ALBERT

30 December 2023 09:12

ALBERT (A Little BERT) - https://arxiv.org/pdf/1909.11942.pdf

Increasing model size often leads to better performance on downstream tasks. After some point it gets hard due to the presence of the GPU/TPU memory limitations and longer training times

- ALBERT uses two parameter reduction techniques that help overcome major obstacles in scaling pre-trained models

    - #1 : Vocabulary embedding matrix is split into two small matrices. We separate the size of the hidden layers from the size of the vocabulary embeddings.
        - Total number of parameters = V*H ,V - vocabulary size, H - embedding dimension (which is typically equal to the hidden layer size)
        - This separation makes it easier to grow the hidden size without significantly increasing the parameter size of the vocabulary embeddings

    - #2 : Cross layer parameter sharing
        - This technique prevents the parameters from growing with the depth of the network

- Both the above mentioned techniques reduce the number of parameters for BERT without significantly hurting the performance thus improving parameter efficiency
- ALBERT configuration similar to BERT large has 18 times fewer parameters and can be trained about 1.7 times faster
- Parameter reduction techniques also act as a form of regularization that stabilizes the training and helps with generalization .

- Introduces a Self-supervised loss for sentence order prediction (SOP). This performs better than Next sentence prediction (NSP) loss proposed in the original BERT.

E - the number of dimensions in the embedding space used to represent words or tokens
L - The depth of the Transformer encoder architecture, reflecting the number of processing layers through which information flows.
H - The number of neurons in each hidden layer of the model. (Dictates model's ability to represent and process information at each layer.

ELEMENTS OF ALBERT

- It follows the BERT notation conventions and denote the vocabulary embedding size as E, the number of encoder layers as L, and the hidden size as H. Following Devlin et al. (2019), we set the feed-forward/filter size to be **4H** and the number of attention heads to be **H/64**. There are three main contributions that ALBERT makes over the design choices of BERT.
    - Within each layer of the transformer architecture there's a feed forward network that processes the information after the attention mechanism.
    - The attention head is the allows the model to focus on relevant parts of the input text dynamically

- In BERT and subsequent improvements, the word-piece embeddings size E is tied with the hidden layer size H, This is sub-optimal.
- From a modelling perspective, WordPiece embeddings are meant to learn context-independent representations, whereas hidden-layer embeddings are meant to learn context-dependent representations.
- De -coupling H and E allows for a smaller E while maintaining a larger H, which ensures a more efficient usage of the model parameters. (H>>E)
- ALBERT uses a factorization techniques. Decomposes them into two smaller matrices O(V x H) - O(V x E + E x H)
- Cross Layer Parameter Sharing
    - Ways to improve parameter efficiency
    - Sharing the same set of parameters and saving on a number of parameters to be trained for therefore smaller size and early training with low inference time
    - Networks with cross-layer parameter sharing get better performance on the language modelling and sub-verb agreement than standard transformers