Huggingface Project Proposal: BEIT

This document has two sections. First, I describe the proposed project. Second, I have taken notes on the BEIT paper that will be useful for the project.

Proposal

For this project, I propose to replicate and extend the recent BEIT paper. Broadly, I am interested in generative representation learning for computer vision. The first and most important part of the project will be reproducing the results of BEIT on the tasks of image classification and semantic segmentation. This entails pre-training on ImageNet for 800 epochs and then finetuning on ImageNet/ADE20K for 100 epochs.

The BEIT architecture is straightforward: it is a ViT that predicts visual tokens given by a VQ-VAE. I will use the Huggingface transformers ViT implementation and the VQ-VAE from DALL-E (or possibly the VQ-GAN from the Taming Transformers paper, we shall see). I have a lot of experience with these types of models, but I do not have much experience with Jax/Flax, so that should be a good learning experience.

The BEIT training procedure, data loading, and image augmentations are similar to those used in popular vision transformer papers (ViT, DeiT, etc.). All hyperparameters are listed in the appendix of the paper.

After reproducing the BEIT results, I have a number of ideas for extending the architecture. First, I'd like to explore ways of combining contrastive self-supervised objectives (e.g. SimCLR, MoCo, DINO, etc.) with BEIT. Appendix D in the BEIT paper explored one way of combining BEIT with DINO, but I think there could be better ways of combining the masked language model and contrastive objectives.

Paper Notes

BEIT: BERT Pre-Training of Image Transformers

- Arxiv
- Code (not yet released)
- Summary
    - This paper trains a ViT using a masked language modeling objective. The interesting difference from prior approaches is that the input takes the form of a raw image and the output takes the form of visual tokens.
- Introduction
    - *After pre-training BEIT, we directly fine-tune the model parameters on downstream tasks by appending task layers upon the pretrained encoder.*
        - This is the usual NLP setup; nothing new here.
    - *For example, base-size BEIT achieves 83.2% top-1 accuracy on ImageNet-1K, significantly outperforming from-scratch DeiT training*
        - Strong performance (although we shall see what tricks they use)

    - *A straightforward alternative is regarding the task as a regression problem, which predicts the raw pixels of masked patches. However, such pixel-level recovery task tends to waste modeling capability on pre-training short-range dependencies and high-frequency details*
        - My experience training MLMs for vision with MSE loss matches this
    - *We perform self-supervised learning and then fine-tune the pretrained BEIT on two downstream tasks, i.e., image classification, and semantic segmentation.*
        - It's good to see that they also do a dense task (segmentation); it makes sense that their architecture would be well-suited to this, as it has more spatial information than most self-supervised models (e.g. MoCo).
    - *Performance of BEIT can be further improved by intermediate fine-tuning with ImageNet labels.*
        - No surprise here.
- Methods
    - *We directly use the publicly available image tokenzier described in [CLIP].*
    - *Section 2.3 Pre-Training BEIT: Masked Image Modeling*
        - They randomly mask 40% of patches and replace the masked patches with a learnable embedding.
        - They choose patches in a blockwise manner. Each block contains at least 16 patches. To get the blocks, they (1) randomly sample the total number of patches in the block, and (2) randomly choose an aspect ratio for the patches. They continue this until they have masked enough patches (75 patches; 40%).
    - *2.5 Pre-Training Setup*
        - They use all the standard supervised learning augmentations
        - *The pre-training runs for about 500k steps (i.e., 800 epochs) with 2k batch size. Adam (Kingma and Ba, 2015) with β1 = 0.9, β2 = 0.999 is employed for optimization. The learning rate is set to 1.5e-3, with a warmup of 10 epochs, and cosine learning rate decay. The weight decay is 0.05. We employ stochastic depth (Huang et al., 2016) with a 0.1 rate, and disable dropout. The 500k training steps take about five days using 16 Nvidia Telsa V100 32GB GPU cards.*
            - 800 epochs is a lot! Much more than the usual 300 for supervised training. Do they compare to DeiT at 300? If so, that seems unfair.
            - 80 GPU-days is quite a bit.
    - *2.6 Fine-Tuning BEIT on Downstream Vision Tasks*
        - They then train on labeled data. From Figure 2, it looks like they finetune for 100 epochs. So many epochs for finetuning! They could easily train an entire NF-Net from scratch in that number of epochs.
        - For image classification, they use average pooling and then a linear layer, rather than the [CLS] token
        - For semantic segmentation, they use the setup from SETR-PUP (not described but pretty straightforward)
- Experiments
    - *[Table 1, Table 2]*

        - The ImageNet results are strong. Performance on ImageNet improves with a larger model trained on higher-resolution inputs, in contrast to a model trained with supervision.
        - Note: I wonder whether SAM would help here or not. That's something to explore
    - *[Table 3] and Section 3.2 Semantic Segmentation*
        - They "use the same task layer of SETR-PUP… for a fair comparison" but they train for 3x the number of steps.
        - They are noticeably better than DINO for semantic segmentation
        - Intermediate Fine-Tuning helps substantially -- no surprise
    - *Ablation Studies*
        - Blockwise masking helps a bit for semantic segmentation, but not a ton
        - Directly regressing pixel values instead of visual tokens is definitely worse (81.0 vs 82.9 on IN and 41.4 vs 44.7 on ADE20K) but it's honestly not as bad as I expected. It might be a viable strategy if a better loss were used.
            - Idea: What if you took the patches, did PCA, and then tried to predict that instead. Maybe you could get some more "semantic" information without the hassle of the tokens.
- Conclusion
    - *We will conduct multi-modal pre-training in a more unified way, using the similar objectives and the shared architecture for texts and images.*
        - This is coming
- Appendix
    - *Hyperparameters*
        - It looks like everything is here, which is nice – this should be replicable
    - *Multi-Task Pre-Training with DINO*
        - Gives a boost in segmentation performance
        - They compare throughput -- DINO is really slow!
        - I think there could be better ways of combining DINO + BEIT