The Vanishing Gradient Problem

The Problem, Its Causes, Its Significance, and Its Solutions

Chi-Feng Wang · Follow

Published in Towards Data Science

3 min read · Jan 8, 2019

Listen Share

Title Image // Source

The problem:

As more layers using certain activation functions are added to neural networks, the gradients of the loss function approaches zero, making the network hard to train.

Why:

Certain activation functions, like the sigmoid function, squishes a large input space into a small input space between 0 and 1. Therefore, a large change in the input of the sigmoid function will cause a small change in the output. Hence, the derivative becomes small.

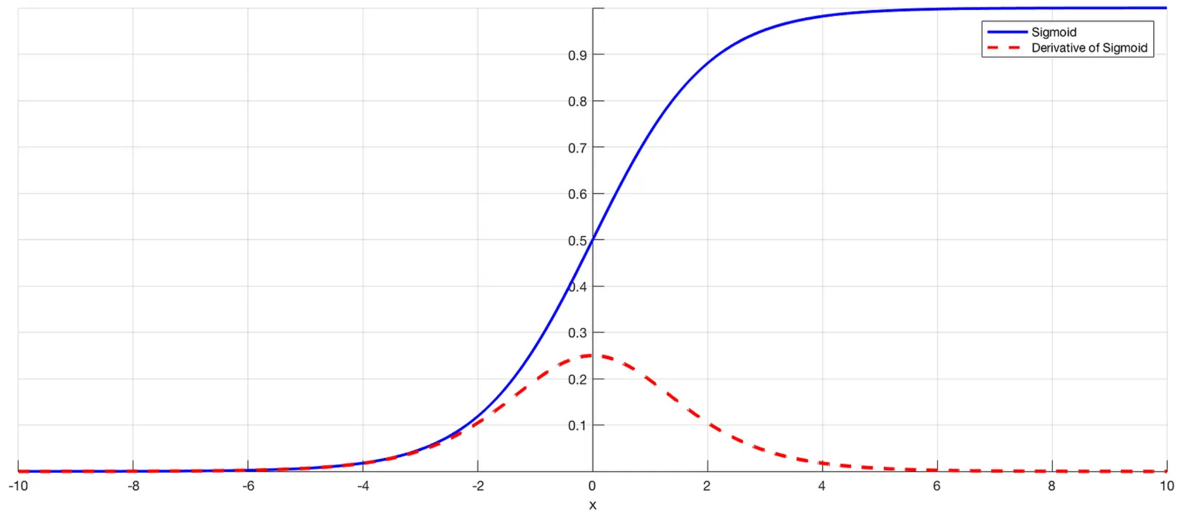


Image 1: The sigmoid function and its derivative // Source

As an example, Image 1 is the sigmoid function and its derivative. Note how when the inputs of the sigmoid function becomes larger or smaller (when |x| becomes bigger), the derivative becomes close to zero.

Why it's significant:

For shallow network with only a few layers that use these activations, this isn't a big problem. However, when more layers are used, it can cause the gradient to be too small for training to work effectively.

Gradients of neural networks are found using backpropagation. Simply put, backpropagation finds the derivatives of the network by moving layer by layer from the final layer to the initial one. By the chain rule, the derivatives of each layer are multiplied down the network (from the final layer to the initial) to compute the derivatives of the initial layers.

However, when *n* hidden layers use an activation like the sigmoid function, *n* small derivatives are multiplied together. Thus, the gradient decreases exponentially as we propagate down to the initial layers.

A small gradient means that the weights and biases of the initial layers will not be updated effectively with each training session. Since these initial layers are often crucial to recognizing the core elements of the input data, it can lead to overall inaccuracy of the whole network.

Solutions:

The simplest solution is to use other activation functions, such as ReLU, which doesn’t cause a small derivative.

Residual networks are another solution, as they provide residual connections straight to earlier layers. As seen in Image 2, the residual connection directly adds the value at the beginning of the block, **x**, to the end of the block (F(x)+x). This residual connection doesn’t go through activation functions that “squashes” the derivatives, resulting in a higher overall derivative of the block.

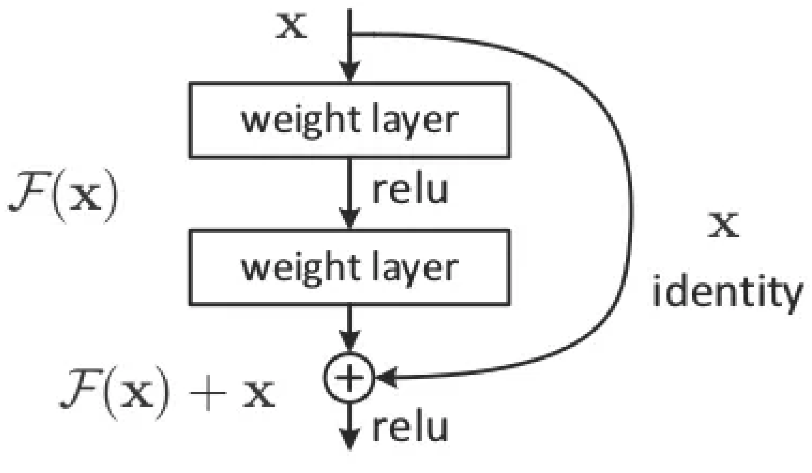
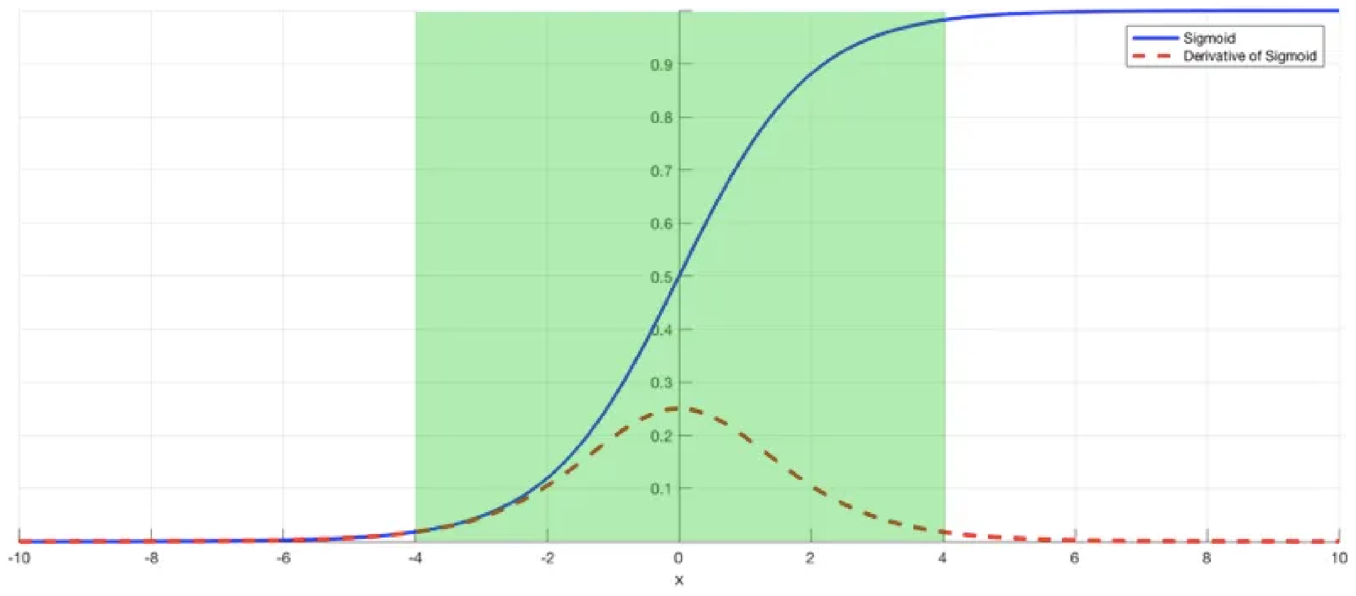

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Image 2: A residual block

Finally, batch normalization layers can also resolve the issue. As stated before, the problem arises when a large input space is mapped to a small one, causing the derivatives to disappear. In Image 1, this is most clearly seen at when |x| is big. Batch normalization reduces this problem by simply normalizing the input so |x| doesn't reach the outer edges of the sigmoid function. As seen in Image 3, it normalizes the input so that most of it falls in the green region, where the derivative isn't too small.



Do leave a comment below if you have any questions or suggestions :)

Read these articles for more information:

- https://www.quora.com/What-is-the-vanishing-gradient-problem
- https://en.wikipedia.org/wiki/Vanishing_gradient_problem
- https://towardsdatascience.com/intuit-and-implement-batch-normalization-c05480333c5b

Machine Learning Vanishing Gradient Neural Networks Artificial Intelligence

Activation Functions

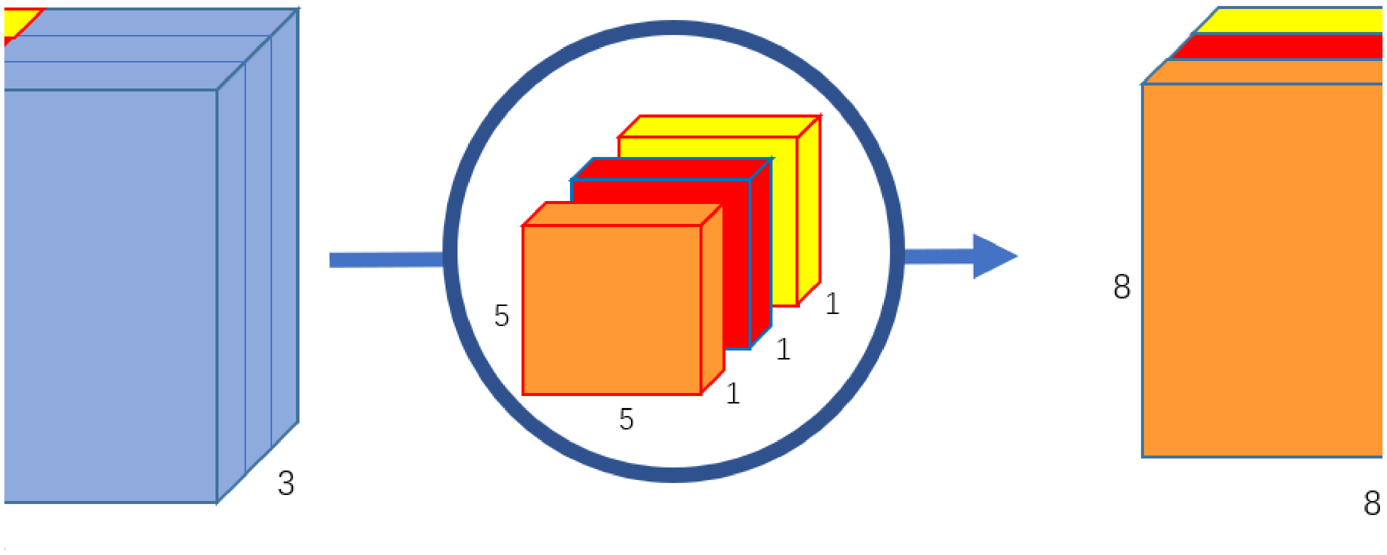
Follow

Written by Chi-Feng Wang

1.6K Followers · Writer for Towards Data Science

Student at UC Berkeley; Machine Learning Enthusiast

More from Chi-Feng Wang and Towards Data Science



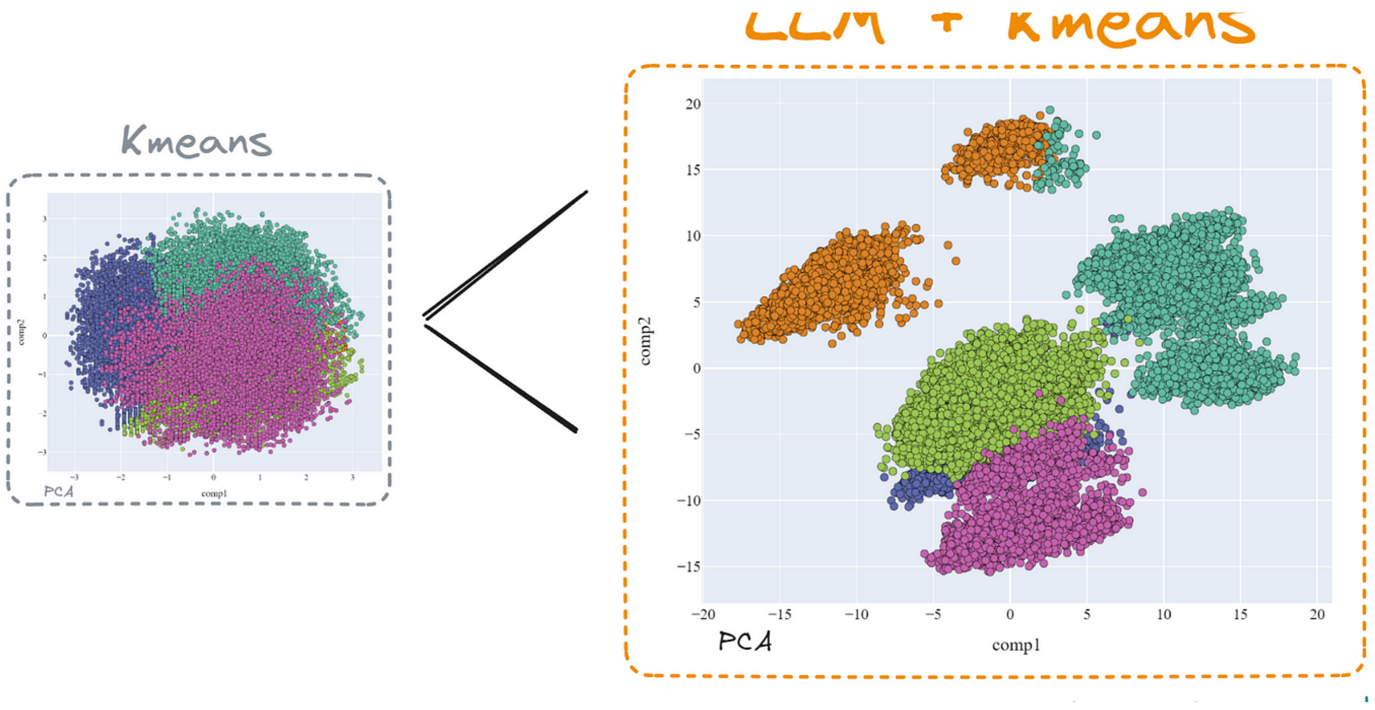
Chi-Feng Wang in Towards Data Science

A Basic Introduction to Separable Convolutions

Explaining spatial separable convolutions, depthwise separable convolutions, and the use of 1×1 kernels in a simple manner.

8 min read · Aug 13, 2018

7.3K 45



Damian Gil in Towards Data Science

Mastering Customer Segmentation with LLM

Unlock advanced customer segmentation techniques using LLMs, and improve your clustering models with advanced techniques

23 min read · Sep 26

2.4K 22

Giuseppe Scalamogna in Towards Data Science

New ChatGPT Prompt Engineering Technique: Program Simulation

A potentially novel technique for turning a ChatGPT prompt into a mini-app.

9 min read · Sep 3

1.7K 17

Chi-Feng Wang in Towards Data Science

Finding the Gradient of a Vector Function

Part 3 of Step by Step: The Math Behind Neural Networks

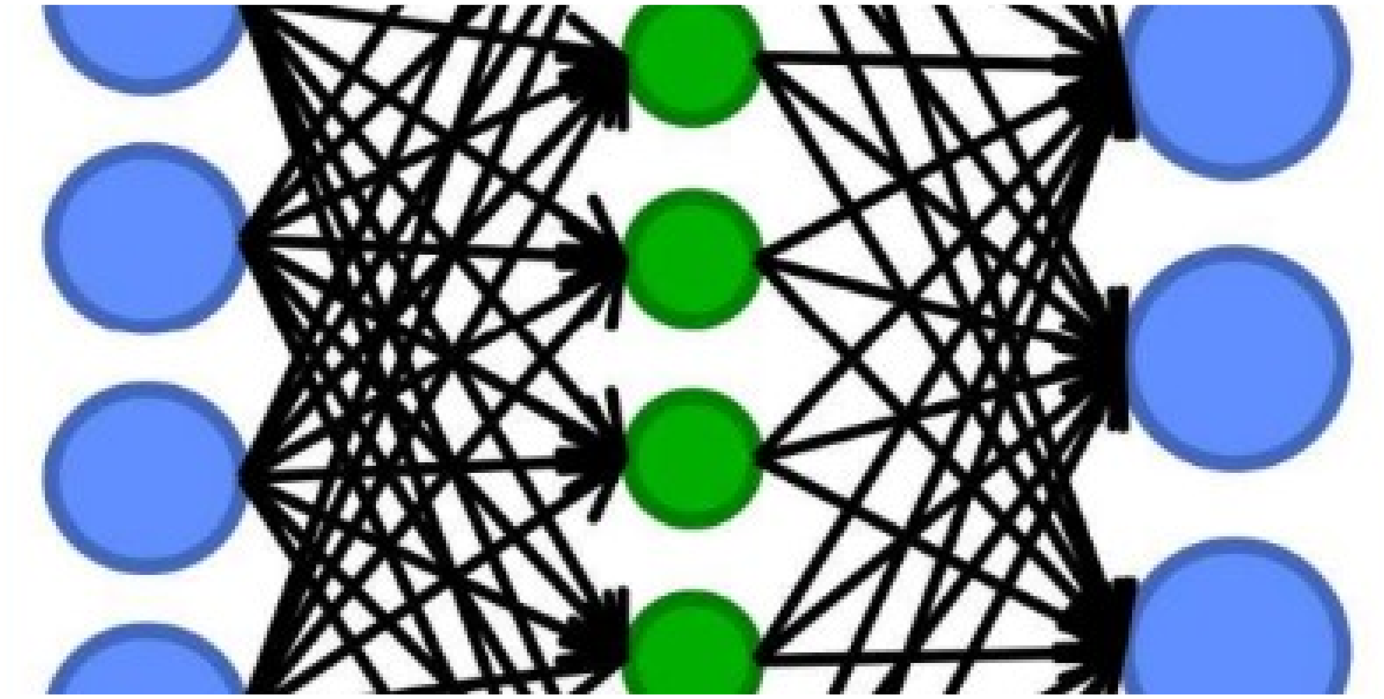
9 min read · Oct 20, 2018

471 1

See all from Chi-Feng Wang

See all from Towards Data Science

Recommended from Medium



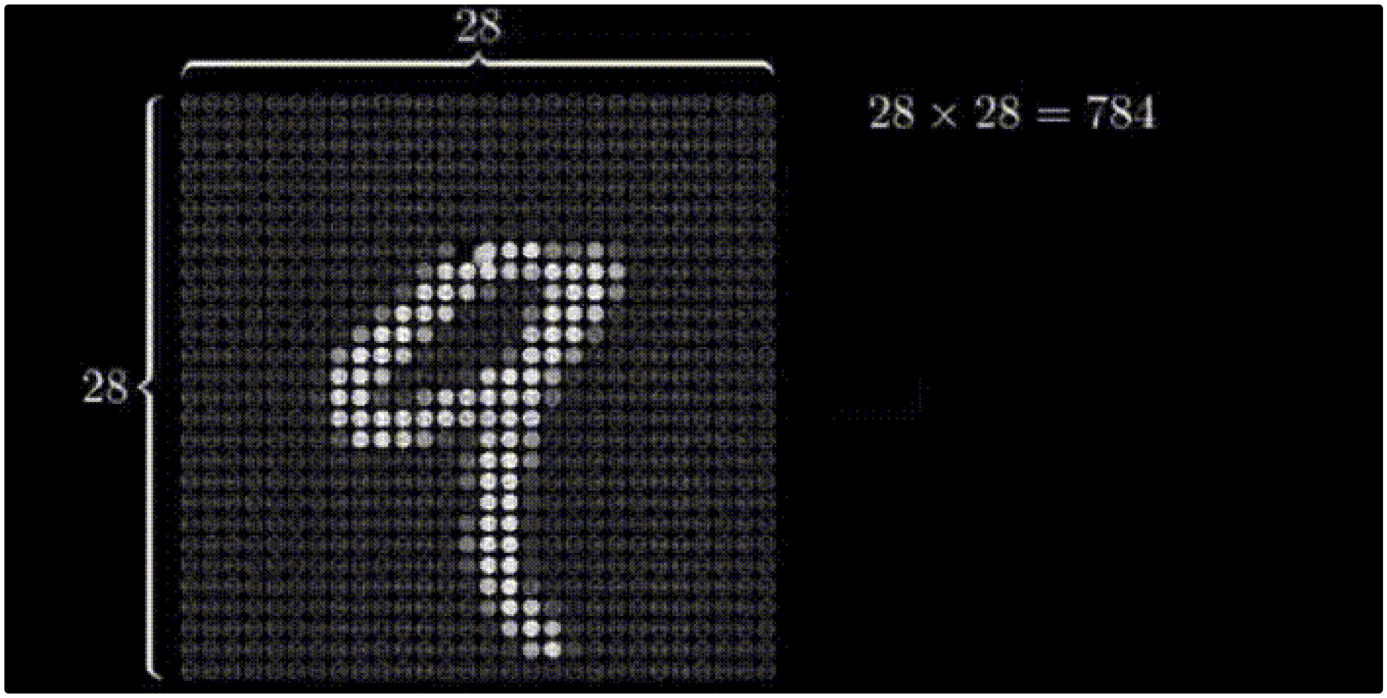
Tamanna

Backpropagation in Neural Networks: A Comprehensive Guide

Backpropagation is a popular algorithm used in artificial neural networks (ANNs) for training deep learning models. It is a supervised...

5 min read · May 4

551



Sadaf Saleem

Neural Networks in 10mins. Simply Explained!

What are Neural Networks?

9 min read · May 15

257 2

Lists

Predictive Modeling w/ Python

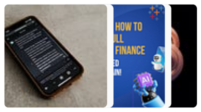
20 stories · 453 saves

AI Regulation

6 stories · 142 saves

Natural Language Processing

669 stories · 286 saves



ChatGPT prompts

24 stories · 466 saves

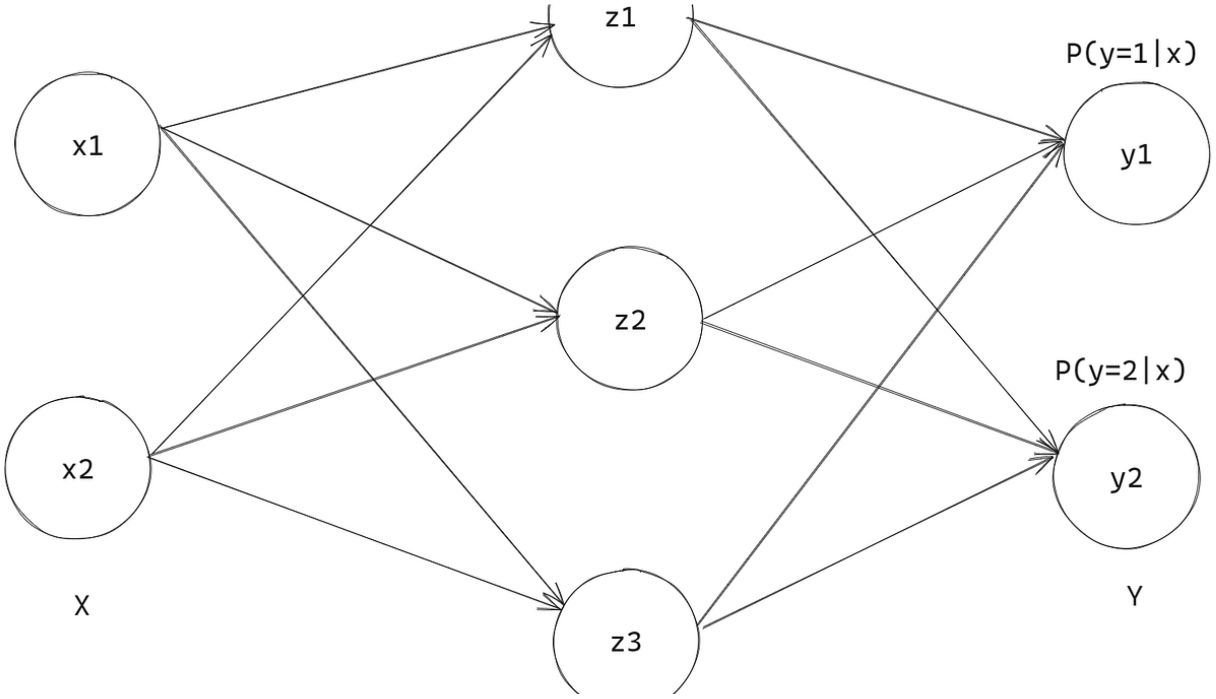
Ebin Babu Thomas

Backpropagation

Backpropagation is a supervised learning algorithm used in training artificial neural networks. The main intuition behind backpropagation…

2 min read · May 14

1



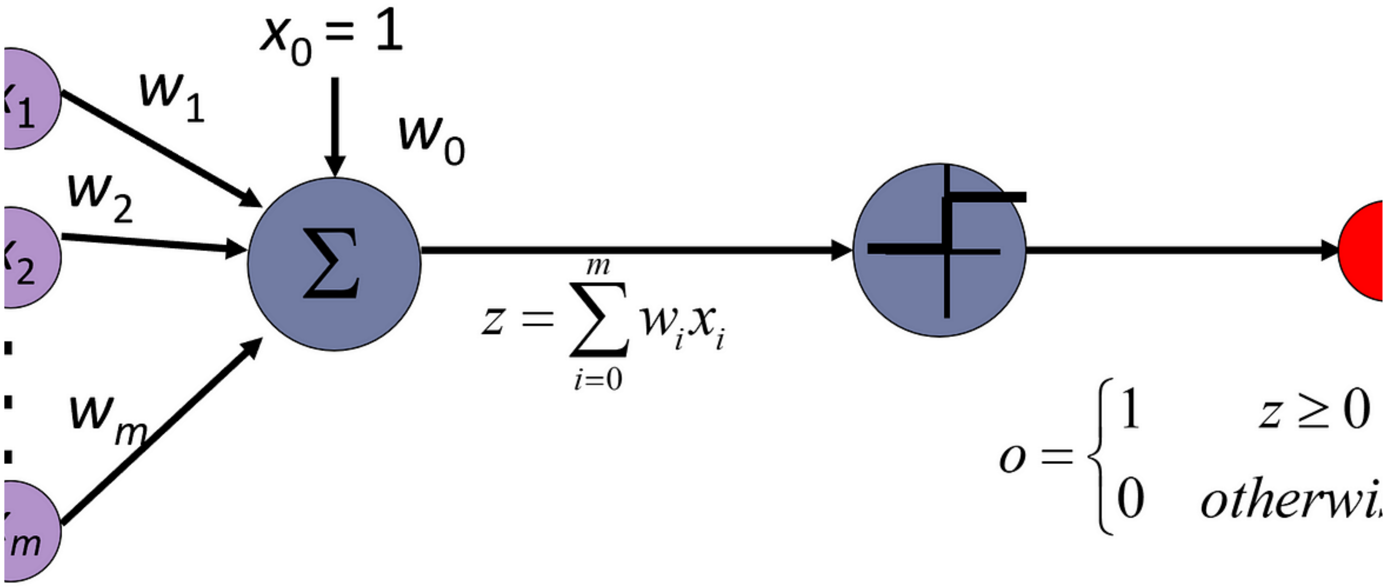
Ramanthind

Machine Learning from Scratch: Neural Networks

Work in progress……

13 min read · Sep 17

7


$x_0 = 1$
w_1
w_0
w_2
Σ
$z = \sum_{i=0}^{m} w_i x_i$
w_m
$o = \begin{cases} 1 & z \geq 0 \\ 0 & otherwi \end{cases}$

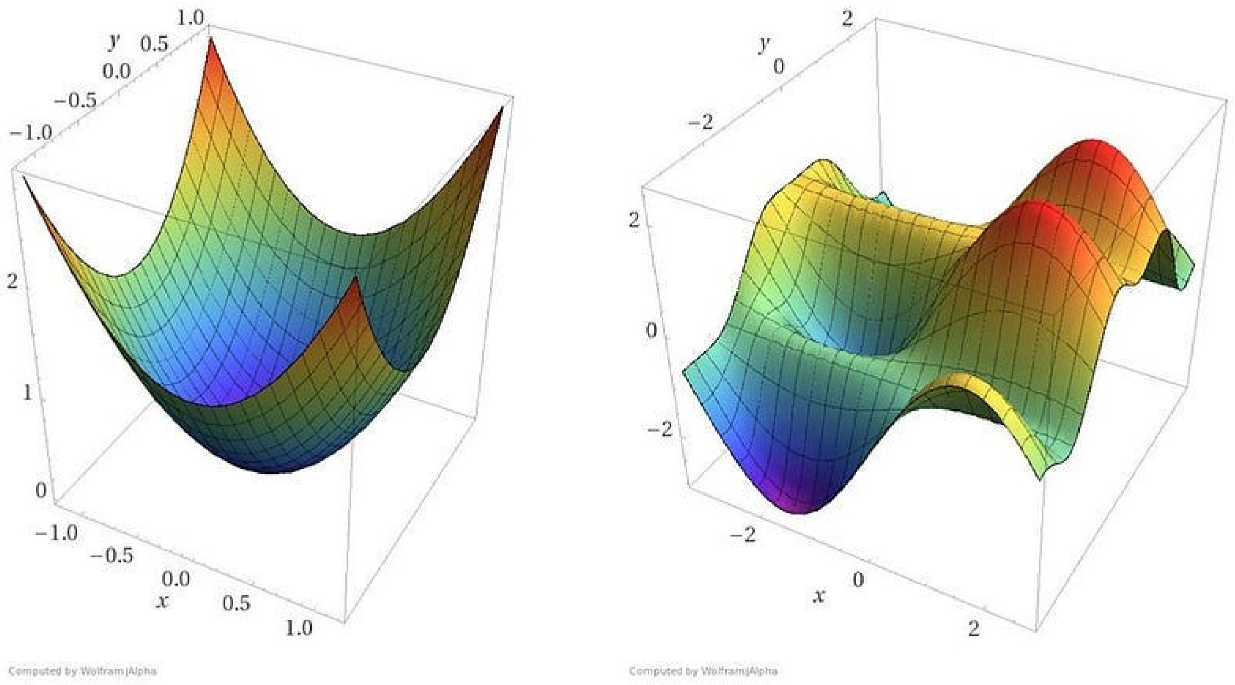
Dr. Roi Yehoshua in Towards Data Science

Perceptrons: The First Neural Network Model

Overview and implementation in Python Perceptrons are one of the earliest computational models of neural networks (NNs), and they form the

✦ · 14 min read · Mar 28

247 7



Deepjyoti Bhattacharjee

Gradient Descent

Gradient descent is an optimization algorithm used to minimize a cost function in machine learning. It works by iteratively adjusting the…

2 min read · Jul 16

5

See more recommendations