Prof. Ryan Cotterell

Johanna Ribas: Assignment 1

jribas@student.ethz.ch, 22-944-003.

04/11/2022 - 10:50h

Question 1: Efficiently Computing the Hessian

a) If we rewrite the Hessian in the two following steps:

$$\nabla^2 f(\boldsymbol{x}) \overset{(1)}{=} \begin{bmatrix} \boldsymbol{e_1} \nabla^2 f(\boldsymbol{x}) \\ \vdots \\ \boldsymbol{e_n} \nabla^2 f(\boldsymbol{x}) \end{bmatrix} \overset{(2)}{=} \begin{bmatrix} \nabla(\boldsymbol{e_1^T} \nabla f(\boldsymbol{x}))^T \\ \vdots \\ \nabla(\boldsymbol{e_n^T} \nabla f(\boldsymbol{x}))^T \end{bmatrix}$$

For (1), I decompose the matrix in its components. Multiplying the basis vectors by each component gives the value of each row. For (2), I use the linearity of the derivative.

b) The hessian is symetric because f is twice continuously differentiable. Thus, the expression of the i_{th} column is: $\nabla(\boldsymbol{e_n^T} \nabla f(\boldsymbol{x}))$

c) Using the results in a), to compute $\nabla^2 f(\boldsymbol{x})$ we need to first compute $\nabla f(\boldsymbol{x})$. We can calculate this gradient using the backpropagation algorithm and then use it again to calculate $\nabla(\nabla f(\boldsymbol{x}))$. The runtime will be O(M) because the derivative of a funcion that runs in $O(M)$ will also run in $O(M)$. Likewise, the number of edges of an addition graph is bounded by 2M and so the runtime over this new graph will also be $O(M)$.

With the more naive algorithm that uses Bauer's Formula the run time will be exponential.

d) To compute the k-th order derivative, we will apply the same methodology k-1 times. The computation graph has m edges and performs in $O(M)$ time and we will do the backprop k-1 times to obtain the k-th order derivative. Then the run time is $O(M(k-1)) = O(M)$.

Link to backpropagation colab notebook:
`https://colab.research.google.com/drive/1cIzc5A1gZp0bUy04EGbPCGFthRKClVBK?usp=sharing`