Medical Image Analysis and Processing

Medical Image Segmentation
Pixel Classification - Clustering

Emad Fatemizadeh

Distance/online Course: Session 19

Date: 02 May 2021, 12th Ordibehesht 1400

Contents

- Clustering Cost Function
- K-means Clustering
- FCM Clustering
- GMM Clustering
- Mean-Shift Clustering
- Supervised Clustering (brief)
- Intensity inhomogeneity aware segmentation (AFCM)

Clustering Cost Function

› An useful cost function for hard clustering:

$$J = \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} u_{ij}\left\|x_i - \omega_j\right\|_2^2$$

› ω_j: Cluster prototype (cluster center)

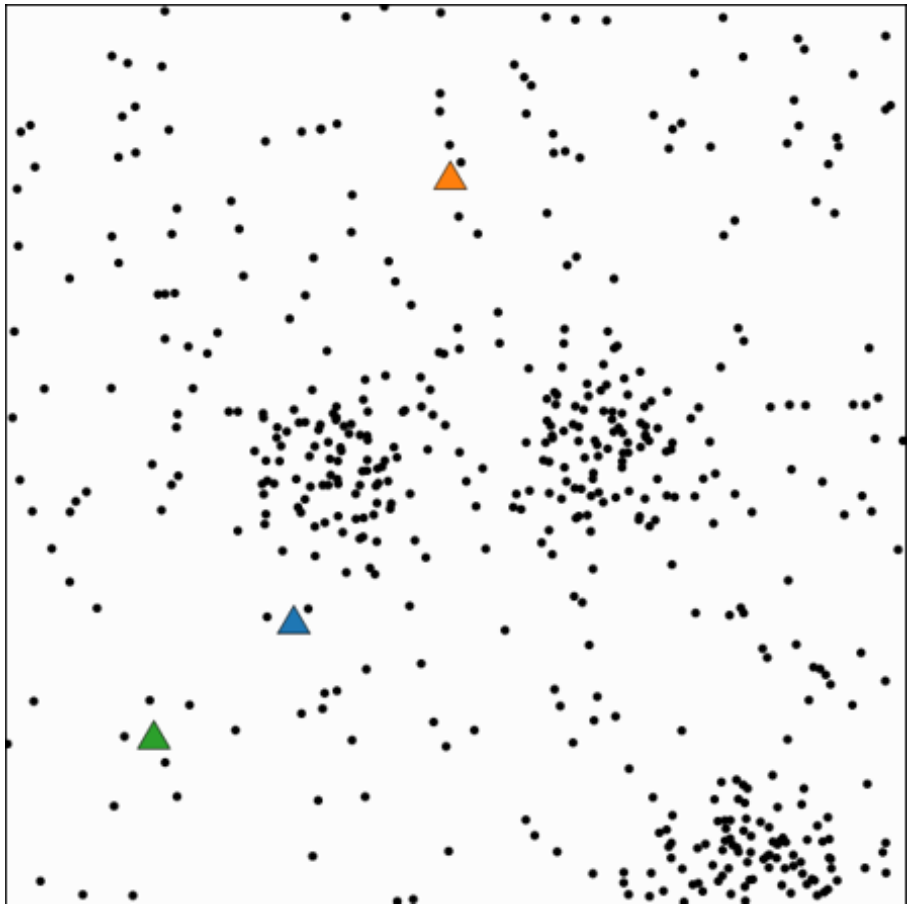
› This function is not differentiable (why?)

› K-means approach:

- Determine u_{ij} based on minimum distance (x_i with respect to ω_j)
- Determine ω_j based on minimum internal energy (CoG)

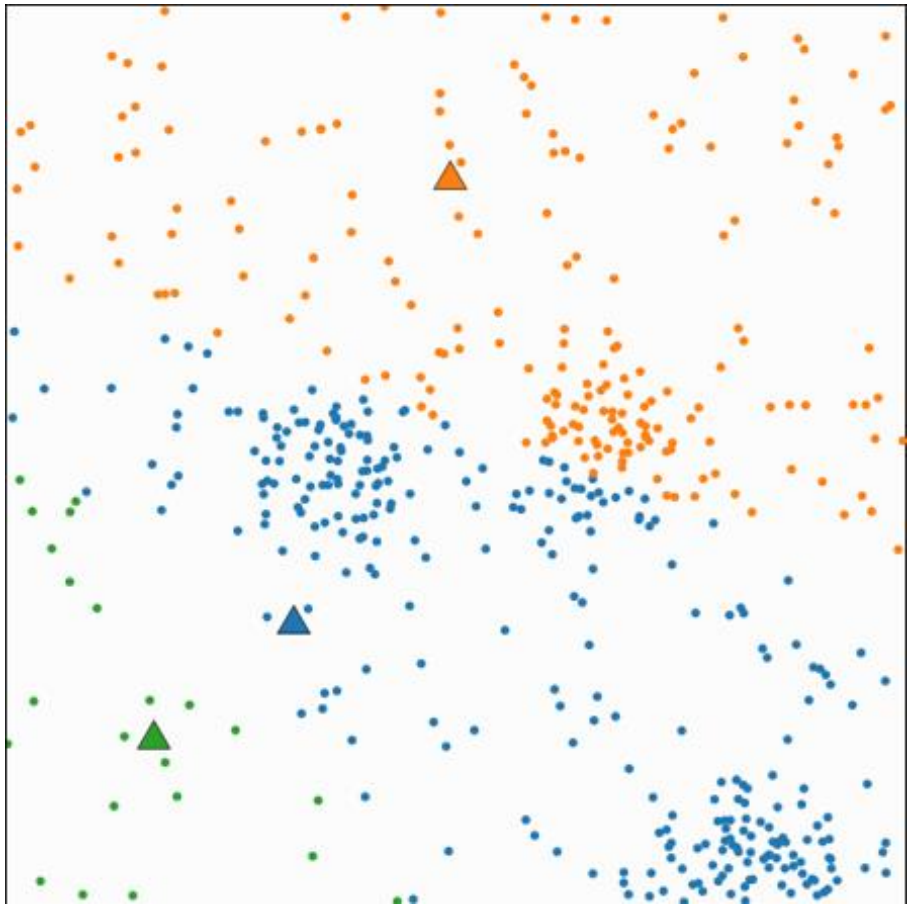
K-means

› Initialization:



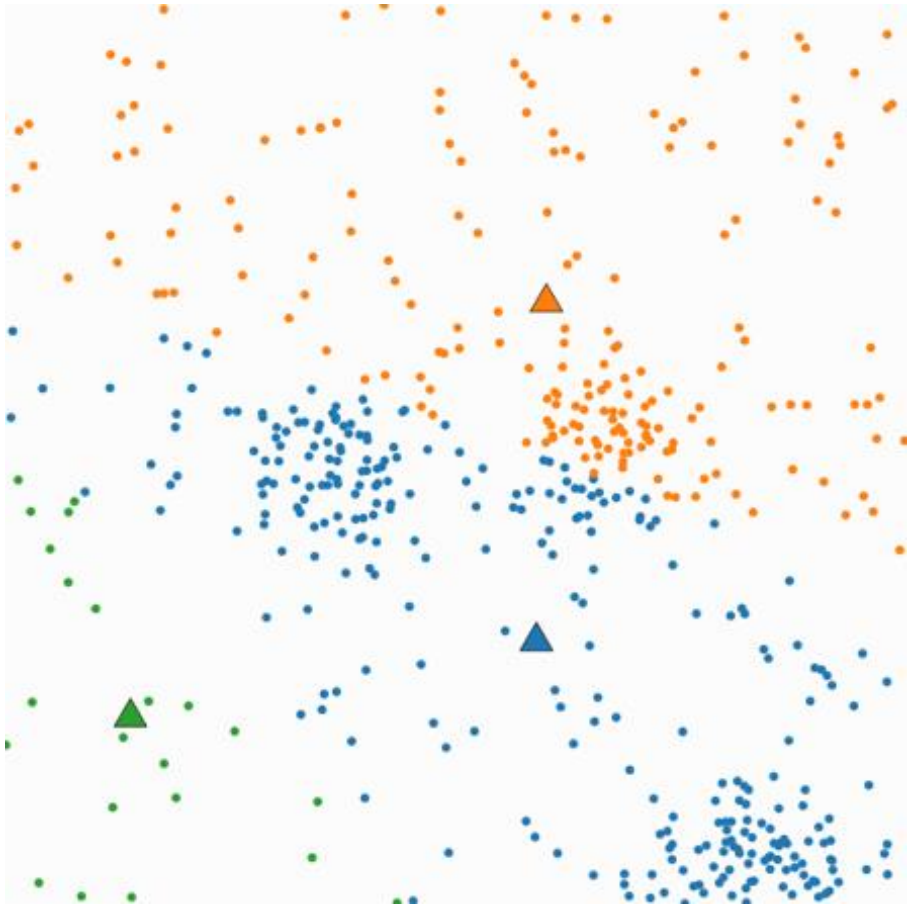
K-means

› Form Clusters:



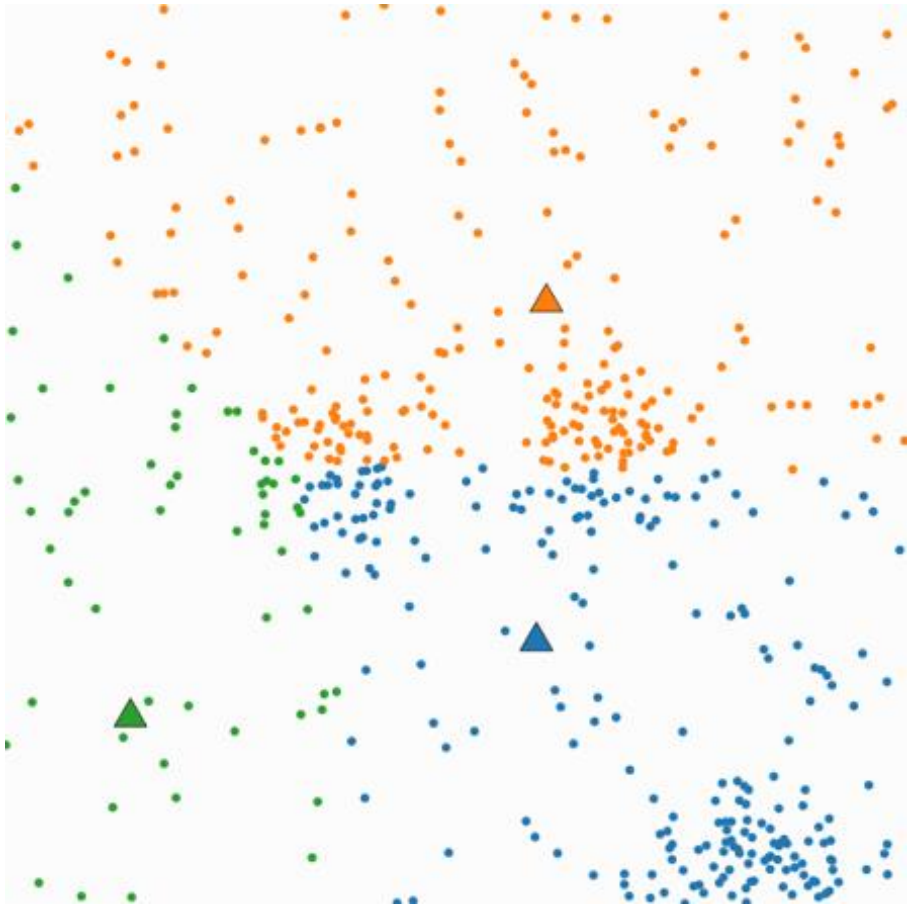
K-means

› Update Centers:



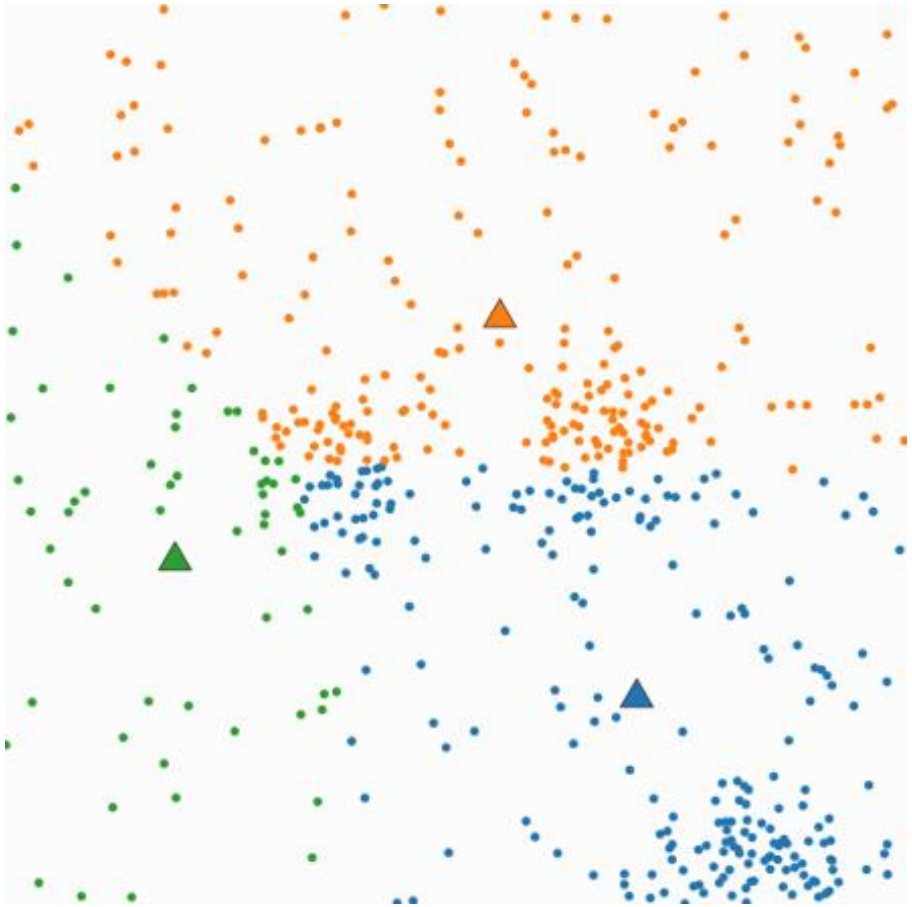
K-means

› Update Clusters:



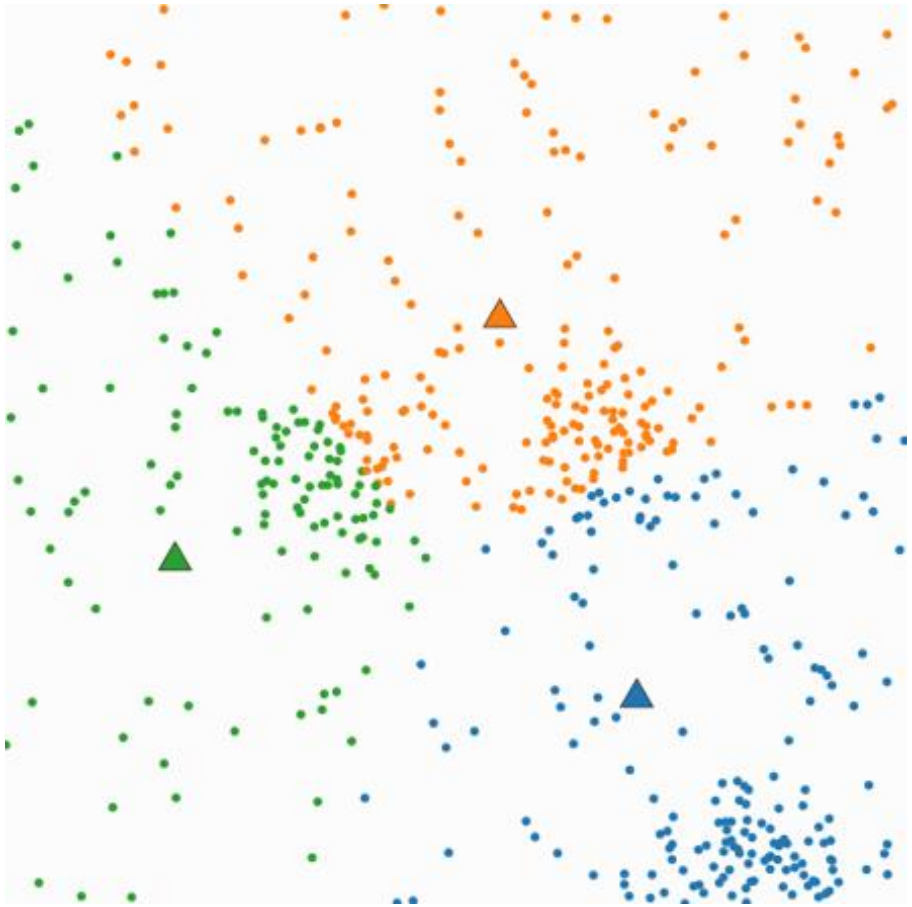
K-means

› Update Centers:



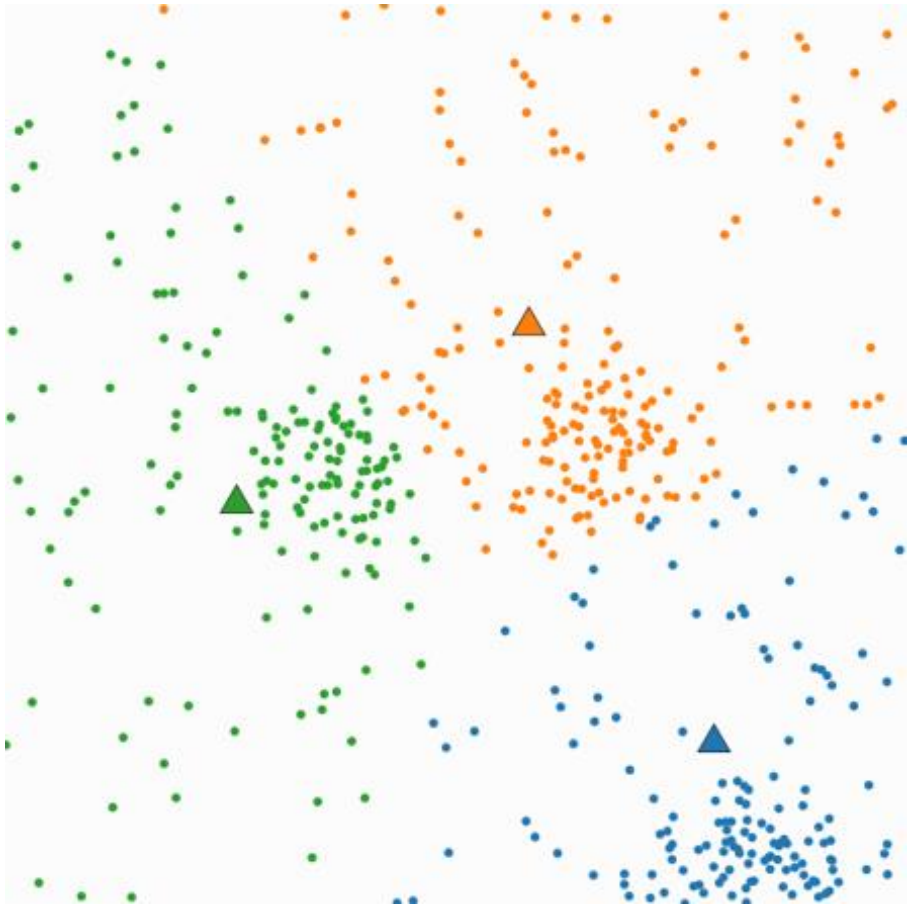
K-means

› Update Clusters:



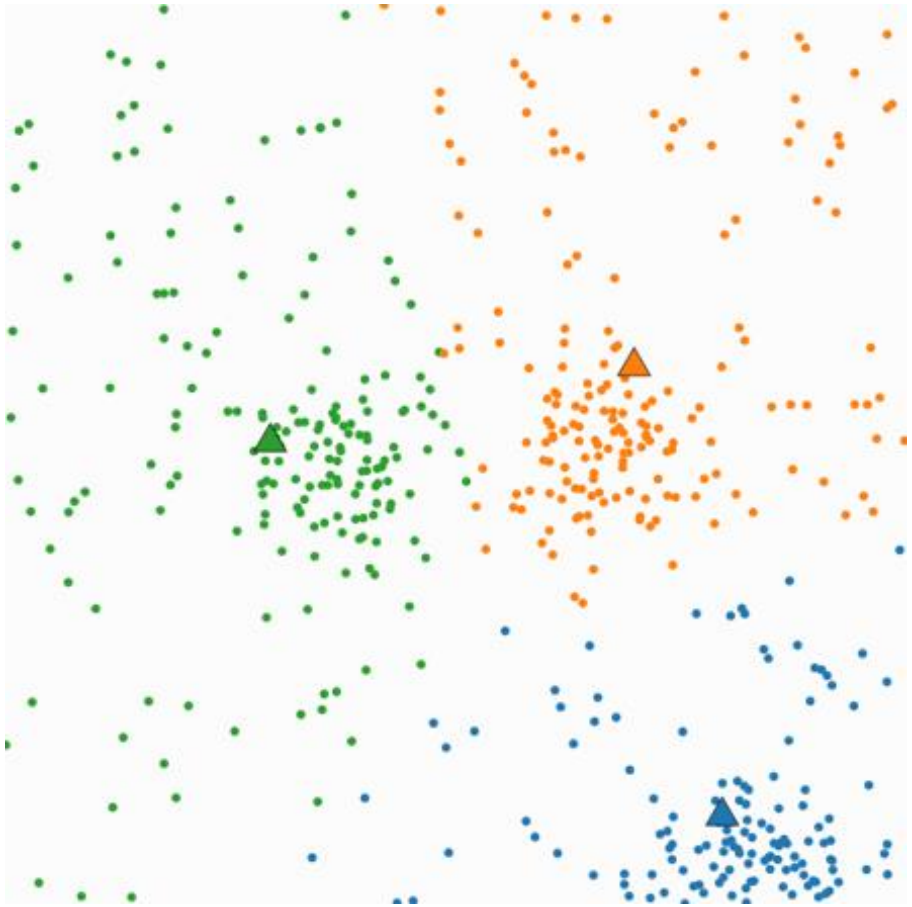
K-means

› Update Centers:



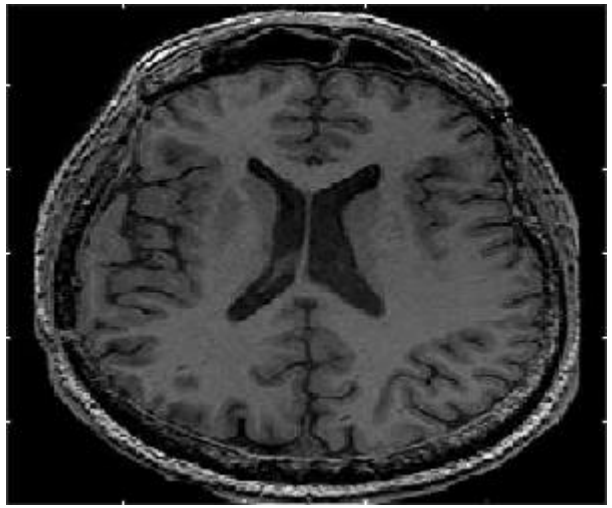
Pixel Classification – K-means

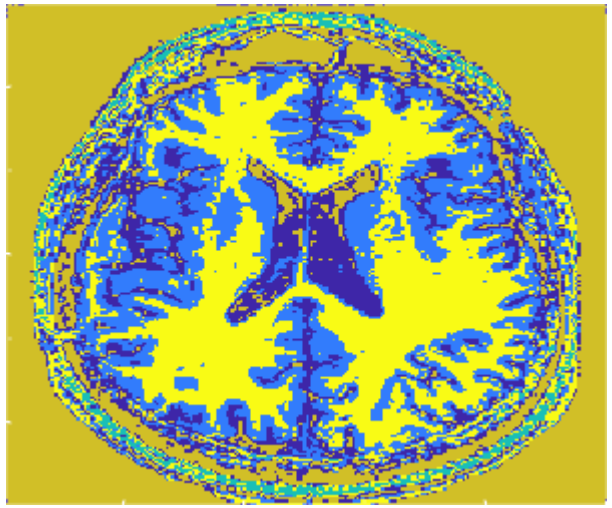
› Final Clusters:



K-means

› MRI Segmentation





Fuzzy C-Means (FCM)

› An useful cost function for soft clustering :

$$J = \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} u_{ij}^{2}\left\|x_i - \omega_j\right\|_2^2, \qquad s.t.\left(\sum_{j=1}^{K} u_{ij} - 1{=}0\right), \forall i = 1,2,\dots,N$$

› Where $q > 1$ {usually $q = 2$) is fuzziness parameter,

› Using Lagrange multiplier:

$$J = \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} u_{ij}^{2}\left\|x_i - \omega_j\right\|_2^2 + \sum_{i=1}^{N}\lambda_i\left(\sum_{j=1}^{K} u_{ij} - 1\right)$$

Fuzzy C-Means (FCM)

› Gives:

$$u_{ij} = \frac{\frac{1}{\|x_i - \omega_j\|_2^2}}{\sum_{l=1}^{K} \frac{1}{\|x_i - \omega_l\|_2^2}} \quad (*)$$

› and:

$$\omega_j = \frac{\sum_{i=1}^{N} u_{ij}^2 \, x_i}{\sum_{i=1}^{N} u_{ij}^2} \quad (**)$$

› Solve $(*)$ and $(**)$ iteratively.

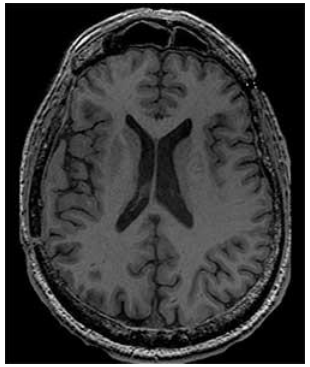
› Useful for partial volume effect.

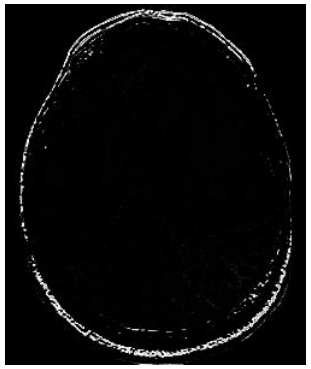
Fuzzy C-Means (FCM)

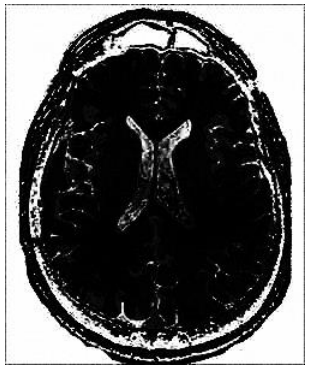
› FCM Example:

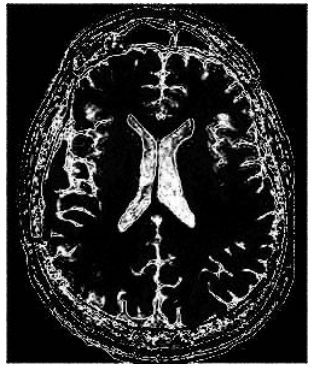
› Image (Top)

› u_{ij}'s maps (Bottom)









GMM (xMM) Clustering

› GMM (xMM) Clustering:

› Recall mixture model formulation:

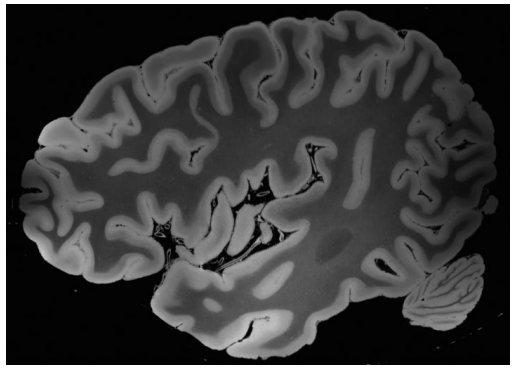
$$p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{k=1}^{m} \pi_k G(\boldsymbol{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$$

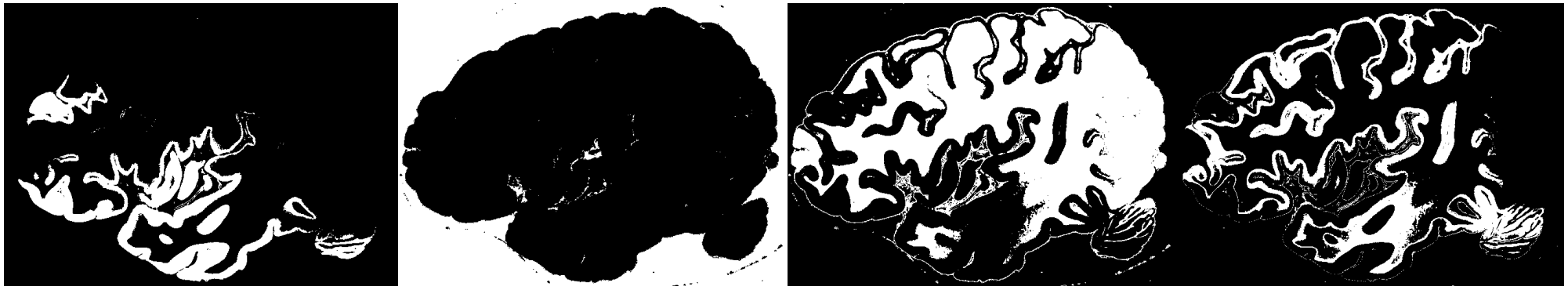
› Each $G(\boldsymbol{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$ is probability map of segment $\#k$

GMM (xMM) Clustering Example

› Input image and Hard Clustering Results (maximum)

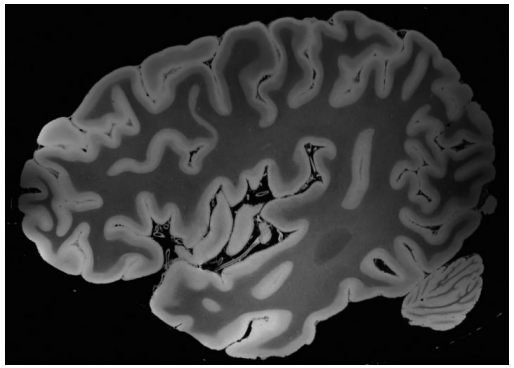
› Feature: raw data!

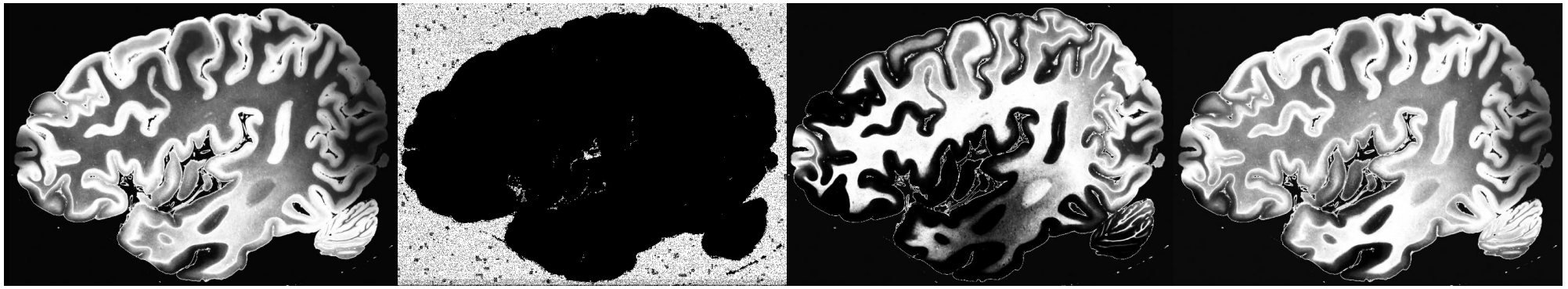




GMM (xMM) Clustering Example

› Input image and mixtures map, $p(\boldsymbol{x}|k)\pi_k$





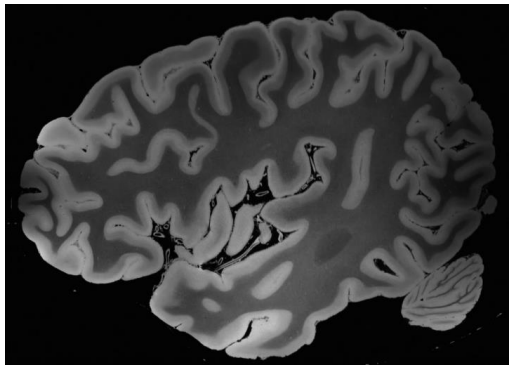
GMM (xMM) Clustering

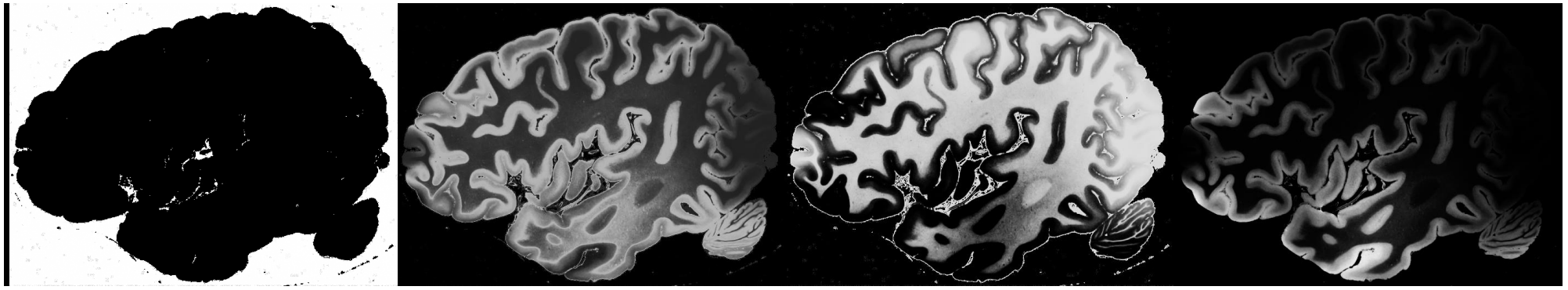
› Probability map of segment #k is calculated using Bayes rule:

$$p(k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|k)\pi_k}{p(\boldsymbol{x})} = \frac{\pi_k G(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k=1}^{m} \pi_k G(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

GMM (xMM) Clustering Example

› Input image and probability map, $p(k|\boldsymbol{x})$.





Meanshift

› Recall KDE formulation:

$$\hat{p}(x) = \frac{1}{Nh^D}\sum_{i=1}^{N} K\left(\frac{\boldsymbol{x}_i - \boldsymbol{x}}{h}\right) = \frac{c_D}{Nh^D}\sum_{i=1}^{N} k\left(\left\|\frac{\boldsymbol{x}_i - \boldsymbol{x}}{h}\right\|^2\right)$$

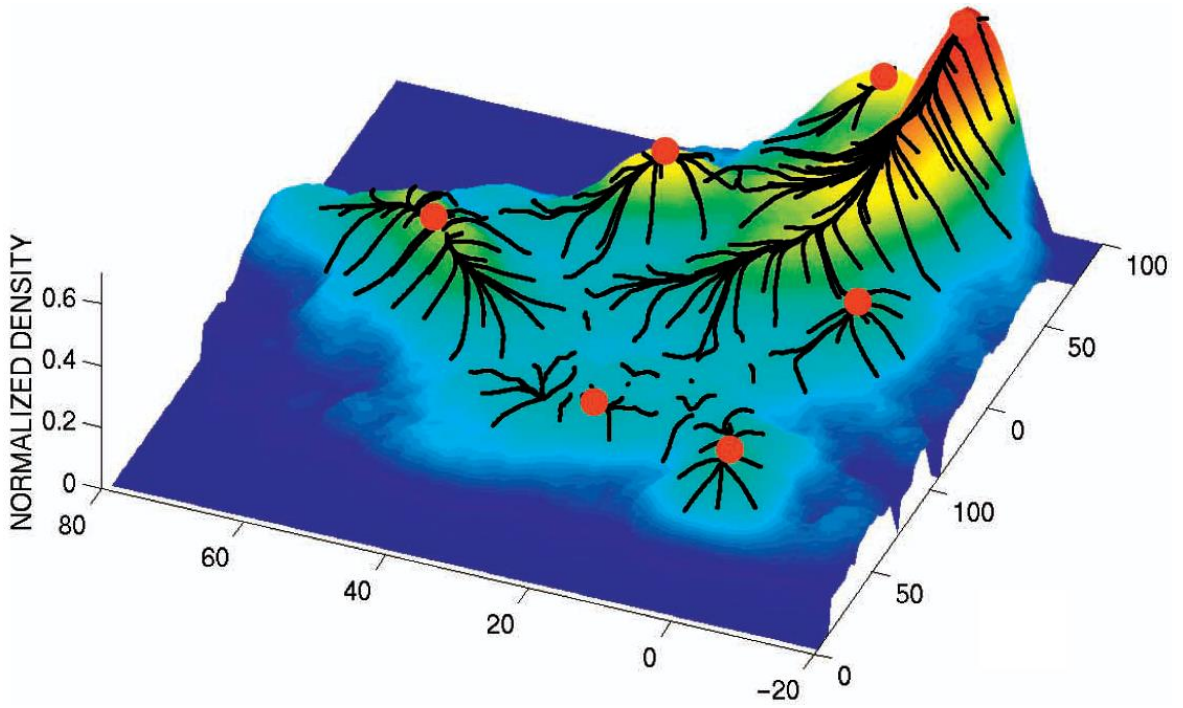
› Meanshift is a method of mode (local maximum) finding in pdf:

› We define the function $g(x) = -k'(x), K(\boldsymbol{x}) = c_D k(\|\boldsymbol{x}\|^2)$

Meanshift

› Modes are *stable* points of the following iteration:

$$\boldsymbol{y} = \frac{\sum_{i=1}^{N} \boldsymbol{x}_i g\left(\left\|\frac{\boldsymbol{x}_i - \boldsymbol{y}}{h}\right\|^2\right)}{\sum_{i=1}^{N} g\left(\left\|\frac{\boldsymbol{x}_i - \boldsymbol{y}}{h}\right\|^2\right)}$$



Meanshift Segmentation

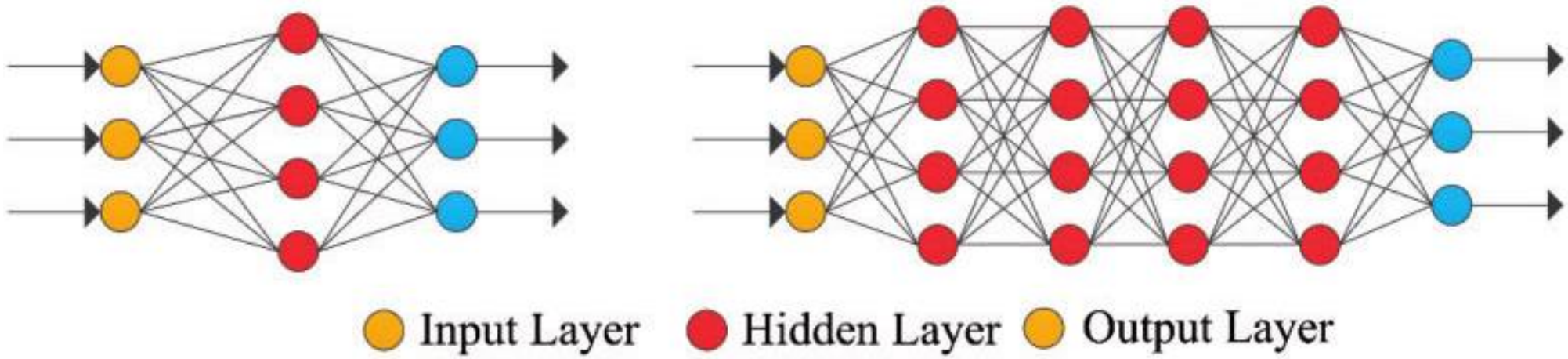
- Algorithm:
- input: $\{\boldsymbol{x}_i\}_{i=1}^{N} \in \mathbb{R}^D$
- Run meanshift procedure starting each $\boldsymbol{x}_i$, as iteration starting point, and store convergence point as $\boldsymbol{z}_i$
- Cluster $\{\boldsymbol{z}_i\}_{i=1}^{N}$ to C cluster by grouping together all $\boldsymbol{z}_i's$ which are closer than a threshold.

Pixel Classification – Supervised

- Any supervised machine learning is useful:
  - MLP (Multi Layer Perceptron)
  - Deep Neural Networks (CNN)
  - SVM (Support Vector Machine)
  - RBFN (Radian Basis Function Network)
  - LDA (Linear Discrimination Analysis)
  - Decision Tree
  - Bayes/Naïve Bayes
  - …

Pixel Classification – Supervised

› Multi Layer Perceptron (Shallow or Deep): A Highly nonlinear Mapper



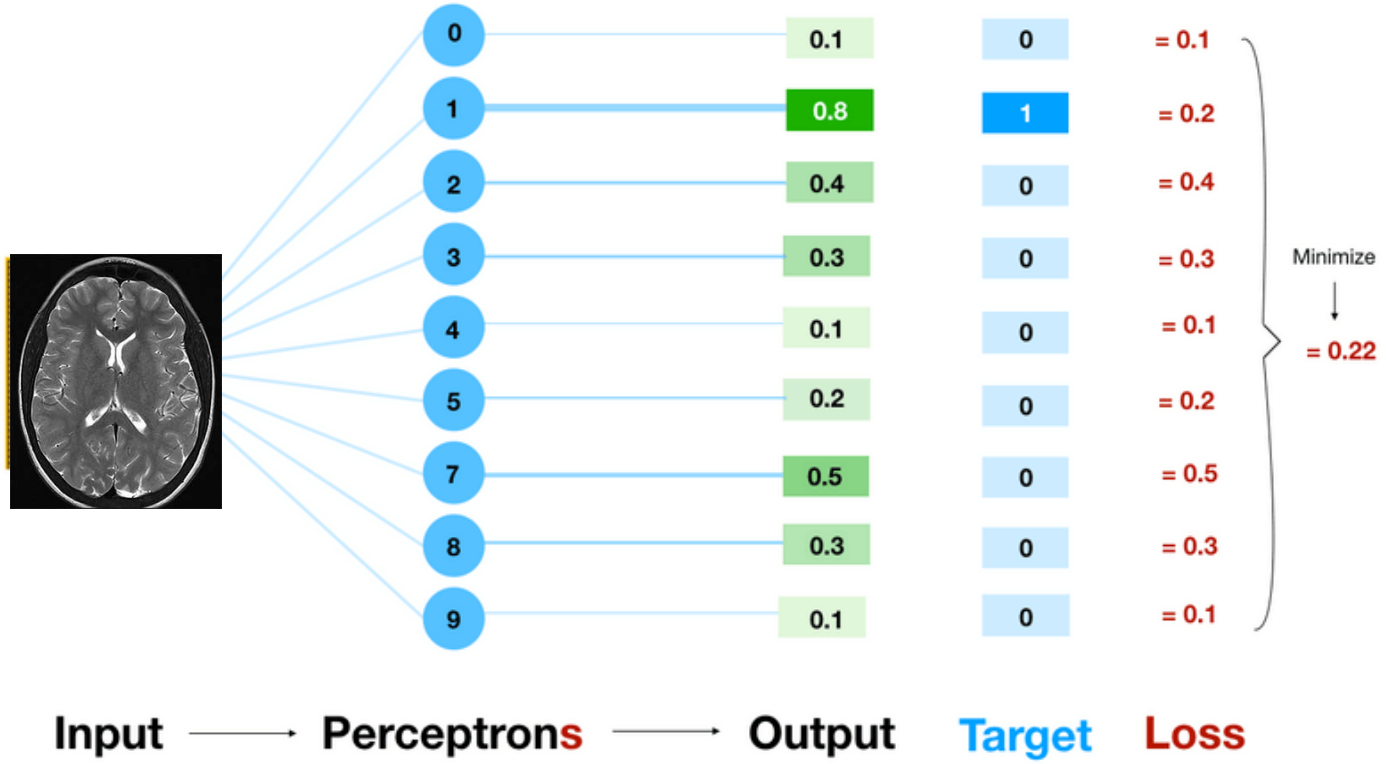
Pixel Classification – MLP

- MLP for segmentation:
  - Input: feature vector
  - Output: One-Hot label for each segment
  - One-Hot label:

[0,0,0,1,0,0]

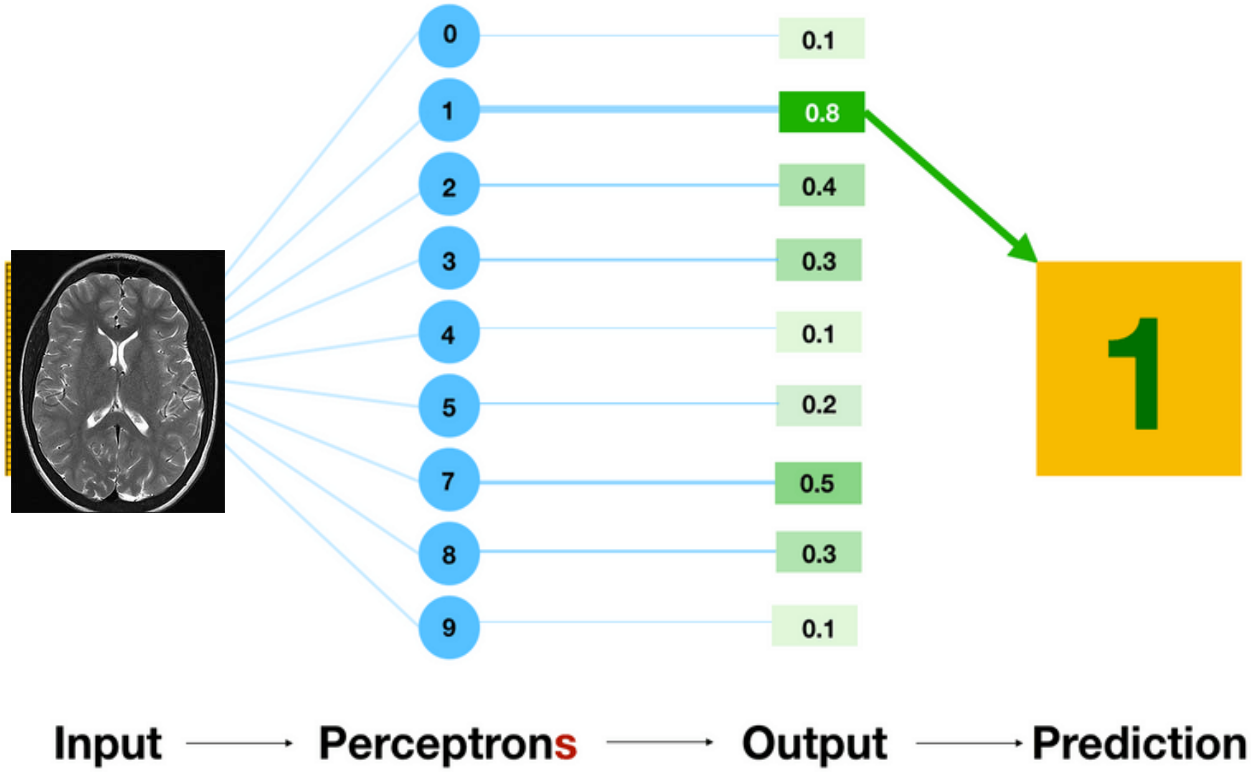
Pixel Classification – MLP

› MLP Training with one-hot encoding in output:



Pixel Classification – MLP

› MLP Test with one-hot encoding in output:



Pixel Classification – Image Segmentation in Presence of Intensity Inhomogeneity

› Intensity Inhomogeneity model (MRI):

› Notation:

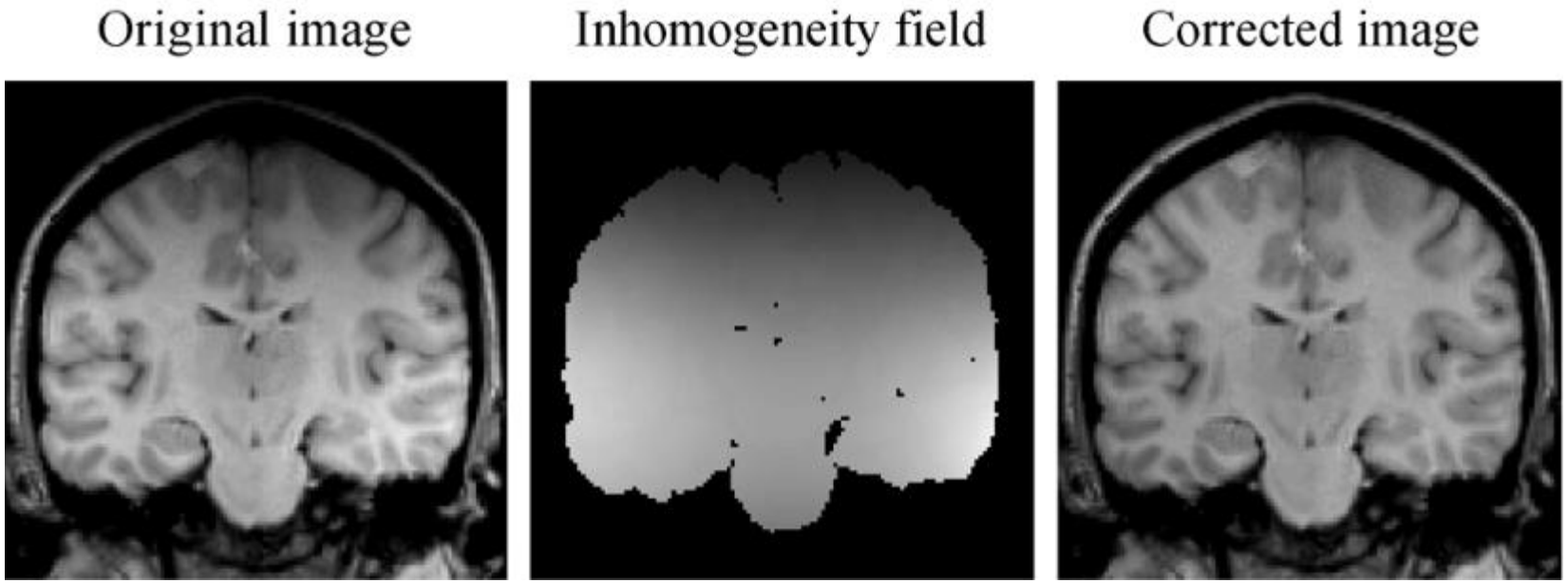
- $u(\boldsymbol{x})$: inhomogeneity and noise free image
- $v(\boldsymbol{x})$: acquired image (corrupted with noise and inhomogeneity)
- $n(\boldsymbol{x})$: additive noise
- $b(\boldsymbol{x})$: intensity inhomogeneity field, slow variation and smooth

› Models:

- $v(\boldsymbol{x}) = u(\boldsymbol{x})b(\boldsymbol{x}) + n(\boldsymbol{x})$
- $v(\boldsymbol{x}) = \big(u(\boldsymbol{x}) + n(\boldsymbol{x})\big)b(\boldsymbol{x})$
- $logv(\boldsymbol{x}) = logu(\boldsymbol{x}) + logb(\boldsymbol{x}) + n'(\boldsymbol{x})$

Pixel Classification – Image Segmentation in Presence of Intensity Inhomogeneity

› Illustration:



Pixel Classification – Image Segmentation in Presence of Intensity Inhomogeneity

- Methods:
  - Preprocessing
  - Intensity Inhomogeneity aware segmentation

Pixel Classification – Image Segmentation in Presence of Intensity Inhomogeneity

› Preprocessing (filtering) approaches:

› Model:

$$logv(\boldsymbol{x}) = logu(\boldsymbol{x}) + logb(\boldsymbol{x}) + n'(\boldsymbol{x})$$

› Restoration:

$$log\hat{u}(\boldsymbol{x}) = logv(\boldsymbol{x}) - \underbrace{LPF\big(logv(\boldsymbol{x})\big)}_{log\big(b(\boldsymbol{x})\big)} + C_N$$

› where the normalization constant, C_N, is added to preserve the mean or maximum intensity of the corrected image.

Pixel Classification – Image Segmentation in Presence of Intensity Inhomogeneity

› Preprocessing (filtering) approaches:

› Model:

$$v(\boldsymbol{x}) = \big(u(\boldsymbol{x}) + n(\boldsymbol{x})\big)b(\boldsymbol{x})$$

› Restoration:

$$\hat{u}(\boldsymbol{x}) = C_N v(\boldsymbol{x}) \Big/ \underbrace{LPF\big(v(\boldsymbol{x})\big)}_{b(\boldsymbol{x})}$$

› where the normalization constant, C_N, is added to preserve the mean or maximum intensity of the corrected image.

Pixel Classification – Image Segmentation in Presence of Intensity Inhomogeneity

- *Intensity Inhomogeneity Aware* segmentation:
- Two core article:
  - Adaptive Fuzzy C-Means (AFCM)
  - Bias Corrected FCM (BCFCM)

Adaptive Fuzzy C-Means (AFCM):

› Notation:

- $y(i,j)$: Acquired image intensity at location (i,j),
- C: # of segments,
- q: fuzziness parameter of algorithm (we assume $q = 2$),
- $u_k(i,j)$: Membership value at pixel location (i,j) for segment $\#k$,
- v_k: Centroid of segment $\#k$

› Conventional Fuzzy C-Means (FCM) cost function:

$$J_{FCM} = \sum_{(i,j)} \sum_{k=1}^{C} u_k^2(i,j) \|y(i,j) - v_k\|_2^2, \qquad s.t. \sum_{k=1}^{C} u_k(i,j) = 1$$

Adaptive Fuzzy C-Means (AFCM):

› AFCM cost function:

$$J_{AFCM} = \sum_{(i,j)} \sum_{k=1}^{C} u_k^2(i,j) \| y(i,j) - m(i,j) v_k \|_2^2 + \cdots$$

$$\lambda_1 \sum_{(i,j)} \left((m(i,j) * D_i)^2 + \left(m(i,j) * D_j \right)^2 \right) + \cdots$$

$$\lambda_2 \sum_{(i,j)} \left((m(i,j) ** D_{ii})^2 + 2\left(m(i,j) ** D_{ij} \right)^2 + \left(m(i,j) ** D_{jj} \right)^2 \right)$$

› $m(i,j)$: unknown multiplier field (model the brightness variation), slow variation!

› D_i and D_j are 1st order finite Difference operator for partial derivatives $(\frac{\partial m}{\partial x}, \frac{\partial m}{\partial y})$

› D_{ii}, D_{jj} and D_{ij} are 2nd order finite Difference operator for partial derivatives $(\frac{\partial^2 m}{\partial x^2}, \frac{\partial^2 m}{\partial y^2}, \frac{\partial^2 m}{\partial x \partial y})$

Adaptive Fuzzy C-Means (AFCM):

› The second and third terms are measure of smoothness:

› For 2D continuous function f:

$$S_1 = \iint_{-\infty}^{+\infty} \left(\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right) dxdy = \iint_{-\infty}^{+\infty} |\nabla f|^2 dxdy$$

$$S_2 = \iint_{-\infty}^{+\infty} \left(\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right) dxdy = \iint_{-\infty}^{+\infty} \|H(f)\|_F^2 dxdy$$

Adaptive Fuzzy C-Means (AFCM):

› Algorithm steps:

1. Initial guess for $\{v_k\}_{k=1}^{C}$ using k-means, FCM, … and set $m(i,j) = 1$
2. Compute membership functions:

$$u_k(i,j) = \frac{\|y(i,j) - m(i,j)v_k\|^{-2}}{\sum_{l=1}^{C}\|y(i,j) - m(i,j)v_l\|^{-2}}$$

3. Compute centroids:

$$v_k = \frac{\sum_{(i,j)} u_k^2(i,j)m(i,j)y(i,j)}{\sum_{(i,j)} u_k^2(i,j)m^2(i,j)}$$

4. Compute multiplier field:

$$y(i,j)\sum_{k=1}^{C} u_k^2(i,j)v_k = m(i,j)\sum_{k=1}^{C} u_k^2(i,j)v_k^2 + \lambda_1\big(m(i,j) ** H_1(i,j)\big) + \lambda_2\big(m(i,j) ** H_2(i,j)\big)$$

where $H_1(i,j) = D_i * \check{D}_i + D_j * \check{D}_j$ and $H_2(i,j) = D_{ii} * \check{D}_{ii} + +2\big(D_{ij} * \check{D}_{ij}\big) + D_{jj} * \check{D}_{jj}$, $\check{f}(i) = f(-i)$

5. Step over 2-3-4 until convergence

The End

› AnY QuEsTiOn?