Training Truncated SVD Using LoRA

1 Overview of SVD and LoRA

In standard Singular Value Decomposition (SVD), the weight matrix W is decomposed as follows:

$$W = U\Sigma V^T$$

where:

- $U \in \mathbb{R}^{D\times r}$: Left singular vectors,
- $\Sigma \in \mathbb{R}^{r\times r}$: Singular values (diagonal matrix),
- $V \in \mathbb{R}^{K\times r}$: Right singular vectors.

In the Low-Rank Adaptation (LoRA) technique, the weight matrix W is updated by introducing two low-rank matrices X and Y:

$$W_{\text{LoRA}} = W + XY$$

where:

- $X \in \mathbb{R}^{D\times r'}$ and $Y \in \mathbb{R}^{r'\times K}$ are learnable low-rank matrices with $r' \ll \min(D, K)$.

2 Motivation

In SVD , we can think of Singular values to represent the "strength" or "importance" of corresponding basis vectors in the U and V matrices of SVD. My intuition for this is driven by the usage of SVD to compress data, for example Image. For example , we can represent an image as a combination of singular vectors, where the contribution of each vector is determined by its corresponding singular value. Through selection of significant singular values, we can effectively compress the image.
Therefore , to get a trade-off between the full rank computation and the number of trainable parameters we have we can do it in two steps:

1. Truncate the singular values obtained from the SVD decomposition , by retaining only the most significant singular values (which in our case keeping only the most important features with respect to a certain weight matrix).
2. Introducing learnable low-rank matrices which adds more trainable parameters to our model.

Moreover, we need to introduce a new hyper-parameter which control the amount of trade-off between giving up the full-rank computation offered by SVD and use truncated SVD and introducing more trainable parameters using Low rank matrices.

3 Truncated SVD with LoRA

Our goal is to fine-tune the model using truncated SVD and apply a LoRA-like low-rank update to the decomposition. First, we perform truncated SVD on the weight matrix W, keeping the top r singular values:

$$W \approx U_r\Sigma_r V_r^T$$

where $U_r \in \mathbb{R}^{D\times r}$, $\Sigma_r \in \mathbb{R}^{r\times r}$, and $V_r \in \mathbb{R}^{K\times r}$.
We then apply a low-rank adaptation similar to LoRA by adding two learnable low-rank matrices X and Y, resulting in:

$$W_{\text{SVD+LoRA}} = U_r(\Sigma_r + XY)V_r^T$$

where $X \in \mathbb{R}^{D\times r'}$ and $Y \in \mathbb{R}^{r'\times K}$ are trained during fine-tuning. This allows the model to adjust without reconstructing the entire weight matrix.

4 Parameter Efficiency and flexibility

By using truncated SVD and LoRA-like updates, we limit the number of parameters involved in fine-tuning:

- SVD fine-tuning: We fine-tune the singular values in Σ_r.
- LoRA-like update: We introduce low-rank matrices X and Y to capture additional transformations.

Moreover , we introduce some flexibility in both retaining the top r singular values and adding more trainable parameters (r') to fit the data.

5 Trade-off Between Rank and Trainable Parameters

By adjusting the rank of truncated SVD (r) and the rank of the LoRA matrices (r'), we can control the trade-off between model capacity and the number of trainable parameters. Specifically:

- A higher SVD rank (r) retains more singular values, preserving more information from the original matrix.
- A higher LoRA rank (r') introduces more parameters for the fine-tuning process, increasing the model's expressiveness.

The final number of trainable parameters is determined by both r and r', providing flexibility in model adaptation without excessive computational costs.