Global Minima of Overparametrized Neural Networks

Ph.D Student: Vlad Raul Constantinescu[1,2]
Advisor: Prof. dr. Ionel Popescu[3,4]

[1]Şcoala doctorală a Facultăţii de Matematică, Universitatea din Bucureşti
[2]Institutul de Statistică Matematică şi Matematică Aplicată "Gheorghe Mihoc- Caius Iacob" al Academiei Române
[3] Facultatea de Matematică şi Informatică, Universitatea din Bucureşti
[4] Institutul de Matematică al Academiei Române "Simion Stoilow"

WORKSHOP "Doctoral Research Days"

Abstract

In this presentation we give an answer to the following questions:

- In which conditions can a neural network interpolate a data set?
- How many solutions do we have for our interpolation problem?
- Can we give a description for the Hessian eigenspectrum of the loss function evaluated at a global minima point?

Abstract

In this presentation we give an answer to the following questions:

- In which conditions can a neural network interpolate a data set?
- How many solutions do we have for our interpolation problem?
- Can we give a description for the Hessian eigenspectrum of the loss function evaluated at a global minima point?

Abstract

In this presentation we give an answer to the following questions:

- In which conditions can a neural network interpolate a data set?
- How many solutions do we have for our interpolation problem?
- Can we give a description for the Hessian eigenspectrum of the loss function evaluated at a global minima point?

- A mathematical formulation for the machine learning problem is the following. Suppose we have a data set $(x_i, y_i)_{i=\overline{1,N}}$ sampled independently after a distribution μ, where x_i are the input data and y_i are the associated labels. In this context, the problem is to determine a function $f : \mathbb{R}^p \to \mathbb{R}$ such that we have $y_i = f(x_i)$.
- Finding such a function is done in a familiy of functions parametrized after a parameter $\theta \in \mathbb{R}^k$. Finding a good f_θ means that we have to define a loss function $L(f_\theta(x), y)$ which measures how far is $f_\theta(x)$ of y.
- In principle we want to minimise the expected value $\mathbb{E}_\mu[L(f_\theta(x), y)]$. Unfortunately, the distribution μ is not known, but instead we can aproximate our expected value with the empirical version

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(f_\theta(x_i), y_i)$$

- A mathematical formulation for the machine learning problem is the following. Suppose we have a data set $(x_i, y_i)_{i=\overline{1,N}}$ sampled independently after a distribution μ, where x_i are the input data and y_i are the associated labels. In this context, the problem is to determine a function $f : \mathbb{R}^p \to \mathbb{R}$ such that we have $y_i = f(x_i)$.
- Finding such a function is done in a familiy of functions parametrized after a parameter $\theta \in \mathbb{R}^k$. Finding a good f_θ means that we have to define a loss function $L(f_\theta(x), y)$ which measures how far is $f_\theta(x)$ of y.
- In principle we want to minimise the expected value $\mathbb{E}_\mu[L(f_\theta(x), y)]$. Unfortunately, the distribution μ is not known, but instead we can aproximate our expected value with the empirical version

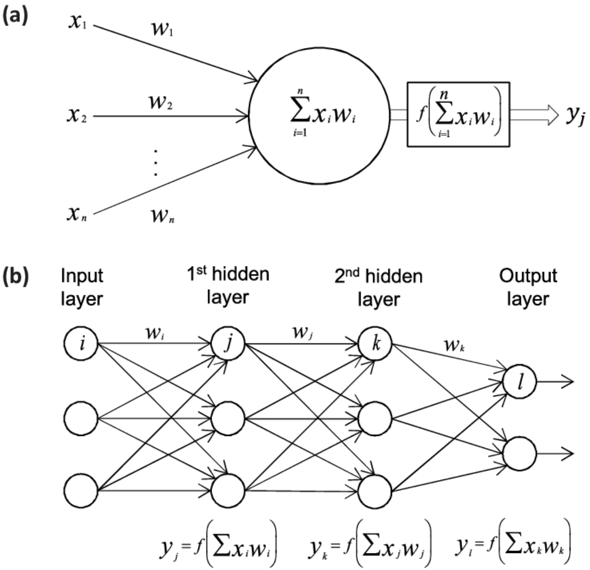
$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} L(f_\theta(x_i), y_i)$$

The machine learning problem

- A mathematical formulation for the machine learning problem is the following. Suppose we have a data set $(x_i, y_i)_{i=\overline{1,N}}$ sampled independently after a distribution μ, where x_i are the input data and y_i are the associated labels. In this context, the problem is to determine a function $f : \mathbb{R}^p \to \mathbb{R}$ such that we have $y_i = f(x_i)$.
- Finding such a function is done in a familiy of functions parametrized after a parameter $\theta \in \mathbb{R}^k$. Finding a good f_θ means that we have to define a loss function $L(f_\theta(x), y)$ which measures how far is $f_\theta(x)$ of y.
- In principle we want to minimise the expected value $\mathbb{E}_\mu[L(f_\theta(x), y)]$. Unfortunately, the distribution μ is not known, but instead we can aproximate our expected value with the empirical version

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(f_\theta(x_i), y_i)$$

Feedforward neural networks



- A neural network, with activation function σ, can be described in matrix form as

$$f_{w,b}(x) = W_l\sigma(W_{l-1}\sigma(\dots\sigma(W_1x - b_1)\dots) - b_{l-1}) - b_l,$$

where $W_i \in \mathcal{M}_{n_{i-1}\times n_i}(\mathbb{R}), b_i \in \mathbb{R}^{n_i}$ and $n_0 = p, n_l = 1$.

- Moreover, we use the convention that σ applied on a vector is simply the component-wise evaluation:

$$\sigma(v_1, v_2, \dots, v_k) = (\sigma(v_1), \sigma(v_2), \dots, \sigma(v_k)).$$

- A neural network, with activation function σ, can be described in matrix form as

$$f_{w,b}(x) = W_l\sigma(W_{l-1}\sigma(\ldots\sigma(W_1x - b_1)\ldots) - b_{l-1}) - b_l,$$

  where $W_i \in \mathcal{M}_{n_{i-1}\times n_i}(\mathbb{R}), b_i \in \mathbb{R}^{n_i}$ and $n_0 = p, n_l = 1$.
- Moreover, we use the convention that σ applied on a vector is simply the component-wise evaluation:

$$\sigma(v_1, v_2, \ldots, v_k) = (\sigma(v_1), \sigma(v_2), \ldots, \sigma(v_k)).$$

Let $\sigma \in C(\mathbb{R})$. We define the set $\mathcal{M}(\sigma)$ to be

$$\mathcal{M}(\sigma) := span\{\sigma(w \cdot x - b) : w \in \mathbb{R}^n, b \in \mathbb{R}\}.$$

We are interested for which σ, $\mathcal{M}(\sigma)$ is dense in $C(\mathbb{R}^n)$, in the topology of uniform convergence on compacts.

THEOREM 1([PIN99],THEOREM 3.1)

Let $\sigma \in C(\mathbb{R})$. Then $\mathcal{M}(\sigma)$ is dense in $C(\mathbb{R}^n)$, in the topology of uniform convergence on compacts, if and only if σ is not a polynomial function.

The Interpolation Problem

A consequence of Theorem 1 is that neural networks, with a continuous activation function and not a polynomial function, can interpolate any data set. One of our original results is a generelisation of this consequence for activation functions which are locally integrable and not polynomials.

Theorem 2

Let $(x_i, y_i)_{i=\overline{1,d}}$ be a data set with $x_i \in \mathbb{R}^p, y_i \in \mathbb{R}$, and the x_i are distinct. Assume that the activation function σ is locally integrable and not a polynomial function of degree less than $d-2$ almost everywhere. Then, in the familiy a feedforward neural nets of l hidden layers, with last hidden layer $h \geq d$, and remaining hidden layers of any width, we can find one that interpolates our data set, i.e. $f_{w,b}(x_i) = y_i$ for all i.

The Interpolation Problem

Sketch of proof

- W.L.O.G., we can assume that our neural net has only one hidden layer and the activation function σ is C^∞. The interpolation problem is reduced in finding $(a_i, b_i, m_i)_{i=\overline{1,d}}$ such that

$$\sum_{j=1}^{d} m_j \sigma(a_j t_i - b_j) = y_i,$$

  for any i.
- The above system of equations has solutions if and only if $\sigma(at_i - b)$ (with respect to a and b) are linearly independent.

Sketch of proof

- Suppose that our d functions are linearly dependent. This means that we can find nontrivial coefficients $(c_i)_{i=\overline{1,d}}$ such that

$$\sum_{i=1}^{d} c_i \sigma(at_i - b) = 0. \tag{1}$$

- If we differentiate k times relation (1) with respect to a, we get

$$\sum_{i=1}^{d} c_i t_i^k \sigma^{(k)}(at_i - b) = 0.$$

Sketch of proof

- Since σ is not a polynomial of degree less or equal than $d-2$, for any $k=\overline{0,d-1}$ we can find $b_k \in \mathbb{R}$ such that $\sigma^{(k)}(-b_k) \neq 0$. Taking $a=0$ and $b=b_k$ for each equation, we get a system of d equations

$$\sum_{i=1}^{d} c_i t_i^k = 0, \tag{2}$$

for each $k=\overline{0,d-1}$. Since the matrix system of (2) is a Vandermonde matrix, and the t_i are distinct, we get that all c_i must be equal to 0.

If σ is a polynomial, the interpolation problem depends very much on the x_i and the degree of σ. More precisely, we have the following result

Proposition 1

Let $(x_i, y_i)_{i=\overline{1,d}}$ be a data set with $x_i \in \mathbb{R}^p, y_i \in \mathbb{R}$, and the x_i are distinct. If σ is a polynomial of degree m, and if $d > \sum_{k=1}^{m} \binom{p+k}{k}$, then the interpolation problem is not possible

The locus of global minima

In this section we take $\mathcal{L}$ to be the mean squared loss function, i.e.

$$\mathcal{L}(\theta) = \frac{1}{d}\sum_{i=1}^{d}(f_\theta(x_i) - y_i)^2$$

Let $M = \mathcal{L}^{-1}(0)$ the locus of global minima. By Theorem 2, M is not the empty set. Moreover, if f_θ is of class C^∞, then we have the following result.

Proposition 2 ([Coo18],Theorem 2.1)

The set $M = \mathcal{L}^{-1}(0)$ is generically (that is, possibly after an arbitrarily small change to the data set) a smooth $n - d$ dimensional submanifold (possibly empty) of $\mathbb{R}^n$.

By Theorem 2 and Proposition 2 we have the following result

THEOREM 3

Let $(x_i, y_i)_{i=\overline{1,d}}$ be a data set with $x_i \in \mathbb{R}^p, y_i \in \mathbb{R}$, and the x_i are distinct. Assume that the activation function σ is C^∞ and not a polynomial of degree less than $d-2$. Let $\mathcal{L}$ be the mean squared loss function of a feedforward neural network with l hidden layers, and with the last one of width $h \geq d$. Then, the set $M = \mathcal{L}^{-1}(0)$ is generically (that is, possibly after an arbitrarily small change to the data set) a smooth $n-d$ dimensional submanifold nonempty of $\mathbb{R}^n$.

The Hessian for the global minima

In this section, we give a description of the eigenvalues of the Hessian of the loss function $\mathcal{L}$ evaluated at a point $m \in M = \mathcal{L}^{-1}(0)$. Let $f_i(\theta) := f_\theta(x_i) - y_i$. We have the following result

Proposition 3([Coo18],Proposition 2.3)

Let $M = \mathcal{L}^{-1}(0) = \bigcap M_i$, where $M_i = f_i^{-1}(0)$, be the locus of global minima of $\mathcal{L}$. If each M_i is a smooth codimension 1 submanifold of $\mathbb{R}^n$, M is nonempty, and the M_i intersect transversally at every point of M, then at every point $m \in M$, the Hessian evaluated at m has d positive eigenvalues and $n - d$ eigenvalues equal to 0.

Consider now a feedforward neural net as in Theorem 3. We also assume that the activation function σ is strictly monotone, which is equivalent to the nonvanishing of σ'. Then we have the following Corrolary of Proposition 3 :

Corrolary 1

Let $\mathcal{L}$ be the mean square loss function of a neural net as described above. Then, M is nonempty, and the Hessian of $\mathcal{L}$, evaluated at any point $m \in M = \mathcal{L}^{-1}(0)$ has d positive eigenvalues and $n - d$ eigenvalues equal to 0.

The Hessian for the global minima

Sketch of proof

- The nonemptyness of M follows from Theorem 2. Each M_i is smooth of codimension 1, again by Theorem 2 for $d = 1$.
- We assume that the intersection at m is not transversal. This means that that the tangent space $T_m M_1 = T_m M_i$ for all i. From our notations, we have that

$$f_i(w, b) = W_l \sigma(W_{l-1}\sigma(\dots\sigma(W_1 x_i - b_1)\dots) - b_{l-1}) - b_l - y_i,$$

Sketch of proof

- The nonemptyness of M follows from Theorem 2. Each M_i is smooth of codimension 1, again by Theorem 2 for $d = 1$.
- We assume that the intersection at m is not transversal. This means that that the tangent space $T_m M_1 = T_m M_i$ for all i. From our notations, we have that

$$f_i(w, b) = W_l \sigma(W_{l-1}\sigma(\ldots \sigma(W_1 x_i - b_1)\ldots) - b_{l-1}) - b_l - y_i,$$

SKETCH OF PROOF

- The equality of the tangent spaces at m is equivalent to the collinearity of the normal vectors, i.e. $\nabla f_i(w,b) = \alpha_i \nabla f_1(w,b)$ for some $\alpha_i \in \mathbb{R}$. If we compute the partial derivatives with respect to W_1, b_1, and b_l, we get

$$\frac{\partial f_i}{\partial W_1}(w,b) = -\frac{\partial f_i}{\partial b_1}(w,b) \otimes x_i$$
$$\frac{\partial f_i}{\partial b_l}(w,b) = -1$$

The Hessian for the global minima

Sketch of proof

- From the partial derivative with respect to b_l, we get that $\alpha_i = 1$ for all i. Thus,

$$\frac{\partial f_i}{\partial b_1}(w,b) = \frac{\partial f_j}{\partial b_1}(w,b)$$
$$\frac{\partial f_i}{\partial b_1}(w,b) \otimes x_i = \frac{\partial f_j}{\partial b_1}(w,b) \otimes x_j$$

- Since σ' is nonvanishing, this implies that any partial derivative of f_i with respect to all parameters of b_1 are different from 0. So from the last two relations, we get that $x_i = x_j$ for all i, j, which is a contradiction with our assumption of our data set.

- If we consider a neural network with the last hidden layer of width at most $d-1$, then do we still have the interpolation property?
- In which conditions the algorithms gradient descent and stochastic gradient descent converge to the locus of global minima?

Future research directions

- If we consider a neural network with the last hidden layer of width at most $d - 1$, then do we still have the interpolation property?
- In which conditions the algorithms gradient descent and stochastic gradient descent converge to the locus of global minima?

Bibliography I

- Sébastien Bubeck et al., *Convex optimization: Algorithms and complexity*, Foundations and Trends® in Machine Learning **8** (2015), no. 3-4, 231–357.
- Yaim Cooper, *The loss landscape of overparameterized neural networks*, arXiv preprint arXiv:1804.10200 (2018).
- Rong Ge, Furong Huang, Chi Jin, and Yang Yuan, *Escaping from saddle points—online stochastic gradient for tensor decomposition*, Conference on learning theory, PMLR, 2015, pp. 797–842.
- Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan, *How to escape saddle points efficiently*, International Conference on Machine Learning, PMLR, 2017, pp. 1724–1732.

Bibliography II

- Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht, *Gradient descent only converges to minimizers*, Conference on learning theory, PMLR, 2016, pp. 1246–1257.
- Chaoyue Liu, Libin Zhu, and Mikhail Belkin, *Toward a theory of optimization for over-parameterized systems of non-linear equations: the lessons of deep learning*, arXiv preprint arXiv:2003.00307 (2020).
- Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar, *Foundations of machine learning*, MIT press, 2018.
- Allan Pinkus, *Approximation theory of the mlp model in neural networks*, Acta numerica **8** (1999), 143–195.

THANK YOU!