# Optimizers in Deep Learning

Musstafa · Follow

6 min read · Mar 28, 2021

338

## What is an optimizer?

**Optimizers** are algorithms or methods used to minimize an error function(*loss function*)or to maximize the efficiency of production. Optimizers are mathematical functions which are dependent on model's learnable parameters i.e Weights & Biases. Optimizers help to know how to change weights and learning rate of neural network to reduce the losses.

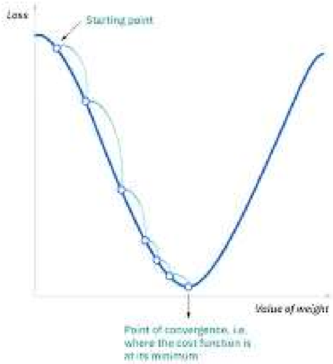
This post will walk you through the optimizers and some popular approaches.

## Types of optimizers

Let's learn about different types of optimizers and how they exactly work to minimize the loss function.

## Gradient Descent

Gradient descent is an optimization algorithm based on a convex function and tweaks its parameters iteratively to minimize a given function to its local minimum. Gradient Descent iteratively reduces a loss function by moving in the direction opposite to that of steepest ascent. It is dependent on the derivatives of the loss function for finding minima. uses the data of the entire training set to calculate the gradient of the cost function to the parameters which requires large amount of memory and slows down the process.



Gradient Descent

$$W_{new} = W_{old} - \alpha * \frac{\partial(Loss)}{\partial(W_{old})}$$

**Advantages of Gradient Descent**

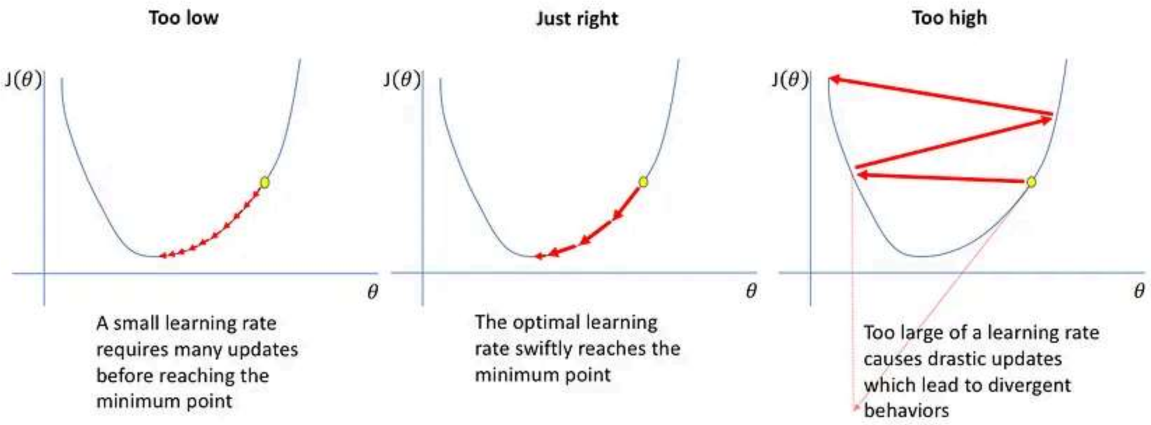
1. Easy to understand
2. Easy to implement

**Disadvantages of Gradient Descent**

1. Because this method calculates the gradient for the entire data set in one update, the calculation is very slow.
2. It requires large memory and it is computationally expensive.

## Learning Rate

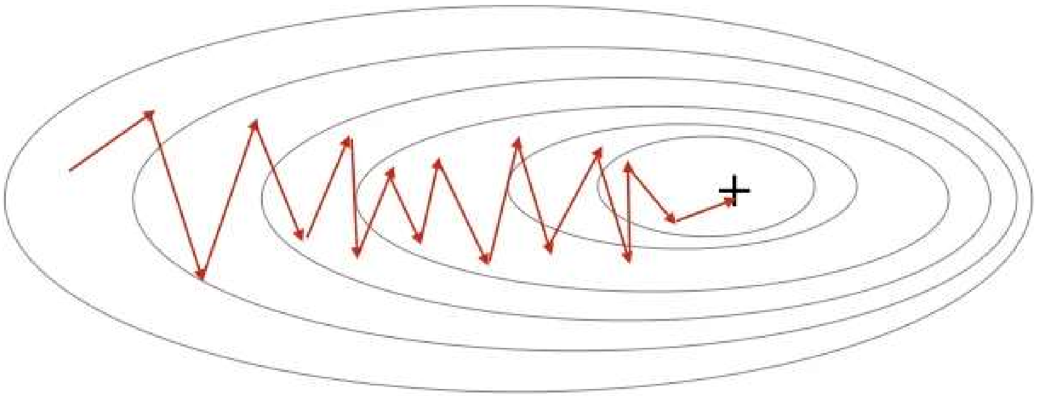
How big/small the steps are gradient descent takes into the direction of the local minimum are determined by the learning rate, which figures out how fast or slow we will move towards the optimal weights.


Too low
J(θ)
θ
A small learning rate requires many updates before reaching the minimum point
Just right
J(θ)
θ
The optimal learning rate swiftly reaches the minimum point
Too high
J(θ)
θ
Too large of a learning rate causes drastic updates which lead to divergent behaviors

Learning Rate

## Stochastic Gradient Descent

It is a variant of Gradient Descent. It update the model parameters one by one. If the model has 10K dataset SGD will update the model parameters 10k times.



Stochastic Gradient Descent

**Advantages of Stochastic Gradient Descent**

1. Frequent updates of model parameter
2. Requires less Memory.
3. Allows the use of large data sets as it has to update only one example at a time.

**Disadvantages of Stochastic Gradient Descent**

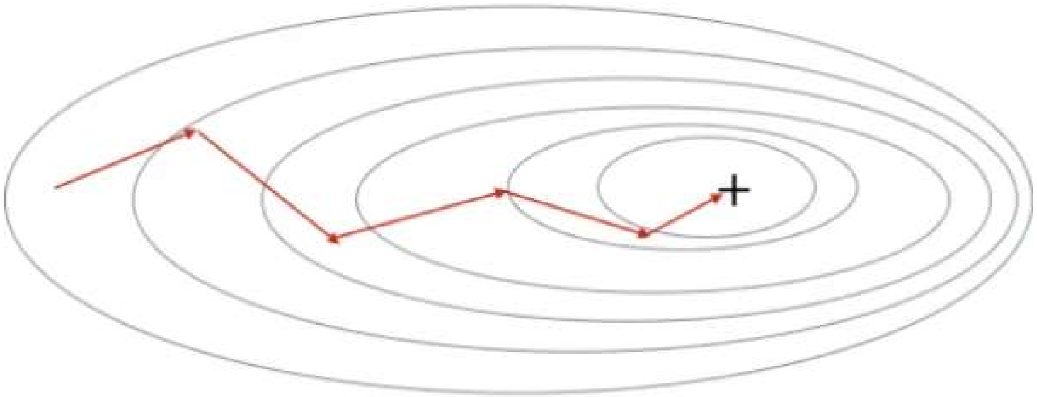
1. The frequent can also result in noisy gradients which may cause the error to increase instead of decreasing it.

2. High Variance.

3. Frequent updates are computationally expensive.

## Mini-Batch Gradient Descent

It is a combination of the concepts of SGD and batch gradient descent. It simply splits the training dataset into small batches and performs an update for each of those batches. This creates a balance between the robustness of stochastic gradient descent and the efficiency of batch gradient descent. it can reduce the variance when the parameters are updated, and the convergence is more stable. It splits the data set in batches in between 50 to 256 examples, chosen at random.



Mini Batch Gradient Descent

**Advantages of Mini Batch Gradient Descent:**

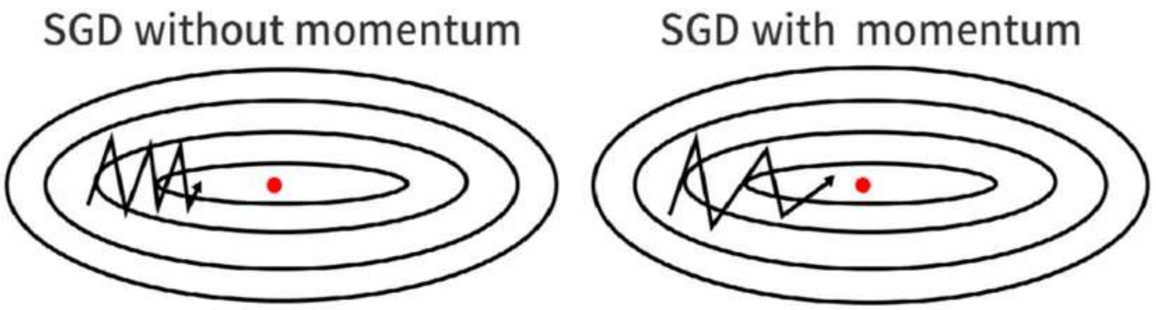
1. It leads to more stable convergence.
2. more efficient gradient calculations.
3. Requires less amount of memory.

**Disadvantages of Mini Batch Gradient Descent**

1. Mini-batch gradient descent does not guarantee good convergence,
2. If the learning rate is too small, the convergence rate will be slow. If it is too large, the loss function will oscillate or even deviate at the minimum value.

## SGD with Momentum

**SGD with Momentum** is a stochastic optimization method that adds a momentum term to regular stochastic gradient descent. Momentum simulates the inertia of an object when it is moving, that is, the direction of the previous update is retained to a certain extent during the update, while the current update gradient is used to fine-tune the final update direction. In this way, you can increase the stability to a certain extent, so that you can learn faster, and also have the ability to get rid of local optimization.



SGD with Momentum

$$V_{new} = \eta * V_{old} - \alpha * \frac{\partial(Loss)}{}$$

Momentum Formula

**Advantages of SGD with momentum**

1. Momentum helps to reduce the noise.
2. Exponential Weighted Average is used to smoothen the curve.

**Disadvantage of SGD with momentum**

1. Extra hyperparameter is added.

## AdaGrad(Adaptive Gradient Descent)

In all the algorithms that we discussed previously the learning rate remains constant. The intuition behind AdaGrad is can we use different Learning Rates for each and every neuron for each and every hidden layer based on different iterations.

$$W_{new} = W_{old} + \frac{\alpha}{\sqrt{cache_{new}} + \epsilon} * \frac{\partial(Loss)}{\partial(W_{old})}$$

**Advantages of AdaGrad**

1. Learning Rate changes adaptively with iterations.
2. It is able to train sparse data as well.

**Disadvantage of AdaGrad**

1. If the neural network is deep the learning rate becomes very small number which will cause dead neuron problem.

## RMS-Prop (Root Mean Square Propagation)

RMS-Prop is a special version of Adagrad in which the learning rate is an exponential average of the gradients instead of the cumulative sum of squared gradients. RMS-Prop basically combines momentum with AdaGrad.

$$cache_{new} = \gamma * cache_{old} + (1 - \gamma) * (\frac{\partial(Loss)}{\partial(W_{old})})^2$$

**Advantages of RMS-Prop**

1. In RMS-Prop learning rate gets adjusted automatically and it chooses a different learning rate for each parameter.

**Disadvantages of RMS-Prop**

1. Slow Learning

## AdaDelta

Adadelta is an extension of Adagrad and it also tries to reduce Adagrad's aggressive, monotonically reducing the learning rate and remove decaying learning rate problem. In Adadelta we do not need to set the default learning rate as we take the ratio of the running average of the previous time steps to the current gradient.

**Advantages of Adadelta**

1. The main advantage of AdaDelta is that we do not need to set a default learning rate.

**Disadvantages of Adadelta**

1. Computationally expensive

## Adam(Adaptive Moment Estimation)

Adam optimizer is one of the most popular and famous gradient descent optimization algorithms. It is a method that computes adaptive learning rates for each parameter. It stores both the decaying average of the past gradients , similar to momentum and also the decaying average of the past squared gradients , similar to RMS-Prop and Adadelta. Thus, it combines the advantages of both the methods.

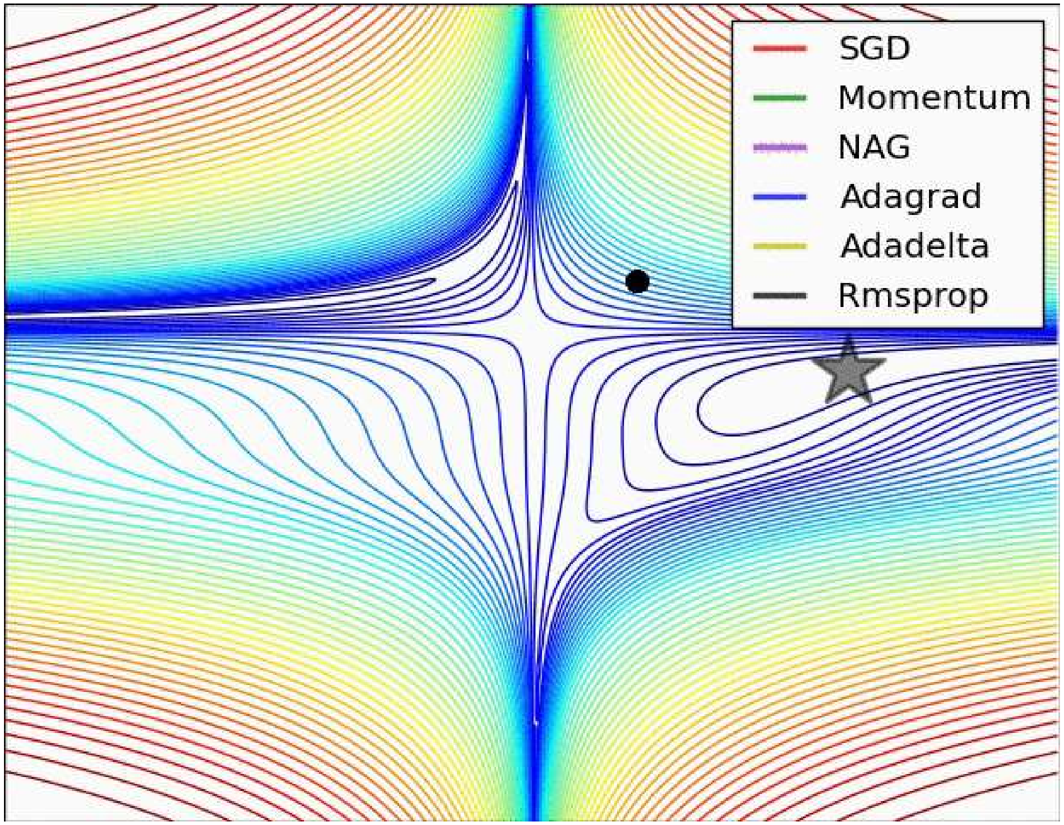
$$\mathrm{w}_{\mathrm{t}} = \mathrm{w}_{\mathrm{t}-1} - \frac{\eta}{\sqrt{\mathrm{S}_{\mathrm{dw}_{\mathrm{t}}} - \varepsilon}} * \mathrm{V}_{\mathrm{dw}_{\mathrm{t}}}$$

$$b_{\mathrm{t}} = b_{\mathrm{t}-1} - \frac{\eta}{\sqrt{\mathrm{S}_{db_{\mathrm{t}}} - \varepsilon}} * \mathrm{V}_{db_{\mathrm{t}}}$$

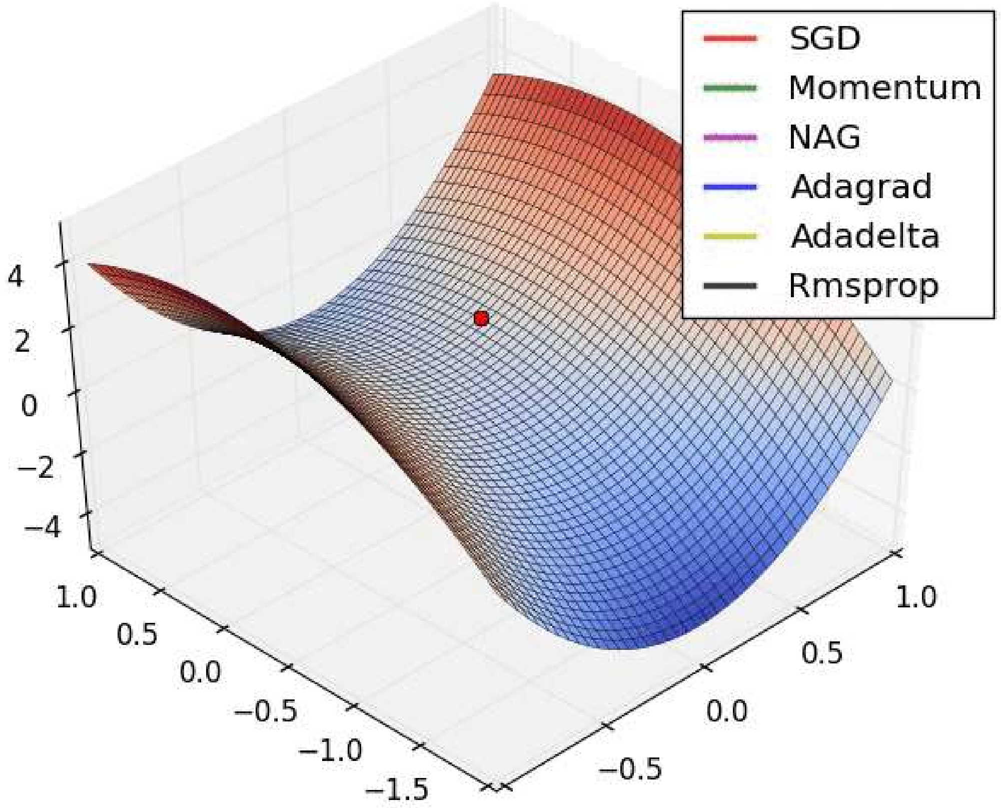
**Advantages of Adam**

1. Easy to implement
2. Computationally efficient.
3. Little memory requirements.

## Comparison



Optimizers Comparison



Optimization on saddle point

## How to choose optimizers?

- If the data is sparse, use the self-applicable methods, namely Adagrad, Adadelta, RMSprop, Adam.
- RMSprop, Adadelta, Adam have similar effects in many cases.
- Adam just added bias-correction and momentum on the basis of RMSprop,
- As the gradient becomes sparse, Adam will perform better than RMSprop.

I hope this article has helped you learn and understand more about these concepts.

**Mlearning.ai Submission Suggestions**

How to become a writer on Mlearning.ai

medium.com

Become a ML Writer

Optimizer Deep Learning Adam Rmsprop Ml So Good

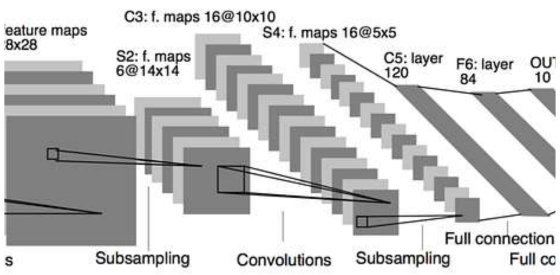
## Written by Musstafa

41 Followers · 3 Following

Follow

Data Scientist at TCS

## More from Musstafa



Musstafa

## LeNet in depth

LeNet was the first architecture in modern CNN introduced in 1998 by Yann LeCun, Léo...

Mar 25, 2021 5

Maximilian Vogel

## The ChatGPT list of lists: A collection of 3000+ prompts, GPT...

Updated Sep-01, 2024. Added New Introductions, Prompts, Lists and Tools

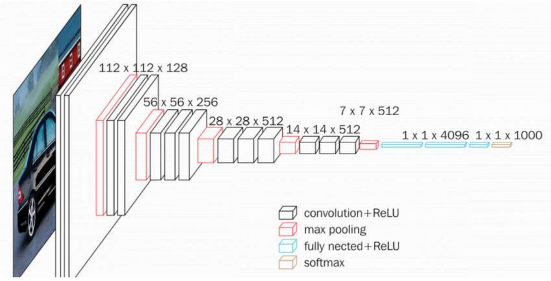
Feb 8, 2023 12.7K 155

Maximilian Vogel

## The 10 Best Free Prompt Engineering Courses & Resource...

Updated Jan-21, 2024: Added a bonus resource.

Sep 7, 2023 1.4K 20



Musstafa

## VGG in depth

In my previous post we are talking about AlexNet which was a revolutionary...

Mar 29, 2021 8

See all from Musstafa

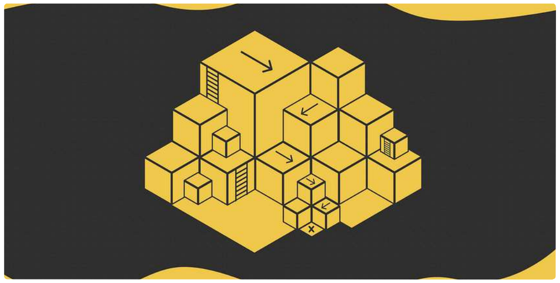
## Recommended from Medium

In The Deep Hub by Jorgecardete

### Convolutional Neural Networks: A Comprehensive Guide

Exploring the power of CNNs in image analysis

Feb 7 2.6K 38



In Towards Data Science by Vyacheslav Efimov

### Understanding Deep Learning Optimizers: Momentum, AdaGra...

Gain intuition behind acceleration training techniques in neural networks

✦ Dec 30, 2023 456 4

Lists

Natural Language Processing

1842 stories · 1466 saves

Practical Guides to Machine Learning

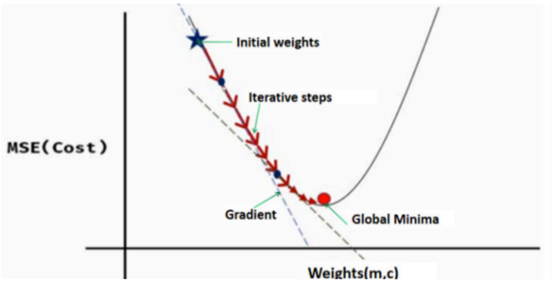
10 stories · 2068 saves

data science and AI

40 stories · 296 saves

Tech & Tools

21 stories · 361 saves



Samuel Ozechi

## Stochastic Gradient Descent For Deep Learning

Understanding the parameter optimization process for deep learning models.

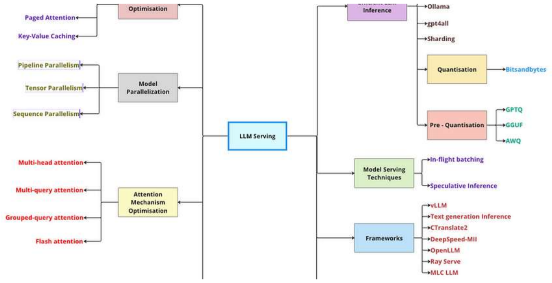
Sep 22

Mikel

## Pinterest ML Internship Summer 2025

Looking for a tech internship for the Summer 2025. I share my recent Interview experienc...

Oct 27 4 1



Vipra Singh

### Building LLM Applications: Serving LLMs (Part 9)

Learn Large Language Models ( LLM ) through the lens of a Retrieval Augmented...

✦ Apr 18 👏 881 💬 6

In Towards AI by Abhinav Kimothi

### Gradient Descent and the Melody of Optimization Algorithms

Source : Image generated using AI by Author

✦ Jan 11 👏 231 💬 6

See more recommendations