# Channel Equilibrium Networks for Learning Deep Representation

Wenqi Shao,

VALSE, July, 28, 2021

The Chinese University of Hong Kong

# • Overview

- Convolutional Neural Networks (CNNs) facilitate various applications in computer vision

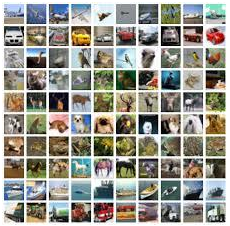
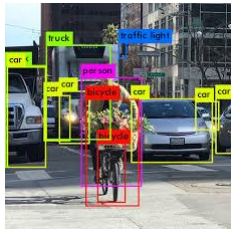


Image Classification




Object Detection

Semantic Segmentation

Person ReID

- AlexNet, ResNet, ResNeXt, Wide-ResNet, MobileNet …

***over-parameterized*** network architectures

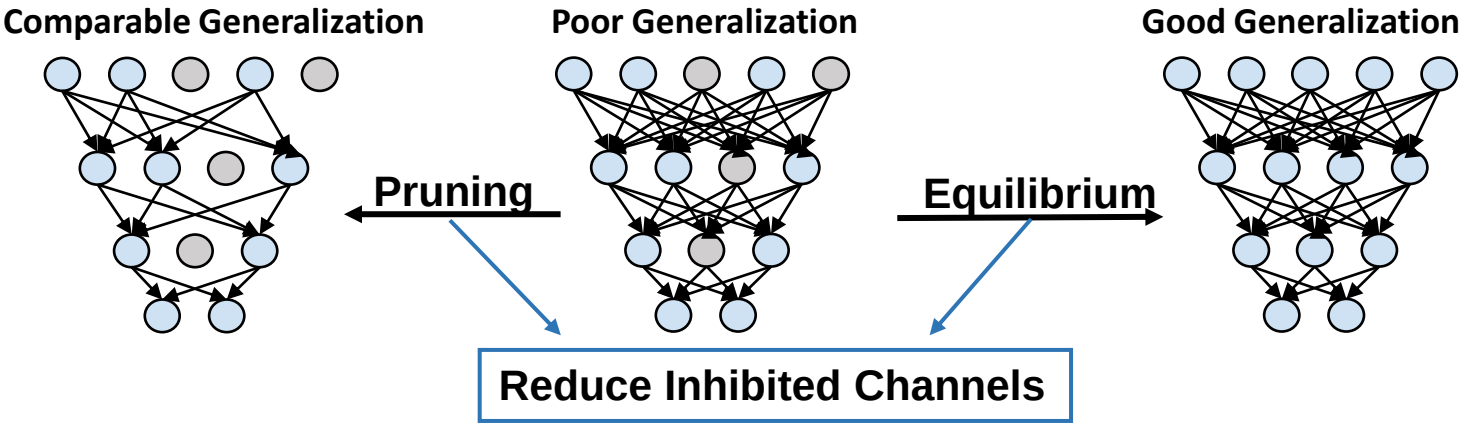
- Observation.

◆ An over-parameterized CNN always contains **unimportant ('inhibited') channels** whose feature values are extremely small.

◆ Some channels in a well-trained CNN can easily be **pruned without significant loss** of performance.

- Overview

- A network's reliance on single directions is a good predictor of generalization performance. [1] ***A network generalizing well often rely less on single direction (neuron, channel etc.).***




**A natural idea**: Why not wake up those 'inhibited' channels?

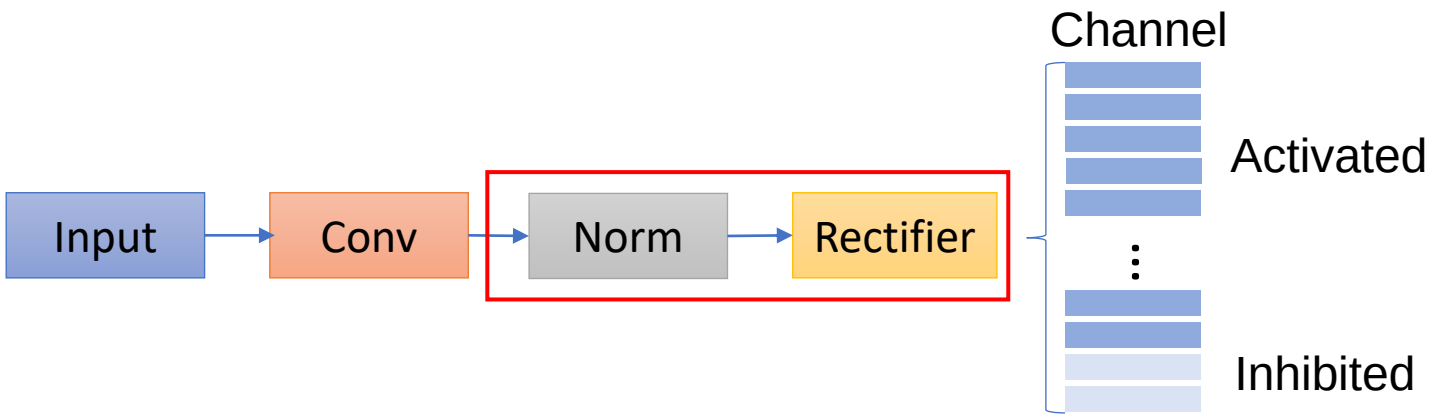
**inhibited channels** ↓ → **reliance on single directions** ↓ → **generalization** ↑

[1] Ari S. Morcos, et al. ON THE IMPORTANCE OF SINGLE DIRECTIONS FOR GENERALIZATION. ICLR'18

# Overview

- How inhibited channels are related with BN
- How whitening method can avoid channel collapse Channel – CE block
- Connection between CE and Nash Equilibrium
- Experimental results
- Conclusion

- Inhibited Channels – where and how?
  - **Where:** Normalization and Rectifier block




  - **Norm:** BatchNorm, InstanceNorm, GroupNorm, LayerNorm…
    a. reduce covariance shift
    b. stabilize and accelerate the training process
  - **Rectifier:** ReLU, Exponential Linear Unit (ELU), Leaky ReLU (LReLU)…
    a. highly non-linear representation,
    b. allow faster and effective training of deep neural architectures
  - **'Norm+Rectifier'** block leads to inhibited channels.

- Inhibited Channels – where and how?

  - **How:** small feature values which contribute little to the learned feature representation

  - **Formulation:** $\boldsymbol{x} \in \boldsymbol{R}^{N\times C\times H\times W}$ represents the feature map in a layer,

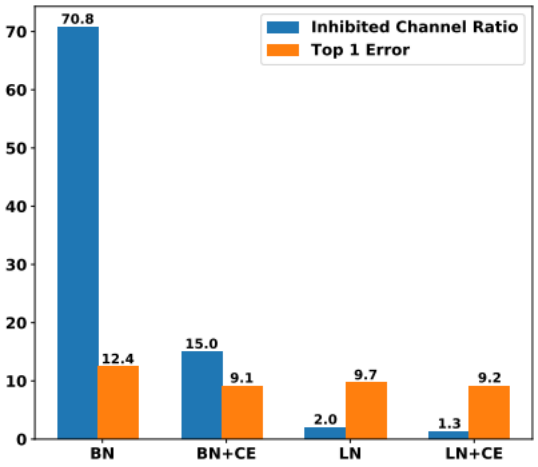
    $x_{ncij}$ denotes a element of $\boldsymbol{x}$.

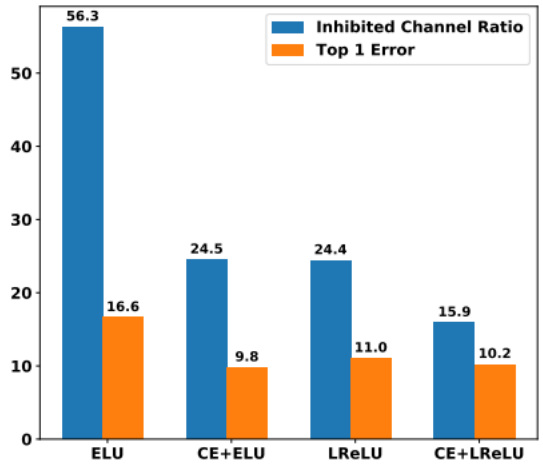
| | |
|---|---|
| **Norm** | $\tilde{x}_{ncij} = \gamma_c \bar{x}_{ncij} + \beta_c, \quad \bar{x}_{ncij} = (x_{ncij} - \mu_k)/\sigma_k.$ |
| **Rectifier** | $y_{ncij} = g(\tilde{x}_{ncij}),$ |

  - Inhibited channels emerge when $\gamma$ and $\beta$ satisfy the condition in Remark 1

**Remark 1.** *Let a random variable* $z \sim \mathcal{N}(0,1)$ *and* $y = max\{0, \gamma_c z + \beta_c\}$. *Then we have* $\mathbb{E}_z[y] = 0$ *and* $\mathbb{E}_z[y^2] = 0$ *if and only if* $\beta_c \leq 0$ *and* $\gamma_c$ *sufficiently approaches* $0$.

- Inhibited Channels – Empirical Observation




- Inhibited channels emerge in various 'Norm+Rectifier' blocks
- Condition of inhibited channels in various ResNets trained on the ImageNet dataset.

| CNNs | ResNet18 | ResNet50 | ResNet101 |
|---|---|---|---|
| $(\gamma_c < 0.1)$ | 1.2 | 18.8 | 21.4 |

*Table 9.* Ratios of $(\gamma_c < 0.1)$ after traing on various CNNs.

| CNNs | ResNet18 | ResNet50 | ResNet101 |
|---|---|---|---|
| $(\beta_c \leq 0)$ | 76.0 | 76.7 | 81.8 |

*Table 4.* Ratios of $(\beta_c \leq 0)$ after traing on various CNNs.

# • Channel Equilibrium Block

- We use **decorrelation** method that decorrelate each output of normalization to achieve **channel equilibrium (CE)**.
- **Formulation** of CE: $\boldsymbol{p}_{nij} = \boldsymbol{D}_n^{-\frac{1}{2}}(\mathrm{Diag}(\boldsymbol{\gamma})\bar{\boldsymbol{x}}_{nij} + \boldsymbol{\beta})$ , $\boldsymbol{D}_n^{-\frac{1}{2}}$ is a decorrelation operator
- **Construction** **Goal:** better model the dependencies between channels

$$\boldsymbol{D}_n = \lambda\boldsymbol{\Sigma} + (1-\lambda)\mathrm{Diag}(\boldsymbol{v}_n), \quad \boldsymbol{v}_n = f(\tilde{\boldsymbol{\sigma}}_n^2),$$

$\boldsymbol{\Sigma}$, Covariance matrix estimated by a batch of sample

$\boldsymbol{v_n}$, Adaptive instance variance computed by the current sample

- **Approximation**

$$\begin{aligned}\boldsymbol{D}_n^{-\frac{1}{2}} &= [\lambda\boldsymbol{\Sigma} + (1-\lambda)\mathrm{Diag}(\boldsymbol{v}_n)]^{-\frac{1}{2}} \\ &\preceq \lambda \underbrace{\boldsymbol{\Sigma}^{-\frac{1}{2}}}_{\text{batch decorrelation}} + (1-\lambda)\underbrace{[\mathrm{Diag}(\boldsymbol{v}_n)]^{-\frac{1}{2}}}_{\text{instance reweighting}},\end{aligned}$$

High-order terms are controlled by the spectral of $\boldsymbol{\Sigma}$ and $\boldsymbol{v_n}$

- **Decomposition**

CE is decomposed into two branches,
Batch Decorrelation (**BD**) and instance reweighting (**IR**)

# • Channel Equilibrium Block

- BD branch computes $\boldsymbol{\Sigma}^{-\frac{1}{2}}$, by Newton's iteration

$$\boldsymbol{\Sigma} = \boldsymbol{\gamma}\boldsymbol{\gamma}^{\mathsf{T}} \odot \frac{1}{M}\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^{\mathsf{T}}, \qquad \text{covariance matrix after BN layer}$$

$$\begin{cases} \boldsymbol{\Sigma}_0 = \boldsymbol{I} \\ \boldsymbol{\Sigma}_k = \frac{1}{2}(3\boldsymbol{\Sigma}_{k-1} - \boldsymbol{\Sigma}_{k-1}^3\boldsymbol{\Sigma}),\ k = 1, 2, \cdots, T. \end{cases} \qquad \text{Newton's iteration}$$

- **BD branch** firstly calculates a normalized covariance matrix and then applies Newton's Iteration to obtain its inverse square root

- IR branch calculates $\mathrm{Diag}[\boldsymbol{f}(\tilde{\boldsymbol{\sigma}}_{\boldsymbol{n}}^2)]^{-\frac{1}{2}}$

$$\tilde{\boldsymbol{\sigma}}_n^2 = \mathrm{diag}(\boldsymbol{\gamma}\boldsymbol{\gamma}^{\mathsf{T}}) \odot \frac{(\boldsymbol{\sigma}_{\mathrm{IN}}^2)_n}{\boldsymbol{\sigma}_{\mathrm{BN}}^2}, \qquad \text{instance variance after BN layer}$$

$$[\mathrm{Diag}(\boldsymbol{v}_n)]^{-\frac{1}{2}} = \mathrm{Diag}(\mathrm{Sigmoid}(\tilde{\boldsymbol{\sigma}}_n^2; \boldsymbol{\theta})) \cdot s^{-\frac{1}{2}}, \qquad \text{reparameterization with attention}$$

- **IR branch** uses a reparameterization trick , where an attention mechanism with Sigmoid activation is used to control the strength of the inverse square root of variance for each channel

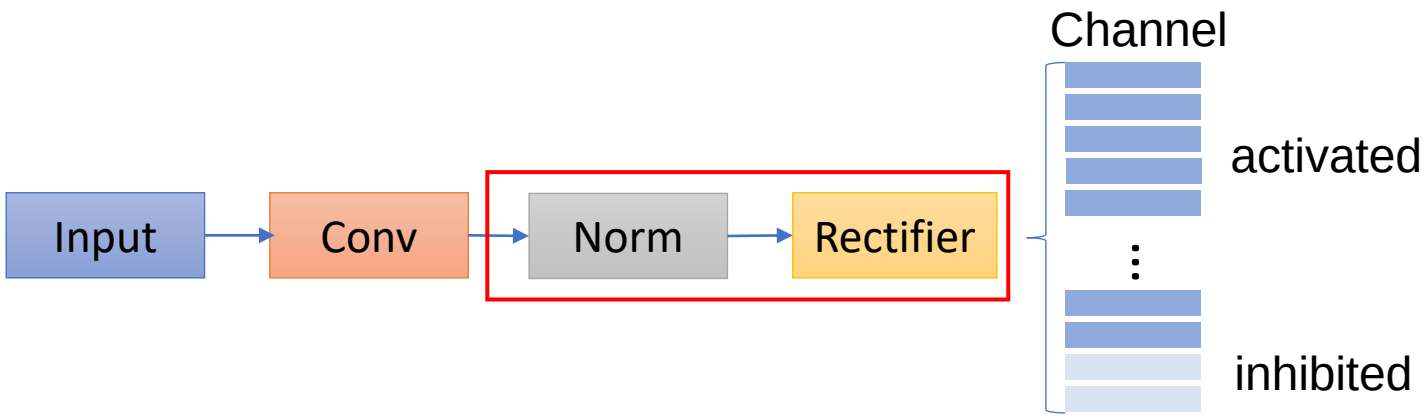
- Channel Equilibrium Block
  - BD branch increases the magnitude of gamma and feature channels.
  - **CE:** $\boldsymbol{p}_{nij} = \boldsymbol{D}_n^{-\frac{1}{2}}(\mathrm{Diag}(\boldsymbol{\gamma})\bar{\boldsymbol{x}}_{nij} + \boldsymbol{\beta})$
  - **Decomposition** $\boldsymbol{D}_n^{-\frac{1}{2}} = [\lambda\boldsymbol{\Sigma} + (1-\lambda)\mathrm{Diag}(\boldsymbol{v}_n)]^{-\frac{1}{2}}$
    $\preceq \lambda \underbrace{\boldsymbol{\Sigma}^{-\frac{1}{2}}}_{\text{batch decorrelation}} + (1-\lambda)\underbrace{[\mathrm{Diag}(\boldsymbol{v}_n)]^{-\frac{1}{2}}}_{\text{instance reweighting}},$
  - **BD:** $\boldsymbol{p}_{nij}^{\mathrm{BD}} = \mathrm{Diag}(\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\gamma})\bar{x}_{nij} + \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\beta}$
  - **Equivalent gamma in BD:** $\hat{\boldsymbol{\gamma}} = \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\gamma}.$

**Proposition 1.** *Let* $\boldsymbol{\Sigma}$ *be covariance matrix of feature maps after batch normalization. Then, (1) assume that* $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}^{-\frac{1}{2}}, \forall k = 1, 2, 3, \cdots, T$, *we have* $|\hat{\gamma}_c| > |\gamma_c|, \forall c \in [C]$. *(2) Denote* $\boldsymbol{\rho} = \frac{1}{M}\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^{\mathsf{T}}$ *in Eqn.(5) and* $\tilde{\boldsymbol{x}}_{nij} = \mathrm{Diag}(\boldsymbol{\gamma})\bar{\boldsymbol{x}}_{nij} + \boldsymbol{\beta}$. *Assume* $\boldsymbol{\rho}$ *is full-rank, then* $\left\|\boldsymbol{\Sigma}^{-\frac{1}{2}}\tilde{\boldsymbol{x}}_{nij}\right\|_2 > \|\tilde{\boldsymbol{x}}_{nij}\|_2$

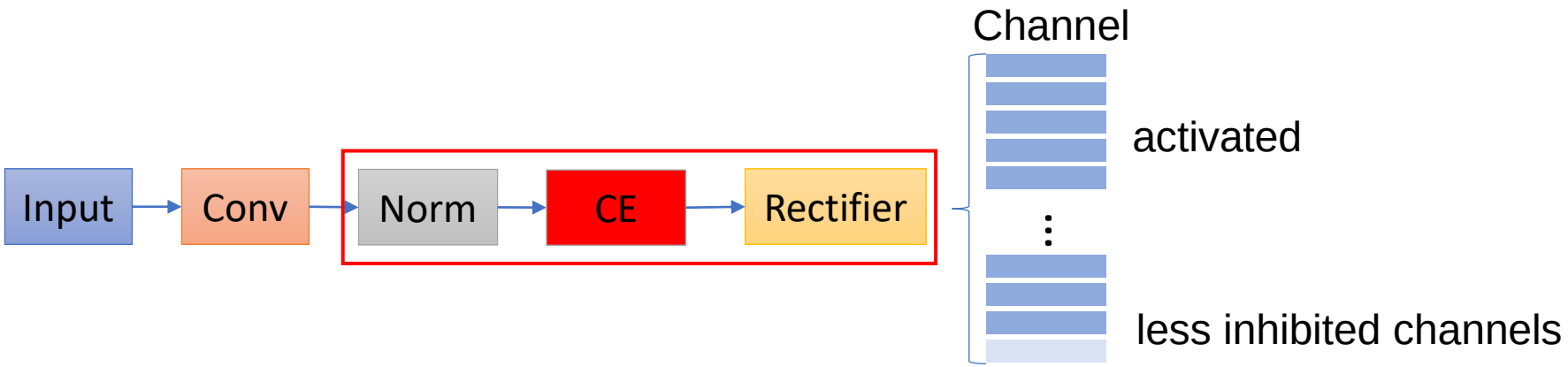
magnitude of gamma for every channel increases

prevent inhibited channels!

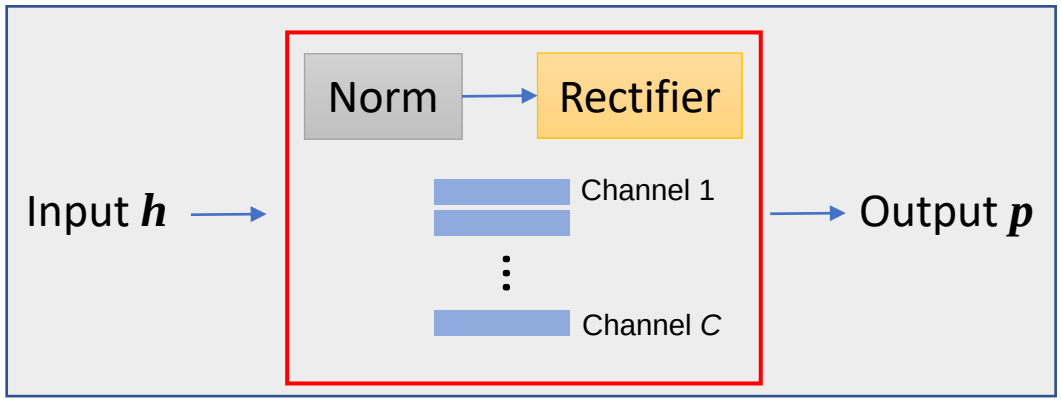
- Channel Equilibrium Block
  - Integration Strategy of **CE**




  - **before rectifier and after normalization**



- Connection between CE and Nash Equilibrium




Objective: Each channels wants to maximize its own benefit

Non-cooperative game!

**Formulation**

- Objective
- Limited budget
- Non-negative output

$$\max \ \mathcal{C}_c(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_C) = \sum_{i,j=1}^{h,W} \ln\left(1 + \frac{g_{cc}p_{cij}}{\sum_{d\neq c} g_{cd}p_{dij} + \sigma_c/h_{cij}}\right)$$

$$s.t. \quad \begin{cases} \sum_{i,j=1}^{H,W} p_{cij} = P_c, \\ p_{cij} \geq 0, \end{cases} \qquad \forall i \in [H], j \in [W]$$

Allows unique Nash Equilibrium

# • Connection between CE and Nash Equilibrium

- When all channels are activated, we have **Nash Equilibrium Point** by Taylor expansion

$$\boldsymbol{p}_{ij}^{*} = \boldsymbol{G}^{-1}\left(\mathrm{Diag}(\boldsymbol{\sigma})\bar{\boldsymbol{h}}_{ij} + \mathrm{Diag}(\boldsymbol{v}_0)^{-1}\mathrm{diag}(\boldsymbol{G}) + (2+\boldsymbol{\delta})\boldsymbol{\sigma}\right)$$

- **CE:** $\boldsymbol{p}_{nij} = \boldsymbol{D}_n^{-\frac{1}{2}}\left(\mathrm{Diag}(\boldsymbol{\gamma})\bar{\boldsymbol{x}}_{nij} + \boldsymbol{\beta}\right)$

**The same form with the expression of CE.**

- We expect further investigation on modelling of relationship among channels by techniques in game theory

# • Experimental Results

- CE improves the generalization of various ResNets and mobile backbones with marginal increase of computational burden.

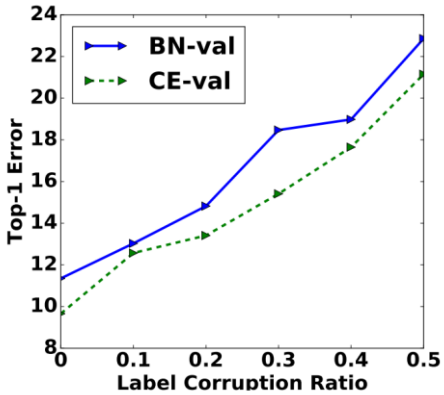
| | ResNet18 | | | ResNet50 | | | ResNet101 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Baseline | SE | CE | Baseline | SE | CE | Baseline | SE | CE |
| Top-1 | 70.4 | 71.4 | **71.9** | 76.6 | 77.6 | **78.3** | 78.0 | 78.5 | **79.0** |
| Top-5 | 89.4 | 90.4 | **90.8** | 93.0 | 93.7 | **94.1** | 94.1 | 94.1 | **94.6** |
| GFLOPs | 1.82 | 1.82 | 1.83 | 4.14 | 4.15 | 4.16 | 7.87 | 7.88 | 7.89 |
| CPU (s) | 3.69 | 3.69 | 4.13 | 8.61 | 11.08 | 11.06 | 15.58 | 19.34 | 17.05 |
| GPU (s) | 0.003 | 0.005 | 0.006 | 0.005 | 0.010 | 0.009 | 0.011 | 0.040 | 0.015 |

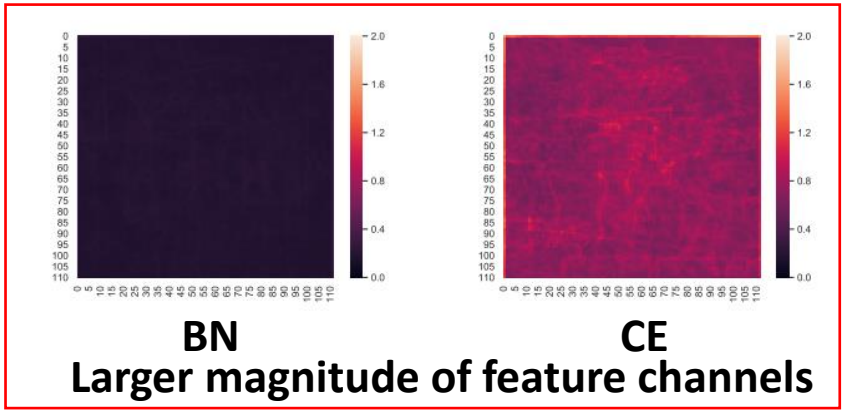
- CE outperforms the baseline BN-ResNets and SE-ResNets.

| | MobileNetv2 1× | | | ShuffleNetv2 0.5× | | | ShuffleNetv2 1× | | |
|---|---|---|---|---|---|---|---|---|---|
| | top-1 | top-5 | GFLOPs | top-1 | top-5 | GFLOPs | top-1 | top-5 | GFLOPs |
| Baseline | 72.5 | 90.8 | 0.33 | 59.2 | 82.0 | 0.05 | 69.0 | 88.6 | 0.15 |
| SE | 73.5 | 91.7 | 0.33 | 60.2 | 82.4 | 0.05 | 70.7 | 89.6 | 0.15 |
| CE | **74.6** | **91.7** | 0.33 | **60.5** | **82.7** | 0.05 | **71.2** | **89.8** | 0.16 |

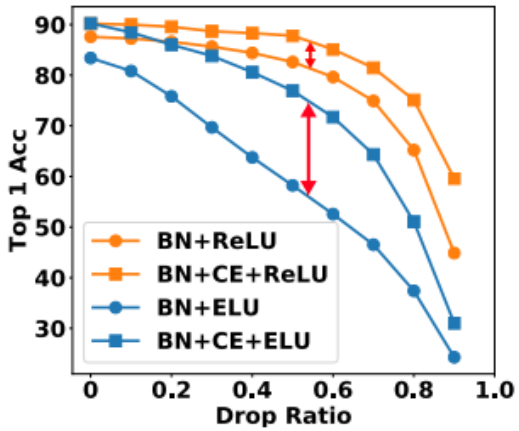
# • Experimental Results

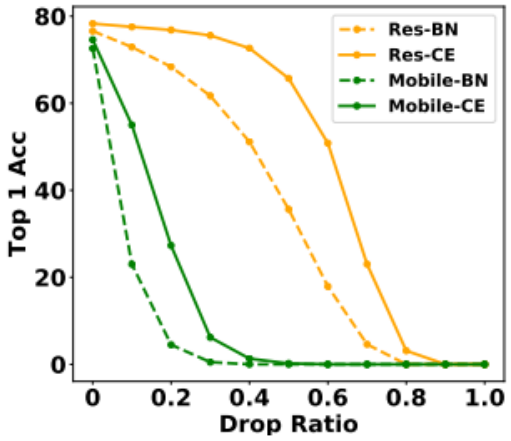
- **CE improves generalization ability in corrupted label setting.**




- **CE encourages channels to contribute more equally to the learned feature representation.**




**Cumulative Ablation Curves**

- CE presents a more gentle accuracy drop curve
- CE makes the network less reliant on specific channels.

# • Conclusion

- We presented an effective and efficient network block, termed as Channel Equilibrium (CE)

- CE encourages channels at the same layer to contribute equally to learned feature representation, enhancing the generalization ability of the network.

- We hope that the analyses of CE could bring a new perspective for future work in architecture design

# Thanks!