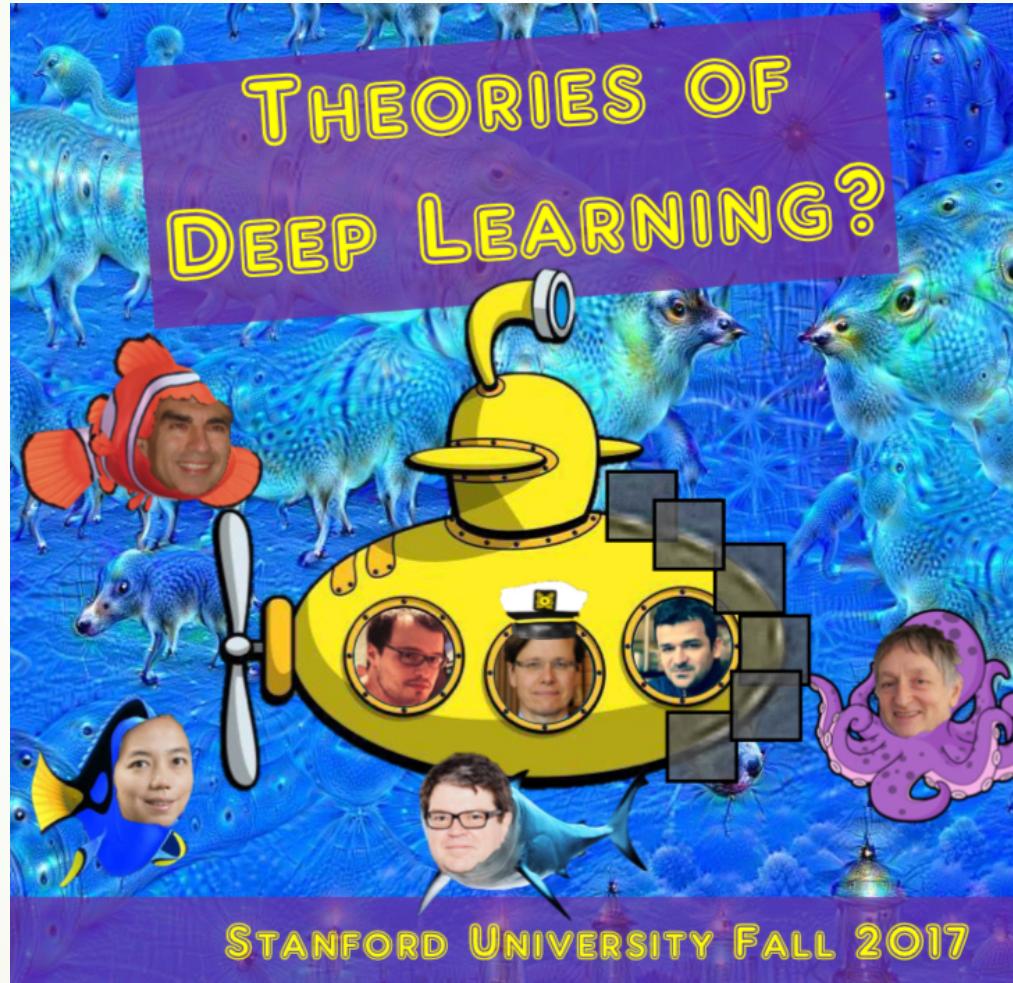


On Mathematical Theories of Deep Learning

Yuan YAO
HKUST

Acknowledgement



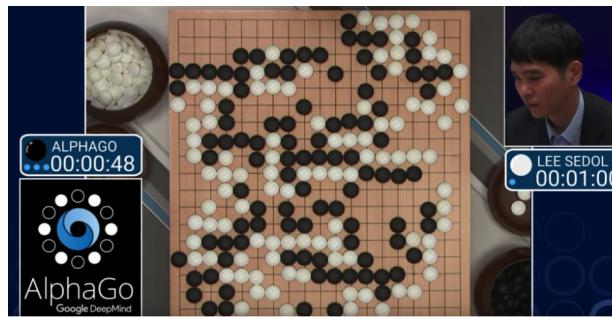
A following-up course at HKUST: <https://deeplearning-math.github.io/>



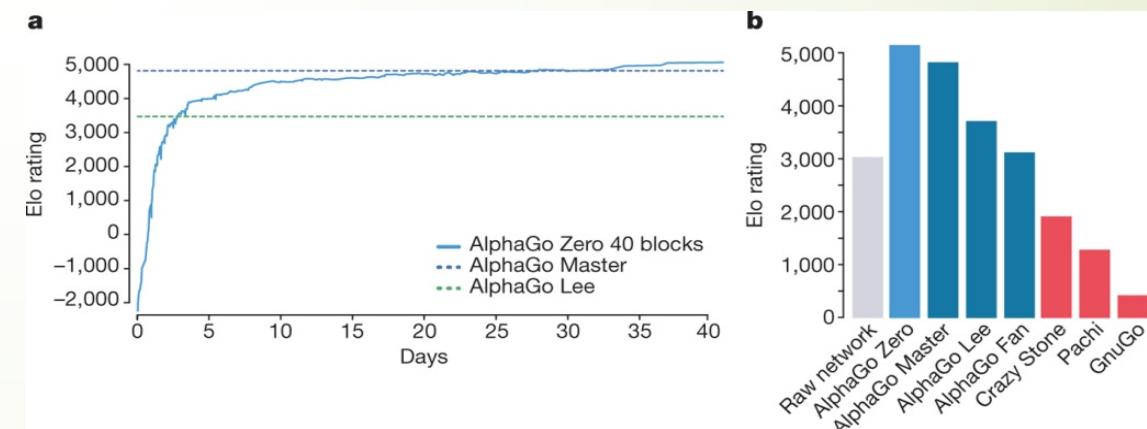
Outline

- ▶ Why mathematical theories of Deep Learning?
 - ▶ The tsunami of deep learning in recent years...
- ▶ What Theories Do We Have or Need?
 - ▶ Harmonic Analysis: what are optimal representation of functions?
 - ▶ Approximation Theory: when deep networks are better than shallow ones?
 - ▶ Statistics: how deep net models can generalize well against noise?
 - ▶ Optimization: when the landscapes of loss are simple and how to efficiently find a good solution?

Reaching Human Performance Level



AlphaGo “LEE” 2016



ImageNet Dataset

- 14,197,122 labeled images
- 21,841 classes
- Labeling required more than a year of human effort via Amazon Mechanical Turk

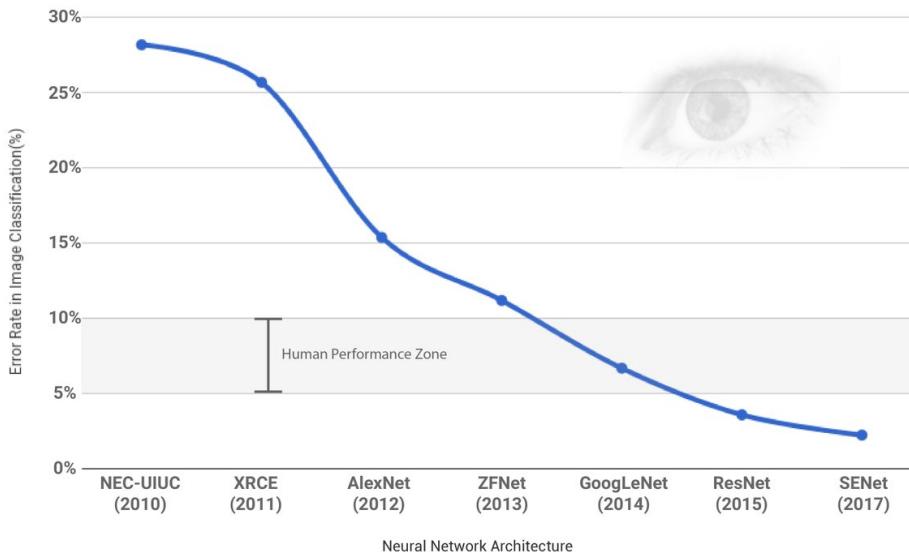
IMAGENET



Stanford University

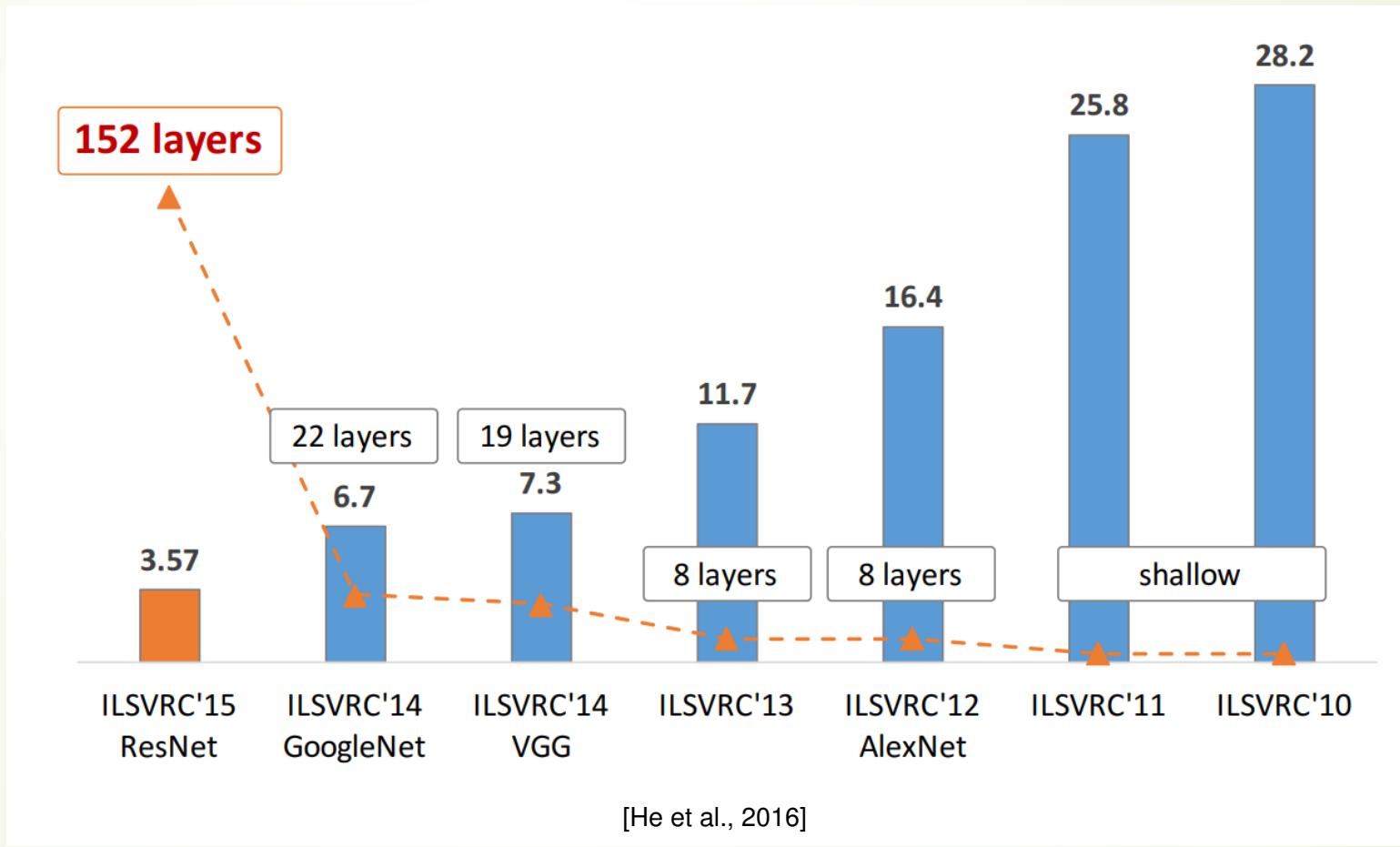
ImageNet Top 5 classification error

- ImageNet (subset):
 - 1.2 million training images
 - 100,000 test images
 - 1000 classes
- ImageNet large-scale visual recognition Challenge



source: <https://www.linkedin.com/pulse/must-read-path-breaking-papers-image-classification-muktabh-mayank>

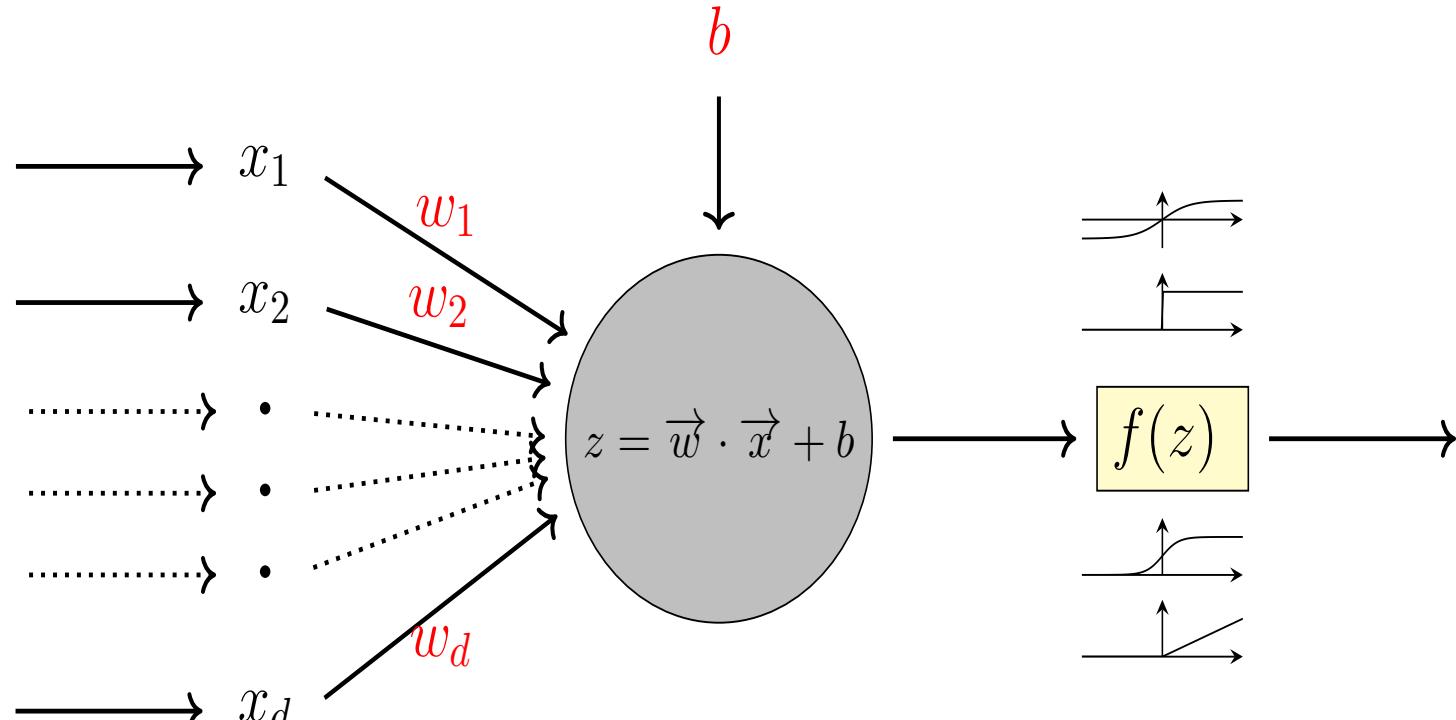
Depth as function of year



Perceptron: single-layer



- Invented by Frank Rosenblatt (1957)



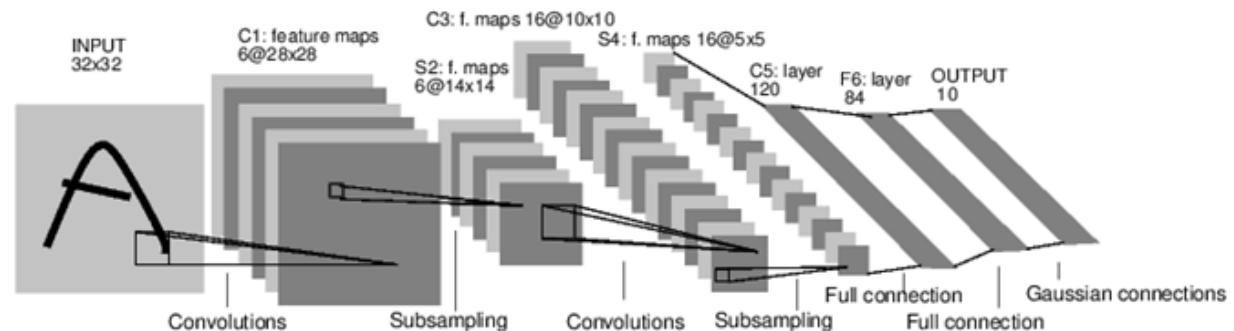
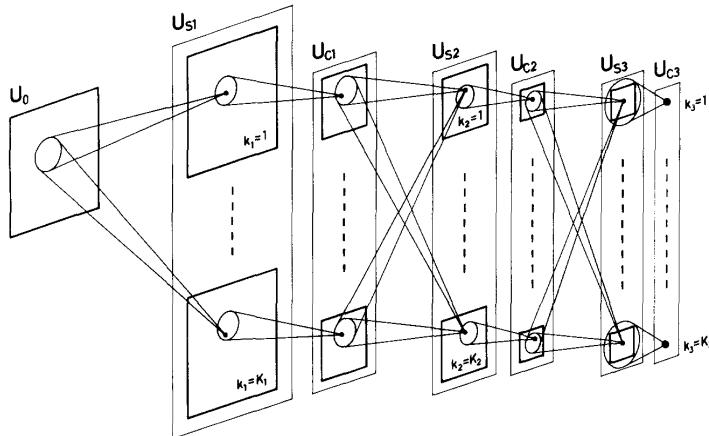
Convolutional Neural Networks

- Can be traced to *Neocognitron* of Kunihiko Fukushima (1979)
- Yann LeCun combined convolutional neural networks with back propagation (1989)
- Imposes **shift invariance** and **locality** on the weights
- Forward pass remains similar
- Backpropagation slightly changes – need to sum over the gradients from all spatial positions

Biol. Cybernetics 36, 193–202 (1980)

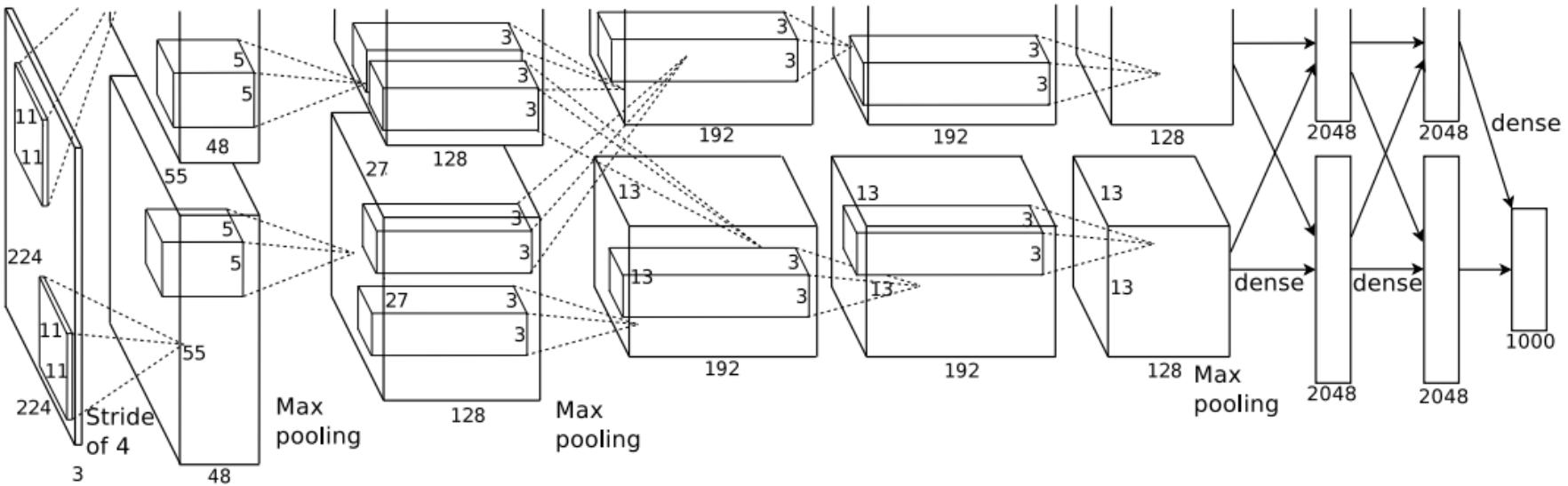
Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position

Kunihiko Fukushima
NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan



AlexNet (2012)

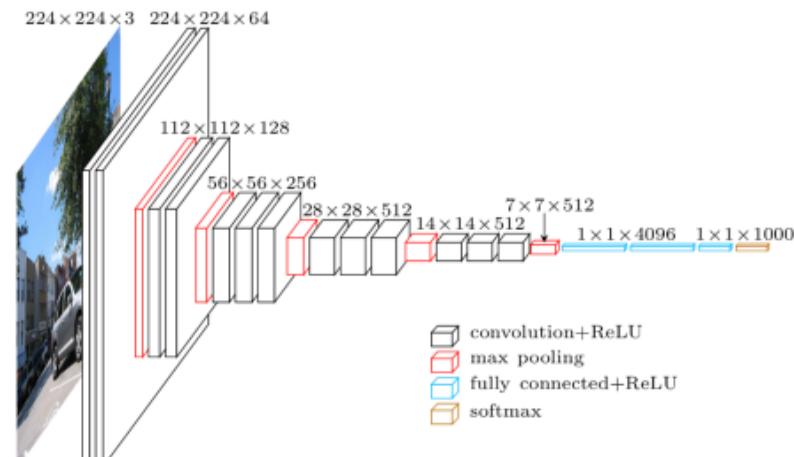
- 8 layers: first 5 convolutional, rest fully connected
- ReLU nonlinearity
- Local response normalization
- Max-pooling
- Dropout



Source: [Krizhevsky et al., 2012]

VGG (2014) [Simonyan-Zisserman'14]

- Deeper than AlexNet: 11-19 layers versus 8
- No local response normalization
- Number of filters multiplied by two every few layers
- Spatial extent of filters 3×3 in all layers
- Instead of 7×7 filters, use three layers of 3×3 filters
 - Gain intermediate nonlinearity
 - Impose a regularization on the 7×7 filters

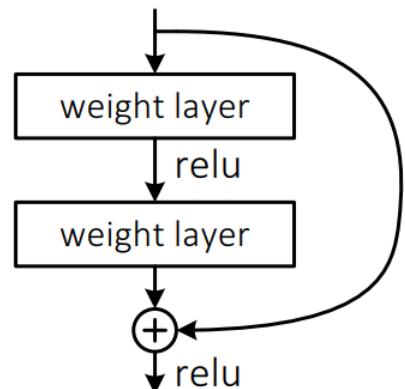


Stanford University

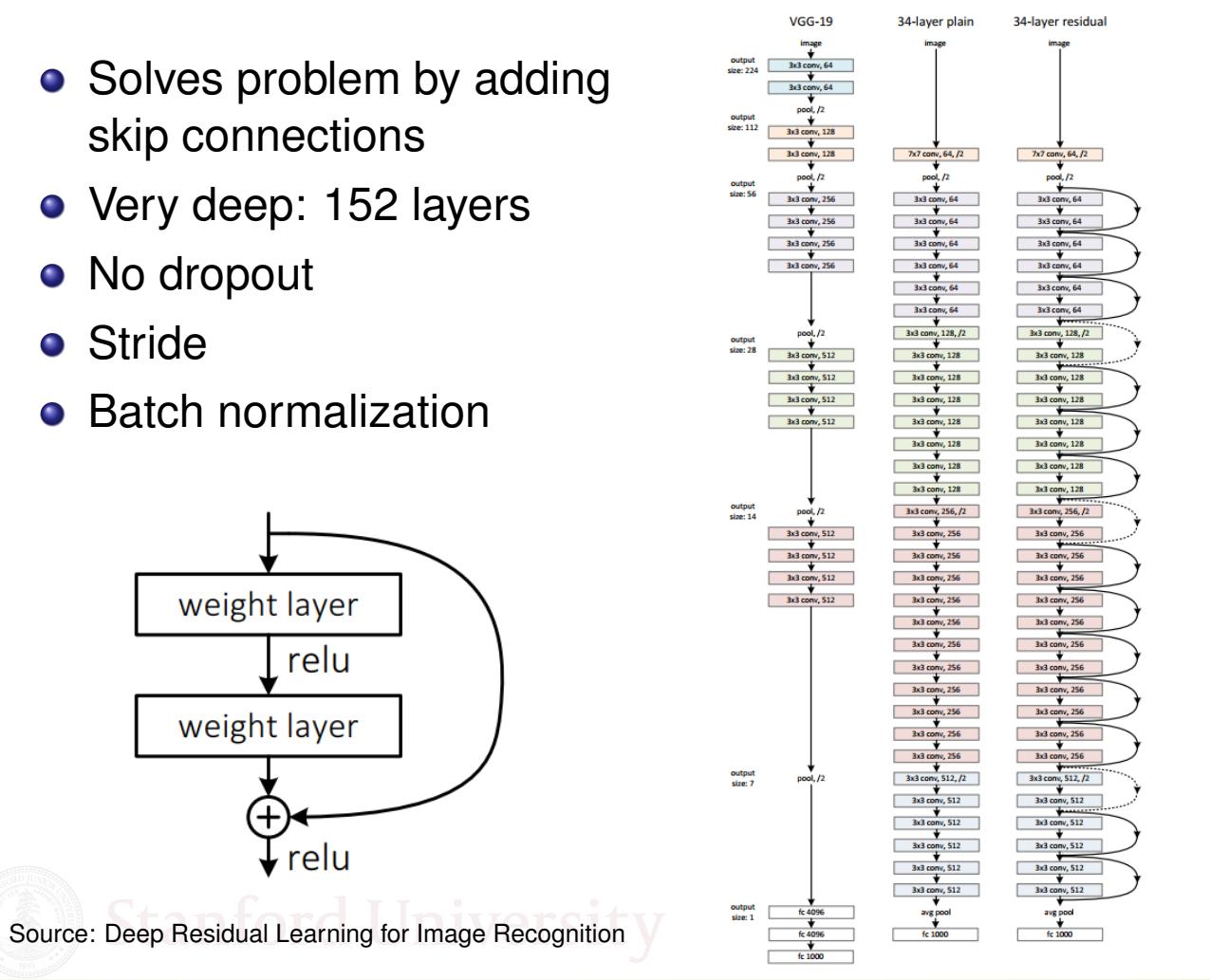
Source: <https://blog.heuritech.com/2016/02/29/>

ResNet (2015) [HGRS-15]

- Solves problem by adding skip connections
- Very deep: 152 layers
- No dropout
- Stride
- Batch normalization



Source: Deep Residual Learning for Image Recognition



Visualizing Deep Neural Networks

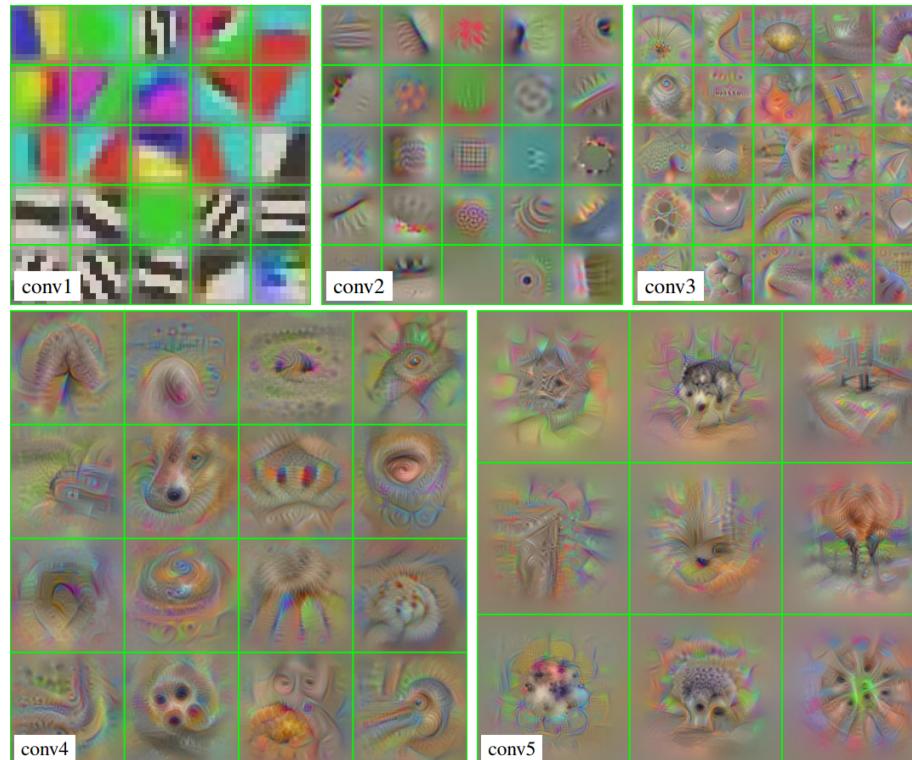
- Filters in first layer of CNN are easy to visualize, while deeper ones are harder
- *Activation maximization* seeks input image maximizing output of the i -th neuron in the network
- Objective

$$x^* = \arg \min_x \mathcal{R}(x) - \langle \Phi(x), e_i \rangle$$

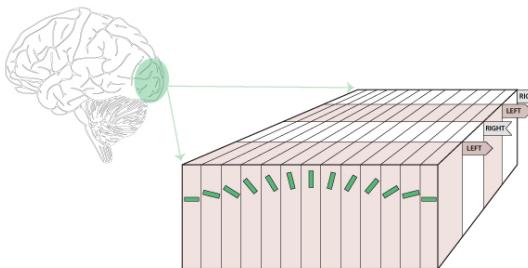
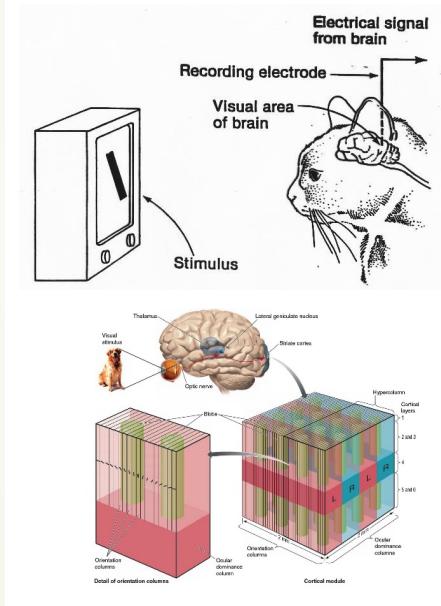
- e_i is indicator vector
- $\mathcal{R}(x)$ is simple natural image prior

Visualizing VGG

- Gabor-like images in first layer
- More sophisticated structures in the rest

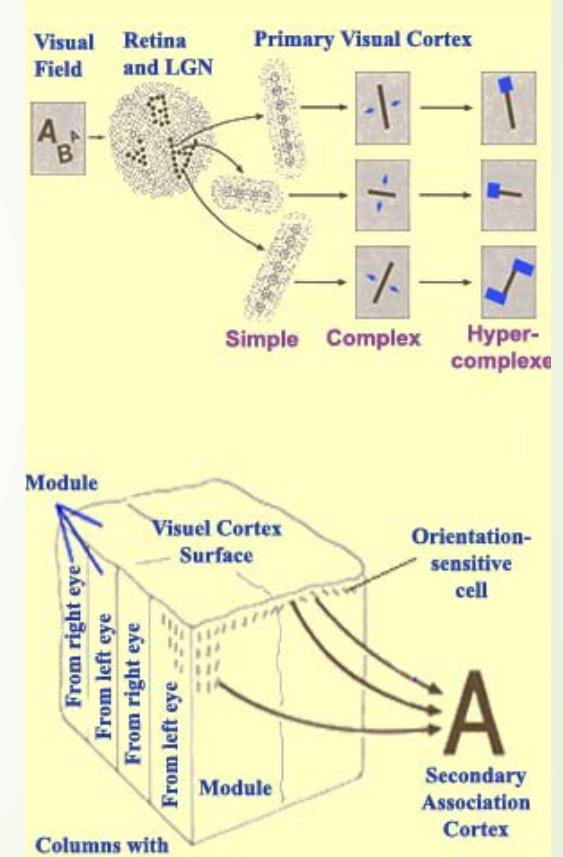
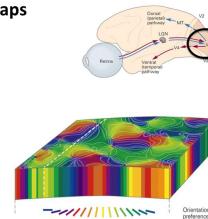


Visual Neuroscience: Hubel/Wiesel, ...



V1 orientation maps

- Continuous maps of orientation preference: "pinwheels"
- Consistent preference though the depth of cortex: columnar architecture



Olshausen and Field 1996

Experimental Neuroscience uncovered the

- ▶ ... neural architecture of Retina/LGN/V1/V2/V3/ etc
- ▶ ... existence of neurons with weights and activation functions (simple cells)
- ▶ ... pooling neurons (complex cells)

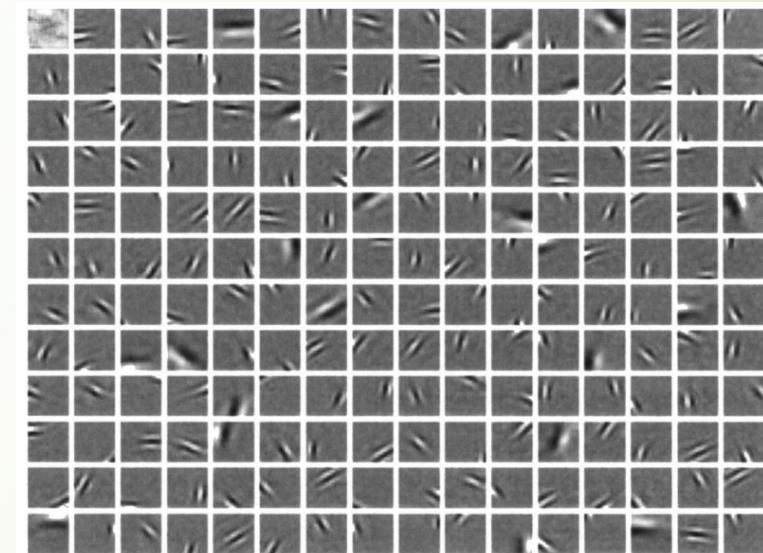
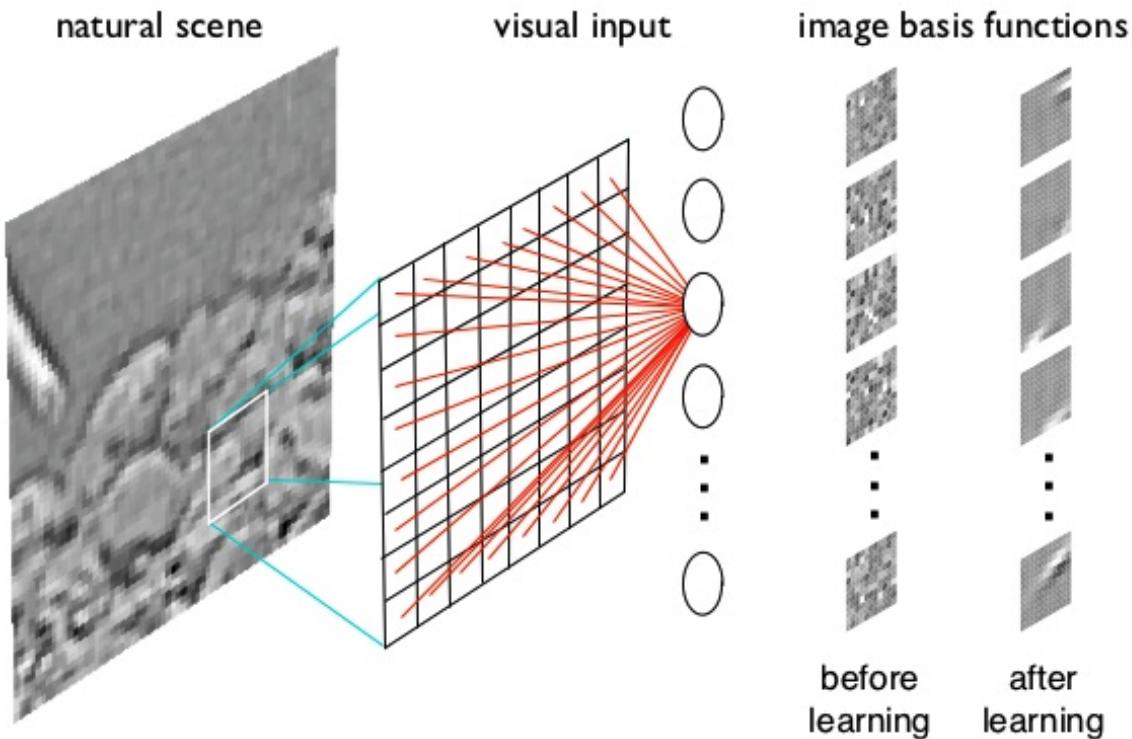
All these features are somehow present in today's sucessful Deep Learning systems

Neuroscience	Deep Network
Simple cells	First layer
Complex cells	Pooling Layer
Grandmother cells	Last layer

Theorists Olshausen and Field (Nature, 1996) demonstrated that receptive fields learned from image patches

First layers learned ...

Efficient coding of natural images: Olshausen and Field, 1996

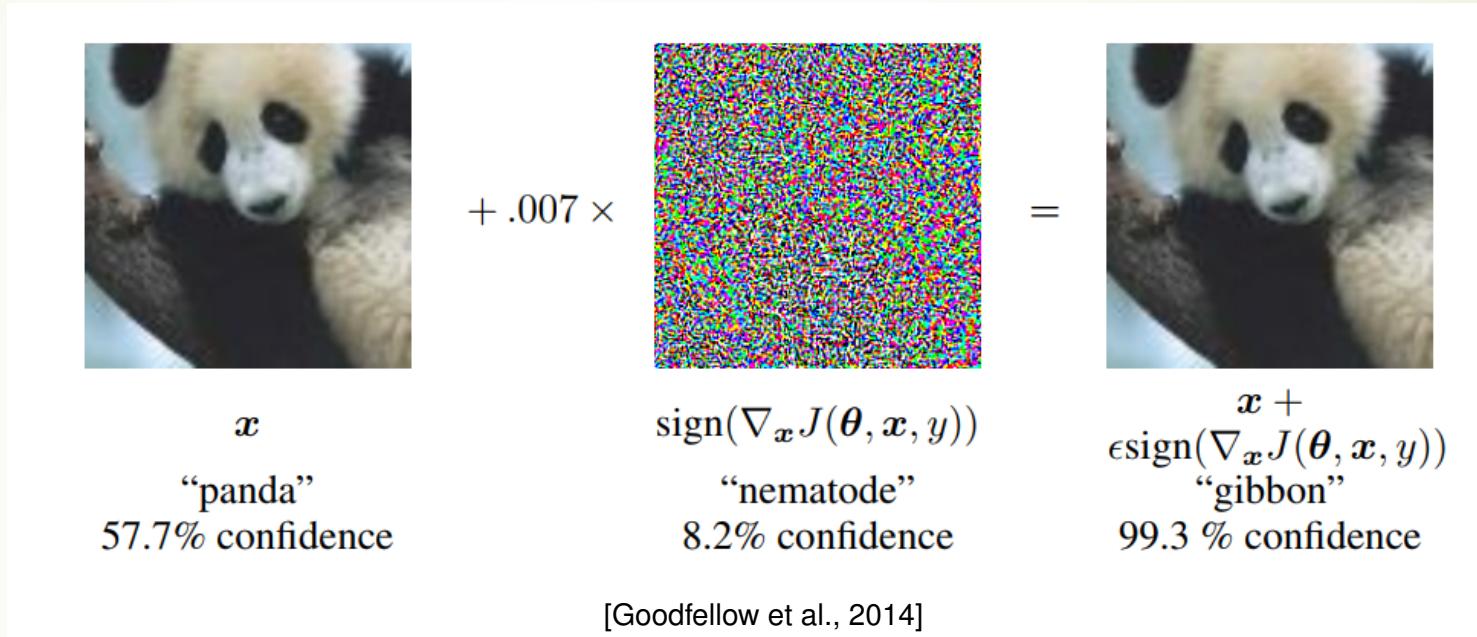


Network weights are adapted to maximize coding efficiency:
minimizes redundancy and maximizes the independence of the outputs

Transfer Learning

- Filters learned in first layers of a network are transferable from one task to another
- When solving another problem, no need to retrain the lower layers, just fine tune upper ones
- Is this simply due to the large amount of images in ImageNet?
- Does solving many classification problems simultaneously result in features that are more easily transferable?
- Does this imply filters can be learned in unsupervised manner?
- Can we characterize filters mathematically?

But Deep Learning may be fragile in generalization against noise!



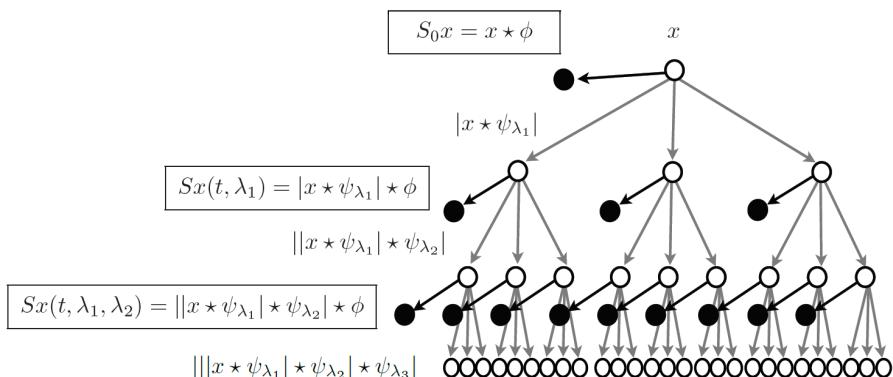
- Small but malicious perturbations can result in severe misclassification
- Malicious examples generalize across different architectures
- What is source of instability?
- Can we robustify network?

Some Open Theoretical Problems

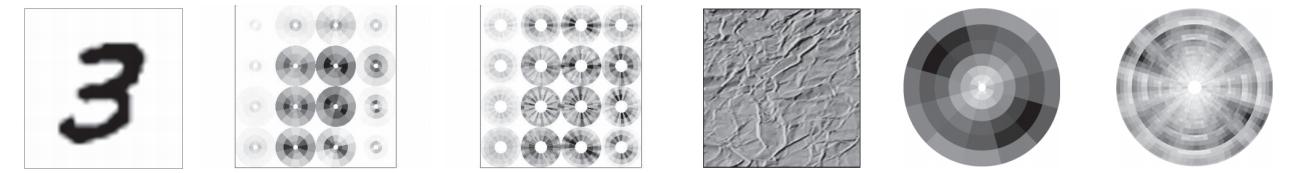
- ▶ *Harmonic Analysis:* What are the optimal (transferrable) representations of functions as input signals (sounds, images, ...)?
- ▶ *Approximation Theory:* When and why are deep networks better than shallow networks?
- ▶ *Optimization:* What is the landscape of the empirical risk and how to minimize it efficiently?
- ▶ *Statistics:* How can deep learning generalize well without overfitting the noise?

Harmonic Analysis

- Harmonic analysis: optimal representation of input signals
- Wavelets are optimal sparse representations for certain class of images
- Deep Scattering Net: translational and rotational invariances
- Yesterday: **Mathew Hirn** talked about scattering net for energy functions on 3-D densities (images)



Scattering Transform:
Mallat'12



Approximation Theory

- ▶ Class prediction rule can be viewed as function $f(x)$ of high-dimensional argument
- ▶ *Curse of Dimensionality*
 - ▶ Traditional theoretical obstacle to high-dimensional approximation
 - ▶ “*Functions of high dimensional x can wiggle in too many dimensions to be learned from finite datasets*”

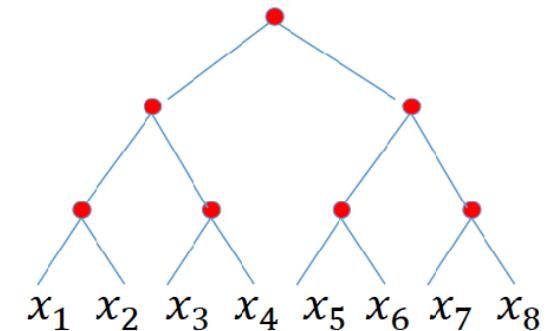
- 
- ▶ Ridge Functions $\rho(u'x)$ mathematically same as deep learning first layer outputs.
 - ▶ Sums of Ridge Functions mathematically same as input to second layer.
 - ▶ Approximation by Sums of Ridge Functions $f \approx \sum_i \rho_i(u'_i x)$ studied for decades
 - ▶ Theorists (1990's-Today): certain functions $f(x)$ approximated by ridge sums with no curse of dimensionality

Compositional Functions

- ▶ Compositional functions $f(x) = h(g_1(x_{i_{1,1}}, \dots, x_{i_{1,k}}), g_2(x_{i_{2,1}}, \dots, x_{i_{2,k}}), \dots, g_\ell(x_{i_{\ell,1}}, \dots, x_{i_{\ell,k}}))$ are functions of small number of functions; $\ell, k \ll d$.
- ▶ VGG Nets are deep compositions
- ▶ Approximation by Compositional Functions studied for decades
- ▶ Theorists (1990's-Today): certain functions $f(x)$ avoid curse of dimensionality using multilayer compositions
- ▶ T. Poggio (MIT) and Hrushikesh Mhaskar (Caltech) have several papers analyzing deepnets as deep compositions.

Mhaskar-Poggio-Liao'16

$$f(x_1, x_2, \dots, x_8) = g_3(g_{21}(g_{11}(x_1, x_2), g_{12}(x_3, x_4)), g_{22}(g_{11}(x_5, x_6), g_{12}(x_7, x_8)))$$



Theorem (informal statement)

Suppose that a function of d variables is hierarchically, locally, compositional . Both shallow and deep network can approximate f equally well. The number of parameters of the shallow network depends exponentially on d as $O(\varepsilon^{-d})$ with the dimension whereas for the deep network dance is $O(d\varepsilon^{-2})$

Next talks

- ▶ Tuesday:
 - ▶ **Ding-Xuan ZHOU** *Approximation Analysis of Distributed Learning and Deep CNNs*
- ▶ Wednesday:
 - ▶ **Philipp Grohs** *Approximation Results for Deep Neural Networks*
- ▶ Thursday:
 - ▶ **Gitta Kutyniok** *Optimal Approximation with Sparsely Connected Deep Neural Networks*
 - ▶ **Philipp Petersen** *Optimal Approximation of Classifier Functions by Deep ReLU Networks*

Modern Statistical Theory

Interactions with key themes in modern statistical theory

- ▶ Estimation of functions in high dimensions
- ▶ VC Classes/Generalization Bound
- ▶ Statistical Models
 - ▶ Projection Pursuit
 - ▶ Compositional Models
 - ▶ Generalized Additive Models
- ▶ Regularization/Overfitting/Model Selection
- ▶ Experimental Design
- ▶ Observational vs Experimental Data
- ▶ Ecological Correlation/Correlation vs Causation

Generalization of Supervised Learning

Collect data: $\{(x_i, y_i) | i = 1, 2, \dots, n\}$

Learn a model: $f: \mathcal{X} \rightarrow \mathcal{Y}, f \in \mathcal{H}$

Predict new data: $x \rightarrow f(x)$

All $(x, y) \sim \mathcal{D}$, where \mathcal{D} is unknown

A common approach to learn:
ERM (Empirical Risk Minimization)

$$\min R_n(w) := \frac{1}{n} \sum_i l(w; x_i, y_i)$$

w : model parameters

$l(w; x, y)$: loss function w.r.t. data

Population Risk: $R(w) := E[l(w; x, y)]$

Generalization Error

- We consider the standard ML setup:

$$\hat{E}(\Theta) = \mathbb{E}_{(X,Y) \sim \hat{P}} \ell(\Phi(X; \Theta), Y) + \mathcal{R}(\Theta)$$
$$E(\Theta) = \mathbb{E}_{(X,Y) \sim P} \ell(\Phi(X; \Theta), Y) .$$

$$\hat{P} = \frac{1}{n} \sum_{i \leq n} \delta_{(x_i, y_i)}$$

$\ell(z)$ convex

$\mathcal{R}(\Theta)$: regularization

- Population loss decomposition (aka "fundamental theorem of ML"):

$$E(\Theta^*) = \underbrace{\hat{E}(\Theta^*)}_{\text{training error}} + \underbrace{E(\Theta^*) - \hat{E}(\Theta^*)}_{\text{generalization gap}} .$$

- Long history of techniques to provably control generalization error via appropriate regularization.
- Generalization error and optimization are entangled [Bottou & Bousquet]

How to control generalization error?

- However, when $\Phi(\mathbf{X}; \Theta)$ is a large, deep network, current best mechanism to control generalization gap has two key ingredients:
 - Stochastic Optimization
 - ❖ “During training, it adds the sampling noise that corresponds to empirical-population mismatch” [Léon Bottou].
 - Make the model as *large* as possible.
 - ❖ see e.g. “Understanding Deep Learning Requires Rethinking Generalization”, [Ch. Zhang *et al*, ICLR’17].

Traditional Learning Theory

Common form of generalization bound (in expectation or high probability)

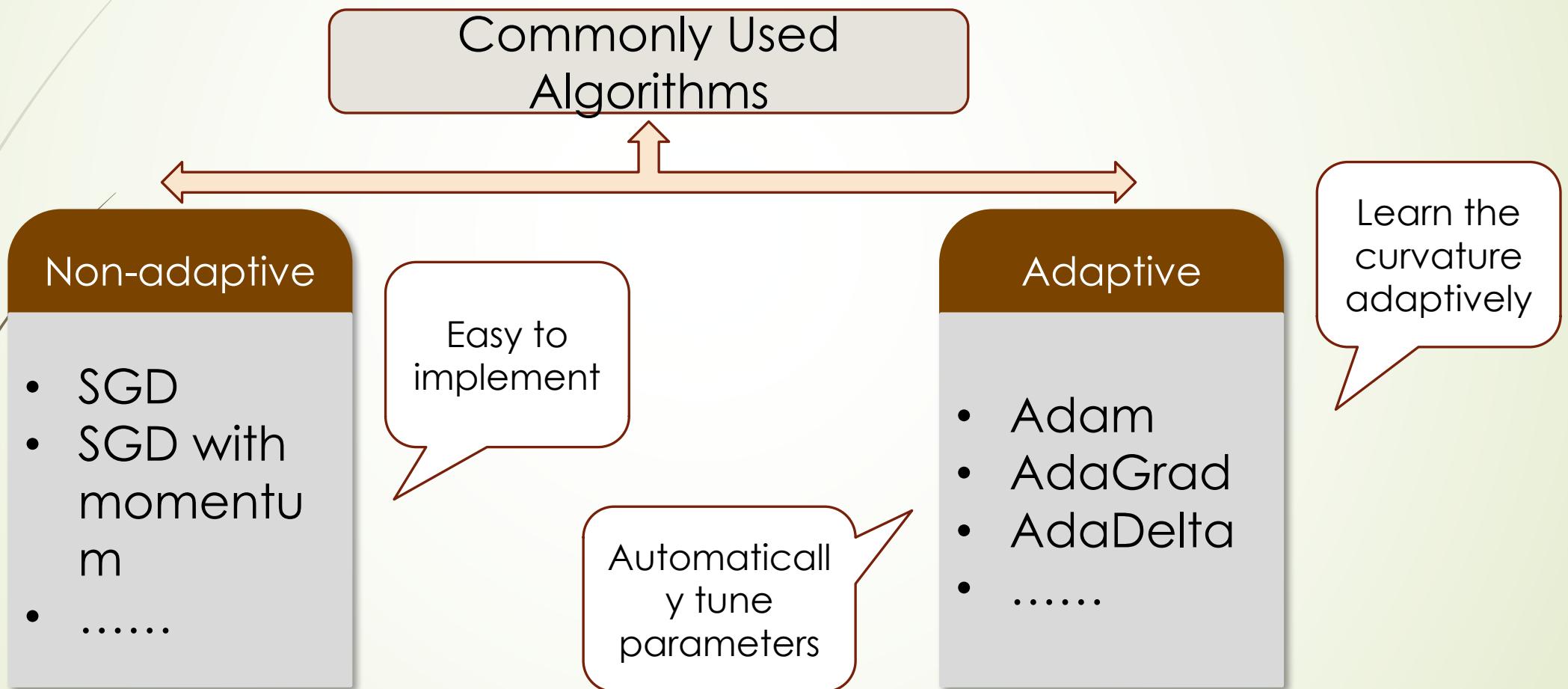
$$R(w) \leq R_n(w) + \sqrt{\frac{\text{Capacity Measure}}{n}}$$

Capacity Measurement	Complexity
VC-dimension	$VC \leq O(E \log E)$
ϵ -Covering number	$\log_2 N_{l_1}(\mathcal{F}, \epsilon, m) \leq O\left(\frac{(AL_\phi)^{L(L+1)}}{\epsilon^{2L}}\right)$
Rademacher Average	$R_m(\mathcal{F}) \leq O(\mu^L)$

$|E|$: # of edges
 L : # of layers

Big model should fail!

Training Algorithms for Deep Learning



Stochastic Gradient Descent

Objective loss function:

$$\min R_n(w) := \frac{1}{n} \sum_i l(w; x_i, y_i)$$

where (x_i, y_i) is the data, w is the parameter vector.

Gradient Descent: $w_{t+1} = w_t - \frac{\eta}{n} \sum \nabla l(w_t; x_i, y_i)$

$O(n)$ time complexity

SGD: $w_{t+1} = w_t - \eta \nabla l(w_t; x_{i_t}, y_{i_t})$, where i_t is uniform in $\{1, \dots, n\}$

$O(1)$ time complexity

Extensions: mini-batch SGD

Stochastic Gradient/Discrete Langevin Dynamics (SGLD)

SGLD is a variant of SGD:

$$w_{t+1} = w_t - \eta \nabla l(w_t; x_{i_t}, y_{i_t}) + \sqrt{\frac{2\eta}{\beta}} z_t, \text{ where } z_t \sim \mathcal{N}(0, I_d)$$

Injection of Gaussian noise makes SGLD completely different with SGD

For small enough step size η_t , Gaussian noise will dominate the stochastic gradient.

Distinctions of SGLD

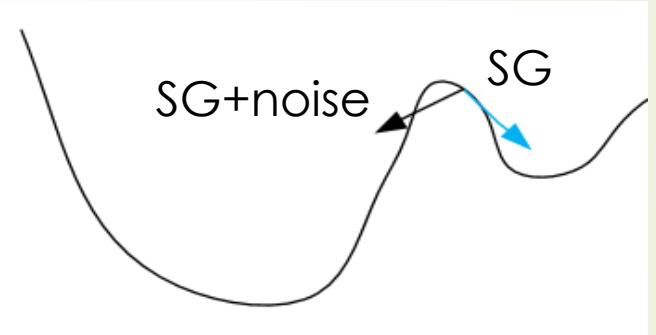
Intuitively, injected isotropic Gaussian noise helps escape saddle points or local minimum

SGLD is the discretization of following SDE

$$dW(t) = -\nabla F(W(t))dt + \sqrt{\frac{2}{\beta}}dB(t)$$

where $F(\cdot)$ is the empirical loss function, $B(t)$ is the standard Brownian motion

Its distribution converges to Gibbs distribution $\propto \exp(-\beta F(w))$



Large β will concentrate on the global minimizer of $F(w)$

From the view of stability theory:

Under mild conditions of (surrogate) loss function, the generalization error of SGLD at N -th round satisfies

$$E[l(w_S, z)] - E_S[l(w_S, z)] \leq O\left(\frac{1}{n} \left(k_0 + L \sqrt{\beta \sum_{k=k_0+1}^N \eta_k} \right)\right)$$

where L is the Lipschitz constant, and $k_0 := \min \{k : \eta_k \beta L^2 < 1\}$

If consider high probability form, there is an additional $\tilde{O}(\sqrt{1/n})$ term

Lipschitz Bound by Liwei Wang et al. 2017

From the view of PAC-Bayesian theory:

For regularized ERM with $R(w) = \lambda \|w\|^2/2$. Under mild conditions, with high probability, the generalization error of SGLD at N -th round satisfies

$$E[E[l(w_S, z)]] - E_S[E[l(w_S, z)]] \leq O\left(\sqrt{\frac{\beta}{n} \sum_{k=1}^N \eta_k e^{-\lambda(T_N-T_k)/2} E[\|g_k\|^2]}\right)$$

where $T_k = \sum_{j=1}^k \eta_j$, g_k is the stochastic gradient in each round.

Comparison Two Results

Both bounds suggest “train faster, generalize better”, which explain the random label experiments in ICLR17

- In expectation, stability bound has a faster $O\left(\frac{1}{n}\right)$ rate.
- PAC-Bayes bound is data dependent, and doesn’t rely on Lipschitz condition.
- Effect of step sizes in PAC-Bayes exponentially decay with time.

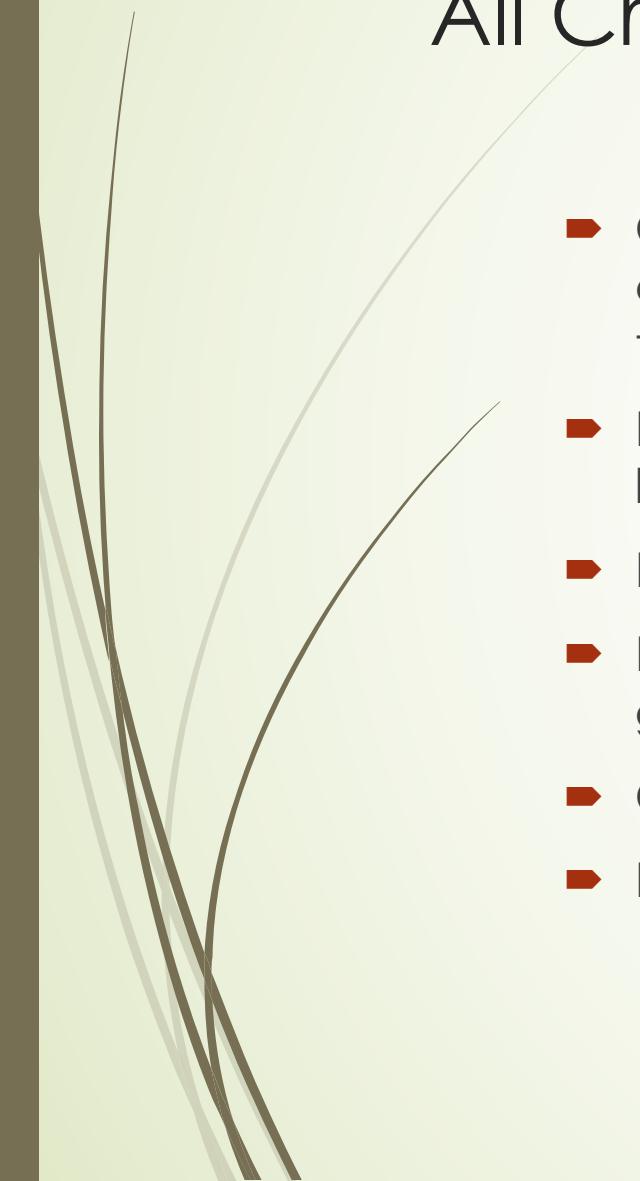
Next talks from Statistics

- ▶ Wednesday:
 - ▶ **Tony Cai** Statistical and Computational Limits for Submatrix Localization and Sparse Matrix Detection
- ▶ Friday:
 - ▶ **Xiaoming Huo** Nonconvex Penalization and Connections to Deep Learning

The Landscape of Risks

- However, when $\Phi(X; \Theta)$ is a large, deep network, current best mechanism to control generalization gap has two key ingredients:
 - Stochastic Optimization
 - ❖ “during training, it adds the sampling noise that corresponds to empirical-population mismatch” [Léon Bottou].
 - Make the model as *large as possible*.
 - ❖ see e.g. “Understanding Deep Learning Requires Rethinking Generalization”, [Ch. Zhang *et al*, ICLR’17].

- We first address how overparametrization affects the energy landscapes $E(\Theta), \hat{E}(\Theta)$.



A ‘Deep’ Dream: All Critical Point/local optima = Global Optima?

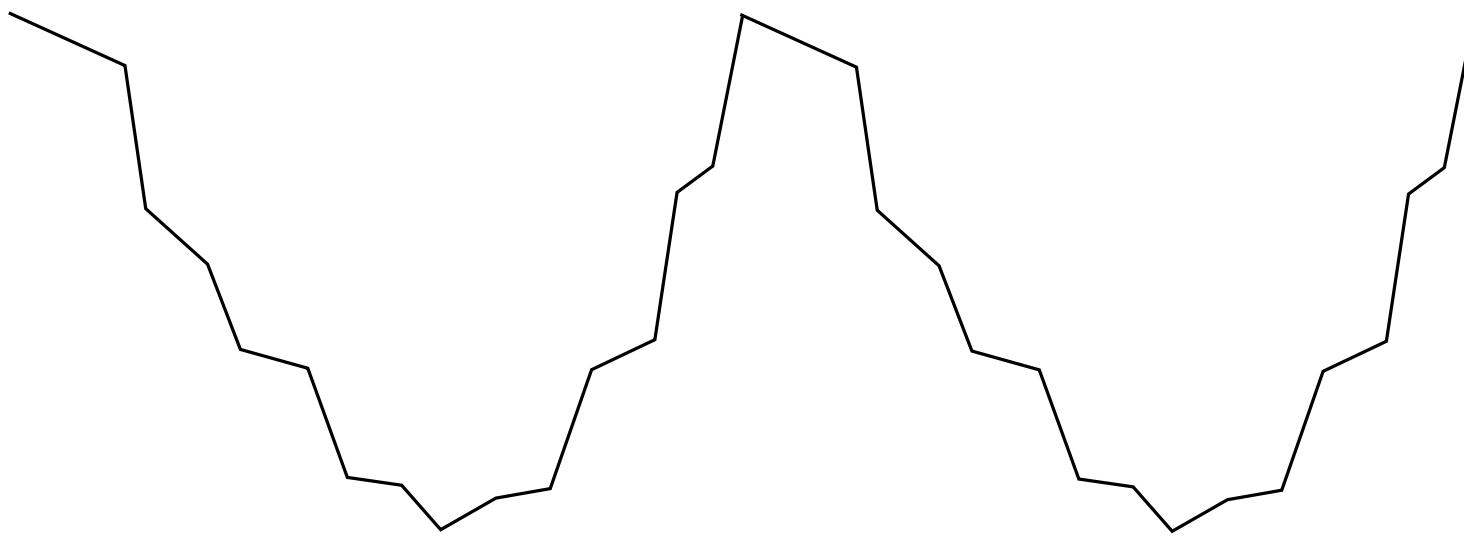
- ▶ Choromanska-LeCun-Ben Arous'15: most of critical values are concentrated in a narrow bind of global optima, using random Morse theory on sphere (spin class models)
- ▶ Haeffele et al.'15,16: overparameterized tensor factorization models, every local optima are global optima
- ▶ Kawaguchi'16: linear networks have no poor local optima
- ▶ Bruna et al.'16,17: simple sublevel set topology of multilinear regression, with group symmetry, and some nonlinear networks
- ▶ Chaudhari et al'17: Moreau envelope of empirical risk
- ▶ Pennington & Bahri'17: Hessian Analysis using Random Matrix Theory



A Dream: All Critical Point = Global Optima?

- Models from Statistical physics have been considered as possible approximations [Dauphin et al.'14, Choromanska et al.'15, Segun et al.'15]
- Tensor factorization models capture some of the non convexity essence [Anandukar et al'15, Cohen et al. '15, Haeffele et al.'15]
- [Shafran and Shamir,'15] studies bassins of attraction in neural networks in the overparametrized regime.
- [Soudry'16, Song et al'16] study Empirical Risk Minimization in two-layer ReLU networks, also in the over-parametrized regime.
- [Tian'17] studies learning dynamics in a gaussian generative setting.
- [Chaudhari et al'17]: Studies local smoothing of energy landscape using the local entropy method from statistical physics.
- [\[Pennington & Bahri'17\]: Hessian Analysis using Random Matrix Th.](#)
- [Soltanolkotabi, Javanmard & Lee'17]: layer-wise quadratic NNs.

Non-convex



$$F(\theta) = F(g.\theta), \quad g \in G \text{ compact.}$$

- We can perturb any convex function in such a way it is no longer convex, but such that gradient descent still converges.
- E.g. quasi-convex functions.
- In particular, deep models have internal symmetries.

Linear Networks

- Some authors have considered linear "deep" models as a first step towards understanding nonlinear deep models:

$$E(W_1, \dots, W_K) = \mathbb{E}_{(X,Y) \sim P} \|W_K \dots W_1 X - Y\|^2 .$$

$$X \in \mathbb{R}^n , \quad Y \in \mathbb{R}^m , \quad W_k \in \mathbb{R}^{n_k \times n_{k-1}} .$$

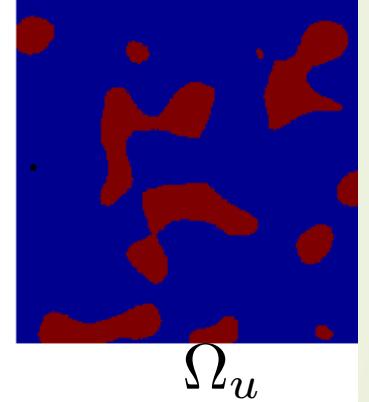
Theorem: [Kawaguchi'16] If $\Sigma = \mathbb{E}(XX^T)$ and $\mathbb{E}(XY^T)$ are full-rank and Σ has distinct eigenvalues, then $E(\Theta)$ has no poor local minima.

- studying critical points.
- later generalized in [Hardt & Ma'16, Lu & Kawaguchi'17]

Topology of Nonconvex Landscape

- Given loss $E(\theta)$, $\theta \in \mathbb{R}^d$, we consider its representation in terms of level sets:

$$E(\theta) = \int_0^\infty \mathbf{1}(\theta \in \Omega_u) du, \quad \Omega_u = \{y \in \mathbb{R}^d ; E(y) \leq u\}$$

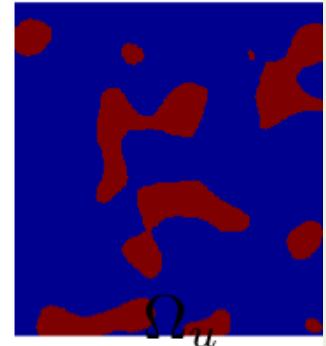


- A first notion we address is about the topology of the level sets .
- In particular, we ask how connected they are, i.e. how many connected components N_u at each energy level u ?

Simple Topology

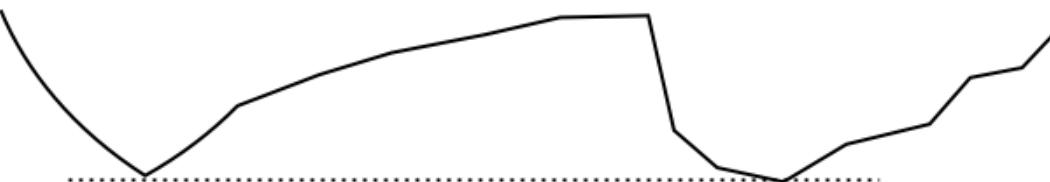
- A first notion we address is about the topology of the level sets .
 - In particular, we ask how connected they are, i.e. how many connected components N_u at each energy level u ?
- This is directly related to the question of global minima:

Proposition: If $N_u = 1$ for all u then E has no poor local minima.



(i.e. no local minima y^* s.t. $E(y^*) > \min_y E(y)$)

- We say E is *simple* in that case.
- The converse is clearly not true.



Simple Topology of Linear Networks

[Bruna-Freeman'16]

$$E(W_1, \dots, W_K) = \mathbb{E}_{(X,Y) \sim P} \|W_K \dots W_1 X - Y\|^2 .$$

Proposition: [BF'16]

1. If $n_k > \min(n, m)$, $0 < k < K$, then $N_u = 1$ for all u .
2. (2-layer case, ridge regression)
 $E(W_1, W_2) = \mathbb{E}_{(X,Y) \sim P} \|W_2 W_1 X - Y\|^2 + \lambda(\|W_1\|^2 + \|W_2\|^2)$
satisfies $N_u = 1 \forall u$ if $n_1 > \min(n, m)$.

- We pay extra redundancy price to get simple topology.

Group Symmetries [Bruna-Venturi-Bandiera'17]

- Q: How much extra redundancy are we paying to achieve $N_u = 1$ instead of simply no poor-local minima?
 - In the multilinear case, we don't need $n_k > \min(n, m)$
 - ❖ We do the same analysis in the quotient space defined by the equivalence relationship $W \sim \tilde{W} \Leftrightarrow W = \tilde{W}U$, $U \in GL(\mathbb{R}^n)$.

Corollary [LBB'17]: The Multilinear regression $\mathbb{E}_{(X,Y) \sim P} \|W_1 \dots W_k X - Y\|^2$ has no poor local minima.

- ❖ Construct paths on the Grassmannian manifold of subspaces.
- ❖ Generalizes best known results for multilinear case (no assumptions on data covariance).

Nonlinear ReLU network

- Good behavior is recovered with nonlinear ReLU networks, provided they are sufficiently overparametrized:
- Setup: two-layer ReLU network:
 $\Phi(X; \Theta) = W_2 \rho(W_1 X)$, $\rho(z) = \max(0, z)$. $W_1 \in \mathbb{R}^{m \times n}$, $W_2 \in \mathbb{R}^m$

Theorem [BF'16]: For any $\Theta^A, \Theta^B \in \mathbb{R}^{m \times n}, \mathbb{R}^m$, with $E(\Theta^{\{A,B\}}) \leq \lambda$, there exists path $\gamma(t)$ from Θ^A and Θ^B such that
 $\forall t, E(\gamma(t)) \leq \max(\lambda, \epsilon)$ and $\epsilon \sim m^{-\frac{1}{n}}$.

- Overparametrisation "wipes-out" local minima (and group symmetries).
- The bound is cursed by dimensionality, ie exponential in n .
- Open question: polynomial rate using Taylor decomp of $\rho(z)$?

Better Optimization Algorithms?

- ▶ Backpropagation Algorithm (made popular by Rumelhart-Hinton-Williams'1986) as stochastic gradient descent is equivalent to Lagrangian Multiplier method with gradient descent on weights (prox-linear)
 - ▶ Used in control theory (dynamic programming) in 1960s
 - ▶ **Qianxiao Li's** talk yesterday (PMP in continuous case)
- ▶ It suffers from **vanishing of gradients** due to the chain rule of gradient map

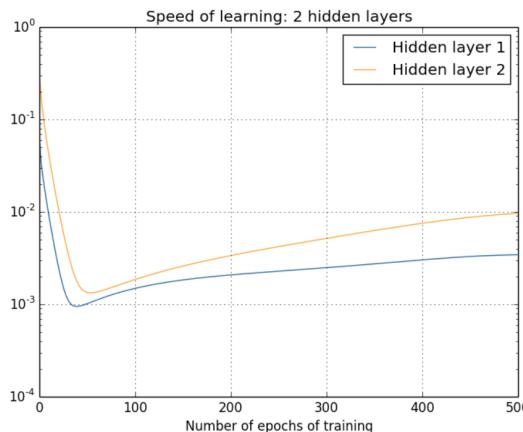
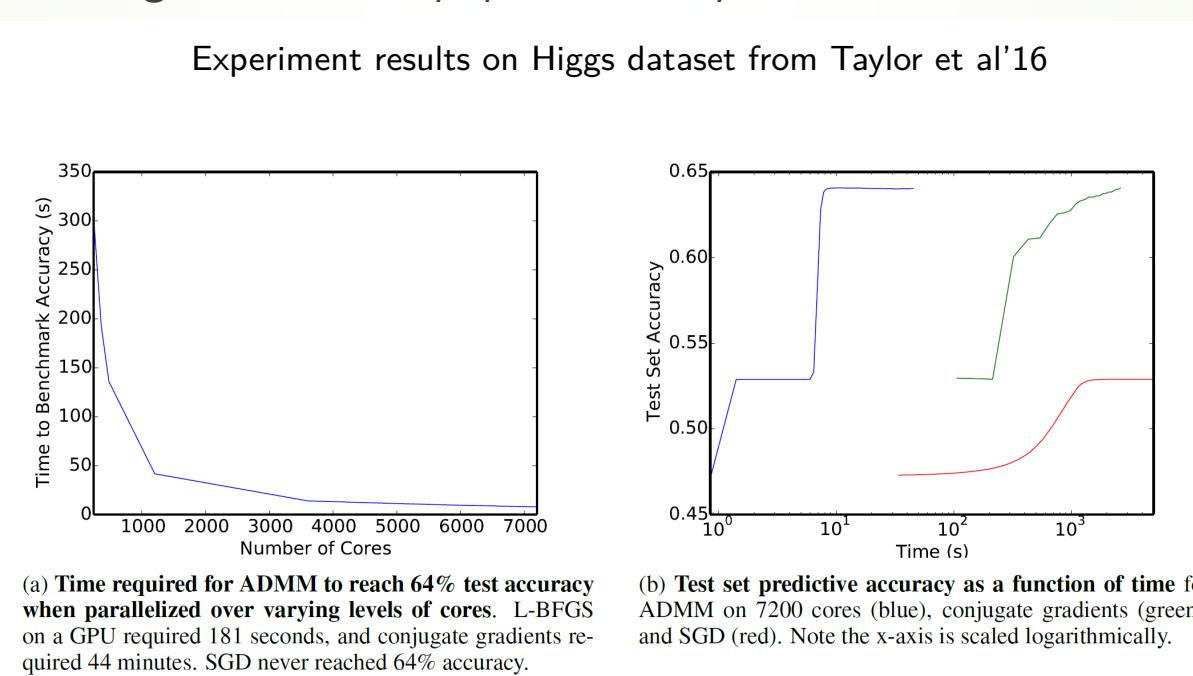


Figure: BP on MNIST[2]: Two hidden layers, speed of learning

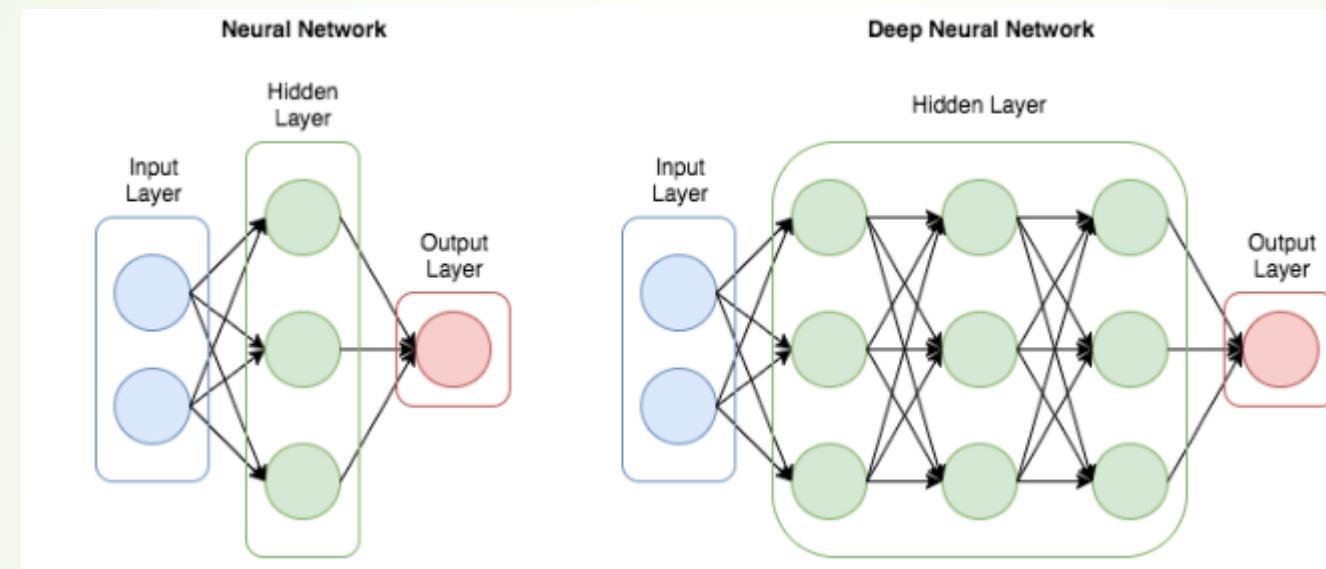
Alternative: (Augmented) Lagrangian Multiplier with Block Coordinate Descent

- ▶ ADMM-type: Taylor et al. ICML 2016
- ▶ BCD with zero Lagrangian multiplier: Zhang et al. NIPS 2017
- ▶ Discrete EMSA of PMP: **Qianxiao Li** et al 2017
 - ▶ No-vanshing gradients and parallelizable
- ▶ Some convergence theory: preliminary results on ADMM with **Jinshan Zeng'17**



ADMM for DNN Learning

► Deep Neural Network



- l layers (i.e., $l - 1$ hidden layers)
 - Input: d_1 neurons
 - Output: d_2 neurons
 - Hidden layer i : r_i neurons
 - Layer j : $W_j \in \mathbb{R}^{r_j \times r_{j-1}}$, $j = 1, \dots, l$
 - $r_0 := d_1$, $r_l := d_2$

ADMM for DNN Learning

- Deep Neural Network (DNN) Learning
 - Neural network model: (omit “bias” for simplicity)

$$f(x; \{W_j\}_{j=1}^l) = W_l \sigma(W_{l-1} \sigma(\cdots \sigma(W_1 x)))$$

σ : activation function (e.g., sigmoid, ReLU)

- DNN Learning: given n samples $(x_i, y_i)_{i=1}^n$

$$\min_{\{W_j\}_{j=1}^l} \underbrace{\frac{1}{n} \sum_{i=1}^n \ell(f(x_i; \{W_j\}_{j=1}^l), y_i)}_{\text{Empirical risk}} + \underbrace{\frac{\lambda}{2} \sum_{j=1}^l \|W_j\|_F^2}_{\text{Regularization}}$$

ℓ : loss function, e.g., ℓ_2 , logistic, hinge

ADMM for DNN Learning

► Solved via ADMM

$$\min_{\{W_j\}_{j=1}^l} \ell(f(X; \{W_j\}_{j=1}^l), Y) + \frac{\lambda}{2} \sum_{j=1}^l \|W_j\|_F^2$$

- Auxiliary variables: let $V_0 := X$,

$$V_i = \sigma(W_i V_{i-1}), \quad i = 1, \dots, l-1; \quad V_l = W_l V_{l-1}.$$

- DNN Learning \Rightarrow ADMM-ready formulation:

$$\min_{\{W_j, V_j\}_{j=1}^l} \ell(V_l, Y) + \frac{\lambda}{2} \sum_{j=1}^l \|W_j\|_F^2$$

$$\text{s.t. } V_i = \sigma(W_i V_{i-1}), \quad i = 1, \dots, l-1 \\ V_l = W_l V_{l-1}.$$

ADMM for DNN Learning

► Solved via ADMM (Augmented Lagrangian)

- Augmented Lagrangian Function

$$\begin{aligned}\mathcal{L}(\{W_i\}_{i=1}^l, \{V_i\}_{i=1}^l, \{\Lambda_i\}_{i=1}^l) \\ = \ell(f(X; \{W_j\}_{j=1}^l), Y) + \frac{\lambda}{2} \sum_{j=1}^l \|W_j\|_F^2 \\ + \sum_{i=1}^{l-1} \langle \Lambda_i, \sigma(W_i V_{i-1}) - V_i \rangle + \langle \Lambda_l, W_l V_{l-1} - V_l \rangle \\ + \sum_{i=1}^{l-1} \frac{\beta_i}{2} \|\sigma(W_i V_{i-1}) - V_i\|_F^2 + \frac{\beta_l}{2} \|W_l V_{l-1} - V_l\|_F^2\end{aligned}$$

ADMM for DNN Learning

- Solved via ADMM (KKT conditions)

KKT conditions:

$$0 = \lambda_1 W_1 + (\Lambda_1 \odot \sigma'(W_1 V_0)) V_0^T,$$

$$0 = \lambda_i W_i + (\Lambda_i \odot \sigma'(W_i V_{i-1})) V_{i-1}^T, \quad i = 2, \dots, l-1$$

$$0 = \lambda_l W_l + \Lambda_l V_{l-1}^T,$$

$$0 = -\Lambda_i + W_{i+1}^T (\Lambda_{i+1} \odot \sigma'(W_{i+1} V_i)), \quad i = 1, \dots, l-2$$

$$0 = -\Lambda_{l-1} + W_l^T \Lambda_l,$$

$$0 = -\Lambda_l + (V_l - Y),$$

$$0 = \sigma(W_i V_{i-1}) - V_i, \quad i = 1, \dots, l-1$$

$$0 = W_l V_{l-1} - V_l$$

ADMM for DNN Learning

- Solved via ADMM (Algorithm)

Algorithm 1 ADMM for DNN Learning

Samples: $Z := \{(x_i, y_i)\}_{i=1}^n$, $X := [x_1, \dots, x_n] \in \mathbb{R}^{d_1 \times n}$, $Y := [y_1, \dots, y_n] \in \mathbb{R}^{d_2 \times n}$, $V_0 := X$

Initialization: $\{W_i^0, V_i^0, \Lambda_i^0\}_{i=1}^l$

Parameters: $\lambda_i, \beta_i > 0, i = 1, \dots, l$

for $k=1, \dots$ **do**

(Backward Estimation)

for $i = l : -1 : 1$ **do**

$$W_i^k \leftarrow \arg \min_{W_i} \mathcal{L}(W_{<i}^{k-1}, W_i, W_{>i}^k, \{V_i^{k-1}\}_{i=1}^l, \{\Lambda_i^{k-1}\}_{i=1}^l)$$

end for

(Forward Correction)

for $j = 1 : l$ **do**

$$V_i^k \leftarrow \arg \min_{V_i} \mathcal{L}(\{W_i^k\}_{i=1}^l, V_{<i}^k, V_i, V_{>i}^{k-1}, \{\Lambda_i^{k-1}\}_{i=1}^l)$$

end for

(Updating Multipliers)

$$\Lambda_1^k \leftarrow \Lambda_1^{k-1} + \beta_1(\sigma(W_1^k V_0) - V_1^k)$$

$$\Lambda_i^k \leftarrow \Lambda_i^{k-1} + \beta_i(\sigma(W_i^k V_{i-1}^k) - V_i^k), \quad i = 2, \dots, l-1$$

$$\Lambda_l^k \leftarrow \Lambda_l^{k-1} + \beta_l(W_l^k V_{l-1}^k - V_l^k)$$

$$k \leftarrow k + 1$$

end for

ADMM for DNN Learning

► Convergence of ADMM

- ℓ_2 loss + sigmoidal activation function (i.e., $\sigma(u) = \frac{1}{1+e^{-u}}$)

Theorem (Convergence of ADMM for DNN Learning)

If the parameters $\{\beta_i\}_{i=1}^l$ are sufficiently large, then ADMM converges to a KKT point of DNN learning problem from any finite initial point.

ADMM for DNN Learning

- Convergence of ADMM (3-layers for example)

- $\mathcal{B} := \max \left\{ \frac{2\sqrt{r_1}}{\|X\|_F/\sqrt{n}}, \left(\frac{2\sqrt{r_2}}{\|X\|_F/\sqrt{n}} \right)^{1/2}, \left(\frac{\|Y\|_F}{\|X\|_F} \right)^{1/3} \right\}$

- Assumptions on β_i 's:

- (a) $\beta_3 \geq 2$,
- (b) $\beta_2 \geq 16(\beta_3 + 1)\mathcal{B}^2$,
- (c) $\beta_1 \geq 11(\beta_2 + 1)\mathcal{B}^4\|X\|_F$.

- Assumption on λ :

$$\lambda \geq \max \left\{ \frac{3}{4}\beta_1\mathcal{B}\|X\|_F^3, 10\beta_2\mathcal{B}^4\|X\|_F^3, 4\beta_3\mathcal{B}^4\|X\|_F^2 \right\}$$

Notes: regularization parameters may grow with number of layers in the worst case!

Thank you!

