Interpolation Based Data Augmentation Techniques for NLP

Arijit Ghosh Chowdhury
arijit2@illinois.edu
Department of Computer Science
University of Illinois at Urbana Champaign

ABSTRACT

Models with large number of parameters are prone to over-fitting and often fail to capture the underlying input distribution. Recent works use latent feature interpolations of word embeddings and intermediate layer representations to construct virtual examples for training through a technique called Mixup that has been effective in Computer Vision tasks. This technique has been extended to Natural Language Processing tasks, usually while fine-tuning BERT. It has been demonstrated that the virtual examples thus constructed lead to more robust models with greater generalization capacity. Models trained with mixup perform better on text classification and are robust to noise and sparsification. This report provides a comprehensive survey of recent developments in interpolation based data augmentation techniques for NLP.

ACM Reference Format:
Arijit Ghosh Chowdhury. 2018. Interpolation Based Data Augmentation Techniques for NLP. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Context and Scope: In this report, we deal with the problem of sentence classification. Many state-of-the-art deep learning approaches for sentence classification have millions of parameters and hence require a large number of training examples, which may be time-consuming and expensive to obtain. .

Motivation: We need to be able to train machine learning models with a small amount of data as obtaining large amounts of data may be costly. We need to extend classifiers beyond the distribution of data fed into it. Models should be robust to slight changes in the input distribution. In general, it is difficult to come up with rules for language transformation similar to image transformations; hence universal data augmentation techniques have not thoroughly been explored in NLP yet. One such technique prominently used in computer vision is called *Mixup* - it uses systematic transformations to make sure the model trains on samples from the vicinity distribution [1] along with the original distribution of the training data. Mixup is a data-agnostic data augmentation method that is proven to be effective in Computer Vision tasks by introducing samples from the vicinity distribution [9]. This augmentation framework has been extended to NLP too in recent years.

2 RELATED WORK

Data Augmentation has been vastly used in Computer Vision [5], which employs techniques like flipping, scaling, and rotation. Data augmentation has also been explored in NLP to some extent. [10] replaced words with their synonyms according to a geometric distribution. [7] used k-k-NN and Cosine Similarity metrics to find a similar word for replacement. [8] introduced EDA (Easy Data Augmentation) for NLP which includes *Synonym Replacement, Random Insertion, Random Deletion, and Random Swap*. These methods have the risk of completely changing the context of the sentence with the replacement of a word which alters the entire meaning of the sentence. We create virtual examples by interpolating the latent spaces of two sentences and mix the corresponding labels accordingly to ensure that the interpolated sentence still conforms to the correct label.

Mixup [9], and more recently Manifold Mixup [6] has shown improvements in the accuracy of image classification models in Computer Vision. In NLP, [3] trained a CNN based classifier on various sentiment classification datasets with Input Mixup and found an increase in the generalization capabilities of the model.

3 METHODOLOGY

3.1 Mixup for Sentence Classification

For sentence classification: we first take a sentence $s = \{w_1, w_2, w_3 .... w_n\}$, where w_i is the word embedding of the i_{th} word. We pass s through the BERT base model to obtain feature representation f. We pass f through a linear layer followed by softmax activation to obtain the predicted label y for sentence s ($y = softmax(Linear(f))$). We use Mixup in broadly two ways i) Mixup on the **Word Embeddings Space** (Figure 1) ii) Mixup on the **Intermediate Representations** of BERT (Figure 2).

These two methods can be combined into four different variants of Mixup:

SenMix [3] : In this work, word embeddings of the input sentences (k1 and k2) are interpolated. Let w_i^1 and w_i^2 denote the word embeddings of the i_{th} word in two sentences. The authors interpolate them as - $w_i^{mix} = \lambda w_i^{k1} + (1-\lambda) w_i^{k2}$. The parameter $\lambda \in [0,1]$ is distributed according to a Beta distribution: $\lambda \sim \beta(\alpha, \alpha)$.

TMix [2]: In this, a layer is sampled randomly from the set of BERT layers at each mini-batch and interpolate the features from that layer for both the sentences using the same formulation as in Input Mixup. Formally it is defined as - $w_i^{mix} = \lambda f(w_i)_k^{k1} + (1-\lambda) f(w_i)_k^{k2}$. During backpropagation, the gradients flow through the entire computational graph, including the layers before the Mixup layer k.

Woodstock '18, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/10.1145/1122445.1122456

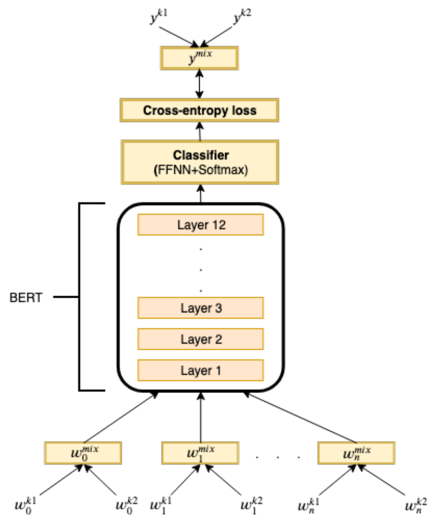

Figure 1: Mixup applied to the word embeddings fed into BERT.

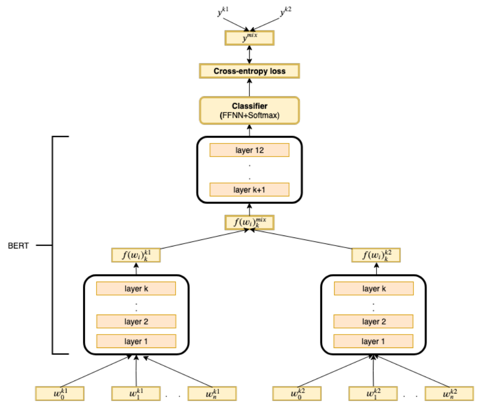

Figure 2: Mixup applied to the intermediate representations from BERT.

If the sampled set of layers, $L = \{0\}$, then manifold Mixup reduces to Input Mixup.

Emix [4] takes the difference of text energies [?] of the samples into consideration analogous to how tokozume2017learning takes into account the energy of the speech samples, in order to make the perception of the mixed sample $x_i : x_j = \lambda : (1 - \lambda)$. We define t using the standard deviation per text sample (σ_i and σ_j) so that the ratio of text energy becomes $x_i : x_j = \lambda : (1 - \lambda)$. We solve $t\sigma_i : (1 - t)\ \sigma_j = \lambda : (1 - \lambda)$ and obtain the proposed mixing method:

$$mix(x_i, x_j) = \frac{t(x_i - \mu_i) + (1-t)(x_j - \mu_i)}{\sqrt{t^2 + (1-t)^2}} \tag{1}$$

$$where\ t = \frac{1}{1 + \frac{\sigma_i}{\sigma_j} \cdot \frac{1-\lambda}{\lambda}}$$

3.2 Training Paradigm

Let $g(., \theta)$ denote the classification model used, where θ denotes the model parameters. Assuming this model has M layers, we choose to mix the hidden representations at the m-th layer, $m \in [0, M]$. Mathematically the m-th layer is denoted as $g_m(., \theta)$, hence the hidden representation of the m-th layer is $h_m = g_m(h_{m-1}, \theta)$. The 0-th layer is considered as the embedding layer. Hence, for two text samples x_i and x_j, $h_0^i = W_E x_i$, $h_0^j = W_E x_j$ and the following hidden representations are as follows:

$$h_l^i = g_l(h_{l-1}^i, \theta), l \in [1, m] \tag{2}$$

$$h_l^j = g_l(h_{l-1}^j, \theta), l \in [1, m] \tag{3}$$

These hidden representations at the m-th layer are mixed using equation 1. We denote this mixed representation as $\tilde{h}_m$. Mixup at the m-th layer is thus defined as follows:

$$\tilde{h}_m = \frac{t h_m^i + (1-t) h_m^j}{\sqrt{t^2 + (1-t)^2}} \tag{4}$$

The continued forward pass after the mixed hidden representation has been generated is defined as follows:

$$\tilde{h}_l = g_l(\tilde{h}_{l-1}, \theta), l \in [m+1, M] \tag{5}$$

The layers chosen for mixup are denoted by S where $S = \{S_1, S_2, ...\}$ where each $S_i \in [0, M]$. The layer m where mixup occurs is chosen randomly from S with equal probability given to each layer in S and sampled separately for each pair of examples that are mixed.

.

For all variants, the output labels are also interpolated as follows: $y^{mix} = \lambda y^{k1} + (1 - \lambda) y^{k2}$.

4 OBSERVATIONS

Mixup has shown great improvements in sentence classification problems across all works. show that it also acts as a regularizer, providing an additional signal compared to dropout. note that Mixup can be used as an important step in semi supervised learning paradigms. Additionally note that hidden space interpolations improve performance under high levels of sparsification and for out of distribution samples as well.

REFERENCES

[1] Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. 2001. Vicinal risk minimization. In *Advances in neural information processing systems*. 416–422.

[2] Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2147–2157. https://doi.org/10.18653/v1/2020.acl-main.194

[3] Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting Data with Mixup for Sentence Classification: An Empirical Study. *arXiv preprint arXiv:1905.08941* (2019).

[4] Amit Jindal, Arijit Ghosh Chowdhury, Aniket Didolkar, Di Jin, Ramit Sawhney, and Rajiv Ratn Shah. 2020. Augmenting NLP models using Latent Feature Interpolations. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 6931–6936. https://doi.org/10.18653/v1/2020.coling-main.611

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[6] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. 2018. Manifold mixup: Better representations by interpolating hidden states. *arXiv preprint arXiv:1806.05236* (2018).

[7] William Yang Wang and Diyi Yang. 2015. That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 2557–2563. https://doi.org/10.18653/v1/D15-1306

[8] Jason W Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196* (2019).

[9] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[10] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*. 649–657.