

# Representations in Random Deep Neural Networks

Hadi Daneshmand

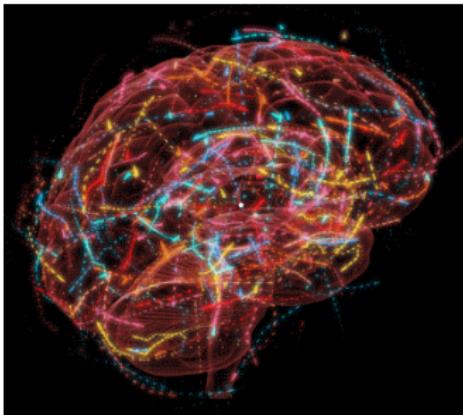
Princeton University

WSS 2022

# "The noisy brain"



- ▶ Brain is a collection of noisy firing neurons
- ▶ Random is essential for many brain tasks including decision-making attention<sup>1</sup>
- ▶ Neural synchrony causes brain dysfunctions such as schizophrenia



---

<sup>1b.</sup>

# Evolution of computational neural networks



3

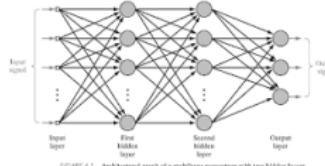
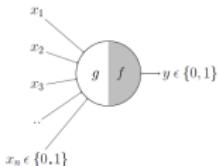
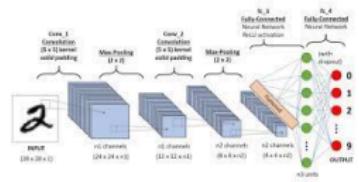


FIGURE 4.1 Architecture graph of a multilayer perceptron with two hidden layers



## McCulloch-Pitts Neuron

## MLP

## Convolutional networks

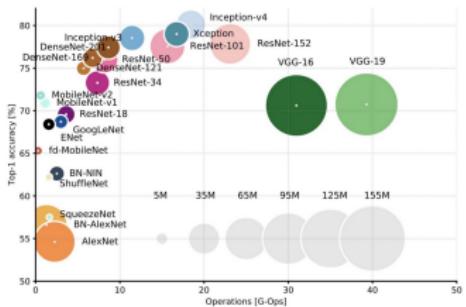


Figure: Images credit: Alfredo Canziani et. al.

# 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<sup>2</sup>.

Why?

---

<sup>2</sup>[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  $H_1$ ,<sup>3</sup> characterizes e.e.d. of  $H_1$  as batchsize and width tends to  $\infty$  denoted by  $p$

---

<sup>3</sup>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  $I$ :

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

- ▶ Given  $G = \mathbf{E} [H_1^\top H_1]$ , the following holds<sup>4</sup>

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

- ▶  $s(z)$  is the solution of

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

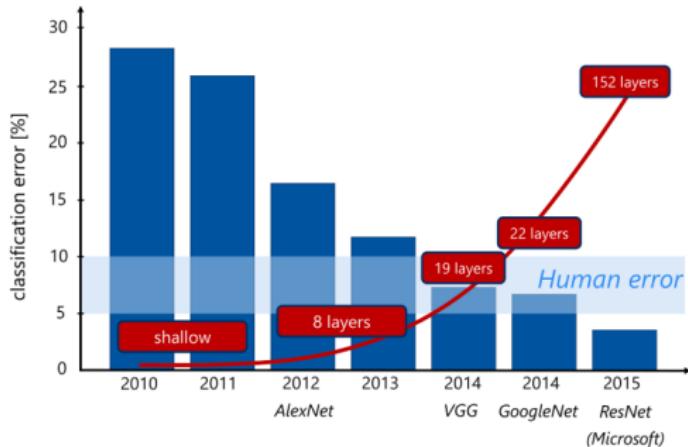
---

<sup>4</sup>Louart, C., Liao, Z., Couillet, R., et al. A random matrix approach to neural networks. *The Annals of Applied Probability* (2018).

# Depth in computational neural network



7



But, deep networks are difficult to train

ImageNet Competition Summary

# Historical random deep networks



- ▶ Let  $x_\ell \in \mathbb{R}^{\text{width}}$  be representation of input  $x_0$  at layer  $\ell$
- ▶ 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.

# Information Bottleneck Principle (Tishby & Zaslavsky)



Capture relevant information

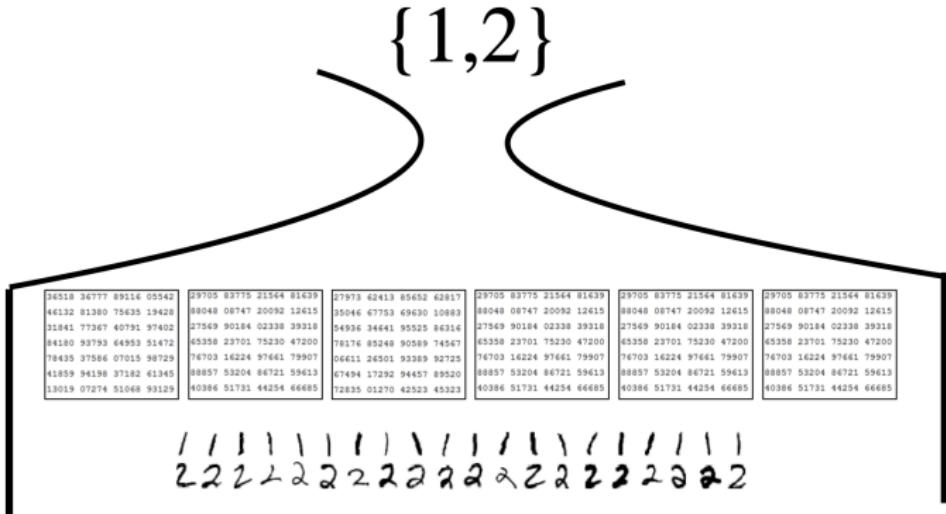
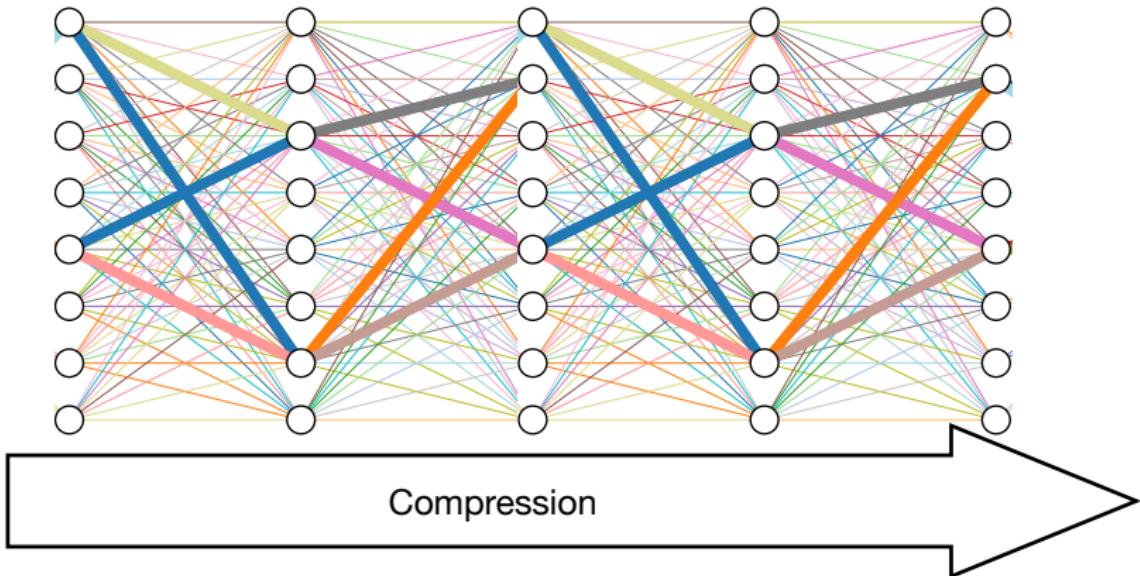


Figure: Compress irrelevant features (image idea:Xu, Tao TPAMI14)

# Compression in deep learning



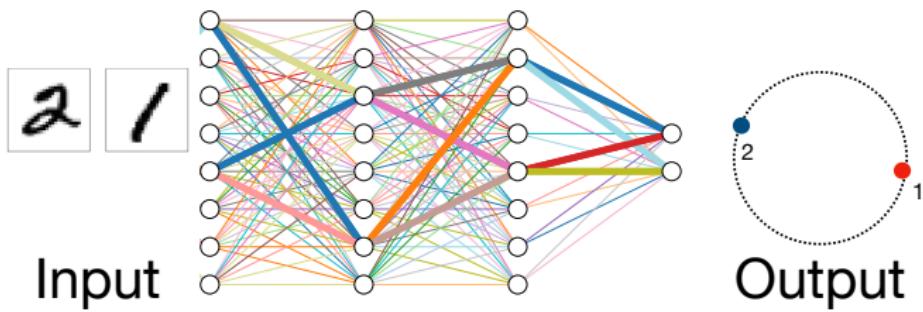
Random networks compress their inputs through their layers.



# Compression study



11



# Over-compression with depth



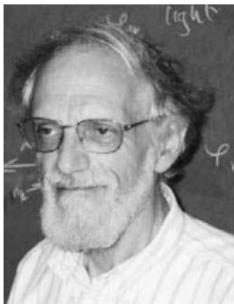
12

Deep **random** conventional networks compress inputs to a byte.

# Over-compression with depth



Deep **random** conventional networks compress inputs to a byte.



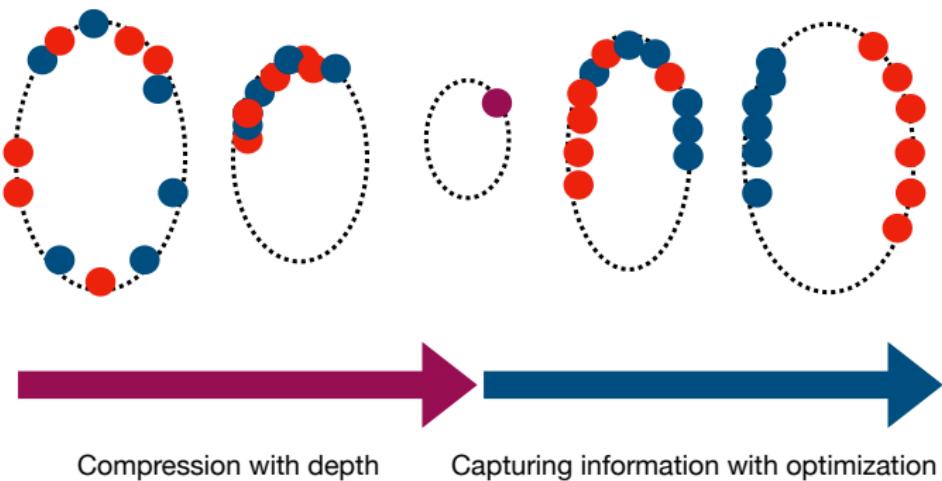
This compression is established by Markov chain studies in 1960 by Hillel Furstenberg [3]

# Information Bottleneck for deep learning



13

After compression in depth, networks capture relevant information during optimization.

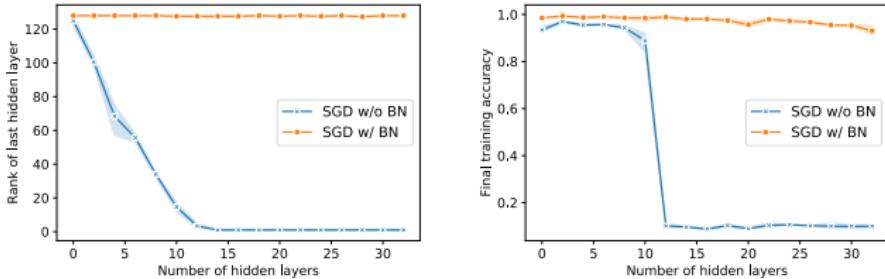


# Compression influences optimization



14

Over-compression with depth slows optimization<sup>5</sup>



<sup>5</sup>Daneshmand\*, H. et al. Batch Normalization Provably Avoids Rank Collapse for Randomly Initialised Deep Networks. *NeurIPS* (2020).

# Modern deep neural networks



15

1960

Overcompression



Slow optimization

55 years  
Engineering

2015  
Compression?

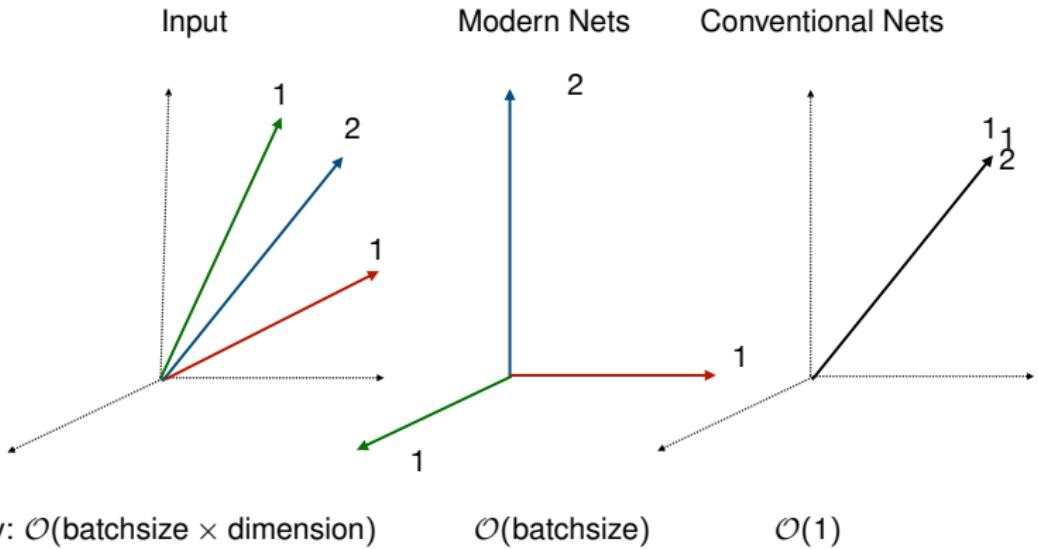


Fast optimization

# Mild compression with modern networks



16



# Batch Normalization (BN)



BN is the fundamental component of modern neural networks.

---

<sup>6</sup>Santurkar et al. NeurIPS18, Bjorck et al. NeurIPS18, Arora et al. ICLR19, Yao et al. IEEE BigData20, Sun et al. AAAI20, Lubana et al. NeurIPS21

<sup>7</sup>**Daneshmand, H., Joudaki, A. & Bach, F.** Batch Normalization Orthogonalizes Representations in Deep Random Networks. *NeurIPS (2021)*.

# Batch Normalization (BN)



17

BN is the fundamental component of modern neural networks.

The underlying mechanisms of BN has been

- ▶ a fundamental open problem in machine learning

---

<sup>6</sup>Santurkar et al. NeurIPS18, Bjorck et al. NeurIPS18, Arora et al. ICLR19, Yao et al. IEEE BigData20, Sun et al. AAAI20, Lubana et al. NeurIPS21

<sup>7</sup>Daneshmand, H., Joudaki, A. & Bach, F. Batch Normalization Orthogonalizes Representations in Deep Random Networks. *NeurIPS* (2021).

# Batch Normalization (BN)



BN is the fundamental component of modern neural networks.

The underlying mechanisms of BN has been

- ▶ a fundamental open problem in machine learning
- ▶ explored by recent researches in machine learning<sup>6</sup>

**Compression.** *BN makes representations increasingly orthogonal across layers<sup>7</sup>.*

---

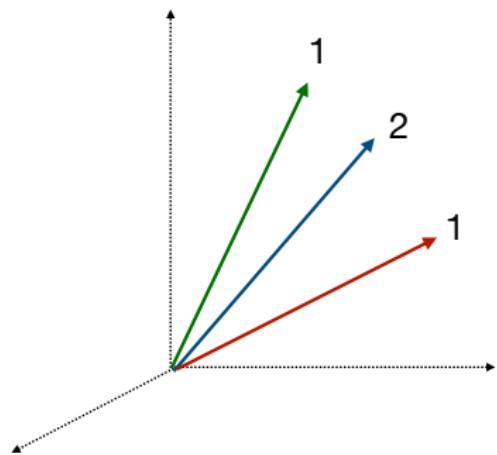
<sup>6</sup>Santurkar et al. NeurIPS18, Bjorck et al. NeurIPS18, Arora et al. ICLR19, Yao et al. IEEE BigData20, Sun et al. AAAI20, Lubana et al. NeurIPS21

<sup>7</sup>Daneshmand, H., Joudaki, A. & Bach, F. Batch Normalization Orthogonalizes Representations in Deep Random Networks. *NeurIPS (2021)*.

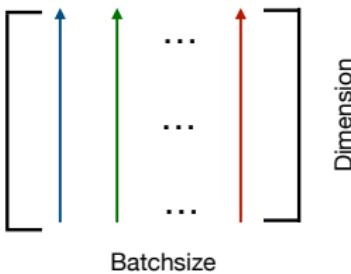
# The Markov chain of representations



18



$$H_0 =$$



Representations

$$\widehat{H}_{\ell+1} = BN(\widehat{W}_\ell H_\ell)$$

$$BN(M) = (\text{diag}(MM^\top))^{-1/2} M$$

BN(.) makes the rows (features) unit norm.

# Coupled representations



19

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

# Coupled representations



19

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

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  $\ell \rightarrow \infty$ .

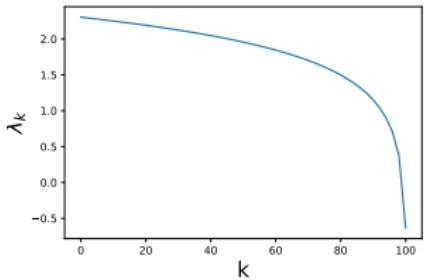
# Lyapunov exponents



- ▶ Definition:

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

- ▶ Computation<sup>8</sup>:  $\lambda_k = \frac{1}{2}(\log(2) + \Psi(\frac{d-k+1}{2}))$



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

<sup>8</sup>Newman, C. M. The distribution of Lyapunov exponents: Exact results for random matrices. *Communications in mathematical physics* (1986).

# Modern NN with Batch normalization (BN)



22

- ▶ BN is one of the main building block of modern neural networks<sup>9</sup>
- ▶ Representation  $H_\ell : \text{width} \times \text{batchsize}$ .

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

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

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

- ▶ Learning only parameters  $\alpha$  and  $\beta$  (per unit) leads to surprisingly good performance<sup>10</sup>

---

<sup>9</sup>Ioffe, S. & Szegedy, C. *Batch normalization: Accelerating deep network training by reducing internal covariate shift.* in *ICML* (2015).

<sup>10</sup>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<sup>1112</sup>.

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

---

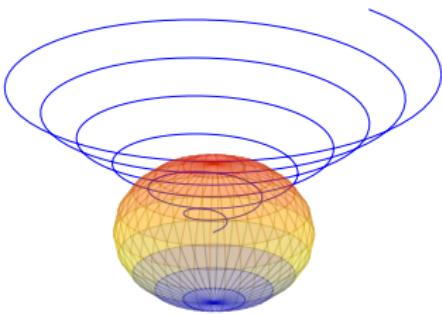
<sup>11</sup>[Daneshmand, H., Joudaki, A. & Bach, F.](#) Batch Normalization Orthogonalizes Representations in Deep Random Networks. *NeurIPS* (2021).

<sup>12</sup>[Daneshmand\\*, H. et al.](#) Batch Normalization Provably Avoids Rank Collapse for Randomly Initialised Deep Networks. *NeurIPS* (2020).

# Theoretical results



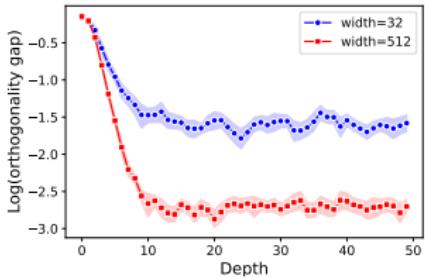
- ▶  $\mathbf{E} \left[ \text{orthogonality gap}(H_\ell) \right] = \mathcal{O} \left( (1 - \alpha)^\ell + \frac{\text{batchsize}}{\alpha \sqrt{\text{width}}} \right)$
- ▶  $\text{Wasser._2} (W_\ell H_\ell, \text{Gaussian})^2 = \mathcal{O} \left( (1 - \alpha)^\ell (\text{batchsize}) + \frac{(\text{batchsize})^2}{\alpha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



# Orthogonalization



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



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

# Modern NN vs. historical NN



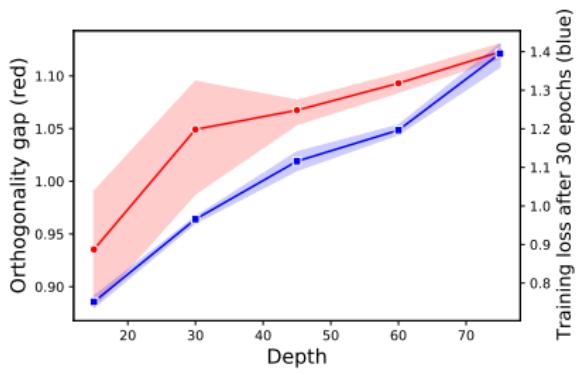
BN	Without BN

# Modern NN vs. historical NN

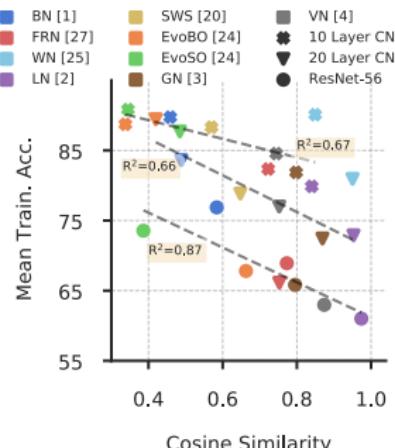


BN	Without BN
$\mathbf{E} \left[ \text{orth. gap}(H_\infty) \right] = \mathcal{O} \left( \frac{\text{batch size}}{\alpha \sqrt{\text{width}}} \right)$	$\mathbf{E} \left[ \text{orth. gap}(H'_\infty) \right] = \Theta(1)$

# The orthogonality influences training



Without normalization



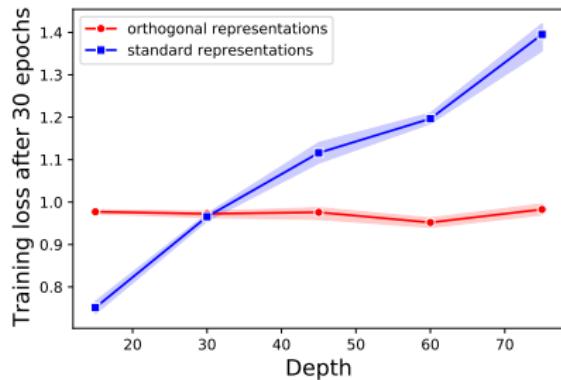
With Normalization<sup>13</sup>

<sup>13</sup>lubana2021beyond.

# Replacing BN with orthogonalization



Saving training time by starting from orthogonal representations



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

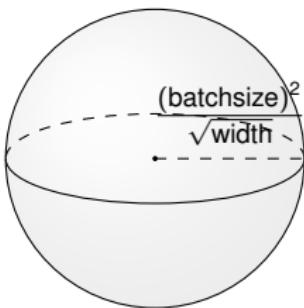
Red: standard initialization with low orthogonality gaps

Blue: novel initialization ensuring orthogonal representations

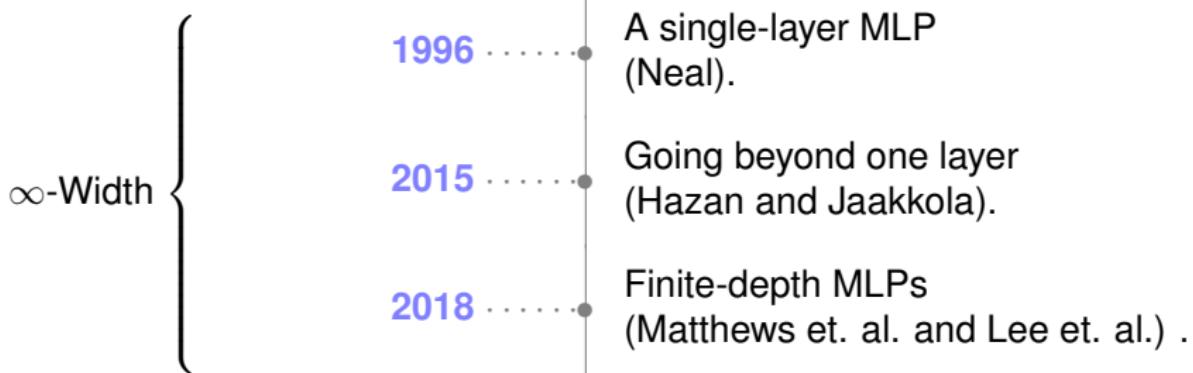
# Gaussian approximation



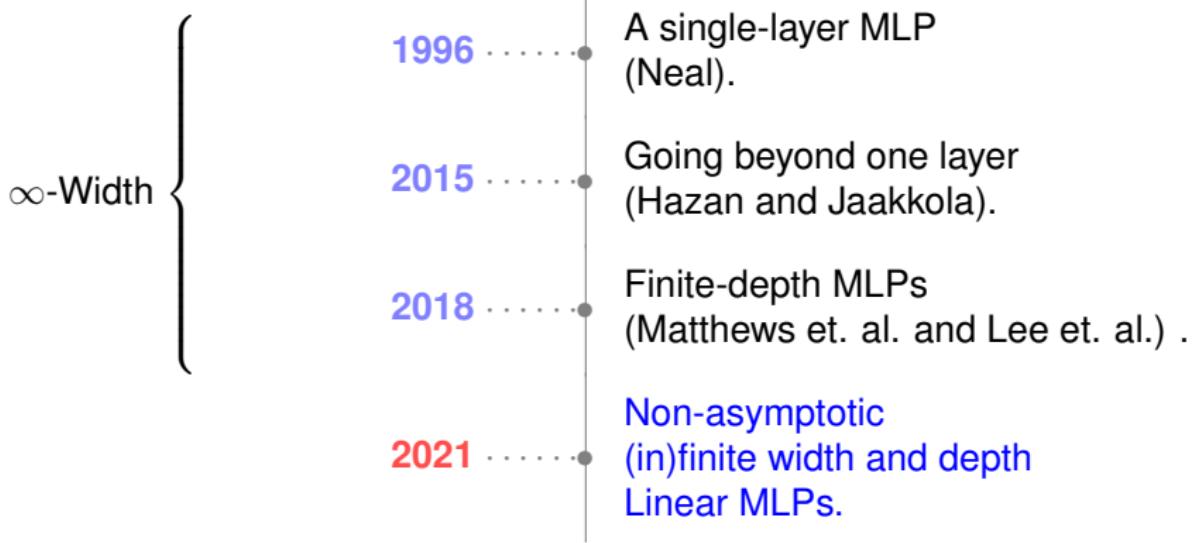
$$\text{Wasserstein}_2(W_\ell H_\ell, \text{Gaussian})^2 = \mathcal{O}\left((1 - \alpha)^\ell (\text{batchsize}) + \frac{(\text{batchsize})^2}{\alpha \sqrt{\text{width}}}\right)$$



# History of Gaussian approximation for NNs



# History of Gaussian approximation for NNs



# Applications of the Gaussian approximation



# Vision



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