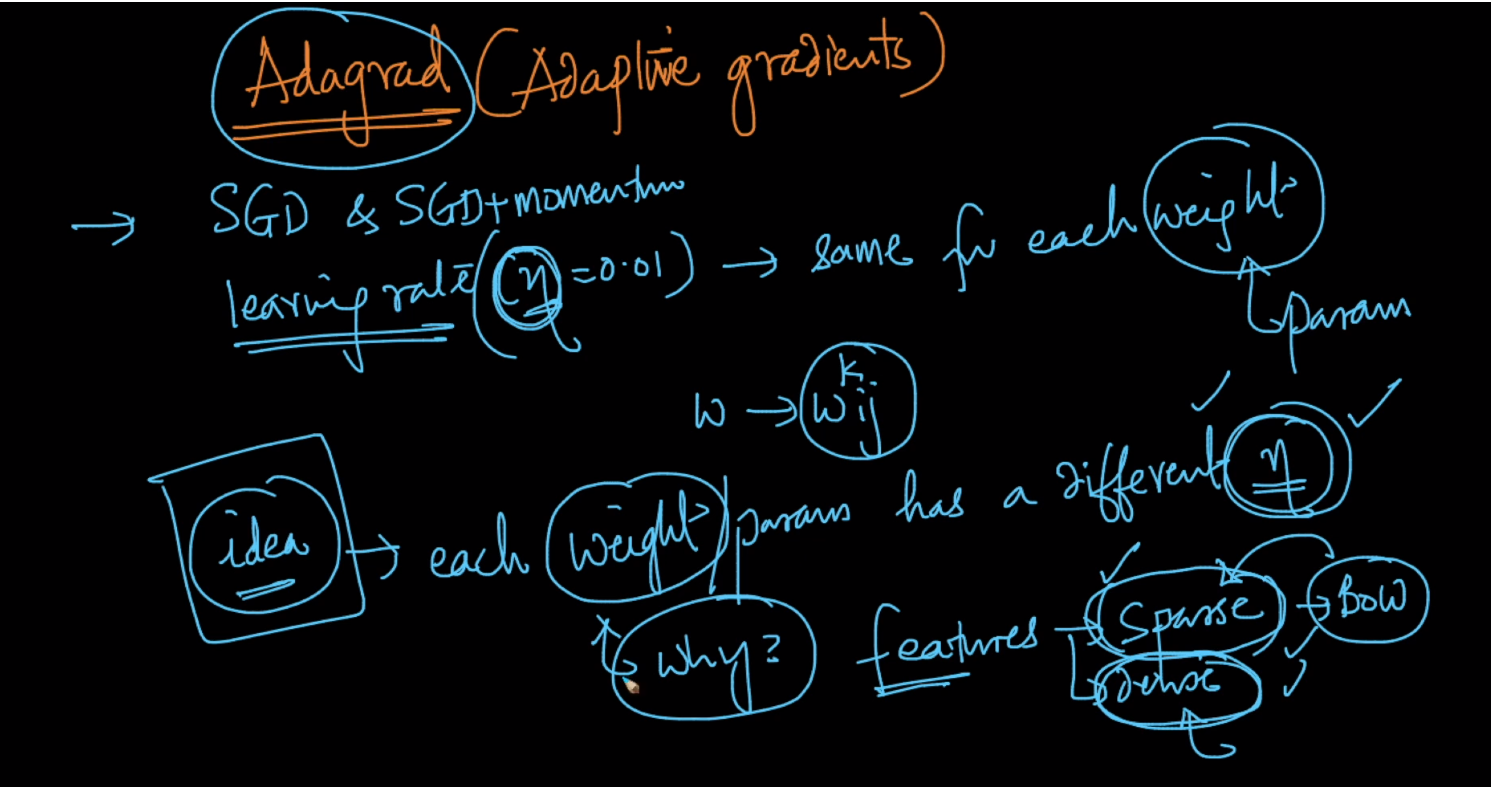
**Optimizers:AdaGrad**

In Sgd and sgd + momentum we use same learning rate eta most likely 0.01 which is same for each weight/param.

Now so in Adagrad we use different learning rate or adaptive learning rate eta for each weight/param.

Why we use this suppose our feature has sparse and dense and now it is not good to use same learning rate for each weight.

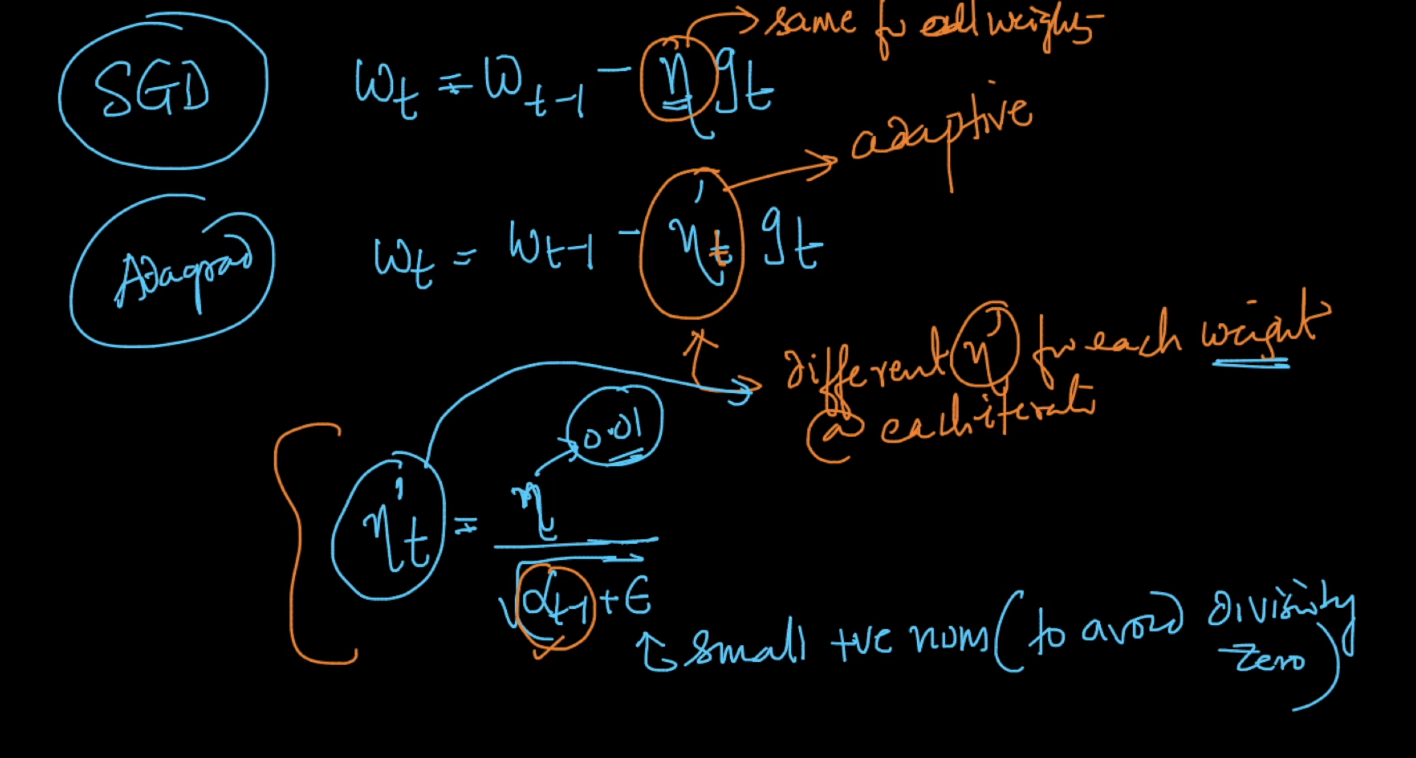


Update equation of SGD and Adagrad is shown below both equation is same except in adagrad we use different learning rate eta’ which is adaptive learning rate i.e there is diff. eta’ for each weight at each iteration.

Formulae for eta’ is shown below

Eta’ = eta / sqrt(alphat-1 + epsilon)

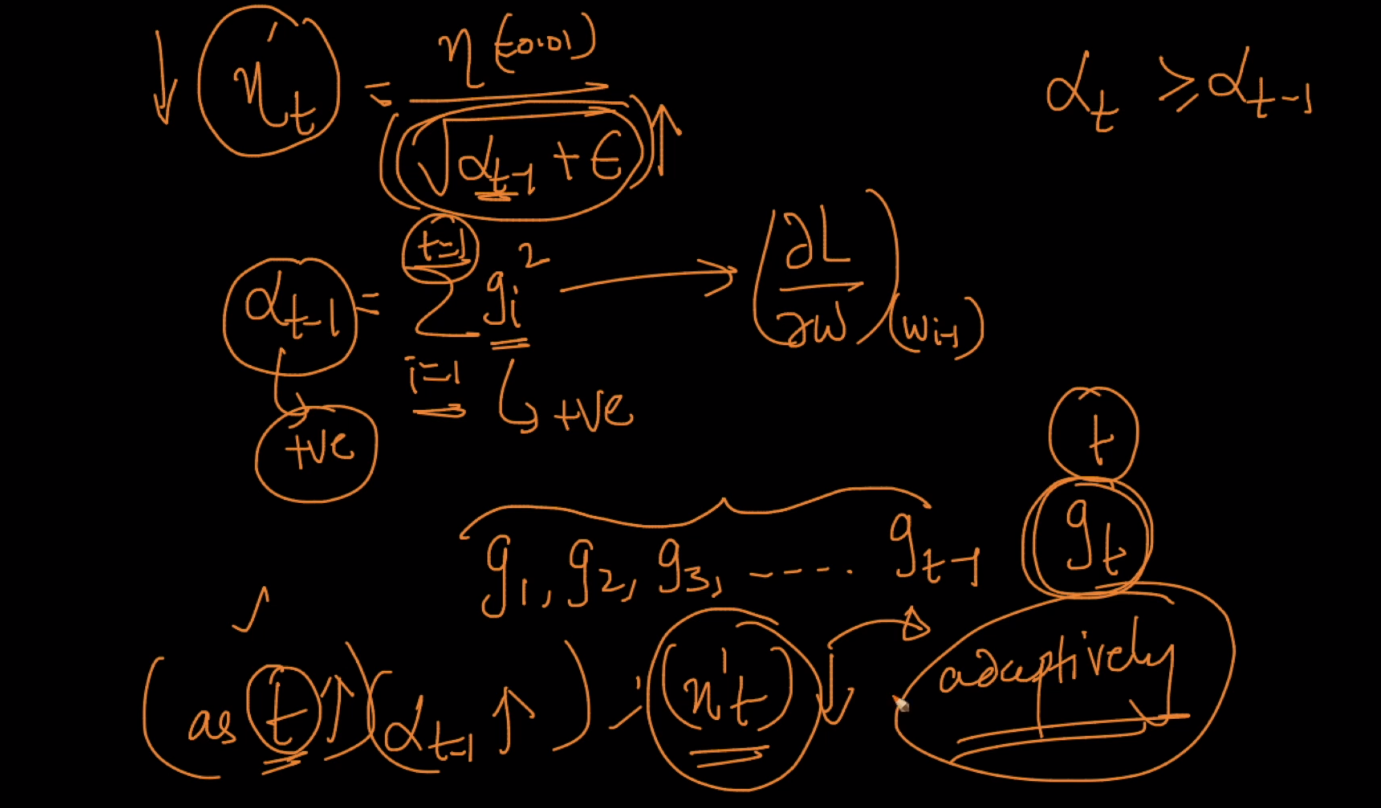
In above formula eta is constant typically 0.01 and in deom we add epsilon this is also constant small value which is use to avoid division by 0



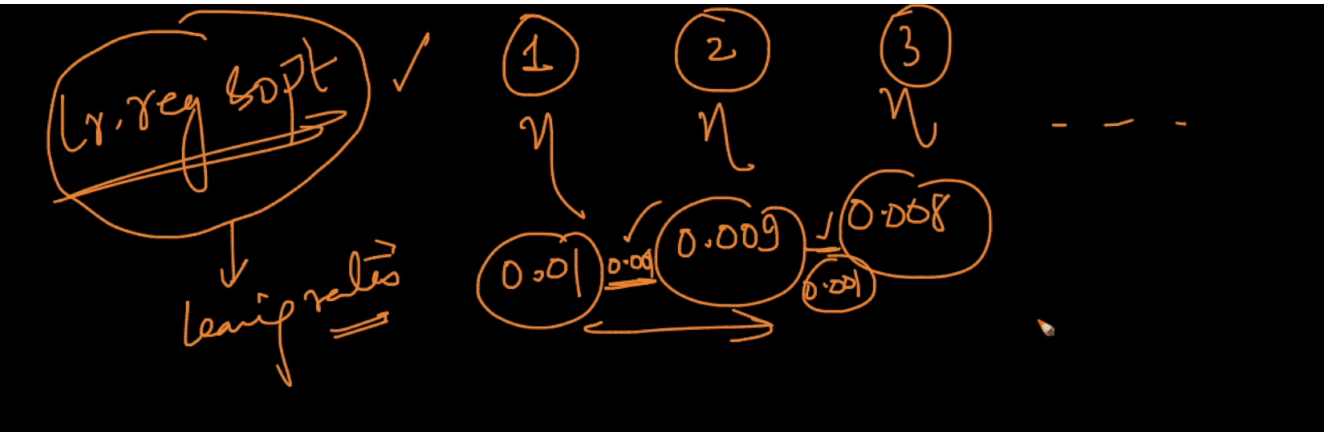
Formulae for alpha\_t-1 = sum\_i=1\_t-1 gi2 here gi is derivative term and it is always positive because we take square of gi .

Value of alpha\_t-1 is decreasing at each iteration because it is adding all previous gradients, therefore alpha\_t >= alpha\_t-1

Therefore as t increases, value of alpha\_t-1 increases and therefore eta’ decreases adaptively



Therefore adagrad resolves the problem of time decay learning rate we use in linear reg. in which we reduce eta at each iteration by same constant.



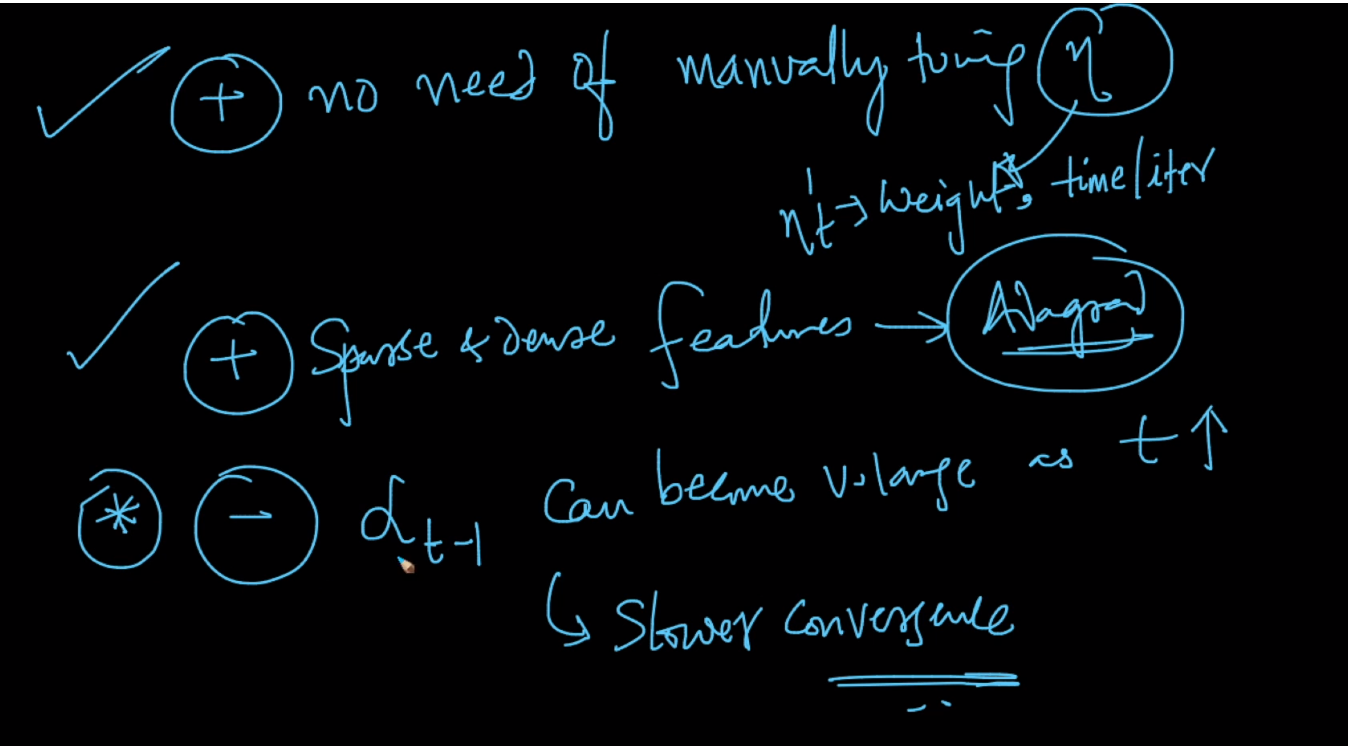
Advantages :

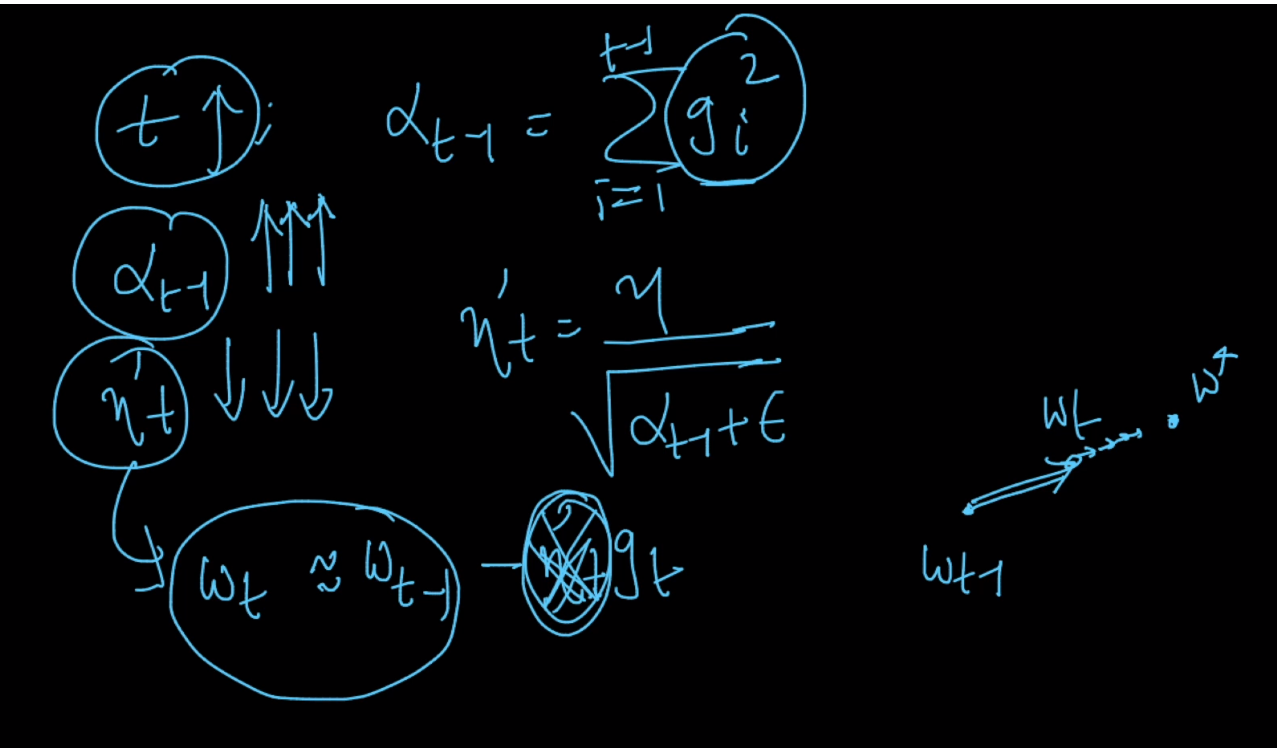
1. There is no need of manually tuning of eta because we diff eta’ for each weight and time/iter.
2. For sparse & dense features

Disadvantage :

1. Major disadvantage is as alpha\_t-1 can become very large as t increases and by this eta decreases very large i.e it comes closer to 0 and thus while updating new wt becomes approxy equals to old wt-1 therefore it moves very slowly towards w\* i.e slow convergence

Why alpha decreases so much because we are taking squares of gradient from 1 to t-1





Comments :

