Unimodal Knowledge Distillation

To validate whether unimodal knowledge distillation even works, which is undoubtedly a simpler task than the multimodal knowledge distillation we are trying to develop, we will first conduct experiments on unimodal knowledge distillation. That is, the usual knowledge distillation introduced in section (TODO: cite kd).

Vision

Method

Our approach to vision KD involves using BEiTv2, self-supervised pretrained on ImageNet-1K [1], [2], as the teacher model and training a shallow version of the vision variant from Data2Vec2 [3] as the student model. This approach might be seen as unusual by some, as a more intuitive choice would be to construct a small BEiTv2 variant for the student model, and then training it to mimic the large model. DistilBERT [4] for example select half of the layers from a pretrained BERT [5] model, and organizes them into a smaller model, which is then trained to replicate the output of the pretrained BERT. However, we use layers of a different pretrained model and organize them into our student, as this step will be inevitable in the multimodal case. This is becuase the multimodal student model will later not only be far more complex, but its multimodal nature also requires a different architecture than the teacher. We believe that taking a similar approach here will give us valuable insights about the feasibility (of multimodal KD) on a comparatively simple exampe, and provide us with a foundation on which we can build our approach to multimodal KD.

BEiTv2 is a self-supervised model, and therefore does not provide a probability distribution over classes that can be predicted using KL-Divergence. Instead, we only have access to the model's activations for each layer, so we have to resort to feature-based knowledge distillation. One option would be to predict the teacher's output for the cls token $\boldsymbol{h}^t_{v,L,[\texttt{I\_CLS}]}$, which aggregates the high level content of the image, and then use the mean squared error as the loss function. However, this neglects the activations for individual image patches and activations of intermediate layers.

This argument is quite similar to that of Data2Vec, a general framework for self-supervised pretraining of unimodal image, text and audio models [6]. The authors introduce "contextualized representations", which are the activations of all layers of a model for each time step of the input. Because of Self-Attention in Transformers, the activations for each image patch (time step) are influenced by all other image patches, and therefore not only encode information about a patches content, but also about its context in the image, i.e. the relationship to other patches. Consequently, contextualized representations are more informative than a single cls token, as they encode information about the image at different levels of abstraction, and how the model aggregates low level features to high level concepts. Since the goal of KD is to "mimic" the behavior of a teacher model for a given input in a compressed way, this is the exact information that should be transferred from the teacher to the student. Simply predicting the cls token would only "mimic" what information the teacher extracts from the image, but not how the information is extracted.

While the dimensions of our student model match those of the teacher model, they both have a hidden size of $d = 768$ and intermediate size of $d_{\text{ff}} = 3072$ for the feed-forward layers in the Transformer blocks, the number of layers in the student model ($L_s = 12$) is only half of that of the teacher model ($L_t = 12$). It is therefore not possible for each layer of the student model to mimic the behavior of the corresponding layer in the teacher model. Fortunately, experiments of the Data2Vec authors show that predicting the mean of all layer activations for each time step works as well as predicting the activations of each layer individually [6]. This suits our approach well, as the only mismatch between the teacher and student model is the number of layers, which is irrelevant when predicting the mean of all layer activations for each time step. Additionally, the authors apply

instance normalization to the activations of each layer before averaging, and then perform parameter-less layer normalization, which we perform likewise. The target and prediction are therefore given by:

$$\begin{aligned} {\boldsymbol{H}'}_{v,l}^{t} &= \text{InstanceNorm}\Big(\boldsymbol{H}_{v,l}^{t}\Big), l \in \{1, 2, ..., L_t\} \\ {\boldsymbol{H}'}_{v,l}^{s} &= \text{InstanceNorm}\Big(\boldsymbol{H}_{v,l}^{s}\Big), l \in \{1, 2, ..., L_s\} \end{aligned} \tag{1}$$

$$\begin{aligned} \widehat{\boldsymbol{H}}_{v}^{t} &= \frac{1}{L_t} \sum_{l=1}^{L_t} {\boldsymbol{H}'}_{v,l}^{t} \\ \widehat{\boldsymbol{H}}_{v}^{s} &= \frac{1}{L_s} \sum_{l=1}^{L_s} {\boldsymbol{H}'}_{v,l}^{s} \end{aligned} \tag{2}$$

$$\begin{aligned} \boldsymbol{Y} &= \Big[\boldsymbol{y}_{[\texttt{I\_CLS}]}, \boldsymbol{y}_1, ..., \boldsymbol{y}_N\Big] = \text{LayerNorm}\Big(\widehat{\boldsymbol{H}}_{v}^{t}\Big) \\ \widehat{\boldsymbol{Y}} &= \Big[\hat{\boldsymbol{y}}_{[\texttt{I\_CLS}]}, \hat{\boldsymbol{y}}_1, ..., \hat{\boldsymbol{y}}_N\Big] = \text{LayerNorm}\Big(\widehat{\boldsymbol{H}}_{v}^{s}\Big) \end{aligned} \tag{3}$$

The loss for a single sample (image) is defined in the following:

$$\mathcal{L}_{\text{KD}}\Big(\boldsymbol{Y}, \widehat{\boldsymbol{Y}}\Big) = \|\boldsymbol{Y} - \widehat{\boldsymbol{Y}}\|_2^2 = \frac{1}{N+1}\left(\sum_{n=1}^{M} \mathcal{L}_{\text{MSE}}(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n) + \mathcal{L}_{\text{MSE}}\Big(\boldsymbol{y}_{[\texttt{I\_CLS}]}, \hat{\boldsymbol{y}}_{[\texttt{I\_CLS}]}\Big)\right) \tag{4}$$

We denote $\boldsymbol{y}_i$ and $\hat{\boldsymbol{y}}_i$ as the average representation for image patch i over all layers from the teacher and student model, respectively. This includes instance norm before averaging, and layer norm afterwards. $\text{InstanceNorm}(\cdot)$ and $\text{LayerNorm}(\cdot)$ are defined as specified in (TODO: cite notation), and $\mathcal{L}_{\text{MSE}}(\cdot, \cdot)$ is the mean squared error between two d-dimensional vectors, defined in (TODO: cite equation) in (TODO: cite notation).

Pretraining

Finetuning

Language

Method

Pretraining

Finetuning

Bibliography

[1] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[2] Z. Peng, L. Dong, H. Bao, Q. Ye, and F. Wei, "BEiT v2: Masked Image Modeling with Vector-Quantized Visual Tokenizers," 2022.

[3] A. Baevski, A. Babu, W.-N. Hsu, and M. Auli, "Efficient Self-supervised Learning with Contextualized Target Representations for Vision, Speech and Language." 2022.

[4] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," in *NeurIPS EMC^2 Workshop*, 2019.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[6] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A general framework for self-supervised learning in speech, vision and language," *arXiv abs/2202.03555*, 2022.