Autoencoders: from unsupervised to generative deep learning

Generative and Deep Learning (GDL)

Davide Bacciu (davide.bacciu@unipi.it)

UNIVERSITÀ DI PISA

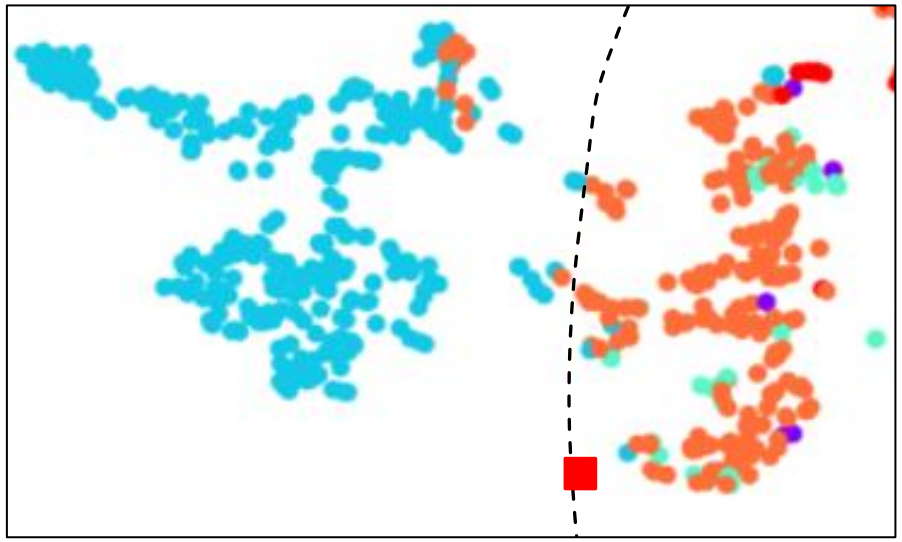
Lecture Outline

- Introduction to the Generative DL module
  - Motivations and taxonomy
- Autoencoders
  - The unsupervised way to deep learning
  - Representation learning and the manifold hypothesis
- Notable autoencoders
  - Neural autoencoders
  - Probabilistic autoencoders
- Autoencoder applications

Generative Deep Learning Module

Why generative?

- Focusing too much on discrimination rather than on characterizing data can cause issues
  - Reduced interpretability
  - Adversarial examples



- Generative models (try to) characterize data distribution
  - Understand the data ⟹ Understand the world
  - Understand data factors of variation ⟹ Learn to steer them
  - Understand normality ⟹ Detect anomalies

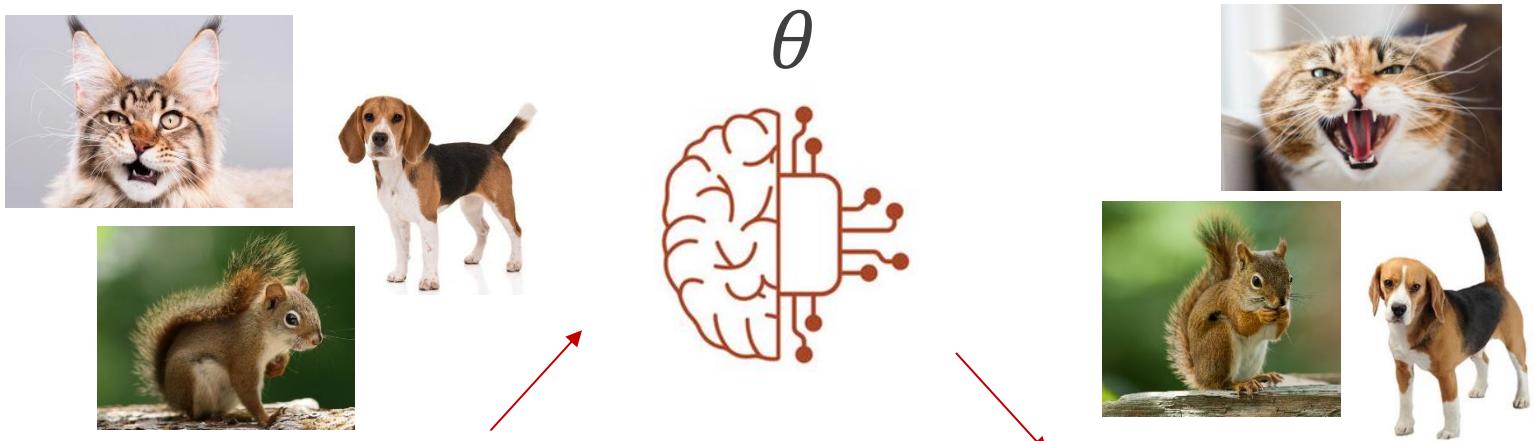
Generative learning is (close to) unsupervised learning

- Task-labelled data is costly and difficult to obtain
- Leand task independent representations at scale instead
  - Learning to reconstruct (**self-supervised**)
  - Learning from similarities (**contrastive**)
- Then, fine-tune with supervised information on fewer samples

The foundation model paradigm

Approaching the problem from a DL perspective

Given training data, learn a (deep) neural network that can generate new samples from (an approximation of) the data distribution



Training data $\sim P(x)$

Generated data $\sim P_\theta(x)$

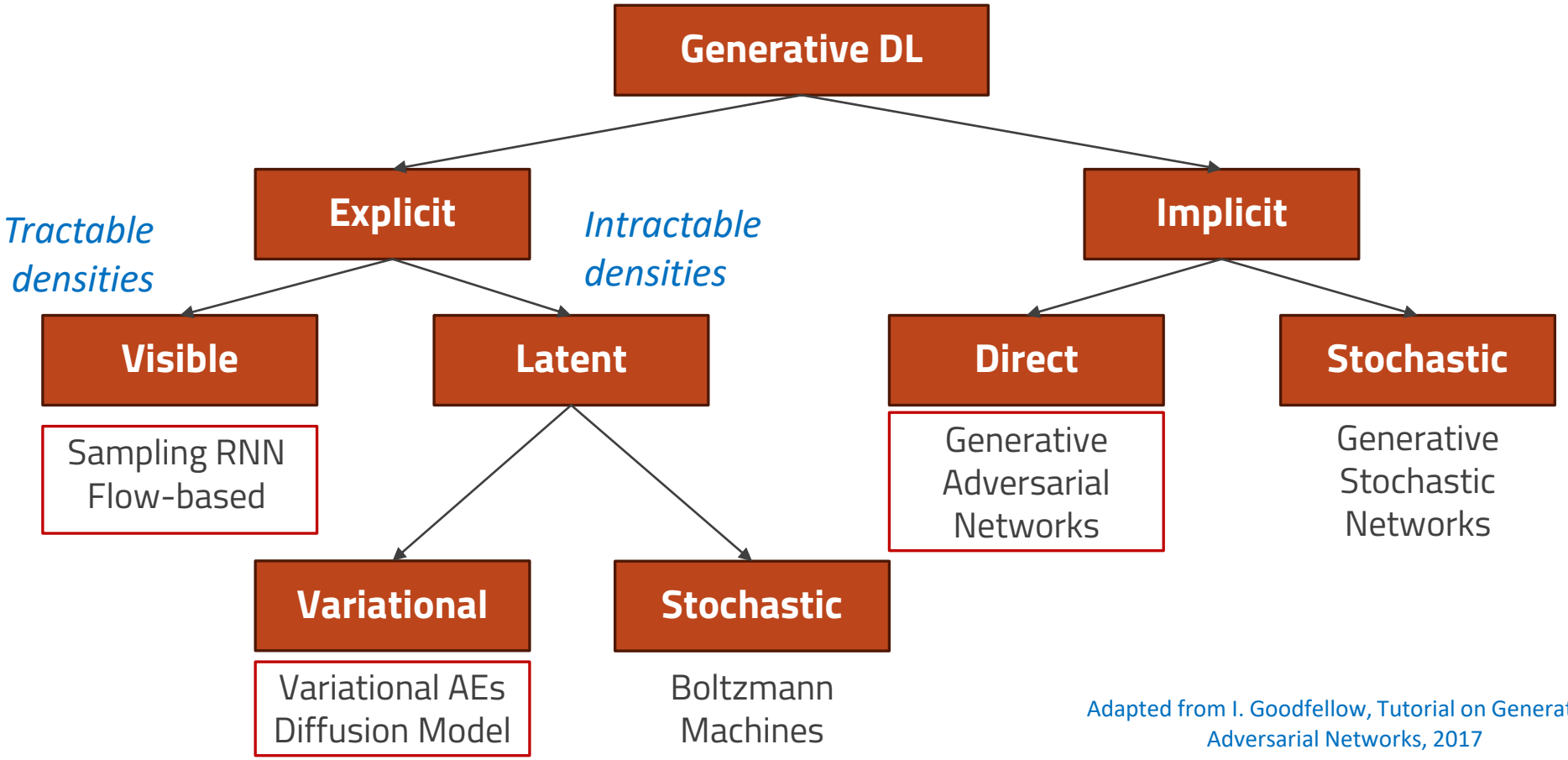
Approaching the problem from a DL perspective

Given training data, learn a (deep) neural network that can generate new samples from (an approximation of) the data distribution

Two broad families of approaches

- Explicit $\Longrightarrow$ Learn a model density $P_\theta(x)$
- Implicit $\Longrightarrow$ Learn a process that samples data from $P_\theta(x) \approx P(x)$

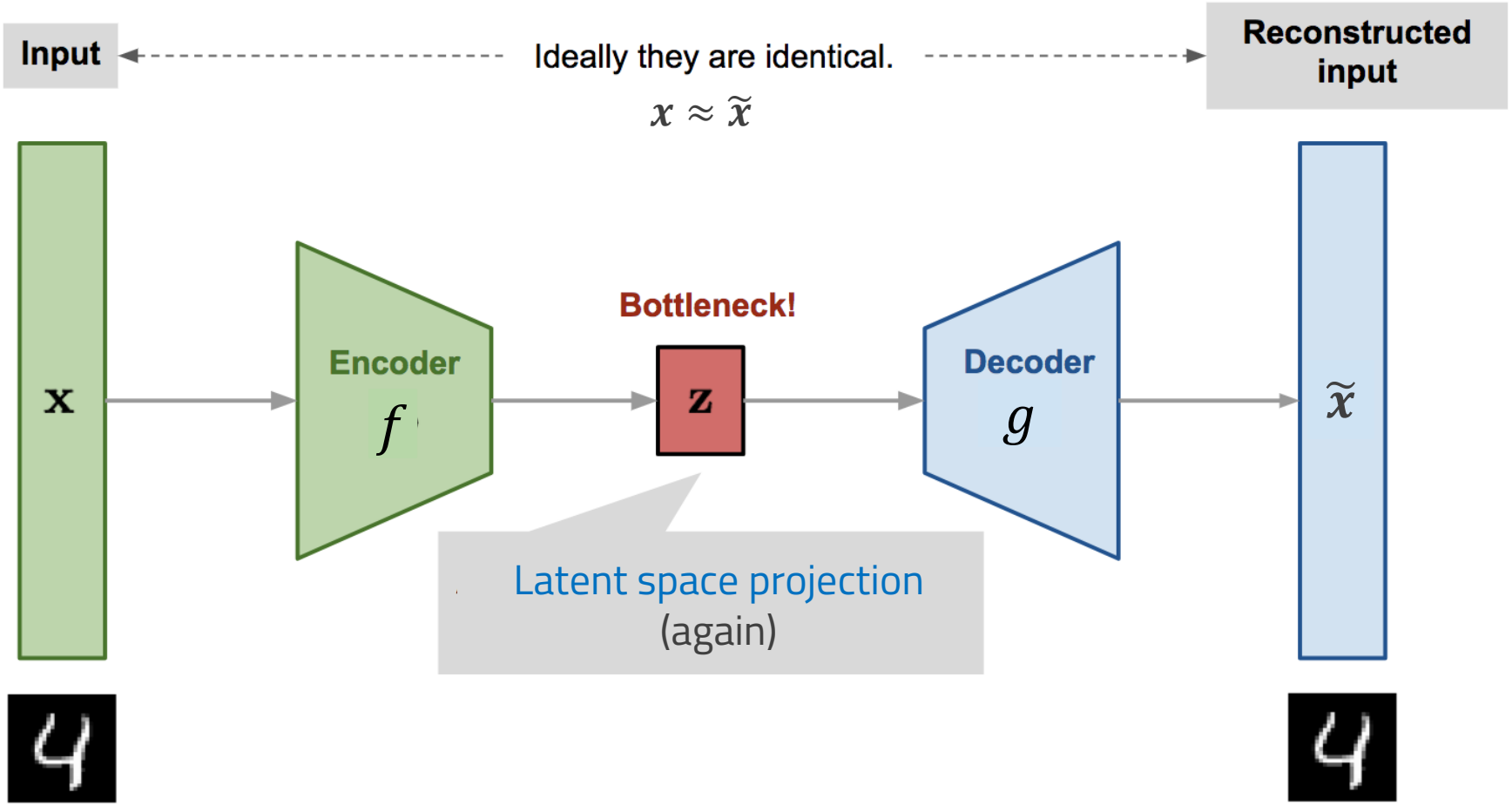
A Taxonomy



Adapted from I. Goodfellow, Tutorial on Generative Adversarial Networks, 2017

Autoencoders

Autoencoder (AE)



The autoencoder task

- Train a model to reconstruct its input
- Passing through some form of information bottleneck
  - $K \ll D$, or?
  - $\boldsymbol{z}$ sparsely active
- Unsupervised task learned by reconstruction error minimization

$$L(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = L(\boldsymbol{x}, g(f(\boldsymbol{x}))\,)$$

Neural Autoencoders

Generally, we would like to train nonlinear AEs, with possibly $K > D$, that do not learn trivial identity

- Regularized autoencoders
  - Sparse AE
  - Denoising AE
  - Contractive AE
- Autoencoders with dropout layers

Sparse Autoencoder

Add a term to the cost function to penalize $\boldsymbol{z}$ (want the number of active units to be small)

$$J_{SAE}(\theta) = \sum_{\boldsymbol{x}\in D}(L(\boldsymbol{x}, \tilde{\boldsymbol{x}}) + \lambda\Omega(\boldsymbol{z}(\boldsymbol{x})))$$

Typically

$$\Omega(\boldsymbol{z}(\boldsymbol{x})) = \Omega(f(\boldsymbol{x})) = \sum_j \left|z_j(\boldsymbol{x}))\right|$$

Probabilistic Interpretation (Oh No, Again!)

Training with regularization is (akin to) MAP inference

$$\max \log P(\boldsymbol{z}, \boldsymbol{x}) = \max \left(\log P(\boldsymbol{x}|\boldsymbol{z}) + \log P(\boldsymbol{z})\right)$$

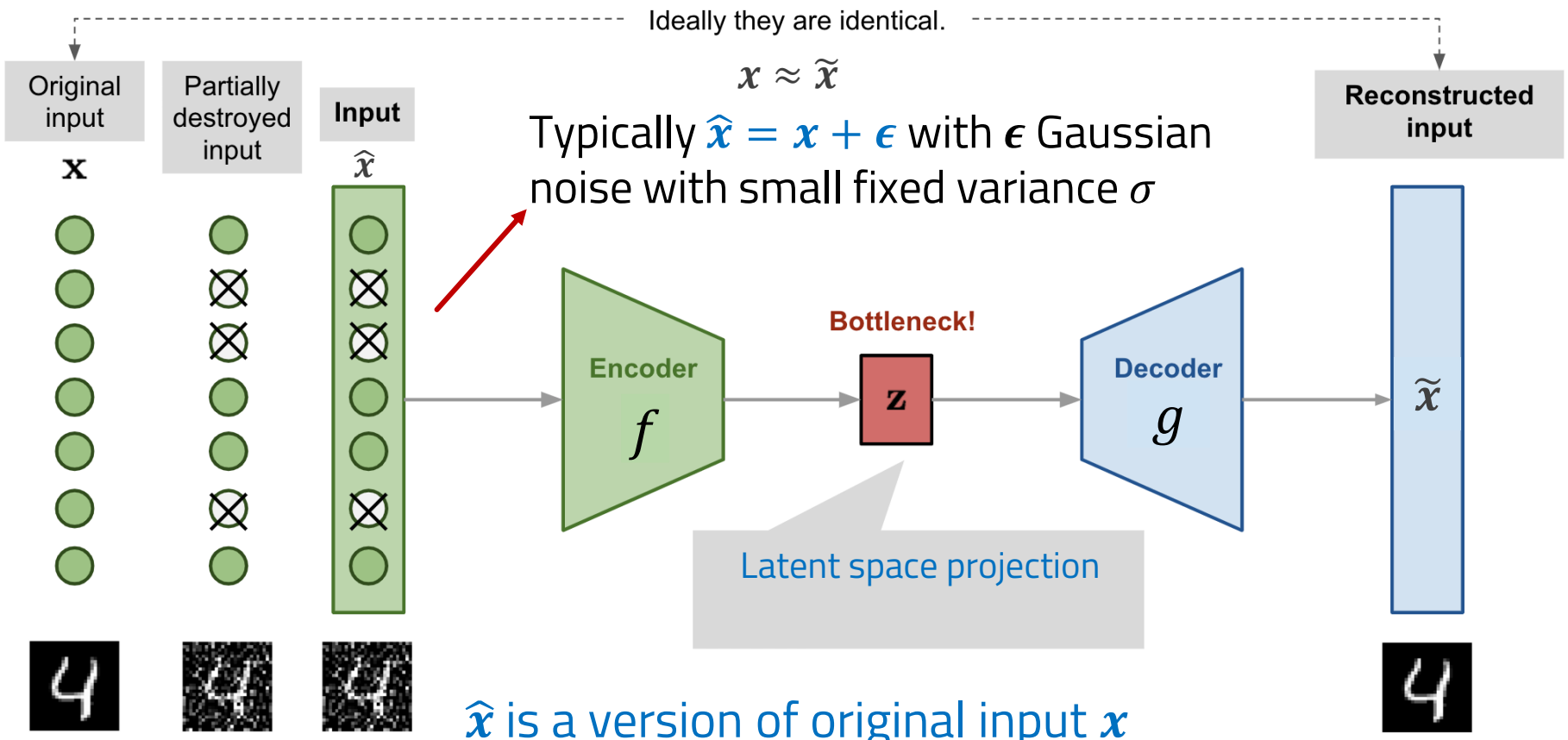
Likelihood

Prior (not really!)

$$P(\boldsymbol{z}) = \frac{\lambda}{2} \exp\left(-\frac{\lambda}{2} |\boldsymbol{z}|_1\right) \Rightarrow \Omega(\boldsymbol{z}) = \lambda ||\boldsymbol{z}||_1$$

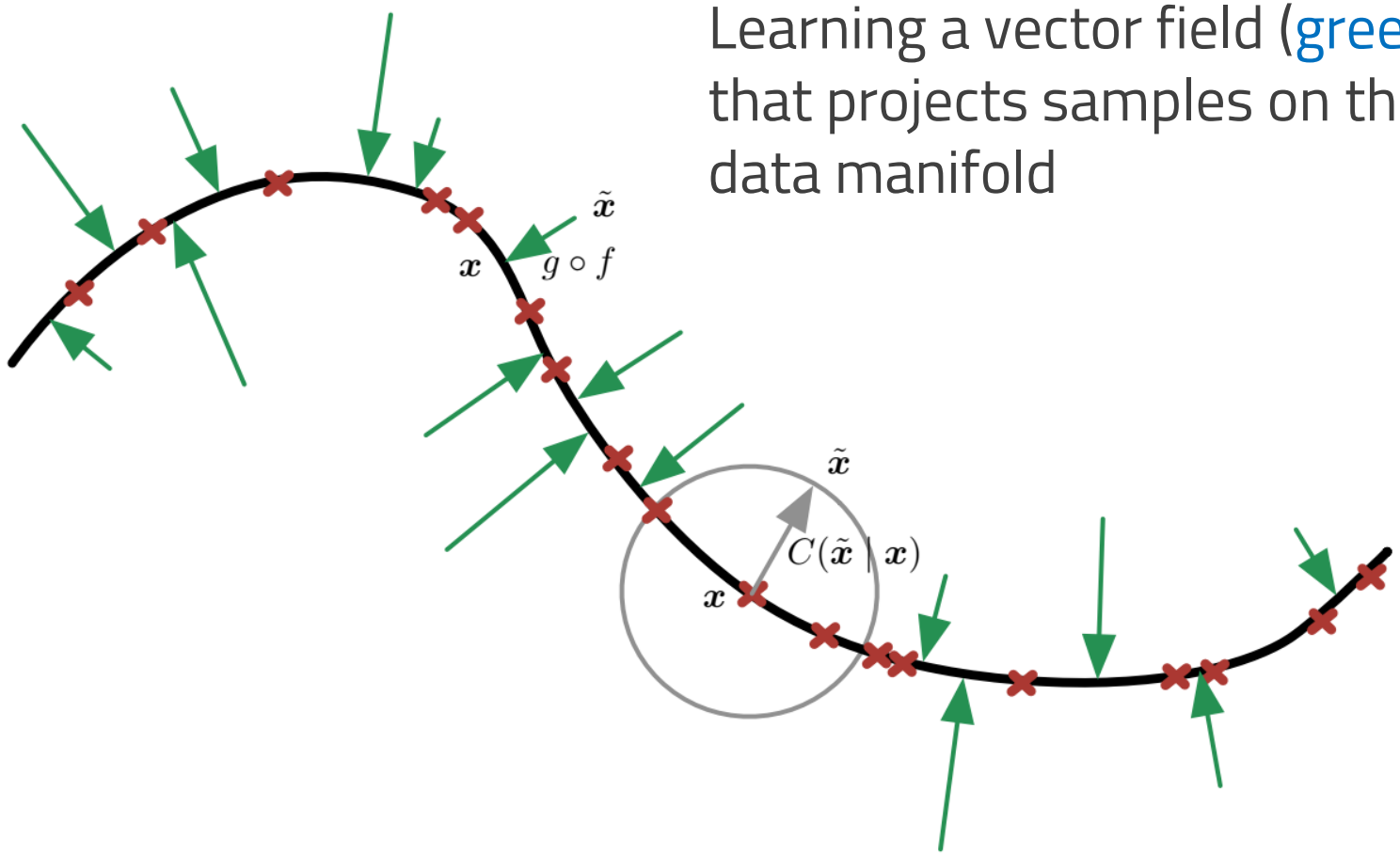
Laplace distribution

Denoising Autoencoder (DAE)



$\hat{x}$ is a version of original input x corrupted by some noise process $C(\hat{x}|x)$

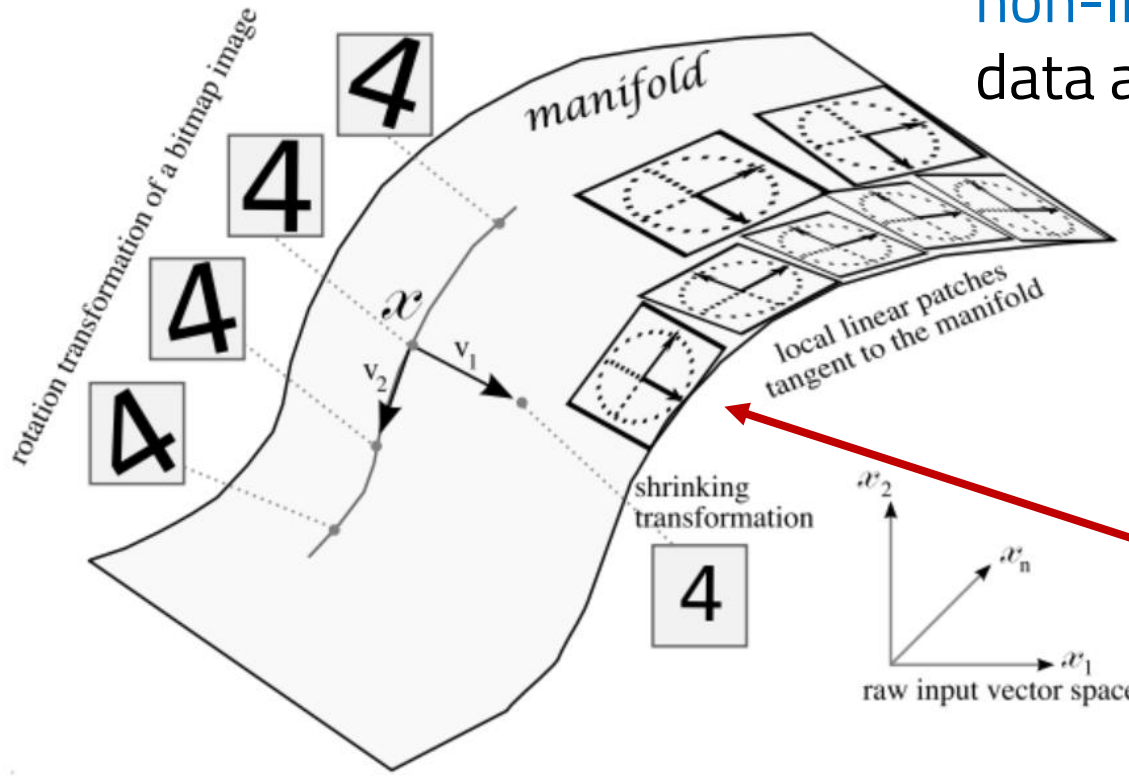
DAE as Manifold Learning



Learning a vector field (green arrows) that projects samples on the original data manifold

The Manifold Assumption

Yoshua Bengio, Learning Deep Architectures for AI, Foundations and Trends in Machine Learning, 2009.



Assume data lies on a lower dimensional non-linear manifold since variables in data are typically dependent

Regularized AE can afford to represent only variations that are needed to reconstruct training examples

AE mapping is sensitive only to changes in manifold direction

Contractive Autoencoder

- DAE's intuition is that learned representations should be robust to partial destruction of the input
- Makes sense to generalize this to learning encoding functions that are robust to infinitesimal variations in the input
- Formally: penalize encoding function for input sensitivity

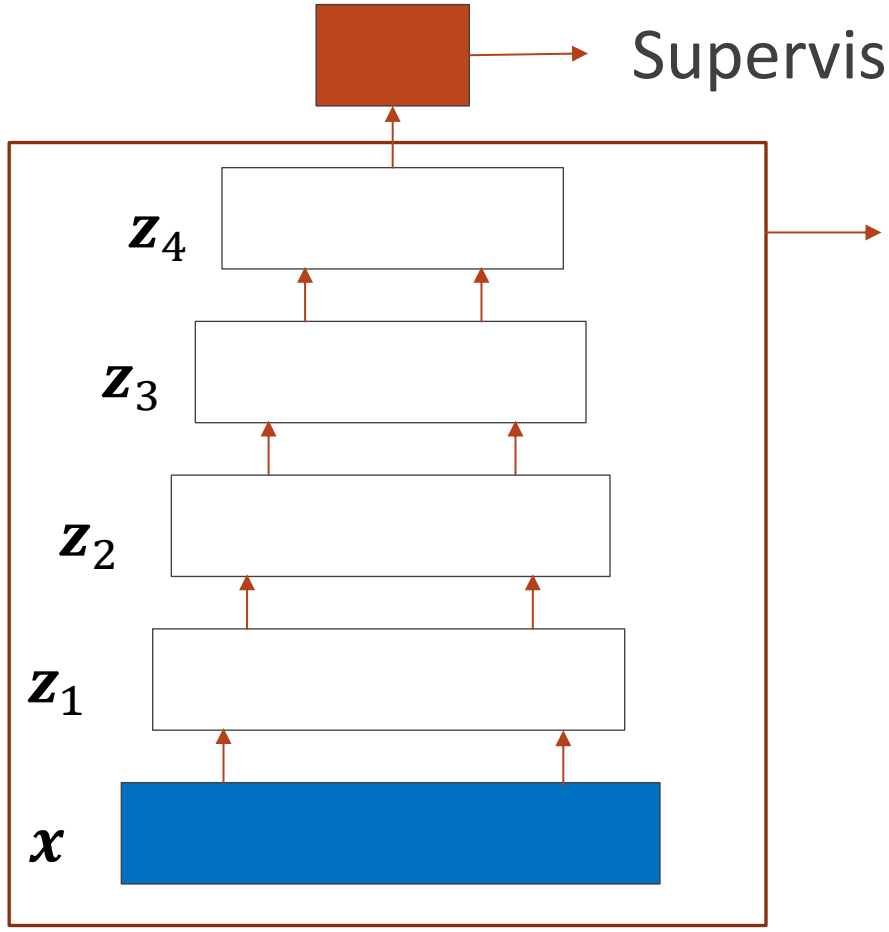
$$J_{CAE}(\theta) = \sum_{\boldsymbol{x} \in S}(L(\boldsymbol{x}, \tilde{\boldsymbol{x}}) + \lambda \Omega(\boldsymbol{z}))$$

$$\Omega(\boldsymbol{z}) = \Omega(f(\boldsymbol{x})) = \left\| \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} \right\|_F$$

- You can as well penalize on higher order derivatives

Deep Autoencoders

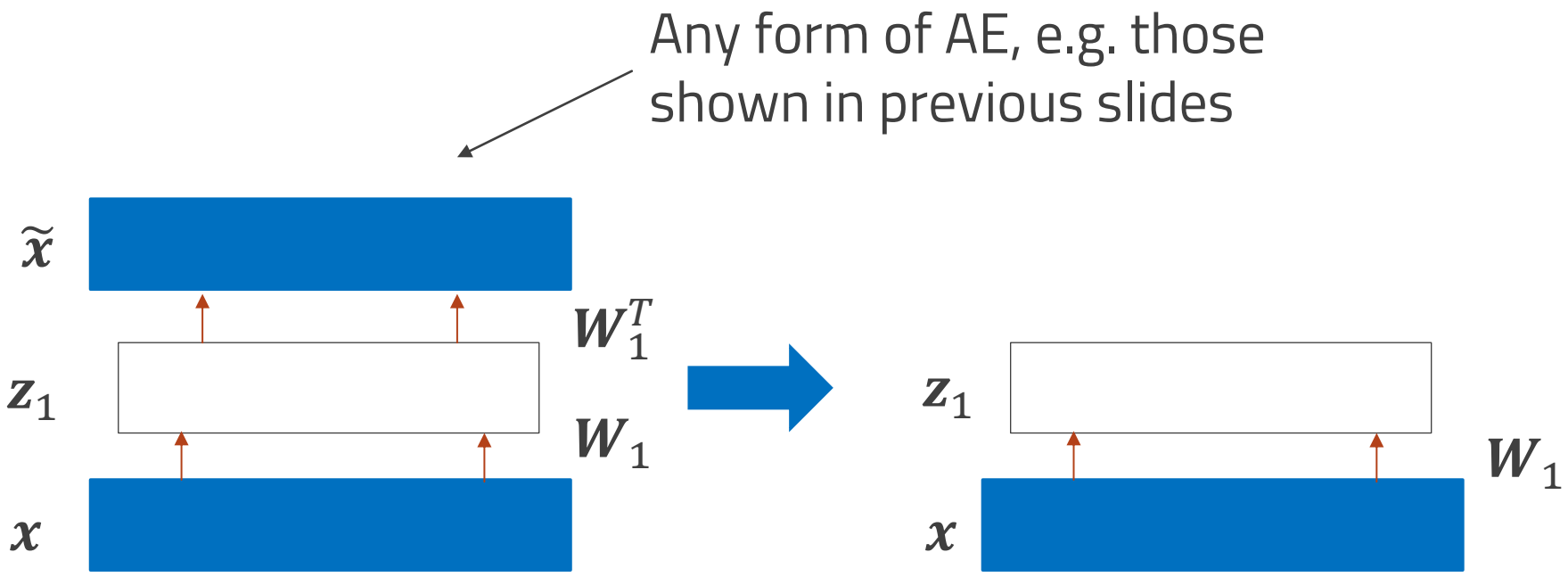
Deep Autoencoder (AE)



- Unsupervised training
- Hierarchical autoencoder
- Extracts a representation of data that facilitates
  - Data visualization, exploration, indexing,…
  - Realization of a supervised task

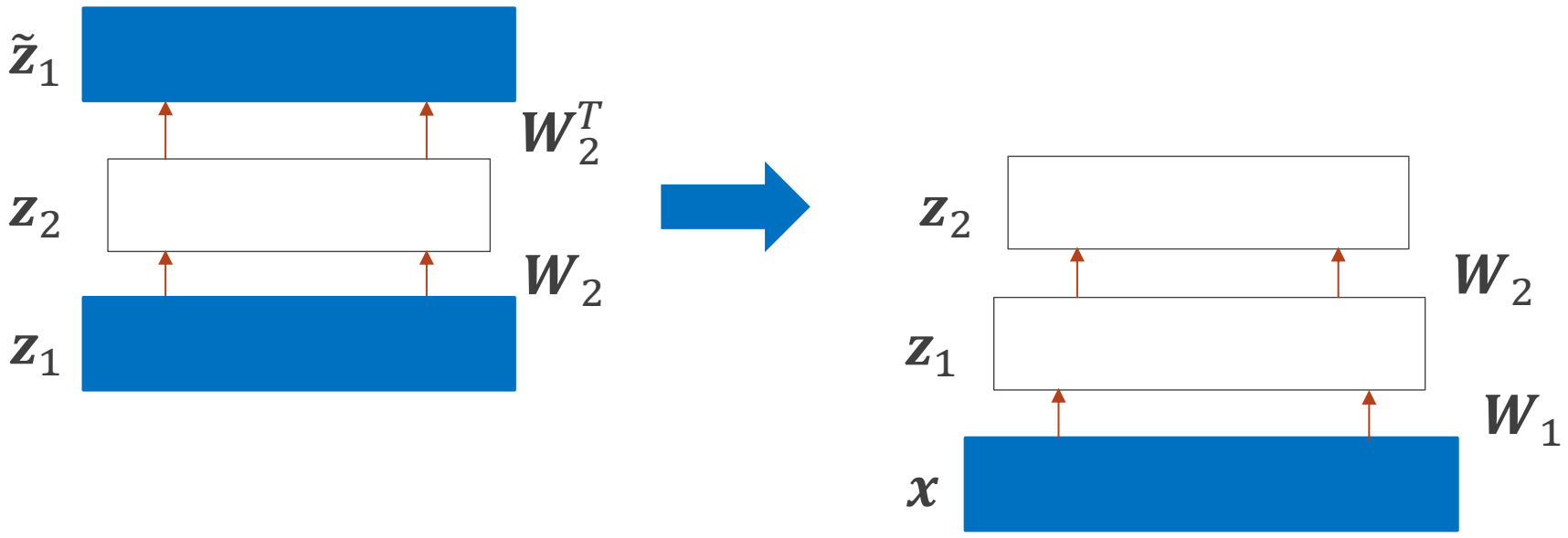
Unsupervised Layerwise Pretraining

Incremental unsupervised construction of the Deep AE



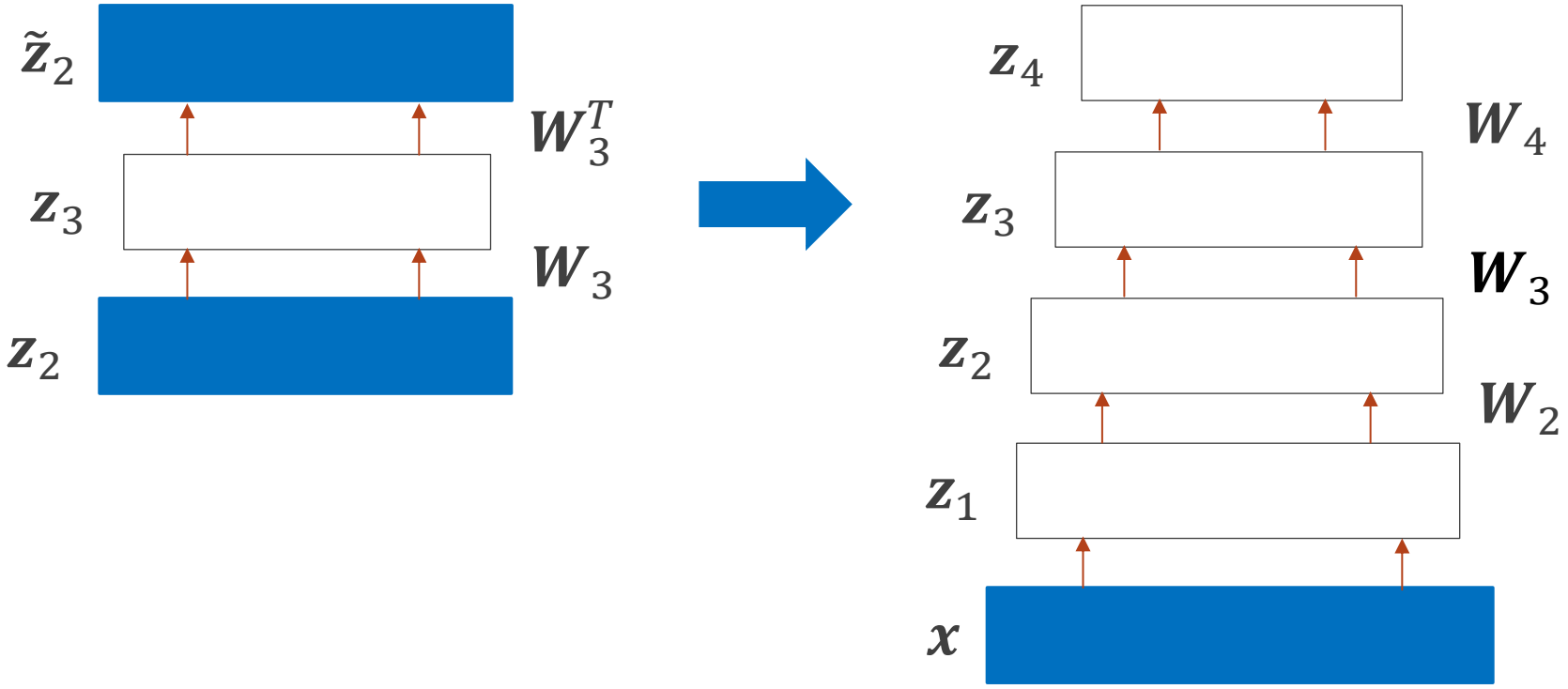
Unsupervised Layerwise Pretraining

Incremental unsupervised construction of the Deep AE



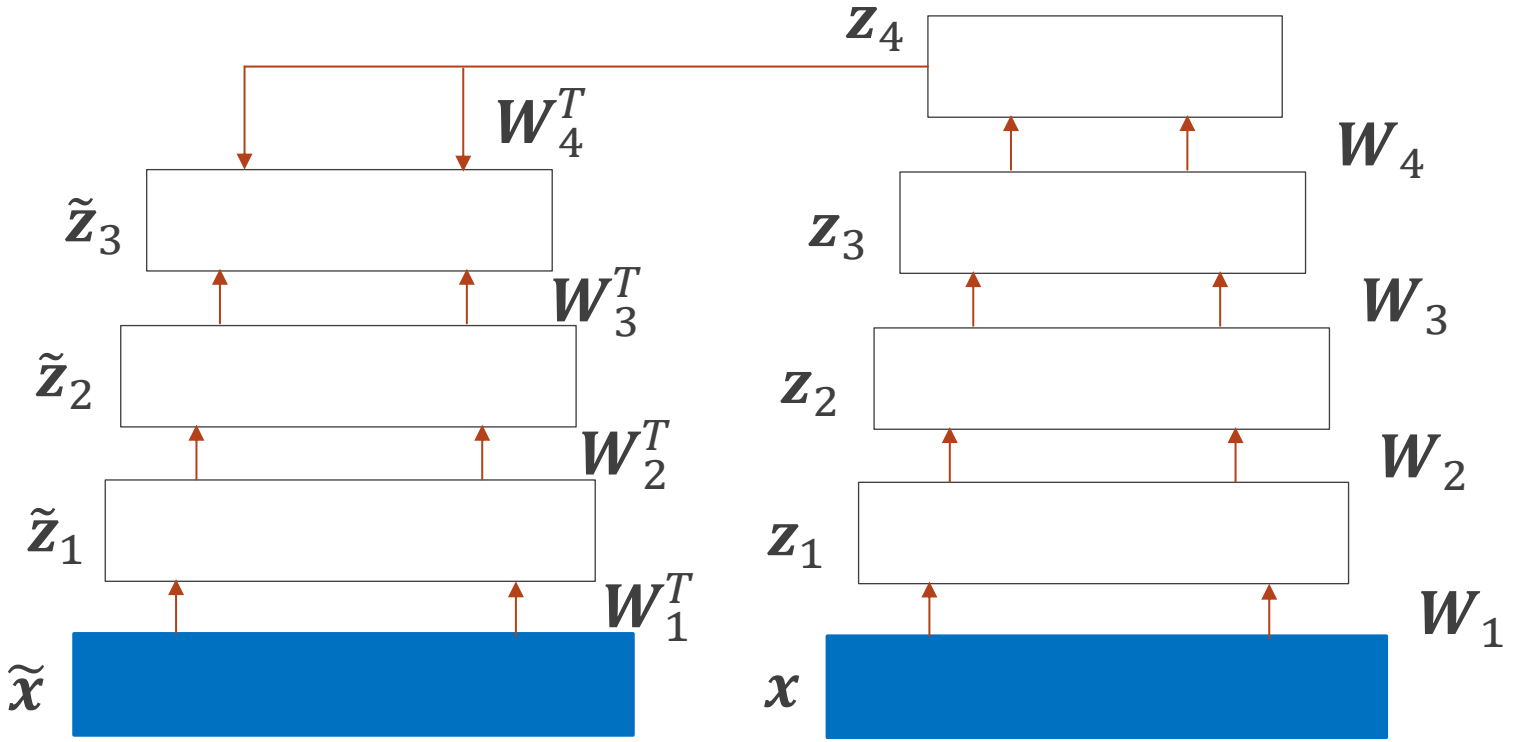
Unsupervised Layerwise Pretraining

Incremental unsupervised construction of the Deep AE



Optional Fine Tuning

Fine tune the whole autoencoder to optimize input reconstruction (also using backpropagation)

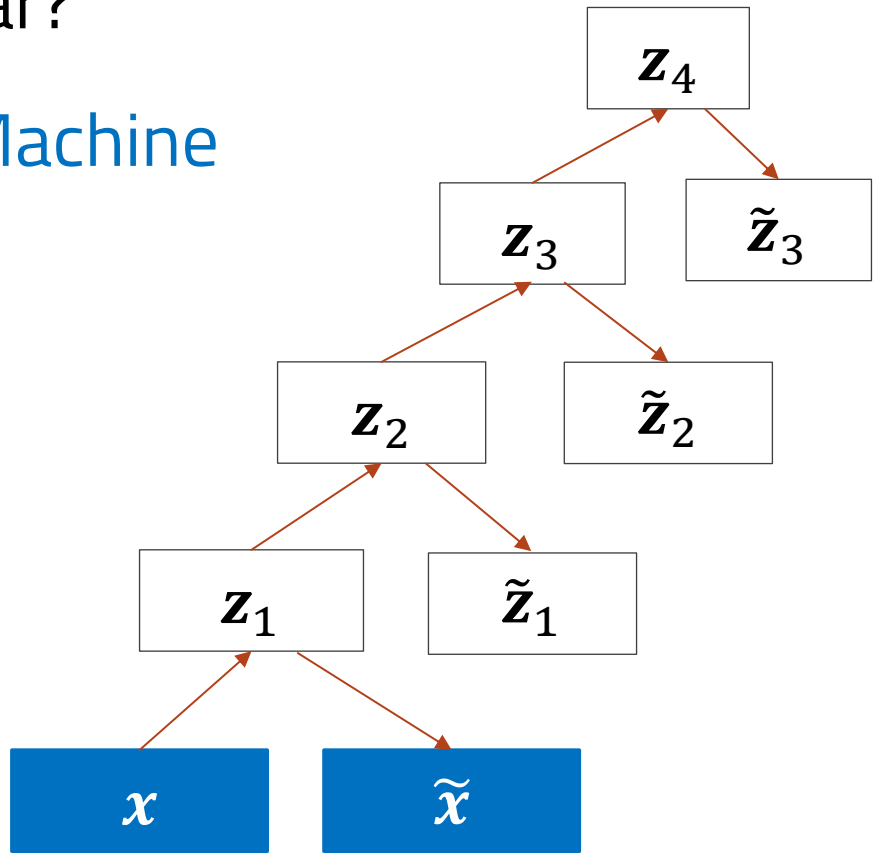


Rearranging the Graphics

Does it look like something familiar?

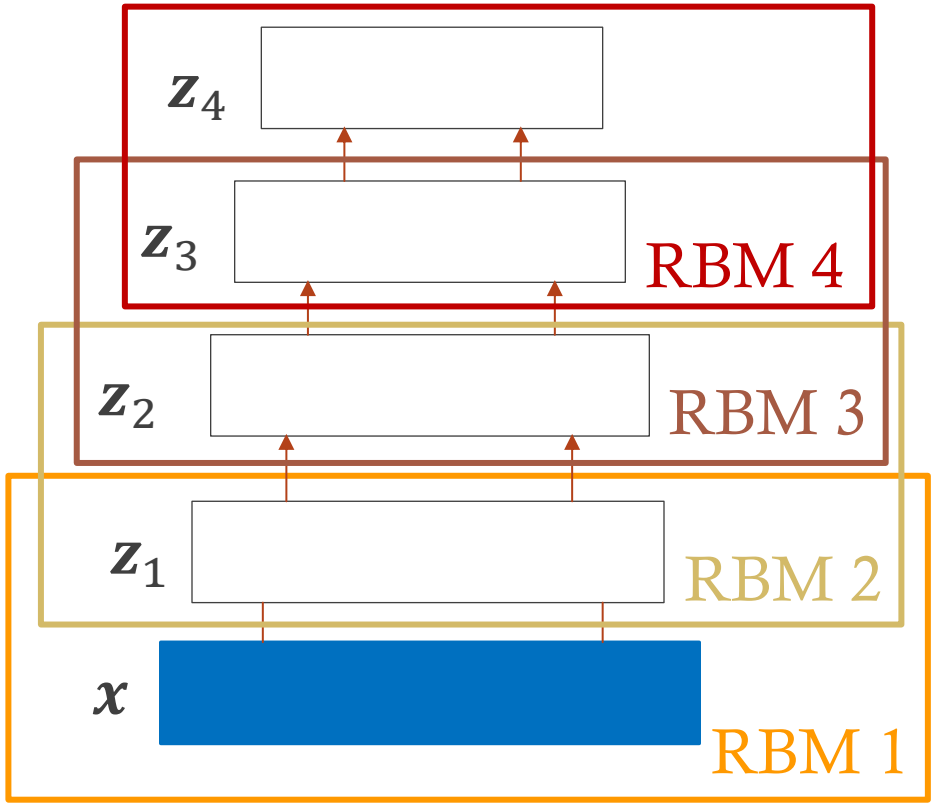
A layered Restricted Boltzmann Machine

Can use RBM to perform layerwise pretraining and learn the matrices $\boldsymbol{W}_i$



Deep Belief Network (DBN)

A stack of pairwise RBM



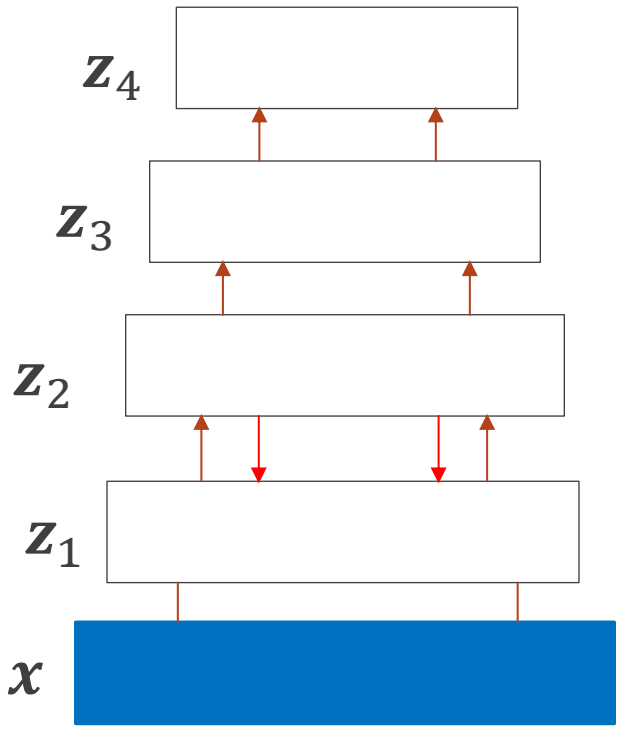
IMPORTANT NOTE

A DBM is a deep autoencoder but it is NOT a deep RBM

It is (mostly) directed!

Deep Boltzmann Machine (DBM)

How do we get this?



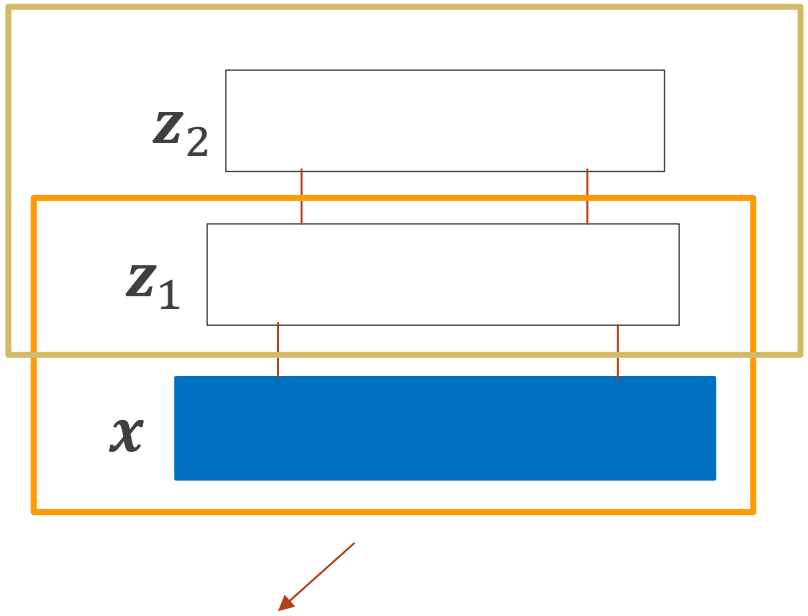
Training requires some attention because of the recurrent interactions from higher layers to the bottom

$$P\left(z_j^1 \middle| \boldsymbol{x}, \boldsymbol{z}^2\right) = \sigma\left(\sum_i W_{ij}^1 \, x_i + \sum_m W_{jm}^2 \, z_m^2\right)$$

$$P(x_i|\boldsymbol{z}^1) = \sigma\left(\sum_j W_{ij}^1 \, z_j^1\right)$$

Pretraining DBM

How do we get this?



1) (Pre)training the first layer entails fitting this model

2) (Pre)training the second layer changes z^1 prior by

$$P(z^1|W^2) = \sum_{z^2} P(z^1, z^2|W^2)$$

When putting things together, we need to average between the two

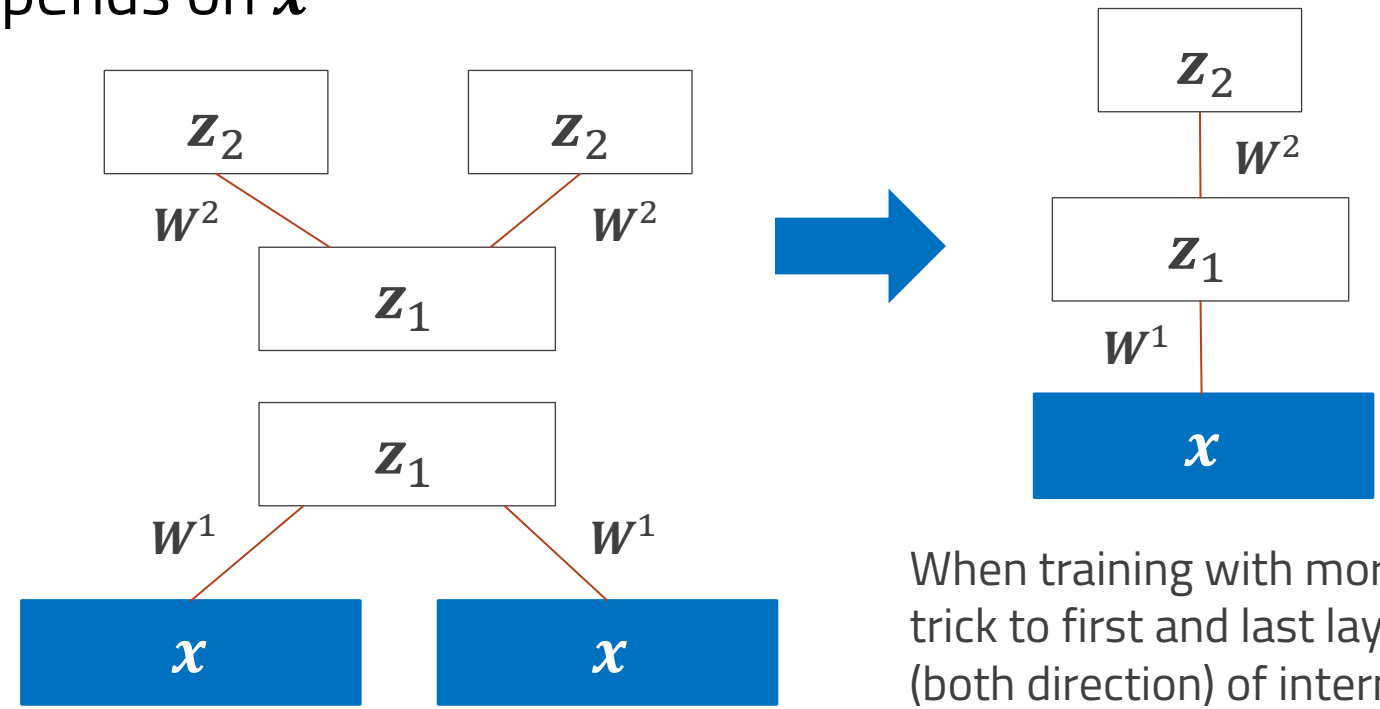
$$P(z^1|W^1) = \sum_{x} P(z^1, x|W^1)$$

$$P(x|\theta) = \sum_{z^1} P(z^1|W^1)P(x|z^1, W^1)$$

Pretraining DBM - Trick

Averaging the two models of $\mathbf{z}^1$ can be approximated by taking half contribution from $\boldsymbol{W}^1$ and half from $\boldsymbol{W}^2$

- Using full $\boldsymbol{W}^1$ and $\boldsymbol{W}^2$ would double count $\boldsymbol{x}$ contribution as $\mathbf{z}^2$ depends on $\boldsymbol{x}$



When training with more than two RBMs apply trick to first and last layers and halve weights (both direction) of intermediate RBM

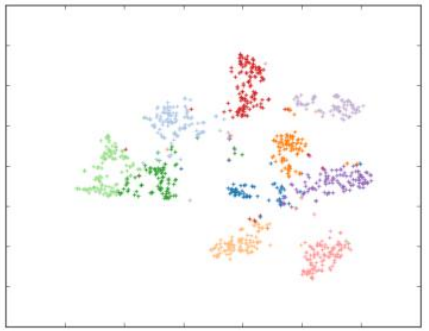
Autoencoders in use

Software

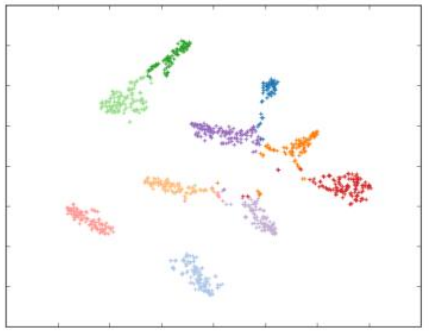
- All deep learning frameworks offer facilities to build deep AE and DBN; DBM implementations exist for all major deep learning libraries
- A variety of deep AE in Keras and their counterpart in in Pytorch
- Deep Boltzmann machine implementation (Tensorflow-based) with image processing application, pre-trained networks and notebooks
- Deepnet – A Toronto-based implementation of deep autoencoders (neural and generative)

AE Applications - Visualization

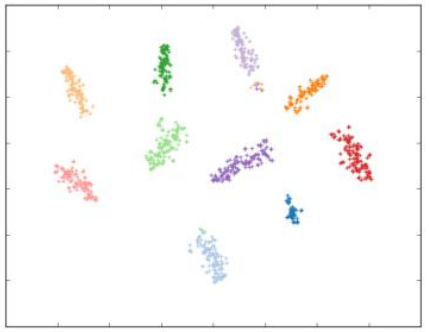
Visualizing complex data in learned latent space



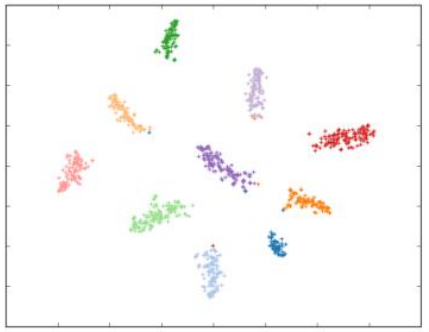
(a) Epoch 0



(b) Epoch 3

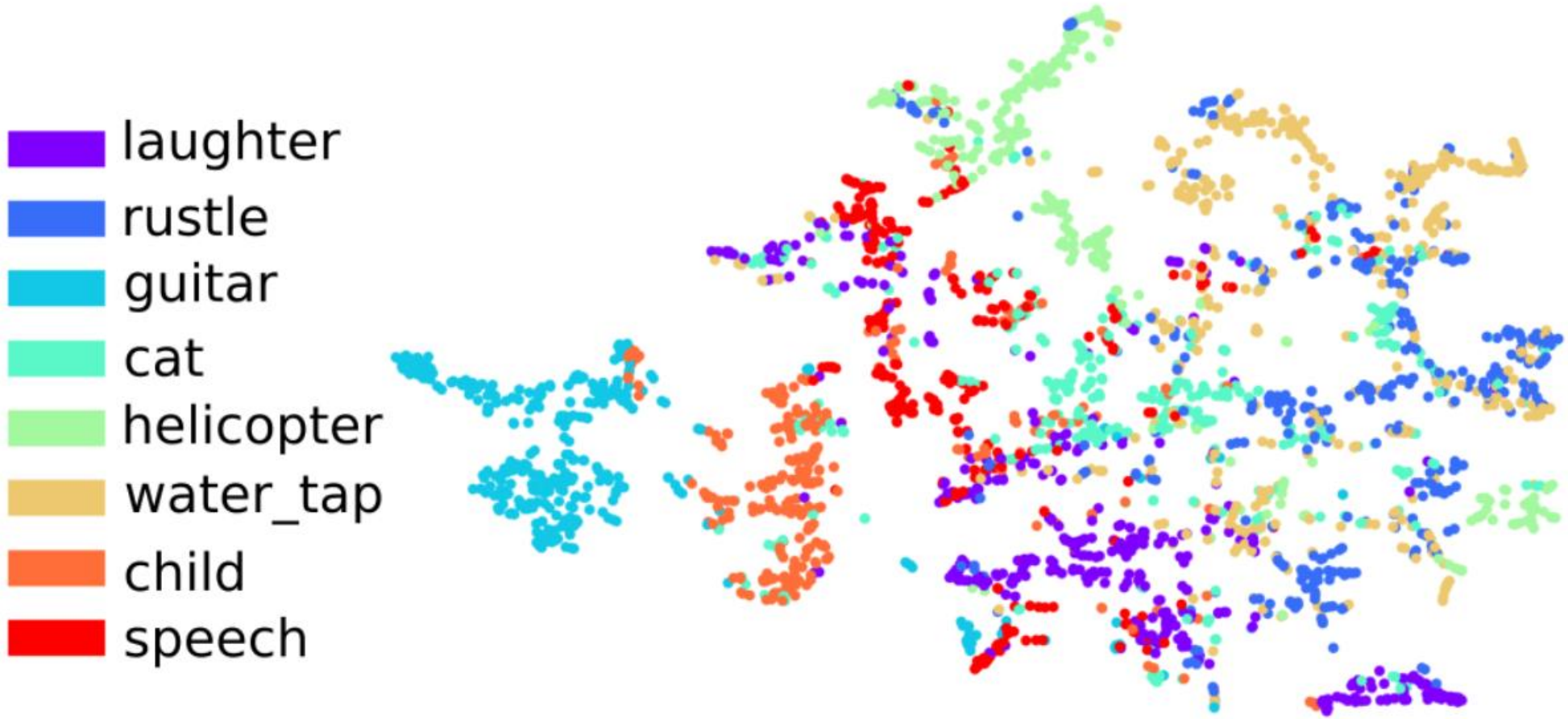


(d) Epoch 9

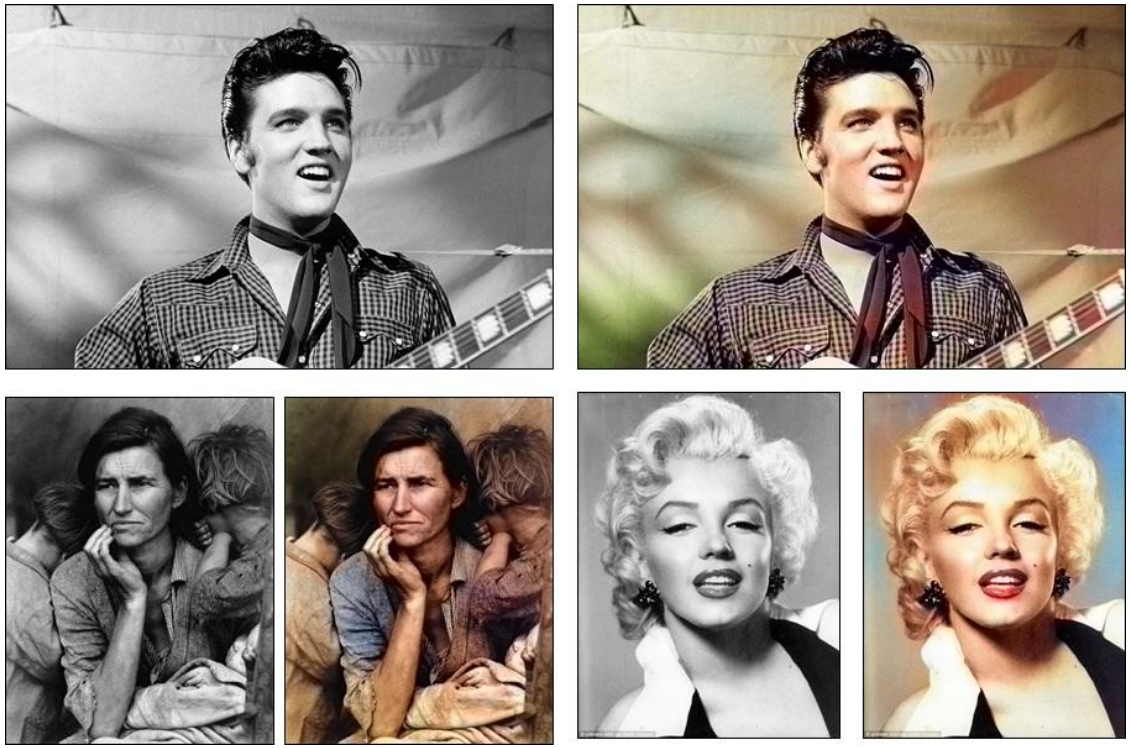


(e) Epoch 12

Visualizing Sound

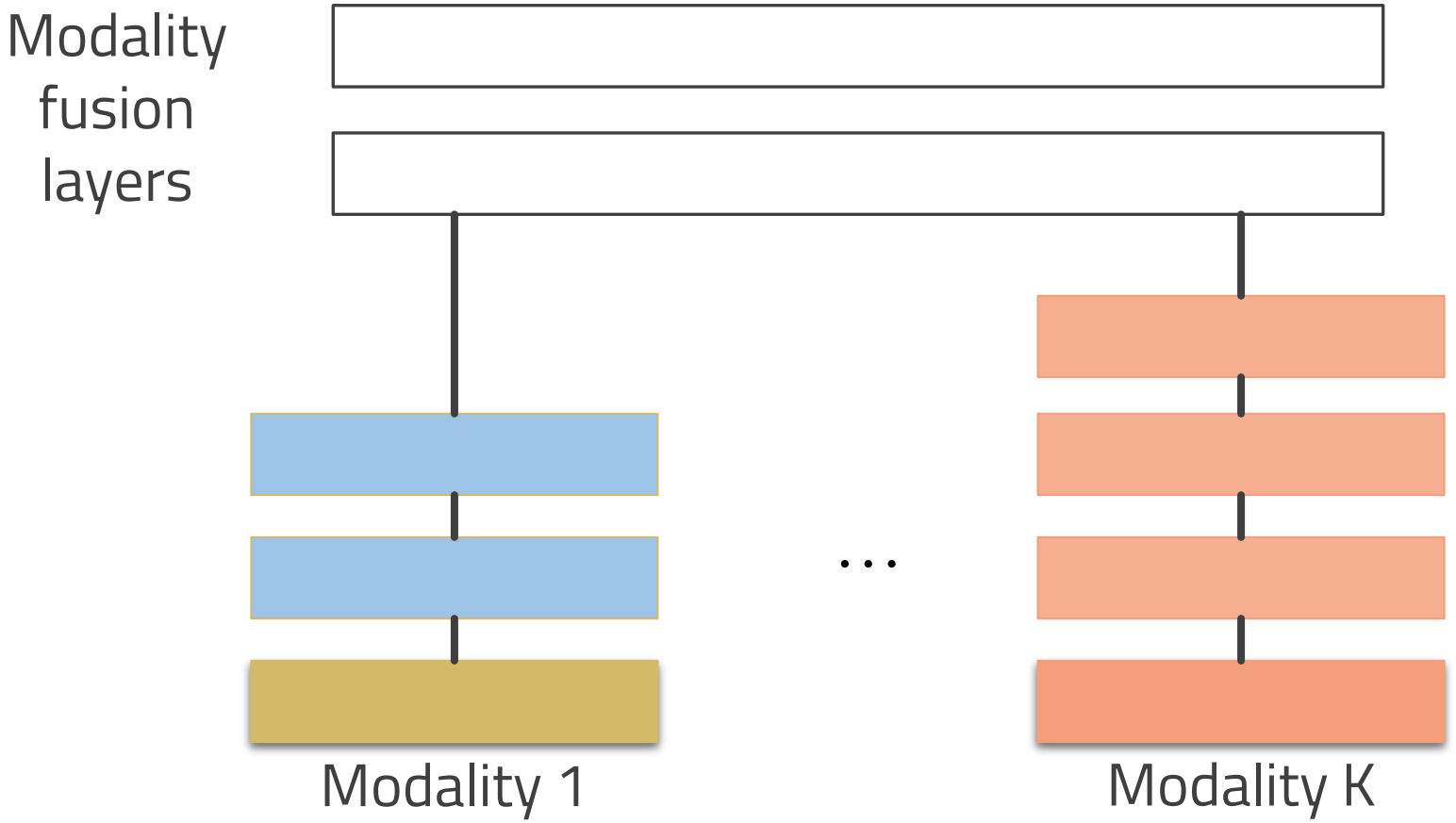


AE Applications – Image Restoration/Colorization



Apply autoencoder construction with advanced building blocks (e.g. CNN layers)

Multimodal DBM



N. Srivastava, R. Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, JMLR 2014

Multimodal DBM – Image and Text

$P(txt|img)$

Image	Given Tags	Generated Tags	Input Tags	Nearest neighbors to generated image features	
	pentax, k10d, kangarooisland, southaustralia, sa, 300mm, australia, australiansealion	beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves	nature, hill, scenery, green, clouds		
	< no text >	night, lights, christmas, nightshot, nacht, nuit, notte, longexposure, noche, nocturna	flower, nature, green, flowers, petal, petals, bud		
	aheram, 0505, sarahc, moo	portrait, bw, balckandwhite, people, faces, girl, blackwhite, person, man	blue, red, art, artwork, painted, paint, artistic, surreal, gallery, bleu		
	unseulpixel, naturey crap	fall, autumn, trees, leaves, foliage, forest, woods, branches, path	bw, blackandwhite, noiretblanc, bianconero, blancoynegro		

$P(img|txt)$

N. Srivastava, R. Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, JMLR 2014

Multimodal DBM – Sampling

Step 50	Step 100	Step 150	Step 200	Step 250
travel	beach	sea	water	italy
trip	ocean	beach	canada	water
vacation	waves	island	bc	sea
africa	sea	vacation	britishcolumbia	boat
earthasia	sand	travel	reflection	italia
asia	nikon	ocean	alberta	mare
men	surf	caribbean	lake	venizia
2007	rocks	tropical	quebec	acqua
india	coast	resort	ontario	ocean
tourism	shore	trip	ice	venice

Input tags	Step 50	Step 100	Step 150	Step 200	Step 250
purple, flowers					
car, auto-mobile					

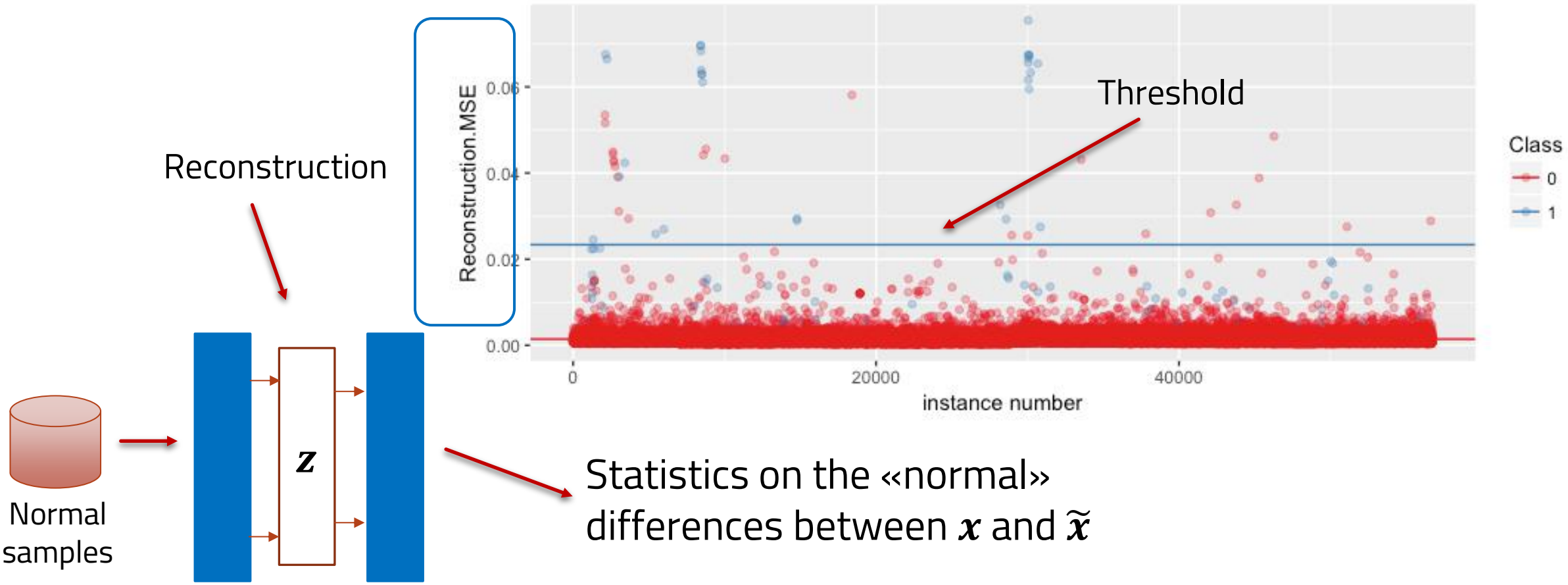
N. Srivastava, R. Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, JMLR 2014

Multimodal DBM – Multimodal Quering

Multimodal Query	Top 4 retrieved results			
 hongkong, causewaybay, shoppingcentre, building, mall	 howell, bridge, genesee, river, rochester, downtown, building	 london, uk, night, skyline, river, thames, lights, bridge	 edinburgh, scotland, dusk, bank	 arcoiris, fincadehierro, lluvia, sannicolas, valencia
 me, myself, eyes, blue, hair	 urban, me, abigfave, fiveflickrfavs,	 trisha, mynewcamera, lake, field, girl	 me, ofme, self, selfportrait	 pink, prettyinpink, explored

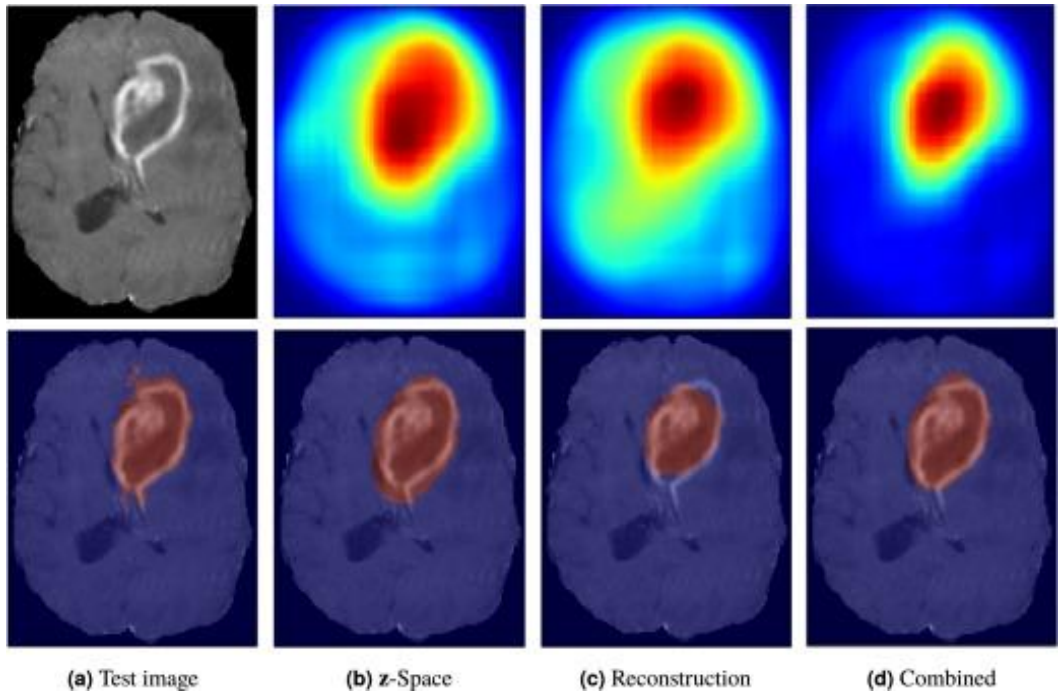
N. Srivastava, R. Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, JMLR 2014

Anomaly Detection



Unsupervised pathology detection in biomedical images

Unsupervised pathology detection of a brain tumor image and resulting anomaly scores (growing values: blue → red)



https://www.sciencedirect.com/science/article/pii/B9780128243497000153

Wrap-up

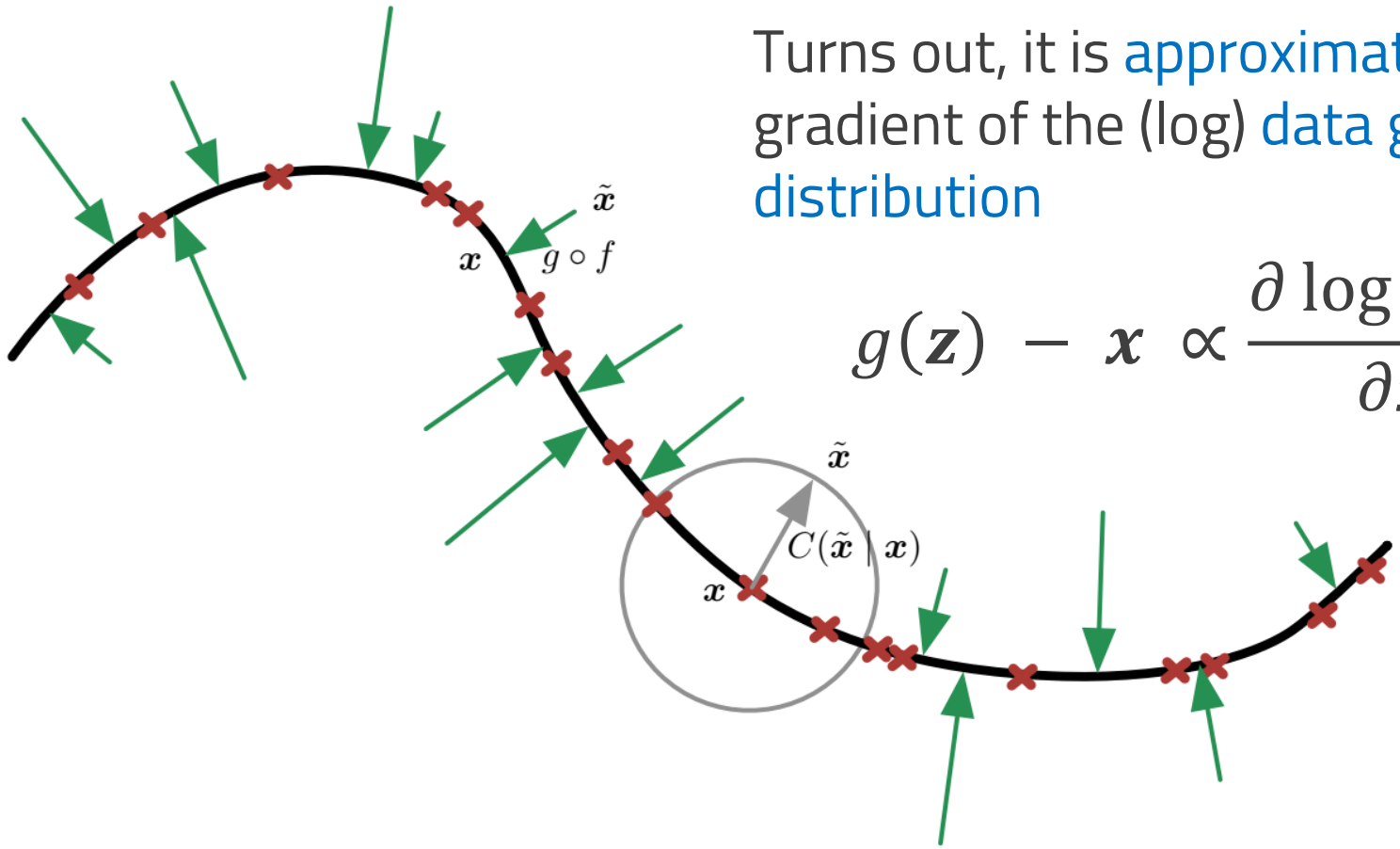
Take Home Messages

- Regularized autoencoder
  - Optimize reconstruction quality
  - Constrain stored information
- Autoencoder training is manifold learning
  - Learn a latent space manifold where input data resides
  - Store only variations that are useful to represent training data
- Deep AE: pretraining, fine tuning, supervised optimization
- Use AE for finding new/useful data representations or anomaly detection

Wait! Wasn't this supposed to be a module on generative deep learning?
I see no probabilities in autoencoders…

And yet...

Remember the vector field in DAE?



Turns out, it is approximating the gradient of the (log) data generating distribution

$$g(\boldsymbol{z}) - \boldsymbol{x} \propto \frac{\partial \log p(\boldsymbol{x})}{\partial x}$$

Next Lecture

Variational Autoencoders

- Autoencoders as explicit density learning models
- Variational inference strikes back (this time with neural models)
- Dissecting variational autoencoder design
- Variational autoencoders and representation learning