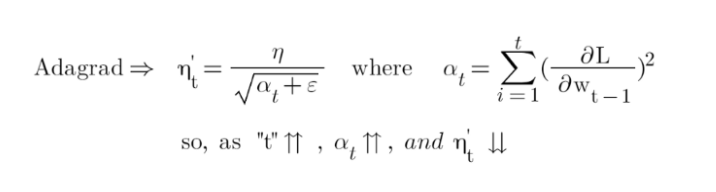
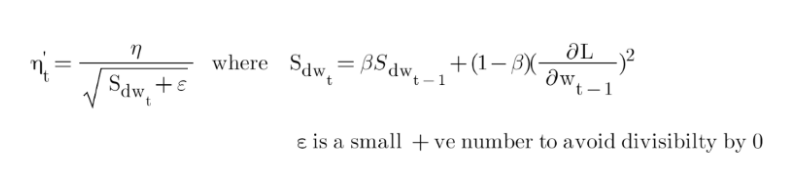
[Adadelta](https://golden.com/wiki/Adadelta) is an extension of Adagrad that attempts to solve its radically diminishing learning rates. The idea behind Adadelta is that instead of summing up all the past squared gradients from 1 to “t” time steps, what if we could restrict the window size. For example, computing the squared gradient of the past 10 gradients and average out. This can be achieved using Exponentially Weighted Averages over Gradient.

****

The above equation shows that as the time steps “t” increase the summation of squared gradients “α” increases which leads to a decrease in learning rate “η”. In order to resolve the exponential increase in the summation of squared gradients “α”, we replaced the “α” with exponentially weighted averages of squared gradients.

****

So, here unlike the alpha “α '' in Adagrad, where it increases exponentially after every time step. In Adadelta, using the exponentially weighted averages over the past Gradient, an increase in “S*dw”* is under control. The calculation for *“Sdw'' is* similar to the example I did in the Exponentially Weighted Averages section.

Reference:

<https://www.youtube.com/watch?v=9wFBbAQixBM>