

MC.AI

[HOME](#)[CONTACT](#)[RECENT POSTS](#)

A Dummies Guide to Gradient Descent and Backpropagation

🕒 10. JULY 2018

A lot of data scientists use the term Gradient Descent and Backpropagation interchangeably, but contrary to popular opinion they are not the same thing.

What is Gradient Descent?

Gradient Descent is an optimization technique in the machine learning process which minimizes the cost function. Every machine learning algorithm has a cost function.

What is the Cost Function and why do I need to minimize it?

Whenever we use machine learning algorithms we are using the training data to make predictions which we later map on the test data. The cost function is used to inform us how close are the real values from the training data to the predicted values in the test data. The closer the values, the lesser the accuracy error and the better our algorithm's prediction prowess.

Gradient Descent is essentially an optimization algorithm we use in machine learning which find the minimum point in the cost function (local minima or global minima). Think of gradient descent as a detective which is trying to find where the minimum point of the cost function. The minimum point of the cost function would mean the least amount of error.

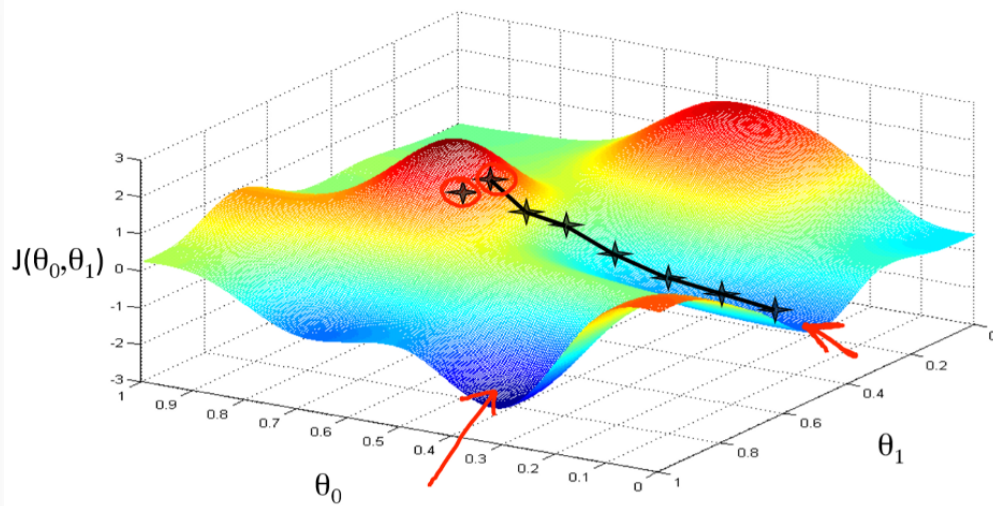
ABOUT

MC.AI –
Aggregated news
about artificial
intelligence

MC.AI collects interesting articles and news about artificial intelligence and related areas. The contributions come from various open sources and are presented here in a collected form. The copyrights are held by the original authors, the source is indicated with each contribution.

Contributions which should be deleted from this platform can be reported using the appropriate form (within the contribution).





MC.AI is open for direct submissions, we look forward to your contribution!

SEARCH ON MC.AI

Cost function mapped looks like a hill

If we map all the values of the cost function, we will see something like a 3D model akin to an undulating plane or mountain. Gradient descent is the method which is trying to find the lowest point in the 3D model. Again, the lowest point would mean for us the least error for our model. You can also picture gradient descent as a hiker trying to find the plateau on a hill. If the hiker finds the plateau he can't go any lower. Similarly, if the gradient descent algorithm find the minimum point, it will stop and wont go any further. We can say the model has converged.



Repeat until convergence

{

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

}

The elegant gradient descent formula

The alpha (learning rate) in the formula is how fast the algorithm converges. basically, we can control how slow or fast the hiker can go down the hill. If the learning rate is too small the hiker will very very slow and wont be able to reach the plateau. Similarly, if the learning rate is too large the hiker may walk past the plateau! The method by which the gradient descent algorithm works is by calculating derivatives which measures the rate of change of a slope.

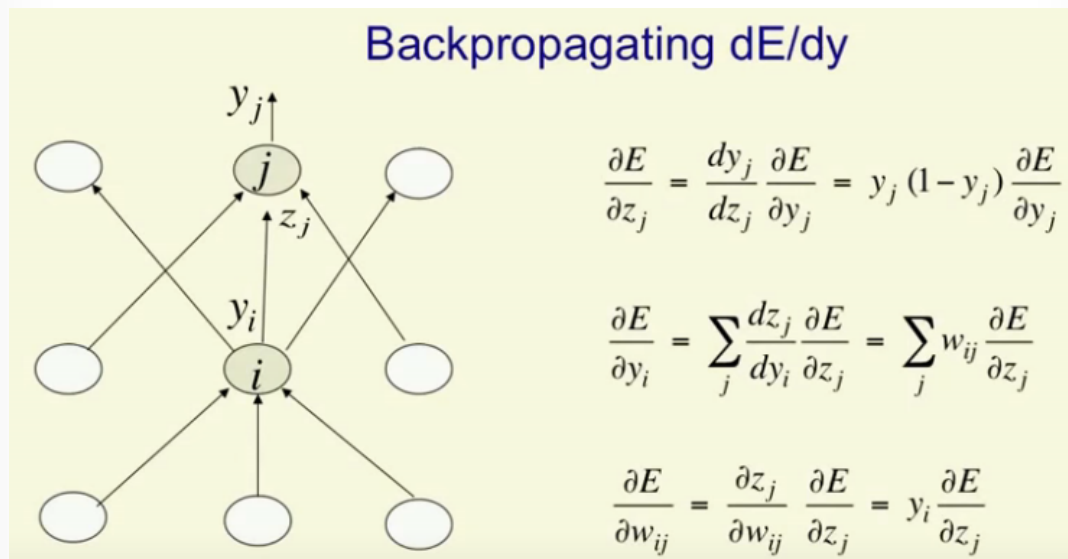
What is Backpropagation?

In simple English, backpropagation is the method of computing the gradient of a cost function in deep neural nets. Unlike the gradient descent algorithm, backpropagation algorithm does not have a learning rate. Backpropagation instead finds the partial derivatives of the cost function.

In neural nets we have input layers for our input features, hidden layers for transformation and output layers for our outputs. Every neuron in the neural net has a weight and we change the weight to improve our performance. If by changing the weight we can improve the performance of the neural net we save the weight. To change one weight the neural net has to perform



forward passes depending on the number of units, this is a slow and inefficient process. This is why we use backpropagation. Backpropagation allows us to compute the error derivatives of all hidden units at the same time.



This is how backpropagation works

From the picture we can see that backpropagation works like forward pass but moves backwards from the hidden layer (i) to the input layer (j) and combines the error derivatives of all hidden layers at once.

Conclusion

In a nutshell, the backpropagation algorithm computes how the error changes as we change the weights in the connections while the gradient descent algorithm optimizes the error of the cost function. In deep neural nets we use gradient descent and backpropagation in tandem.

More Dummies Guides:

[A Dummies Guide to Machine Learning](#)

[A Dummies Guide to Neural Nets](#)



[A Dummies Guide to the Math of AI](#)

[A Dummies Guide to Python](#)

[A Dummies Guide to Data Normalization](#)

Source: [Deep Learning on Medium](#)

Related Articles

**Paper repro:
Deep
Metalearning
using "MAML"
and "Reptile"**

2 APRIL 2019

**NLP & fastai |
Language
Model**

10. SEPTEMBER 2019

**How to Convert
Your Keras
Model to ONNX**

27. NOVEMBER 2019

**Artificial
Intelligence
Wrote This
Fake Metallica
Song –
MetalSucks**

← DAY 1 : Deep Learning Vs Machine Learning

Disassembling Recurrent Neural Networks →

REQUEST FOR DELETION

Fields marked with an * are required

The blog posts on this website are all collected from different sources (via feeds).

If you are an author of a post and would like to have it deleted from this page, you can request it using this form.

Or if you consider a contribution to be SPAM or inappropriate, please let me know!

Requests will be processed as soon as possible.

Name *

Email *

URL of post to be deleted Reason for the request *

*



https://mc.ai/a-dummies-guide-to-gradient-descent-and-backpropagation/

please state the reason for your request

☐

**I understand, that this form is for "requests for deletion" only!
Other comments will not be handled. ***

Anti-Spam: How much is $5+2$? *

Type your answer here

SUBMIT



mc.ai aggregates articles from different sources - copyright remains at original authors

