

Feature learning using matrix factorization and neural networks

Miami Machine Learning Meetup
January 2018

<https://github.com/rikturr/mml-feature-learning>

* record

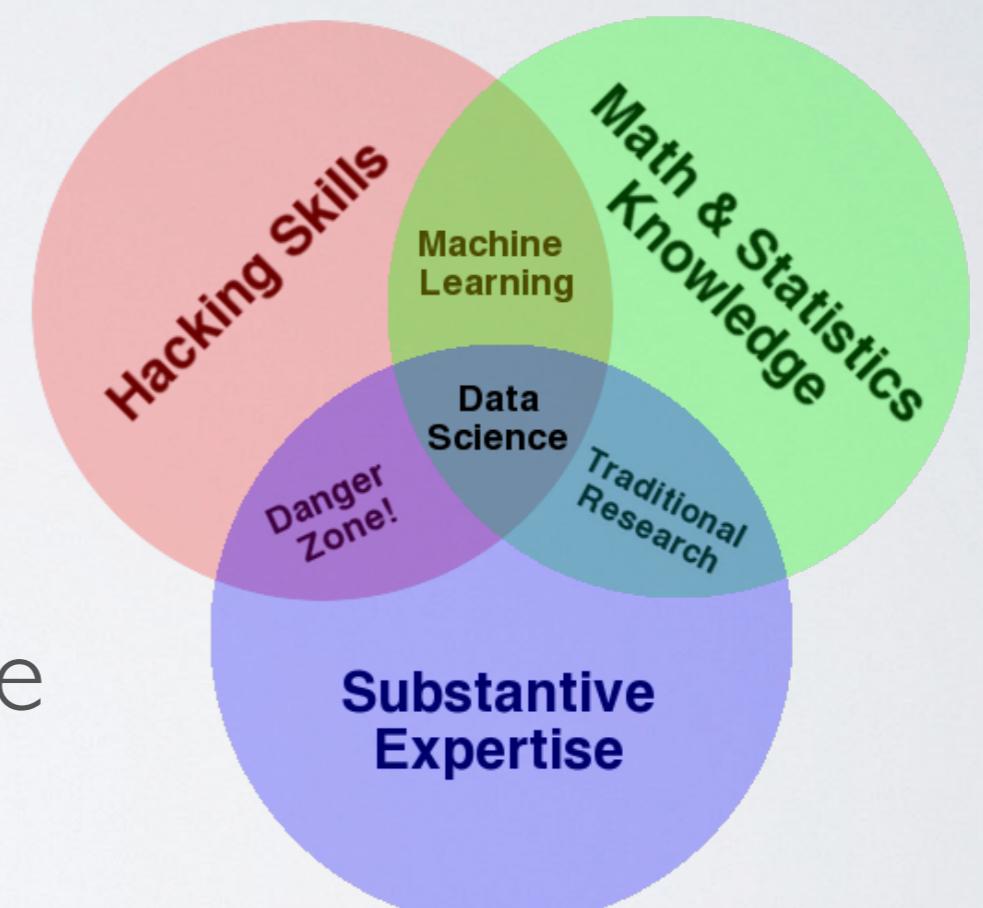


MODERNIZING
MEDICINE



FLORIDA ATLANTIC
UNIVERSITY

- Aaron Richter
- Data Scientist
 - Thrive in the danger zone
- PhD Candidate, Computer Science
-  @rikturr  rikturr@gmail.com



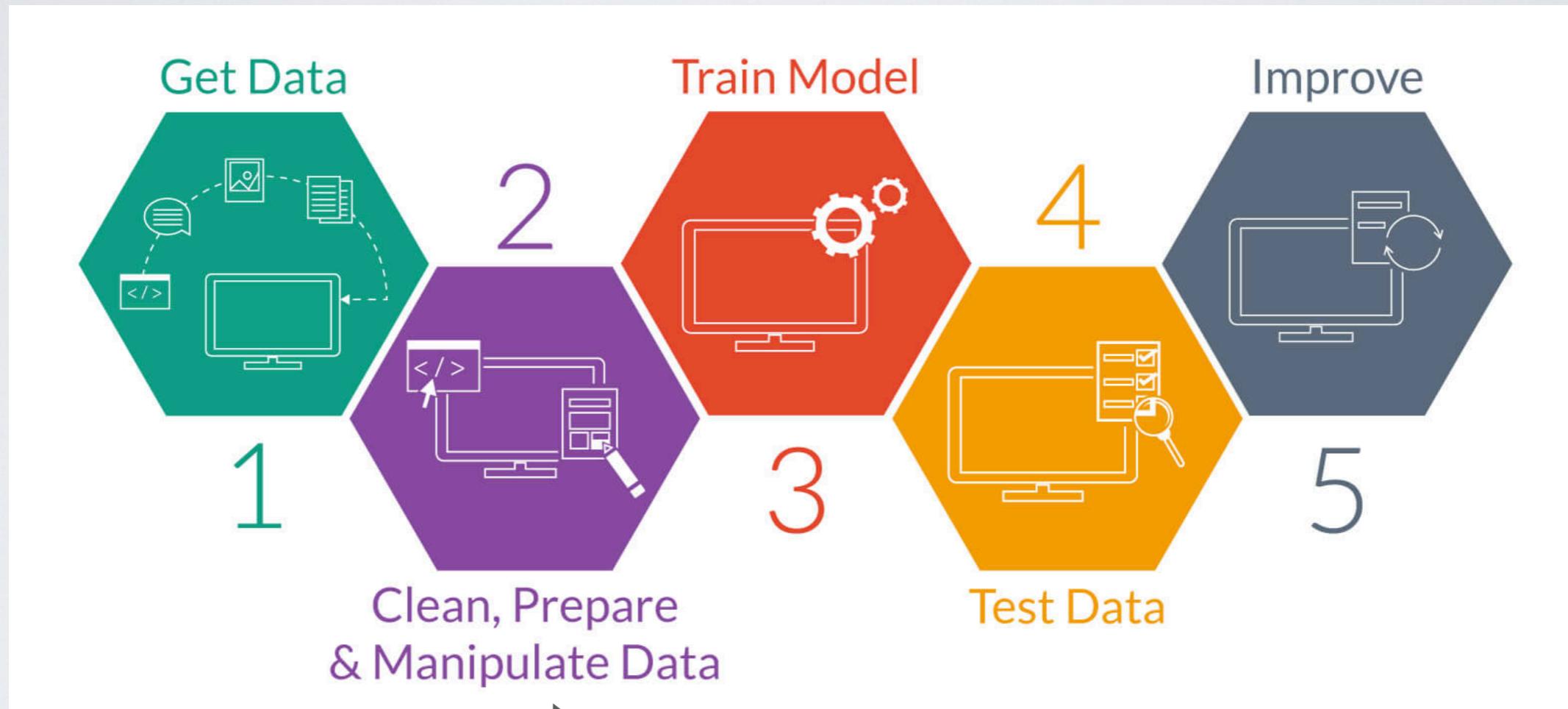
What we'll cover

- What is feature learning?
- Why feature learning?
- How to do feature learning?
 - Matrix Factorization
 - Neural networks
- Code samples

What is feature learning?

- Automatically extracting features from raw data using machine learning techniques
- Create latent feature space for input to predictive models
- Also known as feature embedding, feature extraction, representation learning

Why feature learning?



Feature engineering

Why feature learning?

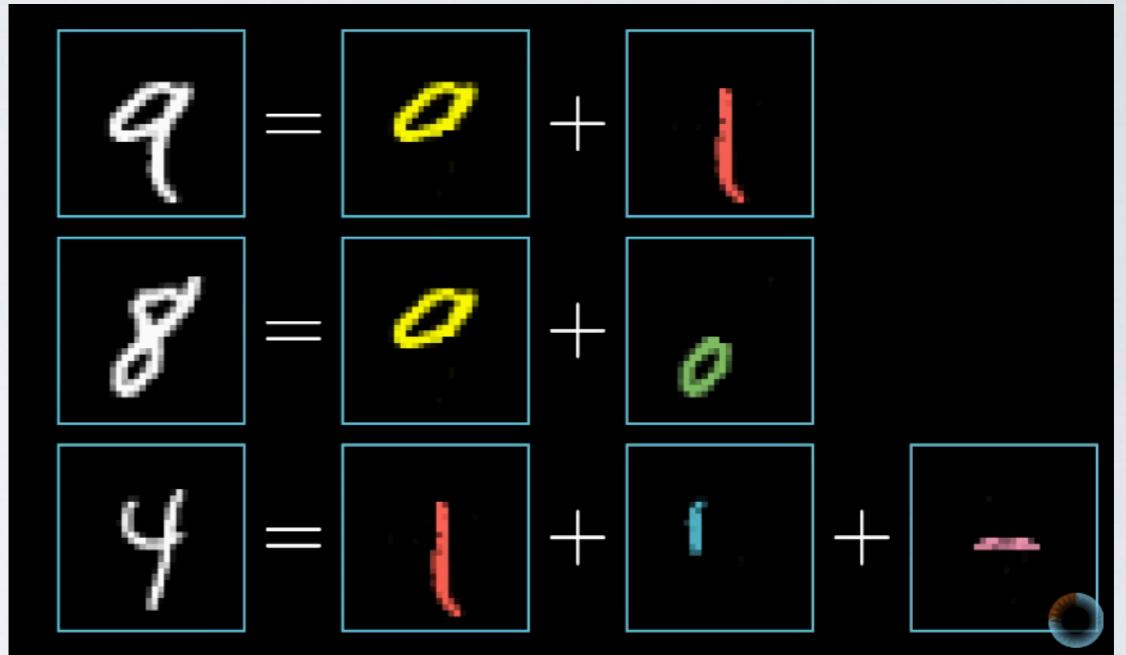
- Feature engineering is hard!
- “Coming up with features is difficult, time-consuming, requires expert knowledge. Applied machine learning is basically feature engineering.”
 - Andrew Ng

https://en.wikipedia.org/wiki/Feature_engineering

<https://forum.stanford.edu/events/2011/2011slides/plenary/2011plenaryNg.pdf>

Example applications

- Computer vision
- Text mining
- How do you tell the computer what to look for?



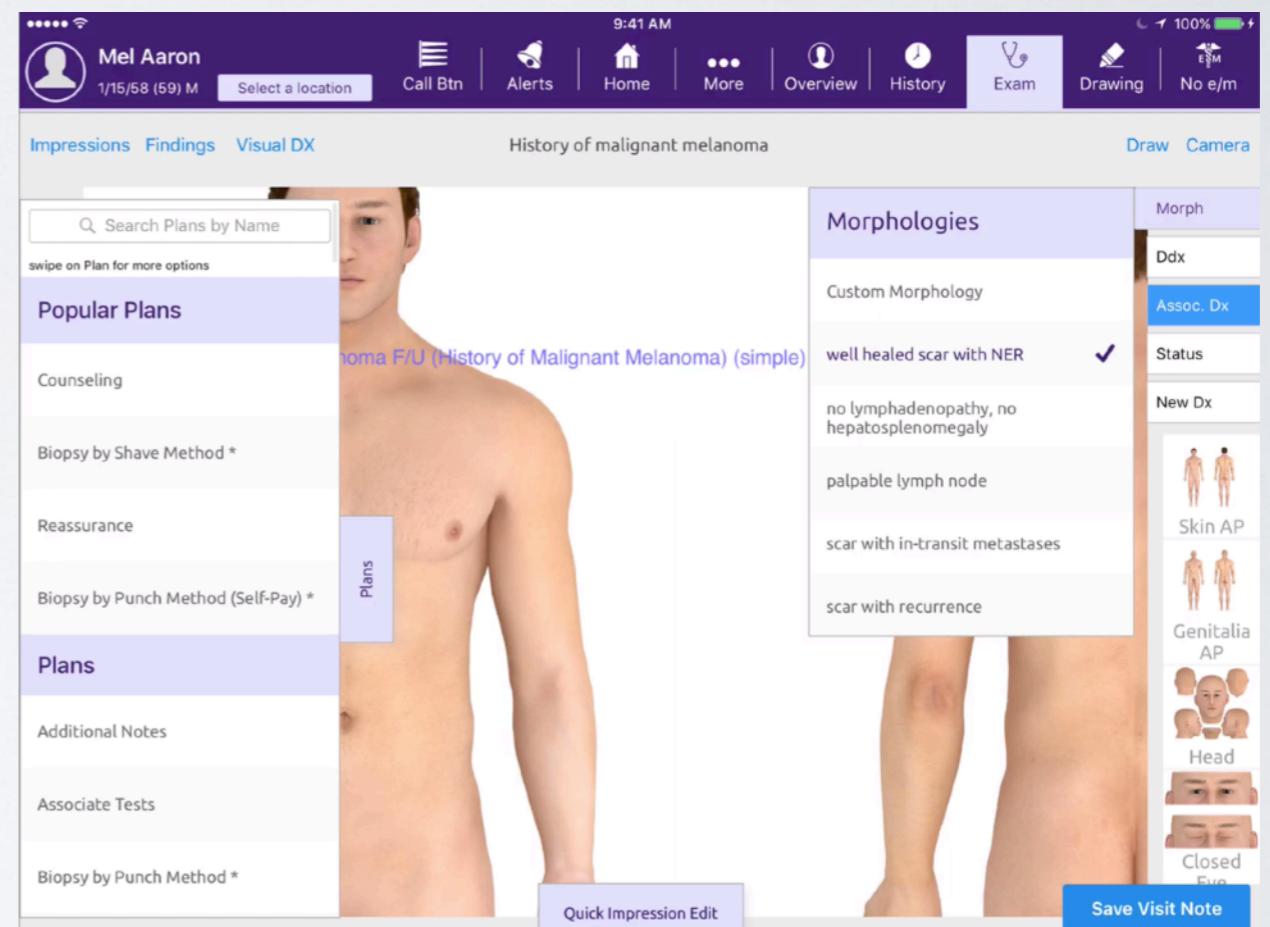
Source Text	Training Samples
The quick brown fox jumps over the lazy dog. →	(the, quick) (the, brown)
The quick brown fox jumps over the lazy dog. →	(quick, the) (quick, brown) (quick, fox)
The quick brown fox jumps over the lazy dog. →	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown fox jumps over the lazy dog. →	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

<https://www.youtube.com/watch?v=aircArUvnKk&t=25s>

<http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>

Example applications

- Predicting sentinel node status in melanoma from a real-world EHR dataset
(Richter et al., 2017)
- 30 engineering features / 100,000s of raw features
- 5,000 labeled instances / 2,000,000+ unlabeled instances
- 6 month project, 2 weeks of actual machine learning



Considerations

- Works best with high-dimensional “raw” data
 - Dimensionality reduction
- Model interpretability can be limited when using a latent feature space
 - Active area of research!

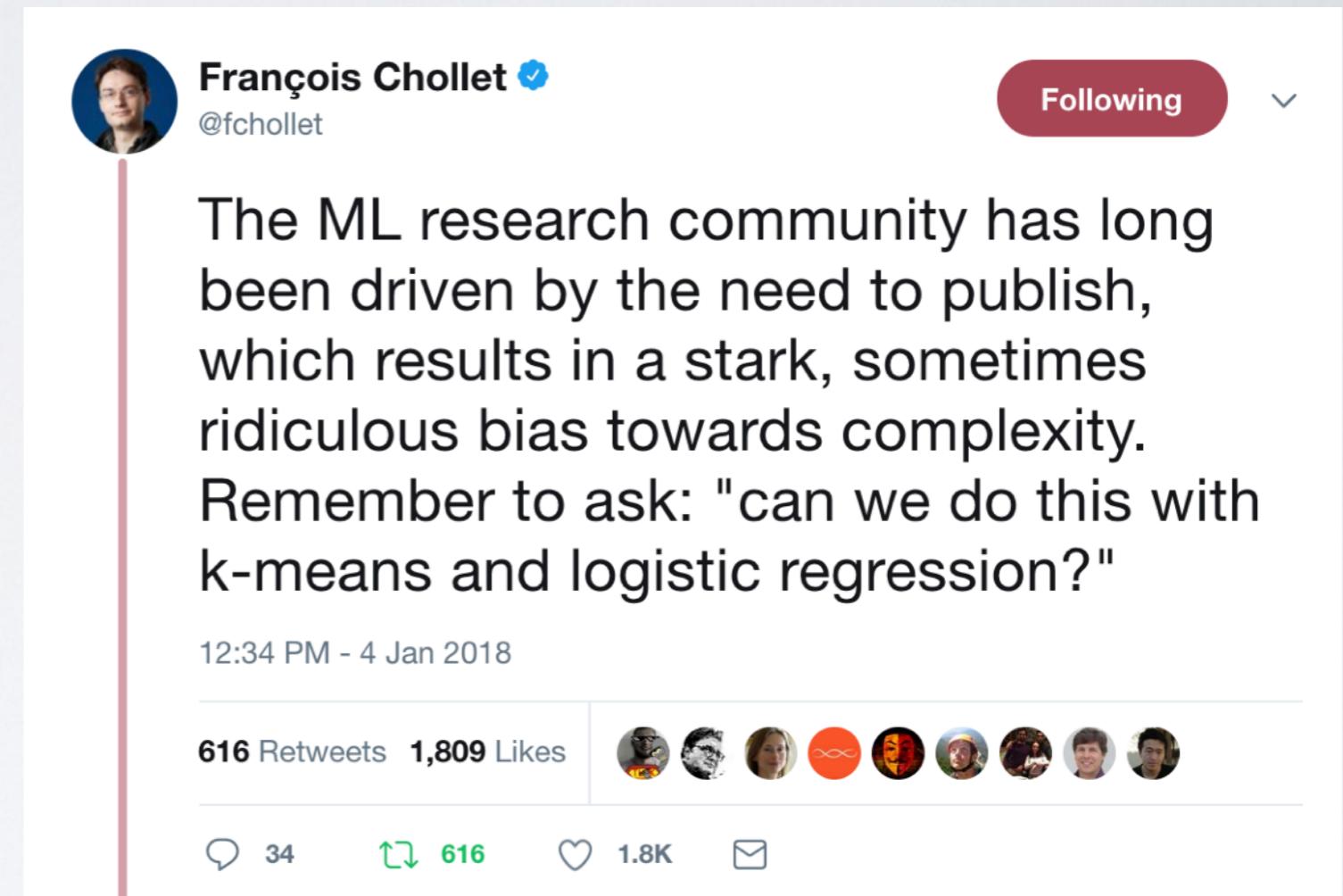
How to do feature learning?

- Unsupervised
 - Matrix factorization
 - Clustering
 - Neural networks (Autoencoders, Restricted Boltzmann machines)
- Supervised
 - Multilayer neural networks (hidden layers create embedding)
 - Transfer learning

https://en.wikipedia.org/wiki/Feature_learning

Aside

- More complexity != better ML
- Sometimes simple methods are good enough
- Don't need deep learning for everything!



François Chollet 
@fchollet

Following

The ML research community has long been driven by the need to publish, which results in a stark, sometimes ridiculous bias towards complexity. Remember to ask: "can we do this with k-means and logistic regression?"

12:34 PM - 4 Jan 2018

616 Retweets 1,809 Likes

34 616 1.8K

<https://twitter.com/fchollet/status/948970872802443264>

Matrix factorization

- (exactly what it sounds like)
- Also known as matrix decomposition
- Very efficient methods of dimensionality reduction
- Commonly used in ML
 - Eigendecomposition (spectral decomposition)
 - Singular value decomposition (SVD)

SVD

- Singular Value Decomposition
- One way to factorize a matrix
- Diagonal of Σ are the *singular values*

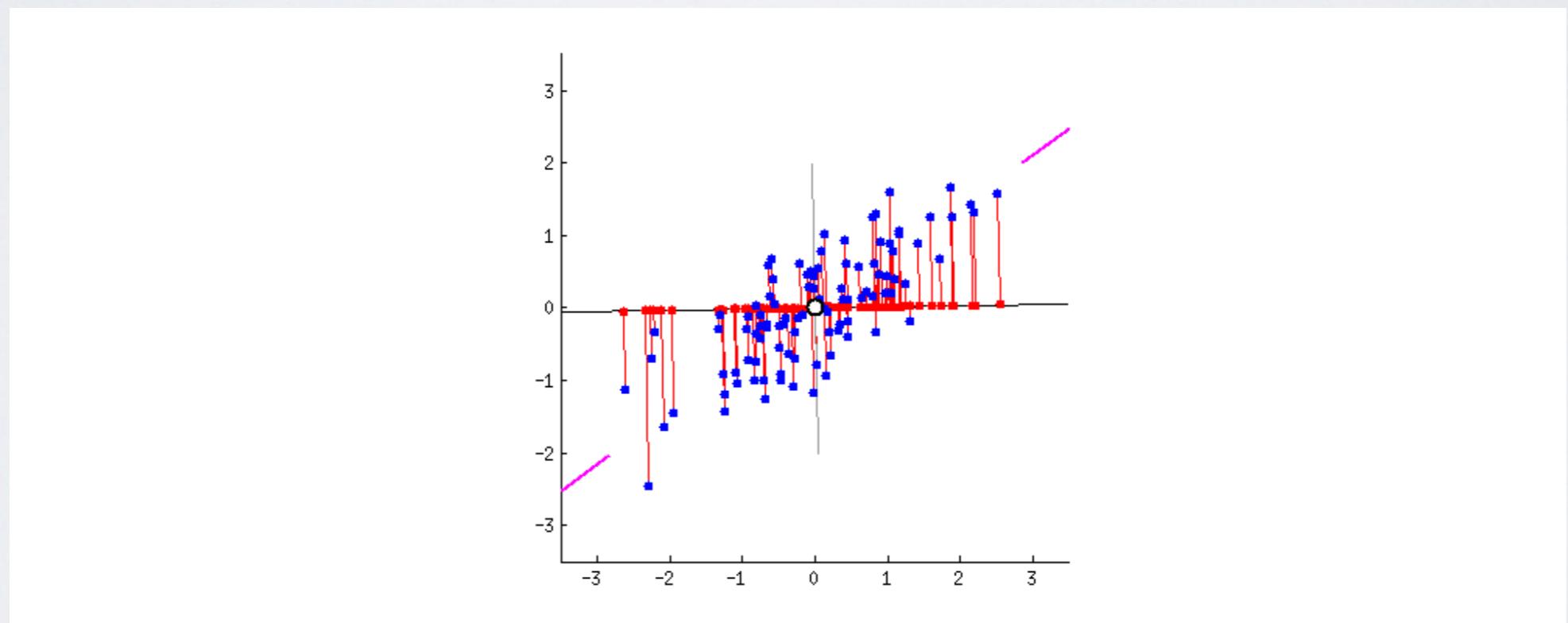
$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^*$$

where

- \mathbf{U} is an $m \times m$ **unitary matrix** over K (if $K = \mathbb{R}$, unitary matrices are **orthogonal matrices**),
- Σ is a **diagonal** $m \times n$ matrix with non-negative real numbers on the diagonal,
- \mathbf{V} is an $n \times n$ **unitary matrix** over K , and
- \mathbf{V}^* is the **conjugate transpose** of \mathbf{V} .

PCA

- Principal Component Analysis
- Find components (new features) that maximize variation between the input features
- Each subsequent principal component is orthogonal to the previous
 - First component captures the most variance



PCA

- Two ways to obtain:
 - (1) Eigenvalue decomposition of correlation matrix
 - (2) SVD of data matrix
- Must center and scale data!
 - Can use Truncated SVD if data is sparse
- General rule of thumb: (1) is more efficient (2) more numerically stable

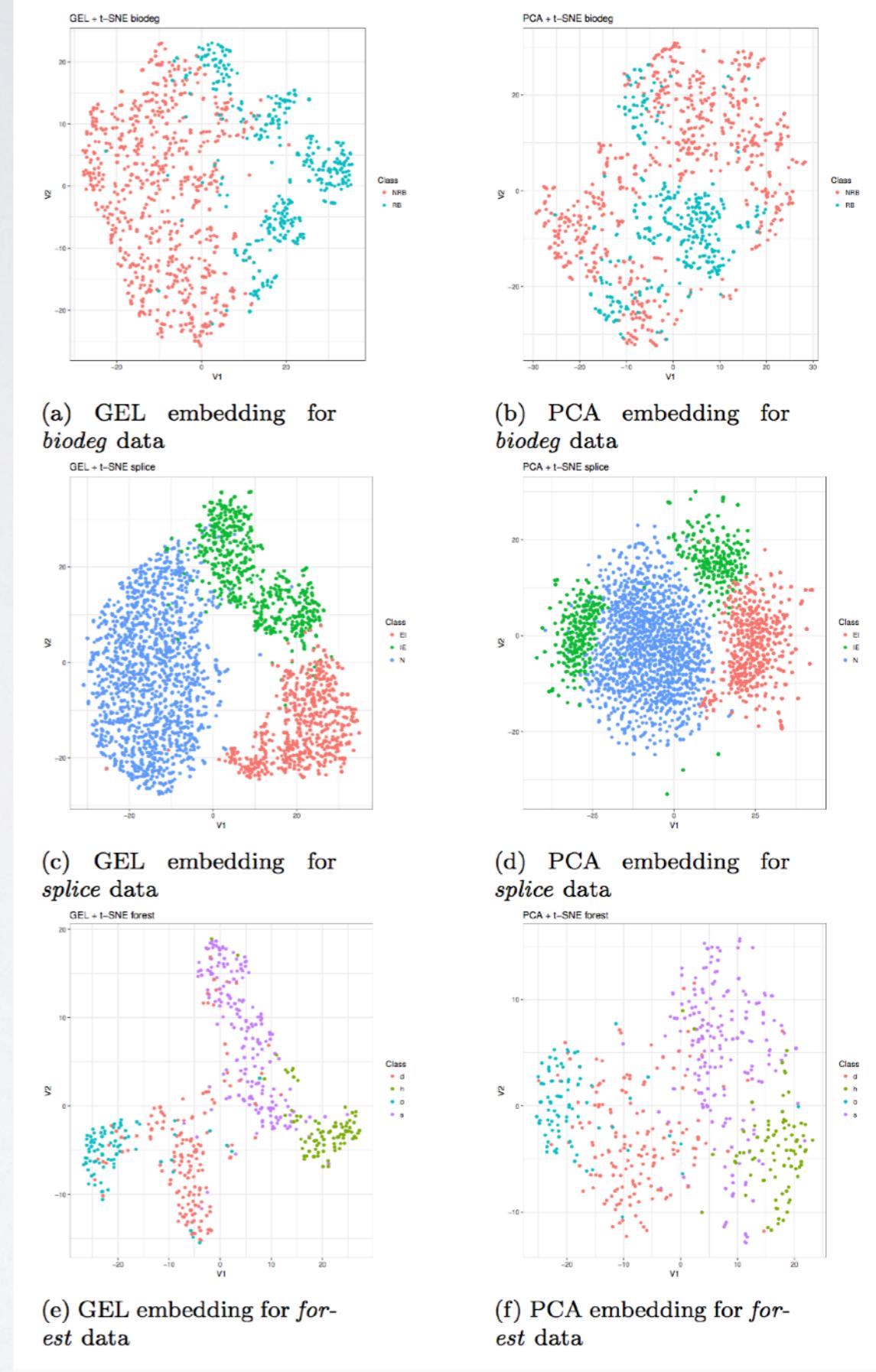
<http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html>

<https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca>

<https://stats.stackexchange.com/questions/79043/why-pca-of-data-by-means-of-svd-of-the-data>

GEL

- Generalized feature embedding learning
- Based on class partitioned instance representation and matrix decomposition
- Can be used in unsupervised or supervised manner
- New algorithm created by Eric Golinko (@egolinko)

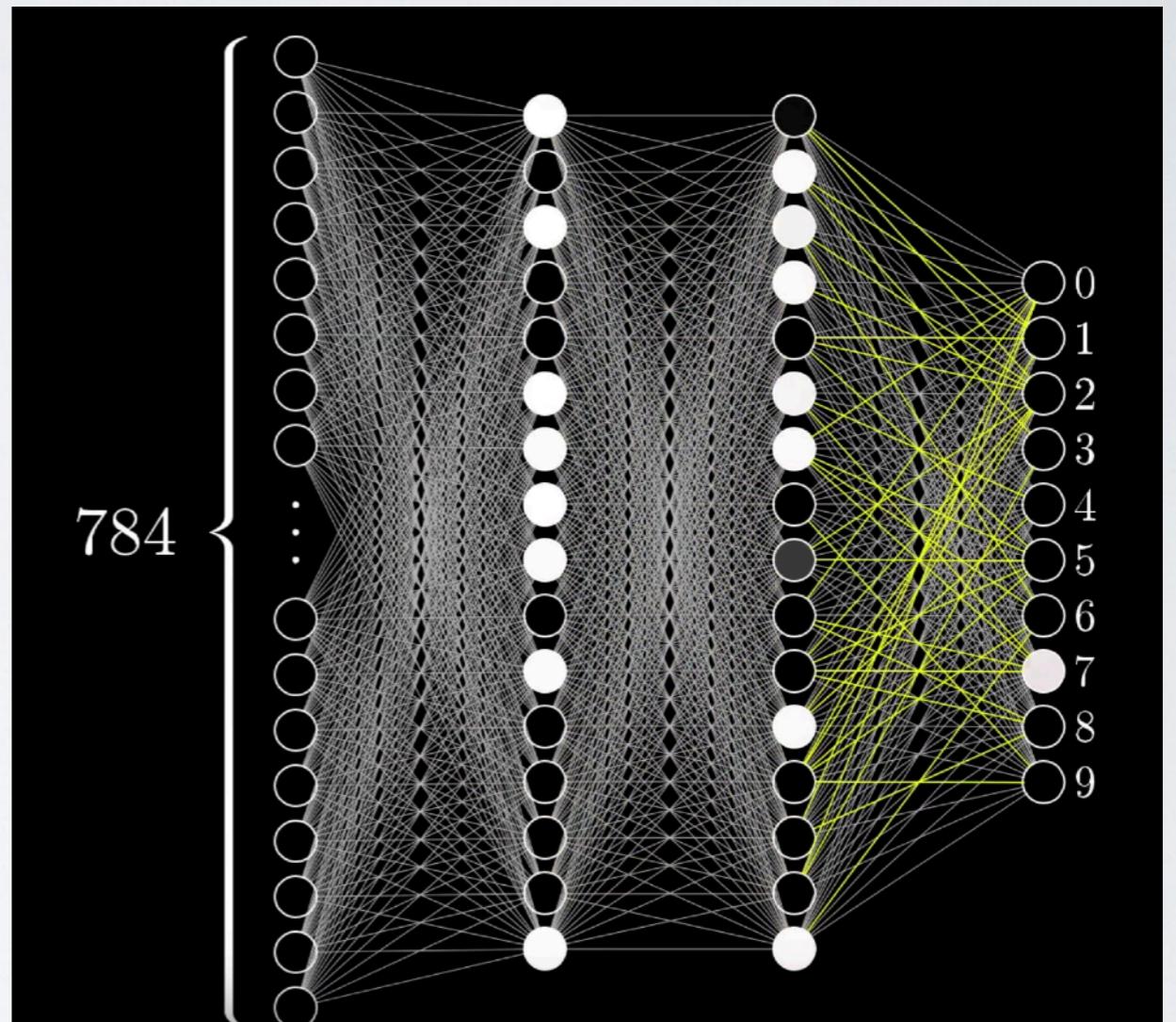


<http://ieeexplore.ieee.org/document/8102942/>

<https://github.com/egolinko/GEL>

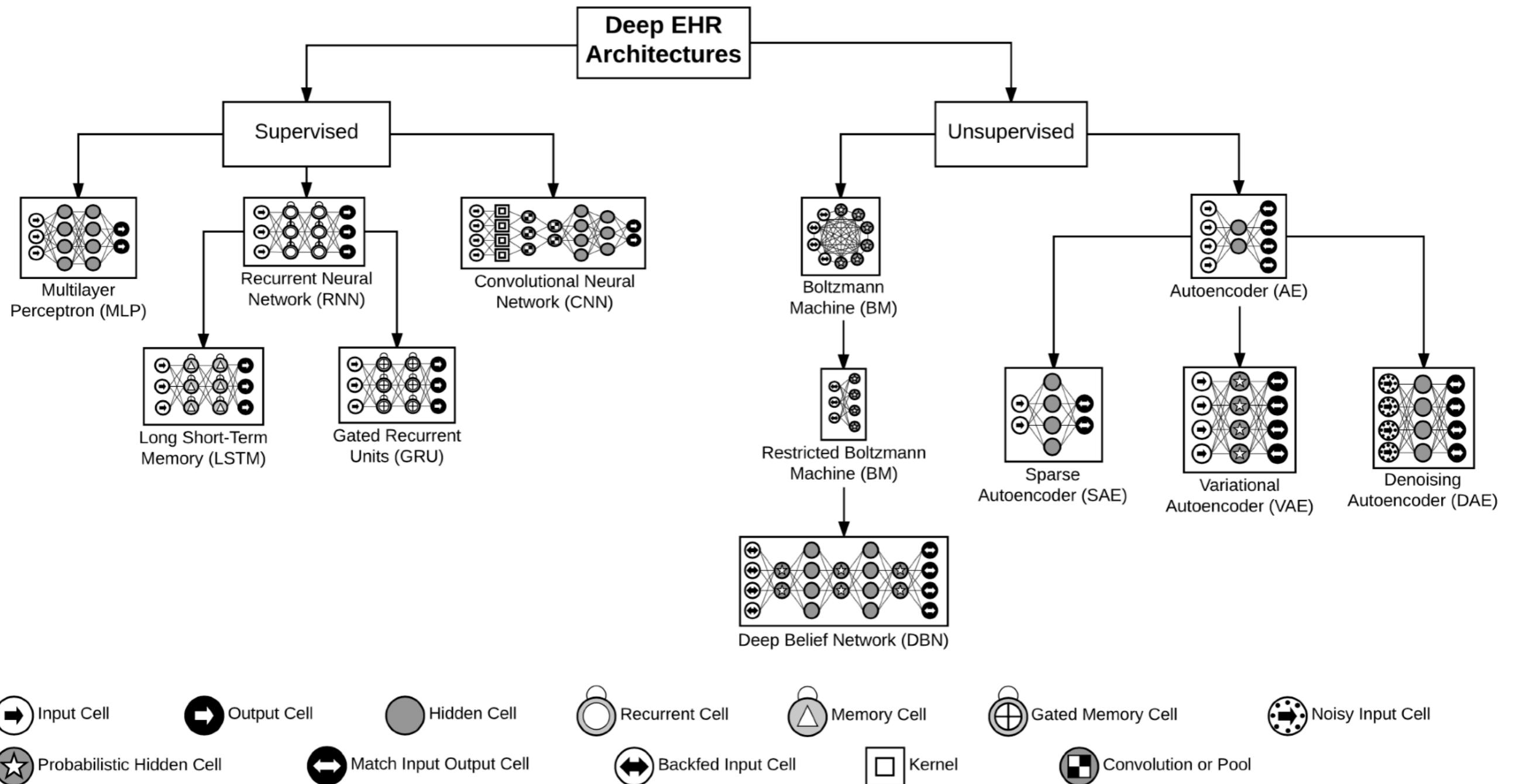
Neural Networks

- Big group of regression functions
- Output of each neuron is based on a non-linear function of inputs
- Learns by minimizing a cost function on the final (output layer) and setting weights through back propagation
- Multiple hidden layers = “deep learning”
- Really great intro to neural networks ->



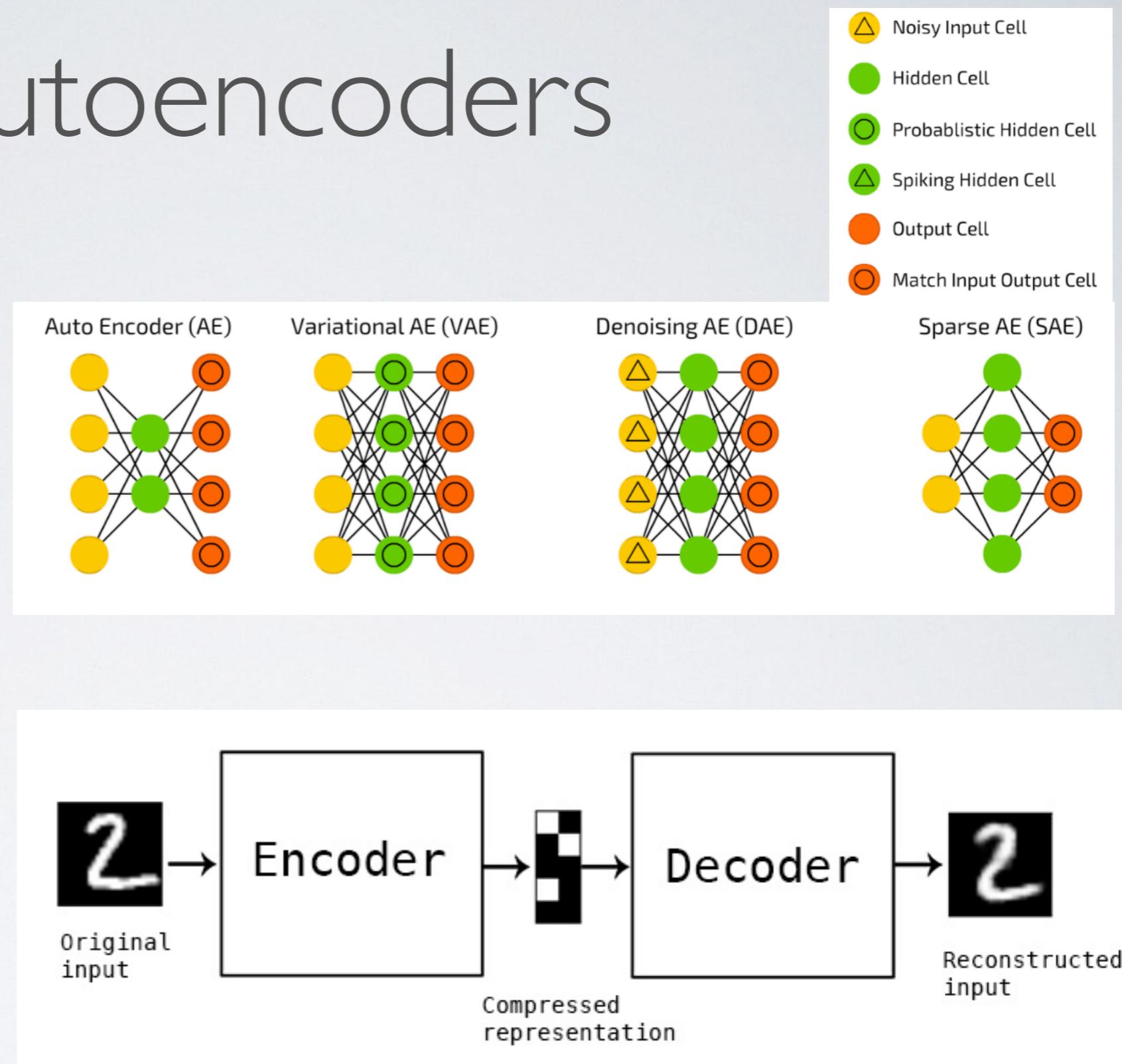
<https://www.youtube.com/watch?v=aircAruvnKk&t=374s>

Neural Networks



Autoencoders

- Reconstruct input
- Use intermediate layer as feature representation
- Useful when you have a large number of unlabeled examples

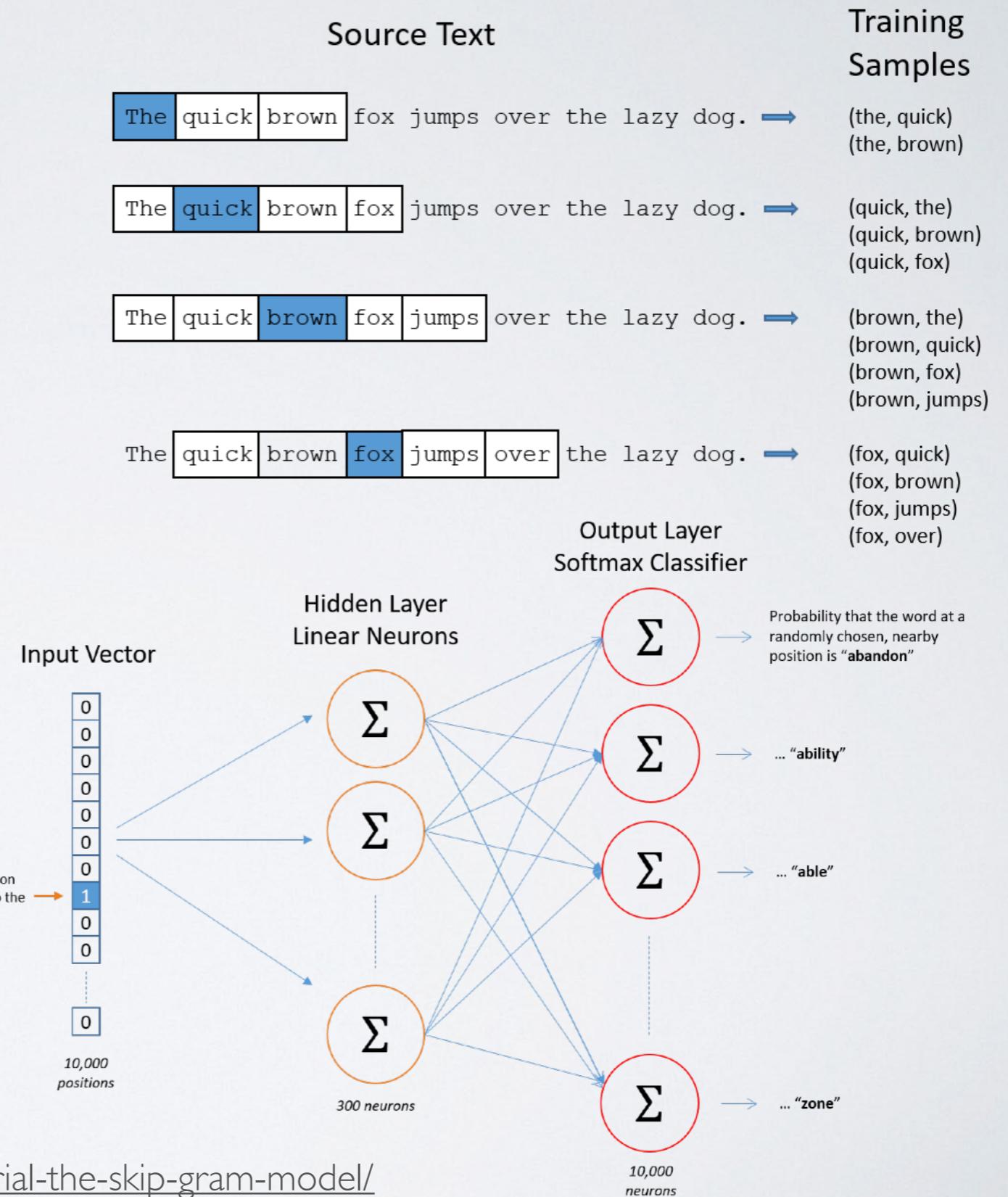


<http://www.asimovinstitute.org/neural-network-zoo/>

<https://blog.keras.io/building-autoencoders-in-keras.html>

Skip-gram / Word2Vec

- Model probability that words will be in proximity of a specific word
- Hidden layer creates word vectors (embeddings)
- Places words in coordinate space



<https://code.google.com/archive/p/word2vec/>

<http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>

Supervised + Transfer Learning

- Train a deep neural network on a supervised learning problem with a large number of labels
- Chop off output layer, attach new output layer for different dependent variable
- Results in pre-initialized weights, can freeze weights in certain layers when training on new problem

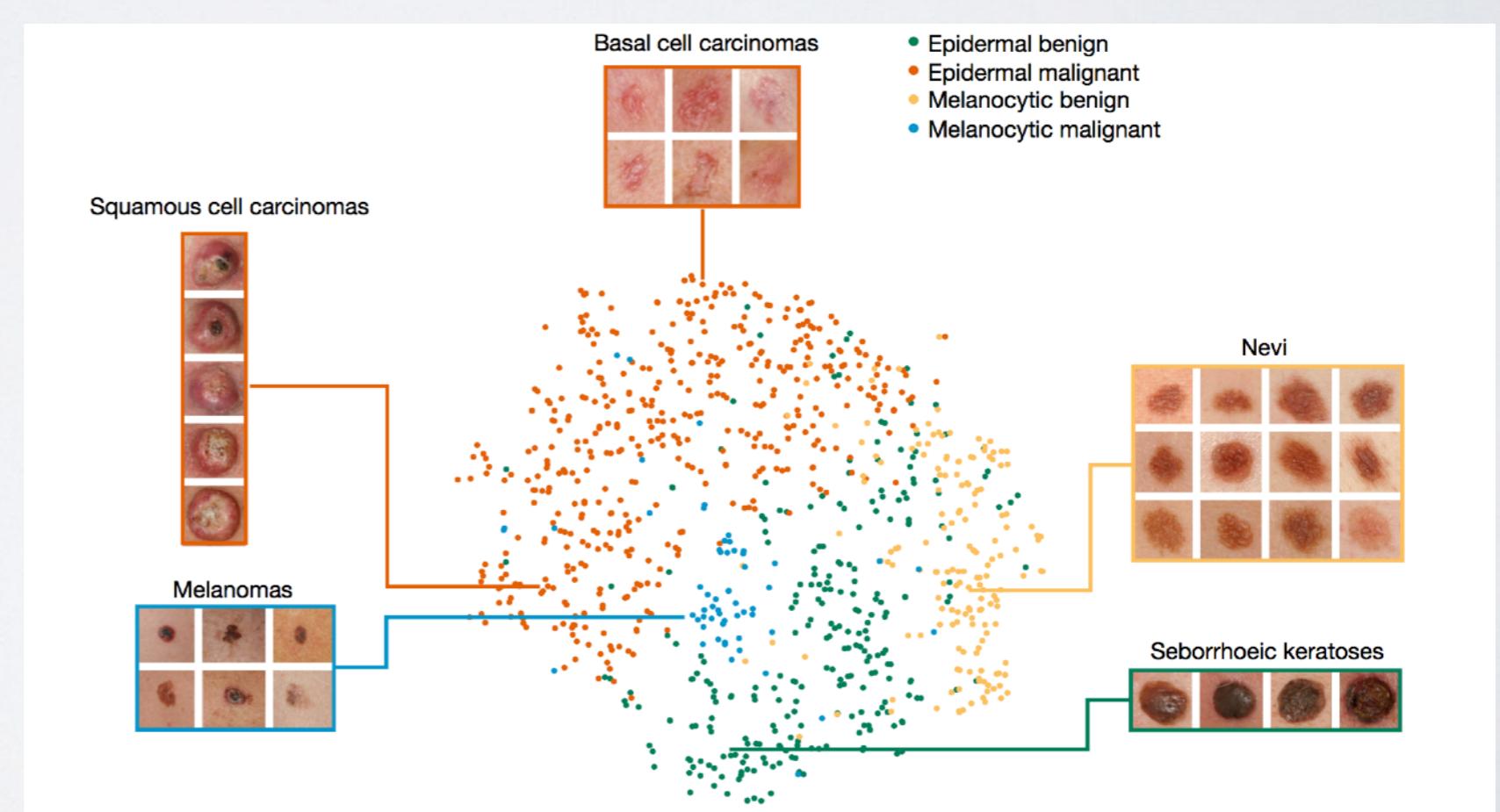
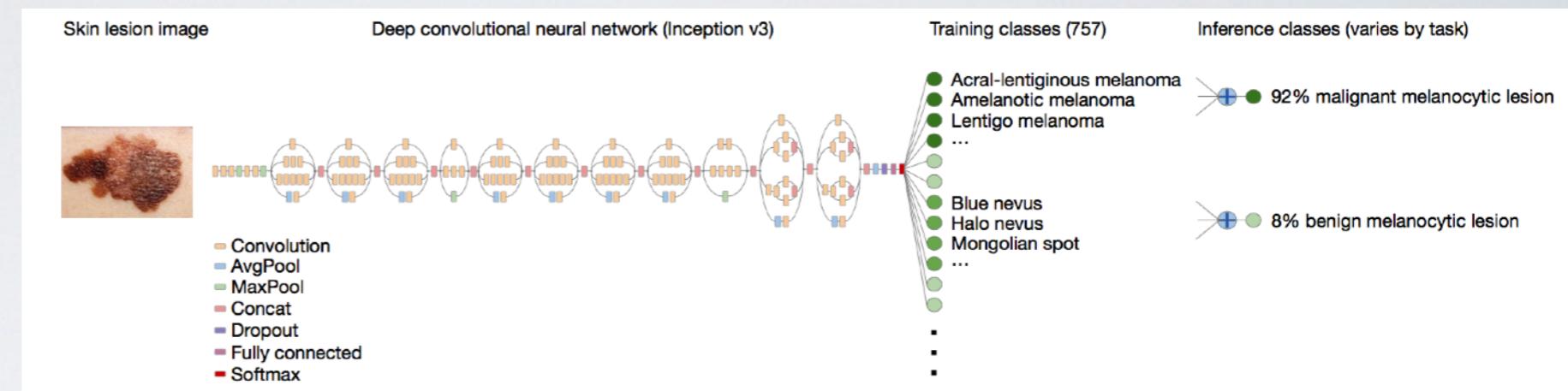
Supervised + Transfer Learning

- Dermatologist-level classification of skin cancer with deep neural networks

(Esteva et al., 2017)

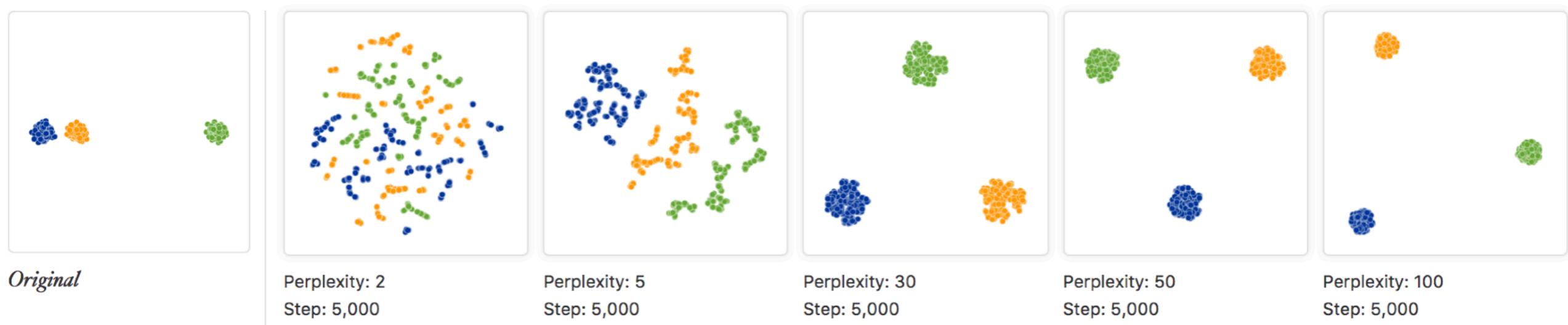
- Used pre-trained Google Inception V3 CNN pre-trained on ImageNet

NOT REPLACING DOCTORS!

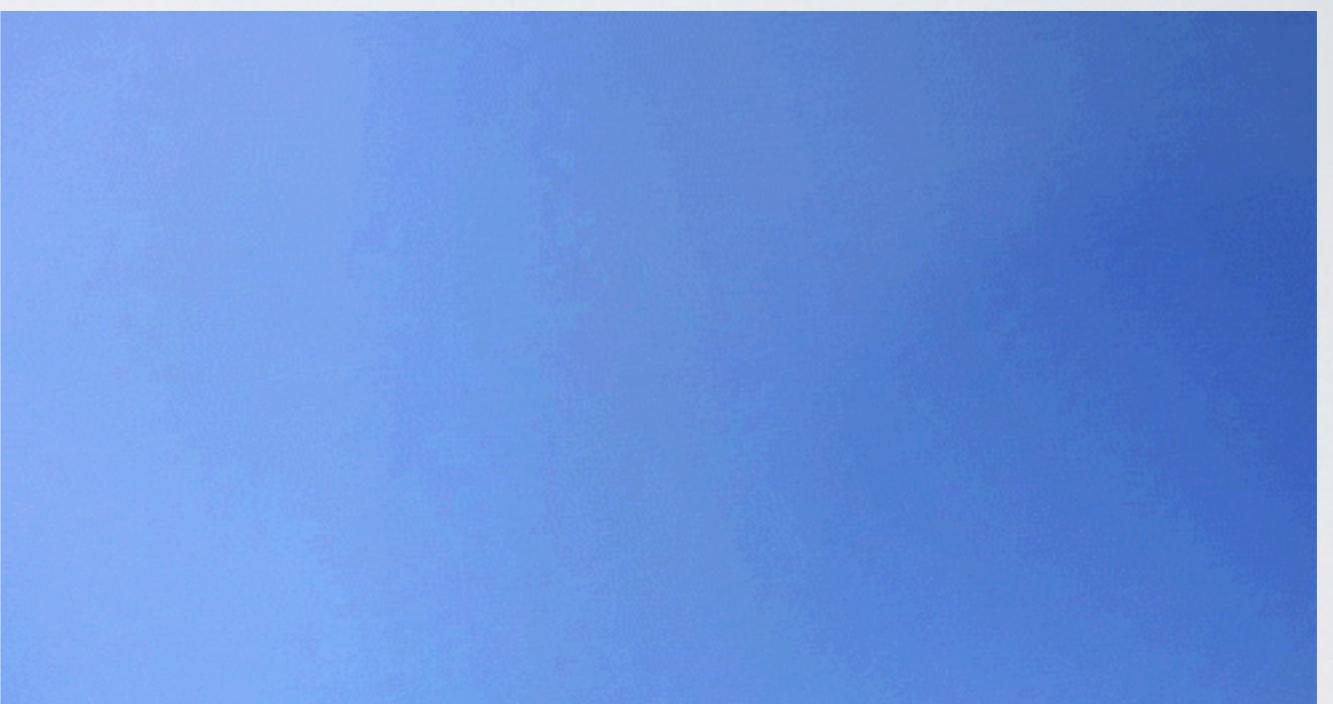


Visualizing feature representations

- t-SNE
- They say it is best used with < 50 features, if more, use PCA first to reduce
 - (I have found it can still work with large feature spaces)
- Don't ask me to explain it :)

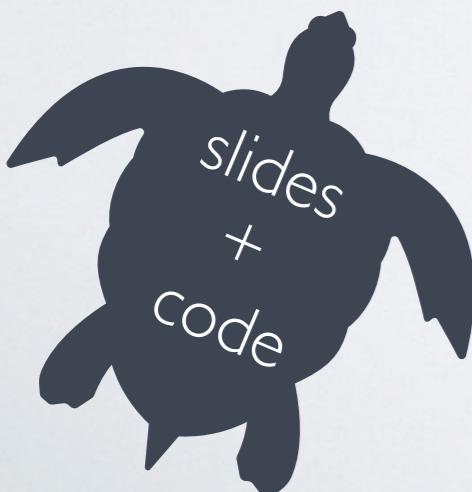


Code!!



Thank you!

- Aaron Richter
-  @rikturr  rikturr@gmail.com
- *** We're hiring!!



<https://github.com/rikturr/mml-feature-learning>