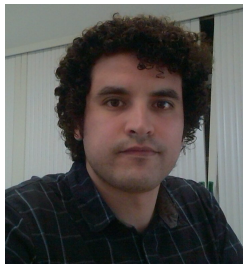
Representation Learning

UCA Deep Learning School - Deep in France

Nice 2017



Soufiane Belharbi

Romain Hérault

Clément Chatelain

Sébastien Adam

soufiane.belharbi@insa-rouen.fr

LITIS lab., Apprentissage team - INSA de Rouen, France

13 June, 2017

My PhD work

3rd year PhD student at LITIS lab. Deep learning, structured output prediction, learning representations.

1. S. Belharbi, C. Chatelain, R.Hérault, S. Adam, ***Learning Structured Output Dependencies Using Deep Neural Networks***. 2015. in: Deep Learning Workshop in the 32nd International Conference on Machine Learning (ICML), 2015.
2. S. Belharbi, R.Hérault, C. Chatelain, S. Adam, ***Deep multi-task learning with evolving weights***, in: European Symposium on Artificial Neural 445 Networks (ESANN), 2016
3. S. Belharbi, C. Chatelain, R.Hérault, S. Adam, ***Multi-task Learning for Structured Output Prediction***. 2017. submitted to Neurocomputing. ArXiv: arxiv.org/abs/1504.07550.
4. S. Belharbi, R.Hérault, C. Chatelain, R. Modzelewski, S. Adam, M. Chastan, S. Thureau, ***Spotting L3 slice in CT scans using deep convolutional network and transfer learning***, in: Computers in Biology and Medicine, 2017.

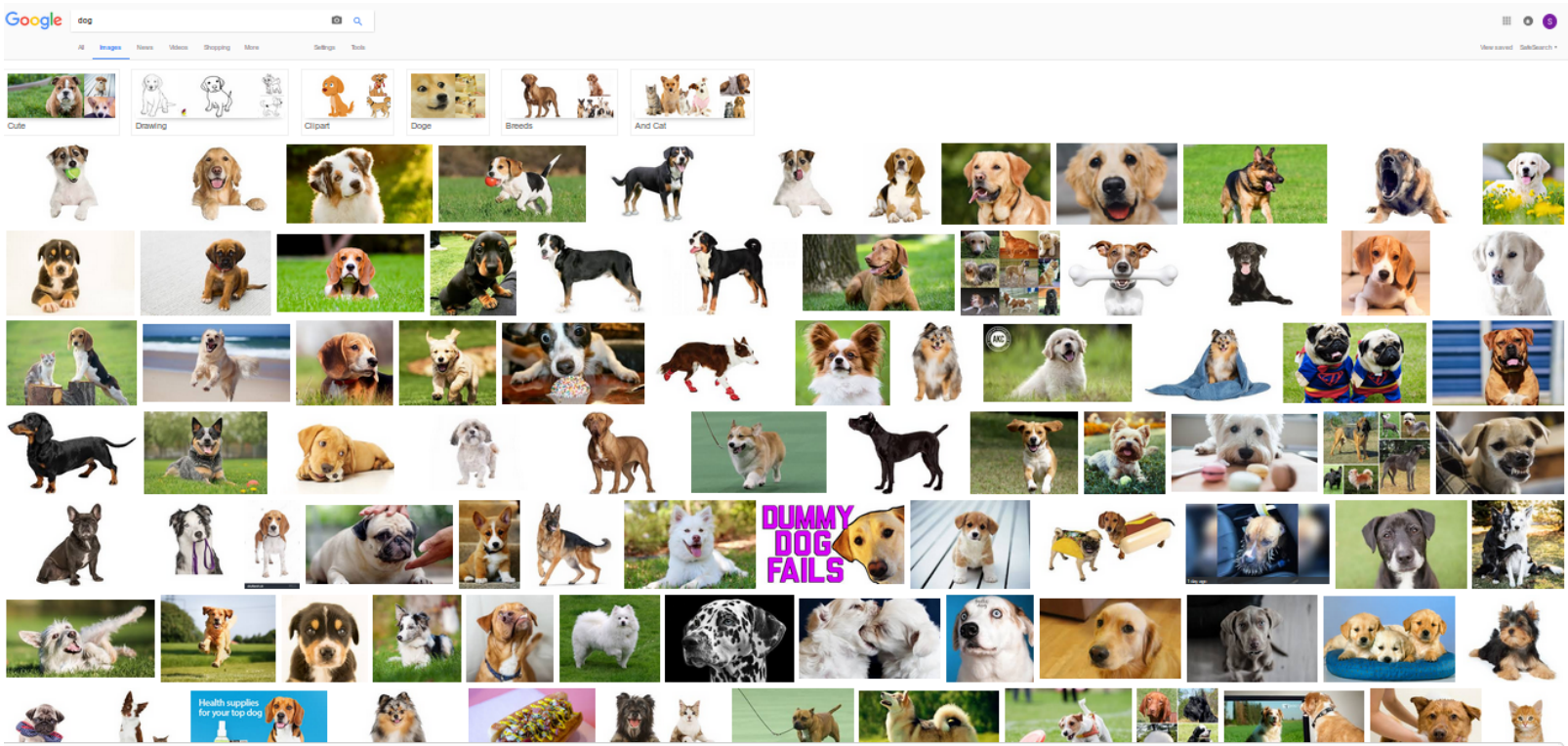
Current work: Learning **invariance** within neural networks.
S. Belharbi, C. Chatelain, R.Hérault, S. Adam, ***Class-invariance hint: a regularization framework for training neural networks***. Coming up soon.

Roadmap

Representation Learning

Representation Learning is fundamental in Machine Learning

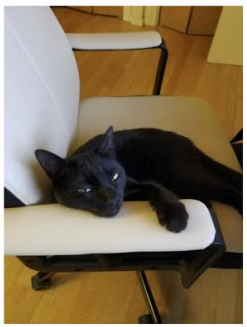
How to represent the class “dog”? (input variations)



Conference: ICLR `www.iclr.cc` (since 2013).

Representation Learning

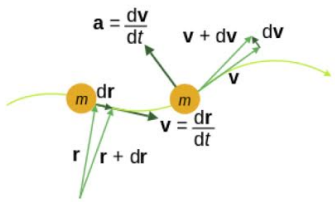
Things...

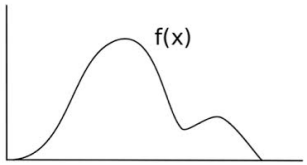


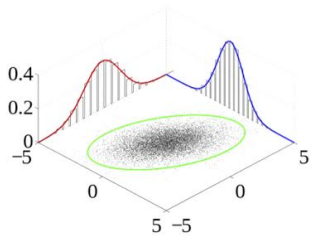
My heart beats as if the world is dropping, you may not feel the love but i do its a heart breaking moment of your life. enjoy the times that we have, it might not sound good but one thing it rhymes it might not be romantic but i think it is great,the best rhyme i've ever heard.

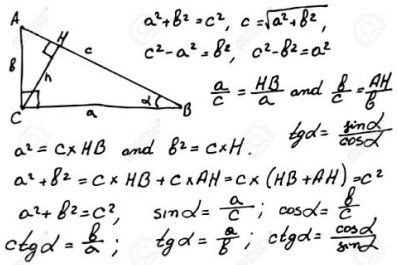
Representation

Engineering Knowledge...





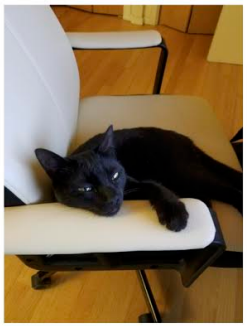




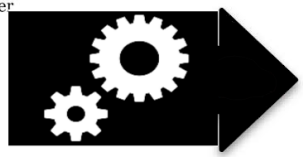
Stanford, CS331B.

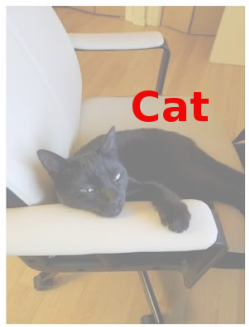
Representation Learning

Things...



My heart beats as if the world is dropping, you may not feel the love but i do its a heart breaking moment of your life. enjoy the times that we have, it might not sound good but one thing it rhymes it might not be romantic but i think it is great,the best rhyme i've ever heard.





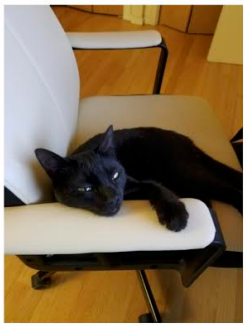
My heart beats as if the world is dropping, you may not feel the love but i do its a heart breaking moment of your life. enjoy the times that we have, it might not sound good but one thing it rhymes it might not be romantic but i think it is great,the best rhyme i've ever heard.

Where was the cat in the morning?

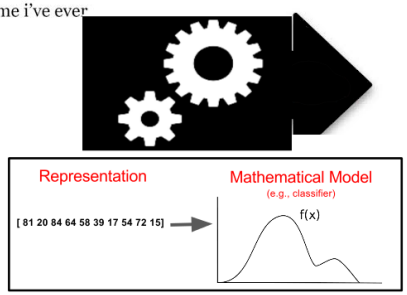
Task

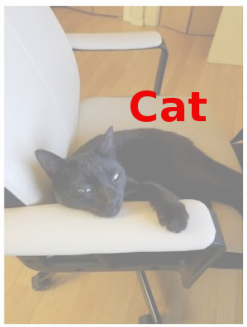
Representation Learning

Things...



My heart beats as if the world is dropping, you may not feel the love but i do its a heart breaking moment of your life. enjoy the times that we have, it might not sound good but one thing it rhymes it might not be romantic but i think it is great,the best rhyme i've ever heard.





My heart beats as if the world is dropping, you may not feel the love but i do its a heart

Where was the cat in the morning?

Task

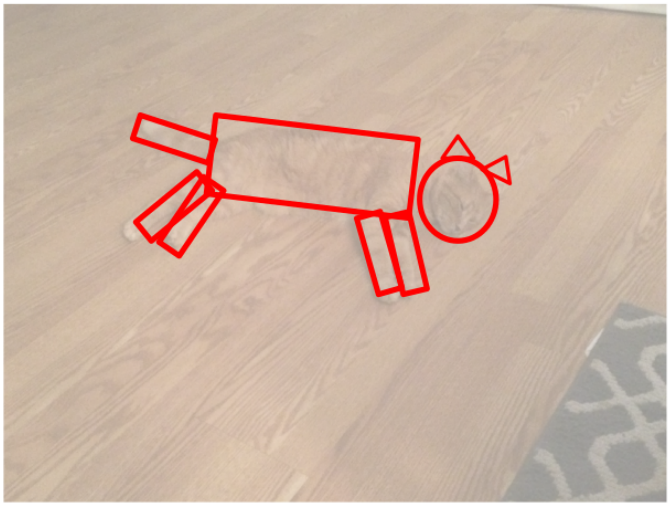
Features representation: Handcrafting

Let us build a cat detector . . .

Stanford, A.Zamir

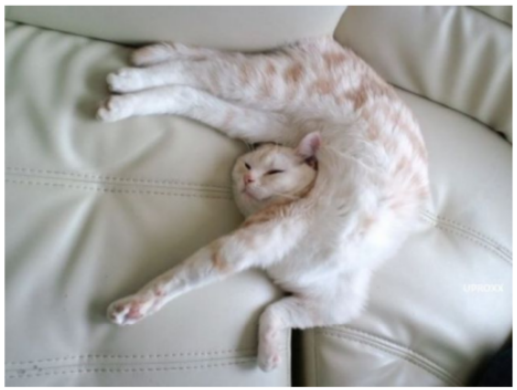
Features representation: Handcrafting

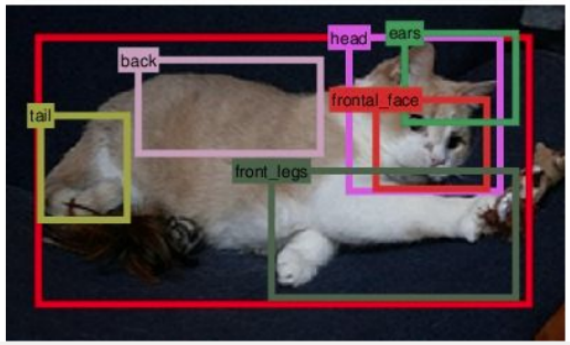
Let us build a cat detector . . .



Features representation: Handcrafting

Let us build a cat detector . . .



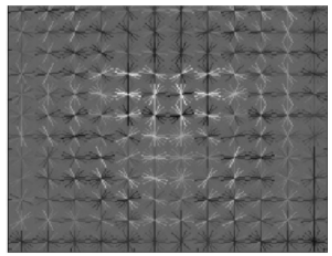


Features representation: Handcrafting

Let us build a cat detector . . .







Features representation: Handcrafting

Handcrafted features . . .
Pros:

- Was the only way for a long time.
- Works quite good.
- Sometimes you need to combine many features.
- Generic.

Features representation: Handcrafting

Handcrafted features . . .
Cons:

- Generic.
- Time consuming.
- What you will do if nothing works?
- In many cases, it is difficult to build discriminative features.

Figure 2: Classifier: Happy vs Sad

Ideal:
Application-dependent features $\Rightarrow$ Deep Learning

Representation Learning Approaches

Two main approaches

Supervised

Stanford, A.Zamir

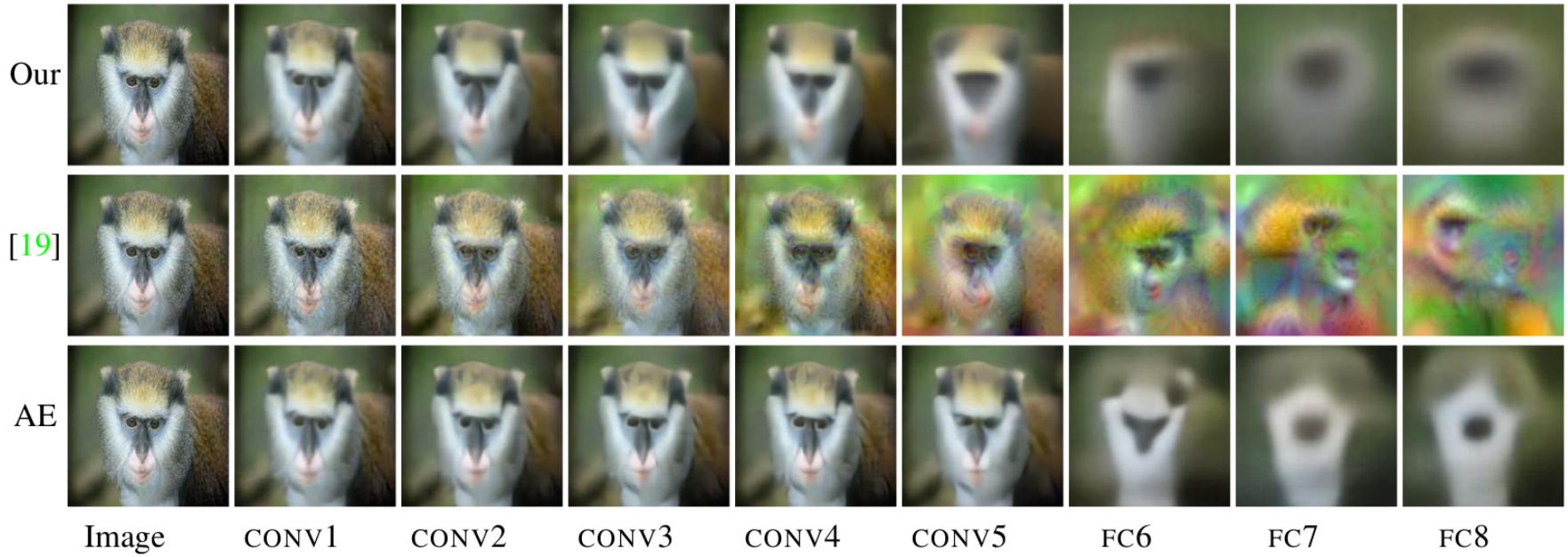
Unsupervised: **Representation constrained on reconstruction**

Representation Learning Approaches

[81 20 84 64 58 39 17 54 72 15]

Inverting a representation

Representation Learning Approaches



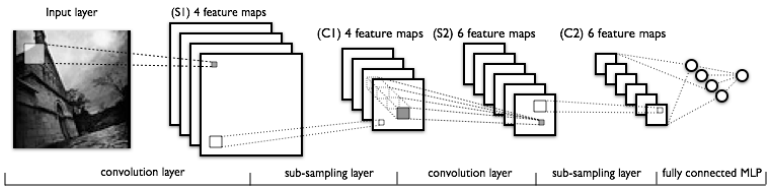
Inverting a representation

Dosovitskiy and Brox, 2015.

Representation Learning Approaches

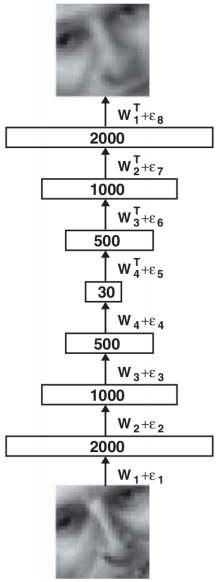
Two main approaches

Supervised



LeCun et al. 1998.

Unsupervised: **Representation constrained on reconstruction**



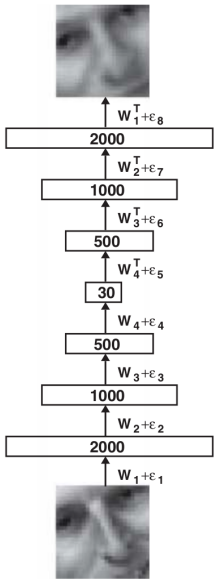
Hinton et al. 2006.

Representation Learning Approaches

Unsupervised: **Representation constrained on reconstruction**

Pros:
Exploit **millions of unlabeled data** from the internet:

- Images.
- Text (Wikipedia, . . .)
- Records and videos.



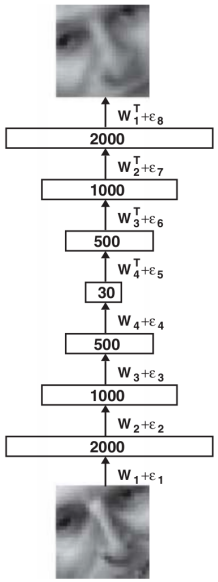
Hinton et al. 2006.

Representation Learning Approaches

Unsupervised: **Representation constrained on reconstruction**

The reconstruction loss: how to reconstruct? L2 pixel loss.
Applications:

- Data compression.
- Dimenstionality reduction.
- Pre-train neuranl networks (initialization).



Hinton et al. 2006.

Unsupervised Representation Learning Methods

- Sparse coding.
- Auto-encoders (AEs).
- Restricted Boltzmann Machines (RBMs).

Sparse Coding

Objective:

$$x = \sum_{i=1}^{k} a_i \phi_i,$$

where ϕ_i is a set of basis (dictionnary).

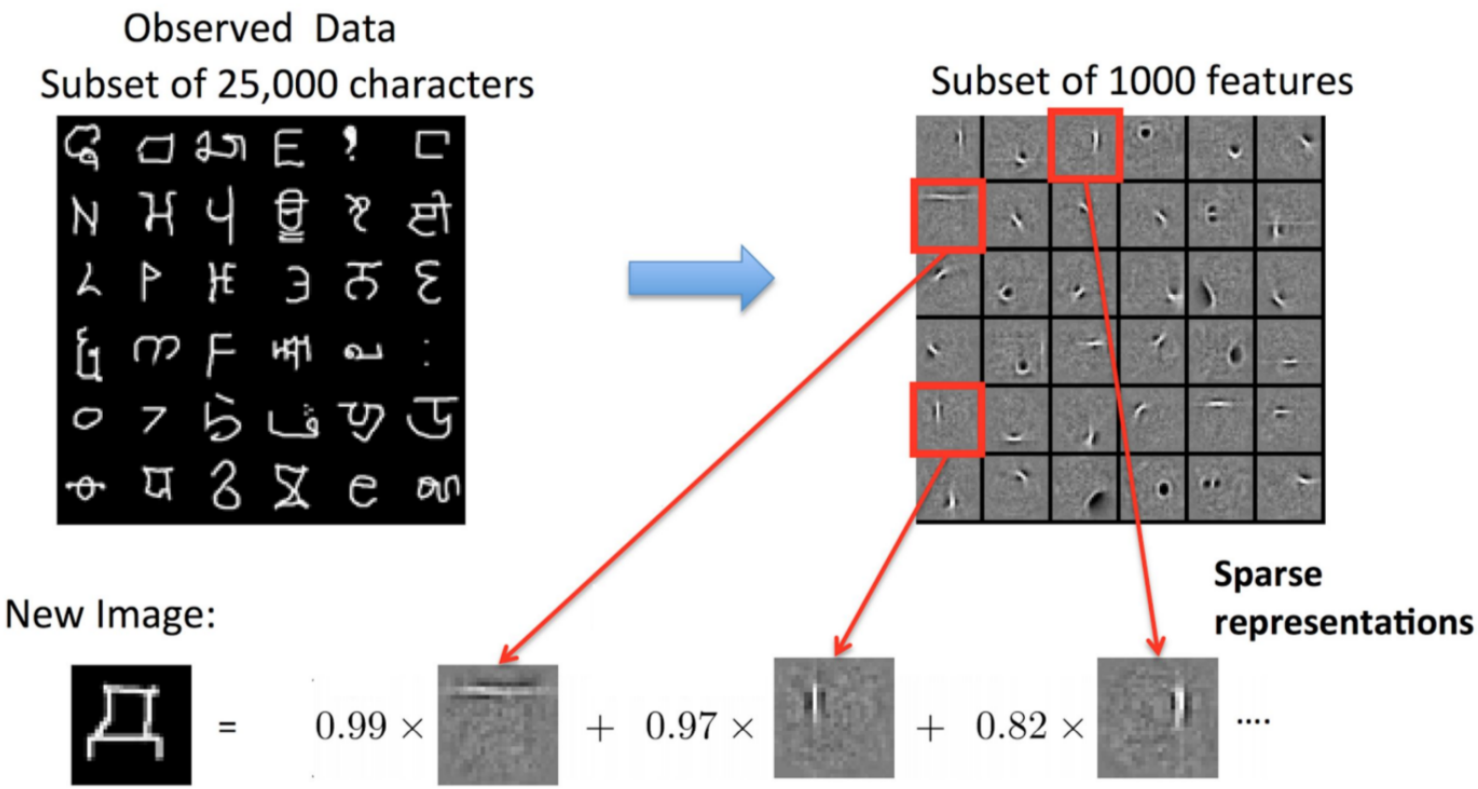
Cost function on a set of m input vectors:

$$\min_{a_i^{(j)}, \phi_i} \underbrace{\sum_{j=1}^{m} \left|\left| x^{(j)} - \sum_{i=1}^{k} a_i^{(j)} \phi_i \right|\right|^2}_{\text{reconstruction term}} + \underbrace{\lambda \sum_{i=1}^{k} S(a_i^{(j)})}_{\text{sparsity term}} .$$

Similar to:

$$\min_{a_i^{(j)}, \phi_i} \underbrace{\sum_{j=1}^{m} \left|\left| x^{(j)} - H^{(j)} W \right|\right|^2}_{\text{reconstruction term}} + \underbrace{\lambda ||H^{(j)}||_1}_{\text{sparsity term}} .$$

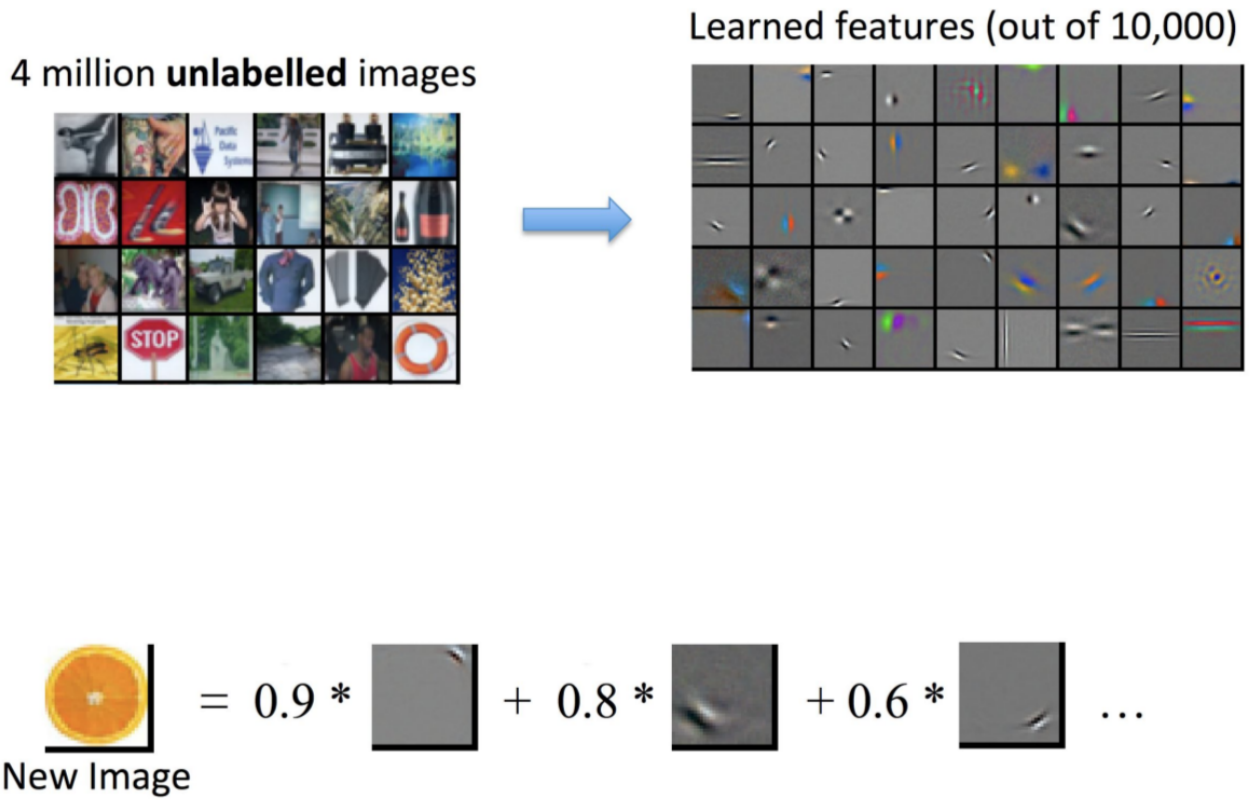
Sparse Coding



Lee et al. 2006.
Salakhutdinov. 2016.

Andrew Ng.

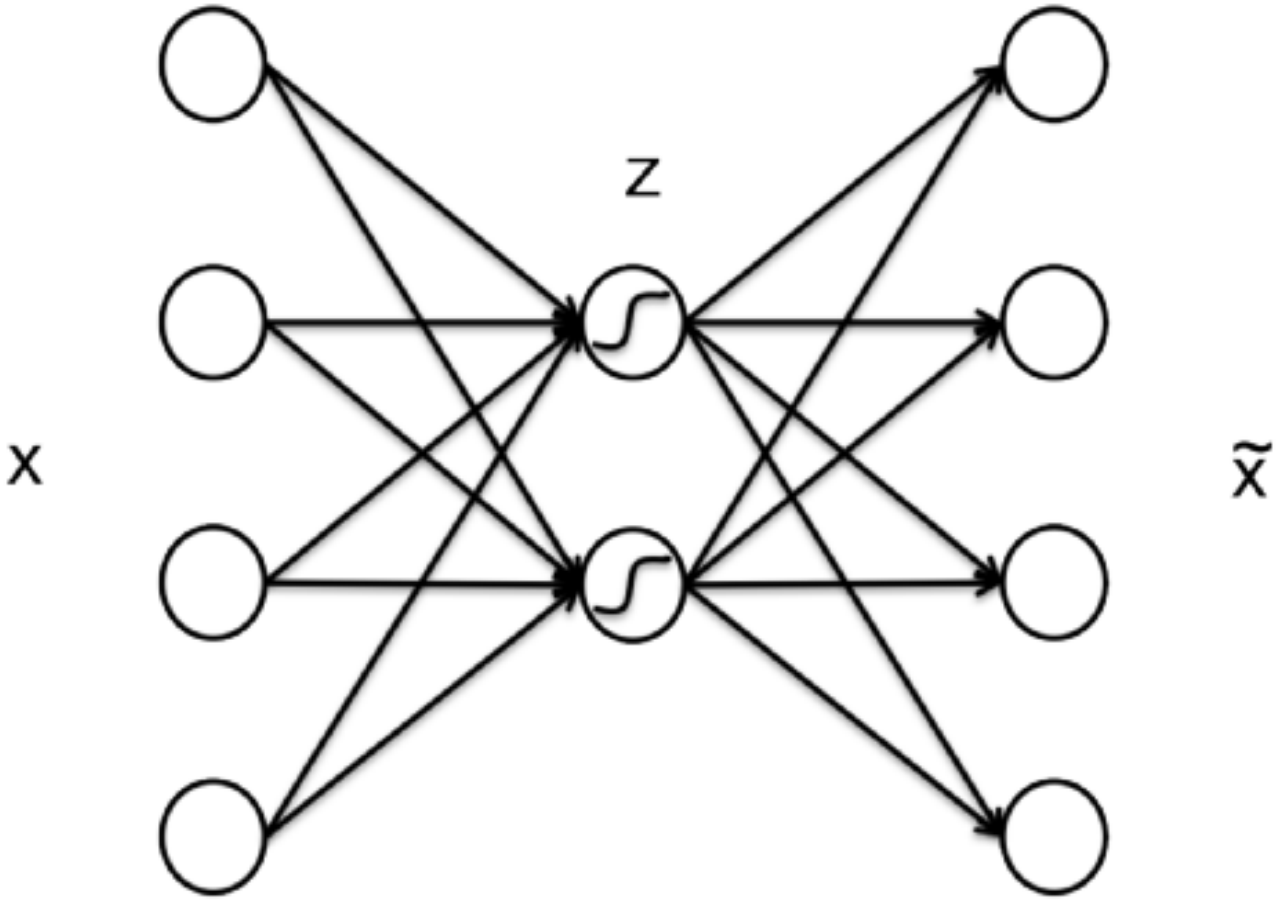
Sparse Coding



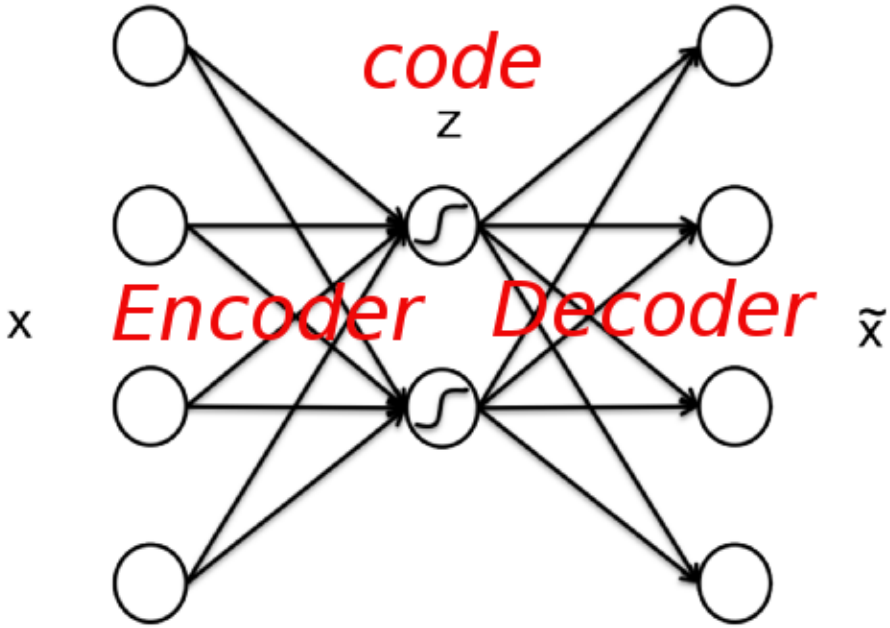
Lee et al. 2006.
Salakhutdinov. 2016.

Andrew Ng.

Auto-encoders



Auto-encoders

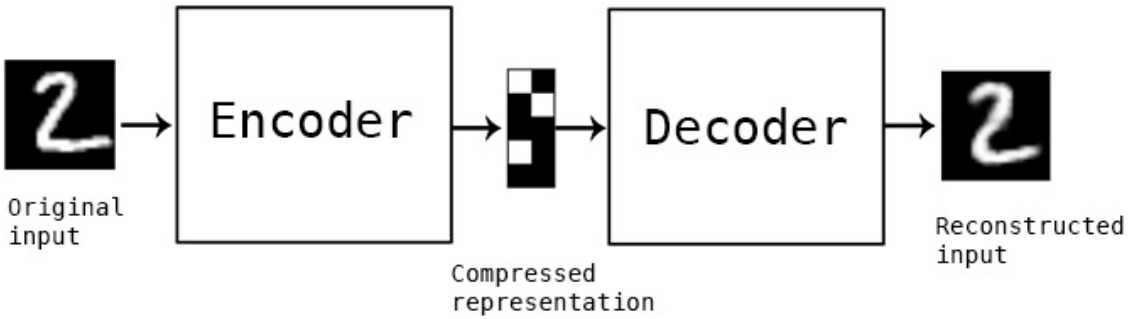


Encoder: $f(x) = s(Wx + b) = z$.
Decoder: $g(z) = s(W' + b') = \widetilde{x}$, $W' = W^T$ (tied weight).
Objective over a set of n examples x:

$$J(x; W, b, b') = \frac{1}{n} \sum_{i=1}^{n} ||x - \widetilde{x}||^2.$$

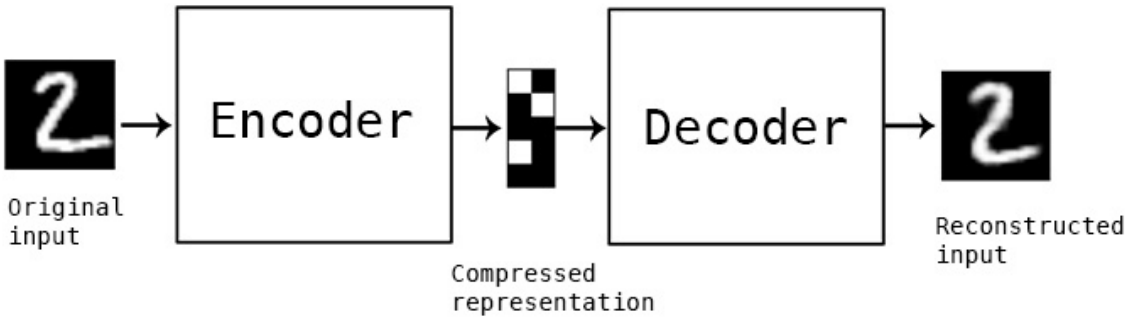
Similar to PCA.

Auto-encoders



Keras blog.

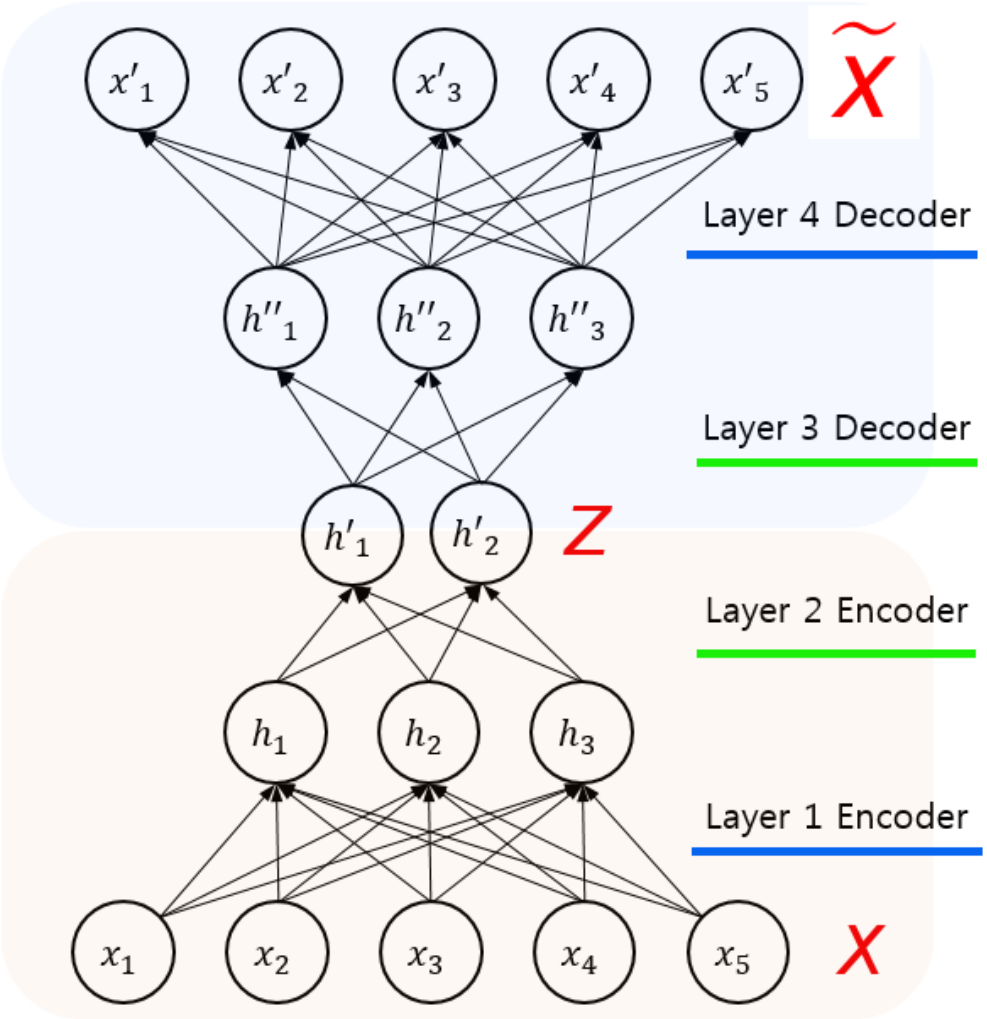
Auto-encoders



Example:

Keras blog.

Deep Auto-encoders



wikidocs.

Denoising Auto-encoders

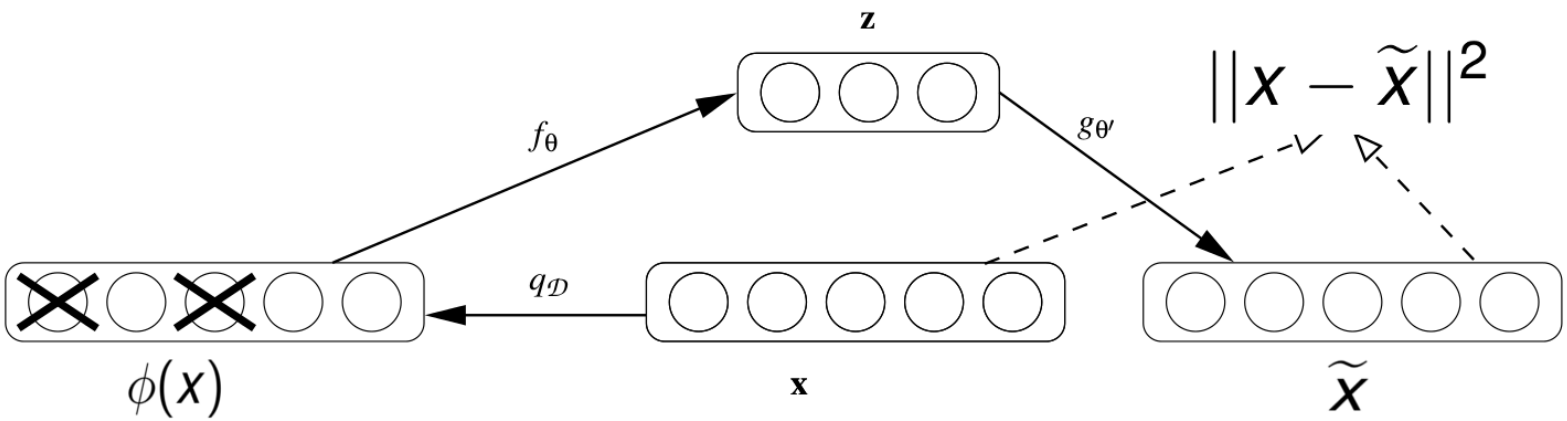
Basic auto-encoder:

$$J(x, W, b, b') = \frac{1}{n}\sum_{i=1}^{n} ||x - \underbrace{s(W^T(s(Wx + b))) + b'}_{\widetilde{x}}||^2$$

Denoising auto-encoder: build good representations by **recovering a corrupted input**.

$$J(x, W, b, b') = \frac{1}{n}\sum_{i=1}^{n} ||x - \underbrace{s(W^T(s(W\phi(x) + b))) + b'}_{\widetilde{x}}||^2$$

Denoising Auto-encoders



P.Vincent, 2010.

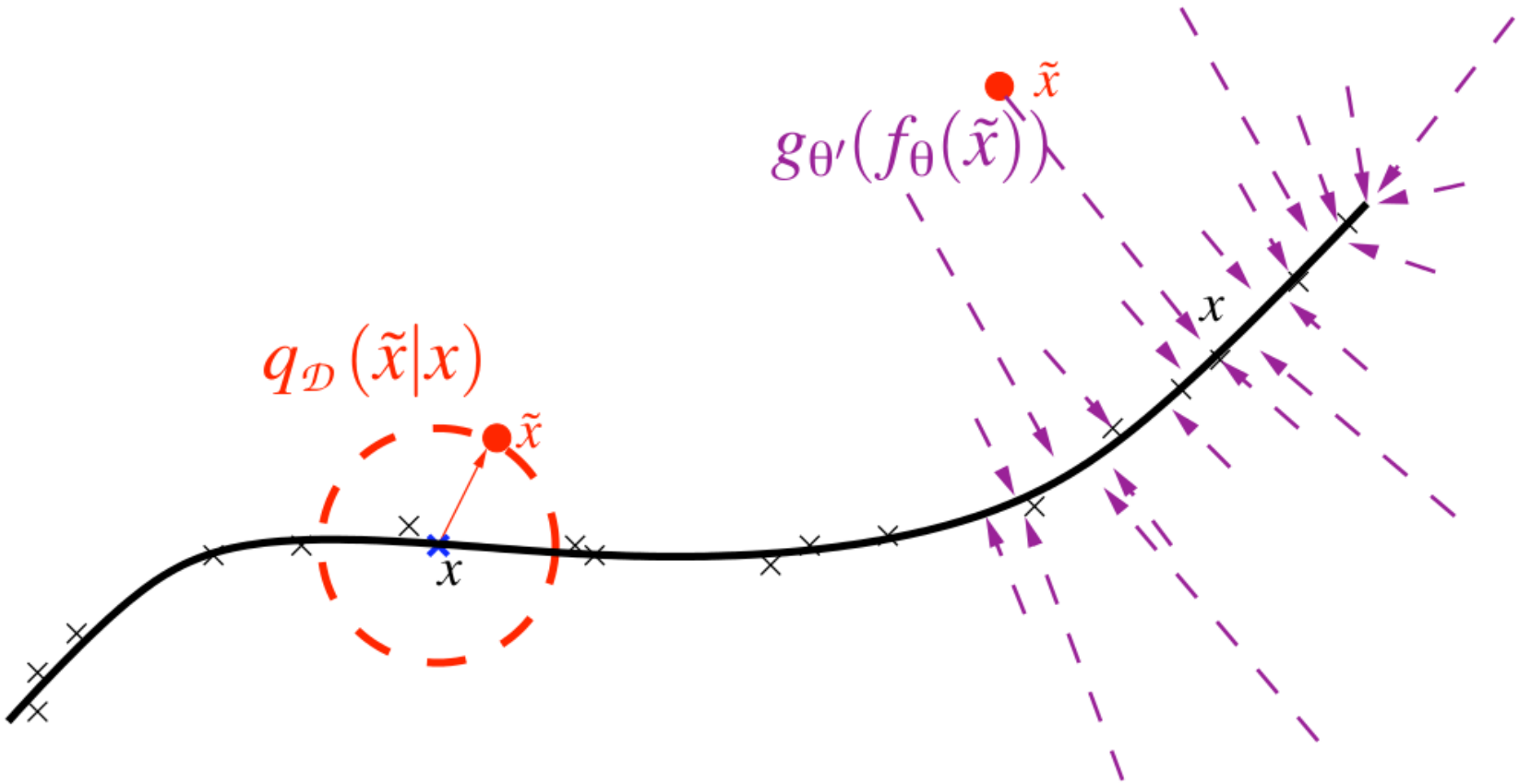
Denoising Auto-encoders

Unsupervised Manifold hypothesis:
Data in high dimensional spaces concentrate in sub-manifolds of much lower dimensionality.

Denoising Auto-encoders

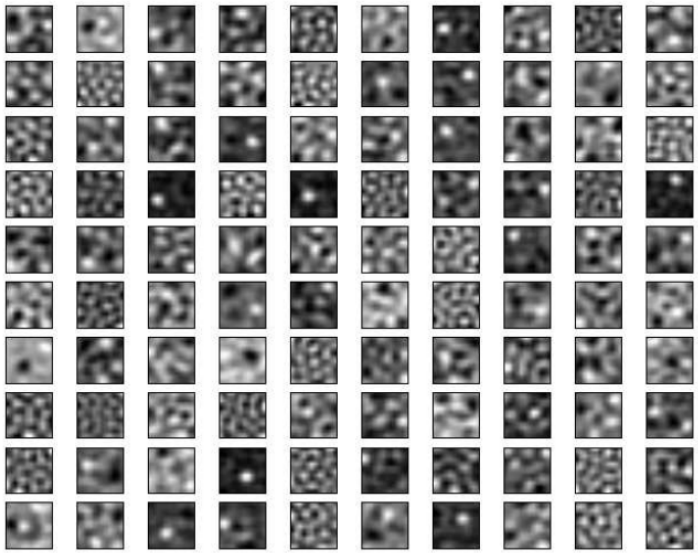
Manifolds. (G.Mesnil.)

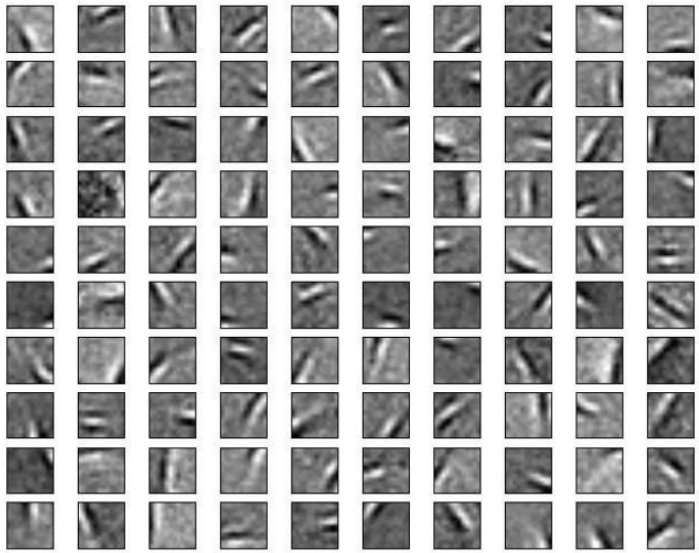
Denoising Auto-encoders



Manifold learning perspective. (P.Vincent, 2010.)

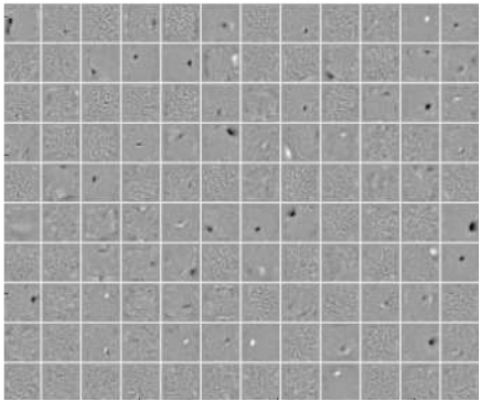
Denoising Auto-encoders



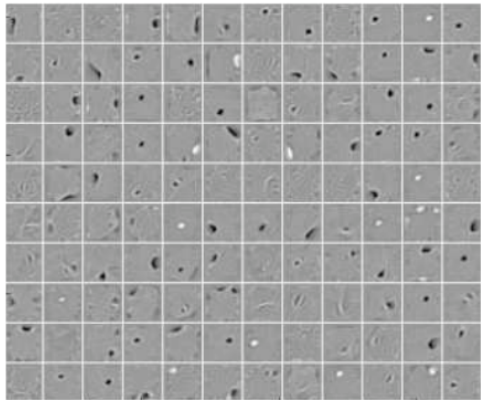


Left: filters of basic AE. Right: filters of DAE (Gaussian noise). (trained on natural images) (P.Vincent, 2010.)

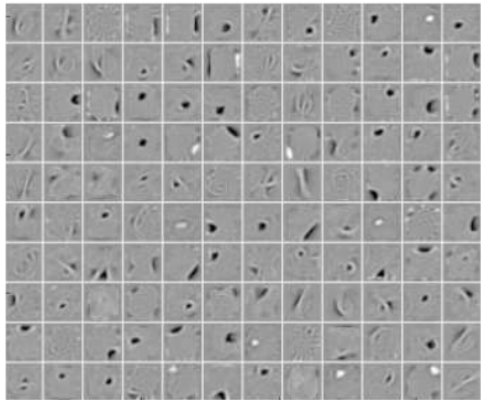
Denoising Auto-encoders



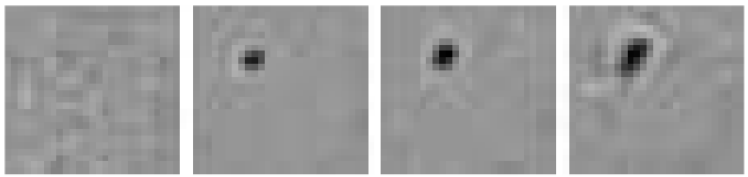
(a) No corruption



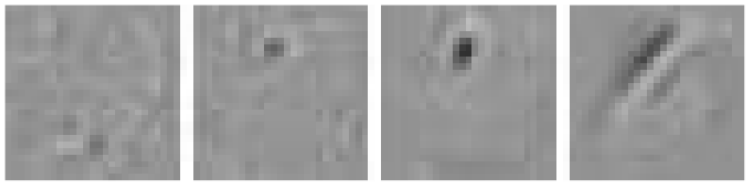
(b) 25% corruption



(c) 50% corruption



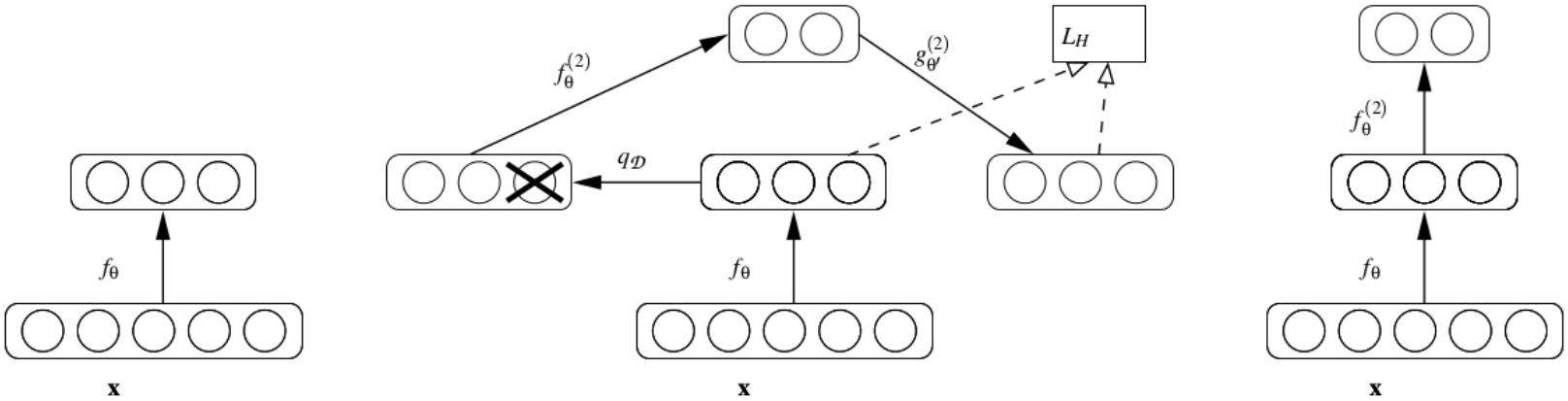
(d) Neuron A (0%, 10%, 20%, 50% corruption)



(e) Neuron B (0%, 10%, 20%, 50% corruption)

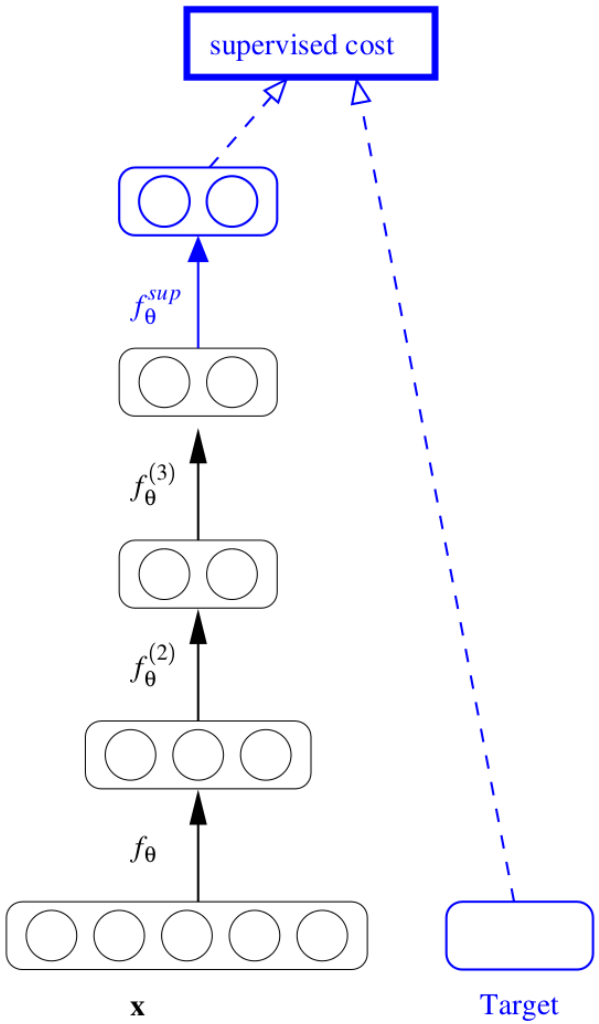
filters of DAE (zero-masking noise). (trained on MNIST) (P.Vincent, 2010.)

Stacked Denoising Auto-encoders



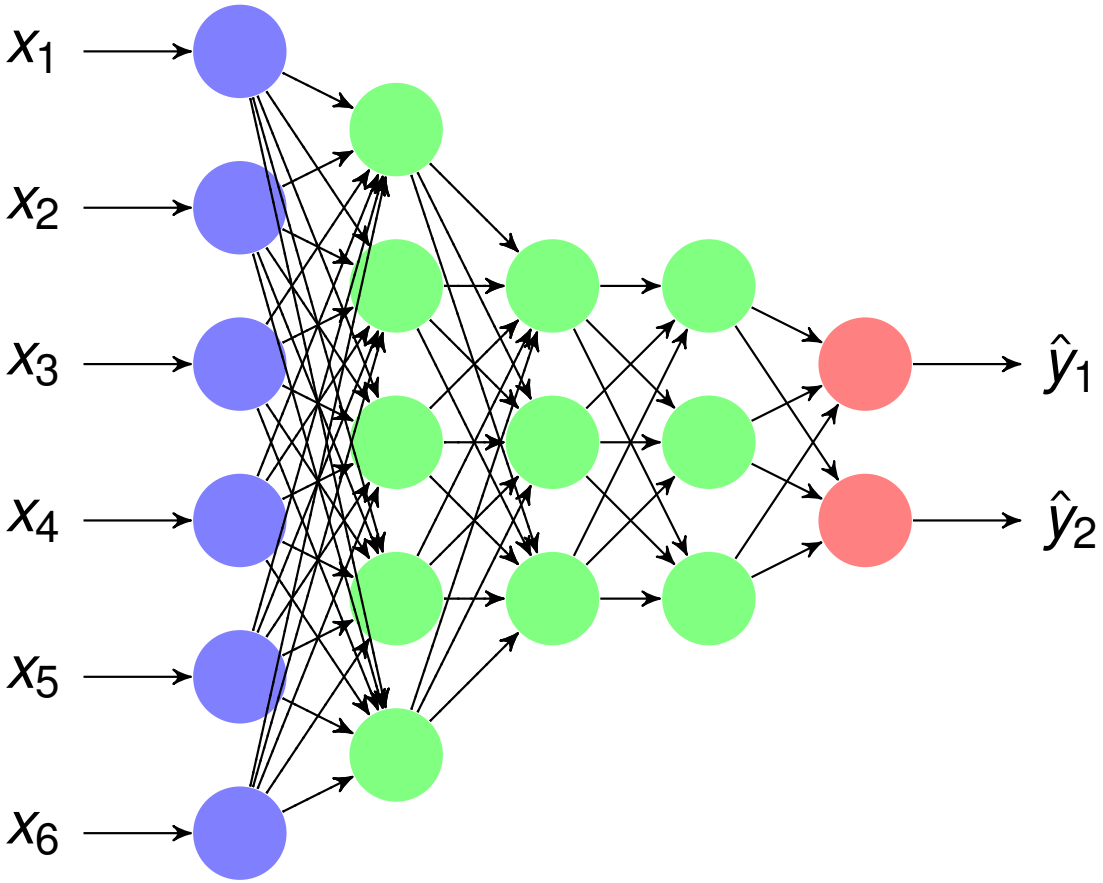
Stacked denoising AEs. (pre-training) (P.Vincent, 2010.)

Stacked Denoising Auto-encoders



Stacked denoising AEs. (fine-tunning) (P.Vincent, 2010.)

Layer-wise pre-training: auto-encoders

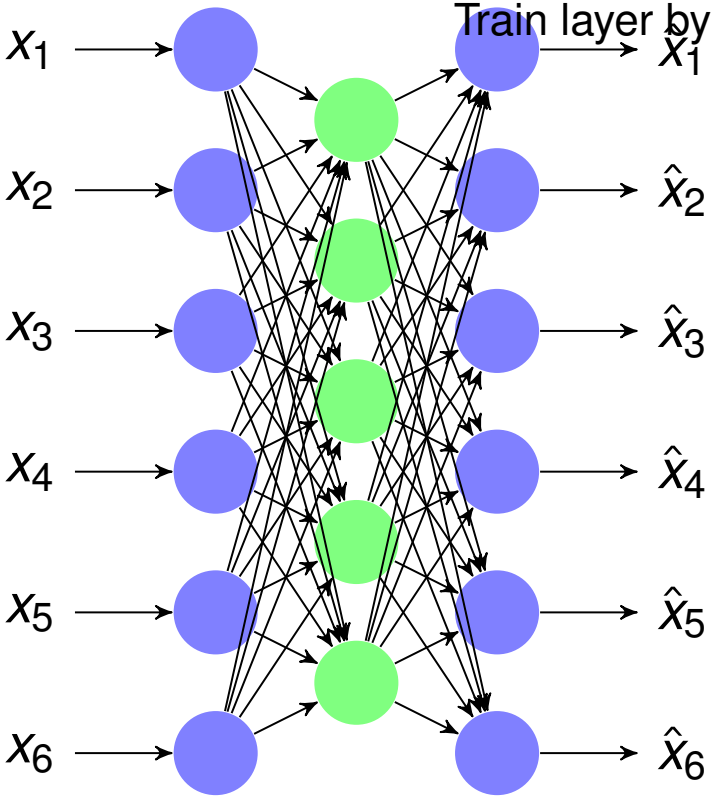


A DNN to train

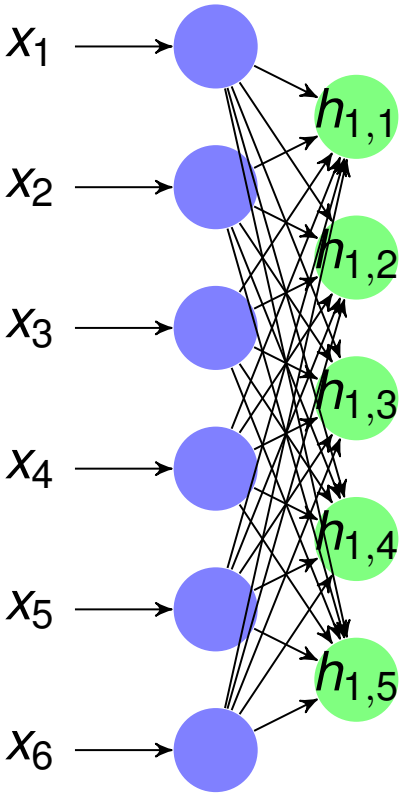
Layer-wise pre-training: auto-encoders

1) Step 1: **Unsupervised layer-wise pre-training**

Train layer by layer **sequentially** using **only x** (labeled or unlabeled)



Layer-wise pre-training: auto-encoders



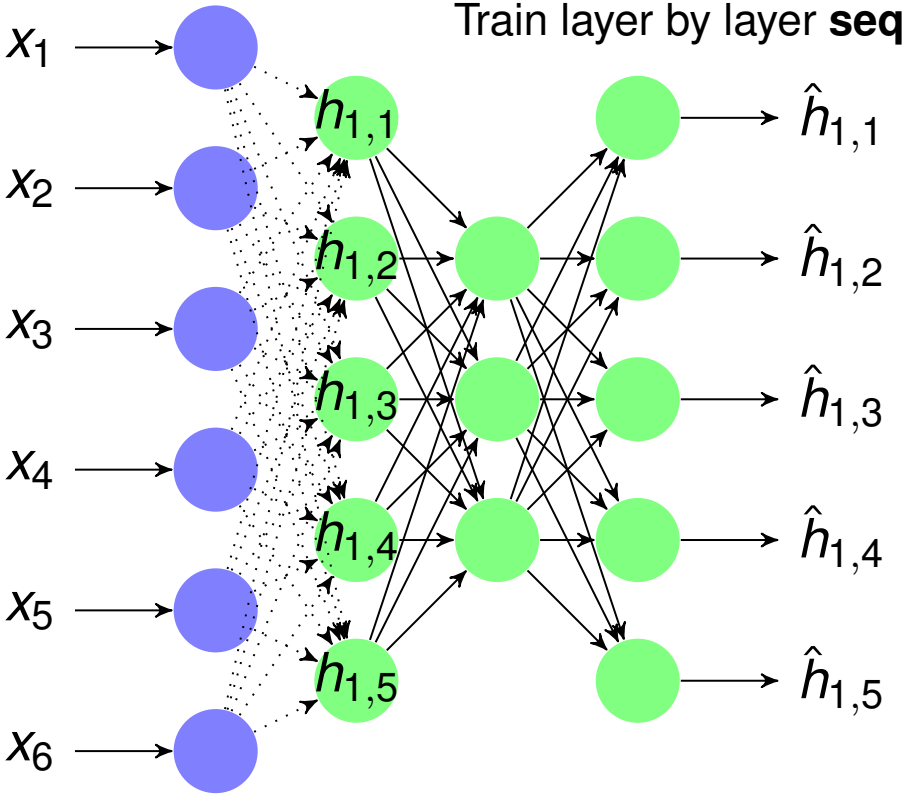
1) Step 1: **Unsupervised layer-wise pre-training**

Train layer by layer **sequentially** using **only x** (labeled or unlabeled)

Layer-wise pre-training: auto-encoders

1) Step 1: **Unsupervised layer-wise pre-training**

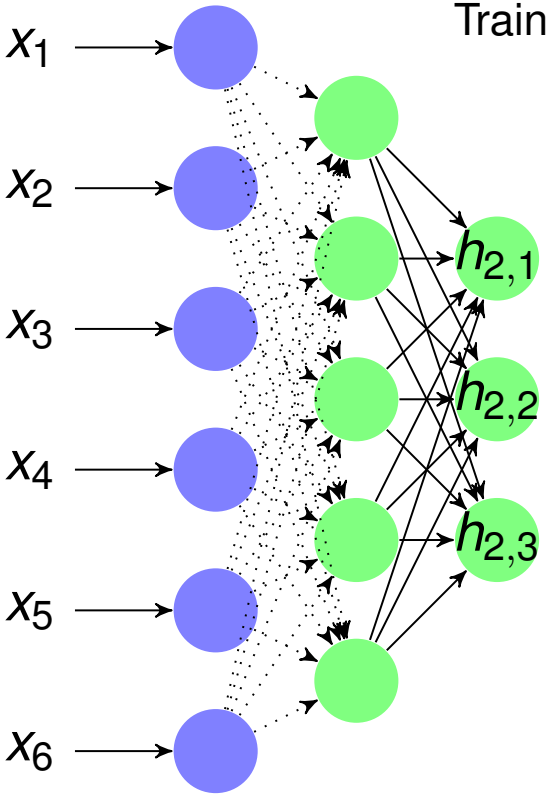
Train layer by layer **sequentially** using **only x** (labeled or unlabeled)



Layer-wise pre-training: auto-encoders

1) Step 1: **Unsupervised layer-wise pre-training**

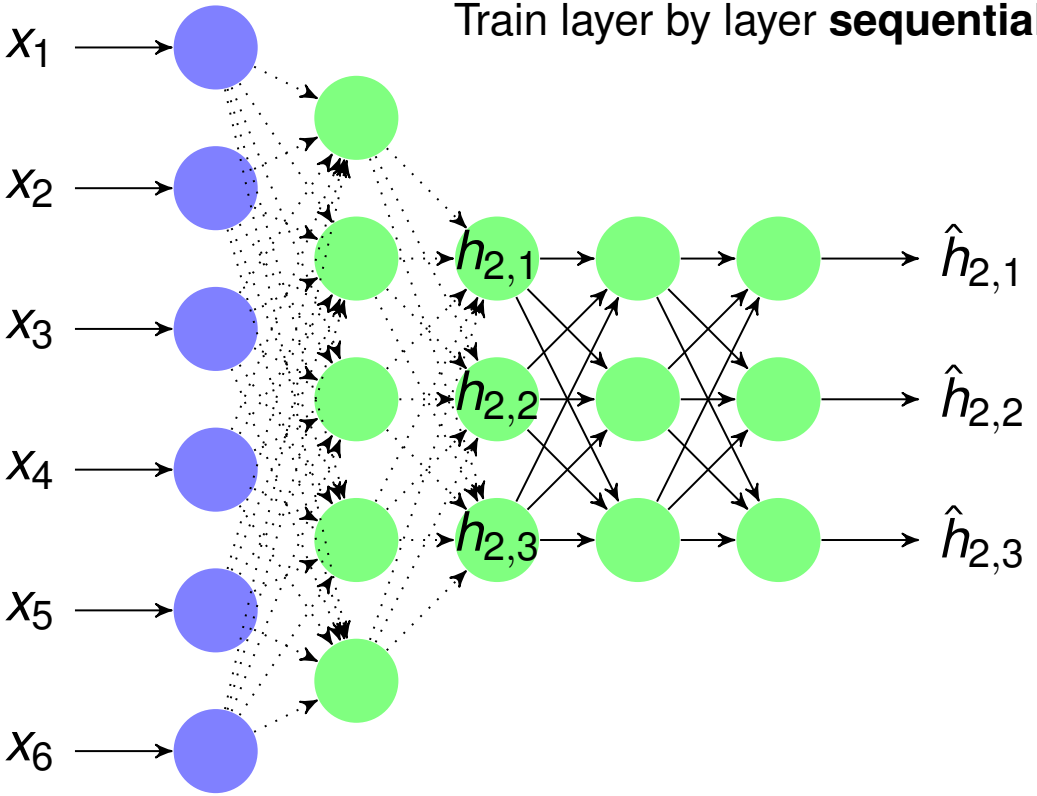
Train layer by layer **sequentially** using **only x** (labeled or unlabeled)



Layer-wise pre-training: auto-encoders

1) Step 1: **Unsupervised layer-wise pre-training**

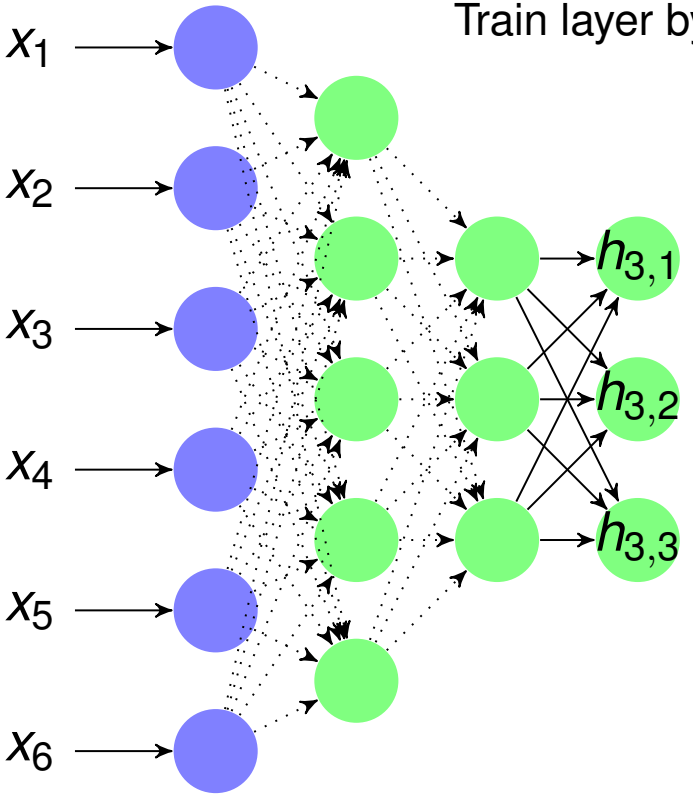
Train layer by layer **sequentially** using **only x** (labeled or unlabeled)



Layer-wise pre-training: auto-encoders

1) Step 1: **Unsupervised layer-wise pre-training**

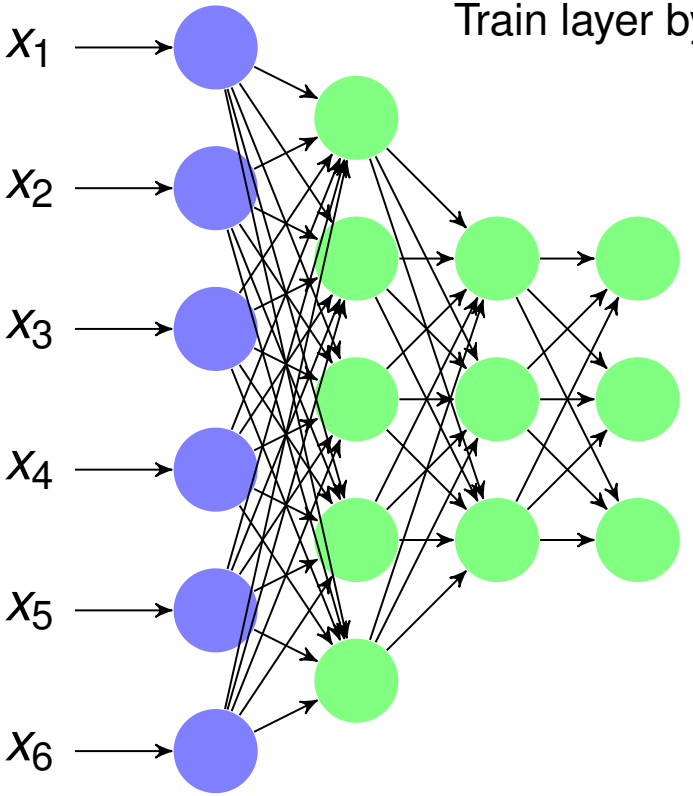
Train layer by layer **sequentially** using **only x** (labeled or unlabeled)



Layer-wise pre-training: auto-encoders

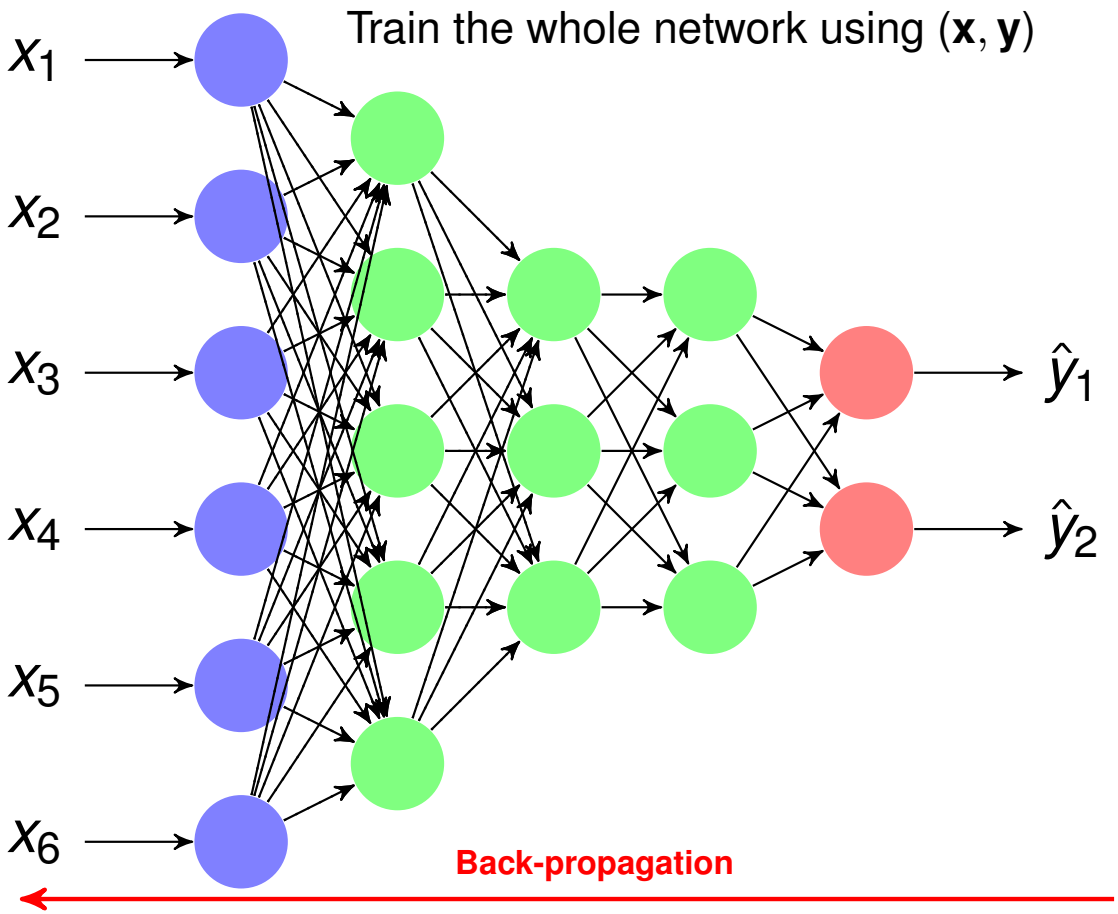
1) Step 1: **Unsupervised layer-wise pre-training**

Train layer by layer **sequentially** using **only x** (labeled or unlabeled)

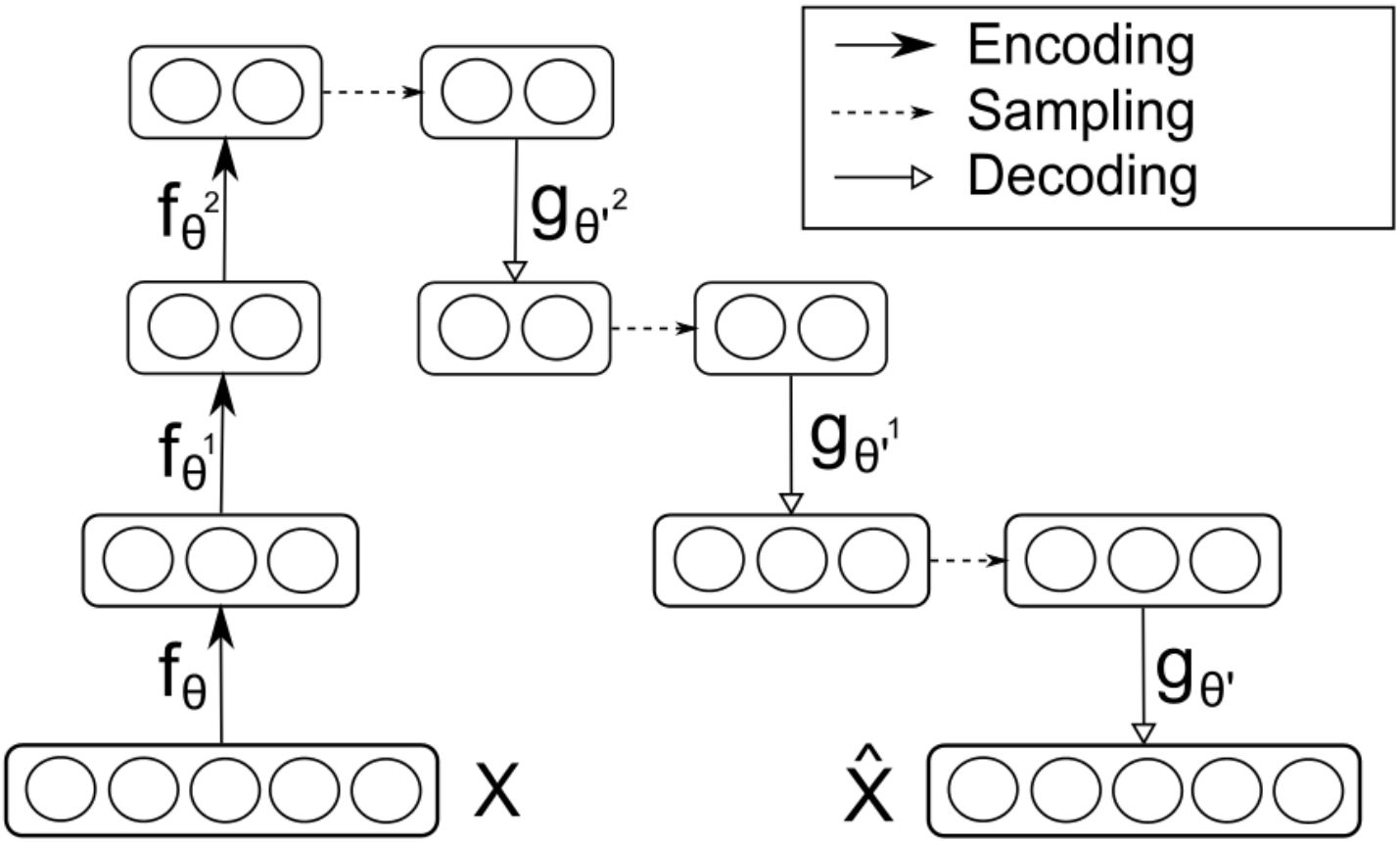


Layer-wise pre-training: auto-encoders

2) Step 2: **Supervised training**

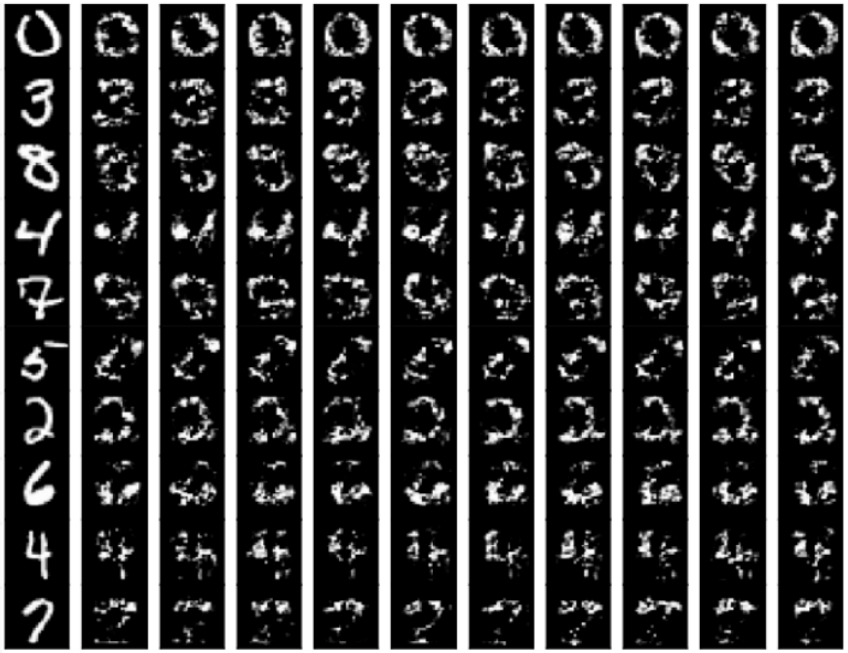


Generating Samples with SAE, SDAE

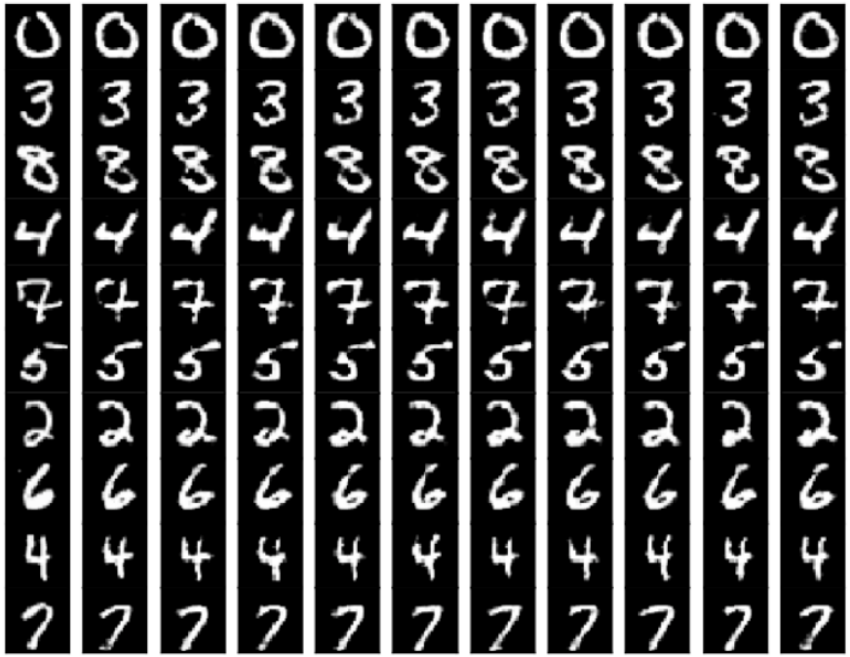


P.Vincent, 2010.

Generating Samples with SAE, SDAE



(a) SAE



(b) SDAE

P.Vincent, 2010.

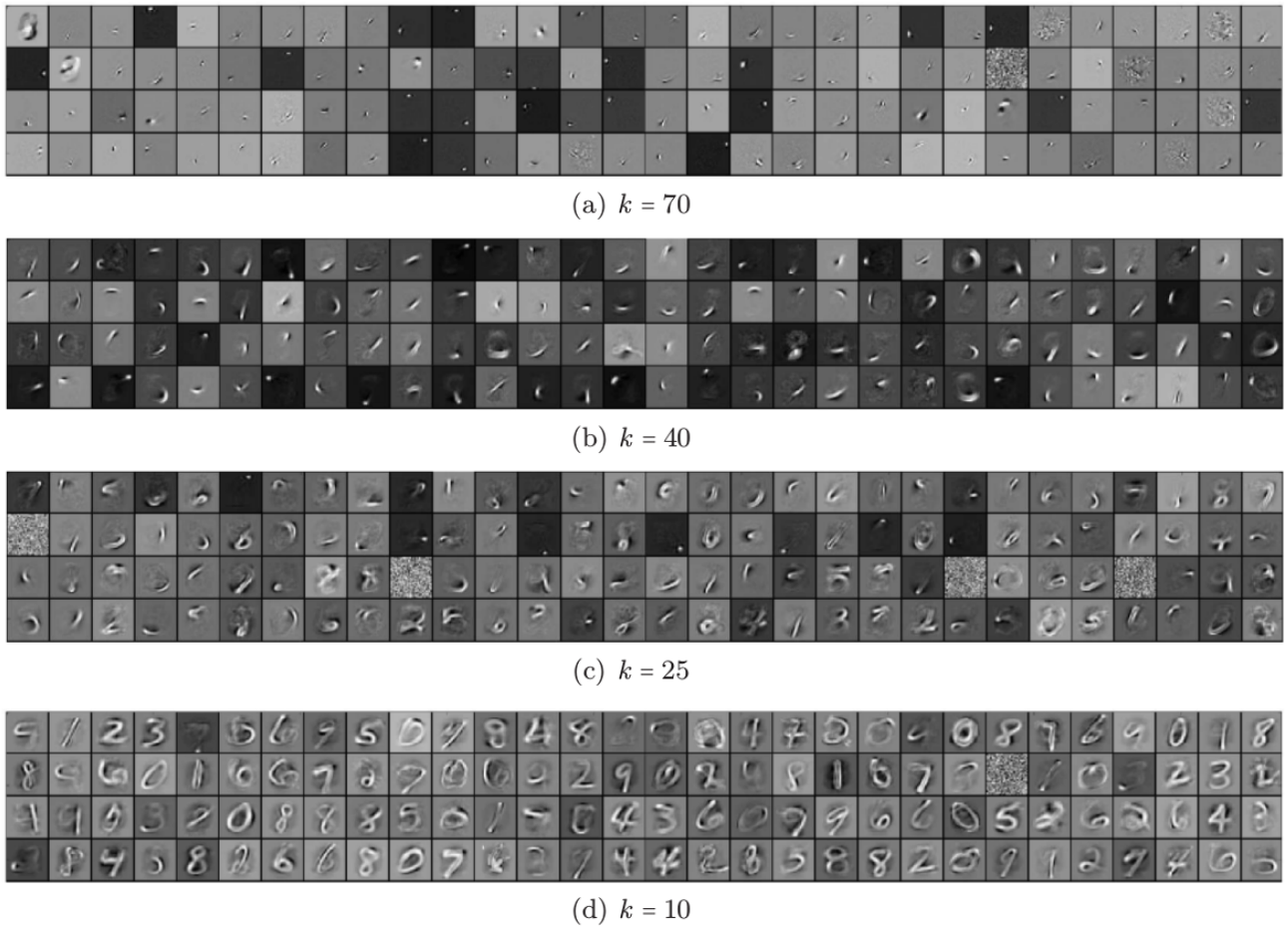
Sparse Auto-encoders

$$Cost = reconstruction + sparsity$$

Objective: build **sparse features** (most components are zeros).
Example: K-sparse AEs. (Makhzani et al. 13)

$$\hat{\boldsymbol{z}}_i = \underset{\boldsymbol{z}}{\text{argmin}} \|\boldsymbol{x}_i - W\boldsymbol{z}\|_2^2 \quad s.t. \quad \|\boldsymbol{z}\|_0 < k$$

Sparse Auto-encoders



(a) $k = 70$

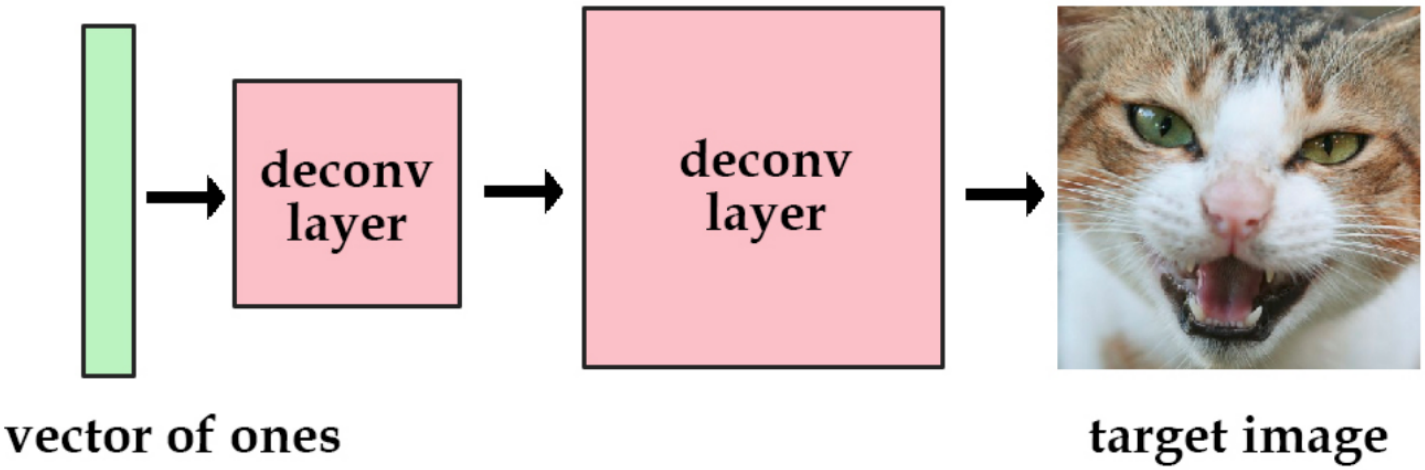
(b) $k = 40$

(c) $k = 25$

(d) $k = 10$

Filters of the k -sparse auto-encoder for different sparsity levels k, learnt from MNIST. (A. Makhzani, 2013.)

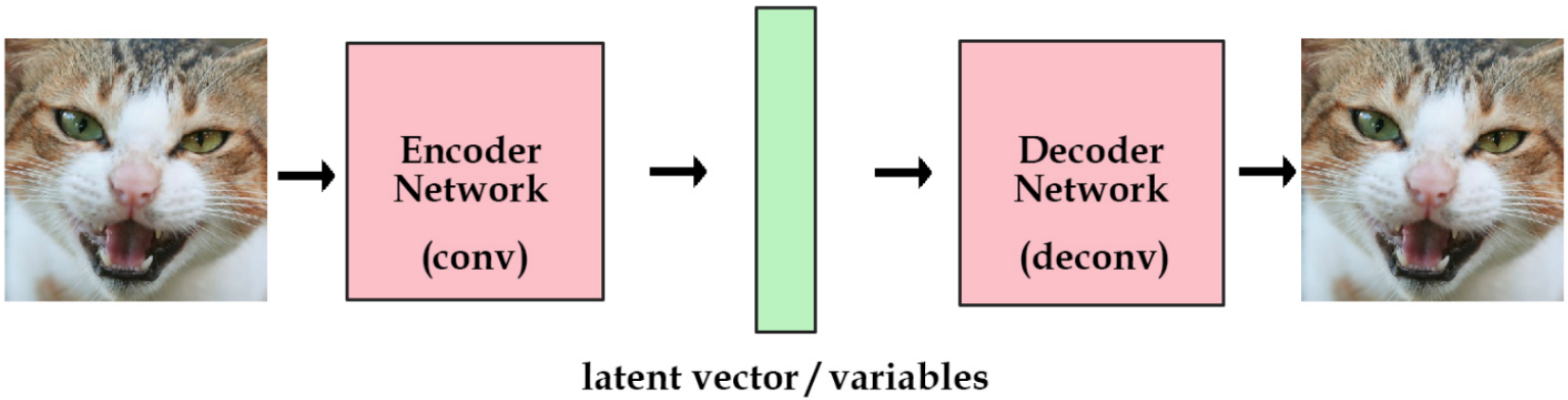
Variational Auto-encoders



How to generate samples?

kvfrans.

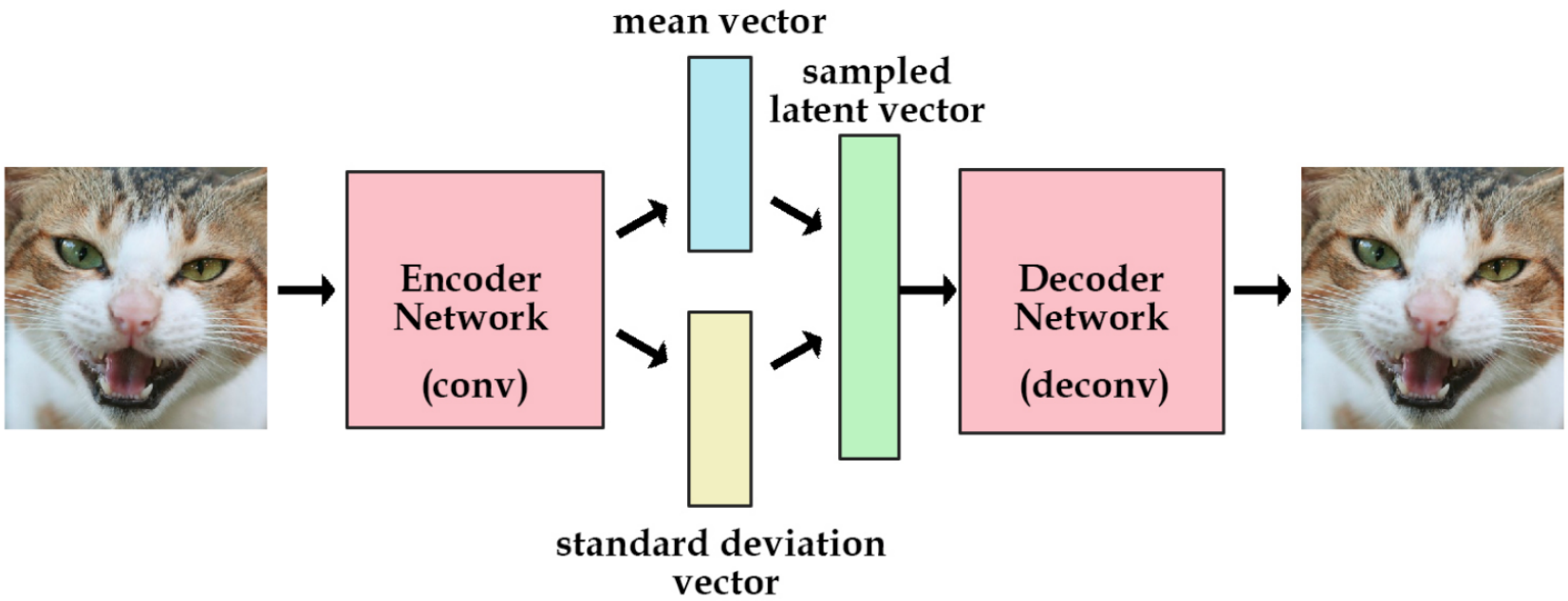
Variational Auto-encoders



Sample AE. (no constraints on the latent code)

kvfrans.

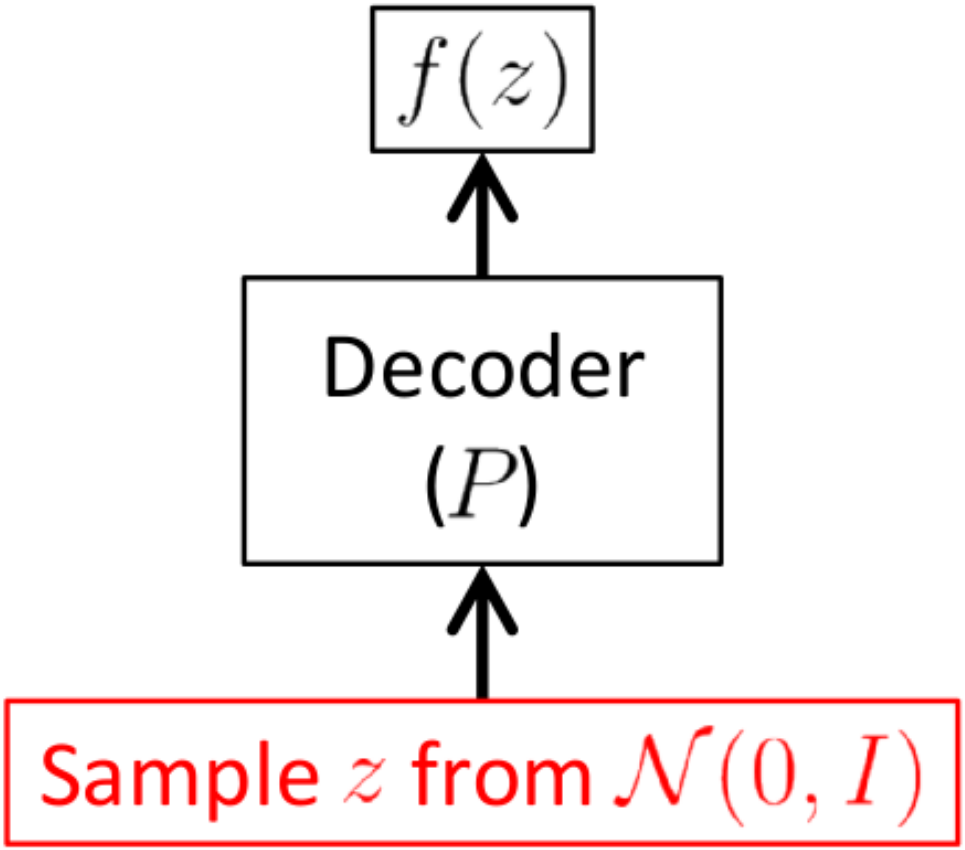
Variational Auto-encoders



VAE: AE with constraints on the latent code. (unit Gaussian distribution)

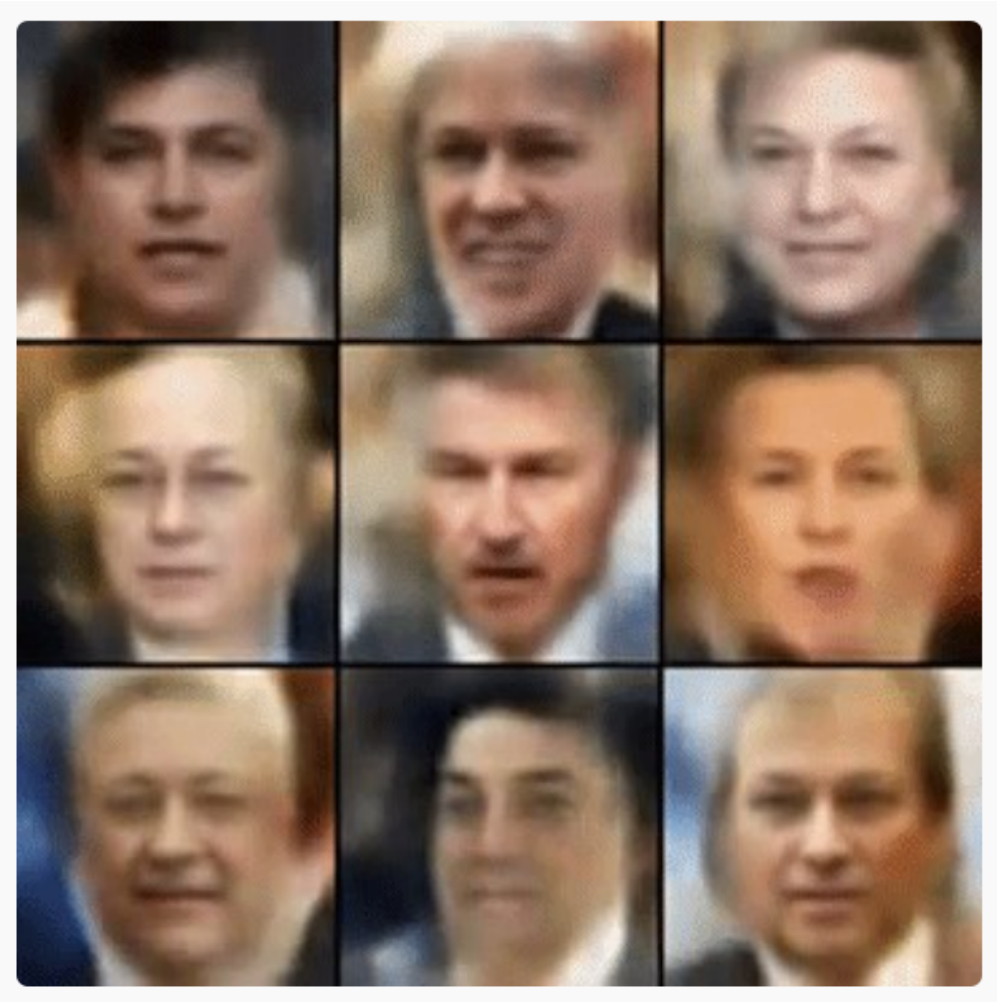
kvfrans.

Variational Auto-encoders



Sampling from VAE.

Variational Auto-encoders



Fictional celebrity faces generated by a variational autoencoder.
Link to video: `https://www.youtube.com/watch?v=XNZIN7Jh3Sg`. (J.Altosaar.)

Contractive Auto-encoders

Regularize the AE so the **features** will be **robust to slight in the input.** (local invariance)

$$cost = reconstruction + contraction$$

$$\begin{cases} h & = f(x) = s_f(Wx + b_h) \quad \text{(code)} \\ ||J_f(x)||_F^2 & = \sum_{ij} \left(\frac{\partial h_j(x)}{\partial x_i}\right)^2 \quad \text{(contraction)} \\ \mathcal{J}_{CAE}(x;\theta) & = \frac{1}{n}\sum_{x \in D_n}(||x - \widetilde{x}||^2 + \lambda||J_f(x)||_F^2) \end{cases}$$

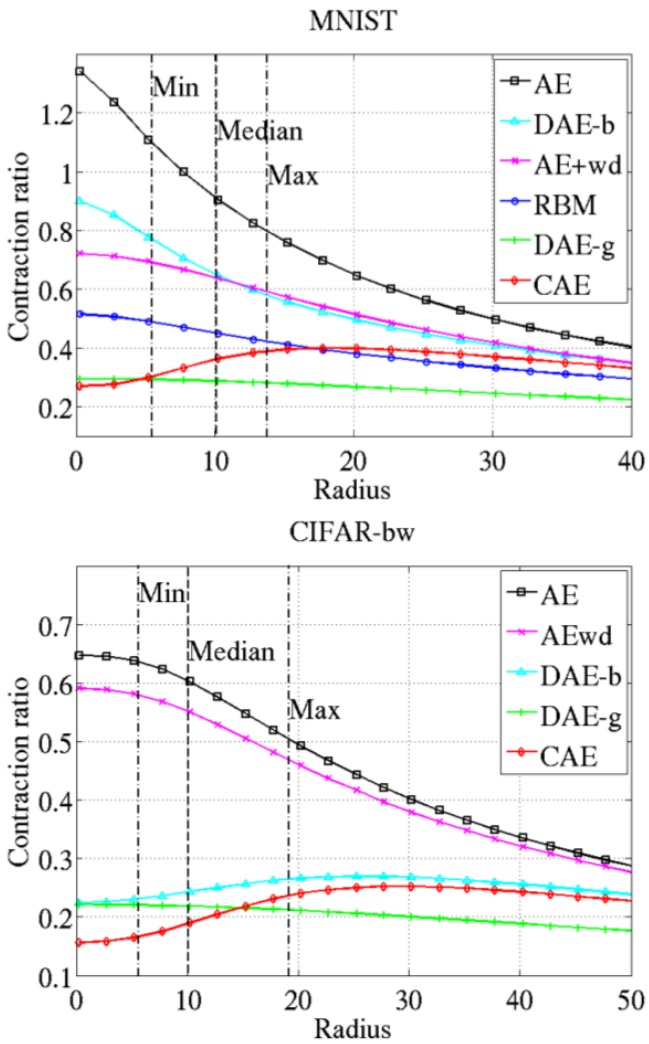
Contractive Auto-encoders

Regularize the AE so the **features** will be **robust to slight in the input.** (local invariance)

$$cost = reconstruction + contraction$$

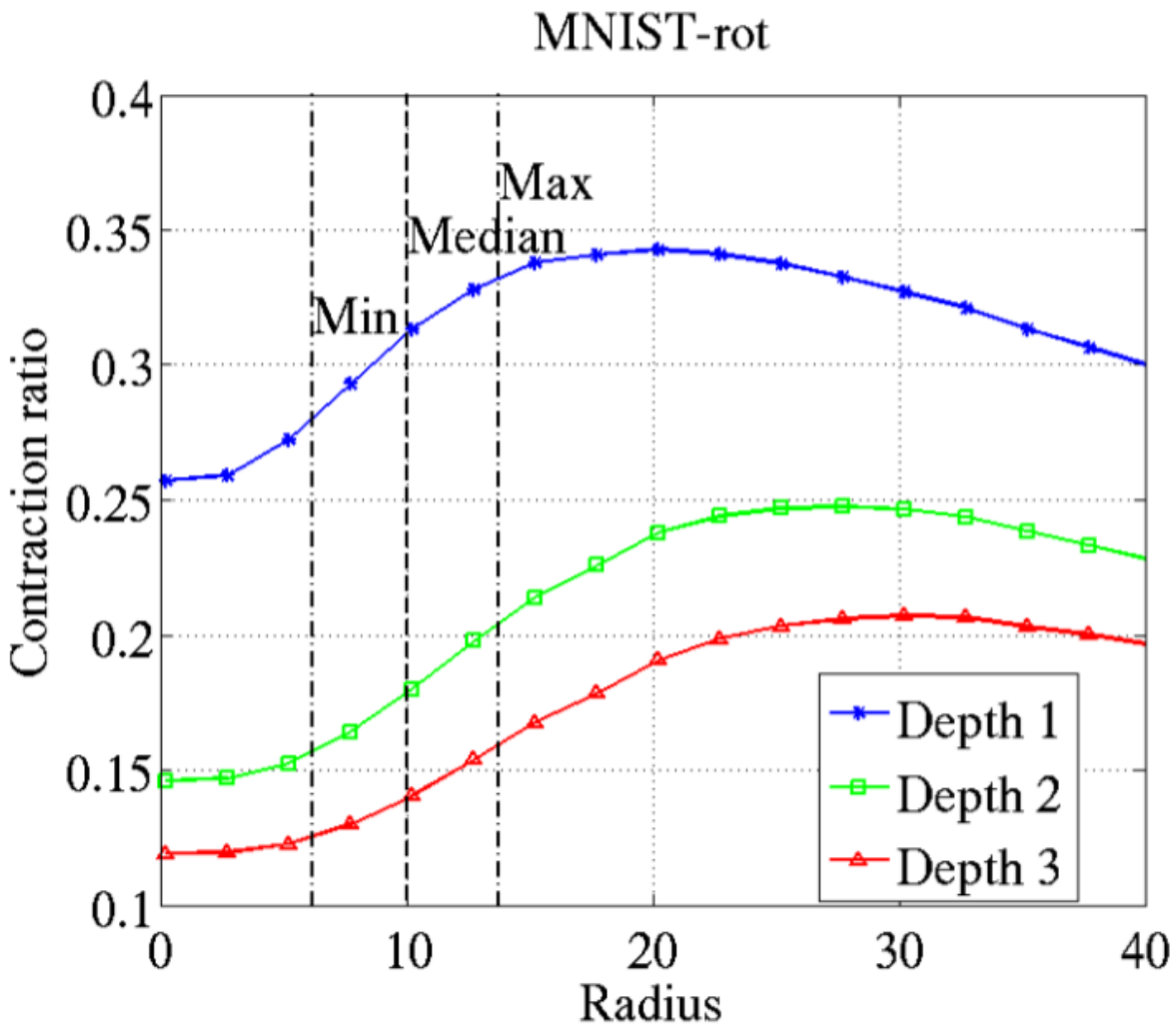
$$\begin{cases} h & = f(x) = s_f(Wx + b_h) \quad \text{(code)} \\ ||J_f(x)||_F^2 & = \sum_{ij} \left(\frac{\partial h_j(x)}{\partial x_i}\right)^2 \quad \text{(contraction)} \\ \mathcal{J}_{CAE}(x;\theta) & = \frac{1}{n}\sum_{x \in D_n}(||x - \widetilde{x}||^2 + \lambda||J_f(x)||_F^2) \end{cases}$$

Contractive Auto-encoders



Contraction ration: MNIST (top) and CIFAR-bw (bottom). (S.Rifai,

Contractive Auto-encoders



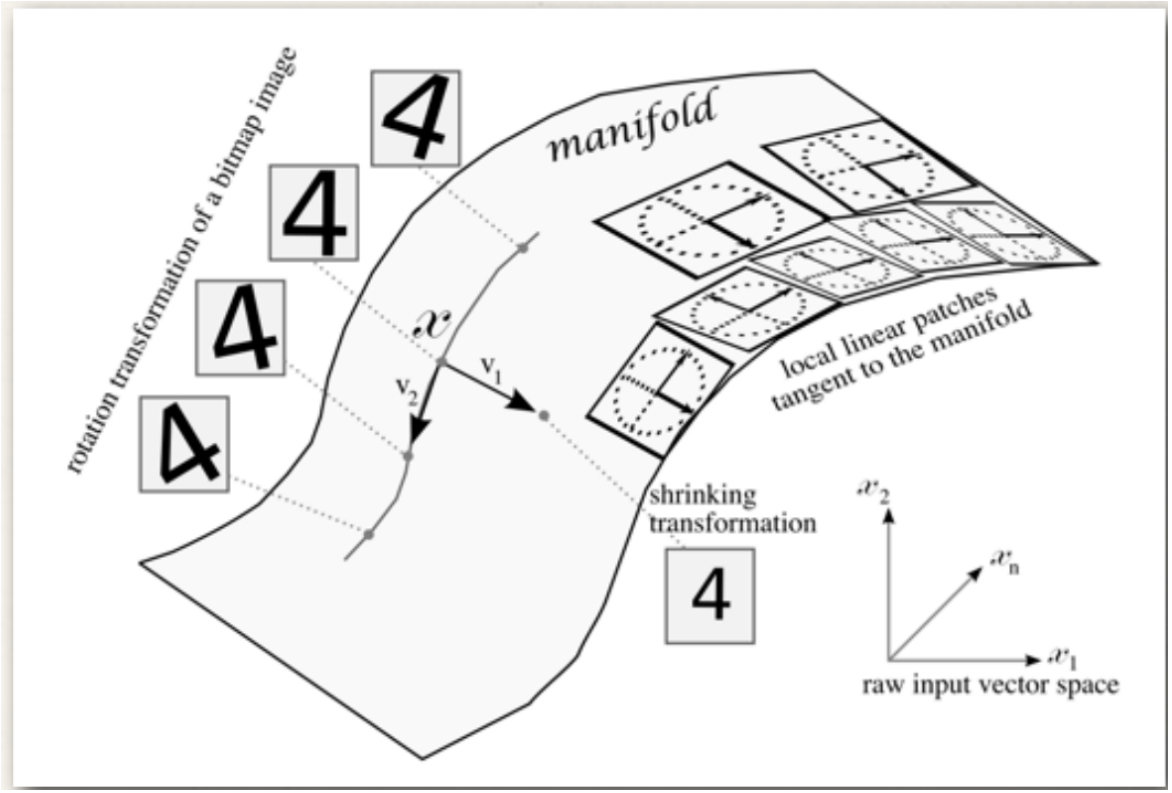
Contraction ration with respect to depth. (S.Rifai, 2011.)

Higher Order Contractive Auto-encoders

$$cost = reconstruction + \underbrace{contraction}_{\text{1st derivative}} + \underbrace{\text{curvature of the contraction}}_{\text{2nd derivative}}$$

Objective: extract local charts (local directions of variations)

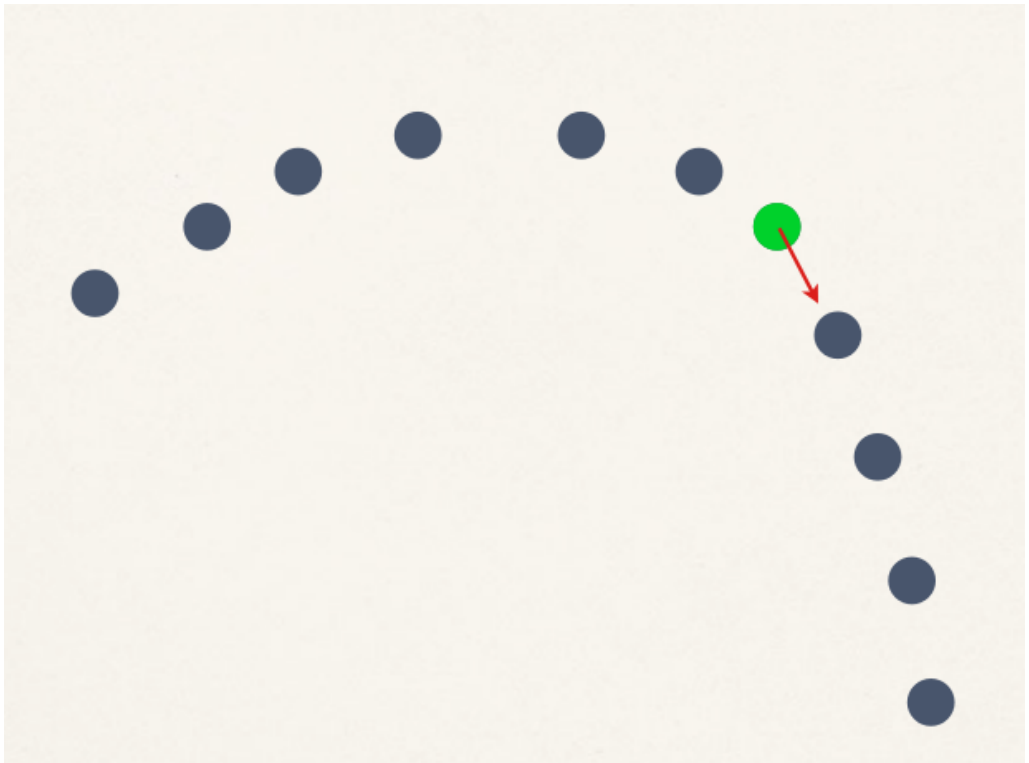
Higher Order Contractive Auto-encoders



Unsupervised invariance to local transformations. (C.H.Martin.)

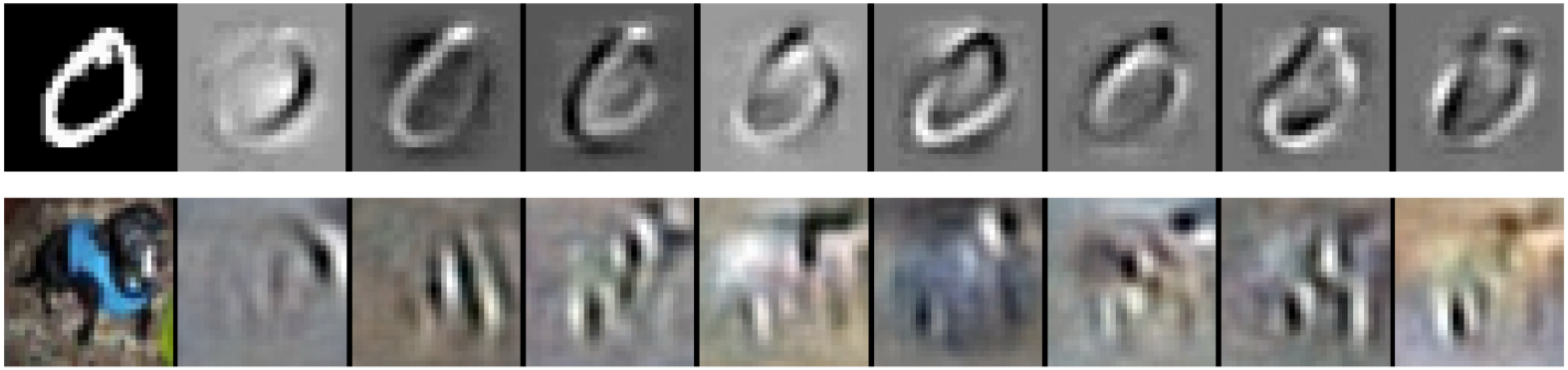
Higher Order Contractive Auto-encoders

Use High CAE to extract local tangents directions of variations to ensure invariance of classifier to these directions.

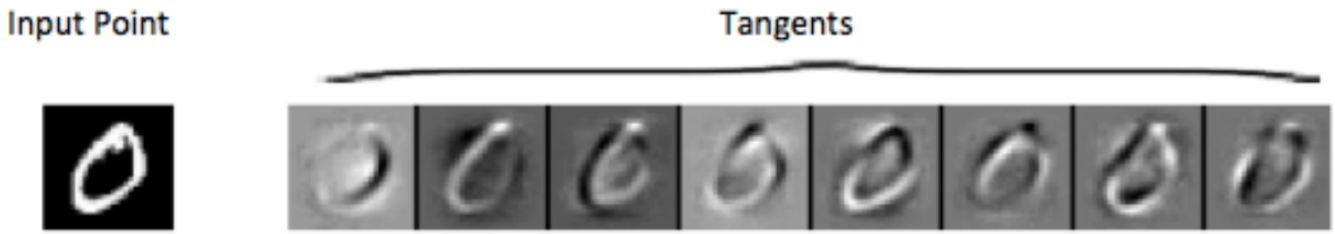


(S.Rifai, 2011.)

Higher Order Contractive Auto-encoders



Learned tangents over MNIST and CIFAR-10. (S.Rifai, 2011.)

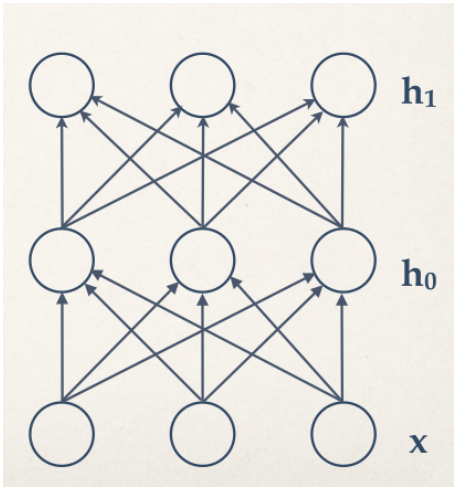


Higher Order Contractive Auto-encoders

1. Train stack of CAE layers.
2. Compute the tangents of each example by SVD of $\frac{\partial h_1}{\partial x}$.

$$\begin{cases} D & = \{x^{(1)}, \ldots, x^{(n)}\} \\ \Downarrow & \\ D' & = \{(x^{(1)}, \frac{\partial h_1}{\partial x}(x^{(1)})), \ldots, (x^{(1)}, \frac{\partial h_1}{\partial x}(x^{(1)}))\} \end{cases}$$

(S.Rifai, 2011.)



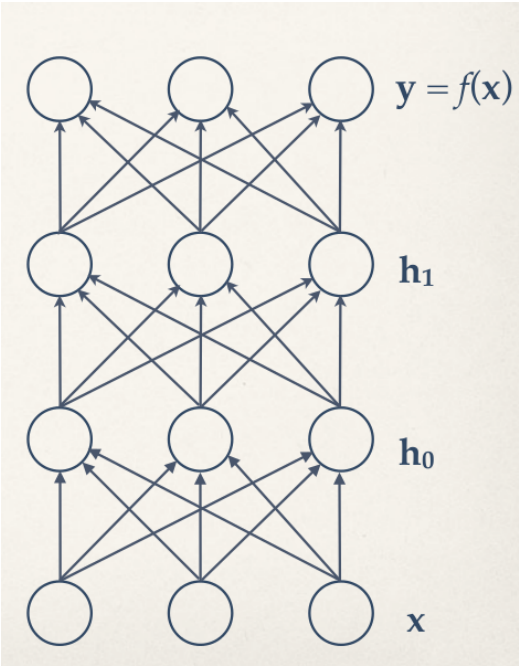
Higher Order Contractive Auto-encoders

1. Train stack of CAE layers.
2. Compute the tangents of each example by SVD of $\frac{\partial h_1}{\partial x}$.

$$\begin{cases} D & = \{x^{(1)}, \ldots, x^{(n)}\} \\ \Downarrow & \\ D' & = \{(x^{(1)}, \frac{\partial h_1}{\partial x}(x^{(1)})), \ldots, (x^{(1)}, \frac{\partial h_1}{\partial x}(x^{(1)}))\} \end{cases}$$

3. Train a logistic regression on top of the CAE stack with the tangent propagation penalty.

(S.Rifai, 2011.)

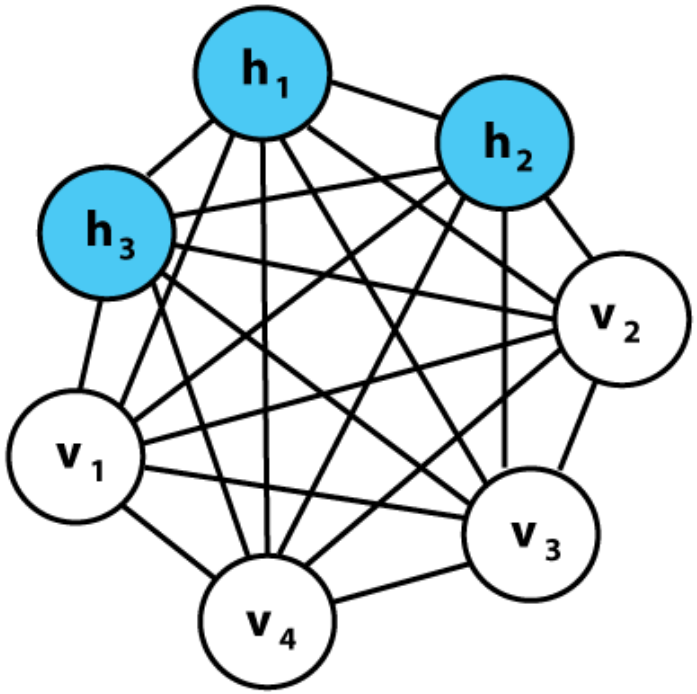


Next generation AEs

Toward: Adversarial Autoencoders, GANs+VAE, CNN+AEs.

Restricted Boltzmann Machines

Boltzmann Machines: **undirected graphical** model.
Fully connected graph.



v: visible units. **h**: hidden units. G.Hinton et al. 85

Restricted Boltzmann Machines

Restricted Boltzmann Machines

The input x is not **fully observed** $\Rightarrow$

$$x = \underbrace{v}_{\text{visible part}} + \underbrace{h}_{\text{hidden part}} .$$

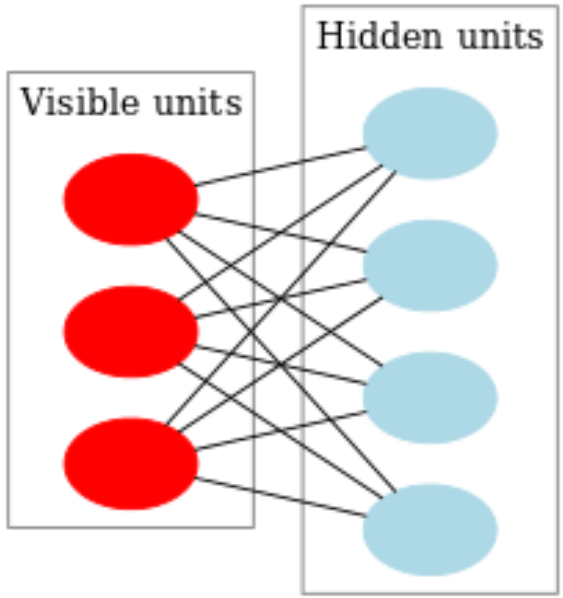


Figure 3: visible $\Longleftrightarrow$ hidden.

Binary variables. (possible extension to continuous variables).
(Wikipedia.)

Restricted Boltzmann Machines

Restricted Boltzmann Machines

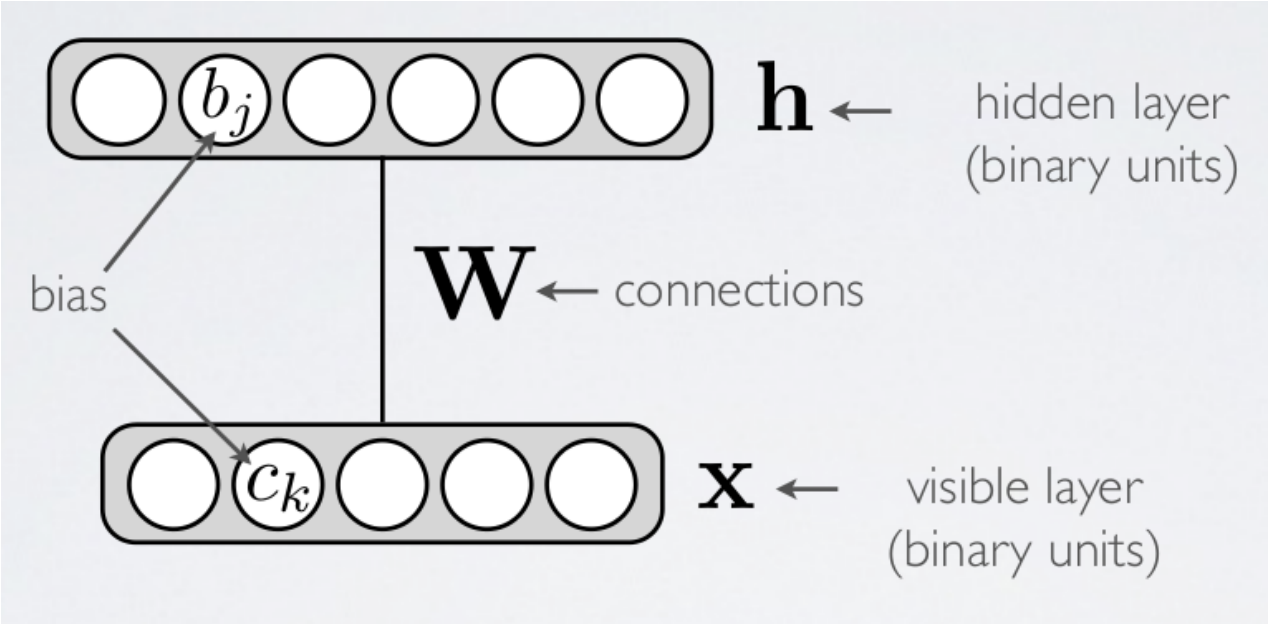


Figure 4: $x \Longleftrightarrow h$.

(H.Larochelle.)

Restricted Boltzmann Machines

Restricted Boltzmann Machines

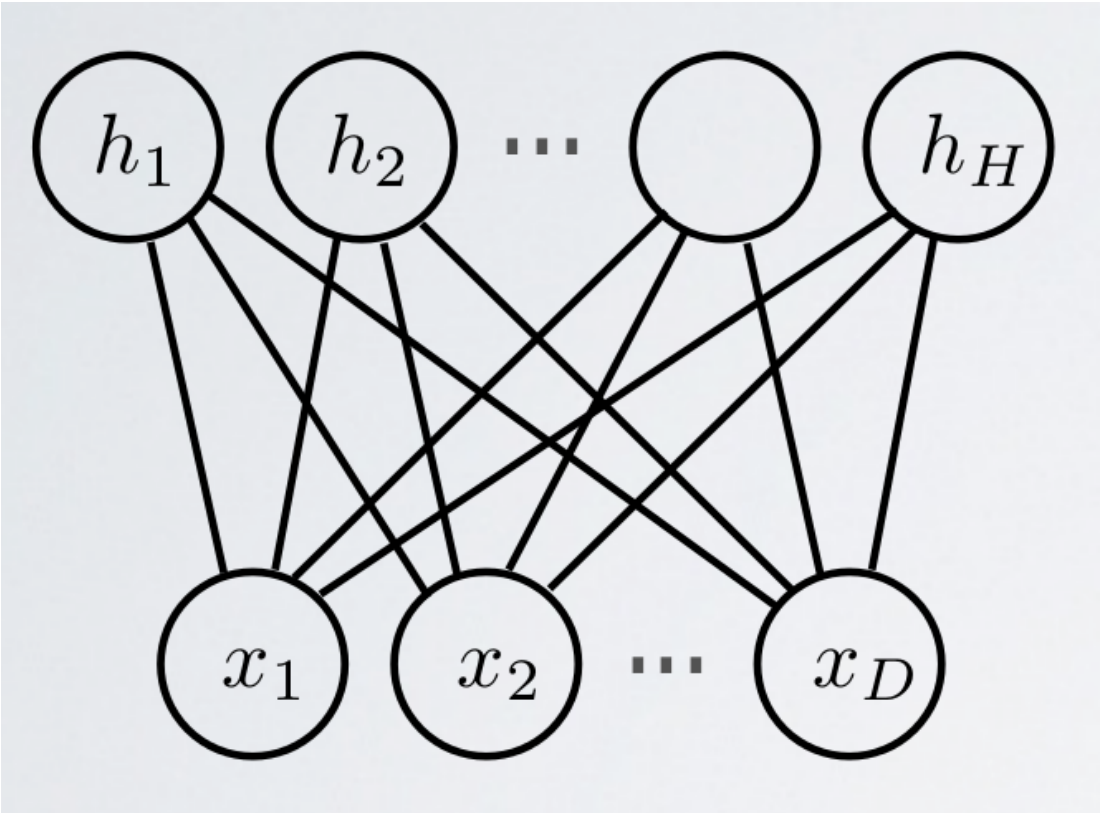


Figure 5: $x \Longleftrightarrow h$.

(H.Larochelle.)

Restricted Boltzmann Machines

Restricted Boltzmann Machines

RBMs are:

- **Generative** models.
- Model the **density** $P(v)$ (i.e. $P(input)$).
- **Energy** based model $P(v) = f(E(v, h))$.

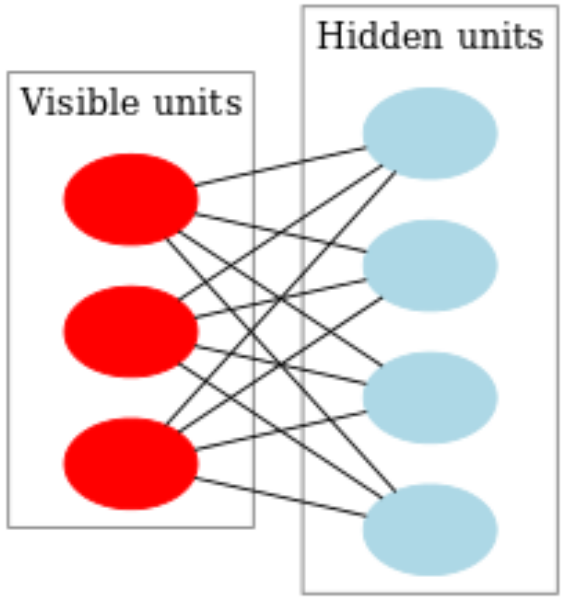
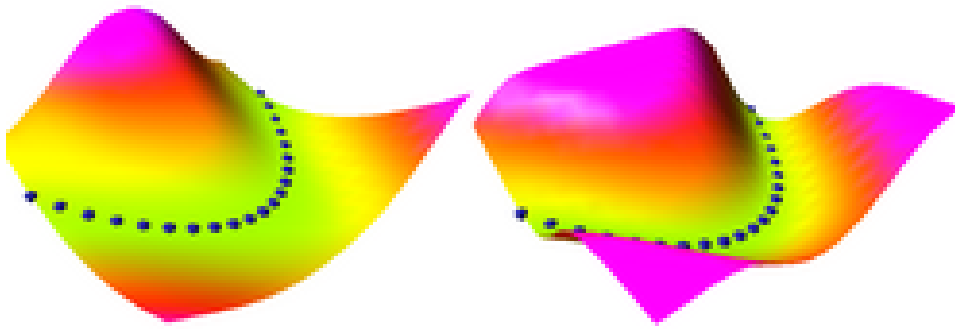


Figure 6: visible $\Longleftrightarrow$ hidden.

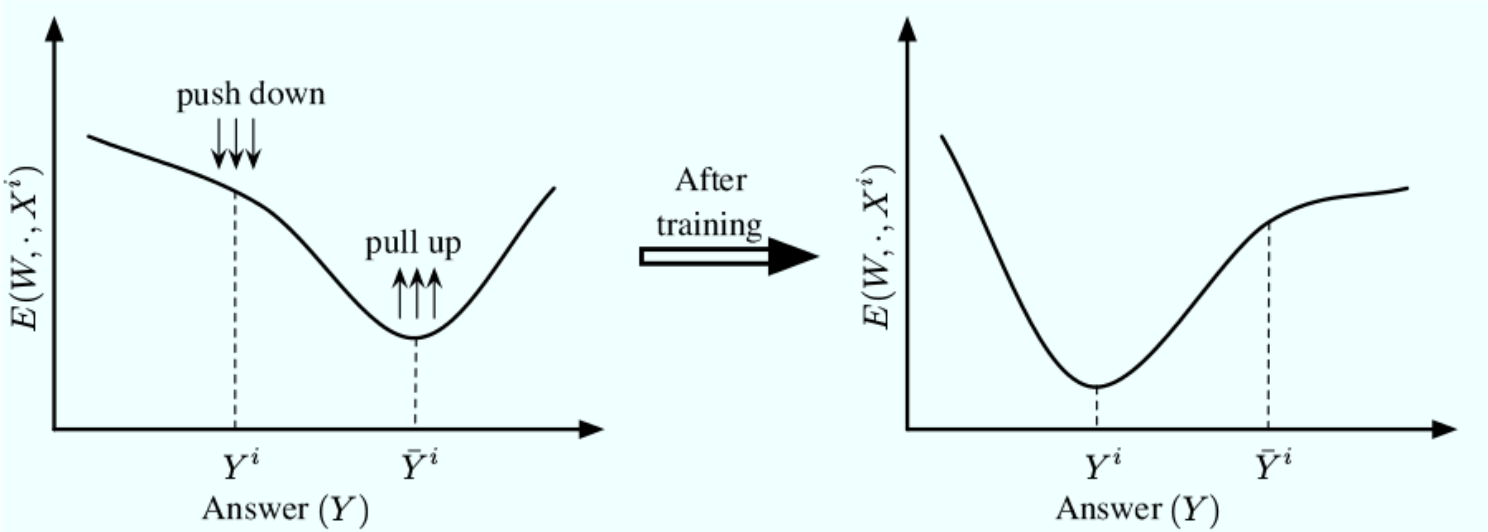
Restricted Boltzmann Machines

- RBMs are **energy** based models.
- It associates an energy $E(v, h)$ with every possible configuration of the system.
- Learning consists in modifying the energy shape.
- Physics: ***lower energy = more stability***.

(Y.LeCun)



Restricted Boltzmann Machines



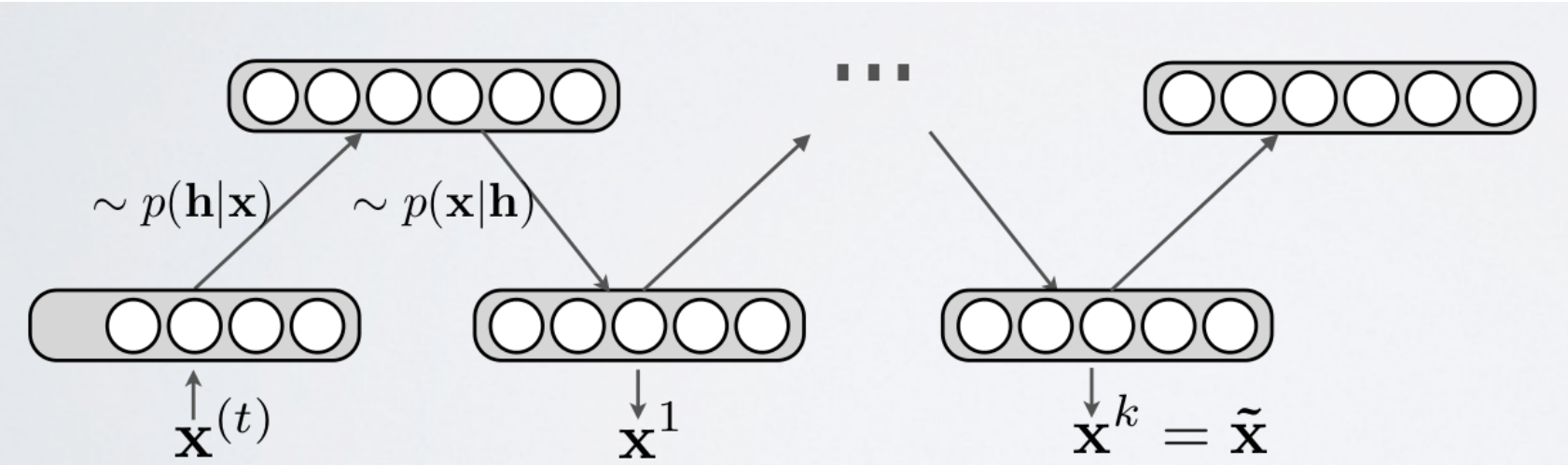
Learning:
Modify the energy shape so that the **desirable configurations** have **lower energy**.
(Y.LeCun)

Restricted Boltzmann Machines

Training of RBMs: **k-steps Contrastive Divergence** (CD-k).

Ingredients:

- Markov chain using Gibbs sampling.
- Gradient approximation (gradient descent).



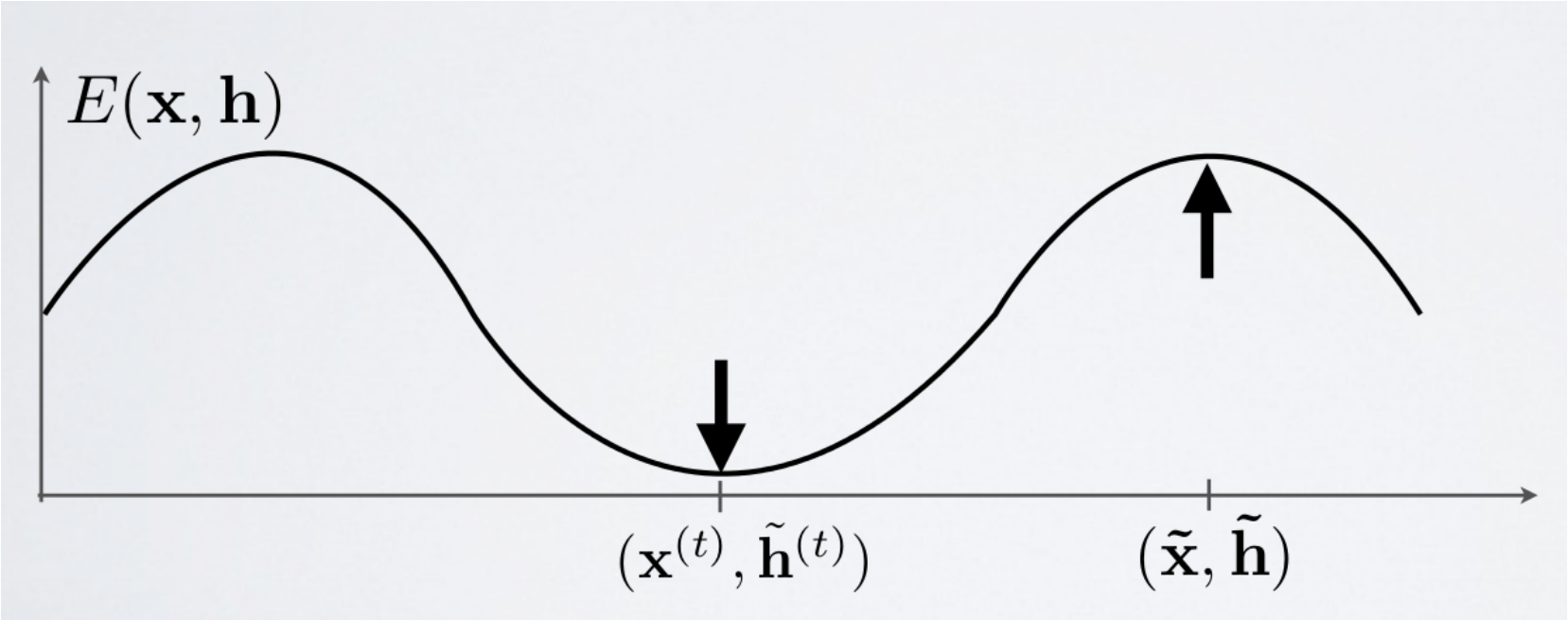
(H.Larochelle.)

Restricted Boltzmann Machines

Training of RBMs: **k-steps Contrastive Divergence** (CD-*k*).

Ingredients:

- Markov chain using Gibbs sampling.
- Gradient approximation (gradient descent).



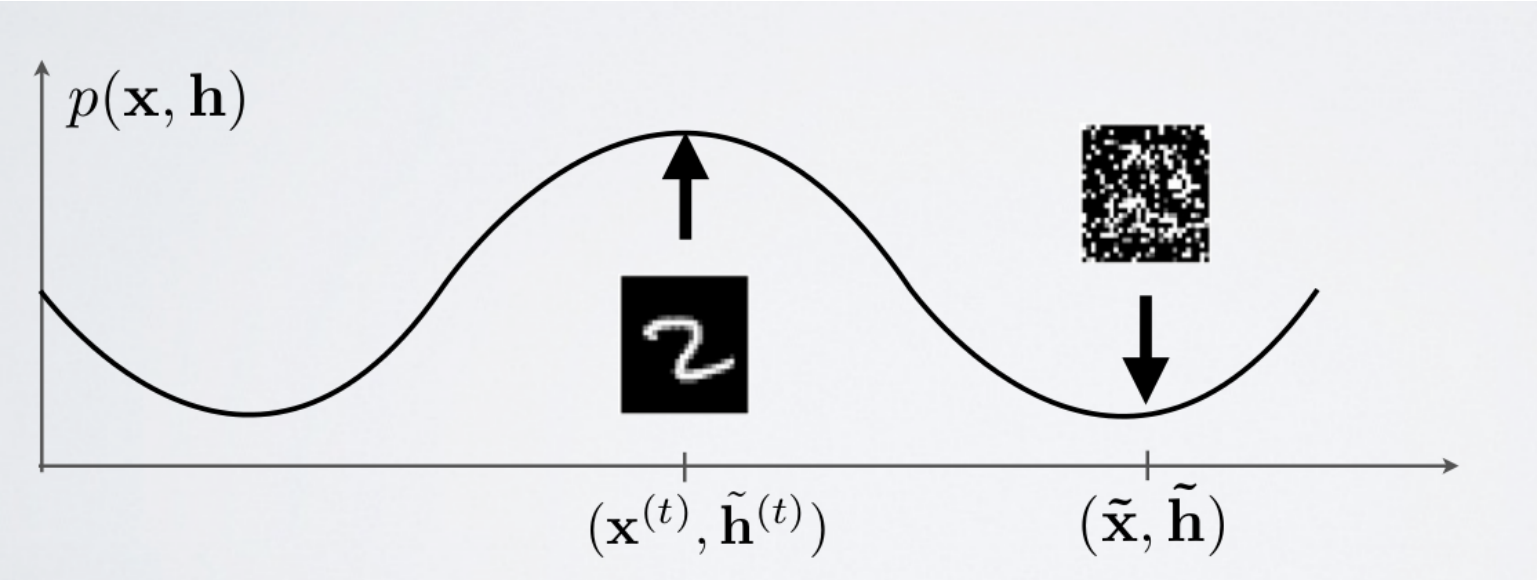
(H.Larochelle.)

Restricted Boltzmann Machines

Training of RBMs: **k-steps Contrastive Divergence** (CD-*k*).

Ingredients:

- Markov chain using Gibbs sampling.
- Gradient approximation (gradient descent).



(H.Larochelle.)

Restricted Boltzmann Machines

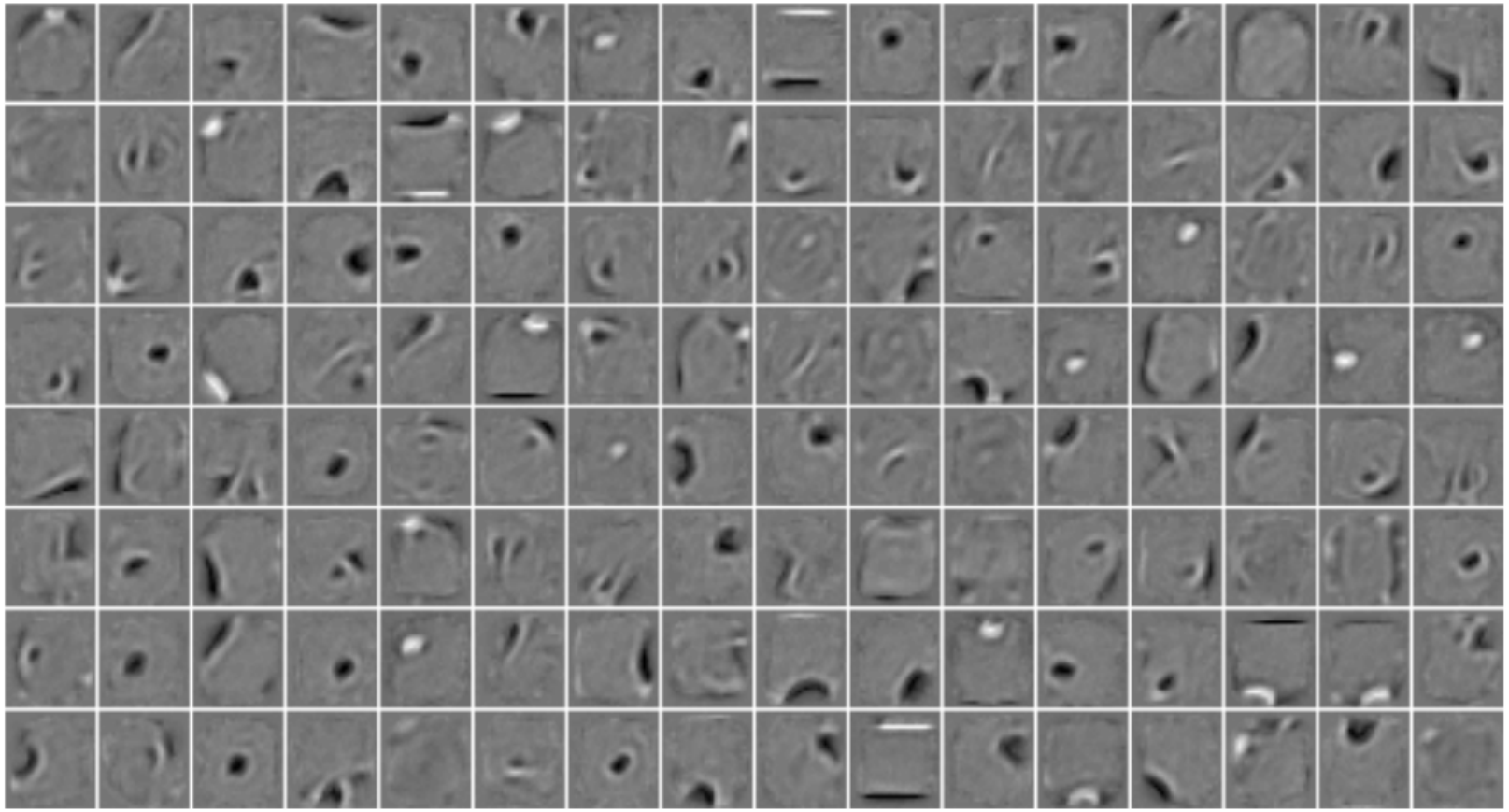


Figure 7: Learned weights of the hidden units of RBM over Mnist.

(H.Larochelle. 2009)

Example 1: Stacked RBMs

Dimensionality reduction (Hinton et al. 06).

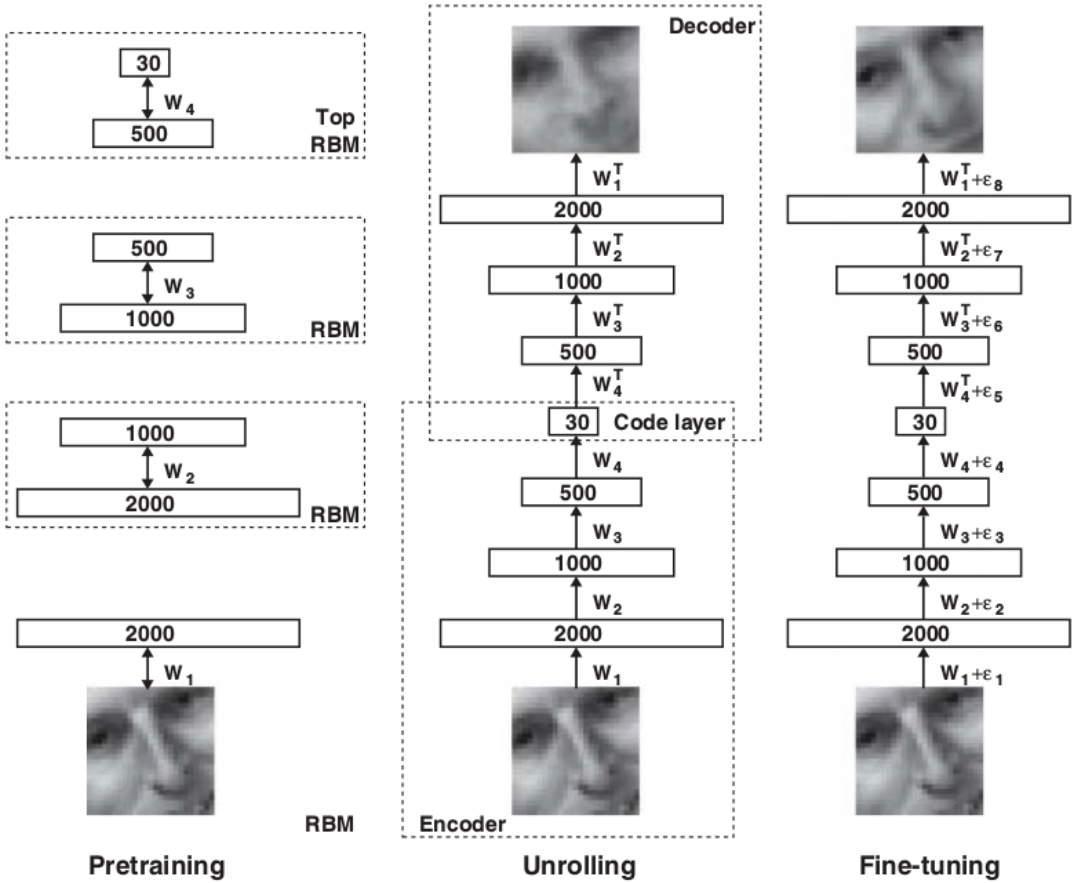


Figure 8: Stacked RBMs

Example 1: Stacked RBMs

Dimensionality reduction (Hinton et al. 06).

Figure 9: Top: origin. middle: reconstructed SRBMs. Bottom: reconstructed PCA.

Example 1: Stacked RBMs

Dimensionality reduction (Hinton et al. 06).

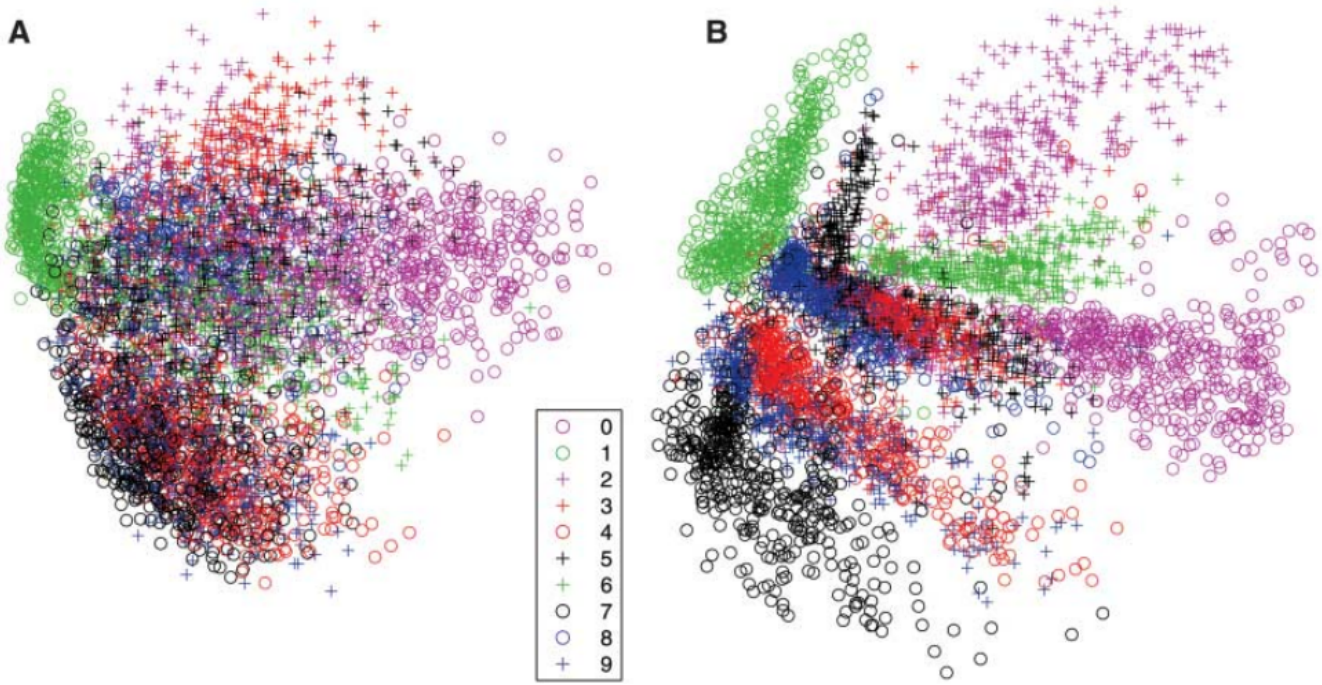


Figure 10: A: original training images (1st two dimensions of PCA). B: two dimensional codes of 784- 1000-500-250-2.

Example 2: Collaborative Filtering

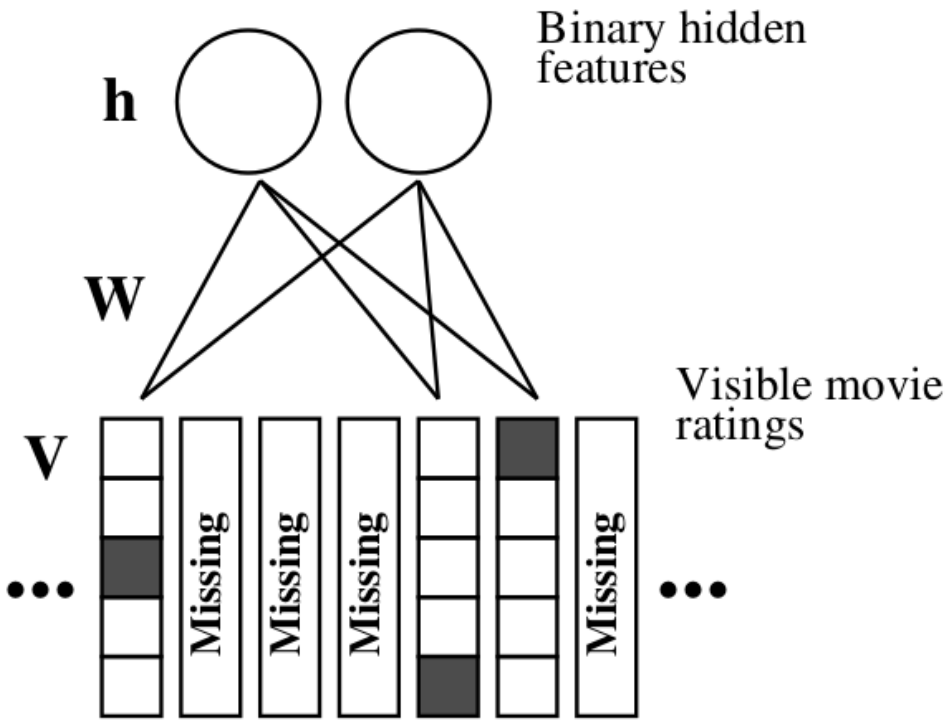
Netflix recommendation (Salakhutdinov et al. 07).
see CF (*.gif): `https://en.wikipedia.org/wiki/File:Collaborative_filtering.gif`

Example 2: Collaborative Filtering

Netflix recommendation (Salakhutdinov et al. 07).
see CF (*.gif): `https://en.wikipedia.org/wiki/File:Collaborative_filtering.gif`

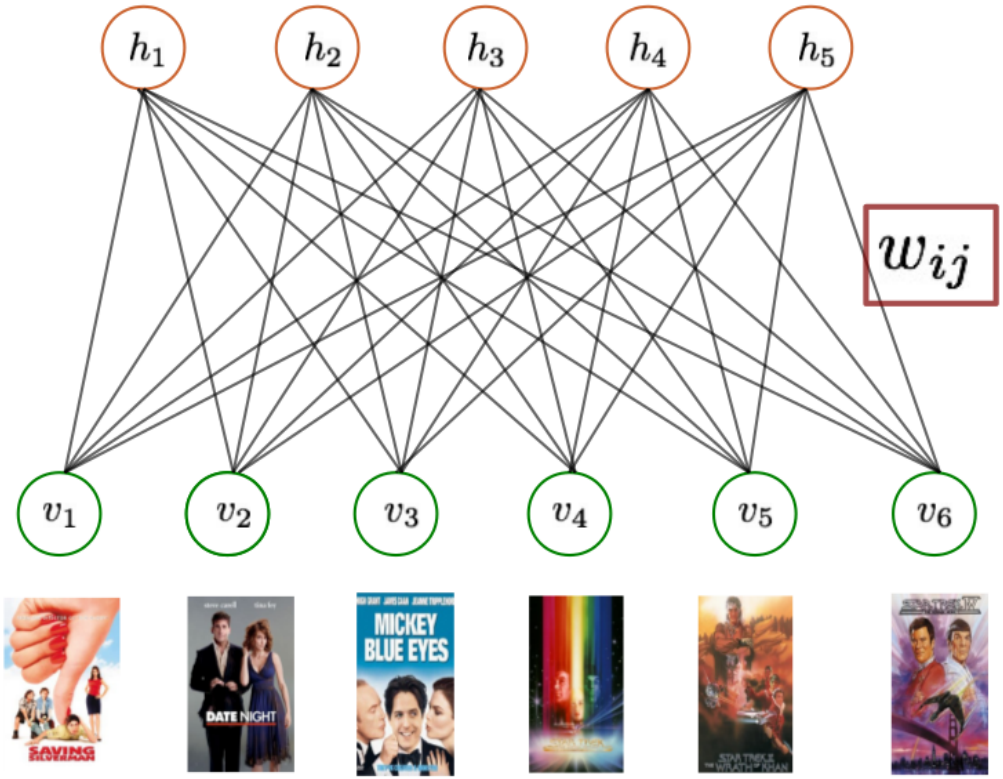
Example 2: Collaborative Filtering

Netflix recommendation (Salakhutdinov et al. 07).
see CF (*.gif): `https://en.wikipedia.org/wiki/File:Collaborative_filtering.gif`



Example 2: Collaborative Filtering

Netflix recommendation (Salakhutdinov et al. 07).
see CF (*.gif): `https://en.wikipedia.org/wiki/File:Collaborative_filtering.gif`



Conclusion

- The way data is represented is important.
- Deep learning offers task-dependant representations.
- Exploit the large amount of unlabeled data: unsupervised learning:
  - Auto-encoders.
  - Restricted Boltzmann Machines.

Questions

Thank you for your attention,

Questions?

soufiane.belharbi@insa-rouen.fr
https://sbelharbi.github.io