EquiBERT: (An Attempt At) Equivariant Fine-Tuning of Pretrained Large Language Models

Stanford CS224N {Default} Project

Patrick Sicurello
Department of Computer Science
Stanford University
psicur@stanford.edu

Abstract

In this project, we first implement and train a small scale BERT model to perform sentiment classification on the Stanford Sentiment Treebank dataset. We extend this model to also perform paraphrase detection and semantic text similarity, using a round robin training method to iterate through batches of all three tasks. We then implement a BERT model that finetunes on the task of generated uniform hypersphere embeddings, with the assumption that this is an underlying symmetry in our data in the spirit of geometric deep learning. We find that directly finetuning BERT on this loss caused it to overall perform worse than baseline, and that likely we would need to introduce another network to perform this transformation of the data.

1 Key Information to include

- Mentor:
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

Data often contain certain equivalence classes of transformations that still preserve the semantic meaning of the transformed data, called symmetries. For example, a rotated image of an apple is still an apple. Training neural networks that respect these symmetries, through making them insensitive to input transformations, can lead to more robust and generally better performing models (Bronstein et al., 2021). We can represent these transformations as equivariant functions on our data, or functions that transform the data in a way that is predictable given knowledge of those symmetries.

However, compared to rotations, symmetries are much more difficult to define on NLP data in a closed-form manner, though they definitely still exist. Word embeddings have intrinsic geometric properties, with distance representing semantic similarity Yin and Shen (2018). Similar geometric embeddings arise through contrastive learning, which forces them to become uniform along a unit hypersphere Wang and Isola (2022).

In lieu of explicitly stating these symmetries to the model, it may be possible to learn them through optimization with an appropriate choice of objective function (Kaba et al., 2023). In this work we consider using explicit metrics of alignment and uniformity as a finetuning task, in order to train our BERT model to output better symmetrized sentence embeddings. We use these finetuned sentence embeddings on three downstream tasks, closely following the ideas of learned canonicalization in Kaba et al. (2023).

3 Related Work

3.1 Equivariance and Canonicalization

When talking about symmetry, it is often the case that you care about the set of all symmetries that behave similarly. This can be formalized as a mathematical group, which can be thought of as an object that respects permutations of elements of a set. Formally, we represent transformations on our data using a group G on a vector space X representing the data itself. The group representation $\rho : G \to GL(X)$ gives us the set of invertible linear transformations on X, since $GL(X)$ is the set of invertible matrices representing the transformations induced by G. And equivariant function is then defined as a function $f : X \to Y$ that behaves according to

$$f(\rho(g) \cdot x) = \rho'(g) \cdot f(x) \tag{1}$$

where ρ' is another group representation that acts on the output Y. The transformation represented by ρ' is induced by G, meaning we know how the data will transform given knowledge of G.

Using the formulation above, our equivariant function $f : X \to Y$ can be written in a decoupled form:

$$f(x) = \rho'(c(x)) \cdot p(\rho(c(x)^{-1} \cdot x) \tag{2}$$

where $c : X \to G$ is called a canonicalization network and $p : X \to Y$ is a prediction network (Mondal et al., 2023). This prediction network can be a network that is difficult to make equivariant, such as a pretrained language model, as it decouples the equivariance requirement to be on the canonicalization network instead.

3.2 Contrastive Learning

Contrastive learning improves sentence embeddings by learning a metric between positive and negative sampled examples (Chen et al., 2020). This metric pushes negative sampled pairs further away in the model's embedding space, and puts positive sampled pairs together. This method was applied to NLP domains by defining positive pairs by generating two sets of embeddings from the same model using, where low probability dropout layers add noise as data augmentation (Gao et al., 2022). This is done through a constrastive loss function on two embeddings h_i, h_i^+:

$$\mathcal{L}(h_i, h_i^+)_{\text{contrast}} = \frac{\exp(sim(h_i, h_i^+)/\tau)}{\sum_{j=1}^{N} \exp(sim(h_i, h_j^-)/\tau)} \tag{3}$$

where $sim(x, y)$ is a similarity metric between x and y and temperature value τ. In this work, we add this loss to our downstream task loss as a weighted regularization function.

This loss asymptotically forces sentence embeddings to become uniform and aligned along a hypersphere in Wang and Isola (2022). For the purpose of our work, we consider this as an appropriate optimized symmetry for a language model.

4 Approach

We implement a method for finetuning BERT to directly output uniform sentence embeddings and compare it to contrastive learning as a baseline. This is in the spirit of a canonicalization network, in that it attempts to learn some relevant geometry in the input data of a downstream task.

4.1 Loss Functions

We train our model on four separate loss functions. First, we use the loss associated to the finetuning task, which we denote as $\mathcal{L}_{task}$. Second, we use the contrastive loss function defined in equation 3. For our similarity metric, we use cosine similarity. Lastly, we use the uniform and alignment metrics defined by

$$\mathcal{L}_{align}(h, h^+) = \mathbb{E}(||h - h^+||_2^2) \tag{4}$$

and

$$\mathcal{L}_{uniform}(h, h^+, t) = \mathbb{E}(e^{-t||h-h^+||_2^2}) \tag{5}$$

We then take a weighted sum of these loss functions in various combinations depending on the model.

4.2 Model

We propose finetuning our pretrained BERT model on the weighted sum of the uniformity and alignment loss functions, and then train our downstream models on these hopefully symmetric embeddings. Figure 1 shows a diagram of this model. Note that the weights of the BERT model are unfrozen during training on the uniform-alignment loss, and frozen when training the feedforward networks on the downstream task. The final loss term can also be substituted to include the contrastive loss term as well.

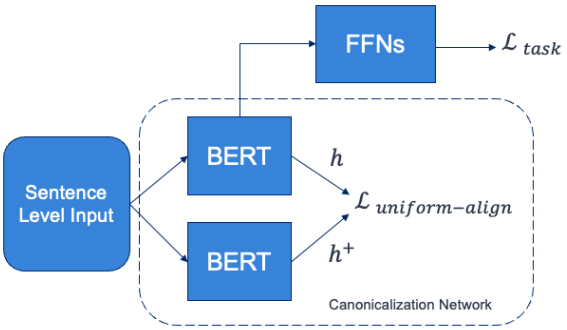

Figure 1: Proposed model architecture. We finetune BERT to the uniform-alignment loss function to train it to output sentence embedings that respect our proposed hypersphere symmetry.

We thus have two models to test against a baseline: one that uses the contrastive loss and another that doesn't.

4.3 Baseline Models

We compare our models against a BERT model finetuned with only the task loss, and a BERT model finetuned with the contrastive loss term. We expect BERT with the constrastive loss to be more comparable to our model.

5 Experiments

5.1 Data

We train and evaluate our models on three different downstream tasks.

5.1.1 Sentiment Classification

One task is multiclass sentiment classification, using the Stanford Sentiment Treebank (Socher et al., 2013). This contains annotated phrases parsed by the Stanford parser. There are 5 output classes total.

5.1.2 Paraphrase Detection

Another task is paraphrase detection – given two sentences, determining if they are paraphrases or not. For this we use the Quora dataset, which consists of sentences for comparison. There are 2 output classes total.

5.1.3 Semantic Text Similarity

The last task is determining semantic equivalence of two sentences using the STS benchmark dataset (Agirre et al., 2013). This is a regression task, where the model attempts to output a continuous value determining how similar pairs of sentences are. This scale goes from 0 to 5, representing unrelated to equivalent, respectively.

5.2 Evaluation method

We evaluate performance on both sentiment classification and paraphrase detection using accuracy, as both are classifcation problems. For semantic text similarity, we use the Pearson correlation of the true similarity values against the prediction similarity values.

5.3 Experimental details

All models were trained with a learning rate of 1×10^{-5}, with 3 training epochs of finetuning on all three downstream tasks. Models with the uniform loss were trained with 5 additional epochs on the SST training data.

Models were trained round robin, with alternating batches between each task dataset. Due to the uneven sizes of the datasets (specifically the Quora dataset being much larger than the others), we iterate multiple times through the smaller datasets and consider an epoch finished once we have finished passing through the largest dataset. In certain experiments, where only the relative performance differences between the models matter, we subsample 128 batches evenly between each task.

5.4 Results

Due to time constraints, we performed a comparison analysis between models through subsampling the data as mentioned above. In the tables below, BERT$_{vanilla}$ is the vanilla BERT multitask model, BERT$_{simCSE}$ is the BERT multitask model with the additional contrastive loss term, BERT$_{unif}$ is BERT trained on the uniform loss function, and BERT$_{simCSEcanon}$ is the same as the previous model with an additional contrastive loss term.

Model	Performance		
	SST-Dev	Para-Dev	STS-Dev
BERT$_{vanilla}$	0.453	0.720	0.342
BERT$_{simCSE}$	0.447	0.712	0.336
BERT$_{unif}$	0.373	0.618	0.049
BERT$_{simCSEunif}$	0.370	0.375	0.199

Table 1: Performance finetuning on small subset of data from all tasks.

These results show that the vanilla model performs the best out of all methods. The performance drop introducing $simCSE$ is possibly due to our choice of treating the loss as a regularization term instead of the actual full loss objective.

It is important to note here that the uniform BERT model is only trained on the SST dataset, while BERT is finetuned on all datasets for the baseline models. To have a realistic comparison, we repeat the same experiment but only finetune the baseline models on the SST dataset. This gives an improved relative performance in our models, but not by much.

As a result, we submit test results using the contrastive loss model. This gave us test results 0.446 on SST, 0.527 on paraphrase, and 0.307 on STS.

6 Analysis

Overall we see that our model performs worse than baseline, even when finetuning on the same dataset. Finetuning all models on the SST dataset does make performance look closer, and the

Model	Performance		
	SST-Dev	Para-Dev	STS-Dev
BERT$_{vanilla}$	0.489	0.517	0.091
BERT$_{simCSE}$	0.505	0.507	0.083
BERT$_{unif}$	0.417	0.618	0.049
BERT$_{simCSEunif}$	0.383	0.618	0.049

Table 2: Performance finetuning on only SST task.

uniform loss model does outperform the baselines for the paraphrase task. However, the model does not expect the way we would expect. Qualitatively, we expect a model that preserves symmetries in sentence embeddings to perform better on the STS benchmark even when training on sentences outside of that dataset. The canonical model fails to do so.

There are two possible reasons for this, the first of which is the scale of the experiments performed. A key assumption with this method is that we can finetune BERT to output embeddings adhering to a hypersphere through the alignment and uniformity losses. However, we only ever finetune BERT on a single dataset for 5 epochs. It is possible that longer finetuning to this task will improve performance, but this may not increase the expressivity of the model and would thus need to be trained on a dataset not in the test set to determine if the results are actually due to the uniformity loss. We also, due to lack of time, were not able to finetune BERT on all task datasets using the uniform-alignment loss, which would likely improve model performance greatly.

Another possible reason for the model's performance is the motivation for decoupling the equivariance requirement from the pretrained model in the first place. If large pretrained models can be regularized to be equivariant, they likely already would be. Instead, this task should be given to a feedforward network or something similar.

7 Conclusion

We found that directly enforcing symmetries through finetuning a pretrained model is not enough to ensure better performance in downstream tasks, likely due to conflicting optimization objectives and lack of sufficient training. Our proposed uniform model, when trained similar to the baselines, outperformed in paraphrase detection but underperformed in the other two tasks. Surprisingly, the model does not do better on the SST dataset, even though it's trained for longer on it. Due to time constraints, we were not able to investigate whether this was the case where our model was finetuning BERT on all tasks with the uniform-alignment loss.

Though this attempt underperforms, we still believe that the connection between the work done by Wang and Isola (2022) and Kaba et al. (2023) is one worth exploring in more detail. If we can use the uniformity-alignment loss on a feedforward network, to act as a symmetry on NLP data up to optimization, then we can have at least a starting point for defined symmetries on NLP data.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. 2021. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. 2023. Equivariance with learned canonicalization functions.

Arnab Kumar Mondal, Siba Smarak Panigrahi, Sékou-Oumar Kaba, Sai Rajeswar, and Siamak Ravanbakhsh. 2023. Equivariant adaptation of large pretrained models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Tongzhou Wang and Phillip Isola. 2022. Understanding contrastive representation learning through alignment and uniformity on the hypersphere.

Zi Yin and Yuanyuan Shen. 2018. On the dimensionality of word embedding.