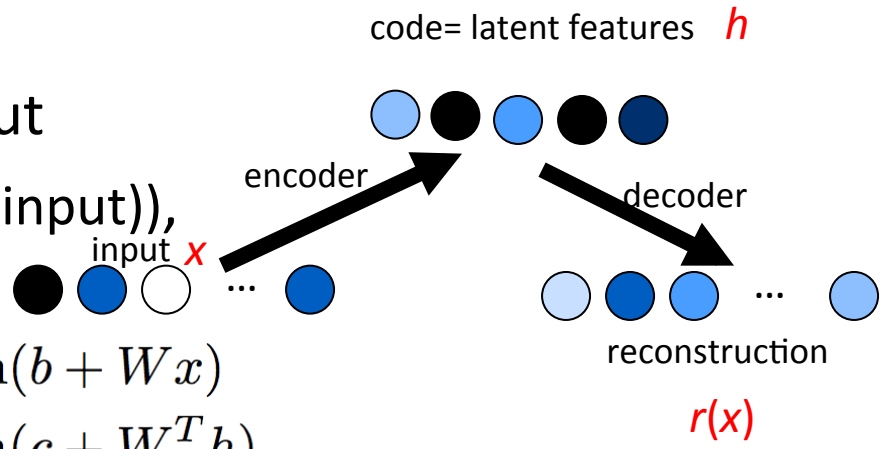
Auto-Encoders & Variants

Auto-Encoders

- MLP whose target output = input
- Reconstruction=decoder(encoder(input)), e.g.



$$
\begin{aligned}
h &= \tanh(b + Wx) \\
\text{reconstruction} &= \tanh(c + W^T h) \\
\text{Loss } L(x, \text{reconstruction}) &= ||\text{reconstruction} - x||^2
\end{aligned}
$$

- With bottleneck, code = new coordinate system
- Encoder and decoder can have 1 or more layers
- Training deep auto-encoders notoriously difficult

Link Between Contrastive Divergence and Auto-Encoder Reconstruction Error Gradient

- (Bengio & Delalleau 2009):
  - CD-2k estimates the log-likelihood gradient from 2k diminishing terms of an expansion that mimics the Gibbs steps
  - reconstruction error gradient looks only at the first step, i.e., is a kind of mean-field approximation of CD-0.5

$$\frac{\partial \log P(x_1)}{\partial \theta} = \sum_{s=1}^{t-1} \left(E\left[\frac{\partial \log P(x_s|h_s)}{\partial \theta} \middle| x_1 \right] + E\left[\frac{\partial \log P(h_s|x_{s+1})}{\partial \theta} \middle| x_1 \right] \right) + E\left[\frac{\partial \log P(x_t)}{\partial \theta} \middle| x_1 \right]$$

Traditional Directed X|θ Models

$$P(X, \theta) = P(X|\theta)P(\theta)$$

$$P(X|\theta) = \frac{e^{-E_\theta(X)}}{Z_\theta}$$

$$\frac{\partial \log Z_\theta}{\partial \theta} = -\sum_X P(X|\theta) \frac{\partial E_\theta(X)}{\partial \theta}$$

What are regularized auto-encoders learning exactly?

- Any training criterion E(X, θ) interpretable as a form of MAP:
- JEPADA: Joint Energy in PArameters and Data (Bengio, Courville, Vincent 2012)

$$P(X, \theta) = \frac{e^{-E(X,\theta)}}{Z}$$

This Z does not depend on θ. If E(X, θ) tractable, so is the gradient

No magic; consider traditional directed model:

$$E(X, \theta) = E_\theta(X) + \log Z_\theta - \log P(\theta)$$

Application: Predictive Sparse Decomposition, regularized auto-encoders, …

Joint Parameter-Data Energy (JEPADA)

- Getting rid of the partition function problem
- Sampling X given θ, even when previously there was no probabilistic interpretation to E(X, θ)
- Sampling θ given X (Bayesian)
- Inference and decision based on the model for which θ was really tuned.

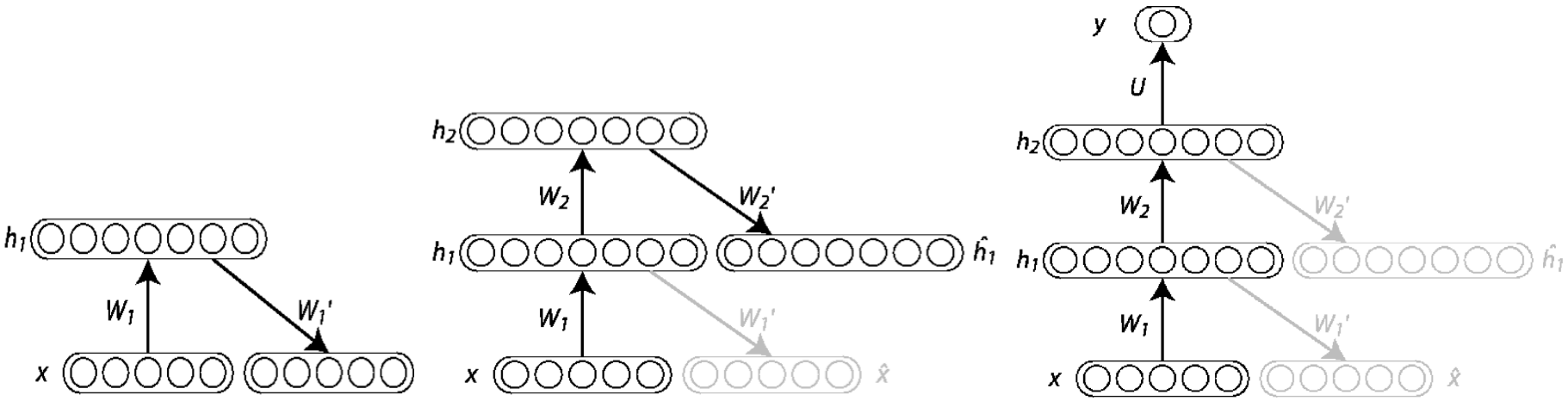
- BUT WHAT MATHEMATICAL FORMS MAKE SENSE? Reconstruction error and pseudo-likelihood-like things seem to work well. What else?

I think I finally understand what auto-encoders do!

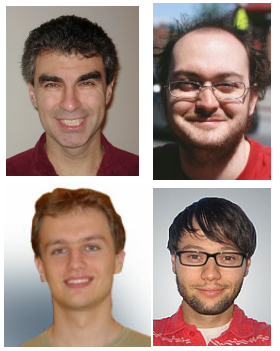
- Try to carve holes in $||r(x)-x||^2$ at training examples

- Vector $r(x)-x$ points in direction of increasing prob., i.e. estimate score = d log $p(x)$ / dx: learn **score** vector field = **local mean**
- Generalize (*valleys*) in between above holes to form *manifolds*
  - $\mathrm{d}\,r(x)/\mathrm{d}x$ estimates the **local covariance** and is linked to the Hessian $\mathrm{d}^2 \log p(x)/\mathrm{d}x^2$
- Regularized AEs estimate 1st and 2nd local moments of the density (imagine a ball around each x), which allows to **sample**

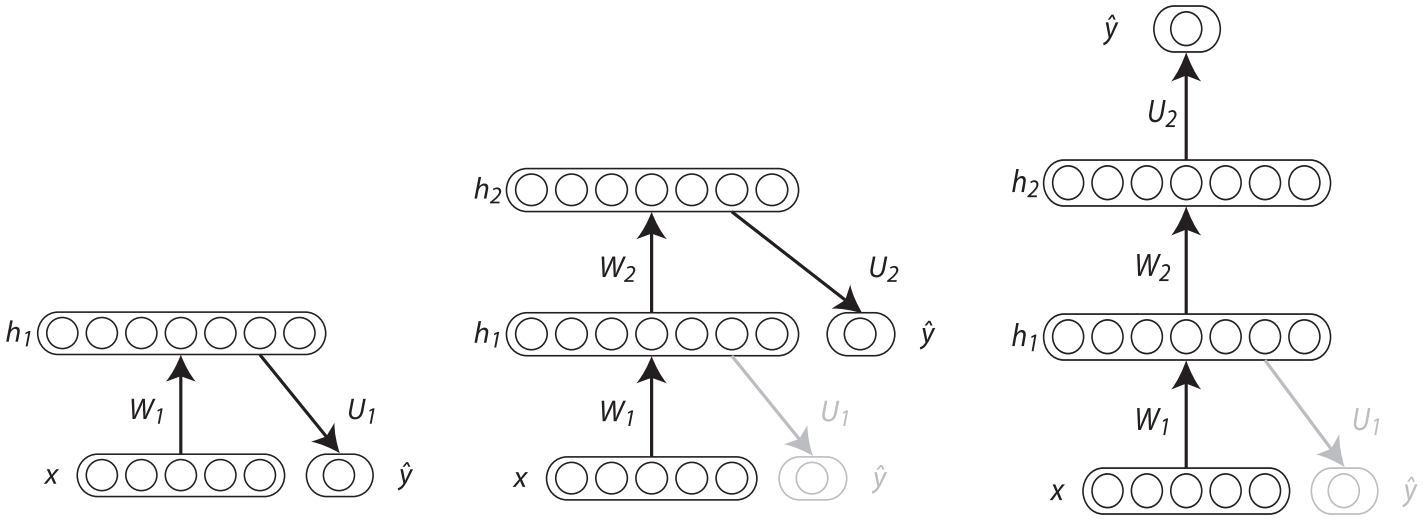
Stacking Auto-Encoders



Auto-encoders can be stacked successfully (Bengio et al NIPS'2006) to form highly non-linear representations, which with fine-tuning overperformed purely supervised MLPs



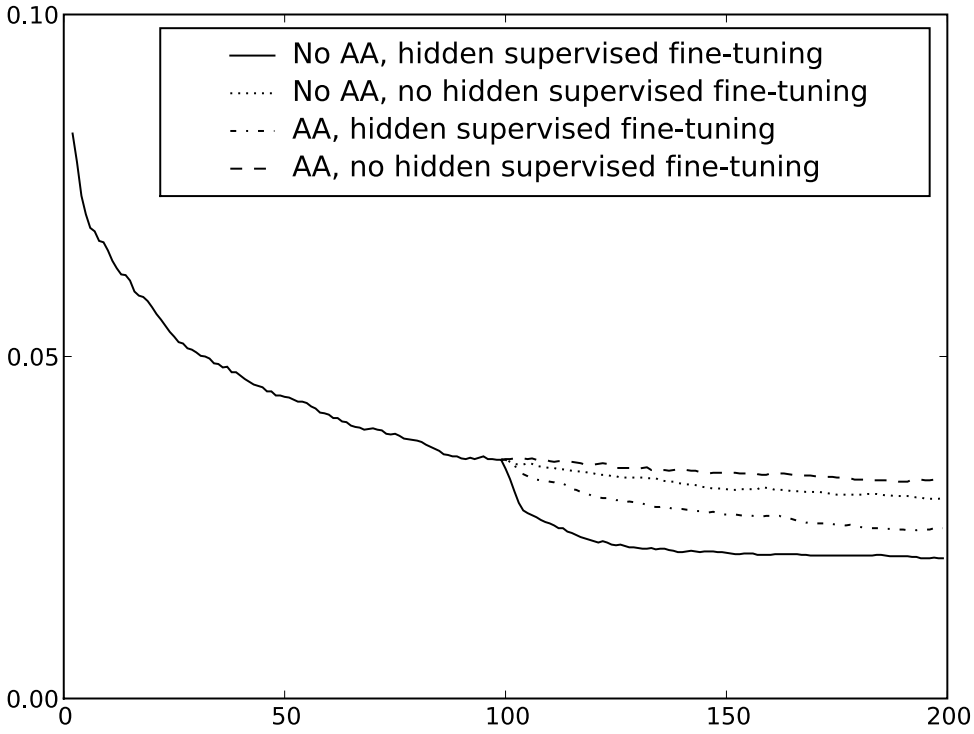
Greedy Layerwise Supervised Training



Generally worse than unsupervised pre-training but better than ordinary training of a deep neural network (Bengio et al. NIPS'2006). Has been used successfully on large labeled datasets, where unsupervised pre-training did not make as much of an impact.

Supervised Fine-Tuning is Important

- Greedy layer-wise unsupervised pre-training phase with RBMs or auto-encoders on MNIST
- Supervised phase with or without unsupervised updates, with or without fine-tuning of hidden layers
- Can train all RBMs at the same time, same results



(Auto-Encoder) Reconstruction Loss

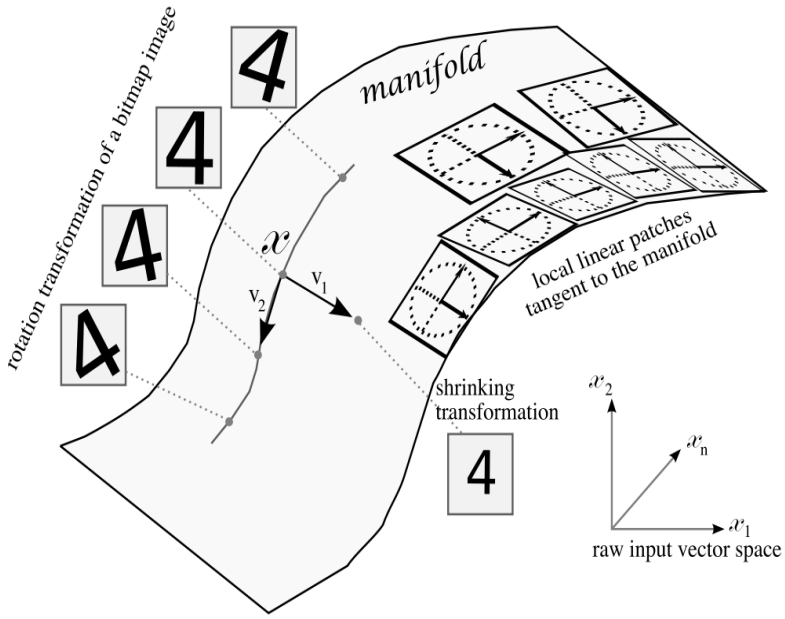
- Discrete inputs: cross-entropy for binary inputs
  - $- \Sigma_i x_i \log r_i(x) + (1-x_i) \log(1-r_i(x))$ (with $0<r_i(x)<1$)

  or log-likelihood reconstruction criterion, e.g., for a multinomial (one-hot) input
  - $- \Sigma_i x_i \log r_i(x)$ (where $\Sigma_i r_i(x)=1$, summing over subset of inputs associated with this multinomial variable)

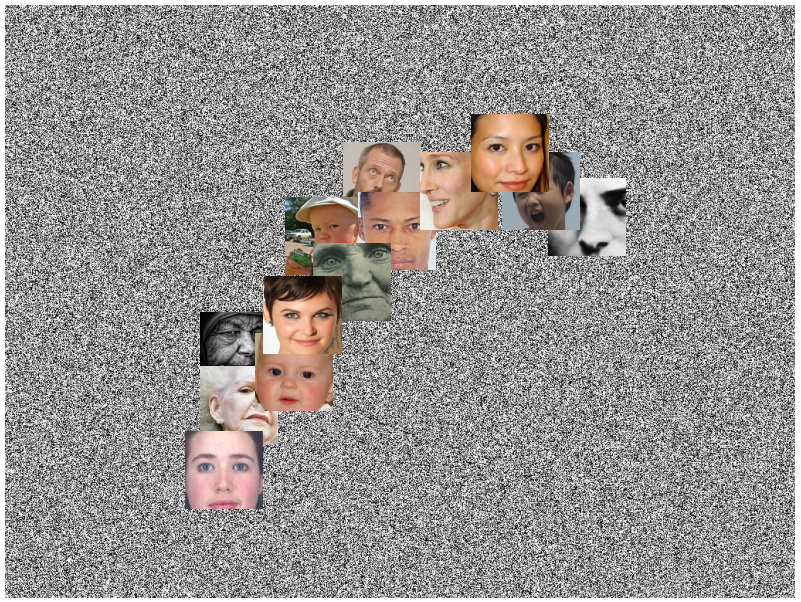
- In general: consider what are appropriate loss functions to predict each of the input variables, typically $-\log P(x|r(x))$ or the equivalent KL divergence.

Manifold Learning

- Additional prior: examples **concentrate** near a lower dimensional "manifold" (region of high density with only few operations allowed which allow small changes while staying on the manifold)

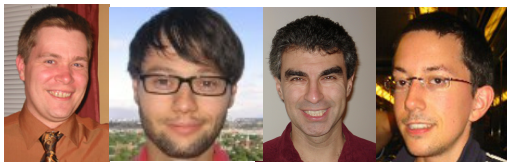


- variable dimension locally?
- Soft # of dimensions?

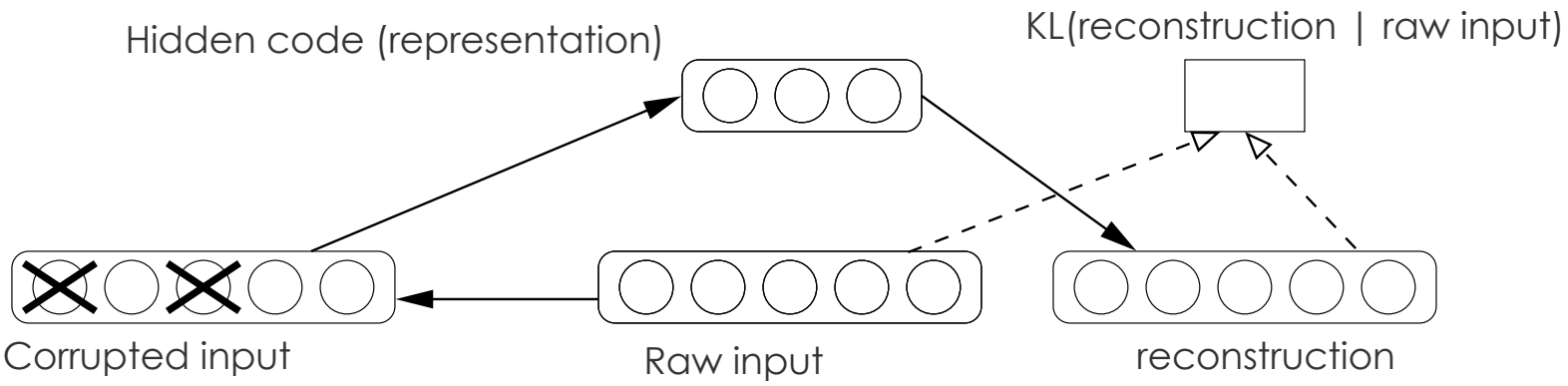


Denoising Auto-Encoder

(Vincent et al 2008)



- Corrupt the input
- 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

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

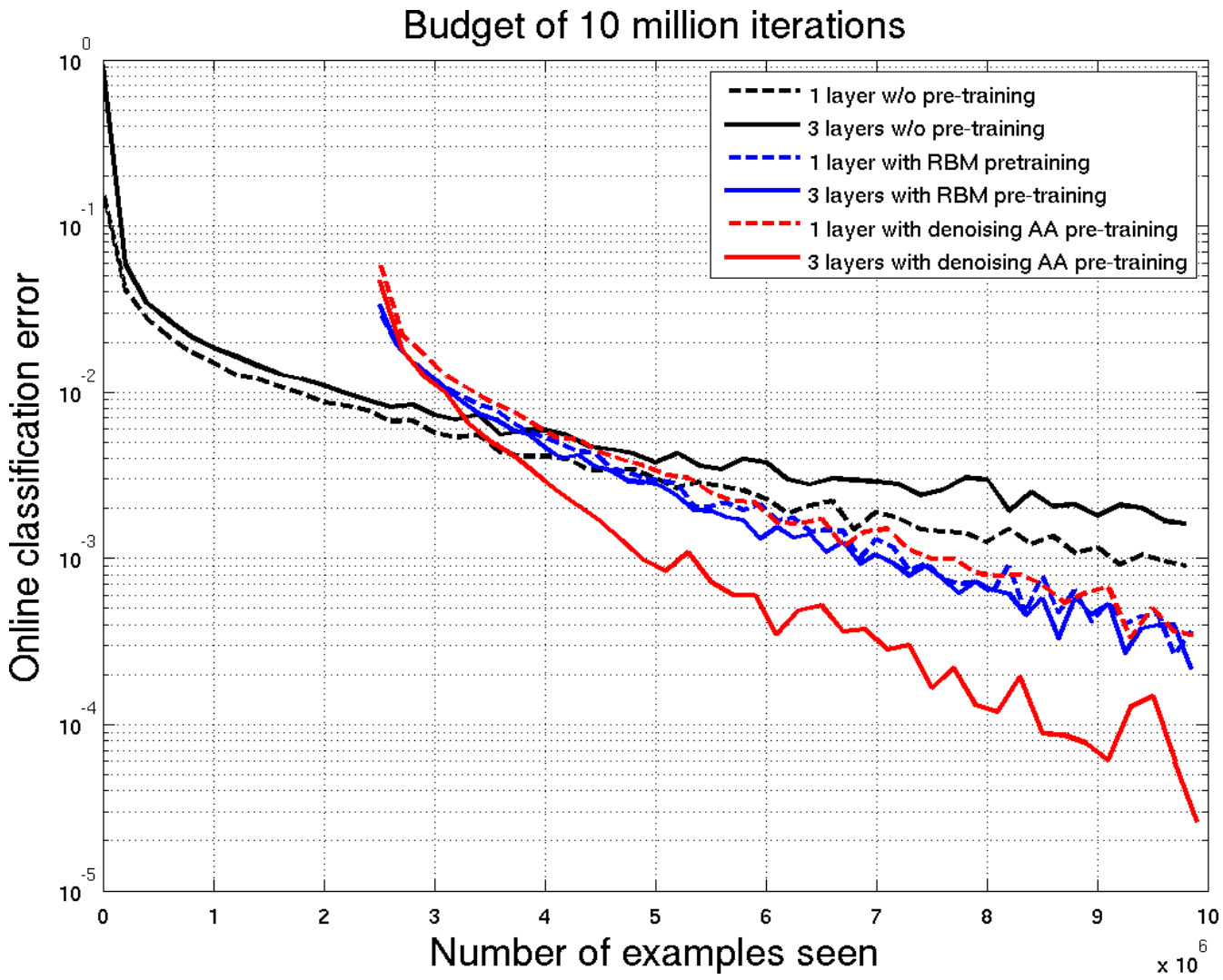
- Some DAEs correspond to a kind of Gaussian RBM with *regularized* Score Matching (Vincent 2011)

  [equivalent when noise→0]

- No partition function, can measure training criterion

prior: examples concentrate near a lower dimensional "manifold"

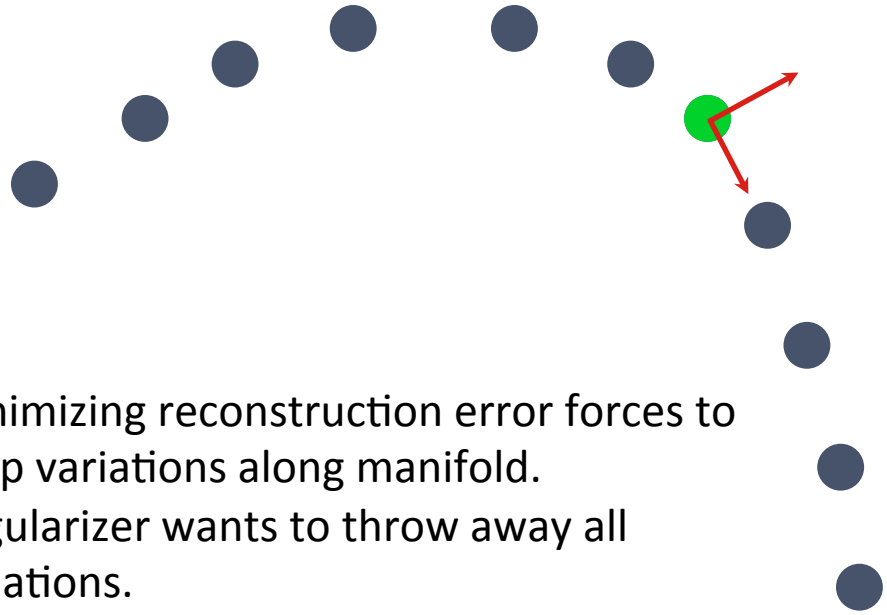
Stacked Denoising Auto-Encoders



Infinite MNIST

Note how advantage of better initialization does not vanish like other regularizers as #exemples→∞

Auto-Encoders Learn Salient Variations, like a non-linear PCA



- Minimizing reconstruction error forces to keep variations along manifold.
- Regularizer wants to throw away all variations.
- With both: keep ONLY sensitivity to variations ON the manifold.

Contractive Auto-Encoders

(Rifai, Vincent, Muller, Glorot, Bengio ICML 2011; Rifai, Mesnil, Vincent, Bengio, Dauphin, Glorot ECML 2011; Rifai, Dauphin, Vincent, Bengio, Muller NIPS 2011)

$$\text{reconstruction}(x) = g(h(x)) = \text{decoder}(\text{encoder}(x))$$

Training criterion:

$$\mathcal{J}_{CAE}(\theta) = \sum_{x \in D_n} \lambda \sum_{ij} \left(\frac{\partial h_j(x)}{\partial x_i} \right)^2 + L(x, \text{reconstruction}(x))$$

wants contraction in all directions

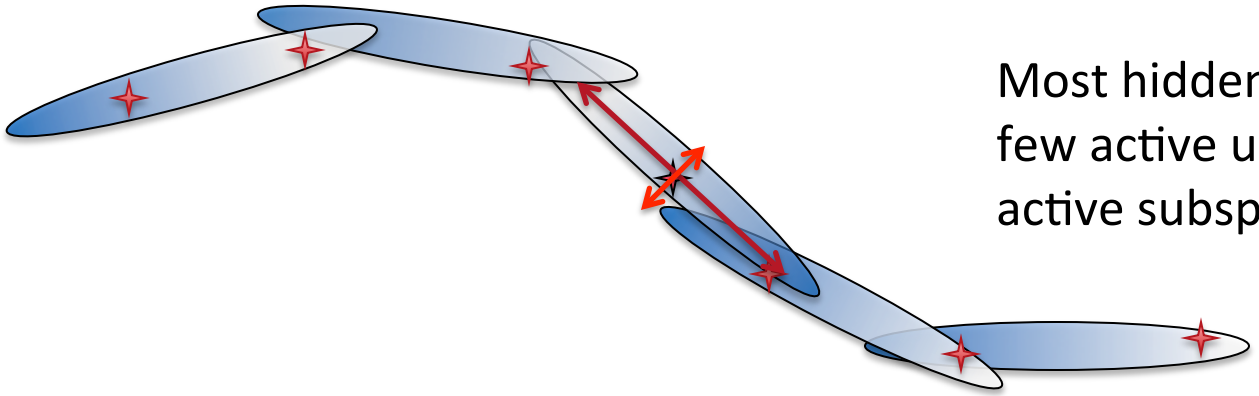
cannot afford contraction in manifold directions

If $h_j = \text{sigmoid}(b_j + W_j x)$

$(dh_j(x)/dx_i)^2 = h_j^2(1-h_j)^2 W_{ji}^2$

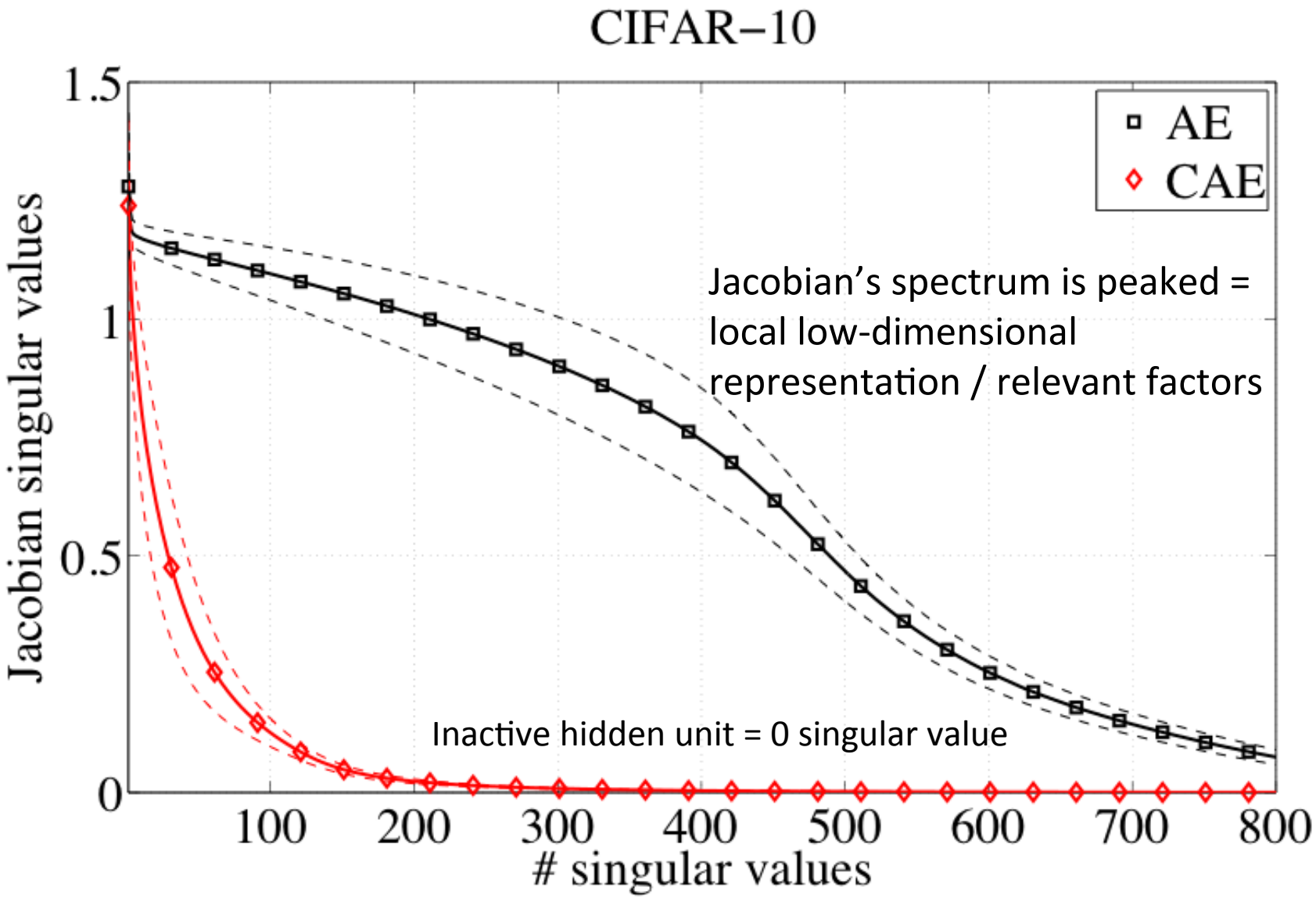
Contractive Auto-Encoders

(Rifai, Vincent, Muller, Glorot, Bengio ICML 2011; Rifai, Mesnil, Vincent, Bengio, Dauphin, Glorot ECML 2011; Rifai, Dauphin, Vincent, Bengio, Muller NIPS 2011)



Most hidden units saturate: few active units represent the active subspace (local chart)

Each region/chart = subset of active hidden units
Neighboring region: one of the units becomes active/inactive
SHARED SET OF FILTERS ACROSS REGIONS, EACH USING A SUBSET


CIFAR−10
AE
CAE
Jacobian's spectrum is peaked = local low-dimensional representation / relevant factors
Inactive hidden unit = 0 singular value
Jacobian singular values
singular values
1.5
1
0.5
0
100
200
300
400
500
600
700
800

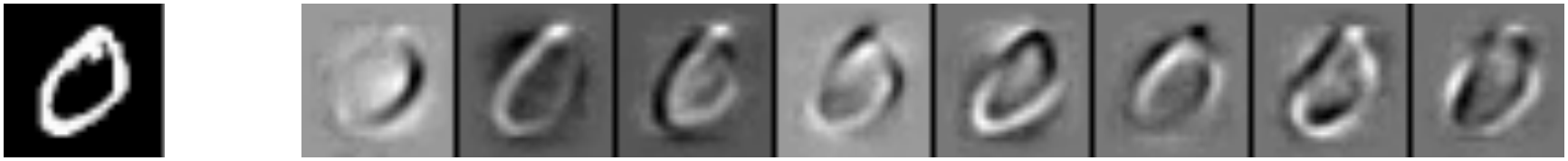
Contractive Auto-Encoders

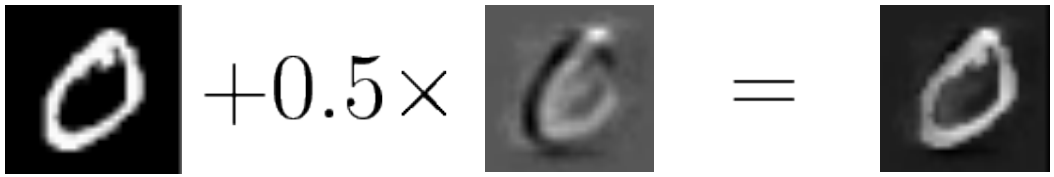
Benchmark of medium-size datasets on which several deep learning algorithms had been evaluated (Larochelle et al ICML 2007)

Data Set	SVM_{rbf}	SAE-3	RBM-3	DAE-b-3	CAE-1	CAE-2
basic	**3.03**$\pm$0.15	3.46$\pm$0.16	3.11$\pm$0.15	2.84$\pm$0.15	2.83$\pm$0.15	**2.48**$\pm$0.14
rot	11.11$\pm$0.28	10.30$\pm$0.27	10.30$\pm$0.27	**9.53**$\pm$0.26	11.59$\pm$0.28	**9.66**$\pm$0.26
bg-rand	14.58$\pm$0.31	11.28$\pm$0.28	**6.73**$\pm$0.22	10.30$\pm$0.27	13.57$\pm$0.30	10.90 $\pm$0.27
bg-img	22.61$\pm$0.379	23.00$\pm$0.37	16.31$\pm$0.32	16.68$\pm$0.33	16.70$\pm$0.33	**15.50**$\pm$0.32
bg-img-rot	55.18$\pm$0.44	51.93$\pm$0.44	47.39$\pm$0.44	**43.76**$\pm$0.43	48.10$\pm$0.44	45.23$\pm$0.44
rect	**2.15**$\pm$0.13	2.41$\pm$0.13	2.60$\pm$0.14	1.99$\pm$0.12	1.48$\pm$0.10	**1.21**$\pm$0.10
rect-img	24.04$\pm$0.37	24.05$\pm$0.37	22.50$\pm$0.37	**21.59**$\pm$0.36	**21.86**$\pm$0.36	**21.54**$\pm$0.36

Input Point

Tangents

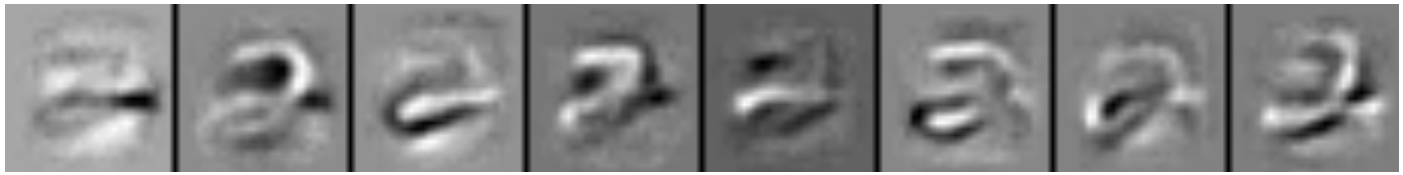




MNIST

Input Point

Tangents



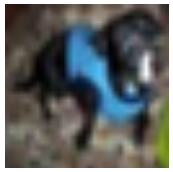
MNIST Tangents

Distributed vs Local (CIFAR-10 unsupervised)

Input Point

Tangents

Local PCA (no sharing across regions)



Contractive Auto-Encoder

Denoising auto-encoders are also contractive!

- Taylor-expand Gaussian corruption noise in reconstruction error:

$$
\begin{aligned}
E\left[\ell(x, r(x+\epsilon))\right] &\approx E\left[\left(x - \left(r(x) + \frac{\partial r(x)}{\partial x}\epsilon\right)\right)^T \left(x - \left(r(x) + \frac{\partial r(x)}{\partial x}\epsilon\right)\right)\right] \\
&= E\left[\|x - r(x)\|^2\right] + \sigma^2 E\left[\left\|\frac{\partial r(x)}{\partial x}\right\|_F^2\right]
\end{aligned}
$$

- Yields a contractive penalty in the reconstruction function (instead of encoder) proportional to amount of corruption noise

Learned Tangent Prop: the Manifold Tangent Classifier

3 hypotheses:

1. Semi-supervised hypothesis ($P(x)$ related to $P(y|x)$)

2. Unsupervised manifold hypothesis (data concentrates near low-dim. manifolds)

3. Manifold hypothesis for classification (low density between class manifolds)

Learned Tangent Prop: the Manifold Tangent Classifier

Algorithm:

1. Estimate local principal directions of variation U(x) by CAE (principal singular vectors of dh(x)/dx)

2. Penalize f(x)=P(y|x) predictor by || df/dx U(x) ||

Makes f(x) insensitive to variations on manifold at x, tangent plane characterized by U(x).

Manifold Tangent Classifier Results

- Leading singular vectors on MNIST, CIFAR-10, RCV1:

Trading & Markets								
	+gilt	-slow	+matur	-percent	+bln	-anti	+interest	-sen
	+yen	-term	+auction	-sent	+coupon	-predict	+calcul	-californ
	+usda	-debt	+treasur	-pressure	+discount	-belgian	+overnight	-introduc

- **Knowledge-free MNIST: 0.81% error**

K-NN	NN	SVM	DBN	CAE	DBM	CNN	MTC
3.09%	1.60%	1.40%	1.17%	1.04%	0.95%	0.95%	**0.81%**

- Semi-sup.

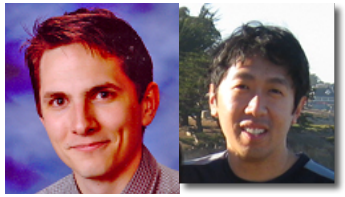
	NN	SVM	CNN	TSVM	DBN-rNCA	EmbedNN	CAE	MTC
100	25.81	23.44	22.98	16.81	-	16.86	13.47	**12.03**
600	11.44	8.85	7.68	6.16	8.7	5.97	6.3	**5.13**
1000	10.7	7.77	6.45	5.38	-	5.73	4.77	**3.64**
3000	6.04	4.21	3.35	3.45	3.3	3.59	3.22	**2.57**

- Forest (500k examples)

SVM	Distributed SVM	MTC
4.11%	3.46%	**3.13%**

Inference and Explaining Away

- Easy inference in RBMs and regularized Auto-Encoders
- But no explaining away (competition between causes)
- (Coates et al 2011): even when training filters as RBMs it helps to perform additional explaining away (e.g. plug them into a Sparse Coding inference), to obtain better-classifying features



- RBMs would need lateral connections to achieve similar effect
- Auto-Encoders would need to have lateral recurrent connections

Sparse Coding (Olshausen et al 97)

- Directed graphical model:

$$P(h) \propto e^{-\lambda|h|_1} \qquad x|h \sim N(W^T h, \sigma^2 I)$$

- One of the first unsupervised feature learning algorithms with non-linear feature extraction (but linear decoder)

$$\min_h \frac{||x - W^T h||^2}{\sigma^2} + \lambda|h|_1$$

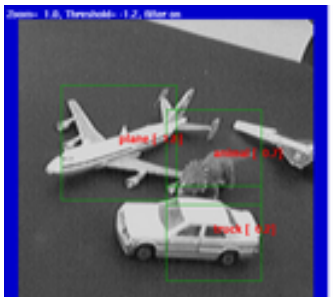
MAP inference recovers sparse h although P(h|x) not concentrated at 0

- Linear decoder, non-parametric encoder
- Sparse Coding inference, convex opt. but expensive

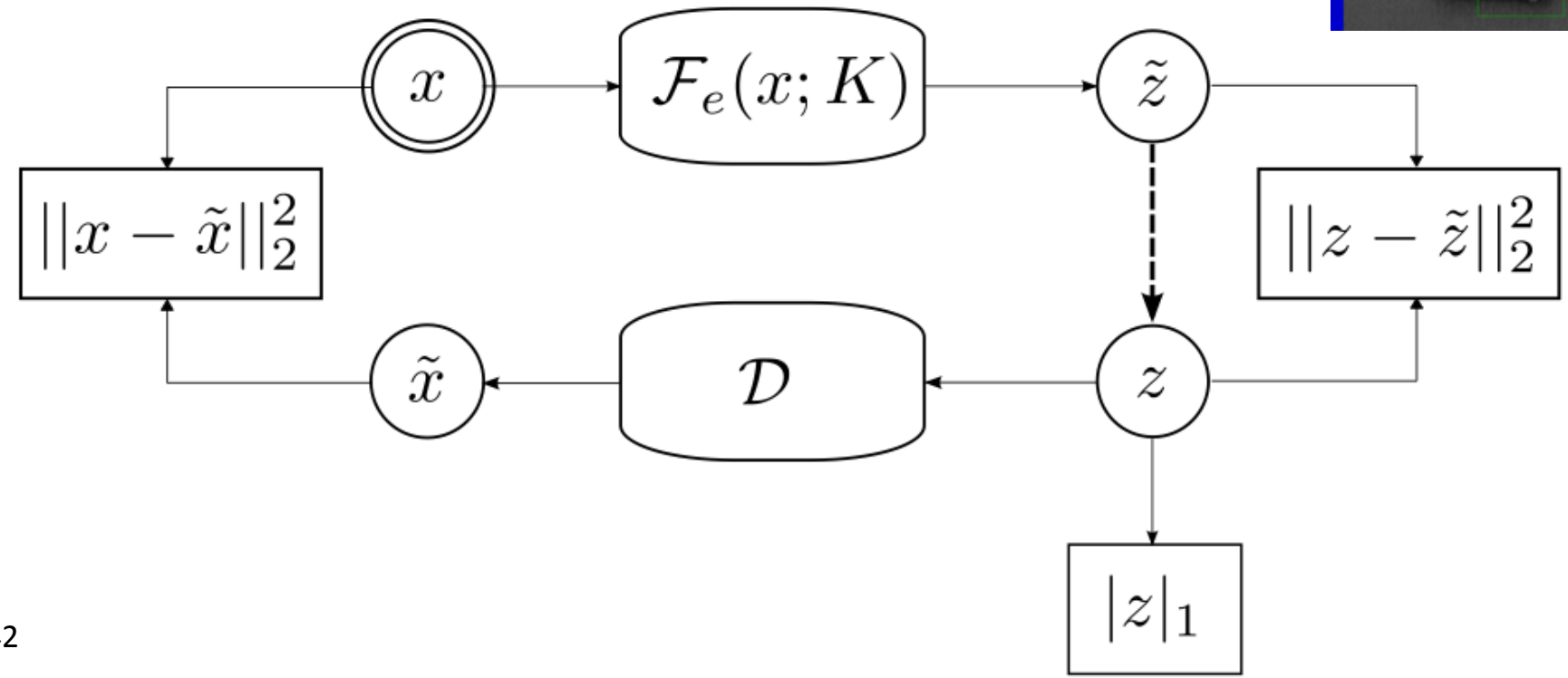
Predictive Sparse Decomposition

- Approximate the inference of sparse coding by an encoder:

Predictive Sparse Decomposition (Kavukcuoglu et al 2008)

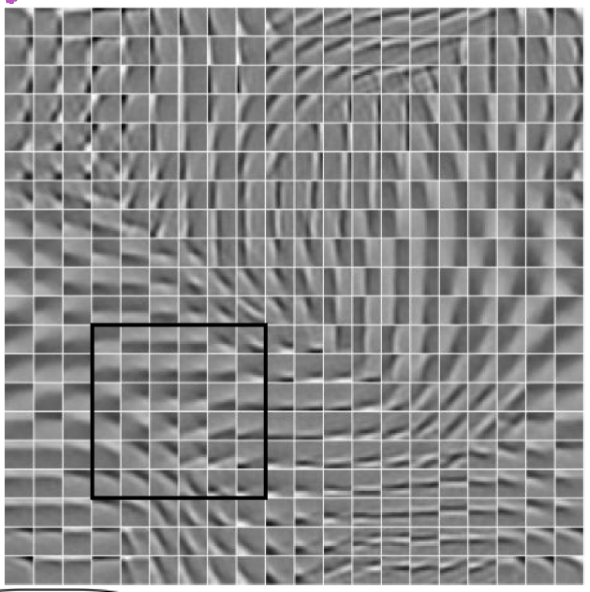


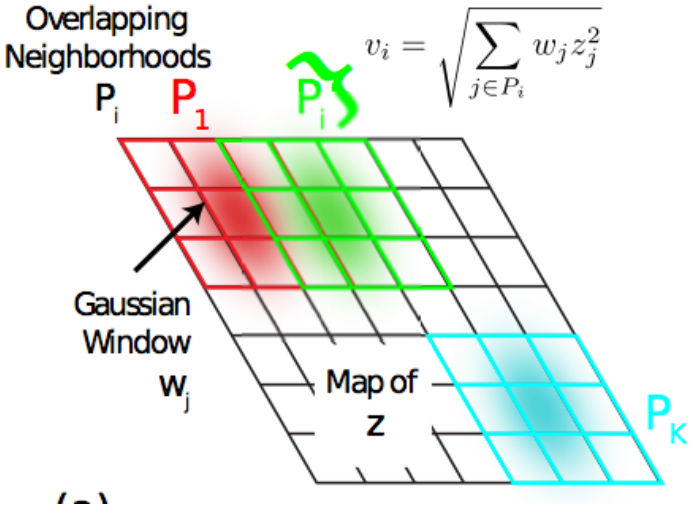
- Very successful applications in machine vision with convolutional architectures

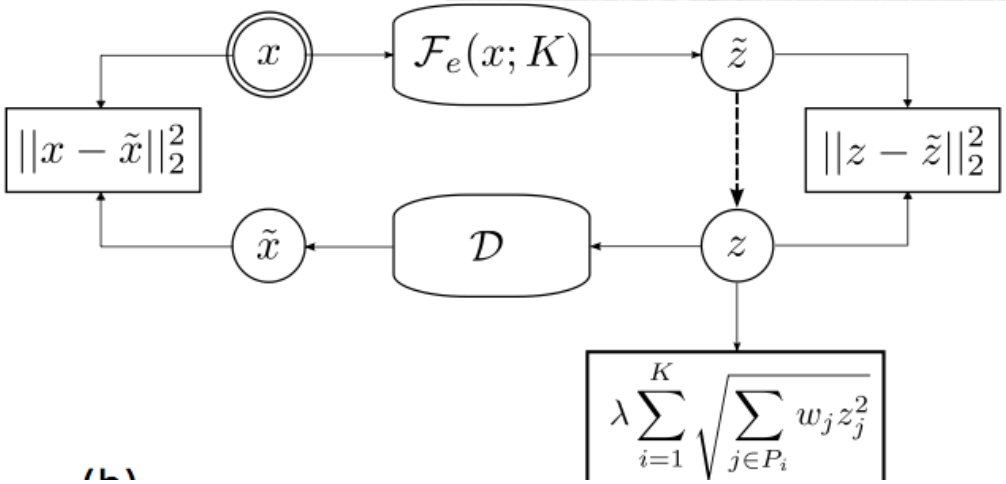


Predictive Sparse Decomposition

- Stacked to form deep architectures
- Alternating convolution, rectification, pooling
- Tiling: no sharing across overlapping filters
- Group sparsity penalty yields topographic maps





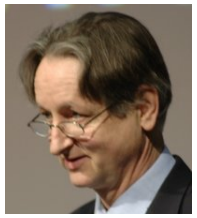


Deep Variants

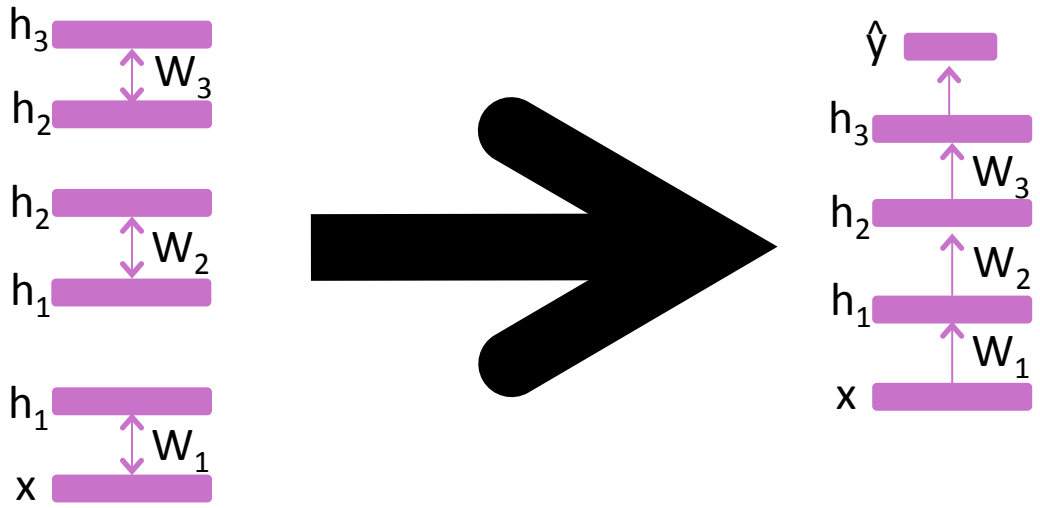
Level-Local Learning is Important

- Initializing each layer of an unsupervised deep Boltzmann machine helps a lot
- Initializing each layer of a supervised neural network as an RBM, auto-encoder, denoising auto-encoder, etc helps a lot
- Helps most the layers further away from the target
- Not just an effect of unsupervised prior
- Jointly training all the levels of a deep architecture is difficult
- Initializing using **a level-local learning algorithm** is a useful trick

Stack of RBMs / AEs → Deep MLP



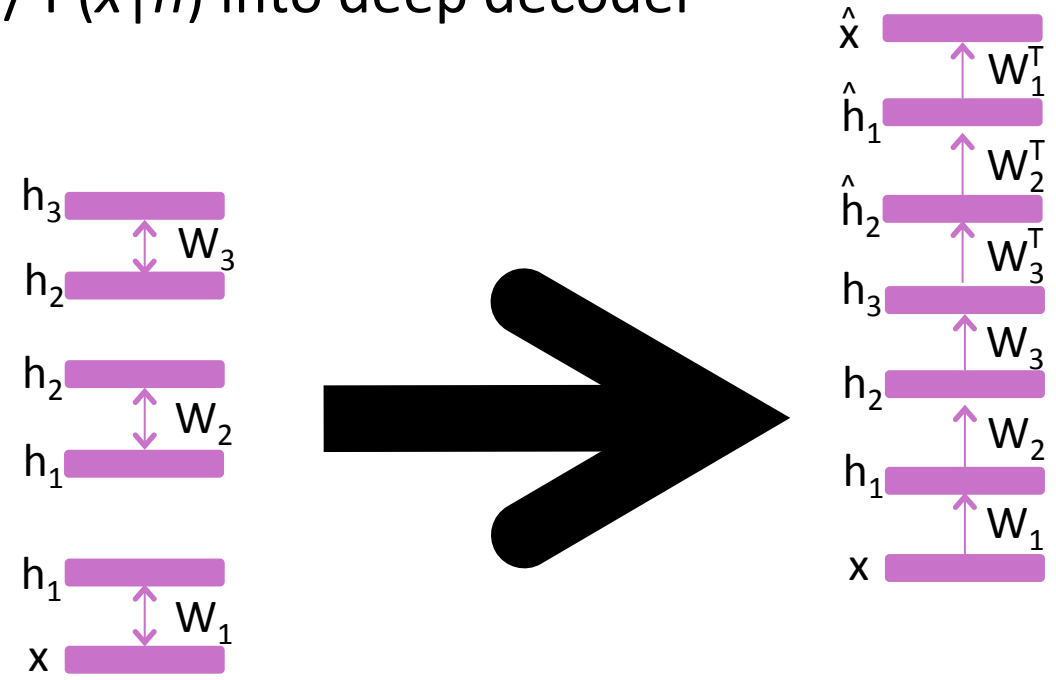
- Encoder or P(*h*|*v*) becomes MLP layer



Stack of RBMs / AEs → Deep Auto-Encoder

(Hinton & Salakhutdinov 2006)

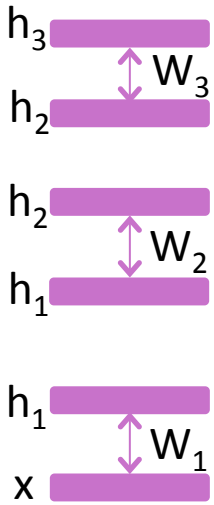
- Stack encoders / P($h|x$) into deep encoder
- Stack decoders / P($x|h$) into deep decoder

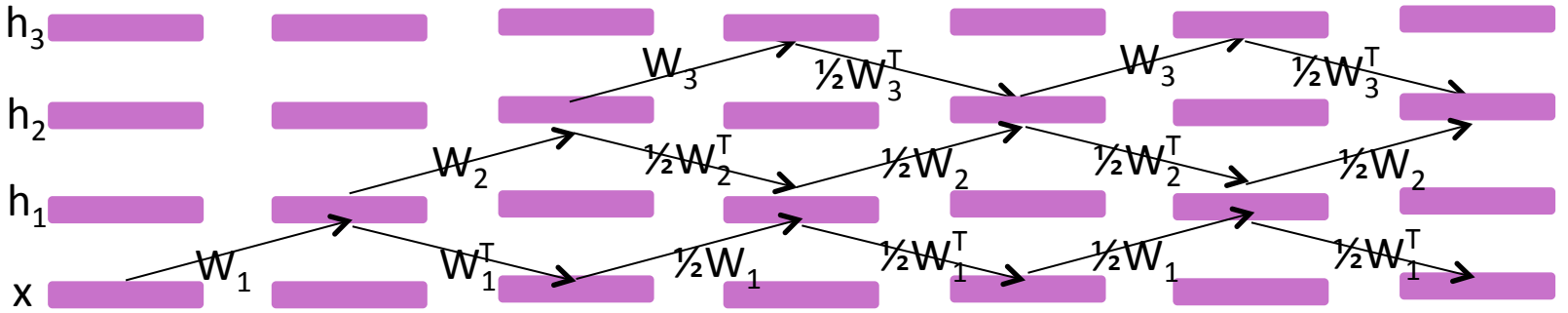


Stack of RBMs / AEs → Deep Recurrent Auto-Encoder

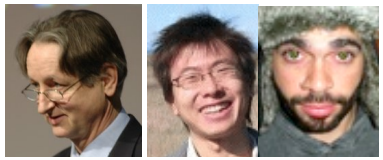
(Savard 2011)

- Each hidden layer receives input from below and above
- Halve the weights
- Deterministic (mean-field) recurrent computation



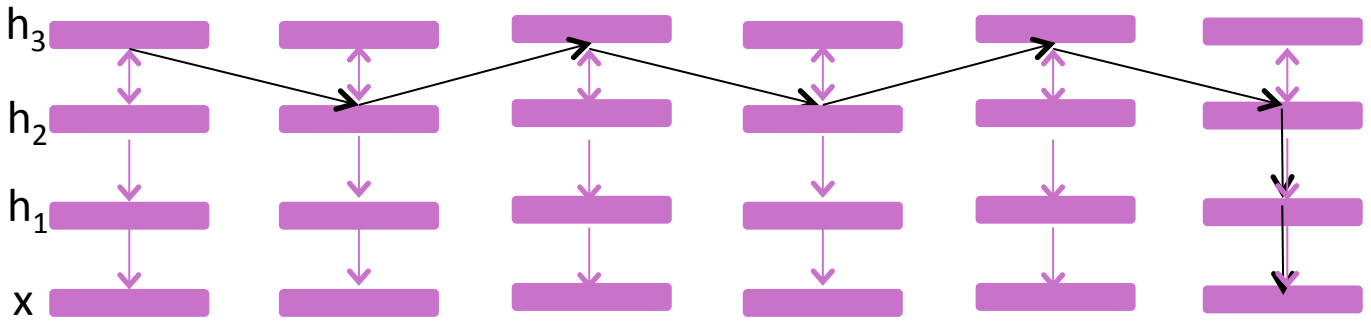


Stack of RBMs → Deep Belief Net



(Hinton et al 2006)

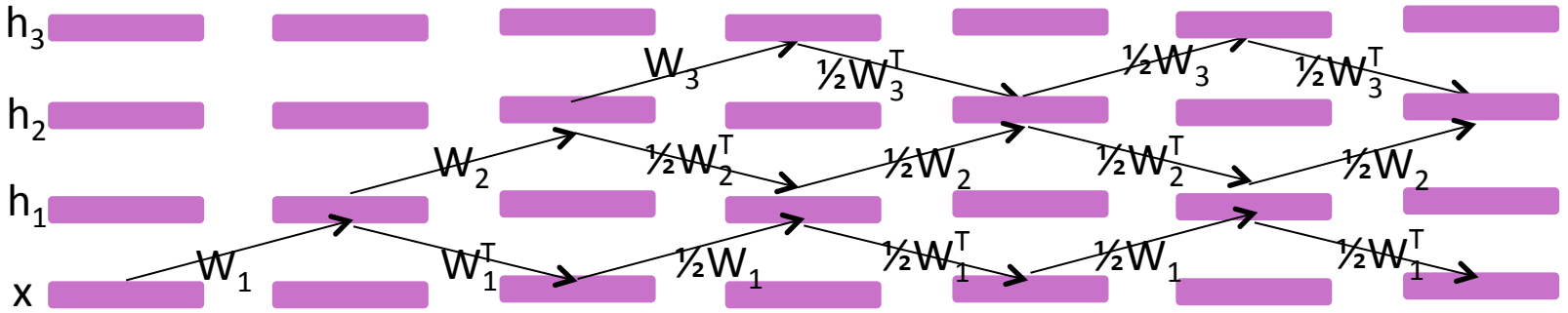
- Stack lower levels RBMs' P($x|h$) along with top-level RBM
- $P(x, h_1, h_2, h_3) = P(h_2, h_3)\, P(h_1|h_2)\, P(x \mid h_1)$
- Sample: Gibbs on top RBM, propagate down



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

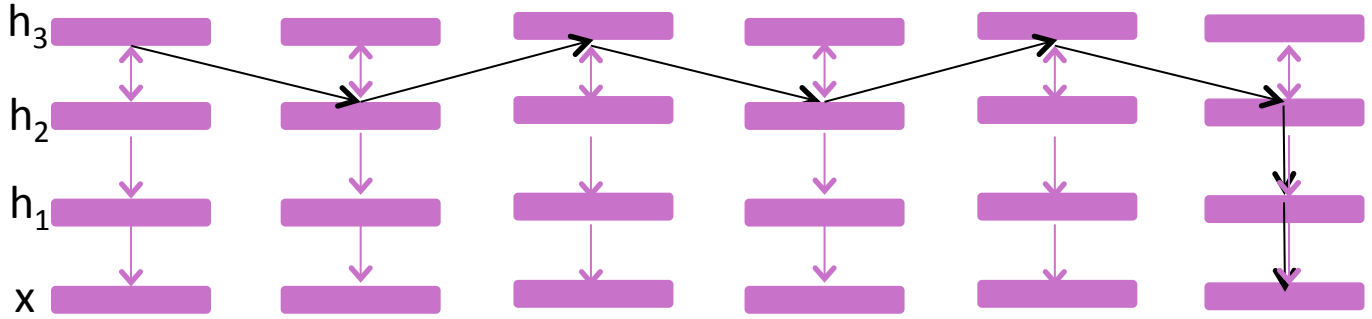


Stack of Auto-Encoders → Deep Generative Auto-Encoder

(Rifai et al ICML 2012)

- MCMC on top-level auto-encoder
  - h_{t+1} = encode(decode(h_t))+σ noise

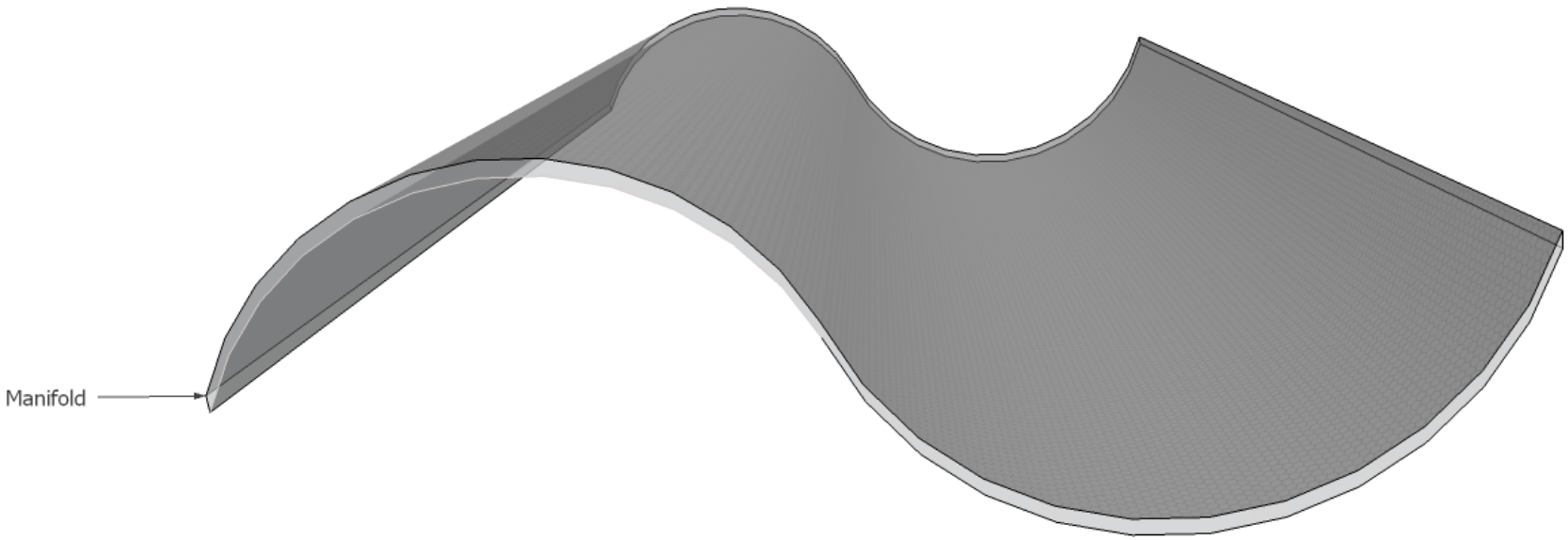
  where noise is Normal(0, d/dh encode(decode(h_t)))
- Then deterministically propagate down with decoders



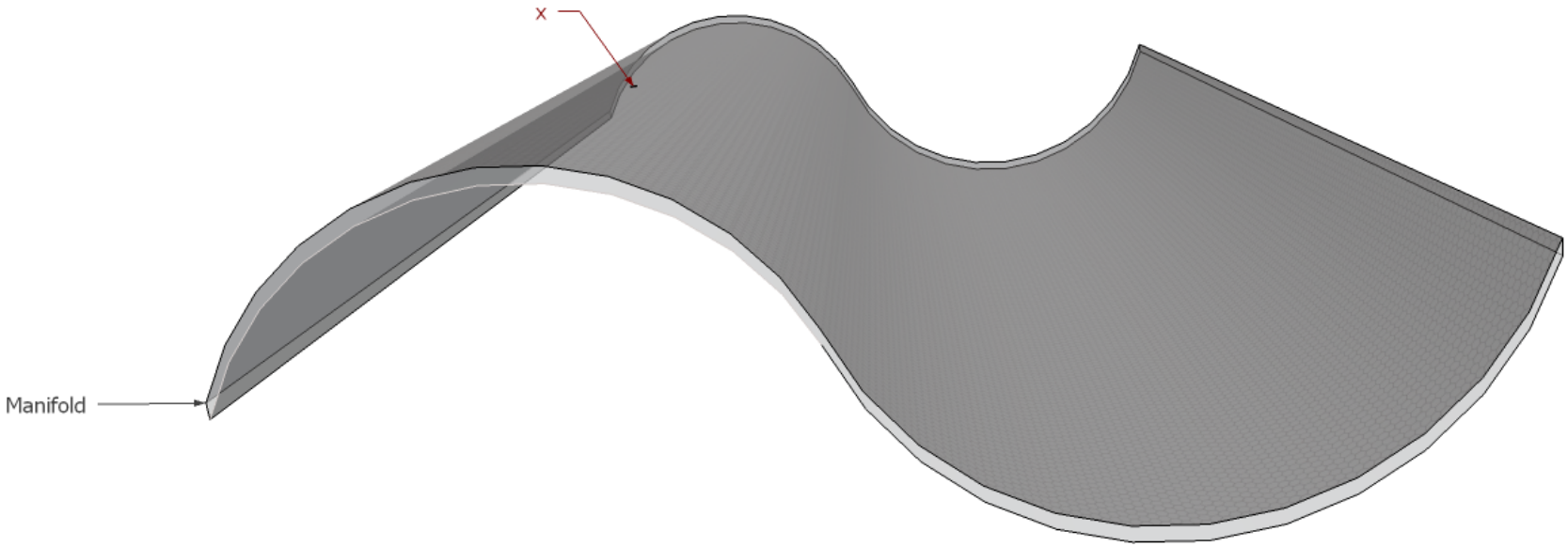
Manifold Learning Interpretation Allows Sampling from Auto-Encoders

- Reconstruction function captures geometry of the input distribution
- *reconstruction(x)-x* points towards high-density (score)
- Jacobian of *reconstruction(x)* has large singular values in directions of local factors of variation (manifold tangents)
- Gives rise to an implicit density estimator and a **sampling algorithm** for contractive and denoising auto-encoders (Rifai et al ICML 2012)

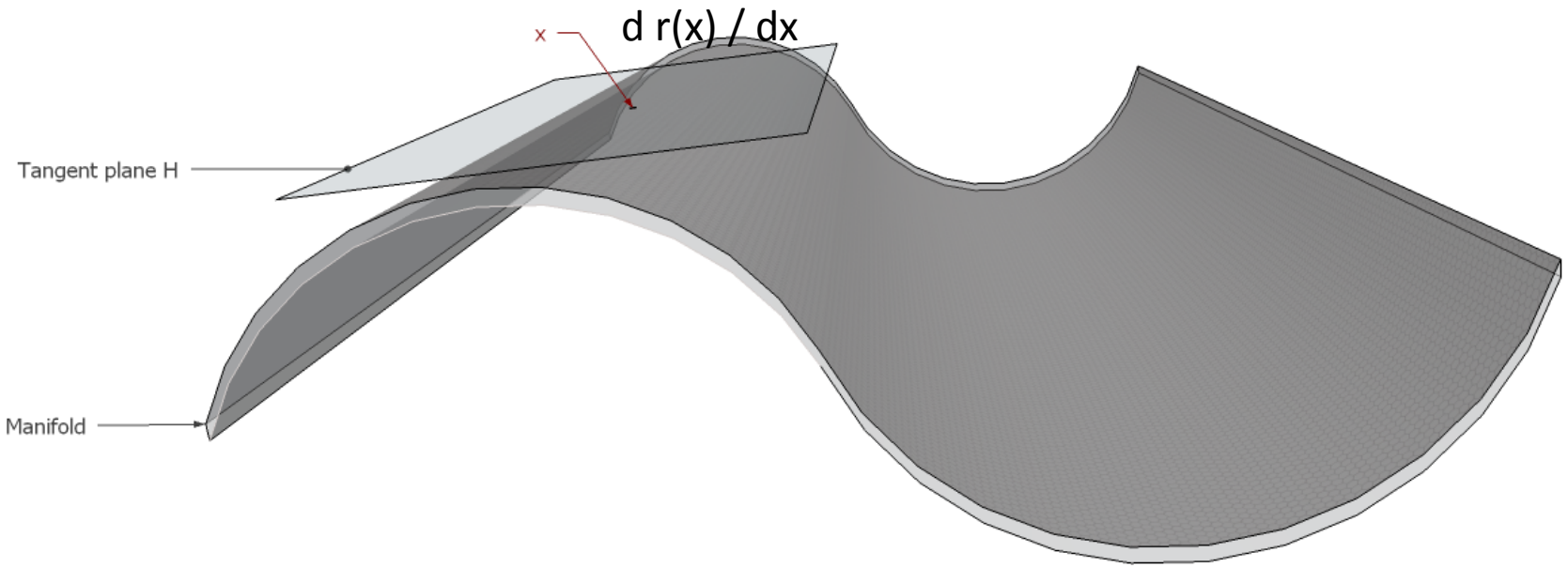
Sampling from a Regularized Auto-Encoder



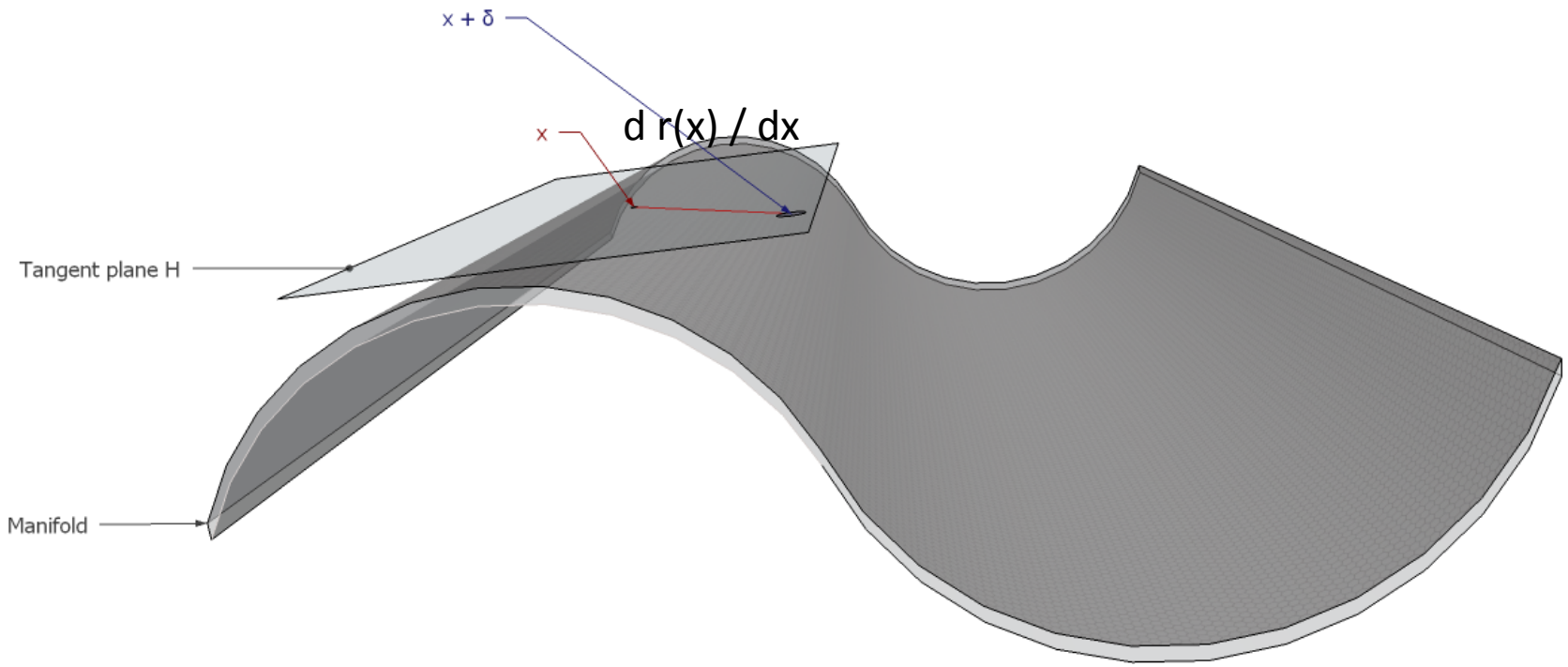
Sampling from a Regularized Auto-Encoder



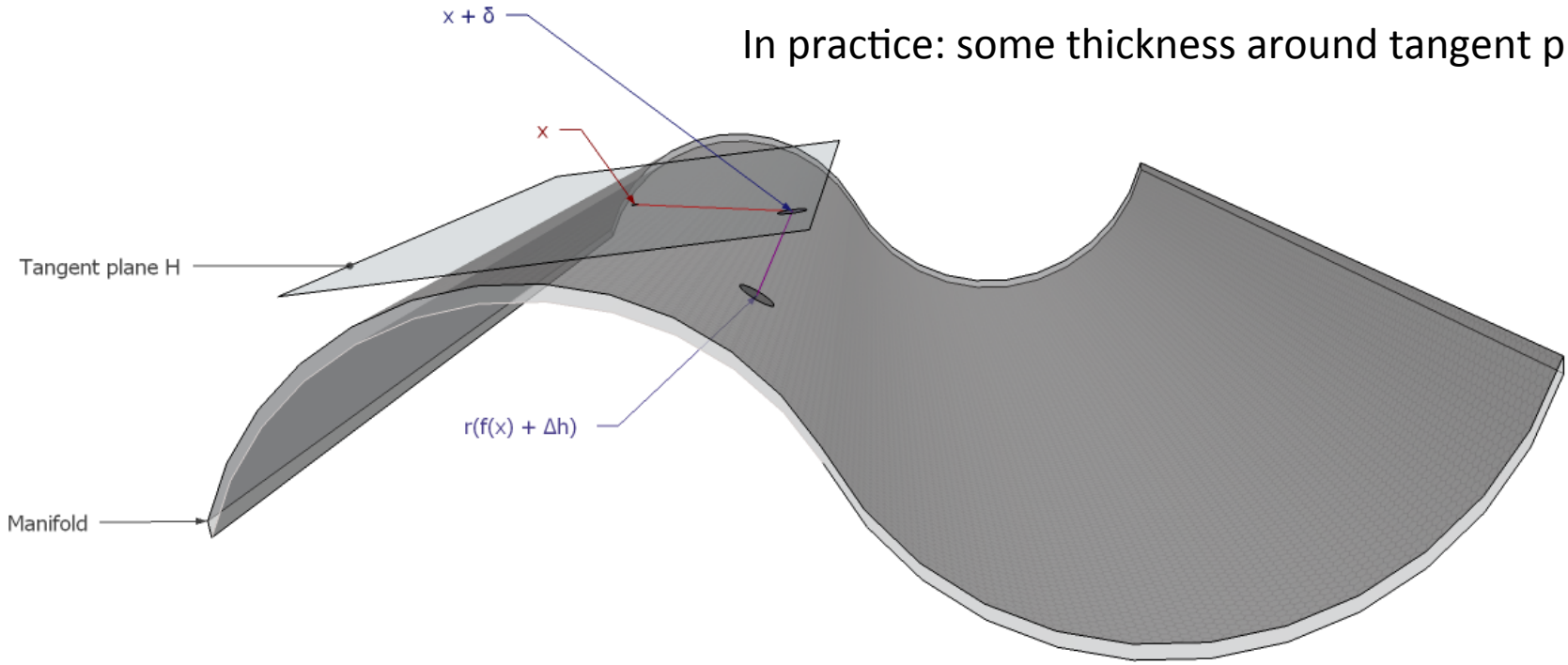
Sampling from a Regularized Auto-Encoder



Sampling from a Regularized Auto-Encoder



Sampling from a Regularized Auto-Encoder



In practice: some thickness around tangent plane..

Samples from a 2-level DAE

- TFD

- MNIST

Samples from a 2-level CAE

(ICML 2012)

Table 1. Log-Likelihoods from Parzen density estimator using 10000 samples from each model

	DBN-2	CAE-2
TFD	1908.80 ± 65.94	**2110.09** ± 49.15
MNIST	**137.89** ± 2.11	121.17 ± 1.59

CAE2

DBN2

MNIST TFD

- Not using local covariance estimator, just isotropic noise: **bad**

MCMC Asymptotic Distribution: Uncountable Gaussian Mixture

- Each step samples next *x* from Gaussian with mean and covariance a function of previous $\tilde{x}$
- Asymptotic distribution (if exists):

$$\pi(x) = \int \pi(\tilde{x}) \mathcal{N}(x; \mu(\tilde{x}), \Sigma(\tilde{x})) d\tilde{x}$$

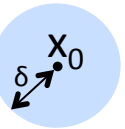
= uncountable gaussian mixture with weights = the density itself

- Thm: If Σ(x) is full-rank and μ(x) in bounded region, then π exists.

Consistency: Samples ↔ Local Moments

(Bengio et al 2012, arXiv paper, "Implicit Density Estimation by Local Moment Matching to Sample from Auto-Encoders")

- Inside-ball density: $p_\delta(x|x_0) = \frac{p(x)1_{||x-x_0||<\delta}}{Z(x_0)}$



- Ball size $\delta \to 0$ around each x_0, MCMC steps of size $\sigma << \delta$

$$
\begin{aligned}
m_0 = E_\pi[x|x_0] &= \frac{1}{Z(x_0)} \int_x x \int_{\tilde{x}} p(\tilde{x}) \mathcal{N}(x; \mu(\tilde{x}), \Sigma(\tilde{x})) d\tilde{x}\, 1_{||x-x_0||<\delta} dx \\
&= \frac{1}{Z(x_0)} \int_{\tilde{x}} p(\tilde{x}) \int_{||x-x_0||<\delta} x \mathcal{N}(x; \mu(\tilde{x}), \Sigma(\tilde{x})) dx d\tilde{x}. \\
&\approx \int_{\tilde{x}} \frac{p(\tilde{x})}{Z(x_0)} 1_{||\mu(\tilde{x})-x_0||<\delta} \mu(\tilde{x}) d\tilde{x} \\
&\approx \mathbb{E}[\mu(x)|x_0]
\end{aligned}
$$

- i.e. the local mean m_0 ≈ expected value of MCMC mean in the ball, and similarly for local covariance C_0 & MCMC covariance.
- *Step size σ controls quality of approximation, which corresponds to a smooth of the estimated density*.

Consistency: Non-Parametric / Asymptotic Minimizer of Criterion

- Training criterion rewritten:

$$\mathcal{L}_{\text{global}} = \int p(x_0) \left(\|x_0 - r(x_0)\|^2 + \alpha \left\| \frac{\partial r(x_0)}{\partial x_0} \right\|_F^2 \right) dx_0$$

$$= \lim_{\delta \to 0} \int p(x_0) \left(\left(\int_x ||x - r(x)||^2 p_\delta(x|x_0) dx \right) + \alpha \left\| \frac{\partial r(x_0)}{\partial x_0} \right\|_F^2 \right) dx_0$$

- Local (non-parametric) parametrization around x_0

$$r(x) = r(x_0) + \left. \frac{\partial r}{\partial x} \right|_{x_0} (x - x_0) = r_0 + J_0(x - x_0)$$

$$\mathcal{L}_{\text{local}}(x_0, \delta) = \int_x ||x - (r_0 + J_0(x - x_0))||^2 p_\delta(x|x_0) dx + \alpha ||J_0||_F^2$$

$$\mathcal{L}_{\text{global}} = \lim_{\delta \to 0} \int p(x_0) \mathcal{L}_{\text{local}}(x_0, \delta) dx_0$$

Consistency: Non-Parametric / Asymptotic Minimizer of Criterion

- Solving: $\frac{\partial \mathcal{L}_{\text{local}}(x_0, \delta)}{\partial r_0} = 0$

  $\frac{\partial \mathcal{L}_{\text{local}}(x_0, \delta)}{\partial J_0} = 0$

  yields:

$$\begin{aligned} r_0 &= (I - J_0)m_0 + J_0 x_0 \\ J_0 &= C_0(\alpha I + C_0)^{-1}. \end{aligned}$$

  i.e. when $\delta \to 0$ (i.e. $J_0 \to 0$), $\asymp$ means lhs / rhs $\to$ 1:

$$\begin{aligned} r_0 &\asymp m_0 \\ J_0 &\asymp \alpha^{-1} C_0 \end{aligned}$$

- **Reconstruction and its Jacobian estimate local mean & covariance**

Implicit Density Estimation

- In general, no explicit analytic formulation of the estimated density, only of its local moments and 1st & 2nd derivatives
- Can obtain samples by MCMC (of a smooth of estimated density)

- Alternatively, can parametrize $r(x)-x$ = derivative of an energy function $energy(x)$ which provides an explicit analytic formulation of the estimated density.

- **We have avoided the partition function and introduced a novel(?) alternative to maximum likelihood**

AE sampling: open questions

- Effects of parametric non-asymptotic setting?
- Training energy-based models as regularized AE
- Why better results when training as CAE vs DAE?