Unsupervised Learning: Dimensionality Reduction

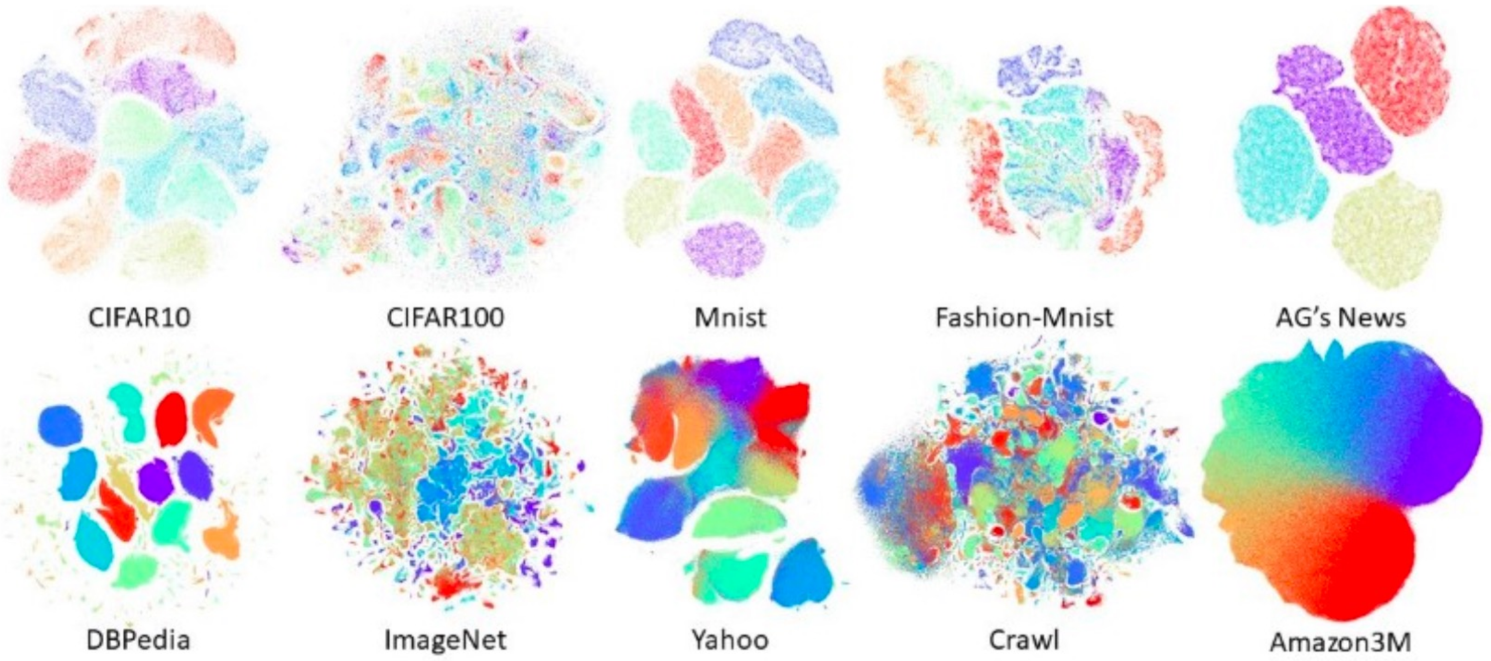
COSC 410: Applied Machine Learning

Spring 2022

Prof. Apthorpe

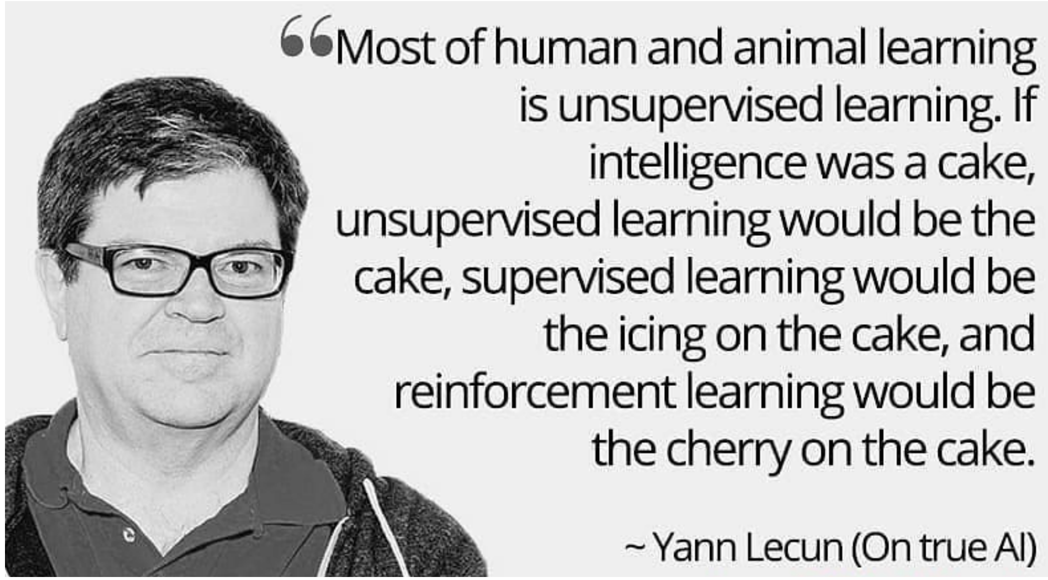
Outline

- Overview
- Principal Component Analysis
- t-SNE
- Autoencoders



Unsupervised Machine Learning

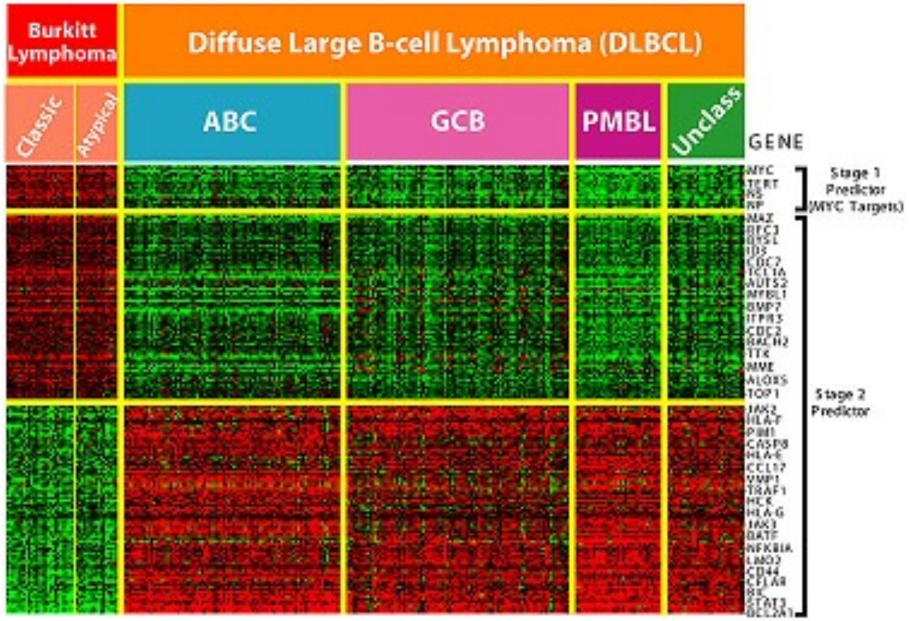
- Data is **unlabeled**
  - Examples with features $\mathbf{X}$
  - No labels $\mathbf{y}$
- *Goal:* Identify patterns in data without "ground-truth" classes



Most data in the world is unlabeled!

Dimensionality Reduction

- Real-world datasets are often high dimensional (many features)
  - Hard to visualize & increases training times
- *Goal:*
  - Reduce # of features
  - Preserve patterns as much as possible

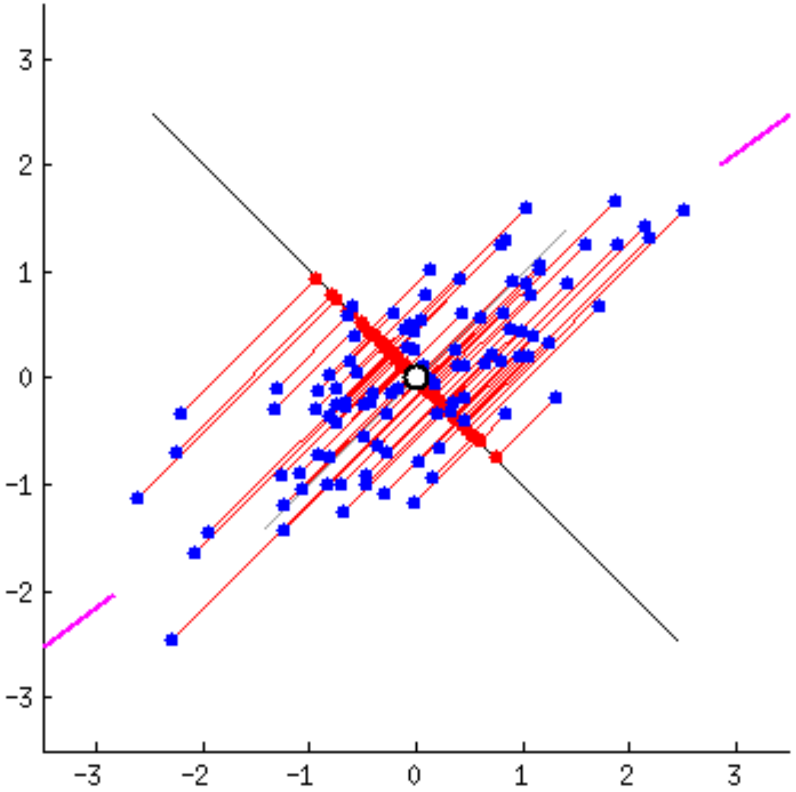


High-dimensional microarray for measuring gene expression

Principal Component Analysis (PCA)

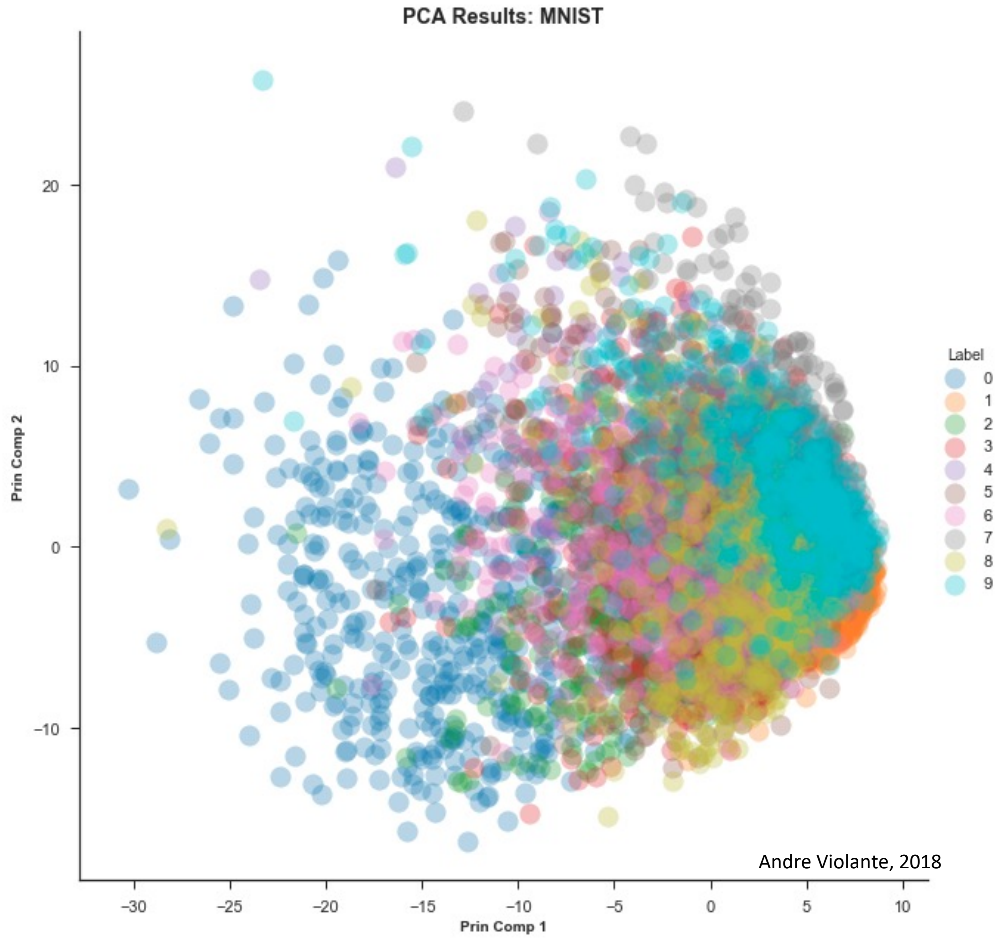
Principal Component Analysis

- Choose dimensionality of target projection *N*
- Find new basis (axes) in *N*-dimensions s.t. projecting examples
  - Maximizes variance (to preserve predictive power)
  - Minimizes mean-square distance from original to projected positions
- The new basis vectors (axes) are the principal components
  - Each example has 1 feature per principal component



Limitations of PCA

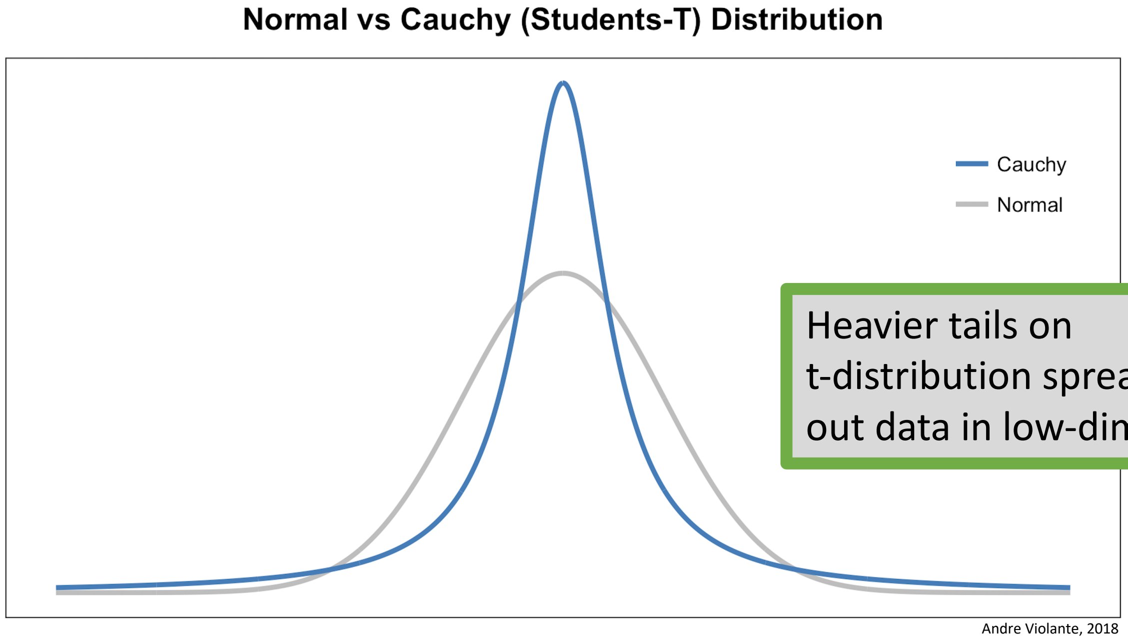
- Sensitive to feature scaling
  - Remember to standardize!
- Resulting features are less interpretable
- Often produces “blobby” visualizations in 2D
- Doesn’t maintain non-linear relationships between examples
  - Can use **kernel PCA** instead



t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE

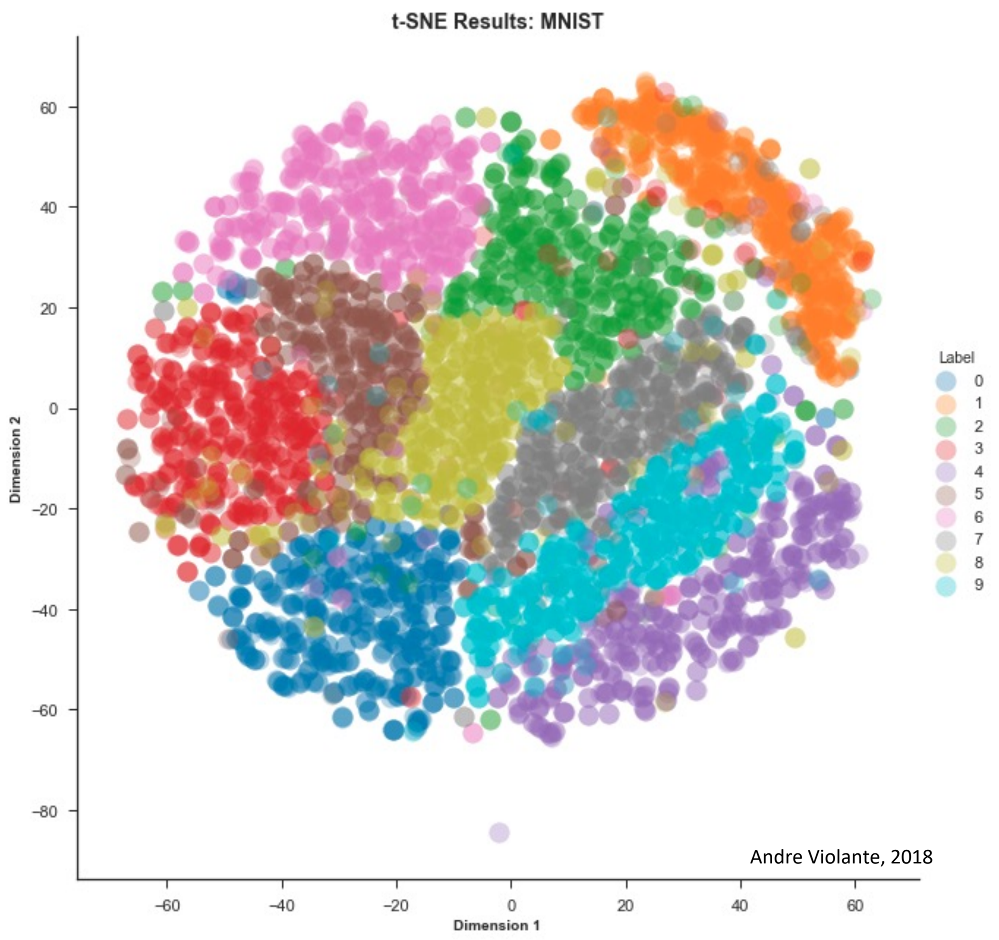
- Compute Gaussian distribution over pairs of examples in **high-dimensional** space
  - **Similar** examples → **higher** probability should be chosen as neighbors
  - **Dissimilar** examples → **lower** probability should be chosen as neighbors
- Find mapping from Gaussian to *t*-distribution in **low-dimensional** space
  - Minimize the divergence between the two distributions across the low-dim space
  - Use gradient descent…
- Nonlinear dimensionality reduction
  - Can preserve non-linear relationships in data



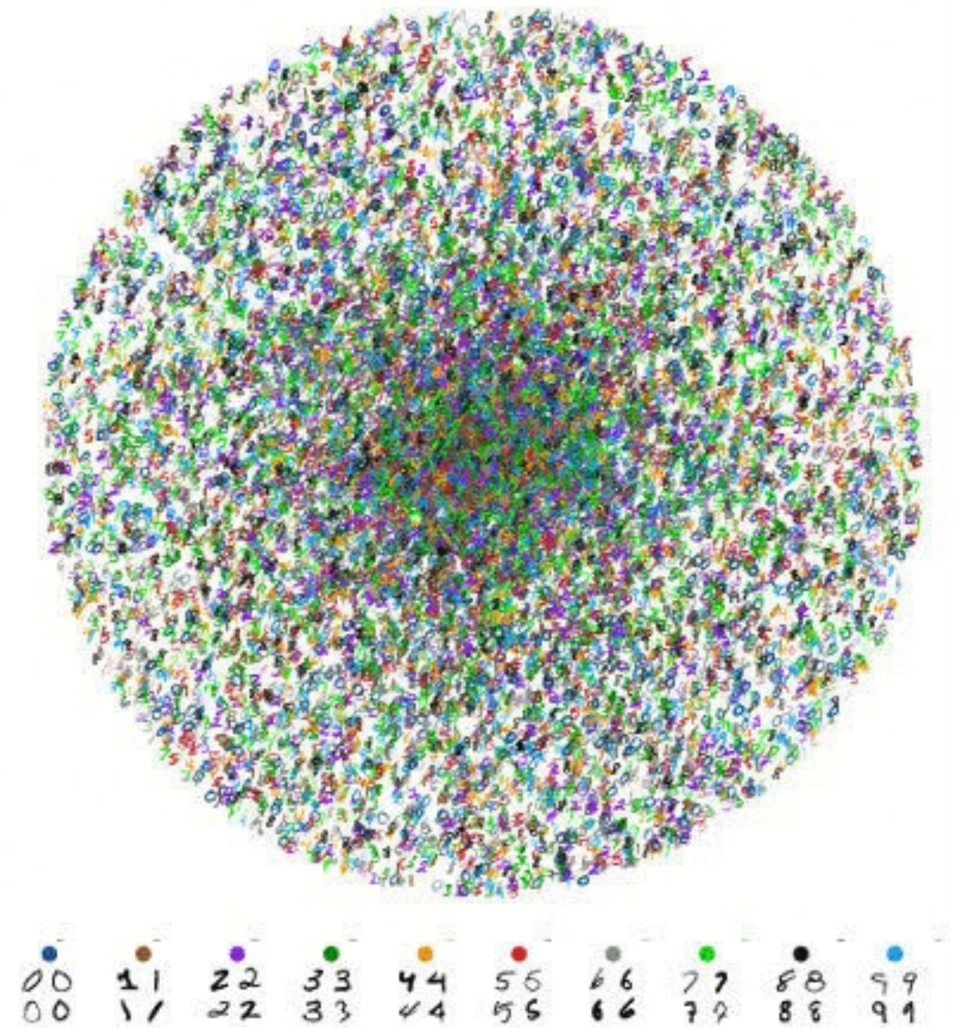
Heavier tails on t-distribution spread out data in low-dim

PCA vs. t-SNE


PCA Results: MNIST
Prin Comp 2
Prin Comp 1
20
10
0
−10
−30
−25
−20
−15
−10
−5
0
5
10
Label
0
1
2
3
4
5
6
7
8
9
Andre Violante, 2018

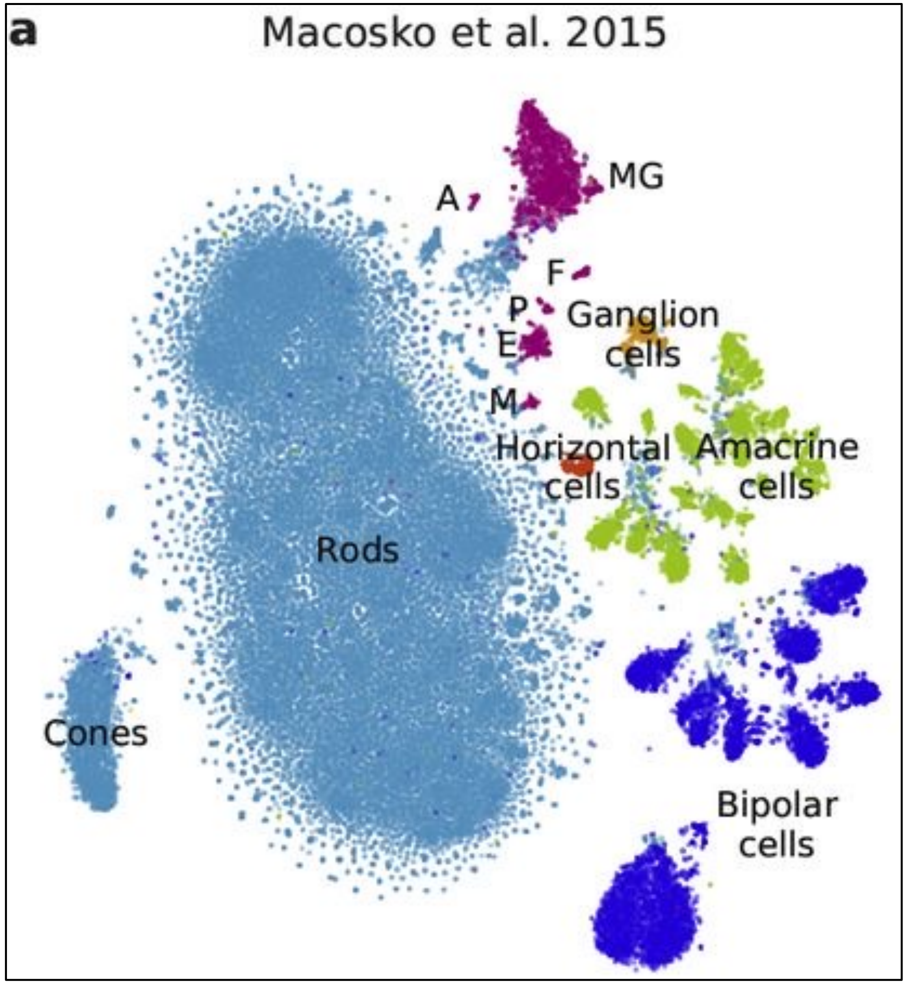

t-SNE Results: MNIST
Dimension 2
Dimension 1
60
40
20
0
−20
−40
−60
−80
−60
−40
−20
0
20
40
60
Label
0
1
2
3
4
5
6
7
8
9
Andre Violante, 2018

t-SNE Visualization on MNIST



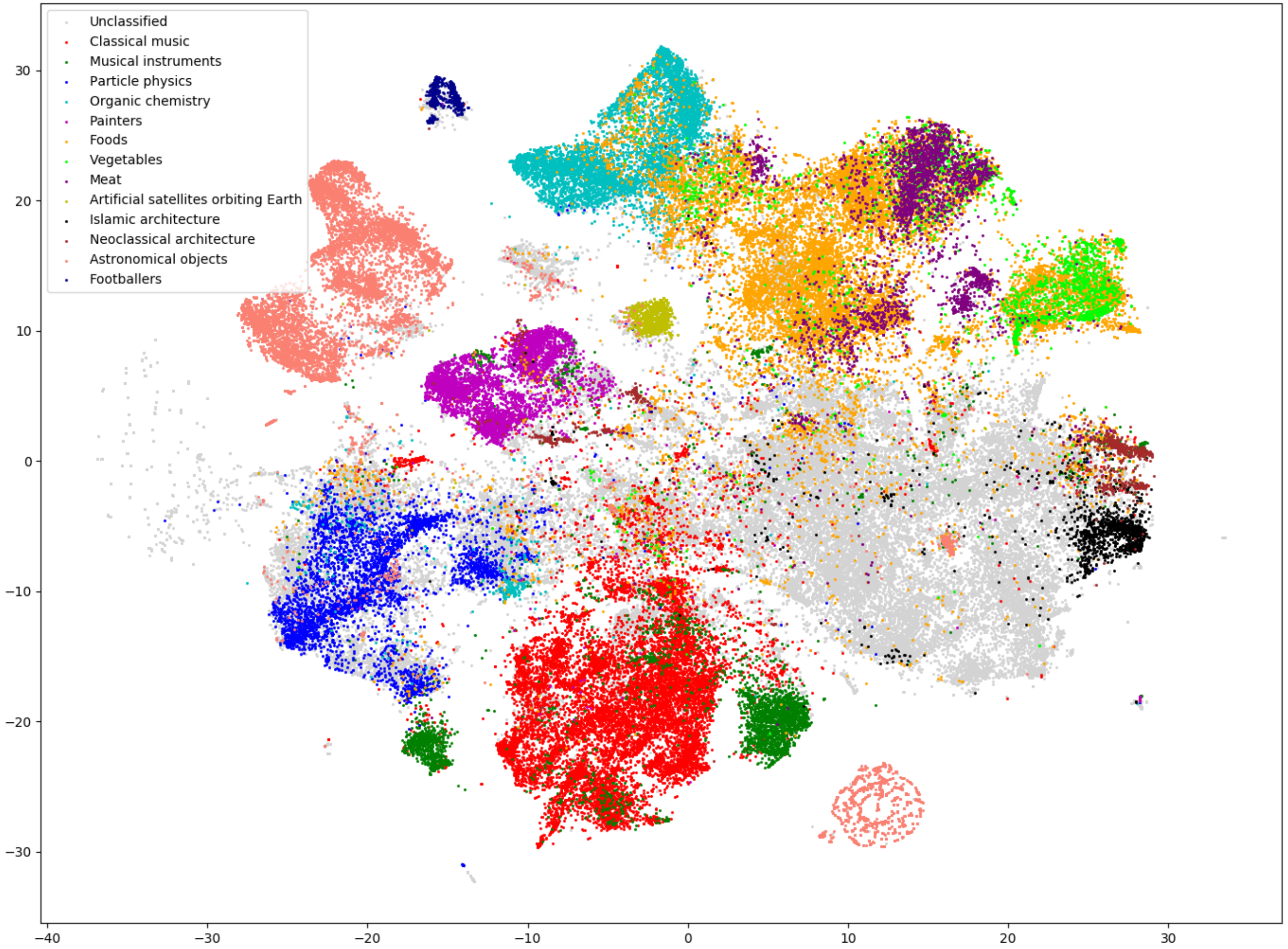
Pezzotti et al., 2018

t-SNE for Biological Data



Macosko et al. (2015) data set, n = 44,808 cells from the mouse retina. Bipolar cells comprise eight clusters, amacrine cells comprise 21 clusters. Non-neural clusters are abbreviated (MG: Mueller glia, A: astrocytes, F: fibroblasts, P: pericytes, E: endothelium, M: microglia).

t-SNE on Natural Language Data



2D visualization of 120k Wikipedia pages

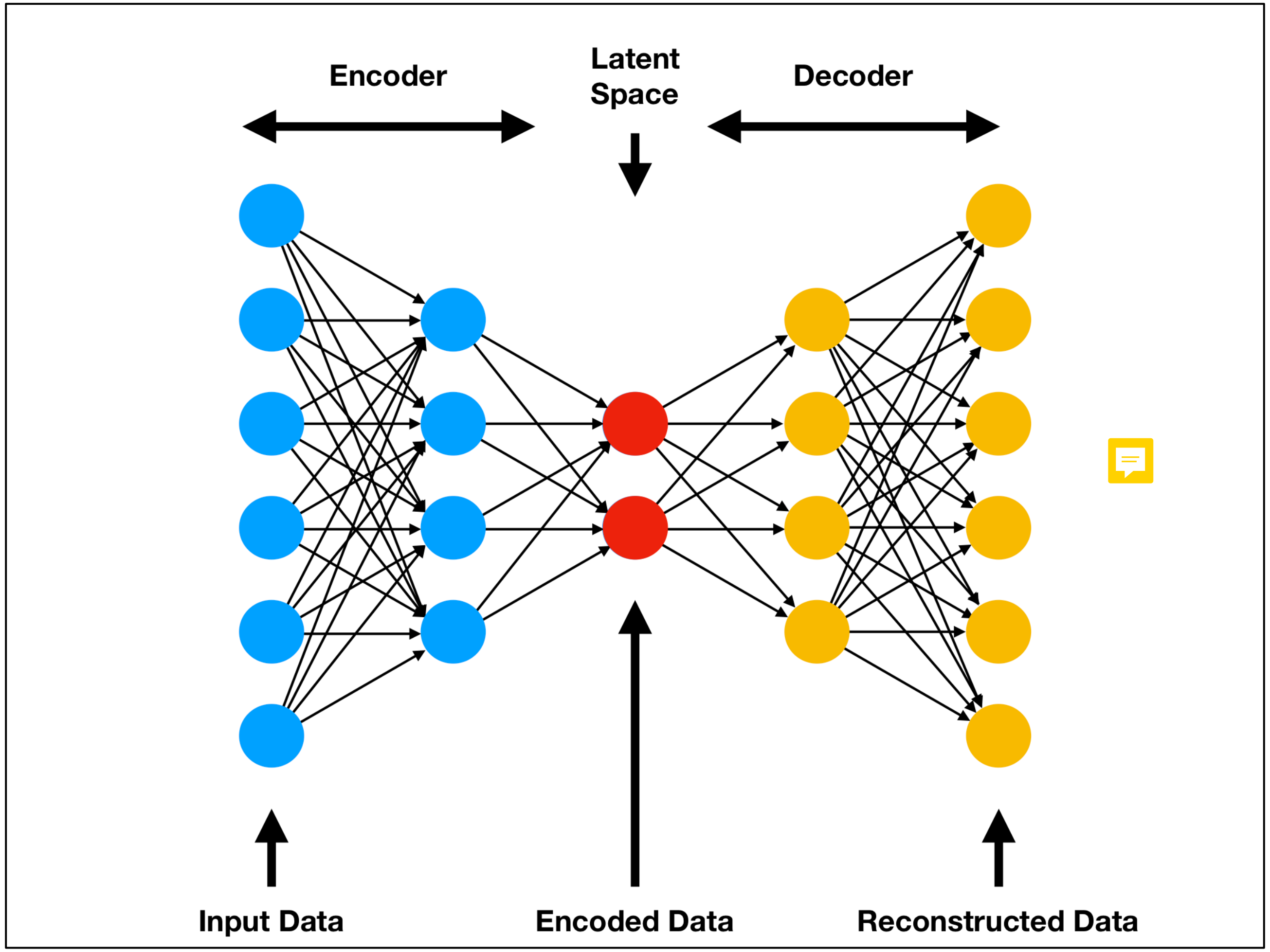
Dr. Vlasta Kůs, Dr. Alessandro Negro, 2018

Limitations of t-SNE

- Computationally expensive
- Non-parametric
  - Can't apply to new examples without rerunning entire algorithm
  - Why does this matter for ML?
- Preferred for visualizations
  - Less effective for reducing features before supervised learning

Autoencoders

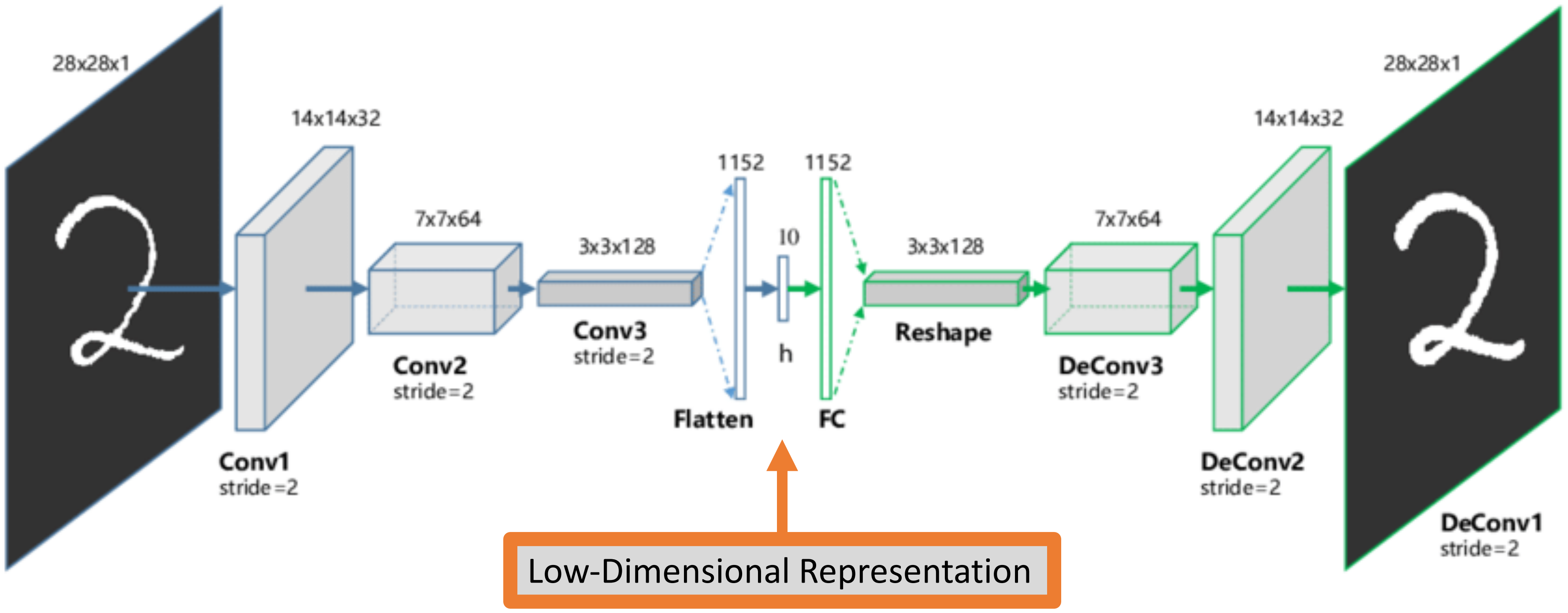
Autoencoders



Steven Flores, 2019

Autoencoders

- Autoencoders can be feedforward, recurrent, or convolutional



Questions?