Vanishing Gradient Problem

Thursday, April 7, 2022 7:45 AM

In machine learning, the **vanishing gradient problem** is encountered when training artificial neural networks with gradient-based learning methods and backpropagation. In such methods, during each iteration of training each of the neural network's weights receives an update proportional to the partial derivative of the error function with respect to the current weight. The problem is that in some cases, the gradient will be vanishingly small, effectively preventing the weight from changing its value. In the worst case, this may completely stop the neural network from further training. ~~As one example of the problem cause, traditional activation~~

GD

1) $0.1 \times 0.1 \times 0.1 \times 0.1 = 0.0001$ → VGP

2) Deep NN

3) Sigmoid / tanh → Af → 0-1

Backprop

0-0.5

deep

sigmoid

sigmoid derivative

$L \to \hat{y} \to o_{11} \to w'_{11}$

$w_n = w_0 - \eta \frac{\partial L}{\partial w}$ → derivative of L wrt weight

$w_0 = 1$ ↑ small change

$\frac{\partial L}{\partial w'_{11}} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial z} \times \frac{\partial z}{\partial o_{11}} \times \frac{\partial o_{11}}{\partial w'_{11}}$

$0 < x < 1$ $0 < x < 1$ 0.1

0.0001

$w_n = 1 - 0.01 \times 0.0001$

$w_n = 0.99999$ → vanishingly small

did not change

{ Sigmoid tanh }

Backprop → model training → x

$\frac{\partial L}{\partial w'_{11}} = 0.000001$

↳ 80's, 90's → main reasons DL failed

0 - 100000

0 - 1

1 - 1

How to recognize?

1) Loss focus → epoch → no changes → VGP

2) weights → graph value

w'_{11}

Keras

{ loss after epoch }

Tensorboard → callbacks

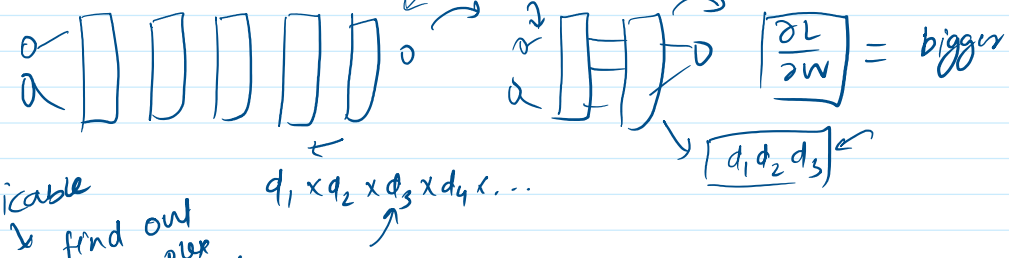
epoch

$w_n = w_0 - \eta \frac{\partial L}{\partial w}$ $\frac{w_0 - w_n}{\eta} =$

$$w_n = w_0 - \eta \left[\frac{\partial L}{\partial w}\right] \rightarrow \qquad \boxed{\frac{w_0 - w_n}{\eta} =}$$

How to handle Vanishing Gradient Problem →

1) Reduce model complexity



Applicable
↳ find out complex patterns

2) Using ReLU Activation functions

max (0, z) → 0, 1

−1000 − 1000 → squish
0 − 1000
0 − 1

z = 1 × 1 × 1 = 1 → vgp

dying relu ↳ derivative $w_n = w_0$

{leaky relu}

3) Proper weight init → Global → xavier

4) Batch norm → layer →

5) Residual Network → CNN → ResNET
↳ building block → ANN

Exploding Gradient Problem → RNN

10, 10, 10, 10 → 10000

big number

$\frac{\partial L}{\partial w_{11}} = d_1 \times d_1 \times \phi_1 \times \phi_3$

$w_n = w_0 - \eta \frac{\partial L}{\partial w}$ (0.1, 1000)

random

Loss ↓ x 1 − 100 → 99 → 1000 ↘

{Gradient clipping}

expo