JHU vision lab

Mathematics of Deep Learning

CDC Tutorial, Melbourne, Australia, December 15th, 2017

Raja Giryes (Tel Aviv University), **Pratik Chaudhari** (UCLA), **René Vidal** (Hopkins)

JOHNS HOPKINS MATHEMATICAL INSTITUTE *for* DATA SCIENCE

THE DEPARTMENT OF BIOMEDICAL ENGINEERING
The Whitaker Institute at Johns Hopkins

Center for IMAGING SCIENCE

CDC 2017 Tutorial Schedule

- 10.00-10.20: **René Vidal** Introduction to Deep Learning
- 10.20-10.40: **René Vidal** Global Optimality in Deep Learning
- 10.40-11.00: **René Vidal** Analysis of Dropout for Factorization
- 11.00-11.20: **Pratik Chaudhari** A Picture of the Energy Landscape of Deep Neural Networks
- 11.20-11.40: **Raja Giryes** Generalization Error for Deep Learning
- 11.40-12.00: **Raja Giryes** Data Structure Based Theory for Deep Learning

More Information

- **Slides**
  - http://vision.jhu.edu/tutorials/CDC17-Tutorial-Math-Deep-Learning.htm
- **Paper**
  - https://arxiv.org/abs/1712.04741

Mathematics of Deep Learning

René Vidal Joan Bruna Raja Giryes Stefano Soatto

***Abstract*— Recently there has been a dramatic increase in the performance of recognition systems due to the introduction of deep architectures for representation learning and classification. However, the mathematical reasons for this success remain elusive. This tutorial will review recent work that aims to provide a mathematical justification for several properties of deep networks, such as global optimality, geometric stability, and invariance of the learned representations.**

sigmoidal activations are universal function approximators [5], [6], [7], [8]. However, the capacity of a wide and shallow network can be replicated by a deep network with significant improvements in performance. One possible explanation is that deeper architectures are able to better capture invariant properties of the data compared to their shallow counterparts. In computer vision, for example, the category of an object

Brief History of Neural Networks

Beginnings

Thresholded Logic Unit — 1943
Perceptron — 1957
Adaline — 1960

1st Neural Winter

XOR Problem — 1969

1982
Multilayer Backprop — 1986
CNNs — 1989

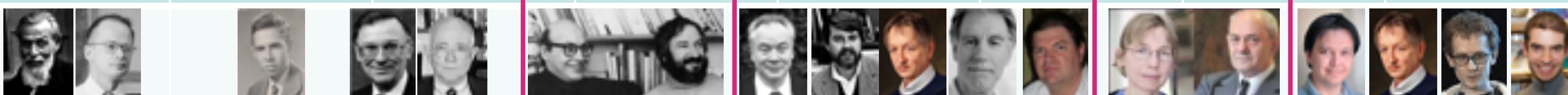
2nd Neural Winter

SVMs — 1995

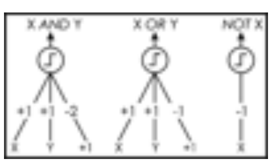
GPU Era

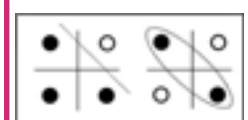
Deep Nets — 2006
Alex Net — 2012

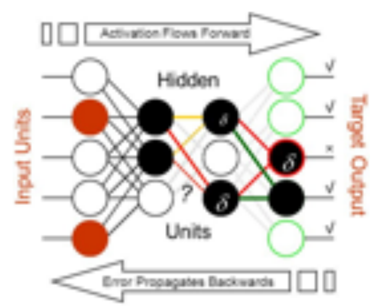
1940	1950	1960	1970	1980	1990	2000	2010

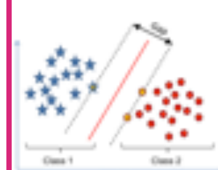


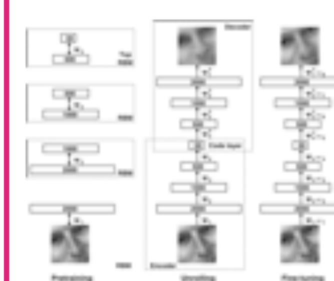
S. McCulloch - W. Pitts
R. Rosenblatt
B. Widrow - M. Hoff
M. Minsky - S. Papert
P. Werbos
D. Rumelhart - G. Hinton - R. Williams
Y. Lecun
C. Cortes - V. Vapnik
R. Salakhutdinov - J. Hinton - A. Krizhevsky - I. Sutskever









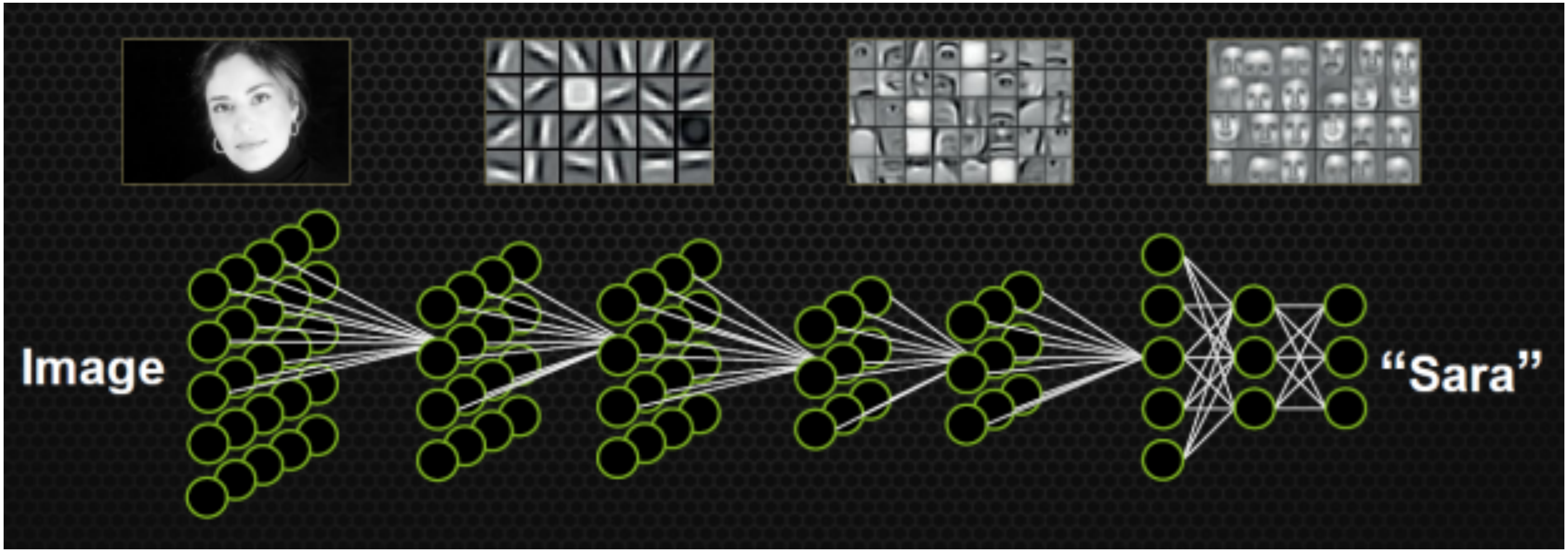


JOHNS HOPKINS
MATHEMATICAL INSTITUTE for DATA SCIENCE

Center for IMAGING SCIENCE

Impact of Deep Learning in Computer Vision

- Deep learning gives ~ 10% improvement on ImageNet
  - 1.2M images
  - 1000 categories
  - 60 million parameters



[1] Krizhevsky, Sutskever and Hinton. ImageNet classification with deep convolutional neural networks, NIPS'12.
[2] Sermanet, Eigen, Zhang, Mathieu, Fergus, LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. ICLR'14.
[3] Donahue, Jia, Vinyals, Hoffman, Zhang, Tzeng, Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. ICML'14.

Impact of Deep Learning in Computer Vision

- 2012-2014 classification results in ImageNet

CNN
non-CNN

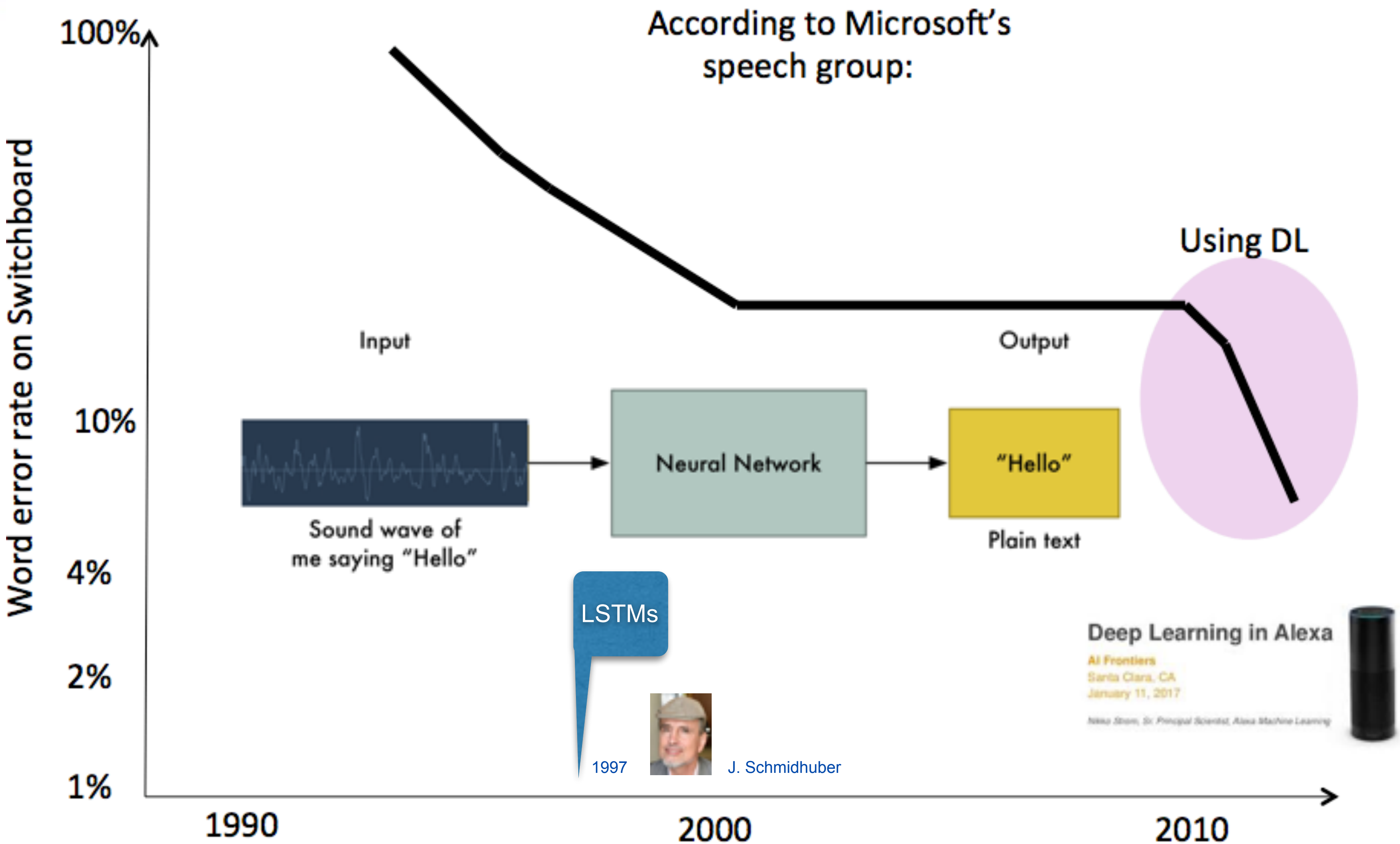
2012 Teams	%error
Supervision (Toronto)	15.3
ISI (Tokyo)	26.1
VGG (Oxford)	26.9
XRCE/INRIA	27.0
UvA (Amsterdam)	29.6
INRIA/LEAR	33.4

2013 Teams	%error
Clarifai (NYU spinoff)	11.7
NUS (singapore)	12.9
Zeiler-Fergus (NYU)	13.5
A. Howard	13.5
OverFeat (NYU)	14.1
UvA (Amsterdam)	14.2
Adobe	15.2
VGG (Oxford)	15.2
VGG (Oxford)	23.0

2014 Teams	%error
GoogLeNet	6.6
VGG (Oxford)	7.3
MSRA	8.0
A. Howard	8.1
DeeperVision	9.5
NUS-BST	9.7
TTIC-ECP	10.2
XYZ	11.2
UvA	12.1

- 2015 results: ResNet under 3.5% error using 150 layers!

Impact of Deep Learning in Speech Recognition

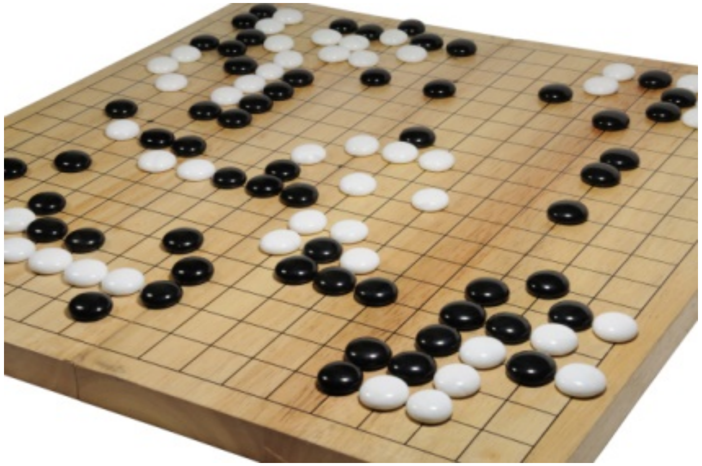


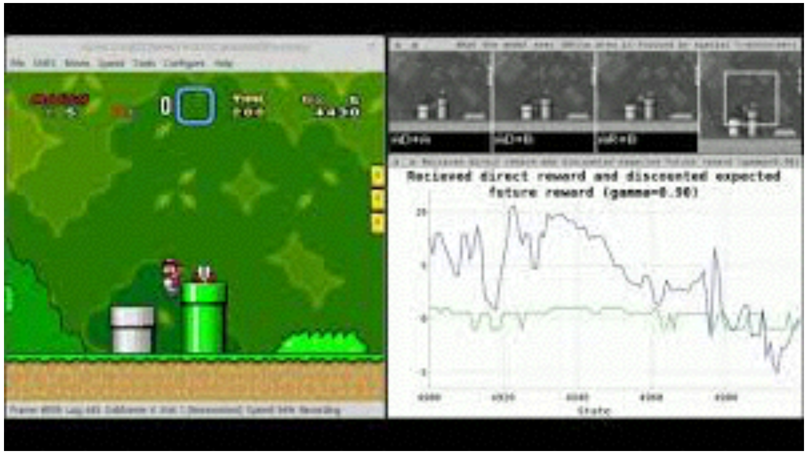
JOHNS HOPKINS
MATHEMATICAL INSTITUTE *for* DATA SCIENCE

Center for IMAGING SCIENCE

Impact of Deep Learning in Game Playing

- **AlphaGo**: the first computer program to ever beat a professional player at the game of Go [1]





- Similar deep reinforcement learning strategies developed to play **Atari Breakout**, **Super Mario**

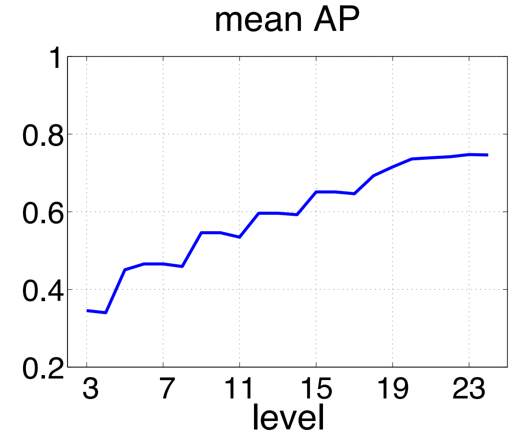
Silver et al. Mastering the game of Go with deep neural networks and tree search, Nature 2016
Artificial intelligence learns Mario level in just 34 attempts, https://www.engadget.com/2015/06/17/super-mario-world-self-learning-ai/, https://github.com/aleju/mario-ai

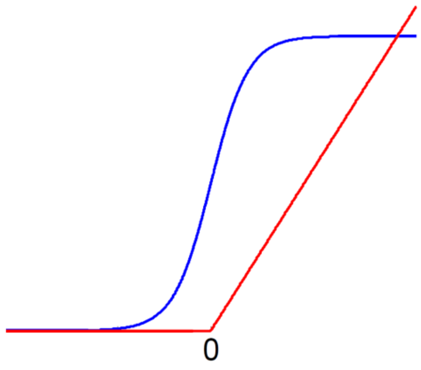
Great Performance in Many Applications

- Disease diagnosis [Zhou, Greenspan & Shen, 2016].
- Language translation [Sutskever et al., 2014]
- Video classification [Karpathy et al., 2014].
- Face detection [Schroff et al., 2015].
- Handwriting recognition [Poznanski & Wolf, 2016].
- Sentiment classification [Socher et al., 2013].
- Image denoising [Burger et al., 2012].
- Super-resolution [Kim et al., 2016], [Bruna et al., 2016].

Why These Improvements in Performance?

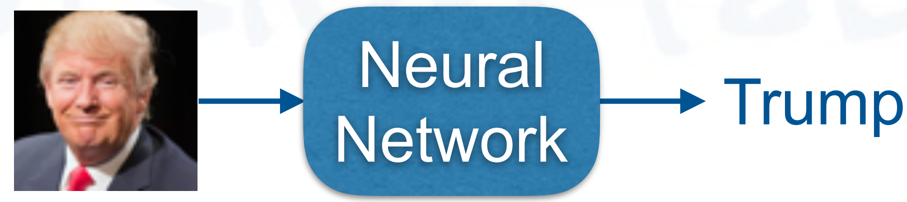
- Features are learned rather than hand-crafted
- More layers capture more invariances [1]
- More data to train deeper networks
- More computing (GPUs)
- Better regularization: Dropout
- New nonlinearities
  - Max pooling, Rectified linear units (ReLU) [2]
- Theoretical understanding of deep networks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

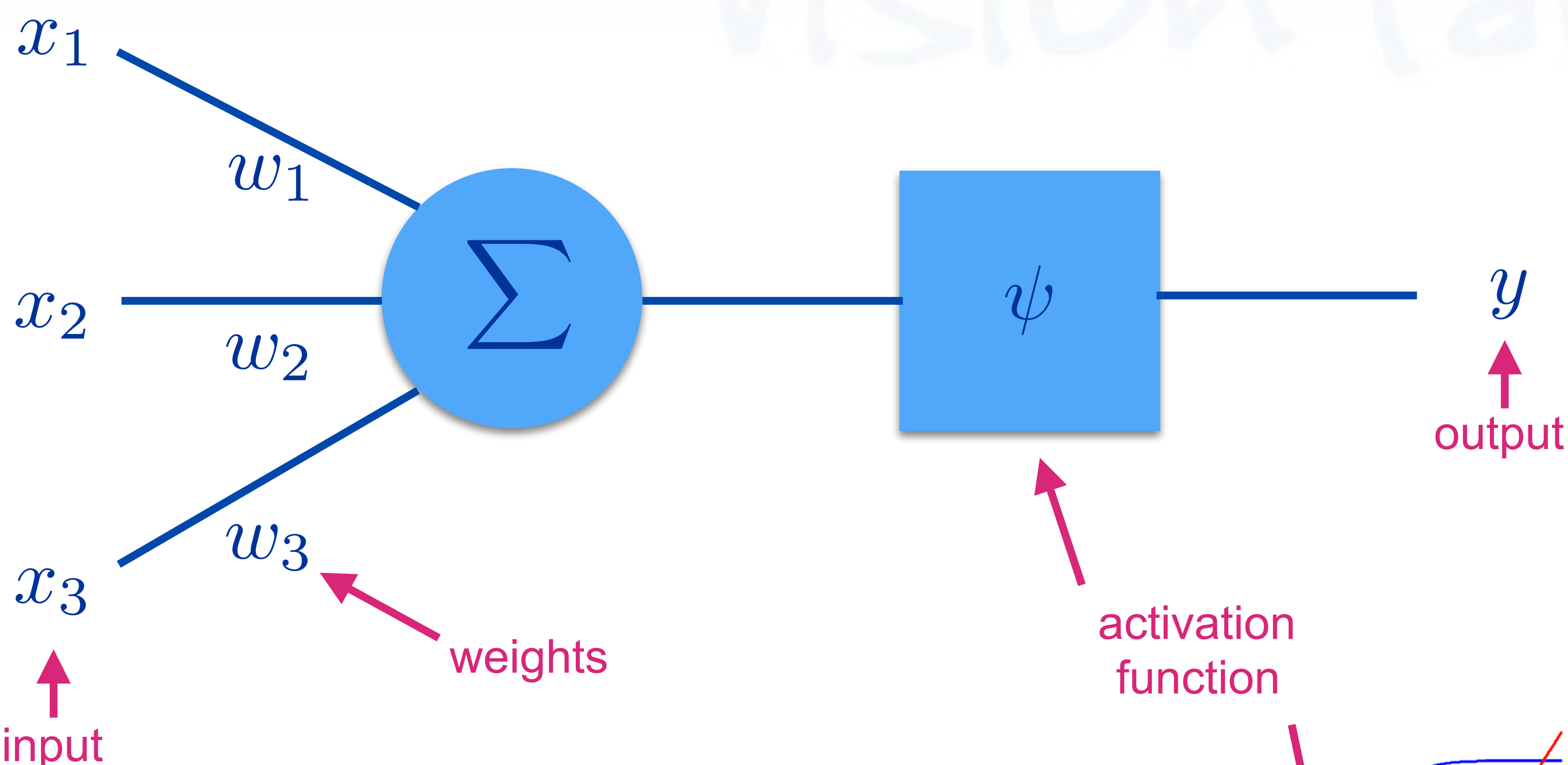
Control Systems vs Neural Networks



- **Control System**
  - **Input**: u
  - **State**: x
  - **Output**: y
  - **System**: (A,B,C,D), f(x,u), h()
- **Neural Network**
  - **Input**: image, audio, data
  - **State**: neuronal responses
  - **Output**: label, label sequence
  - **System**: weights, activations

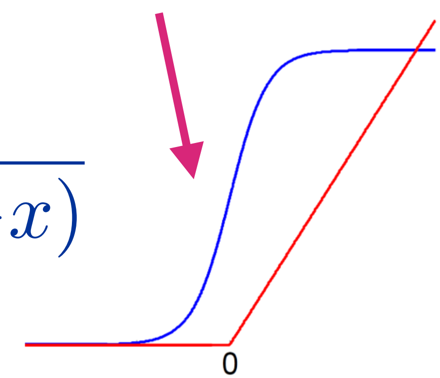
Control System	**Neural Network**
Openloop system	Feedforward network
Closedloop system	Recurrent neural network
State estimation	Inference of hidden variables
System identification	Parameter learning
Prediction error	Loss or risk

Notation: Single Neuron Architecture

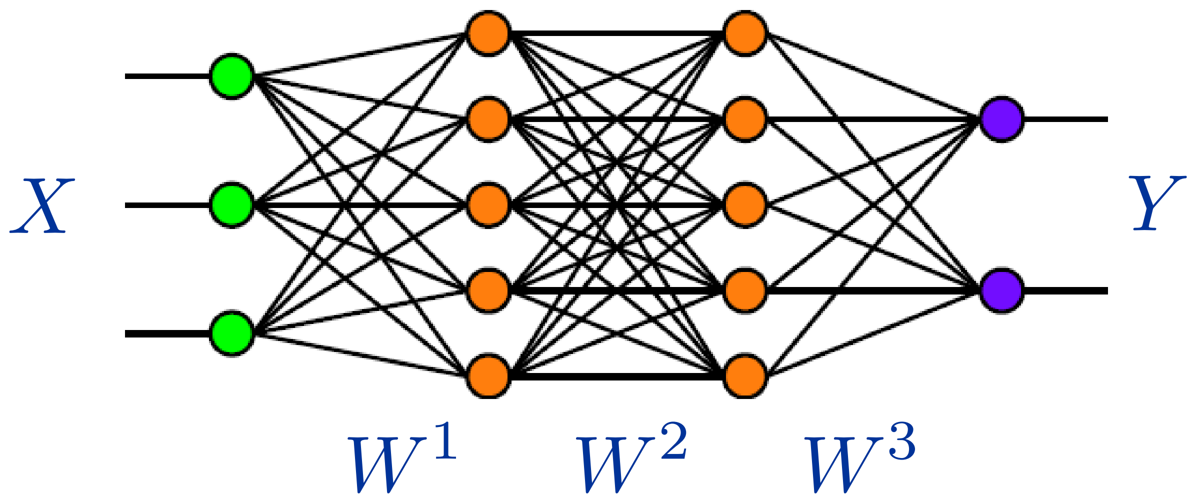


$$\text{sigmoid:} \quad \psi(x) = \frac{1}{1 + \exp(-x)}$$

$$\text{ReLU:} \quad \psi(x) = \max(x, 0)$$



Notation: Multilayer Network Architecture



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

output activation input weights

Notation: Expected and Empirical Loss

- Assume $Y = \Phi^*(X)$. Find W that minimizes **expected loss**

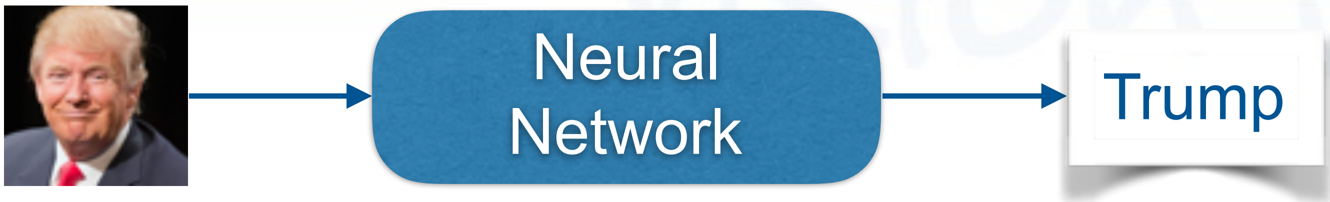
$$W^* = \operatorname*{argmin}_W f(W) = \mathbb{E}_{(X,Y)}[\ell(Y, \Phi(X, W))]$$

- Since joint distribution of (X,Y) is unknown, find W that minimizes **empirical loss**

$$W_N^* = \operatorname*{argmin}_W f_N(W) = \frac{1}{N}\sum_{i=1}^N \ell(Y_i, \Phi(X_i, W))$$

- **Approximation error**: $\mathrm{AE} = f(W^*) - f(\Phi^*)$
- **Generalization error:** $\mathrm{GE} = f(W_N^*) - f(W^*)$
- **Optimization error:** $\mathrm{OE} = f(W_N) - f(W_N^*)$

Notation: Regularized Loss



- Given training examples (X,Y), find model parameters W that minimize regularized loss (classification error)

output (labels) input (data)

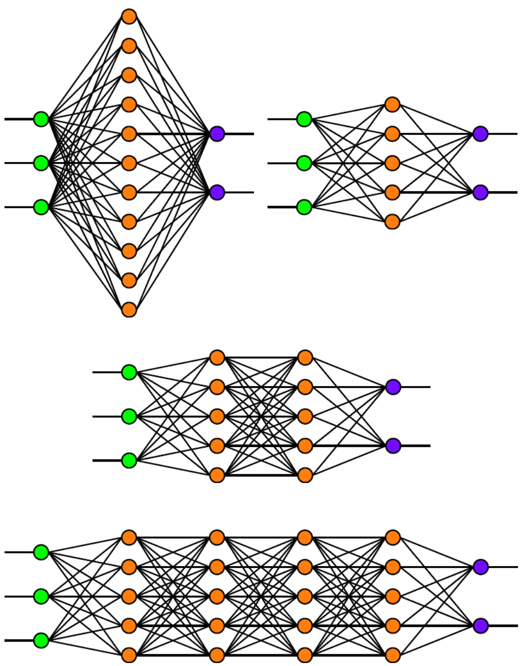
$$\min_{W} \ell(Y, \Phi(X, W)) + \lambda\Theta(W)$$

loss function prediction function regularization function

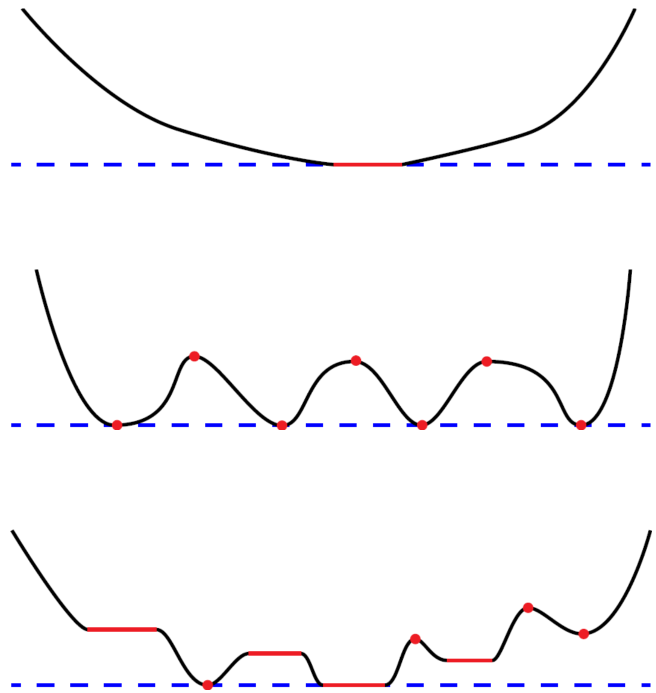
- **Architecture** Φ designed to control **approximation error**
- **Regularizer** Θ designed to control **generalization error**
- **Optimizer** designed to control **optimization error**

Key Theoretical Questions

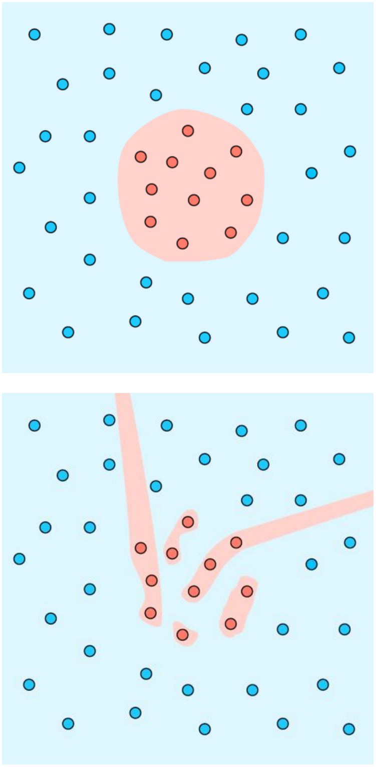
Architecture Design



Optimization

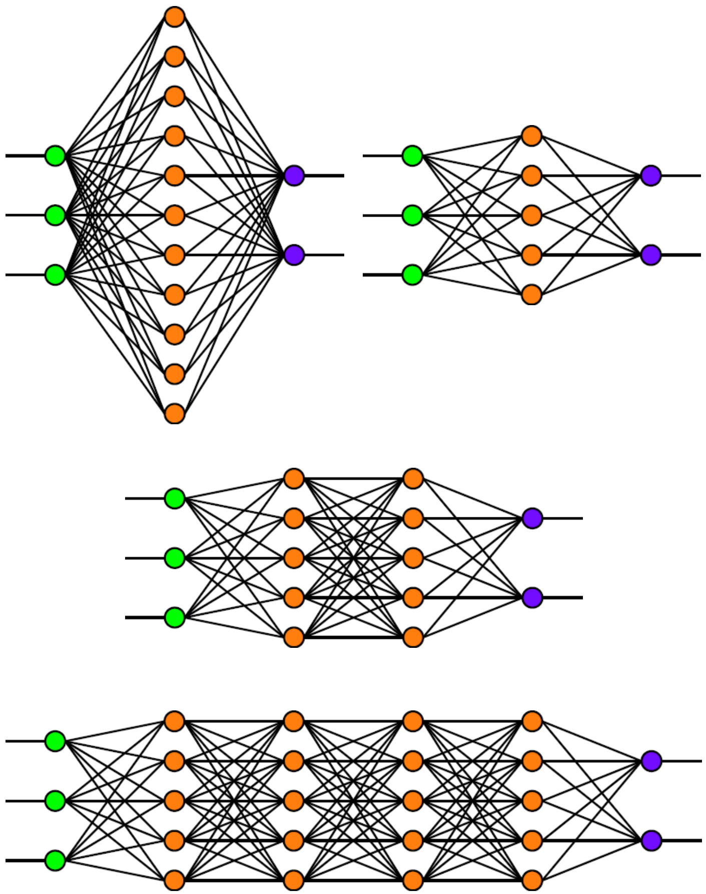


Generalization



Key Theoretical Questions: Architecture

- **Are there principled ways to design networks?**
  - How many layers?
  - Size of layers?
  - Choice of layer types?
  - What classes of functions can be approximated by a feedforward neural network?
  - How does the architecture impact expressiveness? [1]



Slide courtesy of Ben Haeffele

[1] Cohen, et al., "On the expressive power of deep learning: A tensor analysis." COLT. (2016)

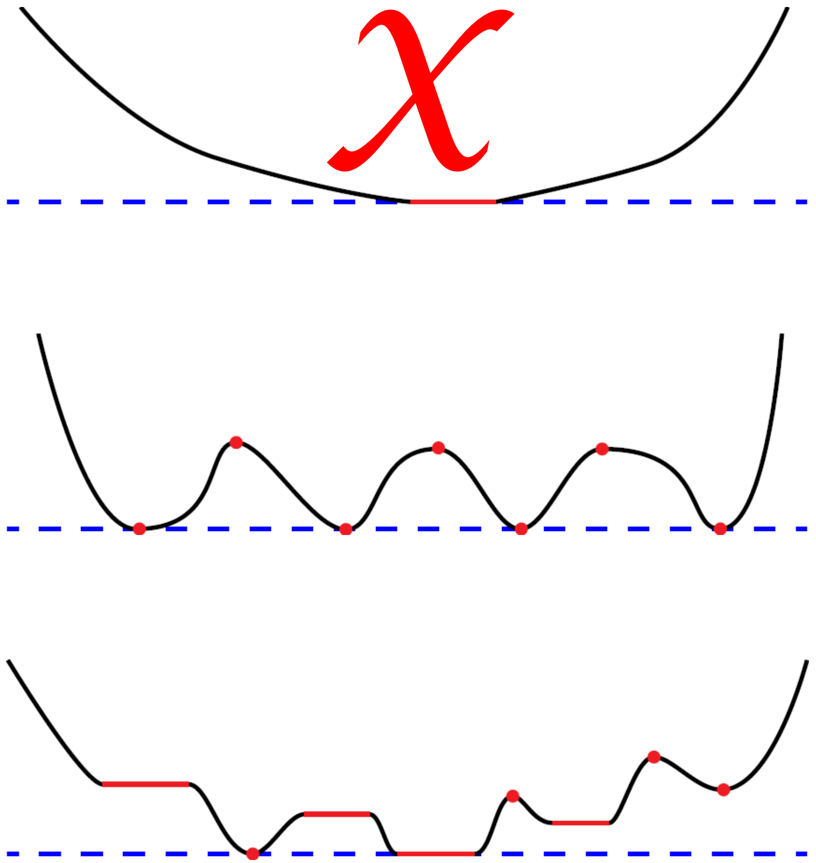
Key Theoretical Questions: Architecture

- **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]
  - Memory-optimal neural network approximation [Bölcskei ’17]

[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. Memory-optimal neural network approximation. Wavelets and Sparsity 2017.

Key Theoretical Questions: Optimization

- **How to train neural networks?**
  - Problem is non-convex
  - What does the error surface look like?
  - How to guarantee optimality?
  - When does local descent succeed?



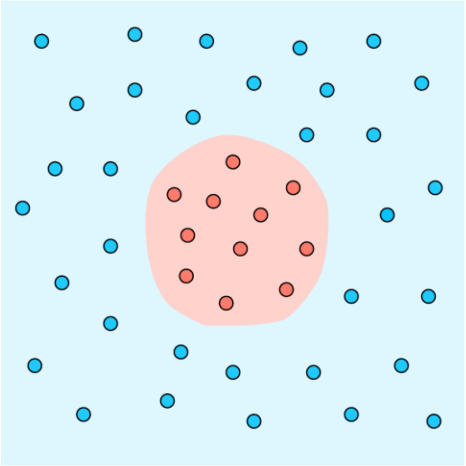
Key Theoretical Questions: Optimization

- **Optimization theory: earlier work**
  - No spurious local minima for linear networks [Baldi-Hornik ’89]
  - 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**
  - Convex neural networks in infinite number of variables [Bengio ’05]
  - Networks with many hidden units can learn polynomials [Andoni ’14]
  - The loss surface of multilayer networks [Choromanska ’15]
  - Attacking the saddle point problem [Dauphin ’14]
  - Effect of gradient noise on the energy landscape: [Chaudhari ‘15]
  - Entropy-SGD is biased toward wide valleys: [Chaudhari ‘17]
  - Deep relaxation: PDEs for optimizing deep nets [Chaudhari ‘17]
  - Guaranteed training of NNs using tensor methods [Janzamin ’15]
  - No spurious local minima for large networks [Haeffele-Vidal’15 Soudry’16]

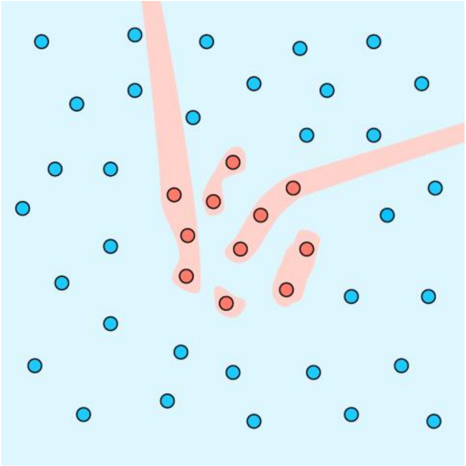
Key Theoretical Questions: Generalization

- **Classification performance guarantees?**
  - How well do deep networks generalize?
  - How should networks be regularized?
  - How to prevent under or over fitting?

✓ Simple



X Complex



Slide courtesy of Ben Haeffele

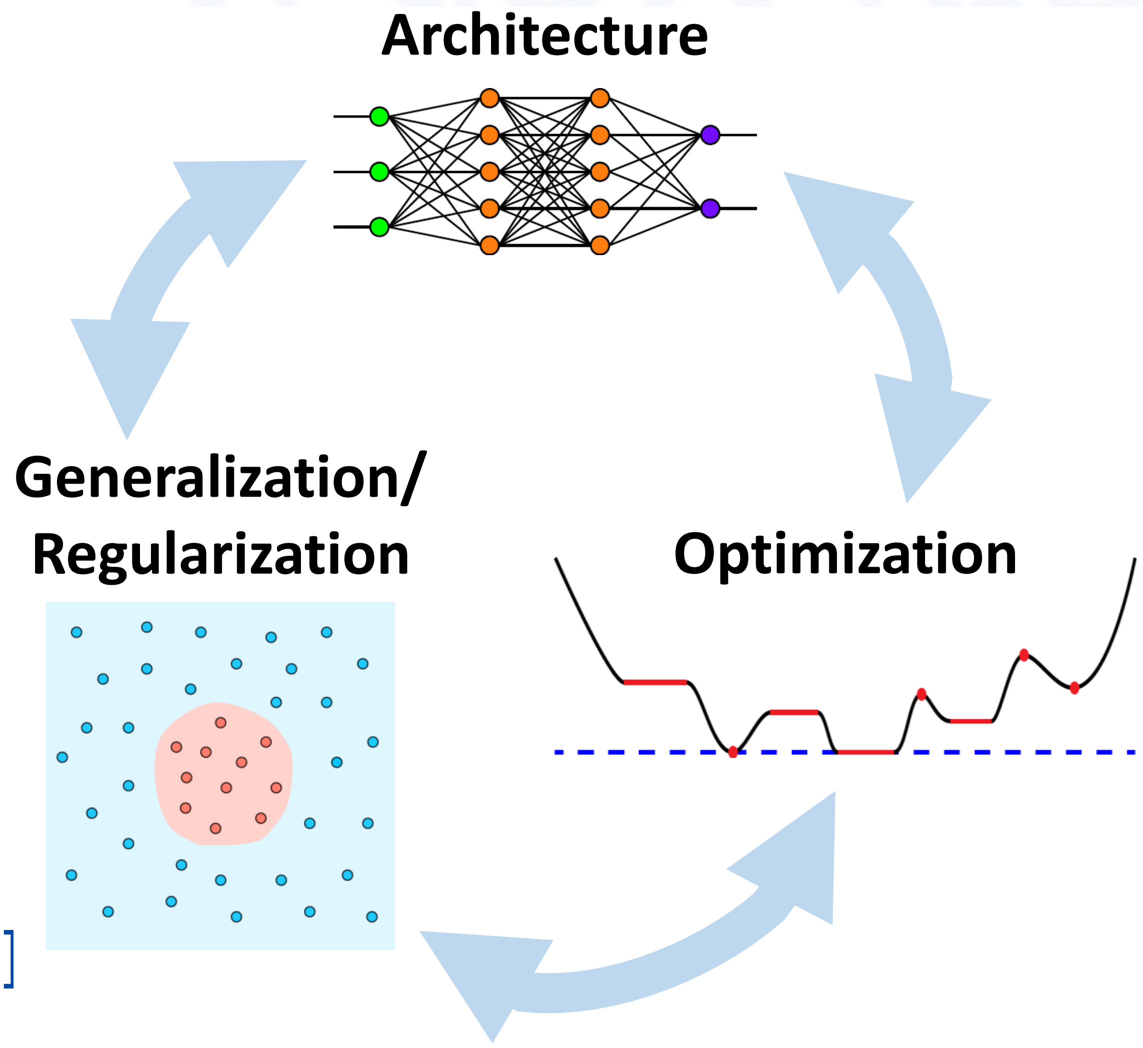
Key Theoretical Questions: Generalization

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

[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. NIPS01.
[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] Giryes, Sapiro, Bronstein. Deep Neural Networks with Random Gaussian Weights. arXiv:1504.08291.
[7] Sokolic. Margin Preservation of Deep Neural Networks, 2015
[8] Neyshabur. Path-SGD: Path-Normalized Optimization in Deep Neural Networks. NIPS 2015
[9] Behnam Neyshabur. Implicit Regularization in Deep Learning. PhD Thesis 2017
[10] Sokolic, Giryes, Sapiro, Rodrigues. Generalization error of invariant classifiers. In AISTATS, 2017.
[11 Sokolić, Giryes, Sapiro, Rodrigues. Robust Large Margin Deep Neural Networks. IEEE Transactions on Signal Processing, 2017.
[12] Shwartz-Ziv, Tishby. Opening the black box of deep neural networks via information. arXiv:1703.00810, 2017.
[13] Achille, Soatto. Information dropout: Learning optimal representations through noisy computation. *arXiv:* 2016.
[14] Liang, Poggio, Rakhlin, Stokes. Fisher-Rao Metric, Geometry and Complexity of Neural Networks. arXiv: 2017.
[15] Zhang, Bengio, Hardt, Recht, Vinyals. Understanding deep learning requires rethinking generalization. ICLR 2017.

Key Theoretical Questions are Interrelated

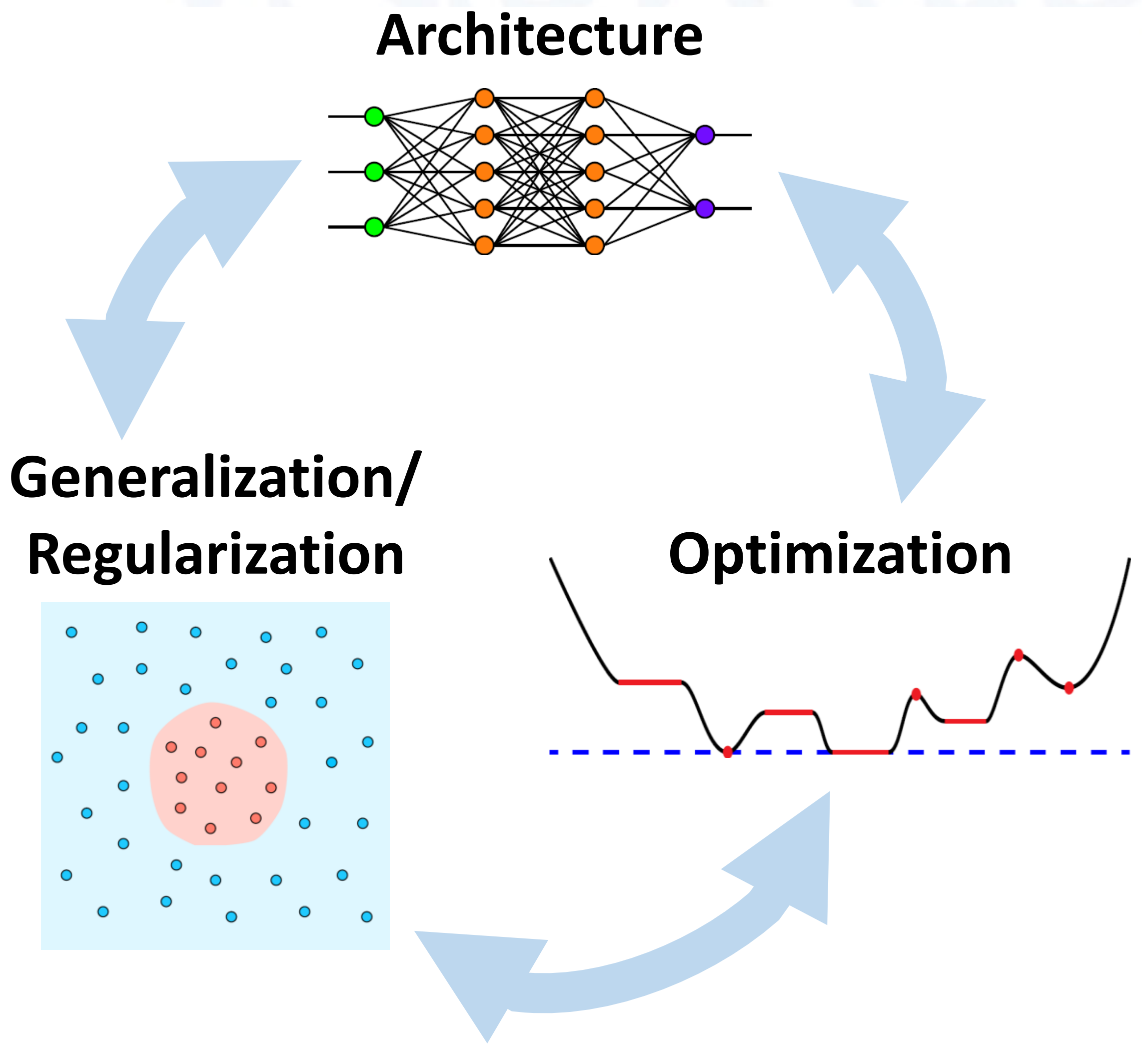
- Optimization can impact generalization [1,2]
- Architecture has strong effect on generalization [3]
- Some architectures could be easier to optimize than others [4]



[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]
- Regularization with product of weights generalizes well [2]
- No spurious local minima for product of weight regularizers [3]



[1] Cavazza, Lane, Moreiro, Haeffele, Murino, Vidal. An Analysis of Dropout for Matrix Factorization, arXiv 2017
[2] Sokolić, R. Giryes, G. Sapiro, and M. Rodrigues. Generalization error of Invariant Classifiers. AISTATS, 2017.
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

CDC 2017 Tutorial Schedule

- 10.00-10.20: **René Vidal** Introduction to Deep Learning
- 10.20-10.40: **René Vidal** Global Optimality in Deep Learning
- 10.40-11.00: **René Vidal** Analysis of Dropout for Factorization
- 11.00-11.20: **Pratik Chaudhari** A Picture of the Energy Landscape of Deep Neural Networks
- 11.20-11.40: **Raja Giryes** Generalization Error for Deep Learning
- 11.40-12.00: **Raja Giryes** Data Structure Based Theory for Deep Learning