Mathematics of Deep Learning

René Vidal
Herschel Seder Professor of Biomedical Engineering
Director of the Mathematical Institute for Data Science
Johns Hopkins University

JHU vision lab

THE DEPARTMENT OF BIOMEDICAL ENGINEERING
The Whitaker Institute at Johns Hopkins

JOHNS HOPKINS
MATHEMATICAL INSTITUTE *for* DATA SCIENCE

Brief History of Neural Networks

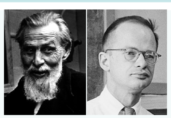
Beginnings

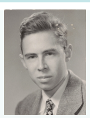
Thresholded Logic Unit — 1943

Perceptron — 1957

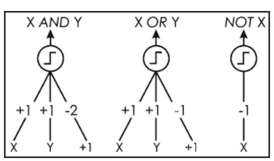
Adaline — 1960

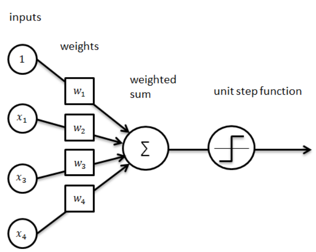
1940 **1950** **1960**


S. McCulloch - W. Pitts


R. Rosenblatt


B. Widrow - M. Hoff



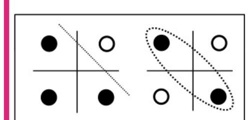


1st Neural Winter

XOR Problem — 1969

1970


M. Minsky - S. Papert

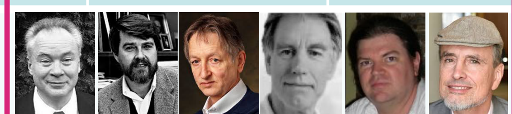


Multilayer Backprop — 1982, 1986

CNNs — 1989

LSTMs — 1997

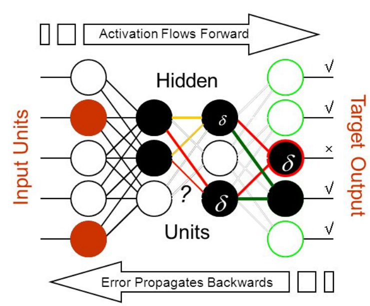
1980 **1990**


P. Werbos

D. Rumelhart - G. Hinton - R. Williams

Y. Lecun

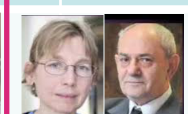
J. Schmidhuber

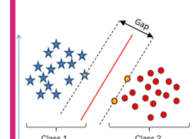


2nd Neural Winter

SVMs — 1995

2000


C. Cortes - V. Vapnik

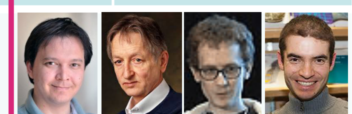


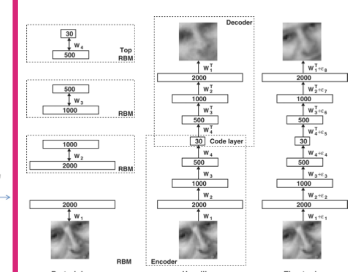
GPU Era

Deep Nets — 2006

Alex Net — 2012

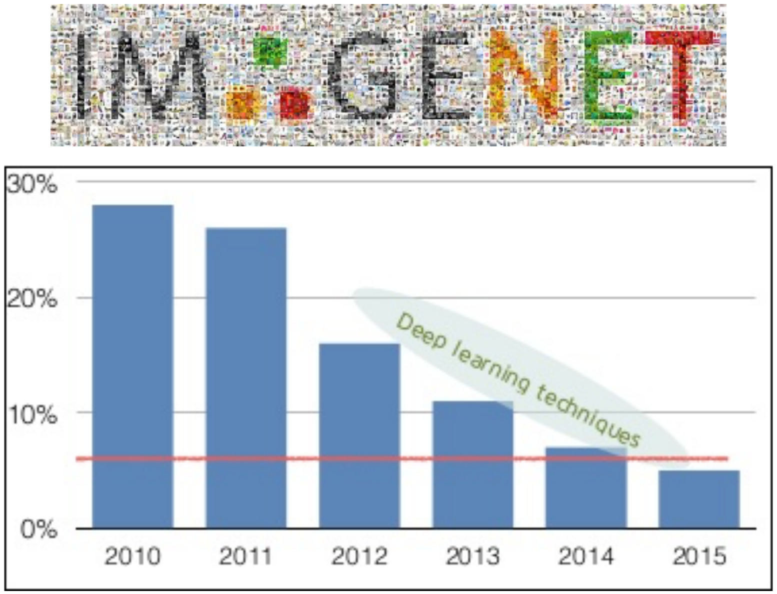
2010

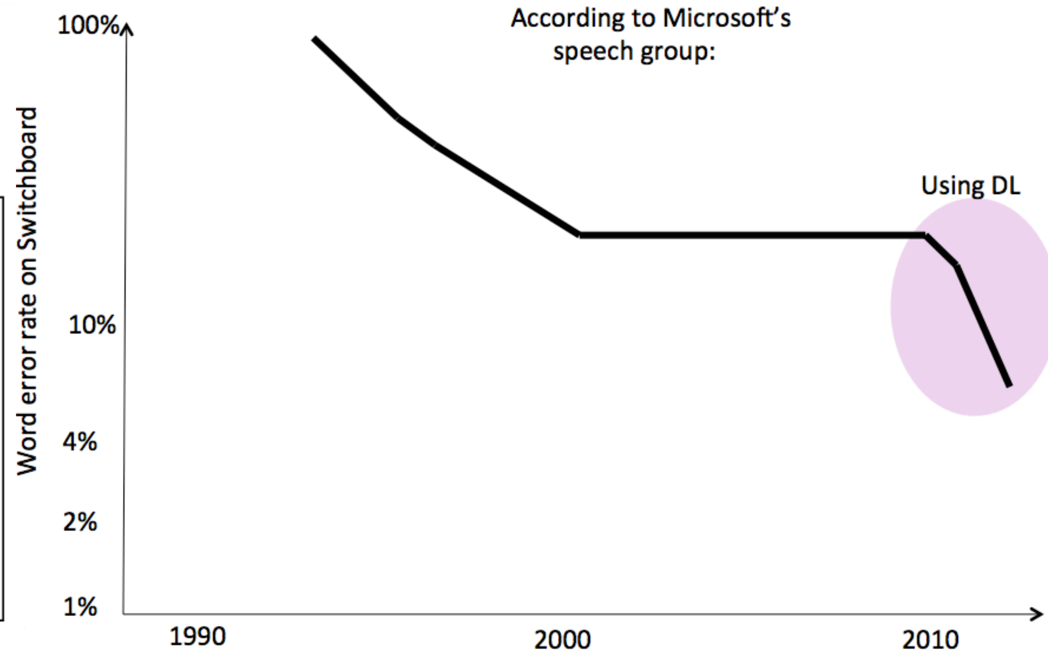

R. Salakhutdinov - J. Hinton - A. Krizhevsky - I. Sutskever

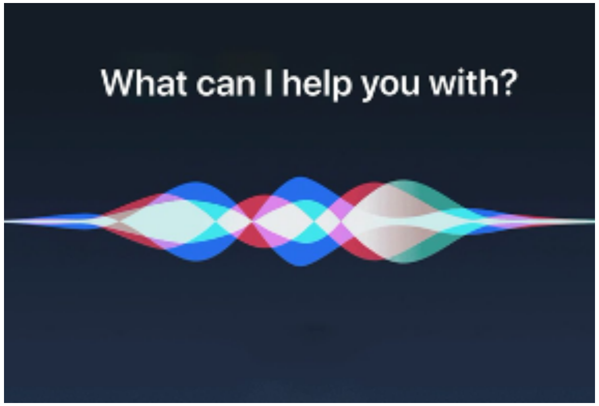


JOHNS HOPKINS
MATHEMATICAL INSTITUTE for DATA SCIENCE

Impact of Deep Learning in AI



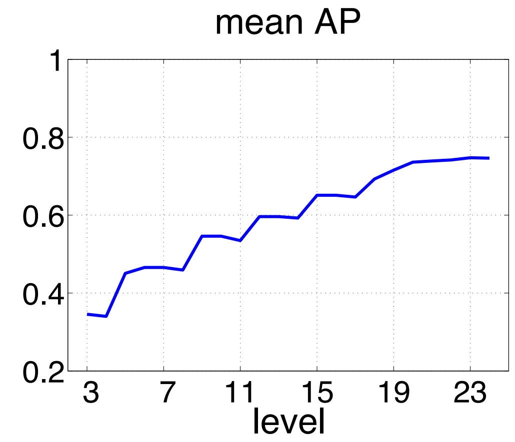


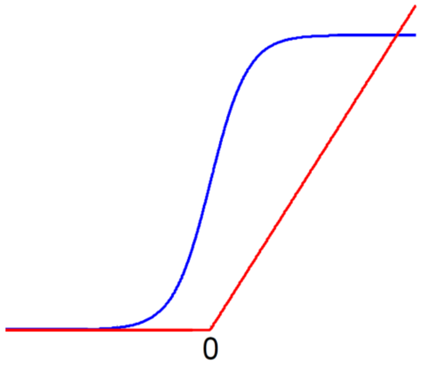


Silver et al. Mastering the game of Go with deep neural networks and tree search, Nature 2016
Artificial intelligence learns Mario level in just 34 attempts, https://www.engadget.com/2015/06/17/super-mario-world-self-learning-ai/, https://github.com/aleju/mario-ai

Why These Improvements in Performance?

- Features are learned rather than hand-crafted
- More layers capture more invariances [1]
- More data to train deeper networks
- More computing (GPUs)
- Better regularization: Dropout
- New nonlinearities
  - Max pooling, Rectified linear units (ReLU) [2]
- Theoretical understanding of deep networks remains shallow

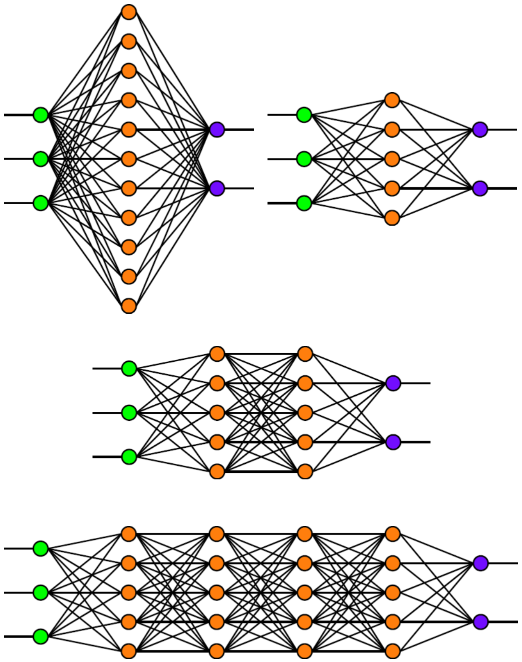




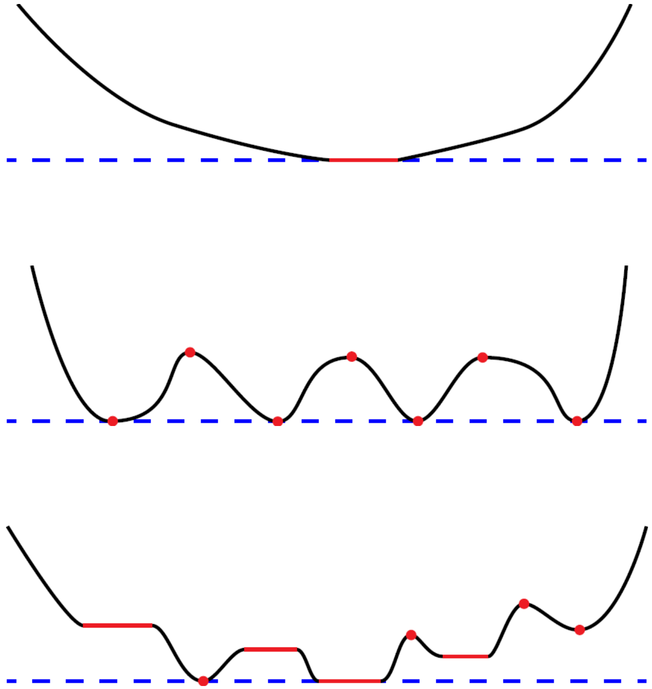
[1] Razavian, Azizpour, Sullivan, Carlsson, CNN Features off-the-shelf: an Astounding Baseline for Recognition. CVPRW'14.
[2] Hahnloser, Sarpeshkar, Mahowald, Douglas, Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature, 405(6789):947–951, 2000.

Key Theoretical Questions in Deep Learning

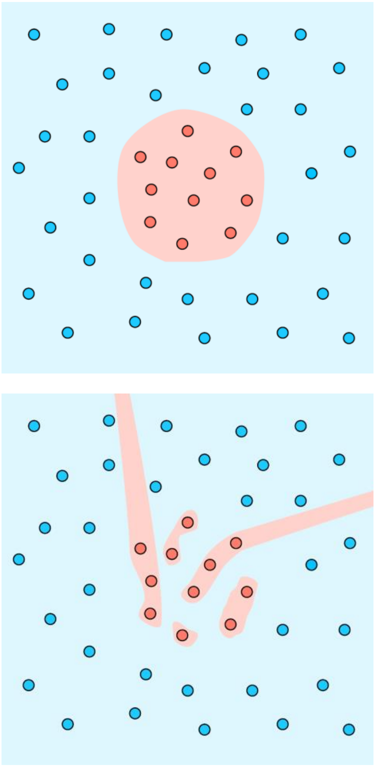
Architecture Design



Optimization



Generalization



Slide courtesy of Ben Haeffele

JOHNS HOPKINS
MATHEMATICAL INSTITUTE
for DATA SCIENCE

Key Theoretical Questions: Architecture

- **Approximation, depth, width and invariance: earlier work**
  - Perceptrons and multilayer feedforward networks are universal approximators [Cybenko '89, Hornik '89, Hornik '91, Barron '93]

Theorem [C'89, H'91] Let $\rho()$ be a bounded, non-constant continuous function. Let I_m denote the m-dimensional hypercube, and $C(I_m)$ denote the space of continuous functions on I_m. Given any $f \in C(I_m)$ and $\epsilon > 0$, there exists $N > 0$ and v_i, w_i, b_i, $i = 1 \ldots, N$ such that

$$F(x) = \sum_{i \leq N} v_i \rho(w_i^T x + b_i) \quad \text{satisfies}$$

$$\sup_{x \in I_m} |f(x) - F(x)| < \epsilon \; .$$

Key Theoretical Questions: Architecture

- **Approximation, depth, width and invariance: earlier work**
  - Perceptrons and multilayer feedforward networks are universal approximators [Cybenko '89, Hornik '89, Hornik '91, Barron '93]
- **Approximation, depth, width and invariance: recent work**
  - Gaps between deep and shallow networks [Montufar'14, Mhaskar'16]
  - Deep Boltzmann machines are universal approximators [Montufar'15]
  - Design of CNNs via hierarchical tensor decompositions [Cohen '17]
  - Scattering networks are deformation stable for Lipschitz non-linearities [Bruna-Mallat '13, Wiatowski '15, Mallat '16]
  - Exponential # of units needed to approximate deep net [Telgarsky'16]
  - Approximation with sparsely connected deep networks [Bölcskei '19]
  - Representation power of GNNs [Jegelka'18]

[1] Cybenko. Approximations by superpositions of sigmoidal functions, Mathematics of Control, Signals, and Systems, 2 (4), 303-314, 1989.
[2] Hornik, Stinchcombe and White. Multilayer feedforward networks are universal approximators, Neural Networks, 2(3), 359-366, 1989.
[3] Hornik. Approximation Capabilities of Multilayer Feedforward Networks, Neural Networks, 4(2), 251–257, 1991.
[4] Barron. Universal approximation bounds for superpositions of a sigmoidal function. IEEE Transactions on Information Theory, 39(3):930–945, 1993.
[5] Cohen et al. Analysis and Design of Convolutional Networks via Hierarchical Tensor Decompositions arXiv preprint arXiv:1705.02302
[6] Montúfar, Pascanu, Cho, Bengio, On the number of linear regions of deep neural networks, NIPS, 2014
[7] Mhaskar, Poggio. Deep vs. shallow networks: An approximation theory perspective. Analysis and Applications, 2016.
[8] Montúfar et al, Deep narrow Boltzmann machines are universal approximators, ICLR 2015, arXiv:1411.3784v3
[9] Bruna and Mallat. Invariant scattering convolution networks. Trans. PAMI, 35(8):1872–1886, 2013.
[10] Wiatowski, Bölcskei. A mathematical theory of deep convolutional neural networks for feature extraction. arXiv2015.
[11] Mallat. Understanding deep convolutional networks. Phil. Trans. R. Soc. A, 374(2065), 2016.
[12] Telgarsky, Benefits of depth in neural networks. COLT 2016.
[13] Bölcskei, Grohs, Kutyniok, Petersen. Optimal approximation with sparsely connected deep neural networks. SIAM J. Math of Data Science, 2019

Key Theoretical Questions: Optimization

- **Optimization theory: earlier work**
  - No spurious local minima for linear networks [Baldi-Hornik'89, Nouiehed'18, Zhu']
  - Backprop fails to converge for nonlinear networks [Brady'89], converges for linearly separable data [Gori-Tesi'91-'92], or it gets stuck [Frasconi'97]
  - Local minima and plateaus in multilayer perceptrons [Fukumizu-Amari'00]
- **Optimization theory: recent work on landscape**
  - Convex neural networks in infinite number of variables [Bengio '05]
  - No spurious local minima for deep linear networks and square loss [Kawaguchi'16]
  - No spurious local minima for positively homogeneous networks [Haeffele-Vidal'15 '17], but infinitely many local minima in general [Yun '19]
  - Role of level sets on spurious valleys [Venturi '18, Nguyen'18'19, Kuditipudi '19]
  - Statistical physics-based analysis of the landscape of two-layer neural networks [Mei '18 '19] and multilayer networks [Choromanska '15, Verpoort-Lee-Wales '20]

[1] Baldi, Hornik, Neural networks and principal component analysis: Learning from examples without local minima, Neural networks, 1989.
[2] Brady, Raghavan, J Slawny. Back propagation fails to separate where perceptrons succeed. IEEE Trans Circuits & Systems, 36(5):665–674, 1989.
[3] Gori, Tesi. On the problem of local minima in backpropagation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 14(1):76–86, 1992.
[4] Frasconi, Gori, Tesi. Successes and failures of backpropagation: A theoretical. Progress in Neural Networks: Architecture, 5:205, 1997.
[5] Fukumizu, Amari. Local minima and plateaus in multilayer perceptrons. Neural Networks, 2000.
[6] Bengio, Le Roux, Vincent, Delalleau, Marcotte. Convex Neural Networks. NeurIPS, 2005
[7] Kawaguchi. Deep learning without poor local minima. NeurIPS, 2016.
[8] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, 2015.
[9] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.
[10] Yun, Sra, Jadbabaie. Small nonlinearities in activation functions create bad local minima in neural networks. ICLR 2019.
[11] Y Cooper. The loss landscape of overparameterized neural networks. arXiv:1804.10200, 2018.
[12] Venturi, A. S. Bandeira, and J. Bruna. Spurious valleys in two-layer neural network optimization landscapes. arXiv preprint arXiv:1802.06384, 2018.
[13] Nguyen. On connected sublevel sets in deep learning. arXiv preprint arXiv:1901.07417, 2019.
[14] Nguyen, Mukkamala, Hein. On the loss landscape of a class of deep neural networks with no bad local valleys. arXiv preprint arXiv:1809.10749, 2018.
[15] Kuditipudi, Wang, Lee, Zhang, Li, Hu, Ge, Arora. Explaining landscape connectivity of low-cost solutions for multilayer nets. NeurIPS, 2019.
[16] Mei, Montanari, Nguyen. A mean field view of the landscape of two-layer neural networks. PNAS, 115(33):E7665–E7671, 2018.
[17] Mei, Misiakiewicz, Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. COLT, 2019
[18] Verpoort, Lee, Wales. Archetypal landscapes for deep neural networks. PNAS, 2020.

Key Theoretical Questions: Optimization

- **Optimization theory: recent work on algorithms**
  - GD on networks with many hidden units can learn polynomials [Andoni ’14]
  - Attacking the saddle point problem [Dauphin ’14]
  - Effect of noise and BN on the landscape [Santurkar’18, Chaudhari’15, Soudry ’16]
  - Entropy-SGD is biased toward wide valleys [Chaudhari ’17]
  - Deep relaxation: PDEs for optimizing deep nets [Chaudhari ’18]
  - Guaranteed training of NNs using tensor methods [Janzamin ’16]
  - Convergence of GD for deep linear neural networks [Arora ’18]
  - Implicit acceleration by over-parameterization [Arora ’18, Tarmoun ’20]
  - Benign landscape [Fang ’19] and convergence of gradient methods in overparametrized models [Chizat ’18, Li ’18, Du ’19, Allen-Zhu’19, Zou ’19]
  - Mean-field and learning dynamics [Nguyen ’19]

[1] Andoni, Panigrahy, Valiant, Zhang. Learning polynomials with neural networks. ICML 2014.
[2] Dauphin, Pascanu, Gulcehre, Cho, Ganguli, Bengio, Identifying and attacking the saddle point problem in high-dimensional non- convex optimization, NeurIPS 2014.
[3] Santurkar, Tsipras, Ilyas, Madry. How does batch normalization help optimization? NeurIPS, 2018.
[4] Soudry, Y Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. arXiv preprint arXiv:1605.08361, 2016.
[5] Chaudhari, Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. ICLR 2018.
[6] Chaudhari, Choromanska, Soatto, LeCun, Baldassi, Borgs, Chayes, Sagun, Zecchina. Entropy-SGD: biasing gradient descent into wide valleys. ICLR 2016, JSM 2019.
[7] Chaudhari, A Oberman, S Osher, S Soatto, G Carlier. Deep relaxation: partial differential equations for optimizing deep neural networks. RMS 2018
[8] Janzamin, Sedghi, Anandkumar, Beating the Perils of Non-Convexity: Guaranteed Training of Neural Networks using Tensor Methods, arXiv:1506.08473, 2016.
[9] Arora, Cohen, Golowich, Hu. A convergence analysis of gradient descent for deep linear neural networks. arXiv preprint arXiv:1810.02281, 2018.
[10] Arora, Cohen, Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. arXiv preprint arXiv:1802.06509, 2018.
[11] Tarmoun, Franca, Haeffele, Vidal. Implicit Acceleration of Gradient Flow in Overparameterized Linear Models.
[12] Fang, Gu, Zhang, Zhang. Convex formulation of overparameterized deep neural networks. arXiv preprint arXiv:1911.07626, 2019.
[13] Chizat, Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. NeurIPS, 2018.
[14] Li, Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. NeurIPS, 2018.
[15] Du, Zhai, Poczos, Singh. Gradient descent provably optimizes over-parameterized neural networks. ICLR, 2019.
[16] Du, Lee, Li, Wang, Zhai. Gradient descent finds global minima of deep neural networks. ICML, 2019.
[17] Allen-Zhu, Li, Song. A convergence theory for deep learning via over-parameterization. ICML, 2019.
[18] Zou, Cao, Zhou, Gu. Gradient descent optimizes over-parameterized deep ReLU networks. Machine Learning 2019
[19] Zou, Gu. An improved analysis of training over-parameterized deep neural networks. NeurIPS, 2019.
[20] Nguyen. Mean field limit of the learning dynamics of multilayer neural networks. arXiv preprint arXiv:1902.02880, 2019.
[21] Dogra, Redman. Optimizing Neural Networks via Koopman Operator Theory, 2020.

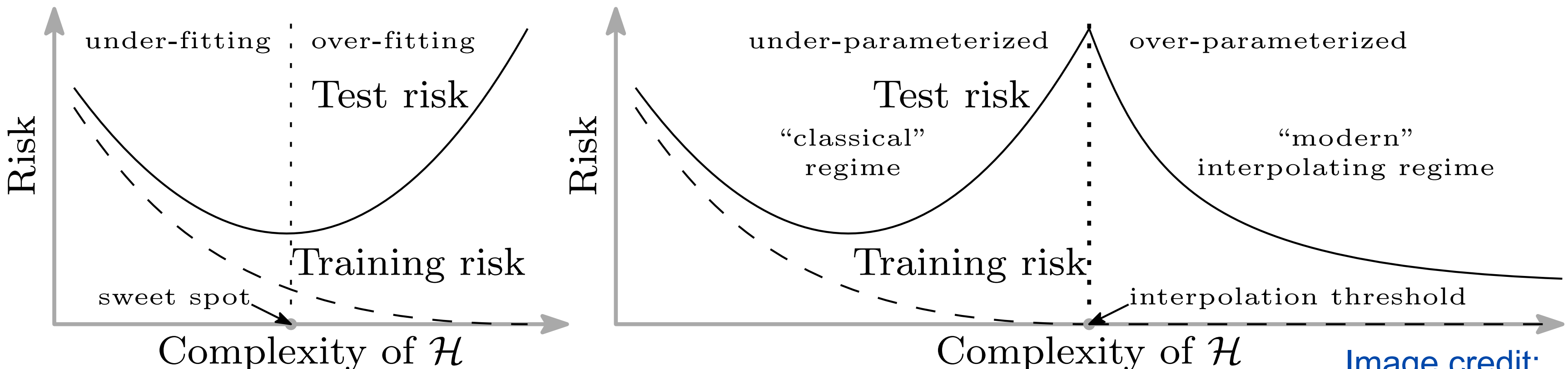
Key Theoretical Questions: Generalization

- **Generalization and regularization theory: earlier work**
  - # training examples grows polynomially with network size [1,2]
- **Regularization methods: earlier and recent work**
  - Early stopping [3]
  - Dropout, Dropconnect, Dropblock and extensions (adaptive, annealed) [4,5]
  - Batch normalization [6]
- **Generalization and regularization theory: recent work**
  - Distance and margin-preserving embeddings [7,8]
  - Path SGD/implicit regularization & generalization bounds [9,10]
  - Product of norms regularization & generalization bounds [11,12]
  - Information theory: info bottleneck, info dropout, Fisher-Rao [13,14,15]
  - Rethinking generalization: [16]

[1] Sontag. VC Dimension of Neural Networks. Neural Networks and Machine Learning, 1998.
[2] Bartlett, Maass. VC dimension of neural nets. The handbook of brain theory and neural networks, 2003.
[3] Caruana, Lawrence, Giles. Overfitting in neural nets: Backpropagation, conjugate gradient & early stopping. NeurIPS 2001.
[4] Srivastava. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. JMLR, 2014.
[5] Wan. Regularization of neural networks using dropconnect. ICML, 2013.
[6] Ioffe, Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv preprint arXiv:1502.03167, 2015
[7] Giryes, Sapiro, Bronstein. Deep Neural Networks with Random Gaussian Weights. arXiv:1504.08291.
[8] Sokolic. Margin Preservation of Deep Neural Networks, 2015
[9] Neyshabur. Path-SGD: Path-Normalized Optimization in Deep Neural Networks. NIPS 2015
[10] Behnam Neyshabur. Implicit Regularization in Deep Learning. PhD Thesis 2017
[11] Sokolic, Giryes, Sapiro, Rodrigues. Generalization error of invariant classifiers. In AISTATS, 2017.
[12] Sokolić, Giryes, Sapiro, Rodrigues. Robust Large Margin Deep Neural Networks. IEEE Transactions on Signal Processing, 2017.
[13] Shwartz-Ziv, Tishby. Opening the black box of deep neural networks via information. arXiv:1703.00810, 2017.
[14] Achille, Soatto. Information dropout: Learning optimal representations through noisy computation. *arXiv:* 2016.
[15] Liang, Poggio, Rakhlin, Stokes. Fisher-Rao Metric, Geometry and Complexity of Neural Networks. arXiv: 2017.
[16] Zhang, Bengio, Hardt, Recht, Vinyals. Understanding deep learning requires rethinking generalization. ICLR 2017.

Key Theoretical Questions: Generalization

- **Generalization and regularization theory: recent work**
  - Implicit regularization of dropout [Cavazza'18, Mianjy'18, Pal'20, Arora'20], batch normalization [Schilling'16, De'20] & GD [Arora'19] in matrix factorization/deep nets
  - Neural tangent kernel (NTK) [Jacot'18, Chizat'19, Arora'19, Wei'19, Ghorbani '20]
  - Over-parametrization can improve generalization [Belkin'19, Allen-Zhu'18, Arora'19, Fang '19, Montanari'19 '20, Cao'19]



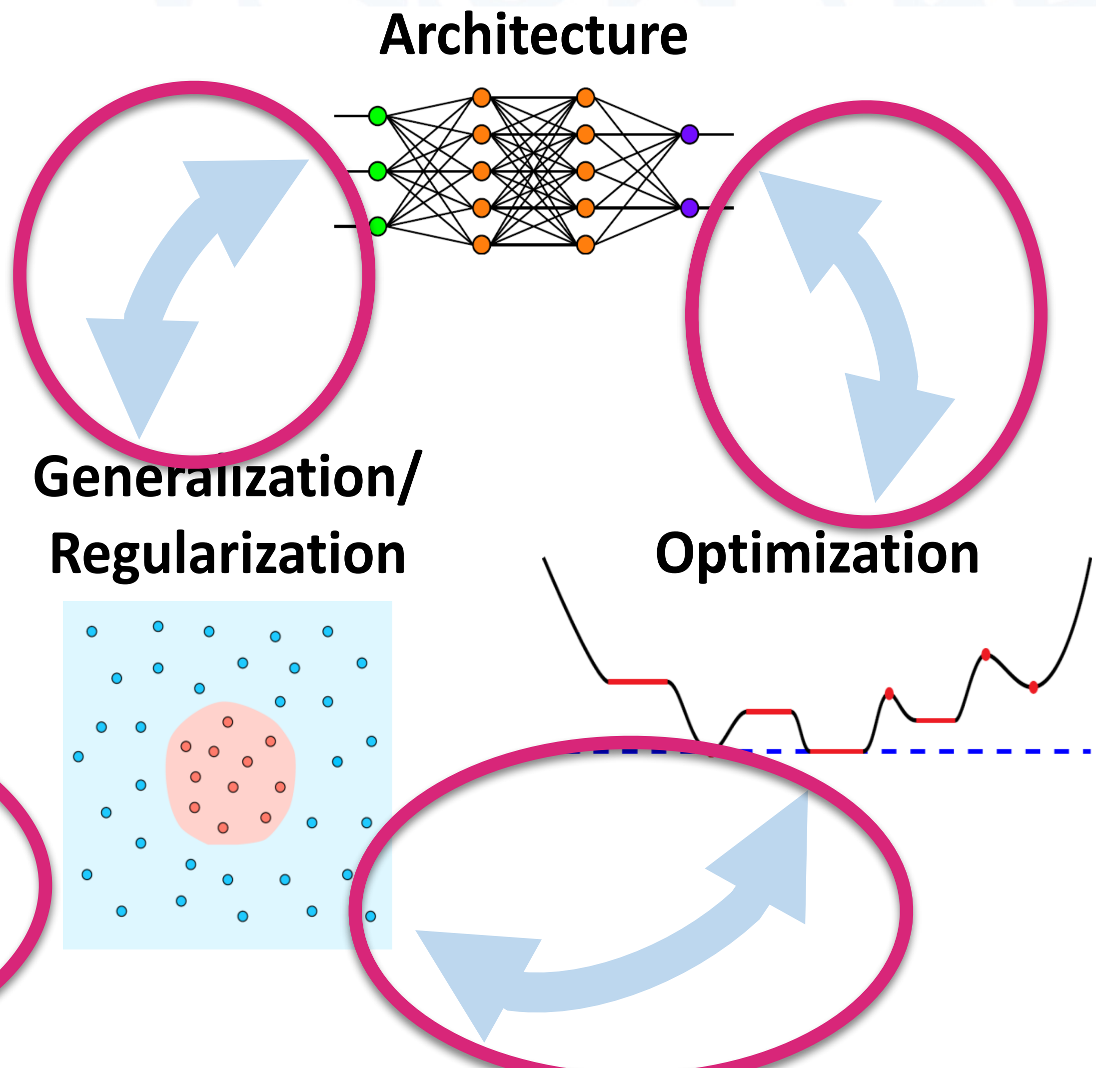
(a) U-shaped "bias-variance" risk curve (b) "double descent" risk curve

Image credit: Mikhail Belkin

[1] Cavazza, Haeffele, Morerio, Lane, Murino, Vidal, Dropout as a Low-Rank Regularizer for Matrix Factorization, AISTATS (2018), https://arxiv.org/abs/1710.03487
[2] Mianjy, Arora, Vidal, On the Implicit Bias of Dropout, ICML (2018), https://arxiv.org/abs/1806.09777
[3] Pal, Lane, Vidal, Haeffele. On the Regularization Properties of Structured Dropout, CVPR (2020). https://arxiv.org/abs/1910.14186
[4] Arora, Bartlett, Mianjy, Srebro. Dropout: Explicit Forms and Capacity Control. arXiv:2003.03397, 2020.
[5] Schilling. The effect of batch normalization on deep convolutional neural networks, 2016.
[6] De, Smith. Batch Normalization Biases Residual Blocks Towards the Identity Function in Deep Networks, 2020.
[7] Jacot, Gabriel, Hongler. Neural tangent kernel: Convergence and generalization in neural networks. NeurIPS, 2018.
[8] Chizat, Oyallon, Bach. On lazy training in differentiable programming. NeurIPS, 2019.
[9] Arora, Du, Hu, Li, Salakhutdinov, Wang. On exact computation with an infinitely wide neural net. arXiv preprint arXiv:1904.11955, 2019.
[10] Wei, Lee, Liu, Ma. Regularization matters: Generalization and optimization of neural nets v.s. their induced kernel. NeurIPS, 2019.
[11] Ghorbani, Mei, Misiakiewicz, Montanari. When Do Neural Networks Outperform Kernel Methods? arXiv preprint arXiv:2006.13409, 2020.
[12] Belkin, Hsu, Ma, Mandal, Reconciling modern machine-learning practice and the classical bias–variance trade-off. PNAS, 2019.
[13] Allen-Zhu, Li, Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. arXiv preprint arXiv:1811.04918, 2018.
[14] Arora, Du, Hu, Li, Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. ICML, 2019.
[15] Fang, Dong, Zhang. Over parameterized two-level neural networks can learn near optimal feature representations. arXiv preprint arXiv:1910.11508, 2019.
[17] Montanari, Ruan, Sohn, Yan. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. arXiv 2019
[18] Montanari, Zhong. The interpolation phase transition in neural networks: Memorization and generalization under lazy training. arXiv preprint arXiv:2007.12826. 2020
[19] Cao, Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. NeurIPS, 2019.

Key Theoretical Questions are Interrelated

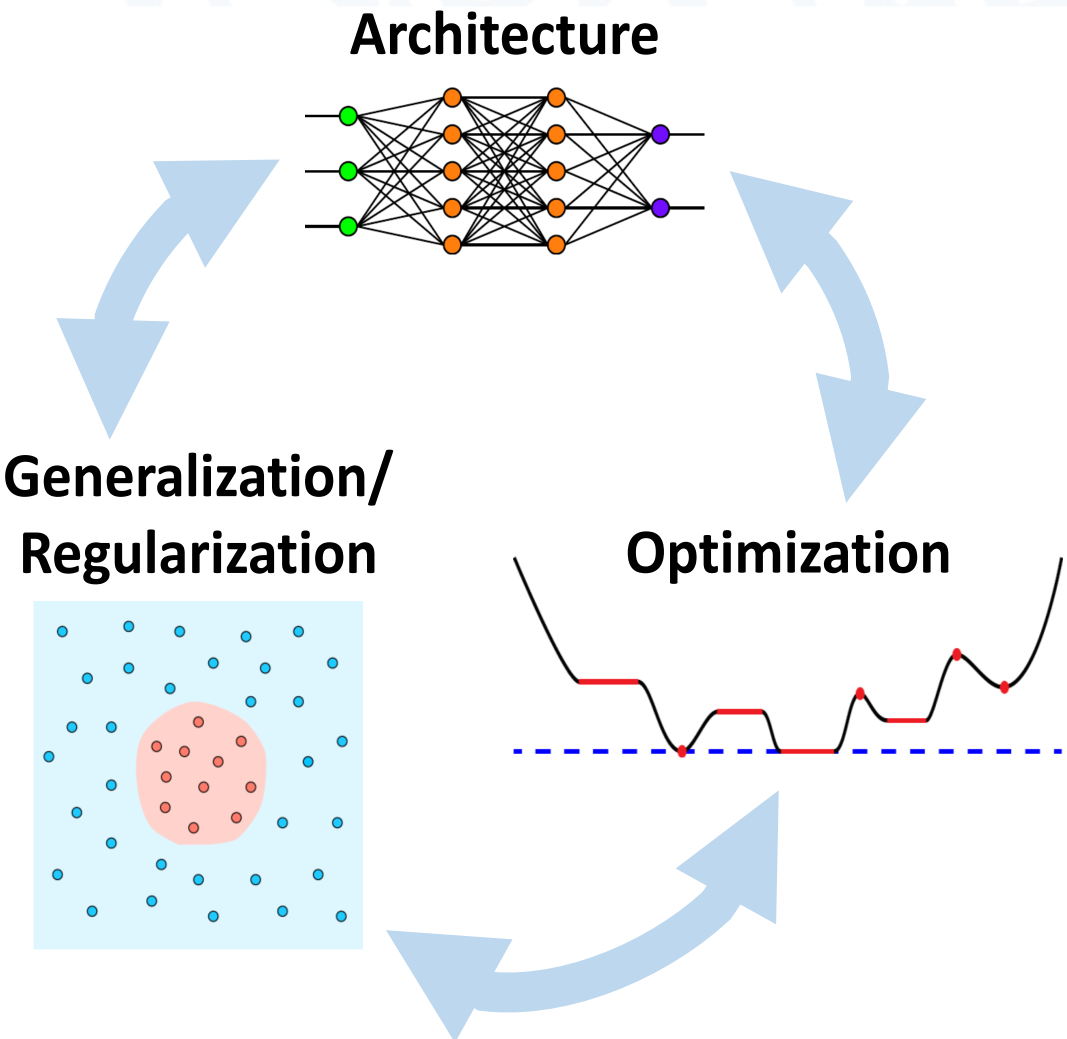
- Optimization can impact generalization [1,2]
- Architecture has strong effect on generalization [3]
- Some architectures could be easier to optimize than others [4]



[1] Neyshabur et. al. In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning." ICLR workshop. (2015).
[2] P. Zhou, J. Feng. The Landscape of Deep Learning Algorithms. 1705.07038, 2017
[3] Zhang, et al., "Understanding deep learning requires rethinking generalization." ICLR. (2017).
[4] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

Toward a Unified Theory?

- Dropout regularization is equivalent to regularization with products of weights [1,2]
- Regularization with product of weights generalizes well [3,4]
- No spurious local minima for product of weight regularizers [5]



[1] Cavazza, Lane, Moreiro, Haeffele, Murino, Vidal. An Analysis of Dropout for Matrix Factorization, AISTATS 2018.
[2] Poorya Mianjy, Raman Arora, Rene Vidal. On the Implicit Bias of Dropout. ICML 2018.
[3] Neyshabur, Salakhutdinov, Srebro. Path-SGD: Path-Normalized Optimization in Deep Neural Networks. NIPS 2015
[4] Sokolic, Giryes, Sapiro, Rodrigues. Generalization error of Invariant Classifiers. AISTATS, 2017.
[5] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

Outline

- **Part I: Optimization Landscape of Linear Networks**
  - All local minima are global
  - Other critical points are saddle points
  - All saddles are strict for one hidden layer
  - Non-strict saddles exist for deeper networks
- **Part II: Optimization Landscape of Positively Homogeneous Networks**
  - If network is wide enough, all local minima are global
  - One can escape local minima by increasing the size of the network
- **Part III: Analysis of Dropout, DropConnect, DropBlock**
  - Dropout is SGD applied to a regularized objective
  - Dropout induces low-rank and balanced solutions
  - Dropblock induces r-support norm regularization

[1] Baldi, Hornik, Neural networks and principal component analysis: Learning from examples without local minima, Neural networks, 1989.
[2] Nouiehed, Razaviyayn. Learning deep models: Critical points and local openness. arXiv preprint arXiv:1803.02968, 2018
[3] Zhu, Soudry, Eldar, Wakin. The Global Optimization Geometry of Shallow Linear Neural Networks. JMIV, 2019.
[4] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[5] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[6] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

Landscape of Linear Networks

- All local minima are global
- Other critical points are saddles
- All saddles are strict for one hidden layer
- Non-strict saddles exist for deeper networks

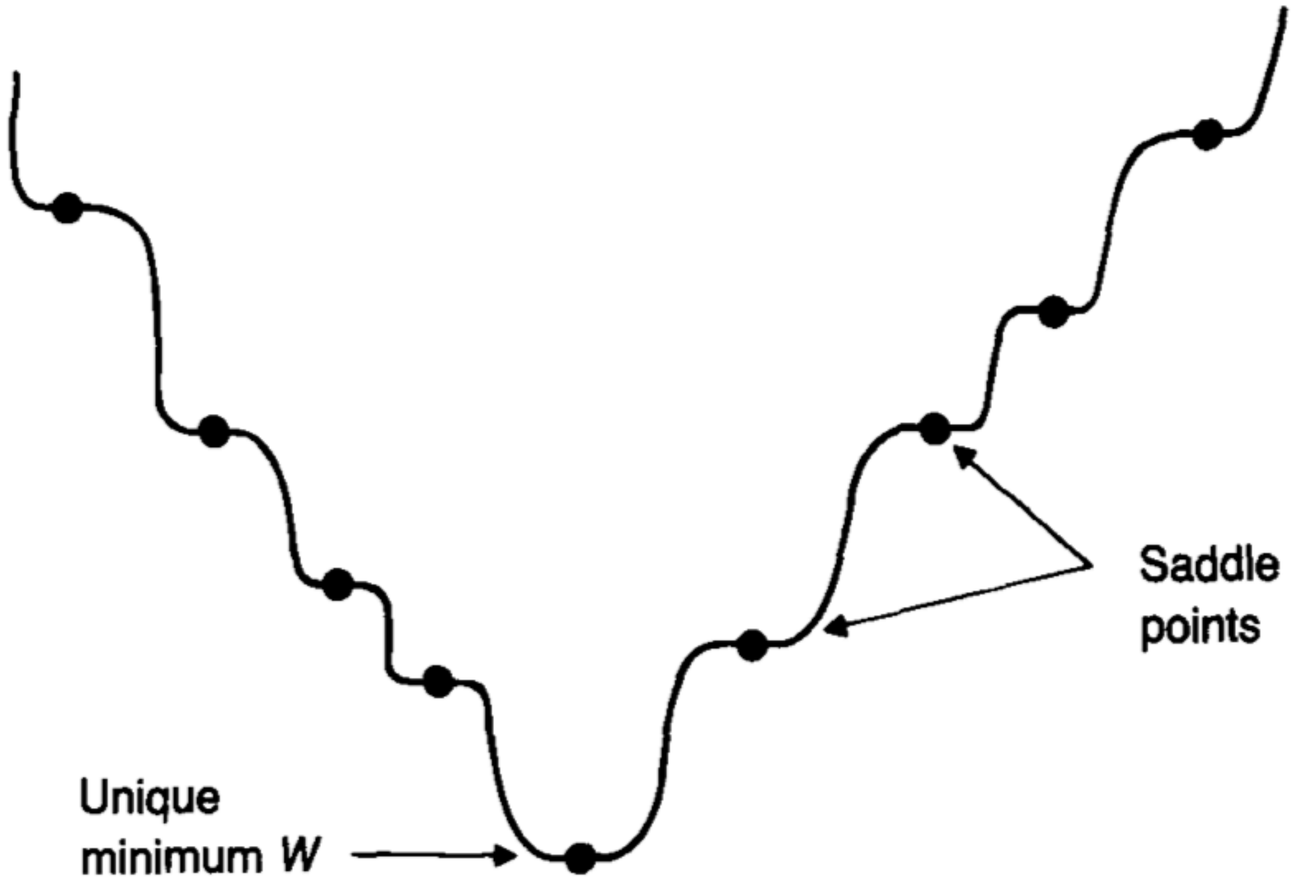
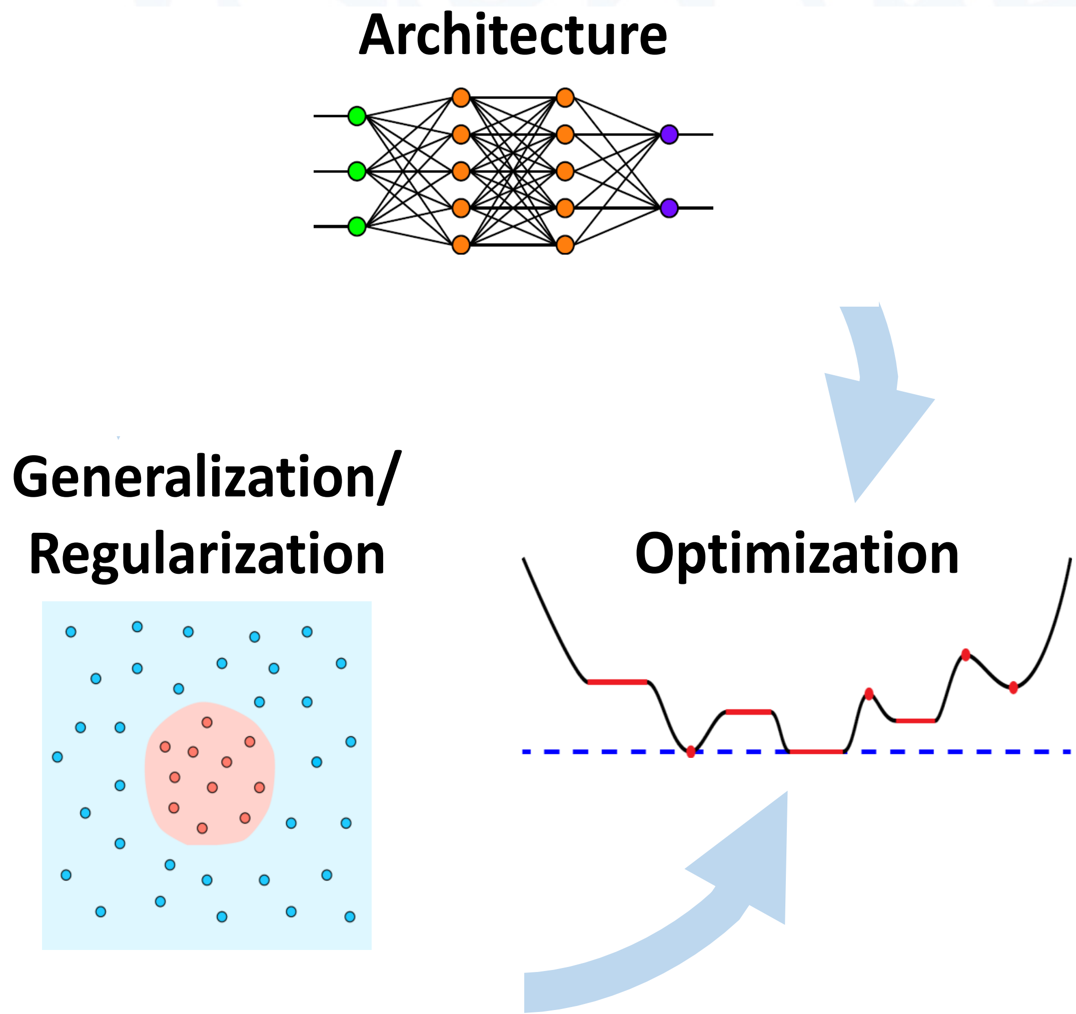


FIGURE 2. The landscape of *E*.

[1] Baldi, Hornik. Neural networks and principal component analysis: Learning from examples without local minima, Neural networks, 1989.
[2] Nouiehed, Razaviyayn. Learning deep models: Critical points and local openness. arXiv preprint arXiv:1803.02968, 2018
[3] Zhu, Soudry, Eldar, Wakin. The Global Optimization Geometry of Shallow Linear Neural Networks. JMIV, 2019.
[4] Kawaguchi. Deep learning without poor local minima. NeurIPS, 2016.

Landscape of Homogeneous Networks

- What properties of the network architecture facilitate optimization?
  - Positive homogeneity
  - Parallel subnetwork structure
- What properties of the regularization function facilitate optimization?
  - Positive homogeneity
  - Adapt network structure to the data [1]



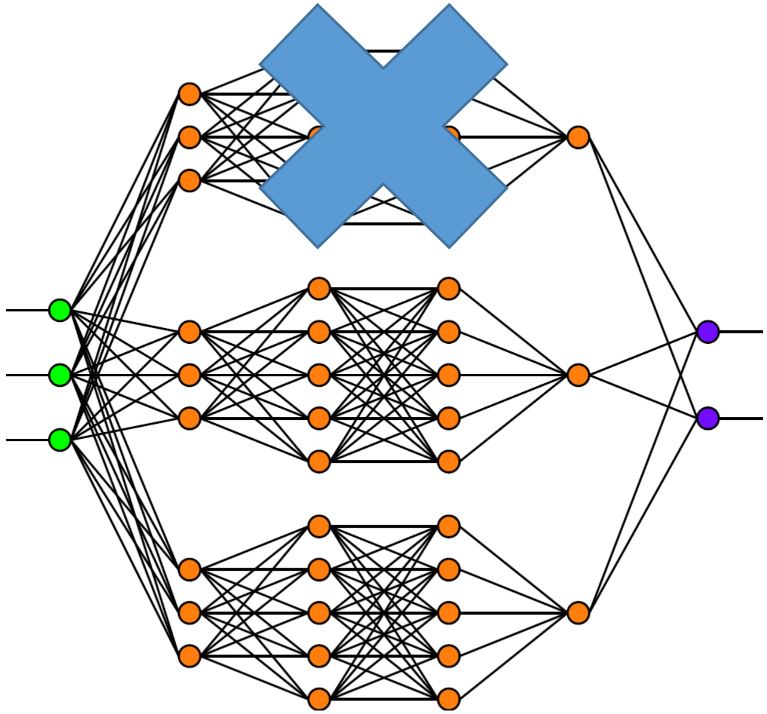
Picture courtesy of Ben Haeffele

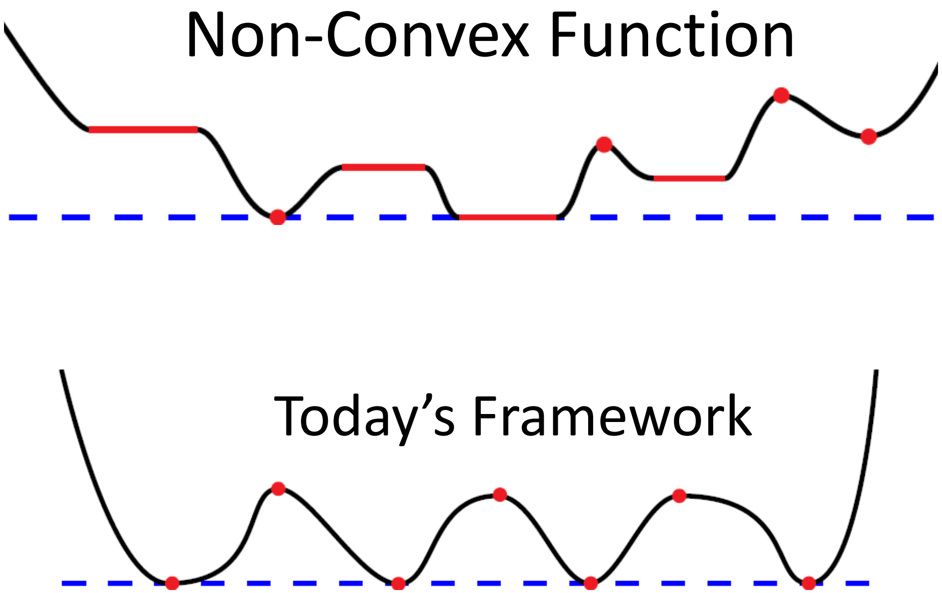
[1] Bengio, et al., "Convex neural networks." NIPS. (2005)

Landscape of Homogeneous Networks

Theorem: A local minimum such that all the weights from one subnetwork are zero is a global minimum

Theorem: If the network size is large enough, local descent can reach a global minimum from any initialization





[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.
[4] Haeffele, Vidal. Structured low-rank matrix factorization: Global optimality, algorithms, and applications. TPAMI 2019.

Analysis of Dropout/DropConnect/DropBlock

- Is dropout a valid optimization algorithm?
- **Theorem**: Dropout is SGD applied to stochastic objective.
- What type of regularization does dropout induce?
- **Theorem**: Dropout induces explicit low-rank regularization.
- What are the properties of the optimal weights?
- **Theorem**: Dropout induces balanced weights.
- Do results extend to DropBlock, DropConnect and deep networks?
- **Theorem**: DropBlock induces r-support norm regularization and balanced weights.

[1] Jacopo Cavazza, Benjamin Haeffele, Pietro Morerio, Connor Lane, Vittorio Murino, René Vidal, Dropout as a Low-Rank Regularizer for Matrix Factorization, AISTATS (2018), https://arxiv.org/abs/1710.03487
[2] Poorya Mianjy, Raman Arora, René Vidal, On the Implicit Bias of Dropout, ICML (2018), https://arxiv.org/abs/1806.09777
[3] Ambar Pal, Connor Lane, René Vidal, Benjamin D. Haeffele. On the Regularization Properties of Structured Dropout, CVPR (2020). https://arxiv.org/abs/1910.14186

JHU vision lab

Optimization Landscape of Linear Networks

René Vidal

Herschel Seder Professor of Biomedical Engineering

Director of the Mathematical Institute for Data Science

Johns Hopkins University

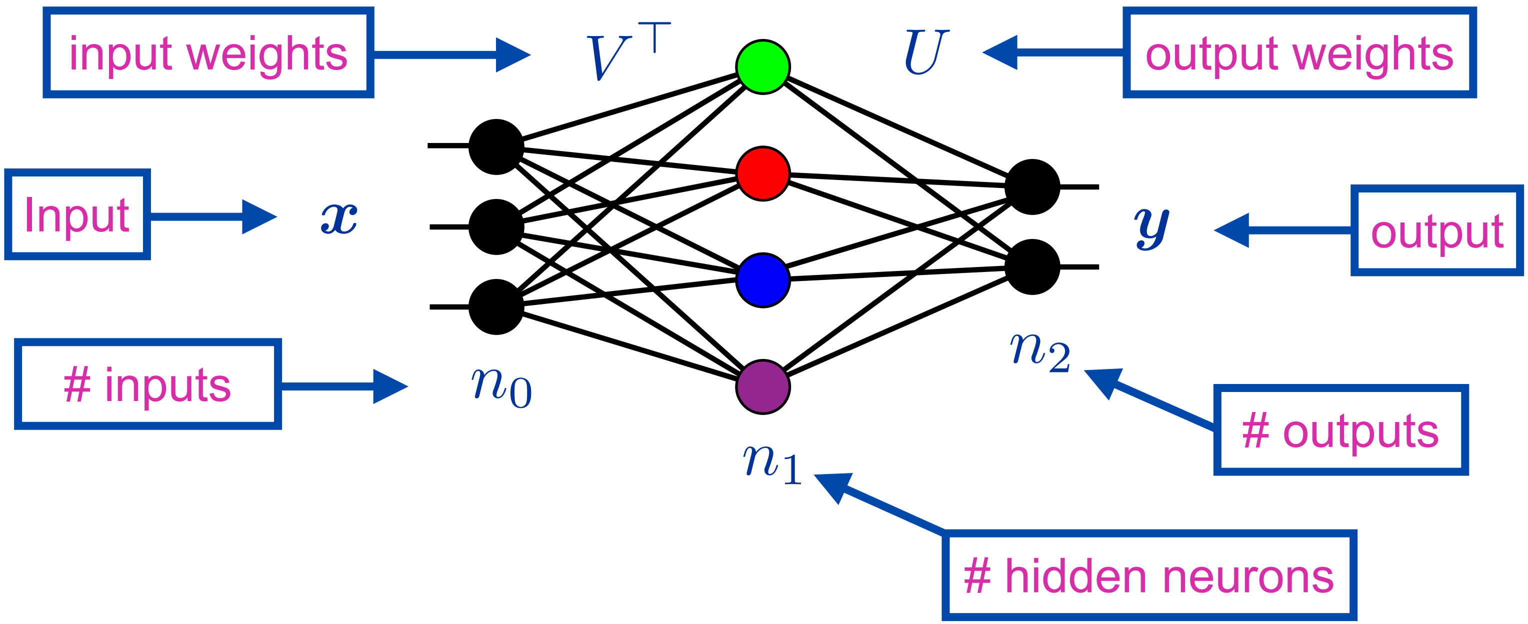
THE DEPARTMENT OF BIOMEDICAL ENGINEERING

The Whitaker Institute at Johns Hopkins

JOHNS HOPKINS MATHEMATICAL INSTITUTE for DATA SCIENCE

Single-Hidden Layer Linear Networks

- Linear Network with One Hidden Layer



- Hypothesis space:

$$\mathcal{F} = \{f \in \mathcal{Y}^{\mathcal{X}} : f(\boldsymbol{x}) = UV^\top \boldsymbol{x}, \text{ where } U \in \mathbb{R}^{n_2 \times n_1} \text{ and } V \in \mathbb{R}^{n_0 \times n_1}\}$$

Single-Hidden Layer Linear Networks

- **Risk**: $\mathcal{R}(U,V) \doteq \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\left[\|\boldsymbol{y} - UV^\top \boldsymbol{x}\|_2^2\right]$

$$= \operatorname{trace}(\Sigma_{\boldsymbol{yy}} - 2\Sigma_{\boldsymbol{yx}} V U^\top + U V^\top \Sigma_{\boldsymbol{xx}} V U^\top)$$

- **Note**: If the hidden layer is large enough ($n_1 \geq \max\{n_0, n_2\}$) so that $Z = UV^\top$ is full rank, and $\Sigma_{\boldsymbol{xx}}$ is invertible, then

$$Z^* = U^* V^{*\top} = \Sigma_{\boldsymbol{yx}} \Sigma_{\boldsymbol{xx}}^{-1}$$

- **Note**: if $\Sigma_{\boldsymbol{xx}}$ is invertible problem becomes matrix factorization

$$\min_{U,V} \|\Sigma_{\boldsymbol{yx}} \Sigma_{\boldsymbol{xx}}^{-1} - UV^\top\|_F^2$$

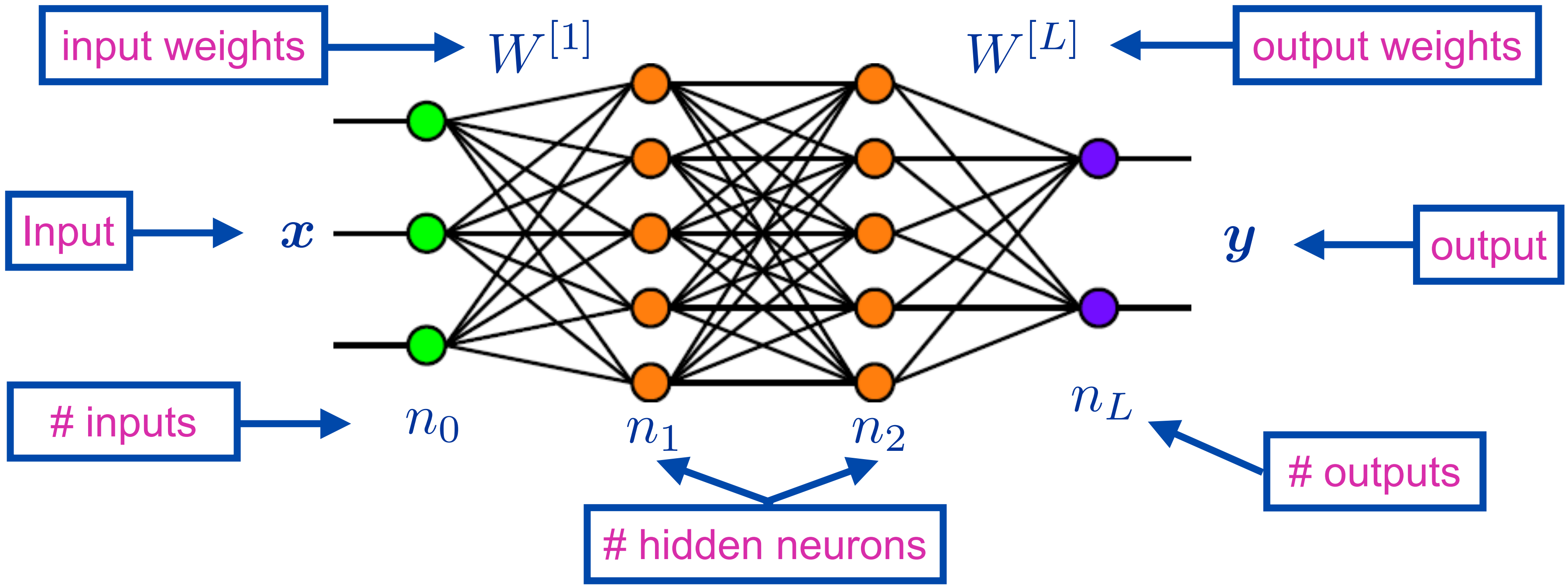
- **Theorem [1]**: If $\Sigma_{\boldsymbol{xx}}$ and $\Sigma = \Sigma_{\boldsymbol{yx}} \Sigma_{\boldsymbol{xx}}^{-1} \Sigma_{\boldsymbol{xy}}$ are invertible, then up to a change of basis, the set of global minima of the risk is:

$$U = Q_{1:n_1}, \quad V = \Sigma_{\boldsymbol{xx}}^{-1} \Sigma_{\boldsymbol{xy}} Q_{1:n_1}, \quad UV^\top = Q_{1:n_1} Q_{1:n_1}^\top \Sigma_{\boldsymbol{yx}} \Sigma_{\boldsymbol{xx}}^{-1}$$

[1] Baldi, Hornik, Neural networks and principal component analysis: Learning from examples without local minima, Neural networks, 1989.
[2] Nouiehed, Razaviyayn. Learning deep models: Critical points and local openness. arXiv preprint arXiv:1803.02968, 2018
[3] Zhu, Soudry, Eldar, Wakin. The Global Optimization Geometry of Shallow Linear Neural Networks. JMIV, 2019.

Deep Linear Networks

- Deep Linear Network with L layers



- Hypothesis space:

$$\mathcal{F} = \{ f \in \mathcal{Y}^{\mathcal{X}} : f(\boldsymbol{x}) = W^{[L]} W^{[L-1]} \cdots W^{[1]} \boldsymbol{x}, \text{ where } W^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}} \}$$

Deep Linear Networks

- **Risk**: $\mathcal{R}(W) \doteq \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\left[\|\boldsymbol{y} - W^{[L]}W^{[L-1]}\cdots W^{[1]}\boldsymbol{x}\|_2^2\right]$

$$= \mathrm{trace}(\Sigma_{\boldsymbol{yy}} - 2\Sigma_{\boldsymbol{yx}}W_{1:L}^\top + W_{1:L}\Sigma_{\boldsymbol{xx}}W_{1:L}^\top)$$

- **Note**: If hidden layers are large enough ($n_l \geq \max\{n_0, n_L\}$) so that $W_{1:L}$ is full rank, and Σ_{xx} is invertible, then

$$W_{1:L}^* = \Sigma_{\boldsymbol{yx}}\Sigma_{\boldsymbol{xx}}^{-1}$$

- **Theorem [1]**: If Σ_{xx} and Σ_{xy} are full rank with $n_L \leq n_0$ and $\Sigma = \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}$ is full rank with n_L distinct eigenvalues, then:
  - Any local minimum is global, other critical points are saddle points
  - A saddle such that $\mathrm{rank}\left(W^{[L-1]}\cdots W^{[1]}\right) = \min_{1\leq l\leq L-1} n_l$ is strict
  - Other saddles may not be strict.

[1] Kawaguchi. Deep learning without poor local minima. NeurIPS, 2016.

Landscape of Linear Networks

- All local minima are global
- Other critical points are saddles
- All saddles are strict for one hidden layer
- Non-strict saddles exist for deeper networks

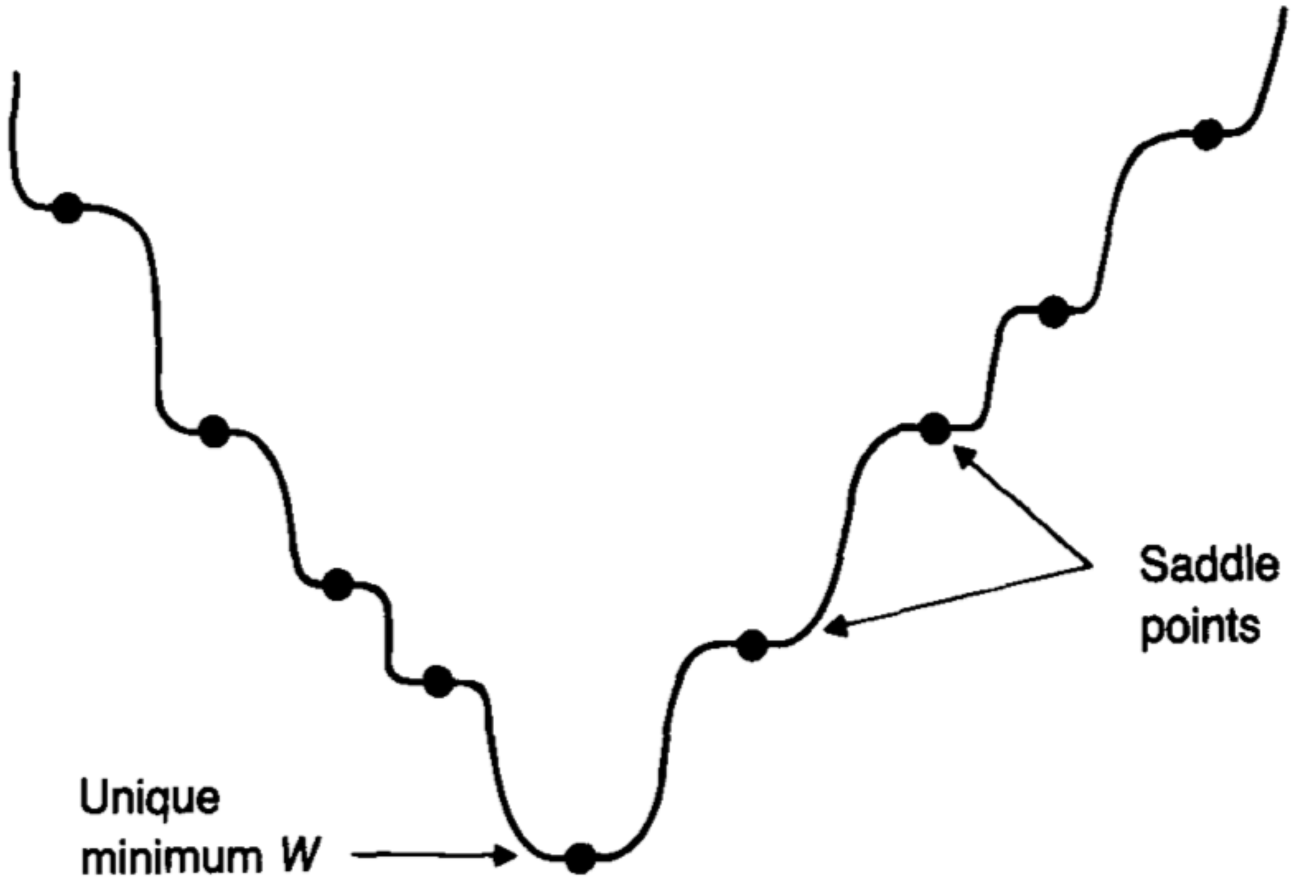


FIGURE 2. The landscape of E.

[1] Baldi, Hornik. Neural networks and principal component analysis: Learning from examples without local minima, Neural networks, 1989.
[2] Nouiehed, Razaviyayn. Learning deep models: Critical points and local openness. arXiv preprint arXiv:1803.02968, 2018
[3] Zhu, Soudry, Eldar, Wakin. The Global Optimization Geometry of Shallow Linear Neural Networks. JMIV, 2019.
[4] Kawaguchi. Deep learning without poor local minima. NeurIPS, 2016.

JHU vision lab

Global Optimality in Matrix and Tensor Factorization, Deep Learning & Beyond

Ben Haeffele and René Vidal

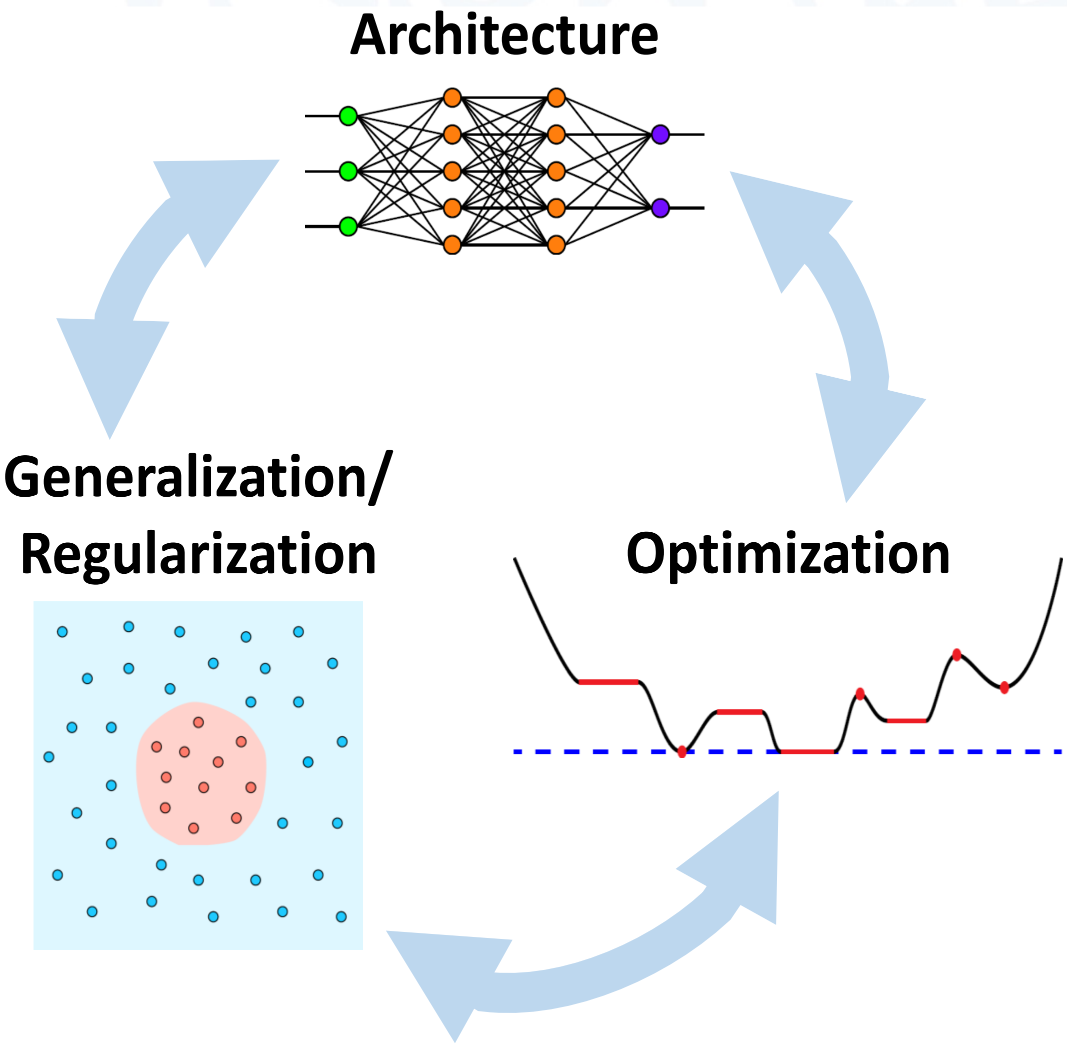
Center for Imaging Science
Mathematical Institute for Data Science
Johns Hopkins University

THE DEPARTMENT OF BIOMEDICAL ENGINEERING
The Whitaker Institute at Johns Hopkins

JOHNS HOPKINS
MATHEMATICAL INSTITUTE
for DATA SCIENCE

Toward a Unified Theory?

- Dropout regularization is equivalent to regularization with products of weights [1,2]
- Regularization with product of weights generalizes well [3,4]
- No spurious local minima for product of weight regularizers [5]



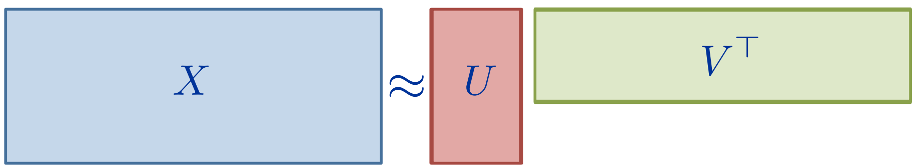
[1] Cavazza, Lane, Moreiro, Haeffele, Murino, Vidal. An Analysis of Dropout for Matrix Factorization, AISTATS 2018.
[2] Poorya Mianjy, Raman Arora, Rene Vidal. On the Implicit Bias of Dropout. ICML 2018.
[3] Neyshabur, Salakhutdinov, Srebro. Path-SGD: Path-Normalized Optimization in Deep Neural Networks. NIPS 2015
[4] Sokolic, Giryes, Sapiro, Rodrigues. Generalization error of Invariant Classifiers. AISTATS, 2017.
[5] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

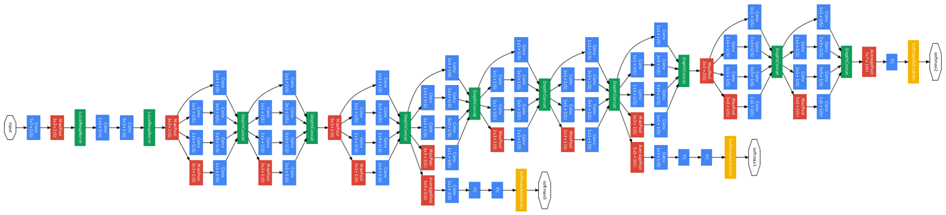
Outline

- **Architecture properties that facilitate optimization**
  – Positive homogeneity
  – Parallel subnetwork structure

- **Regularization properties that facilitate optimization**
  – Positive homogeneity
  – Adapt network structure to the data

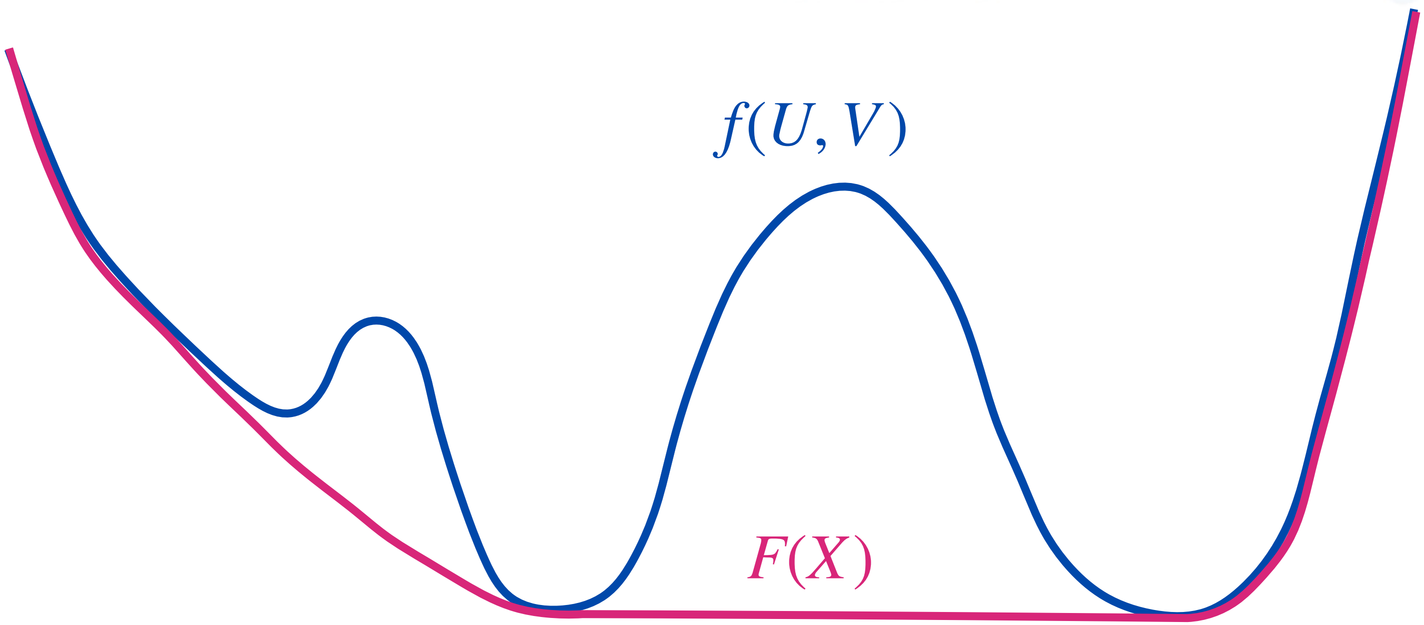
- **Theoretical guarantees**
  – Sufficient conditions for global optimality
  – Local descent can reach global minimizers





[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML ’14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, ’15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

Relating Convex & Factorized Formulations



Convex lower bound: $F(X) \leq f(U, V) \quad UV^\top = X$

Global minima agree: $\min_X F(X) = \min_{UV^\top = X} f(U, V)$

Relating Convex & Factorized Formulations

- **Convex formulations:** **Factorized formulations**

$$\min_{X} \ell(Y, X) + \lambda \|X\|_* \qquad \min_{U,V} \ell(Y, UV^\top) + \lambda \Theta(U, V)$$

- Variational form of the nuclear norm [1,2]

$$\|X\|_* = \min_{U,V,r} \sum_{i=1}^{r} \|U_i\|_2 \|V_i\|_2 \quad \text{s.t.} \quad UV^\top = X$$

$$\|X\|_* = \sum \sigma_i(X)$$

- A natural generalization is the projective tensor norm [3,4]

$$\|X\|_{u,v} = \min_{U,V,r} \sum_{i=1}^{r} \|U_i\|_u \|V_i\|_v \quad \text{s.t.} \quad UV^\top = X$$

[1] Burer, Monteiro. Local minima and convergence in low- rank semidefinite programming. Math. Prog., 2005.
[2] Cabral, De la Torre, Costeira, Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," CVPR, 2013, pp. 2488–2495.
[3] Bach, Mairal, Ponce, Convex sparse matrix factorizations, arXiv 2008.
[4] Bach. Convex relaxations of structured matrix factorizations, arXiv 2013.

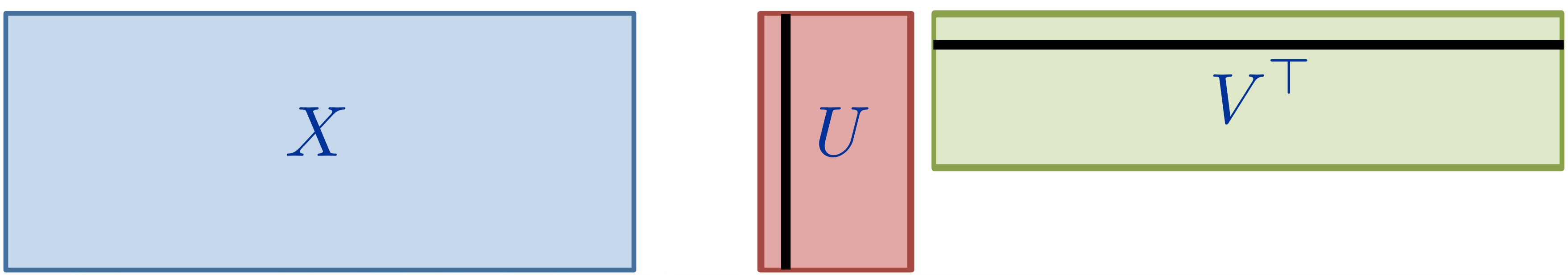
Main Results: Matrix Factorization

- **Theorem 1**: Assume ℓ is convex and once differentiable in X. A local minimizer (U, V) of the non-convex factorized problem

$$\min_{U,V,r} \ell(Y, UV^\top) + \sum_{i=1}^{r} \|U_i\|_u \|V_i\|_v$$

such that for some i $U_i = V_i = 0$, is a global minimizer. Moreover, $UV^\top$ is a global minimizer of the convex problem

$$\min_X \ell(Y, X) + \lambda \|X\|_{u,v}$$

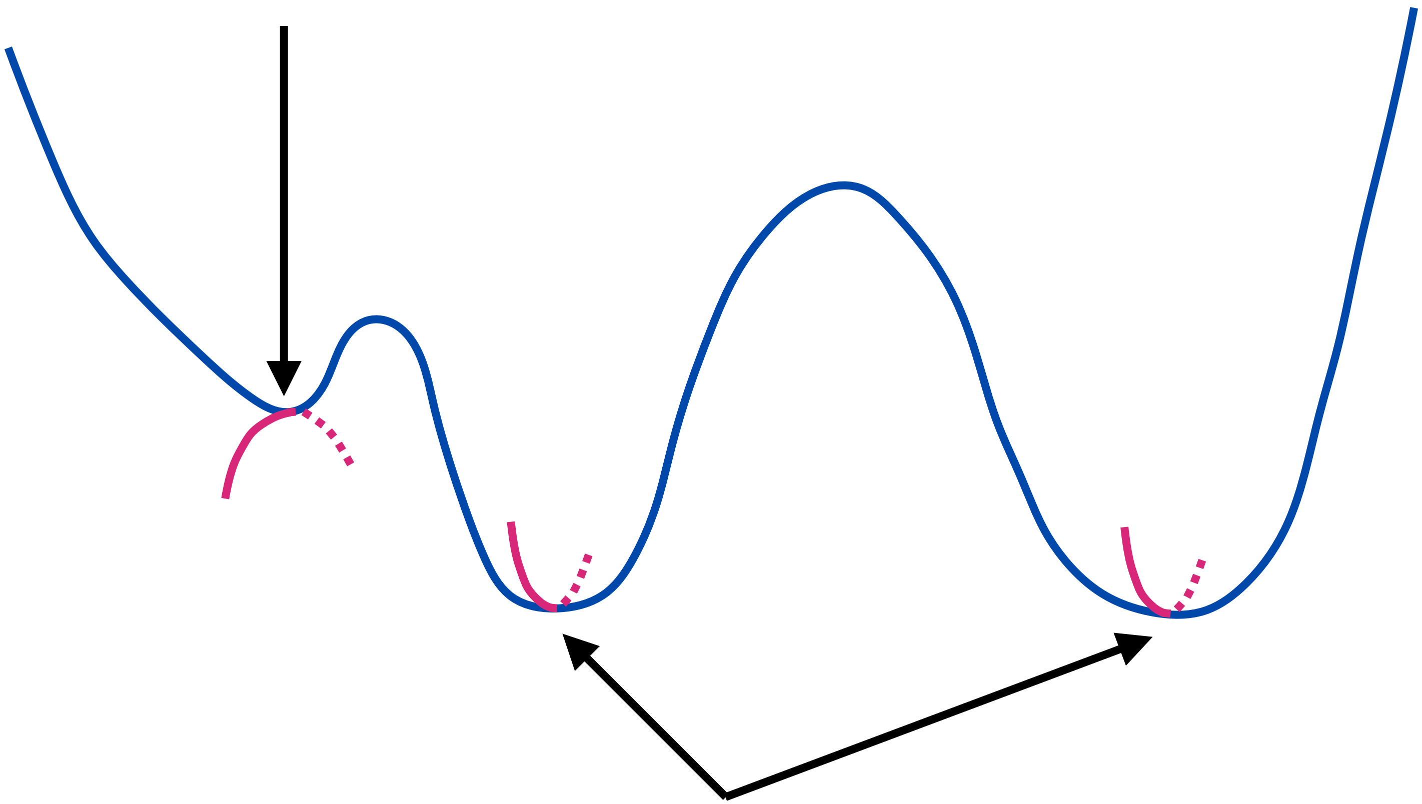


[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv '15

JOHNS HOPKINS
MATHEMATICAL INSTITUTE
for DATA SCIENCE

Main Results: Matrix Factorization

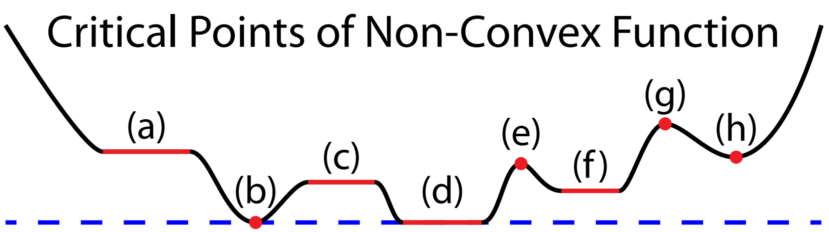
If at a spurious local minima, we can find a descent direction by adding extra dimensions, thus creating a saddle point

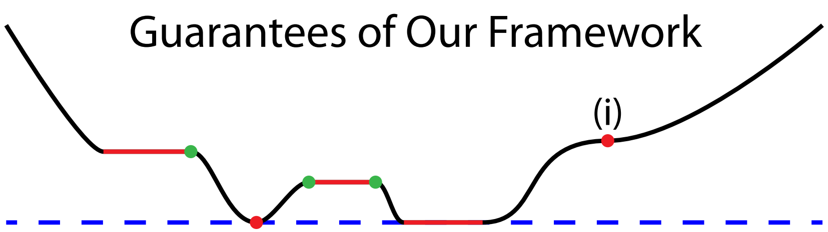


If at a global minima, we cannot find a descent direction

Main Results: Matrix Factorization

- **Theorem 2:** If the number of columns is large enough, local descent can reach a global minimizer from any initialization



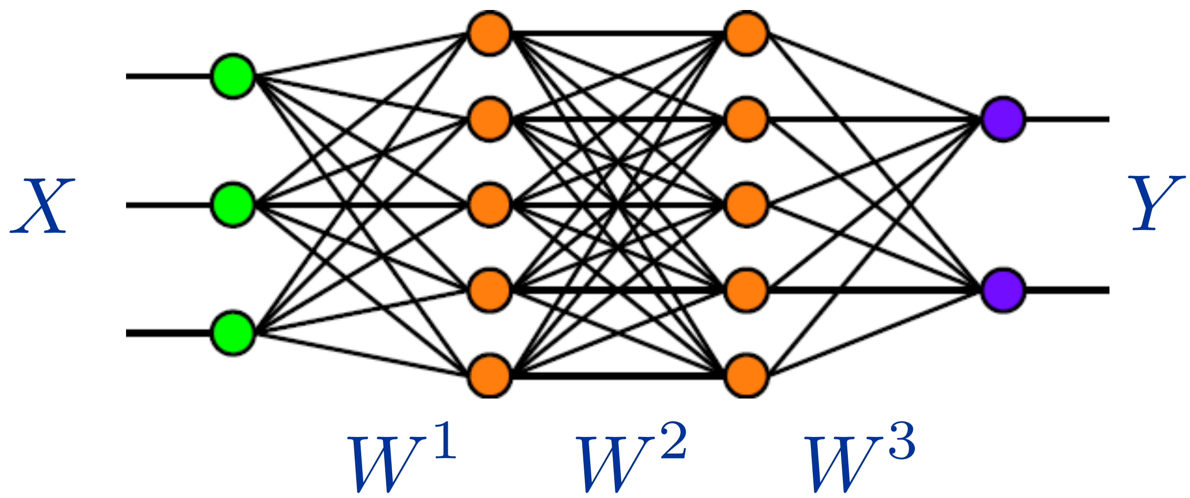


- **Meta-Algorithm:**
  – If not at a local minima, perform local descent
  – At local minima, test if Theorem 1 is satisfied. If yes => global minima
  – If not, increase size of factorization and find descent direction (u,v)

$$r \leftarrow r + 1 \quad U \leftarrow \begin{bmatrix} U & u \end{bmatrix} \quad V \leftarrow \begin{bmatrix} V & v \end{bmatrix}$$

[1] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv '15

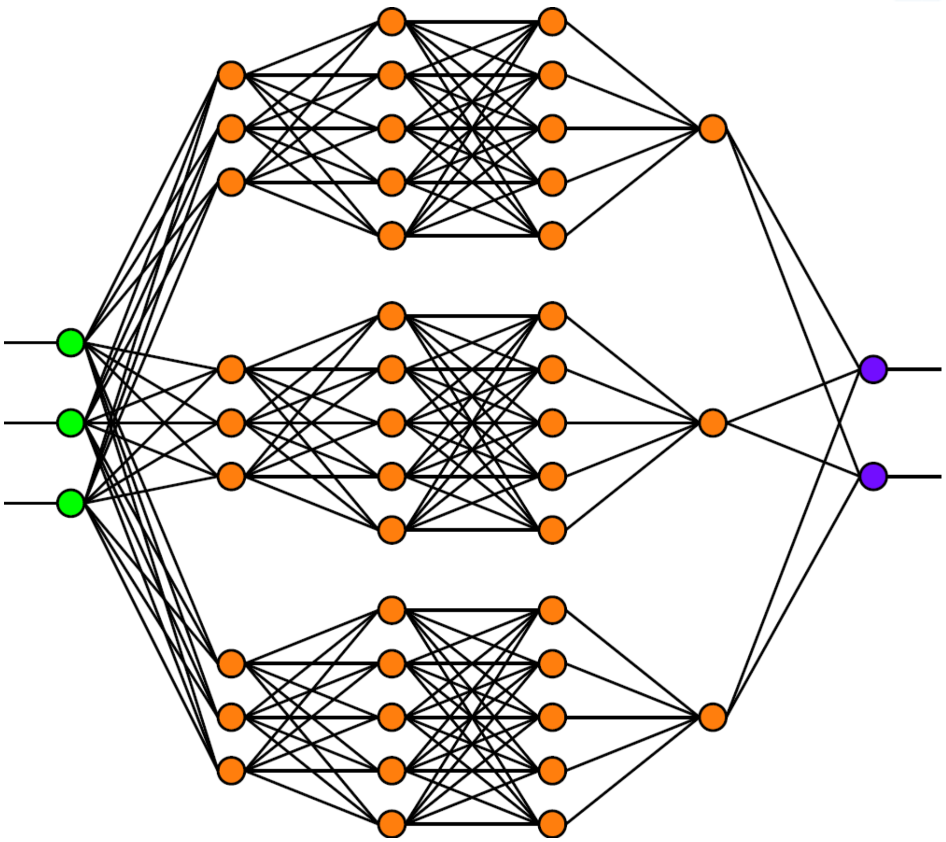
From Matrix Factorization to Deep Learning



$$\Phi(X, W^1, \ldots, W^K) = \psi_K(\cdots \psi_2(\psi_1(XW^1)W^2)\cdots W^K)$$

output activation input weights

From Matrix Factorization to Deep Learning



- In matrix factorization we had

$$\Phi(U,V) = \sum_{i=1}^{r} U_i V_i^\top$$

- In positively homogeneous networks with parallel structure we have

$$\Phi(W^1,\ldots,W^K) = \sum_{i=1}^{r} \phi(W_i^1,\ldots,W_i^K)$$

From Matrix Factorization to Deep Learning

- In matrix factorization we had "generalized nuclear norm"

$$\|Z\|_{u,v} = \min_{U,V,r} \sum_{i=1}^{r} \|U_i\|_u \|V_i\|_v \quad \text{s.t.} \quad UV^\top = Z$$

- By analogy we define "nuclear deep net regularizer"

$$\Omega_{\phi,\theta}(Z) = \min_{\{W^k\},r} \sum_{i=1}^{r} \theta(W_i^1, \ldots, W_i^K) \text{ s.t. } \Phi(W^1, \ldots, W^K) = Z$$

where θ is positively homogeneous of the same degree as ϕ

- **Proposition:** $\Omega_{\phi,\theta}$ is convex
- **Intuition:** regularizer Θ "comes from a convex function"

Main Results: Deep Learning Case

- **Theorem 1:** Assume $\ell(Y,Z)$ convex and differentiable in Z. A local minimizer $(W^1,\dots,W^K)$ of the factorized formulation

$$\min_{\{W^k\}} \ell(Y,\Phi(W^1,\dots,W^K)) + \lambda\Theta(W^1,\dots,W^K)$$

such that for some i and all k $W_i^k = 0$ is a global minimizer. Moreover, $Z = \Phi(W^1,\dots,W^K)$ is a global minimizer of the convex problem

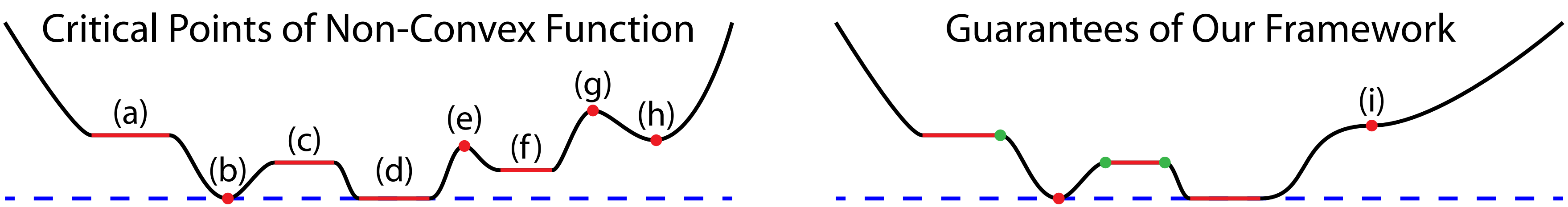
$$\min_Z \ell(Y,Z) + \lambda\Omega_{\phi,\theta}(Z)$$

- Examples
  - Matrix factorization
  - Tensor factorization
  - Deep learning

[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

Main Results: Deep Learning Case

- **Theorem 2:** If the size of the network is large enough, local descent can reach a global minimizer from any initialization



- **Meta-Algorithm:**
  - If not at a local minima, perform local descent
  - At a local minima, test if Theorem 1 is satisfied. If yes => global minima
  - If not, increase size by 1 (add network in parallel) and continue
  - Maximum r guaranteed to be bounded by the dimensions of the network output

[1] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv '15

Summary so Far

- **Size matters**
  - Optimize not only the network weights, but also the network size
  - Today: size = number of neurons or number of parallel networks
  - Tomorrow: size = number of layers + number of neurons per layer
- **Regularization matters**
  - Use “positively homogeneous regularizer” of same degree as network
  - How to build a regularizer that controls number of layers + number of neurons per layer
- **Not done yet**
  - Checking if we are at a local minimum or finding a descent direction can be NP hard
  - Need “computationally tractable” regularizers

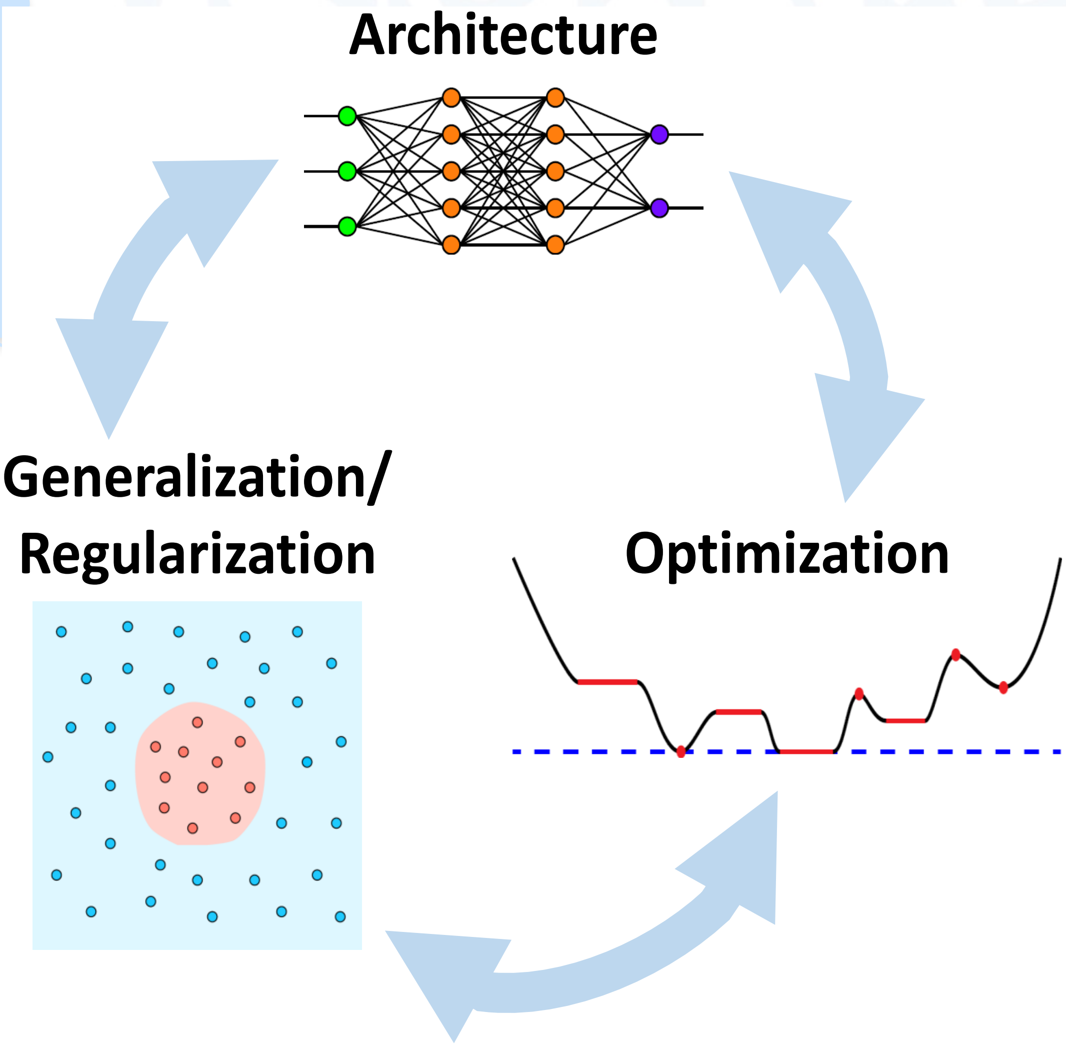
On the Regularization Properties of Structured Dropout

René Vidal
Herschel Seder Professor of Biomedical Engineering
Director of the Mathematical Institute for Data Science
Johns Hopkins University

JHU vision lab
THE DEPARTMENT OF BIOMEDICAL ENGINEERING
The Whitaker Institute at Johns Hopkins
JOHNS HOPKINS MATHEMATICAL INSTITUTE for DATA SCIENCE

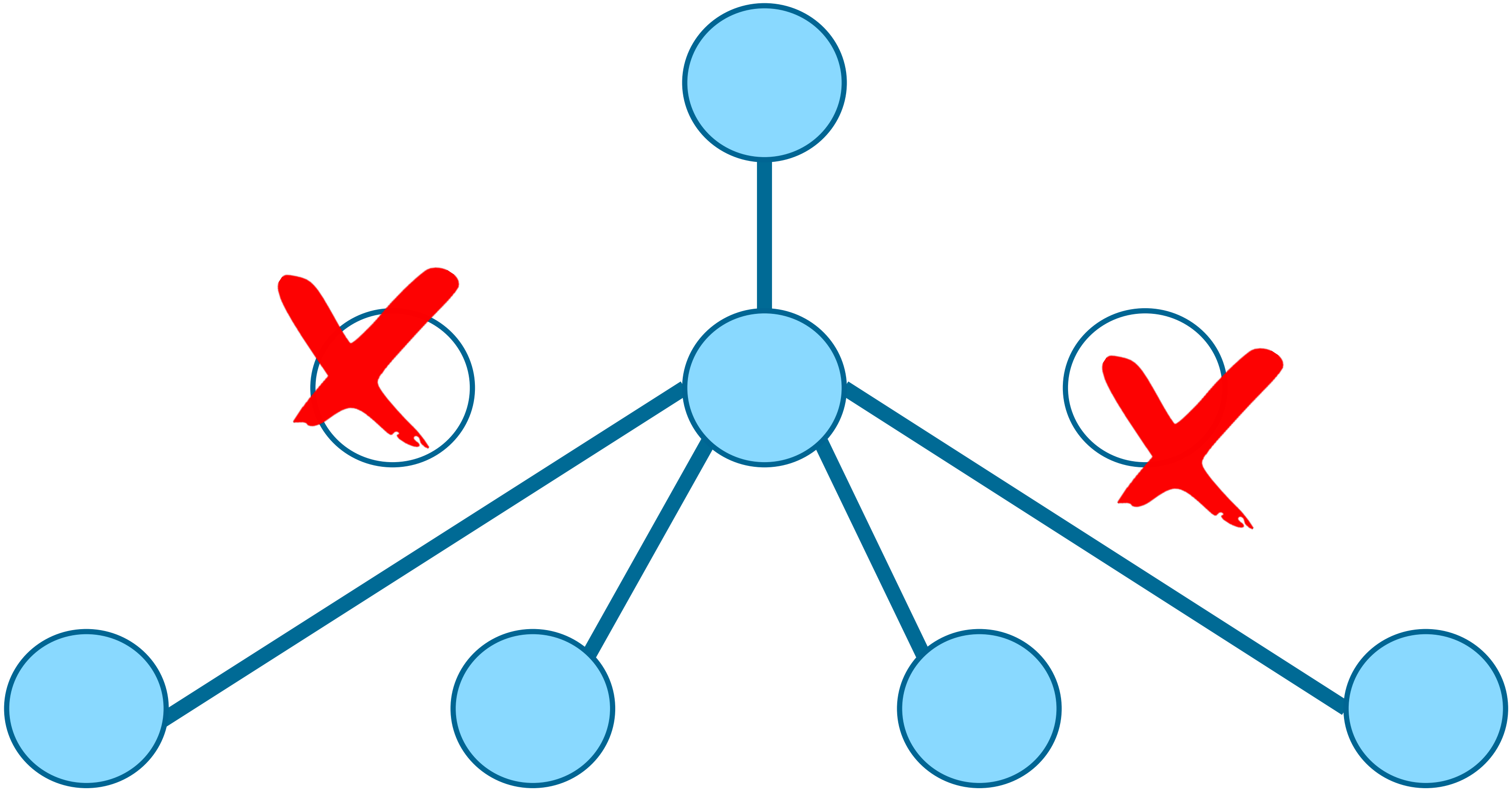
Toward a Unified Theory?

- Dropout regularization is equivalent to regularization with products of weights [1,2]
- Regularization with product of weights generalizes well [3,4]
- No spurious local minima for product of weight regularizers [5]

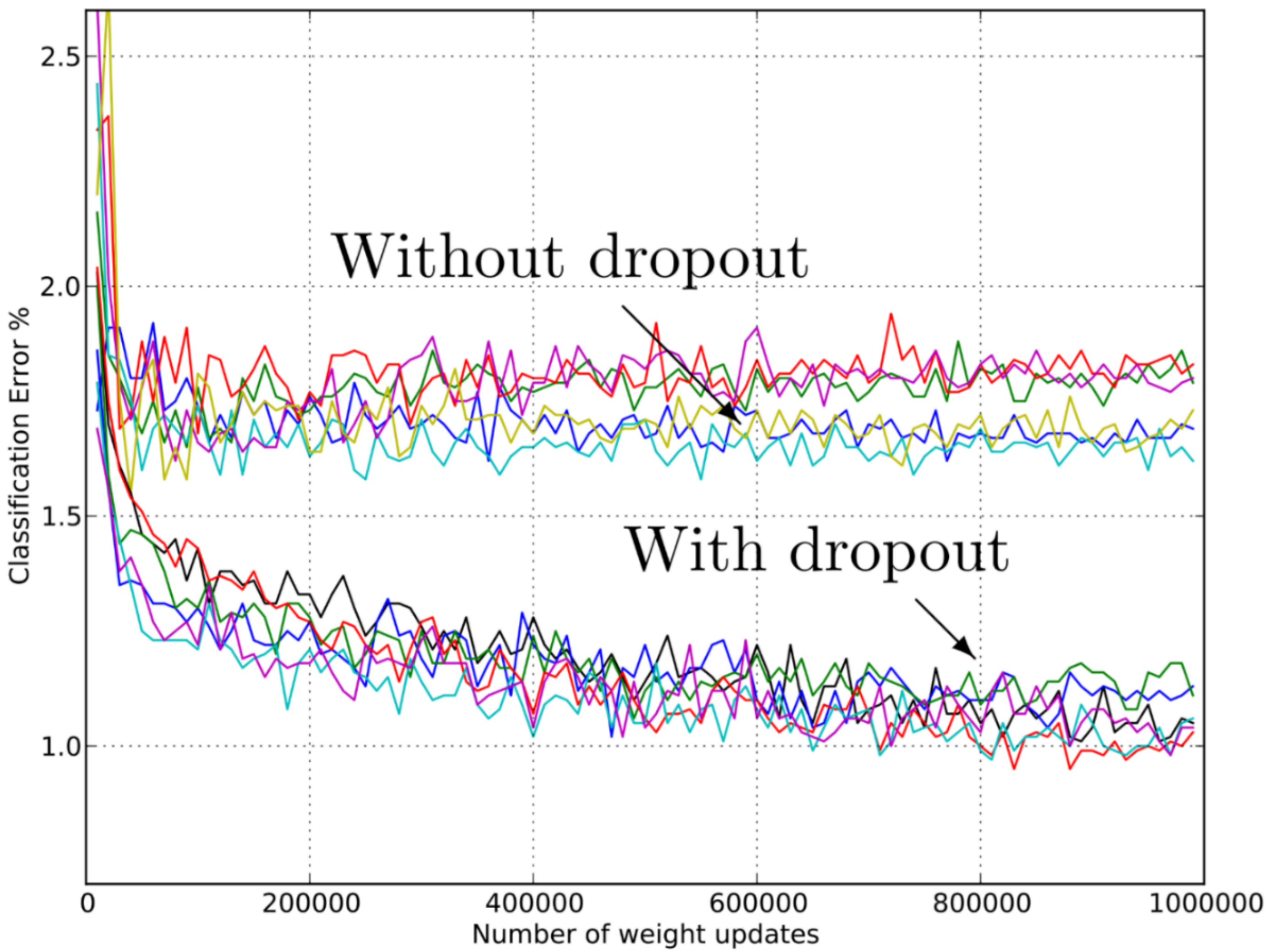


[1] Cavazza, Lane, Moreiro, Haeffele, Murino, Vidal. An Analysis of Dropout for Matrix Factorization, AISTATS 2018.
[2] Poorya Mianjy, Raman Arora, Rene Vidal. On the Implicit Bias of Dropout. ICML 2018.
[3] Neyshabur, Salakhutdinov, Srebro. Path-SGD: Path-Normalized Optimization in Deep Neural Networks. NIPS 2015
[4] Sokolic, Giryes, Sapiro, Rodrigues. Generalization error of Invariant Classifiers. AISTATS, 2017.
[5] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

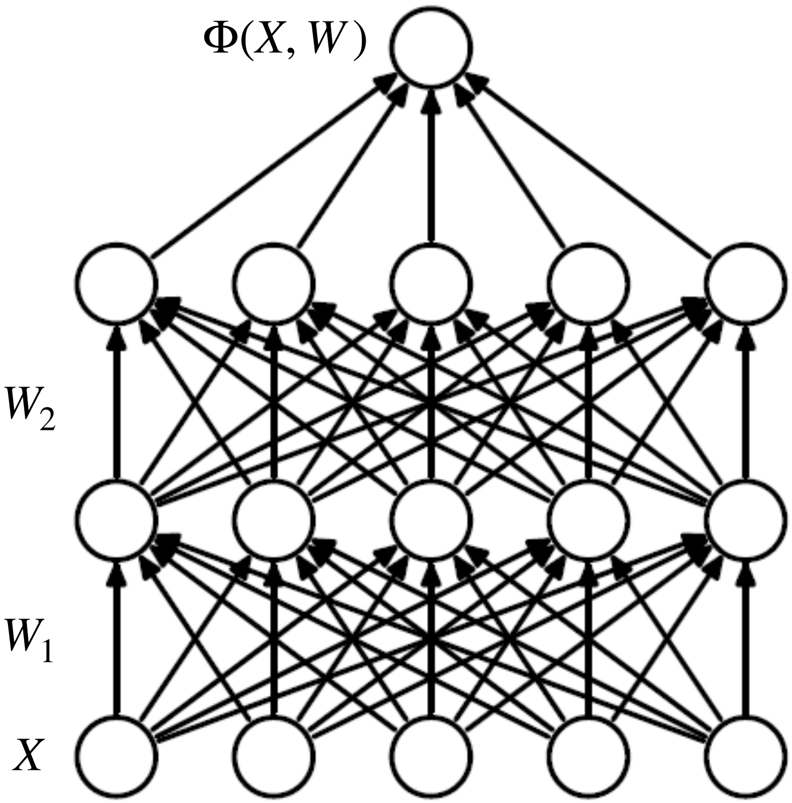
Dropout Training



Dropout Training: Better Learning Curve



Backpropagation vs Dropout Training



(a) Standard Neural Net

- Minimize empirical loss

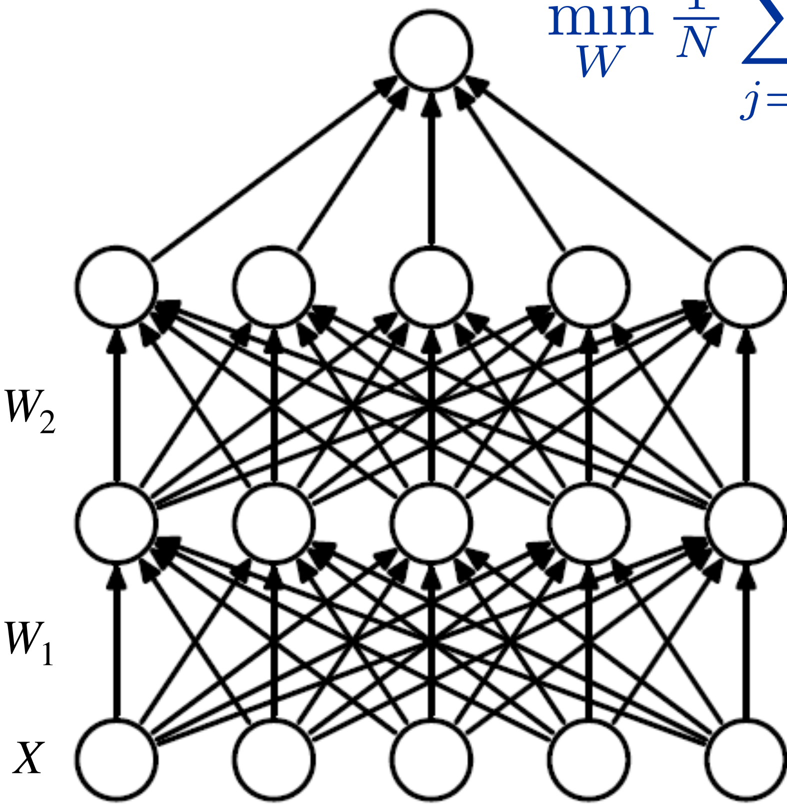
$$\min_W \frac{1}{N} \sum_{j=1}^{N} \ell(Y_j, \Phi(X_j, W))$$

- Stochastic gradient descent

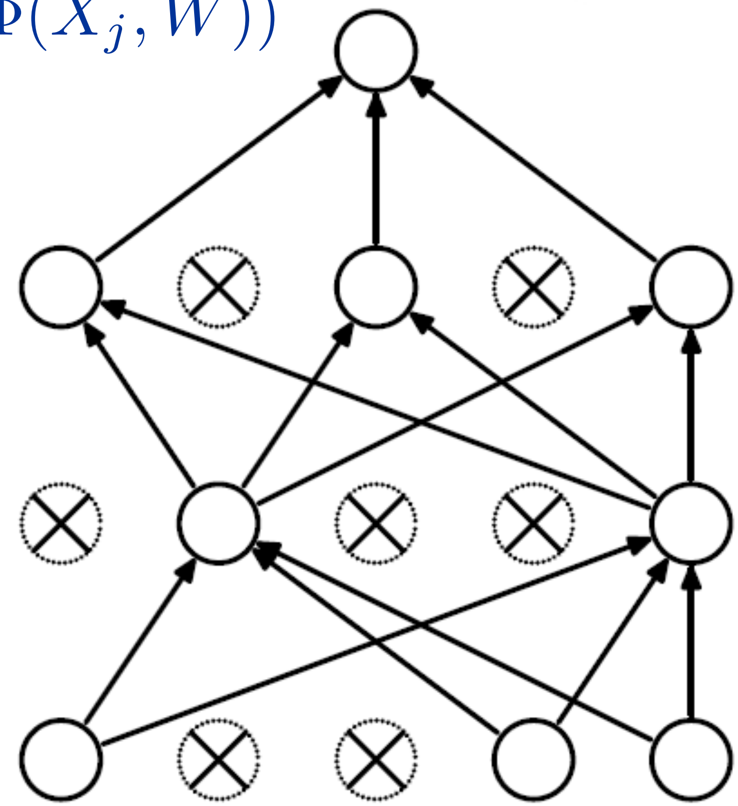
$$W^{t+1} = W^t - \frac{\epsilon}{|\mathcal{B}_t|} \sum_{j \in \mathcal{B}_t} \nabla \ell(Y_j, \Phi(X_j, W^t))$$

Backpropagation vs Dropout Training

$$\min_{W} \frac{1}{N}\sum_{j=1}^{N} \ell(Y_j, \Phi(X_j, W))$$



(a) Standard Neural Net



(b) After applying dropout.

$$W^{t+1} = W^t - \frac{\epsilon}{|\mathcal{B}_t|}\sum_{j\in\mathcal{B}_t} \nabla\ell(Y_j, \Phi(X_j, W^t))$$

$$W^{t+1} = W^t - \frac{\epsilon}{|\mathcal{B}_t|}\sum_{j\in\mathcal{B}_t} \nabla\ell(Y_j, \underbrace{\Phi(X_j, W^t, z^t)}_{\text{set output of drop out neurons to 0}}) \otimes \underbrace{z^t}_{\text{set gradient of dropout neurons to 0}}$$

Srivastava et al. – Dropout: A simple way to prevent neural networks from overfitting – JMLR 2014

JOHNS HOPKINS
MATHEMATICAL INSTITUTE
for DATA SCIENCE

Dropout Induces Low-Rank Solutions

$$\text{Dropout} \approx (\text{Nuclear Norm})^2$$

[1] Jacopo Cavazza, Benjamin Haeffele, Pietro Morerio, Connor Lane, Vittorio Murino, René Vidal, Dropout as a Low-Rank Regularizer for Matrix Factorization, AISTATS (2018), https://arxiv.org/abs/1710.03487

Deterministic vs Stochastic Factorization

- What objective function is being minimized by dropout?
- Deterministic Matrix Factorization (DMF)

$$\min_{U,V} \|Y - UV^\top\|_F^2$$

#outputs x #neurons

#neurons x #inputs

- Stochastic Matrix Factorization (SMF)

$$\min_{U,V} \mathbb{E}_{\boldsymbol{z}} \|Y - \frac{1}{\theta} \underbrace{U \operatorname{diag}(\boldsymbol{z}) V^\top}_{\sum_{i=1}^{r} z_i U_i V_i^\top} \|_F^2, \quad z_i \sim \operatorname{Ber}(\theta), \ \theta \in (0,1)$$

#neurons

[1] Jacopo Cavazza, Benjamin Haeffele, Pietro Morerio, Connor Lane, Vittorio Murino, René Vidal, Dropout as a Low-Rank Regularizer for Matrix Factorization, AISTATS (2018), https://arxiv.org/abs/1710.03487

Dropout as an Explicit Regularizer for SMF

- Using the definition of variance $\mathbb{E}(y^2) = \mathbb{E}(y)^2 + \mathrm{Var}(y)$ we can show that dropout induces an explicit regularizer

$$\mathbb{E}_{\boldsymbol{z}} \left\| Y - \tfrac{1}{\theta} U \mathrm{diag}(\boldsymbol{z}) V^\top \right\|_F^2 = \|Y - UV^\top\|_F^2 + \frac{1-\theta}{\theta} \sum_{i=1}^{r} \|U_i\|_2^2 \|V_i\|_2^2$$

- The second term looks like the nuclear norm (low-rank reg.)

$$\|X\|_* = \min_{U,V,r} \sum_{i=1}^{r} \|U_i\|_2 \|V_i\|_2 \ \text{ s.t. } \ UV^\top = X$$

[1] Jacopo Cavazza, Benjamin Haeffele, Pietro Morerio, Connor Lane, Vittorio Murino, René Vidal, Dropout as a Low-Rank Regularizer for Matrix Factorization, AISTATS (2018), https://arxiv.org/abs/1710.03487

Dropout with Variable Rate => Low Rank

- **Proposition**: Dropout with variable rate induces a regularizer

$$\Omega(X) = \min_{U,V,r} \frac{1-\theta_r}{\theta_r} \sum_{i=1}^{r} \|U_i\|_2^2 \|V_i\|_2^2 \quad \text{s.t.} \quad UV^\top = X$$

whose convex envelope is the (nuclear norm)²

$$\frac{1-\theta_1}{\theta_1} \|X\|_*^2$$

- **Theorem**: Let (U*,V*,r*) be a global minimum of

$$\min_{U,V,r} \|Y - UV^\top\|_F^2 + \frac{1-\theta_r}{\theta_r} \sum_{i=1}^{r} \|U_i\|_2^2 \|V_i\|_2^2$$

Then, $U^* {V^*}^\top = \mathcal{S}_\tau(Y)$ is a global minimum of

$$\min_X \|Y - X\|_F^2 + \frac{1-\theta_1}{\theta_1} \|X\|_*^2$$

singular value thresholding

tau depends on svalues of Y

What About Dropout with Fixed Rate?

- Results so far tell us what the optimal product is for variable *r*, but do not tell us what the optimal factors look like for fixed *r*.

- The weights *(U,V)* are balanced if the product of the norms of incoming and outgoing weights are equal for all neurons

$$\|U_i\|_2\|V_i\|_2 = \|U_j\|_2\|V_j\|_2 \quad \forall i,j = 1,\ldots,r$$

- **Theorem** [balance via rotation] For any pair *(U,V)* there exists a rotation R such that the rotated pair *(U',V')*= *(UR,VR)* gives the same product, i.e., $UV^T = U'V'^T$, and *(U',V')* are balanced.

- **Algorithm to compute *(U',V',R)***: based on Gram matrices, eigenvalue decompositions and matrix diagonalization.

[2] Poorya Mianjy, Raman Arora, René Vidal, On the Implicit Bias of Dropout, ICML (2018), https://arxiv.org/abs/1806.09777

Dropout Minima are Low Rank & Balanced

$$\min_{U,V} \|Y - UV^\top\|_F^2 + \lambda \sum_{i=1}^{r} \|U_i\|_2^2 \|V_i\|_2^2$$

- **Theorem**: (U*,V*) is a global minimum iff it is balanced and

$$U^* V^{*\top} = \mathcal{S}_\tau(Y)$$

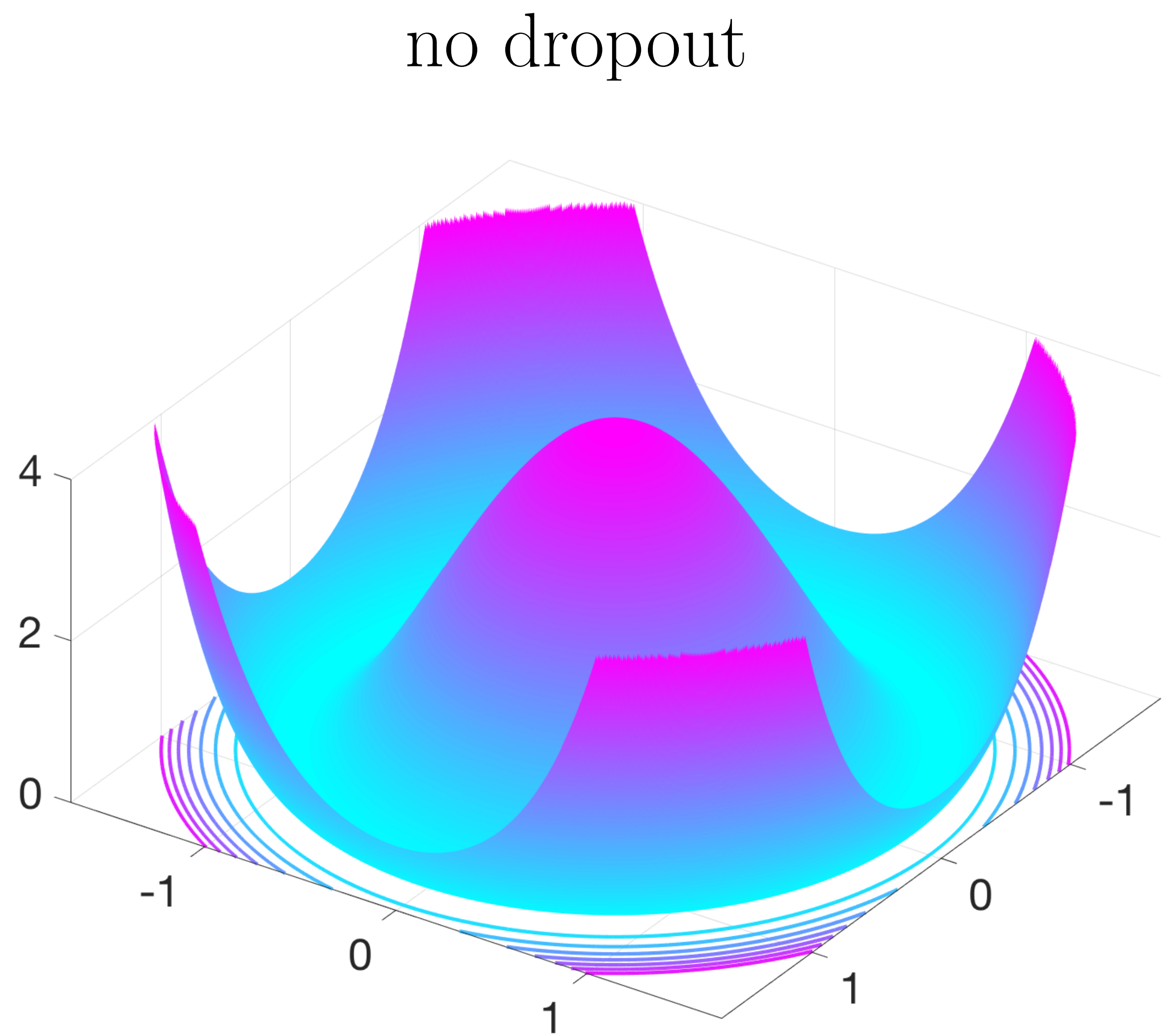
  where tau and optimal r depend on singular values of Y

- **Algorithm:** A global optimum (U*,V*) can be found as follows
  - Find any factorization (U,V) of $\mathcal{S}_\tau(Y)$
  - Balance the factors to obtain (U*, V*) = (UR, VR)

[2] Poorya Mianjy, Raman Arora, René Vidal, On the Implicit Bias of Dropout, ICML (2018), https://arxiv.org/abs/1806.09777

Effect of Dropout Rate on the Landscape

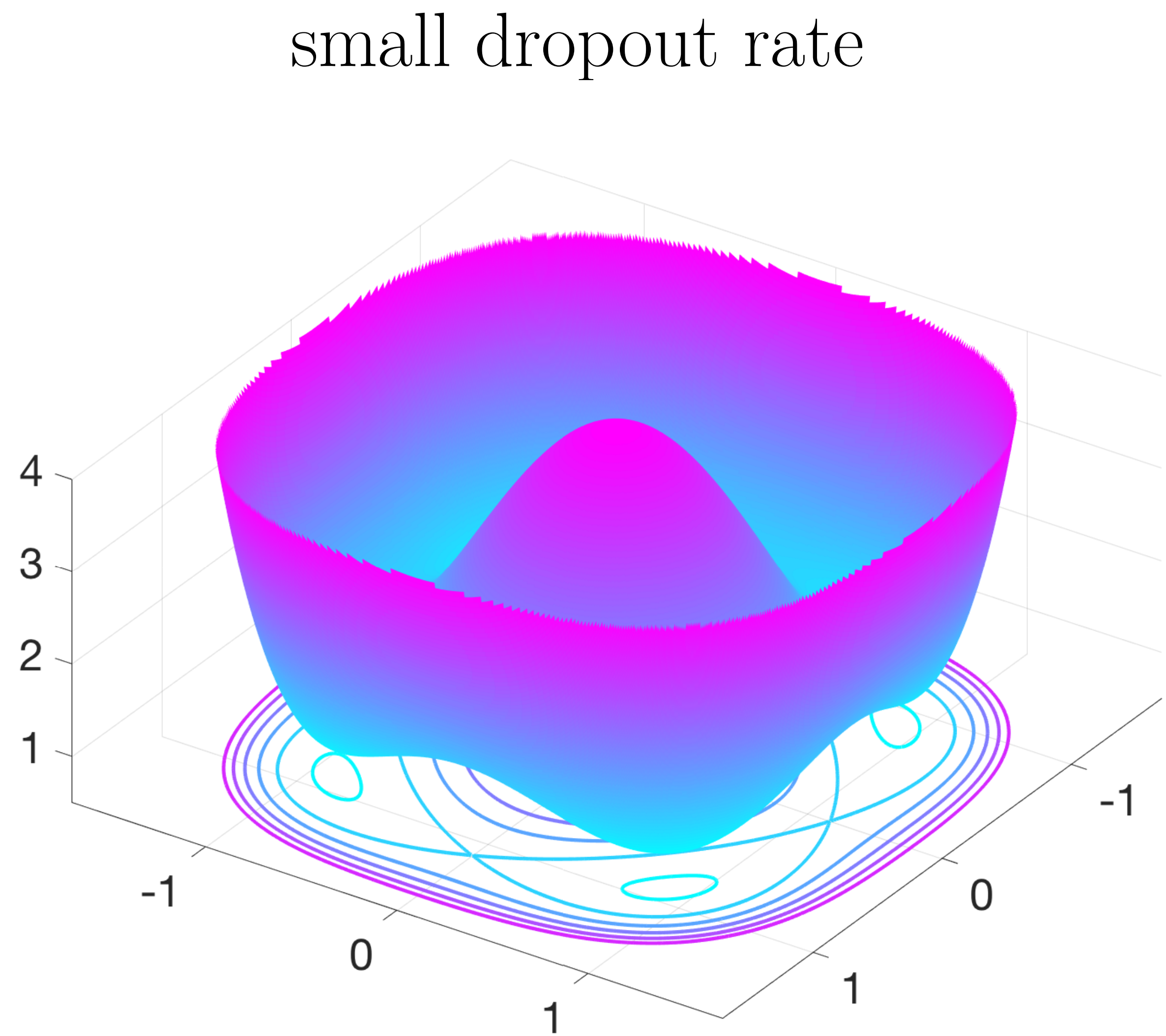
- Linear auto-encoder
- 1 input
- 2 hidden neurons
- 1 output



[2] Poorya Mianjy, Raman Arora, René Vidal, On the Implicit Bias of Dropout, ICML (2018), https://arxiv.org/abs/1806.09777

Effect of Dropout Rate on the Landscape

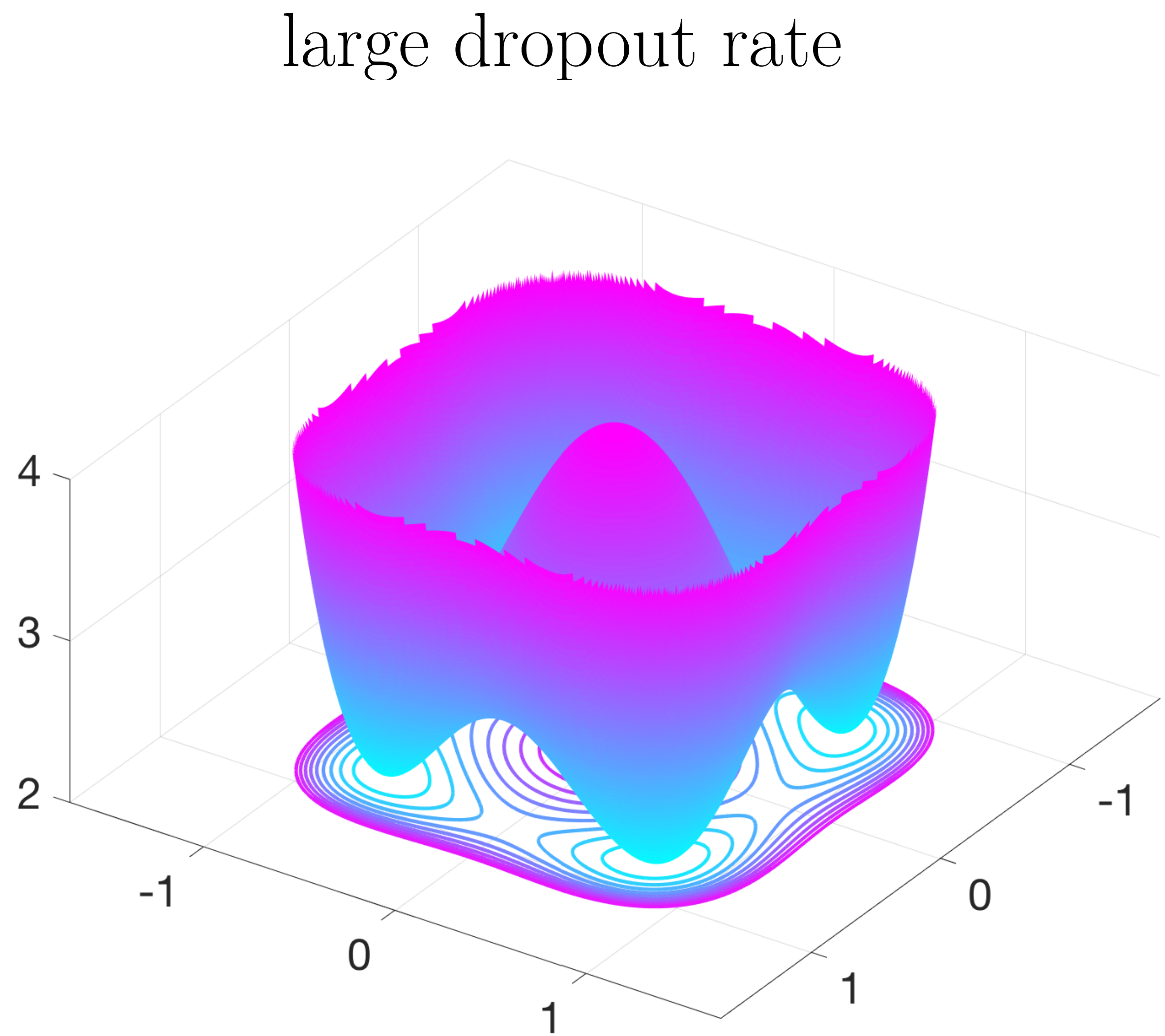
- Linear auto-encoder
- 1 input
- 2 hidden neurons
- 1 output



[2] Poorya Mianjy, Raman Arora, René Vidal, On the Implicit Bias of Dropout, ICML (2018), https://arxiv.org/abs/1806.09777

Effect of Dropout Rate on the Landscape

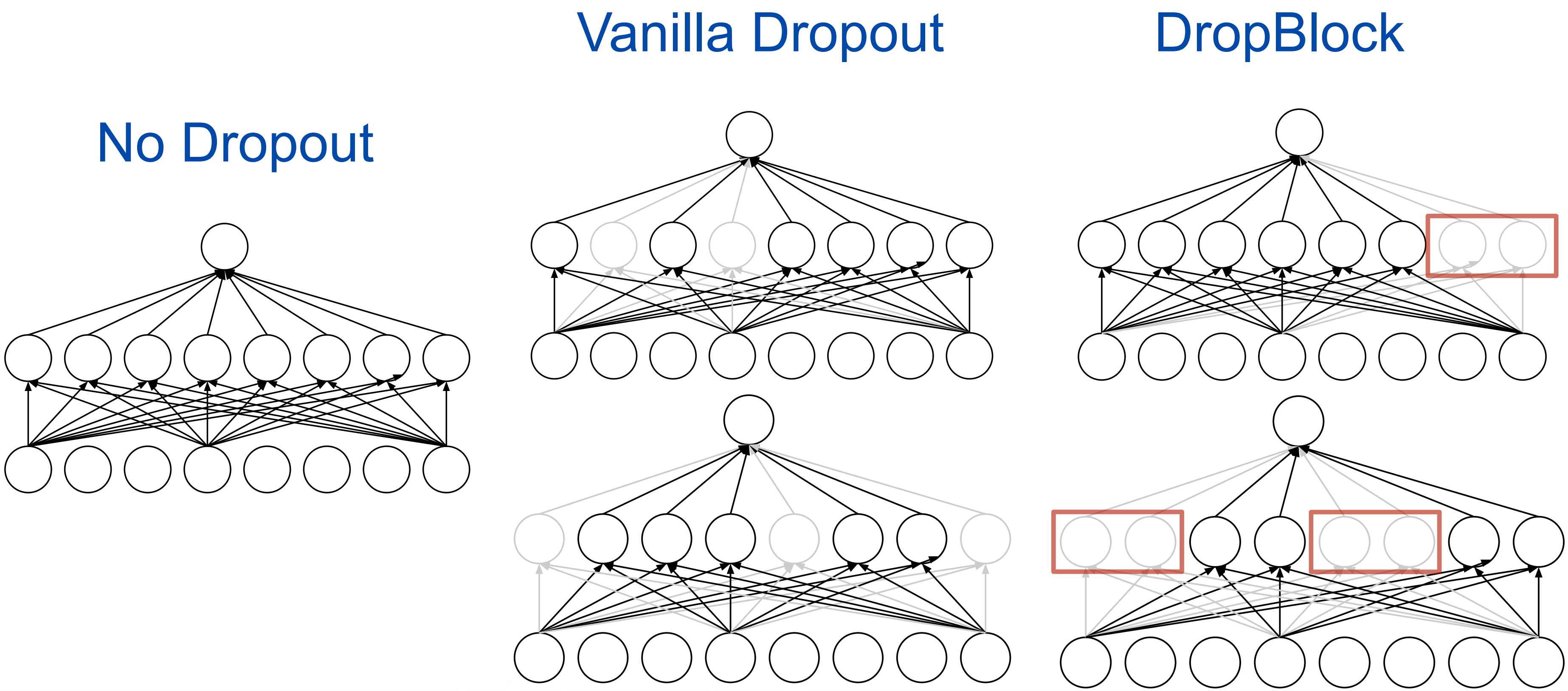
- Linear auto-encoder
- 1 input
- 2 hidden neurons
- 1 output



[2] Poorya Mianjy, Raman Arora, René Vidal, On the Implicit Bias of Dropout, ICML (2018), https://arxiv.org/abs/1806.09777

DropBlock

- Motivation: Prevent *co-adaptation* of correlated units
- Instead of dropping units independently, blocks of a fixed size are dropped together



Dropout as an Explicit Regularizer for SMF

- **Recall**: Dropout is an SGD method for minimizing

#neurons

weights
i-th neuron

$$\mathbb{E}_{\boldsymbol{z}} \|Y - \tfrac{1}{\theta} U \operatorname{diag}(\boldsymbol{z}) V^\top\|_F^2 = \|Y - UV^\top\|_F^2 + \frac{1-\theta}{\theta} \sum_{i=1}^{r} \|U_i\|_2^2 \|V_i\|_2^2$$

- **Theorem**: DropBlock is an SGD method for minimizing

#blocks

weights
i-th block

$$\mathbb{E}_{\boldsymbol{w}} \|Y - \tfrac{1}{\theta} U (\operatorname{diag}(\boldsymbol{w}) \otimes I_r) V^\top X\|_F^2 = \|Y - UV^\top\|_F^2 + \frac{1-\theta}{\theta} \sum_{i=1}^{r} \|U_i\|_F^2 \|V_i\|_F^2$$

DropBlock Induces r-support regularization

- **Proposition**: DropBlock induces spectral r-support norm

$$\Omega(X) = \min_{U,V,r} \frac{1-\theta_r}{\theta_r} \sum_{i=1}^{r} \|U_i\|_F^2 \|V_i\|_F^2 : UV^\top = X$$

$$= \max_{\rho \in \{1,2,\ldots,r\}} \left(\sum_{i=1}^{\rho-1} \sigma_i^2 + \frac{\left(\sum_{i=\rho}^{r} \sigma_i\right)^2}{r-\rho+1} \right)$$

- Tradeoff between ℓ_2^2 and ℓ_1^2 penalties
- If $\rho^* = 1$ then $\Omega(X)$ is the scaled Nuclear norm $\|X\|_*^2$
- As $\rho^* \to r$, $\Omega(X)$ moves towards the Frobenius norm $\|X\|_F^2$

DropBlock Induces Balance & Low-Support

- **Theorem**: A global minimum (U^*, V^*, r^*) of DropBlock

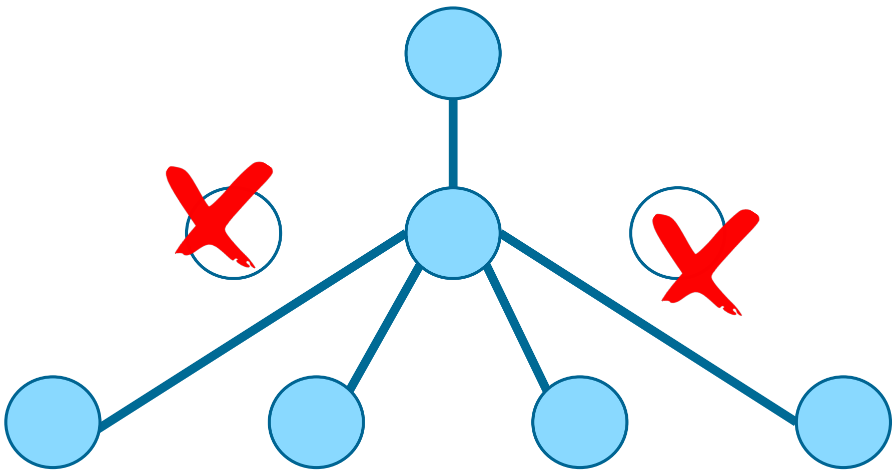
$$\min_{\substack{U,V,r \\ UV^\top = X}} \|Y - UV^\top\|_F^2 + \frac{1-\theta_r}{\theta_r} \sum_{i=1}^{r} \|U_i\|_F^2 \|V_i\|_F^2$$

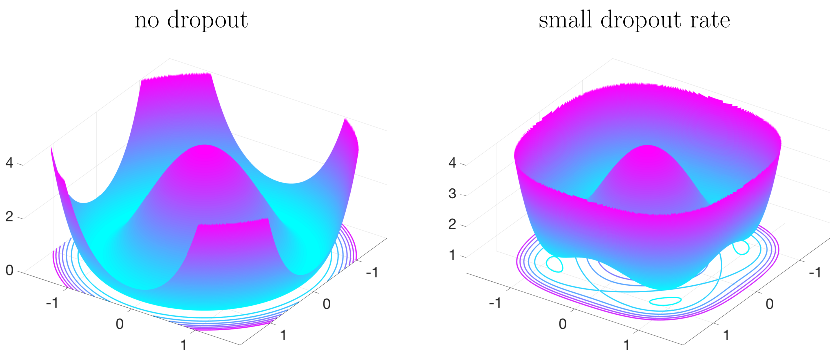
is balanced: $\|U_1^* {V_1^*}^\top\|_F = \|U_2^* {V_2^*}^\top\|_F = \ldots = \|U_r^* {V_r^*}^\top\|_F$

Moreover, $X^* = U^* {V^*}^\top$ can be computed in closed form and is the global minimum of

$$\min_X \|Y - X\|_F^2 + \frac{1-\theta_1}{\theta_1} \|X\|_{r-\mathrm{support}}^2$$

Conclusions





- **Theorem**: Dropout is SGD applied to stochastic objective.
- **Theorem**: Dropout induces explicit low-rank regularization.
- **Theorem**: Dropout induces balanced weights.
- **Theorem**: DropBlock induces r-support norm regularization and balanced weights.

[1] Jacopo Cavazza, Benjamin Haeffele, Pietro Morerio, Connor Lane, Vittorio Murino, Rene Vidal, Dropout as a Low-Rank Regularizer for Matrix Factorization, AISTATS (2018), https://arxiv.org/abs/1710.03487
[2] Poorya Mianjy, Raman Arora, Rene Vidal, On the Implicit Bias of Dropout, ICML (2018), https://arxiv.org/abs/1806.09777
[3] Ambar Pal, Connor Lane, René Vidal, Benjamin D. Haeffele. On the Regularization Properties of Structured Dropout. https://arxiv.org/abs/1910.14186

JOHNS HOPKINS
MATHEMATICAL INSTITUTE
for DATA SCIENCE

Mathematical Institute for Data Science (MINDS)

- Created in November 2017
- Brings together 30 faculty from
  - Applied Mathematics and Statistics
  - Biomedical Engineering, Computer Science
  - Electrical and Computer Engineering
  - Math, Medicine and Biostatistics
- Focus
  - Mathematical, Statistical, Computational Foundations of Data Science
- Funding
  - NSF-Simons Math of Deep Learning
  - NSF TRIPODS Found Graph & Deep Learning
- We are hiring
  - 6 Faculty Positions
  - 4 Bloomberg Distinguished Professors

More Information,

Vision Lab @ JHU

http://www.vision.jhu.edu

Center for Imaging Science @ JHU

http://www.cis.jhu.edu

Mathematical Institute for Data Science @ JHU

http://www.minds.jhu.edu

Thank You!