Efficient Algorithms may not be those we think

Yann LeCun,
Computational and Biological Learning Lab
The Courant Institute of Mathematical Sciences
New York University

http://yann.lecun.com

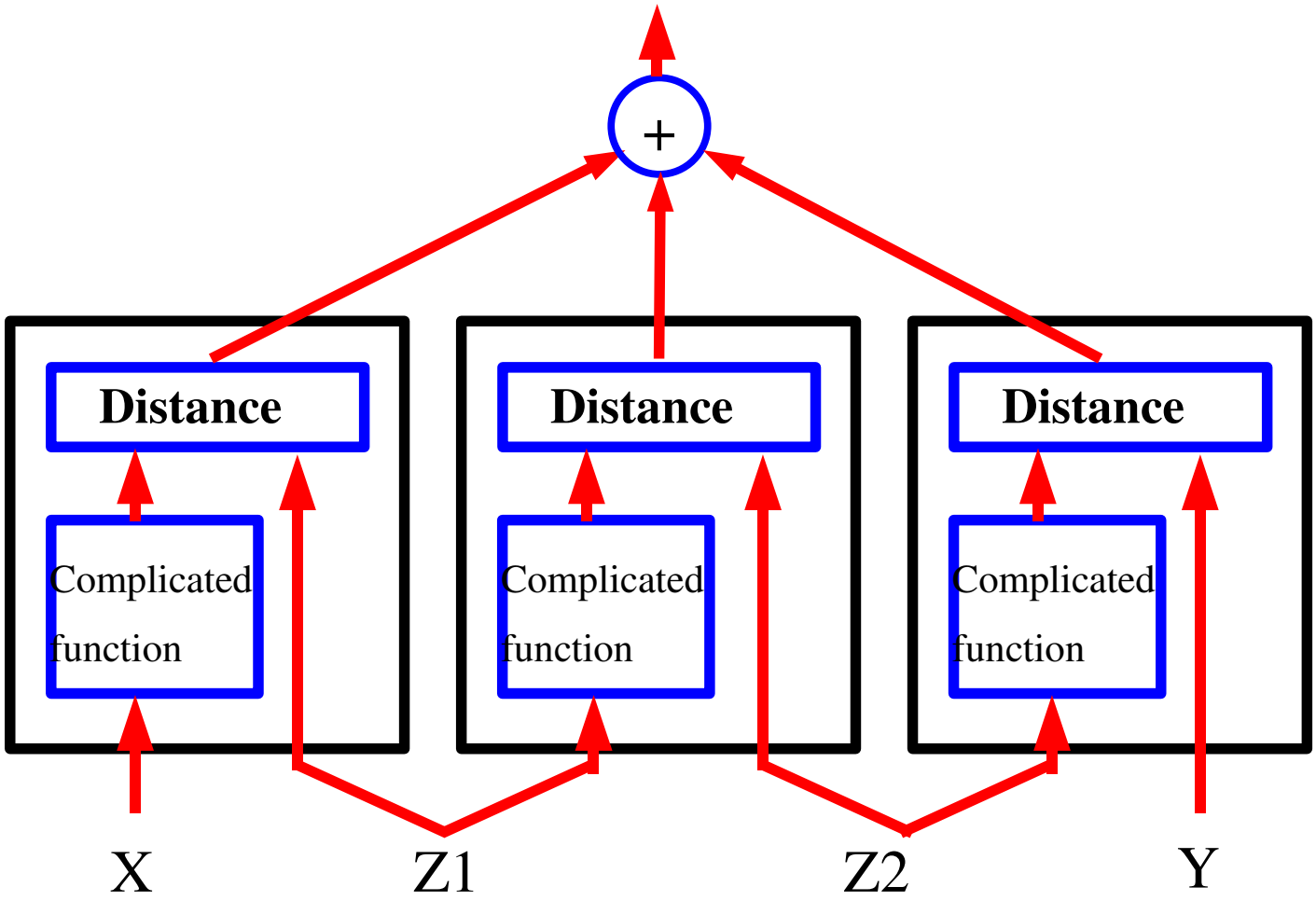
http://www.cs.nyu.edu/~yann

The Importance of Architecture

- The internal structure of factors influences the efficiency of inference

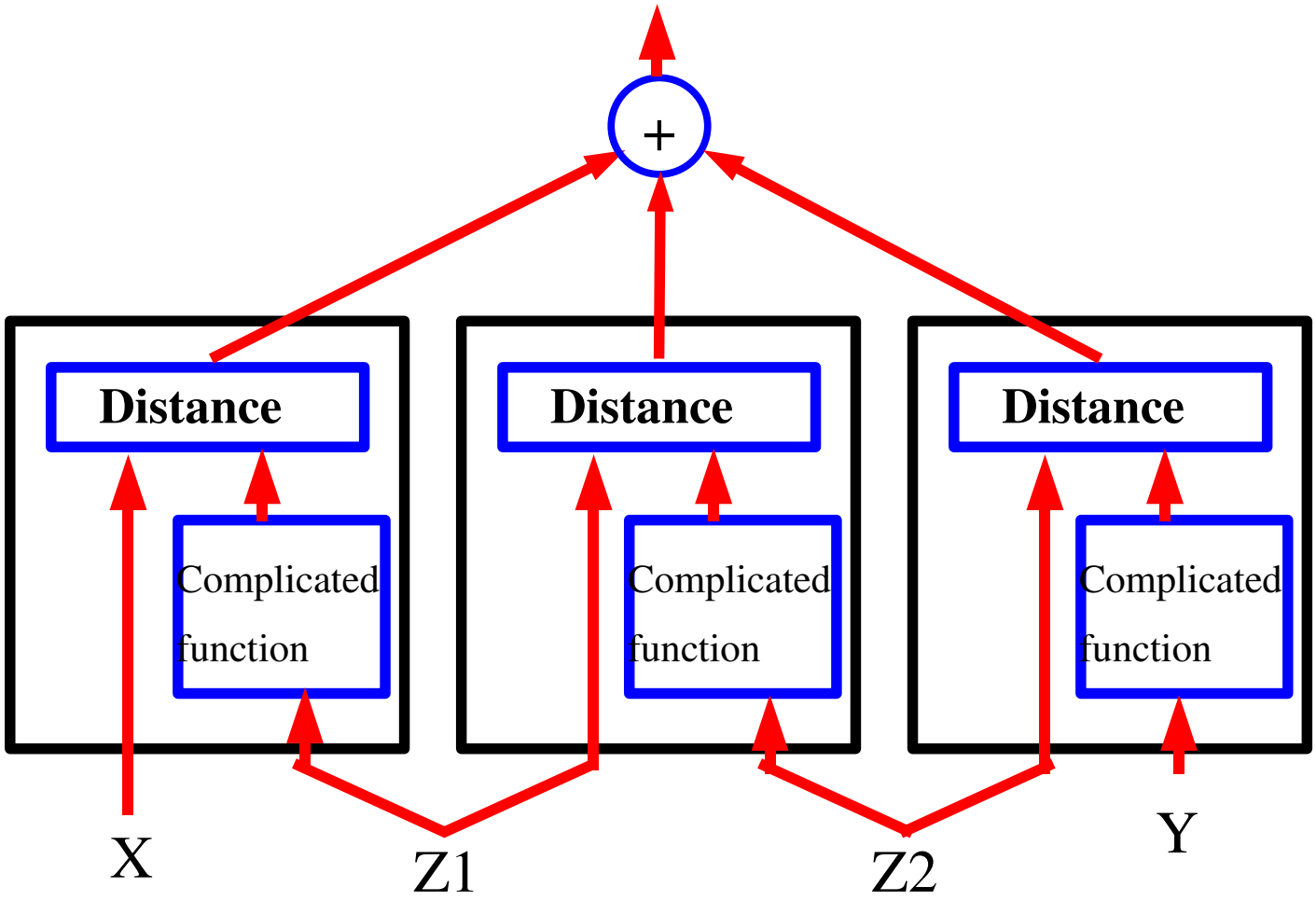
Feed-Forward (bottom up) Deep Belief Net

- Predicting Y from X is trivial
- Predicting X from Y is very hard



Feedback (top down) Deep Belief Net

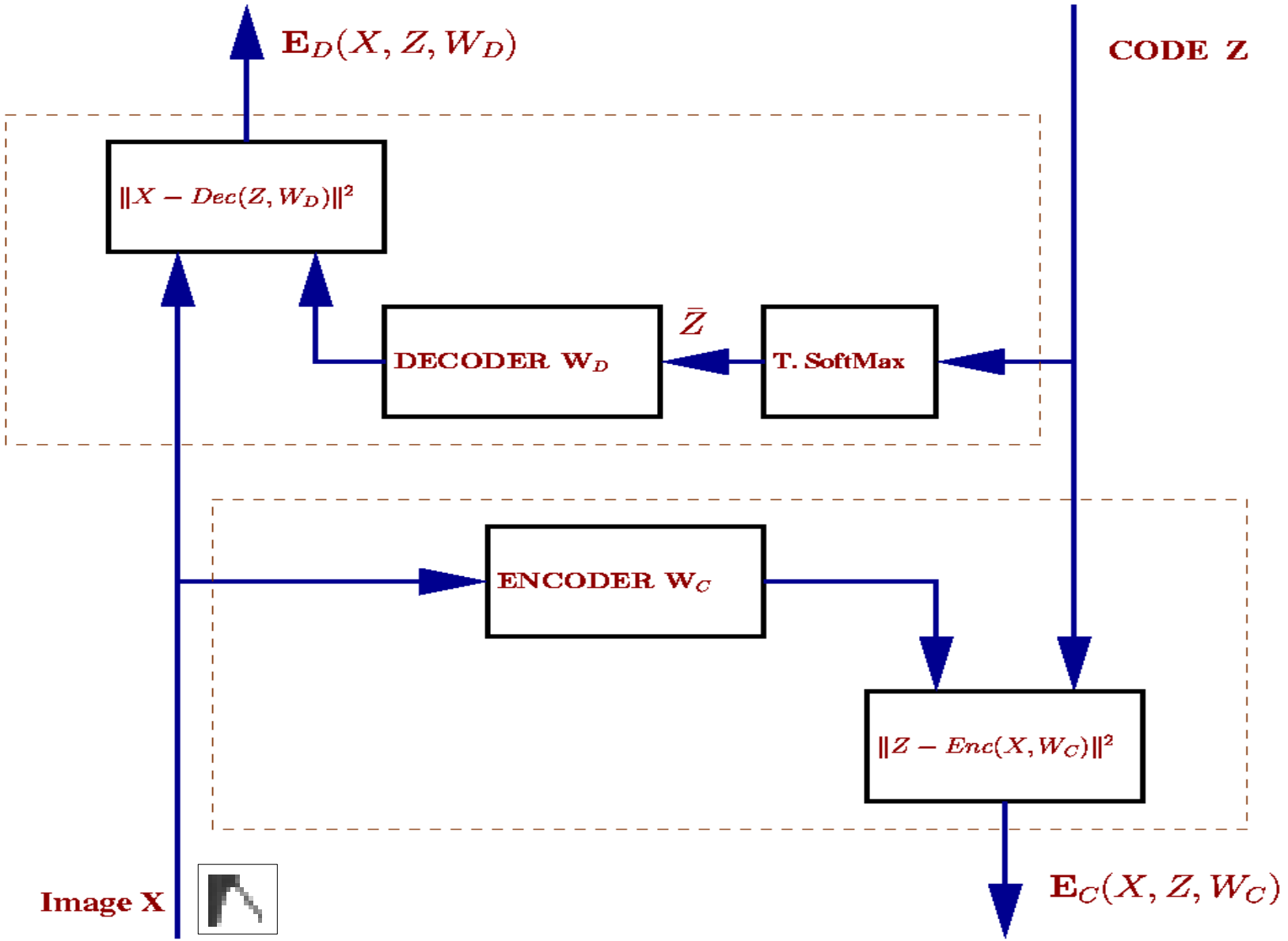
- Predicting X from Y is trivial
- Predicting Y from X is very hard



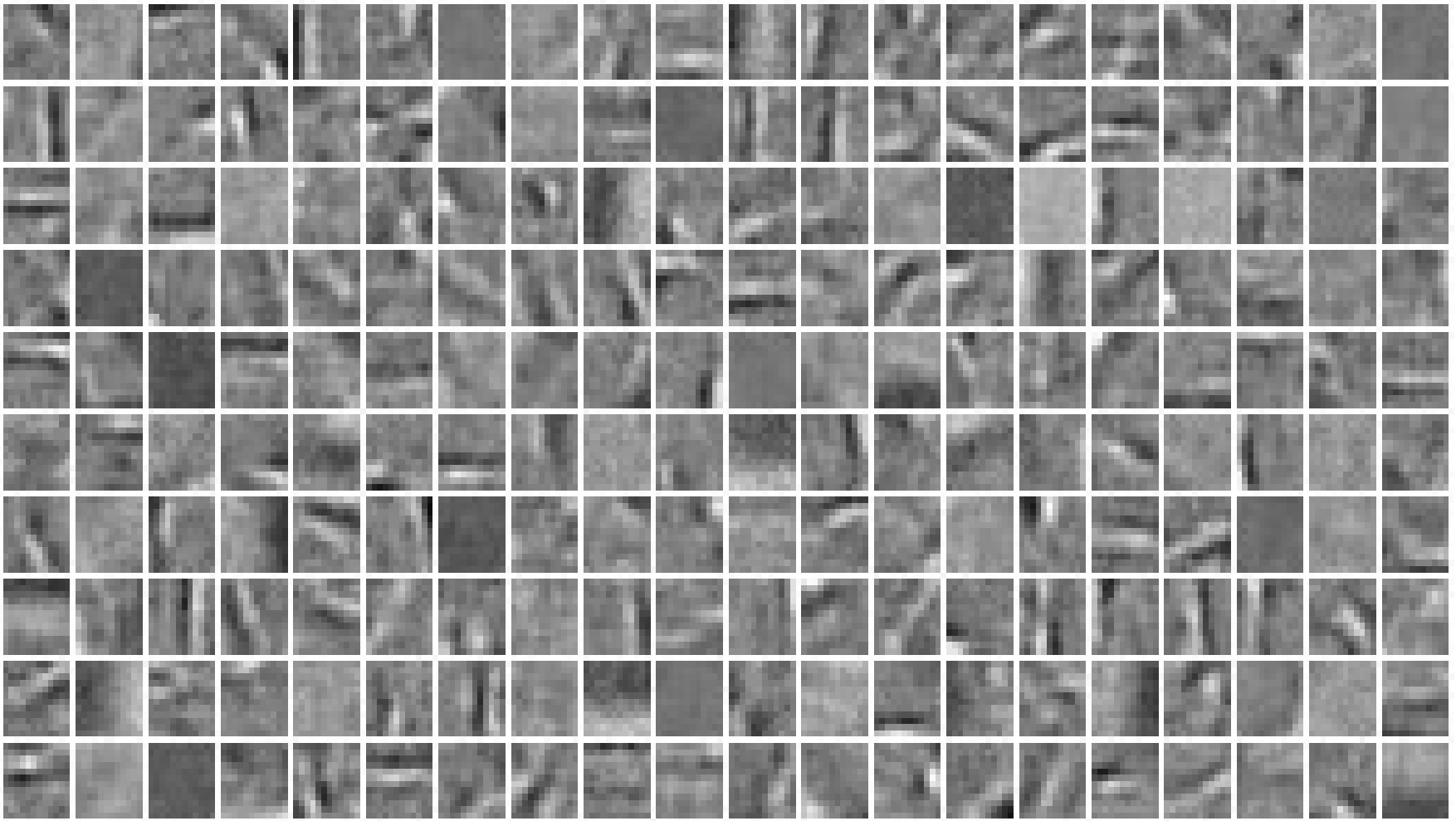
Feed Forward vs Feedback?

- **How far can we go with Feed-Forward architectures?**
  - Animals have a vested interest in picking out certain objects very quickly (e.g. Predators, preys, obstacles...)
- **Feedback (top-down) systems are slower, but smarter**
  - Couldn't we use a feedback system to drive/train a feed-forward one?
  - 1. Run the top-down inference system (slow, but good)
  - 2. Train a feed-forward system to predict the same answer
  - ==> you get a fast, bottom-up/top-down system that has the best of both worlds.

Inference & Learning

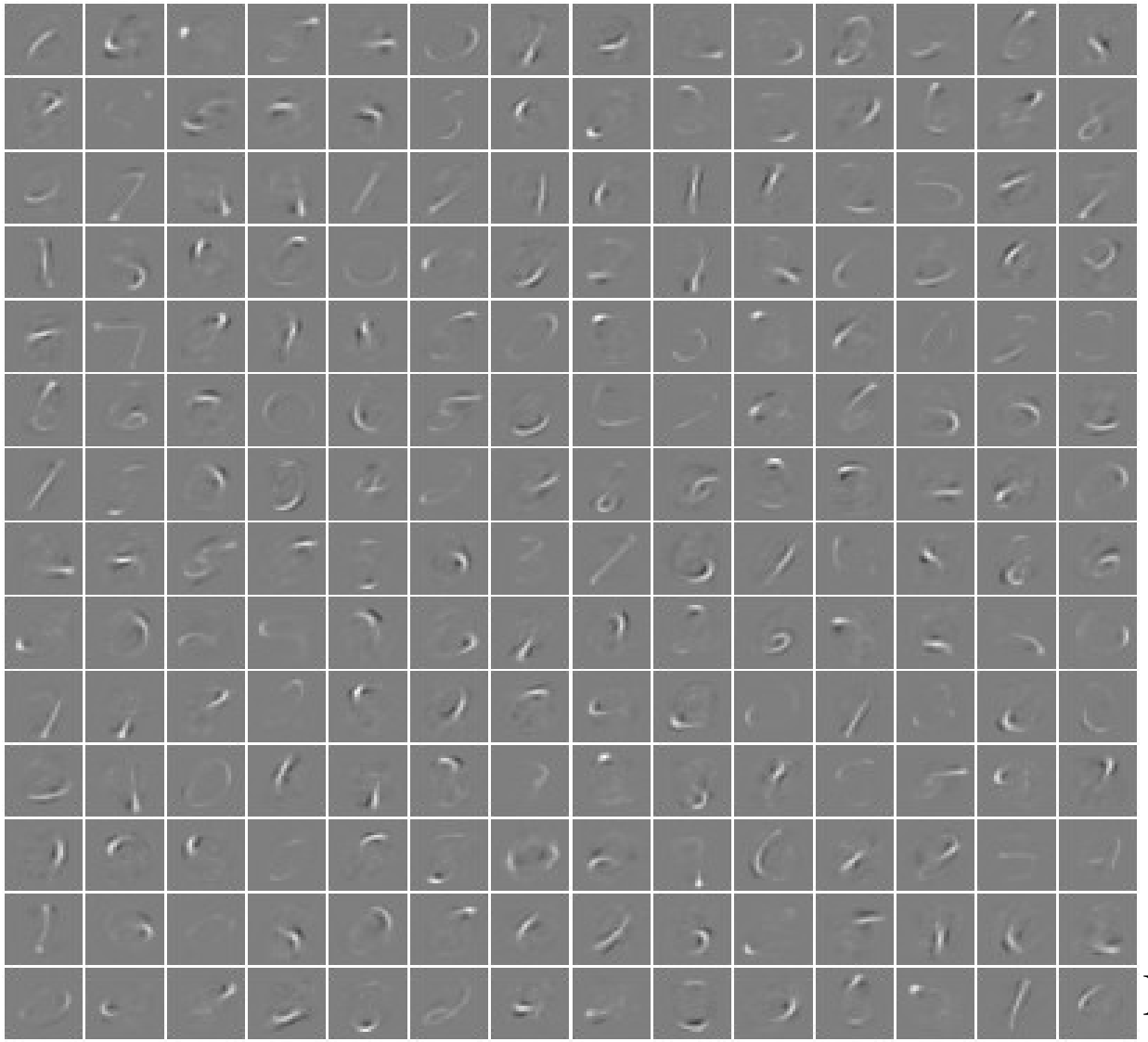


Natural image patches - Berkeley



200 decoder filters (reshaped columns of matrix $\mathbf{W_d}$)

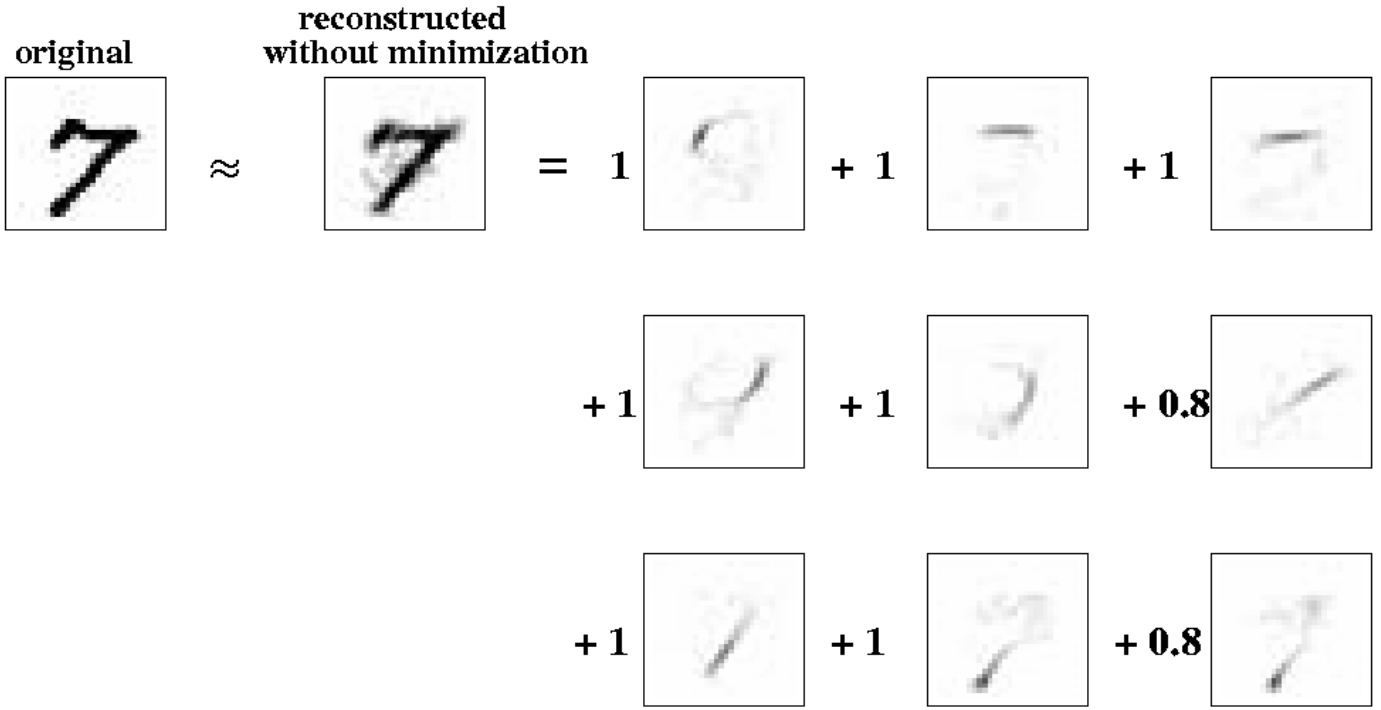
Handwritten digits - MNIST

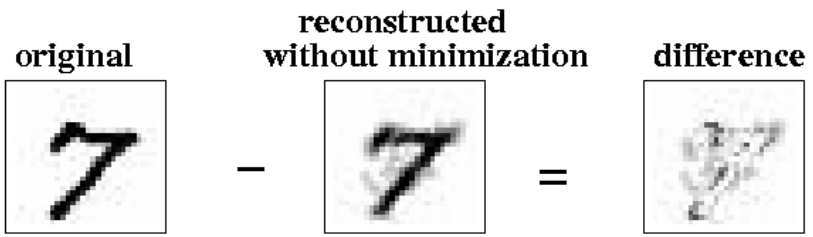


- 60,000 28x28 images
- 196 units in the code
- η 0.01
- β_1
- learning rate 0.001
- L1, L2 regularizer 0.005

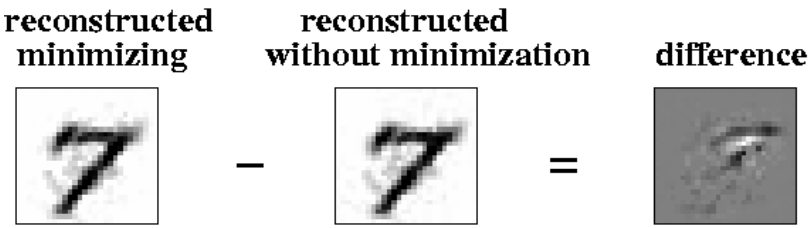
Encoder *direct* filters

Handwritten digits - MNIST



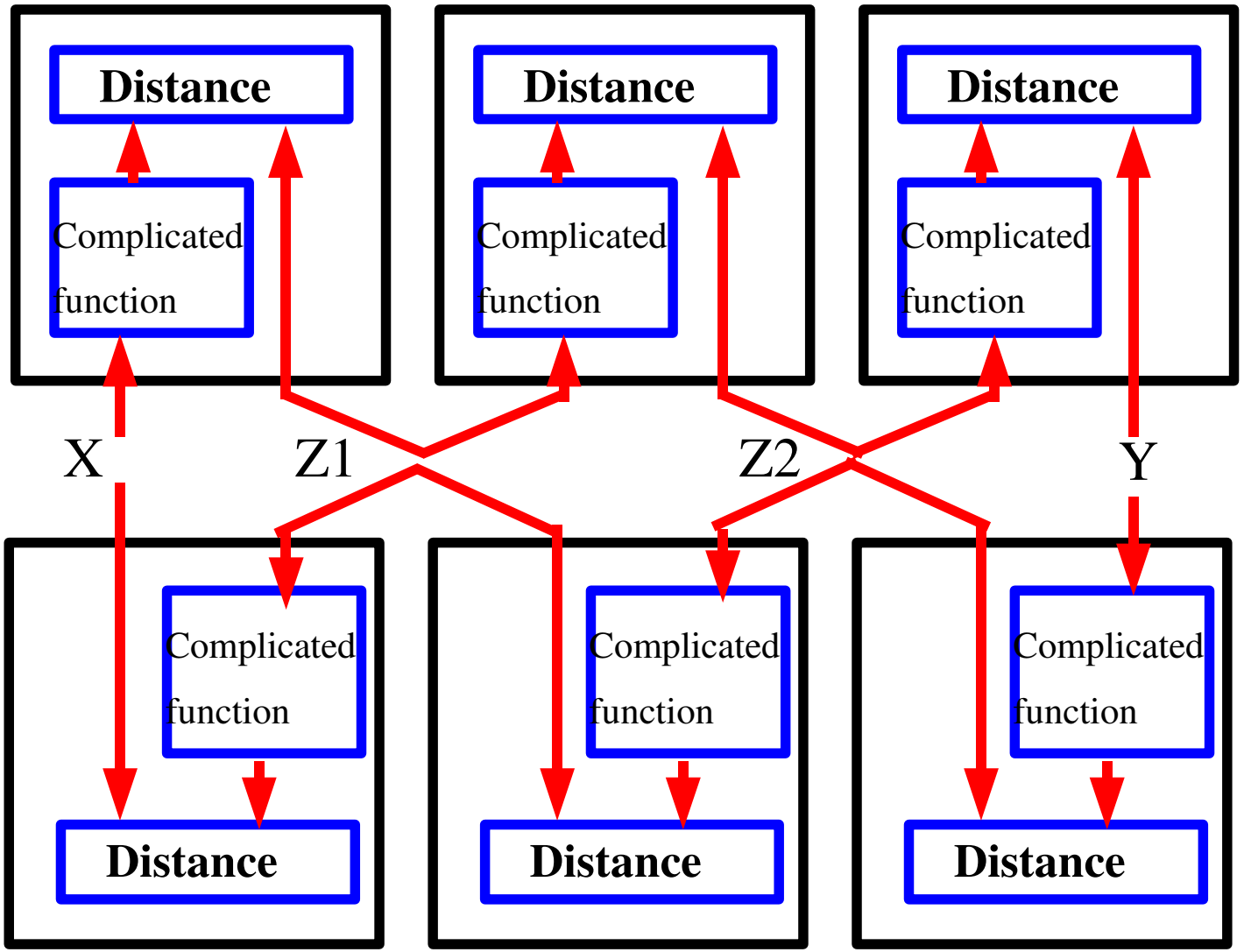


forward propagation through encoder and decoder



after training there is no need to minimize in code space

Top-Down+Bottom-Up



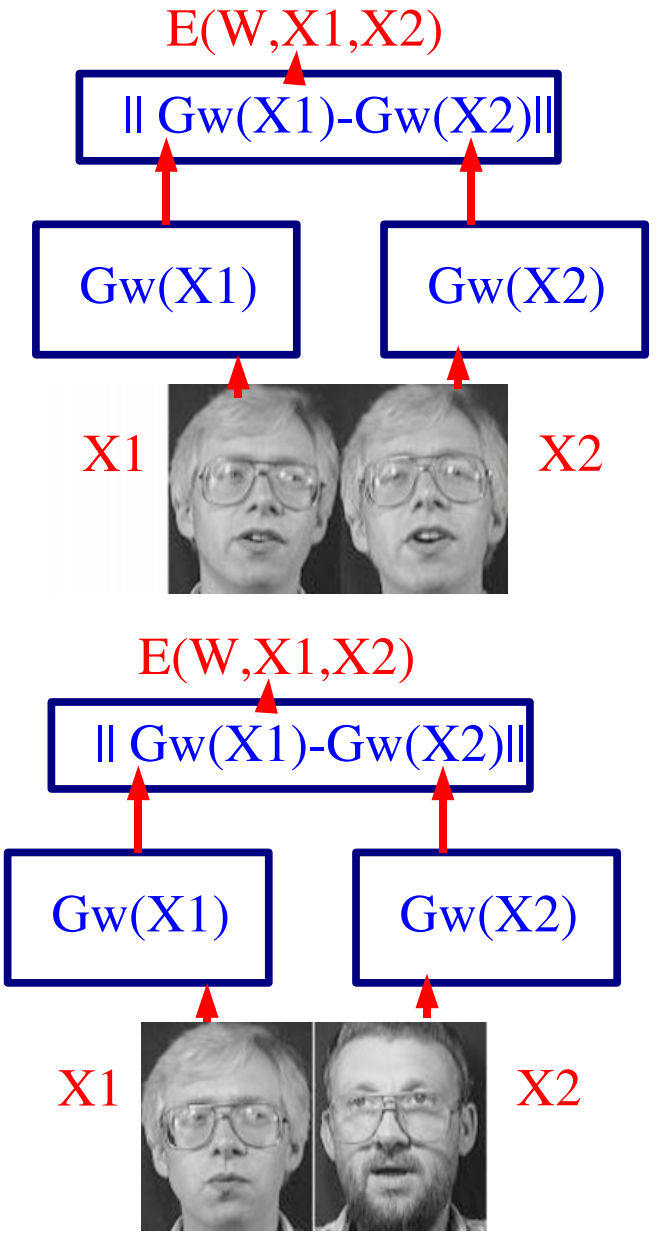
Low-dimensional representations for matching?

- **A solution to the many-class problem**
- **Example: face recognition**
  - We do not have pictures of every person
- **We must be able to learn something without seeing all the classes**
- **Possible Solution: learn a similarity metric to make matching efficient**
- **Map images to a low dimensional space in which**
  - Two images of the same person are mapped to nearby points
  - Two images of different persons are mapped to distant points

Comparing Objects: Learning an Invariant Dissimilarity Metric

[Chopra, Hadsell, LeCun CVPR 2005]

- Training a **parameterized, invariant dissimilarity metric** may be a solution to the **many-category problem**.
- Find a mapping **Gw(X)** such that the Euclidean distance **||Gw(X1)- Gw(X2)||** reflects the "semantic" distance between X1 and X2.
- Once trained, a trainable dissimilarity metric can be used to classify **new categories using a very small number of training samples** (used as prototypes).
- This is an example where probabilistic models are too constraining, because we would have to limit ourselves to models that can be normalized over the space of input pairs.
- With EBMs, we can put what we want in the box (e.g. A convolutional net).
- **Siamese Architecture**
- **Application:** face verification/recognition



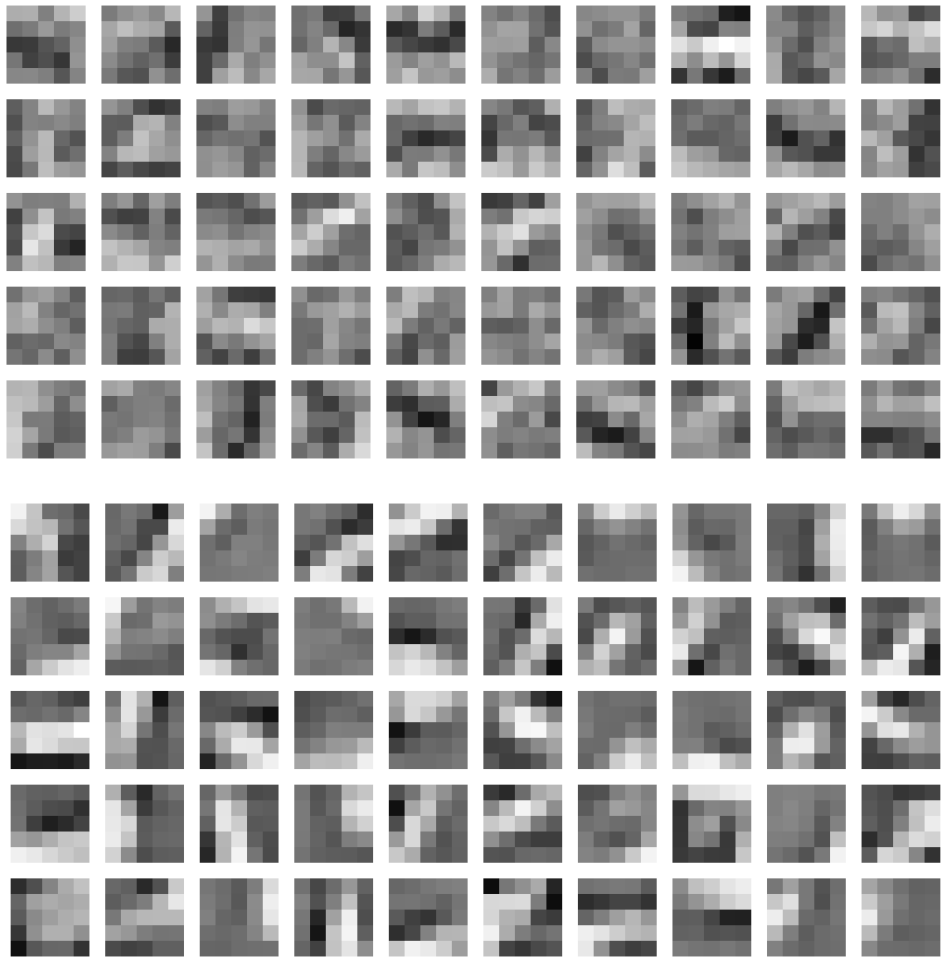
Internal state for genuine and impostor pairs

Sparse vs Dense Features?

- **Low-dimensional, dense feature representations**
  - Efficient in terms of information content per variable
  - Efficient for matching
  - Difficult to use for classification
- **High-dimensional, sparse feature representations**
  - Diluted information (inefficient for memory)
  - May be inefficient for matching
  - Easy to use for classification
- **Any family of function can be parameterized linearly using a sufficiently high-dimensional and sparse representation of the input variable**
  - e.g. Using a kernel representation
- **But that representation may be too large!**

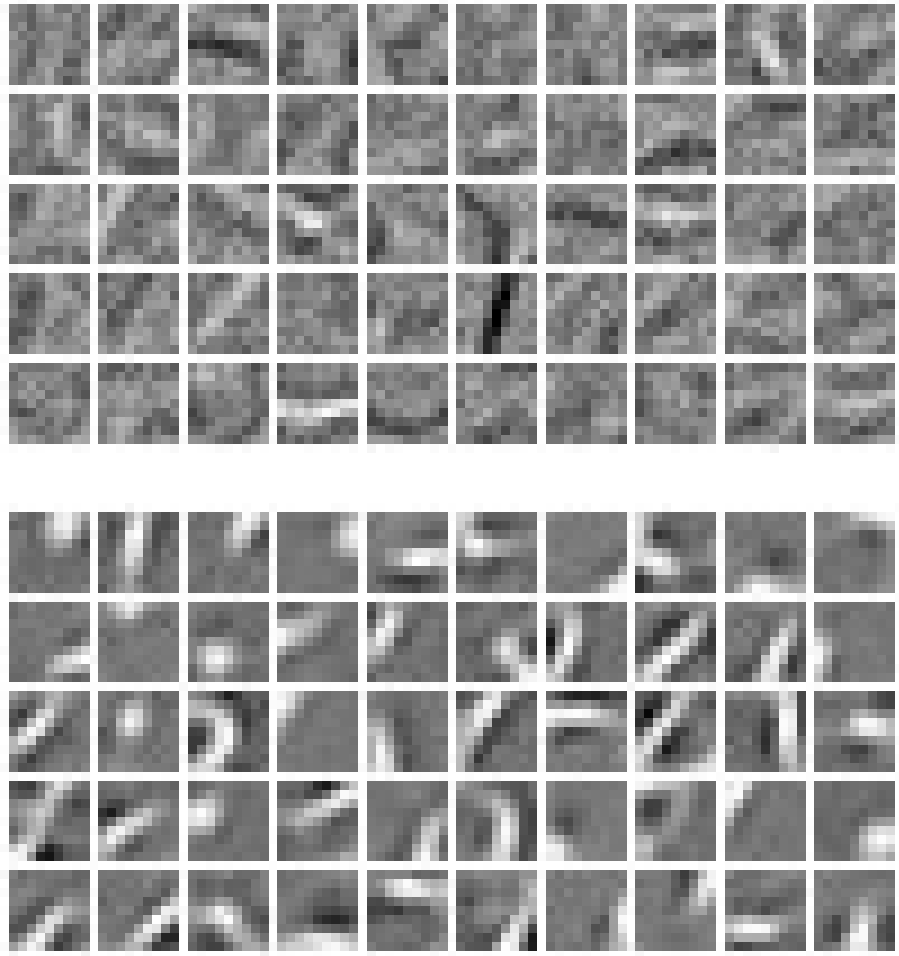
Initializing a Convolutional Net with SPoE

- **Architecture: LeNet-6**
  - 1->50->50->200->10
- **Baseline: random initialization**
  - 0.7% error on test set
- **First Layer Initialized with SpoE**
  - 0.6% error on test set
- **Training with elastically-distorted samples:**
  - 0.38% error on test set



Initializing a Convolutional Net with SPoE

- **Architecture: LeNet-6**
  - 1->50->50->200->10
  - 9x9 kernels instead of 5x5
- **Baseline: random initialization**
- **First Layer Initialized with SpoE**



Best Results on MNIST (from raw images: no preprocessing)

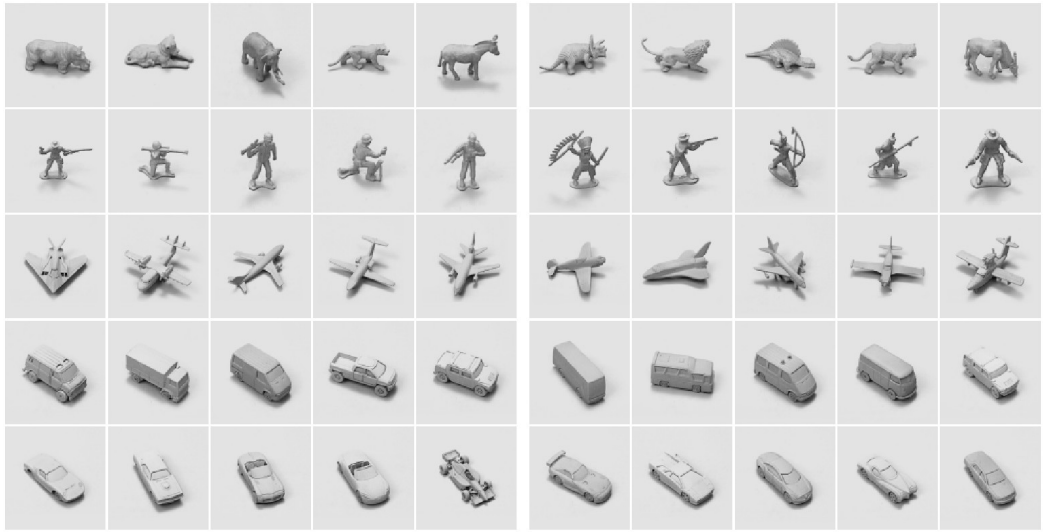
CLASSIFIER	DEFORMATION	ERROR	Reference
Knowledge-free methods			
2-layer NN, 800 HU, CE		1.60	Simard et al., ICDAR 2003
3-layer NN, 500+300 HU, CE, reg		1.53	Hinton, in press, 2005
SVM, Gaussian Kernel		1.40	Cortes 92 + Many others
Unsupervised Stacked RBM + backprop		0.95	Hinton, in press, 2005
Convolutional nets			
Convolutional net LeNet-5,		0.80	LeCun 2005 Unpublished
Convolutional net LeNet-6,		0.70	LeCun 2006 Unpublished
Conv. net LeNet-6- + unsup learning		0.60	LeCun 2006 Unpublished
Training set augmented with Affine Distortions			
2-layer NN, 800 HU, CE	Affine	1.10	Simard et al., ICDAR 2003
Virtual SVM deg-9 poly	Affine	0.80	Scholkopf
Convolutional net, CE	Affine	0.60	Simard et al., ICDAR 2003
Training et augmented with Elastic Distortions			
2-layer NN, 800 HU, CE	Elastic	0.70	Simard et al., ICDAR 2003
Convolutional net, CE	Elastic	0.40	Simard et al., ICDAR 2003
Conv. net LeNet-6- + unsup learning	Elastic	0.38	LeCun 2006 Unpublished

Convexity is Overrated

- **Using a suitable architecture (even if it leads to non-convex loss functions) is more important than insisting on convexity (particularly if it restricts us to unsuitable architectures)**
  - e.g.: Shallow (convex) classifiers versus Deep (non-convex) classifiers
- **Even for shallow/convex architecture, such as SVM, using non-convex loss functions actually improves the accuracy and speed**
  - See "trading convexity for efficiency" by Collobert, Bottou, and Weston, ICML 2006 (best paper award)

Normalized-Uniform Set: Error Rates

- **Linear Classifier on raw stereo images:** **30.2% error.**
- **K-Nearest-Neighbors on raw stereo images:** **18.4% error.**
- **K-Nearest-Neighbors on PCA-95:** **16.6% error.**
- **Pairwise SVM on 96x96 stereo images:** **11.6% error**
- **Pairwise SVM on 95 Principal Components:** **13.3% error.**
- **Convolutional Net on 96x96 stereo images:** **5.8% error.**



Training instances Test instances

Normalized-Uniform Set: Learning Times

	SVM	Conv Net				SVM/Conv
test error	11.6%	10.4%	6.2%	5.8%	6.2%	5.9%
train time (min*GHz)	480	64	384	640	3,200	50+
test time per sample (sec*GHz)	0.95	0.03				0.04+
#SV	28%					28%
parameters	σ=2,000 C=40					dim=80 σ=5 C=0.01

SVM: using a parallel implementation by Graf, Durdanovic, and Cosatto (NEC Labs)

Chop off the last layer of the convolutional net and train an SVM on it

Experiment 2: Jittered-Cluttered Dataset

- 291,600 training samples, 58,320 test samples
- SVM with Gaussian kernel — 43.3% error
- Convolutional Net with binocular input: — 7.8% error
- Convolutional Net + SVM on top: — 5.9% error
- Convolutional Net with monocular input: — 20.8% error
- Smaller mono net (DEMO): — 26.0% error
- Dataset available from http://www.cs.nyu.edu/~yann

Jittered-Cluttered Dataset

	SVM	Conv Net			SVM/Conv
test error	43.3%	16.38%	7.5%	7.2%	5.9%
train time (min*GHz)	10,944	420	2,100	5,880	330+
test time per sample (sec*GHz)	2.2	0.04			0.06+
#SV	5%				2%
parameters	σ=10^4 C=40				dim=100 σ=5 C=1

OUCH!

The convex loss, VC bounds and representers theorems don't seem to help

Chop off the last layer, and train an SVM on it it works!

Optimization algorithms for learning

- **Neural nets:**
  - conjugate gradient, BFGS, LM-BFGS, ... don't work as well as stochastic gradient
- **SVM:**
  - "batch" quadratic programming methods don't work as well as SMO. SMO don't work as well as recent on-line methods
- **CRF:**
  - Iterative scaling (or whatever) doesn't work as well as stochastic gradient (Schraudolph et al ICML 2006)
  - The discriminative learning folks in speech and handwriting recognition have known this for a long time
- **Stochastic gradient has no good theoretical guarantees**
  - That doesn't mean we shouldn't use them, because the empirical evidence that it works better is overwhelming

Theoretical Guarantees are overrated

- When Empirical Evidence suggests a fact for which we don't have theoretical guarantees, it just means the theory is inadequate.
- When empirical evidence and theory disagree, the theory is wrong.
- Let's not be afraid of methods for which we have no theoretical guarantee, particularly if they have been shown to work well
- But, let's aggressively look to those theoretical guarantees.
- We should use our theoretical understanding to expand our creativity, not to restrict it.