香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Batch Normalization Increases Adversarial Vulnerability and Decreases Adversarial Transferability: A Non-Robust Feature Perspective[1]

Rongjun Tang

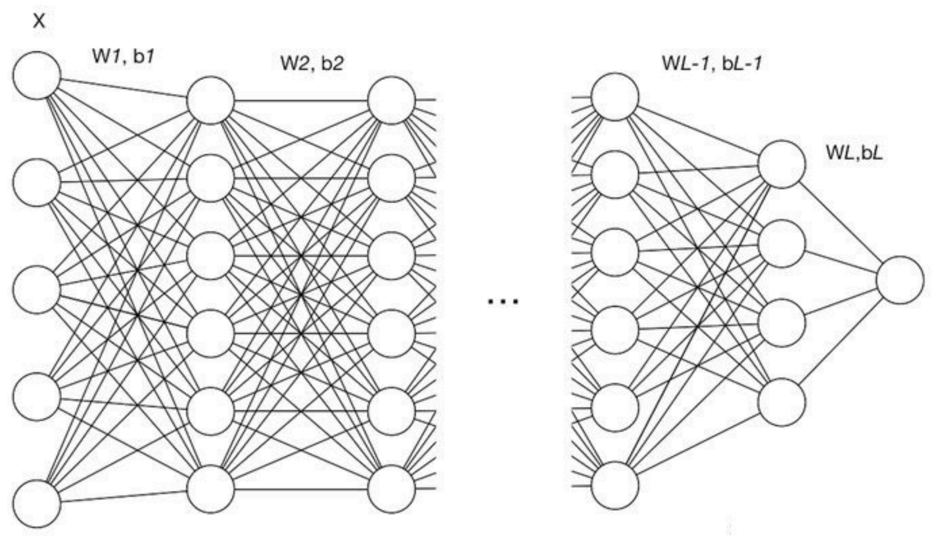
2021.10.20

[1] Benz, Philipp, Chaoning Zhang, and In So Kweon. "Batch Normalization Increases Adversarial Vulnerability and Decreases Adversarial Transferability: A Non-Robust Feature Perspective." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021..

Introduction

Why normalization?

Internal Covariate Shift (内部协变量偏移): the change in the distribution of network activations due to the change in network parameters during training, and leads to the poor learning speed of the whole network, even not converge.



$$Z^{[l]} = W^{[l]} \times input + b^{[l]}$$

$$A^{[l]} = g^{[l]}(Z^{[l]})$$

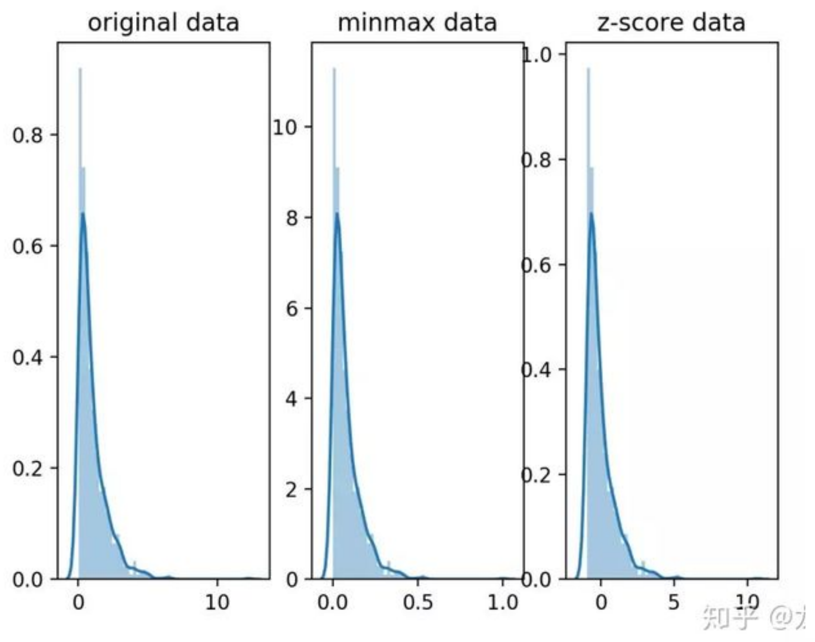
$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$$

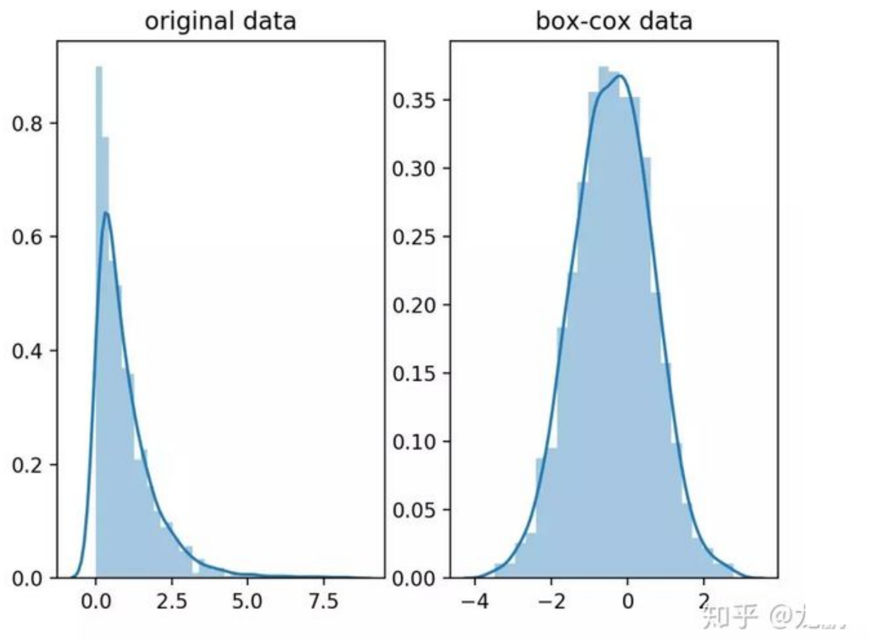
Lost in saturated regime.

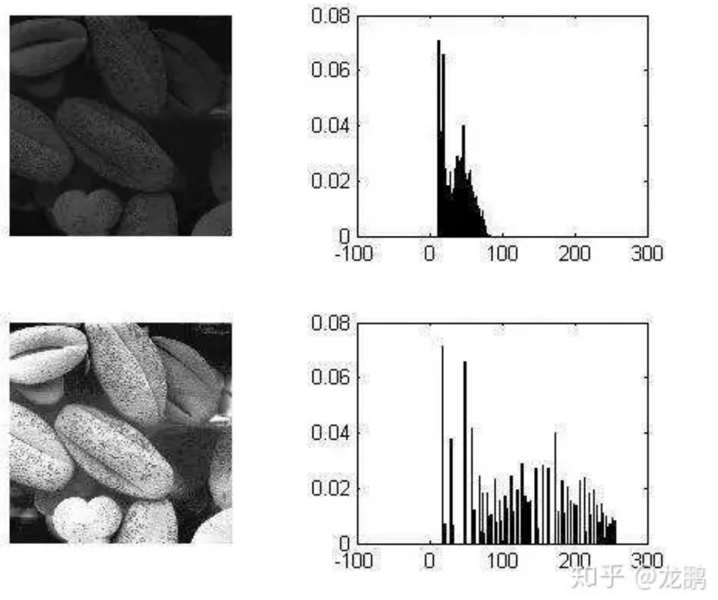
Introduction

What is batch normalization[2]?
Adjust the data distribution and then project back. An algorithmic method which makes the training of Deep Neural Networks (DNN) faster and more stable.

Some other normalization methods:
Min-max normalization; Z-score; Box-Cox transform; Histogram Equalization.







[2] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[C]//ICML. 2015

Whitening

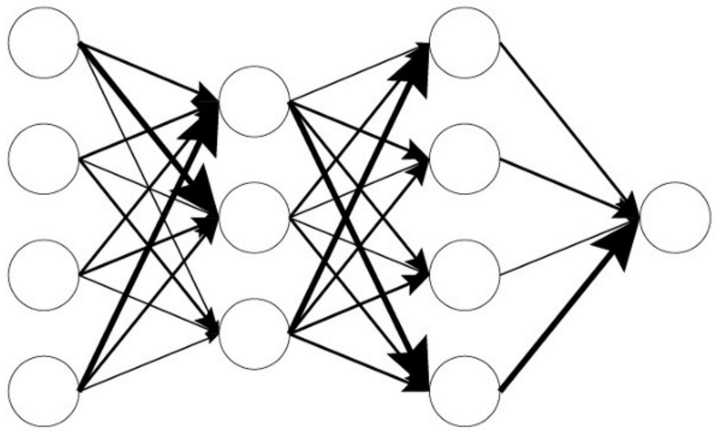
PCA and ZCA whitening: **make the input less redundant;** more formally, your training input will have the following characteristics: (i) the **features are less correlated with each other**, and (ii) **the features all have the same variance.**

However:
Large cost; Change the distribution of the layers in the network, thus lead to the information loss of the bottom layers.

Batch Normalization.

How to do Batch Normalization

Use a minibatch, and introduce two learnable variables which is not related to the previous layers: better for optimization.

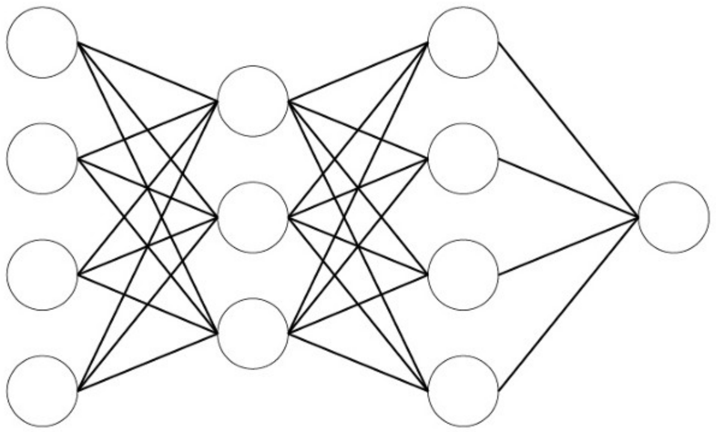


- **Raw** signal
- **High interdependancy** between distributions
- **Slow** and **unstable** training

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} Z_j^{(i)}$$

$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(Z_j^{(i)} - \mu_j)^2$$

$$\hat{Z}_j = \frac{Z_j - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$



- **Normalized** signal
- **Mitigated interdependancy** between distributions
- **Fast** and **stable** training

$$\tilde{Z}_j = \gamma_j \hat{Z}_j + \beta_j$$

Especially, when γ and β equal to σ and μ, we can achieve identity transform.

Batch Normalization: Extension

Instance Normalization, Layer Normalization, Group Normalization

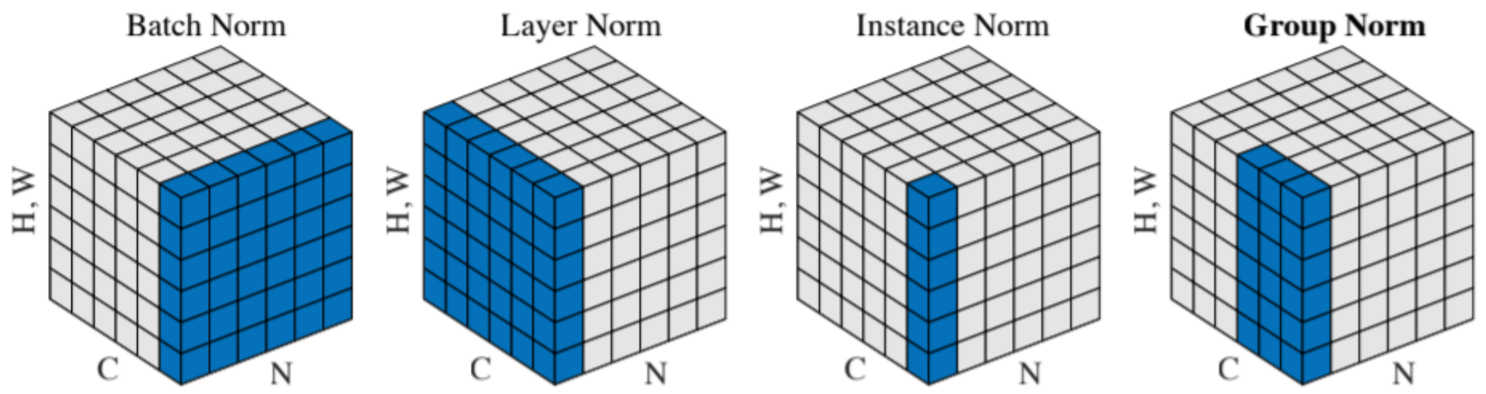


Figure 2. **Normalization methods**. Each subplot shows a feature map tensor, with N as the batch axis, C as the channel axis, and (H, W) as the spatial axes. The pixels in blue are normalized by the same mean and variance, computed by aggregating the values of these pixels.

$$\mu_i = \frac{1}{m} \sum_{k \in \mathcal{S}_i} x_k, \quad \sigma_i = \sqrt{\frac{1}{m} \sum_{k \in \mathcal{S}_i} (x_k - \mu_i)^2 + \epsilon}$$

A unified expression for BN, IN, LN and GN, where GN is the most general case.

$\mathcal{S}_i$ is the set of pixels in which the mean and std are computed.

Batch Normalization: Extension

BN在batch的维度上normalization，归一化维度为[N, H, W]，对batch中对应的channel归一化；

LN避开了batch维度，归一化的维度为[C, H, W]；

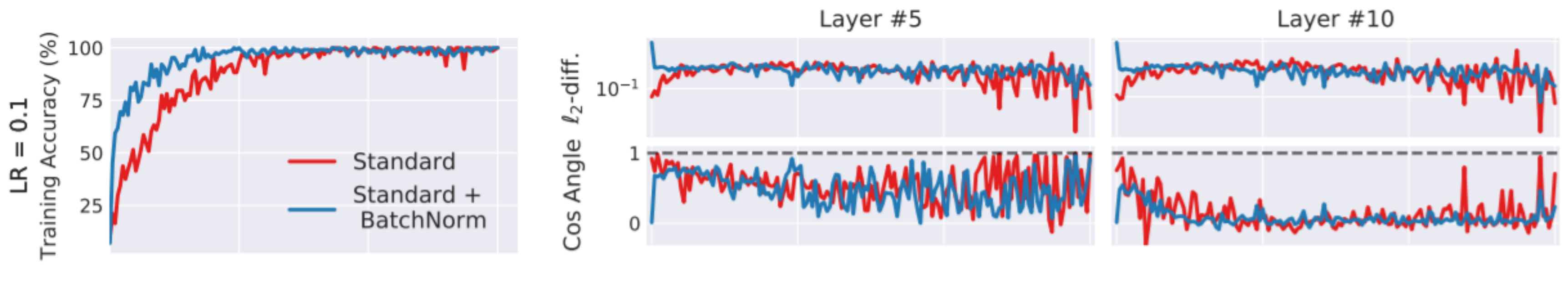
IN 归一化的维度为[H, W]；

而GN介于LN和IN之间，其首先将channel分为许多组（group），对每一组做归一化，即先将feature的维度由[N, C, H, W] reshape为 [N, G, C//G, H, W]，归一化的维度为[C//G, H, W]

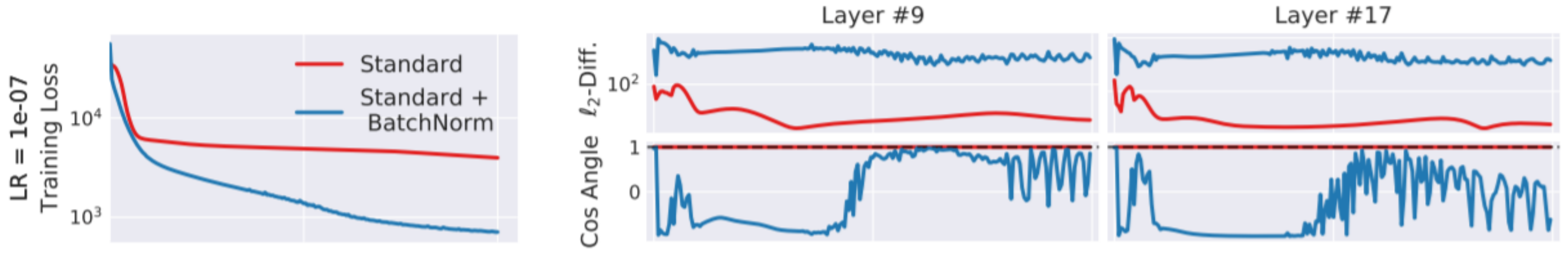
How Does Batch Normalization(IN, LN, GN) Work?

Not clear yet;

The original idea about Internal Covariate Shift may not explicitly explain how BN work[3].



(a) VGG



(b) DLN

The numerical proof of Lipschitz can help explain BN effect.

$$\left\|\nabla_{\boldsymbol{y}_j}\widehat{\mathcal{L}}\right\|^2 \leq \frac{\gamma^2}{\sigma_j^2}\left(\left\|\nabla_{\boldsymbol{y}_j}\mathcal{L}\right\|^2 - \frac{1}{m}\left\langle \mathbf{1}, \nabla_{\boldsymbol{y}_j}\mathcal{L}\right\rangle^2 - \frac{1}{\sqrt{m}}\left\langle \nabla_{\boldsymbol{y}_j}\mathcal{L}, \hat{\boldsymbol{y}}_j\right\rangle^2\right)$$

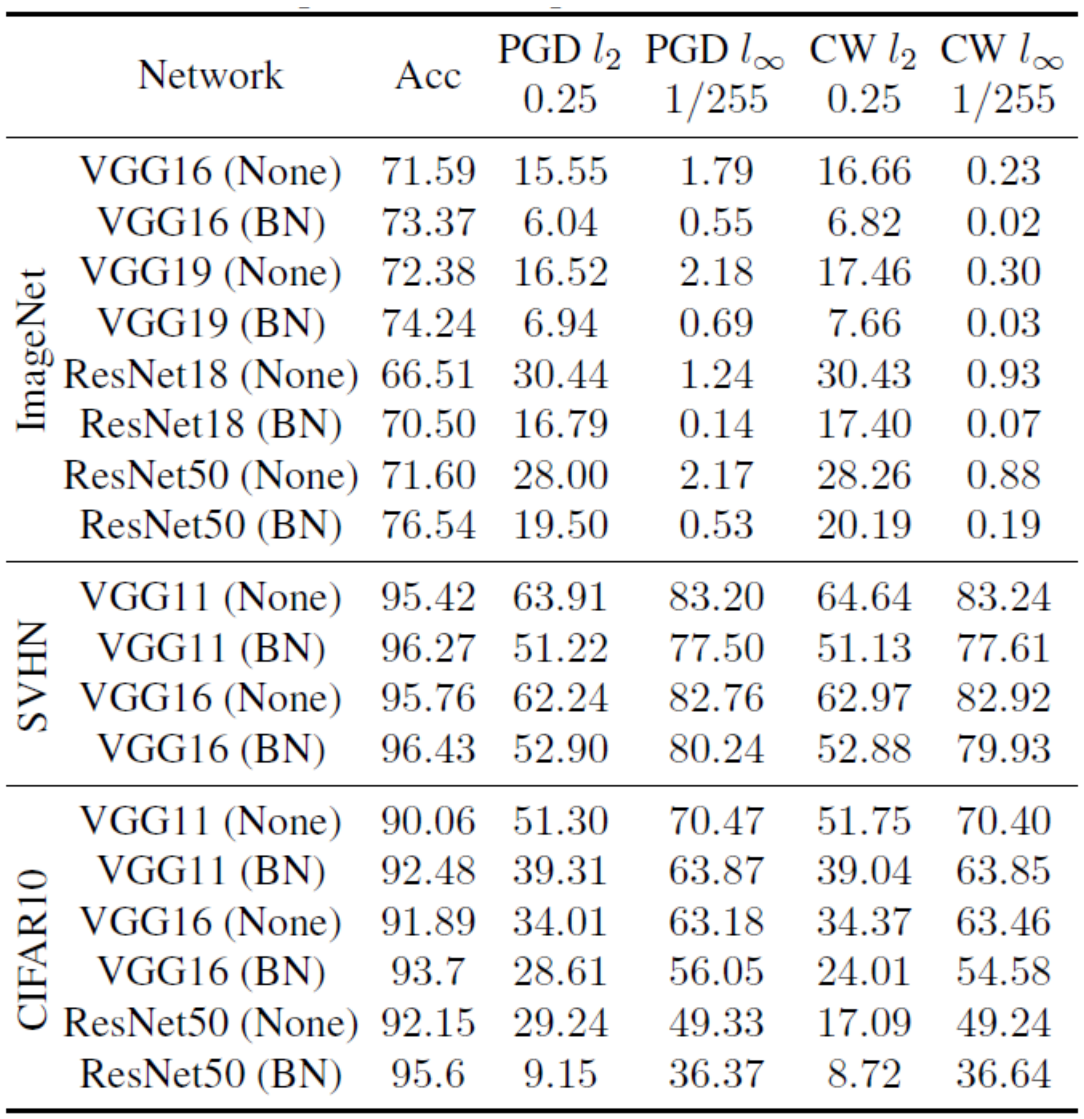
[3] Santurkar, Shibani, et al. "How does batch normalization help optimization?." Proceedings of the 32nd international conference on neural information processing systems. 2018.

Intuition: BN increases the clean accuracy

However, this comes at the cost of lower robust accuracy, i.e., accuracy on adversarial images.

DNN can learn robust features (RF) and non-robust features (NRF)[4], then which is BN's preferring one?

BN shifts the model to rely more on NRFs than RFs for classification.



	Network	Acc	PGD l_2 0.25	PGD l_∞ $1/255$	CW l_2 0.25	CW l_∞ $1/255$
ImageNet	VGG16 (None)	71.59	15.55	1.79	16.66	0.23
	VGG16 (BN)	73.37	6.04	0.55	6.82	0.02
	VGG19 (None)	72.38	16.52	2.18	17.46	0.30
	VGG19 (BN)	74.24	6.94	0.69	7.66	0.03
	ResNet18 (None)	66.51	30.44	1.24	30.43	0.93
	ResNet18 (BN)	70.50	16.79	0.14	17.40	0.07
	ResNet50 (None)	71.60	28.00	2.17	28.26	0.88
	ResNet50 (BN)	76.54	19.50	0.53	20.19	0.19
SVHN	VGG11 (None)	95.42	63.91	83.20	64.64	83.24
	VGG11 (BN)	96.27	51.22	77.50	51.13	77.61
	VGG16 (None)	95.76	62.24	82.76	62.97	82.92
	VGG16 (BN)	96.43	52.90	80.24	52.88	79.93
CIFAR10	VGG11 (None)	90.06	51.30	70.47	51.75	70.40
	VGG11 (BN)	92.48	39.31	63.87	39.04	63.85
	VGG16 (None)	91.89	34.01	63.18	34.37	63.46
	VGG16 (BN)	93.7	28.61	56.05	24.01	54.58
	ResNet50 (None)	92.15	29.24	49.33	17.09	49.24
	ResNet50 (BN)	95.6	9.15	36.37	8.72	36.64

[4] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In NeurIPS, 2019

Other Normalization Results

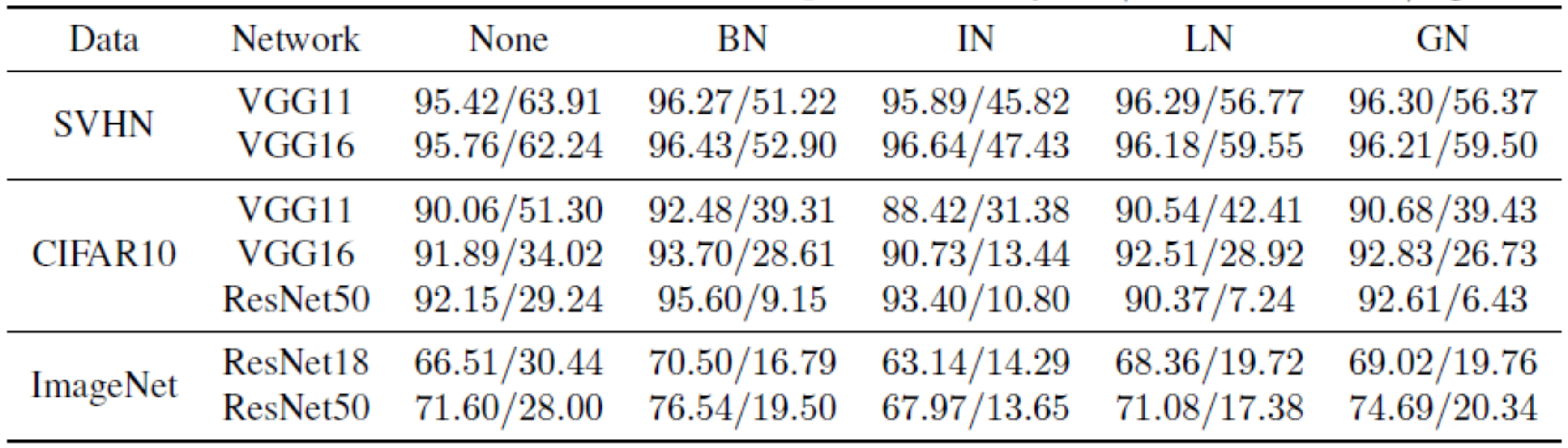


Table 2. Influence of various normalization techniques on accuracy (left/) and robustness (/right).

Data	Network	None	BN	IN	LN	GN
SVHN	VGG11	95.42/63.91	96.27/51.22	95.89/45.82	96.29/56.77	96.30/56.37
	VGG16	95.76/62.24	96.43/52.90	96.64/47.43	96.18/59.55	96.21/59.50
CIFAR10	VGG11	90.06/51.30	92.48/39.31	88.42/31.38	90.54/42.41	90.68/39.43
	VGG16	91.89/34.02	93.70/28.61	90.73/13.44	92.51/28.92	92.83/26.73
	ResNet50	92.15/29.24	95.60/9.15	93.40/10.80	90.37/7.24	92.61/6.43
ImageNet	ResNet18	66.51/30.44	70.50/16.79	63.14/14.29	68.36/19.72	69.02/19.76
	ResNet50	71.60/28.00	76.54/19.50	67.97/13.65	71.08/17.38	74.69/20.34

RF and NRF

Feature is the key concept in CV.

RF (Robust Feature) : a feature f is robust if there exists a $\gamma > 0$ for it to be γ-robustly useful under some specified set of valid perturbations Δ, i.e. $\mathbb{E}_{(x,\, y)\sim D}\left[\inf_{\delta \epsilon \Delta(x)} y \cdot f(x+\delta)\right] \geq \gamma$.

NRF (Non-Robust Feature) : a feature f is non-robust if $\gamma > 0$ does not exist.

!!! But we still need to distinguish robust usefulness from robust for a network!

Usefulness, Robust Usefulness and Robustness

Define: a DNN classifier as a set of features, i.e. $F = \{f\}$

F usefulness: F is ρ -useful ($\rho > 0$) if it is correlated with the true label in expectation, i.e. $\mathbb{E}_{(x,\, y) \sim D}[y \cdot F(x)] \geq \rho$.

F robust usefulness: F is γ-robustly useful if there exists a $\gamma > 0$ under some specified set of valid perturbations Δ, i.e. $\mathbb{E}_{(x,\, y) \sim D}\left[\inf_{\delta \epsilon \Delta(x)} y \cdot F(x+\delta) \right] \geq \gamma$.

Robust usefulness correlates to usefulness, and we can not trivially define F robustness by measuring its correlation with the true label in expectation to ensure the orthogonality to usefulness.

Usefulness, Robust Usefulness and Robustness

F robustness: A feature set F is β-robust if the local linearity is larger than β $(\beta > 0)$, i.e. $\mathbb{E}_{(x,\ y)\sim D,\ \upsilon\sim\Delta}\left[sim(\nabla l(x,\ y), \nabla l(x\ +\ \nu,\ y))\right]\ \geq\ \beta$.

Here we use cosine similarity, termed Local Input Gradient Similarity (LIGS).

$$\mathbf{E}_{(x,y)\sim D,\nu\sim\Delta}\left[\frac{\nabla l(x,y)\cdot\nabla l(x+\nu,y)}{\|\nabla l(x,y)\|\cdot\|\nabla l(x+\nu,y)\|}\right]$$

So by definition, F robustness, F usefulness, and F robust usefulness can be measured by LIGS, clean accuracy, and robust accuracy, respectively.

Experimental Results

Train with or without BN.

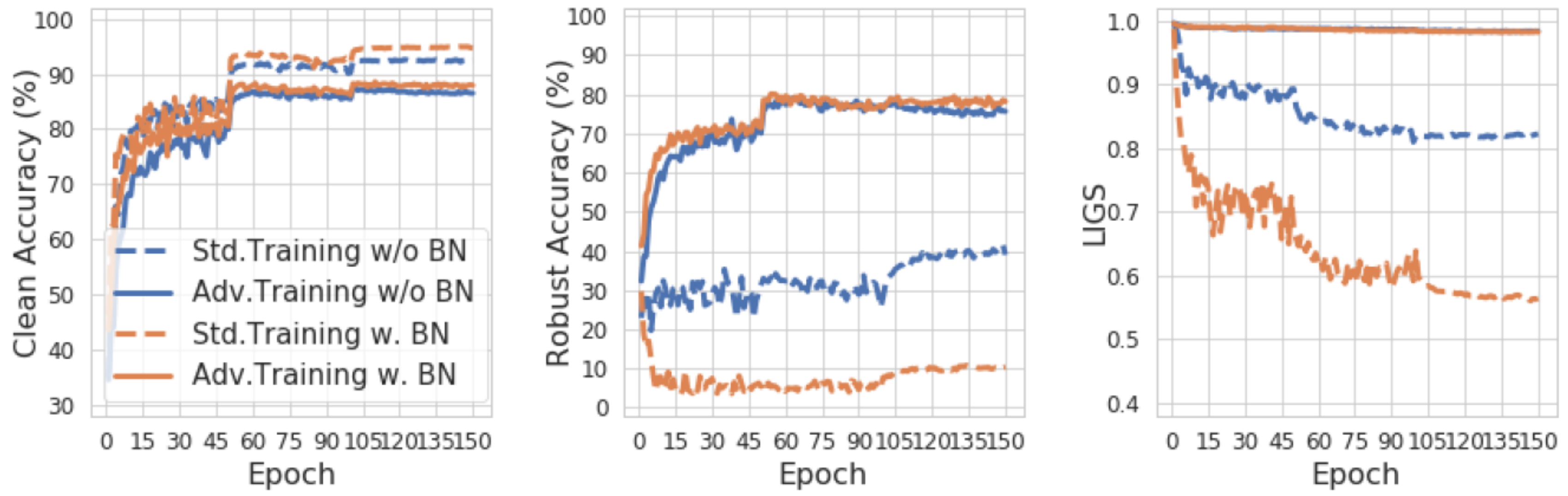


Figure 3. Trend of clean accuracy, robust accuracy, LIGS with ResNet18 on CIFAR10.

Experimental Results

Increasing LIGS through regularization improves the robust accuracy by a large margin despite a small influence on clean accuracy.

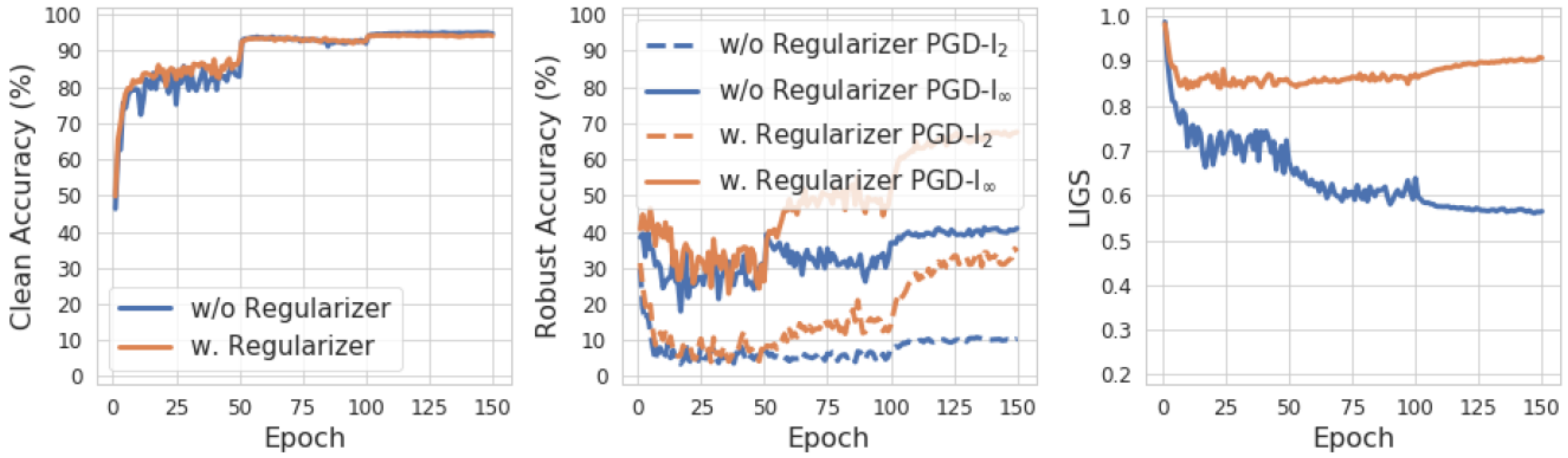


Figure 4. Effect of regularizing LIGS.

Conclusion

The model learns first RFs and then NRFs because RFs are essential for training the model in the early stage, and that BN is crucial for learning NRFs.

Use BN carefully in medical image processing: batch size, and other kinds of normalization methods.