Hyperparameters in the optimization routines and their notation

In this PDF, we discuss the hyperparameters that are present in the different optimizers.

All the routines are iterative in nature. Further, they have tolerance `tol` (i.e. the difference between the values of successive objective is less than this value) and maximum number of iterations `maxiter` as common hyperparameters for the terminating the algorithm.

The default values of `tol` and `maxiter` are 1e-5 and 100 respectively.

Gradient Descent with Momentum

Let W_i be the parameters of the model at the i-th iteration and ∇L be the gradient of the objective.

$$V_t = \mu V_{t-1} - \alpha \nabla L(W_{t-1})$$
$$W_t = W_{t-1} + V_t$$

Here, $\mu \in [0, 1)$ is the `mass` and $\alpha > 0$ is the `learning_rate`.

The default values of `mass` and α are 0.9 and 0.05 respectively.

RMSProp

Let W_i be the parameters of the model at the i-th iteration and ∇L be the gradient of the objective.

$$R_t = \gamma R_{t-1} + (1-\gamma)\nabla L_t\,(W_{t-1})^2$$

$$W_t = W_{t-1} - \alpha \frac{\nabla L_t\,(W_{t-1})}{\sqrt{R_t + \epsilon}}$$

Here, $\gamma \in [0, 1)$ is the `gamma` which the exponential weighting factor (smaller the value, more emphasis on recent weights), and $\alpha > 0$ is the `learning_rate`.

Adam

Let W_i be the parameters of the model at the i-th iteration and ∇L be the gradient of the objective.

$M_0 = \mathbf{0}, R_0 = \mathbf{0}$ (Initialization)

For $t = 1, \ldots, T$:

$M_t = \beta_1 M_{t-1} + (1 - \beta_1) \nabla L_t (W_{t-1})$ (1st moment estimate)

$R_t = \beta_2 R_{t-1} + (1 - \beta_2) \nabla L_t (W_{t-1})^2$ (2nd moment estimate)

$\hat{M}_t = M_t / \left(1 - (\beta_1)^t\right)$ (1st moment bias correction)

$\hat{R}_t = R_t / \left(1 - (\beta_2)^t\right)$ (2*nd* moment bias correction)

$$W_t = W_{t-1} - \alpha \frac{\hat{M}_t}{\sqrt{\hat{R}_t + \epsilon}} \quad \text{(Update)}$$

Here, $\beta_1 \in [0, 1)$ is 1st moment decay rate, $\beta_2 \in [0, 1)$ is 2nd moment decay rate, and $\alpha > 0$ is the `learning_rate`.

Newtons CG

Newtons CG is an iterative, inexact Newton method which uses Conjugate Gradient method to compute the search direction. The details are given on Pg 168 Nocedal, J, and S J Wright. 2006, Numerical Optimization, Springer New York.

From the perspective of end user, one needs to only specify the tolerance `tol` condition (i.e. the difference between the values of successive objective is less than this value) or maximum iterations `maxiter`.