An Attention Module for Convolutional Neural Networks

Zhu, Baozhou; Hofstee, Peter; Lee, Jinho; Al-Ars, Zaid

DOI
10.1007/978-3-030-86362-3_14

Publication date
2021

Document Version
Accepted author manuscript

Published in
Artificial Neural Networks and Machine Learning – ICANN 2021 - 30th International Conference on Artificial Neural Networks, Proceedings

An Attention Module for Convolutional Neural Networks

Baozhou Zhu[1], Peter Hofstee[1,2], Jinho Lee[3], and Zaid Al-Ars[1]

[1] Delft University of Technology, Delft, The Netherlands
{b.zhu-1, z.al-ars}@tudelft.nl
[2] IBM Systems, Austin, TX, USA
[3] Yonsei University, Seoul, South Korea

Abstract. Attention mechanism has been regarded as an advanced technique to capture long-range feature interactions and to boost the representation capability for convolutional neural networks. However, we found two ignored problems in current attentional activations-based models: the approximation problem and the insufficient capacity problem of the attention maps. To solve the two problems together, we initially propose an attention module for convolutional neural networks by developing an AW-convolution, where the shape of attention maps matches that of the weights rather than the activations. Our proposed attention module is a complementary method to previous attention-based schemes, such as those that apply the attention mechanism to explore the relationship between channel-wise and spatial features. Experiments on several datasets for image classification and object detection tasks show the effectiveness of our proposed attention module. In particular, our proposed attention module achieves 1.00% Top-1 accuracy improvement on ImageNet classification over a ResNet101 baseline and 0.63 COCO-style Average Precision improvement on the COCO object detection on top of a Faster R-CNN baseline with the backbone of ResNet101-FPN. When integrating with the previous attentional activations-based models, our proposed attention module can further increase their Top-1 accuracy on ImageNet classification by up to 0.57% and COCO-style Average Precision on the COCO object detection by up to 0.45. Code and pre-trained models will be publicly available.

Keywords: Attention mechanism · Convolution · Representation.

1 Introduction

Recent literature [6, 12, 31] have investigated the attention mechanism since it can improve not only the representation power but also the representation of interests. Convolutional neural networks can extract informative features by blending cross-channel and spatial information [9]. Attention modules [19, 29] can learn "where" and "what" to attend in channel and space axes, respectively, by focusing on important features and suppressing unnecessary ones of the activations. Dynamic Filter Networks [13, 17] generate the filters conditioned on

the input and show the flexibility power of such filters because of their adaptive nature, which has become popular in prediction [15] and Natural Language Processing [30]. Both Dynamic Filter Networks and attention-based models are adaptive based on the inputs, but there are significant differences between them. Attention-based models [9, 29] produce attention maps using the attention mechanism to operate on the activations of convolution. On the contrary, Dynamic Filer Networks [22] generate input information-specific kernels, such as position-specific kernels [22] and few-shot learning setting-specific kernels [32], which work as the weights of convolution. Our proposed attention module leverages the attention mechanism to compute the attention maps for attending the activations of convolution, so it is clear to categorized the models applied with our proposed attention module as attention-based models instead of Dynamic Filter Networks.

In this paper, we analyze two ignored problems of the current attentional activations-based models: the approximation problem and the insufficient capacity problem of the attention maps. To address the two problems together, we originally propose an attention module by developing an AW-convolution, where the shape of the attention maps matches that of the weights instead of the activations. Besides, we present and refine the architecture of calculating attention maps A. Our proposed attention module is a complementary method to previous attention mechanism-based modules, such as Attention Augmented (AA) convolution [2], the SE [10] and CBAM [29] modules in the attentional activations-based models. Integrating with our proposed attention module, the accuracy of SE-Net, and CBAM-Net will be improved further.

We use image classification and object detection tasks to demonstrate the effectiveness of our proposed attention module. With negligible computational complexity increase, our proposed attention module can boost the image classification and object detection task performance, and it can achieve better accuracy when integrating with other attention-based models. In particular, our proposed attention module achieves 1.00% Top-1 accuracy improvement on ImageNet classification over a ResNet101 baseline and 0.63 COCO-style Average Precision improvement on the COCO object detection on top of a Faster R-CNN baseline with the backbone of ResNet101-FPN. When integrating with the previous attentional activations-based models, our proposed attention module can further increase their Top-1 accuracy on ImageNet classification by up to 0.57% and COCO-style Average Precision on the COCO object detection by up to 0.45.

2 Related work

2.1 Network engineering

Increasing the depth of convolutional neural networks has been regarded as an intuitive way to boost performance, which is the philosophy of VGGNet and ResNet [7]. In addition, since the skip connection from ResNet shows a strong ability to assist the gradient flow, WideResNet, PyramidNet, Inception-ResNet [23], and ResNeXt are ResNet-based versions proposed to explore further the

influence of the width, the increase of the width, the multi-scale and the cardinality of convolution, respectively. In terms of efficiency, DenseNet [11] reuses the feature maps by concatenating the feature maps from different layers. In particular, MobileNet [8] and ShuffleNet [20] series present the advantage of depthwise convolution and the shuffle operation between various group convolutions, respectively. Another design approach uses automated neural architecture search, which achieves state-of-the-art performance regarding both accuracy and efficiency across a range of computer vision tasks [24].

2.2 Attention mechanism

The attention mechanism plays an important role in the human vision perceptron since it can allocate the available resources to selectively focus on processing the salient part instead of the whole scene [5]. Multiple attention mechanisms are used to address a known weakness in convolution [3, 4, 10, 14, 19], by capturing long-range information interactions [1, 26]. The Inception family of architectures [23], Multigrid Neural Architectures [14], and Octave Convolution [3] aggregate the scale-space information, while Squeeze-and-Excitation Networks [10] and Gather-Excite [9] adaptively recalibrate channel-wise response by modeling interdependency between channels. GALA [19], CBAM [29], and BAM [21] refine the feature maps separately in the channel and spatial dimensions. Attention Modules [27] and self-attention [2, 25] can be used to exploit global context information. Precisely, non-local networks [28] deploy self-attention as a generalized global operator to capture the relationship between all pairwise convolutional feature maps interactions. Except for applying the attention mechanism to computer vision tasks [16], it has been a widespread adoption to modeling sequences in Natural Language Processing [30].

3 Proposed attention module

In this section, we analyze the two ignored problems in current attentional activations-based models and develop an attention module that mainly refers to the AW-convolution. Besides, we refine the branch of calculating the attention maps. Last but not least, we integrate our proposed attention module with other attention-based models.

3.1 Motivation

First, we define basic notations in a traditional convolutional layer. In a traditional convolutional layer, the input activations, weights, and output activations are denoted as I, K, and O, respectively. For the input activations $I \in R^{N \times C_1 \times H \times W}$, N, C_1, H, and W refer to the batch size, the number of input channels, the height, and width of the input feature maps, respectively. For the weights $K \in R^{C_2 \times C_1 \times h \times w}$, C_2, h and w refer to the number of output channels, the height and width of the weights, respectively. For the output

activations $O \in R^{N \times C_2 \times H \times W}$, it is computed as the convolution between the input activations I and the weights K. In particular, every individual value of the output activations $O_{[l,p,m,n]}$ is calculated as follows.

$$O_{[l,p,m,n]} = \text{Convolution}(I, K) = \sum_{o=1}^{C_1} \sum_{j=1}^{h-1} \sum_{k=1}^{w-1} I_{[l,o,m'+j,n'+k]} \times K_{[p,o,j,k]} \tag{1}$$

where $l = 0, ..., N-1$, $m = 0, ..., H-1$, $n = 0, ..., W-1$, $o = 0, ..., C_1 - 1$, $p = 0, ..., C_2 - 1$, $m' = m - \frac{h-1}{2}$, $n' = n - \frac{w-1}{2}$.

To apply the attention mechanism on the input activations I, previous attentional activations-based models produce the channel attention maps $A_c \in R^{N \times C_1 \times 1 \times 1}$ and spatial attention maps $A_s \in R^{N \times 1 \times H \times W}$ separately. For example, applying the channel attention maps A_c on the input activations I is presented as $O = \text{Convolution}((I \odot A_c), K)$, where $\odot$ refers to the Hadamard product and broadcasting during element-wise multiplication is omitted.

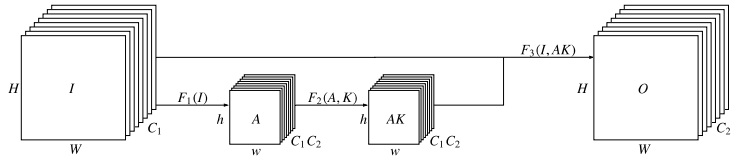
Approximation problem of the attention maps To thoroughly attend the input activations I, we need to compute the attention maps $A_f \in R^{N \times C_1 \times H \times W}$ and apply it as $O = \text{Convolution}((I \odot A_f), K)$, which requires too much computational and parameter overhead. Thus, all the current attentional activations-based models produce the attention maps separately into the channel attention maps A_c and spatial attention maps A_s. We use A_c and A_s to approximate the four-dimensional attention map A_f, which leads to the approximation problem of attention maps.

Inspired by convolution, we adopt local connection and attention maps sharing to reduce the size of the attention maps. We compute the attention maps $A_a \in R^{N \times C_1 \times h \times w}$ as follows, where $\otimes$ is a special element-wise multiplication since it only works associated with convolution.

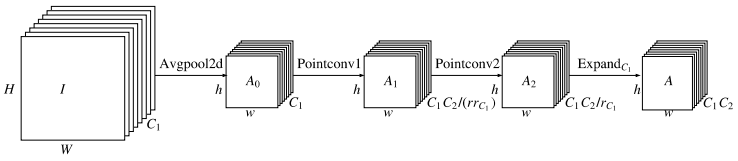
$$\begin{aligned} O_{[l,p,m,n]} &= \text{Convolution}(I \otimes A_a, K) \\ &= \sum_{o=1}^{C_1} \sum_{j=1}^{h-1} \sum_{k=1}^{w-1} (I_{[l,o,m'+j,n'+k]} \times A_{a\,[l,o,j,k]}) \times K_{[p,o,j,k]} \end{aligned} \tag{2}$$

Insufficient capacity problem of the attention maps To compute different channels of the output activations of the convolution, the input activations are constrained to be recalibrated by the same attention map, i.e., the four-dimensional attention map A_f, which indicates the insufficient capacity of the attention maps. As each channel of the feature maps is considered as a feature detector, different channels of the output activations of the convolution expect the input activations to be adapted by different attention maps.

Take two channels of output activations of a convolutional layer as an example, the two channels are responsible for recognizing rectangle shape and triangle shape, respectively. Thus, it is reasonable for the two channels to expect that there are different attention maps for attending the input activations of the convolution (i.e., the attention maps to compute the channel of recognizing



(a) The AW-convolution architecture.



(b) The architecture of calculating attention maps A.

Fig. 1: The architecture of our proposed attention module.

the rectangle shape should be different from the attention maps to compute the channel of recognizing the triangle shape). To meet this expectation, we need to compute the five-dimensional attention map $A_{ic} \in R^{N\times C_2\times C_1\times 1\times 1}$ and apply it on the input activations as follows.

$$\begin{aligned} O_{[l,p,m,n]} &= \text{Convolution}(I \odot A_{ic[l,p,:,:,:]}, K) \\ &= \sum_{o=1}^{C_1}\sum_{j=1}^{h-1}\sum_{k=1}^{w-1}(I_{[l,o,m'+j,n'+k]} \times A_{ic[l,p,o,0,0]}) \times K_{[p,o,j,k]} \end{aligned} \tag{3}$$

To solve the approximation problem and the insufficient capacity problem of the attention maps together (i.e., combining the solution of Equation.2 and the solution of Equation. 3), we introduce our proposed attention module by developing the AW-convolution. Specifically, we propose to compute the attention maps $A \in R^{N\times C_2\times C_1\times h\times w}$ and apply it as follows where the attention maps $A_{[l,:,:,:,:]}$ has the same shape as that of the weights instead of the input activations. In this paper, "Attentional weights" refers to the element-wise multiplication result between the attention maps and the weights. Similarly, "Attentional activations" refers to the element-wise multiplication result between the attention maps and the activations in previous attentional activations-based models. Thus, $I \otimes A$ and $A_{[l,:,:,:,:]} \odot K$ represent the attentional activations and attentional weights, respectively. To reduces half the number of element-wise multiplications, we calculate attentional weights instead of attentional activations as follows.

$$\begin{aligned} O_{[l,p,m,n]} &= \text{Convolution}(I \otimes A, K) \\ &= \sum_{o=1}^{C_1}\sum_{j=1}^{h-1}\sum_{k=1}^{w-1} I_{[l,o,m'+j,n'+k]} \times (A_{[l,p,o,j,k]} \times K_{[p,o,j,k]}) \\ &= \text{Convolution}(I, A_{[l,:,:,:,:]} \odot K) = \text{AW-Convolution}(I, A \odot K) \end{aligned} \tag{4}$$

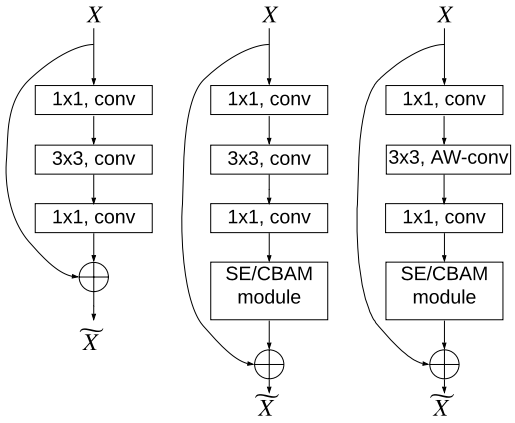


Fig. 2: The schema of bottlenecks when integrating with our proposed attention module. **Left:** bottleneck in ResNet. **Middle:** bottleneck in SE-ResNet/CBAM-ResNet. **Right:** bottleneck in AW-SE-ResNet/AW-CBAM-ResNet.

3.2 AW-convolution in proposed attention module

The AW-convolution in our proposed attention module is presented in Figure 1a. In this figure, the attention maps A has five dimensions, which is computed from the input activations I as $A = F_1(I)$. F_1 is a function to calculate the attention maps A given the input activations I. Then, the attentional weights $AK \in R^{N\times C_2\times C_1\times h\times w}$ is calculated as $AK = F_2(A, K) = K + A \odot K$. F_2 is a function to calculate the attentional weights AK given the weights K and the attention maps A. Finally, the output activations O is calculated from the input activations I and the attentional weights AK as follows.

$$\begin{aligned} &O_{[l,p,m,n]} = F_3(I, AK) = \text{AW-Convolution}(I, AK) \\ &= \sum_{i=1}^{C_1}\sum_{j=1}^{h-1}\sum_{k=1}^{w-1} I_{[l,o,m'+j,n'+k]} \times AK_{[l,p,o,j,k]} = \text{Convolution}(I, AK_{[l,:,:,:,:]}) \end{aligned} \tag{5}$$

where F_3 is a function to calculate the output activations O given the input activations I and the attentional weights AK. Compared with the traditional convolution, the attentional weights AK of the AW-convolution in our proposed attention module has five dimensions rather than four dimensions, which are different from each other for every individual sample of the input activations batch to convolute.

It is also worth explaining the definition of the function F_2. $AK = K + A \odot K$ instead of $AK = A \odot K$ is used to describe the function F_2 since it can be regarded as a residual design as follows.

$$\begin{aligned} O &= F_3(I, AK) = \text{AW-Convolution}(I, F_2(A, K)) \\ &= \text{Convolution}(I, K) + \text{AW-Convolution}(I, A \odot K) \end{aligned} \tag{6}$$

3.3 Calculating the attention maps A

As shown in Figure 1b, the architecture to compute the attention maps A (i.e., the definition of the function F_1) is presented, which can be expressed as follows. Avgpool2d aggregates feature responses from the whole spatial extent and embeds them into A_0, and Pointconv1 and Pointconv2 followed by Relu redistribute the pooled information to capture the dynamic and no-linear dependencies between channels and spatial spaces.

$$\begin{aligned} A = F_1(I) &= \text{Expand}_{C_1}(A_2) = \text{Expand}_{C_1}(\text{Pointconv2}(A_1)) \\ &= \text{Expand}_{C_1}(\text{Pointconv2}(\text{Pointconv1}(A_0))) \\ &= \text{Expand}_{C_1}(\text{Pointconv2}(\text{Pointconv1}(\text{Avgpool2d}(I)))) \end{aligned} \tag{7}$$

where Pointconv1 and Pointconv2 are pointwise convolutions. We add Batch Normalization and Relu layers after Pointconv1, while adding Batch Normalization and Sigmoid layers after Pointconv2, and they are omitted here to provide a clear expression.

In Figure 1b, Expand function along C_1 dimension, denoted as Expand_{C_1}, is used as an example, and Expand function can be also executed along N, C_2, h, and w dimensions in a similar way. Expand_{C_1} function is used to expand the tensor $A_2 \in R^{N \times (C_2 C_1 / r_{C_1}) \times h \times w}$ into the attention maps $A \in R^{N \times C_2 \times C_1 \times h \times w}$ with the reduction ratio r_{C_1}, including necessary squeeze, reshape, and expand operations. Expand_{C_1} can be expressed as follows.

$$\begin{aligned} A = \text{Expand}_{C_1}(A_2) = A_2.\text{reshape}(N, C_2, C_1/r_{C_1}, h, w).\text{unsqueeze(dim=3)} \\ .\text{expand}(N, C_2, C_1/r_{C_1}, r_{C_1}, h, w).\text{reshape}(N, C_2, C_1, h, w) \end{aligned} \tag{8}$$

Calculating the five-dimension attention maps A is not an easy computational task without careful design. Thus, we analyze the additional computational complexity of an AW-convolution compared with a traditional convolution as a reference to refine this design. Considering the trade-off between computational complexity and accuracy, all the experiments in the remainder of this paper use the same settings for the architecture of calculating the attention maps A in our proposed attention module, including $r_{C1} = C_1$, $r_{C_2} = r_{hw} = 1$, $r = 16$, used in all the stages, and $AK = K + A \odot K$ as the definition for the function F_2.

3.4 Integrating with other attention-based modules

In this section, we show how to integrate our proposed attention module with the previous attention-based convolutional neural networks to demonstrate the complementary relationship between our proposed attention module and other attention-based modules. Since applying our proposed attention module is using the AW-convolution to replace the traditional convolution, we can easily integrate our proposed attention module with any convolutional neural networks consisting of traditional convolution, including all the recently developed attention-based models [2, 10, 19, 21, 29].

We choose the recent attentional activations-based models, i.e., SE-Net and CBAM-Net, as examples to show how to integrate our proposed attention module with other attention-based models. Here we use the popular ResNet [7] as the backbone to apply the attention mechanism. As shown in Figure 2, the left side is the structure of a primary bottleneck in ResNet. The middle one is the structure of a bottleneck with SE/CBAM modules in SE-ResNet/CBAM-ResNet. Integrating the central bottleneck with our proposed attention module is completed by replacing its 3×3 convolution with a 3×3 AW-convolution, and its final structure in AW-SE-ResNet/AW-CBAM-ResNet is shown on the right side. In summary, our proposed attention module is a general module to be integrated seamlessly with any CNNs architectures, including previous attention-based CNNs.

4 Experimental results

4.1 ImageNet image classification

According to the results shown in Table 1, our proposed attention module is complementary to other attentional activations-based models. AW-ResNet50 achieves a 1.18% Top-1 error reduction compared with the ResNet50 baseline. Integrating with our proposed attention module, SE-ResNet50 [10] can improve further by 0.42% Top-1 accuracy. The Top-1 accuracy of our AW-SE-ResNet101 is 1.60% and 0.57% higher than that of ResNet101 and SE-ResNet101, respectively. To integrate with CBAM-ResNet [29] more carefully, we define CBAM-ResNet (MaxPool) and CBAM-ResNet (Spatial) separately to reduce computational complexity. We do not use max-pooled features in CBAM-ResNet. The Top-1 accuracy of AW-CBAM-ResNet50 is better than AW-ResNet50 by 0.18% but worse than AW-SE-ResNet50. The number of additional parameters for our proposed attention module is 0.16 M, which is much smaller than 2.83 M (i.e., one-sixteenth) of SE and CBAM modules. Moreover, it takes only 0.01 GFLOPs to apply our proposed attention module on the ResNet50 model on ImageNet classification, which is comparable with 0.01 GFLOPs and 0.04 to adopt the SE and CBAM modules and is negligible in terms of FLOPs to implement the baseline model.

Resource-constrained architecture To inspect the generalization of our proposed attention module in this resource-constrained scenario, we conduct the ImageNet classification with the MobileNet architecture [8]. We apply our proposed attention module to pointwise convolution instead of depthwise convolution in every two depthwise separable convolutions. When integrating with the CBAM models [29], we remove the max-pooled features and keep spatial attention maps. As shown in Table 1, AW-SE-MobileNet and AW-CBAM-MobileNet achieve 0.56% and 0.19% Top-1 accuracy improvements compared with SE-MobileNet [10] and CBAM-MobileNet, respectively. It is an impressive result that the Top-1 accuracy of AW-CBAM-MobileNet is 2.57% better than that of the MobileNet

Table 1: Comparisons of attention-based models on ImageNet classification. * refers to the baseline results from [29]. All the rest results are produced using the source code from [29].

Model	Top-1 Error	Top-5 Error	GFLOPs	Parameters (M)
ResNet50 [7] *	24.56%(+0.00%)	7.50%	3.86	25.56
AW-ResNet50	23.38%(+1.18%)	6.79%	3.87	25.72
SE-ResNet50 [10] *	23.14%(+1.42%)	6.70%	3.87	28.09
AW-SE-ResNet50	22.72%(+1.84%)	6.47%	3.88	28.25
AW-CBAM-ResNet50 (MaxPool)	22.82%(+1.74%)	6.41%	3.89	28.25
AW-CBAM-ResNet50 (Spatial)	23.20%(+1.36%)	6.58%	3.90	28.25
ResNet101 Baseline [7] *	23.38%(+0.00%)	6.88%	7.57	44.55
AW-ResNet101	22.38%(+1.00%)	6.21%	7.58	44.95
SE-ResNet101 [10] *	22.35%(+1.03%)	6.19%	7.58	49.33
AW-SE-ResNet101	21.78%(+1.60%)	5.74%	7.59	49.73
AW-CBAM-ResNet101 (MaxPool)	21.64%(+1.74%)	5.76%	7.60	49.73
AW-CBAM-ResNet101 (Spatial)	22.32%(+1.06%)	6.18%	7.61	49.73
MobileNet Baseline [8] *	31.39%(+0.00%)	11.51%	0.569	4.23
SE-MobileNet [10] *	29.97%(+1.42%)	10.63%	0.581	5.07
AW-SE-MobileNet	29.41%(+1.98%)	10.59%	0.623	5.52
CBAM-MobileNet [29]	29.01%(+2.38%)	9.99%	0.611	5.07
AW-CBAM-MobileNet (Spatial)	28.82%(+2.57%)	9.98%	0.652	5.52

baseline. For the MobileNet model, our proposed attention module increases the computation by 0.041 GFLOPs, while SE and CBAM modules need 0.012 and 0.041 GFLOPs, respectively. Also, the required parameters for our proposed attention module are 0.45 M, which is much less than 0.84 M for SE and CBAM modules.

4.2 Object Detection on COCO

To show the generalization of our proposed attention module, we apply it to object detection tasks. We evaluate our proposed attention module further on the COCO dataset, which contains $118K$ images (i.e., train2017) for training and $5K$ images (i.e., val2017) for validating. Here we intend to evaluate the benefits of applying our proposed attention module on the ResNet101-FPN backbone [18], where all the lateral and output convolutions of the FPN adopt our AW-convolution. The SE and CBAM modules are placed right before the lateral and output convolutions. As shown in Table 2, applying our proposed attention module on ResNet101-FPN boosts mAP@[0.5, 0.95] by 0.63 for the Faster R-CNN baseline. Integrating with attentional activations-based models, Faster R-CNNs with the backbones of ResNet101-AW-SE-FPN and ResNet101-AW-CBAM-FPN outperform Faster R-CNNs with the backbones of ResNet101-SE-FPN and ResNet101-CBAM-FPN by 0.34 and 0.45 on COCO's standard metric AP.

Table 2: Comparisons of attention-based Faster R-CNN on COCO. All the results are produced using Pytorch.

Backbone	Detector	mAP@[0.5, 0.95]	mAP@0.5	mAP@0.75
ResNet101-FPN [18]	Faster R-CNN	37.13(+0.00%)	58.28	40.29
ResNet101-AW-FPN	Faster R-CNN	37.76(+0.63%)	59.17	40.91
ResNet101-SE-FPN [10]	Faster R-CNN	38.11(+0.98%)	59.41	41.33
ResNet101-AW-SE-FPN	Faster R-CNN	38.45(+1.32%)	59.70	41.86
ResNet101-CBAM-FPN [29]	Faster R-CNN	37.74(+0.61%)	58.84	40.77
ResNet101-AW-CBAM-FPN	Faster R-CNN	38.19(+1.06%)	59.52	41.43

5 Conclusion

In this paper, we analyze the two ignored problems in attentional activations-based models: the approximation problem and the insufficient capacity problem of the attention maps. To address the two problems together, we propose an attention module by developing the AW-convolution, where the shape of the attention maps matches that of the weights rather than the activations, and integrate it with attention-based models as a complementary method to enlarge their attentional capability. We have implemented extensive experiments to demonstrate the effectiveness of our proposed attention module, both on image classification and object detection tasks.

Acknowledgment This work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative.

Bibliography

[1] Bello, I., Kulkarni, S., Jain, S., Boutilier, C., Chi, E.H., Eban, E., Luo, X., Mackey, A., Meshi, O.: Seq2slate: Re-ranking and slate optimization with rnns. CoRR **abs/1810.02019** (2018), http://arxiv.org/abs/1810.02019

[2] Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V.: Attention augmented convolutional networks. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)

[3] Chen, Y., Fan, H., Xu, B., Yan, Z., Kalantidis, Y., Rohrbach, M., Yan, S., Feng, J.: Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)

[4] Chen, Y., Kalantidis, Y., Li, J., Yan, S., Feng, J.: Aˆ 2-nets: Double attention networks. In: Advances in Neural Information Processing Systems. pp. 352–361 (2018)

[5] Corbetta, M., Shulman, G.L.: Control of goal-directed and stimulus-driven attention in the brain. Nature reviews neuroscience **3**(3), 201 (2002)

[6] Gregor, K., Danihelka, I., Graves, A., Rezende, D., Wierstra, D.: Draw: A recurrent neural network for image generation. In: International Conference on Machine Learning. pp. 1462–1471 (2015)
[7] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
[8] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR **abs/1704.04861** (2017), http://arxiv.org/abs/1704.04861
[9] Hu, J., Shen, L., Albanie, S., Sun, G., Vedaldi, A.: Gather-excite: Exploiting feature context in convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 9401–9411 (2018)
[10] Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
[11] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
[12] Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in neural information processing systems. pp. 2017–2025 (2015)
[13] Jia, X., De Brabandere, B., Tuytelaars, T., Gool, L.V.: Dynamic filter networks. In: Advances in Neural Information Processing Systems. pp. 667–675 (2016)
[14] Ke, T.W., Maire, M., Yu, S.X.: Multigrid neural architectures. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6665–6673 (2017)
[15] Klein, B., Wolf, L., Afek, Y.: A dynamic convolutional layer for short range weather prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4840–4848 (2015)
[16] Li, H., Liu, Y., Ouyang, W., Wang, X.: Zoom out-and-in network with map attention decision for region proposal and object detection. International Journal of Computer Vision **127**(3), 225–238 (2019)
[17] Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 510–519 (2019)
[18] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
[19] Linsley, D., Shiebler, D., Eberhardt, S., Serre, T.: Learning what and where to attend with humans in the loop. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=BJgLg3R9KQ
[20] Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 116–131 (2018)
[21] Park, J., Woo, S., Lee, J.Y., Kweon, I.S.: Bam: Bottleneck attention module. In: British Machine Vision Conference (BMVC). British Machine Vision Association (BMVA) (2018)
[22] Su, H., Jampani, V., Sun, D., Gallo, O., Learned-Miller, E., Kautz, J.: Pixel-adaptive convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11166–11175 (2019)

[23] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
[24] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2820–2828 (2019)
[25] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
[26] Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 2692–2700. Curran Associates, Inc. (2015), http://papers.nips.cc/paper/5866-pointer-networks.pdf
[27] Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3156–3164 (2017)
[28] Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7794–7803 (2018)
[29] Woo, S., Park, J., Lee, J.Y., So Kweon, I.: Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
[30] Wu, F., Fan, A., Baevski, A., Dauphin, Y., Auli, M.: Pay less attention with lightweight and dynamic convolutions. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=SkVhlh09tX
[31] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015)
[32] Zhao, F., Zhao, J., Yan, S., Feng, J.: Dynamic conditional networks for few-shot learning. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 19–35 (2018)