Deep Learning Theory and Applications

Autoencoders Part 2

Yale

CPSC/AMTH 663

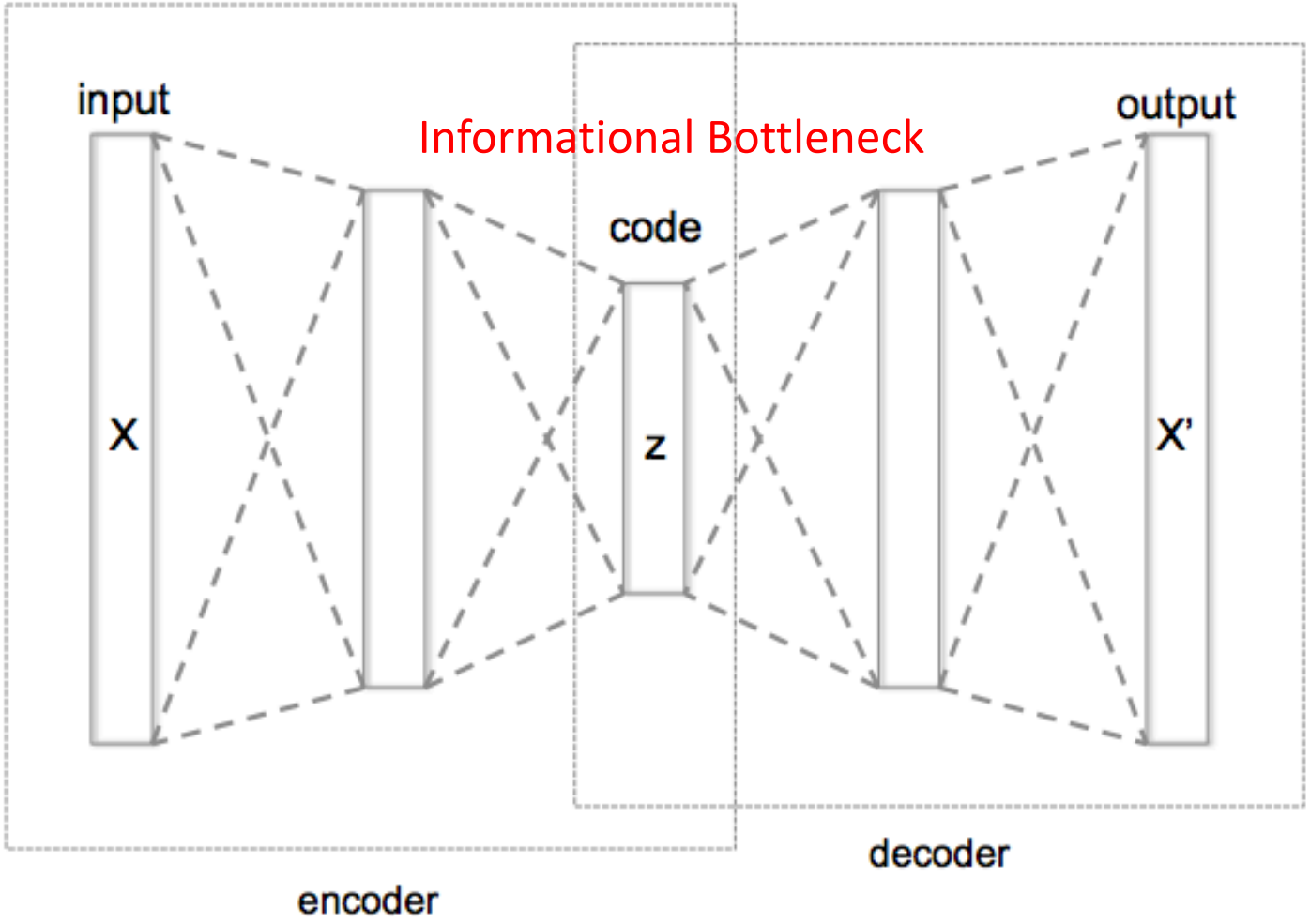
Outline

1. Unsupervised Learning
2. PCA
3. Autoencoders

Autoencoder (AE)

- Neural network trained to copy its input x to its output
- Hidden layer z describes a ***code*** to represent the input
- Two parts
  - Encoder: $\mathrm{z} = f(x)$
  - Decoder: $x' = g(z)$
- Goal: minimize $L\left(x, g\left(f(x)\right)\right)$
  - L penalizes $g\left(f(x)\right)$ for being dissimilar from x
  - Example: mean squared error
- How do you train an autoencoder?
  - Backpropagation
  - Recirculation (rarely)

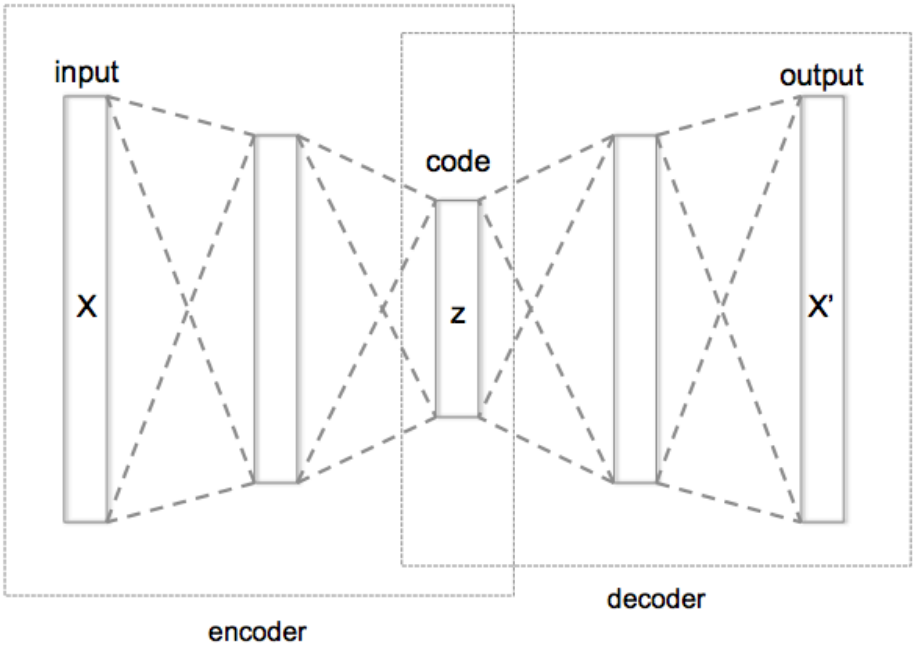
Autoencoder



[Hinton, Salakhutdinov, Science 2006]

Autoencoder Applications

- Data denoising
- Dimensionality reduction
  - Including visualization
- Feature learning
- Information retrieval
- Data compression
- Data generation



Are Autoencoders useful?

- Can't we just set $g\big(f(x)\big) = x$ everywhere?
  - Not very useful
- Design the AE so it can't learn to copy the input perfectly
  - Restrict the AE to copy only approximately
  - Can prioritize certain aspects of the input
- Examples
  - Restrict the size of the code (i.e. add a bottleneck)
  - Add regularization

Denoising Autoencoder

Add noise to the input while training and teach the autoencoder to remove noise [Bengio ‘09]

Information retrieval with AEs

Information retrieval: find entries in a database that resemble a query entry

- Train an AE to produce a low-dimensional *binary* code
  - Can store database entries in a hash table
  - Return all entries with the same binary code
- Do this with saturated sigmoids on the final layer of the encoder
  - Add additive noise to the sigmoid nonlinearity during training
  - The AE fights the noise until saturation occurs

The Autoencoder compromise

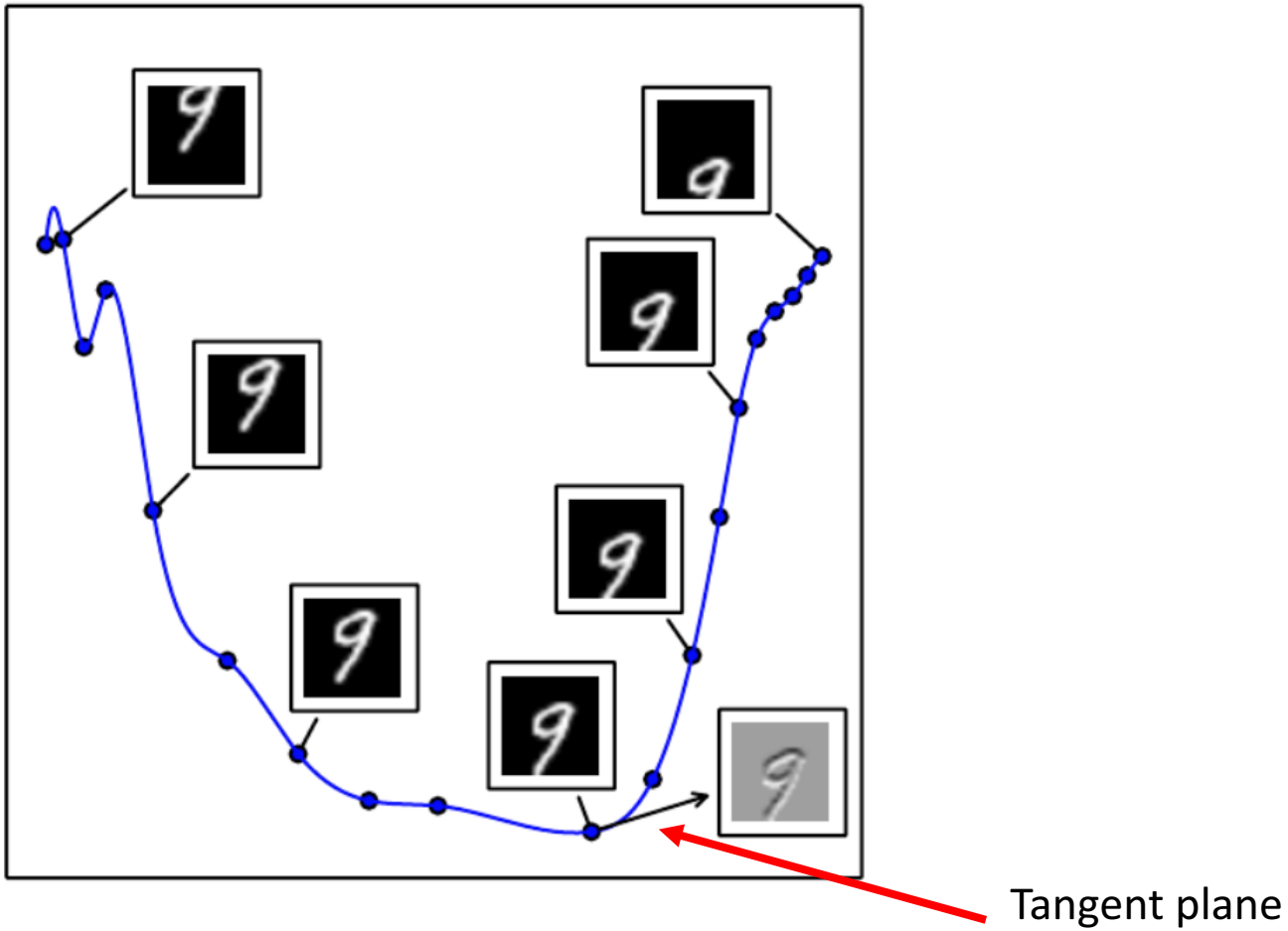
Two opposing forces:

1. Learn a representation z of x s.t. x can be approximately recovered from z
   - x is drawn from the training data ⇒ the AE learns to reconstruct only probable inputs
2. Satisfy the constraint or regularization penalty
   - E.g. architectural constraint or regularization term
   - Less sensitivity to the input is preferred

- Neither force alone is useful
- Together they force z to capture information about the structure of the data-generating distribution

Manifold learning

- Tangent plane: plane indicating the local directions of variation allowed on the manifold



Contractive Autoencoders

- Add a regularizer on the code $z = f(x)$
  - Encourages the derivatives of f to be small:

$$\Omega(z) = \lambda \left\| \frac{\partial f(x)}{\partial x} \right\|_F^2$$

  - $\Omega(z)$ is the squared Frobenius norm (sum of squared elements) of the (Jacobian) matrix of partial derivatives
- Connection to denoising AEs
  - For small Gaussian noise, the denoising reconstruction error is equivalent to a contractive penalty on $g\big(f(x)\big)$
  - I.e., denoising AEs make the reconstruction resist small, finite perturbations of the input
  - Contractive AEs make the reconstruction resist infinitesimal perturbations of the input

Contractive Autoencoders

- Why the name?
- A contractive AE maps a neighborhood of input points to a smaller neighborhood of output points
  - It contracts the input neighborhood to a smaller output neighborhood
- But this contraction is only local
  - All perturbations of x are mapped near to $f(x)$
- Contraction may not occur globally
  - Possible to map two points x and x' to points $f(x)$ and $f(x')$ that are farther apart than the original points

Contractive Autoencoders

- Why the name?
- A more formal approach:
- Can view the Jacobian matrix J at a point x as approximating $f(x)$ as a linear operator
- A linear operator is contractive if $\|Jx\| \leq 1$ for all x s.t. $\|x\| = 1$
  - I.e. J is contractive if it shrinks the unit sphere
  - Contractive AEs penalize the Frobenius norm of a local linear approximation of $f(x)$ at every training point x to encourage a contraction

Contractive Autoencoders

The two forces in a contractive AE:

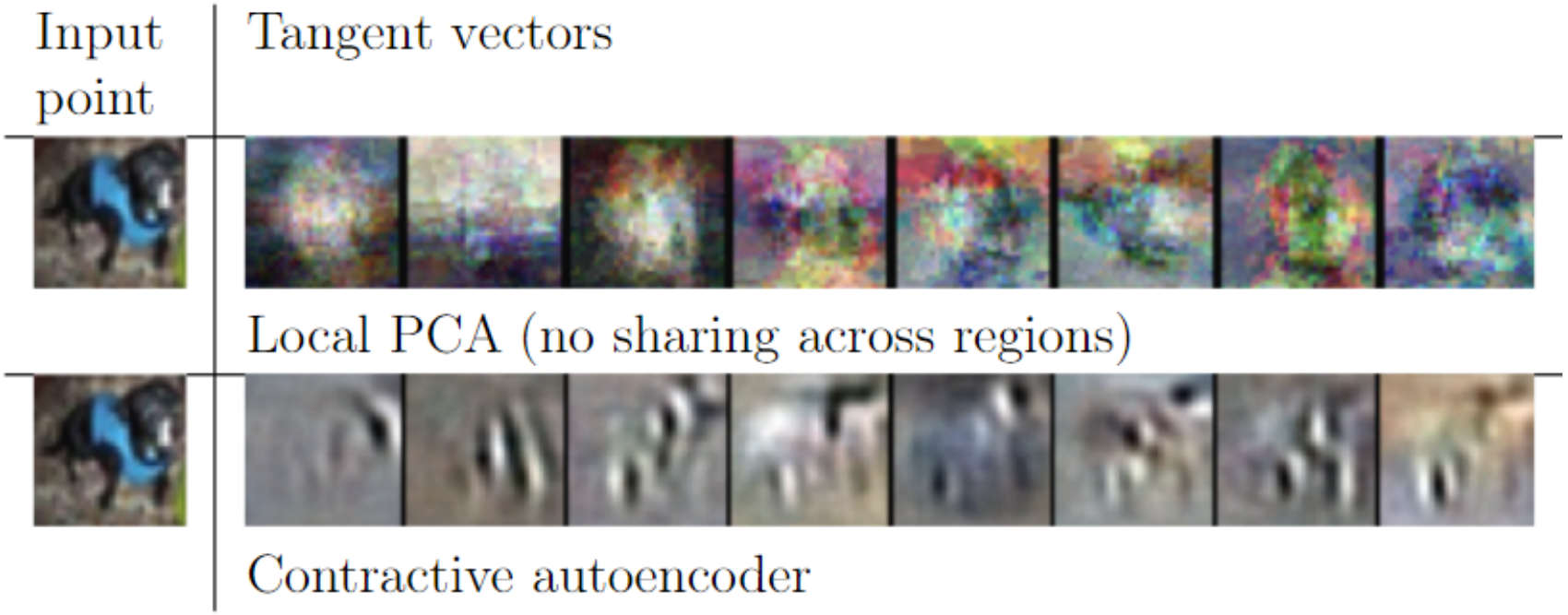
- Reconstruction error and the contractive penalty $\Omega(z)$
  - Reconstruction error alone would encourage the identity function
  - Contractive penalty alone would encourage constant features wrt x
- The compromise yields an AE with mostly tiny derivatives $\frac{\partial f(x)}{\partial x}$
  - Only a small number of directions may have significant derivatives

Contractive Autoencoders

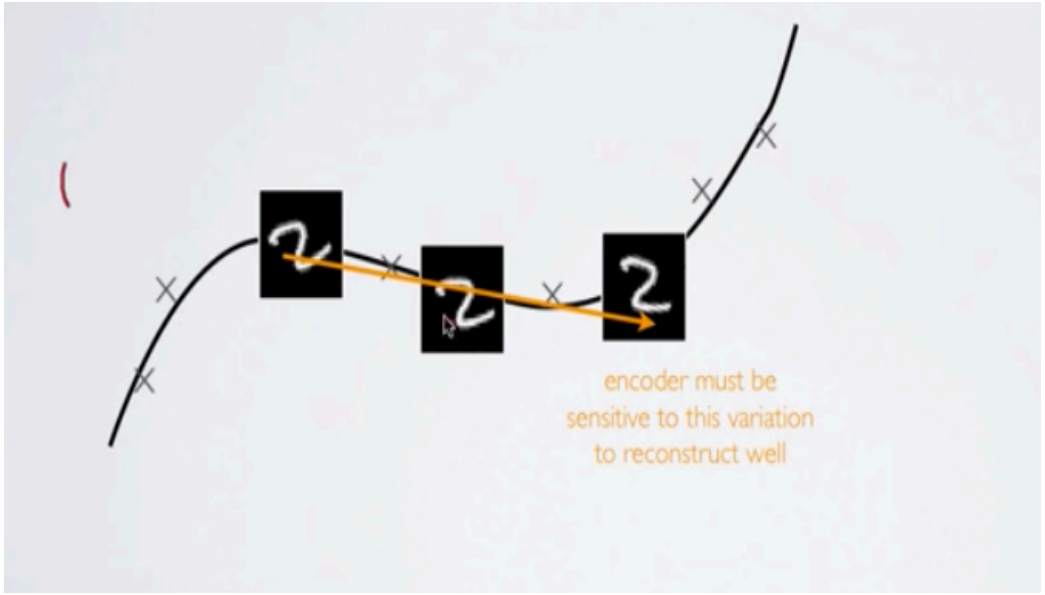
- Goal of a contractive AE is to learn the manifold structure
- Directions x with large Jx rapidly change z
  - Likely to approximate the directions (i.e. tangent planes) of the manifold
- Experimentally, training contractive AE results in most singular values of J being less than 1 (i.e. contractive)
- Directions with the largest singular values interpreted as the tangent directions

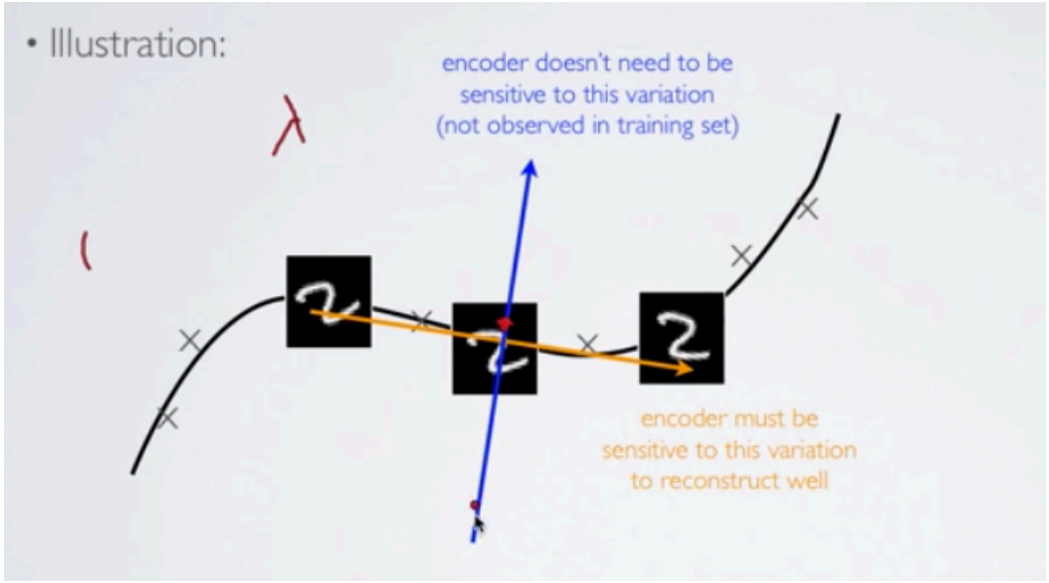
Contractive Autoencoders

- Experimentally obtained tangent vectors
  - Correspond to meaningful transformations



Contractive Autoencoders

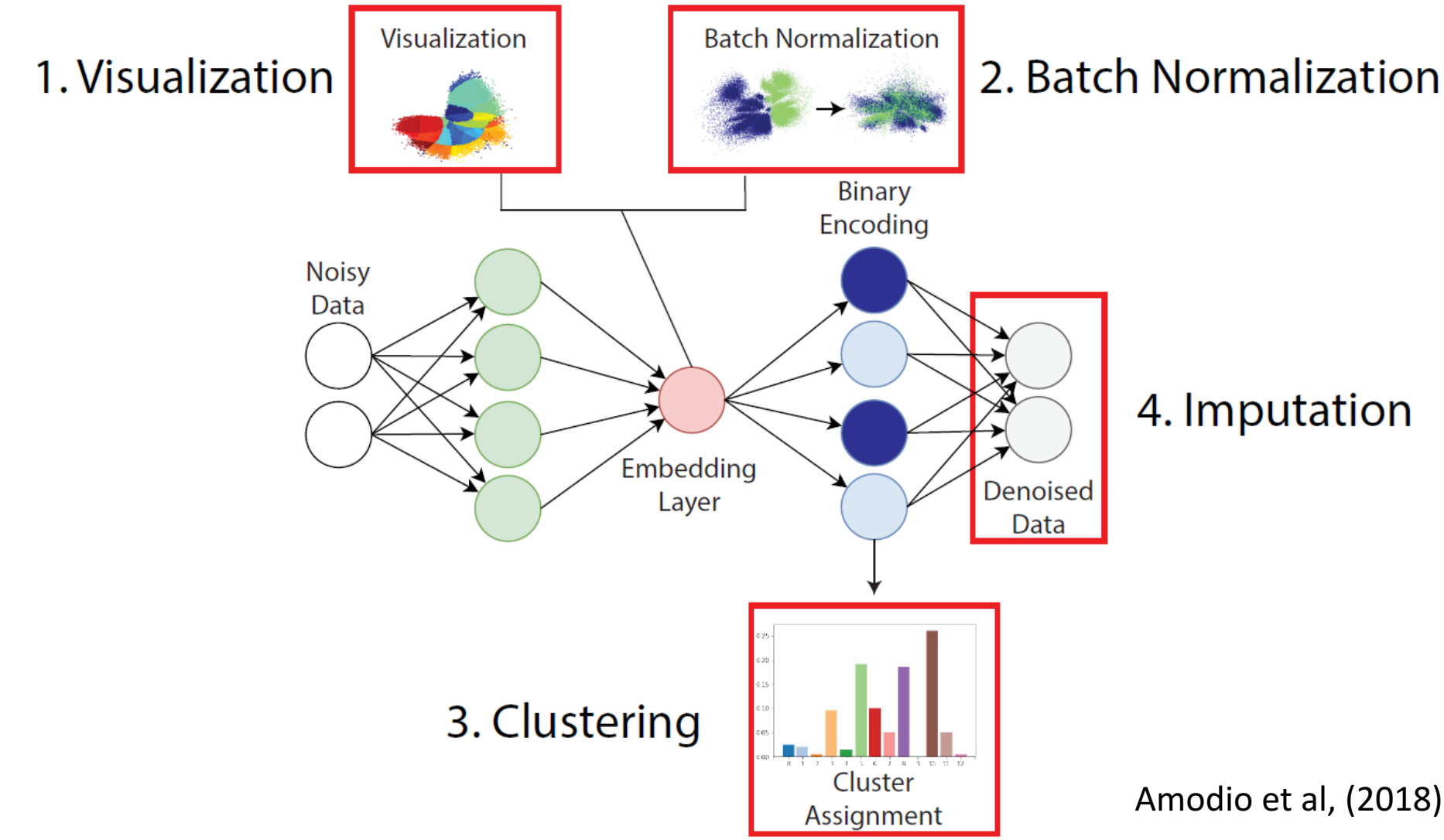




Contractive Autoencoders

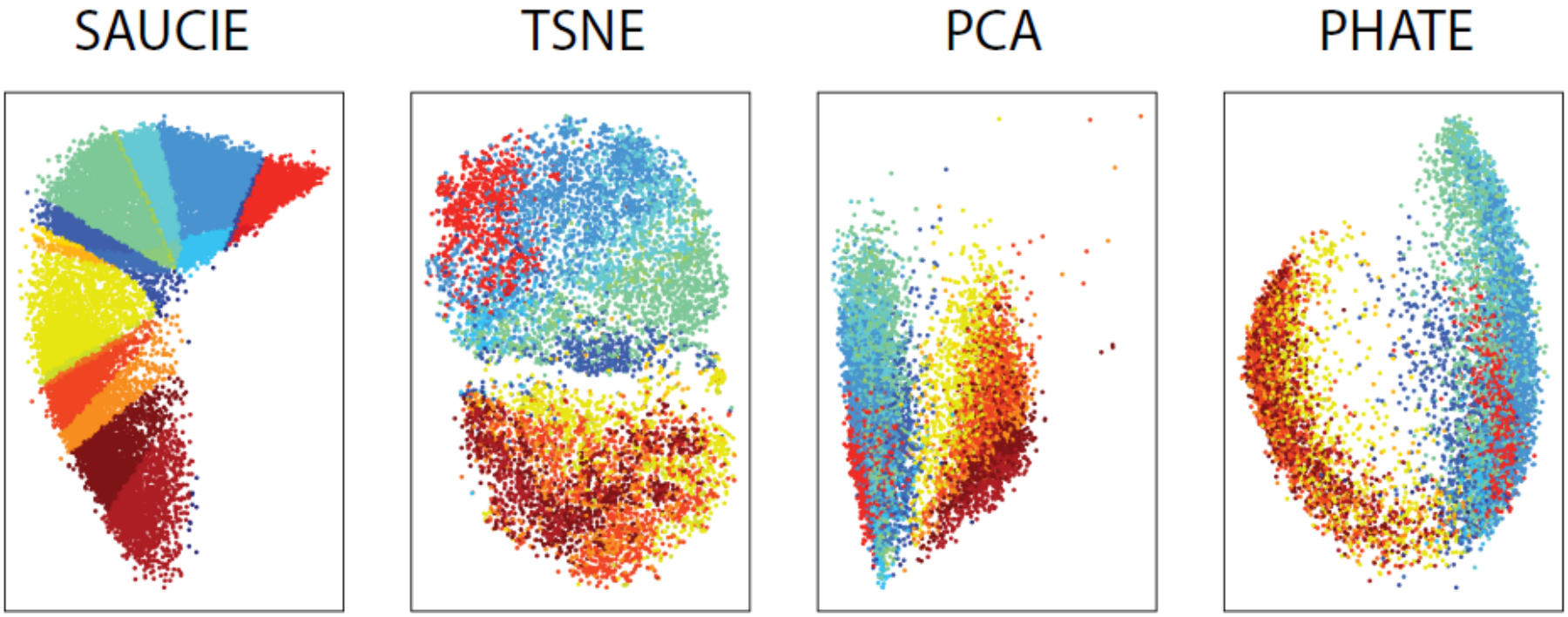
- Practical issue: deep contractive AEs expensive to compute
- One strategy:
  - Separately train a series of single-layer autoencoders to reconstruct the previous autoencoder's hidden layer
  - Compose these together to form a deep AE
- Practical issue: Can obtain useless results
  - E.g., encoder multiply by ϵ and decoder divide by ϵ
    - Driving ϵ to zero satisfies the contractive penalty
- Strategy: tie the weights of f and g together
  - E.g., set the weight matrix of g to be the transpose of the weight matrix of f

Autoencoder example: SAUCIE



Amodio et al, (2018)

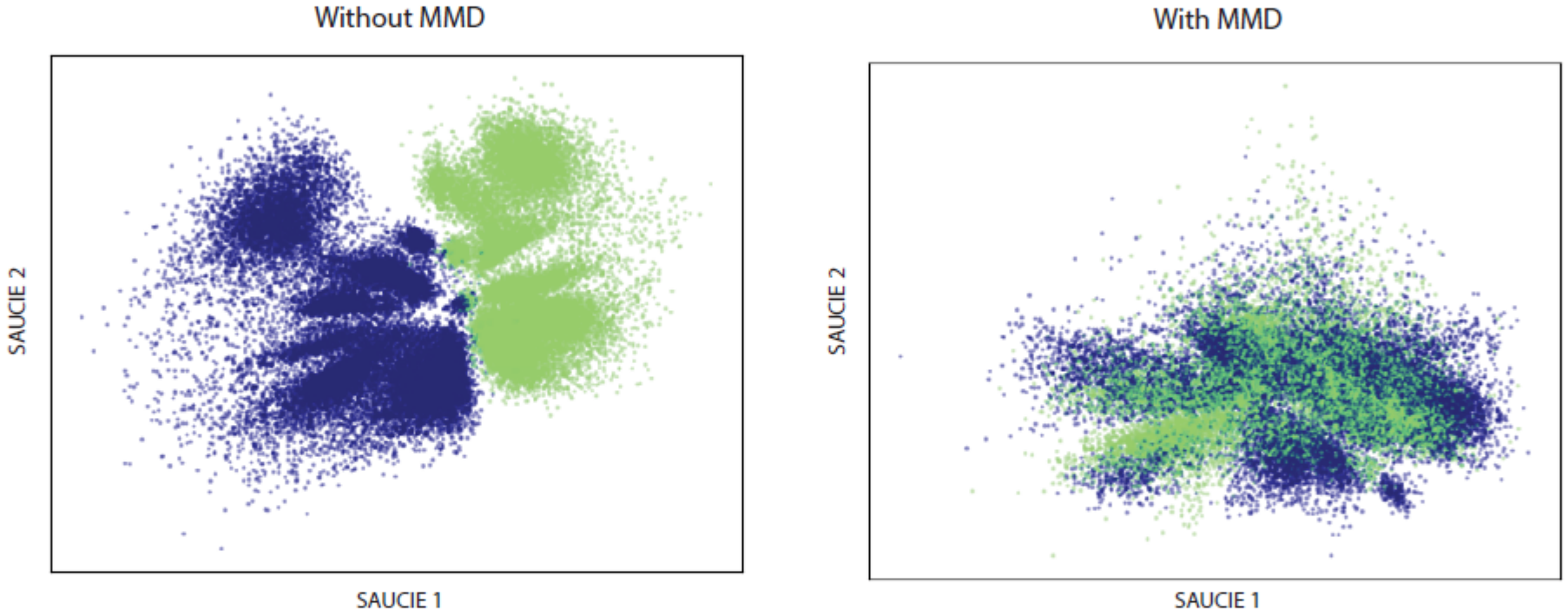
SAUCIE: Visualization



- The embedding layer gives a non-linear dimensionality reduction for visualization

Amodio et al, (2018)

SAUCIE: Batch Correction



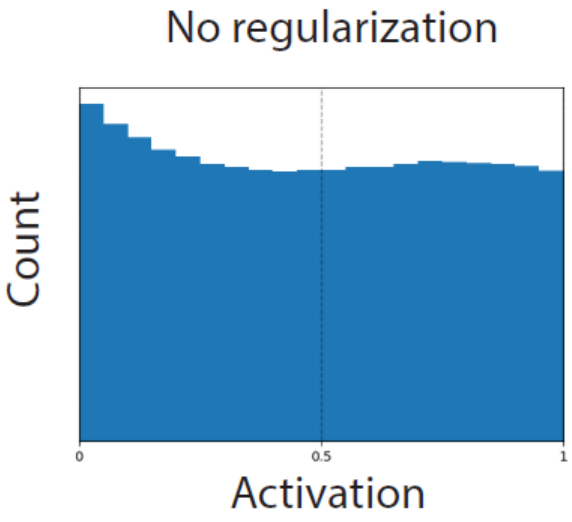
- Batch correction of CyTOF panels run on different days

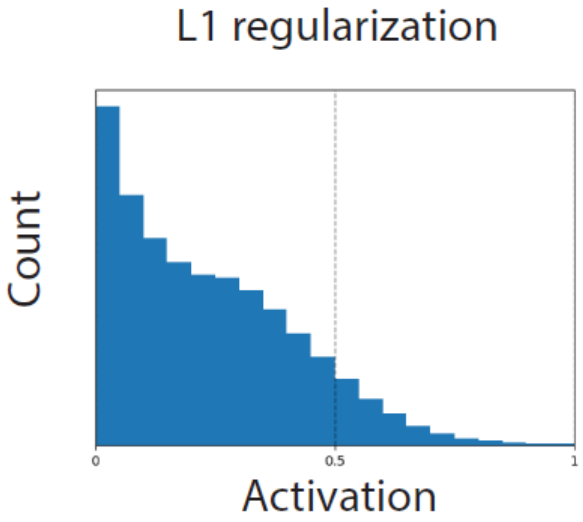
$$MMD = \frac{1}{n^2} \sum_{ij=1}^{n} k(x_i, x_j) + \frac{1}{m^2} \sum_{ij=1}^{m} k(y_i, y_j) - \frac{2}{mn} \cdot \sum_{ij=1}^{m} k(x_i, y_j)$$

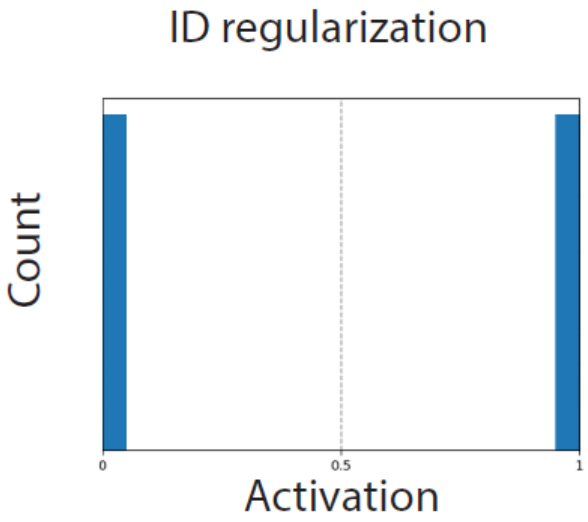
Amodio et al, (2018)

SAUCIE: Clustering

- Clustering layer gives binary codes
  - Clustering of extremely large datasets
- Information Dimensionality (ID) Regularization
  - Entropy of the activations



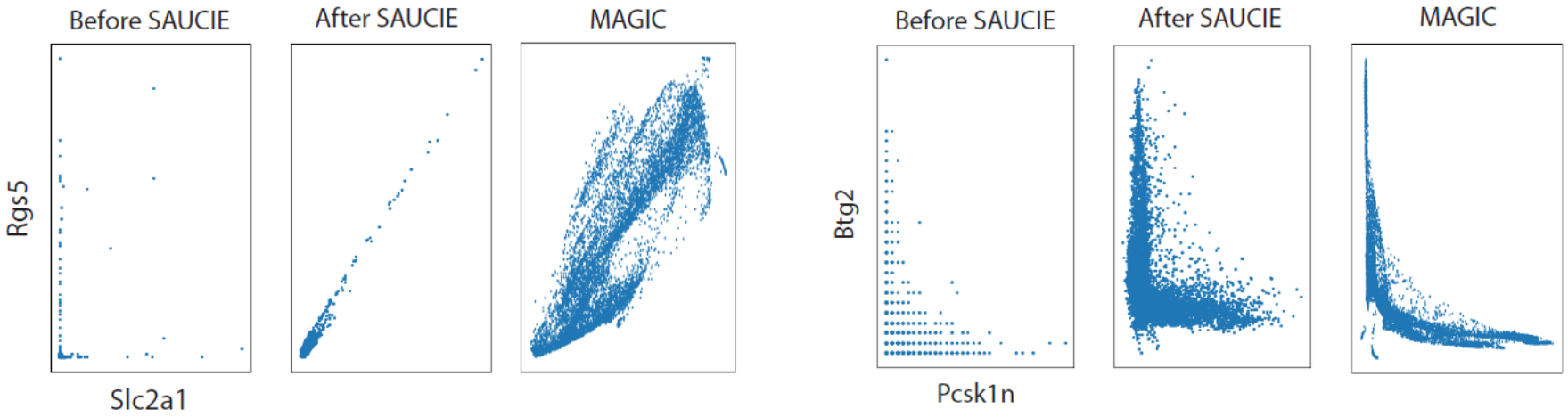




Amodio et al, (2018)

SAUCIE: Imputation

- Reconstructed output imputes dropout in scRNA-seq data



Amodio et al, (2018)

Representation learning revisited

- How the information is represented matters
- Example: easy to divide 210 by 6 using Arabic numerals
  - What about Roman numerals?
  - Divide CCX by VI
- Example: insert a number into a sorted list
  - $O(n)$ for a linked list
  - $O(\log n)$ for a red-black tree
- Similarly, a good representation can make machine learning tasks easier

Representation learning revisited

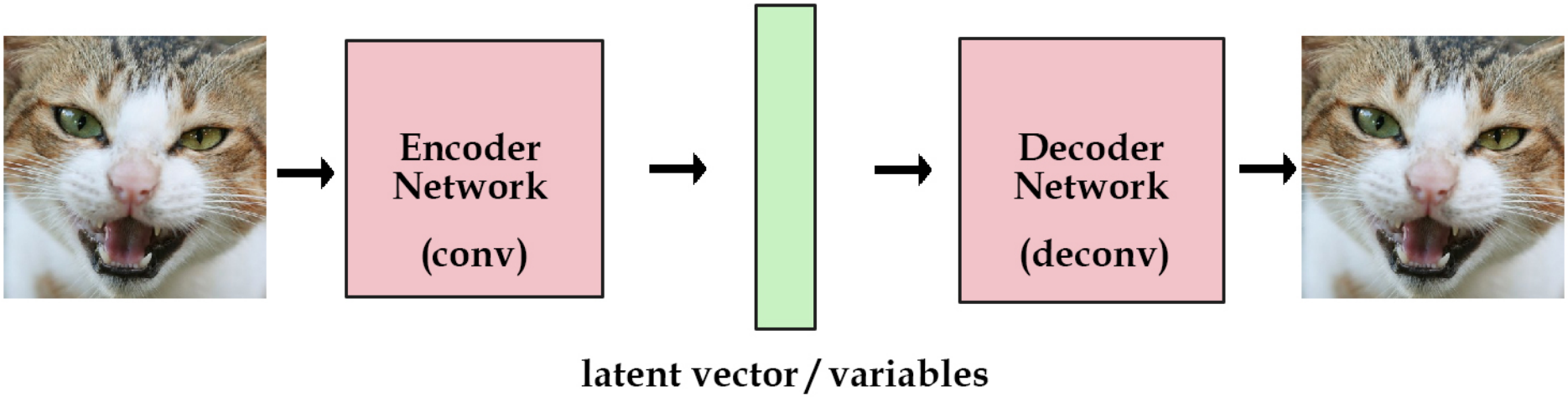
- A feedforward network for classification is doing some kind of representation learning
  - Last layer gives the classification
  - The rest of the network learns a good representation for that classification
- Other kinds of representation learning can be designed (e.g. autoencoders)
- Particularly useful for unsupervised and semi-supervised learning

Unsupervised Pretraining

- Greedy layer-wise unsupervised pretraining
  - Enabled deep networks for the first time (without special architectures like convolution or recurrence)
  - A representation learned for one task (unsupervised learning) can be sometimes useful for another task (supervised learning with the same input domain)
- Approach: Train a single layer at a time to learn a representation
  - E.g., use a single layer autoencoder
  - Each layer is pretrained to take the output of the previous layer and produce a new representation that is simpler
  - Finally, train the entire network to perform the task

Variational Autoencoders (VAEs)

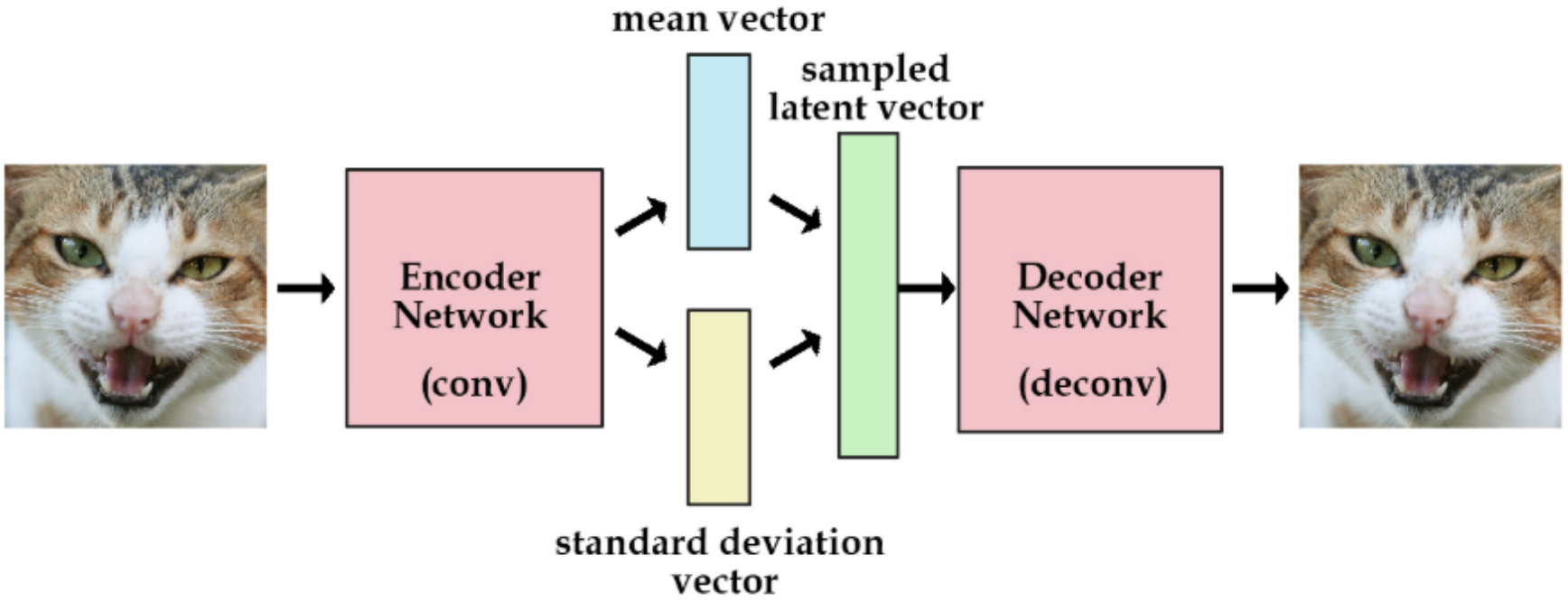
- Standard autoencoder



- We want to generate new examples
- Can we use an autoencoder?

Variational Autoencoders (VAEs)

- Force the latent vectors to have (roughly) a unit Gaussian distribution
- Generate images by sampling a unit Gaussian and passing it into the decoder



Variational Autoencoders (VAEs)

- Tradeoff between accuracy and the unit Gaussian approximation
- Build this directly into the loss
  - Kullback-Leibler (KL) divergence: a measure of difference between probability distributions P and Q

$$D_{KL}(P||Q) = -\sum_{i} P(i) \log\left(\frac{Q(i)}{P(i)}\right)$$

  - Include a regularization term that penalizes the KL divergence between a unit Gaussian and the latent vector distribution
  - How do we do this in practice?

Variational Autoencoders (VAEs)

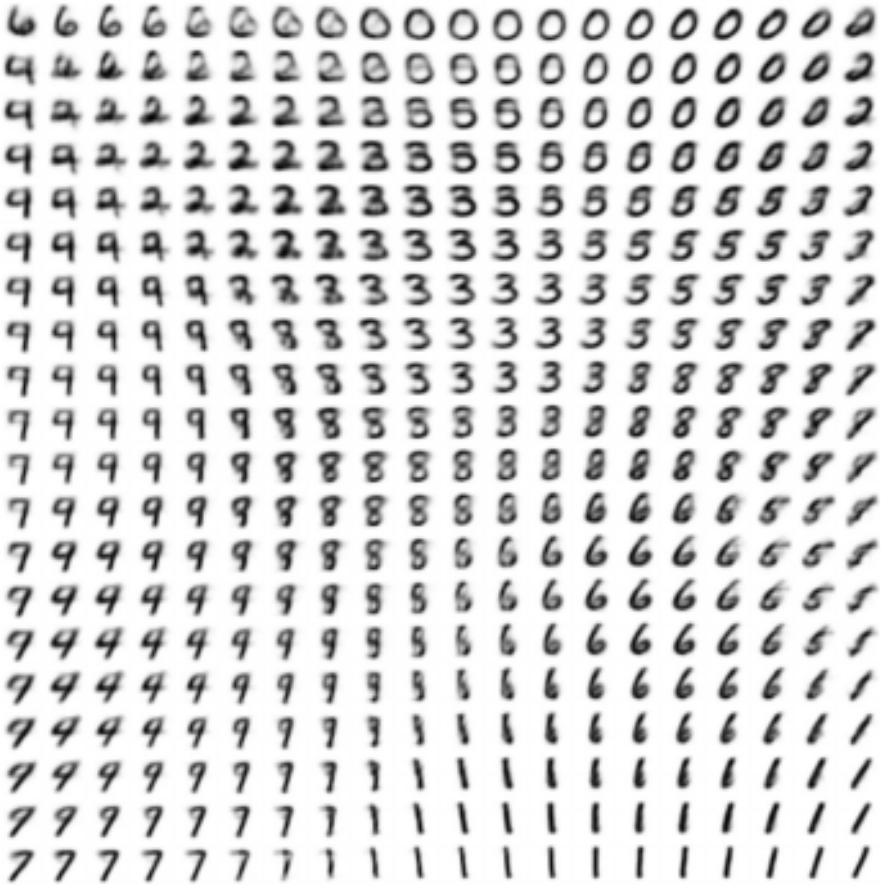
1st Epoch

9th Epoch

Original

Variational Autoencoders (VAEs)

2D latent mapping of results from VAEs



Question?

- How can you generate from a normal autoencoder

Further reading

- http://kvfrans.com/variational-autoencoders-explained/
- https://blog.keras.io/building-autoencoders-in-keras.html
- Goodfellow et al., chapters 14 and 15