Why we multiply by batch size in the gradient hook of EKFAC

Vinay Ramasesh

03/26/19

Whatever loss function we take, for a given mini-batch the loss function is of the form

$$L(\theta) = \frac{1}{N_{batch}} \sum_i L_i(\theta), \tag{1}$$

where i indexes the data points in each mini-batch, and L_i is the contribution to the loss of the i^{th} data point.

When we use automatic differentiation to take the derivative of the loss function with respect to the parameter vector θ, what we are computing is $\frac{dL}{d\theta}$. This is fine if we're already averaging over the mini-batch, but if we want to store the individual loss-function values for each data point to compute later (and possibly average after further computation), what we need to store, for each data point, is $\frac{dL_i}{d\theta}$. Since we don't have access to this directly, i.e. the gradient which is passed through backpropagation comes from the total loss function $L(\theta)$ rather than $L_i(\theta)$, when we save this gradient for future computation, we must multiply by N_{batch} so we are saving $\nabla L_i(\theta)$, and not $\nabla L_i(\theta)/N_{batch}$.