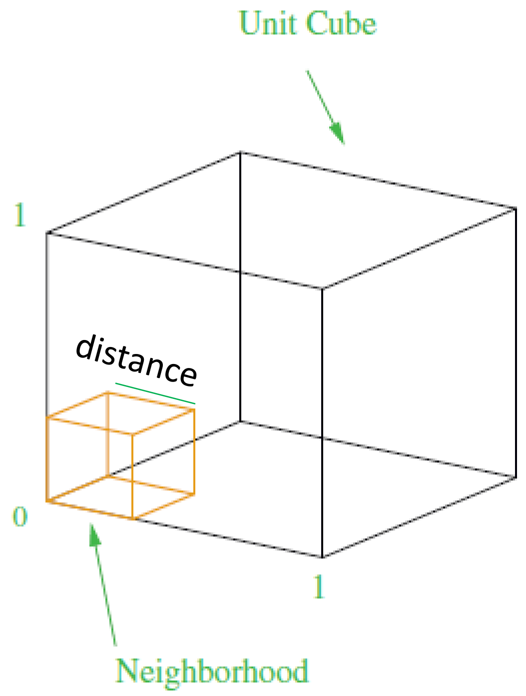
Manifolds

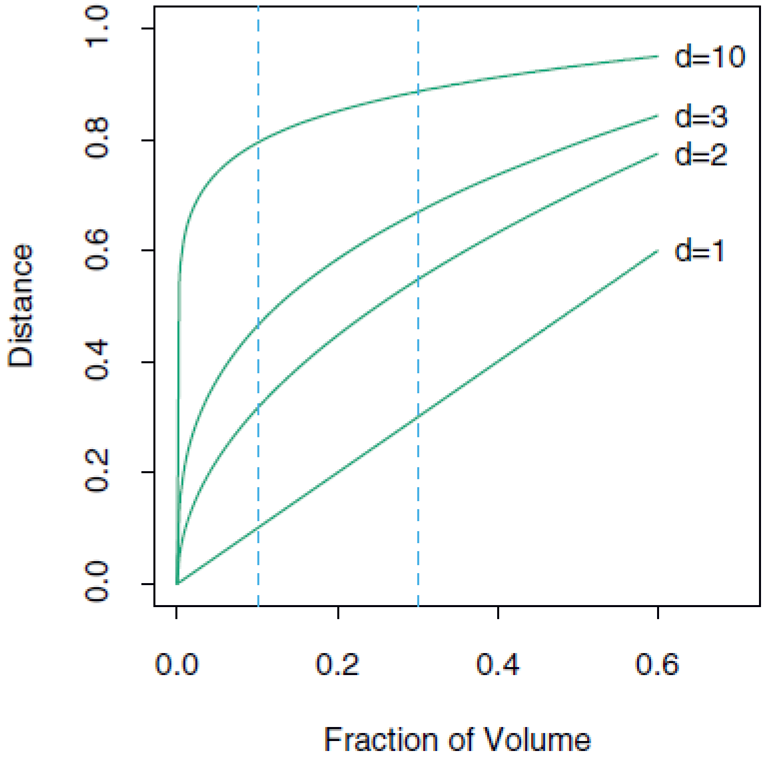
Professor Marie Roch

San Diego State University

Munch's *The Scream*

Curse of dimensionality



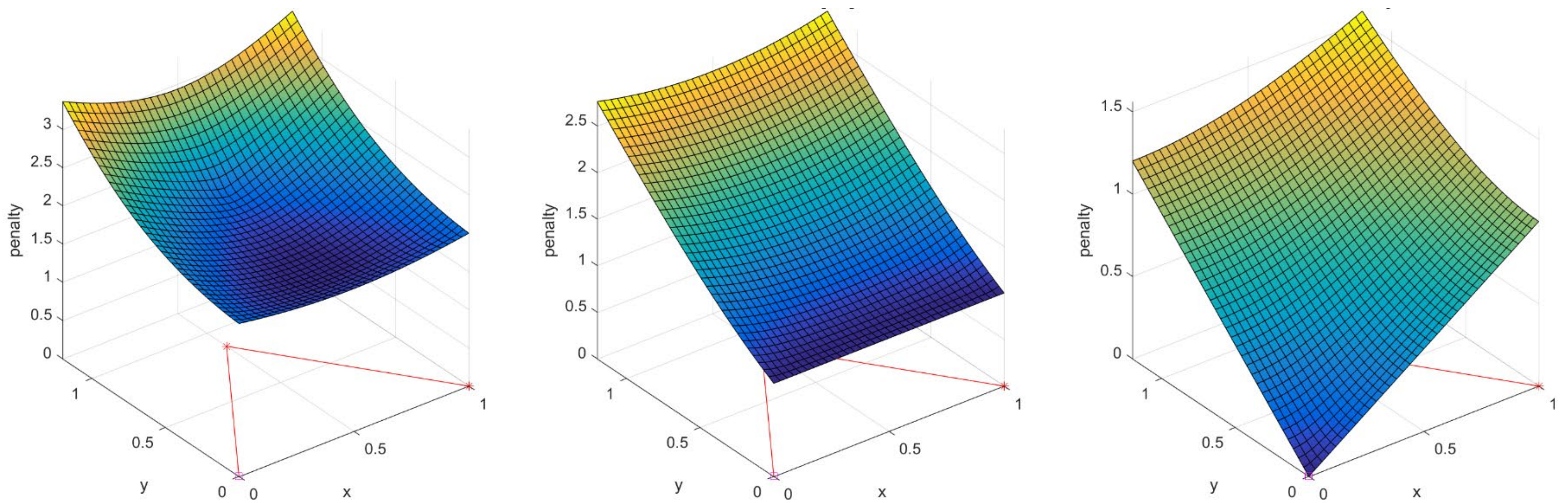


Hastie et al. (2009), p. 23

Training features should cover much of the space…

Manifolds

- Frequently what we measure can be expressed more compactly.
- Low dimensional representation of higher dimension object



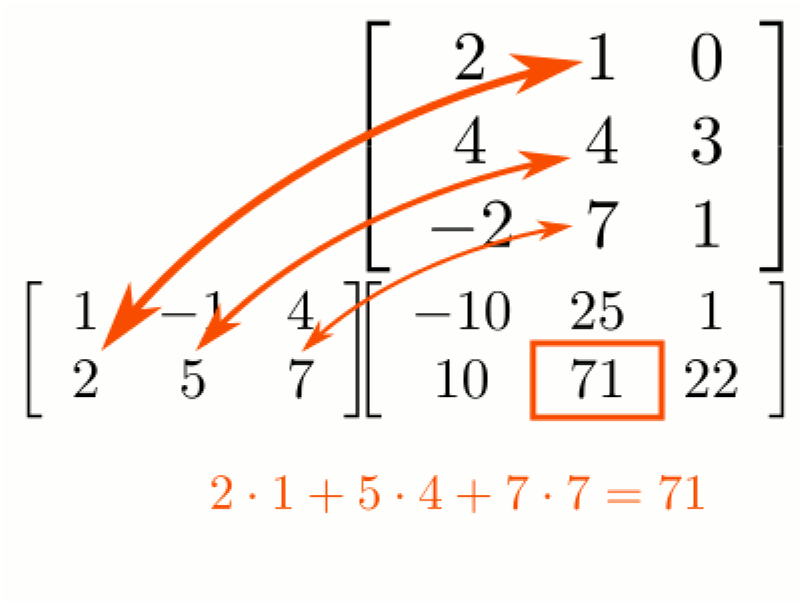
Manifolds

- Your text covers how to do this in a non-linear manner (chapter 14).
- For now, we will use principle components analysis, but we'll need a brief review of linear algebra

Linear algebra review

Goodfellow et al. 2.2 for details

- Matrix multiplication
  - inner dimension must match:
    [2 x 3][3 x 3]=[2 x 3]



calculus.seas.upenn.edu/?n=Main.Foreshadowing

Eigen vectors

Goodfellow et al. 2.7 for details

- Special vectors *x* such that in

$$Ax = \lambda x \text{ for some } \lambda \in \Re$$

- Vector x is merely scaled
  (∃line through origin, x and λx.)
- All such vectors are uncorrelated.
- When the $||x|| = 1$ (unit vector), λ is the eigen value.

numpy.linalg.eig

see Powell and Lehe demo

Principal components analysis (PCA)

- Finds new basis set to represent data
- Relies on eigen vectors and values
- Bases account for different amounts of variance in data and low contributors can be discarded

Munch The Scream

PCA – Let's get our hands dirty

- Let X be an *NxD* data matrix
- Assume expected value has been subtracted (no loss of generality)
- Can think of feature space as
  - having basis vectors u_1, u_2, …, u_D along axes
  - each row of X is a combination of those vectors

PCA

- Goal: Pick a new set of basis vectors
  - First vector

$$X\begin{bmatrix} w_{(1),1} \\ w_{(1),\dots} \\ w_{(1),D} \end{bmatrix} = y$$

produces vector of y values in direction $w_{(1)}$

  - Select such that var(y) is maximized

  - Repeat finding next largest uncorrelated basis

$$\sum_{i=1}^{D} w_{(1),i}^2 = 1$$

PCA

- In practice, this becomes an eigenvector problem on the variance-covariance matrix
- Principal components are eigenvectors ordered by descending eigenvalue.

- Remember:
  - Eigen vectors are the direction in which a matrix transforms data
  - Eigen values are the amount by which they scale
  - Need a quick introduction: watch Youtube lecture on eigen values/vectors of symmetric matrices (~ 5 min)

Computing PCA

- Estimate covariance matrix ∑ of X
  X is an NxD data matrix.
- Remember:

$$cov\left(X_i, X_j\right) = E([(x_i - \mu_i)^2\left(x_j - \mu_j\right)^2]$$

  If we detrend X (subtract off means: $cov\left(X_i, X_j\right) = E[x_i^2 \cdot x_j^2]$
  or $\Sigma = X^T X$
- Compute eigen vectors e_i and values λ_i of ∑ and arrange by largest eigen value to smallest:

$$e_1, e_2, \ldots, e_D$$

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D$$

Using PCA

- To project data, multiply by the number of bases desired, e.g. for data matrix X

$$\underbrace{\begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,D} \\ x_{2,1} & x_{2,2} & \dots & x_{2,D} \\ \dots & & & \\ x_{N-1,1} & x_{N-1,2} & \dots & x_{N-1,D} \\ x_{N,1} & x_{N,2} & \dots & x_{N,D} \end{bmatrix}}_{NxD}^{\text{original data}} \underbrace{\begin{bmatrix} e_{1,1} & \dots & e_{m,1} \\ \vdots & & \vdots \\ e_{1,D} & & e_{m,D} \end{bmatrix}}_{Dxm}^{\text{most important eigen vectors}} = \underbrace{\begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,m} \\ b_{2,1} & b_{2,2} & \dots & b_{2,m} \\ \dots & & & \\ b_{N-1,1} & b_{N-1,2} & \dots & b_{N-1,m} \\ b_{N,1} & b_{N,2} & \dots & b_{N,m} \end{bmatrix}}_{Nxm}^{\text{data projected to m dimensions}}$$

$1 \leq m \leq D$

project to m dimensions

Use [numpy.dot] to multiply matrices

How much of the variance is captured?

- Sum of variances (trace of covar) is the same as the sum of the eigen values: $tr(\Sigma) = \sum_{i=1}^{D} \lambda_i$
- The first m dimensions contain the variance represented by the sum of their eigen values:

$$\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{j=1}^{D} \lambda_j}$$

Component loadings

Loadings give the correlation between the bases and the features, e.g. for eigen vector e_i:

$$\frac{e_{i,1}\sqrt{\lambda_i}}{\sigma_{f_1}} \quad \frac{e_{i,2}\sqrt{\lambda_i}}{\sigma_{f_2}} \quad \cdots \quad \frac{e_{i,D-1}\sqrt{\lambda_i}}{\sigma_{f_{D-1}}} \quad \frac{e_{i,D}\sqrt{\lambda_i}}{\sigma_{f_D}}$$

PCA of correlation matrix

- The same analysis can be done on the sample correlation matrix R
- Eigen values will add up to D. Why?
- What is the qualitative difference with this type of analysis?

Nonlinear manifolds exist

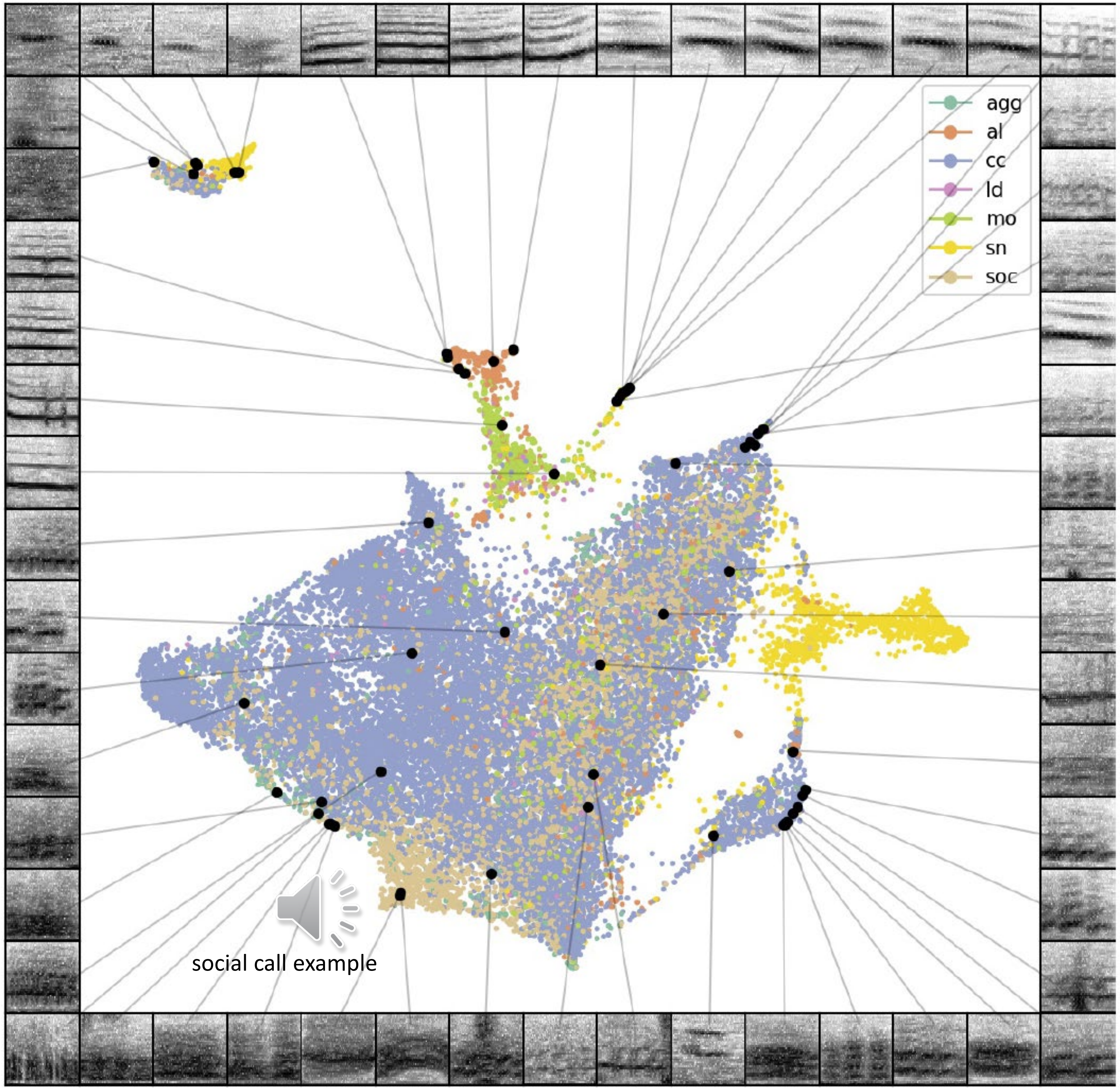
- t-distributed stochastic neighbor embedding (t-SNE)
- uniform manifold approximation and projection (UMAP)
- autoencoders

- Insufficient time to cover these, basic ideas…

Non-linear mappings (e.g. t-SNE/UMAP)

- Given *N* points in high-dimensional space, select *N* points in a lower-dimensional space
- Use information theoretic measures to align the distribution of the high dimensional points with that of the low dimensional points (moving the low dimensional points).
- Methods pay attention to local structure
- UMAP has a penalty term that tends to better preserve gaps between clusters

UMAP example



courtesty Mara Thomas

based on techniques by Sainburg et al 2019

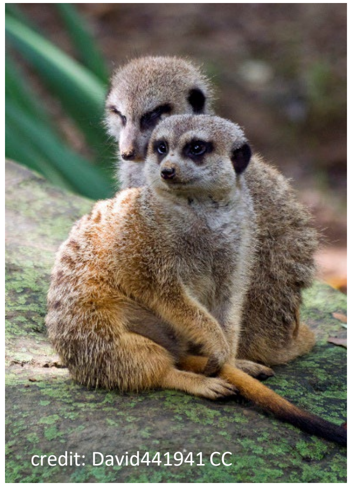
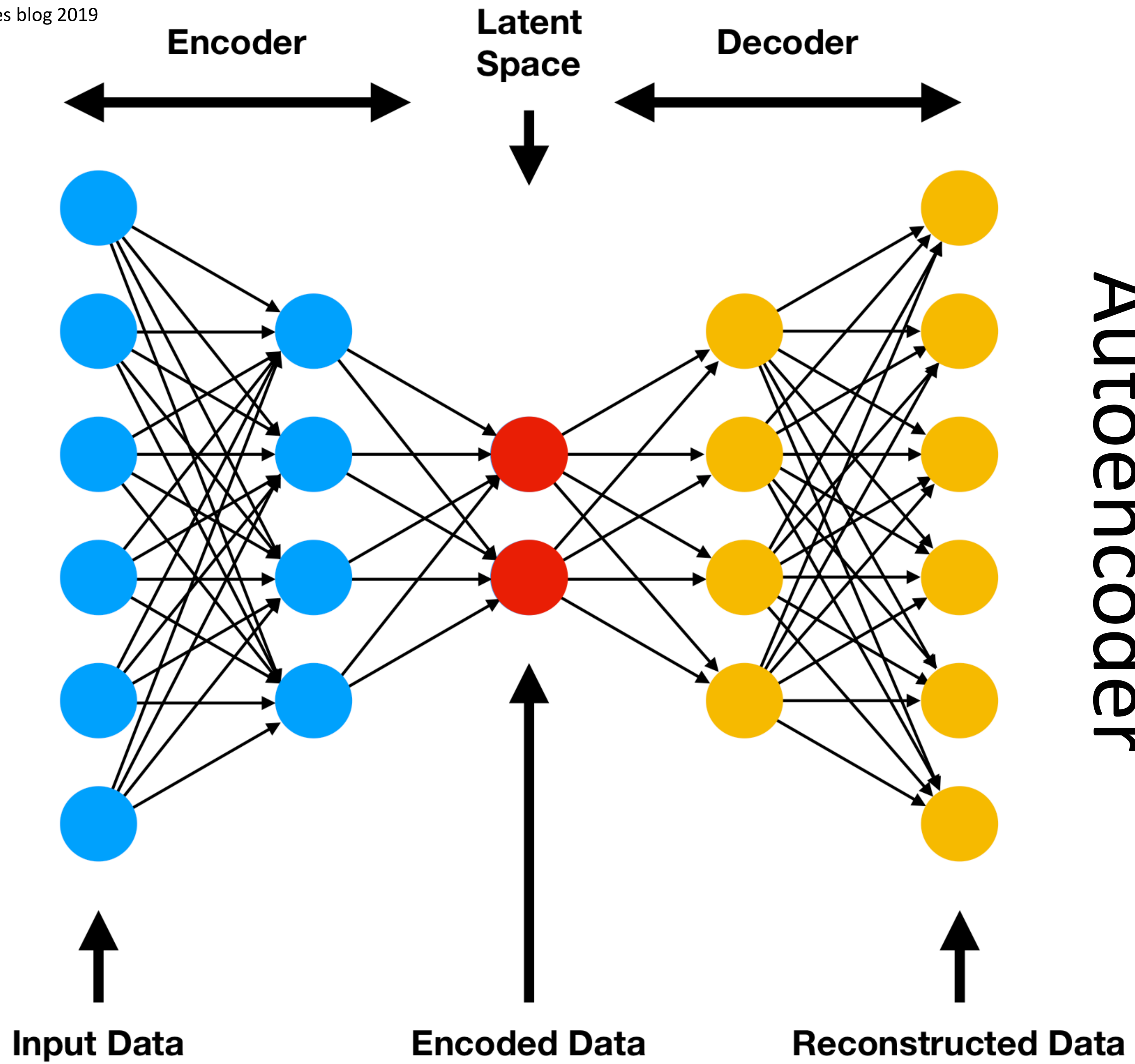


Image: Steven Flores blog 2019



Autoencoder