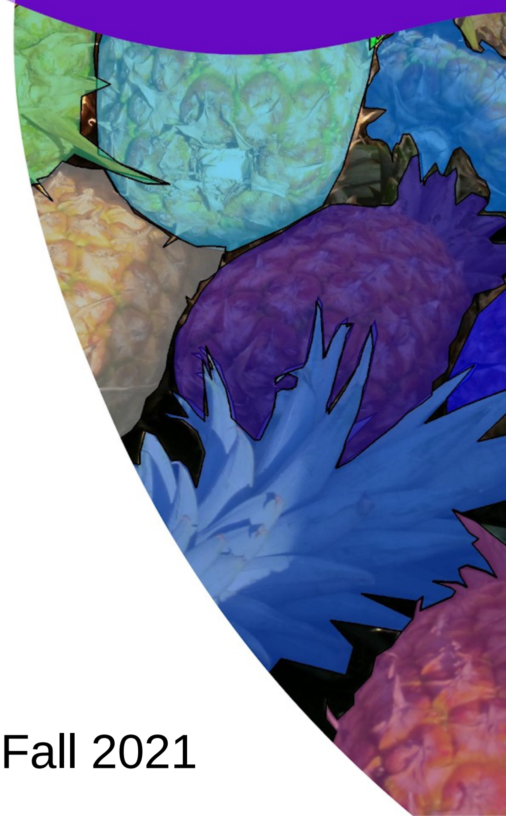
NEW YORK UNIVERSITY

Self-Supervised Learning

Yann LeCun
NYU - Courant Institute & Center for Data Science
Facebook AI Research
http://yann.lecun.com



Deep Learning, NYU, Fall 2021

FACEBOOK AI

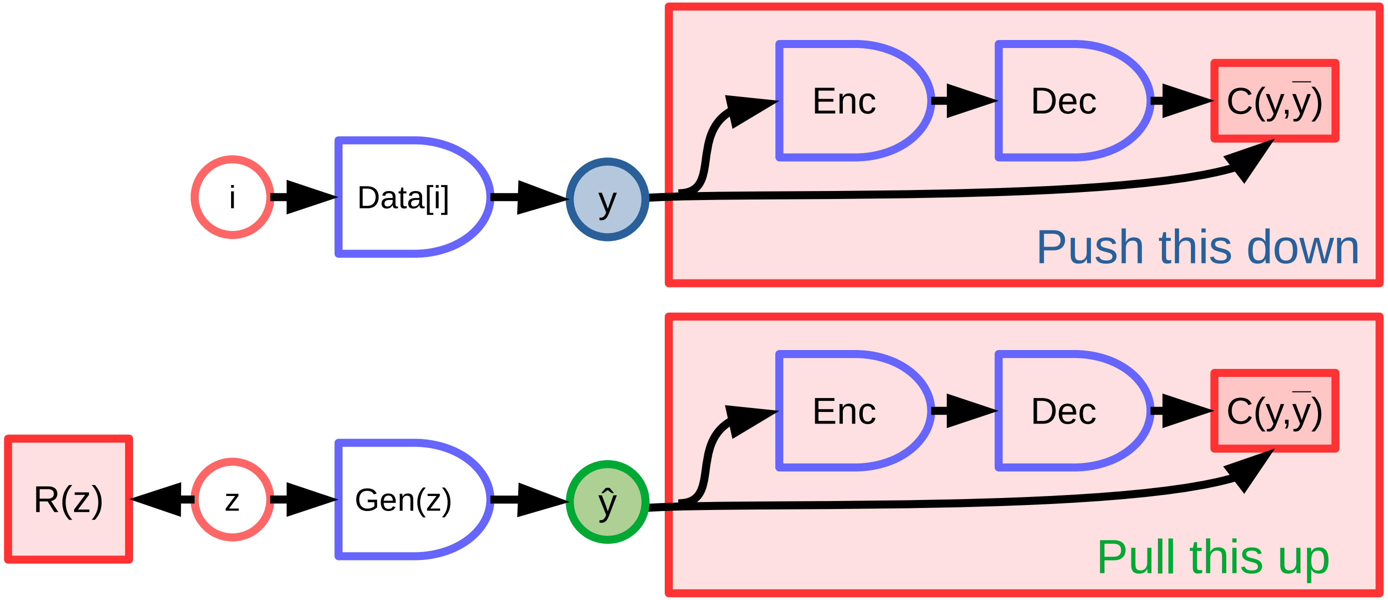
Before that, let’s revisit GANs

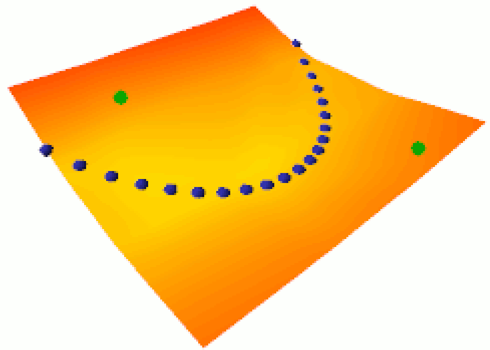
GANs are contrastive EBMs

GANs are secretly a contrastive method for EBM

- **Energy-Based GAN** [Zhao 2016]**, Wasserstein GAN** [Arjovsky 2017]**,...**
  - GANs generate nice images
  - But learning representations of image has not been very successful.

$$\mathcal{L}(x, y, \hat{y}, w) = H(F_w(x, y), F_w(x, \hat{y}), m(y, \hat{y}))$$



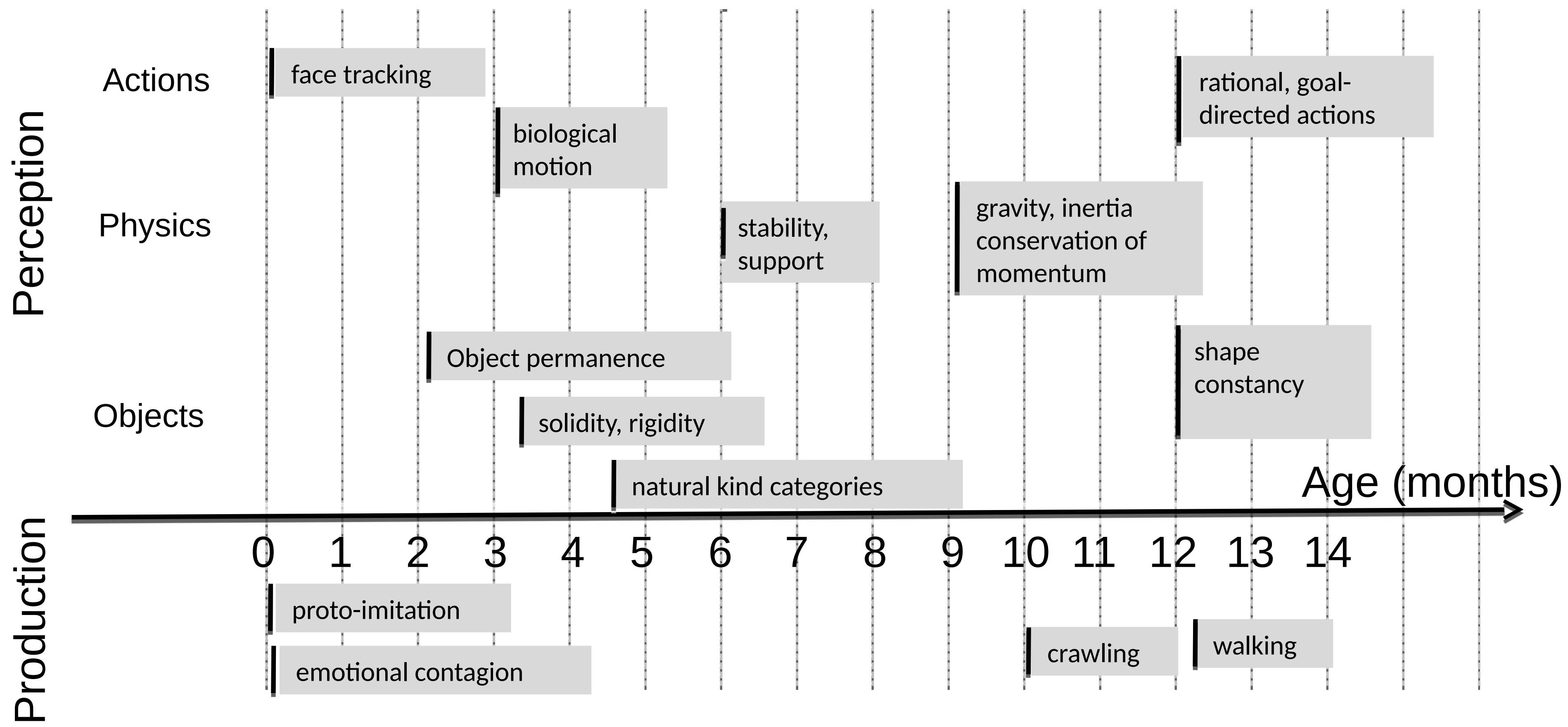


FACEBOOK AI

How do humans and animals learn so quickly?

Not supervised.
Not Reinforced.

When infants learn models of the world [after Emmanuel Dupoux]

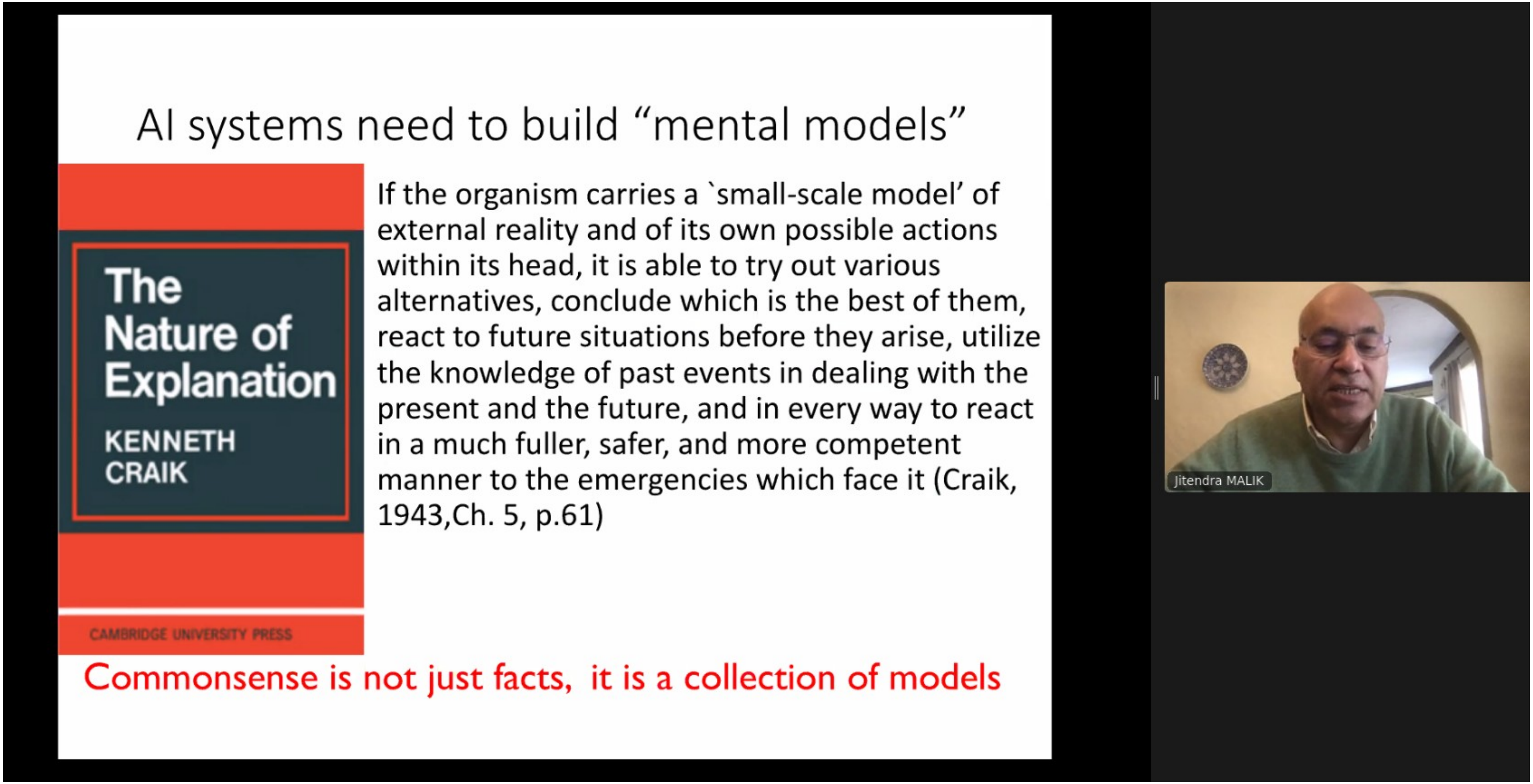


How do Human and Animal Babies Learn?

- **How do they learn how the world works?**
- **Largely by observation, with remarkably little interaction (initially).**
- **They accumulate enormous amounts of background knowledge**
  - About the structure of the world, like intuitive physics.
- **Perhaps common sense emerges from this knowledge?**

Photos courtesy of Emmanuel Dupoux

From Jitendra Malik's talk



FACEBOOK AI

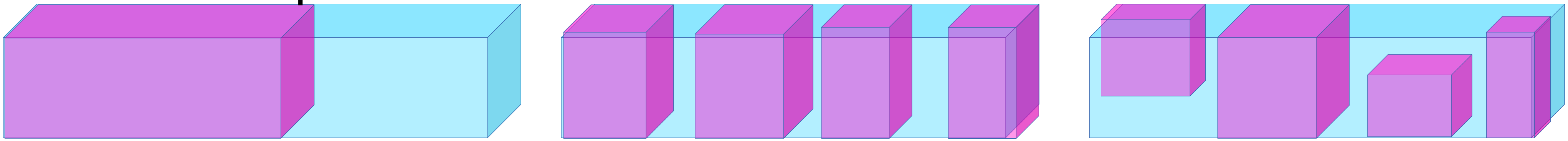
Self-Supervised Learning

Capture dependencies.
Predict everything from everything else.

Self-Supervised Learning = Learning to Fill in the Blanks

- **Reconstruct the input or Predict missing parts of the input.**

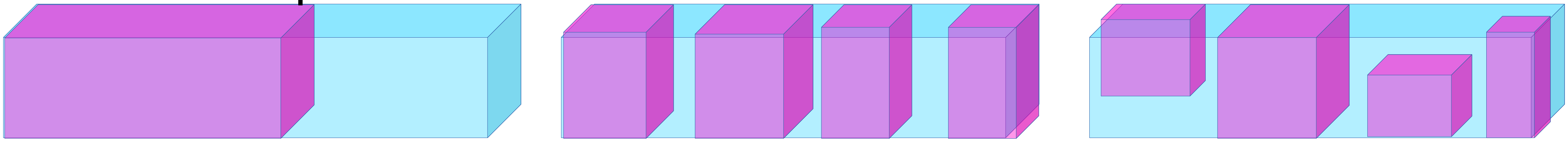
time or space →



Self-Supervised Learning = Learning to Fill in the Blanks

- **Reconstruct the input or Predict missing parts of the input.**

time or space →



Two Uses for Self-Supervised Learning

- **1. Learning hierarchical representations of the world**
  - SSL pre-training precedes a supervised or RL phase
- **2. Learning predictive (forward) models of the world**
  - Learning models for Model-Predictive Control, policy learning for control, or model-based RL.
- **Question: how to represent uncertainty & multi-modality in the prediction?**

FACEBOOK AI

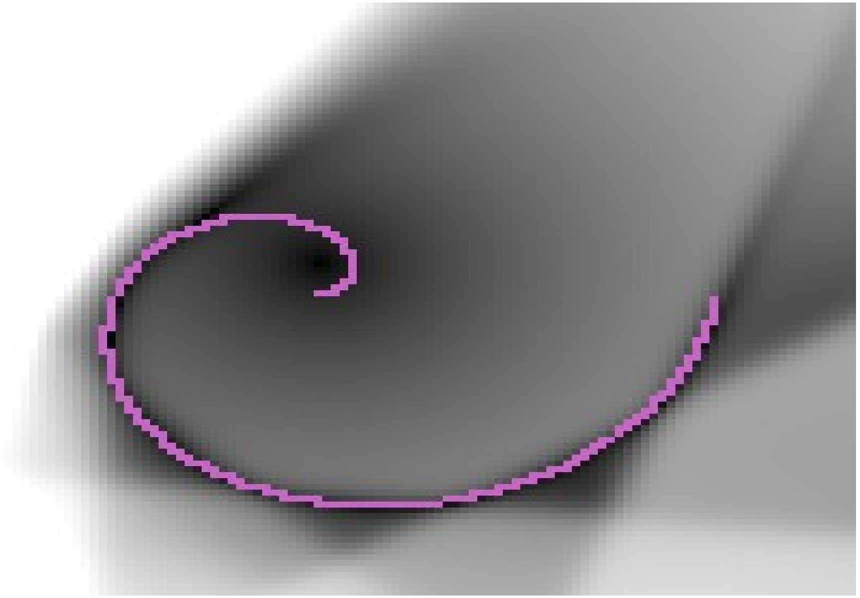
Sparse Coding
Sparse Modeling

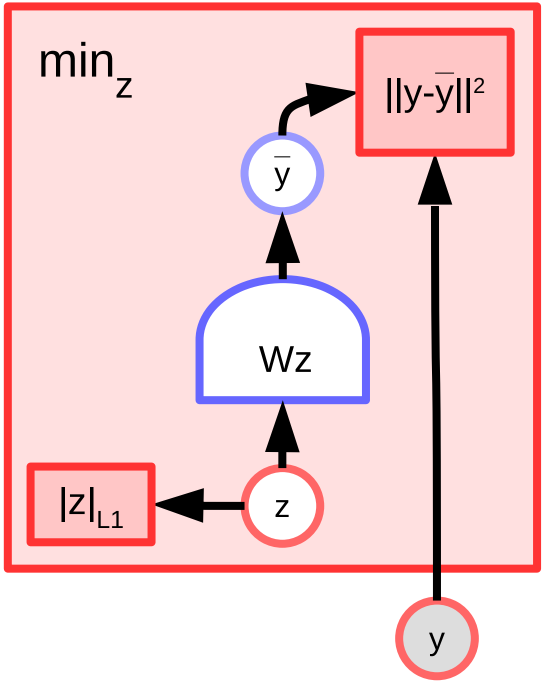
Regularized latent-variable
generative EBM with sparsity penalty

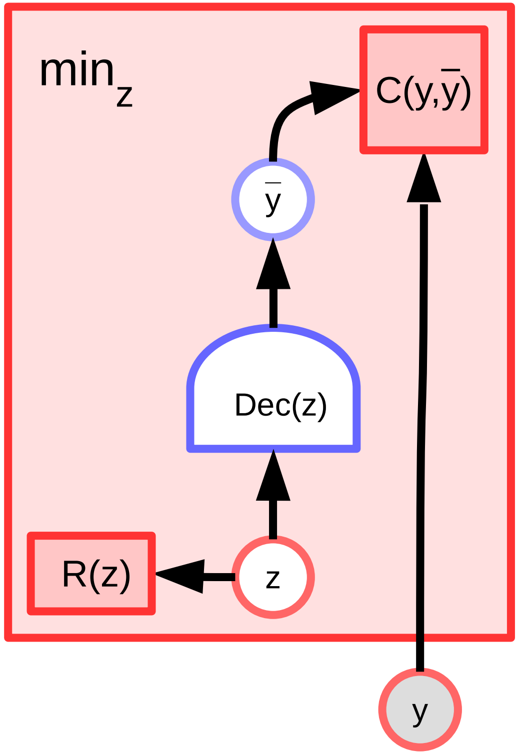
Unconditional Regularized Latent Variable EBM

- **Unconditional form. Reconstruction (no x, no predictor).**
- **Example: sparse coding / sparse modeling**
  - Linear decoder
  - L1 regularizer on Z

$$E(y,z)=\|y-Wz\|^2+\lambda|z|$$







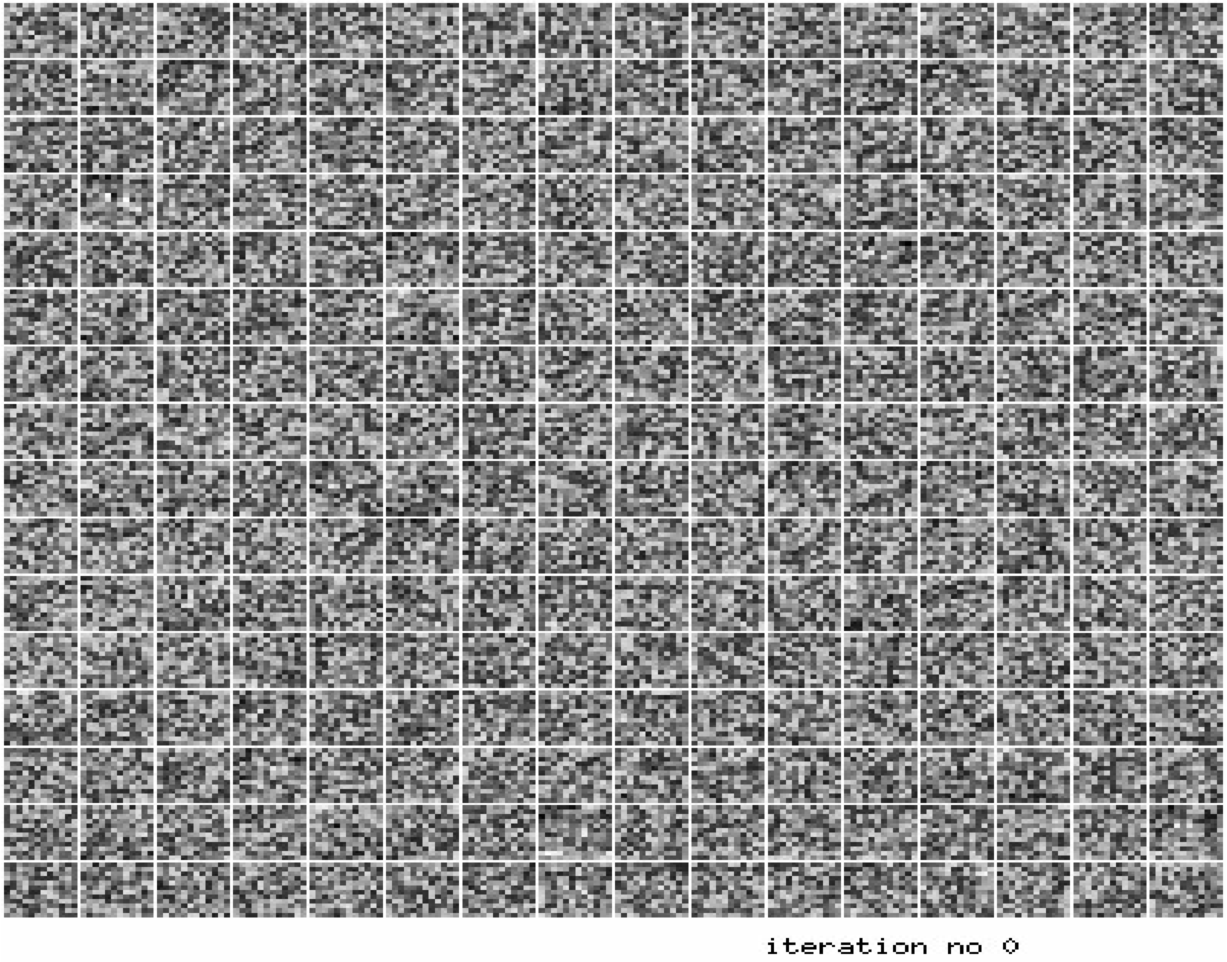
Sparse Modeling [Olshausen & Field 1997]

- **Energy function** $$E(y,z) = ||y - wz||^2 + \alpha \sum_j |z_j|$$
  - Capacity of latent variable z limited by L1 norm (sparsity)
  - Columns of dictionary matrix w are normalized

- **Procedure: repeat:**
  - 1. pick a training sample y
  - 2. inference: find the optimal z $$\check{z} = \text{argmin}_z E(y,z)$$
    - Use the ISTA algorithm $$z(t+1) = \text{Shrink}_{\alpha\eta}\left[z(t) - \eta w^t (wz(t) - y)\right]$$
  - 3. Update W $$w \leftarrow w - \eta \partial E(y,\check{z})/\partial w = w + \eta z (y - wz)^t$$
  - 4. Normalize columns of W to a constant (e.g. 1).

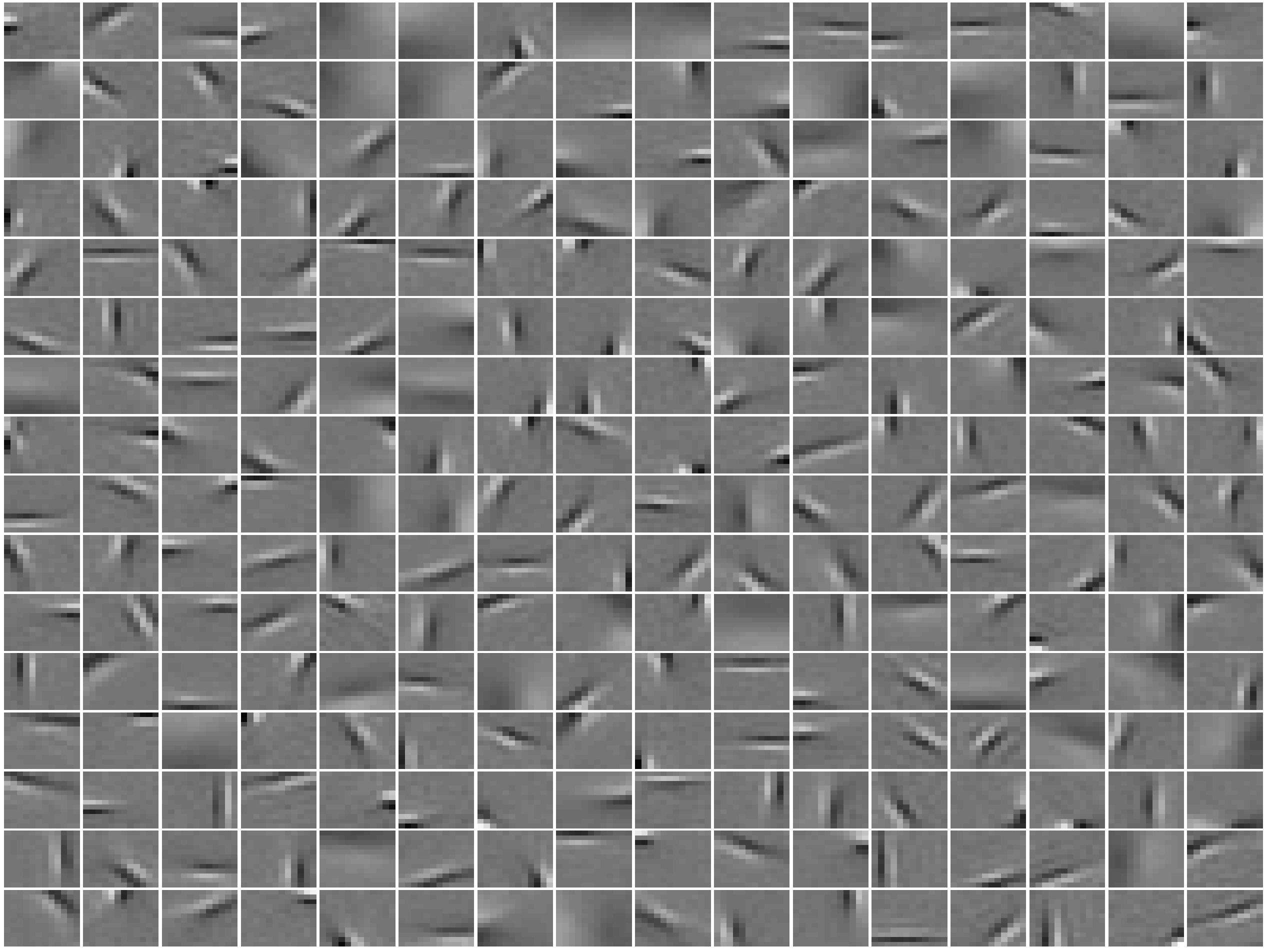
Predictive Sparse Decomposition (PSD): Training

- **Training on natural images patches.**
  - 12X12
  - 256 basis functions



Learned Features on natural patches: V1-like receptive fields

Y. LeCun



Amortized Inference

- **Training an encoder to give an approximate solution to the inference optimization problem**

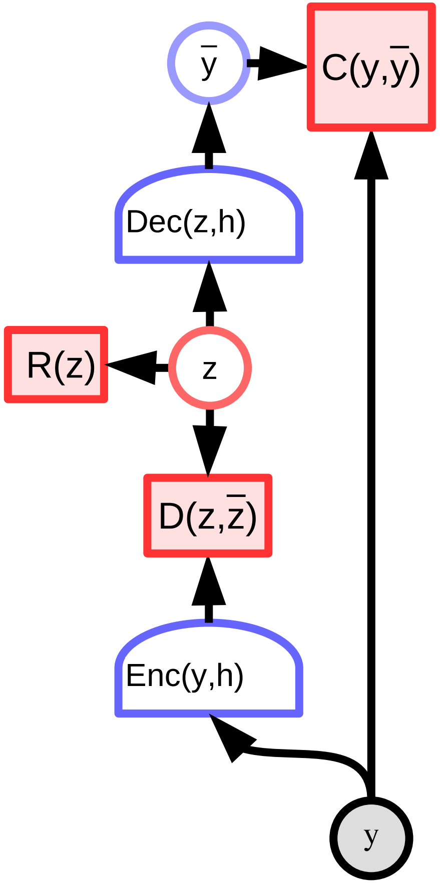
- **Regularized Auto-Encoder, Sparse AE, LISTA**

$$\mathrm{E(y,z)} = \mathrm{C(y,\ Dec((z))} + \mathrm{D(z,Enc(y))} + \lambda R(z)$$

$$F(y) = \min_z E(y,z)$$

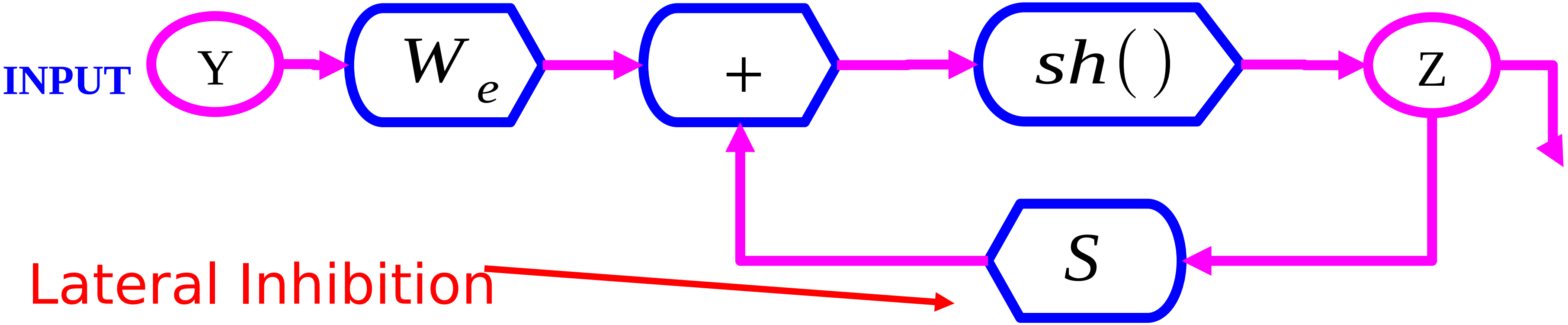
- **Variational AE**
  - Approximated by sampling and variational approximation

$$F(y) = -\log \int_z e^{-E(y,z)}$$



Giving the “right” structure to the encoder

- **ISTA/FISTA: iterative algorithm that converges to optimal sparse code**



$$z(t+1) = \mathrm{Shrink}_{\alpha\eta}\left[z(t) - \eta w^t(wz(t) - y)\right]$$

- **ISTA/FastISTA reparameterized:**

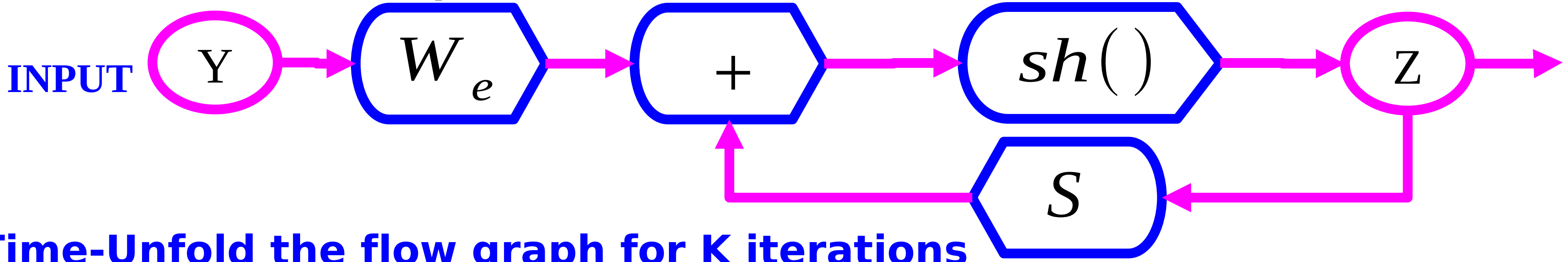
$$z(t+1) = \mathrm{Shrink}_{\alpha\eta}\left[sz(t) + w_e y\right]; \quad w_e = \eta w; \quad s = I - \eta w^T w$$

- **LISTA (Learned ISTA): learn the We and S matrices to get fast solutions**

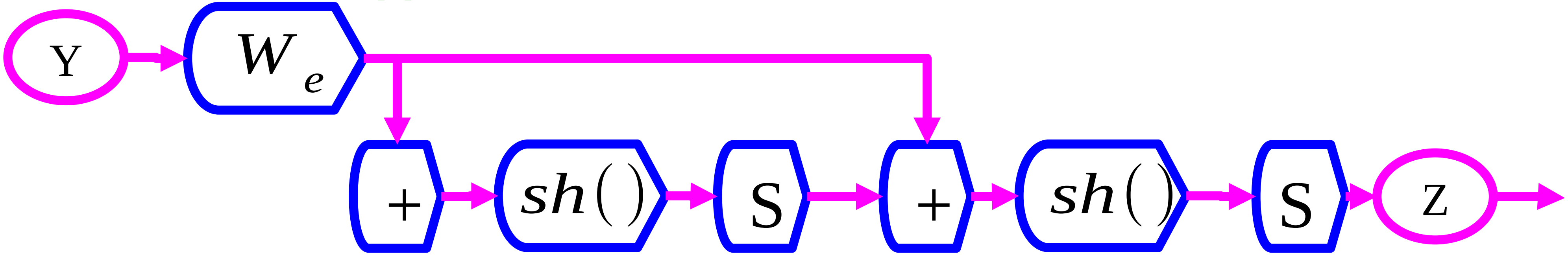
[Gregor & LeCun, ICML 2010], [Bronstein et al. ICML 2012], [Rolfe & LeCun ICLR 2013]

LISTA: Train We and S matrices to give a good approximation quickly

- **Think of the Fast ISTA flow graph as a recurrent neural net where We and S are trainable parameters**

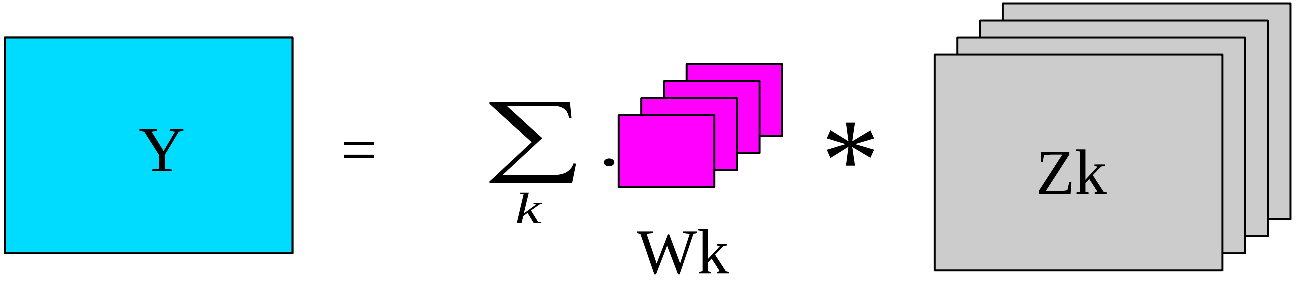


- **Time-Unfold the flow graph for K iterations**
- **Learn the We and S matrices with "backprop-through-time"**
- **Get the best approximate solution within K iterations**



Convolutional Sparse Coding

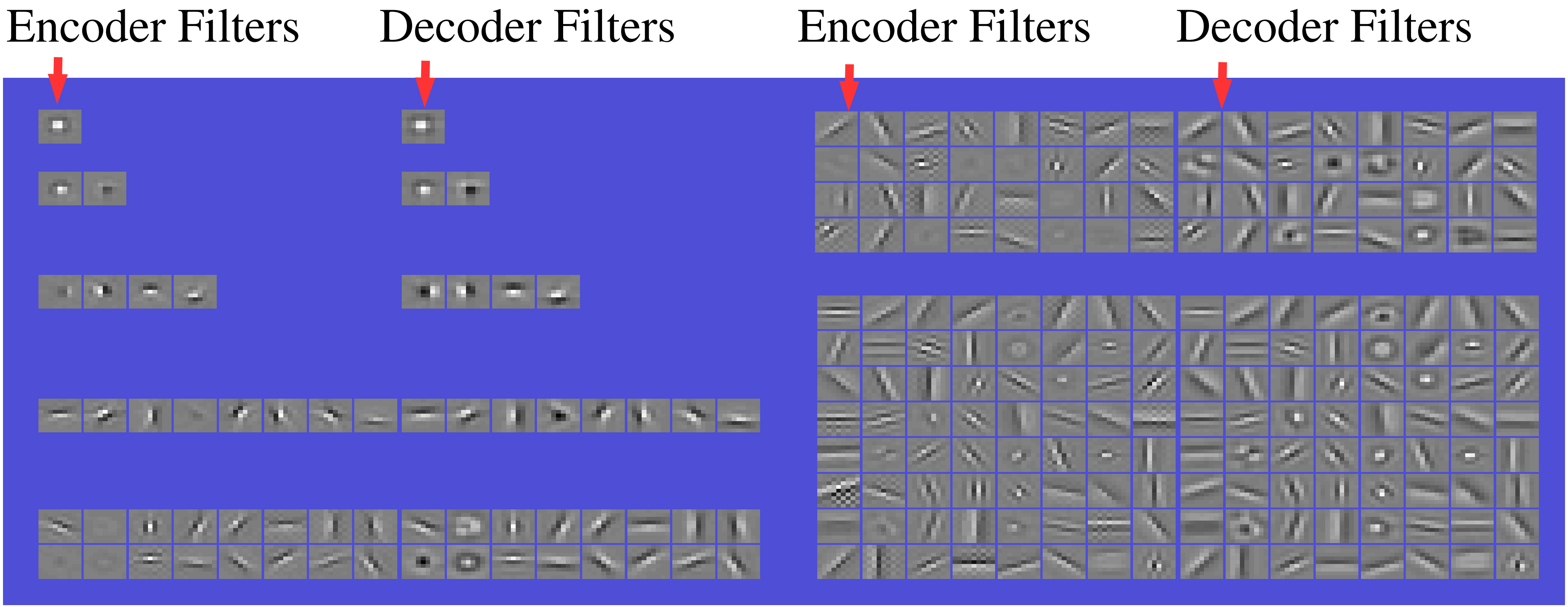
- **Replace the dot products with dictionary element by convolutions.**
  - Input Y is a full image
  - Each code component Zk is a feature map (an image)
  - Each dictionary element is a convolution kernel
- **Regular sparse coding** $E(Y,Z) = ||Y - \sum_k W_k Z_k||^2 + \alpha \sum_k |Z_k|$
- **Convolutional S.C.** $E(Y,Z) = ||Y - \sum_k W_k * Z_k||^2 + \alpha \sum_k |Z_k|$



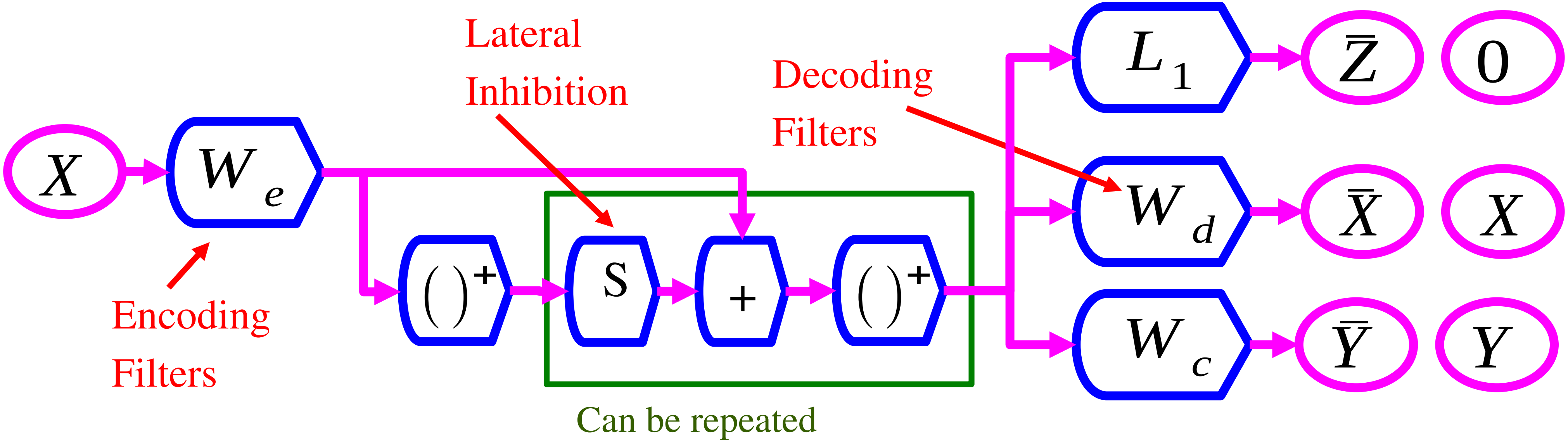
Also used in “deconvolutional networks” [Zeiler, Taylor, Fergus CVPR 2010]

Convolutional Sparse Auto-Encoder on Natural Images

- **Encoder filters and decoder filters. Decoder is linear (convolutional)**
  - with 1, 2, 4, 8, 16, 32, and 64 filters [Kavukcuoglu NIPS 2010]



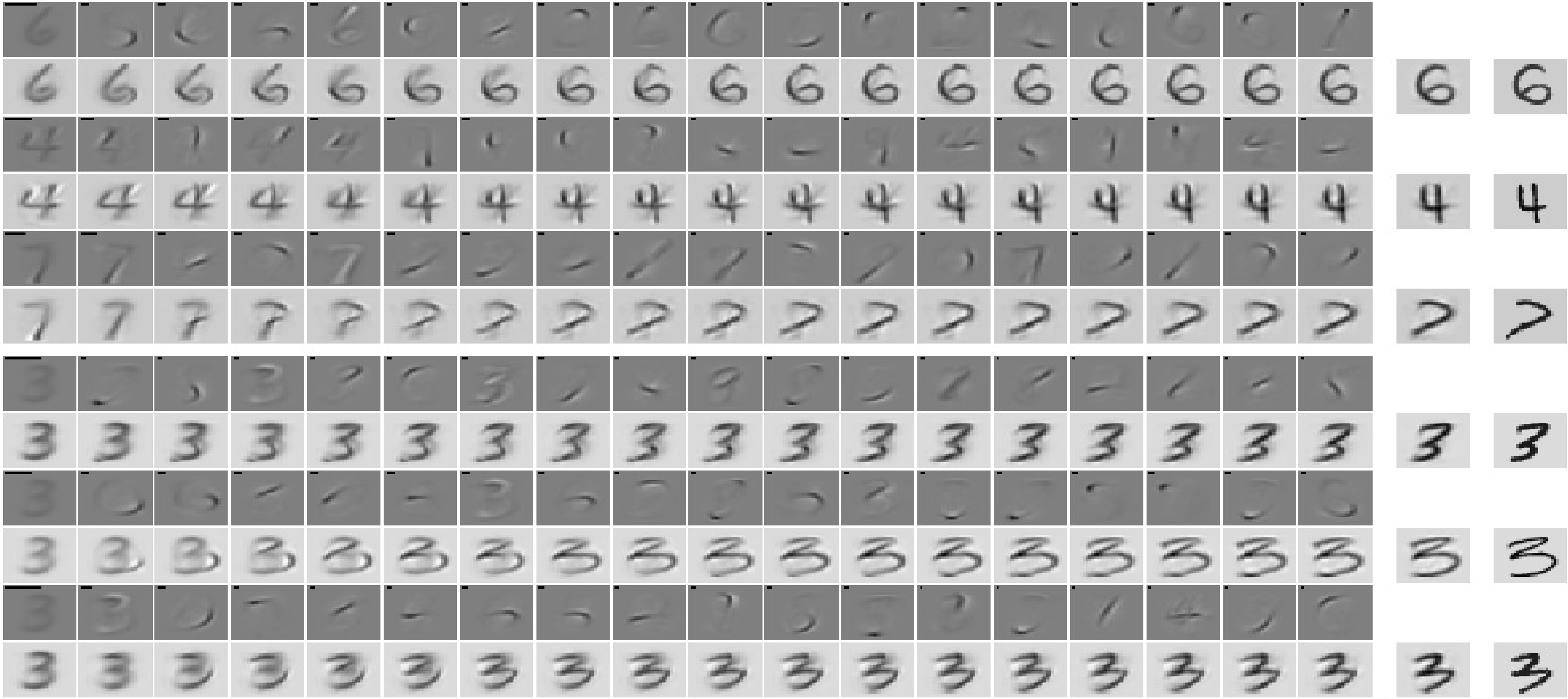
Discriminative Recurrent Sparse Auto-Encoder (DrSAE)



[Rolfe & LeCun ICLR 2013]

DrSAE Discovers manifold structure of handwritten digits

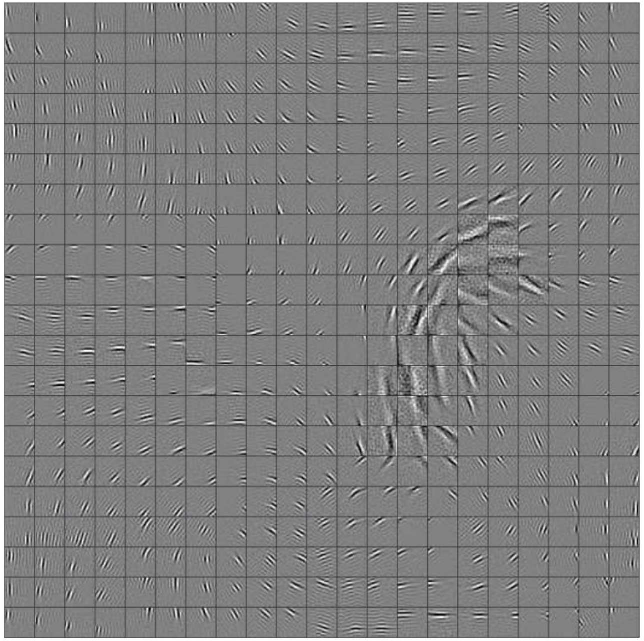
- Image = prototype + sparse sum of “parts” (to move around the



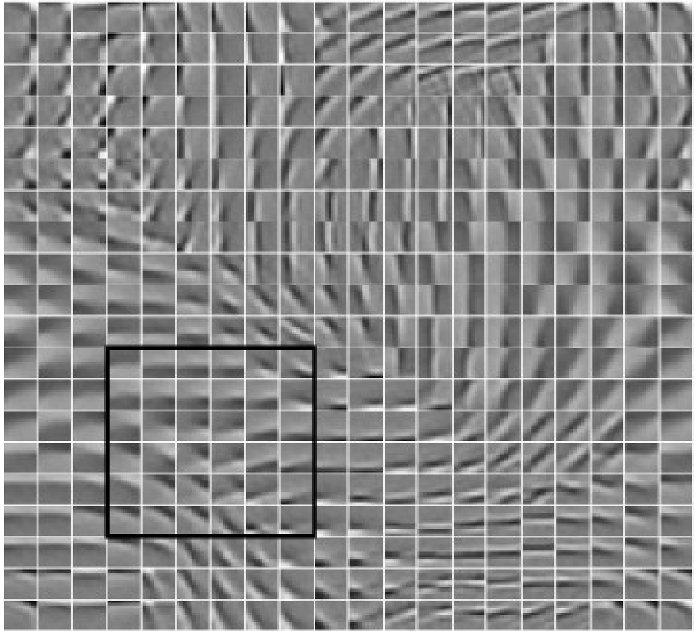
Learning invariant features

- **Sparsity over pooling units → group sparsity**
  - [Hyvarinen & Hoyer 2001], [Osindero et al. Neural Comp. 2006], [Kavukcuoglu et al. CVPR 2009], [Gregor & LeCun arXiv:1006.044], [Mairal et al. 2011].

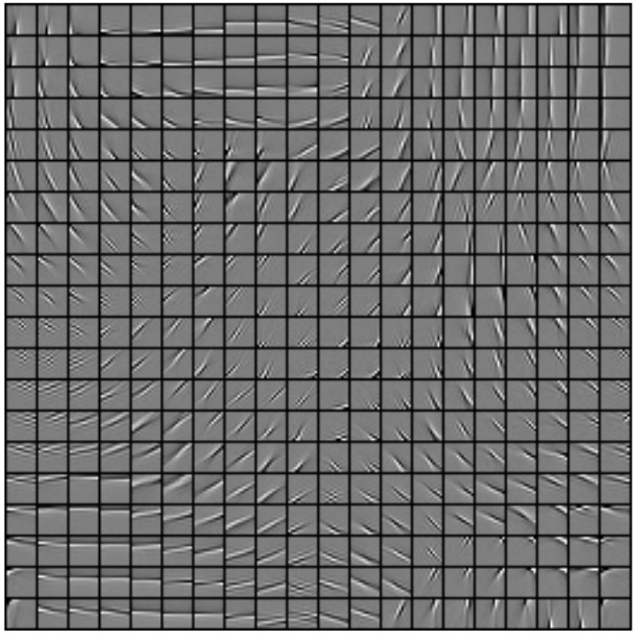
[Osindero 2006]



[Kavukcuoglu 2009]

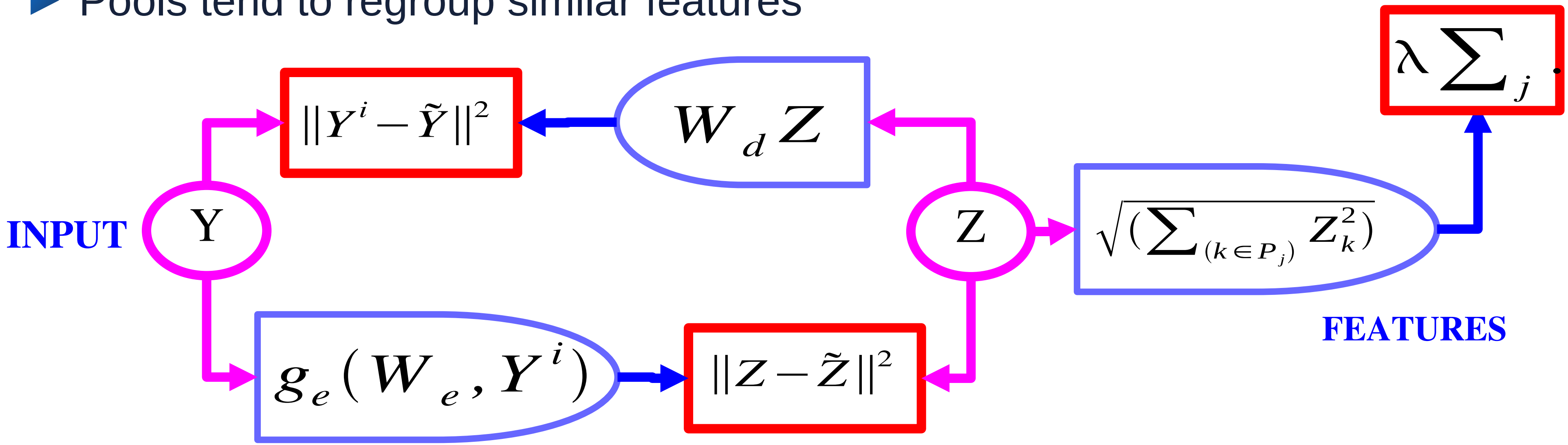


[Mairal 2011]



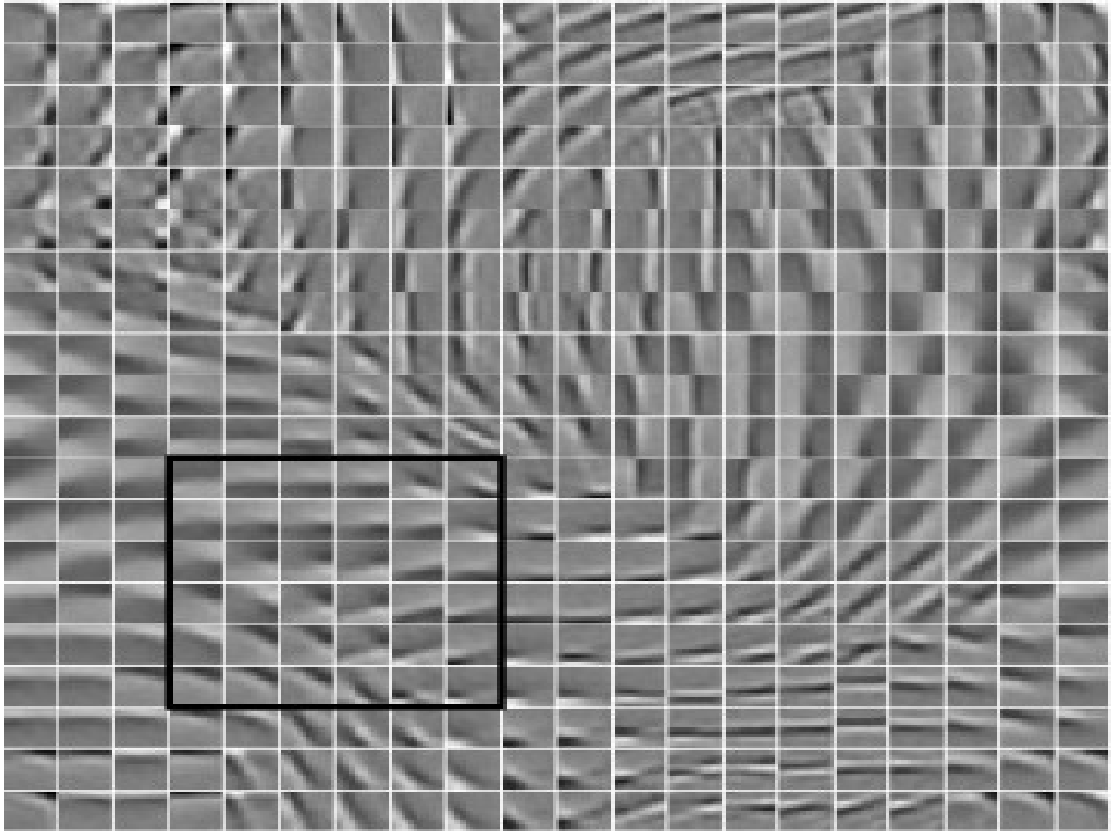
AE with Group Sparsity [Kavukcuoglu et al. CVPR 2009]

- **Trains a sparse AE to produce invariant features**
- **Could we devise a similar method that learns the pooling layer as well?**
- **Group sparsity on pools of features**
  - Minimum number of pools must be non-zero
  - Number of features that are on within a pool doesn't matter
  - Pools tend to regroup similar features



Pooling over features in a topographic map.

- **The pools are 6x6 blocks of features arranged in a 2D torus**
- **While training, the filters arrange themselves spontaneously so that similar filters enter the same pool.**
- **The pooling units can be seen as complex cells**
- **They are invariant to local transformations of the input**
  - For some it's translations, for others rotations, or other transformations.



Pinwheels!

- **The so-called "pinwheel" pattern is observed in the primary visual cortex.**

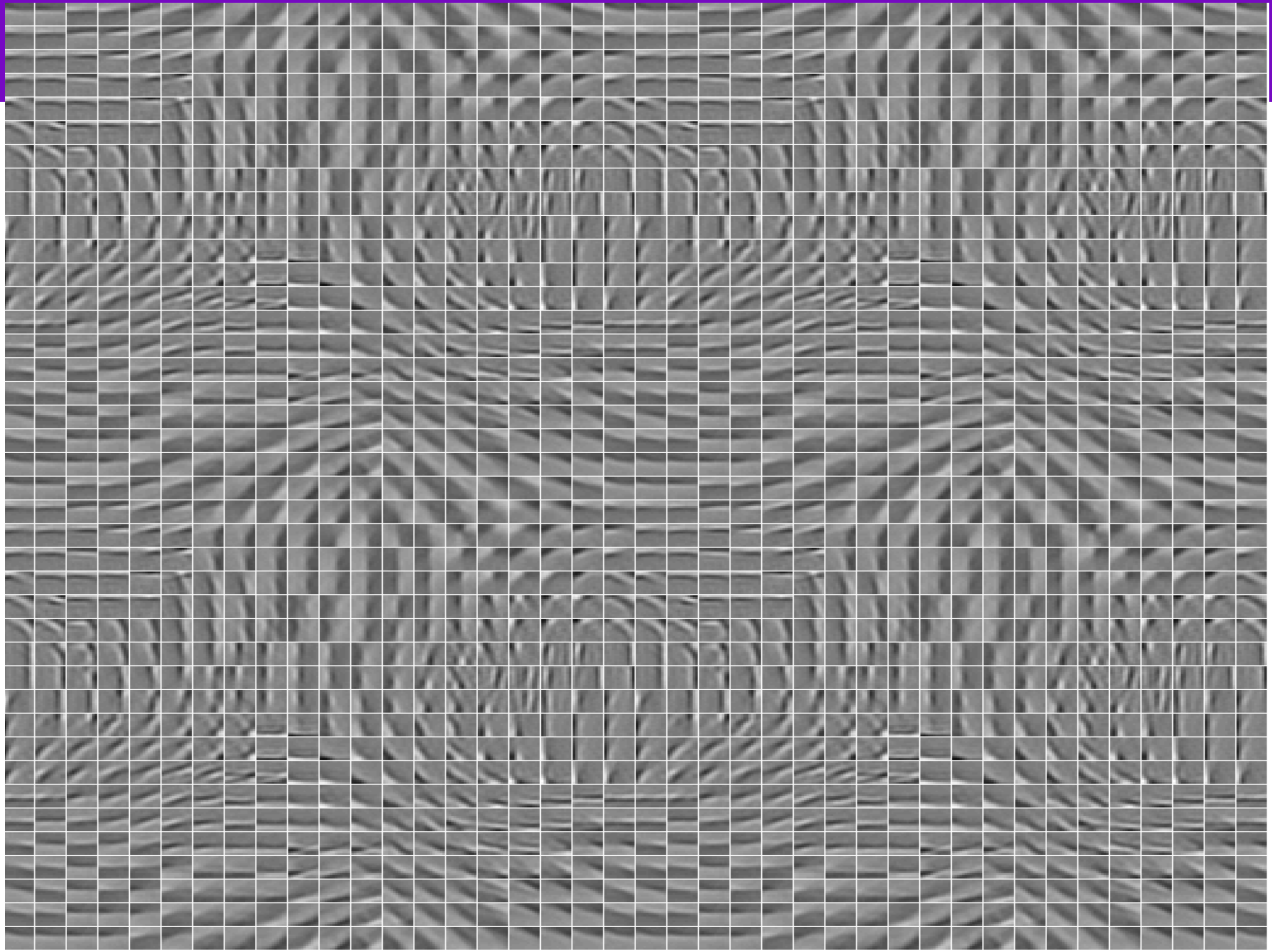


Image-level training, local filters but no weight sharing!

- **Training on 115x115 images. Kernels are 15x15 (not shared across space!)**
- **[Gregor & LeCun arXiv:1006.044]**
  - "Emergence of Complex-Like Cells in a Temporal Product Network with Local Receptive Fields"

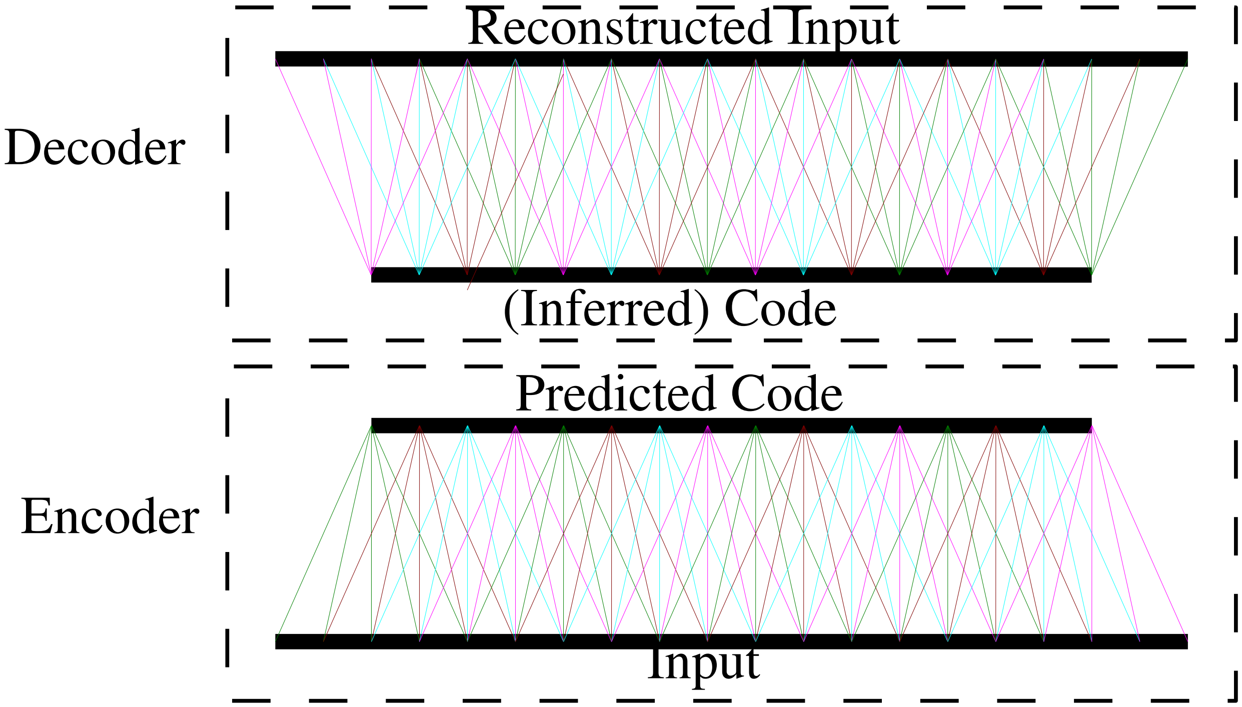


Image-level training, local filters but no weight sharing!

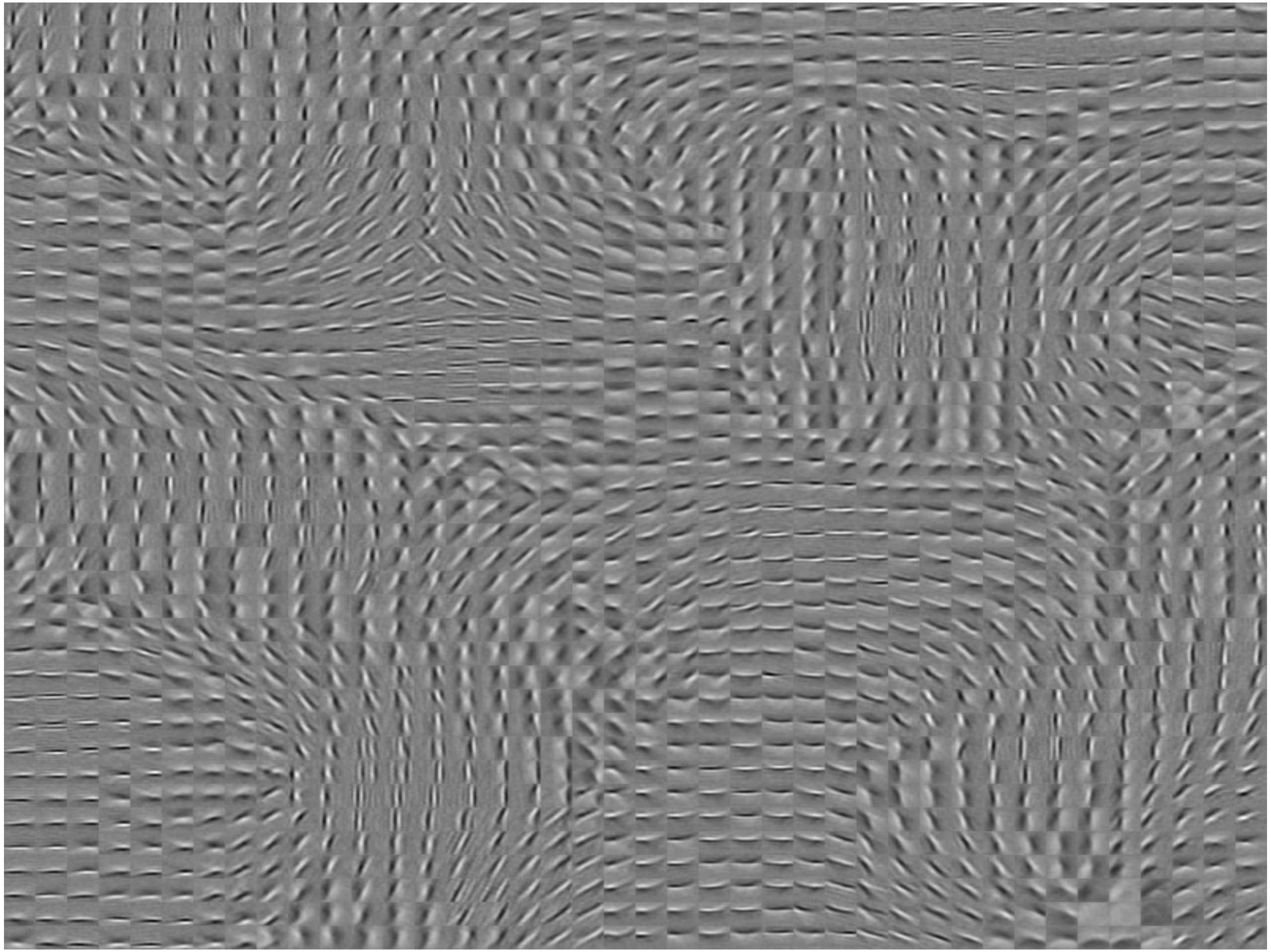
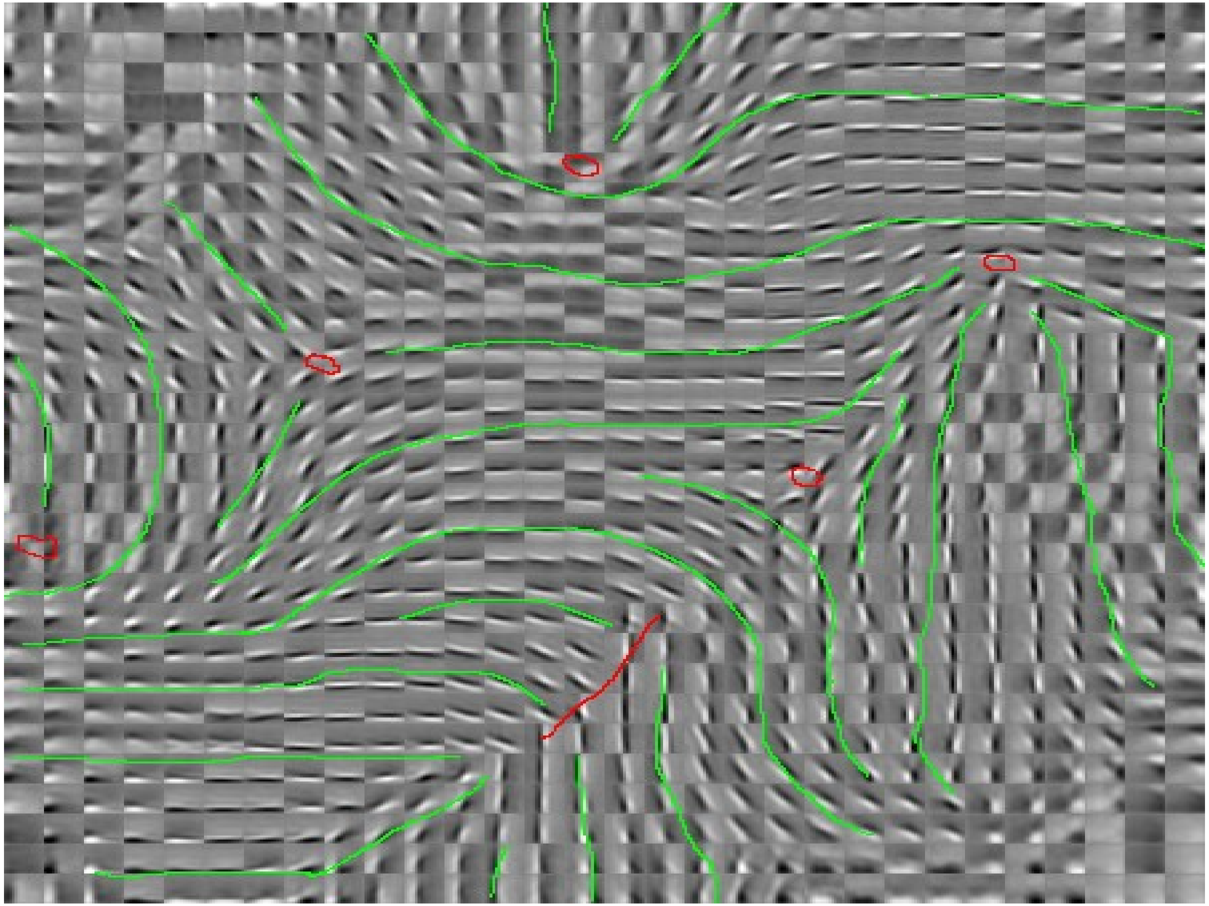


Image-level training, local filters but no weight sharing!

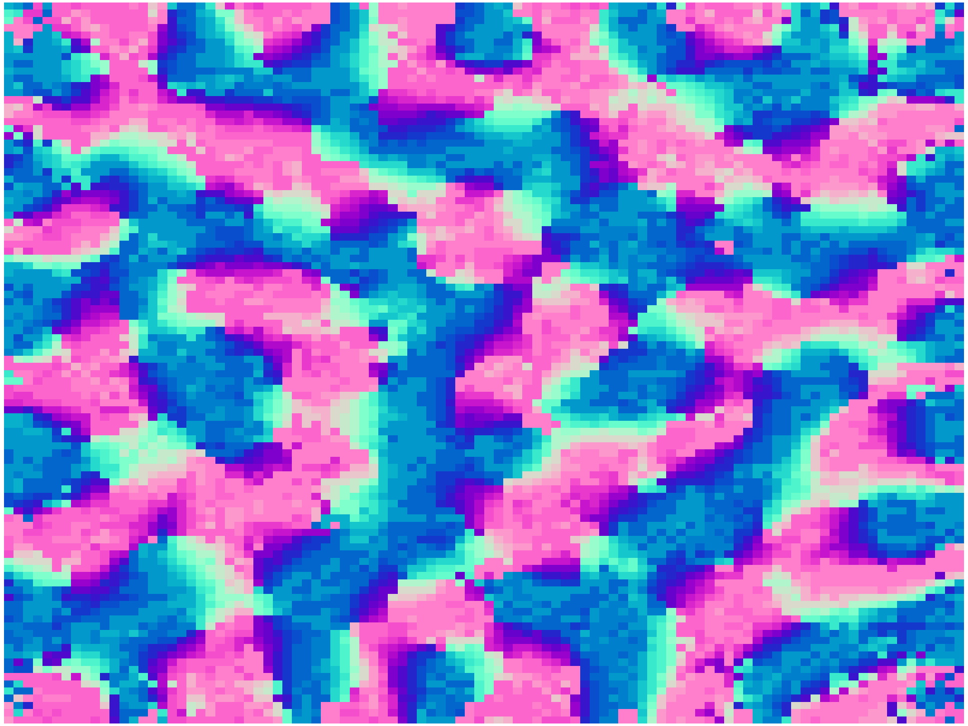
- **Training on 115x115 images. Kernels are 15x15 (not shared across space!)**



Orientation Selectivity Map

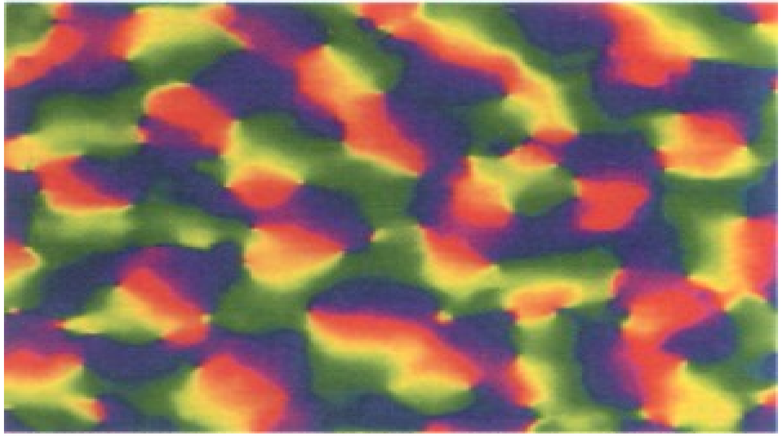
119x119 Image Input, 100x100 Code

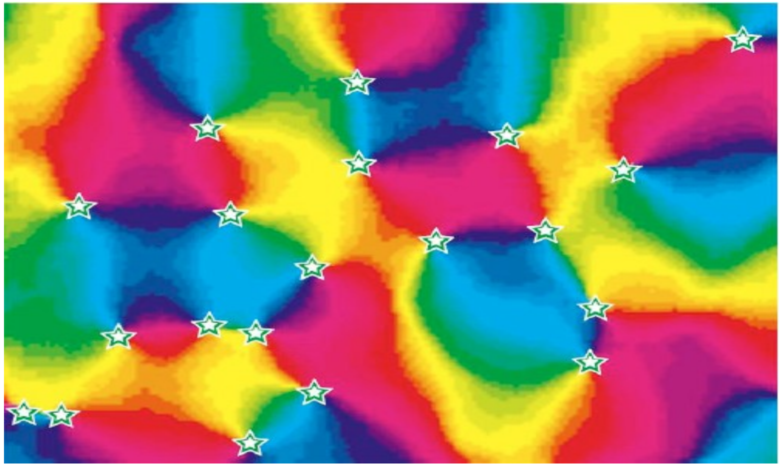
20x20 Receptive field size, sigma=5



Michael C. Crair, et. al. The Journal of Neurophysiology Vol. 77 No. 6 June 1997, pp. 3381-3385 (**Cat**)

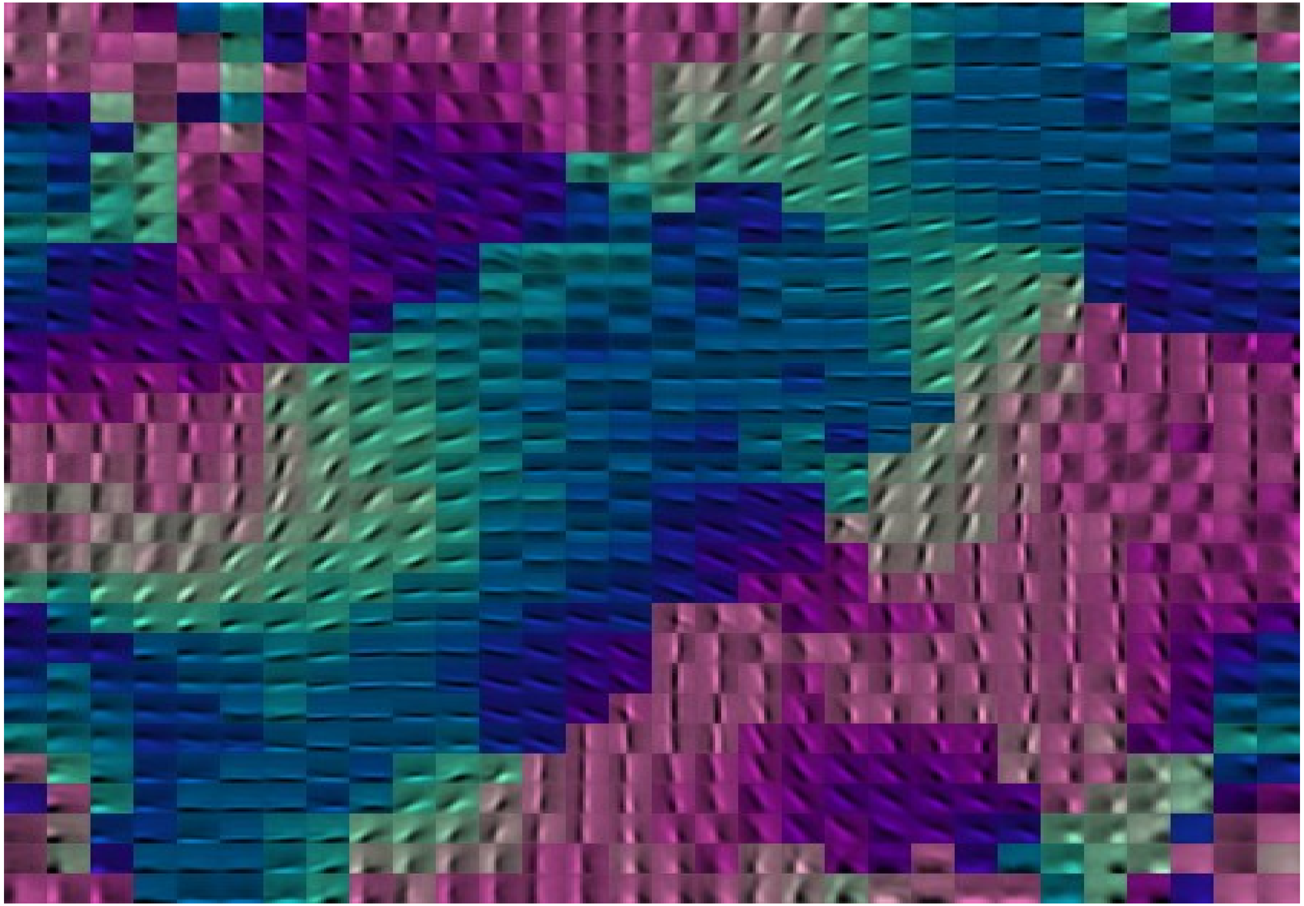
K Obermayer and GG Blasdel, Journal of Neuroscience, Vol 13, 4114-4129 (**Monkey**)





Same Method, withTraining at the Image Level (vs patch)

- Color indicates orientation (by fitting Gabors)



FACEBOOK AI

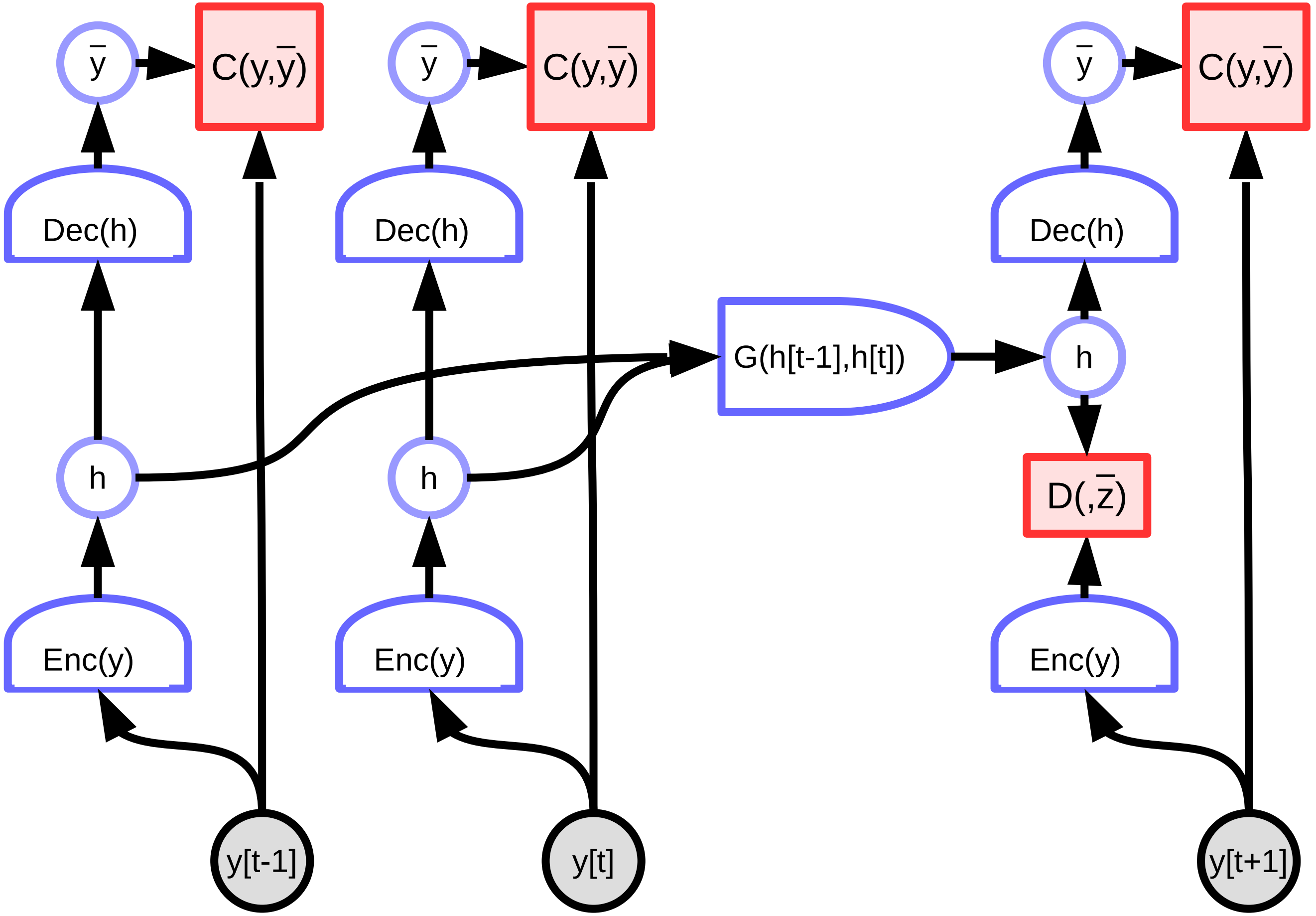
Regularization through Temporal Consistency

Learning to predict invariant features from video

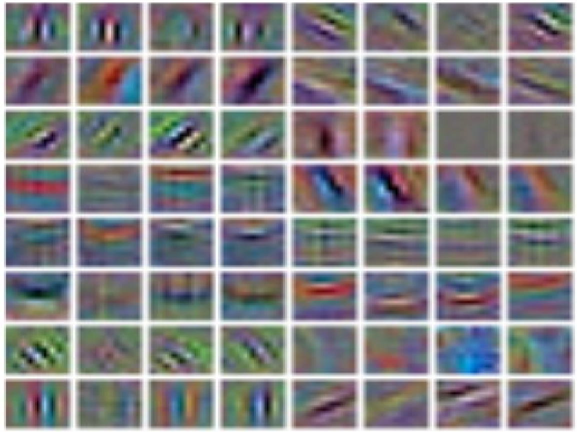
Temporal Regularization Methods

- **Favors "flatness" and predictability of the representation.**
  - Temporal invariance [Goroshin ICCV'15]
  - Linear predictability [Goroshin NIPS'16]
  - Minimal curvature

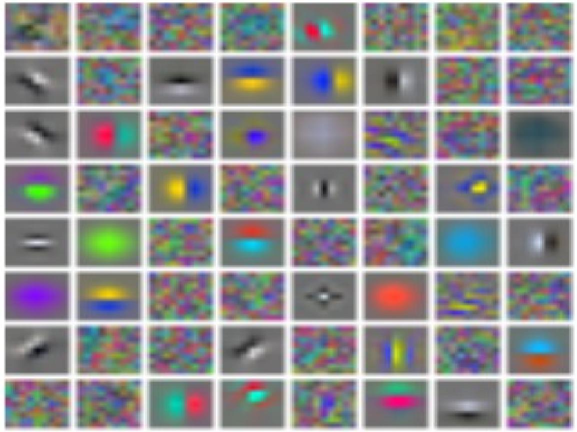
    [O. Hénaff 2019]
- **Temporal proximity is an instance of similarity graph.**
- **Decoder alleviates need for contrastive samples**



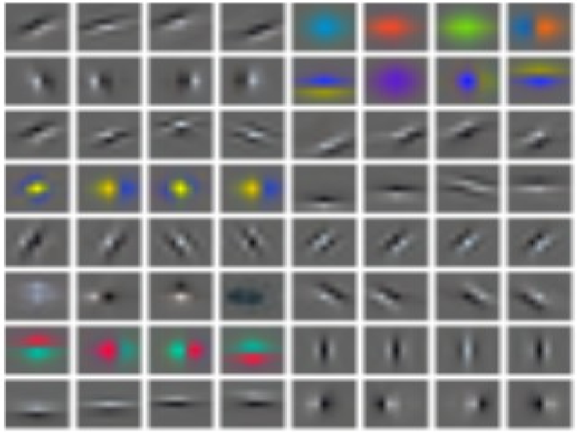
Sparse Auto-Encoder with “Slow Feature” Penalty



- Supervised filters CIFAR10

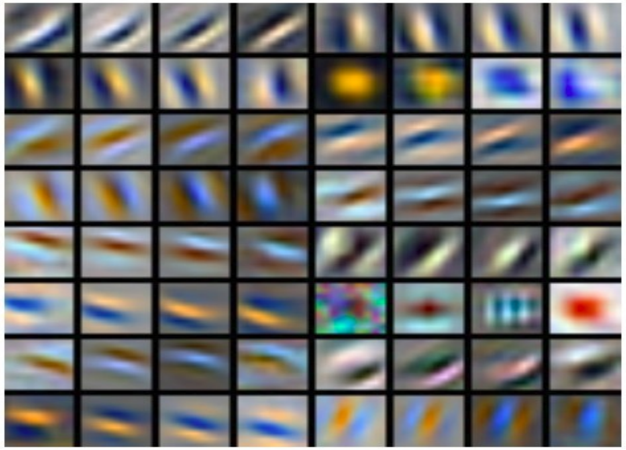


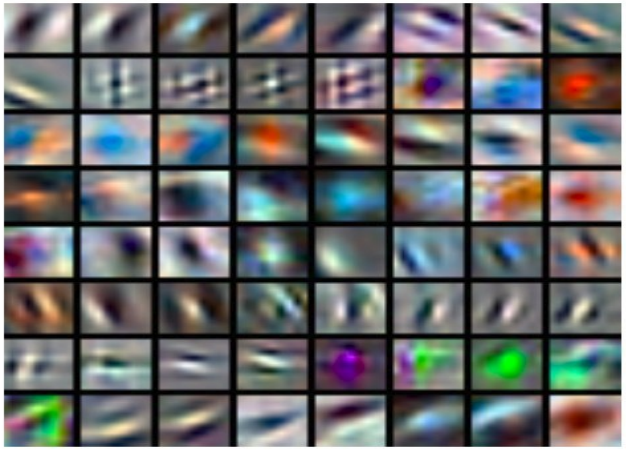
sparse conv. auto-encoder



slow & sparse convolutional AE trained on YouTube videos

- Representation is pooled over non-overlapping groups of 4 features





- Representation is pooled over overlapping groups of 4 features

FACEBOOK AI

Variational AE

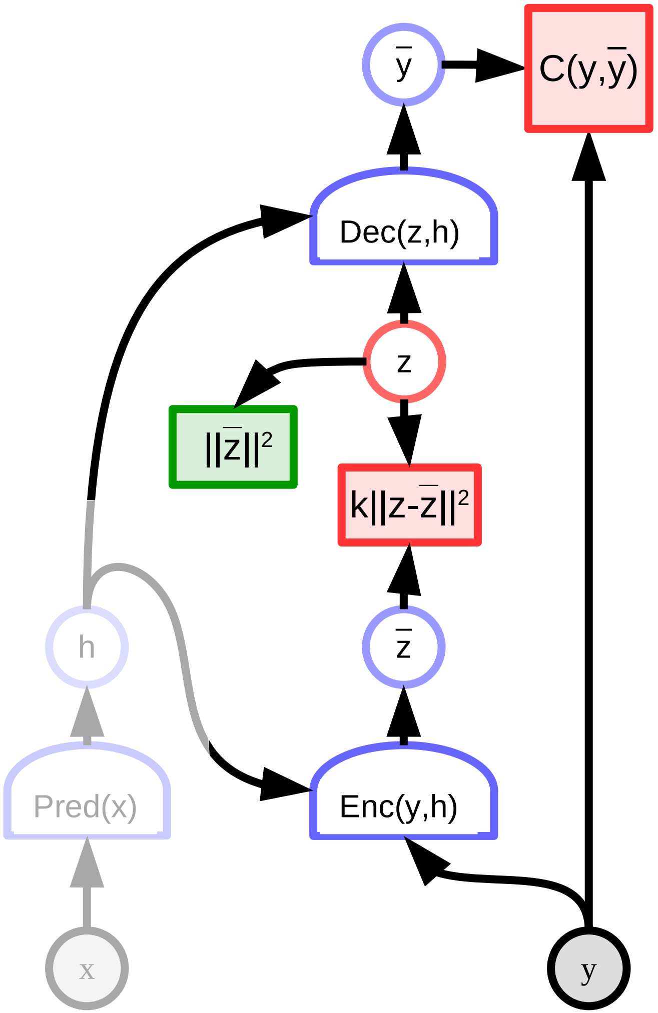
The energy-based approach.
It’s a kind of noise-regularized
latent var model with amortized inference.

Variational Auto-Encoder

- **Limiting the information capacity of the code by adding Gaussian noise**
- **The energy term $k||z-\bar{z}||^2$ is seen as the log of a prior from which to sample z**
- **The encoder output is regularized to have a mean and a variance close to zero.**

$$E(y,z) = C(y, Dec(z)) + (z - Enc(y))^T M (z - Enc(y)) + \gamma ||z||^2$$

Inverse covariance matrix



Variational Auto-Encoder

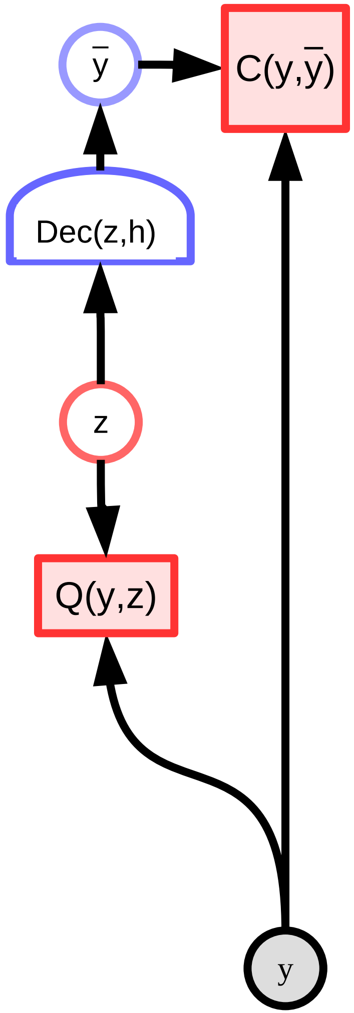
- **Variational approximation of marginalization over z**

$$E(y,z) = C(y, Dec(z))$$

$$F(y) = -\frac{1}{\beta}\log\int_z e^{-\beta E(y,z)}$$

$$F(y) = -\frac{1}{\beta}\log\int_z q(z|y)\left[\frac{e^{-\beta E(y,z)}}{q(z|y)}\right]$$

$$q(z|y) = \frac{e^{-\beta Q(y,z)}}{\int_{z'} e^{-\beta Q(y,z')}}$$



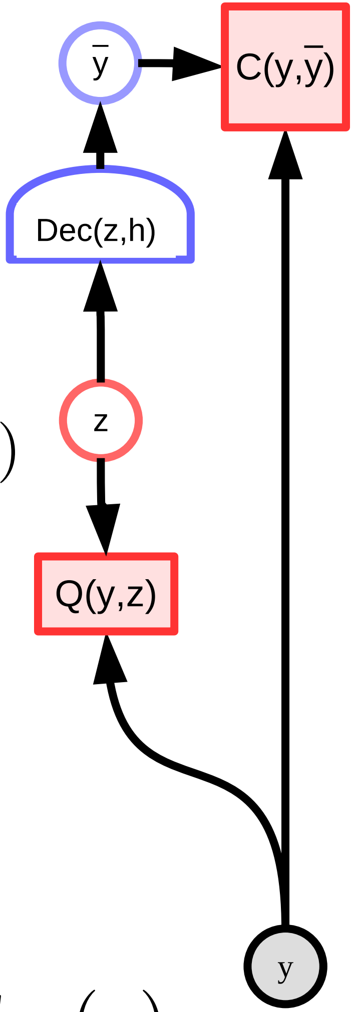
Variational Auto-Encoder

- **Variational approximation of marginalization over z**

$$F(y) = -\frac{1}{\beta}\log\int_z q(z|y)\frac{e^{-\beta E(y,z)}}{q(z|y)}$$

Jensen's inequality:
-log is convex, hence

$$-\log(\mathrm{mean}_z(h(z))) \leq \mathrm{mean}_z(-\log(h(z)))$$



$$F(y) \leq \tilde{F}_q(y) = \int_z q(z|y)\left[-\frac{1}{\beta}\log\frac{e^{-\beta E(y,z)}}{q(z|y)}\right]$$

$$\tilde{F}_q(y) = \int_z q(z|y)E(y,z) + \frac{1}{\beta}\int_z q(z|y)\log(q(z|y))$$

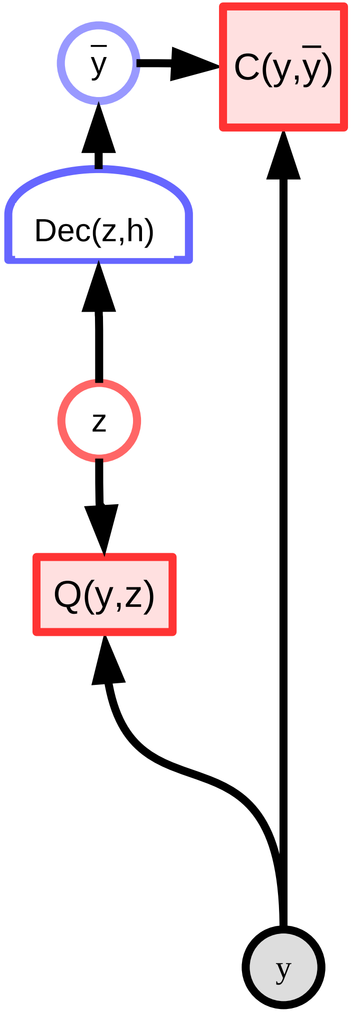
$$\tilde{F} = \langle E\rangle - TS \qquad\qquad L(w) = \tilde{F}_{qw}(y)$$

Variational Auto-Encoder

- **Q: What distribution q minimizes the variational free energy?**
- **A: the (intractable) "posterior" distribution over z:**

$$\check{q}(z|y) = \frac{e^{-\beta E(y,z)}}{\int_{z'} e^{-\beta E(y,z')}} \qquad F(y) = \tilde{F}_{\check{q}}(y)$$

$$F(y) = \int_z \check{q}(z|y) E(y,z) + \frac{1}{\beta} \int_z \check{q}(z|y) \log(\check{q}(z|y))$$



- **The variational approximation trick replaces the intractable posterior by a family of tractable distributions.**

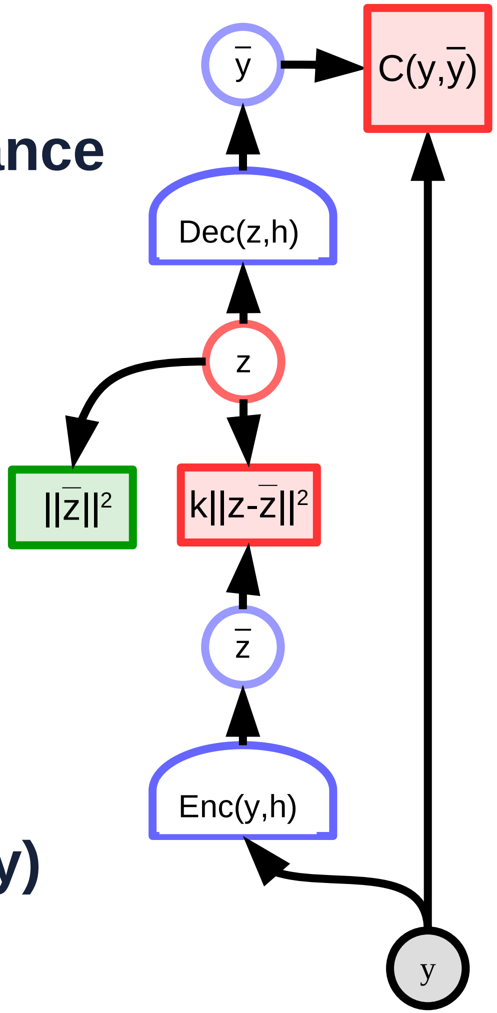
Variational Auto-Encoder

- **Simple case: q(z|y) is Gaussian with fixed covariance**
  - **Entropy is constant**

$$q(z|y) = \frac{e^{-Q(y,z)}}{\int_z' e^{-Q(y,z')}}$$

$$\mathrm{Q(y,z)} = ||z - Enc(y)||^2 + \gamma\, ||z||^2$$

- **Mean of the Gaussian is Enc(y)**
  - **Denominator independent of parameters of Enc(y)**



Variational Auto-Encoder

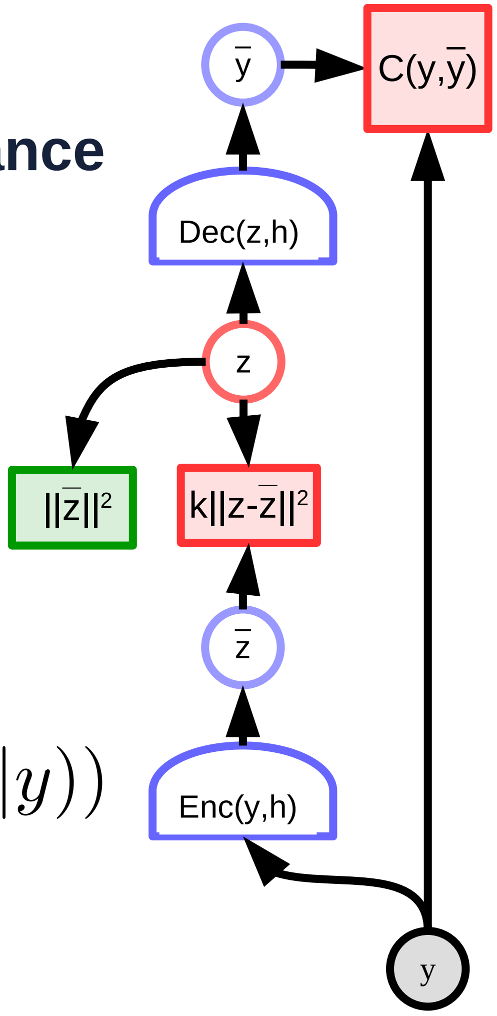
- **Simple case: q(z|y) is Gaussian with fixed covariance**
  - **Entropy is constant**

$$q(z|y) = \frac{e^{-Q(y,z)}}{\int_{z'} e^{-Q(y,z')}}$$

$$\mathrm{Q(y,z)} = ||z - Enc(y)||^2 + \gamma\, ||z||^2$$

$$\tilde{F}(y) = \int_z q(z|y)E(y,z) + \frac{1}{\beta}\int_z q(z|y)\log(q(z|y))$$

$$\tilde{F}(y) = \int_z q(z|y)E(y,z) + \text{Constant}$$



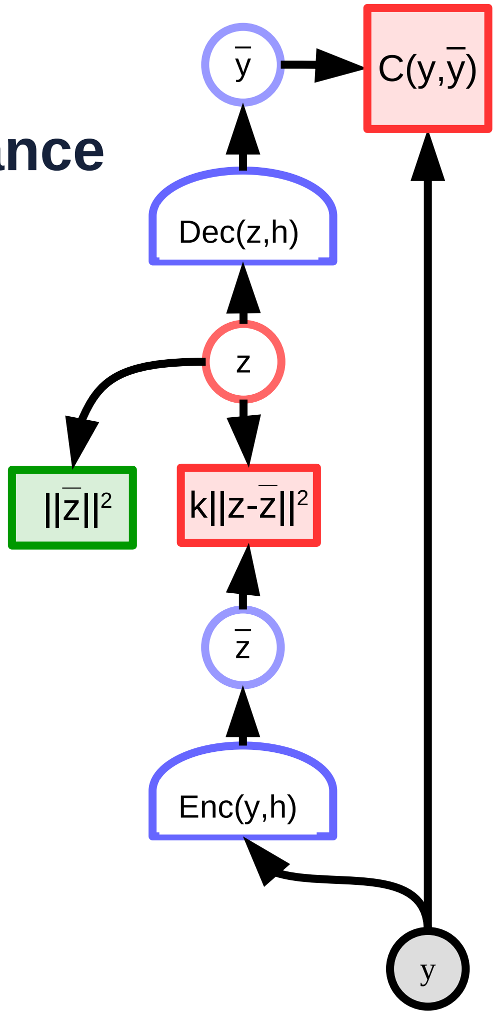
Variational Auto-Encoder

- **Simple case: q(z|y) is Gaussian with fixed covariance**
  - **Entropy is constant**

$$q(z|y) = \frac{e^{-Q(y,z)}}{\int_z' e^{-Q(y,z')}}$$

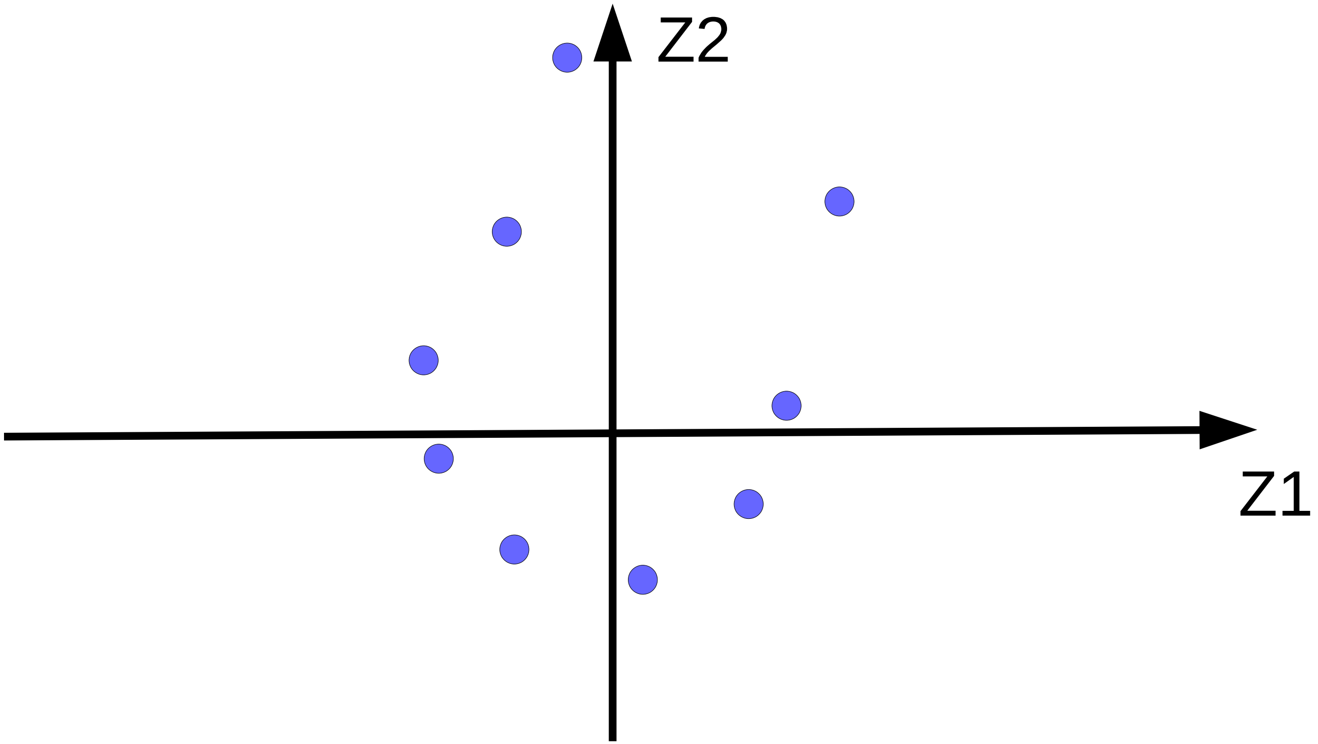
$$\mathrm{Q(y,z)} = ||z - Enc(y)||^2 + \gamma\, ||z||^2$$

$$\frac{\partial \tilde{F}(y)}{w_e} \simeq \frac{\partial E(y, \mathrm{Enc}(y))}{\partial w_e}$$



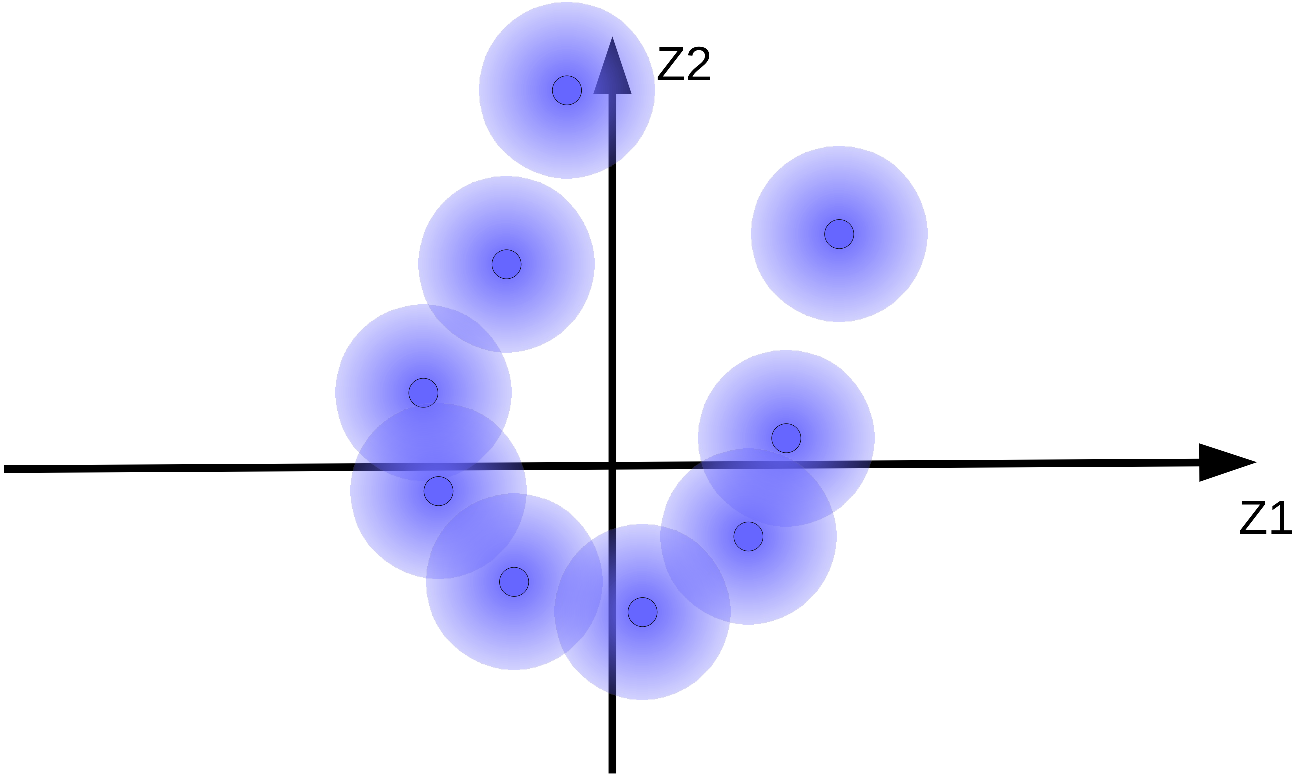
Variational Auto-Encoder

- **Code vectors for training samples**



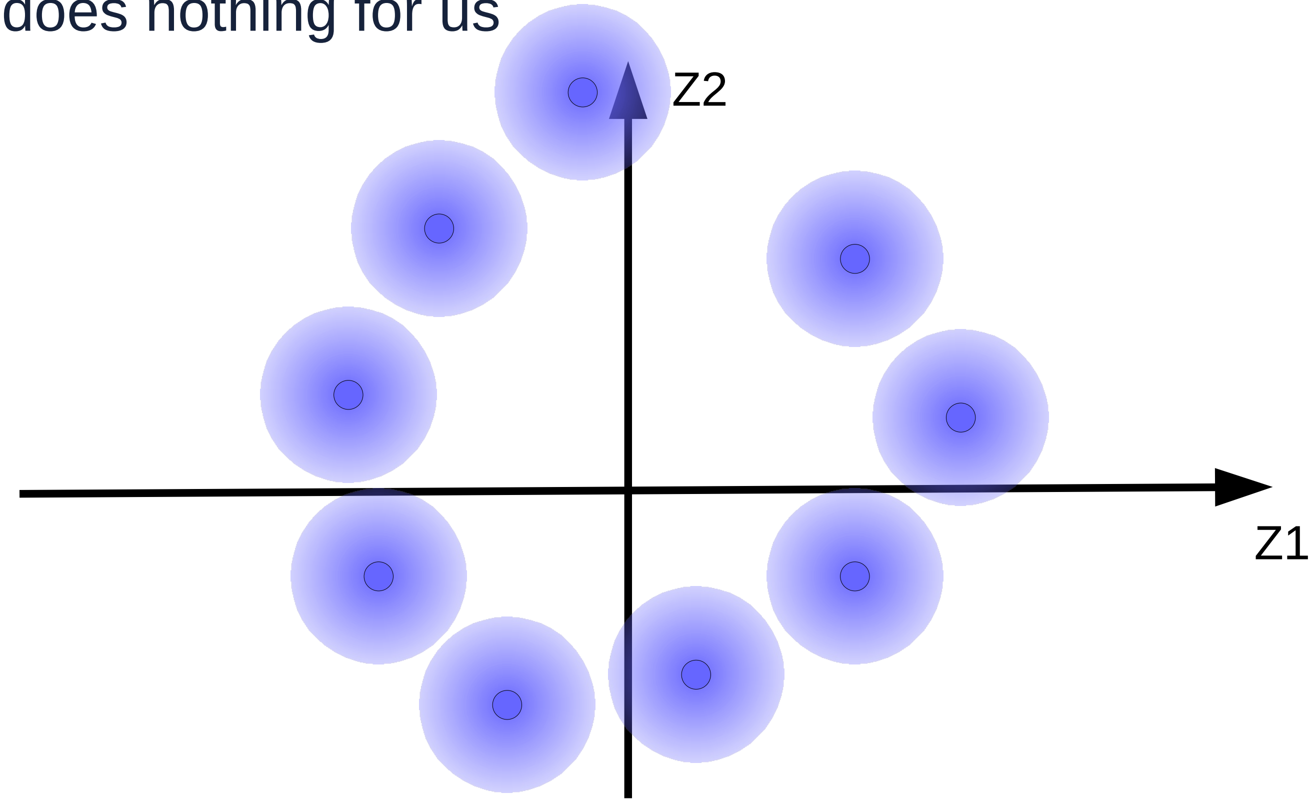
Variational Auto-Encoder

- **Code vectors for training sample with Gaussian noise**
  - Some fuzzy balls overlap, causing bad reconstructions



Variational Auto-Encoder

- **The code vectors want to move away from each other to minimize reconstruction error**
  - But that does nothing for us



Variational Auto-Encoder

- **Attach the balls to the center with a spring, so they don’t fly away**
  - Minimize the square distances of the balls to the origin
- **Center the balls around the origin**
  - Make the center of mass zero
- **Make the sizes of the balls close to 1 in each dimension**
  - Through a so-called KL term