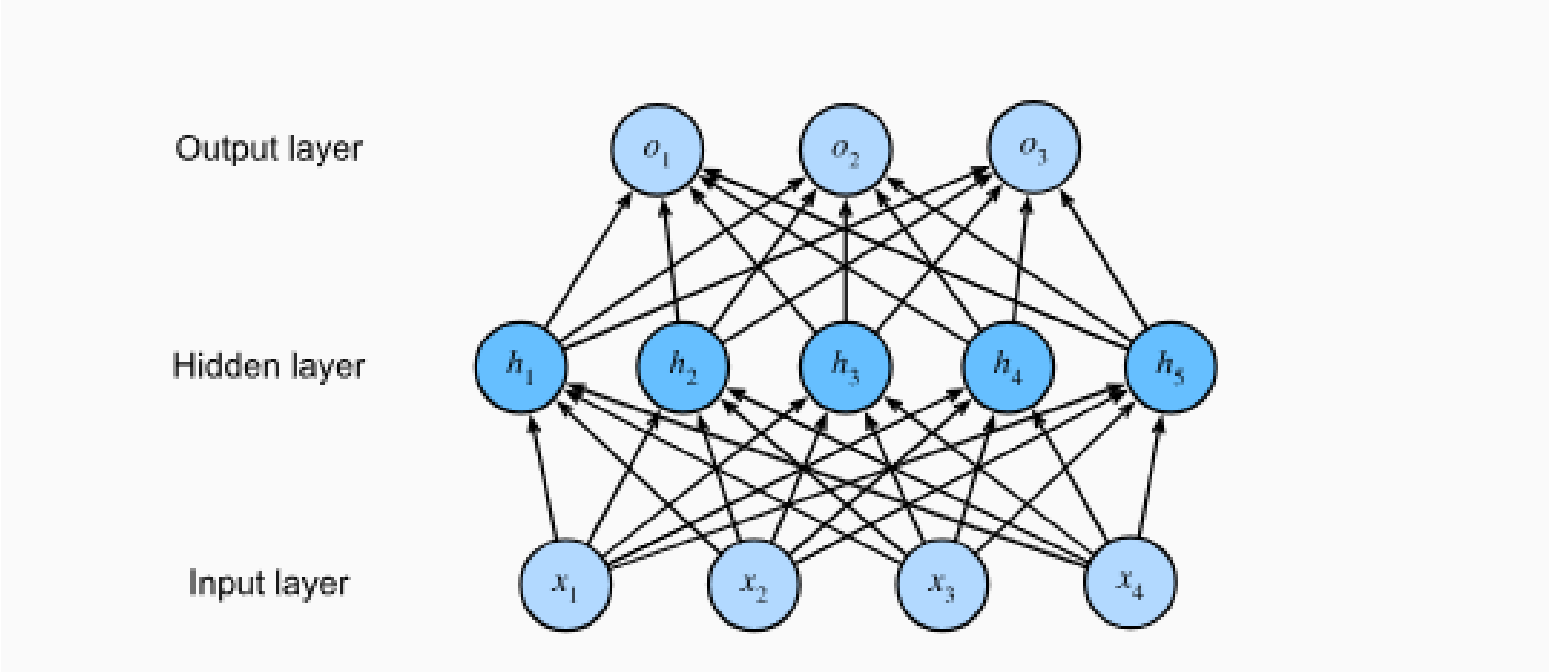
Medical Applications of Neural Networks

- Image interpretation
- Predictive models for clinical outcomes
- Automated pathology slide analysis
- Natural Language Processing (NLP) for clinical notes
- Drug discovery & genomics applications

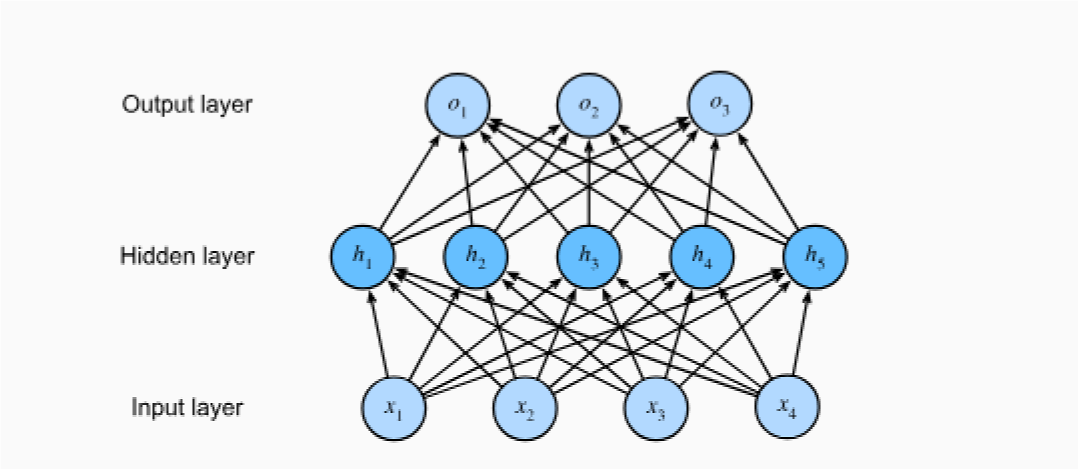
What Is a Neural Network?

- Pattern recognition inspired by the brain



Role of the Layers

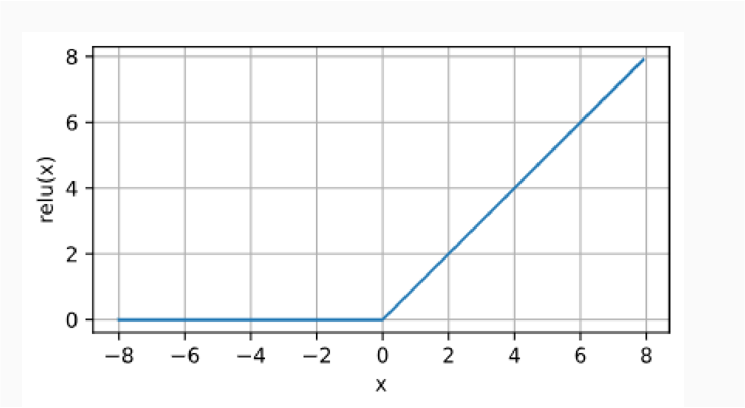
- Typically there are many many more layers (1000s) than input variables (features)
- And very few output layers (sometimes only one)

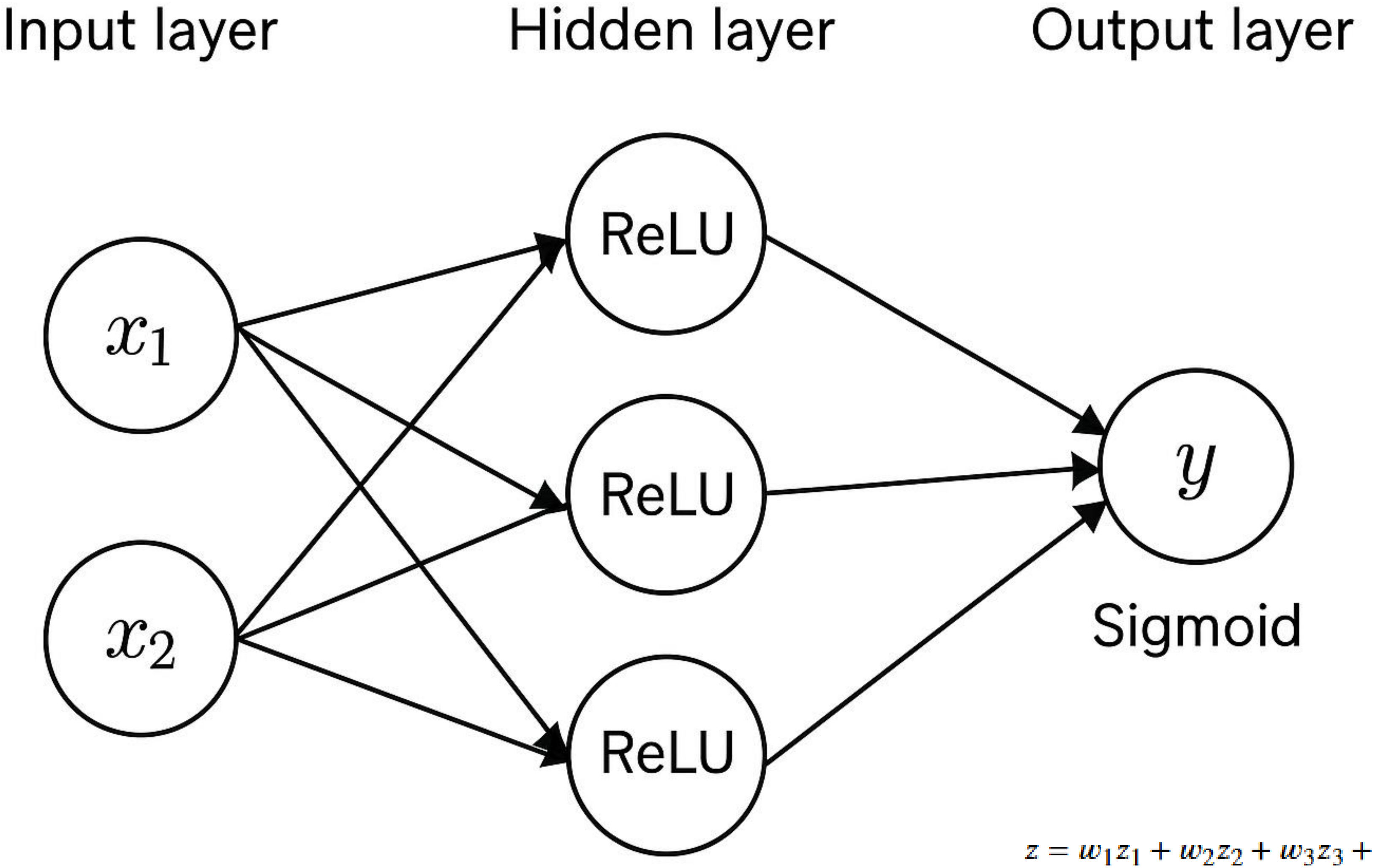


Neural Networks Conceptually

- Each arrow in the network is a transform
  - i.e. a possibly nonlinear function that takes input values from the layer below and returns a value for the layer above
- A popular form is ReLU which mimics neurons firing

- ReLU (rectified linear unit) function





$$z = w_1 z_1 + w_2 z_2 + w_3 z_3 + b$$

z1 = max{0, (x1 * v11) + (x2 * v12) + b1}
z2 = max{0, (x1 * v21) + (x2 * v22) + b2}
z3 = max{0, (x1 * v31) + (x2 * v32) + b3}

$$y = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Training a Neural Network

- Using data to estimate (learn) the coefficients (weights)
- It involves taking derivatives of the nonlinear functions (ReLUs and the sigmoid in our example)
  - Backpropogation: Automatic way of doing this
- Then minimizing a loss function (like prediction error) using these derivatives
  - Gradient descent: A popular set of algorithms to do this efficiently

Universal Approximation

- There is some theoretical work that says with enough hidden layers and neurons, a neural network can approximate any function arbitrarily well
- But it does not say that you can learn a network with finite amount of data

Key Architectures

- Feed-forward networks
- Convolutional neural networks
- Recurrent neural networks
- Encoder/Decoder
- Transformers
- Generative models

Feed-Forward Networks (FFNNs)

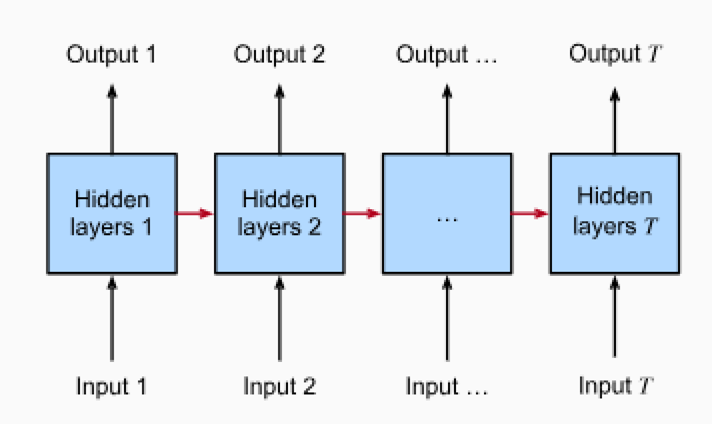
- The example we have is an FFN
- Input → Hidden Layers → Output
- A very general form of regression
- Basic risk prediction or classification architecture

Convolutional Neural Networks (CNNs)

- Good for working with images
- Inputs are not numbers anymore, but they are matrices representing images
- There are concepts of invariance and proximity
  - Invariance: A tumor’s location within the organ does not change the fact that it is a tumor
  - Proximity: Two pixels close to one another are likely to have similar values

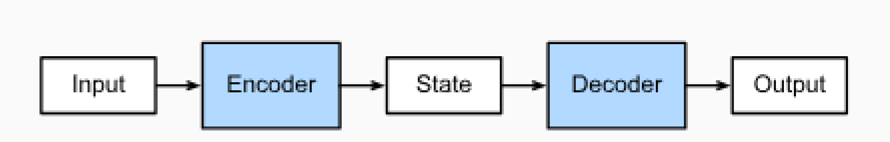
Recurrent Networks & Time Series

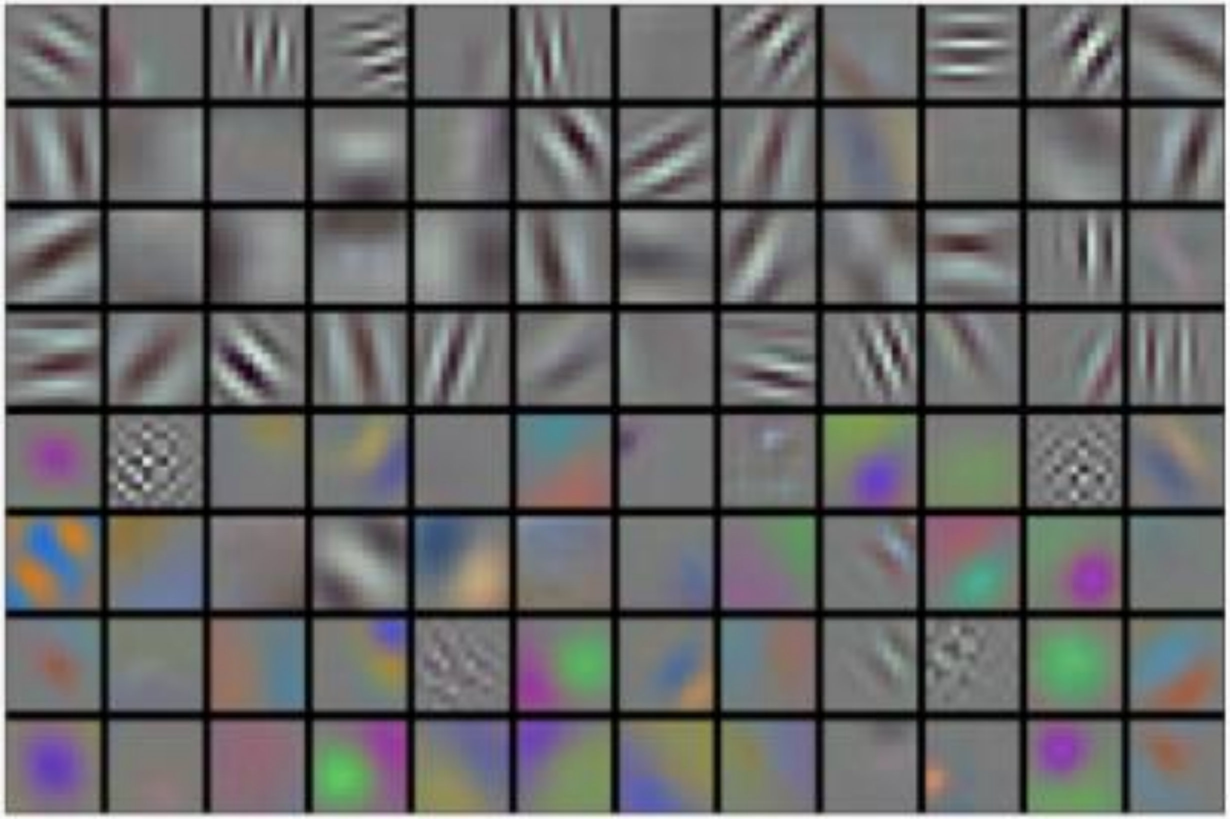
- When you have data over time
  - ECG
  - Longitudinally collected biomarkers



Encoder/Decoder Architecture

- Encode: Find a small number of latent features
- Develop a model based on these features
- Decode: Convert the encoded representation to the output scale





Transformers (Vision & Text)

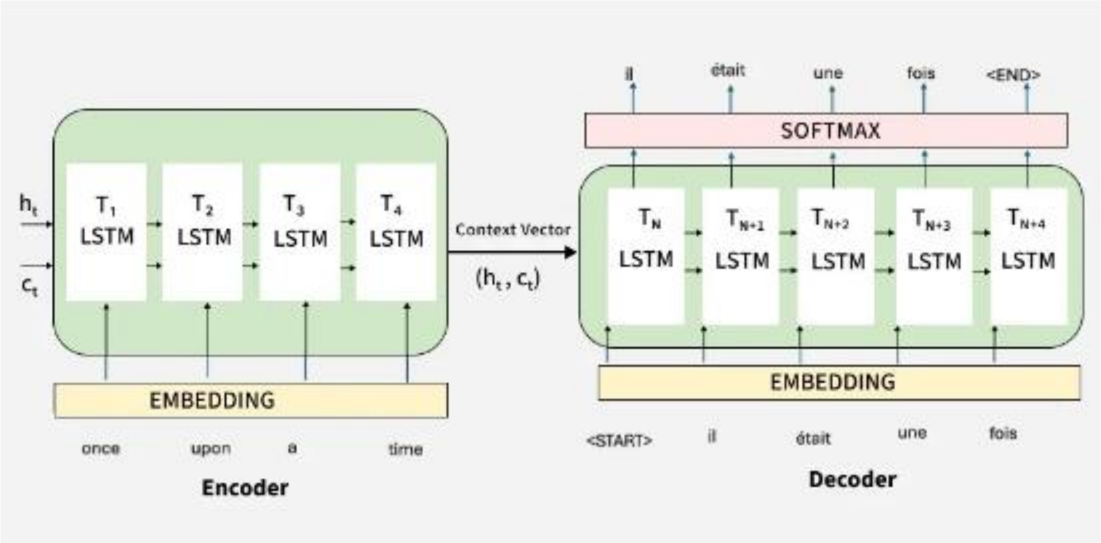
- Instead of processing input (like language or a set of images like a video) one (word) by one, process it as a whole
- This means uncovering relationships between words
- Which is accomplished through a relevance score for every pair of words

Example of Relevance

- The man refused to cross the field because **it** was too muddy
- ”It” refers to man or field?
  - Generates a score between each word and “it”
  - Chooses the one with the highest score

More on transformers

- Since inputs are not read sequentially parallel processing is possible and easier to scale up (long texts and videos can be processed faster)
- Architecture behind generative (large language models)


il
était
une
fois
<END>
SOFTMAX
h_t
c_t
T_1 LSTM
T_2 LSTM
T_3 LSTM
T_4 LSTM
Context Vector
(h_t, c_t)
T_N LSTM
T_{N+1} LSTM
T_{N+2} LSTM
T_{N+3} LSTM
T_{N+4} LSTM
EMBEDDING
once
upon
a
time
Encoder
EMBEDDING
<START>
il
était
une
fois
Decoder

Importance of Evaluation

- All of these architectures tend to overfit
- Held-out validation sets are essential
- Typical metrics
  - Sensitivity / specificity
  - ROC / AUC
  - Calibration curve
- Clinical input in model development and oversight in implementation

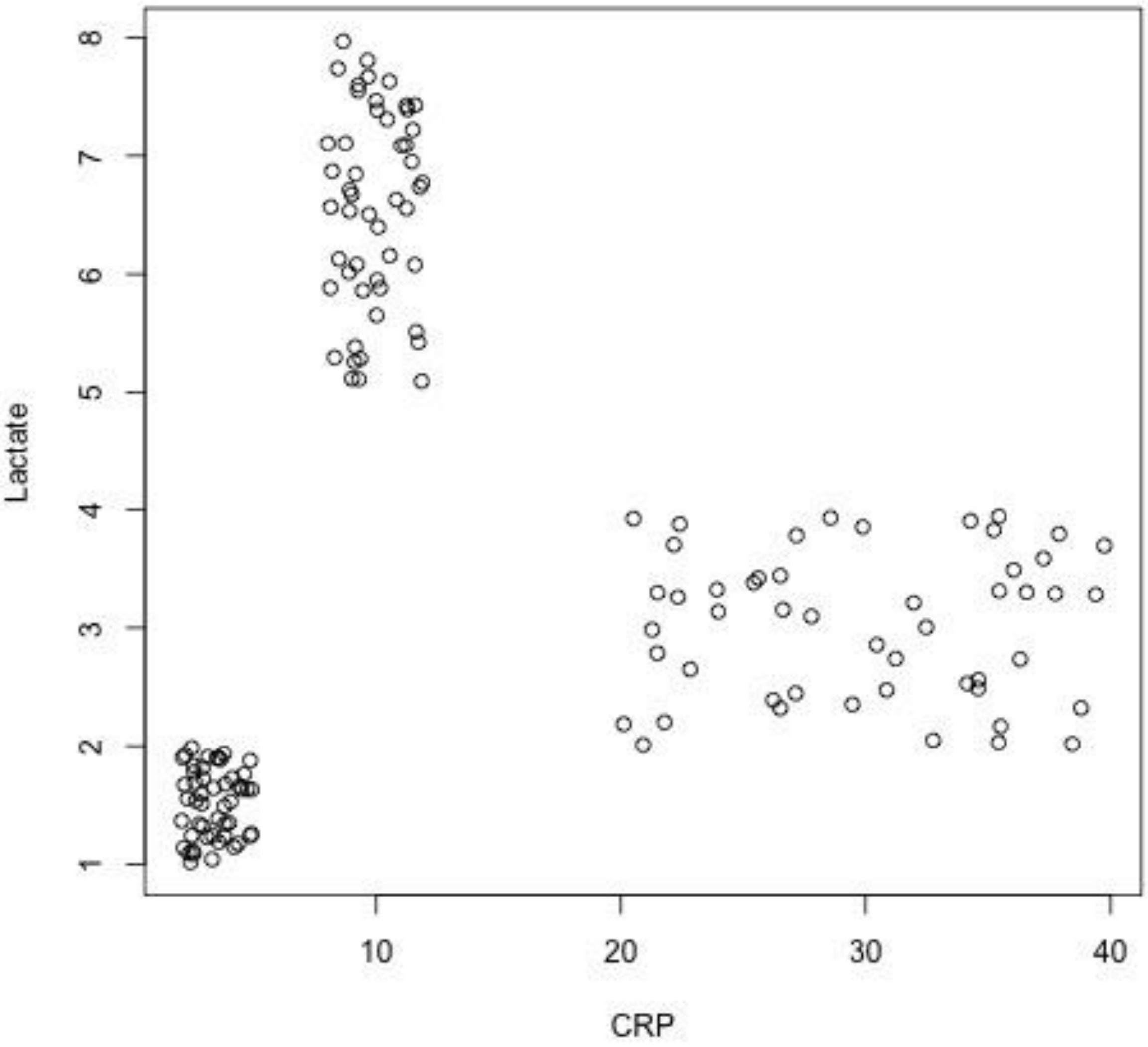
Unsupervised Learning

- Everything we learned so far (regression, random forests, neural networks) is supervised learning
- Unlabeled data (we do not know the outcome, we only know the covariates)
- Can we still learn something?

Two Types of Unsupervised Learning

- Clustering
  - Find subsets of data where the covariate distributions are similar
  - Concept of distance (dissimilarity) is key
- Dimensionality Reduction
  - Find a small number of latent variables that retain most of the information

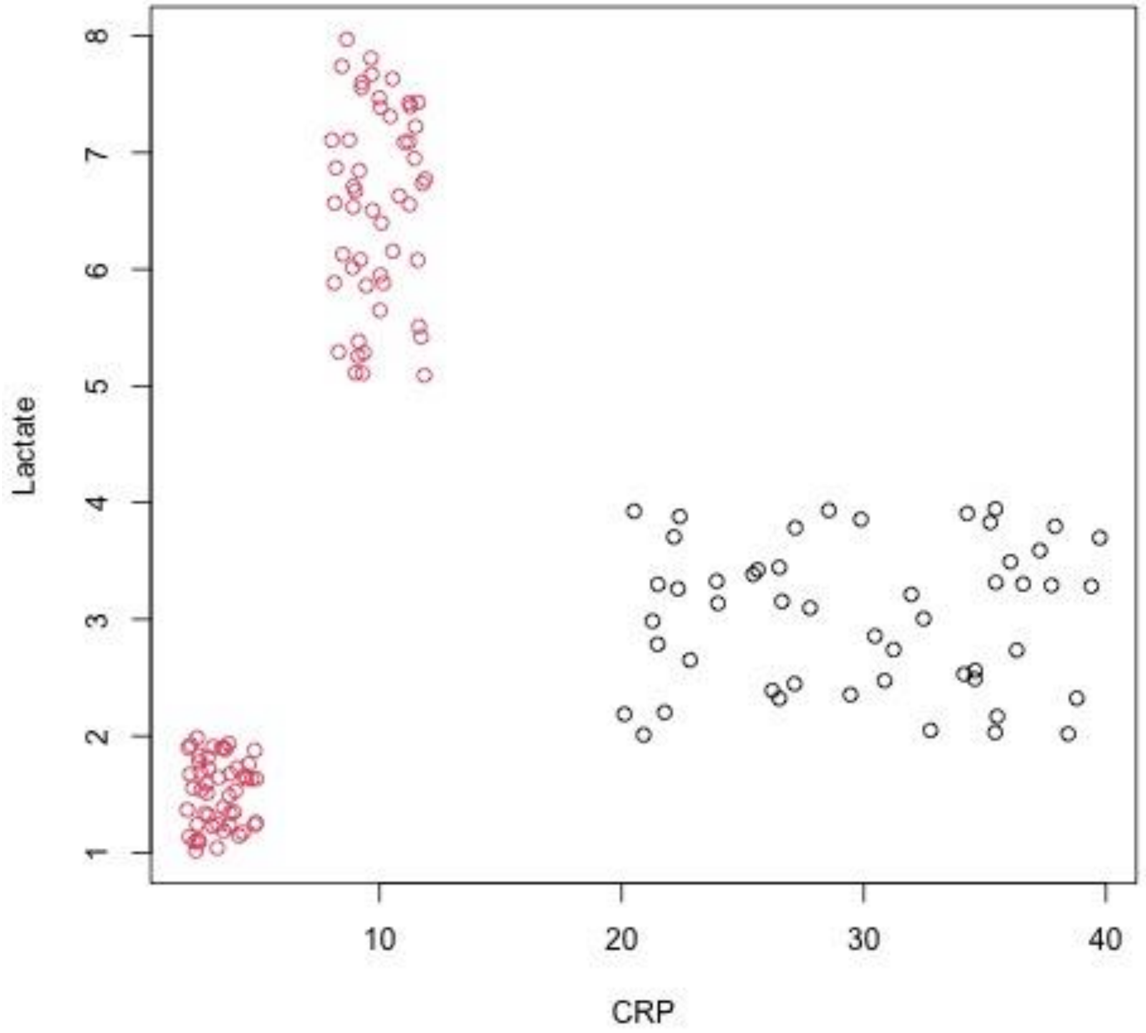
Clustering Intuition: Grouping similar patients



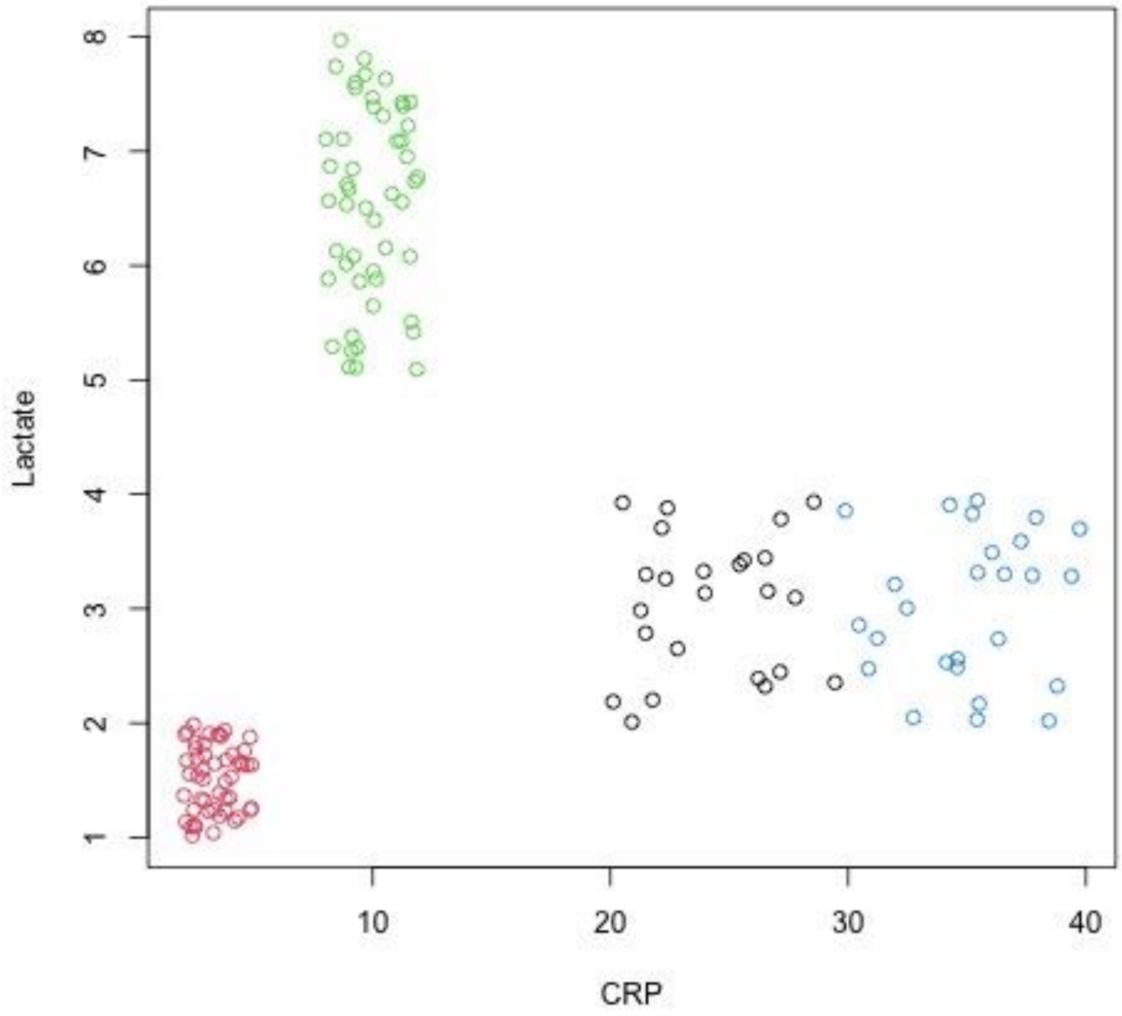
k-Means

- Pre-specify the number of clusters
- Randomly choose cluster centers
- Assign each patient to the nearest center
- Recalculate cluster centers
- Reassign patients
- Continue until no patient is reassigned to a different cluster

2 Clusters



4 Clusters

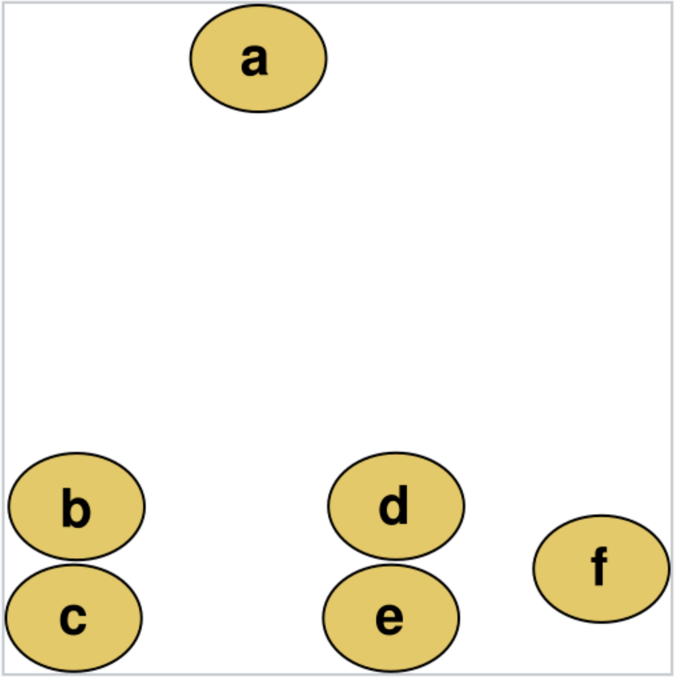


K-Means Summary

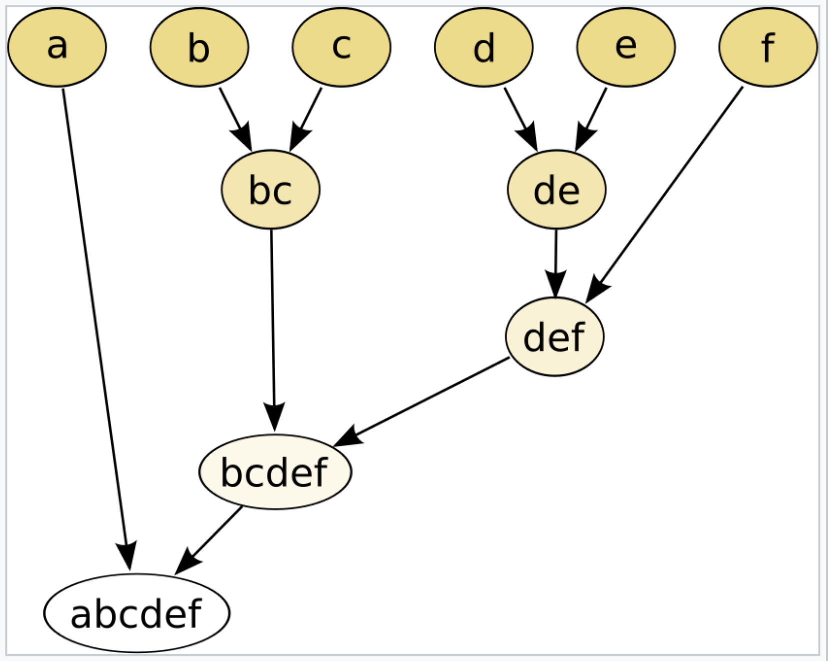
- Nonparametric, no distributions assumed
- Sensitive to the pre-specified number of clusters (and sometimes starting points)
- Iterative nature means convergence to local solutions are possible
- Grouping similar patients
- Easy to scale up, flexible with choice of distance (one can cluster images for example)

Hierarchical Clustering

- Calculate the distance between all pairs of patients
- Join the two that are closest and make one patient out of them
- Keep going until all patients are joined
- One needs the concept of “linkage” after patients are merged into one



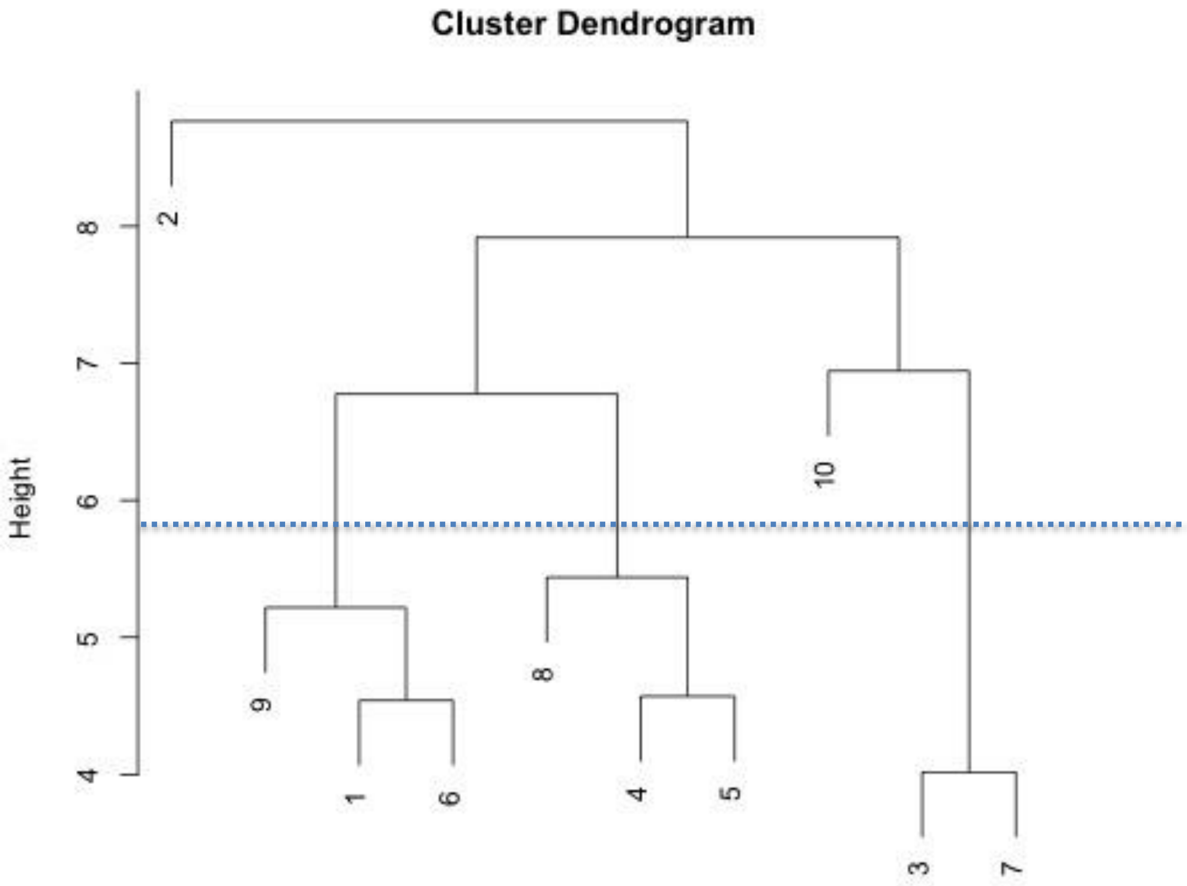
Raw data



Hierarchical Clustering

- This is known as agglomerative, alternative you can do divisive
- Start all patients as one, then begin peeling away individual patients
- No evidence that one is better than the other
- Linkage is way to calculate distance between sets of patients (merged patients)

Hierarchical Clustering



One can "cut" the tree at a given "height"

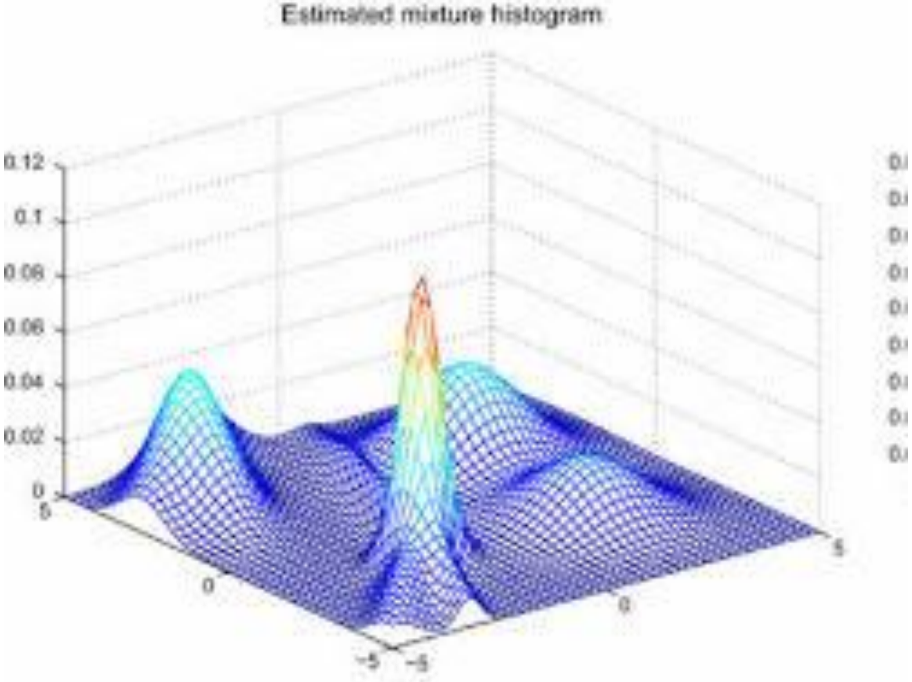
No best method to choose the height

K-means vs Hierachical

- HC + Dendogram works best in small data sets with high dimensional data (like gene expression)
- K-means works well in most settings but sensitive to the pre-specified number of clusters
- Both are exploratory data analysis tools

Density based clustering

- Sometimes called mixture models
- We start by assuming not only a number of clusters but also a specific probability density for the distribution of observations in each cluster



(a)

Density-Based Clustering

- Works well with big data sets, arbitrarily shaped clusters
- Does not work well with categorical data (true to some extent for K-means and HC as well)
- Requires some prior knowledge, less exploratory than the other two methods of clustering

Dimensionality Reduction

- Can we represent a collection of variables with a smaller set of derived (latent) variables?
- This way we can analyze/visualize the smaller set
- Remember the encoder/decoder

Principal Components (PCA)

- Consider all possible linear combinations (w1*x1 + w2*x2 …) of the raw variables
- Find the that retains most of the original information in the raw variables, this is your first PC
- Do the same but limit yourself to linear combinations that are orthogonal to (independent of) PC1

Dummy Example

- Three variables: Test Score 1, Test Score 2, Hours of Study
- PC1 = 0.71*Score1 + 0.68*Score2 + 0.11*Hours
- PC2 = 0.17*Score1 + 0.21*Score2 + 0.94*Hours
- PC3 = 0.64*Score1 + 0.66*Score2 + 0.23*Hours

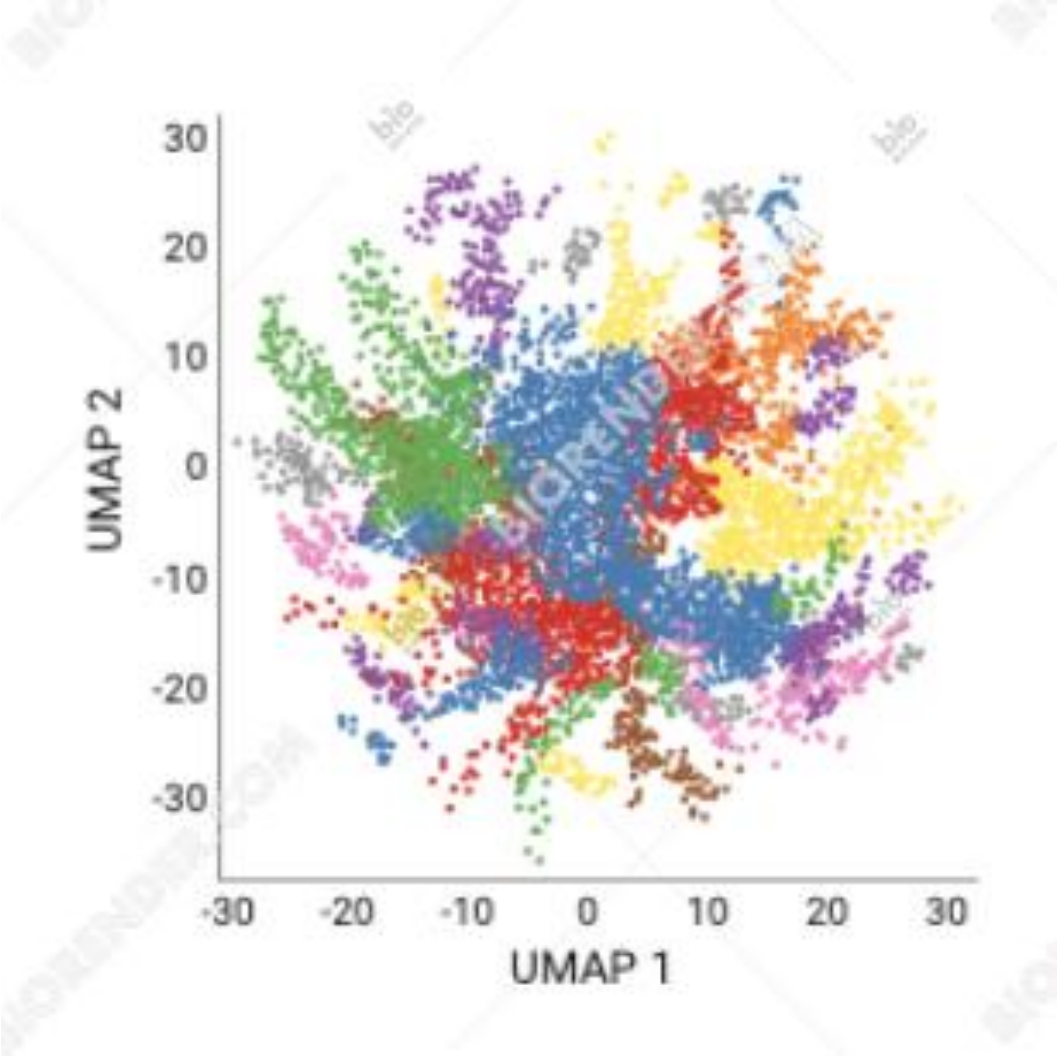
Variance Explained

- Also called information retained
  - PC1 = 0.71
  - PC2 = 0.24
  - PC3 = 0.05
- Rule of thumb: choose the first PCs that explain 80% of the variance

Nonlinear dimension reduction

- PC is linear
- t-SNE and UMAP are modern alternatives to PCA
- UMAP is more recent and more popular
- Nonlinearity means a lot of tuning parameters, results easy to manioulate

Clustering + Dimension Reduction


UMAP 2
UMAP 1