FinalTerm Project Report

Aakash Patel and Nikita Jat
Department of Computer Science
Indian Institute of Science

January 21, 2021

1 Introduction

1.1 Problem

Given huge amounts of unstructured text data and very few labelled examples, we want to perform classification on data in a semi-supervised manner. For this purpose, we will use two semi-supervised approaches, delta training and layer partitioning.

1.2 Motivation

Text classification can provide us with many useful insights into the data. Many tasks can be performed under text classification like spam filtering, news classification, sentiment analysis, etc. Deep learning-based classification algorithms have performed well in various NLP tasks[6], but the performance is not always satisfactory when utilizing small data. Many times it is necessary to have more data for better performance. Although collecting unlabeled text data is relatively easy, labeling in and of itself requires a considerable amount of human labor. To incorporate unlabeled data into a task, we have to label the data following the task's class policies. Still, the labeling process requires not only human labor but also domain knowledge in the classes.

2 Preliminaries

2.1 Self-training

Given labeled data $\{(x_1; y_1); ..; (x_n; y_n)\}$ and unlabeled data $\{(x_{n+1}); ..; (x_{n+l})\}$, self-training[3] , first builds a model m using labeled data. Next, it simply predicts the unlabeled data using pre-trained model m. If the confidence score of the predicted label is higher than a predefined threshold T, then adds the label-by-prediction data to the training set.

2.2 π model

The input and the prediction function in π-Model are both stochastic, and hence it can produce different outputs for the same input x. π-Model [2] adds a consistency loss which encourages the distance between a network's output for different passes of x through the network to be small. However, the teacher network targets are very unstable and can change rapidly along the training.

2.3 Temporal Ensembling and Mean Teacher

This method proposed to obtain a more stable target by setting the teacher network as an exponential moving average of model parameters from previous training steps, or by directly calculating the moving average of previous targets **ref**.

3 Model

3.1 Δ-Training

This method consists of two classifiers: one is randomly initialized(m_{rand}; random network), and the other is using pre-trained word vectors (m_{emb}; embedded network). For ensembling, we duplicate the same classifier, $M_{rand} = (m_{rand1}; ..; m_{randn}) and M_{emb} = (m_{emb1}; ...; m_{embn})$, respectively.
First, we train the classifiers using the training set with early-stopping, and return their predictions on unlabeled data. We consider the predictions of M_{emb} on the unlabeled data as label-by-prediction since M_{emb} always outperforms M_{rand} according to our hypothesis. After labeling the unlabeled data, we select the data with conditions that

- each ensembled classifiers are predicting the same class.
- the predictions of M_{rand} and M_{emb} are different.

Condition (a) helps to pick out the data labeled with high confidence by the classifiers and Condition (b) helps to pick out the data which is incorrect in M_{rand} but correct in M_{emb}. The ratio in which labels might be correct in M_{rand} but incorrect in M_{emb} is relatively small than vice versa (will be also presented in Section 7). We add the selected data and its pseudo-label by M_{emb} to training set, and then train the classifiers again from the very first step to validate our hypothesis. We denote one such iterative process,training and pseudo-labeling, as a meta-epoch.

3.2 Layer Partitioning

Let M be a neural network model with n layers, we could then split M into two parts U and F, where F contains the lower layers {1;...; l} and U contains the higher layers {l + 1;...;n}. We propose to freeze the layers in F and use them

as a feature extractor. We only update the task-specific layers in U. In fact, assuming we have samples $x_1; :::; x_N$, freezing F is equivalent to training the model U with transformed inputs $F(x_1); : : : ; F(x_N)$.
Each time x passes through F, the output will contain different perturbation. After properly perturbing the discrete input, we can used state-of-the-art semi-supervised learning method to train U. We used π-Model.
Approaching the end of the training, we will gradually unfreeze the layers in F. The motivation is that U has been well trained on {F(x)} and becomes saturated, we can then unfreeze F to let it also learn some specific features of the task domain.
The loss is a weighted sum of the cross-entropy loss(CE) and the consistency loss and is given as below:

$$\text{Loss(x, y)} = \text{CE(z , y)} + \text{w(t).MSE(z , } \hat{z}\text{)}$$

where w(t) is the weight of the consistency loss and is a function of the iteration t.
y is the actual label, z is the predicted label and $\hat{z}$ is the predicted output for the perturbed input.

3.3 Our Idea

In the Layer Partitioning approach, Instead of using π-model, we've used another semi-supervised learning algorithm namely, Temporal Ensembling[1, 5]. The temporal ensembling model is similar to π-Model, except that the "teacher" targets $\hat{z}$ is an ensemble of previous predictions. More formally,

$$Z_i \leftarrow Z_i + (1 - \alpha)z_i \text{ and then,}$$
$$\hat{z}_i = Z_i = (1 - \alpha^{T_i})$$

where T_i is the number of times that x_i has been used. The factor $1 - \alpha^{T_i}$ is to correct the zero initialization bias.

4 Experiments

4.1 Layer Partitioning

Transformer encoder with 16 layers, 410 dimensional embedding, 2100 dimensional hidden layer and 10 heads for each multi-head attention layer. The encoder was pre-trained with a linear language model heading on the WikiText-103 data. We also use the BERT-tokenize where a [CLS] token is appended to each sentence.
Then we simply add a linear classification layer on top of the embedding of the [CLS] token to predict the class.
We freeze the word embedding layer and the first layer of the transformer encoder.

4.2 Δ-Training

5 Datasets

- IMDB : It is a movie review dataset, containing 50,000 movie review with two class labels.
- AGNews: It is News review dataset, containing 127,600 movie review with four class labels.
- TREC-6: This dataset consists of open-domain, fact-based 5952 questions divided into six semantic categories.

References

[1] Samuli Laine and Timo Aila, *"Temporal ensembling for semi-supervised learning," arXiv preprint arXiv:1610.02242, 2016.*

[2] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen, *"Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in Advances in Neural Information Processing Systems, 2016, pp. 1163–1171.*

[3] David Yarowsky. 1995. *Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the 33rd annual meeting on Association for Computational Linguistics, pages 189–196. Association for Computational Linguistics.*

[4] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. *"Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," IEEE Transactions on Neural Networks, vol. 20, no. 3, pp. 542–542, 2009.*

[5] Antti Tarvainen and Harri Valpola. *"Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in Advances in neural information processing systems, 2017, pp. 1195–1204.*

[6] Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. *Word representations: a simple and general method for semi-supervised learning. In Proceedings of the 48th annual meeting of the association for computational linguistics, pages 384–394. Association for Computational Linguistics.*