Keep it simple! How to understand Gradient Descent algorithm

2017 Silver KDnuggets Blog

Previous post
Next post

Tags: Algorithms, Gradient Descent

In Data Science, Gradient Descent is one of the important and difficult concepts. Here we explain this concept with an example, in a very simple way. Check this out.

---

comments

By Jahnavi Mahanta.

Source: dilbert.com

When I first started out learning about machine learning algorithms, it turned out to be quite a task to gain an intuition of what the algorithms are doing. Not just because it was difficult to understand all the mathematical theory and notations, but it was also plain boring. When I turned to online tutorials for answers, I could again only see equations or high level explanations without going through the detail in a majority of the cases.

It was then that one of my data science colleagues introduced me to the concept of working out an algorithm in an excel sheet. And that worked wonders for me. Any new algorithm, I try to learn it in an excel at a small scale and believe me, it does wonders to enhance your understanding and helps you fully appreciate the beauty of the algorithm.

Let me explain to you using an example.

Most of the data science algorithms are optimization problems and one of the most used algorithms to do the same is the Gradient Descent Algorithm.

Now, for a starter, the name itself Gradient Descent Algorithm may sound intimidating, well, hopefully after going though this post,that might change.

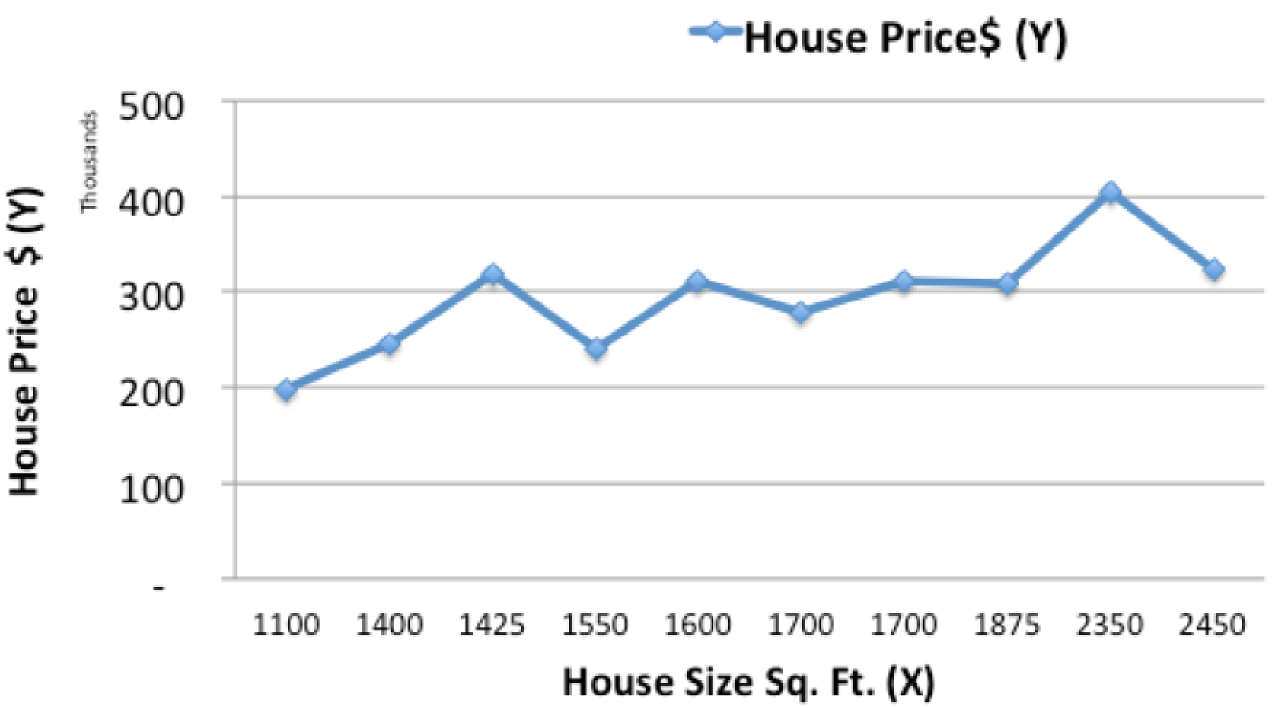
Lets take the example of predicting the price of a new price from housing data:

Now, given **historical housing data,**the task is to create a model that predicts the price of a new house given the house size.

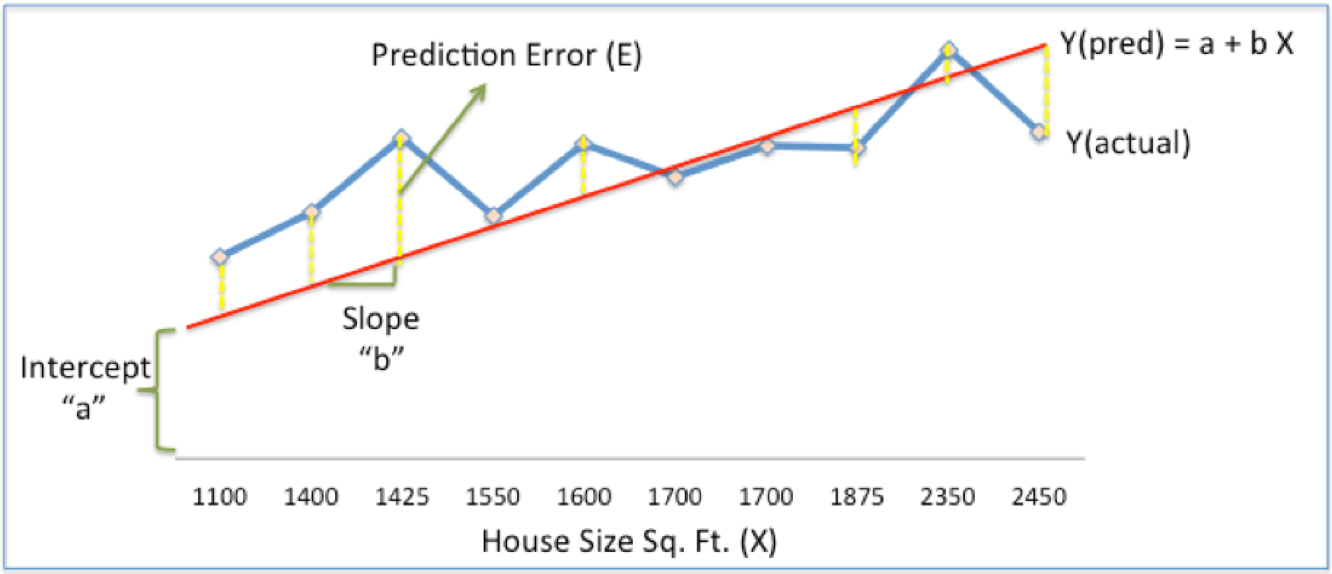
House Size sq.ft (X)	1400	1600	1700	1875	1100	1550	2350	2450	1425	1700
House Price$ (Y)	245,000	312,000	279,000	308,000	199,000	219,000	405,000	324,000	319,000	255,000

The task – for a new house, given its size (X), what will its price (Y) be?

Lets start off by plotting the historical housing data:



Now, we will use a simple linear model, where we fit a line on the historical data, to predict the price of a new house (Ypred) given its size (X)



In the above chart, the red line gives the predicted house price (Ypred) given house size (X).

Ypred = a+bX

The blue line gives the actual house prices from historical data (Yactual)

The difference between Yactual and Ypred (given by the yellow dashed lines) is the prediction error (E)

So, we need to find a line with optimal values of a,b (called weights) that best fits the historical data by reducing the prediction error and improving prediction accuracy.

So, our objective is to find optimal **a, b** that minimizes the error between actual and predicted values of house price (1/2 is for mathematical convenience since it helps in calculating gradients in calculus)

Sum of Squared Errors (SSE) = ½ Sum (Actual House Price – Predicted House Price)^2

= **½ Sum(Y – Ypred)^2**

(Please note that there are other measures of Error. SSE is just one of them.)

This is where Gradient Descent comes into the picture. Gradient descent is an optimization algorithm that finds the optimal weights (a,b) that reduces prediction error.

Lets now go step by step to understand the **Gradient Descent algorithm:**

Step 1: Initialize the weights(a & b) with random values and calculate Error (SSE)

Step 2: Calculate the gradient i.e. change in SSE when the weights (a & b) are changed by a very small value from their original randomly initialized value. This helps us move the values of a & b in the direction in which SSE is minimized.

Step 3: Adjust the weights with the gradients to reach the optimal values where SSE is minimized

Step 4: Use the new weights for prediction and to calculate the new SSE

Step 5: Repeat steps 2 and 3 till further adjustments to weights doesn't significantly reduce the Error

We will now go through each of the steps in detail (I worked out the steps in excel, which I have pasted below). But before that, we have to standardize the data as it makes the optimization process faster.

HOUSING DATA	
House Size (X)	House Price (Y)
1,100	1,99,000
1,400	2,45,000
1,425	3,19,000
1,550	2,40,000
1,600	3,12,000
1,700	2,79,000
1,700	3,10,000
1,875	3,08,000
2,350	4,05,000
2,450	3,24,000

Min-Max Standardization	
X (X-Min/Max-min)	Y (Y-Min/Max-Min)
0.00	0.00
0.22	0.22
0.24	0.58
0.33	0.20
0.37	0.55
0.44	0.39
0.44	0.54
0.57	0.53
0.93	1.00
1.00	0.61

Step 1: To fit a line Ypred = a + b X, start off with random values of a and b and calculate prediction error (SSE)

a	b	X	Y	YP=a+bX	SSE=1/2(Y-YP)^2
0.45	0.75	0.00	0.00	0.45	0.101
		0.22	0.22	0.62	0.077
		0.24	0.58	0.63	0.001
		0.33	0.20	0.70	0.125
		0.37	0.55	0.73	0.016
		0.44	0.39	0.78	0.078
		0.44	0.54	0.78	0.030
		0.57	0.53	0.88	0.062
		0.93	1.00	1.14	0.010
		1.00	0.61	1.20	0.176
				Total SSE	**0.677**

Step 2: Calculate the error gradient w.r.t the weights

∂SSE/∂a = – (Y-YP)

∂SSE/∂b = – (Y-YP)X

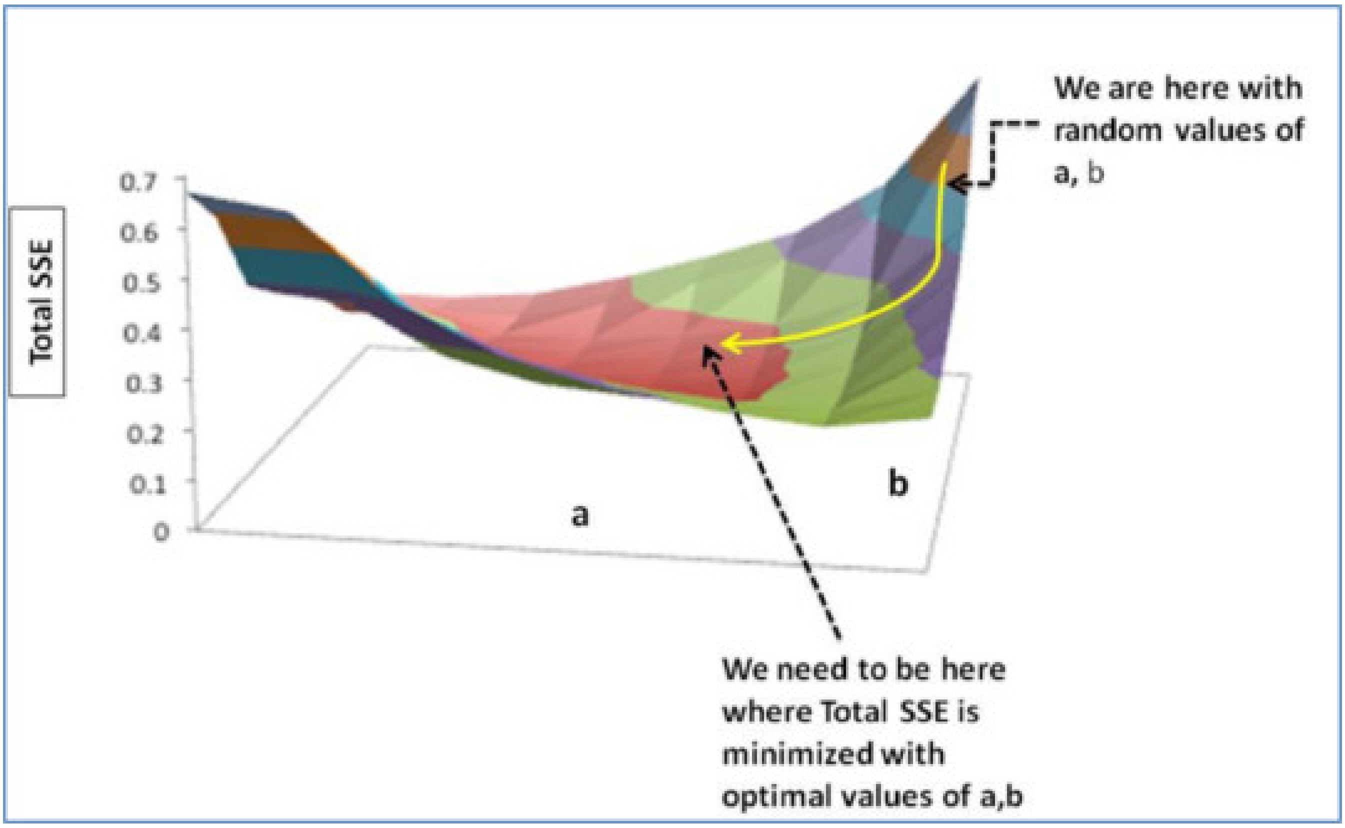
Here, SSE=½ $(Y\text{-}YP)^2$ = ½$(Y\text{-}(a+bX))^2$

You need to know a bit of calculus, but that's about it!!

∂SSE/∂a and ∂SSE/∂b are the **gradients** and they give the direction of the movement of a,b w.r.t to SSE.

a	b	X	Y	YP=a+bX		SSE		∂SSE/∂a = -(Y-YP)	∂SSE/∂b = -(Y-YP)X
0.45	0.75	0.00	0.00	0.45		0.101		0.45	0.00
		0.22	0.22	0.62		0.077		0.39	0.09
		0.24	0.58	0.63		0.001		0.05	0.01
		0.33	0.20	0.70		0.125		0.50	0.17
		0.37	0.55	0.73		0.016		0.18	0.07
		0.44	0.39	0.78		0.078		0.39	0.18
		0.44	0.54	0.78		0.030		0.24	0.11
		0.57	0.53	0.88		0.062		0.35	0.20
		0.93	1.00	1.14		0.010		0.14	0.13
		1.00	0.61	1.20		0.176		0.59	0.59
					Total SSE	0.677	Sum	3.300	1.545

Step 3: Adjust the weights with the gradients to reach the optimal values where SSE is minimized



We need to update the random values of a,b so that we move in the direction of optimal a, b.

Update rules:

- a – ∂SSE/∂a
- b – ∂SSE/∂b

So, update rules:

1. New a = a – r * **∂SSE/∂a =** 0.45-0.01*3.300 = 0.42
2. New b = b – r * **∂SSE/∂b=** 0.75-0.01*1.545 = 0.73

here, r is the learning rate = 0.01, which is the pace of adjustment to the weights.

Step 4: Use new a and b for prediction and to calculate new Total SSE

a	b	X	Y	YP=a+bX		SSE		∂SSE/∂a	∂SSE/∂b
0.42	0.73	0.00	0.00	0.42		0.087		0.42	0.00
		0.22	0.22	0.58		0.064		0.36	0.08
		0.24	0.58	0.59		0.000		0.01	0.00
		0.33	0.20	0.66		0.107		0.46	0.15
		0.37	0.55	0.69		0.010		0.14	0.05
		0.44	0.39	0.74		0.063		0.36	0.16
		0.44	0.54	0.74		0.021		0.20	0.09
		0.57	0.53	0.84		0.048		0.31	0.18
		0.93	1.00	1.10		0.005		0.10	0.09
		1.00	0.61	1.15		0.148		0.54	0.54
					Total SSE	0.553	Sum	2.900	1.350

You can see with the new prediction, the total SSE has gone down (0.677 to 0.553). That means prediction accuracy has improved.

Step 5: Repeat step 3 and 4 till the time further adjustments to a, b doesn't significantly reduces the error. At that time, we have arrived at the optimal a,b with the highest prediction accuracy.

This is the Gradient Descent Algorithm. This optimization algorithm and its variants form the core of many machine learning algorithms like Neural Networks and even Deep Learning.

Disclaimers:

Please note that this post is primarily for tutorial purposes, hence:

1. The data used is fictitious and data size is extremely small. Also to simplify the example, the data and model, is a one variable example.
2. This post is primarily meant to highlight how we can simplify our understanding of the math behind algorithms like Gradient descent by working them out in excel, hence there is no claim here that gradient descent gives better /worse results as compared to least square regression.
3. Since the data is very small, for tutorial purposes, the entire data is being used for training. However, while building actual predictive models, various data validation techniques are leveraged (Example – Train / Test split or N- Cross Validation)

Liked what you read? To learn other algorithms in a similar simplified manner, register for the 8 week Data science course on www.deeplearningtrack.com.

Bio: Jahnavi is a machine learning and deep learning enthusiast, having led multiple machine learning teams in American Express over the last 13 years. She is the co-founder of Deeplearningtrack, an online instructor led data science training platform— **www.deeplearningtrack.com**

Related:

- Learning to Learn by Gradient Descent by Gradient Descent
- Getting Up Close and Personal with Algorithms
- A Concise Overview of Standard Model-fitting Methods

28 Comments **KDnuggets** 1 **Login**

Recommend 23 Tweet Share **Sort by Best**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name


HwangC • a year ago

Thank you for the awesome article about the Gradient Descent Algorithm.

34 ^ | v • Reply • Share ›


Munish Kohli • a year ago

Jahanvi, I went through Stanford Machine learning course by Andrew Ng, but had a hard time understanding. Your this article cleared my thoughts and encouraged me to do follow your steps in an Excel. Although even after executing steps on sample data its not producing results I am expecting, but I feel I am making a progress. Can you post similar practical article on stochastic gradient? That will be helpful

2 ^ | v • Reply • Share ›

2 ^ | v • Reply • Share ›

Ash ➜ Munish Kohli • a year ago

yea...it was indeed a great explaination... even i went through Andrew course.... one thing i didnt understand...if there are more than one features in X .... how will you calculate min-max standardisation? the way jahanvi has done it... coz in Andrew course ...he has used some 1/2m summation of(h theta (x) -y)^2

1 ^ | v • Reply • Share ›

JerryK ➜ Ash • a month ago

The goal is go the correct way to reduce the error. So you can add a constant in front and not change the this fact. The reason 1/2 is selected as this constant is because when take the derivative of the squared term in the parenthesis it becomes 2/2 * derivative of (j theta(x)-y). 2/2 is 1 so answer so the term and the associated multiplication operation can be dropped. This make the operation faster. This may seem trivial, but when you might do the 10 million times (or more) for a 10 Mpixel image even tiny the reduction in computation becomes significant.

^ | v • Reply • Share ›

Lakshminadh Javvadi • 2 years ago

excellent jahnavi. Thanks for simplifying the concept.

1 ^ | v • Reply • Share ›

vinay kumar • 2 years ago • edited

A video tutorial on Gradient Descent for beginners:
https://quickkt.com/tutoria...

1 ^ | v • Reply • Share ›

Дмитрий Ватлин • a year ago • edited

Thank you for the explanation! Just before Step 1 there is a Housing Data table. We see that 1550 sq.m. house costs 240 000 USD but at the beginning it costs 219 000 USD. Mistake?

1 ^ | v • Reply • Share ›

Vitor Mizumoto • 2 months ago

Thank you so much for having invested your time in this article. I was struggling to understand these concepts

^ | v • Reply • Share ›

fatma • 3 months ago

Great Job

^ | v • Reply • Share ›

jeffross • 7 months ago

or does the parameter assumes the value theta * standardizedfactor where standardizedfactor is standardizedx value times (xmax-xmin) over all training examples?

^ | v • Reply • Share ›

jeffross • 7 months ago

pretty neat work this. However two questions. 1) the parameters computed here would work for the standardized training set above but might not work if used with the non standardized training set 2) Also if a value is to be predicted given a house size((which is either less than min above or greater than max) then then obviously we would have to standardize again, which means the above standardized values would be invalidated?

^ | v • Reply • Share ›

Kiran Prasad • 7 months ago

can you please explain Stochastic Gradient Descent like this

^ | v • Reply • Share ›

Kiran Prasad • 7 months ago

awesome , very celarly explained. Thank you

^ | v • Reply • Share ›

Rawan • 9 months ago

Thanks for clarification! Love how simple you explained many points

Thanks for clarification! Love how simple you explained many points.

^ | v • Reply • Share ›

Mahdi Soltanzadeh • 9 months ago

God Bless Your Soul

^ | v • Reply • Share ›

Ash • a year ago • edited

It was a great explanation indeed :) ... really helped me understanding this concept to the core....

^ | v • Reply • Share ›

Mamta Pandey • a year ago

excellent and very helpful for me. How van I apply on multiple data sets.

^ | v • Reply • Share ›

Maryam Saleki • a year ago

The best explanation i have ever seen! thank you so much Jahnavi.

^ | v • Reply • Share ›

Wira Maharddhika • a year ago

Easy to understand, thanks for this good explanation

^ | v • Reply • Share ›

Enthusiast • 2 years ago • edited

Thanks Jahnavi,
Question: What's the reason you don't take the average after the summation of -(Y-YP) and -(Y-YP)X ?

^ | v • Reply • Share ›

Aown Mohammad → Enthusiast • a year ago

Because she did't use Summation Symbol. She computed error for only two values and divided it by 2. If you run a loop for N values, divide it by N.

^ | v • Reply • Share ›

Hemendra Akuthota → Aown Mohammad • 8 months ago

Its really nice article and easy to understand.
But, I too have the same question as above: here we did run through the loop of values, but at the end didn't average. Can you help me to understand why didn't we average.

^ | v • Reply • Share ›

Show more replies

sudharsan tk • 2 years ago

Hi , to be honest this was the simplest explanation i came across. Could you please let me know how to do the same if we have more than one variable?

^ | v • Reply • Share ›

disqus_5kgkbbaz50 • 2 years ago

Nice very simplified explanation.Thanks

^ | v • Reply • Share ›

William Lucas • 2 years ago

This was amazing! Could you post a google sheet (or other) so that we could copy/play around with the data ourselves?

^ | v • Reply • Share ›

Kumar Siddappa • a year ago

how did you get this equation
∂SSE/∂a = – (Y-YP)

∂SSE/∂b = – (Y-YP)X

^ | v • Reply • Share ›

Amit Singh Baghel ➦ Kumar Siddappa • 10 months ago

Would like to know the same. Please explain.

^ | ˅ • Reply • Share ›

Amit Singh Baghel ➦ Amit Singh Baghel • 10 months ago

https://www.codeproject.com...

^ | ˅ • Reply • Share ›

Previous post
Next post

Top Stories Past 30 Days

Most Popular

1. A Step-by-Step Guide to Transitioning your Career to Data Science Part 1
2. 7 Steps to Mastering SQL for Data Science 2019 Edition
3. How (not) to use Machine Learning for time series forecasting: Avoiding the pitfalls
4. The 3 Biggest Mistakes on Learning Data Science
5. 7 Steps to Mastering Intermediate Machine Learning with Python 2019 Edition
6. The Third Wave Data Scientist
7. The most desired skill in data science

Most Shared

1. 7 Steps to Mastering SQL for Data Science — 2019 Edition
2. Python leads the 11 top Data Science, Machine Learning platforms: Trends and Analysis
3. Jupyter Notebooks: Data Science Reporting
4. A Step-by-Step Guide to Transitioning your Career to Data Science Part 1
5. 7 Steps to Mastering Intermediate Machine Learning with Python — 2019 Edition
6. The Data Fabric for Machine Learning – Part 1
7. Machine Learning in Agriculture: Applications and Techniques

Latest News

- How to Learn Python for Data Science the Right Way
- Show off your Data Science skills with Kaggle Kernels
- 5 Useful Statistics Data Scientists Need to Know
- Berkeley Lab: Associate Laboratory Director for Computi...
- First hand experience from Uber, Microsoft & more ...
- Why Machine Learning is vulnerable to adversarial attac...

KDnuggets Home » News » 2017 » Apr » Tutorials, Overviews » Keep it simple! How to understand Gradient Descent algorithm (17:n17)