FORMATTING INSTRUCTIONS FOR ICLR 2020 CONFERENCE SUBMISSIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

The abstract paragraph should be indented 1/2 inch (3 picas) on both left and right-hand margins. Use 10 point type, with a vertical spacing of 11 points. The word ABSTRACT must be centered, in small caps, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 INTRODUCTION

2 BACKGROUND ON ATTENTION MECHANISM AND RELATED WORK

2.1 CONVOLUTION LAYER

A convolution layer is defined by the size of its kernel The output of one layer of convolution for an input $\boldsymbol{X} \in \mathbb{R}^{W \times H \times C}$ is

2.2 SELF ATTENTION LAYER

2.3 ATTENTION MECHANISM IN VISION

Successful work to use attention on images (Ramachandran et al., 2019; Bello et al., 2019).

Model	relative	sinusoids	learned
Vaswani et al. (2017)		✓	
Radford et al. (2018)			✓
Devlin et al. (2018)			✓
Dai et al. (2019)	✓	✓	
Yang et al. (2019)	✓	✓	
Bello et al. (2019)	✓		✓
Ramachandran et al. (2019)	✓		✓

Table 1: Type of position encoding used by transformers models applied to text (*top*) and images (*bottom*). When multiple types have been tried by the authors, we report the one advised.

3 ATTENTION LAYER CAN IMPLEMENT CONVOLUTION LAYER

Our goal is to show that attention layers have the expressive power to learn convolution filter and that we can learn such filters in practice. The ability of the Attention mechanism to encode spacial filters (3×3 kernels) depends heavily on how we encode position in the image.

Understanding how attention can apply convolutions can help to:

- find better position encoding scheme,
- explain the need of multiple heads.

We take inspiration from (Dai et al., 2019, TransformerXL) to define relative positional encoding. They decompose the computation of the attention coefficients as follow:

$$\begin{aligned}\mathbf{A}^{\text{abs}}_{i,j} &= (\boldsymbol{E}_{x_i} + \boldsymbol{U}_i)^\top \boldsymbol{W}_q^\top \boldsymbol{W}_k (\boldsymbol{E}_{x_j} + \boldsymbol{U}_j) &(1)\\ &= \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(b)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(c)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(d)} &(2)\end{aligned}$$

with $\boldsymbol{E}_{x_i}$ the embedding of token x_i, $\boldsymbol{U}_i$ the absolute position encoding, $\boldsymbol{R}_{i-j}$ relative position encoding. They replace all absolute position encoding with relative ones:

$$\mathbf{A}^{\text{rel}}_{i,j} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(b)} + \underbrace{\boldsymbol{u}^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(c)} + \underbrace{\boldsymbol{v}^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(d)} \tag{3}$$

To mimic the CNN computation, each attention head focuses on a pixel at a given relative position. Hence reproducing a 3×3 kernel requires 9 attention heads and the ability for each of them to perfectly attend on one pixel at a relative position. Given that spatial convolution filters are not conditioned on the input data, we set $\boldsymbol{W}_{k,E}$ and $\boldsymbol{W}_{k,R}$ to 0, leaving only the (d) term in equation 3. We further write the attention computation in the 2D setting,

$$\mathbf{A}^{\text{conv}}_{i,j} = \boldsymbol{v}^\top \mathbf{W}_{k,R} \mathbf{R}_{\boldsymbol{p}_i - \boldsymbol{p}_j} \tag{4}$$

with $\boldsymbol{p}_i$ the row and column position of pixel i. As both $\boldsymbol{v}$ and $\mathbf{W}_{k,R}$ are learnable parameters, we simplify the expression to learn a single vector $\mathbf{t}_h$ for each head h.

3.1 Attention on a Pixel at any Relative Position

How to discern relative position? The goal is to reproduce the 3×3 pattern of CNN kernel with attention instead. Each head should be able to chose which pixel to attend based on its relative position to the query token x_i. The most basic attention mechanism is the dot product $(\boldsymbol{p}_i - \boldsymbol{p}_j)^\top \mathbf{t}_h$ between the relative positions of the input pixels and the relative target positions $\mathbf{t}_h = (x_h, y_h)^\top$. The dot product is large for pixels in the same direction as $\mathbf{t}_h$, however it also grows for farther pixels. To be able to focus on a relative pixel position (direction and distance), we propose the following relative position encoding

$$\boldsymbol{R}_\Delta = \begin{pmatrix}\Delta_x & \Delta_y & \Delta_x^2 & \Delta_y^2 & 1 & 1\end{pmatrix}^\top \tag{5}$$

Instead of learning the target vector $\mathbf{t}_h$, we parametrize it for each head to attend to the relative pixel position (x_h, y_h),

$$\mathbf{t}_h = -\alpha_h \begin{pmatrix}-2x_h & -2y_h & 1 & 1 & x_h^2 & y_h^2\end{pmatrix}^\top \tag{6}$$

The attention coefficient for head h between pixel i and j is given by the dot product,

$$\boldsymbol{R}_\Delta^\top \mathbf{t}_h = -\alpha_h((\Delta_x - x_h)^2 + (\Delta_y - y_h)^2), \tag{7}$$

with $\Delta = \boldsymbol{p}_i - \boldsymbol{p}_j$. The maximum attention coefficient is 0 when $\Delta = (x_h, y_h)$, i.e. the center of attention of a given head. The α coefficient controls how spiky the attention is (analogous to temperature in the softmax). Figure 1 gives a representation of the attention weights for a fixed query pixel i and different α and (x_h, y_h).

Discussion. Some relations can be drawn to RBF kernels centered at the target pixels. We are expressing a more general form of convolutions where *(i)* the input pixels do not follow a grid shape but has any pattern and *(ii)* the inputs are not single pixels but weighted averages of local patches in the image.

Simplification. One further simplification is to remove the constant terms of $\boldsymbol{R}_\Delta$ because the softmax (renormalization) is shift invariant and constant terms for a head are useless.

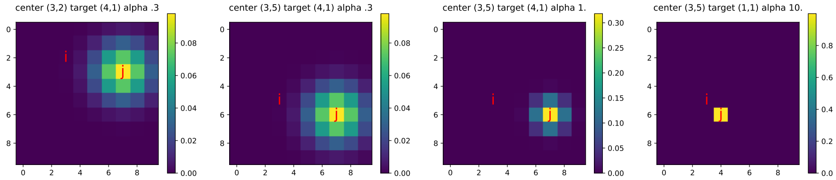


Figure 1: Attention coefficient for different center i and target j with varying α parameters. Target designate the relative position that the head attends to.

3.2 Non Isotropic Gaussian

In this section, we present an extension of the relative positional encoding introduced above to model non-isotropic Gaussians.

$$f_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{x}) = \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{8}$$

We drop the normalization factor of the Gaussian distribution as it is replaced by the softmax of the attention mechanism. The matrix $\boldsymbol{\Sigma}^{-1}$ must be positive semi-definite. We parametrize it as

$$\boldsymbol{\Sigma}^{-1} = (\boldsymbol{\Sigma}^{-1/2})^\top \boldsymbol{\Sigma}^{-1/2} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \tag{9}$$

Our goal is to write the exponent as a dot product of two vectors:

- one vector $R_{\boldsymbol{x}}$ dependent only on $\boldsymbol{x}$, the relative positional encoding,
- one vector $\mathbf{t}_h$ dependent only on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, the attention head.

$$R_{\boldsymbol{x}} = \begin{pmatrix} x_1 & x_2 & x_1 & x_2 & x_1^2 & x_2^2 & x_1 x_2 & 1 & 1 & 1 \end{pmatrix} \tag{10}$$

$$\mathbf{t}_h = -\frac{1}{2}\begin{pmatrix} -2a\mu_1 & -2c\mu_2 & -2b\mu_2 & -2b\mu_1 & a & c & 2b & a\mu_1^2 & c\mu_2^2 & 2b\mu_1\mu_2 \end{pmatrix} \tag{11}$$

$$R_{\boldsymbol{x}}^\top \mathbf{t}_h = -\frac{1}{2}\left[a(x_1-\mu_1)^2 + 2b(x_1-\mu_1)(x_2-\mu_2) + c(x_2-\mu_2)^2\right] = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \tag{12}$$

We can check that setting $b = 0$ and $a = c = \alpha$ recovers the expression of the isotropic Gaussian derived above.

3.3 Implications to Sinusoid or Learned Positional Encoding

3.4 Exploit Attention on Features?

- Does the terms (a) (b) (c) allow to condition the CNN filters on the input data?

4 Experiments

We want to validate that not only Transformer architecture can express CNN filters but that it can also learn such filters with SGD from the data. It has already been shown by (Bello et al., 2019) that adding attention features to CNNs can improve performence on Cifar-100 and ImageNet. We want to stick to a fully attentional model, and show that it matches its siblings ResNet of equivalent depth.

4.1 Performance Against ResNet

We design a smaller ResNet (i.e. ResNet10) which have similar number of layers as our transformer. The sizes of the models are displayed in Table 2.

Model	Number of parameters
ResNet10	4.9M
BERT 4 layers	1.85M
BERT 6 layers	9.56M

Table 2: Number of parameters per model.

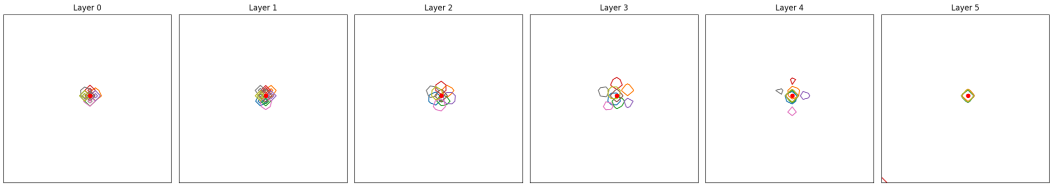


Figure 2: Contours of attention weights per layer for the 6 layers fully attentional model.

5 Discussion

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

References

Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention Augmented Convolutional Networks. *arXiv:1904.09925 [cs]*, April 2019.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *CoRR*, abs/1901.02860, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *CoRR*, abs/1906.05909, 2019. URL http://arxiv.org/abs/1906.05909.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237, 2019. URL http://arxiv.org/abs/1906.08237.

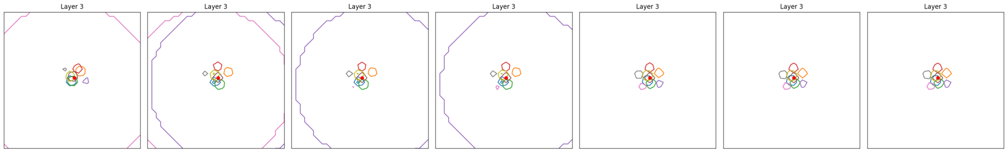
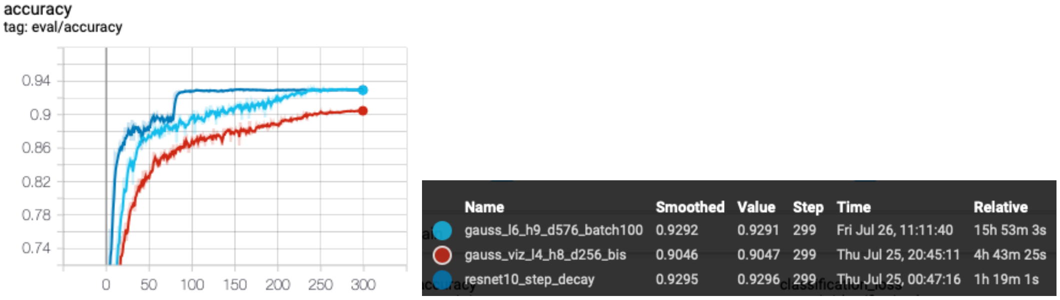


Figure 3: Contours of attention weights during training (300 epochs) at layer 3.



	Name	Smoothed	Value	Step	Time	Relative
●	gauss_l6_h9_d576_batch100	0.9292	0.9291	299	Fri Jul 26, 11:11:40	15h 53m 3s
●	gauss_viz_l4_h8_d256_bis	0.9046	0.9047	299	Thu Jul 25, 20:45:11	4h 43m 25s
●	resnet10_step_decay	0.9295	0.9296	299	Thu Jul 25, 00:47:16	1h 19m 1s

Figure 4: Evaluation accuracy on CIFAR-10 of a small ResNet and two Gaussian Attention models.