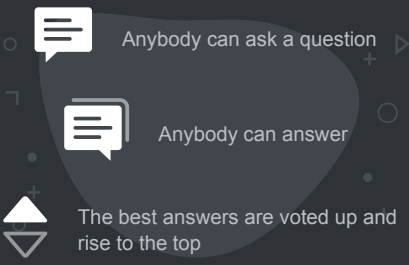



Search, Home, Trophies, Help, Notifications, Log in, Sign up

Cross Validated is a question and answer site for people interested in statistics, machine learning, data analysis, data mining, and data visualization. It only takes a minute to sign up.

Sign up to join this community



 Cross Validated

Ask Question

A list of cost functions used in neural networks, alongside applications

Asked 4 years, 11 months ago Active 1 year, 7 months ago Viewed 154k times

What are common cost functions used in evaluating the performance of neural networks?

148 Details

(feel free to skip the rest of this question, my intent here is simply to provide clarification on notation that answers may use to help them be more understandable to the general reader)

204

I think it would be useful to have a list of common cost functions, alongside a few ways that they have been used in practice. So if others are interested in this I think a community wiki is probably the best approach, or we can take it down if it's off topic.

Notation

So to start, I'd like to define a notation that we all use when describing these, so the answers fit well with each other.

This notation is from [Neilsen's book](#).

A Feedforward Neural Network is a many layers of neurons connected together. Then it takes in an input, that input "trickles" through the network and then the neural network returns an output vector.

More formally, call a_j^i the activation (aka output) of the j^{th} neuron in the i^{th} layer, where a_j^1 is the j^{th}

$$a_j^i = \sigma(\sum_k (w_{jk}^i \cdot a_k^{i-1}) + b_j^i)$$

where

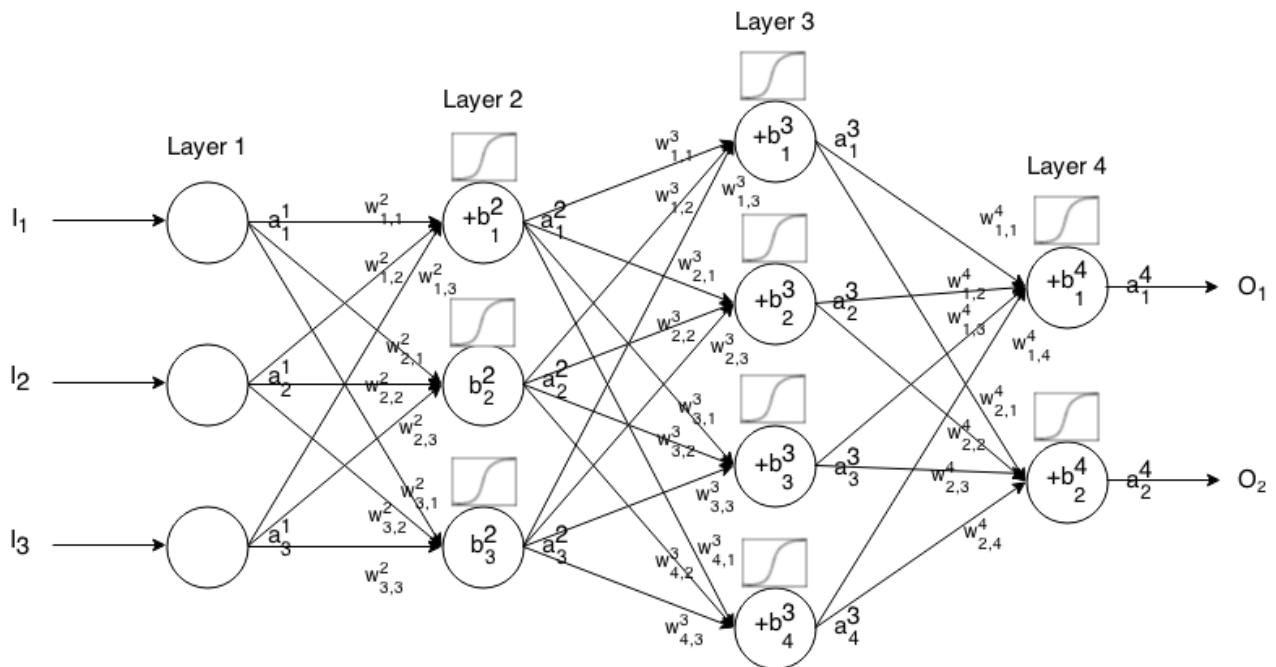
σ is the activation function,

w_{jk}^i is the weight from the k^{th} neuron in the $(i-1)^{th}$ layer to the j^{th} neuron in the i^{th} layer,

b_j^i is the bias of the j^{th} neuron in the i^{th} layer, and

a_j^i represents the activation value of the j^{th} neuron in the i^{th} layer.

Sometimes we write z_j^i to represent $\sum_k (w_{jk}^i \cdot a_k^{i-1}) + b_j^i$, in other words, the activation value of a neuron before applying the activation function.



For more concise notation we can write

$$a^i = \sigma(w^i \times a^{i-1} + b^i)$$

To use this formula to compute the output of a feedforward network for some input $I \in \mathbb{R}^n$, set $a^1 = I$, then compute a^2, a^3, \dots, a^m , where m is the number of layers.

Introduction

A cost function is a measure of "how good" a neural network did with respect to it's given training sample

[Log in](#)[Sign up](#)

whole.

Specifically, a cost function is of the form

$$C(W, B, S^r, E^r)$$

where W is our neural network's weights, B is our neural network's biases, S^r is the input of a single training sample, and E^r is the desired output of that training sample. Note this function can also potentially be dependent on y_j^i and z_j^i for any neuron j in layer i , because those values are dependent on W , B , and S^r .

In backpropagation, the cost function is used to compute the error of our output layer, δ^L , via

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} \sigma'(z_j^i)$$

.

Which can also be written as a vector via

$$\delta^L = \nabla_a C \odot \sigma'(z^i)$$

.

We will provide the gradient of the cost functions in terms of the second equation, but if one wants to prove these results themselves, using the first equation is recommended because it's easier to work with.

Cost function requirements

To be used in backpropagation, a cost function must satisfy two properties:

1: *The cost function C must be able to be written as an average*

$$C = \frac{1}{n} \sum_x C_x$$

over cost functions C_x for individual training examples, x .

This is so it allows us to compute the gradient (with respect to weights and biases) for a single training example, and run Gradient Descent.

2: *The cost function C must not be dependent on any activation values of a neural network besides the output values a^L .*

Technically a cost function can be dependent on any a_j^i or z_j^i . We just make this restriction so we can backpropagate, because the equation for finding the gradient of the last layer is the only one that is dependent on the cost function (the rest are dependent on the next layer). If the cost function is dependent on other activation layers besides the output one, backpropagation will be invalid because the idea of "trickling backwards" no longer works.

Also, activation functions are required to have an output $0 \leq a_j^L \leq 1$ for all j . Thus these cost functions need to only be defined within that range (for example, $\sqrt{a_j^L}$ is valid since we are guaranteed $a_j^L \geq 0$).

[Log in](#)[Sign up](#)[share](#) [cite](#) [improve this question](#) [follow](#)

edited May 31 '15 at 19:57

community wiki

Phylliida

- 3 This is a Q&A site, and the format of this post doesn't really fit that. You should probably put the majority of the content in an answer, and leave just the question (e.g. What is a list of cost functions used in NNs?). – Roger Fan May 31 '15 at 19:47

Okay, is that better? I think the definitions are important otherwise the answers become vague for those that aren't familiar with the terminology the writer uses. – Phylliida May 31 '15 at 19:49

But what if a different answer uses different notation or terminology? – Roger Fan May 31 '15 at 19:52

- 3 The idea is that everyone uses the same terminology here, and that if it's different we convert it to this, so the answers "fit" with each other. But I suppose I could remove that piece if you don't think it's helpful. – Phylliida May 31 '15 at 19:54
- 1 I just think the detail the question goes into isn't really necessary or relevant. It seems a bit excessive and limiting, but that's just me. – Roger Fan May 31 '15 at 19:56

[add a comment](#)

2 Answers

Active	Oldest	Votes
--------	--------	-------



Here are those I understand so far. Most of these work best when given values between 0 and 1.

92

Quadratic cost



Also known as *mean squared error*, *maximum likelihood*, and *sum squared error*, this is defined as:



$$C_{MST}(W, B, S^r, E^r) = 0.5 \sum_j (a_j^L - E_j^r)^2$$

The gradient of this cost function with respect to the output of a neural network and some sample r is:

$$\nabla_a C_{MST} = (a^L - E^r)$$

Cross-entropy cost

Also known as *Bernoulli negative log-likelihood* and *Binary Cross-Entropy*

$$C_{CE}(W, B, S^r, E^r) = - \sum_j [E_j^r \ln a_j^L + (1 - E_j^r) \ln (1 - a_j^L)]$$

The gradient of this cost function with respect to the output of a neural network and some sample r is:

$$\nabla_a C_{CE} = \frac{(a^L - E^r)}{(1 - a^L)(a^L)}$$

Exponential cost

This requires choosing some parameter τ that you think will give you the behavior you want. Typically you'll just need to play with this until things work good.

[Log in](#)[Sign up](#)

where $\exp(x)$ is simply shorthand for e^x .

The gradient of this cost function with respect to the output of a neural network and some sample r is:

$$\nabla_a C = \frac{2}{\tau} (a^L - E^r) C_{EXP}(W, B, S^r, E^r)$$

I could rewrite out C_{EXP} , but that seems redundant. Point is the gradient computes a vector and then multiplies it by C_{EXP} .

Hellinger distance

$$C_{HD}(W, B, S^r, E^r) = \frac{1}{\sqrt{2}} \sum_j (\sqrt{a_j^L} - \sqrt{E_j^r})^2$$

You can find more about this [here](#). This needs to have positive values, and ideally values between 0 and 1. The same is true for the following divergences.

The gradient of this cost function with respect to the output of a neural network and some sample r is:

$$\nabla_a C = \frac{\sqrt{a^L} - \sqrt{E^r}}{\sqrt{2}\sqrt{a^L}}$$

Kullback–Leibler divergence

Also known as *Information Divergence*, *Information Gain*, *Relative entropy*, *KLIC*, or *KL Divergence* (See [here](#)).

Kullback–Leibler divergence is typically denoted

$$D_{KL}(P \parallel Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)}$$

,

where $D_{KL}(P \parallel Q)$ is a measure of the information lost when Q is used to approximate P . Thus we want to set $P = E^i$ and $Q = a^L$, because we want to measure how much information is lost when we use a_j^i to approximate E_j^i . This gives us

$$C_{KL}(W, B, S^r, E^r) = \sum_j E_j^r \log \frac{E_j^r}{a_j^L}$$

The other divergences here use this same idea of setting $P = E^i$ and $Q = a^L$.

The gradient of this cost function with respect to the output of a neural network and some sample r is:

$$\nabla_a C = - \frac{E^r}{a^L}$$

[Log in](#)[Sign up](#)

From [here](#).

$$C_{GKL}(W, B, S^r, E^r) = \sum_j E_j^r \log \frac{E_j^r}{a_j^L} - \sum_j (E_j^r) + \sum_j (a_j^L)$$

The gradient of this cost function with respect to the output of a neural network and some sample r is:

$$\nabla_a C = \frac{a^L - E^r}{a^L}$$

Itakura–Saito distance

Also from [here](#).

$$C_{GKL}(W, B, S^r, E^r) = \sum_j \left(\frac{E_j^r}{a_j^L} - \log \frac{E_j^r}{a_j^L} - 1 \right)$$

The gradient of this cost function with respect to the output of a neural network and some sample r is:

$$\nabla_a C = \frac{a^L - E^r}{(a^L)^2}$$

Where $\left((a^L)^2\right)_j = a_j^L \cdot a_j^L$. In other words, $(a^L)^2$ is simply equal to squaring each element of a^L .

[share](#) [cite](#) [improve this answer](#) [follow](#)

edited Aug 27 '18 at 0:28

community wiki

5 revs

Phylliida

Thanks for sharing, you can also consider these: github.com/torch/nn/blob/master/doc/criterion.md – Yannis Assael May 31 '15 at 19:52

2 you have a small mistake in the denominator of the cross-entropy derivative, it should be $a * (1-a)$ not $a * (1+a)$ – Amro May 16 '16 at 17:51

1 It would also be cool to show the pinball loss function to minimize error quantiles rather than average error. Very used in decision support systems. – Ricardo Cruz Nov 10 '16 at 13:56

where can I see graphs for these? – coiso Aug 24 '18 at 15:46

1 With regards to the Quadratic Cost Function, you should note that "mean squared error" \neq "maximum likelihood" \neq "sum squared error". Authors may use the name (incorrectly) interchangeably, but they are *not* the same thing. – Jon Jan 2 '19 at 19:26

[show 2 more comments](#)

Don't have the reputation to comment, but there are sign errors in those last 3 gradients.

In the KL divergence,

[Log in](#)[Sign up](#)

$$= \sum_j E_j \log(E_j) - E_j \log(a_j)$$

$$dC = - \sum_j E_j d\log(a_j)$$

$$= - \sum_j (E_j/a_j) da_j$$

$$\nabla_a C = \frac{-E}{a}$$

This same sign error appears in the Generalized KL divergence.

In the Itakura-Saito distance ,

$$C = \sum_j (E_j/a_j) - \log(E_j/a_j) - 1$$

$$= \sum_j (E_j/a_j) - \log(E_j) + \log(a_j) - 1$$

$$dC = \sum_j (-E_j/a_j^2) da_j + d\log(a_j)$$

$$= \sum_j (1/a_j) da_j - (E_j/a_j^2) da_j$$

$$= \sum_j (a_j - E_j)/a_j^2 da_j$$

$$\nabla_a C = \frac{a - E}{(a)^2}$$

[share](#) [cite](#) [improve this answer](#) [follow](#)

answered **Nov 24 '16 at 18:31**

[community wiki](#)
[frank](#)

[add a comment](#)



Highly active question. Earn 10 reputation in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.

Not the answer you're looking for? Browse other questions tagged [your own question](#).

[machine-learning](#)

[neural-networks](#)

or [ask](#)

[Log in](#)[Sign up](#)

✍ A message from our CEO: The way Forward

✍ The Overflow #20: Sharpen your skills

Featured on Meta

🗨 Introducing the Moderator Council - and its first, pro-tempore, representatives

🗨 CEO Blog: The Way Forward

Linked

- 1 What are the applications of different cost functions and which one to choose?
- 19 Backpropagation algorithm
- 11 What is the essential difference between a neural network and nonlinear regression?
- 1 Back propagation and and various cost functions
- 1 Requirements of a loss function for an NN

Related

- 8 Exact definition of Maxout
- 1 Vanishing gradient in basic 3-layer neural networks?
- 3 Matrix-Based Approach to Mini-batch SGD for Feedforward Neural Networks
- 2 activation functions of elman recurrent neural network
- 1 Heterogenous layers in Neural Network
- 0 Calculating Softmax derivative independent of cost function
- 7 Neural network softmax activation
- 3 Do neural networks compute functions or compute parameters of a defined function
- 1 Can all neural network cost functions be written as an average of individual cost and as a function of the activations at the output?
- 0 "Forward Propagation" in Neural Networks

Hot Network Questions

[Log in](#)[Sign up](#)

Can a Glamour bard cast a level 1+ spell and cast the Command spell using the Mantle of Majesty feature in the same turn?



Is there any website that allows you to check if a position has been played before?



Why didn't the US replenish its strategic stockpile of face masks after the 2009 pandemic?



70's/80's children's book with a magic paint brush



Paper rejected due to short reference list



Implement the Zundoko Kiyoshi function



What is polymorphism if you can already have methods that are the same defined in different types?



Inch-long block-and-spring part came with drill - what is it?



Reading a Swiss newspaper as a German learner



Why do you have to blow sideways on a flute?



Was it possible for a young Japanese woman to end up enslaved in Great Britain in the mid-1600s?



The connection of the "R" in the ideal gas law and osmotic pressure



I have been practicing a song for 3 hours straight but I keep making mistakes. How do get this right?



How to address a student systematically boosting their grades by filing frivolous complaints?



Unusual applications of regular expressions?



Win 270 electoral college votes



Tandem Breaker is in half-off position. Unable to reset



Is it possible to see Callisto with the naked eye when it's at its greatest elongation from Jupiter?



Are humanities subjects useful for Computer Science jobs?



What's the difference between 鍵 and ？



How to secure a solo-authored paper as a PhD student, while working with a supervisor who has reputation of forcing coauthorship?



Best practice for layout of balanced audio grounds



Would shotguns be a practical weapon to arm certain members of an anti-zombie military unit?



Question feed

CROSS VALIDATED

COMPANY

STACK EXCHANGE
NETWORK

[Blog](#) [Facebook](#) [Twitter](#) [LinkedIn](#) [Instagram](#)

[Tour](#)

[Stack Overflow](#)

[Technology](#) >

[Help](#)

[Stack Overflow Business](#)



Log in

Sign up

[Feedback](#)

[Press](#)

[Science](#) >

[Mobile](#)

[Legal](#)

[Other](#) >

[Disable Responsiveness](#)

[Privacy Policy](#)

site design / logo © 2020 Stack Exchange Inc;
user contributions licensed under [cc by-sa 4.0](#)
with [attribution required](#). rev 2020.5.8.36785