.

III. Geometric stability in deep learning

Next, we will examine some characteristics found in computer vision tasks that can be found within deep learning methods. Specifically, convolutional architectures play a crucial role in the effectiveness of many deep learning vision models. Within these architectures, there exists a concept of geometric stability, which offers a potential framework for comprehending their achievements.

Representation. In image analysis applications, images can be thought of as functions on the unit square $\Omega = [0, 1]^2$. This representation is given by: $X \in L^2(\Omega)$, where

$$X : \Omega \longrightarrow \mathbb{R} \quad \text{with} \quad \int_{-\infty}^{\infty} |X(t)|^2 dt < \infty.$$

The consistency of this representation is given by the analytical and geometrical properties of $L^2(\Omega)$ such as the existence presence of orthogonality and density onto itself.

Additionally, let's define f (the unknown function we want to learn) such as $f : L^2(\Omega) \longrightarrow \mathcal{Y}$. It has been observed that in some computer vision tasks, f has some of the following properties:

- *Stationarity.* Given a translation operator

$$\mathcal{T}_v X(u) = X(u - v) \quad u, v \in \Omega, \tag{3}$$

  f can have the following properties:
  - *Invariance:* $f(\mathcal{T}_v X) = f(X)$ for any $X \in L^2(\Omega)$ and $v \in \Omega$. This is typically the case in object classification tasks.
  - *Equivariance:* $f(\mathcal{T}_v X) = \mathcal{T}_v f(X)$ for any $X \in L^2(\Omega)$ and $v \in \Omega$. Observed in tasks of object localization and semantic segmentation.
- *Local deformations.* Mathematically, given a smooth vector field $\tau : \Omega \longrightarrow \Omega$, if a function L_τ is such that:

$$L_\tau X(u) := X(u - \tau(u)), \tag{4}$$

  we say that L_τ is a deformation. This kind of functions can model local translations, changes in viewpoint and rotations. We want f to be stable under local deformations.

Important facts of convolutional architectures.

- Most tasks studied in computer vision are not only translation invariant/equivariant, but, more importantly, also stable with respect to local deformations [4].
- Whereas long-range dependencies indeed exist in natural images and are critical to object recognition, they can be captured and down-sampled at different scales.
- CNNs strike a good balance in terms of approximation power, optimization, and invariance [5].
- Recently there has been an effort to extend the geometric stability priors to data that is not defined over an Euclidean domain [6].

Example. Brain tumor segmentation task makes use of an architecture called *"encoder-decoder"*. We can observe the mentioned geometric properties in this experiment.

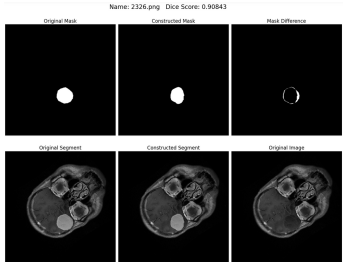

Fig. 2. Brain tumor segmentation task.

IV. Structure Based Theory for Deep Learning

In this section, the authors discuss the relationship between the structure of the data and deep neural networks. The key challenge is to understand the good generalization observed in practice for deep networks with a large number of parameters or deep architectures. Recall that the generalization error is the difference between the expected error and the empirical error of a network, The generalization error is then given as:

$$\mathrm{GE}(\Phi) = \left| \ell_{\exp}(\Phi) - \ell_{\text{emp}}(\Phi) \right|.$$

Where:

$$\ell_{\text{emp}}(\Phi) = \frac{1}{N} \sum_{X_i \in \Upsilon_N} \ell\left(Y_i, \Phi\left(X_i, W\right)\right),$$

represents the empirical loss, which measures the discrepancy between the true labels and the predicted labels of a model on the training data set.
And

$$\ell_{\exp}(\Phi) = \mathbb{E}_{(X,Y)\sim P}[\ell(Y, \Phi(X, W))].$$

is the expected loss which is a measure of how well the model performs on unseen or new data. It represents the average loss that the model is expected to have when making predictions on data drawn from the same underlying distribution as the training data.

Various measures such as VC-dimension, Rademacher or Gaussian complexities, and algorithm robustness have been used to bound the generalization error in deep networks. However, they don't fully explain the good generalization observed in practice for deep networks with a large number of parameters or deep architectures.

When focusing on the VC-dimension, the VC-dimension of halfspaces in Rd is d + 1, which means that the number of examples required to learn halfspaces grows with the dimensionality of the problem. This poses a problem when the dimensionality, d, is very large because it implies that an extremely large number of examples would be needed for accurate learning, however in practice we have seen that deep neural networks can learn when the number of parameters exceed the number of data points in our dataset.

An alternative approach to bound the generalization error is to consider the network's classification margin. The classification margin for a training sample $s_i = (\mathbf{x}_i, y_i)$ measured by a metric d is defined as

$$\gamma^d(s_i) = \sup\{a : d(\mathbf{x}_i, \mathbf{x}) \leq a \Longrightarrow g(\mathbf{x}) = y_i \forall \mathbf{x}\}.$$

The classification margin introduces an additional assumption on the underlying data distribution called "separability with a margin of a." This assumption defines that if the data is separable with a margin a, then the sample complexity (the number of examples required for learning) is reduced.

This is a significant result because it suggests that even if the dimensionality is very large or even infinite, as long as the data satisfies the assumption of separability with a margin, the sample complexity can still be small. In other words, the additional assumption of separability with margin provides a way to overcome the limitations imposed by the VC-dimension and allows for efficient learning with a reduced number of examples.

V. Information-theoretic properties of deep representations

It is well know that for binary classification tasks, the standard loss function is the binary-log loss that allow us maximize the log-likelihood of networks in the training phase. In higher dimensional classification problems (more than two classes) the standard loss function is the empirical cross-entropy defined as:

$$\ell(W) = \mathbb{E}_{P(X,Y)}(-\log(\Phi(X, W))),$$

where $\mathbb{E}_{P(X,Y)}$ is the expected value over the joint distribution of the input X and the target Y, and $\Phi(X, W)$ denotes the output of the network.

In this section the authors introduce two regularized loss functions within an information theory framework that involve the above loss function. The first one cover the overfitting problem that networks present in the learning phase when proposed model gives so much liberty to network parameters, in therms of what they learn about the data. The second one cover the variance problem, this is, when the network doesn't generalize good to new data. As is natural in real world scenarios, this is because presence of noise in training set.

The first regularized loss function is defined as follow:

$$\ell(W) = H(Y|X, W) + \lambda KL(P(W|X, Y)||P(W)),$$

where $P(W)$ is a prior on the weights of the network, H is the empirical conditional cross-entropy and KL is the Kullback–Leibler divergence. The empirical conditional cross-entropy acts as the classical loss function to be minimized; from the data X and actual values for W we want a good prediction of Y. The Kullback-Leibler divergence computes the difference between two distributions; we want the actual distribution of the weights $P(W||X, Y)$ be restricted by the prior distribution $P(W)$. This regularization acts as a limiter of the information the weights can store in the learning phase.

The second regularized loss function is defined as:

$$\ell(W) = H(Y|Z, W) + \lambda I(Z; X),$$

where Z is a stochastic representation of X learned by a layer of the network. Again, H is the empirical conditional cross-entropy. The I term is the mutual information between Z and X; as Z is a learnable representation of X the mutual information how much Z can looks like X. As much as we want to regularize, we are limitating the quantity of noise the network is learning.

VI. Conclusions

The theoretical study of empirical properties of deep networks opens up new mathematical problems that can be of interest not only to the mathematical community. Achieving solutions to these theoretical problems can lead to substantial improvements in many other areas of knowledge that use neural networks as a tool. For mathematicians, the theoretical problems covered in the selected paper can be approached from various perspectives, and it serves as an invitation to explore new areas.

References

[1] A. Choromanska, M. Hena, M. Mathieu, G. B. Arous, and Y. LeCun, "The loss surfaces of multilayer networks," *International Conferenceon Artificial Intelligence and Statistics*, pp. 192,204, 2015.

[2] B. D. Haeffele and R. Vidal, "Global optimality in neural network training," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[3] ——, "Global optimality in tensor factorization, deep learning, and beyond," *arXiv*, 2015.

[4] S. Mallat., "Understanding deep convolutional networks," *Phil. Trans. R. Soc. A*, 2016.

[5] D. R. P. Felzenszwalb R. Girshick, D. McAllester, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[6] M. Bronstein, Y. L. J. Bruna, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *arXiv preprint arXiv:1611.08097*, 2016.