DL: Assignment 1

Jonathan Mitnik
MSc Artificial Intelligence
University of Amsterdam
`jonathan@student.uva.nl`

Abstract

In this report, the main focus is to go through the assignment, starting with a rough Numpy implementation, and then using State-of-the-art PyTorch.

1 Derivatives

Test123asdasdasdasdasd asdasdasd

2 PyTorch Implementation

Test123asdasdasdasdasd asdasdasd

3 Layer normalization

3.2a: Manual implementation of backward pass

I will start by writing down, in essence, all known shapes.

1. $\boldsymbol{X} \in \mathbb{R}^{S*M}$
2. $\boldsymbol{Y} \in \mathbb{R}^{S*M}$ (same shape as X)
3. $L \in \mathbb{R}^1$
4. $\gamma \in \mathbb{R}^M$
5. $\beta \in \mathbb{R}^M$
6. $\frac{\delta L}{\delta \gamma} \in \mathbb{R}^M$
7. $\frac{\delta L}{\delta \beta} \in \mathbb{R}^M$
8. $\frac{\delta L}{\delta \boldsymbol{Y}} \in \mathbb{R}^{S*M}$
9. $\frac{\delta L}{\delta \boldsymbol{X}} \in \mathbb{R}^{S*M}$

To start, the first derivative we calculate is that of $\frac{\delta L}{\delta \gamma}$.

$$
\frac{\delta L}{\delta \gamma}: \boxed{\begin{aligned}
&\frac{\delta L}{\delta \gamma} => [\frac{\delta L}{\delta \gamma}]_i = \frac{\delta L}{\delta \gamma}_i = \sum_{sj} \frac{\delta L}{\delta Y_{sj}} * \frac{\delta Y_{sj}}{\delta \gamma_i} \\
&\quad = \sum_{sj} \frac{\delta L}{\delta Y_{sj}} * \frac{\delta \gamma_j \hat{X}_{sj}}{\delta \gamma_i} \\
&\quad = \sum_{sj} \frac{\delta L}{\delta Y_{sj}} * \delta_{ji} \hat{X}_{sj} \\
&\quad = \sum_{s} \frac{\delta L}{\delta Y_{si}} * \hat{X}_{si} \\
&\quad = \sum_{s} \mathbf{1}_s * \frac{\delta L}{\delta Y_{si}} * \hat{X}_{si} \\
&\quad => \mathbf{1}^T * [\frac{\delta L}{\delta Y} \circ \hat{X}] \qquad \text{where } \mathbf{1} \text{ is a 1xS vector}
\end{aligned}}
$$

In a similar way we can calculate the derivative with respect to β.

$$
\frac{\delta L}{\delta \beta}: \boxed{\begin{aligned}
&\frac{\delta L}{\delta \beta} => [\frac{\delta L}{\delta \beta}]_i = \frac{\delta L}{\delta \beta}_i = \sum_{sj} \frac{\delta L}{\delta Y_{sj}} * \frac{\delta Y_{sj}}{\delta \beta_i} \\
&\quad = \sum_{sj} \frac{\delta L}{\delta Y_{sj}} * \frac{\delta \gamma_j \hat{X}_{sj} + \beta_J}{\delta \beta_i} \\
&\quad = \sum_{sj} \frac{\delta L}{\delta Y_{sj}} * \delta_{ji} \\
&\quad = \sum_{s} \frac{\delta L}{\delta Y_{si}} \\
&\quad = \mathbf{1}^T * \frac{\delta L}{\delta Y} \qquad \text{where } \mathbf{1} \text{ is a 1xS vector}
\end{aligned}}
$$

Now on to calculating the derivative with respect to X. It would be good to first define the starting chain:

$$
\frac{\delta L}{\delta \boldsymbol{X}} => \frac{\delta L}{\delta X_{ri}} = \sum_{sj} \frac{\delta L}{\delta Y_{sj}} * \frac{\delta Y_{sj}}{\delta X_{ri}} \tag{1}
$$

When focusing on $\frac{\delta Y_{sj}}{\delta X_{ri}}$, there are a number of aspects that come to play. To start, we can break it down into a chain, where we explicitly focus on the "contribution" that μ and σ have on X.

$$
\frac{\delta Y_{sj}}{\delta X_{ri}} = \frac{\delta Y_{sj}}{\delta \hat{X}_{sj}} * \frac{\delta \hat{X}_{sj}}{\delta X_{ri}} + \frac{\delta Y_{sj}}{\delta \mu_j} * \frac{\delta \mu_j}{\delta X_{ri}} + \frac{\delta Y_{sj}}{\delta \sigma_j^2} * \frac{\delta \sigma_j^2}{\delta X_{ri}} \tag{2}
$$

This contains quite a number of steps