Representations in Random Deep Neural Networks

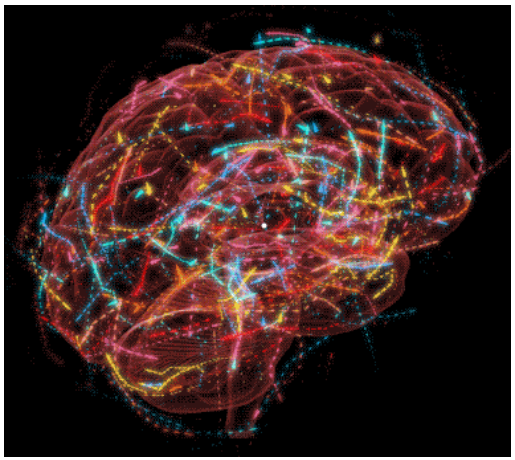
Hadi Daneshmand, Amir Joudaki, Francis Bach

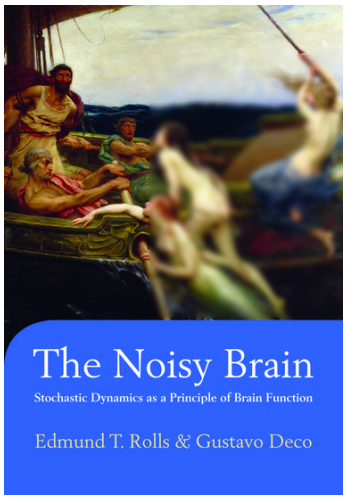
INRIA Paris, Sierra Team

Oct. 2021

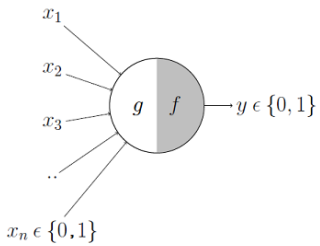
"The noisy brain"

- Brain is a collection of noisy firing neurons
- Microscopically neurons are random but they collectively resemble interesting stochastic processes
- Mean-field analysis of underlying stochastic processes in the brain sheds light into
  - decision-making
  - attention
  - brain dysfunctions such as schizophrenia

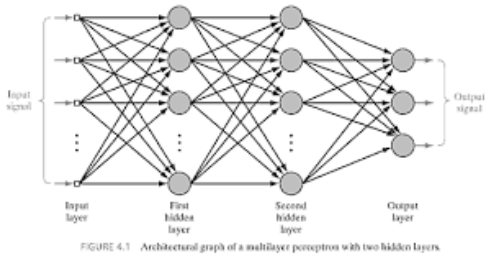




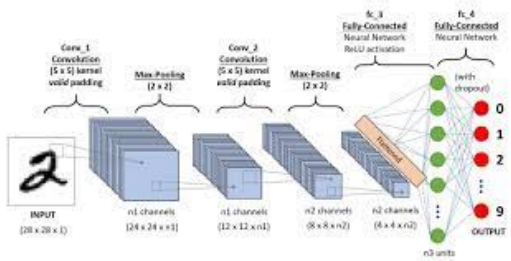
Evolution of computational neural networks



McCulloch-Pitts Neuron



MLP



Convolutional networks

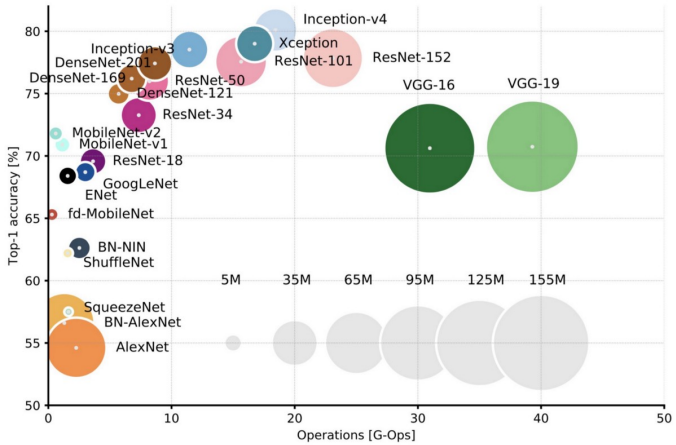


Figure: Images credit: Alfredo Canziani et. al.

Noisy computational networks

- **Historical** neural networks do not work well with random weights
- **Modern** computational neural networks perform surprisingly well with random weights if the neurons are wired well together[1].

Why?

[1] Frankle, J., Schwab, D. J. & Morcos, A. S. Training batchnorm and only batchnorm: On the expressive power of random features in CNNs. *ICLR* (2021).

Historical single-layer MLPs

$$H_1 = \frac{1}{\sqrt{\text{batchsize}}} F(W_0 H_0)$$

- $H_0 \in \mathbb{R}^{\text{width} \times \text{batchsize}}$ is a deterministic matrix
- $W_0 \in \mathbb{R}^{\text{width} \times \text{width}}$ is a random Gaussian matrix
- Empirical eigenvalue distribution (e.e.d): $\frac{1}{\text{width}} \sum_{i=1}^{\text{width}} \delta(\lambda_i(H_1^\top H_1))$
- Given the first two moments of H_1,[2] characterizes e.e.d. of H_1 as batchsize and width tends to ∞ denoted by p

[2] Louart, C., Liao, Z., Couillet, R., *et al.* A random matrix approach to neural networks. *The Annals of Applied Probability* (2018).

Historical single-layer MLPs

- Stieltjes transformation of density p on interval I:

$$S_p(z) = \int_I \frac{p(t)dt}{z-t}, \quad z \in C \backslash I$$

- Given $G = \mathbf{E}\left[H_1^\top H_1\right]$, the following holds[3]

$$S_p(z) = \frac{1}{\text{batchsize}} \text{Tr}\left(\underbrace{\frac{\text{width}}{\text{batchsize}} \frac{G}{1+s(z)} - zI}_{M(s)} \right)^{-1}$$

- $s(z)$ is the solution of

$$s(z) = \frac{1}{\text{batchsize}} \text{Tr}\left(GM^{-1}(s(z))\right)$$

[3] Louart, C., Liao, Z., Couillet, R., *et al.* A random matrix approach to neural networks. *The Annals of Applied Probability* (2018).

Historical random deep networks

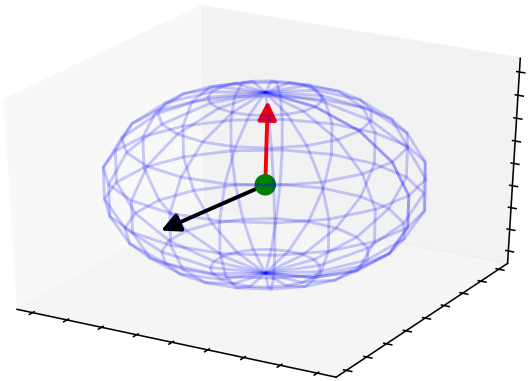
- Let $x_\ell \in \mathbb{R}^{\text{width}}$ be representation of input x_0 at layer ℓ
- The representations make a Markov chain as:

$$x_{\ell+1} = \frac{W_\ell x_\ell}{\| W_\ell x_\ell \|}$$

- Suppose the elements of $W_\ell \in \mathbb{R}^{\text{width} \times \text{width}}$ are i.i.d. Gaussian.

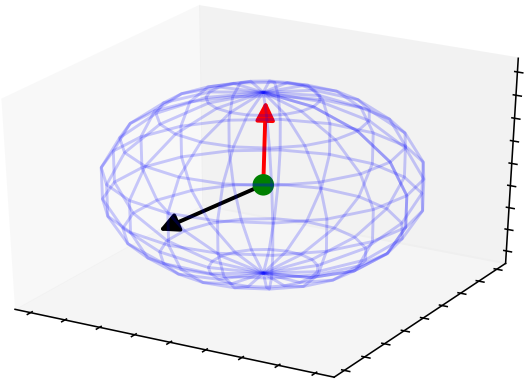
Coupled representations

$$x_{\ell+1} = \frac{W_\ell x_{\ell+1}}{\|W_\ell x_{\ell+1}\|}, \qquad y_{\ell+1} = \frac{W_\ell y_{\ell+1}}{\|W_\ell y_{\ell+1}\|}$$



Coupled representations

$$x_{\ell+1} = \frac{W_\ell x_{\ell+1}}{\|W_\ell x_{\ell+1}\|}, \qquad y_{\ell+1} = \frac{W_\ell y_{\ell+1}}{\|W_\ell y_{\ell+1}\|}$$



The chains contracts to a random directions independent from the starting state.

Product of random matrices

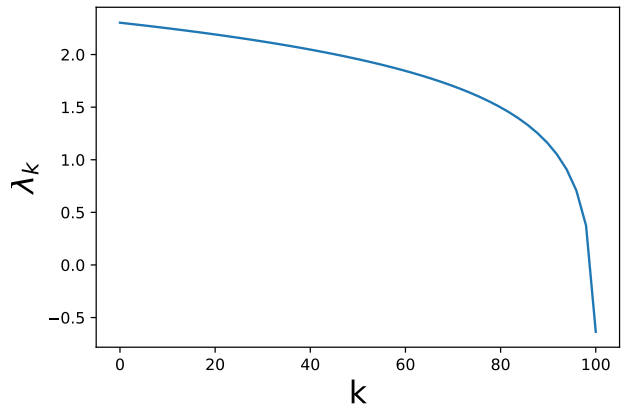
- Consider the product of Gaussian matrix as $S_\ell = W_\ell \dots W_1$
- Claim: $S_\ell / \|S_\ell\|$ becomes rank one in limit.
- Therefore, $(S_\ell x)/\|S_\ell\|$ becomes independent from x as $\ell \to \infty$.

Lyapunov exponents

- Definition:

$$\lambda_k = \lim_{\ell \to \infty} \frac{1}{2\ell} \log \left(k^{\text{th}} \text{ largest eigenvalue of} S_\ell^\top S_\ell \right)$$

- Computation[4]: $\lambda_k = \frac{1}{2}(\log(2) + \Psi\left(\frac{d-k+1}{2}\right))$



- $\lambda_1 - \lambda_2 < 0$ implies $S_\ell / \|S_\ell\|$ becomes rank one in limit.

[4] Newman, C. M. The distribution of Lyapunov exponents: Exact results for random matrices. *Communications in mathematical physics* (1986).

Modern NN with Batch normalization (BN)

- BN is one of the main building block of modern neural networks[5]
- Representation H_ℓ:*width* $\times$ *batchsize*.

$$H_{\ell+1} = F\left(BN_{\alpha,\beta}(W_\ell H_\ell)\right) \tag{1}$$

- $BN_{\alpha,\beta} : \mathbb{R}^{width \times batchsize} \to \mathbb{R}^{width \times batchsize}$

$$[BN_{\alpha,\beta}(M)]_{:} = \alpha_i \text{centered}(M_{i:}) + \beta_i$$

- Learning only parameters α and β (per unit) leads to surprisingly good performance[6]

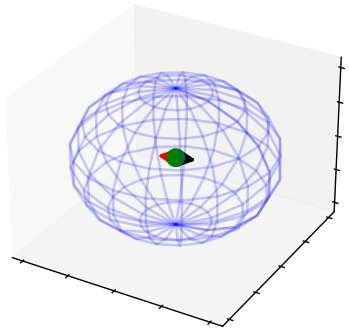
[5] Ioffe, S. & Szegedy, C. *Batch normalization: Accelerating deep network training by reducing internal covariate shift.* in *ICML* (2015).

[6] Frankle, J., Schwab, D. J. & Morcos, A. S. Training batchnorm and only batchnorm: On the expressive power of random features in CNNs. *ICLR* (2021).

The Markov chain of representations

We study the following Markov chain of matrices[7][8].

- $BN(M)$ normalizes M row-wise
- Representations:
  $H_{\ell+1} = \left(\frac{1}{\sqrt{width}}\right) BN(W_\ell H_\ell)$
- W_ℓ: $(width \times width)$ with Gaussian elements

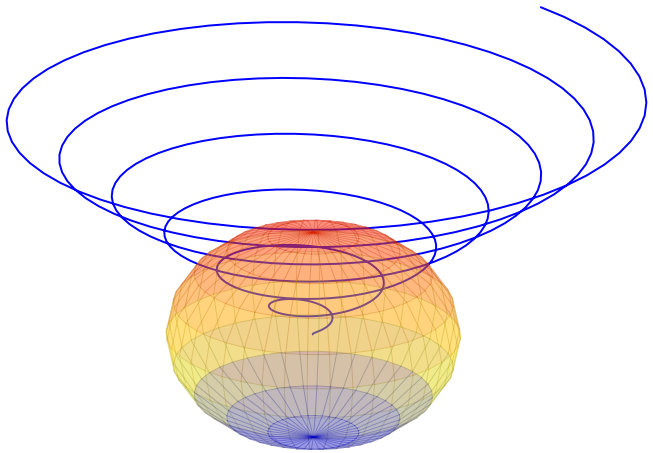


[7] Daneshmand, H., Joudaki, A. & Bach, F. Batch Normalization Orthogonalizes Representations in Deep Random Networks. *NeurIPS21.* **Spotlight presentation (among top %3 of submissions)**.

[8] Daneshmand, H. *et al.* Batch Normalization Provably Avoids Rank Collapse for Randomly Initialised Deep Networks. *NeurIPS20.*

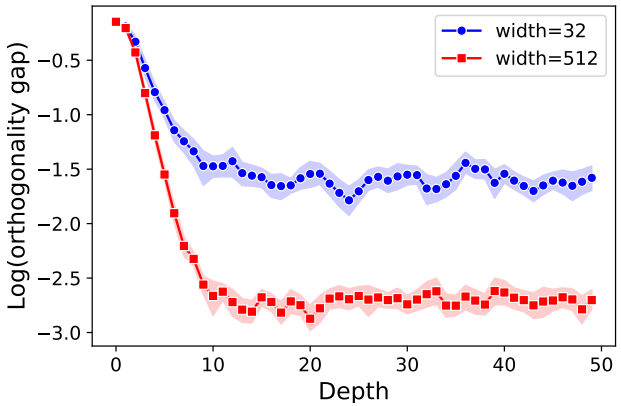
Theoretical results

- $\mathbf{E}\left[\text{orthogonality gap}(H_\ell)\right] = \mathcal{O}\left((1-\alpha)^\ell + \frac{\text{batchsize}}{\alpha\sqrt{\text{width}}}\right)$
- $\text{Wasser.}_2(W_\ell H_\ell, \text{Gaussian})^2 = \mathcal{O}\left((1-\alpha)^\ell\,(\text{batchsize}) + \frac{(\text{batchsize})^2}{\alpha\sqrt{\text{width}}}\right)$



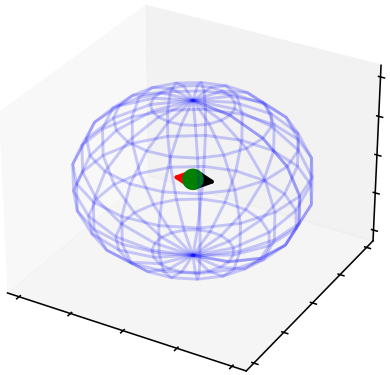
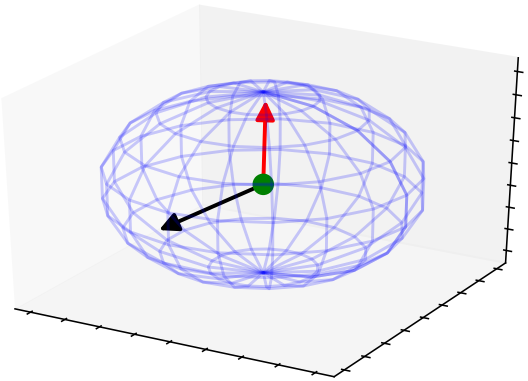
Orthogonalization

- Orthogonality gap$(H) := \left\| \left(\frac{1}{\|H\|_F^2} \right) H^\top H - \left(\frac{1}{\|I_n\|_F^2} \right) I_n \right\|_F$
- $\mathbf{E}\left[\text{orthogonality gap}(H_\ell)\right] = \mathcal{O}\left((1-\alpha)^\ell + \frac{\text{batchsize}}{\alpha\sqrt{\text{width}}}\right)$

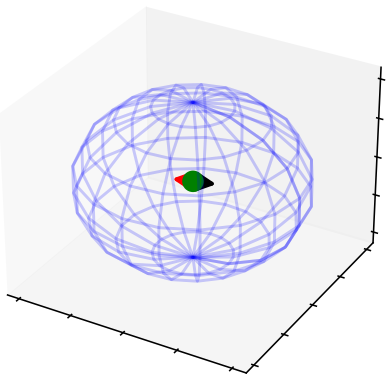
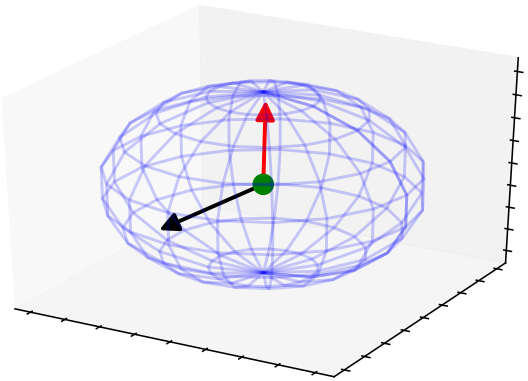


- α is the minimum of smallest singular value of $\{H_1, \ldots, H_\ell\}$.
- To get a non-vacuous bound, we need an α independent from ℓ.

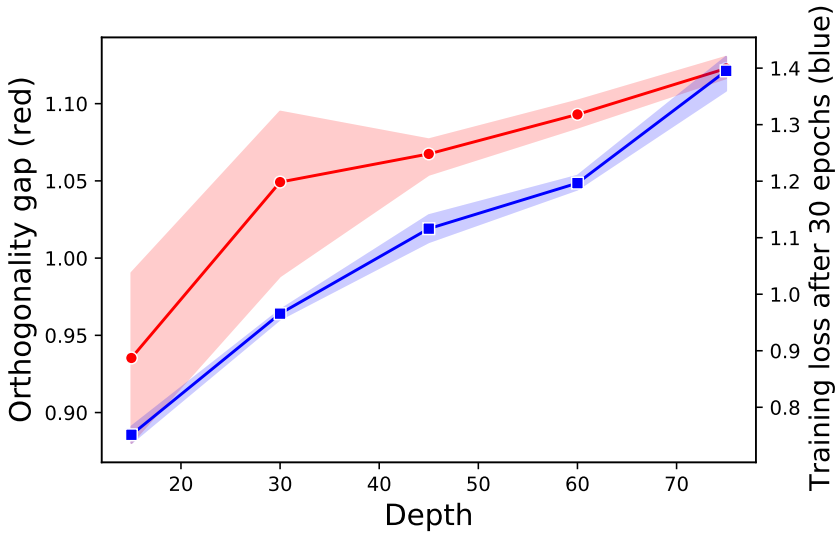
Modern NN vs. historical NN

BN	Without BN
	

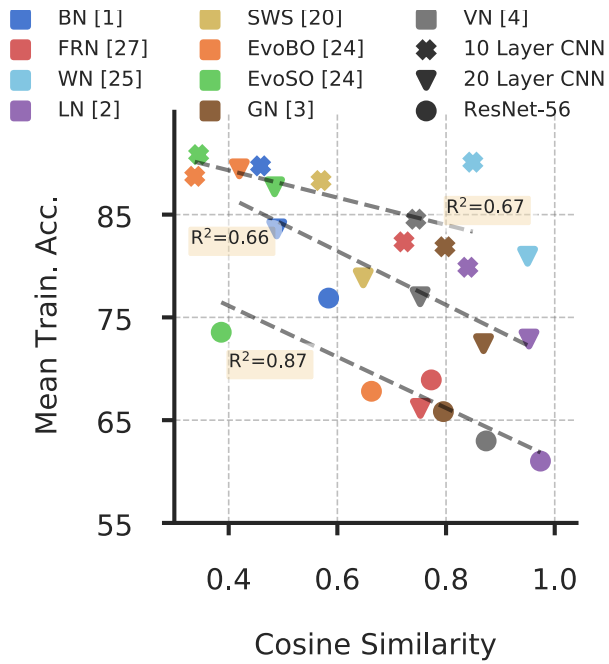
Modern NN vs. historical NN

BN	Without BN
	
$\mathbf{E}\left[\text{orth. gap}(H_\infty)\right] = \mathcal{O}\left(\frac{\text{batch size}}{\alpha\sqrt{\text{width}}}\right)$	$\mathbf{E}\left[\text{orth. gap}(H'_\infty)\right] = \Theta(1)$

The orthogonality influences training



Without normalization

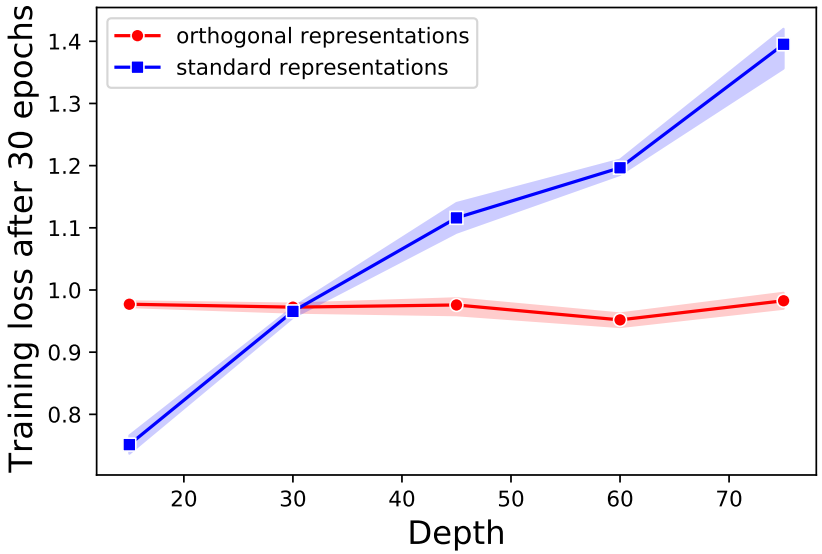


With Normalization[9]

[9] Lubana, E. S., Dick, R. P. & Tanaka, H. Beyond BatchNorm: Towards a General Understanding of Normalization in Deep Learning. *arXiv preprint arXiv:2106.05956* (2021).

Replacing BN with orthogonalization

Saving training time by starting from orthogonal representations



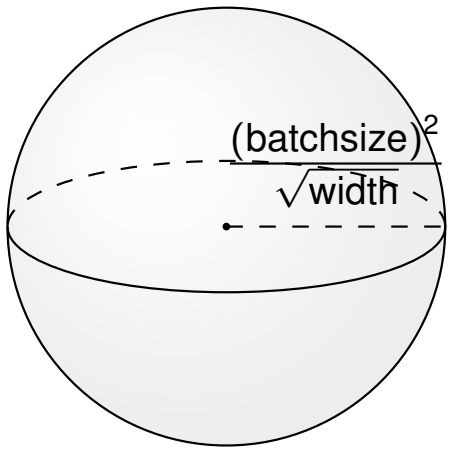
MLPs with ReLU and **without BN** for classifying CIFAR-10

Red: standard initialization with low orthogonality gaps

Blue: novel initialization ensuring orthogonal representations

Gaussian approximation

$$\text{Wasserstein}_2(W_\ell H_\ell, \text{Gaussian})^2 = \mathcal{O}\left((1-\alpha)^\ell (\text{batchsize}) + \frac{(\text{batchsize})^2}{\alpha\sqrt{\text{width}}}\right)$$



History of Gaussian approximation for NNs

∞-Width

1996 A single-layer MLP (Neal).

2015 Going beyond one layer (Hazan and Jaakkola).

2018 Finite-depth MLPs (Matthews et. al. and Lee et. al.) .

History of Gaussian approximation for NNs

∞-Width

1996 A single-layer MLP (Neal).

2015 Going beyond one layer (Hazan and Jaakkola).

2018 Finite-depth MLPs (Matthews et. al. and Lee et. al.) .

2021 Non-asymptotic (in)finite width and depth Linear MLPs.

Applications of the Gaussian approximation

Vision

Deepening our knowledge about representations in deep neural networks will allow us to design more efficient neural networks.