Cost function of neural network is non-convex?

The cost function of neural network is $J(W, b)$, and it is claimed to be **non-convex**. I don't quite understand why it's that way, since as I see that it's quite similar to the cost function of logistic regression, right?

If it is non-convex, so the 2nd order derivative $\frac{\partial J}{\partial W} < 0$, right?

UPDATE

Thanks to the answers below as well as @gung's comment, I got your point, if there's no hidden layers at all, it's convex, just like logistic regression. But if there's hidden layers, by permuting the nodes in the hidden layers as well as the weights in subsequent connections, we could have multiple solutions of the weights resulting to the same loss.

Now more questions,

1) There're multiple local minima, and some of them should be of the same value, since they're corresponding to some nodes and weights permutations, right?

2) If the nodes and weights won't be permuted at all, then it's convex, right? And the minima will be the global minima. If so, the answer to 1) is, all those local minima will be of the same value, correct?

neural-networks loss-functions

edited May 23 '16 at 7:24

asked Jul 9 '14 at 13:59
avocado
863 5 16 37

It is non-convex in that there can be multiple local minima. – gung ♦ Feb 11 '15 at 7:00

2 Depends on the neural network. Neural networks with linear activation functions and square loss will yield convex optimization (if my memory serves me right also for radial basis function networks with fixed variances). However neural networks are mostly used with non-linear activation functions (i.e. sigmoid), hence the optimization becomes non-convex. – Cagdas Ozgenc Feb 11 '15 at 10:57

@gung, I got your point, and now I have more questions, please see my update :-) – avocado May 23 '16 at 7:25

5 At this point (2 years later), it might be better to roll your question back to the previous version, accept one of the answers below, and ask a new, follow-up question that links to this for context. – gung ♦ May 23 '16 at 11:08

1 @gung, yes you're right, but now I'm just not quite sure about some aspects of the the answer I upvoted before. Well, as I've left some new comments on the answers below, I'd wait a while to see if it's necessary to ask a new one. – avocado May 23 '16 at 12:31

5 Answers

The cost function of a neural network is in general neither convex nor concave. This means that the matrix of all second partial derivatives (the Hessian) is neither positive semidefinite, nor negative semidefinite. Since the second derivative is a matrix, it's possible that it's neither one or the other.

To make this analogous to one-variable functions, one could say that the cost function is neither shaped like the graph of x^2 nor like the graph of $-x^2$. Another example of a non-convex, non-concave function is $\sin(x)$ on $\mathbb{R}$. One of the most striking differences is that $\pm x^2$ has only one extremum, whereas $\sin$ has infinitely many maxima and minima.

How does this relate to our neural network? A cost function $J(W, b)$ has also a number of local maxima and minima, as you can see in this picture, for example.

The fact that J has multiple minima can also be interpreted in a nice way. In each layer, you use multiple nodes which are assigned different parameters to make the cost function small. Except for the values of the parameters, these nodes are the same. So you could exchange the parameters of the first node in one layer with those of the second node in the same layer, and accounting for this change in the subsequent layers. You'd end up with a different set of parameters, but the value of the cost function can't be distinguished by (basically you just moved a node, to another place, but kept all the

OK, I understand the permutation explanation you made, I think it makes sense, but now I wonder is this the authentic one to explain why neural net is non-convex? – avocado May 23 '16 at 9:13

1 What do you mean with 'authentic one'? – Roland May 23 '16 at 9:41

I mean, this is how it should be interpreted, not just an analogy. – avocado May 23 '16 at 12:27

4 @loganecolss You are correct that this is not the only reason why cost functions are non-convex, but one of the most obvious reasons. Depdending on the network and the training set, there might be other reasons why there are multiple minima. But the bottom line is: The permuation alone creates non-convexity, regardless of other effects. – Roland Feb 4 '17 at 15:57

1 Sorry, I can not understand the last paragraph. But also I missunderstand why I mentioned max(0,x) here. In any case - I think the correct way to show that there maybe multiple mode (multiple local minimum) is prove it in some way. p.s. If Hessian is indefinite it said nothing - the quasiconvex function can have indefinite Hessian but it's still unimodal. – bruziuz Jul 10 '17 at 19:19

If you permute the neurons in the hidden layer and do the same permutation on the weights of the adjacent layers then the loss doesn't change. Hence if there is a non-zero global minimum as a function of weights, then it can't be unique since the permutation of weights gives another minimum. Hence the function is not convex.

edited May 4 '17 at 9:25
Community ♦
1

answered Feb 11 '15 at 6:23
Abhinav
161 1 4

I got your point, thanks – avocado May 23 '16 at 7:28

Whether the objective function is convex or not depends on the details of the network. In the case where multiple local minima exist, you ask whether they're all equivalent. In general, the answer is no, but the chance of finding a local minimum with good generalization performance appears to increase with network size.

This paper is of interest:

> Choromanska et al. (2015). The Loss Surfaces of Multilayer Networks
>
> http://arxiv.org/pdf/1412.0233v3.pdf

From the introduction:

> - For large-size networks, most local minima are equivalent and yield similar performance on a test set.
> - The probability of finding a "bad" (high value) local minimum is non-zero for small-size networks and decreases quickly with network size.
> - Struggling to find the global minimum on the training set (as opposed to one of the many good local ones) is not useful in practice and may lead to overfitting.

They also cite some papers describing how saddle points are a bigger issue than local minima when training large networks.

answered May 23 '16 at 8:21
user20160
12.5k 1 21 46

Some answers for your updates:

1. Yes, there are in general multiple local minima. (If there was only one, it would be called the global minimum.) The local minima will not necessarily be of the same value. In general, there may be no local minima sharing the same value

introduced by some recursive structure tends to destroy convexity. Another great example of this is MA(q) models in times series analysis.

Side note: I don't really know what you mean by permuting nodes and weights. If the activation function varies across nodes, for instance, and you permute the nodes, you're essentially optimizing a different neural network. That is, while the minima of this permuted network may be the same minima, this is not the same network so you can't make a statement about the multiplicity of the same minima. For an analogy of this in the least-squares framework, you are for example swapping some rows of y and X and saying that since the minimum of $\|y - X\ \|$ is the same as before that there are as many minimizers as there are permutations.

answered May 23 '16 at 7:59
Mustafa S Eisa
909 5 16

1 "one-layer network" would be just what "softmax" or logistic regression looks like, right? – avocado May 23 '16 at 8:25

By "permuting nodes and weights", I mean "swapping", and that's what I got from the above 2 old answers, and as I understood their answers, by "swapping" nodes and weights in **hidden layers**, we might end up having the same output in theory, and that's why we might have multiple minima. You mean this explanation is not correct? – avocado May 23 '16 at 8:28

You have the right idea, but its not quite the same. For networks, the loss may not necessarily be binomial loss, the activation functions may not necessarily be sigmoids, etc. – Mustafa S Eisa May 23 '16 at 8:33

Yes, I don't think it's correct. Even though it's true that you'll get the same performance whether you permute these terms or not, this doesn't define the convexity or non-convexity of any problem. The optimization problem is convex if, for a fixed loss function (not any permutation of the terms in the loss), the objective function is convex in the model parameters and the feasible region upon which you are optimizing is convex and closed. – Mustafa S Eisa May 23 '16 at 8:34

I see, so if it's "one-layer", it might not be "softmax". – avocado May 23 '16 at 8:34

You will have one global minimum if problem is convex or quasiconvex.

About convex "building blocks" during building neural networks (Computer Science version)

I think there are several of them which can be mentioned:

1. max(0,x) - convex and increasing
2. log-sum-exp - convex and increasing in each parameter
3. y = Ax is affine and so convex in (A), maybe increasing maybe decreasing. y = Ax is affine and so convex in (x), maybe increasing maybe decreasing.

Unfortunately it is not convex in (A, x) because it looks like indefinite quadratic form.

4. Usual math discrete convolution (by "usual" I mean defined with repeating signal) Y=h*X Looks that it is affine function of h or of variable X. So it's a convex in variable h or in variable X. About both variables - I don't think so because when h and X are scalars convolution will reduce to indefinite quadratic form.
5. max(f,g) - if f and g are convex then max(f,g) is also convex.

If you substitute one function into another and create compositions then to still in the convex room for y=h(g(x),q(x)), but h should be convex and should increase (non-decrease) in each argument....

Why neural netwoks in non-convex:

1. I think the convolution Y=h*X is not nessesary increasing in h. So if you not use any extra assumptions about kernel you will go out from convex optimization immediatly after you apply convolution. **So there is no all fine with composition**.
2. Also convolution and matrix multiplication is not convex if consider couple parameters as mentioned above. **So there is evean a problems with matrix multiplication: it is non-convex operation in parameters (A,x)**
3. y = Ax can be quasiconvex in (A,x) but also extra assumptions should be taken into account.

Please let me know if you disagree or have any extra consideration. The question is also very interesting to me.

p.s. max-pooling - which is downsamping with selecting max looks like some modification of elementwise max operations with affine precomposition (to pull need blocks) and it looks convex for me.

About other questions

2. If there are not only one global minimum. Nothing can be said about relation between local minimums. Or at least you can not use convex optimization and it's extensions for it, because this area of math is deeply based on global underestimator.

Maybe you have confusion about this. Because really people who create such schemas just do "something" and they receive "something". Unfortunately because we don't have perfect mechanism for tackle with non-convex optimization (in general).

But there are even more simple things beside Neural Networks - which can not be solved like non-linear least squares -- https://youtu.be/l1X4tOoIHYo?t=2992 (EE263, L8, 50:10)

edited Aug 18 '17 at 19:40

answered Jul 10 '17 at 1:19

bruziuz
144 7