# WeightAlign: Normalizing Activations by Weight Alignment

Xiangwei Shi*, Yunqiang Li*, Xin Liu* and Jan van Gemert
Computer Vision Lab, TU Delft, Netherlands

## Introduction

BatchNorm (BN) stabilizes network optimization by normalizing the activations during training and exploits mini-batch sample statistics. BN unfortunately suffers from performance degradation when the statistical estimates become unstable for small batch-size based tasks. This paper we propose WeightAlign: normalizing activations without using sample statistics. Instead of sample statistics, we re-parameterize the weights within a filter to arrive at correctly normalized activations.

## Proposed Method

**Batch Normalization (BN)**:

$$\hat{\mathbf{x}} = \frac{\mathbf{x} - \mu_\beta}{\sigma_\beta}, \quad \mathbf{r} = \gamma\hat{\mathbf{x}} + \beta, \tag{1}$$

where $\mu_\beta$ and $\sigma_\beta$ are functions of *sample statistics* of input features $\mathbf{x}$ in a single channel, and $\gamma$, $\beta$ are a pair of trainable parameters.

**Expressing statistics via weights:** The mean and variance of activation $\mathbf{x}$ can be represented via filter weights,

$$\mu_\beta = E[x] = nE[w]E[Y], \tag{2}$$

$$\sigma^2_\beta = \mathrm{Var}[x] = n\big(E[w^2]E[Y^2] - E^2[w]E^2[Y]\big) \tag{3}$$

where $x$, $Y$ and $w$ present random variables of $\mathbf{x}$, input activations $\mathbf{Y}$ and filter $\mathbf{w}$. The $n = k^2c$ denotes number of weights in a filter.

**WeightAlign (WA):** We expect to have zero mean in Eq.(2) and unit variance in Eq.(3), then,

$$E[w] = 0, \quad \frac{1}{2}n\mathrm{Var}[w] = 1. \tag{4}$$

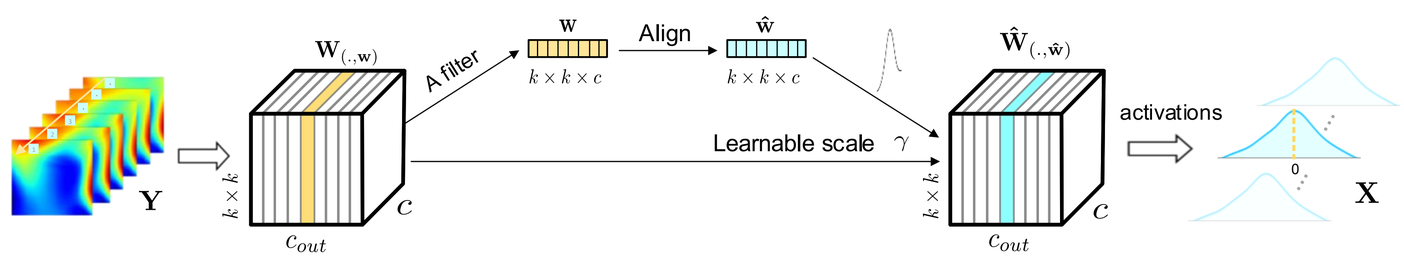
We reparameterize a single filter weights to have zero mean and a standard deviation $\sqrt{2/n}$ that,

$$\hat{w} = \gamma \, \frac{w - E[w]}{\sqrt{n/2 \cdot \mathrm{Var}[w]}}, \tag{5}$$

where $\gamma$ is a learnable scalar parameter.

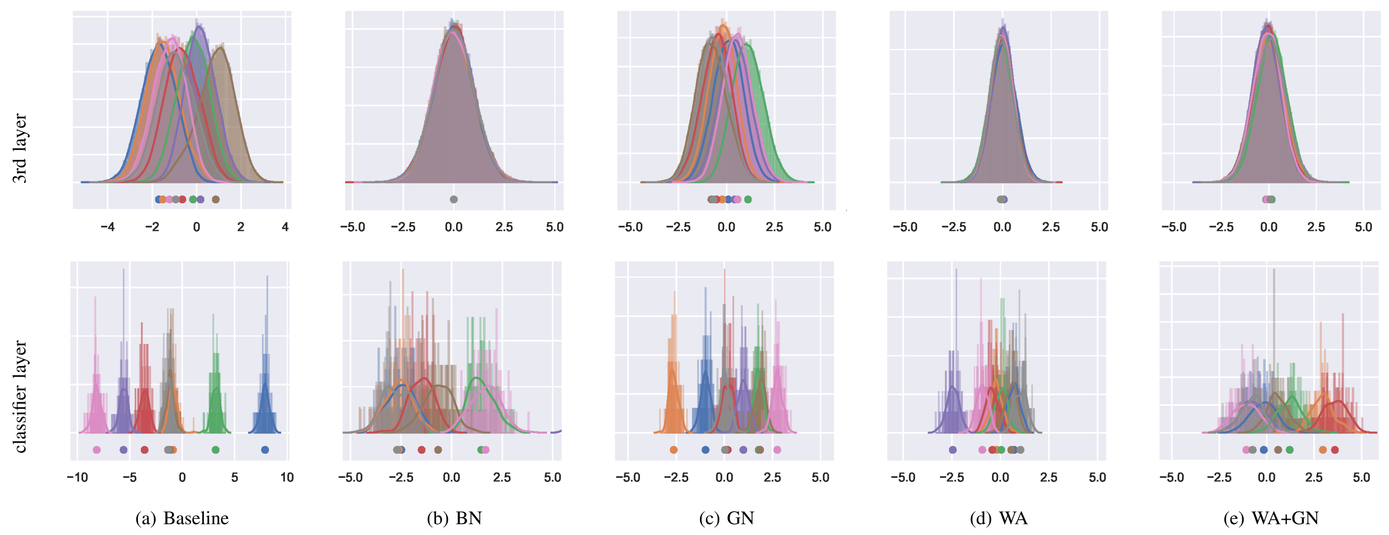
## Pipeline

Overview of WeightAlign (WA): Aligning filter weights allows normalizing channel activations.




## Empirical analysis and examples

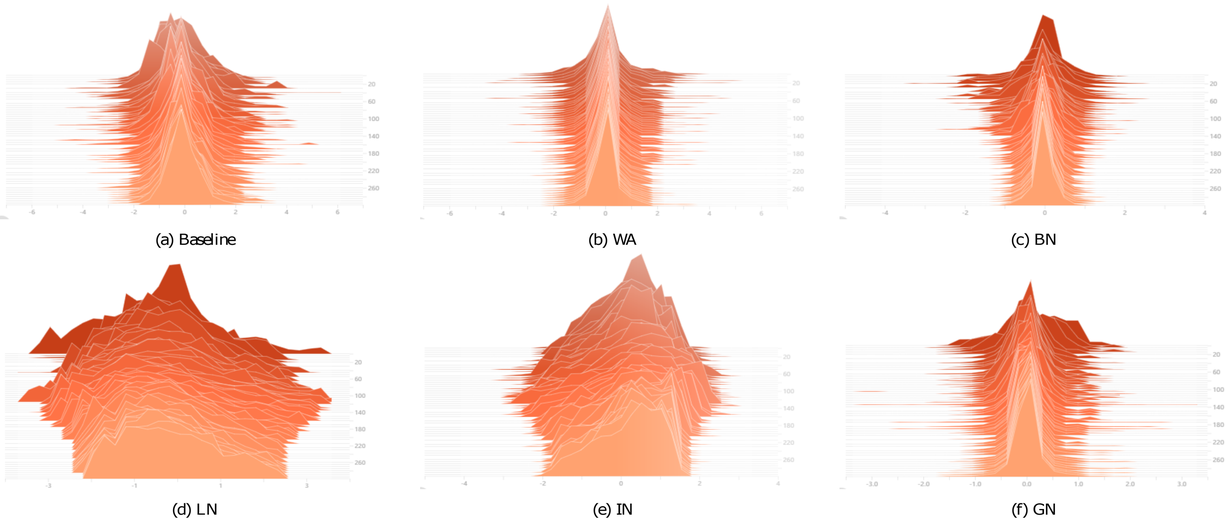
Each color represents the activation distribution of different channels for two different layers. For baseline model, the 'Blue' indexed channel will dominate all other channels, leading to a constant classification result. Our WA method can avoid the constant output as the effect of adding activation normalization layer, e.g. BN and GN.




(a) Baseline (b) BN (c) GN (d) WA (e) WA+GN

## Experiment Results

### i). Visualization of single channel activation in training



(a) Baseline (b) WA (c) BN (d) LN (e) IN (f) GN

### ii).Comparing normalization methods.

| CIFAR-10 | | | | | | | |
|---|---|---|---|---|---|---|---|
| Batch size 64 | | | | Batch size 1 | | | |
| Method | Error | Method | Error | Method | Error | Method | Error |
| Baseline [36] | 6.46 | WA | 6.21 | Baseline | 7.27 | WA | 6.61 |
| BN [23] | 4.30 | BN+WA | 4.29 | BN | - | BN+WA | - |
| IN [20] | 6.49 | IN+WA | 6.42 | IN | 6.91 | IN+WA | 6.50 |
| LN [19] | 5.02 | LN+WA | 5.12 | LN | 6.82 | LN+WA | 5.76 |
| GN [21] | 4.96 | GN+WA | 4.60 | GN | 5.79 | GN+WA | 5.51 |

### iii).Depth of residual networks

| CIFAR-10 | | |
|---|---|---|
| Model (+WA) | Batchsize 1 Error | Batchsize 64 Error |
| ResNet18 | 5.65 | 6.23 |
| ResNet34 | 5.84 | 5.78 |
| ResNet50 | 6.61 | 6.42 |
| ResNet101 | 5.74 | 6.41 |
| ResNet152 | 6.09 | 6.52 |

### iv).Different components

| Components in WA | | VGG16 | ResNet18 |
|---|---|---|---|
| $\mathbb{E}[w] = 0$ | $\mathrm{Var}[w] = \frac{2}{n}$ | Error | Error |
| × | × | - | 28.23 |
| ✓ | × | 35.74 | 27.97 |
| × | ✓ | - | 28.03 |
| ✓ | ✓ | 27.85 | 24.92 |

### v).Image classification on ImageNet

| model | Top-1 (%) Error | Top-5 (%) Error |
|---|---|---|
| VGG16* (Baseline) | 31.30 | 11.19 |
| VGG16 (BN)* | 29.58 | 10.16 |
| VGG16 (WA) | 29.78 | 10.23 |
| VGG16 (BN+WA) | **27.07** | **8.78** |
| ResNet50 (Baseline)† [36] | 27.60 | - |
| ResNet50 (BN)* | 24.89 | 7.71 |
| ResNet50 (WA) | 26.62 | 8.91 |
| ResNet50 (BN+WA) | **24.04** | **7.12** |

## Conclusions

i). We propose WeightAlign that re-parameterizes the weights by the mean and scaled standard derivation computed within a filter. ii). We experimentally demonstrate WeightAlign on five different datasets. iii). WeightAlign can be combined with other activation normalization methods (e.g., BN, GN, LN and IN) and consistently improves their performance.