Week 4 Tutorial Tasks

Sarat Moka

Theoretical Task: Adam's Effective Learning Rate

From lecture notes, recall that the Adam algorithm update rule, without biases corrections, is as follows: Starting with an initial point $\theta^{(0)}$ and with $v^{(0)} = s^{(0)} = 0$,

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \frac{1}{\sqrt{s^{(t+1)}} + \epsilon} v^{(t+1)}, \quad \text{for } t = 1, 2, \dots,$$

where

$$v^{(t+1)} = \beta v^{(t)} + (1-\beta)\nabla C(\theta^{(t)})$$
$$s^{(t+1)} = \gamma s^{(t)} + (1-\gamma)\left(\nabla C(\theta^{(t)}) \odot \nabla C(\theta^{(t)})\right),$$

where $\odot$ denotes the Hadamard or elementwise product between the vectors of the same size.

In this setting, consider a stationary gradients case: $\nabla C(\theta^{(t)}) \approx g$ for all t for a fixed vector g. Then,

1. Prove that for sufficiently large t the *effective* learning rate vector is

$$\alpha_{\texttt{eff}} \approx \alpha \frac{1}{|g|},$$

   where the inverse and $|\cdot|$ are applied elementwise.

2. Based on the above expression, how gradient elements effect the learning rate of each element of θ?

Coding Task: Adam vs GD Learning Rate Sensitivity

Compare optimization paths of Adam and the basic gradient descent with different learning rates on following Rosenbrock function:

$$f(x, y) = (1-x)^2 + 10(y - x^2)^2,$$

which has a unique global minimum at (1,1). Take the initial point $(x^{(0)}, y^{(0)}) = (-1.5, 2.5)$ and vary the learning rate parameter α over $[0.5, 0.1, 0.05, 0.01, 0.005]$.

In Adam implementation, use the default parameter values provided in Remark 4.3.2 in the lecture notes.

Solution to Theoretical Task

Under stationary gradients, since $v^{(0)} = s^{(0)} = 0$, at t

$$v^{(1)} = (1-\beta)g, \quad \text{and} \quad s^{(1)} = (1-\gamma)(g \odot g).$$

Thus,

$$v^{(2)} = (1-\beta)\beta g + (1-\beta)g = (1-\beta)(1+\beta)g,$$

and

$$s^{(2)} = (1-\gamma)\gamma(g \odot g) + (1-\gamma)(g \odot g) = (1-\gamma)(1+\gamma)(g \odot g).$$

Using recursion, we have

$$v^{(t+1)} = (1-\beta)\left(\sum_{\tau=1}^{t} \beta^{\tau-1}\right) g, \quad \text{and} \quad s^{(t+1)} = (1-\gamma)\left(\sum_{\tau=1}^{t} \gamma^{\tau-1}\right)(g \odot g).$$

For large t, we have $\sum_{\tau=1}^{t} \beta^{\tau-1} \approx 1/(1-\beta)$ and $\sum_{\tau=1}^{t} \gamma^{\tau-1} \approx 1/(1-\gamma)$. Thus,

$$v^{(t+1)} \approx g, \quad \text{and} \quad s^{(t+1)} \approx g \odot g.$$

By update of θ, for large t,

$$\begin{aligned}\theta^{(t+1)} &= \theta^{(t)} - \alpha \frac{1}{\sqrt{s^{(t+1)}} + \epsilon} v^{(t+1)} \\ &\approx \theta^{(t)} - \frac{\alpha}{|g| + \epsilon} g \\ &\approx \theta^{(t)} - \frac{\alpha}{|g|} g,\end{aligned}$$

which concludes that the effective learning rate is $\alpha/|g|$.

Gradient Magnitude Impact: Adam's learning rate adapts per parameter inversely to gradient magnitude:

- **Large $|g_i|$:** Small effective learning rate ($\alpha/|g_i|$) prevents overshooting
- **Small $|g_i|$:** Large effective learning rate ($\alpha/|g_i|$) accelerates progress
- **Invariance:** Update magnitude becomes $\alpha \frac{g_i}{|g_i|} = \alpha \, \mathrm{sign}(g_i)$
  $\Rightarrow$ Direction preserved, magnitude normalized

Solution to Coding Task

Check the solution jupyter-notebook.