The statistical physics of deep learning

On infant category learning, dynamic criticality, random landscapes, and the reversal of time.

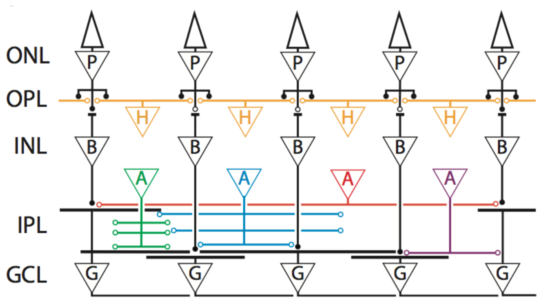
Surya Ganguli

Dept. of Applied Physics,
Neurobiology,
and Electrical Engineering

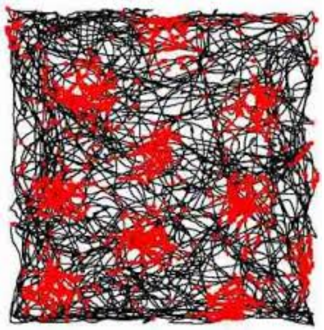
Stanford University

Neural circuits and behavior: theory, computation and experiment

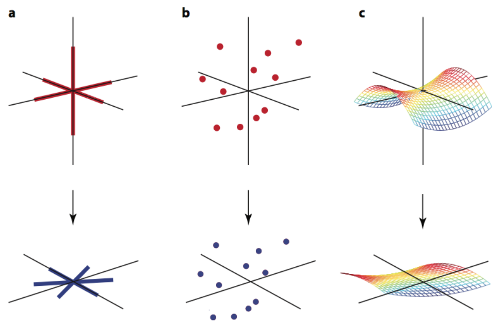
with Baccus lab: inferring hidden circuits in the retina



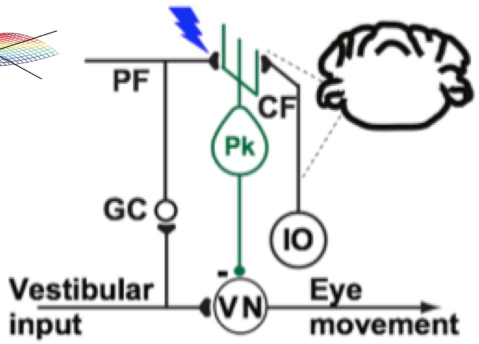
with Clandinin lab: unraveling the computations underlying fly motion vision from whole brain optical imaging



with the Giocomo lab: understanding the internal representations of space in the mouse entorhinal cortex



with the Shenoy lab: a theory of neural dimensionality, dynamics and measurement

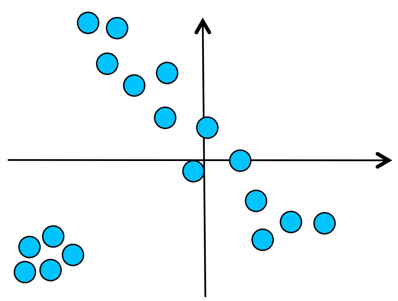
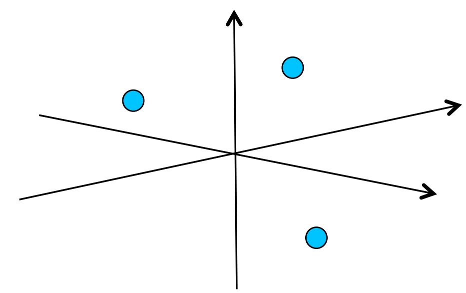


with the Raymond lab: theories of how enhanced plasticity can either enhance or impair learning depending on experience

Statistical mechanics of high dimensional data analysis

N = dimensionality of data
M = number of data points

$\alpha = N / M$

Classical Statistics	Modern Statistics
	
$N \sim O(1)$ $M \to \infty$ $\alpha \to 0$	$N \to \infty$ $M \to \infty$ $\alpha \sim O(1)$

Machine Learning and Data Analysis

Learn statistical parameters by maximizing log likelihood of data given parameters.

Statistical Physics of Quenched Disorder

Energy = - log Prob (data I parameters)

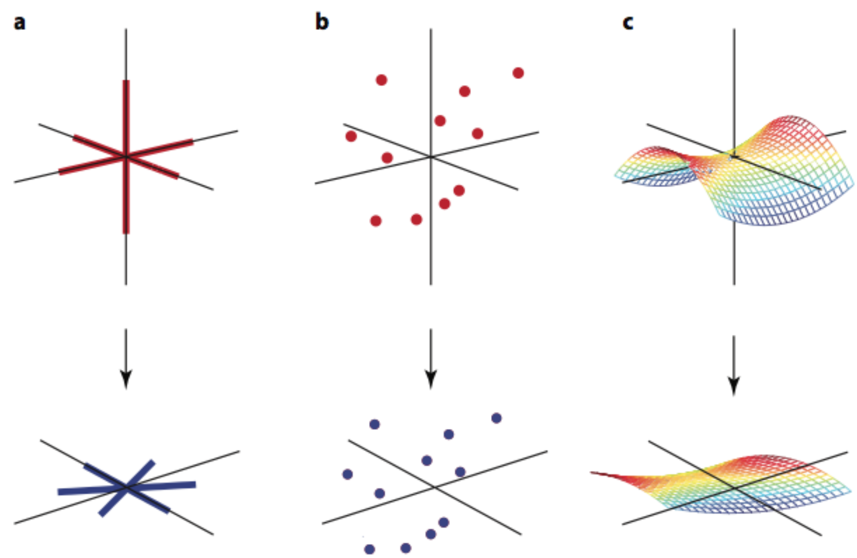
Data = quenched disorder
Parameters = thermal degrees of freedom

Applications to:
1) compressed sensing
2) optimal inference in high dimensions
3) a theory of neural dimensionality and measurement

Statistical mechanics of complex neural systems and high dimensional data

Madhu Advani, Subhaneil Lahiri and Surya Ganguli
Hide affiliations
msadvani@stanford.edu sulahiri@stanford.edu sganguli@stanford.edu
Department of Applied Physics, Stanford University, Stanford, CA, USA

Madhu Advani *et al J. Stat. Mech.* (2013) P03014. doi:10.1088/1742-5468/2013/03/P03014
Received 9 October 2012, accepted for publication 14 January 2013. Published 12 March 20



The project that really keeps me up at night

Motivations for an alliance between theoretical neuroscience and theoretical machine learning: opportunities for statistical physics

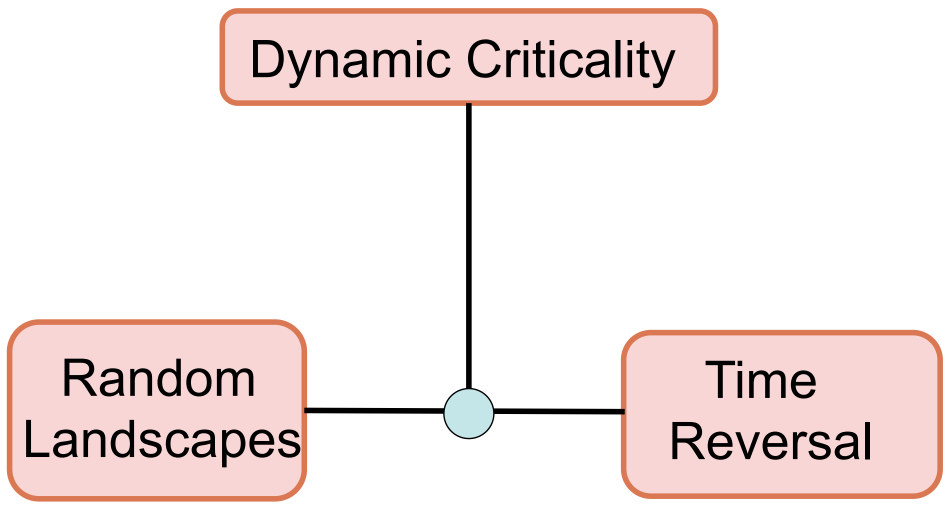
- What does it mean to understand the brain (or a neural circuit?)
- We understand how the connectivity and dynamics of a neural circuit gives rise to behavior.
- And also how neural activity and synaptic learning rules conspire to self-organize useful connectivity that subserves behavior.
- It is a good start, but it is not enough, to develop a theory of either random networks that have no function.
- The field of machine learning has generated a plethora of learned neural networks that accomplish interesting functions.
- We know their connectivity, dynamics, learning rule, and developmental experience, *yet*, we do not have a meaningful understanding of how they learn and work!

On simplicity and complexity in the brave new world of large scale neuroscience,
Gao and Ganguli Curr. Op. Neurobiology 2015

Talk Outline

Original motivation: understanding category learning in neural networks

We find random weight initializations, that make a network dynamically critical and allow rapid training of very deep networks.



Understand and exploit geometry of high dimensional error surfaces: need to escape saddle points not local minima.

Exploit violations of the second law of thermodynamics to create deep generative models

Acknowledgements and Funding

Saxe, J. McClelland, S. Ganguli, Learning hierarchical category structure in deep neural networks, Cog Sci. 2013.

Saxe, J. McClelland, S. Ganguli, Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, ICLR 2014.

J. Sohl-Dickstein, B. Poole, and S. Ganguli, Fast large scale optimization by unifying stochastic gradient and quasi-Newton methods, ICML 2014.

Identifying and attacking the saddle point problem in high dimensional non-convex optimization, Yann Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, Yoshua Bengio, NIPS 2014.

Deep unsupervised learning using non-equilibrium thermodynamics, J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, ICML 2015.

On simplicity and complexity in the brave new world of large-scale neuroscience, P. Gao and S. Ganguli, Current Opinion in Neurobiology, 2015.

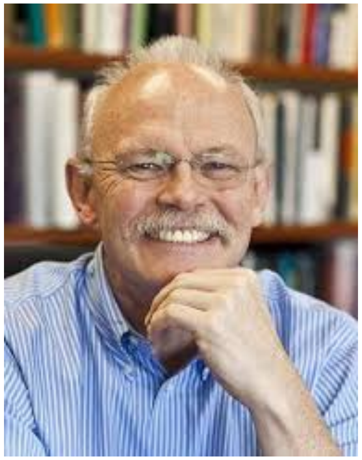
http://ganguli-gang.standford.edu

Funding:

Bio-X Neuroventures
Burroughs Wellcome
Genentech Foundation
James S. McDonnell Foundation
McKnight Foundation
National Science Foundation

Office of Naval Research
Simons Foundation
Sloan Foundation
Simons Foundation
Swartz Foundation
Stanford Terman Award

A Mathematical Theory of Semantic Development*



Joint work with: Andrew Saxe and Jay McClelland

*AKA: The misadventures of an "applied physicist" wondering around the psychology department

What is "semantic cognition"?

Human semantic cognition: Our ability to
learn, recognize, comprehend and produce inferences about properties of objects and events in the world, especially properties that are not present in the current perceptual stimulus

For example:

Does a cat have fur?
Do birds fly?

Our ability to do this likely relies on our ability to form internal representations of categories in the world

Psychophysical tasks that probe semantic cognition

Looking time studies: Can an infant distinguish between two categories of objects? At what age?

Property verification tasks: Can a canary move? Can it sing?
Response latency => central and peripheral properties

Category membership queries: Is a sparrow a bird? An ostrich?
Response latency => typical / atypical category members

Inductive generalization:

(A) Generalize familiar properties to novel objects:
i.e. a "blick" has feathers. Does it fly? Sing?

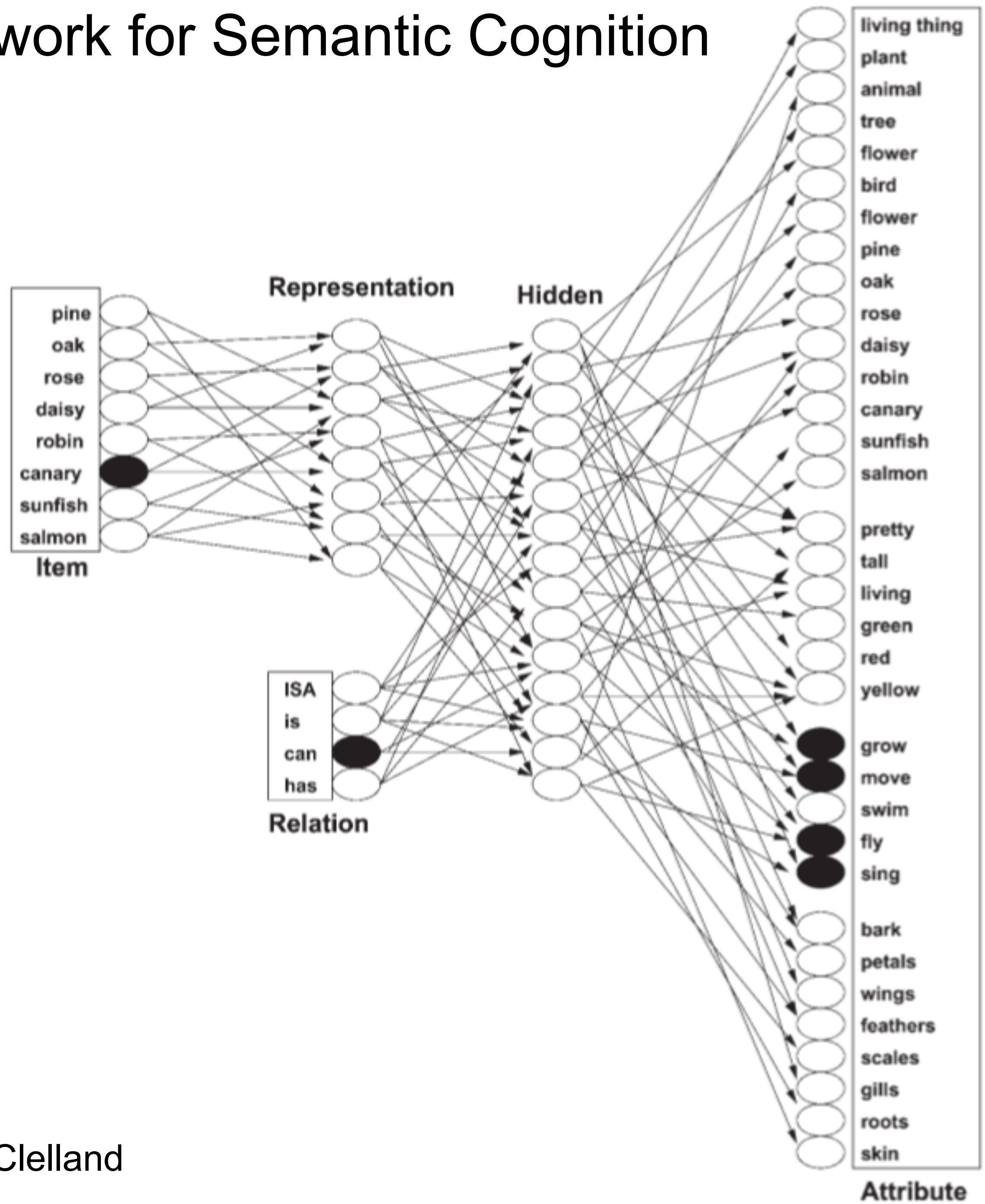
(B) Generalize novel properties to familiar objects:
i.e. a bird has gene "X". Does a crocodile have gene X?
Does a dog?

Semantic Cognition Phenomena

Table 1. *Six key phenomena in the study of semantic abilities*

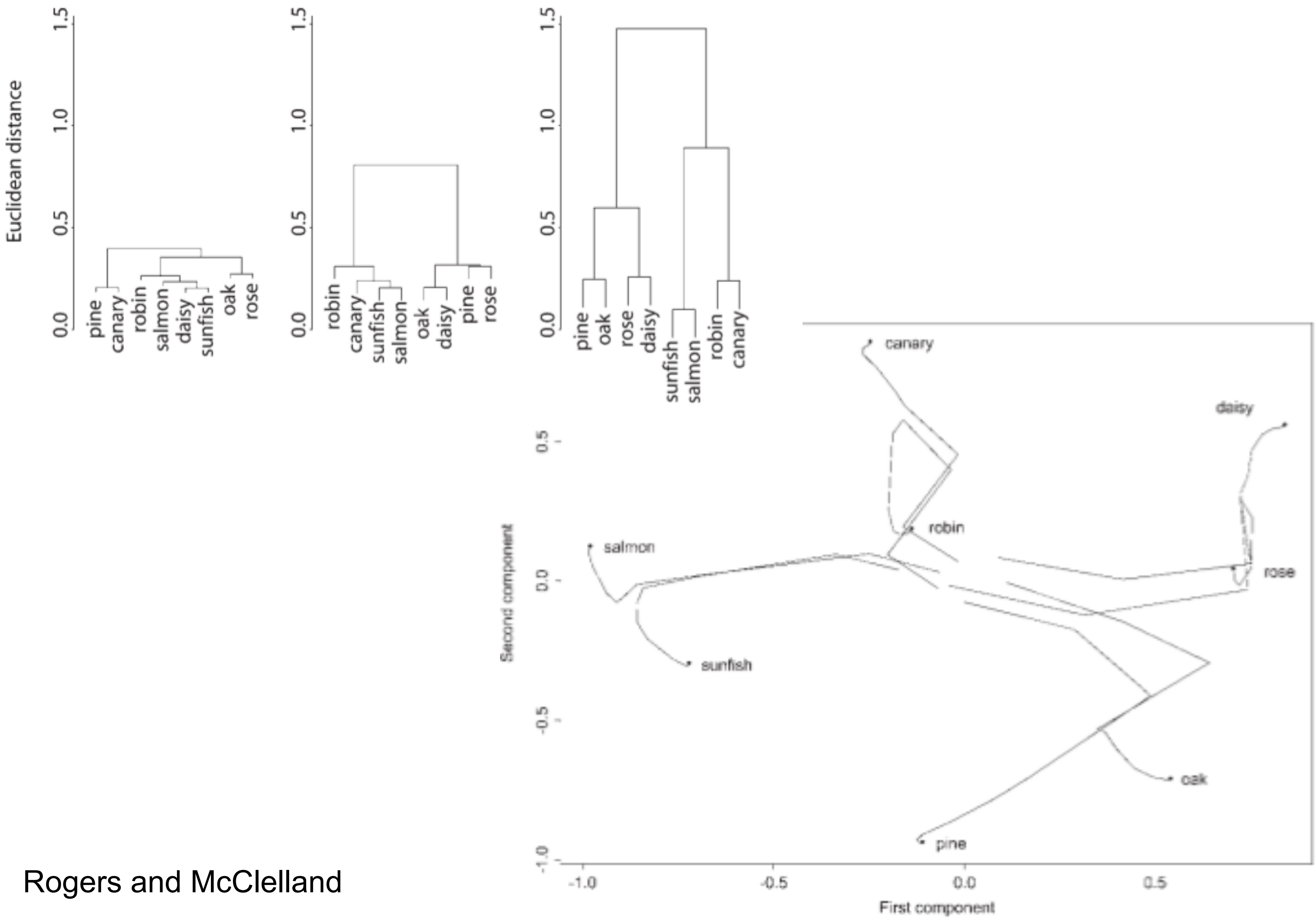
Phenomenon	Example
Progressive differentiation of concepts	Children acquire broader semantic distinctions earlier than more fine-grained distinctions. For example, when perceptual similarity among items is controlled, infants differentiate animals from furniture around 7–9 months of age, but do not make finer-grained distinctions (e.g., between fish and birds or chairs and tables) until somewhat later (Pauen 2002a; Mandler et al. 1991); and a similar pattern of coarse-to-fine conceptual differentiation can be observed between the ages of 4 and 10 in verbal assessments of knowledge about which predicates can appropriately apply to which nouns (Keil 1989).
Category coherence	Some groupings of objects (e.g., "the set of all things that are dogs") seem to provide a useful basis for naming and inductive generalization, whereas other groupings (e.g., "the set of all things that are blue") do not. How does the semantic system "know" which groupings of objects should be used for purposes of naming and inductive generalization, and which should not?
Domain-specific attribute weighting	Some properties seem of central importance to a given concept, whereas others do not. For instance, "being cold inside" seems important to the concept *refrigerator*, whereas "being white" does not. Furthermore, properties that are central to some concepts may be unimportant for others – although having a white color may seem unimportant for *refrigerator*, it seems more critical to the concept *polar bear*. What are the mechanisms that support domain-specific attribute weighting?
Illusory correlations	Children and adults sometimes attest to beliefs that directly contradict their own experience. For example, when shown a photograph of a kiwi bird – a furry-looking animal with eyes but no discernible feet – children may assert that the animal can move "because it has feet," even while explicitly stating that they can see no feet in the photograph. Such illusory correlations appear to indicate some organizing force behind children's inferences that goes beyond "mere" associative learning. What mechanisms promote illusory correlations?
Conceptual reorganization	Children's inductive projection of biological facts to various different plants and animals changes dramatically between the ages of 4 and 10. For some researchers, these changing patterns of induction indicate changes to the implicit theories that children bring to bear on explaining biological facts. What mechanism gives rise to changing induction profiles over development?
The importance of causal knowledge	A variety of evidence now indicates that, in various kinds of semantic induction tasks, children and adults strongly weight causally central properties over other salient but non-causal properties. Why are people sensitive to causal properties?

A Network for Semantic Cognition



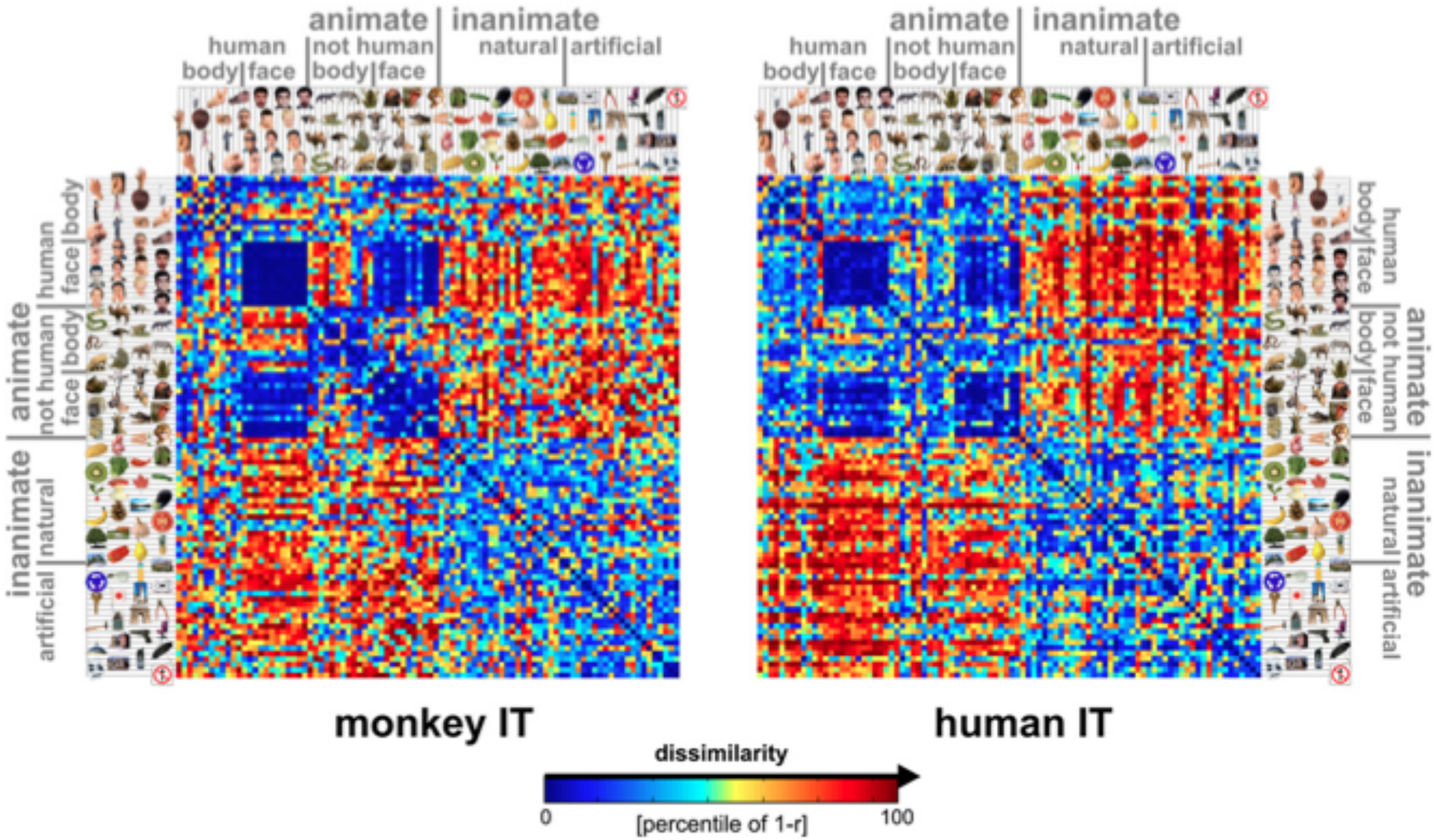
Rogers and McClelland

Evolution of internal representations



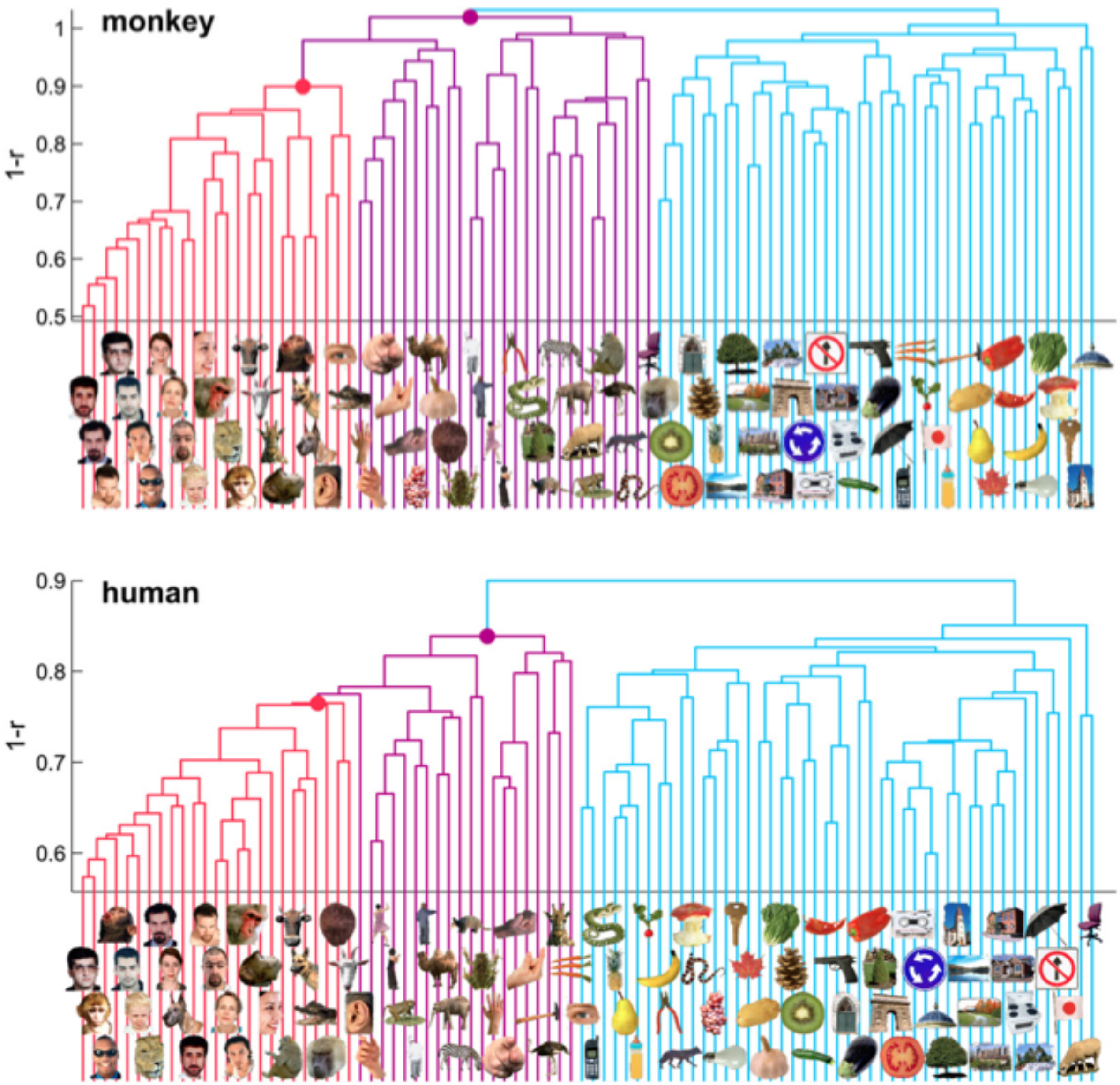
Rogers and McClelland

Categorical representations in human and monkey



Kriegeskorte et. al. Neuron 2008

Categorical representations in human and monkey



Kriegeskorte et. al. Neuron 2008

Evolution of internal representations

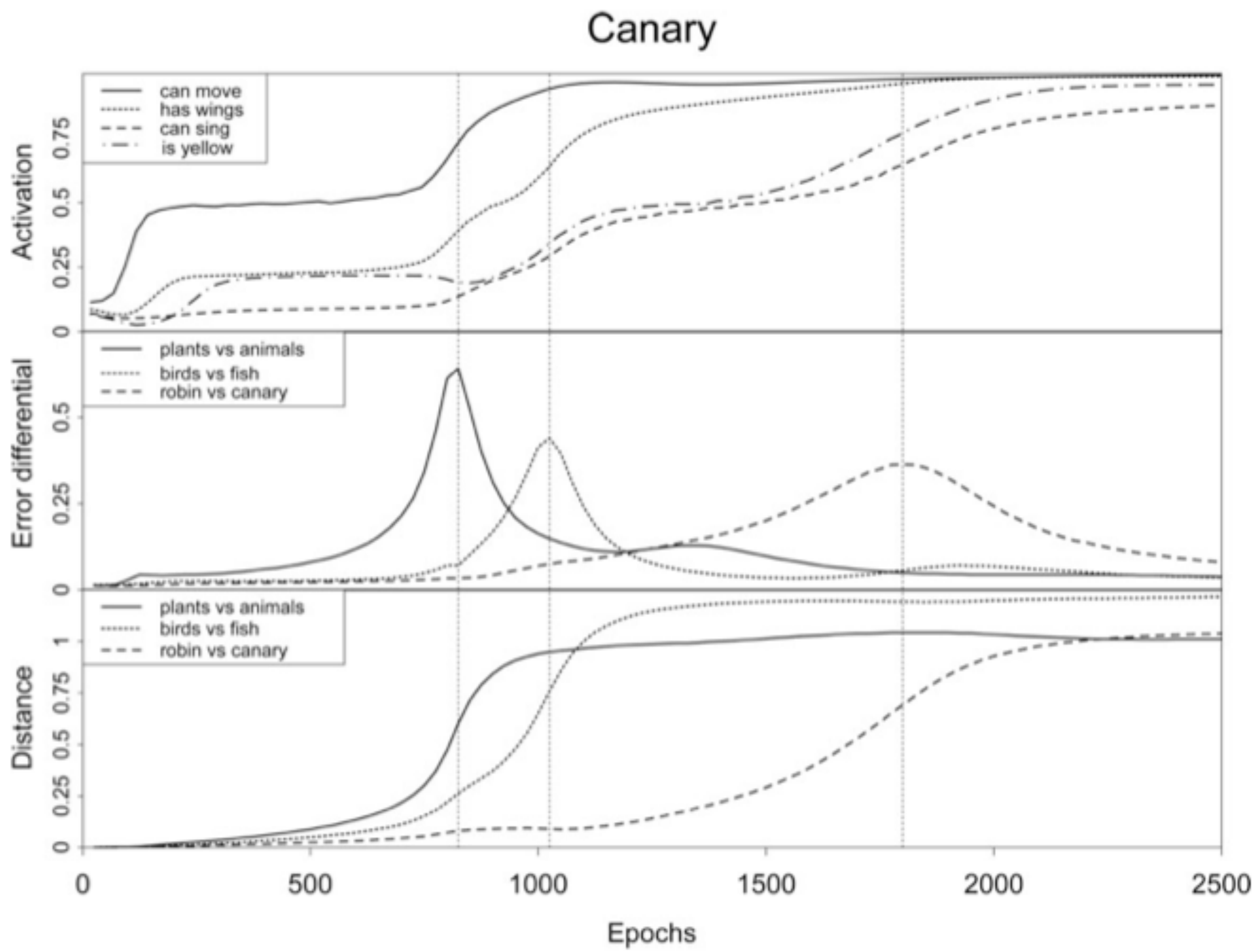


Figure 5. **Bottom**: Mean Euclidean distance between plants and animals, birds and fish, and canary and robin internal representations throughout training. **Middle**: Average magnitude of the error signal propagating back from properties that reliably discriminate plants from animals, birds from fish, or the canary from the robin, at different points throughout training when the model is presented with the canary as input. **Top**: Activation of a property shared by animals (*can move*) or birds (*can fly*), or unique to the canary (*can sing*), when the model is presented with the input canary can at different points throughout training.

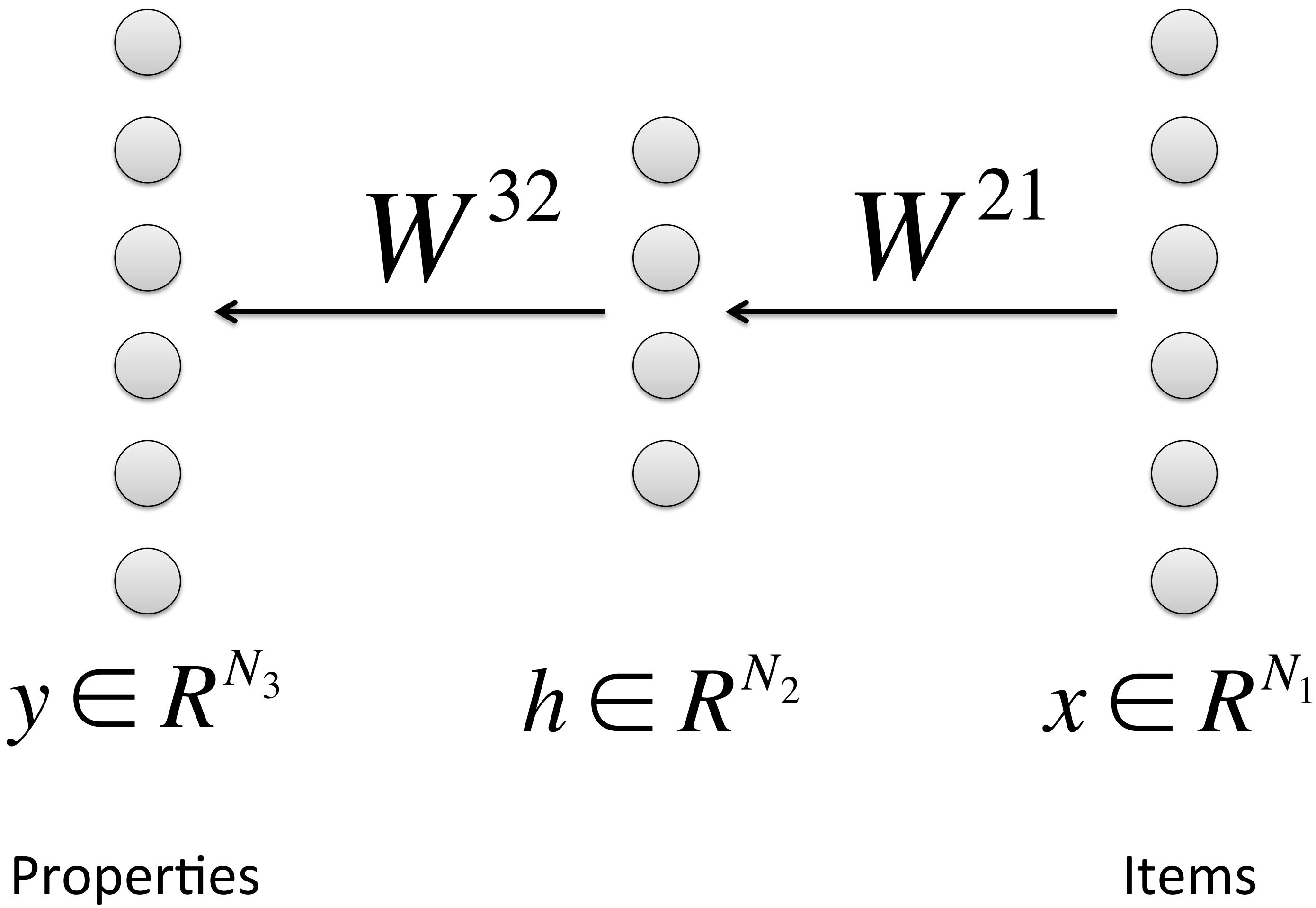
Rogers and McClelland

Theoretical questions

- What are the mathematical principles underlying the hierarchical self-organization of internal representations in the network?
- What are the relative roles of:
  - nonlinear input-output response
  - learning rule
  - input statistics (second order? higher order?)
- What is a mathematical definition of category coherence, and How does it relate the speed of category learning?
- Why are some properties learned more quickly than others?
- How can we explain changing patterns of inductive generalization over developmental time scales?

Problem formulation

We analyze a fully linear three layer network $y = W^{32}W^{21}x$



Learning dynamics

- Network is trained on a set of items and their properties

$$\{x^{\mu}, y^{\mu}\}, \mu = 1, \ldots, P.$$

- Weights adjusted using standard backpropagation:
  - Change each weight to reduce the error between desired network output and current network output

$$\begin{aligned} \Delta W^{21} &= \lambda W^{32^T} \left(y^{\mu} - \hat{y}^{\mu}\right) x^{\mu T} \\ \Delta W^{32} &= \lambda \left(y^{\mu} - \hat{y}^{\mu}\right) h^{\mu T} \end{aligned}$$

- Highlights the error-corrective aspect of this learning process

Learning dynamics

In linear networks, there is an equivalent formulation that highlights the role of the statistics of the training environment:

Input correlations: $\Sigma^{11} \equiv E\left[xx^T\right]$

Input-output correlations: $\Sigma^{31} \equiv E\left[yx^T\right]$

Equivalent dynamics:

$$
\begin{aligned}
\tau \frac{d}{dt} W^{21} &= W^{32^T} \left(\Sigma^{31} - W^{32} W^{21} \Sigma^{11}\right) \\
\tau \frac{d}{dt} W^{32} &= \left(\Sigma^{31} - W^{32} W^{21} \Sigma^{11}\right) W^{21^T}
\end{aligned}
$$

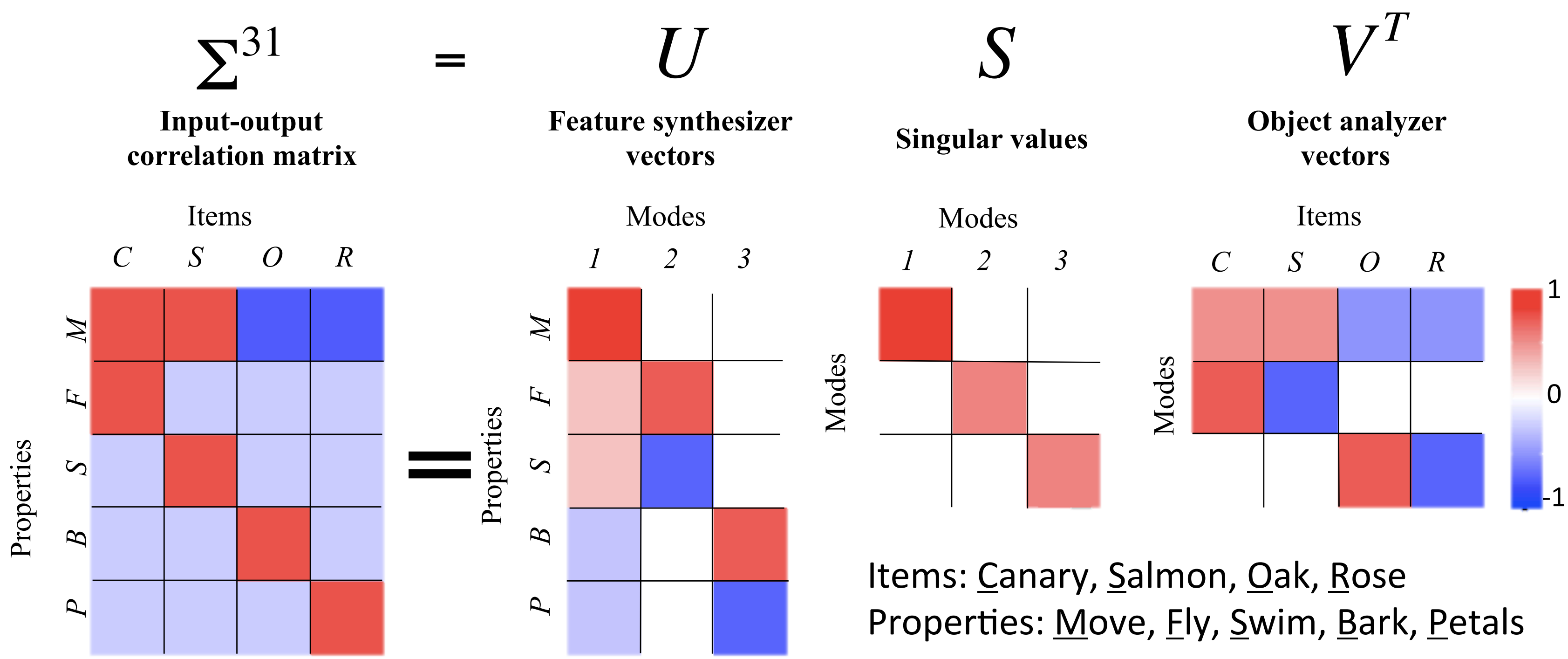
- Learning driven only by correlations in the training data
- Equations coupled and nonlinear

Decomposing input-output correlations

The learning dynamics can be expressed using the SVD of Σ^{31}

$$\Sigma^{31} = U^{33} S^{31} V^{11^T} = \sum_{\alpha=1}^{N_1} s_\alpha u^\alpha v^{\alpha T}$$

Mode *α* links a set of coherently covarying properties u^α to a set of coherently covarying items $v^{\alpha T}$ with strength s_α



Items: Canary, Salmon, Oak, Rose
Properties: Move, Fly, Swim, Bark, Petals

Analytical learning trajectory

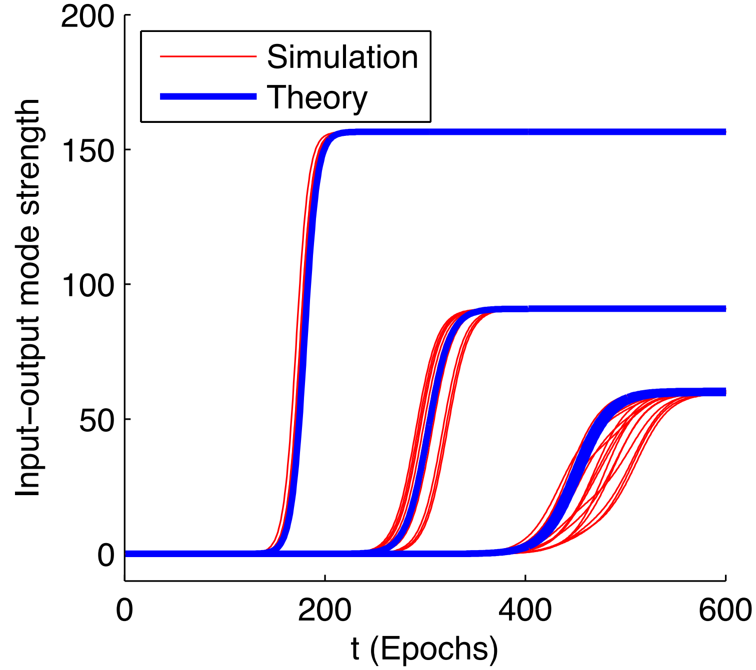
The network's input-output map is exactly

$$W^{32}(t)W^{21}(t) = \sum_{\alpha=1}^{N_2} a(t, s_\alpha, a_\alpha^0)\, u^\alpha v^{\alpha T}$$

where $$a(t, s, a_0) = \frac{s e^{2st/\tau}}{e^{2st/\tau} - 1 + s/a_0}$$

for a special class of initial conditions and $\Sigma^{11} = I$.

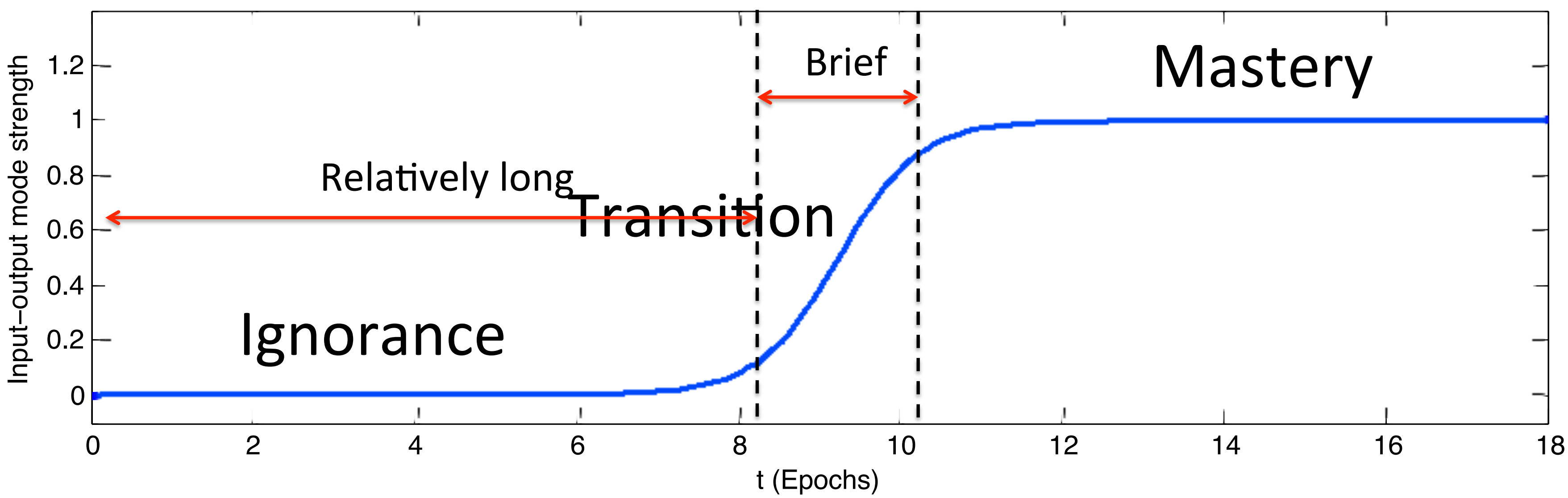
- Each mode evolves independently
- Each mode is **learned in time** $O(\tau/s)$



Stage-like transitions

Empirical evidence suggests transitions during learning can be rapid and stage-like

- Our model exhibits such transitions
- Intuitively, arises from sigmoidal learning trajectories
- The ratio of the *transition period* to the *ignorance period* can be arbitrarily small



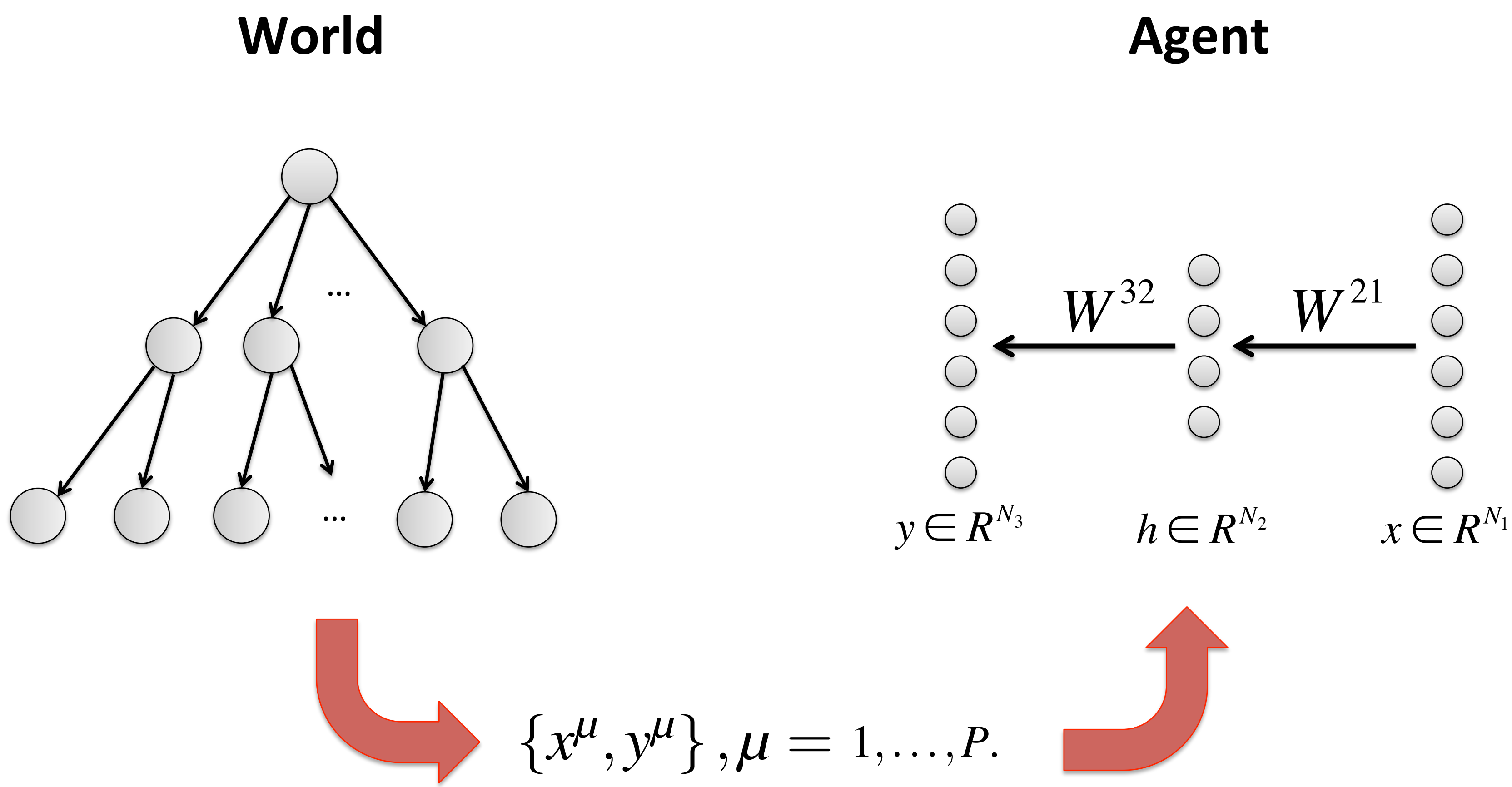
Take home messages so far

- The network learns different modes of covariation between input and output on a time scale inversely proportional to the statistical strength of that covariation.
- The learning curve for an input output mode can be sigmoidal with little evidence of learning for a long time, then a sudden transition to being learned.
- NEXT: What does this have to do with hierarchical differentiation of concepts? To answer this we must understand the second order statistics of hierarchically structured data.

Learning hierarchical structure

- The preceding analysis describes dynamics in response to a **specific** dataset
- Can we move beyond specific datasets to **general** principles when a neural network is exposed to hierarchical structure?
- We consider training a neural network with data generated by a **hierarchical generative model**

Connecting hierarchical generative models and neural network learning

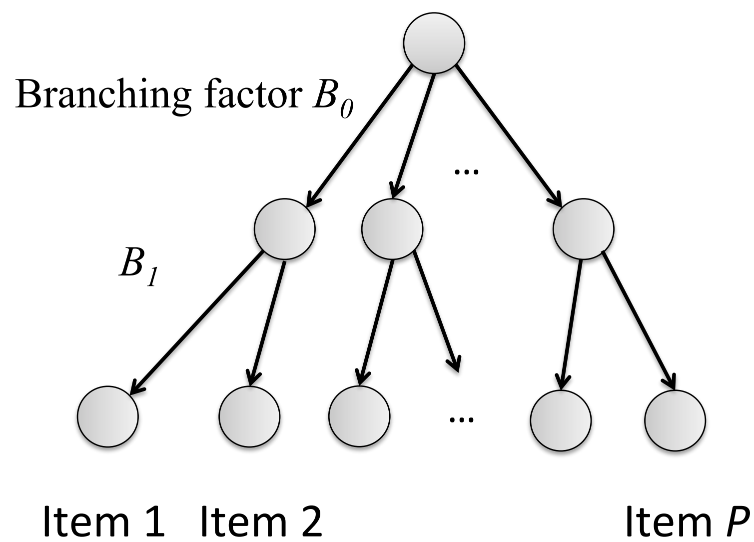


A hierarchical branching diffusion process

Generative model defined by a tree of nested categories

Feature values diffuse down tree with small probability ε of changing along each link

Sampled independently N times to produce N features

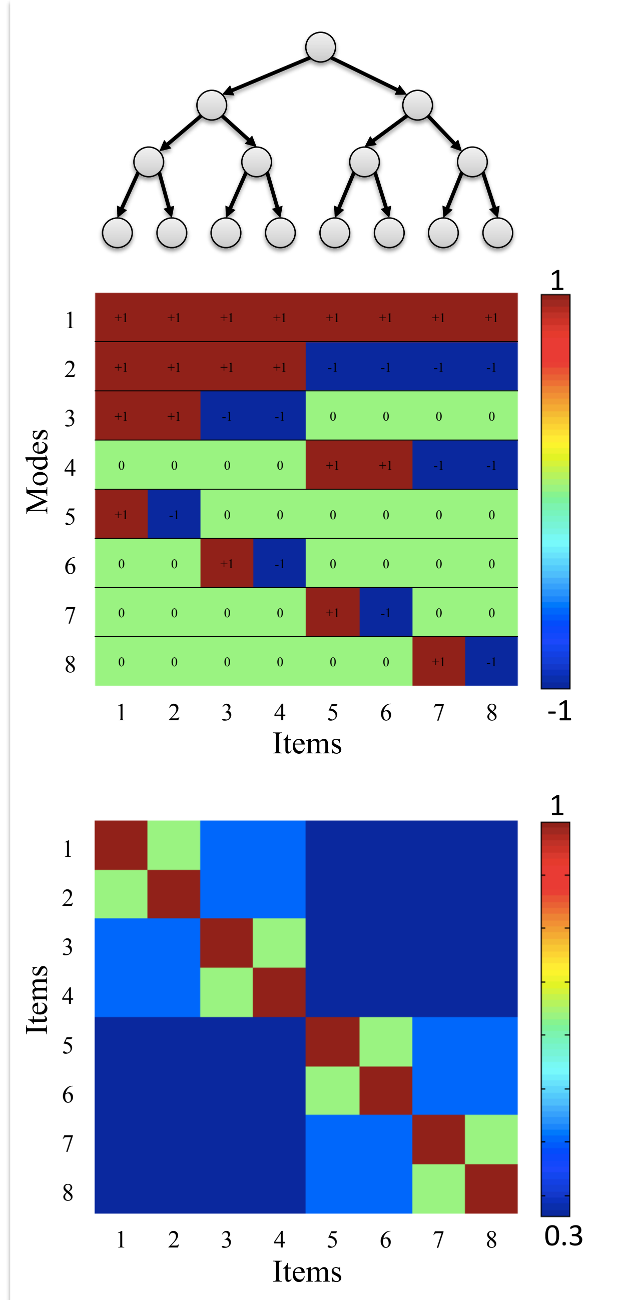


Object analyzer vectors

Assume our network is trained on an infinite amount of data drawn from this model

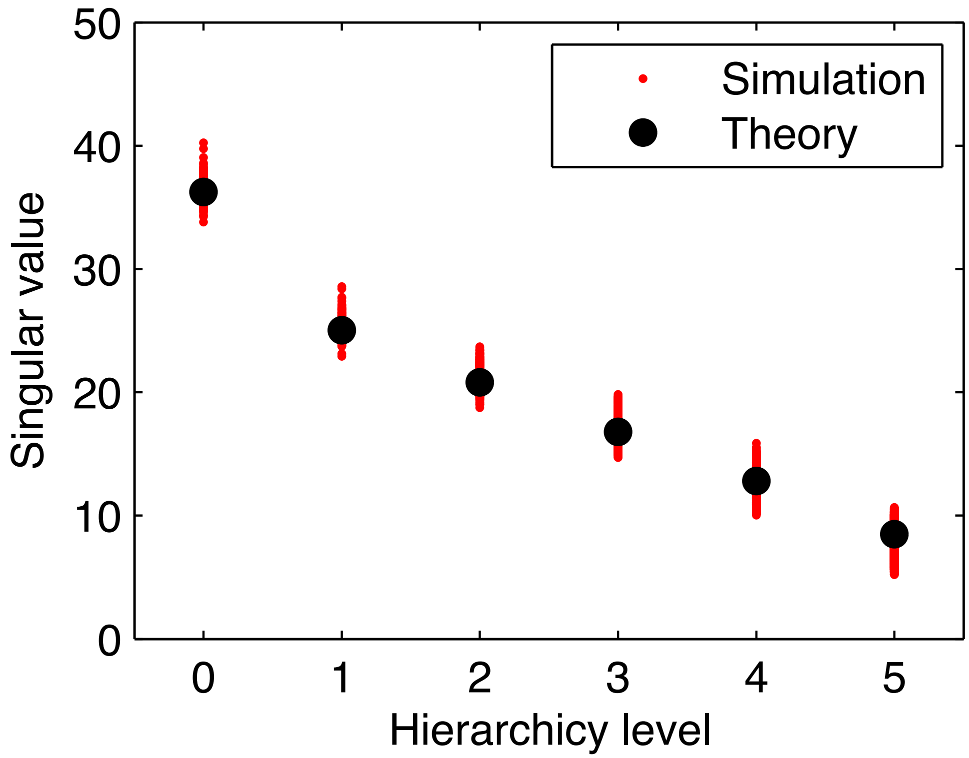
Can analytically compute SVD of the input-output correlation matrix:

The object analyzer vectors **mirror the tree structure**



Singular values

The singular values are **a *decreasing* function** of the hierarchy level.



Progressive differentiation

Hence the network **must** exhibit progressive differentiation on **any** dataset generated by this class of hierarchical diffusion processes:

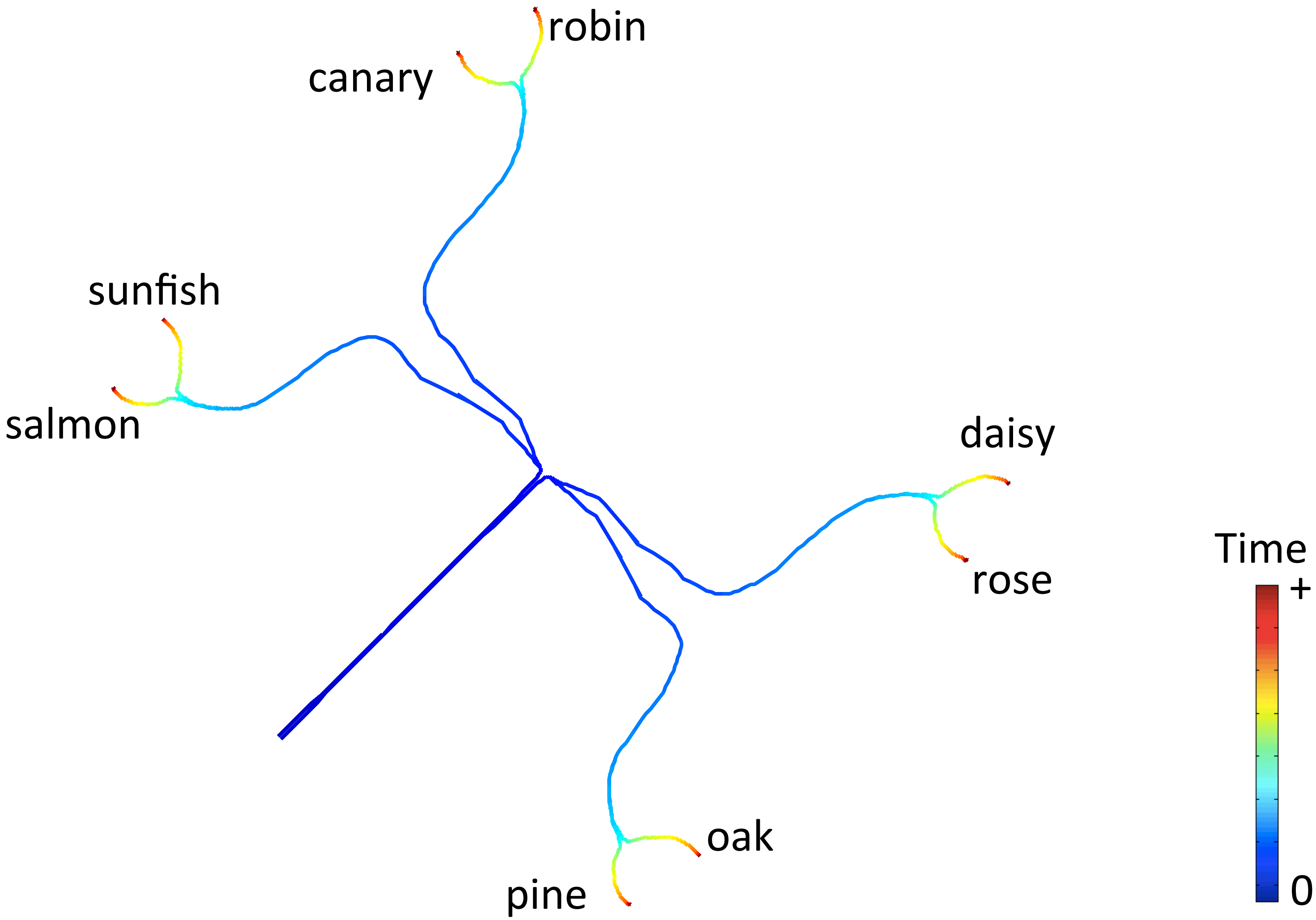
- Network learns input-output modes in time

$$O(\tau/s)$$

- Singular values of broader hierarchical distinctions are larger than those of finer distinctions

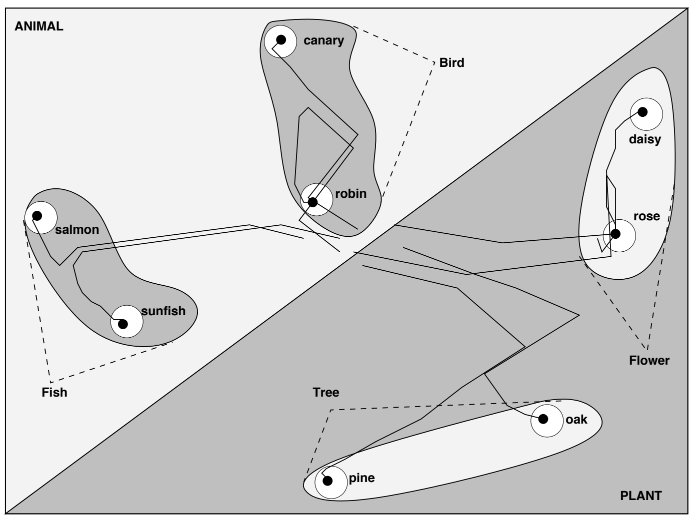
- Input-output modes correspond exactly to the hierarchical distinctions in the underlying tree

Progressive differentiation



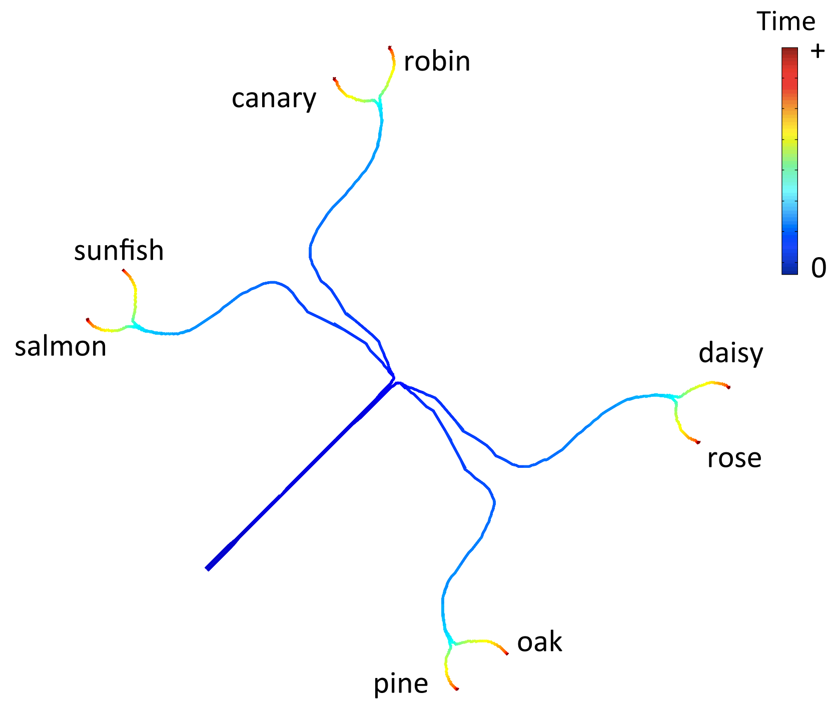
Progressive differentiation

Simulation



Rogers & McClelland, 2004

Analytics



Conclusion

- **Progressive differentiation of hierarchical structure** is a general feature of learning in deep neural networks

- Deep (but not shallow) networks exhibit **stage-like transitions** during learning

- Second order statistics of data are sufficient to drive hierarchical differentiation

Ongoing work

In a position to analytically understand many phenomena previously simulated

- Illusory correlations early in learning
- Familiarity and typicality effects
- Inductive property judgments
- 'Distinctive' feature effects
- Basic level effects
- Category coherence
- Perceptual correlations
- Practice effects

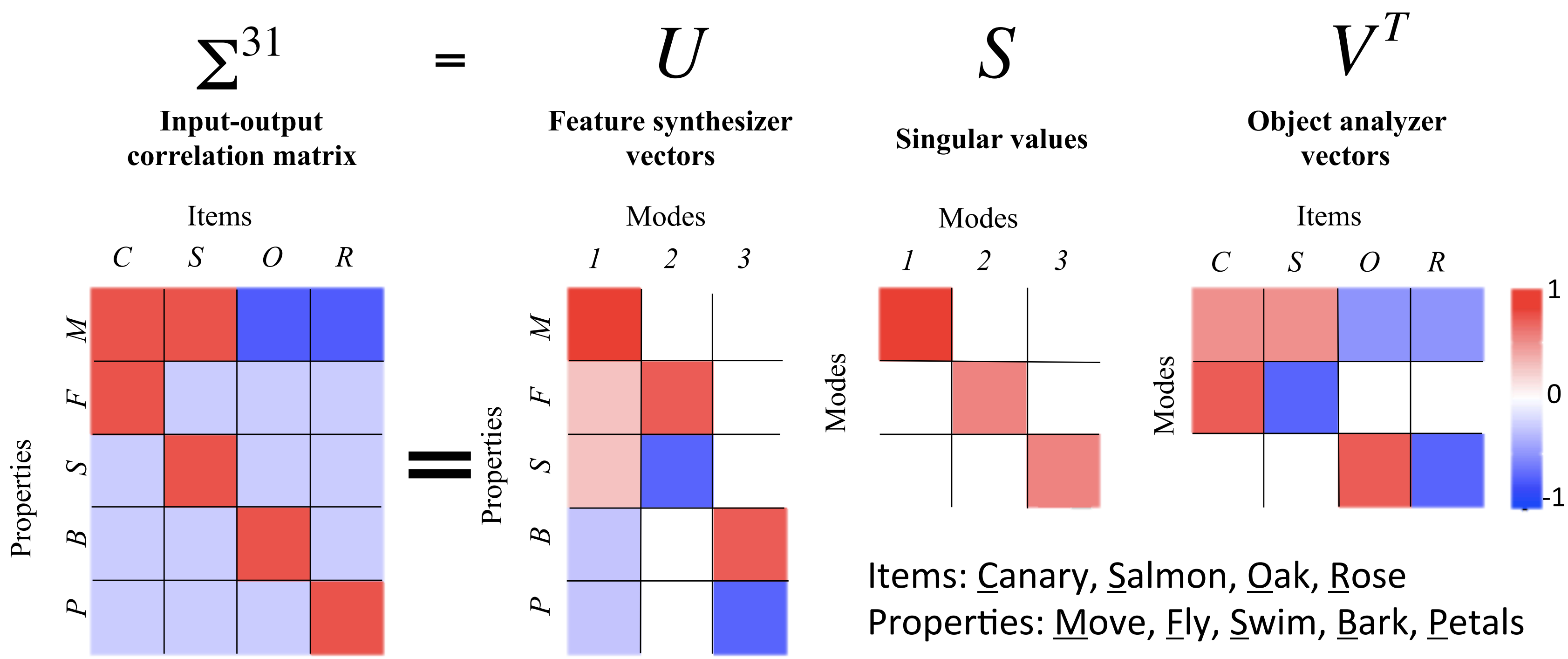
Our framework **connects probabilistic models** and **neural networks**, analytically linking structured environments to learning dynamics.

Why are some properties distinctive, or learned faster?

A property = vector across items
An object analyzer = vector across items

If a property is similar to an object analyzer with large singular value then (and only then) will it be learned quickly.

That property is distinctive for the category associated with that object analyzer (i.e. move for animals versus plant)

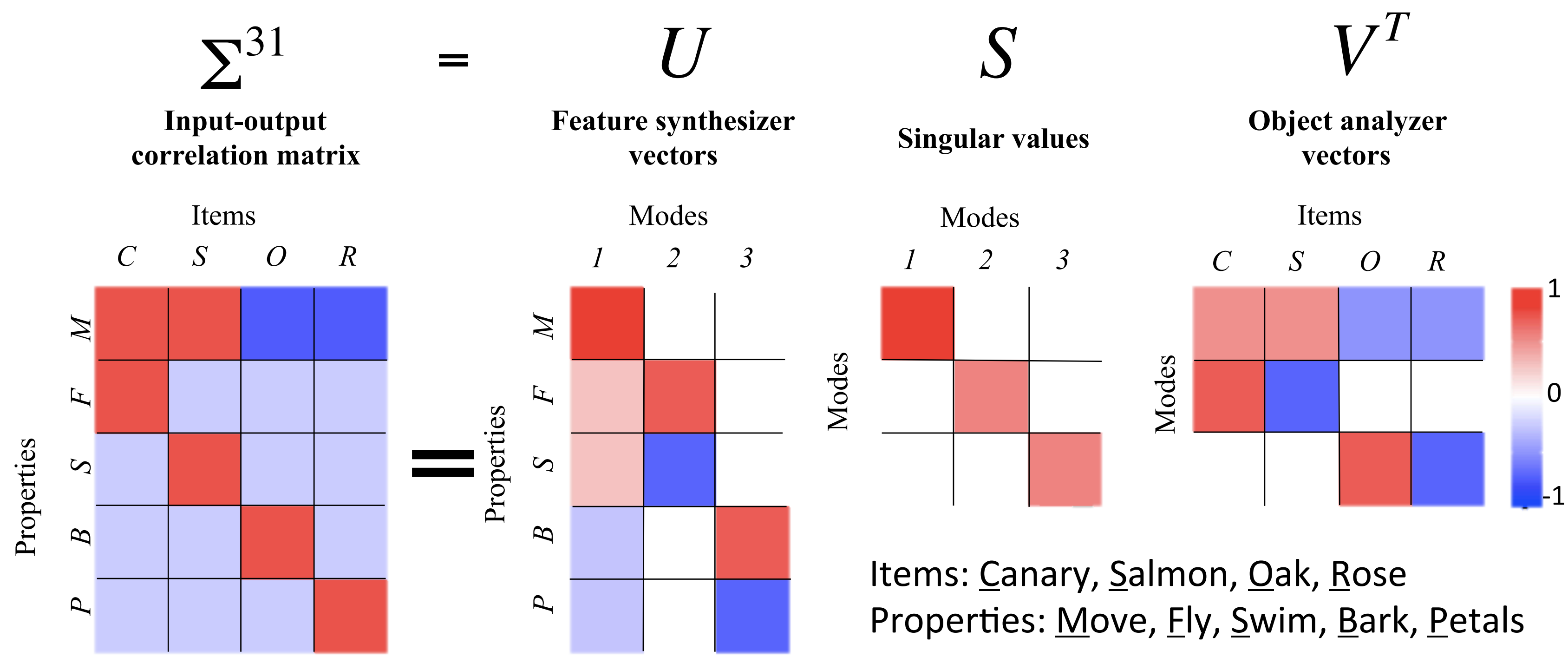


Why are some items more typical members of a category? (i.e. sparrow versus ostrich for the category bird)

An item = vector across properties
A category feature synthesizer = vector across properties

If an item is similar to the feature synthesizer for a category, then it is a typical member of that category.

Category membership verification easier for typical versus atypical items.



How is inductive generalization achieved by neural networks? Inferring familiar properties of a novel item.

Given a new partially described object = vector across subset of properties
What are the rest of the object's properties?

i.e. a "blick" has feathers. Does it fly? Sing?

Partial property vector $\boldsymbol{U^T}$

Neural network internal representation

$\boldsymbol{U}$

Filled in property vector

$$\Sigma^{31} = U \quad S \quad V^T$$

Σ^{31}	=	U	S	V^T
Input-output correlation matrix		**Feature synthesizer vectors**	**Singular values**	**Object analyzer vectors**

How is inductive generalization achieved by neural networks? Inferring which familiar objects have a novel property.

Given a new property = vector across subset of items
Which other items have this property?

i.e. A bird has gene X. Does a crocodile? A dog?

V^T Partial item vector

Neural network internal representation

V Filled in item vector

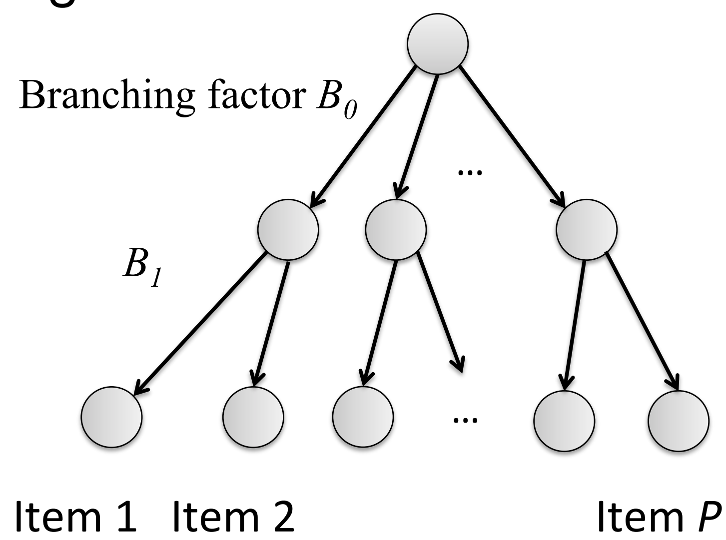
$$\Sigma^{31} = U \quad S \quad V^T$$

Input-output correlation matrix = **Feature synthesizer vectors** · **Singular values** · **Object analyzer vectors**

What is a useful mathematical definition of category coherence?

i.e. “incoherent” = the set of all things that are blue
i.e. “coherent” = the set of all things that are dogs

A natural definition of a coherent category is the singular value of the category, normalized by its level in the hierarchy



Singular value = coherence * exp (- level)

For hierarchically structured data:

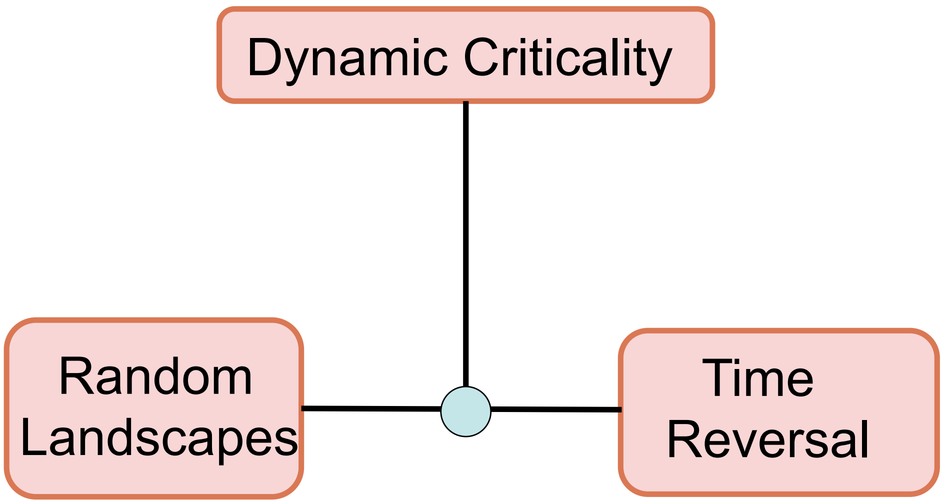
Coherence = similarity of descendants – similarity to nearest out-category

Mathematical Theorem: Coherent categories are learned faster!

Talk Outline

Original motivation: understanding category learning in neural networks

We find random weight initializations, that make a network dynamically critical and allow rapid training of very deep networks.



Understand and exploit geometry of high dimensional error surfaces: need to escape saddle points not local minima.

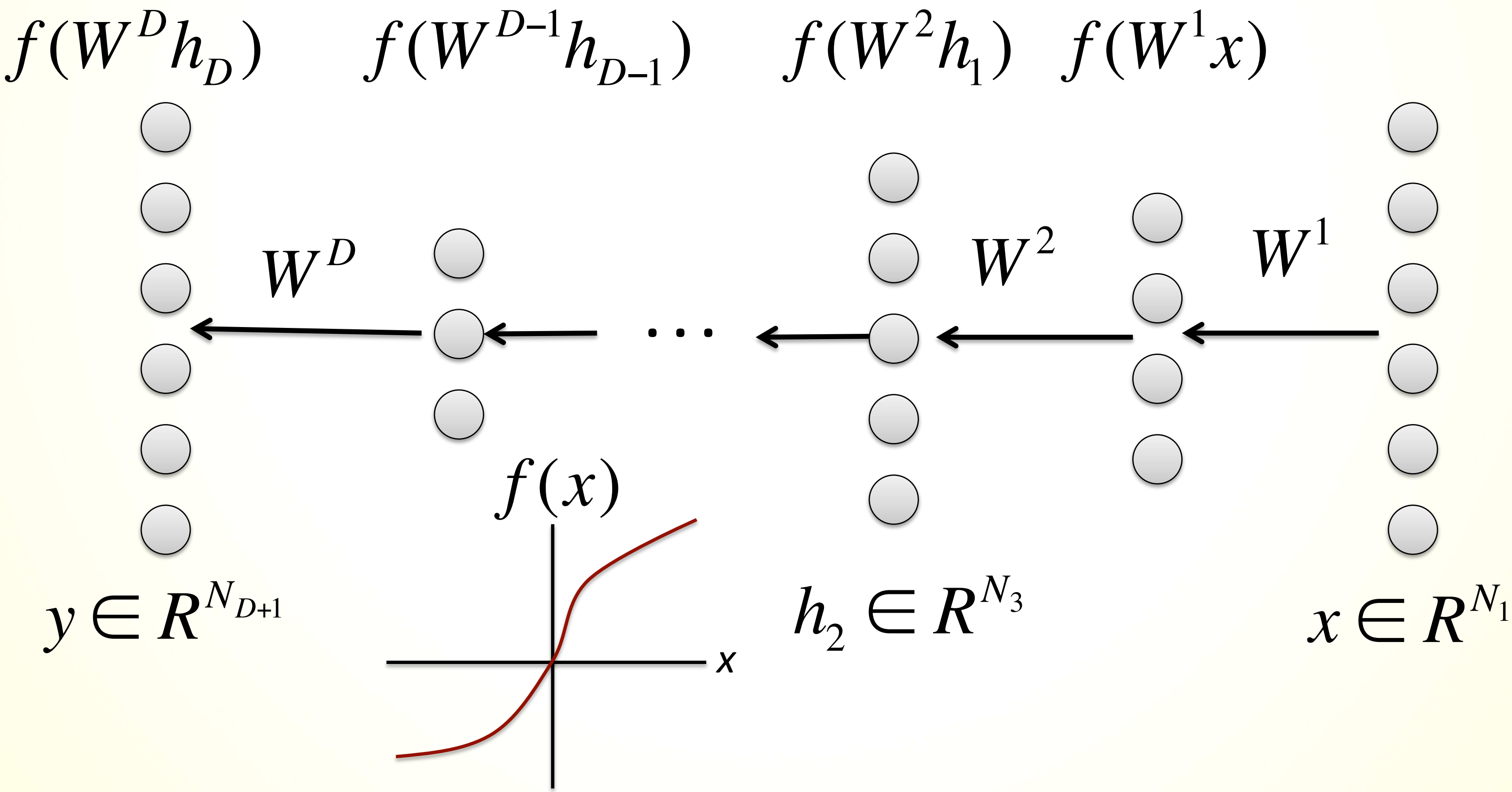
Exploit violations of the second law of thermodynamics to create deep generative models

Towards a theory of deep learning dynamics

- The dynamics of learning in deep networks is non-trivial – i.e. plateaus and sudden transitions to better performance
- How does training time scale with depth?
- How should the learning rate scale with depth?
- How do different weight initializations impact learning speed?
- We will find that weight initializations with *critical dynamics* can aid deep learning and generalization.

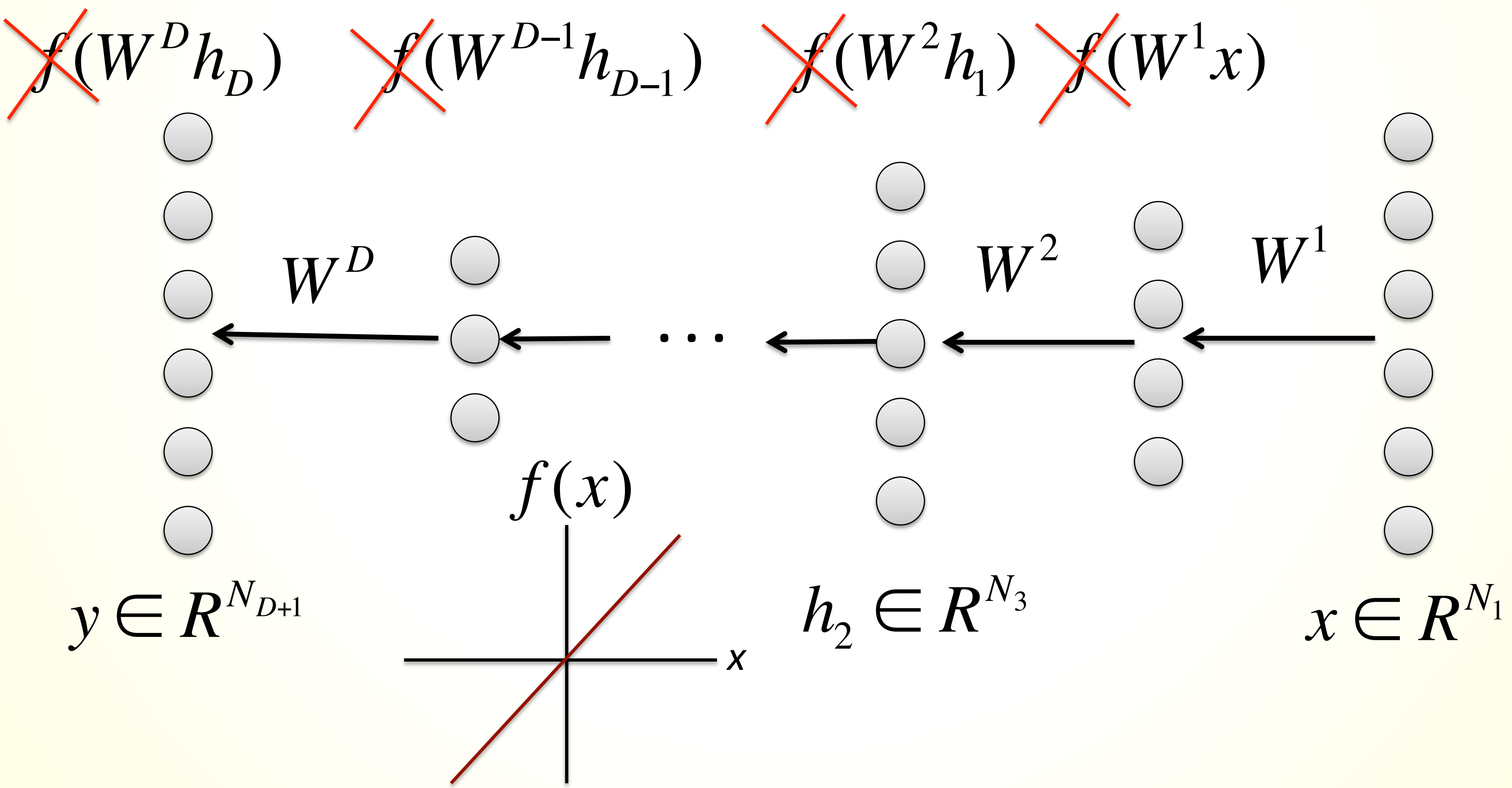
Deep network

- Little hope for a complete theory with arbitrary nonlinearities



Deep *linear* network

- Use a deep *linear* network as a starting point



Deep *linear* network

- Input-output map: Always linear

$$y = \left(\prod_{i=1}^{D} W^i \right) x \equiv W^{tot} x$$

- Gradient descent dynamics: Nonlinear; coupled; nonconvex

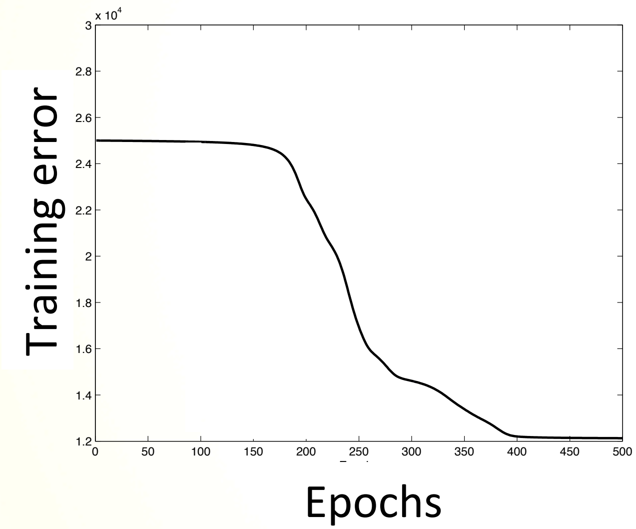
$$\Delta W^l = \lambda \sum_{\mu=1}^{P} \left(\prod_{i=l+1}^{D} W^i \right)^T \left[y^\mu x^{\mu T} - \left(\prod_{i=1}^{D} W^i \right) x^\mu x^{\mu T} \right] \left(\prod_{i=1}^{l-1} W^i \right)^T$$

$$l = 1, \cdots, D$$

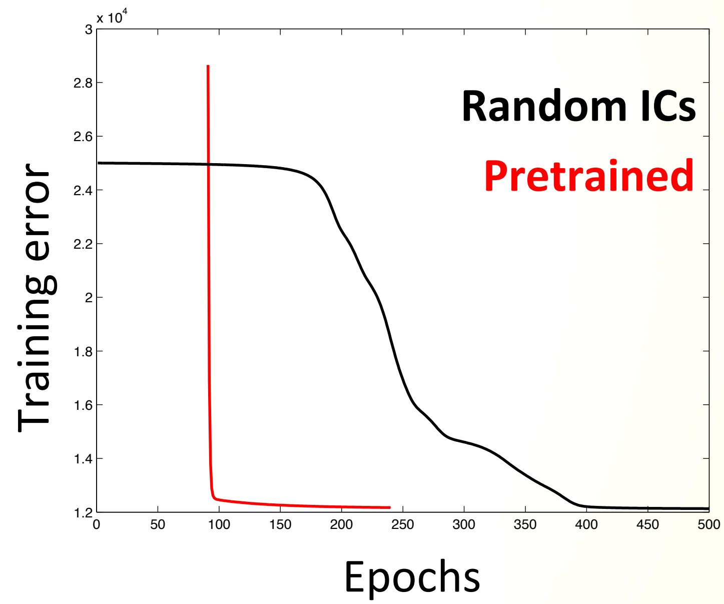
- Useful for studying *learning dynamics,* not representation power.

Nontrivial learning dynamics

Plateaus and sudden transitions

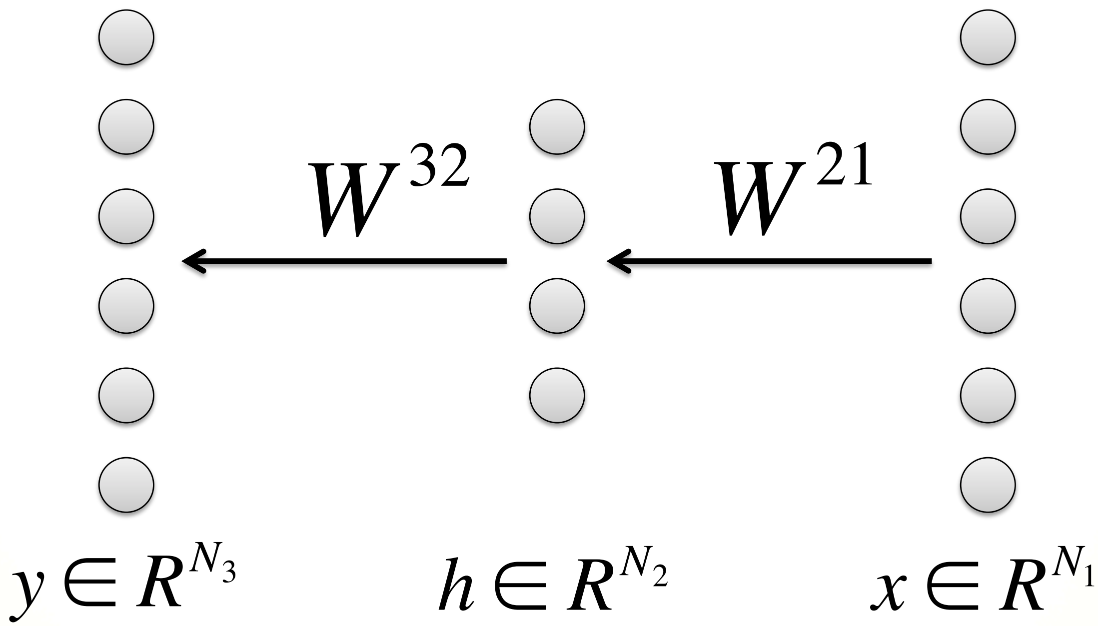


Faster convergence from pretrained initial conditions



- Build intuitions for nonlinear case by analyzing linear case

Three layer dynamics



Problem formulation

- Network trained on patterns $\{x^\mu, y^\mu\}, \mu = 1, \ldots, P.$
- Batch gradient descent on squared error $\|Y - W^{32}W^{21}X\|_F^2$
- Dynamics

$$
\begin{aligned}
\tau \frac{d}{dt} W^{21} &= W^{32^T}\left(\Sigma^{31} - W^{32}W^{21}\Sigma^{11}\right) \\
\tau \frac{d}{dt} W^{32} &= \left(\Sigma^{31} - W^{32}W^{21}\Sigma^{11}\right) W^{21^T}
\end{aligned}
$$

Input correlations: $\Sigma^{11} \equiv E[xx^T] = I$

Input-output correlations: $\Sigma^{31} \equiv E[yx^T]$

(see paper for more general input correlations)

Analytic learning trajectory

SVD of input-output correlations:

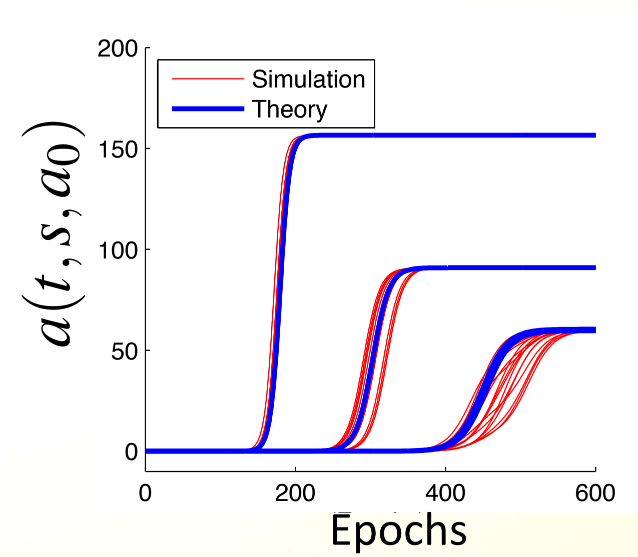
$$\Sigma^{31} = U^{33} S^{31} V^{11^T} = \sum_{\alpha=1}^{N_1} s_\alpha u^\alpha v^{\alpha T}$$

τ	1/Learning rate
s	Singular value
a_0	Initial mode strength

Network input-output map:

$$W^{32}(t) W^{21}(t) = \sum_{\alpha=1}^{N_2} a(t, s_\alpha, a_\alpha^0)\, u^\alpha v^{\alpha T} \quad \text{where} \quad a(t,s,a_0) = \frac{s e^{2st/\tau}}{e^{2st/\tau} - 1 + s/a_0}$$

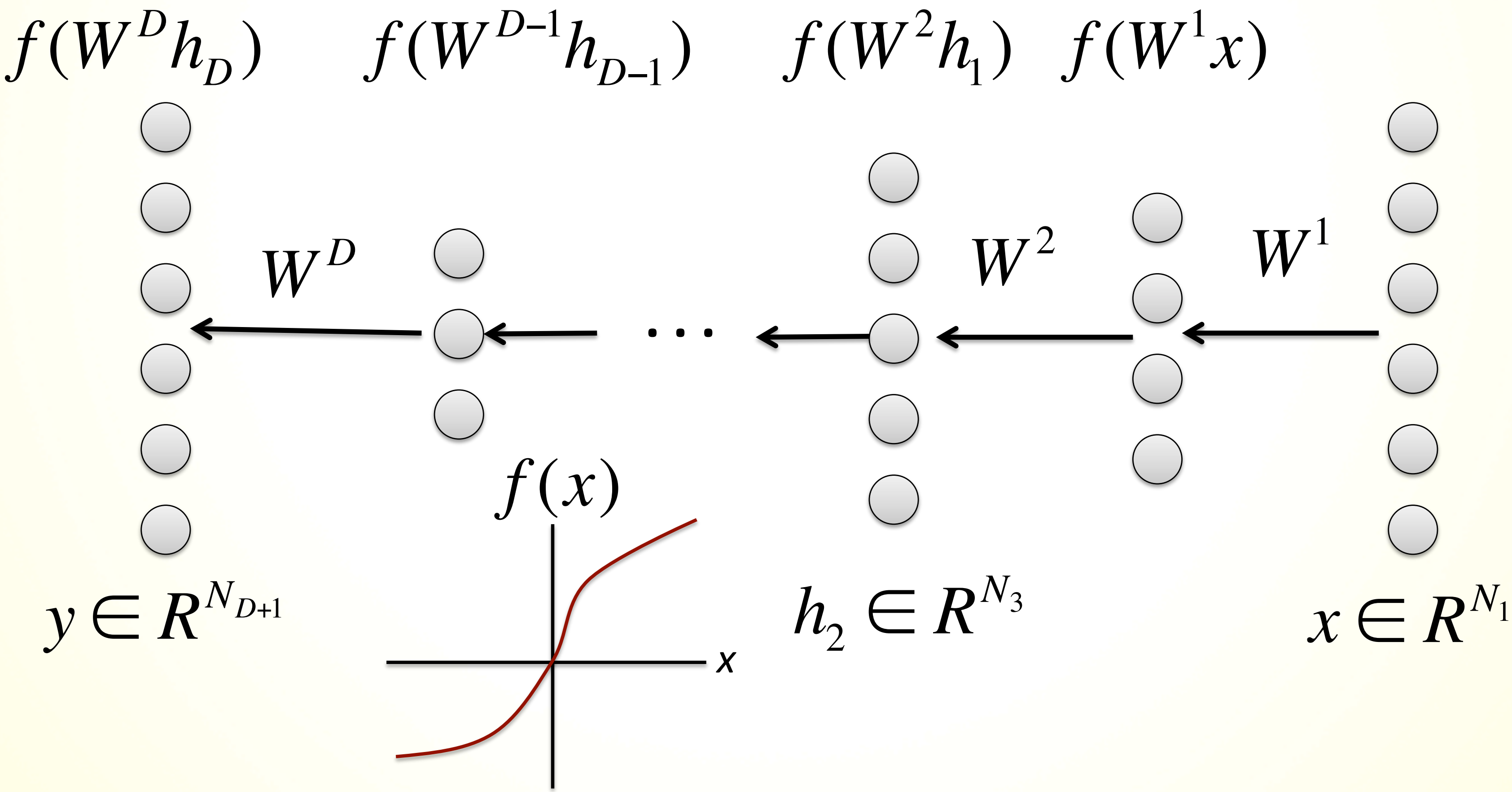
- Starting from decoupled initial conditions.
- Each ‘connectivity mode’ evolves independently
- Singular value s learned at time O(1/s)



Saxe, McCelland, Ganguli, ICLR, 2014

Deeper network learning dynamics

- Jacobian that back-propagates gradients can explode or decay



Deeper networks

- Can generalize to arbitrary depth network

- Each effective singular value *a* evolves independently

$$\tau \frac{d}{dt} a = (N_l - 1) a^{2-2/(N_l-1)} (s - a)$$

τ	1/Learning rate
s	Singular value
N_l	# layers

- In deep networks, combined gradient is $O(N_l/\tau)$

$$a = \prod_{i=1}^{N_l - 1} W_i$$

Deep linear learning speed

- Intuition (see paper for details):
  - Gradient norm $O(N_l)$
  - Learning rate $O(1/N_l)$ (N_l = # layers)
  - Learning time $O(1)$
- Deep learning *can be fast* with the right ICs.

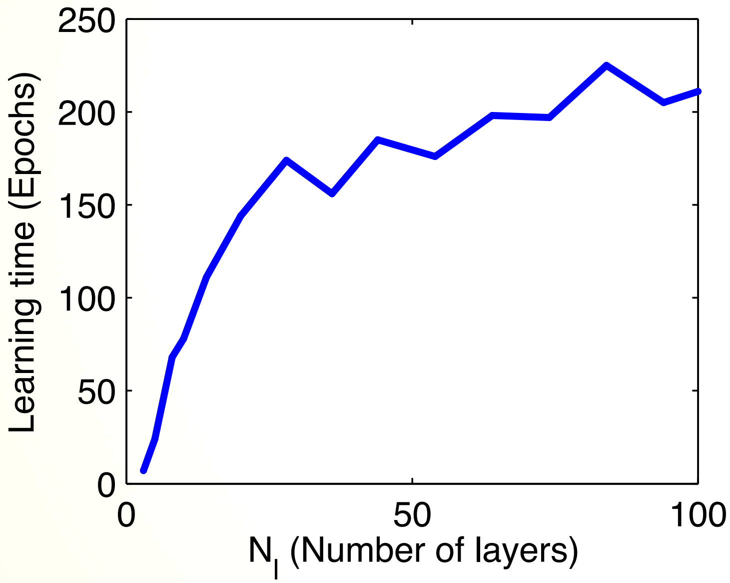
Saxe, McClelland, Ganguli ICLR 2014

MNIST learning speeds

- Trained deep *linear* nets on MNIST
- Depths ranging from 3 to 100
- 1000 hidden units/layer (overcomplete)
- Decoupled initial conditions with fixed initial mode strength
- Batch gradient descent on squared error
- Optimized learning rates for each depth
- Calculated epoch at which error falls below fixed threshold

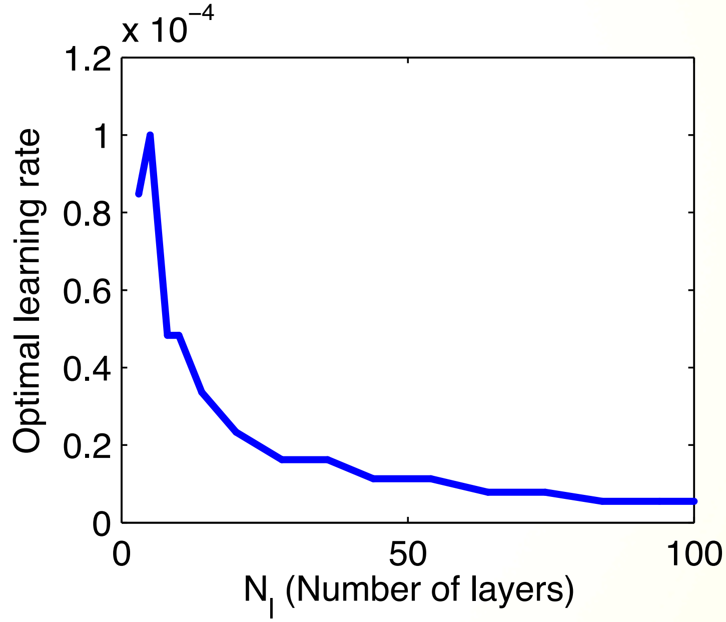
MNIST depth dependence

Time to criterion



Depth

Optimal learning rate



Depth

Deep linear networks

- Deep learning *can be fast* with decoupled ICs and O(1) initial mode strength. **How to find these?**

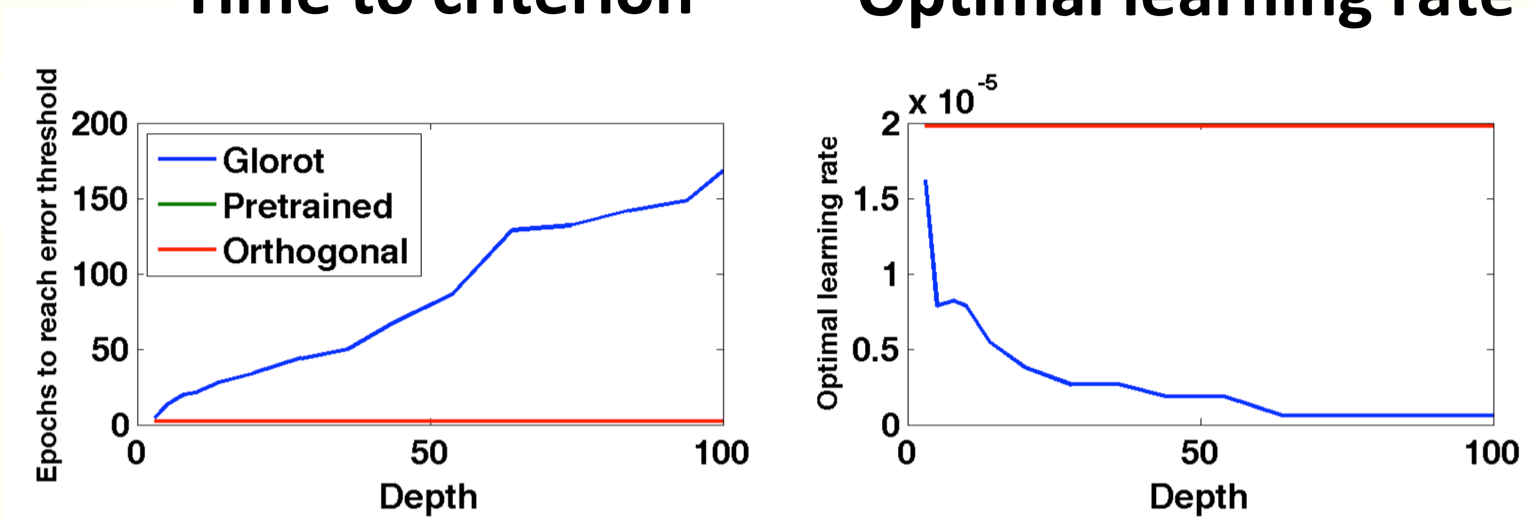
- Answer: Pre-training and random orthogonal initializations can find these special initial conditions that allow depth independent training times!!

- But scaled random Gaussian initial conditions on weights cannot.

Depth-independent training time

- Deep *linear* networks on MNIST
- Scaled random Gaussian initialization (Glorot & Bengio, 2010)

Time to criterion **Optimal learning rate**

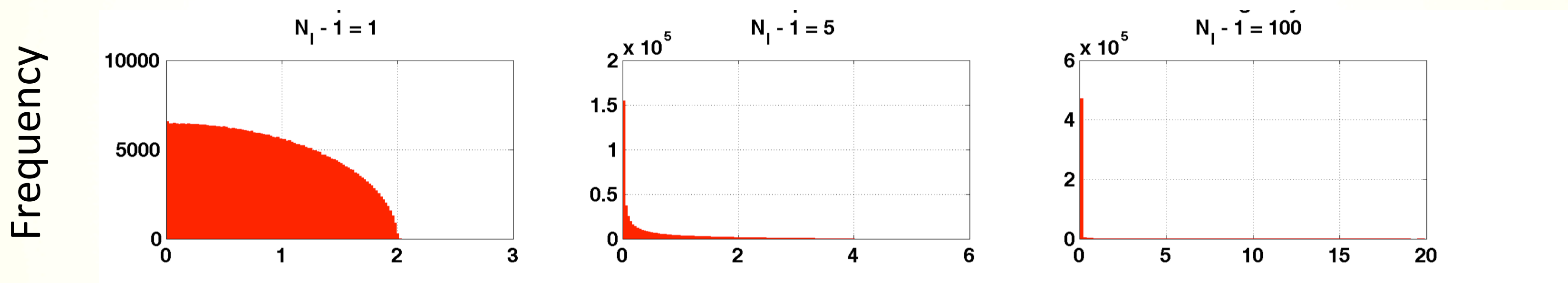


- Pretrained and orthogonal have fast **depth-independent** training times!

Random vs orthogonal

- Gaussian preserves norm of random vector *on average*

1 layer net | 5 layer net | 100 layer net

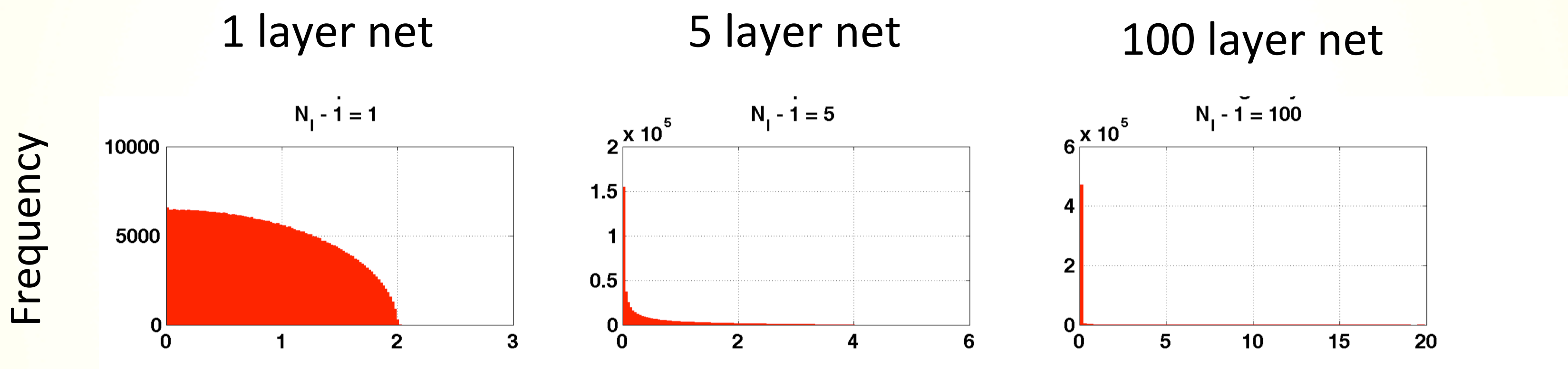


Singular values of $W^{tot} = \prod_{i=1}^{N_l - 1} W^i$

- *Attenuates* on subspace of high dimension
- *Amplifies* on subspace of low dimension

Random vs orthogonal

- Glorot preserves norm of random vector *on average*



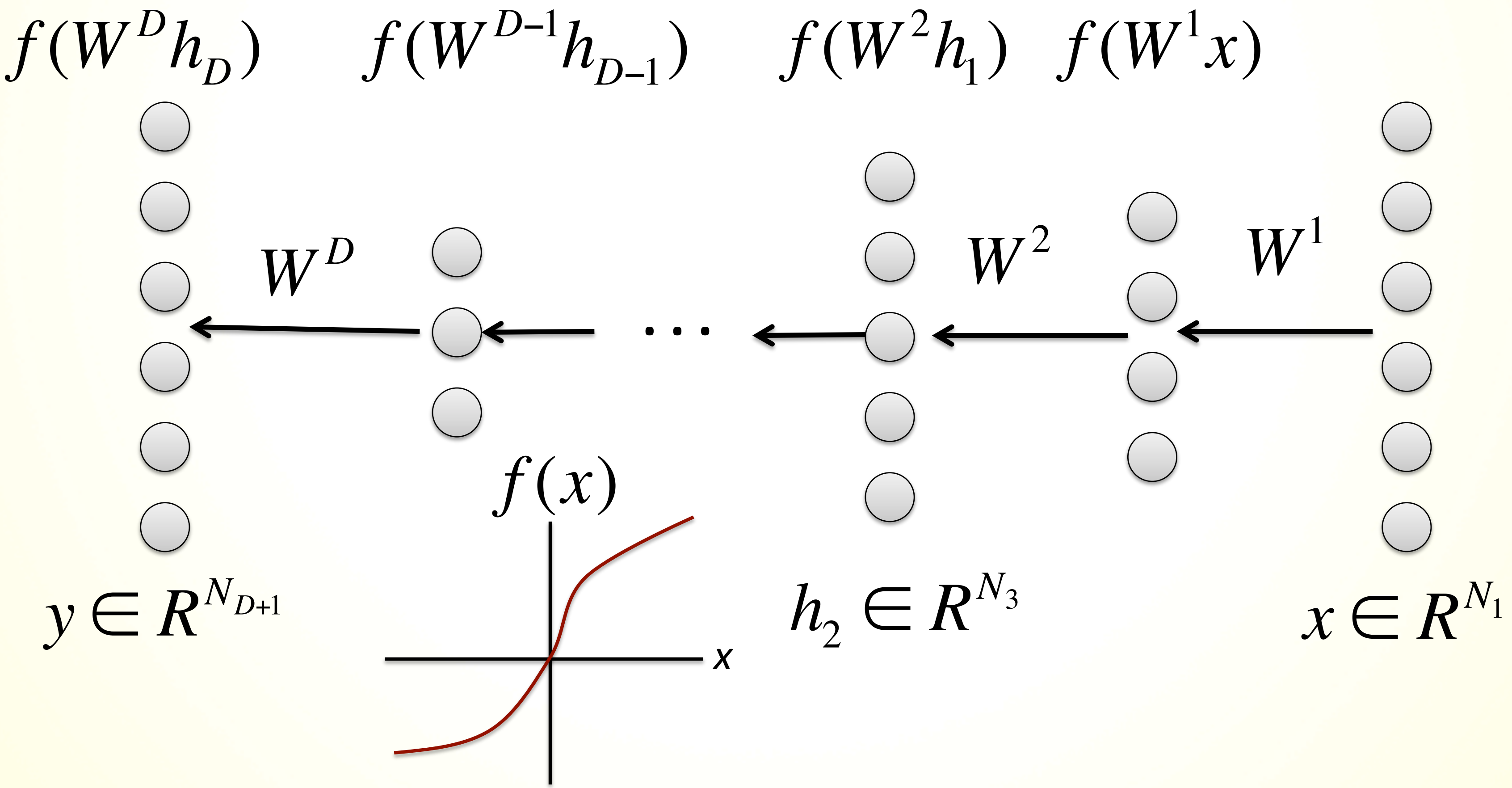
Singular values of $W^{tot} = \prod_{i=1}^{N_l-1} W^i$

- Orthogonal preserves norm of all vectors *exactly*

All singular values of $W^{tot} = 1$

Deeper network learning dynamics

- Jacobian that back-propagates gradients can explode or decay



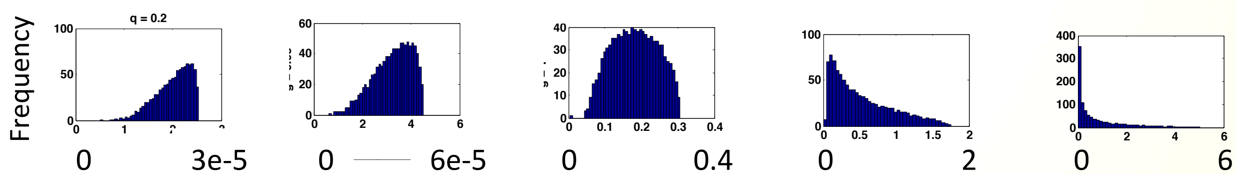
Extensive Criticality yields Dynamical Isometry in *nonlinear* nets

Suggests initialization for *nonlinear* nets

- near-isometry on subspace of large dimension
- Singular values of *end-to-end* Jacobian $J_{ij}^{N_l,1}(x^{N_l}) \equiv \left.\frac{\partial x_i^{N_l}}{\partial x_j^1}\right|_{x^{N_l}}$ concentrated around 1.

Scale orthogonal matrices by gain *g* to counteract contractive nonlinearity

Singular values of *J*



Gain *g=0.9* *g=0.95* *g=1* *g=1.05* *g=1.1*

Just beyond *edge of chaos (g>1)* may be good initialization

Dynamic Isometry Initialization

- g>1 speeds up **30 layer nonlinear** nets
  - Tanh network, softmax output, 500 units/layer
  - No regularization (weight decay, sparsity, dropout, etc)

MNIST Classification error, epoch 1500	**Train Error (%)**	**Test Error (%)**
Gaussian (g=1, random)	2.3	3.4
g=1.1, random	1.5	3.0
g=1, orthogonal	2.8	3.5
Dynamic Isometry (g=1.1, orthogonal)	**0.095**	**2.1**

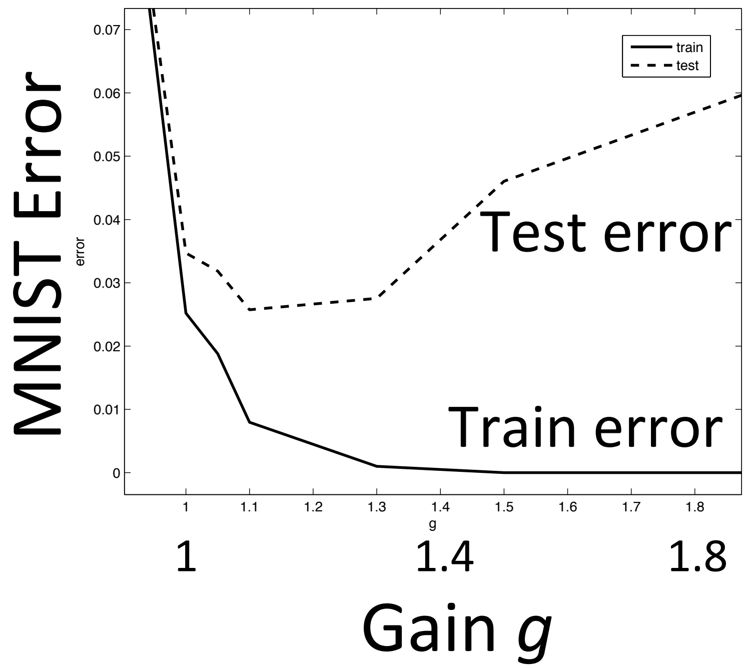
- Dynamic isometry reduces test error by 1.4% pts

Summary

- Deep linear nets have **nontrivial nonlinear learning dynamics.**
- Learning time inversely proportional to strength of input-output correlations.
- With the right initial weight conditions, number of training epochs can remain finite as depth increases.
- Dynamically critical networks just beyond the edge of chaos enjoy **depth-independent** learning times.

Beyond learning: criticality and generalization

- Deep networks + large gain factor *g* train exceptionally quickly
- But large *g* incurs heavy cost in generalization performance

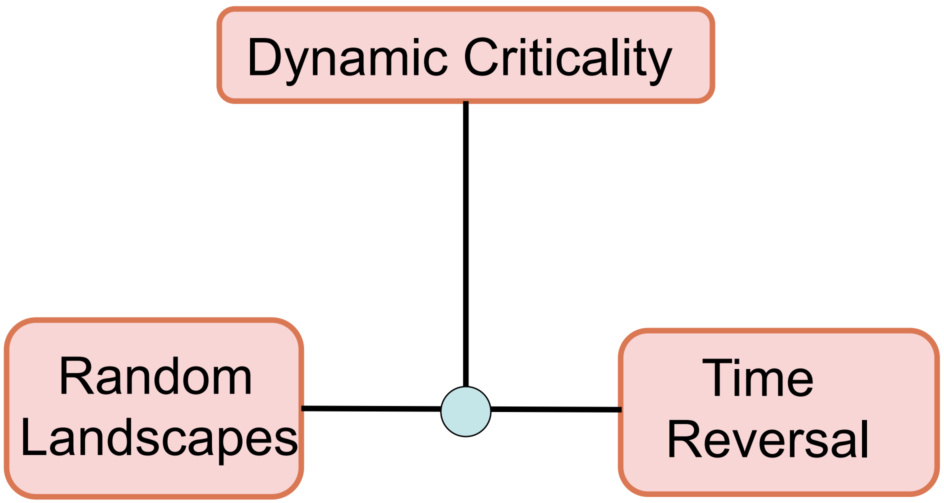


- Suggests small initial weights regularize towards smoother functions

Talk Outline

Original motivation: understanding category learning in neural networks

We find random weight initializations, that make a network dynamically critical and allow rapid training of very deep networks.

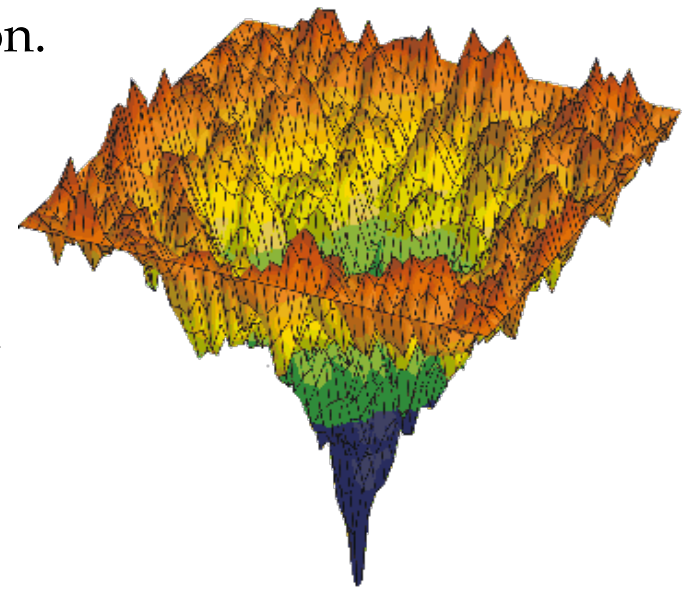


Understand and exploit geometry of high dimensional error surfaces: need to escape saddle points not local minima.

Exploit violations of the second law of thermodynamics to create deep generative models

High dimensional nonconvex optimization

It is often thought that local minima at high error stand as as a major impediment to non-convex optimization.



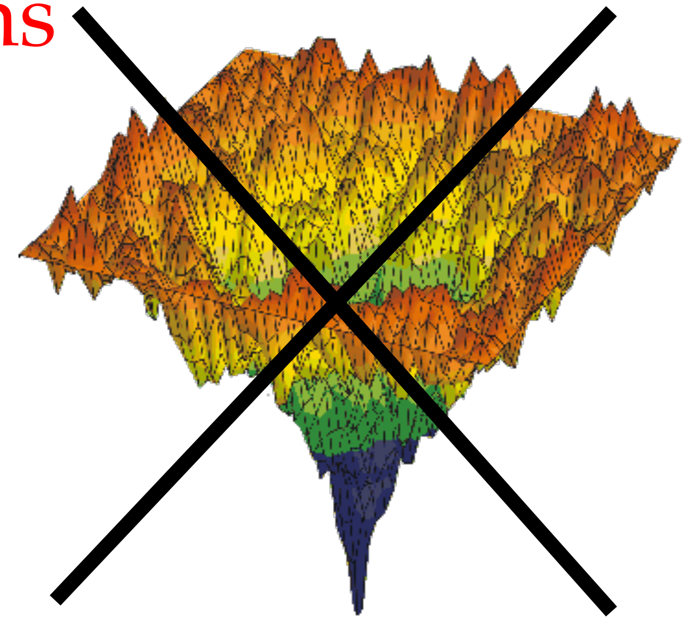
In random non-convex error surfaces over high dimensional spaces, local minima at high error are exponentially rare in the dimensionality.

Instead saddle points proliferate.

We developed an algorithm that rapidly escapes saddle points in high dimensional spaces.

Identifying and attacking the saddle point problem in high dimensional non-convex optimization. Yann Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, Yoshua Bengio. NIPS 2014

General properties of error landscapes in high dimensions



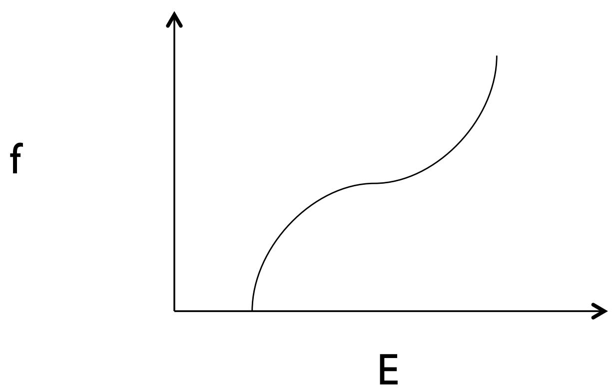
From statistical physics:

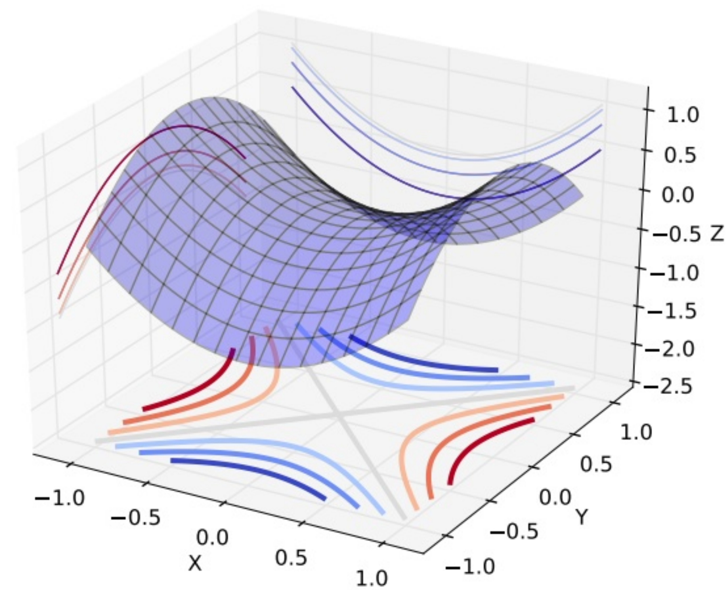
Consider a random Gaussian error landscape over N variables.

Let x be a critical point.

Let E be its error level.

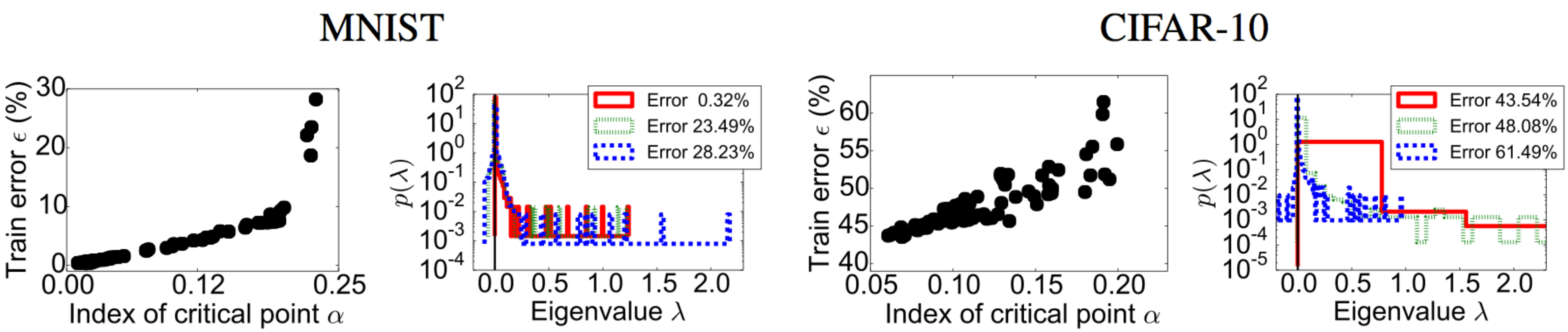
Let f be the fraction of negative curvature directions.





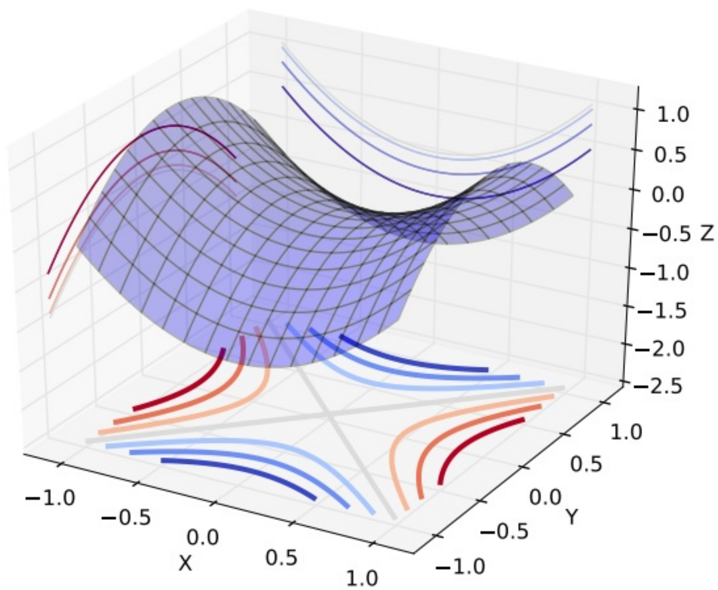
Bray and Dean, Physical Review Letters, 2007

Properties of Error Landscapes on the Synaptic Weight Space of a Deep Neural Net



Qualitatively consistent with the
statistical physics theory of random error landscapes

How to descend saddle points



Newton's Method

$$\Delta x = -H^{-1}\,\nabla f(x)$$

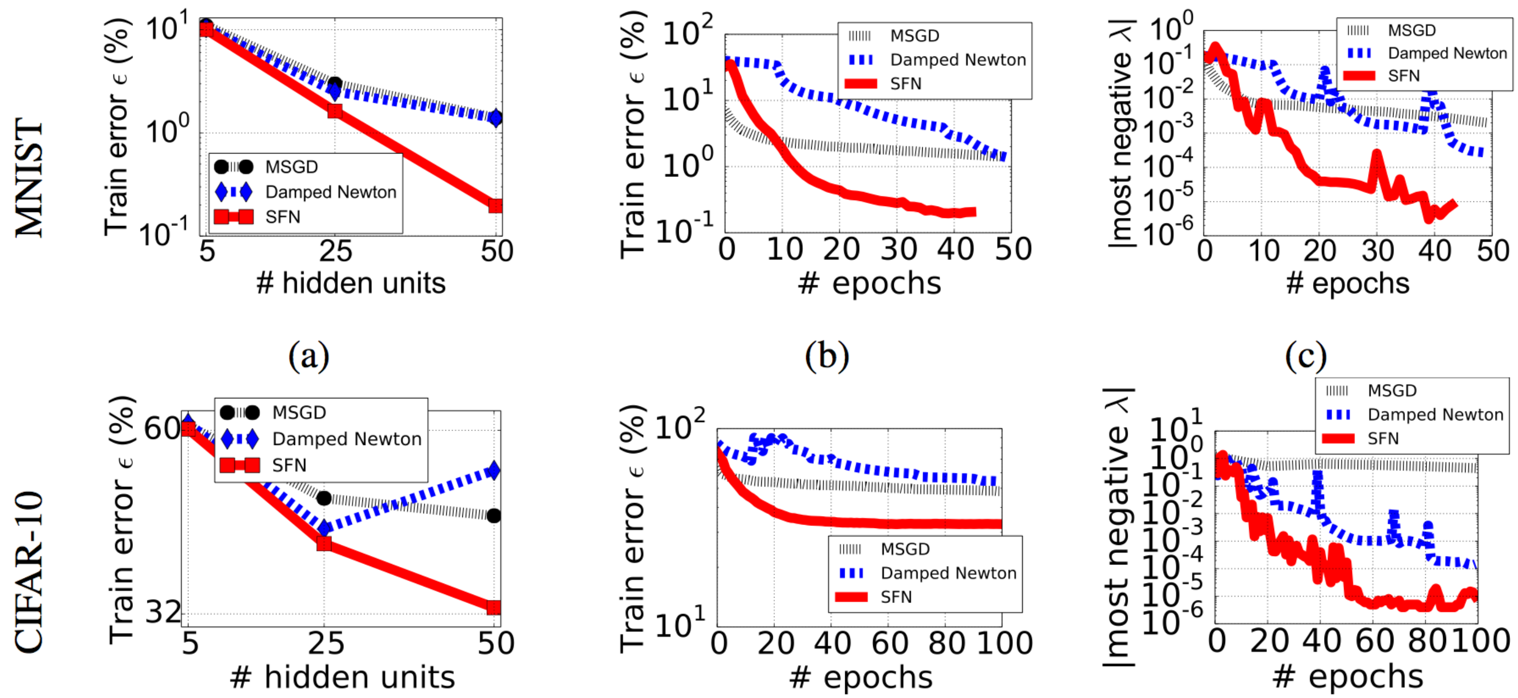
Saddle Free Newton's Method

$$\Delta x = -|H|^{-1}\,\nabla f(x)$$

Intuition: saddle points attract Newton's method, but
repel saddle free Newton's method.

Derivation: minimize a linear approximation to f(x) within a trust region
in which the linear and quadratic approximations agree

Performance of saddle free Newton in learning deep neural networks.



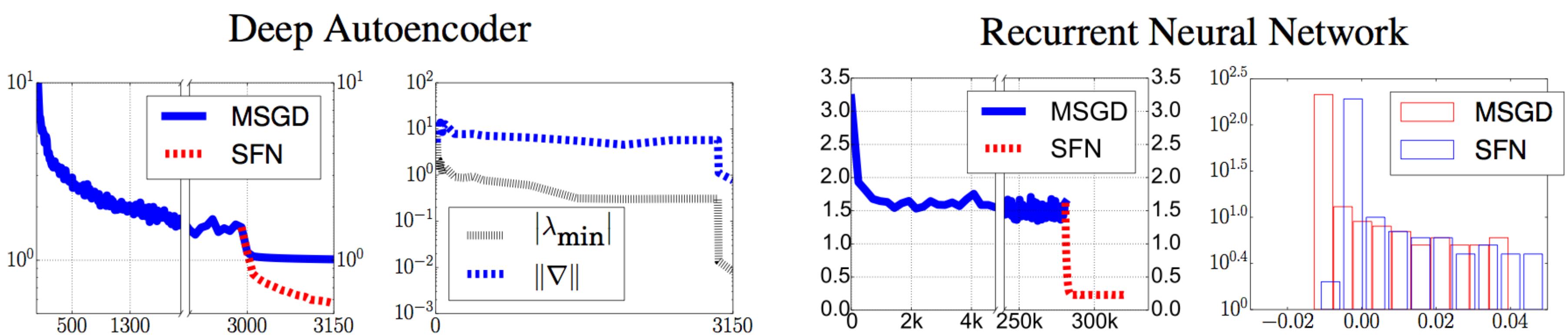
SFN out-performs

(1) minibatch stochastic gradient descent and

(2) damped Newton's method

The performance advantage increases with the problem dimensionality.

Performance of saddle free Newton in learning deep neural networks.



When stochastic gradient descent appears to plateau, switching to saddle Free newton escapes the plateau.

Talk Outline

Original motivation: understanding category learning in neural networks

We find random weight initializations, that make a network dynamically critical and allow rapid training of very deep networks.

Dynamic Criticality

Random Landscapes

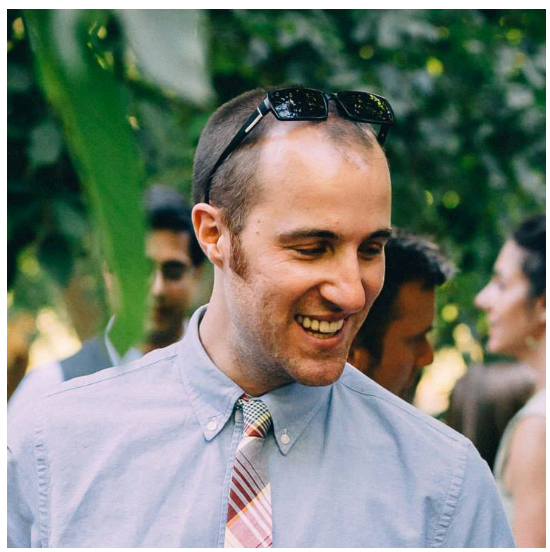
Time Reversal

Understand and exploit geometry of high dimensional error surfaces: need to escape saddle points not local minima.

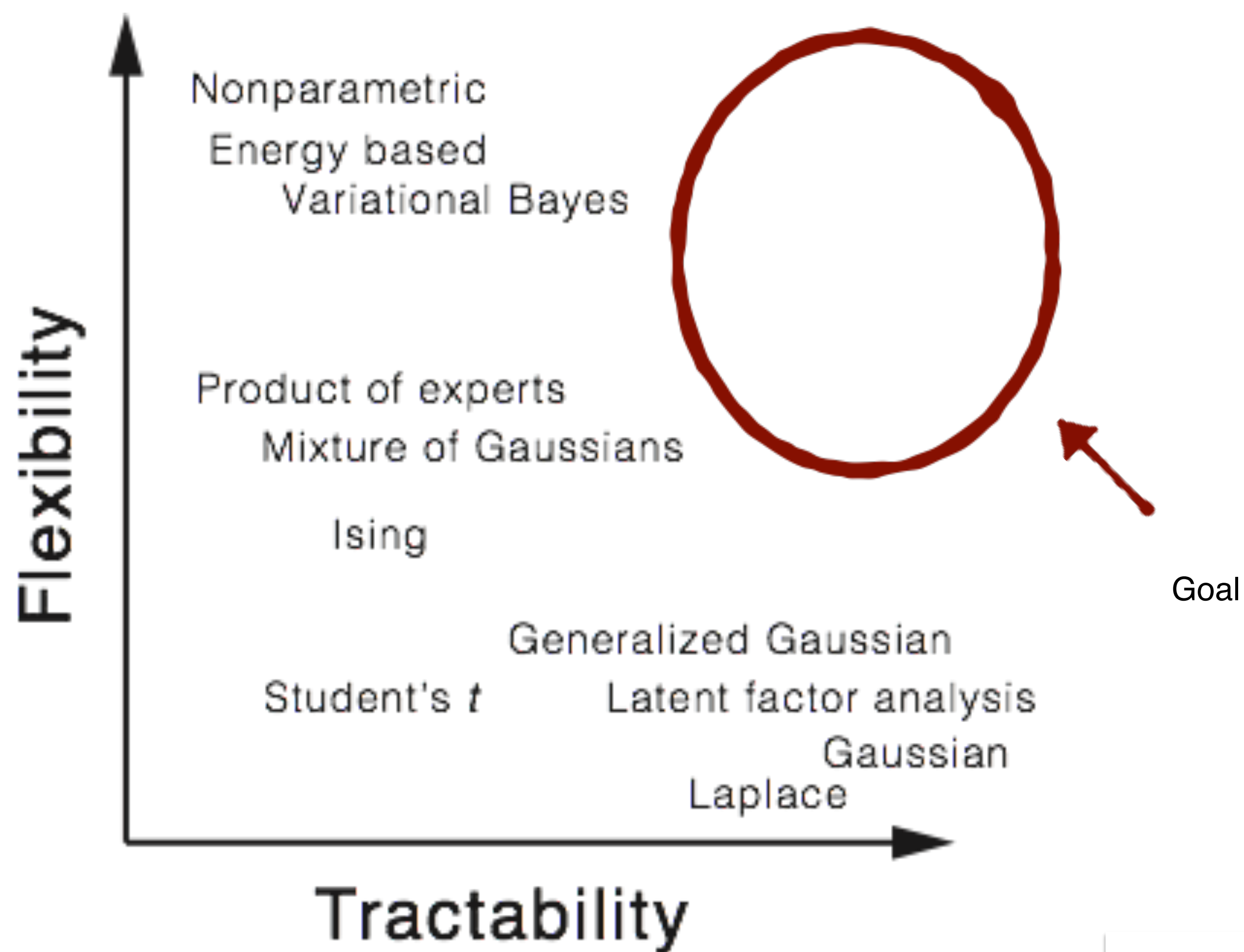
Exploit violations of the second law of thermodynamics to create deep generative models

Modeling Complex Data by ReversingTime

with Jascha Sohl-Dickstein
Eric Weiss, Niru Maheswaranathan



Flexibility-Tractability Tradeoff in Probabilistic Models



Achieving Flexibility and Tractability

- Physical motivation
  - Destroy structure in data through a diffusive process.
  - Carefully record the destruction.
  - Use deep networks to **reverse time and create structure from noise.**
- Inspired by recent results in non-equilibrium statistical mechanics which show that entropy can transiently decrease for short time scales (violations of second law)

Physical Intuition: Destruction of Structure through Diffusion

- Dye density represents probability density
- Goal: Learn structure of probability density
- Observation: Diffusion destroys structure

Data distribution → Uniform distribution

Physical Intuition: Recover Structure by Reversing Time

- What if we could reverse this process?
- Recover data distribution by starting from uniform distribution and running a new type of reverse dynamics (using a trained deep network)

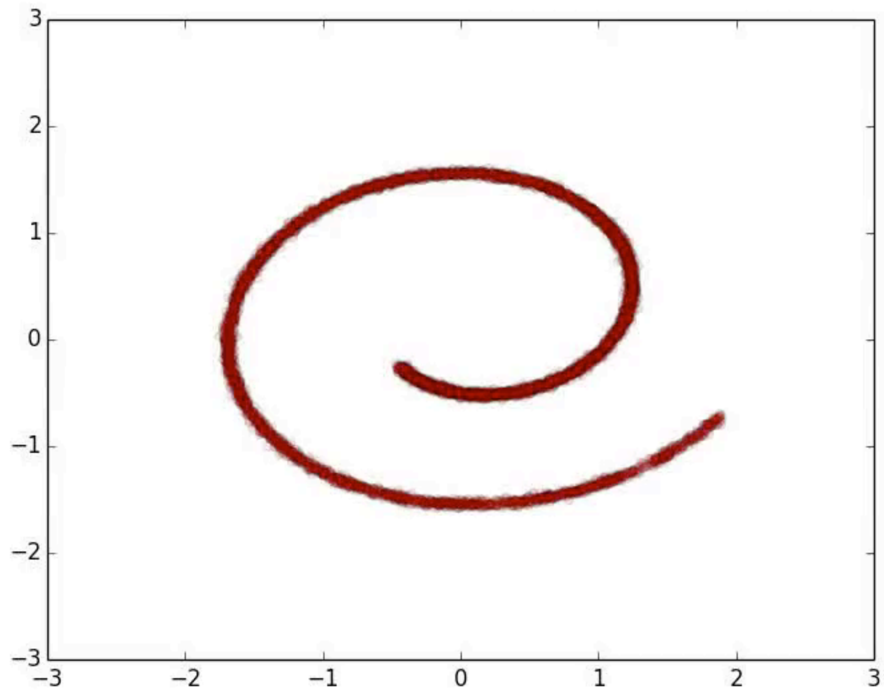
Data distribution ← Uniform distribution

Physical Intuition: Recover Structure by Reversing Time

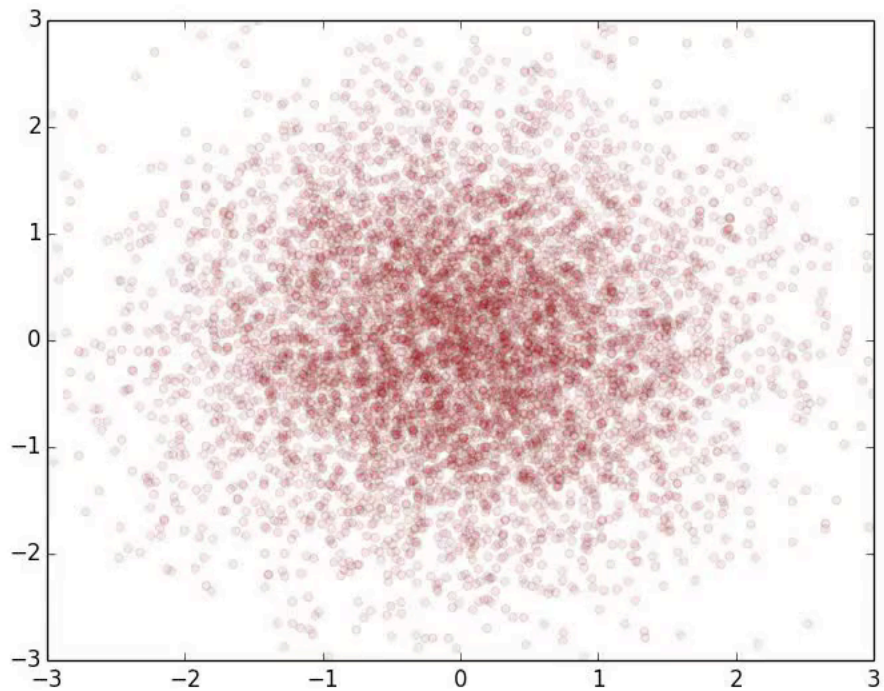
- What if we could reverse time?
- Recover data distribution by starting from uniform distribution and running dynamics backwards (using a trained deep network)

Data distribution ← Uniform distribution

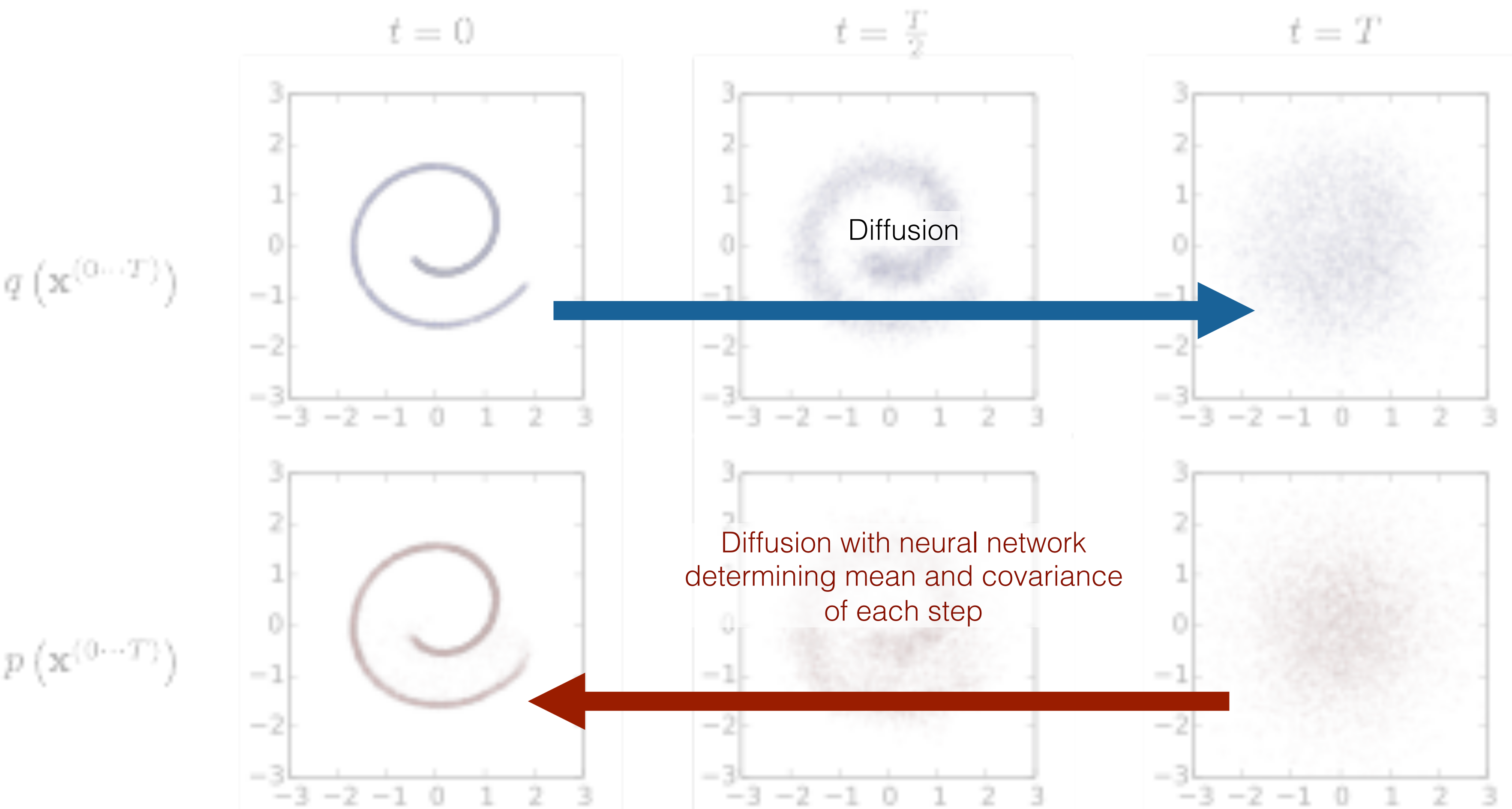
- Forward diffusion process
  - Start at data
  - Run Gaussian diffusion until samples become Gaussian blob



- Reverse diffusion process
  - Start at Gaussian blob
  - Run Gaussian diffusion until samples become data distribution



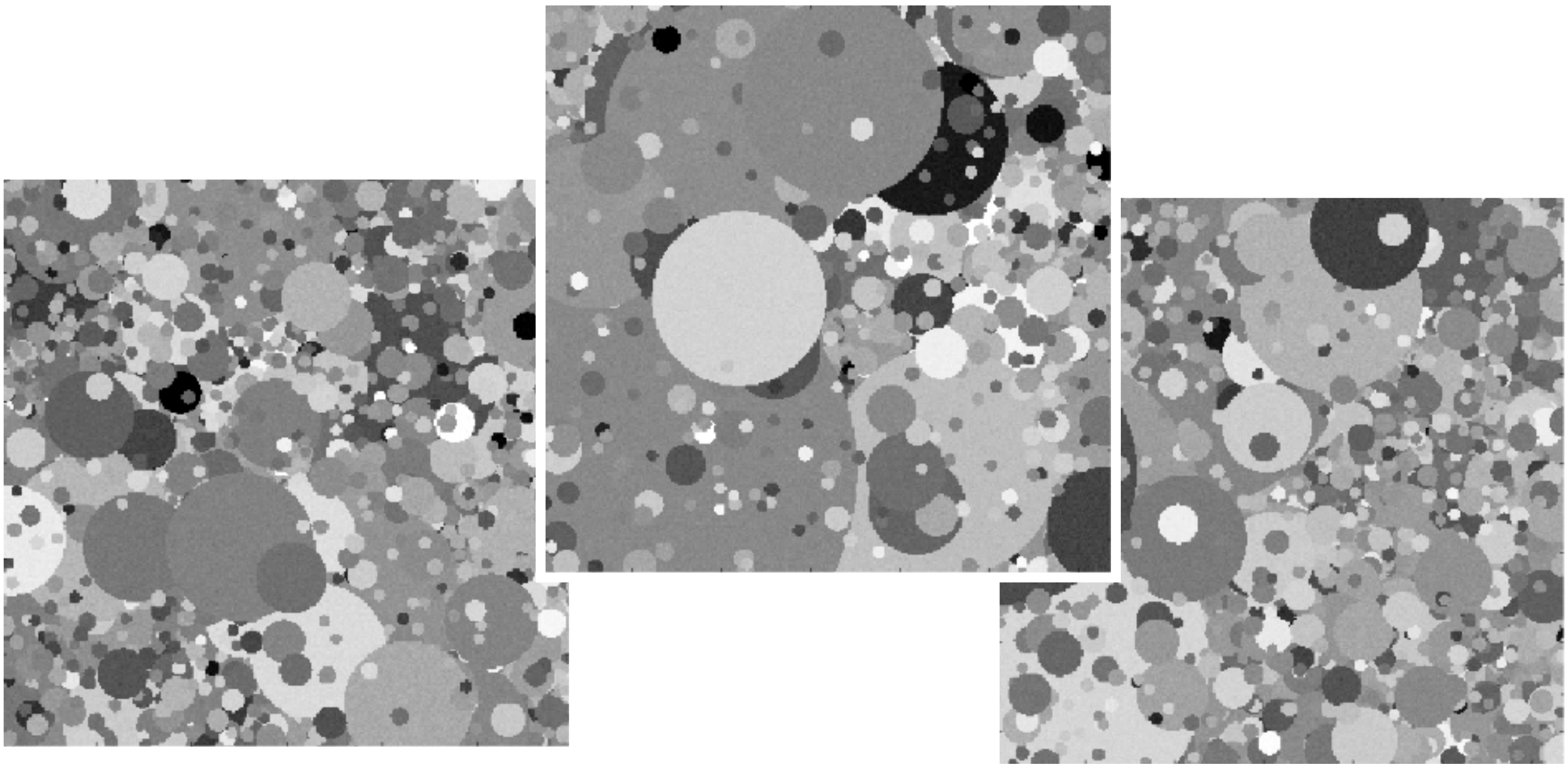
Swiss Roll



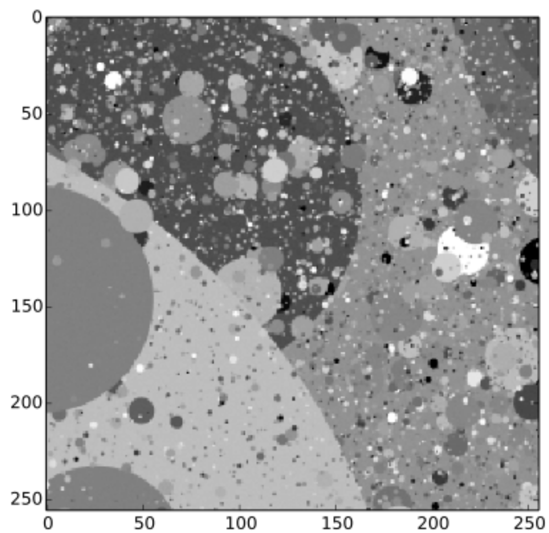
$t = 0$
$t = \frac{T}{2}$
$t = T$
$q\left(\mathbf{x}^{(0\cdots T)}\right)$
$p\left(\mathbf{x}^{(0\cdots T)}\right)$
Diffusion
Diffusion with neural network determining mean and covariance of each step

Dead Leaf Model

- Training data

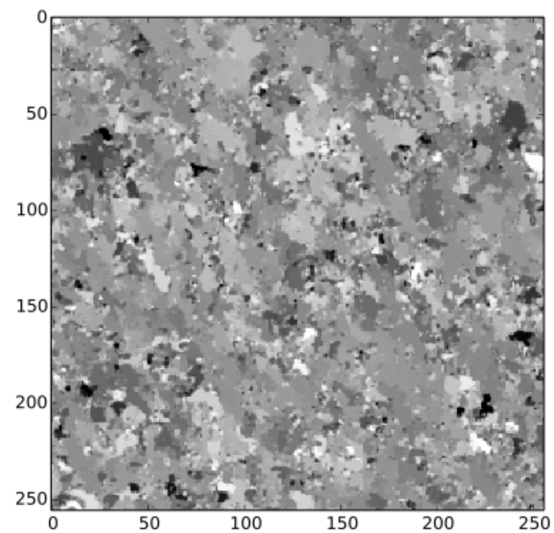


Diffusion Probabilistic Model on Dead Leaves



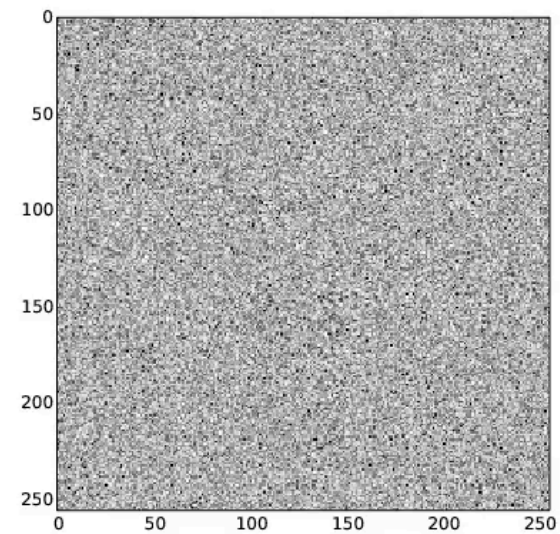
Training Data

Log likelihood
1.24 bits/pixel



Sample from
[Theis *et al*, 2012]

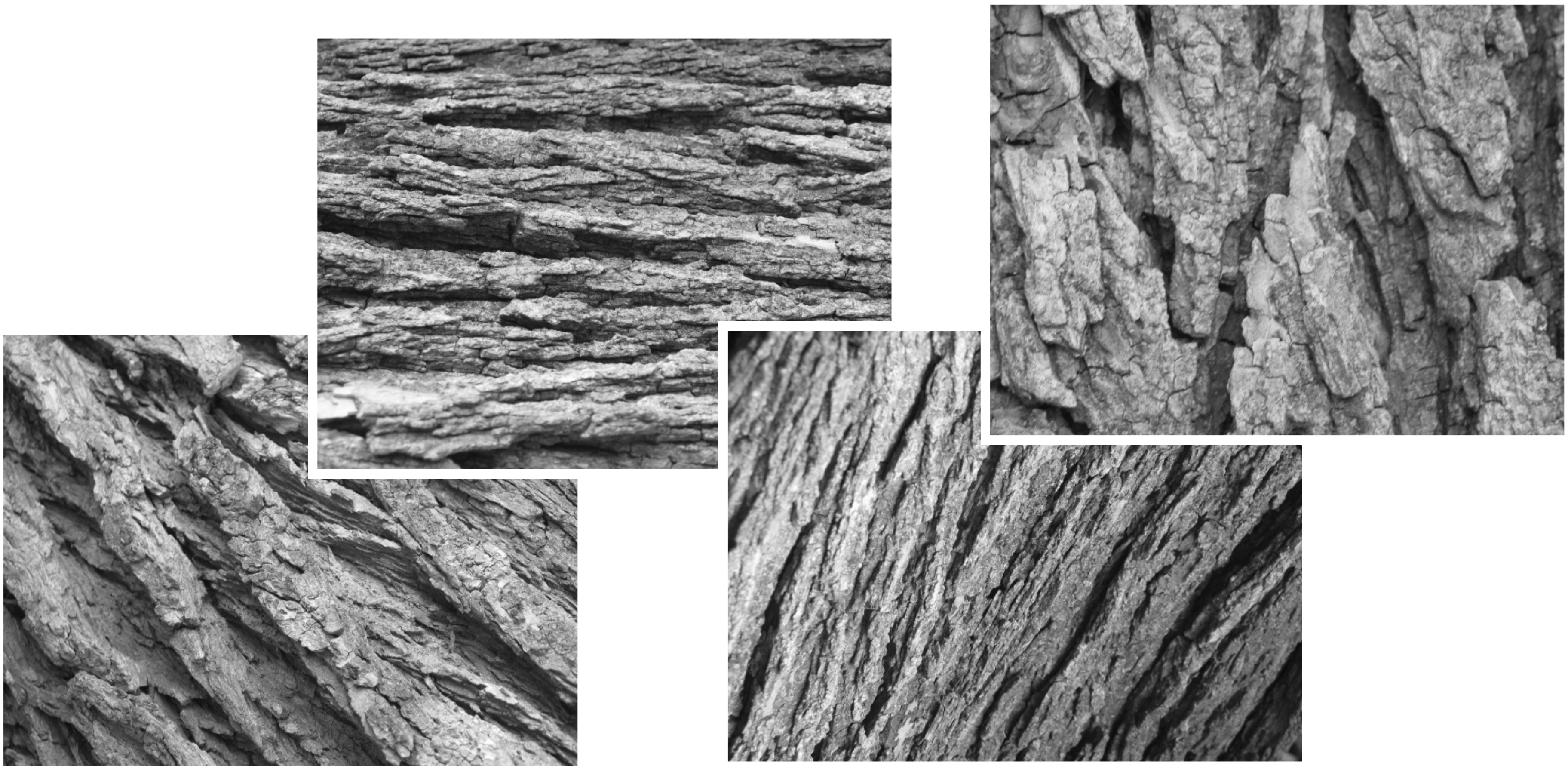
Log likelihood
1.49 bits/pixel



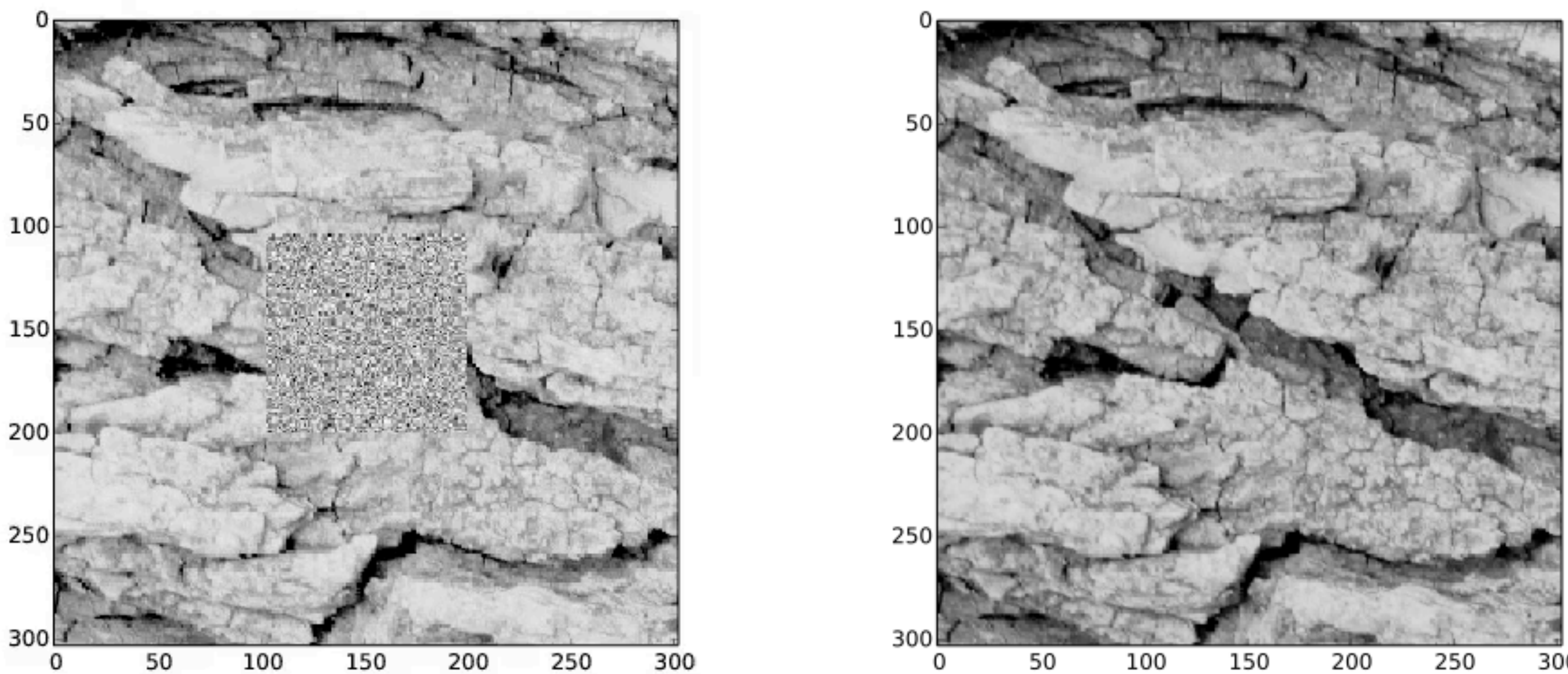
Sample from
diffusion model

Natural Images

- Training data



Diffusion Probabilistic Model
Inpainting



Flexible and Tractable Learning of Probabilistic Models

- Flexible
  - Every distribution has a diffusion process (ongoing work applying to binary spike trains, and full color natural images from diverse scenes)
- Tractable
  - Training: Estimate mean and covariance of Gaussian
  - Sampling: Exact - model defined by sampling chain
  - Inference: Via sampling
  - Evaluation: Cheap - compute probability of sequence of Gaussians

Acknowledgements and Funding

Saxe, J. McClelland, S. Ganguli, Learning hierarchical category structure in deep neural networks, Cog Sci. 2013.

Saxe, J. McClelland, S. Ganguli, Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, ICLR 2014.

J. Sohl-Dickstein, B. Poole, and S. Ganguli, Fast large scale optimization by unifying stochastic gradient and quasi-Newton methods, ICML 2014.

Identifying and attacking the saddle point problem in high dimensional non-convex optimization, Yann Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, Yoshua Bengio, NIPS 2014.

Deep unsupervised learning using non-equilibrium thermodynamics, J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, ICML 2015.

On simplicity and complexity in the brave new world of large-scale neuroscience, P. Gao and S. Ganguli, Current Opinion in Neurobiology, 2015.

Funding:

Bio-X Neuroventures
Burroughs Wellcome
Genentech Foundation
James S. McDonnell Foundation
McKnight Foundation
National Science Foundation

Office of Naval Research
Simons Foundation
Sloan Foundation
Simons Foundation
Swartz Foundation
Stanford Terman Award

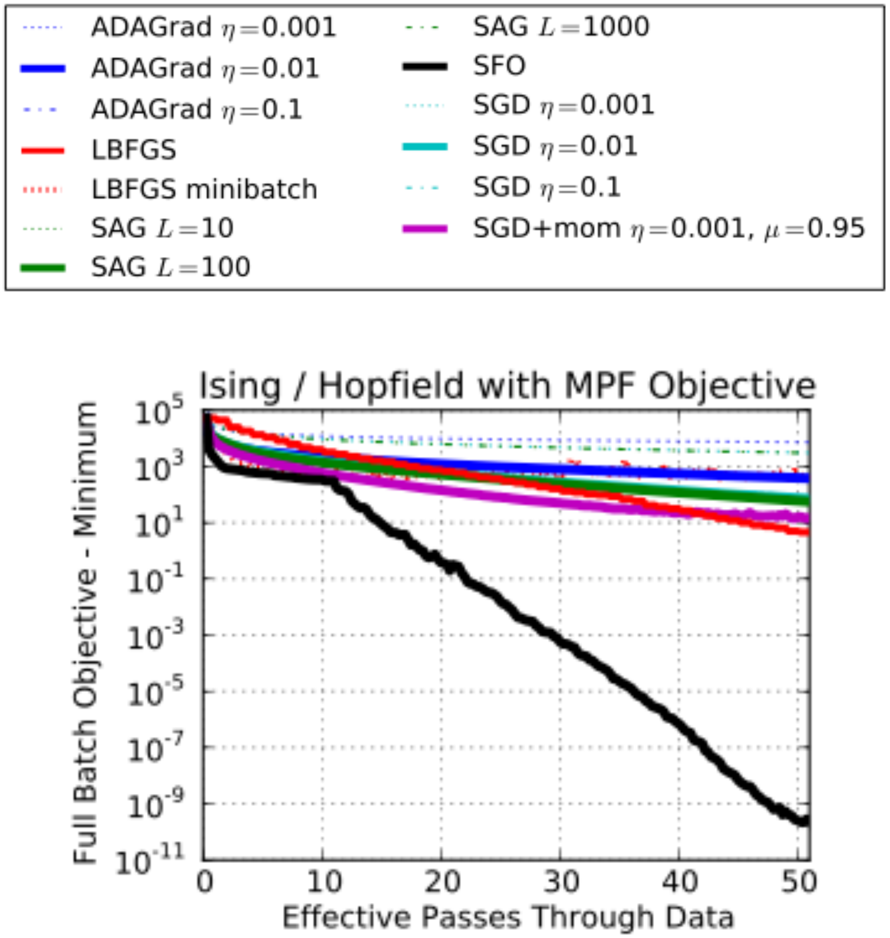
Other Research: A Useful Tool for Optimization

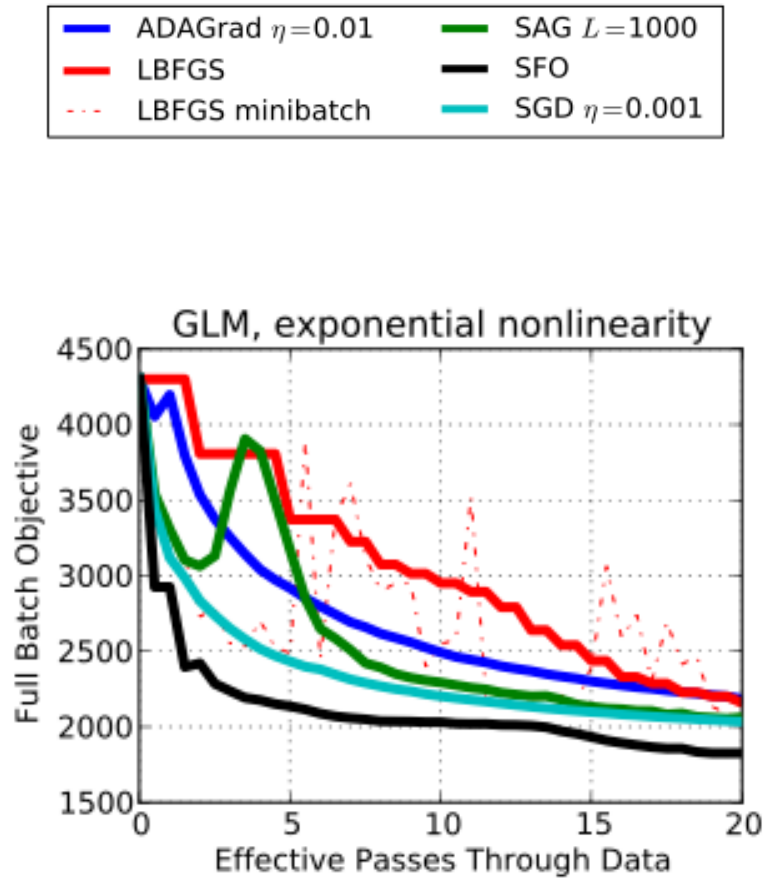
Other Research: A Useful Tool for Optimization

Try me: http://git.io/SFO

- Flexible tool for training functions on minibatches
- Open source Python and MATLAB packages
- No hyperparameters to tune

Optimizer Performance





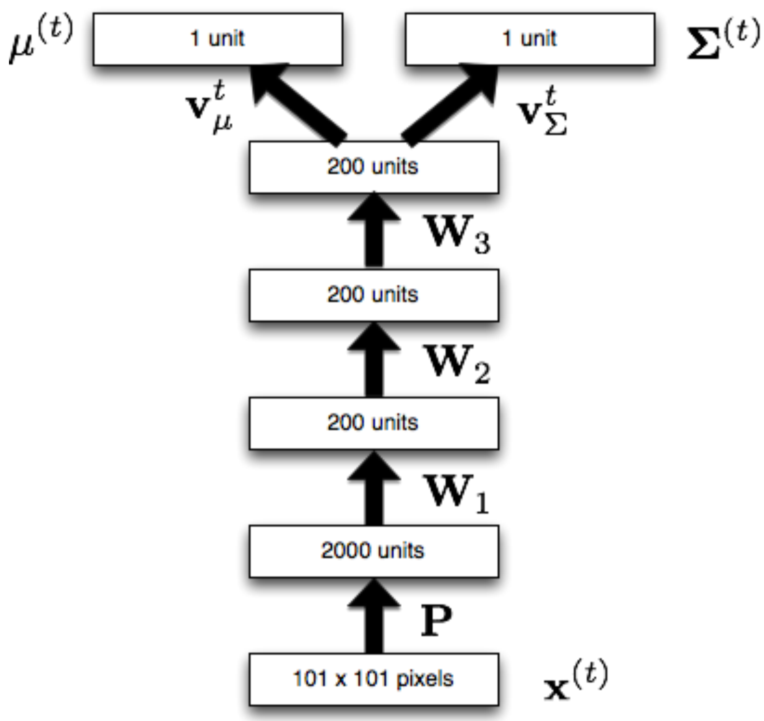
Other Research: A Useful Tool for Optimization

Try me: http://git.io/SFO

- Flexible tool for training functions on minibatches
- Open source Python and MATLAB packages
- No hyperparameters to tune

- Use multilayer neural network to estimate mean and covariance

$$p\left(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}\right) = \mathcal{N}\left(\mathbf{x}^{(t-1)}; \mu_t\left(\mathbf{x}^{(t)}\right), \Sigma_t\left(\mathbf{x}^{(t)}\right)\right)$$



Results

- Inpainting