# NAM: Normalization-based Attention Module


Yichao Liu [1] Zongru Shao [1,2] Yueyang Teng [3] Nico Hoffmann [1]

[1]Helmholtz-Zentrum Dresden-Rossendorf [2]Center for Advanced Systems Understanding

[3]Northeastern University of China



## Abstract

Recognizing less salient features is the key for model compression. However, it has not been investigated in the revolutionary attention mechanisms. In this work, we propose a novel normalization-based attention module (NAM), which suppresses less salient weights. It applies a weight sparsity penalty to the attention modules, thus, making them more computational efficient while retaining similar performance. A comparison with three other attention mechanisms on both Resnet and Mobilenet indicates that our method results in higher accuracy.


## Related work

Many prior works attempt to improve the performance of neural networks by suppressing insignificant weights. Squeeze-and-Excitation Networks (SENet) (hu2018squeeze) integrate the spatial information into channel-wise feature responses and compute the corresponding attention with two multi-layer-perceptron (MLP) layers. Later, Bottleneck Attention Module (BAM) (park2018bam) builds separated spatial and channel submodules in parallel and they can be embedded into each bottleneck block. Convolutional Block Attention Module (CBAM) (woo2018cbam) provides a solution that embeds the channel and spatial attention submodules sequentially. To avoid the ignorance of cross-dimension interactions, Triplet Attention Module (TAM) (misra2021rotate) takes account of dimension correlations by rotating the feature maps. However, these works neglect information from the tuned weights from training. Therefore, we aim to highlight salient features by utilizing the variance measurement of the trained model weights.

## Methodology

We propose NAM as an efficient and lightweight attention mechanism. We *adopt the module integration from CBAM* ([2]) and redesign the channel and spatial attention submodules. Then, a NAM module is embedded at the end of each network block. For residual networks, it is embedded at the end of the residual structures. For the **channel attention** submodule, we use a scaling factor from batch normalization (BN) ([1]), as shown in Equation (1). The scaling factor measures the variance of channels and indicates their importance.

$$B_{out} = BN(B_{in}) = \gamma \frac{B_{in} - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} + \beta \tag{1}$$

where $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ are the mean and standard deviation of mini batch $\mathcal{B}$, respectively; $\gamma$ and $\beta$ are trainable affine transformation parameters (scale and shift) ([1]). The channel attention submodule is shown in Figure 1 and Equation (2), where $\mathbf{M}_c$ represents the its output features. $\gamma$ is the scaling factor for each channel, and the weights are obtained as $W_\gamma = \gamma_i / \sum_{j=0} \gamma_j$.

We also apply a scaling factor of BN to the spatial dimension to measure the importance of pixels. We name it pixel normalization. As a result, the **spatial attention** submodule is designed as shown in Figure 2 and Equation (3), where the output is denoted as $\mathbf{M}_s$. $\lambda$ is the scaling factor, and the weights are $W_\lambda = \lambda_i / \sum_{j=0} \lambda_j$.

To suppress the less salient weights, we add a regularization term into the loss function, as shown in Equation (4), where $x$ denotes the input; $y$ is the output; $W$ represents network weights; $l(\cdot)$ is the loss function; $g(\cdot)$ is the $l_1$ norm penalty function; $p$ is the penalty that balances $g(\gamma)$ and $g(\lambda)$.

$$\mathbf{M}_c = sigmoid(W_\gamma(BN(\mathbf{F}_1))) \tag{2}$$

$$\mathbf{M}_s = sigmoid(W_\lambda(BN_s(\mathbf{F}_2))) \tag{3}$$

$$Loss = \sum_{(x,y)} l(f(x, W), y) + p \sum g(\gamma) + p \sum g(\lambda) \tag{4}$$

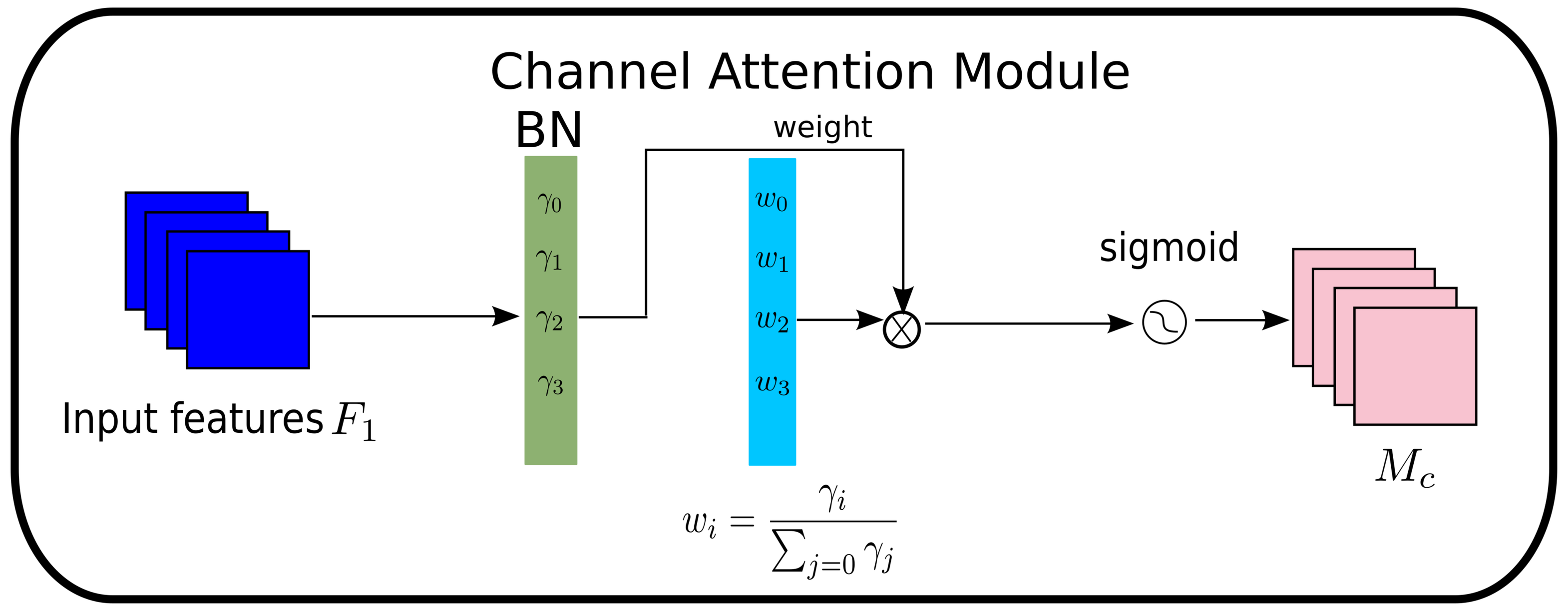


Figure 1. Channel attention mechanism

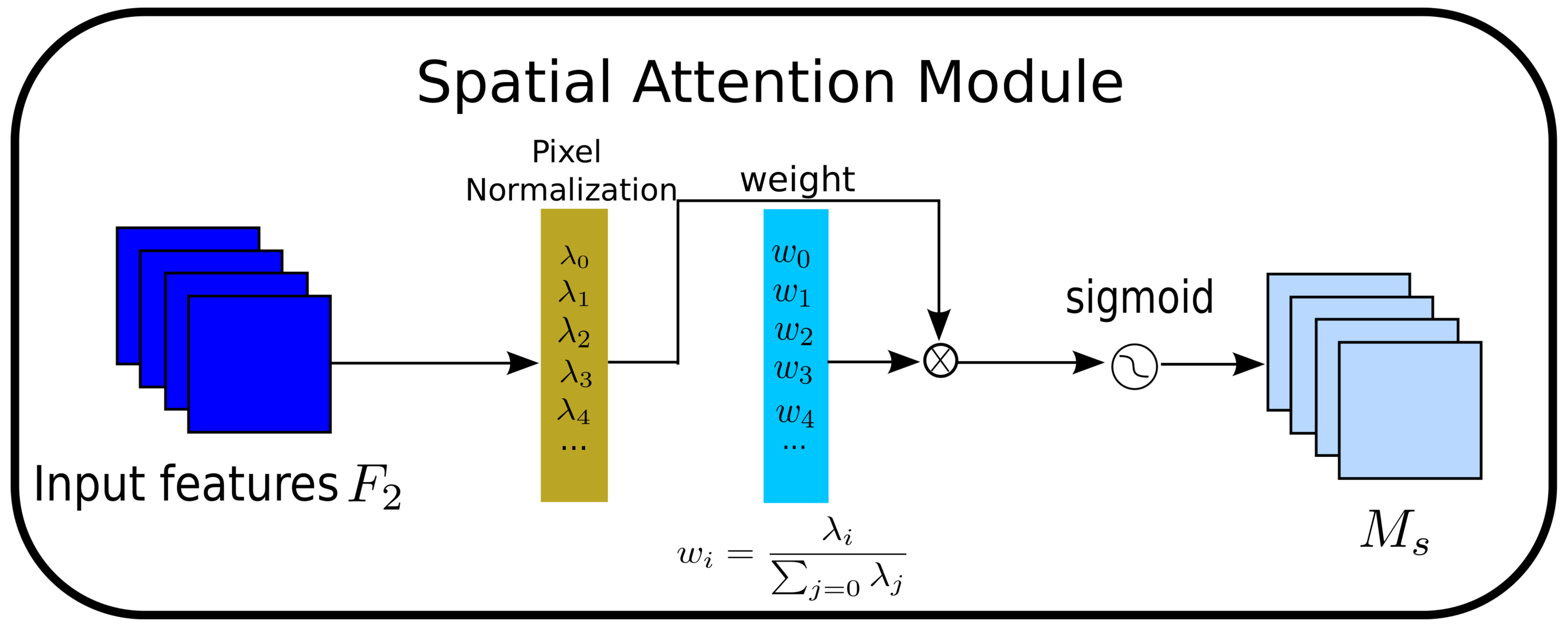


Figure 2. Spatial attention mechanism

## Experiment

Table 1. Classification results on Cifar100

| Architecture | Parameters | FLOPs | Top-1 Error (%) | Top-5 Error (%) |
|---|---|---|---|---|
| ResNet 50 | 23.71M | 1.30G | 22.74 | 6.37 |
| ResNet 50 + SE | 26.22M | 1.31G | 20.29 | 5.18 |
| ResNet 50 + BAM | 24.06M | 1.33G | 19.97 | 5.03 |
| ResNet 50 + CBAM | 26.24M | 1.31G | 19.44 | 4.66 |
| ResNet 50 + TAM | 23.71M | 1.33G | 20.15 | 5.13 |
| ResNet 50 + NAM(ch*) | 23.74M | 1.31G | 19.09 | 4.5 |
| ResNet 50 + NAM(sp*) | 23.71M | 1.31G | 19.38 | 4.72 |

⋆ ch stands for channel attention only; sp indicates spatial attention only.

Table 2. Classification results on ImageNet

| Architecture | Parameters | FLOPs | Top-1 Error (%) | Top-5 Error (%) |
|---|---|---|---|---|
| MobileNet V2 | 3.51M | 0.31G | 30.52 | 11.20 |
| MobileNet V2 + SE | 3.53M | 0.32G | 29.77 | 10.65 |
| MobileNet V2 + BAM | 3.54M | 0.32G | 29.91 | 10.80 |
| MobileNet V2 + CBAM | 3.54M | 0.32G | 29.74 | 10.66 |
| MobileNet V2 + NAM | 3.51M | 0.32G | 29.34 | 10.18 |

## Conclusion

We proposed a NAM module that is more efficient by suppressing the less salient features. Our experiments indicate that NAM provides efficiency gain on both ResNet and MobileNet. We are conducting detailed examination on NAM and adjusting its integration and hyper-parameters. In the future, we plan to investigate NAM on other deep learning architectures and applications. We also plan to optimize NAM with different model compression techniques, which may promote efficiency on recent larger network architectures.

## References


[1] 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.

[2] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.



35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.
workshop of ImageNet: past, present, and future



y.liu@hzdr.de