1 The basics of the algorithm

1.1 What is the Adam algorithm

In machine learning, Adam **(Adaptive moment estimation)** is a highly-efficient optimalization algorithm. It's desigend to adjust the learning-rate to the current situation. Imagine you're navigating on a complex terrain, like mountains. Sometimes, you need to take large strides, while other time your strides are to be cautiosly small. That's basically the mechanism of Adam! **It changes the values of parameters depending on the circumstances**.

2 Adam algorithm's explained

2.1 The math behind Adam

1 First, Adam initializes two vectors, **m** and **v** which are the same shape as parameters θ. The **m** vector stores the moving average of the gradients, while **v** keeps track of the moving average of squared gradients. Another variable, **time step counter, t**, is initialized to zero by the algorithm. The variable t stores the number of iteration Adam has completed.

The initial values of the arguments are set as follows:

1 $t = 0$

2 $v_0 = 0$

3 $m_0 = 0$

2 Compute gradients. For each iteration, Adam computes the gradient of the function for the values of the previous parameters.
Mathematically it can be expressed as:

$$G_t = \nabla_\theta f(\theta_{t-1}) \tag{1}$$

3 Update the **m** vector. The new values of **m** vector is as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) G_t \tag{2}$$

Where β_1 is some parameter being in the interval [0;1]. Usually, β_1 is set to 0.9. The G_t is the gradient at time step, m_{t-1} is the previous status of the m vector.

4 Update v (second moment estimate). This vector gives an estimate of the unvariance of the gradients, therefore it stores the squared gradients that are accumulated. The formula for fresh-new **v** is nearly the same. All what changes is squared gradient.

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) G_t^2 \tag{3}$$

v_t is the second-moment vector at time step t. β_2 is the exponential decay rate for second-moment estimates. It is set to 0.999 commonly.

5 Since **m and v** are initialized to 0, they are biased toward 0, especially during the initial time steps. In order to overcome the bias, Adam corrects the vector by the decay rates (b_1 for m, and b_2 for v). The correction is important as it esnures the moving averages are more representative. Mathematically, the correction process can be expressed as follows:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{4}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{5}$$

Where m_t is the first-moment of the gradient, computed as time step = t, β_1^t is the decay for the moment at time step=t.
Similarly, v_t is the second-moment of the gradient, calculated as time step= t, β_2^t is the decay for the moment at time step=t

6 Updating the parameters! Now it's time for most-exciting moment of the algorithm - for updating the values of the parameters. The mathematical formula for new set of parameters, θ_{t+1} is, boom, boom, boom:

$$\theta_{t+1} = \theta_t - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{6}$$

Symbols explanation

1 θ_{t+1} is an updated set of parameters.

2 θ_t is a previous set of parameters, computed at time step = t.

3 v_t is a second-moment gradient (squared gradients)

4 m_t is a first-moment gradient ("pure" gradients)

5 α is so-called learning rate.

6 ϵ is a small, positive floating-point number which prevents division by zero. Commonly it's set to 10^{-8}

3 What parameters do you need to implement Adaptive Moment algorithm?

The number of iteration is adoptively choosen by the algorithm. When the changes between values of function is less than epsilon, then the loop breaks. The only parameters you need are: ϵ **and** α