Evaluation of Regularization Technique for Transformers

Hugo S. Oliveira[12]
hugo.soares@fe.up.pt
Pedro P. Ribeiro[12]
pribeiro@fc.up.pt
Helder P. Oliveira[12]
helder.f.oliveira@inesctec.pt

[1] INESC TEC
Porto, Portugal
[2] Faculty of Sciences, University of Porto.
Porto, Portugal

Abstract

Regularization techniques enhance deep learning models' generalization and robustness. We evaluate Patch Mix for Transformers. Patch Mix involves random patch replacing within the input sequence with similar patches of a Transformer during training. This encourages the model to learn more robust representations independent of patch locations, avoiding co-adaptation, improving generalization, and mitigating overfitting. Evaluating Patch Mix on benchmark datasets, we compare it with Dropout. Results show Patch Mix effectively reduces overfitting, focusing on meaningful patch interactions rather than specific locations.

Regularization, Transformers, Dropout

1 INTRODUCTION

Deep learning models, particularly Transformer architectures, have succeeded remarkably across various domains, including natural Natural Language Processing (NLP), computer vision, and speech recognition. However, these models often suffer from overfitting, memorizing training data and struggling to generalize to unseen examples. To address this challenge, regularization techniques [2, 3, 4] are employed to improve the models' generalization capabilities and robustness. This study introduces a regularization variation technique called Patch Mix for Transformers, by randomly shuffling and replacing patches within the input sequence during training Fig. 1.

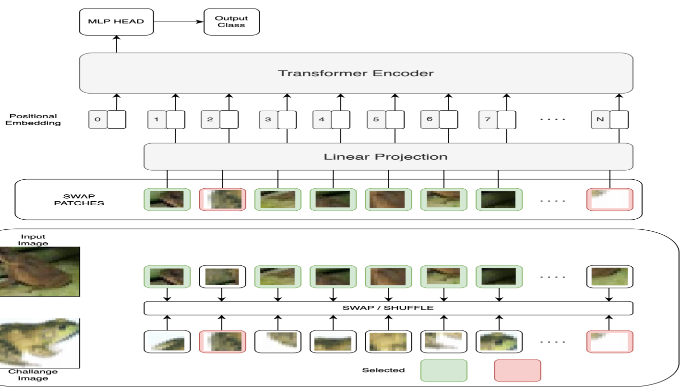

Figure 1: Patch Mix Mix main architecture

While Dropout [1] has proven effective in many cases, it operates at the element level and does not explicitly consider the spatial dependencies present in the input data. To evaluate the effectiveness of Patch Mix, we conduct a comparative analysis with other regularization techniques.

2 LITERATURE REVIEW

Convolutional Neural Networks (CNN)s, and Transformers have performed exceptionally in various domains. However, these models often suffer from overfitting. Among the most common regularization techniques, Dropout and batch normalization [2] is the primary ready-to-use Swiss knife employed in any neural network. The embedding level can also be regularized using DropDim [7], forcing the self-attention to encode meaningful features with a certain number of embedding dimensions erased. Similarly, patch erasing [5] is an effective regularizer, extremely useful in data with very sparse representation. A widely used regularization technique known as Label Smoothing [6]. Inspired by Dropout, Cutout serves as an additional regularization technique. Mixup can be defined as a genre of data augmentation with random samples to combine multiple inputs into a new Mix input.

The main problem of the Transformer and Vision Transformers (VIT) counterpart is the higher degree of overfitting when trained in medium datasets. The choice of the regularization technique depends on the specific characteristics of the model, the dataset, and the task at hand.

3 METHODOLOGY and BACKGROUND

The Patch Mix is a regularization and data argumentation technique that mixes patches between images during training. To effectively evaluate the performance of PatchMix, benchmarks medical datasets such as Medical MNIST 0 containing 6 classes of images among 58954 samples, and ISIC 2019 0 with 25,331 dermoscopic images among 8 classes.

3.1 Vision Transformers (ViT)

VIT is a deep learning architecture that applies the transformer model, originally designed for natural language processing, to the domain of computer vision. It has achieved state-of-the-art performance on various image classification tasks. VIT represents an image as a sequence of patches and employs a self-attention mechanism to capture global and local relationships between these patches.

Mathematical Formulation

Let's define an image $\mathbf{X}$ of size $H \times W \times C$, where H represents the height, W denotes the width, and C corresponds to the number of channels. To convert the image into a sequence of patches, we reshape it into a tensor $\mathbf{P}$ of shape $\left(\frac{H}{p}\right) \times \left(\frac{W}{p}\right) \times (p \times p \times C)$, where p is the patch size. Each patch is a vector of length d, representing its content.

$$\begin{aligned} Q &= P \cdot W_Q \\ K &= P \cdot W_K \\ V &= P \cdot W_V \end{aligned} \tag{1}$$

The input patches $\mathbf{P}$ are then linearly projected into query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ embeddings through weight matrices $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$, respectively:

$$A = softmax\left(\frac{Q\dot{K}^T}{\sqrt{d}}\right) \tag{2}$$

The self-attention mechanism is then applied to $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ to capture the relationships between patches. It computes the attention weights $\mathbf{A}$ by applying the softmax function to the scaled dot-product between $\mathbf{Q}$ and $\mathbf{K}$:

$$A = softmax\left(\frac{Q \cdot K^T}{\sqrt{d}}\right) \tag{3}$$

The attention weights $\mathbf{A}$ are then used to compute the output $\mathbf{O}$ of the self-attention layer:

$$O = A \cdot V \tag{4}$$

Multiple self-attention layers, feed-forward neural networks, and layer normalisation are stacked to form the Vision Transformer model. VIT

[0]https://challenge.isic-archive.com/landing/2019/https://challenge.isic-archive.com/landing/2019/

[0]https://www.kaggle.com/datasets/andrewmvd/medical-mnisthttps://www.kaggle.com/datasets/andrew
mnist

Table 1: Performance of regularization methods

Dataset	Medical MNIST						ISIC 2019					
Patch Size	8		16		32		8		16		32	
Technique	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Cross Entropy	0.803	0.862	0.753	0.845	0.668	0.802	0.549	0.785	0.479	0.732	0.363	0.620
Label Smoothing	0.800	0.882	0.666	0.813	0.899	0.943	0.563	0.740	0.486	0.684	0.369	0.589
Cutout	0.810	0.875	0.762	0.763	0.671	0.793	0.540	0.781	0.472	0.731	0.361	0.617
Mixup	0.810	0.813	0.762	0.855	0.671	0.782	0.605	0.821	0.539	0.771	0.419	0.675
CutMix	0.810	0.893	0.762	0.843	0.6071	0.765	0.622	0.834	0.549	0.788	0.413	0.671
Dropdim	0.810	0.882	0.662	0.842	0.6071	0.765	0.560	0.801	0.512	0.752	0.381	0.597
PatchMix	0.850	0.902	0.801	0.884	0.691	0.763	0.622	0.842	0.561	0.784	0.422	0.683

allows leveraging the transformer model's self-attention mechanism, by treating images as sequences of patches and capturing their relationships through self-attention.

3.2 PatchMix

PatchMix combines a pair of images to train VIT to predict the mixing ratio and the corresponding image categories of two adversarial sets.

Two image pairs $x_a, x_b \in [0,1]^{C\times H\times W}$ with their corresponding number of channels C, height H, and width W,l are used with their corresponding labels y_a and y_b respectively. Given an patch size P, both images x_a, x_b are divided into equal patches of size I_a, I_b, where $I_a = [x_a^1, x_a^2, \dots, x_a^N$ and $I_b = [x_b^1, x_b^2, \dots, x_b^N$, where x^i correspond to the ordered i patch of the image pairs x_a or x_b. The number of patches, N is computed as $N = \frac{H}{P} \times \frac{W}{P}$, conditioned to the partition factor P being integer multiple of H and W.

The mixed image x_s combines sample patches from I_a and I_b, with a sample ratio expressed by λ. The λ ratio corresponds to a sample of the Bates distribution, as $\lambda \approx Beta(\alpha, \alpha)$, with α defining the Bates distribution. The λ value is converted to a discrete one $\lambda^{'} = [\lambda \times N]$. The selection of parches from both I_a and I_b is performed by a set of binary selection masks $M_i = [M^1, M^2, \dots, M^N] \in \{0,1\}^N$, with $M^i = 1$ corresponds to the first image patch. The final image patch x_s with patch combination is then given in Equ. 5 as

$$x_s(x_a, x_b, \lambda^{'}) = I_a * M + I_b * (1 - M) \tag{5}$$

Resulting in a final x_s image series with combinations with patches both from I_a and I_b, sampled according to λ. To train the VIT, cross-entropy is employed with a modified loss to predict labels y_a and y_b, expressed in Equ. 6

$$\zeta_s(x_a, y_a; x_b, y_b; \lambda) = \lambda \zeta_{ce}(f(x_s; \theta_t, y_a) + (1-\lambda)\zeta_{ce}(f(x_s, \theta_t), y_b) \tag{6}$$

with ζ_{ce} the cross-entropy loss, $f(x_s, \theta_t)$ the prediction $\hat{y}$ for image x with the given parameters θ_t. In summary, PatchReplace can be described as in Algo. 3.2.

PatchReplace Vision Transformer

```
Require: $[x_a, y_a], [x_b, y_b]$
  return $\hat{y_a}$and $\hat{y_b}$
Ensure: $y = x^n$
  $W, H = \min(size(x_a, x_b))$
  $N = \frac{H}{P} \times \frac{W}{P}$
  $\lambda \approx Beta(\alpha, \alpha)$
  $x_s(x_a, x_b, \lambda^{'}) = I_a * M + I_b * (1 - M)$
Require: A transformer encoder-decoder architecture
Ensure: A class label for $x_a, x_b$
  * Split $x_a$ and $x_b$ into $N$ patches of size $P$
  * Perform patch Replace with token accordingly
  * Encode the patches into a sequence vector
  * Pass the vectors through the transformer encoder
  * Decode the output of the transformer encoder
  * Classify the patch on the transformer decoder $\hat{y_a}, \hat{y_b}$
```

The VIT algorithm first splits the input image into a sequence of patches. The set of patches gets replaced by another target image. The size of the patches is specified by the patch size P. The number of patches equals the total number of pixels in the image divided by the patch size and encoded into a sequence of vectors. The final step is to classify the image patches I_a or I_b into the respective classes $\hat{y_a}, \hat{y_b}$ based on the output of the decoder and mix ratio.

4 EXPERIMENTS and RESULTS

To evaluate the performance of PatchMix, we deploy the method into the mentioned datasets. The VIT transformer is fine-tuned with a rigid set for comparison purposes, employing 8 encoder blocks with 512 hidden dimensions and a 0.1 dropout rate. Images are normalized into the exact dimensions for easy patch split. The same model architecture is compared against other state-of-the-art regularization techniques such as Cross entropy, Cutout, Mixup, DropDim, CutMix and Label smoothing. Results are summarized in Table 1.

5 CONCLUSIONS

The best results were obtained using a patch size of 8. We tested patch sizes of 8,16, and 32. Smaller patch sizes lead to many image patches, providing a more extensive training sequence. However, this also increases the computational requirements quadratically $O(n^2)$. In this study, we investigated the effectiveness of Patch Mix variation, a novel regularization technique for Transformers, compared to the widely used Dropout method and other regularization techniques. Our findings demonstrate that Patch Mix surpasses Dropout and different ways to improve the generalization and robustness of deep learning models, namely Vision Transformers.

Acknowledgements

This work is co-financed by Component 5 - Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021 - 2026, within project HfPT, with reference 41, and by the Portuguese Foundation of Science and Technology - FCT under the Ph.D grant "2021.06275.BD"

References

[1] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[2] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[3] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pages 4313–4324. PMLR, 2020.

[4] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[5] Hugo S Oliveira, Pedro P Ribeiro, and Helder P Oliveira. Evaluation of regularization techniques for transformers-based models. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 312–319. Springer, 2023.

[6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[7] Hao Zhang, Dan Qu, Keji Shao, and Xukui Yang. DropDim: A regularization method for transformer networks. *IEEE Signal Processing Letters*, 29:474–478, 2022. doi: 10.1109/lsp.2022.3140693. URL `https://doi.org/10.1109%2Flsp.2022.3140693`.