Batch Normalization Provably Avoids Rank Collapse for Randomly Initialised Deep Networks

H. Daneshmand*, J. Kohler*, F. Bach, T. Hofmann. A. Lucchi

ETH Zürich, Switzerland & INRIA Paris, France

Motivation and contributions

Randomly initialized neural networks are known to become harder to train with increasing depth. Making use of the rich literature on random matrices, we pin this problem down to the fact that the **rank of intermediate representations in unnormalized networks collapses quickly with depth**. In this work we highlight the fact that **batch normalization is an effective strategy to avoid rank collapse** for both linear and ReLU networks.

- Leveraging tools from Markov chain theory, we derive a meaningful lower rank bound in deep linear networks.
- Empirically, we also demonstrate that this rank robustness generalizes to ReLU nets.
- Finally, we conduct an extensive set of experiments on real-world data sets, which confirm that rank stability is indeed a crucial condition for training modern-day deep neural architectures.

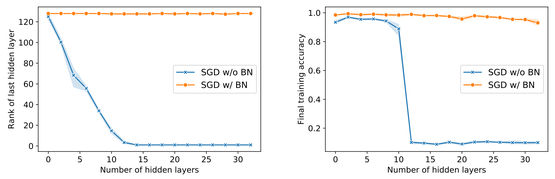
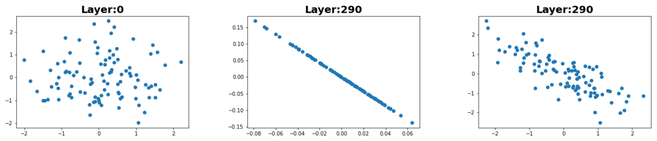


Fig. 1: Effect of depth on rank and learning, on the Fashion-MNIST dataset with ReLU MLPs of depth 1-32. Left: Rank after random initialization. Right: Training accuracy after training 75 epochs with SGD.

Rank collapse for vanilla networks



Input data | Output of a vanilla-net | Output of a BN-net

We consider the hidden states $\widehat{H}_\ell$ of the following linear residual network:

$$\widehat{H}_\ell = \mathbf{B}_\ell X, \quad \mathbf{B}_\ell := \prod_{k=1}^{\ell} (I + \gamma W_k). \tag{1}$$

where $X \in \mathbb{R}^{d\times N}$ is the input matrix containing N samples in $\mathbb{R}^d$. Since the norm of $\widehat{H}_\ell$ is not necessarily bounded, we normalize as $\widetilde{H}_\ell = B_\ell X/\|B_\ell\|$.

Lemma 1(A known result about product of random matrices) Suppose that $\gamma \in (0,1)$ and assume the weights W_ℓ are identically independent random matrices with elements i.i.d. $N(0,1)$. Then the sequence $\{\widetilde{H}_\ell\}$ converges to a rank one matrix.

Batch Normalization prevents rank collapse

Observation we observe that BN effectively avoids the rank collapse.

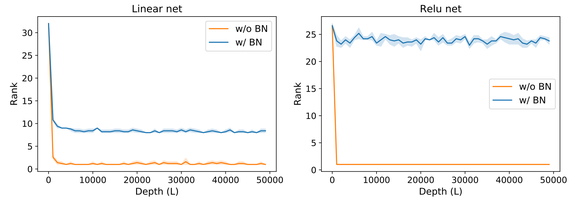


Fig. 2: Log(rank) of the last hidden layer's activation over total number of layers (blue for BN- and orange for vanilla-networks). MLPs with Gaussian inputs. Left: Linear activations. Right: ReLU activations.

Rank definitions To circumvent numerical issues we introduce a soft notion of the rank denoted by $\text{rank}_\tau(H)$ (soft rank). Let $\sigma_1, \ldots, \sigma_d$ be the singular values of H. Then, given a $\tau > 0$, we define $\text{rank}_\tau(H) = \sum_{i=1}^{d} \mathbf{1}(\sigma_i^2/N \geq \tau)$. Clearly that $\text{rank}_\tau(H) \leq \text{rank}(H)$. For analysis purposes, we introduce a lower bound on $\text{rank}_\tau(H)$ that is differentiable w.r.t. H

$$r(H) = \text{Tr}(M(H))^2/\|M(H)\|_F^2, \quad M(H) = HH^\top/N. \tag{2}$$

BN and hidden representations Let $H_\ell^{(\gamma)}$ denote the hidden representation of X in layer ℓ of a BN-network with residual connections. The following recurrence summarizes the network mapping

$$H_{\ell+1}^{(\gamma)} = \text{BN}_{0,\mathbf{1}_d}(H_\ell^{(\gamma)} + \gamma W_\ell H_\ell^{(\gamma)}), \quad H_0^{(\gamma)} = X, \tag{3}$$

where $W_\ell \in \mathbb{R}^{d\times d}$ and γ regulates the skip connection strength. We define the BN operator $\text{BN}_{\alpha,\beta}$ as in the original paper [1], namely

$$\text{BN}_{\alpha,\beta}(H) = \beta \circ (\text{diag}\,(M(H)))^{-1/2} H + \alpha \mathbf{1}_N^\top, M(H) := \frac{1}{N} HH^\top, \tag{4}$$

where $\circ$ is a row-wise product. Both $\alpha \in \mathbb{R}^d$ and $\beta \in \mathbb{R}^d$ are trainable parameters. We assume the initialization $\alpha = 0$ and $\beta = \mathbf{1}_d$.

A theoretical study We study the invariant state of Markov chain of hidden representation, i.e. $\{H_\ell^{(\gamma)}\}$. The invariant property imposes a high rank structure. Assuming that the chain is ergodic, this structure is projected on the average of hidden representation over layers.

Theorem 2 Suppose that the $\text{rank}(X) = d$ and that the weights W_ℓ are initialized in a standard i.i.d. zero-mean fashion. Then, the following limits exist such that

$$\lim_{L\to\infty} \frac{1}{L} \sum_{\ell=1}^{L} \text{rank}_\tau(H_\ell^{(\gamma)}) \geq \lim_{L\to\infty} \frac{(1-\tau)^2}{L} \sum_{\ell=1}^{L} r(H_\ell^{(\gamma)}) = \Omega((1-\tau)^2\sqrt{d}) \tag{5}$$

holds almost surely for a sufficiently small γ (independent of ℓ) and any $\tau \in [0,1)$, under some additional technical assumptions.

The role of the rank for optimization

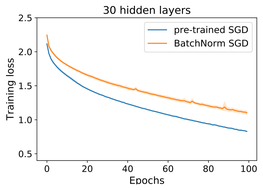


Fig. 3: Beating Batch Norm: Loss over epochs on CIFAR-10 w/ 30 layer MLP.

1. Unsupervised,rank-increasing, pre-training allows SGD to outperform BN: We leverage the lower bound established in Eq. (2) to design a pre-training step that avoids rank collapse and accelerates convergence of SGD on vanilla MLPs. Our proposed procedure is simple and computationally cheap. Specifically, we *maximize* the lower-bound $r(H_\ell)$ (in Eq. (2)) on the rank of the hidden presentation H_ℓ in each layer ℓ by just a few steps of (stochastic) gradient ascent.

2. Breaking batch normalization. We observe that the way that networks are initialized does play a crucial role for the subsequent optimization performance of BN-nets. Particularly, the dashed lines in Fig 4 show that simply moving the domain of the uniform initialziation distribution into the positive realm, breaks the effectivness of BN drastically. A further noteworthy aspect is the clear correlation between the level of pre-training and optimization performance.

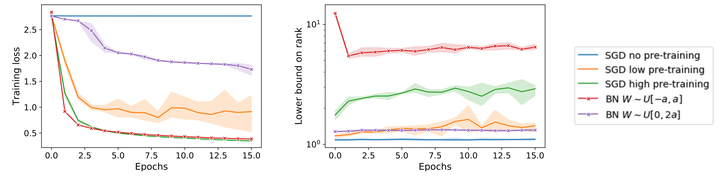


Fig. 4: Fashion-MNIST on MLPs of depth 32 and width 128. (Left) Training accuracy, (Right) Lower bound on rank. Blue line is a ReLU network with standard initialization. Other solid lines are pre-trained layer-wise with 25 (orange) and 75 (green) iterations. Dashed lines are BN nets with standard and asymmetric init.

Future works

Our findings give rise to several interesting follow-up questions such as:

- Can one generalize the analysis of Theorem 2 to ReLU and other non-linear nets to prove the observed rank robustness (e.g. Fig. 2)?
- Is it possible to rigorously prove that SGD updates preserve the rank magnitude throughout optimization, as observed in Fig. 4)?
- Is it possible to use the develop a similarly effective pre-training for convolution and recurrent networks?
- How can one theoretically characterize the connection between the convergence of SGD and the rank quantity (a follow-up on directional gradient vanishing)?