Revisiting the Training of Very Deep Neural Networks without Skip Connections

Oyebade K. Oyedotun, Abdelrahman Shabayek, Djamila Aouada, Björn Ottersten
Interdisciplinary Centre for Security, Reliability and Trust (SnT)
University of Luxembourg, L-1855 Luxembourg
{oyebade.oyedotun, djamila.aouada, bjorn.ottersten}@uni.lu

Presentation by
Oyebade K. Oyedotun

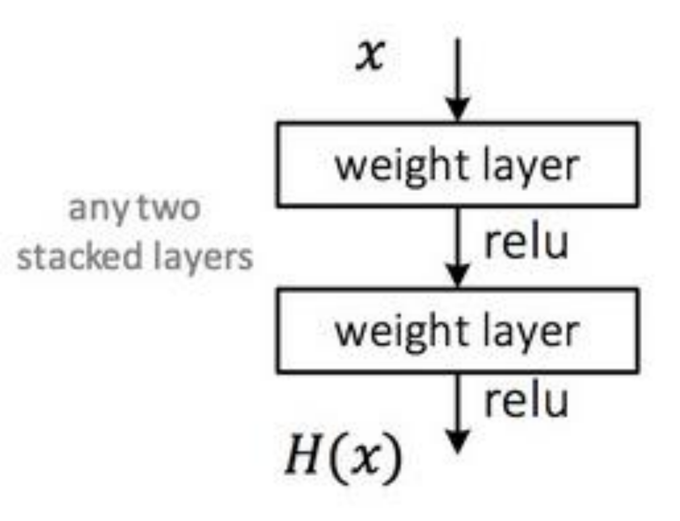
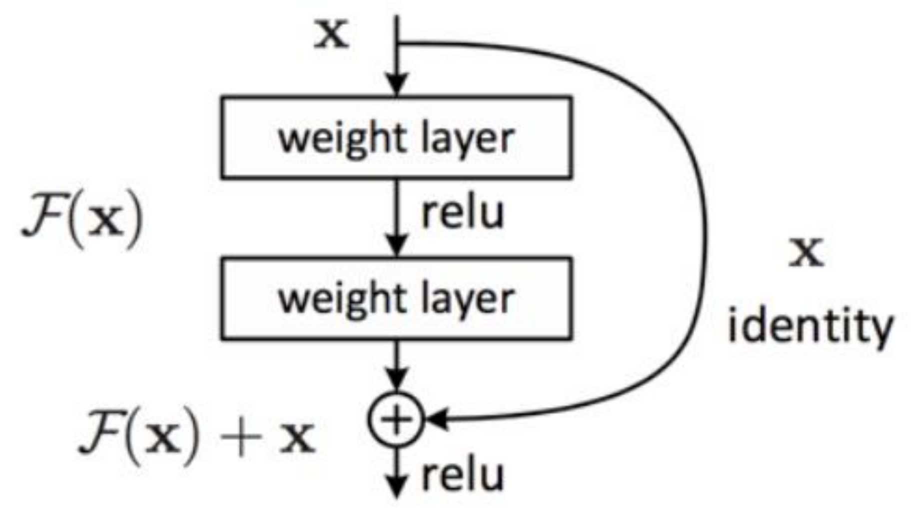
27 October, 2020

Outline

Introduction: training Very Deep Neural Networks

Very deep models:

- These are deep neural networks (DNNs) with over 15 layers

	PlainNets (i.e. no skip connections)	Deep Neural Networks with Skip Connections
Features	• **Few** layers • **Simple** architectures • **Difficult** optimization • Model operation is **explainable** [1]	• **Several** layers • **Complicated** architectures • **Easy** optimization • Model operation is **unclear** [1, 2, 3, 4]
Architecture	• **No skip connections**  	• With skip connections 

Introduction: training very deep PlainNets is difficult

securityandtrust.lu

Problem statement:

- Training very deep PlainNets become difficult with depth increase

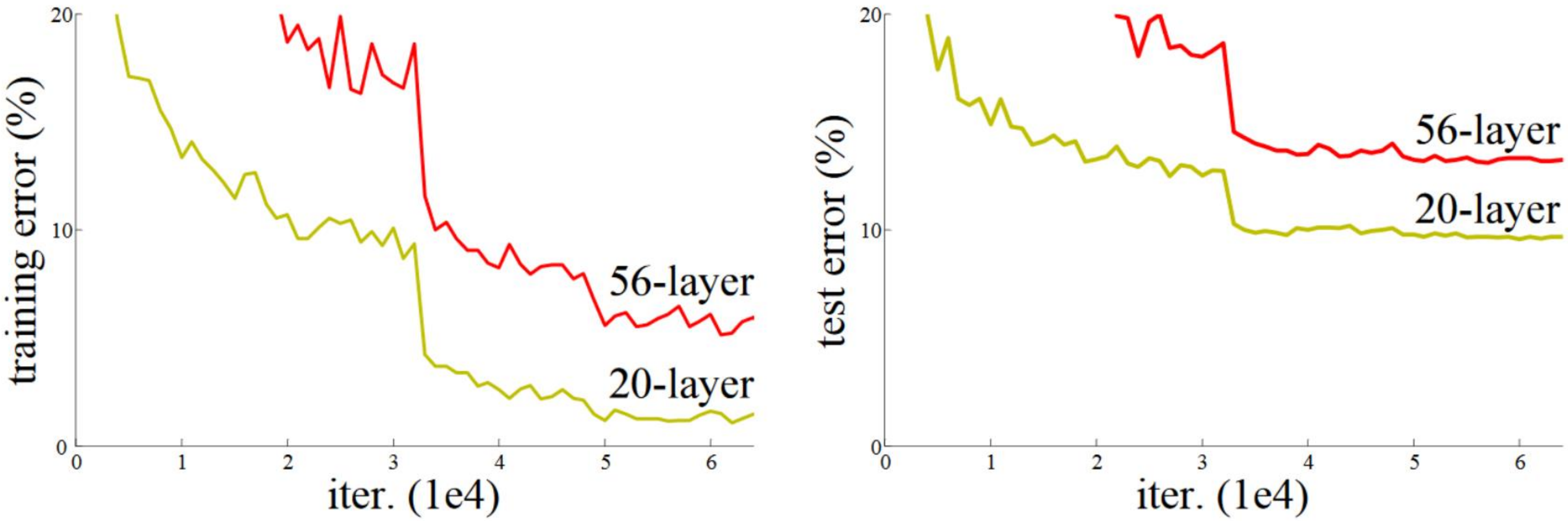
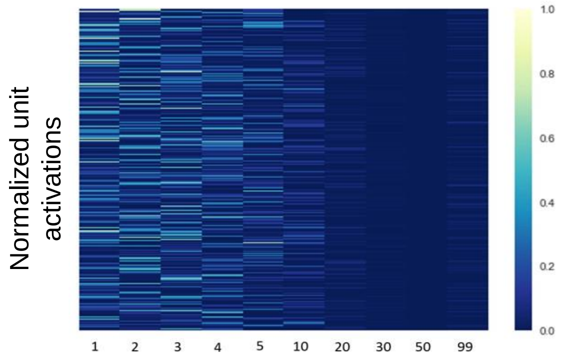


Fig. 2. Error rate increase on the very deep PlainNets trained on CIFAR-10 dataset [5]

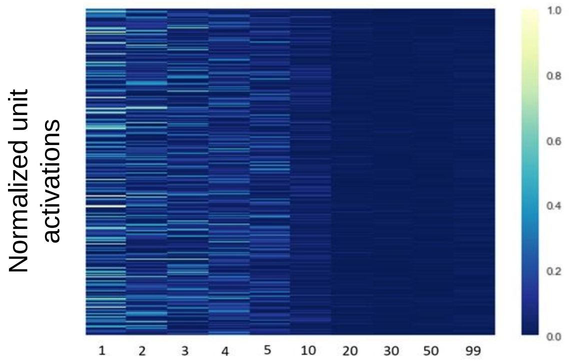
Investigation: vanishing/exploding units' outputs

- ❑ Units' outputs decrease globally with depth

Normalized mean layer activations for a 100 layer PlainNet over COIL-20 dataset

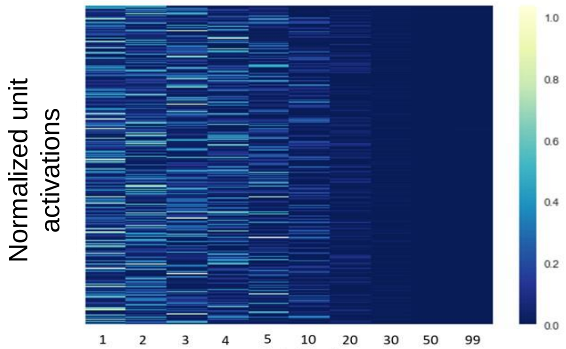


(a) Naive PlainNet: vanishing activations

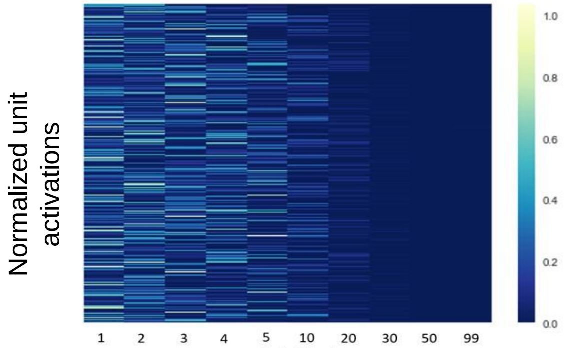


(b) PlainNet-BN: vanishing activations

Normalized mean layer activations for a 100 layer PlainNet over USPS dataset



(a) Naive PlainNet: vanishing activations



(b) PlainNet-BN: vanishing activations

Keys:

- BN → batch normalization [6]
- Naïve PlainNet → no BN
- PlainNet-BN → with BN

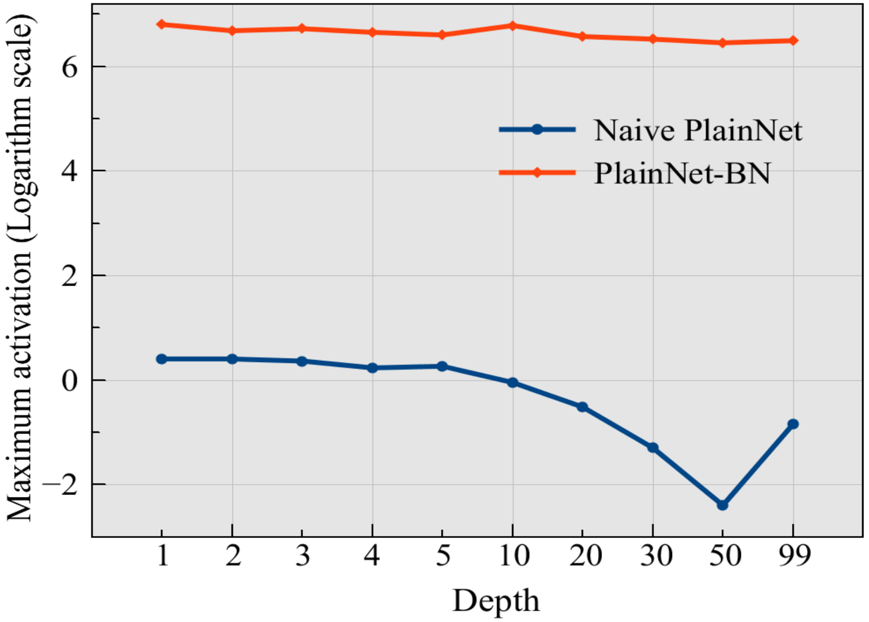
Highlights:

1. Units' outputs decay → info loss

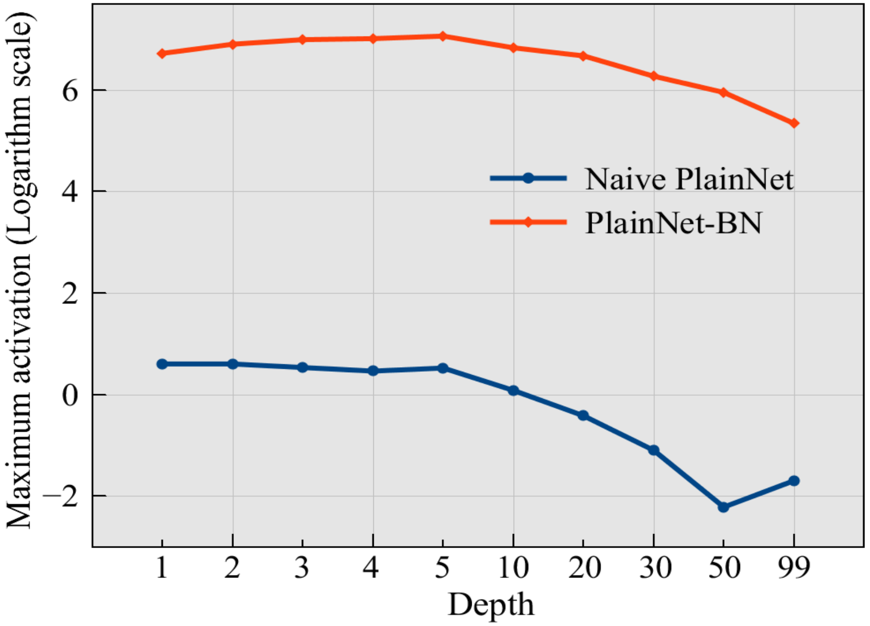
[5] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ICML (pp. 448-456).

Investigation: vanishing/exploding units' outputs

- Units' in PlainNet-BN have extremely high outputs



COIL-20 dataset



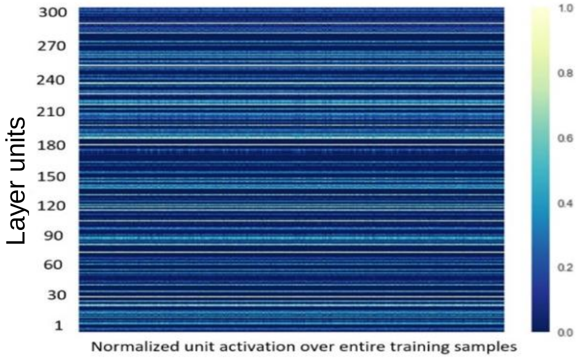
USPS dataset

Highlight:

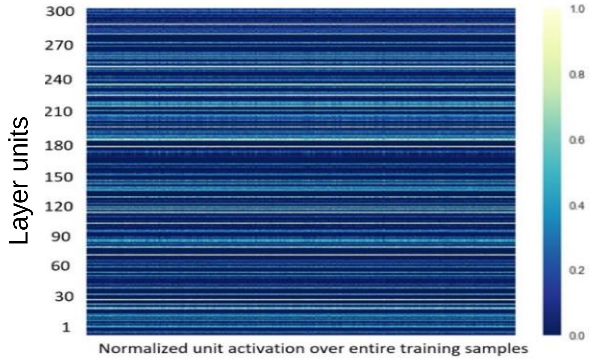
Extremely high or small outputs → bad optimization

Investigation: hidden representation near singularity

- ❑ Units respond in similar fashion to different training samples

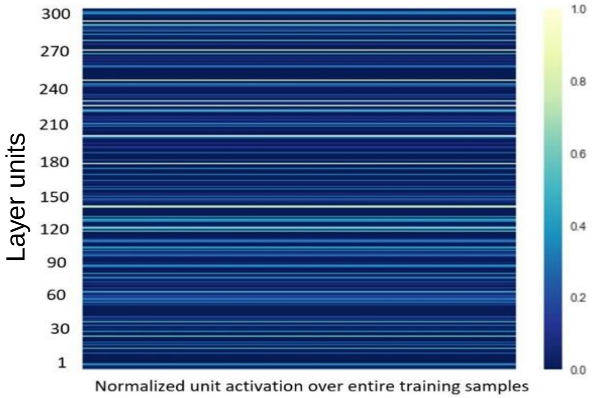


Naive PlainNet: near singularity

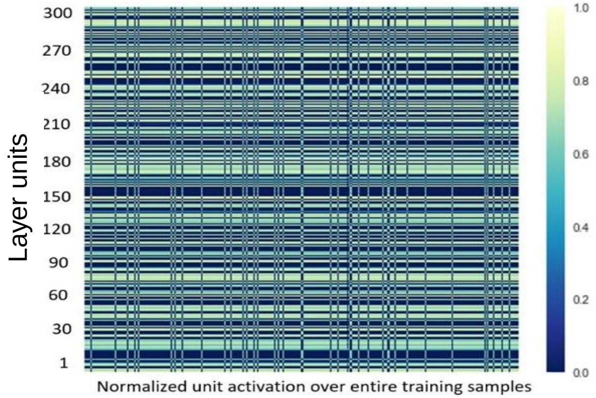


PlainNet-BN: near singularity

50th layer activations in a 100 layer PlainNet for the entire COIL-20 training set



Naive PlainNet: near singularity



PlainNet-BN: near singularity

99th layer activations in a 100 layer PlainNet for the entire COIL-20 training set

Highlights:

1. Units' response → near-singularity
2. Near-singularity → high condition number
3. High condition number → bad optimization

Alleviating the training problems of very deep PlainNets

securityandtrust.lu

- Components of the proposed approach

  - **Leaky Rectified Linear Unit (LReLU) → vanishing/exploding units' outputs**

$$H(x)^l = 1(Z^l \leq 0)(\beta Z^l) + 1(Z^l > 0)(Z^l), \tag{1}$$

where Z^l and β are the pre-activation and leaky scaling factor, respectively.

  - **Max-norm constraint → exploding unit's output**

$$\| \vec{w_j} \| \leq c\ , \tag{2}$$

where c is the specified max-norm.

  - **Weight initialization from uniform distribution → weight diversity**

$$U[\sqrt{6/n_{in}^l}, -\sqrt{6/n_{in}^l}]\ , \tag{3}$$

where n_{in}^l is the number of units feeding into layer l.

Experimental results: proposed solution (PlainNet)

- Table 1: Ablation studies – 100 layer model results using USPS dataset

Model component	Train error	Test error
Batch normalization (BN)	84.56%	83.21%
LReLU	92.37%	92.03%
Max-norm	86.22%	86.85%
BN + LReLU	78.38%	79.52%
BN + max-norm	82.90%	81.86%
LReLU + max-norm	83.62%	82.11%
Proposed: BN + LReLU + max-norm	**0.11%**	**5.48%**

Highlight:

Proposal → the three components give the best results

- Table 2: Model results using CIFAR-10 dataset

Model	Skip conn.	Layers	Parameters	Test error
Highway network [2]	Yes	19	2.30M	7.54%
ResNet [3]	Yes	56	0.85M	6.97%
ResNet [3]	Yes	110	1.7M	6.43%
All CNN [30]	No	8	1.30M	7.25%
NiN [31]	No	10	1.30M	8.81%
Delta init. [15]	No	32	17.80M	18.00%
PlainNet-BN [3]	No	56	0.85M	15.00%
Proposed PlainNet	**No**	**50**	**0.72M**	**6.65%**

Highlight:

Proposal → successful training

Conclusion

securityandtrust.lu

Paper highlights:

- Revisited the problem of training very deep networks without skip connections
- Proposed an approach to tackle identified problems
- The proposed DNN is seen to outperform similar models without skip connections
- The proposed DNN without skip connections achieve competitive results in comparison to DNNs with skip connections.

References

1. Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S., & Pennington, J. (2018, July). Dynamical Isometry and a Mean Field Theory of CNNs: How to Train 10,000-Layer Vanilla Convolutional Neural Networks. In International Conference on Machine Learning (pp. 5393-5402).
2. Veit, A., Wilber, M. J., & Belongie, S. (2016). Residual networks behave like ensembles of relatively shallow networks. In Advances in neural information processing systems (pp. 550-558).
3. Balduzzi, D., Frean, M., Leary, L., Lewis, J. P., Ma, K. W. D., & McWilliams, B. (2017, August). The shattered gradients problem: if resnets are the answer, then what is the question?. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 342-350).
4. Greff, K., Srivastava, R. K., & Schmidhuber, J. (2017). Highway and residual networks learn unrolled iterative estimation. International Conference on Learning Representations.
5. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
6. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In Thirty-first AAAI conference on artificial intelligence.

Thank you !