Optimisation

Kate Farrahi

ECS Southampton

January 24, 2020

[1]

[1] Some of the material in this lecture is based on Andrew Ng's lectures on Optimisation

Why learning can be slow

- If the ellipse is very elongated, the direction of steepest descent is almost perpendicular to the direction towards the minimum
- The gradient vector will have a large component along the short axis of the ellipse and a small component along the long axis of the ellipse.
- This is the opposite of what we want to optimise efficiently

Exponentially Weighted Averages

$$v_t = \beta v_{t-1} + (1 - \beta)\theta_t$$

Exponentially Weighted Averages

$$v_t = \beta v_{t-1} + (1-\beta)\theta_t$$

v_t is approximately average over $\approx \frac{1}{1-\beta}$ days

Exponentially Weighted Averages

$v_t = \beta v_{t-1} + (1 - \beta)\theta_t$

v_t is approximately average over $\approx \frac{1}{1-\beta}$ days

For example
$v_{100} = 0.9 v_{99} + 0.1\theta_{100}$
$v_{99} = 0.9 v_{98} + 0.1\theta_{99}$
$v_{98} = 0.9 v_{97} + 0.1\theta_{98}$
...

Exponentially Weighted Averages

$v_t = \beta v_{t-1} + (1-\beta)\theta_t$

v_t is approximately average over $\approx \frac{1}{1-\beta}$ days

For example
$v_{100} = 0.9v_{99} + 0.1\theta_{100}$
$v_{99} = 0.9v_{98} + 0.1\theta_{99}$
$v_{98} = 0.9v_{97} + 0.1\theta_{98}$
...

$v_{100} = 0.1\theta_{100} + 0.9[0.1\theta_{99} + 0.9[\dots]]]$

Exponentially Weighted Averages

$v_t = \beta v_{t-1} + (1-\beta)\theta_t$

v_t is approximately average over $\approx \frac{1}{1-\beta}$ days

For example
$v_{100} = 0.9 v_{99} + 0.1\theta_{100}$
$v_{99} = 0.9 v_{98} + 0.1\theta_{99}$
$v_{98} = 0.9 v_{97} + 0.1\theta_{98}$
...

$v_{100} = 0.1\theta_{100} + 0.9[0.1\theta_{99} + 0.9[\ldots]]]$

$v_{100} = 0.1\theta_{100} + 0.9 * 0.1 * \theta_{99} + 0.1 * (0.9)^2 * \theta_{98} + 0.1(0.9)^3\theta_{97} + \ldots$

Momentum

- The momentum method allows to accumulate velocity in directions of low curvature that persist across multiple iterations
- This leads to accelerated progress in low curvature directions compared to gradient descent

Gradient Descent (GD) with Momentum

Learning with momentum is given by

On iteration t:
  Compute dW_t on the current mini-batch

$$V_t = \beta V_{t-1} + (1-\beta) dW_t \tag{1}$$
$$w_t = w_{t-1} - \eta V_t \tag{2}$$

Note that dW_t represents the gradient of the cost function (as computed in standard GD). η is the learning rate and $\beta = 0.9$ is a good choice for the exponentially weighted average parameter.

RMSProp

Learning with RMSProp is given by

On iteration t:

Compute dW on current mini-batch

$$S_{dW_t} = \beta S_{dW_{t-1}} + (1-\beta) dW_t^2 \tag{3}$$

$$w_t = w_{t-1} - \eta \frac{dW_t}{\sqrt{S_{dW_t}}} \tag{4}$$

Bias Correction Motivation

- Let's assume that $v_0 = 0$ and $\beta = 0.9$ and we're considering exponentially weighted averages

Bias Correction Motivation

- Let's assume that $v_0 = 0$ and $\beta = 0.9$ and we're considering exponentially weighted averages
- It follows that $v_1 = \beta(0) + (1 - \beta)\theta_1 = 0.1\, \theta_1$

Bias Correction Motivation

- Let's assume that $v_0 = 0$ and $\beta = 0.9$ and we're considering exponentially weighted averages
- It follows that $v_1 = \beta(0) + (1 - \beta)\theta_1 = 0.1\ \theta_1$
- and $v_2 = \beta((1 - \beta)\theta_1) + (1 - \beta)\theta_2 = 0.0196\ \theta_1 + 0.02\ \theta_2$

Bias Correction Motivation

- Let's assume that $v_0 = 0$ and $\beta = 0.9$ and we're considering exponentially weighted averages
- It follows that $v_1 = \beta(0) + (1 - \beta)\theta_1 = 0.1\ \theta_1$
- and $v_2 = \beta((1 - \beta)\theta_1) + (1 - \beta)\theta_2 = 0.0196\ \theta_1 + 0.02\ \theta_2$

Bias Correction

- Add a bias correction term: $\frac{v_t}{1-\beta^t}$

Bias Correction

- Add a bias correction term: $\frac{v_t}{1-\beta^t}$
- $t = 1$: $\frac{v_1}{1-(0.9)^1} = 10 * v_1$

Bias Correction

- Add a bias correction term: $\frac{v_t}{1-\beta^t}$
- $t = 1$: $\frac{v_1}{1-(0.9)^1} = 10 * v_1$
- $t = 2$: $\frac{v_2}{1-(0.9)^2} = 5.263 * v_2$

Bias Correction

- Add a bias correction term: $\frac{v_t}{1-\beta^t}$
- $t = 1$: $\frac{v_1}{1-(0.9)^1} = 10 * v_1$
- $t = 2$: $\frac{v_2}{1-(0.9)^2} = 5.263 * v_2$
- ...
- $t = 10$: $\frac{v_{10}}{1-(0.9)^{10}} = 1.535 * v_{10}$
- ...
- $t = 20$: $\frac{v_{20}}{1-(0.9)^{20}} = 1.138 * v_{20}$

Adam

Initialize parameters: $V_{dW} = 0, S_{dW} = 0$

On iteration t:

Compute dW_t on current mini-batch

$$V_{dW} = \beta_1 V_{dW} + (1-\beta_1)dW, \quad V_{dW}^{corr} = \frac{V_{dW}}{(1-\beta_1^t)} \tag{5}$$

$$S_{dW} = \beta_2 S_{dW} + (1-\beta_2)dW^2, \quad S_{dW}^{corr} = \frac{S_{dW}}{(1-\beta_2^t)} \tag{6}$$

$$w := w - \eta \frac{V_{dW}^{corr}}{\sqrt{(}S_{dW}^{corr} + \epsilon)} \tag{7}$$