CS/ECE 148 –

Data Science Fundamentals

NNs, Unsupervised & Semi-supervised Learning

UCLA Computer Science

Outline

Regularization of NN

- Norm Penalties
- Early Stopping
- Data Augmentation
- Sparse Representation
- **Dropout**

Optimization

- **Challenges in Optimization**
- Momentum (next lectures)
- Adaptive Learning Rate (next lectures)
- Parameter Initialization (next lectures)
- Batch Normalization (next lectures)

Outline

Unsupervised Learning

- K-means
- Mean Shift
- Hierarchical Clustering
- DBSCAN

Semi-supervised Learning

- Self Training

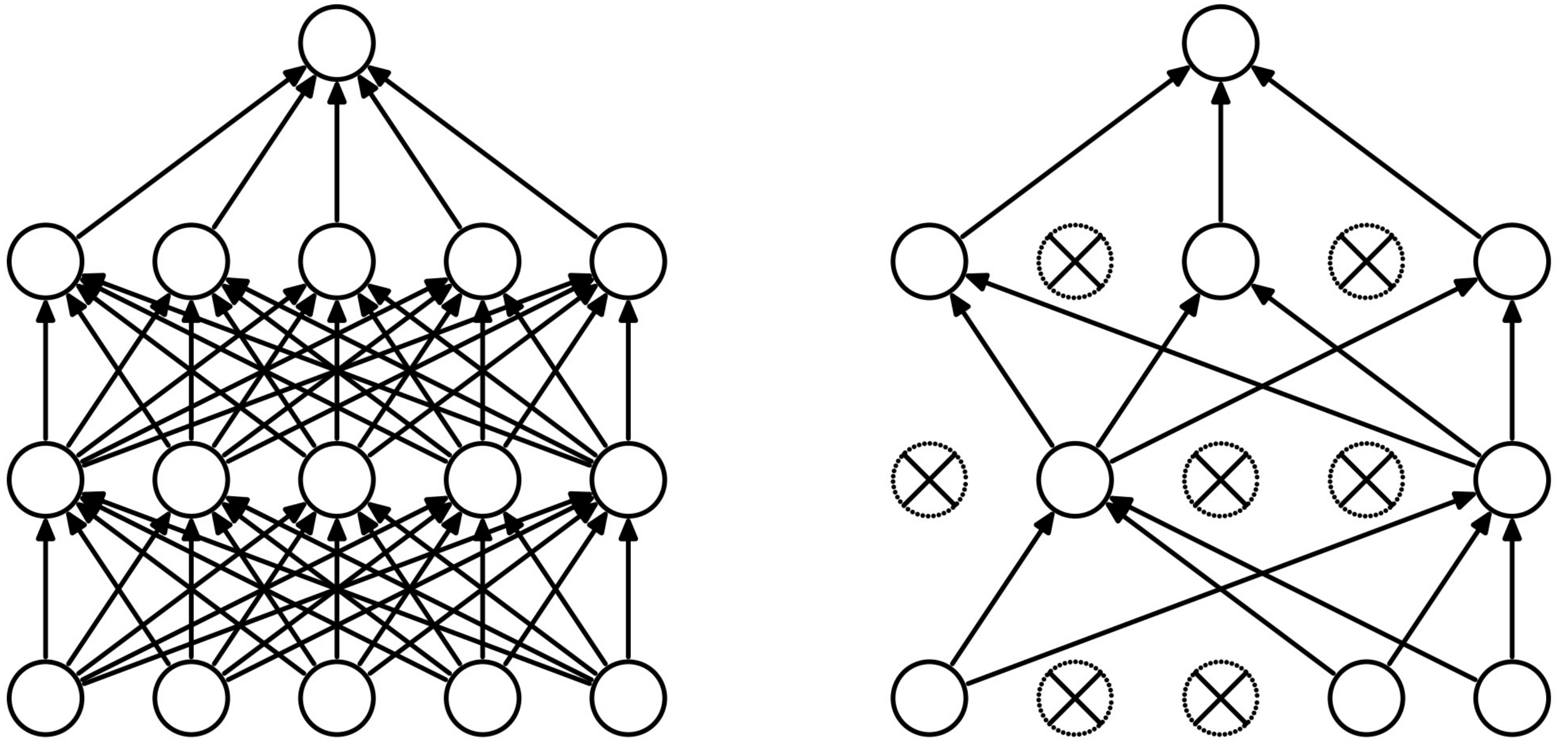
Outline

Regularization of NN

- Norm Penalties
- Early Stopping
- Data Augmentation
- Sparse Representation
- **Dropout**

Dropout

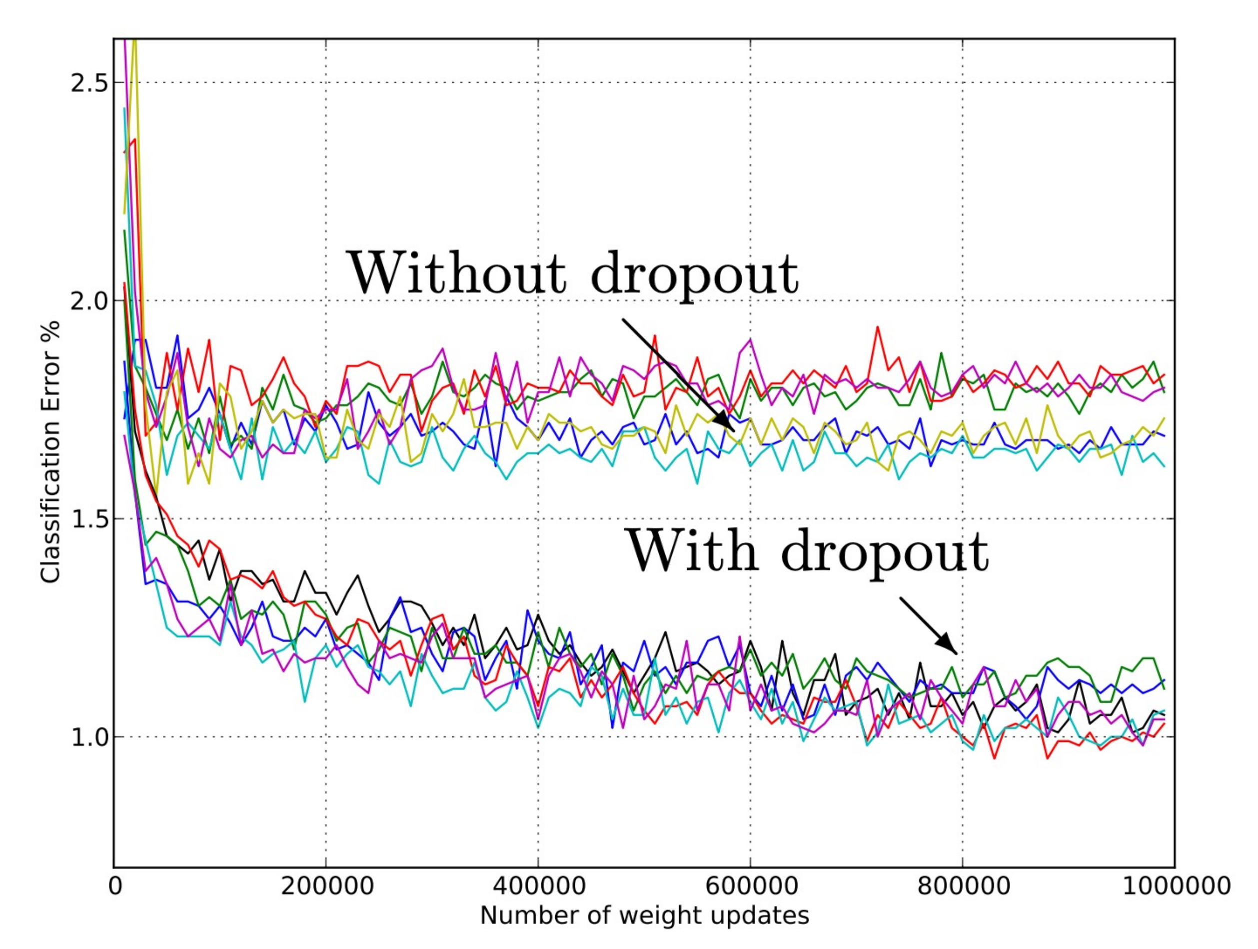
- Randomly set some neurons and their connections to zero (i.e. “dropped”)
- Prevent overfitting by reducing co-adaptation of neurons
- Like training many random sub-networks



(a) Standard Neural Net (b) After applying dropout.

Dropout

- Widely used and highly effective
- Proposed as an alternative to ensembling, which is too expensive for neural nets



Test error for different architectures with and without dropout. The networks have 2 to 4 hidden layers each with 1024 to 2048 units.

http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf

Dropout: Stochastic GD

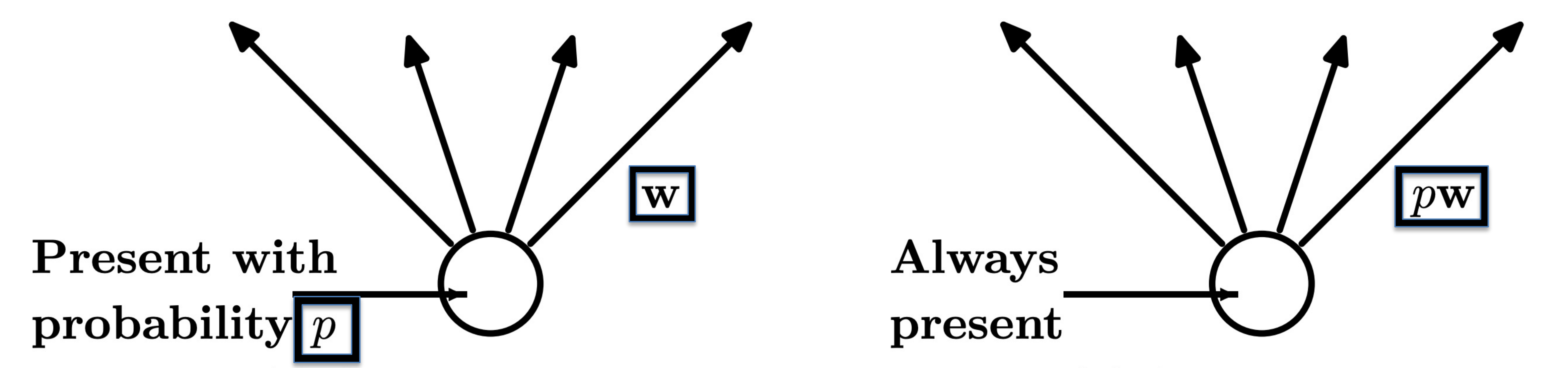
For each new example/mini-batch:

- Randomly sample a binary mask μ independently, where μ_i indicates if input/hidden node i is included
- Multiply output of node i with μ_i, and perform gradient update

Typically, an input node is **included** with **prob=0.8**, hidden node with **prob=0.5.**

Dropout: Weight Scaling

- We can think of dropout as training many of sub-networks
- At **test time**, we can “aggregate” over these sub-networks by **reducing connection weights in proportion to dropout probability, *p***



(a) At training time

(b) At test time

Outline

Optimization

- **Challenges in Optimization**
- Momentum (next lectures)
- Adaptive Learning Rate (next lectures)
- Parameter Initialization (next lectures)
- Batch Normalization (next lectures)

Learning vs. Optimization

Goal of learning: minimize generalization error, or the loss function

$$\mathcal{L}(\boldsymbol{W}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p_{data}}\left[L(f\left(x, W\right), y\right]$$

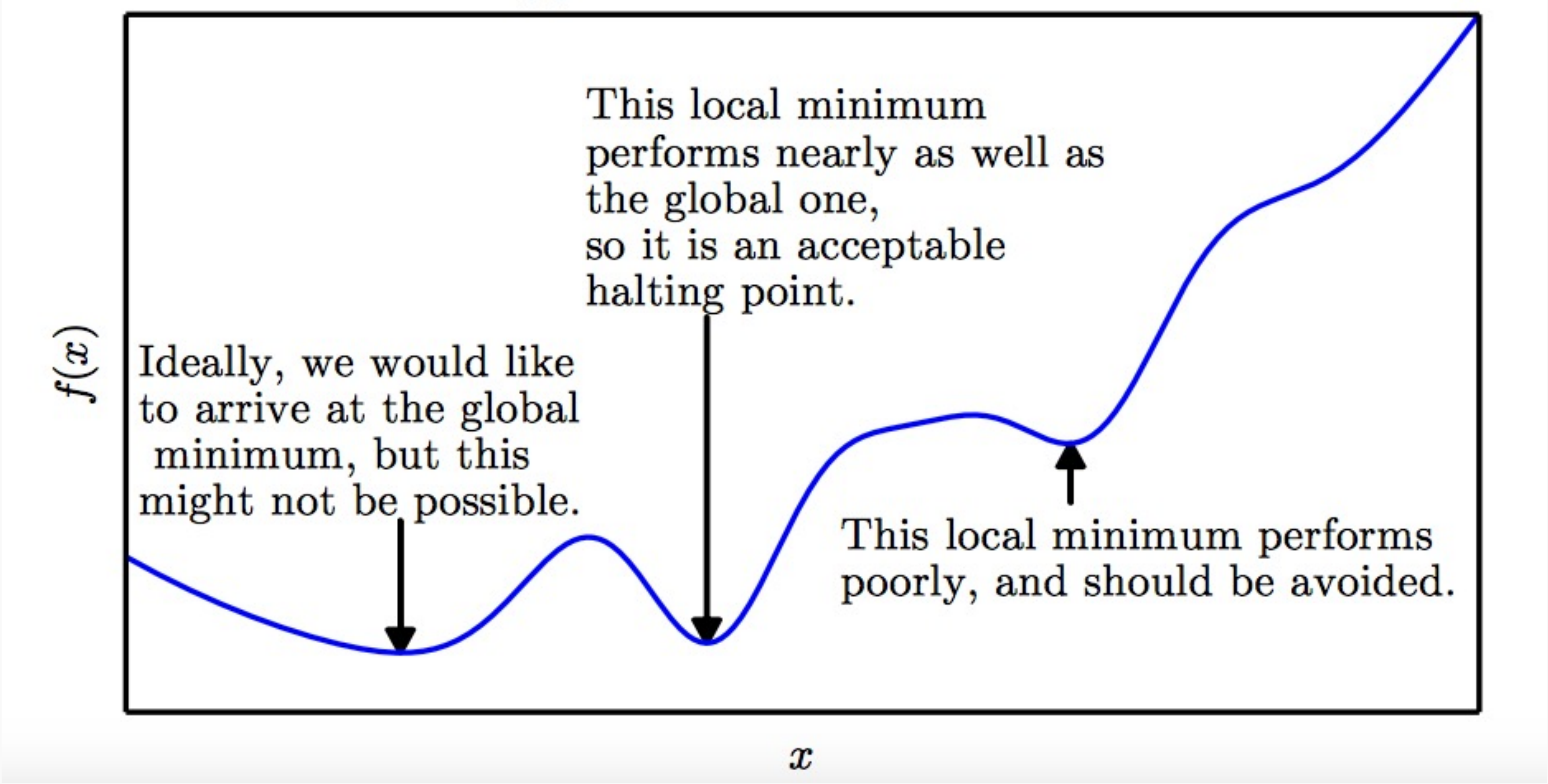
f is the neural network

In practice, empirical risk minimization:

$$\mathcal{L}(W) = \sum_i \left[L(f\left(x_i; W\right), y_i\right]$$

Quantity optimized different from the quantity we care about

Local Minima

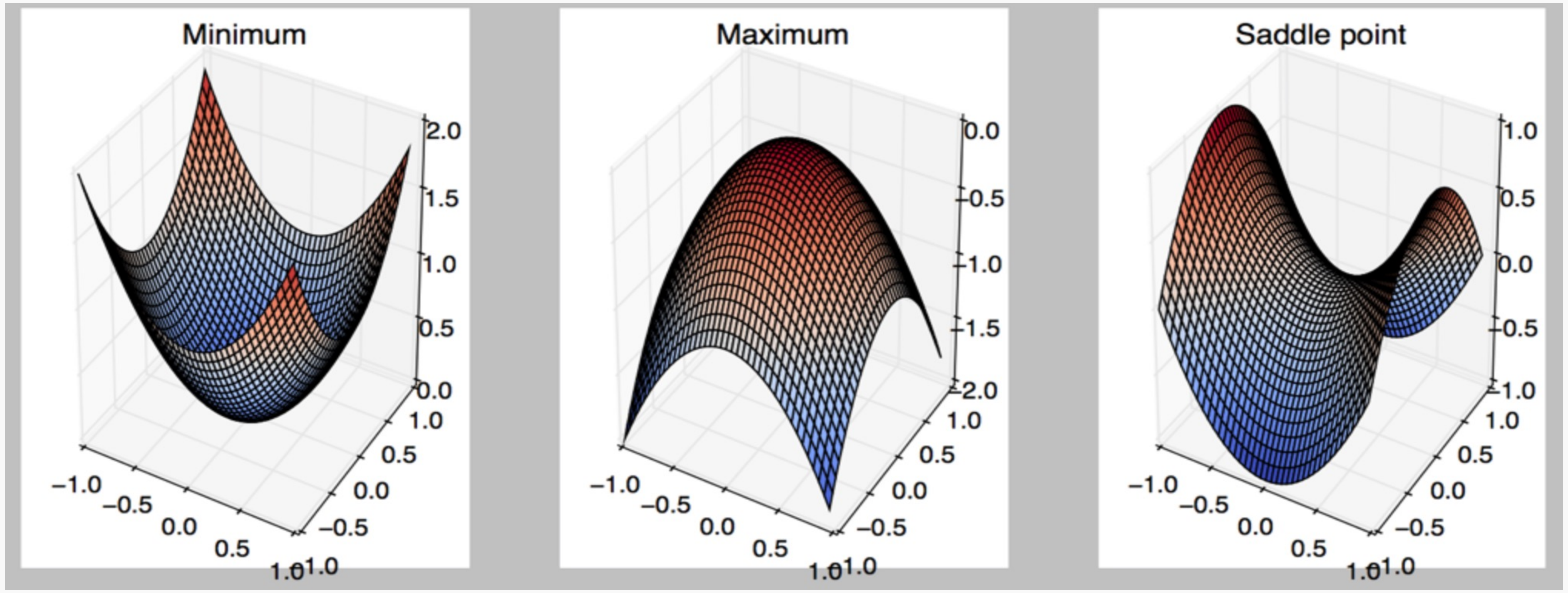


Goodfellow et al. (2016)

Critical Points

Points with zero gradient

2^{nd}-derivate (Hessian) determines curvature



Goodfellow et al. (2016)

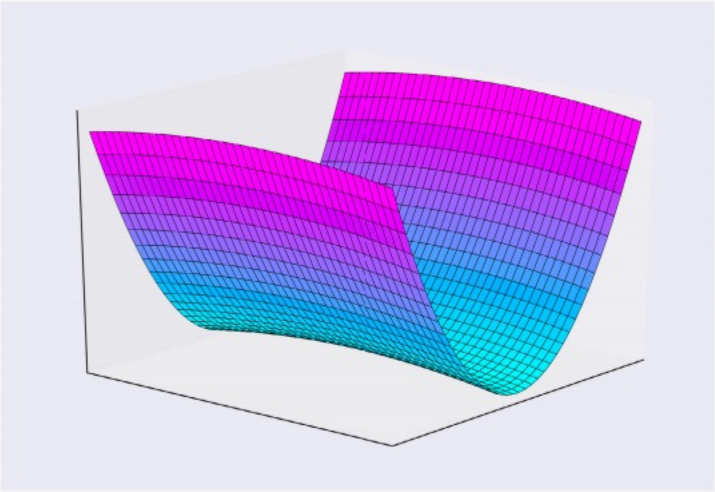
Local Minima

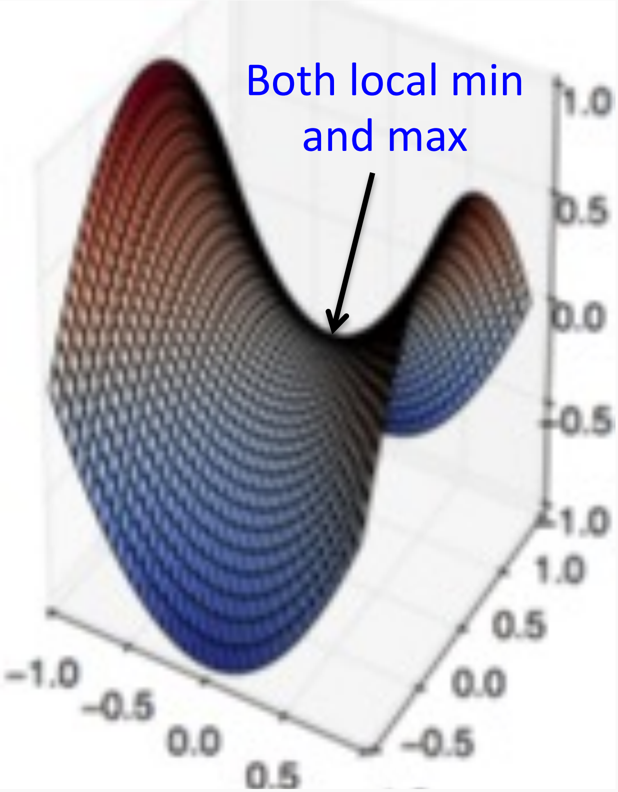
Old view: local minima is major problem in neural network training

Recent view:

- For sufficiently large neural networks, most local minima incur low cost
- Not important to find true global minimum

Saddle Points





Recent studies indicate that in high dim, saddle points are more likely than local min

Gradient can be very small near saddle points

Goodfellow et al. (2016)

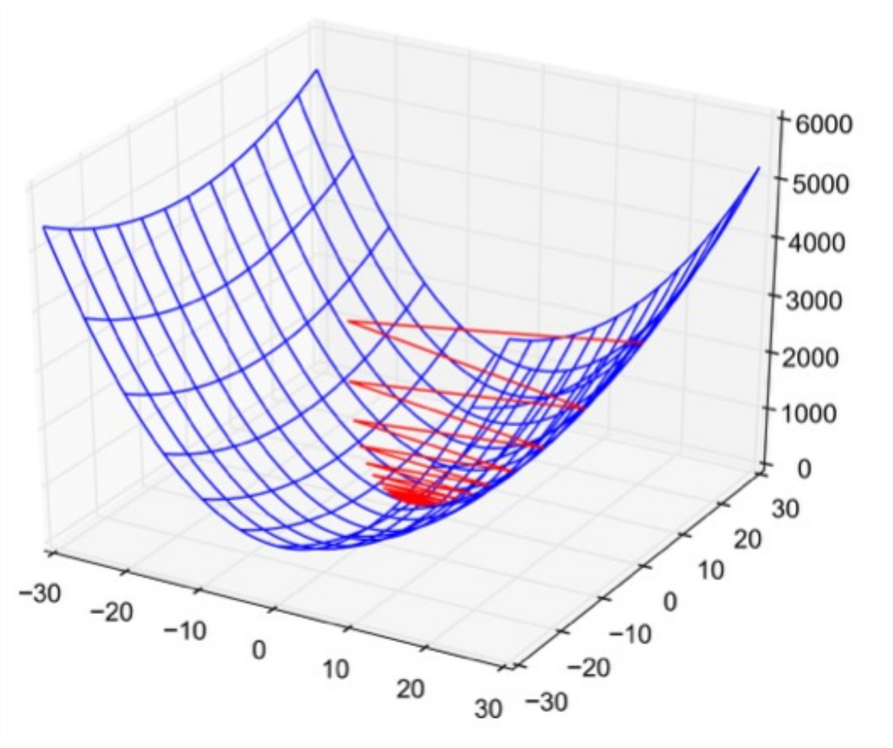
Poor Conditioning

Poorly conditioned Hessian matrix

- High curvature: small steps leads to huge increase

Learning is slow despite strong gradients

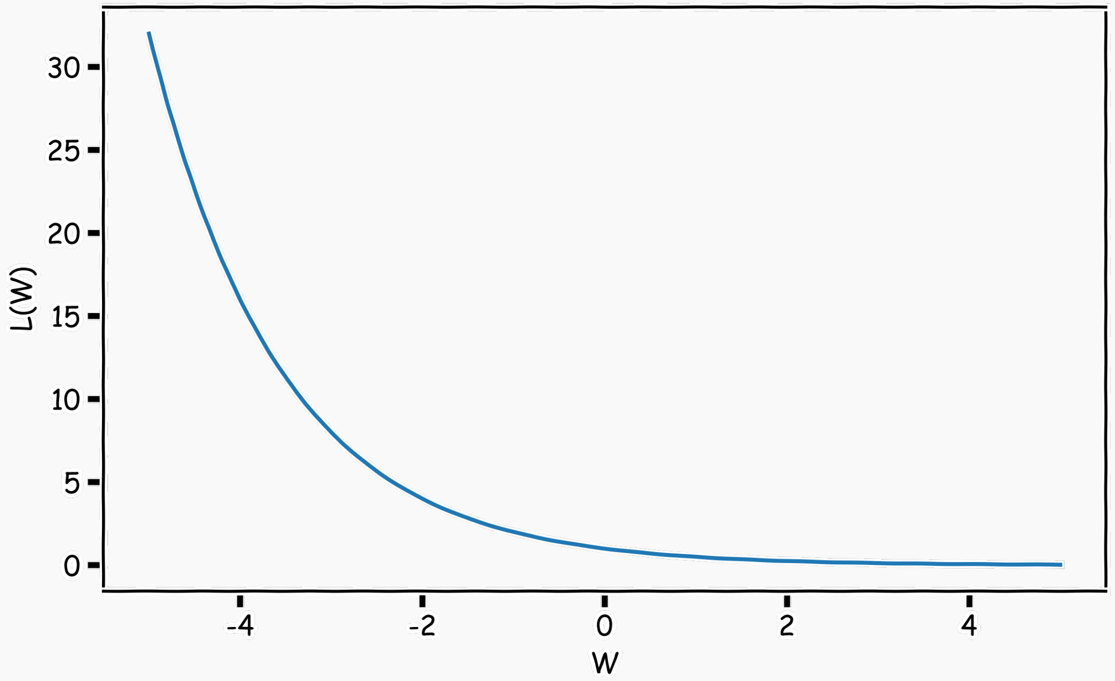
Oscillations slow down progress



No Critical Points

Some cost functions do not have critical points. In particular classification.

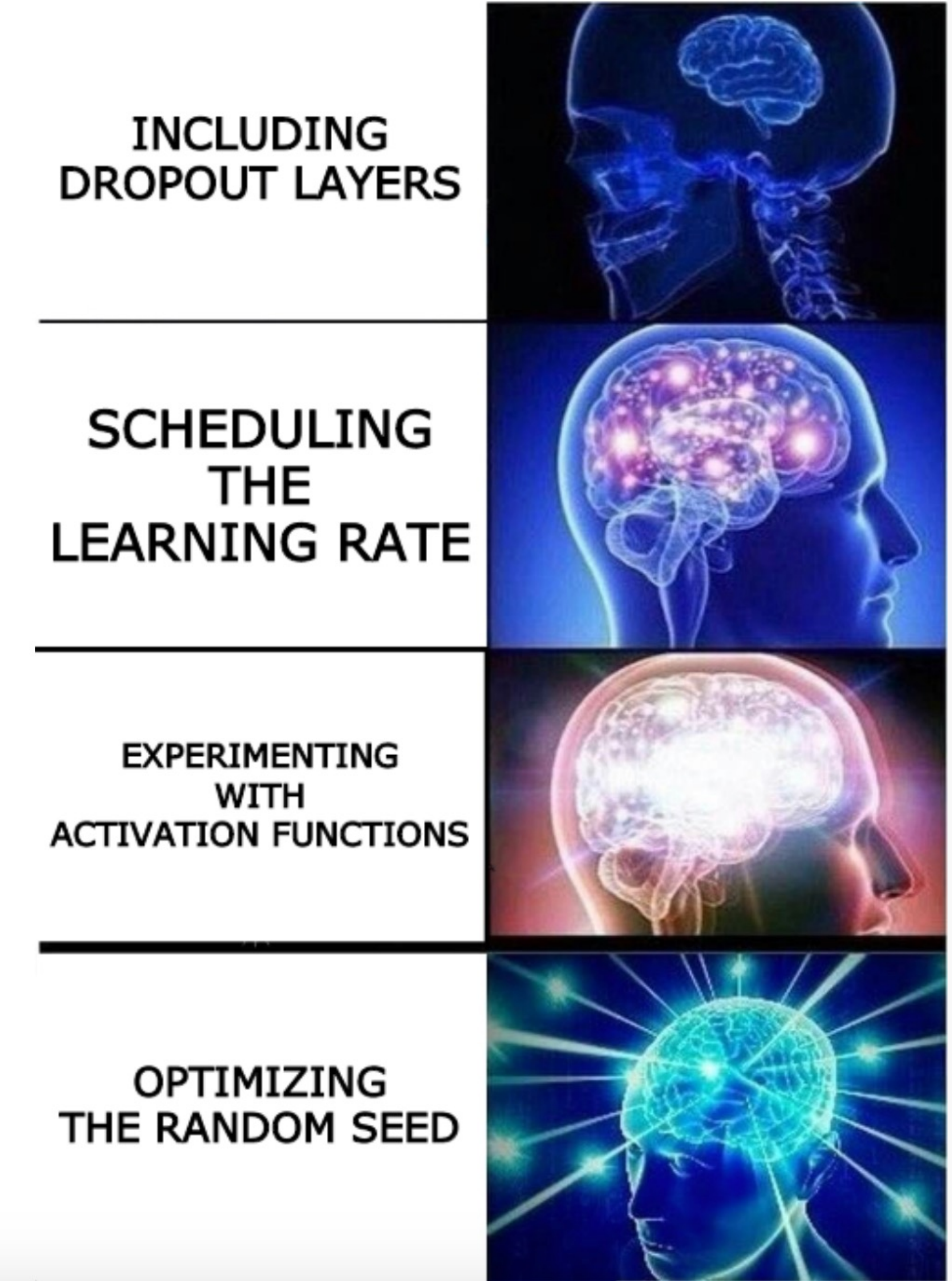
WHY?

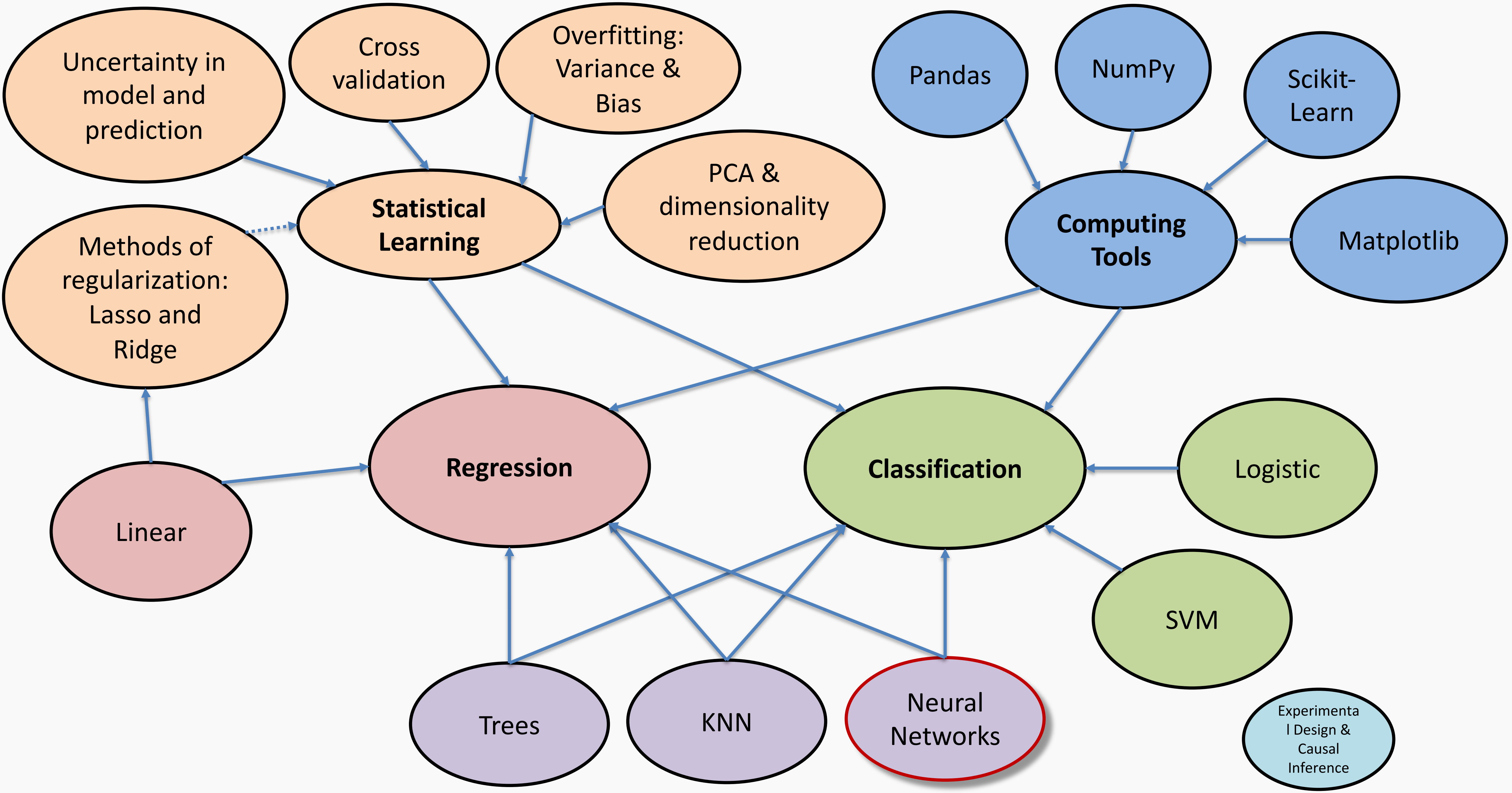


Optimization Challenges

We’ll discuss some solutions in the next lectures

- Momentum (later)
- Adaptive Learning Rate (later)
- Parameter Initialization (later)
- Batch Normalization (later)


INCLUDING DROPOUT LAYERS
SCHEDULING THE LEARNING RATE
EXPERIMENTING WITH ACTIVATION FUNCTIONS
OPTIMIZING THE RANDOM SEED


Uncertainty in model and prediction
Cross validation
Overfitting: Variance & Bias
Pandas
NumPy
Scikit-Learn
Statistical Learning
PCA & dimensionality reduction
Computing Tools
Matplotlib
Methods of regularization: Lasso and Ridge
Regression
Classification
Logistic
Linear
SVM
Trees
KNN
Neural Networks
Experimenta l Design & Causal Inference

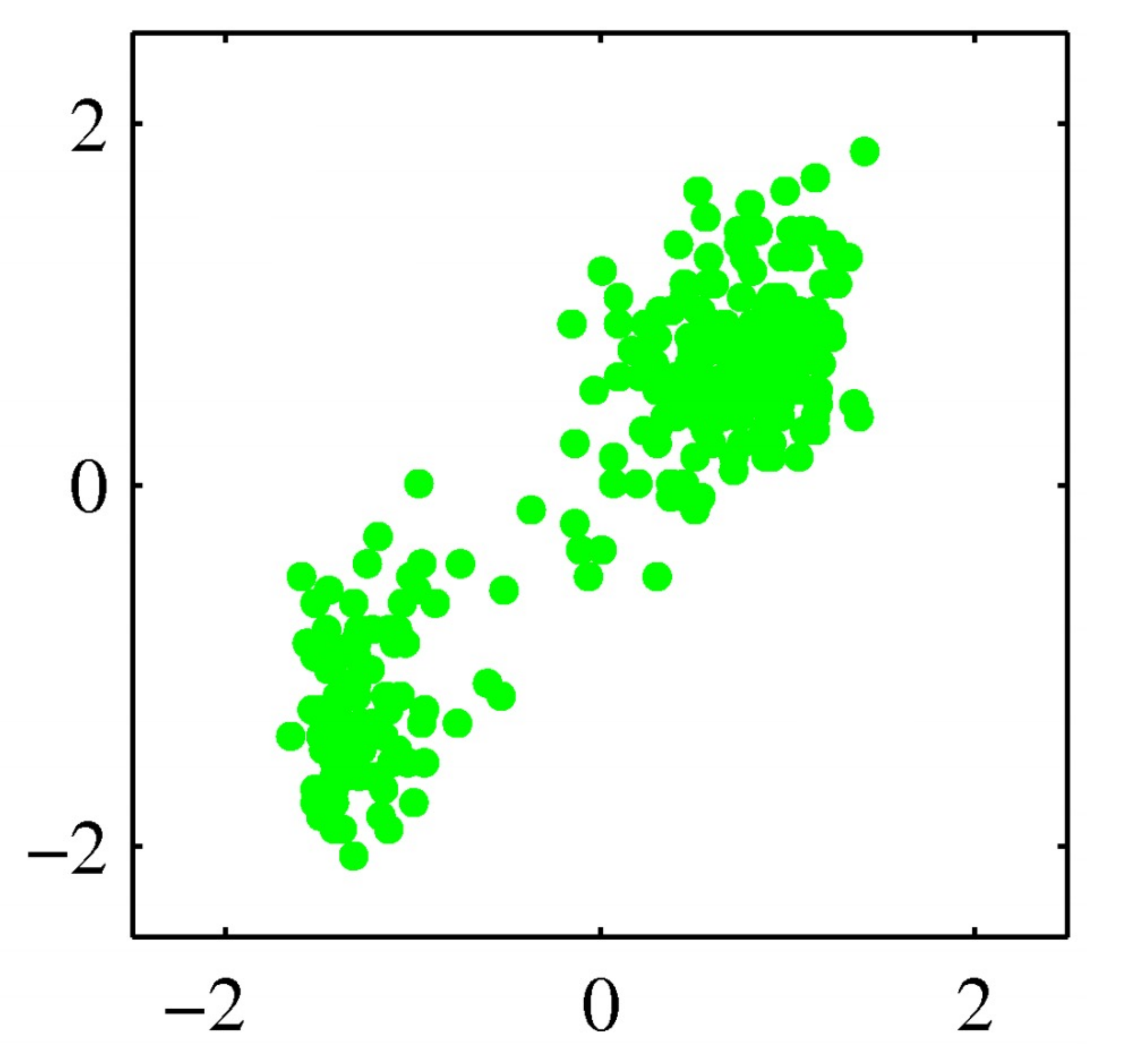
Unsupervised Learning

Unsupervised Learning

- K-means
- Mean-shift
- Hierarchical Clustering
- DBSCAN

- Applications

Unsupervised Setting



Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006

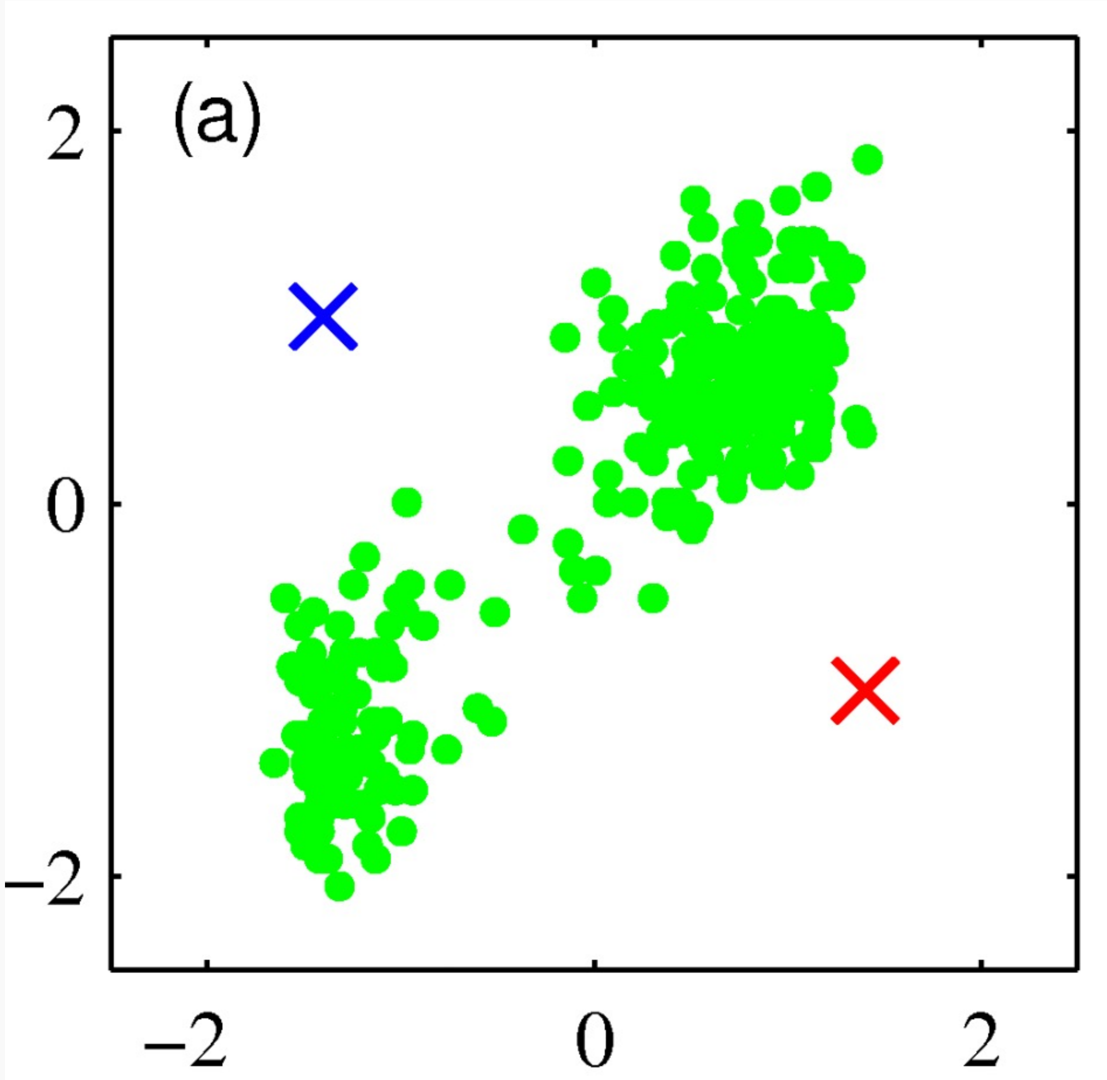
K-means – Algorithm

Initialization:

– choose K random positions

– assign cluster centers μ^j to these positions

K-means



Bishop, "Pattern Recognition and Machine Learning", Springer, 2006

K-means

Until Convergence:

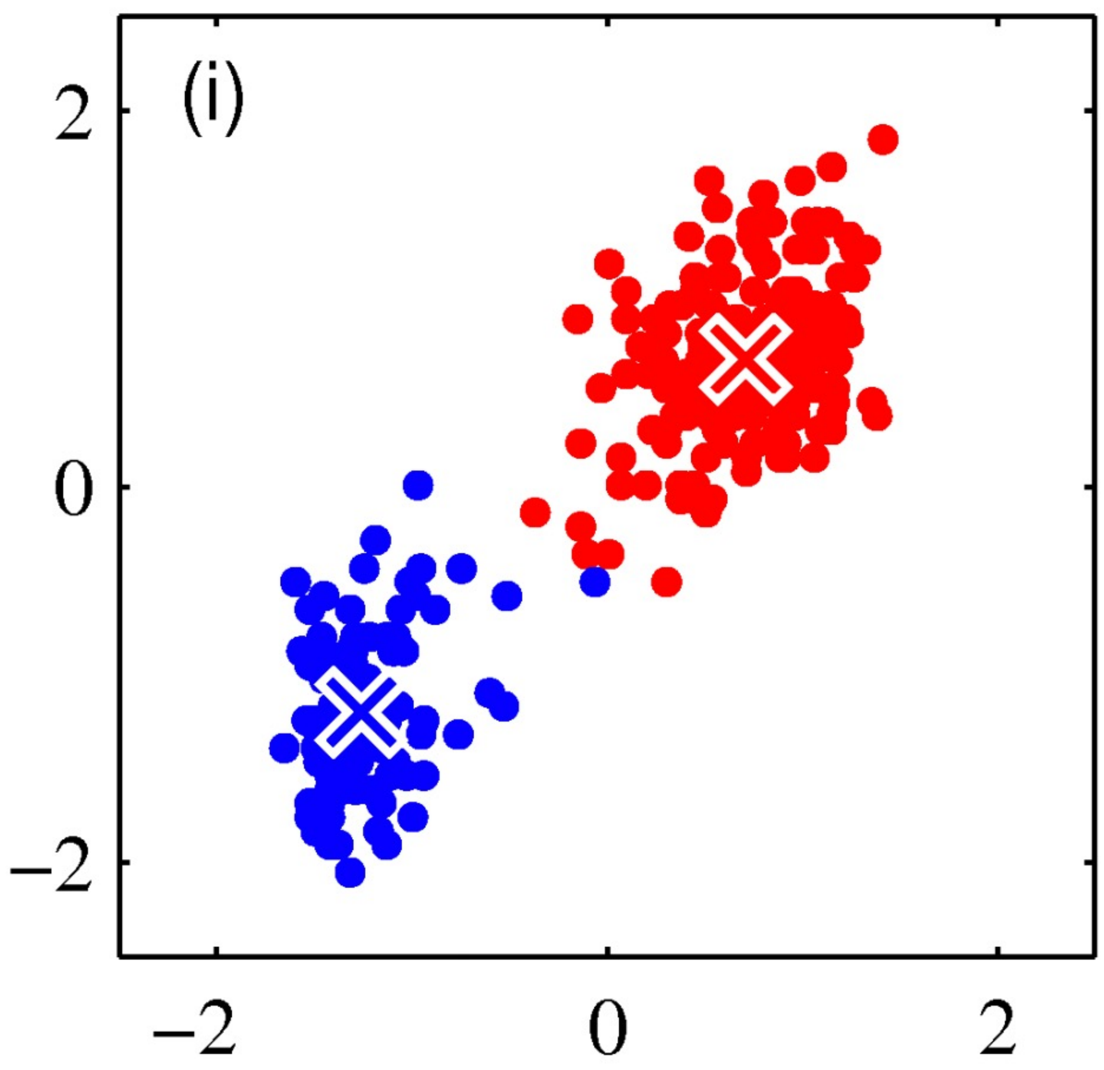
– Compute distances $\left\|x^{(i)} - \mu^{(i)}\right\|$

– Assign points to nearest cluster center

– Update Cluster centers:

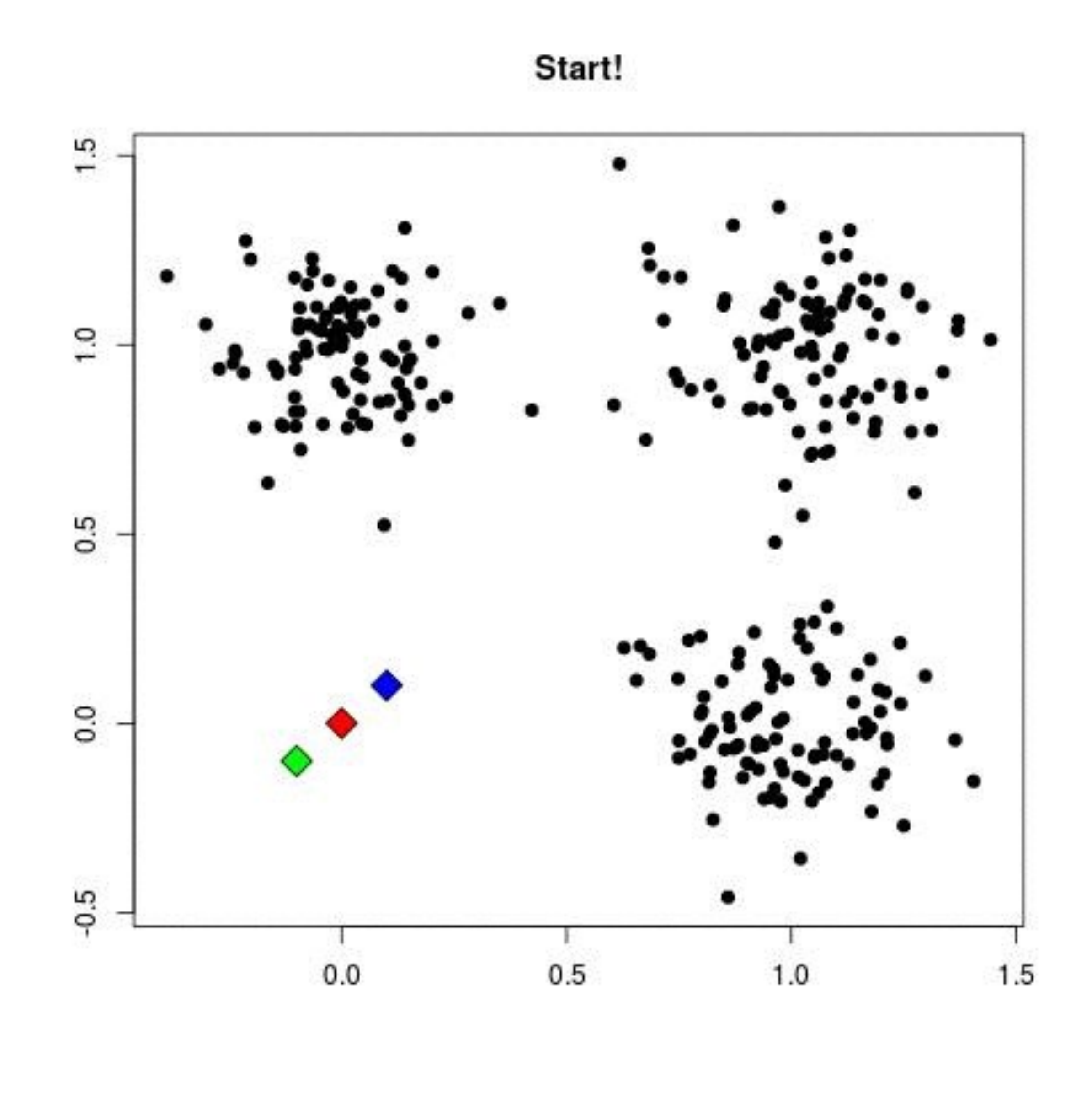
$$\mu^{(j)} = \frac{1}{N_j} \sum_{x_i \in C_j} x_i$$

K-means

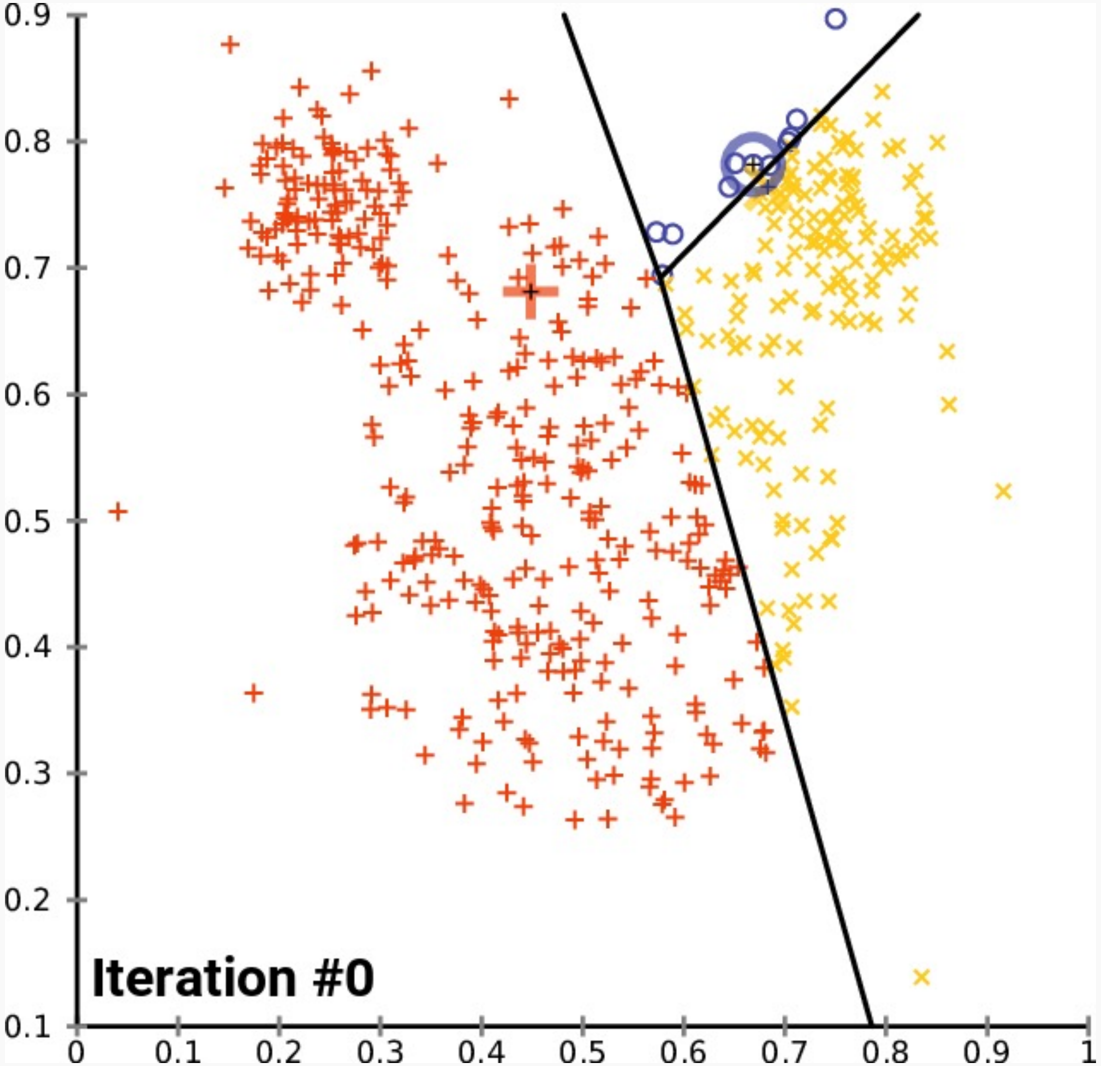


Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006

K-means



K-means



K-means Summary

- Guaranteed to converge
- Result depends on initialization
- Number of clusters is important
- Sensitive to outliers
    - Use median instead of mean for updates

Initialization Methods

- Random Positions
- Random data points as Centers
- Random Cluster assignment to data points

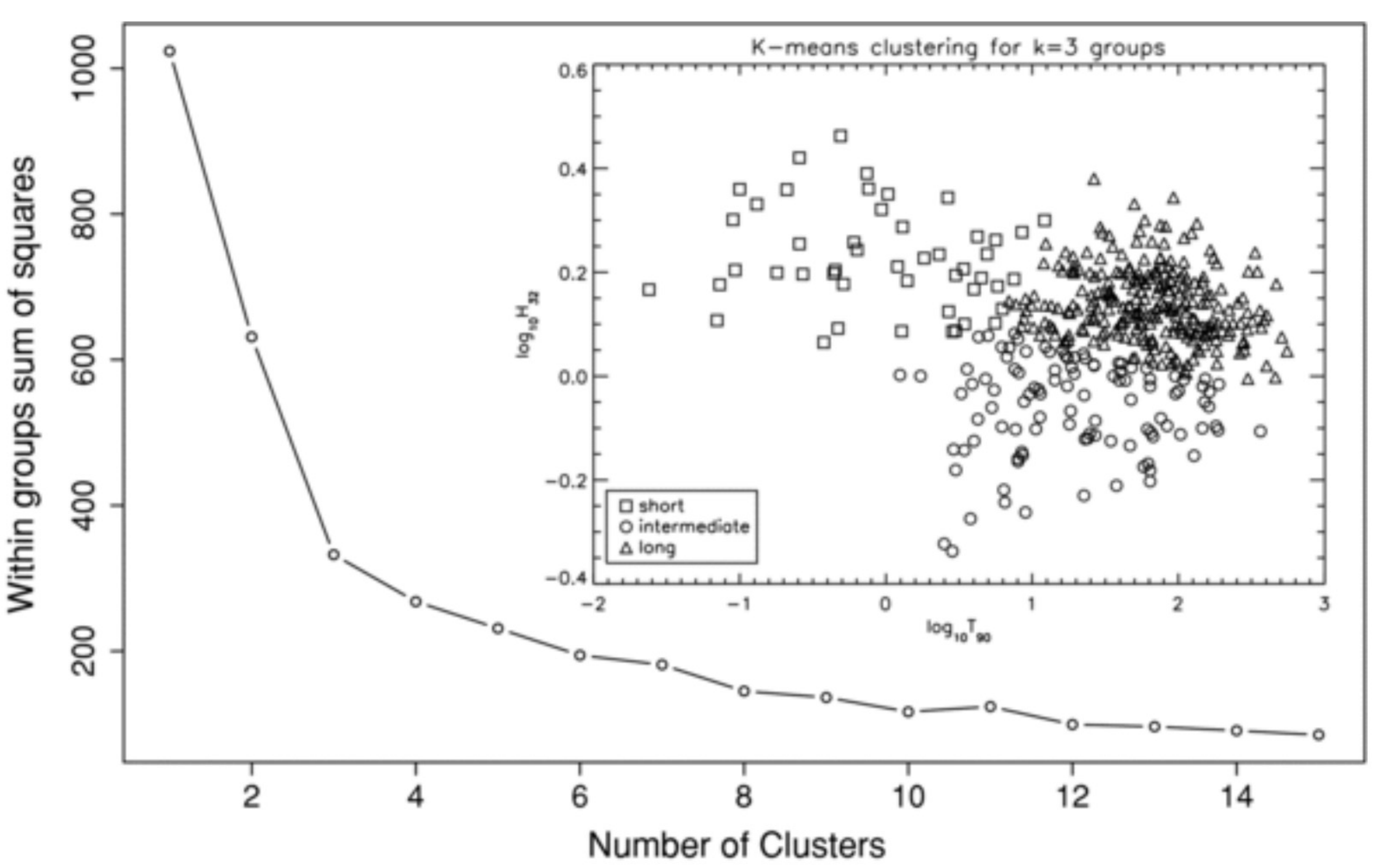
- Start several times

How to find K

- Extreme cases:
  - K=1
  - K=N

- Choose K such that increasing it does not model the data much better.

“Knee” or “Elbow” method



Cross Validation

- Use this if you want to apply your clustering solution to new unseen data

- Partition data into n folds
- Cluster on n-1 folds
- Compute sum of squared distances to centroids for validation set

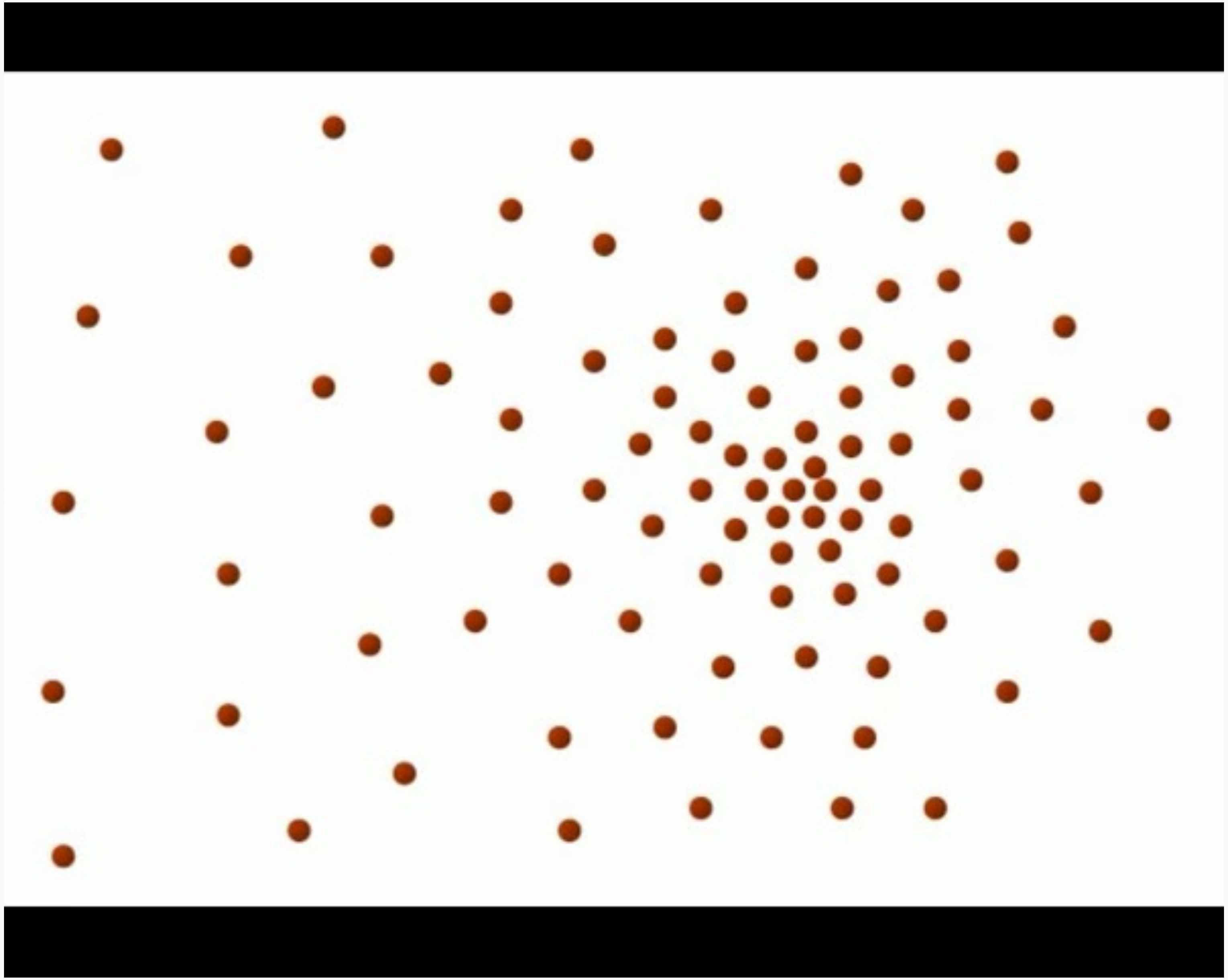
Getting Rid of K

- Having to specify K is annoying
- Can we do without?

Mean Shift

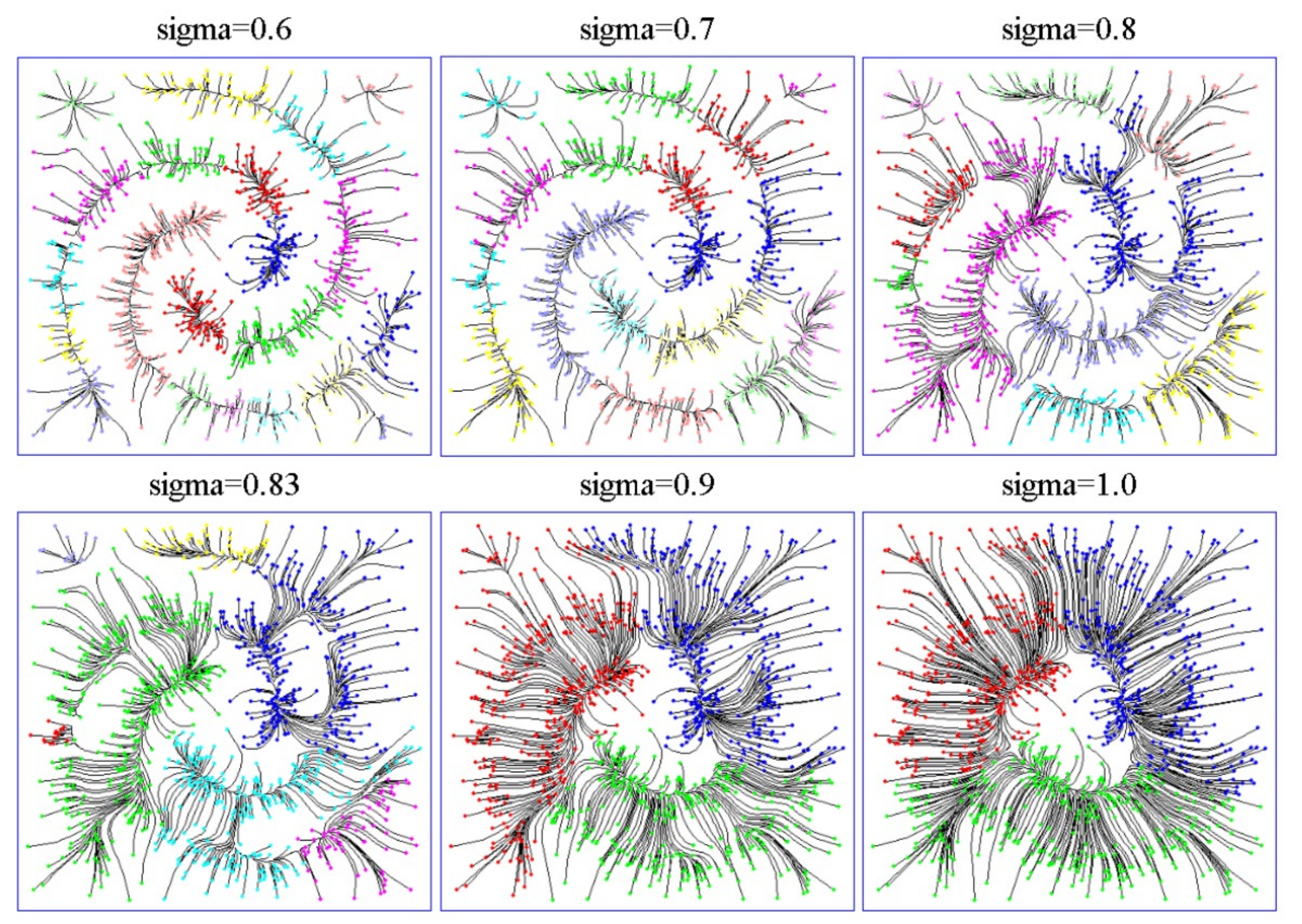
1. Put a window around each point
2. Compute mean of points in the frame.
3. Shift the window to the mean
4. Repeat until convergence

Mean Shift



https://www.youtube.com/watch?v=kmaQAsotT9s

Mean Shift



Fischer et al., “Clustering with the Connectivity Kernel“, NIPS (2003)

Mean Shift Summary

- Does not need to know number of clusters
- Can handle arbitrary shaped clusters
- Robust to initialization
- Needs bandwidth parameter (window size)
- Computationally expensive

- Very good article:

http://saravananthirumuruganathan.wordpress.com/2010/04/01/introduction-to-mean-shift-algorithm/