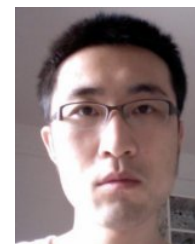


Generative Stochastic Networks Trainable by Backprop



Yoshua Bengio



**with Eric Laufer, Li Yao,
Guillaume Alain & Pascal Vincent**

RepLearn Workshop @ AAI 2013

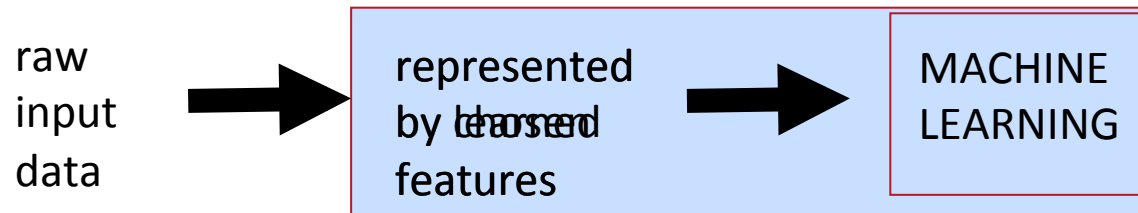
July 15th 2013, Bellevue, WA, USA

Université 
de Montréal

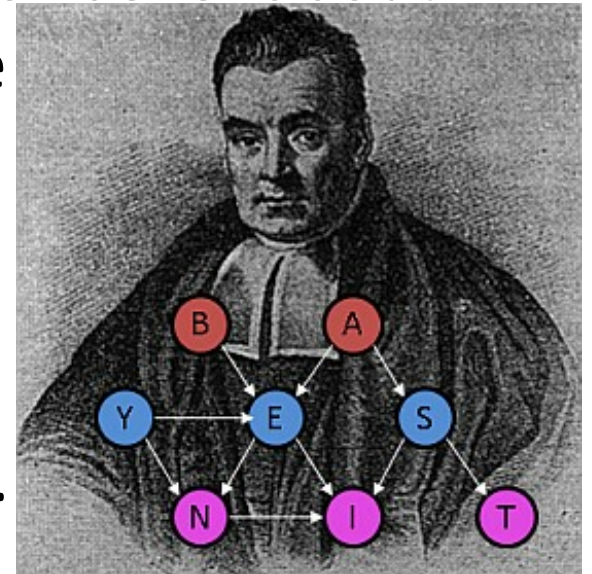


Representation Learning

- Good **features** essential for successful ML

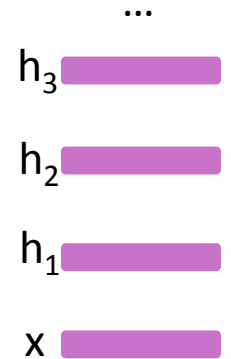


- Handcrafting features vs learning them
- Good representation: captures posterior belief about explanatory causes, disentangles these factors of variation
- Representation learning: **guesses** the features / factors / causes = good representation of observed data.



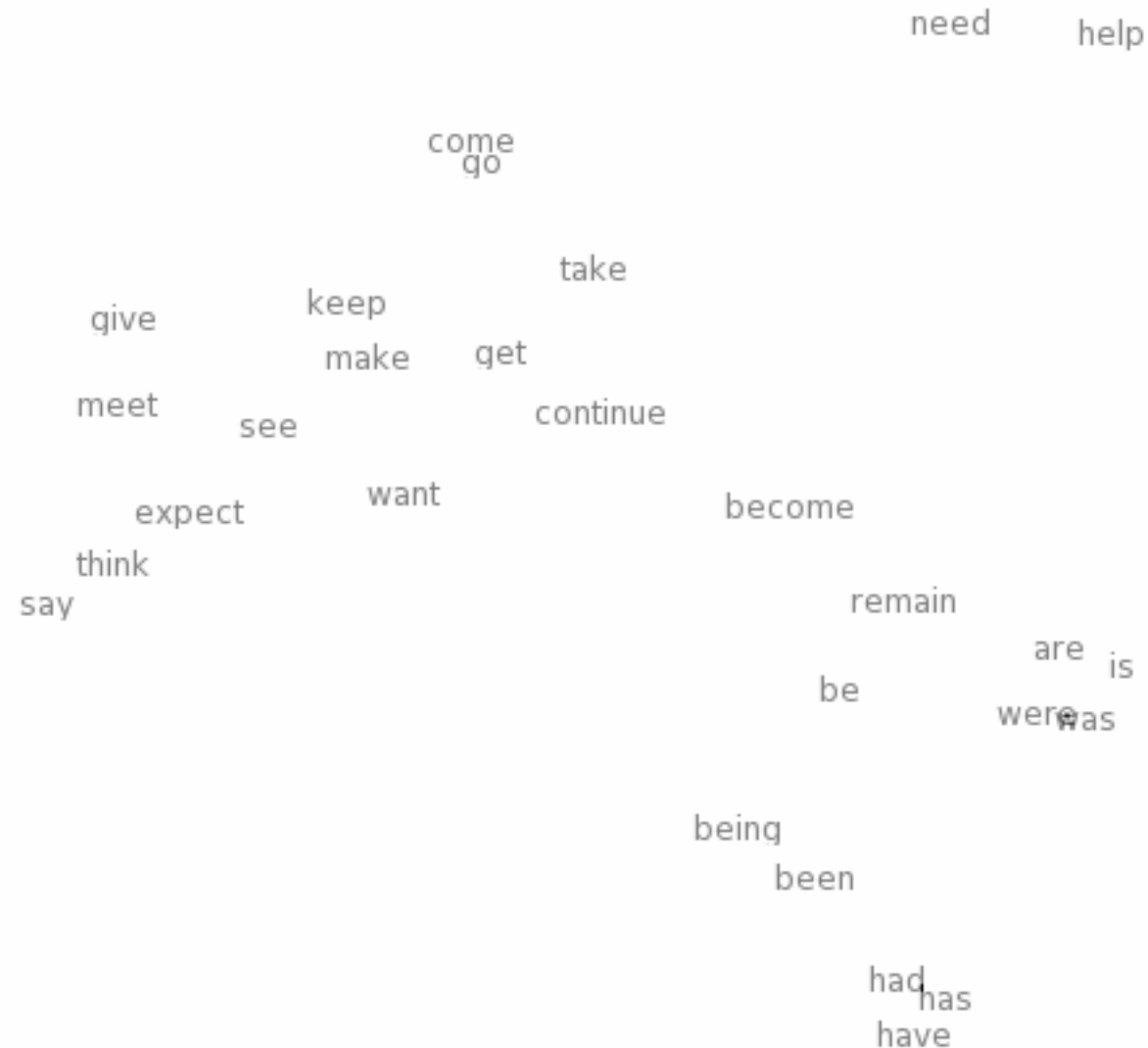
Deep Representation Learning

Learn multiple levels of representation of increasing complexity/abstraction



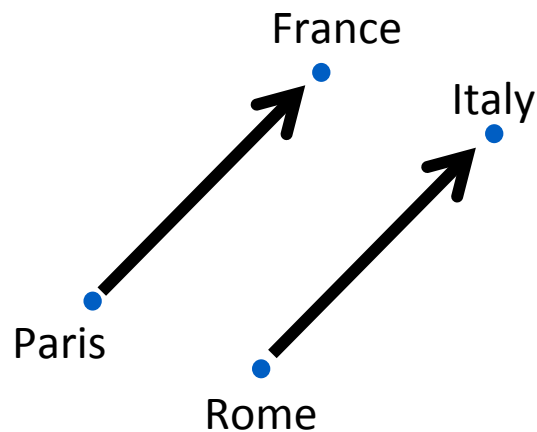
- potentially exponential gain in expressive power
- brains are deep
- humans organize knowledge in a compositional way
- **Better MCMC mixing in space of deeper representations**
(Bengio et al, ICML 2013)
- **They work! SOTA on industrial-scale AI tasks**
(object recognition, speech recognition,
language modeling, music modeling)

Following up on (Bengio et al NIPS'2000) Neural word embeddings - visualization



Analogical Representations for Free (Mikolov et al, ICLR 2013)

- Semantic relations appear as linear relationships in the space of learned representations
- King – Queen \approx Man – Woman
- Paris – France + Italy \approx Rome

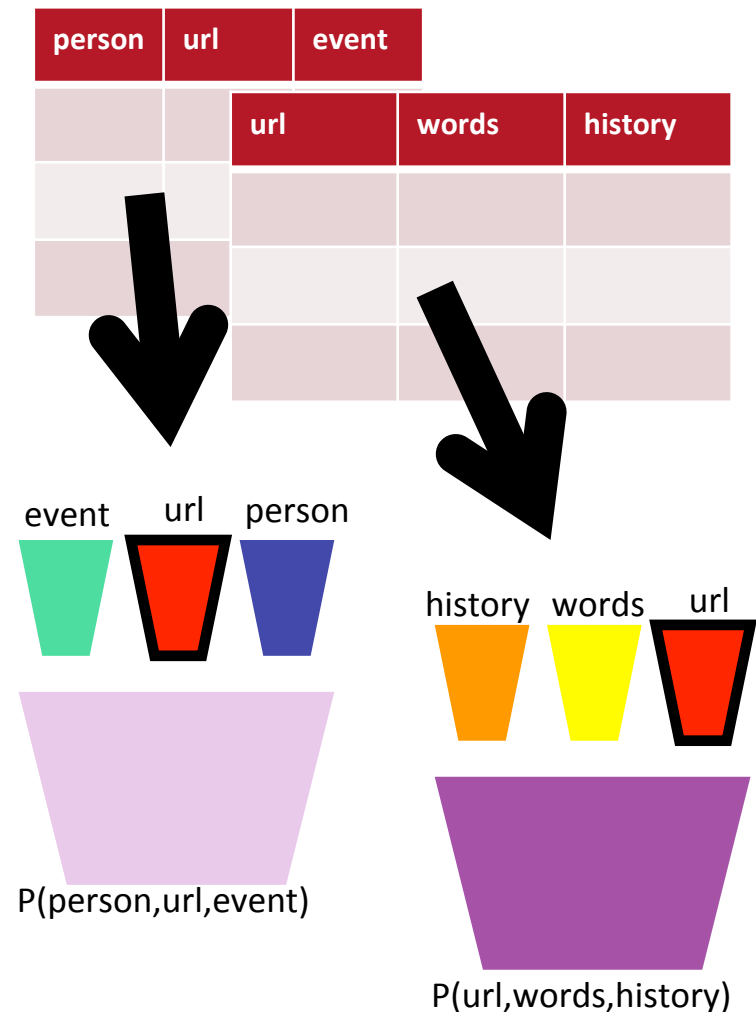


Combining Multiple Sources of Evidence with Shared Representations

- Traditional ML: data = matrix
- Relational learning: multiple sources, different tuples of variables
- Share representations of same types across data sources
- Shared learned representations help propagate information among data sources: e.g., WordNet, XWN, Wikipedia, **FreeBase**, ImageNet...
(Bordes et al AISTATS 2012, ML J. 2013)

- **FACTS = DATA**

- **Deduction = Generalization**



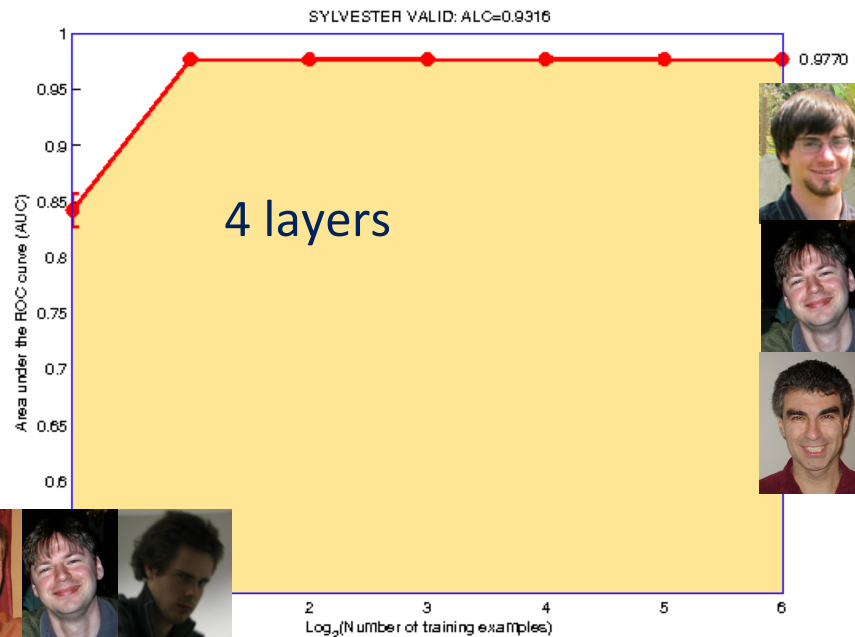
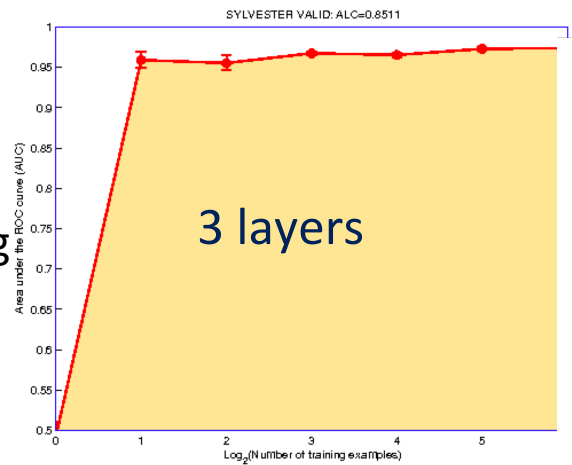
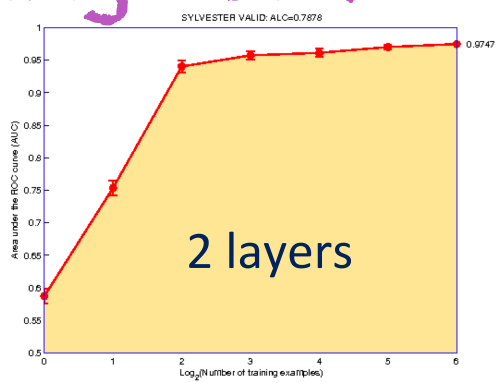
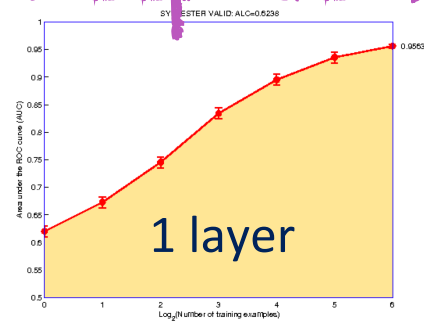
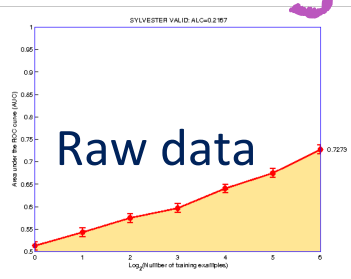
Temporal Coherence and Scales

- Hints from nature about different explanatory factors:
 - Rapidly changing factors (often noise)
 - Slowly changing (generally more abstract)
 - Different factors at different time scales
- Exploit those **hints** to **disentangle** better!
- (Becker & Hinton 1993, Wiskott & Sejnowski 2002, Hurri & Hyvarinen 2003, Berkes & Wiskott 2005, Mobahi et al 2009, Bergstra & Bengio 2009)

How do humans generalize from very few examples?

- They **transfer** knowledge from previous learning:
 - Representations
 - Explanatory factors
- Previous learning from: unlabeled data
 - + labels for other tasks
- **Prior: shared underlying explanatory factors, in particular between $P(x)$ and $P(Y|x)$**
- **→ Need good unsupervised learning of representations**

Unsupervised and Transfer Learning Challenge + Transfer Learning Challenge: Deep Learning 1st Place



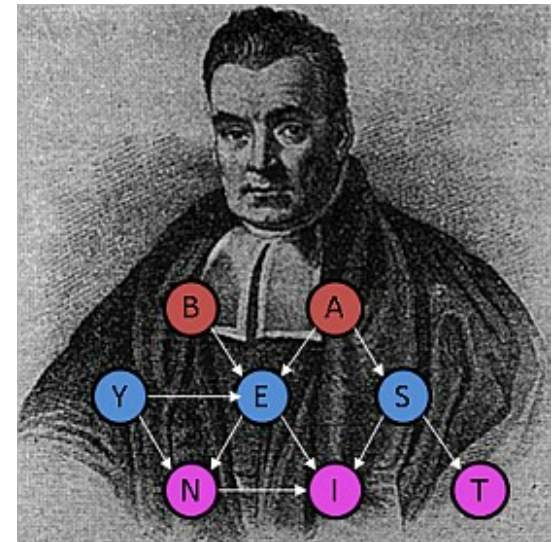
NIPS'2011
Transfer Learning
Challenge
Paper:
ICML'2012

ICML'2011
workshop on
Unsup. &
Transfer Learning



Latent Variables Love-Hate Relationship

- GOOD! **Appealing**: model explanatory factors h
- BAD! Exact inference? Nope. Just **Pain**.
too many possible configurations of h
- WORSE! Learning usually requires inference and/or sampling from $P(h, x)$



Anonymous Latent Variables

- *No pre-assigned semantics*
- Learning **discovers** underlying factors,
e.g., PCA discovers leading directions of variations
- Increases expressiveness of $P(\mathbf{x}) = \sum_h P(\mathbf{x}, h)$
- Universal approximators, e.g. for RBMs
(Le Roux & Bengio, Neural Comp. 2008)

Deep Probabilistic Models

- Linear factor models (sparse coding, PCA, ICA) - shallow
- Restricted Boltzmann Machines (**RBM**s) many variants – shallow
 - Energy(\mathbf{x}, \mathbf{h}) = $-\mathbf{h}' \mathbf{W} \mathbf{x}$
- Deep Belief Nets (**DBN**)
 - $P(\mathbf{x}, \mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3) = P(\mathbf{x} | \mathbf{h}_1) P(\mathbf{h}_1 | \mathbf{h}_2) P(\mathbf{h}_2, \mathbf{h}_3),$
where $P(\mathbf{h}_2, \mathbf{h}_3) = \text{RBM}$, conditionals = sigmoid+affine
- Deep Boltzmann Machines (**DBM**)
 - Energy($\mathbf{x}, \mathbf{h}_1, \mathbf{h}_2, \dots$) = $-\mathbf{h}_1' \mathbf{W}_1 \mathbf{x} - \mathbf{h}_2' \mathbf{W}_2 \mathbf{h}_1 - \dots$

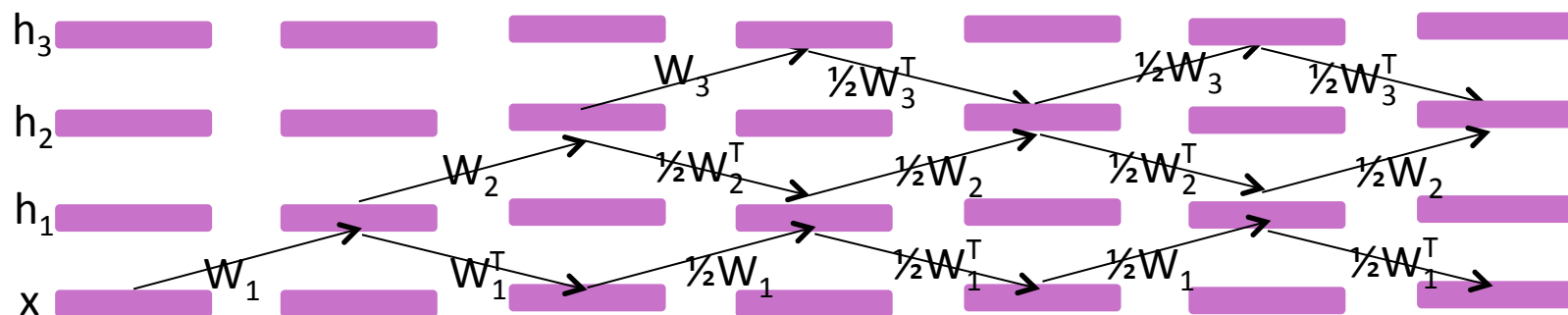


Stack of RBMs

→ Deep Boltzmann Machine

(Salakhutdinov & Hinton AISTATS 2009)

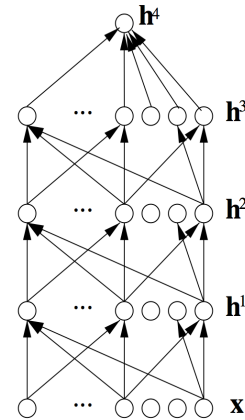
- Halve the RBM weights because each layer now has inputs from below and from above
- Positive phase: (mean-field) variational inference = recurrent AE
- Negative phase: Gibbs sampling (stochastic units)
- train by SML/PCD



Approximate Inference

- **MAP**
 - $h^* \cong \operatorname{argmax}_h P(h|\mathbf{x}) \rightarrow$ assume 1 dominant mode
- **Variational**
 - Look for tractable $Q(h)$ minimizing $KL(Q(.)||P(.|\mathbf{x}))$
 - Q is either factorial or tree-structured
 - \rightarrow strong assumption
- **MCMC**
 - Setup Markov chain asymptotically sampling from $P(h|\mathbf{x})$
 - Approx. marginalization through MC avg over few samples
 - \rightarrow assume a few dominant modes
- *Approximate inference can seriously hurt learning*
(Kulesza & Pereira NIPS'2007)

Computational Graphs



- Operations for particular task
- Neural nets' structure = computational graph for $P(y|\mathbf{x})$
- Graphical model's structure \neq computational graph for inference
- Recurrent nets & graphical models

➔ family of computational graphs sharing parameters

- *Could we have a parametrized family of computational graphs defining "the model"?*

Learned Approximate Inference

1. *Construct a computational graph corresponding to inference*
 - Loopy belief prop. (Ross et al CVPR 2011, Stoyanov et al 2011)
 - Variational mean-field (Goodfellow et al, ICLR 2013)
 - MAP (Kavukcuoglu et al 2008, Gregor & LeCun ICML 2010)
2. *Optimize parameters wrt criterion of interest, possibly decoupling from the generative model's parameters*

Learning can compensate for the inadequacy of approximate inference, taking advantage of specifics of the data distribution

THE PROBLEM

Potentially **Huge** Number of Modes in the Posterior $P(h|x)$

- Foreign speech example, y =answer to question:
 - 10 word segments
 - 100 plausible candidates per word
 - 10^6 possible segmentations
 - Most configurations (999999/1000000) implausible
 - → 10^{20} high-probability modes
- **All known approximate inference scheme break down if the posterior has a huge number of modes (fails MAP & MCMC) and not respecting a variational approximation (fails variational)**

THE SOLUTION

~~• Approximate inference~~

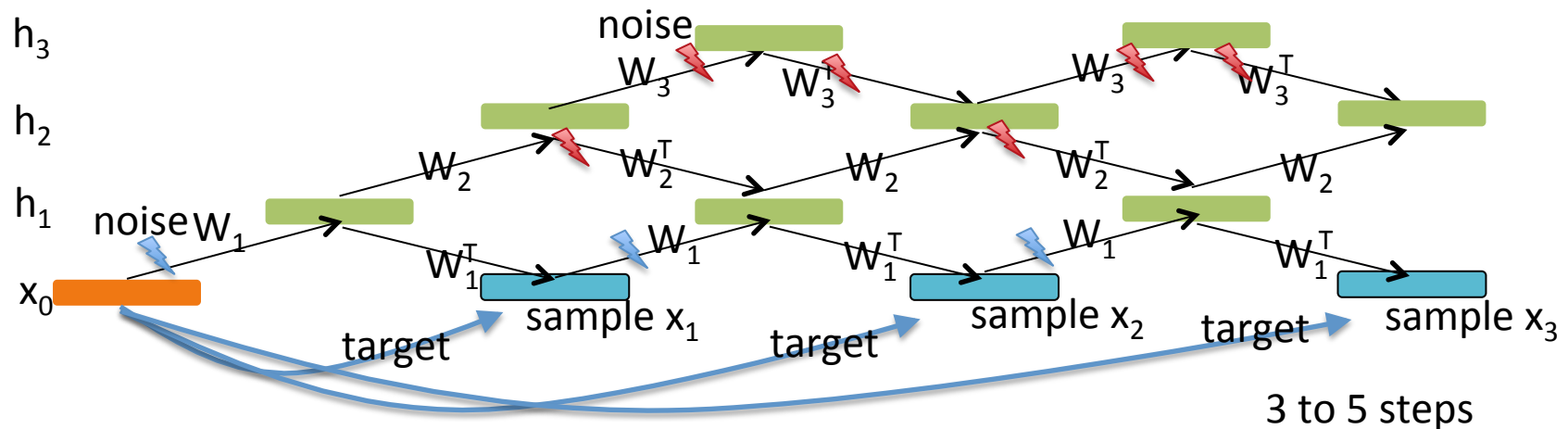
• Function approximation

Hint

- Deep neural nets learn good $P(y|\mathbf{x})$ classifiers even if there are potentially many true latent variables involved
- Exploits structure in $P(y|\mathbf{x})$ that persist even after summing h
- But how do we generalize this idea to full joint-distribution learning and answering any question about these variables, not just one?

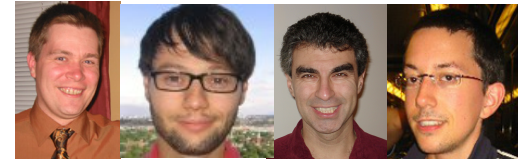
Generative Stochastic Networks (GSN)

- Recurrent parametrized stochastic computational graph that defines a transition operator for a Markov chain whose asymptotic distribution is implicitly estimated by the model
- Noise injected in input and hidden layers
- Trained to max. reconstruction prob. of example at each step
- **Example** structure inspired from the DBM Gibbs chain:

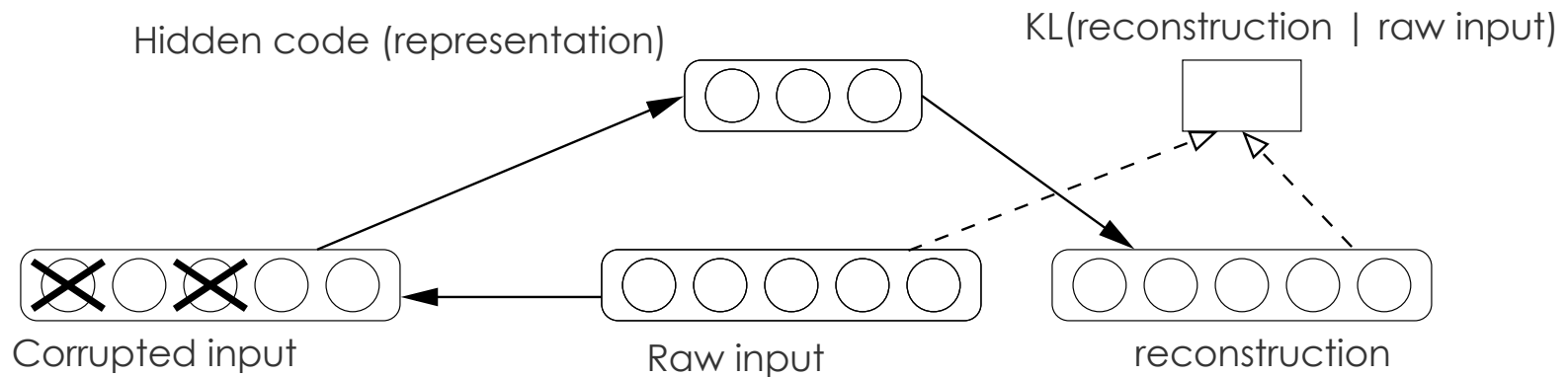


Denoising Auto-Encoder

(Vincent et al 2008)



- Corrupt the input during training only
- Train to reconstruct the uncorrupted input



- Encoder & decoder: any parametrization
- As good or better than RBMs for unsupervised pre-training

Denoising Auto-Encoder

- Learns a vector field pointing towards higher probability direction (Alain & Bengio 2013)



$$r(x)-x \propto d\log p(x)/dx$$

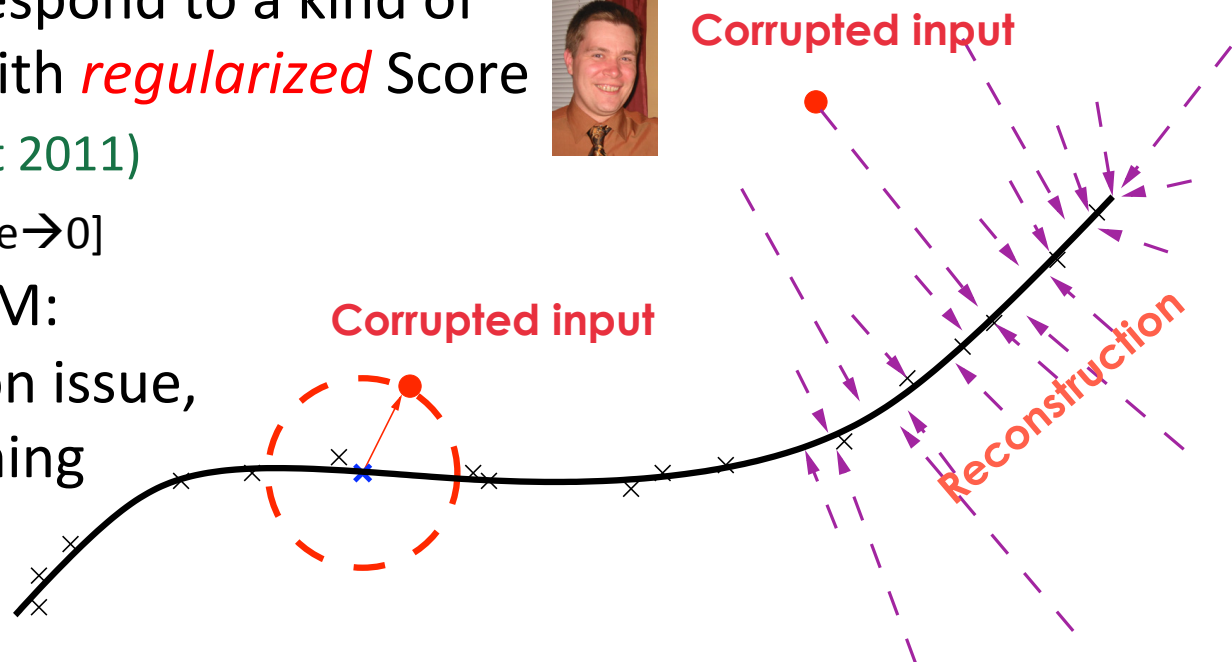
- Some DAEs correspond to a kind of Gaussian RBM with *regularized* Score Matching (Vincent 2011)



[equivalent when noise $\rightarrow 0$]

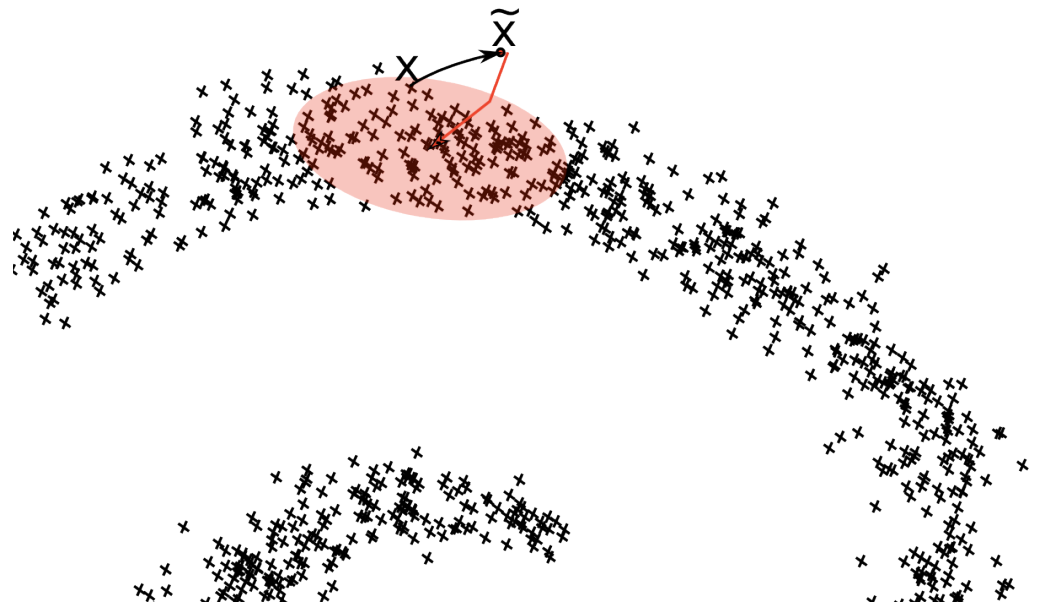
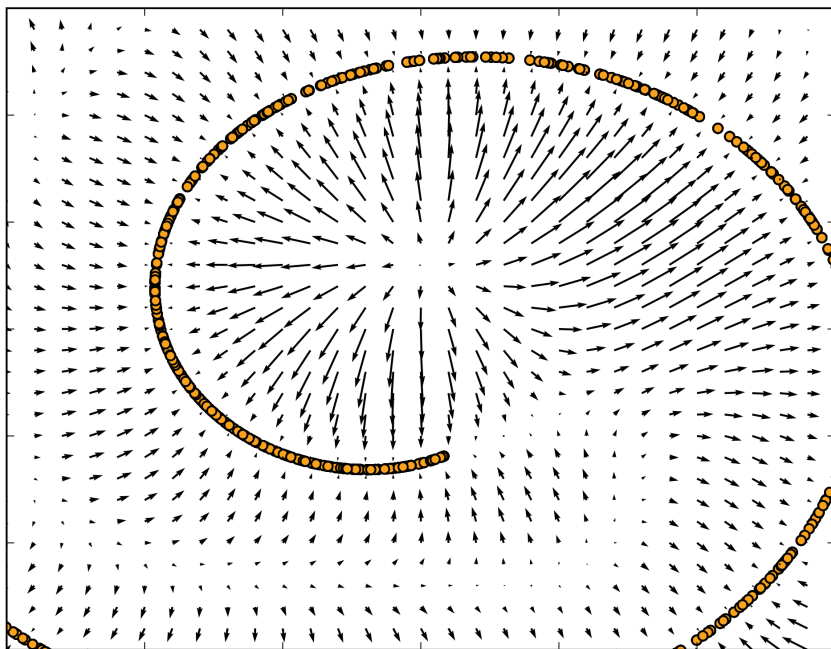
- Compared to RBM:
No partition function issue,
+ can measure training
criterion

prior: examples concentrate near a lower dimensional "manifold"



Regularized Auto-Encoders Learn a Vector Field or a Markov Chain Transition Distribution

- (Bengio, Vincent & Courville, TPAMI 2013) review paper
- (Alain & Bengio ICLR 2013; Bengio et al, arxiv 2013)



Previous Theoretical Results

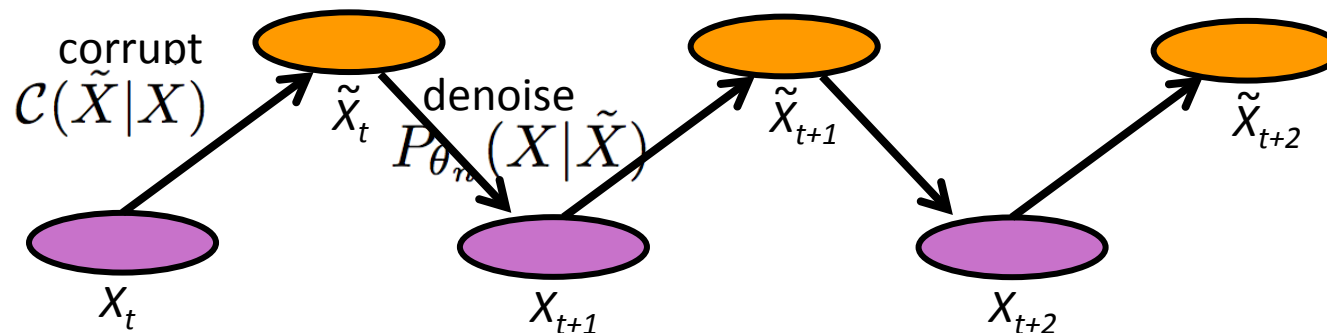
(Vincent 2011, Alain & Bengio 2013)

- Continuous X
- Gaussian corruption
- Noise $\sigma \rightarrow 0$
- Squared reconstruction error $\|r(X+\text{noise})-X\|^2$

$(r(X)-X)/\sigma^2$ estimates the score $d \log p(X) / dX$

Denoising Auto-Encoder Markov Chain

- $\mathcal{P}(X)$: true data-generating distribution
- $\mathcal{C}(\tilde{X}|X)$: corruption process
- $P_{\theta_n}(X|\tilde{X})$: denoising auto-encoder trained with n examples X, \tilde{X} from $\mathcal{C}(\tilde{X}|X)\mathcal{P}(X)$, probabilistically “inverts” corruption
- T_n : Markov chain over X alternating $\tilde{X} \sim \mathcal{C}(\tilde{X}|X)$, $X \sim P_{\theta_n}(X|\tilde{X})$



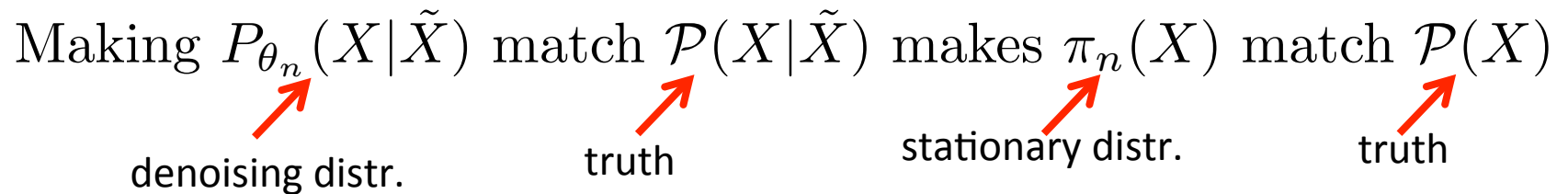
New Theoretical Results: Denoising AE

- Denoising AE are consistent estimators of the data-generating distribution through their Markov chain, so long as they consistently estimate the conditional denoising distribution and the Markov chain converges.

Theorem 1. *If $P_{\theta_n}(X|\tilde{X})$ is a consistent estimator of the true conditional distribution $\mathcal{P}(X|\tilde{X})$ and T_n defines an irreducible and ergodic Markov chain, then as $n \rightarrow \infty$, the asymptotic distribution $\pi_n(X)$ of the generated samples converges to the data generating distribution $\mathcal{P}(X)$.*

Making $P_{\theta_n}(X|\tilde{X})$ match $\mathcal{P}(X|\tilde{X})$ makes $\pi_n(X)$ match $\mathcal{P}(X)$

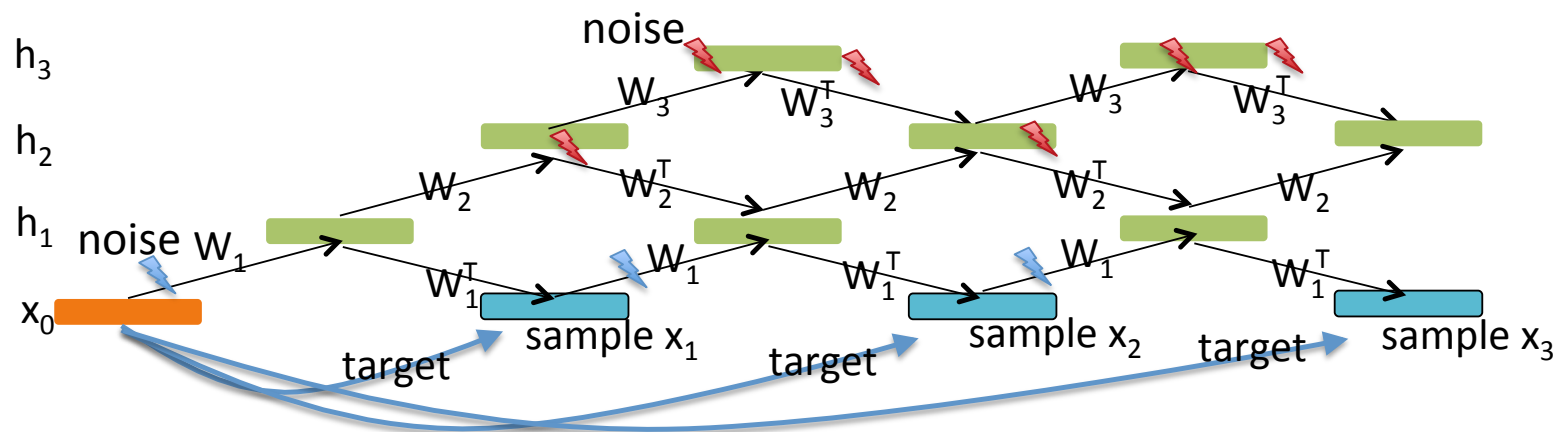
denoising distr. truth stationary distr. truth



Generative Stochastic Networks (GSN)

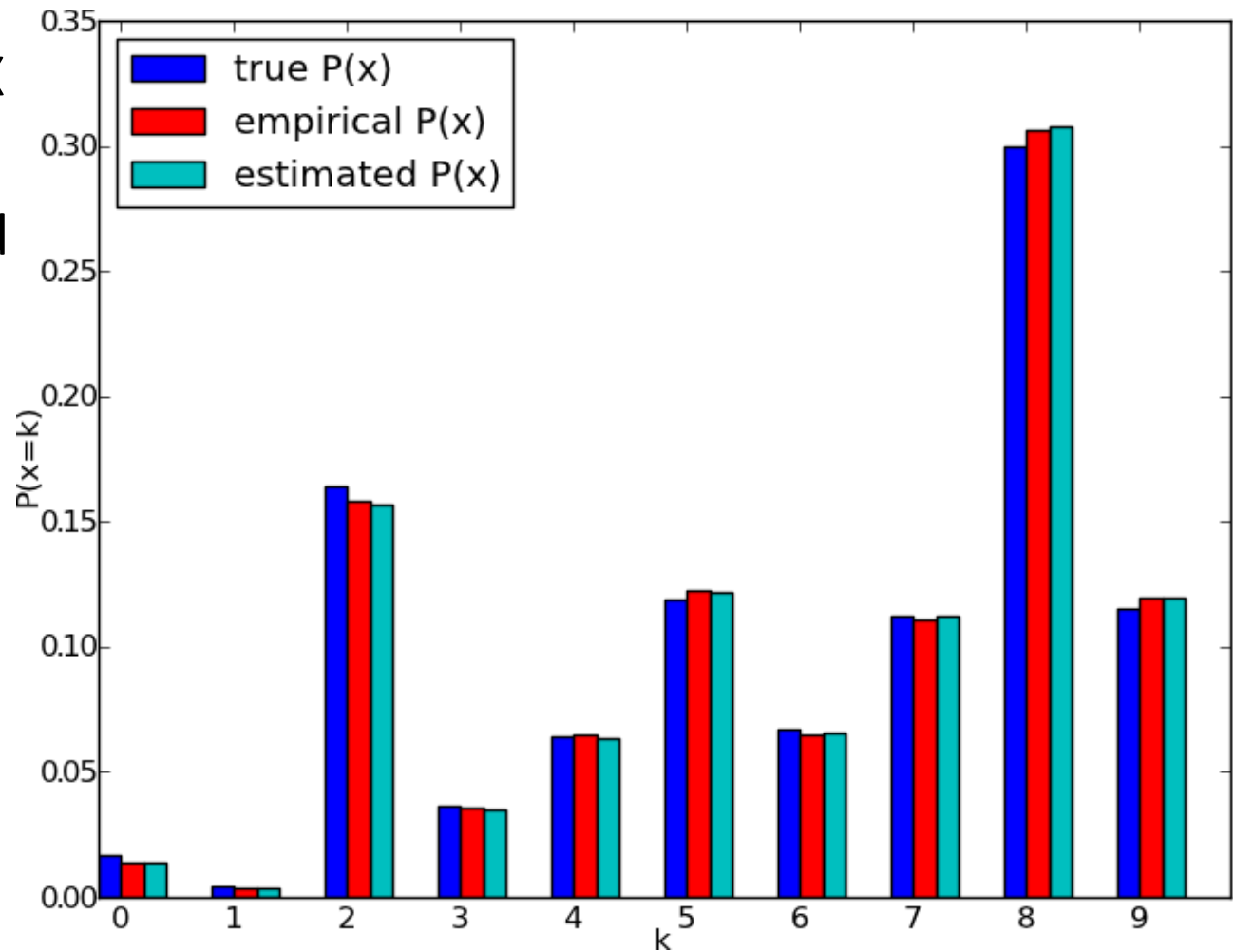
- If we decompose the reconstruction probability into a parametrized noise-dependent part $\tilde{X} = f_{\theta_1}(X, Z)$ and a noise-independent part $P_{\theta_2}(X|\tilde{X})$, we also get a consistent estimator of the data generating distribution, if the chain converges.

Corollary 2. Let training data $X \sim \mathcal{P}(X)$ and independent noise $Z \sim \mathcal{P}(Z)$. Consider a model $P_{\theta_2}(X|f_{\theta_1}(X, Z))$ trained (over both θ_1 and θ_2) by regularized conditional maximum likelihood with n examples of (X, Z) pairs. For a given θ_1 , a random variable $\tilde{X} = f_{\theta_1}(X, Z)$ is defined. Assume that as n increases, P_{θ_2} is a consistent estimator of the true $\mathcal{P}(X|\tilde{X})$. Assume also that the Markov chain $X_t \sim P_{\theta_2}(X|f_{\theta_1}(X_{t-1}, Z_{t-1}))$ (where $Z_{t-1} \sim \mathcal{P}(Z)$) converges to a distribution π_n , even in the limit as $n \rightarrow \infty$. Then $\pi_n(X) \rightarrow \mathcal{P}(X)$ as $n \rightarrow \infty$.



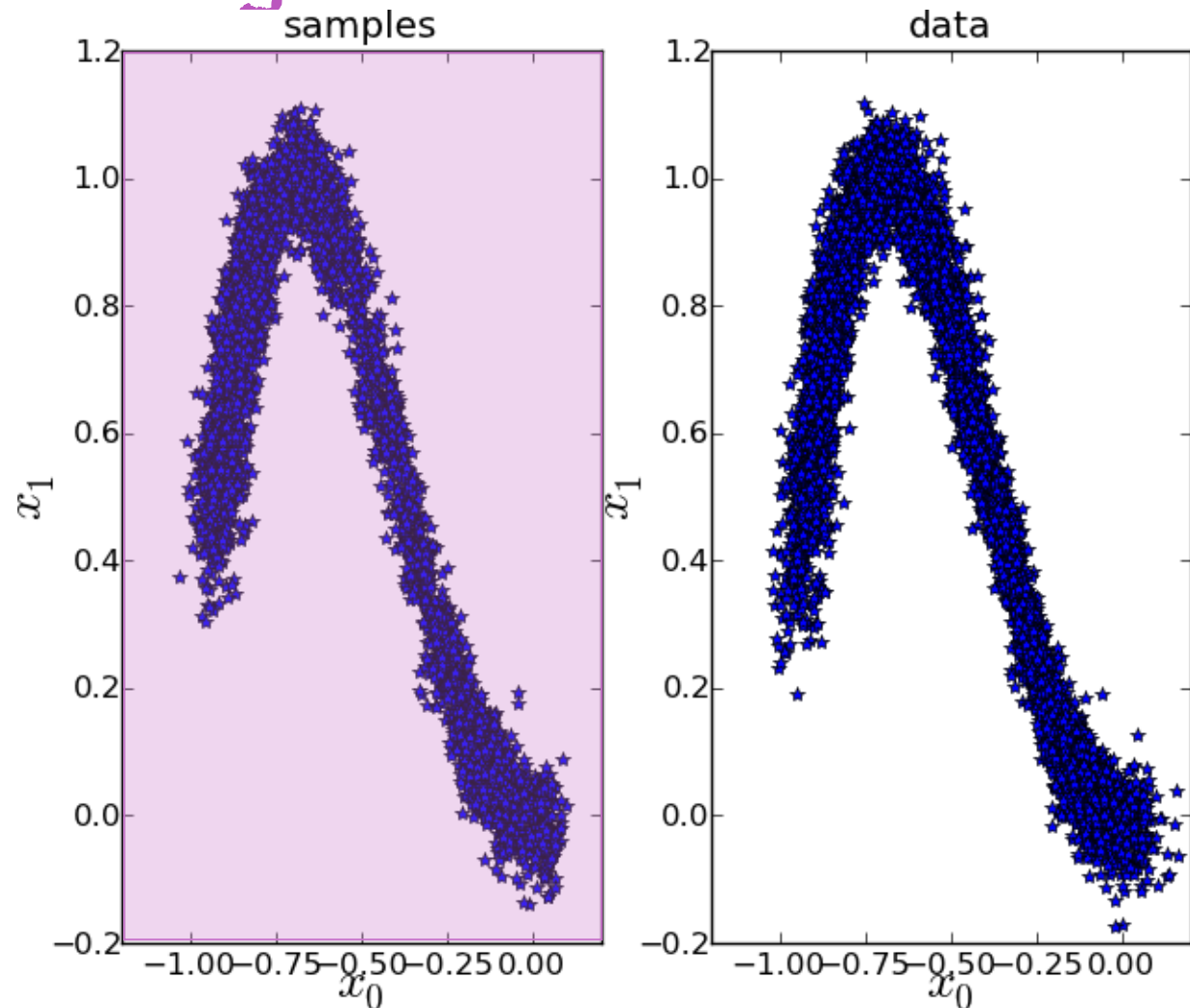
GSN Experiments: validating the theorem in a discrete non-parametric setting

- Discrete data, X in $\{0, \dots, 9\}$
- Corruption: add +/- small int.
- Reconstruction distribution = maximum likelihood estimator (counting)

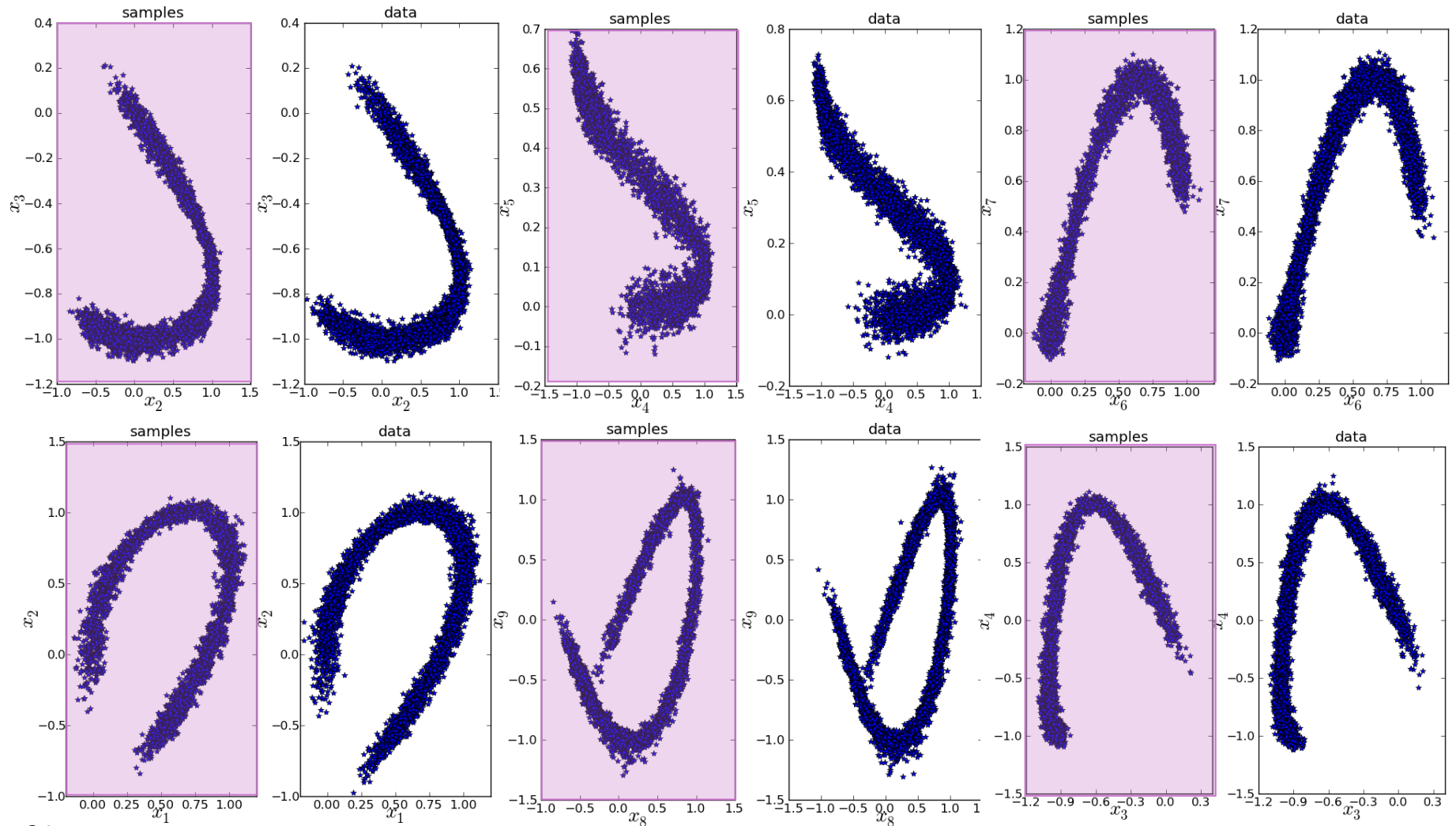


GSN Experiments: validating the theorem in a continuous non-parametric setting

- Continuous data, X in R^{10} , Gaussian corruption
- Reconstruction distribution = Parzen (mixture of Gaussians) estimator
- 5000 training examples, 5000 samples
- Visualize a pair of dimensions

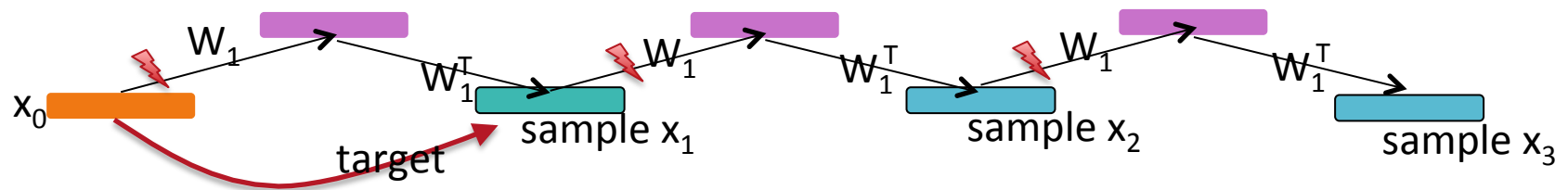


GSN Experiments: validating the theorem in a continuous non-parametric setting



Shallow Model: Generalizing the Denoising Auto-Encoder Probabilistic Interpretation

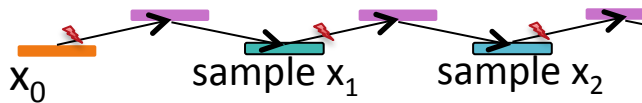
- Classical denoising auto-encoder architecture, single hidden layer with noise only injected in input
- Factored Bernoulli reconstruction prob. distr.
- $\tilde{X} = f_{\theta_1}(X, Z) =$ parameter-less, salt-and-pepper noise on top of X



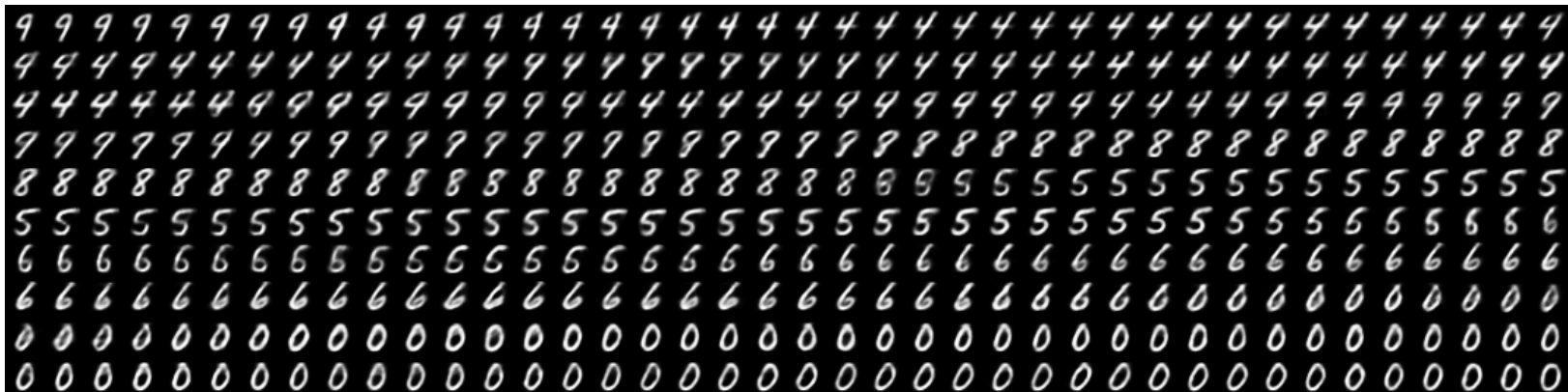
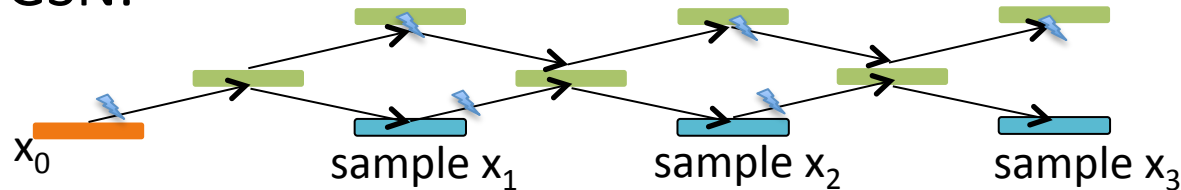
- *Generalizes (Alain & Bengio 2013): not just continuous r.v., any training criterion (as log-likelihood), not just Gaussian but any corruption (no need to be tiny to correctly estimate distribution).*

Experiments: Shallow vs Deep

- Shallow (DAE), no recurrent path at higher levels, state=X only



- Deep GSN:



Quantitative Evaluation of Samples

- Previous procedure for evaluating samples (Breuleux et al 2011, Rifai et al 2012, Bengio et al 2013):
 - Generate 10000 samples from model
 - Use them as training examples for Parzen density estimator
 - Evaluate its log-likelihood on MNIST test data

	GSN-2	DAE	RBM	DBM-3	DBN-2	MNIST
LOG-LIKELIHOOD	214	-152	-244	32	138	24
STANDARD ERROR	1.1	2.2	54	1.9	2.0	1.6

Training examples
↓

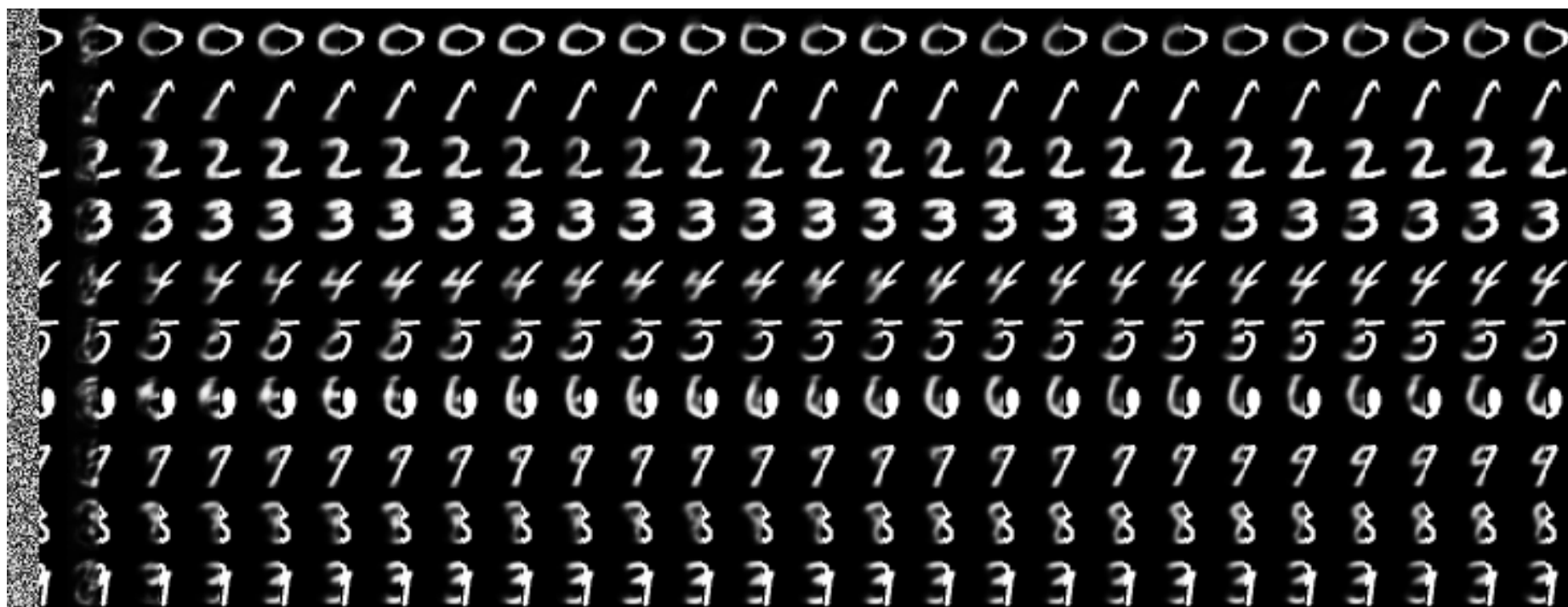
Question Answering, Missing Inputs and Structured Output

- Once trained, a GSN can sample from any conditional over subsets of its inputs, so long as we use the conditional associated with the reconstruction distribution and clamp the right-hand side variables.

Proposition 1. *If a subset $x^{(s)}$ of the elements of X is kept fixed (not resampled) while the remainder $X^{(-s)}$ is updated stochastically during the Markov chain of corollary 2, but using $P(X_{t+1}|f(X_t, Z_t), X_{t+1}^{(s)} = x^{(s)})$, then the asymptotic distribution π_n produces samples of $X^{(-s)}$ from the conditional distribution $\pi_n(X^{(-s)}|X^{(s)} = x^{(s)})$.*

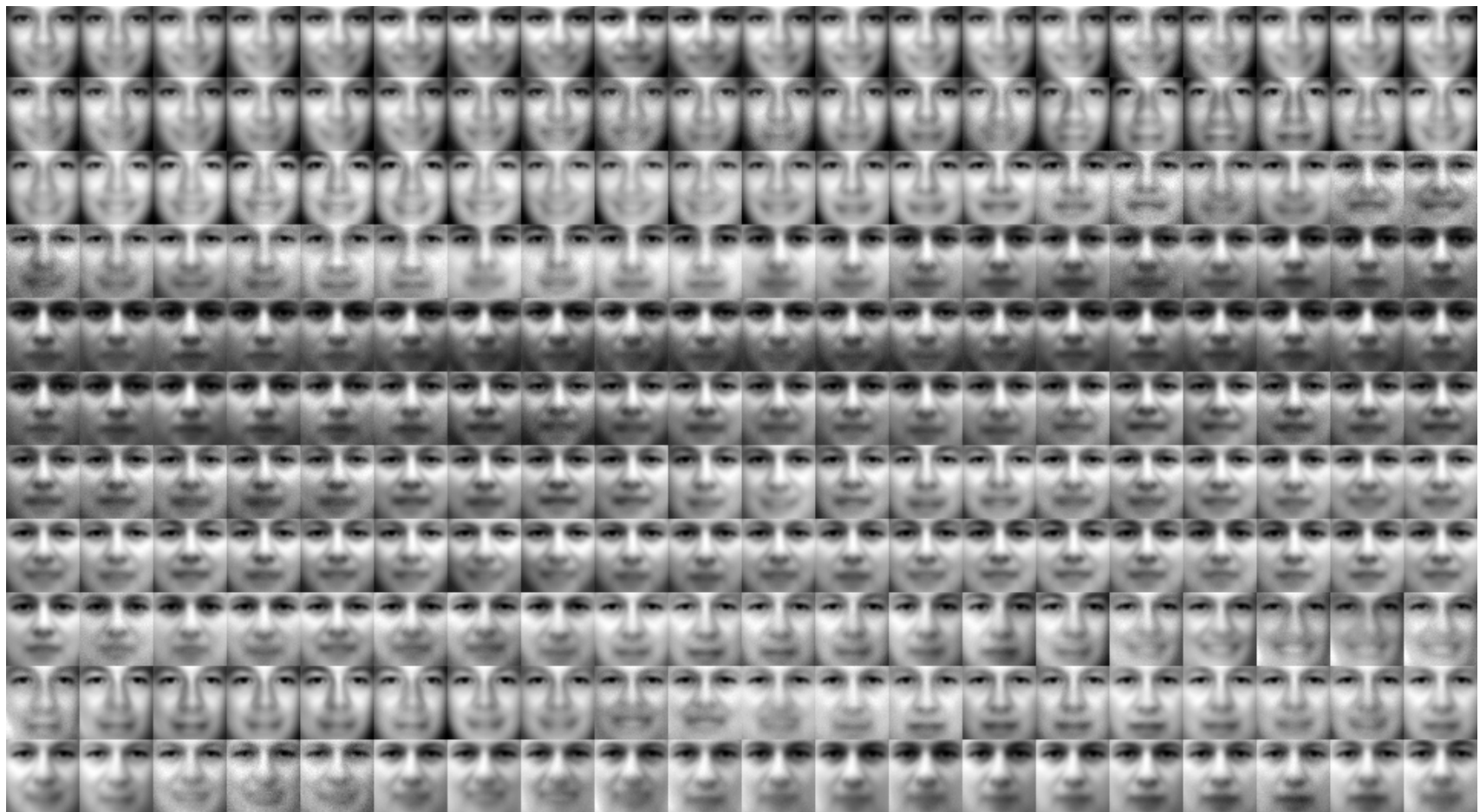
Experiments: Structured Conditionals

- Stochastically fill-in missing inputs, sampling from the chain that generates the conditional distribution of the missing inputs given the observed ones (notice the fast burn-in!)



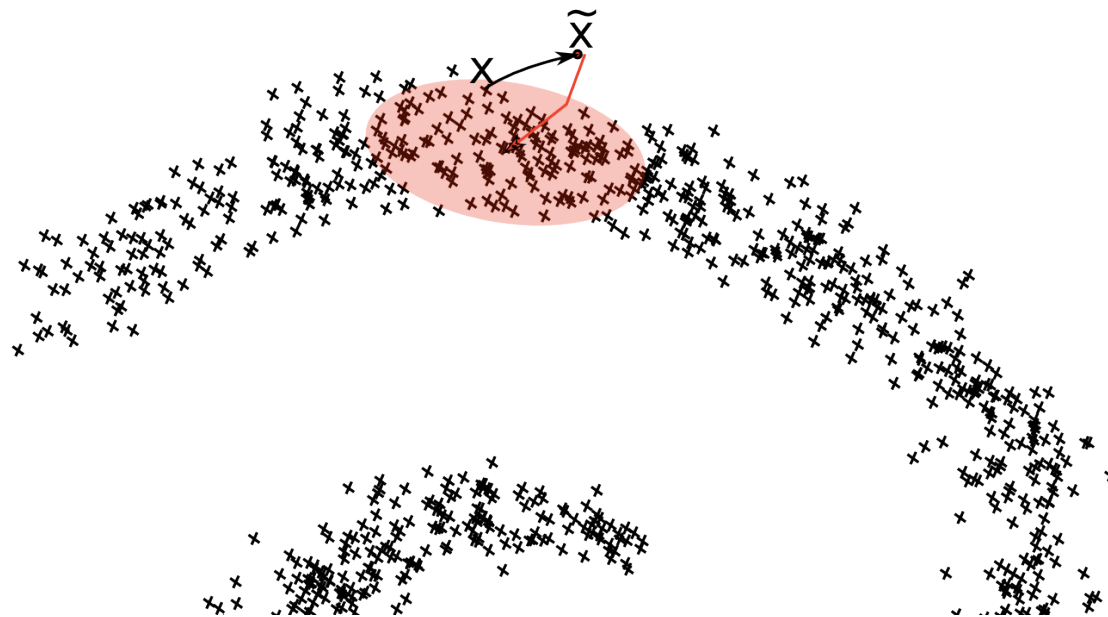
Not Just MNIST: experiments on TFD

- 3 hidden layer model, consecutive samples:



Future Work: Multi-Modal Reconstruction Distributions

- All experiments: unimodal (factorial) reconstruction distribution
- Theorems require potentially multimodal one
- In the limit of small noise, unimodal is enough (Alain & Bengio 2013)



Getting Rid of BackProp Altogether

- Some parts of the network may need to take stochastic hard decisions, can't do backprop
- Discovered an unbiased estimator of the loss gradient wrt to binary stochastic units

$$h_i = f(a_i, z_i) = \mathbf{1}_{z_i > \text{sigm}(a_i)}$$

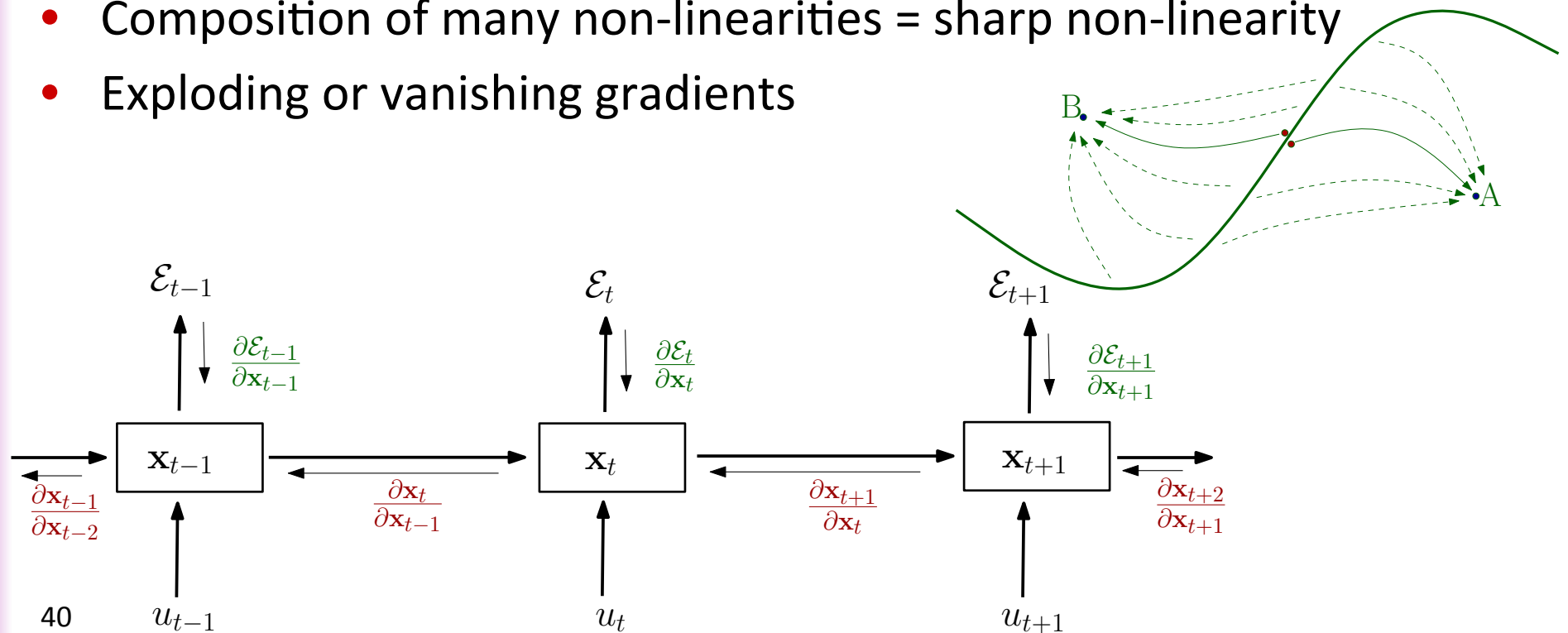
- $\hat{g}_i = (h_i - \text{sigm}(a_i)) \times L$

is an unbiased estimator of the gradient of expectation of L wrt a_i

- A lower variance variant has been demonstrated to learn (NIPS 2013 submission), albeit slower than backprop.
- Hinton also has a proposal for approximating gradient backprop through feedback connections, which could be combined w/ this

The Optimization Challenge in Deep / Recurrent Nets

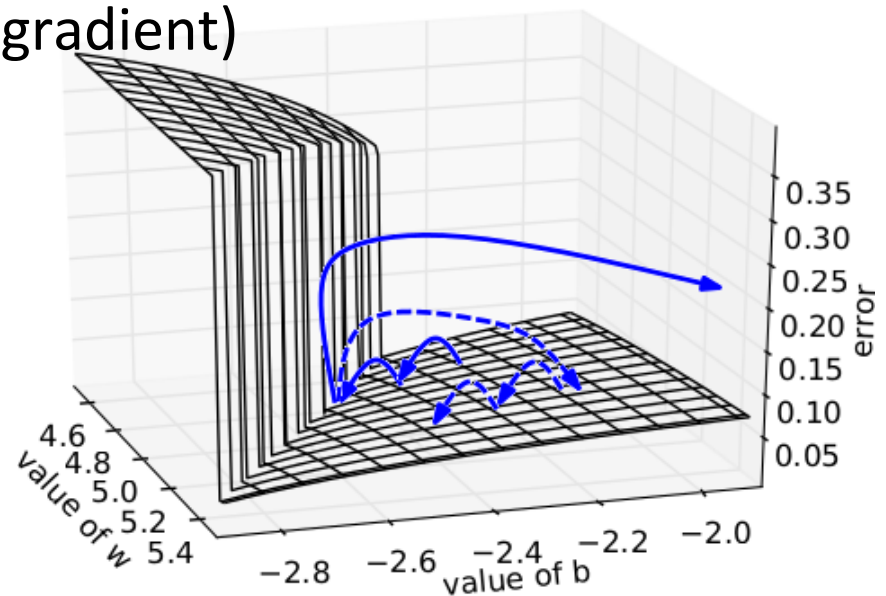
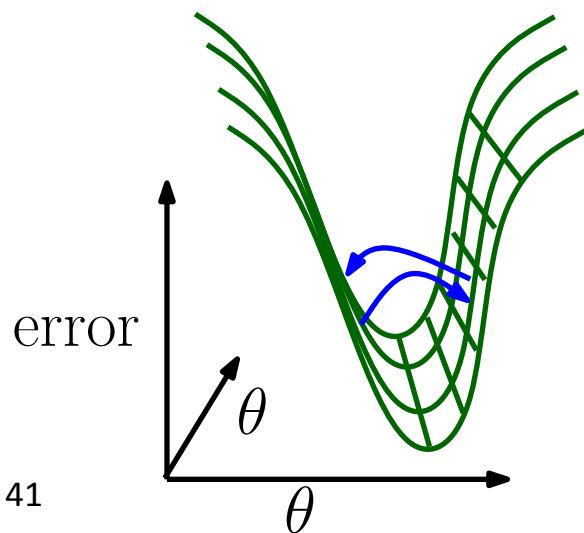
- Higher-level abstractions require highly non-linear transformations to be learned
- Sharp non-linearities are difficult to learn by gradient
- Composition of many non-linearities = sharp non-linearity
- Exploding or vanishing gradients



RNN Tricks

(Pascanu, Mikolov, Bengio, ICML 2013; Bengio, Boulanger & Pascanu, ICASSP 2013)

- Clipping gradients (avoid exploding gradients)
- Leaky integration (propagate long-term dependencies)
- Momentum (cheap 2nd order)
- Initialization (start in right ballpark avoids exploding/vanishing)
- Sparse Gradients (symmetry breaking)
- Gradient propagation regularizer (avoid vanishing gradient)
- LSTM self-loops (avoid vanishing gradient)



Conclusions

- Radically different approach to probabilistic unsupervised learning of generative models through learning a transition operator
- Skips the need for latent variables and approximate inference over them
- Eliminates previous limitations of probabilistic interpretations of regularized auto-encoders
- Any stochastic but smooth computational graph can be trained by **back-prop** with noise injected in the deep network (not just inputs), just like in recent dropout deep nets
- Can model joint / conditional / structured outputs / missing variables

The End

Reading material available on arxiv:

1306.1091	cs.LG	Deep Generative Stochastic Networks Trainable by Backprop
1305.6663	cs.LG	Generalized Denoising Auto-Encoders as Generative Models
1305.2982	cs.LG	Estimating or Propagating Gradients Through Stochastic Neurons
1305.0445	cs.LG	Deep Learning of Representations: Looking Forward