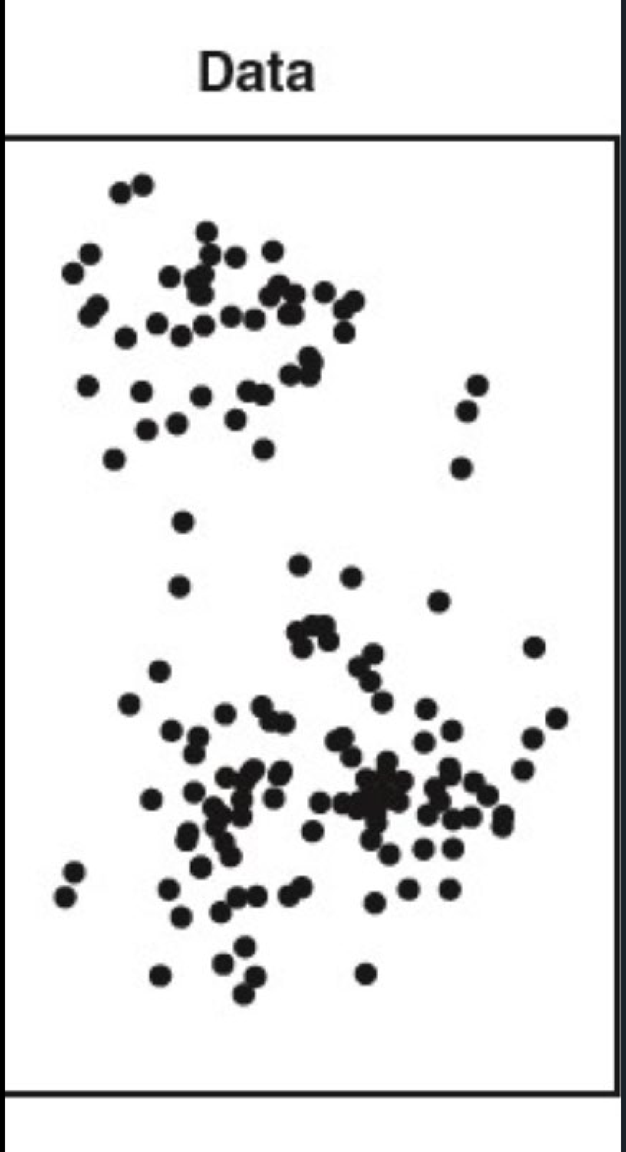
(Unsupervised?)

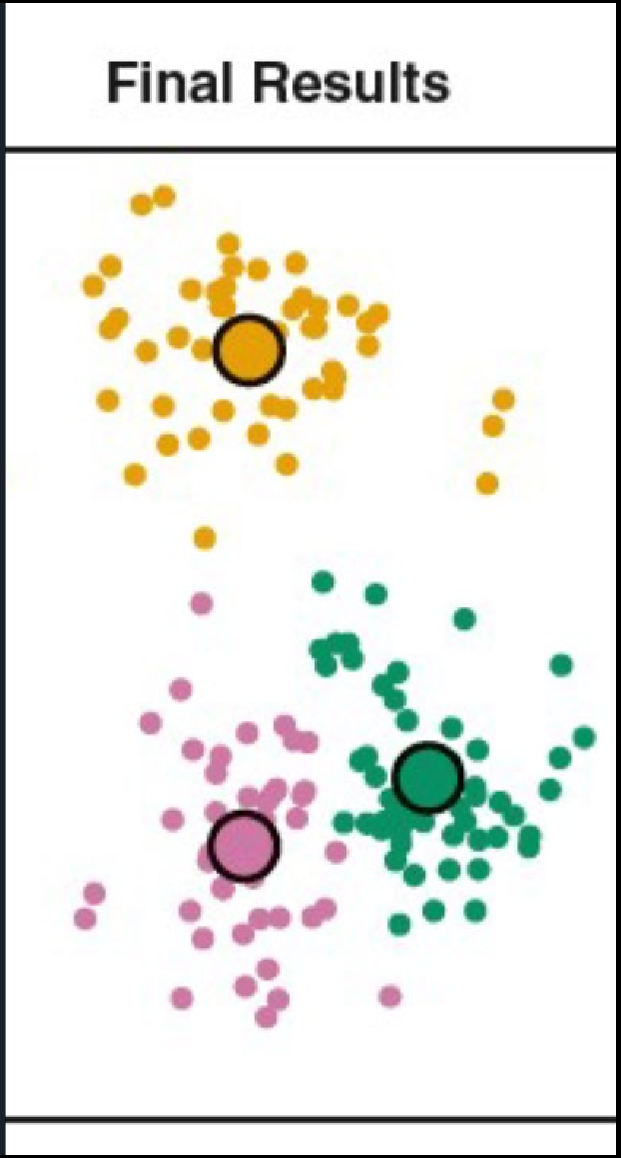
Machine Leaning Techniques in REst-State fMRI

Hanning Su



K-mean clustering an introduction

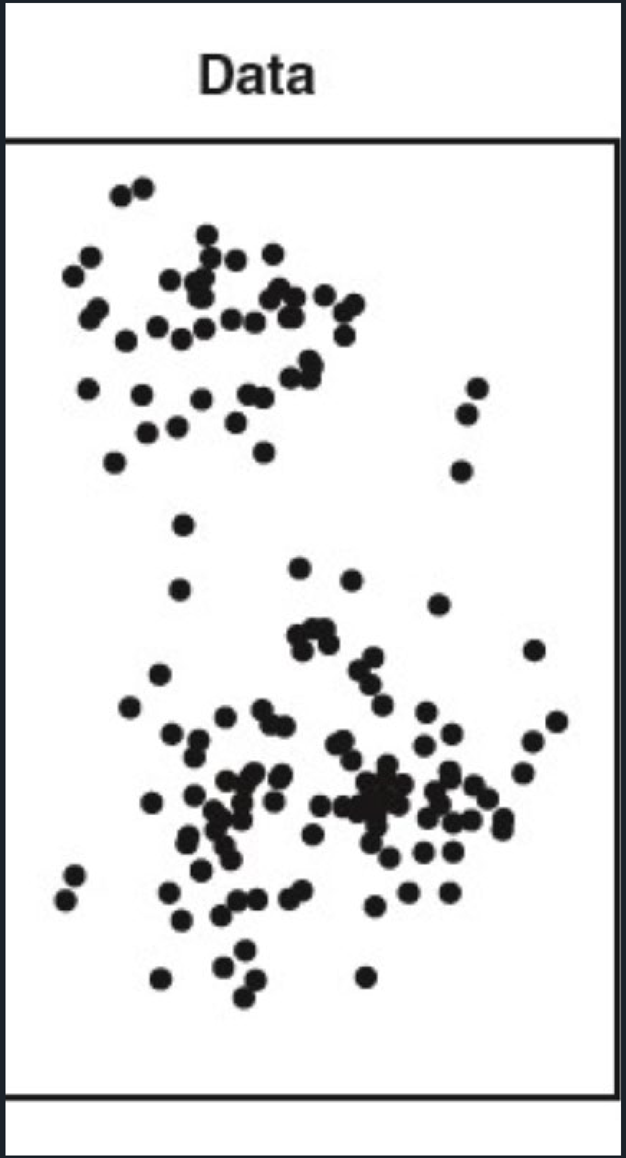
- K-means clustering is a simple and elegant approach for partitioning a dataset into K distinct, non-overlapping clusters.
- To perform K-means clustering, we must first specify the desired number of clusters K, then the K-means algorithm will assign each observation to exactly one of the K clusters.



James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.

K-means Introduction

- Let $C_1, \ldots, C_K$ denote sets containing the indices of the observations in each cluster. These sets satisfy two properties:
- 1. $C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \ldots, n\}$. In other words, each observation belongs to at least one of K clusters.
- 2. $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$. In other words, the clusters are nonoverlapping: no observation belongs to more than one cluster.

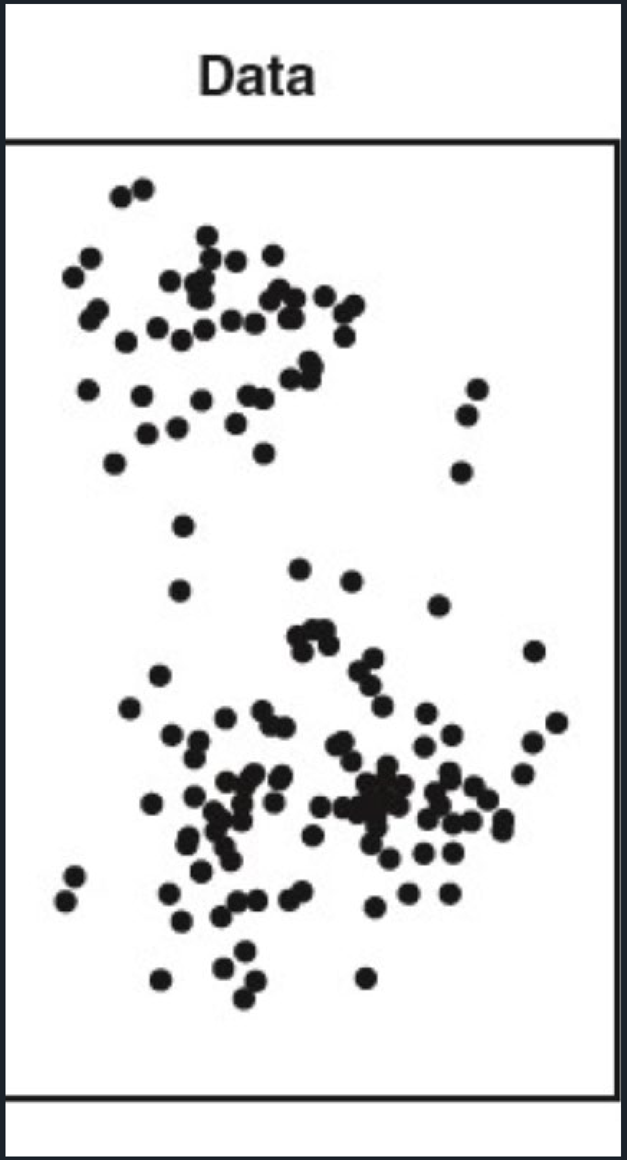


K-means Introduction

- Data format:

Observation (i or I’)	Variable1	Variable2	Variable3
1	3.22	5.52	101.89
2	6.77	10.88	120.55
3	8.30	11.4	130.44
4	111.63	1000.33	0.11
5	108.29	1199.21	0.0032
..			

Note: here, p = number of explanatory variables = 3



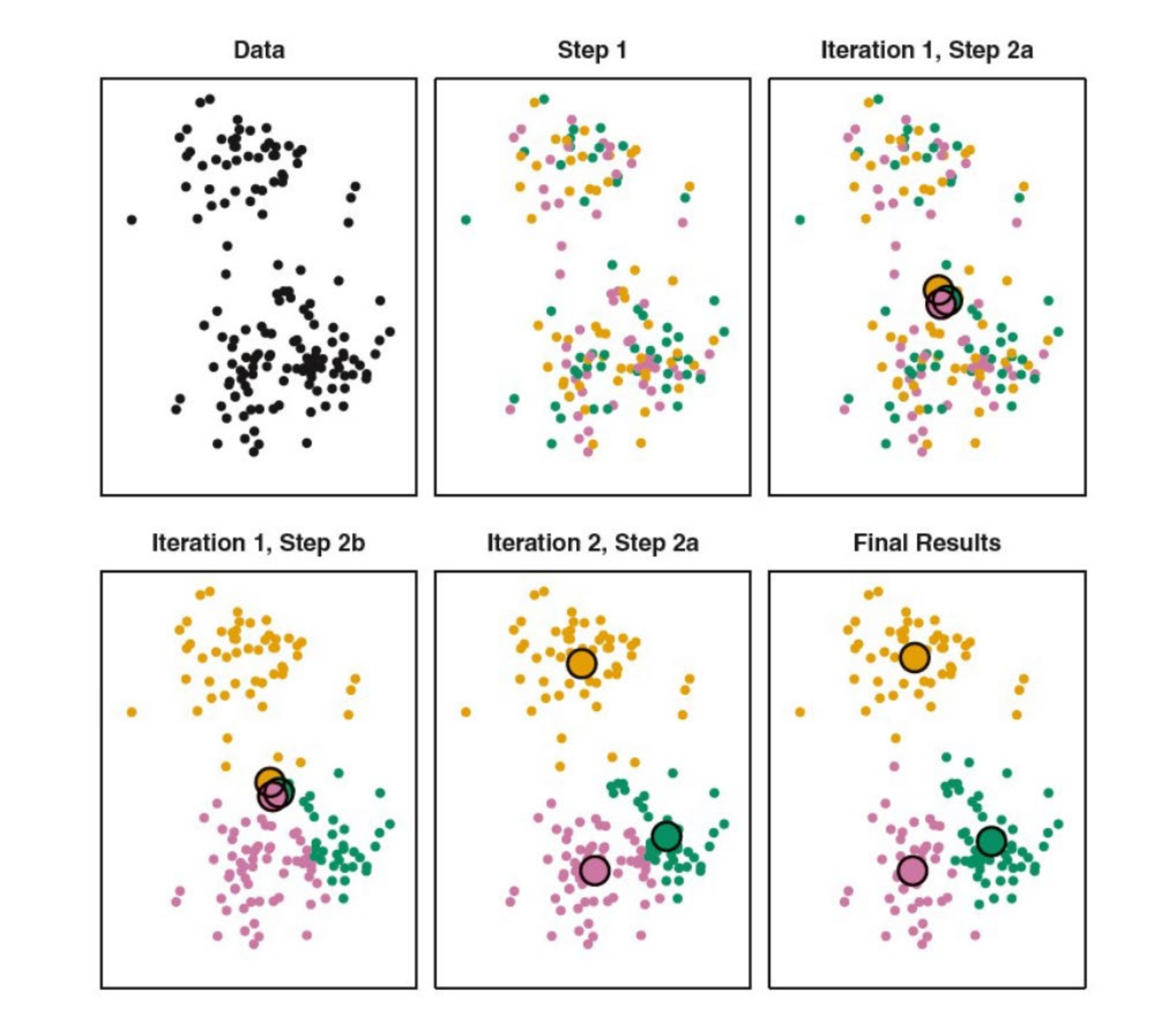
K-means Introduction

- The idea behind K-means clustering is that a good clustering is one for which the within-cluster variation is as small as possible.
- The within-cluster variation for cluster C_k is a measure $W(C_k)$ that is defined as (second equality proof omitted):
- $W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$
- Where $|C_k|$ denotes the number of observations in C_k, and $\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$
- Hence, we have the optimization problem:
- $\min_{C_1, \ldots, C_k} \{\sum_{k=1}^{K} W(C_k)\}$

K-means Clustering Algorithm

- 1. Randomly assign a number from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations, denoted by $C_1^{(1)}, \dots, C_K^{(1)}$. Let $l = 1$.
- 2. (a) For $C_k^{(l)}, k = 1, \dots, K$, compute the cluster *centroid*
- $(\bar{x}_{k1}^{(l)}, \dots, \bar{x}_{kp}^{(l)})$, i.e. the vector of the p feature means of the observations in $C_k^{(l)}$.
- (b) Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance) to generate new clusters $C_1^{(l+1)}, \dots, C_K^{(l+1)}$. If $C_1^{(l+1)}, \dots, C_K^{(l+1)}$ are different from $C_1^{(l)}, \dots, C_K^{(l)}$, then let $l = l + 1$ and repeat Step2; otherwise, return $C_1^{(l)}, \dots, C_K^{(l)}$ as output.

K-means Clustering Algorithm



James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.

Simple exercise

Centroid	variable1	variable2
1	1000	1000
2	10	10

Observation	variable1	variable2
1	9.99	10.2
2	2000	990
3	0.1	0.3

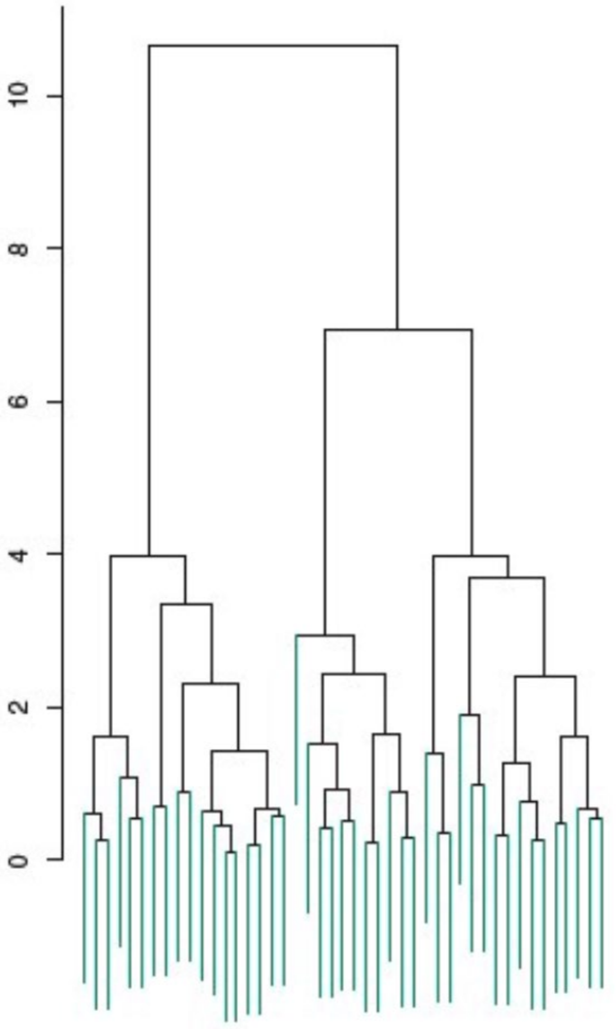
- How to assign the observations?

Papers utilized K-means

- Lee M H, Hacker C D, Snyder A Z, Corbetta M, Zhang D, Leuthardt E C. Clustering of resting state networks. PLoS ONE 2012;7(7):e40370.

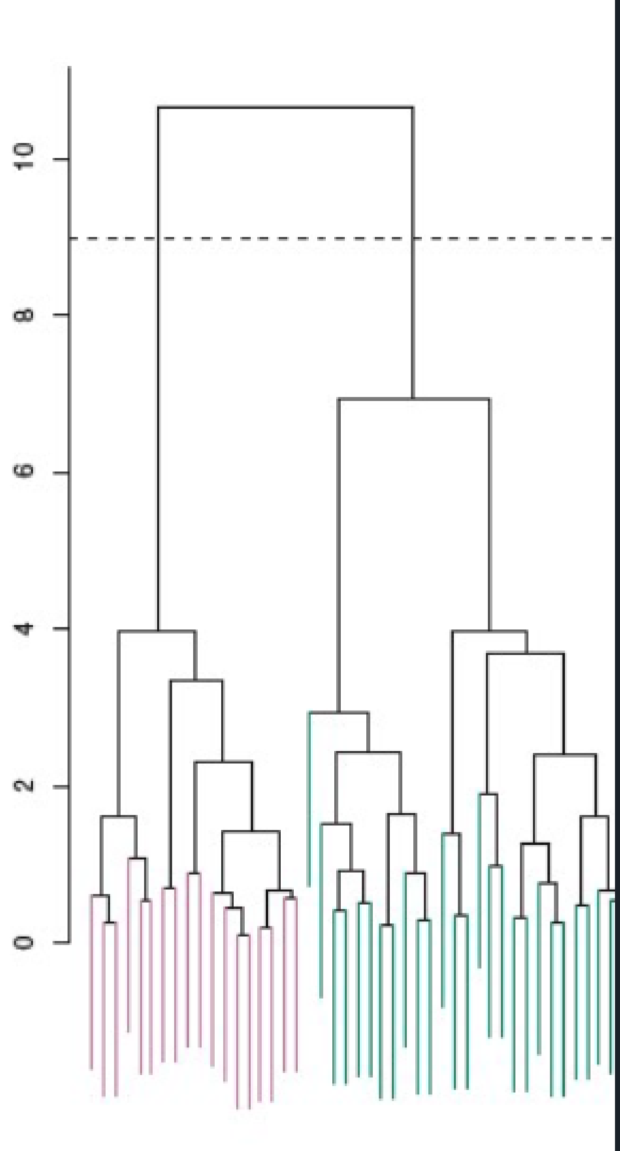
- Kim J H, Lee J M, Jo H J, Kim S H, Lee J H, Kim S T. Defining functional SMA and pre-SMA subregions in human MFC using resting state fMRI: functional connectivity - based parcellation method. Neuroimage 2010;49(3):2375–86.

Hierarchical Clustering Introduction



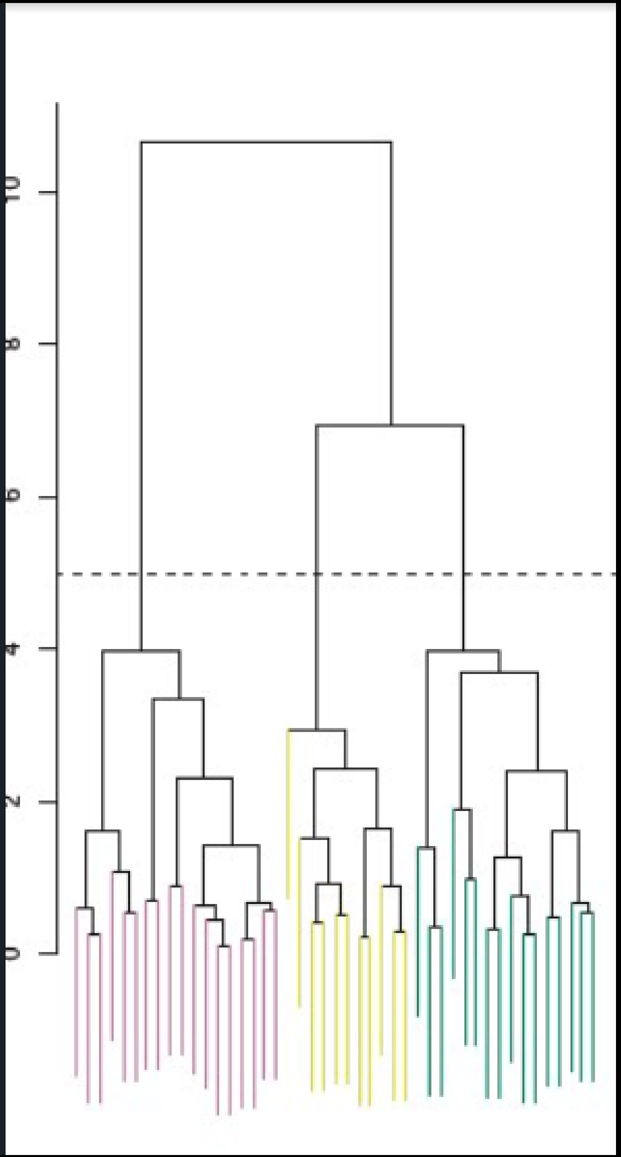
- In the K-means clustering, we have to pre-specify K, which can be a disadvantage.
- Hierarchical clustering results in an attractive tree-based representation of the observations, called a dendrogram.
- By cutting at different height, we can alter the number of clusters easily (more on this later).

James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.

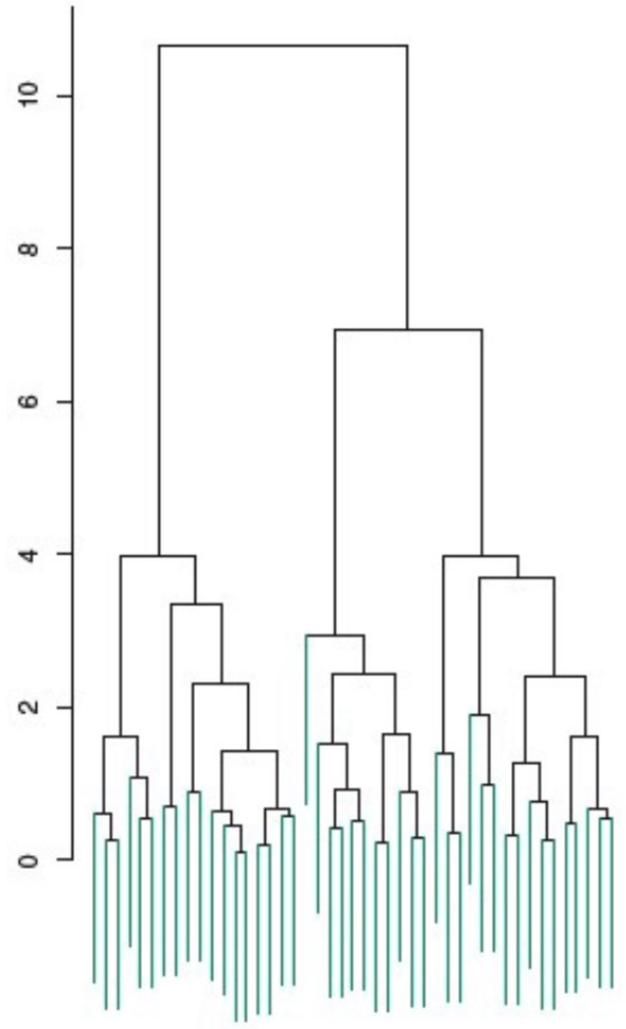


Hierarchical Clustering Introduction

- Here, I describe the bottom up or agglomerate clustering, which refers to the fact that a dendrogram is built starting from the leaves and combining clusters up to the trunk.

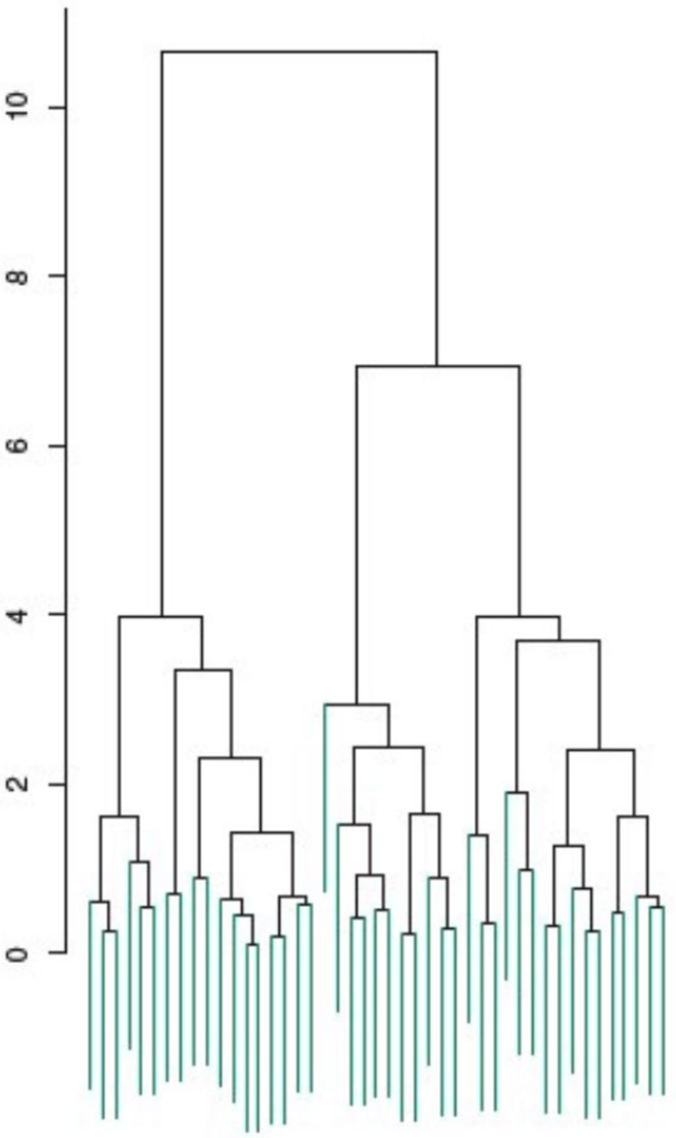


James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.



Hierarchical Clustering Introduction

- Each leaf of the dendrogram represents one of observations (45 observations in figure).
- As we move up the tree, some leaves begin to fuse into branches. These corresponds to observations that are similar to each other.
- As we move higher up the tree, branches themselves fuse, either with leaves or other branches



Hierarchical Clustering Introduction

- Observations that fuse at the very bottom of the tree are quite similar to each other, whereas observations that fuse close to the top of he tree will tend to be quite different.

- For any two observations, we can look for the point in the tree where branches containing those two observations are first fused. The height of this fusion as measured on the vertical axis, indicates how different the two observations are.

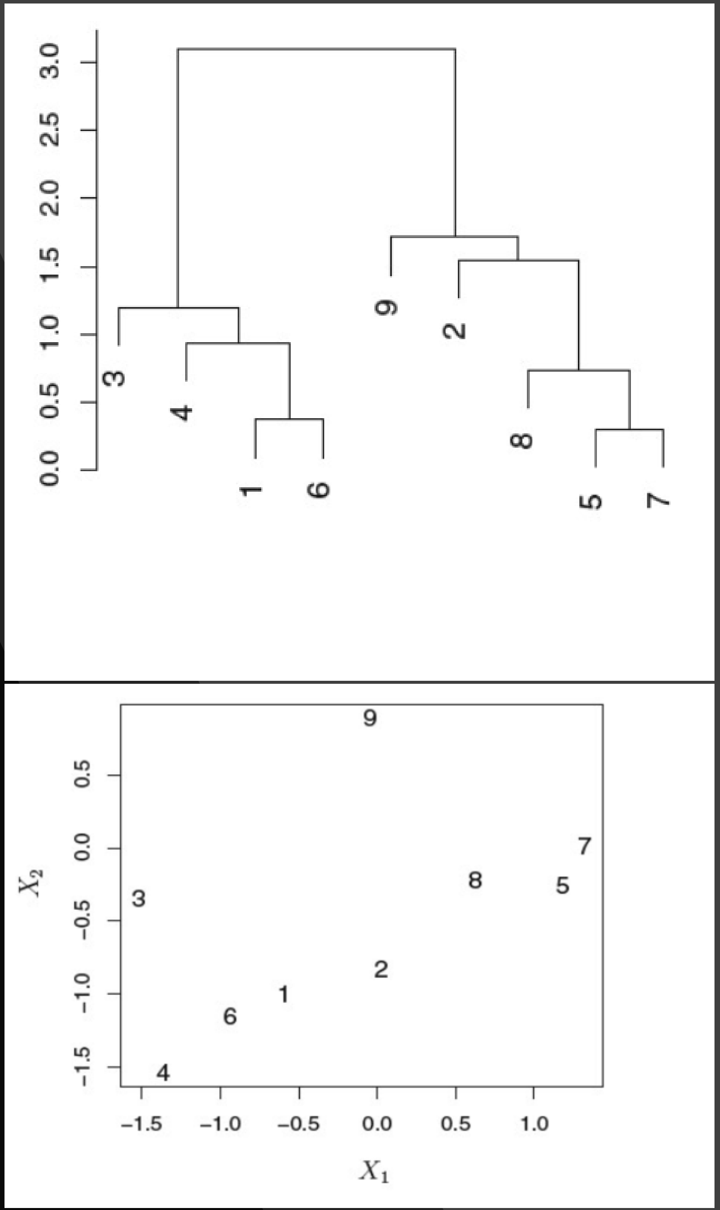
Hierarchical clustering algorithm

- 1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = \frac{n(n-1)}{2}$ pairwise dissimilarities. Treat each observation as its own cluster.
- 2. For $i = n, n-1, \dots, 2$:
- (a) Examine all pairwise inter-cluster dissimilarities (more later) among the i clusters and identify the pair of clusters that are least dissimilar (most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.
- (b) Compare the new pairwise inter-cluster dissimilarities among $i - 1$ remaining clusters.

Identifying clusters using a dendrogram

- To identify clusters, we make a horizontal cut across the dendrogram. The distinct sets of observations beneath the cut can be interpreted as clusters
- The height of the cut to the dendrogram serves as the K in K-means clustering it controls the number of clusters obtained.

James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.



Dissimilarity between two clusters

- Dissimilarity between a pair of observations can be measured by Euclidean distance (there exist other measures).
- Example: observation1: (a_1, a_2, a_3); observation2: (b_1, b_2, b_3)
- Euclidean distance $\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2}$
- If instead we have two groups of observations:

Group1 Observation	variable1	variable2
1	a1	a2
2	b1	b2

Group2 Observation	variable1	variable2
1	c1	c2
2	d1	d2
3	e1	e2

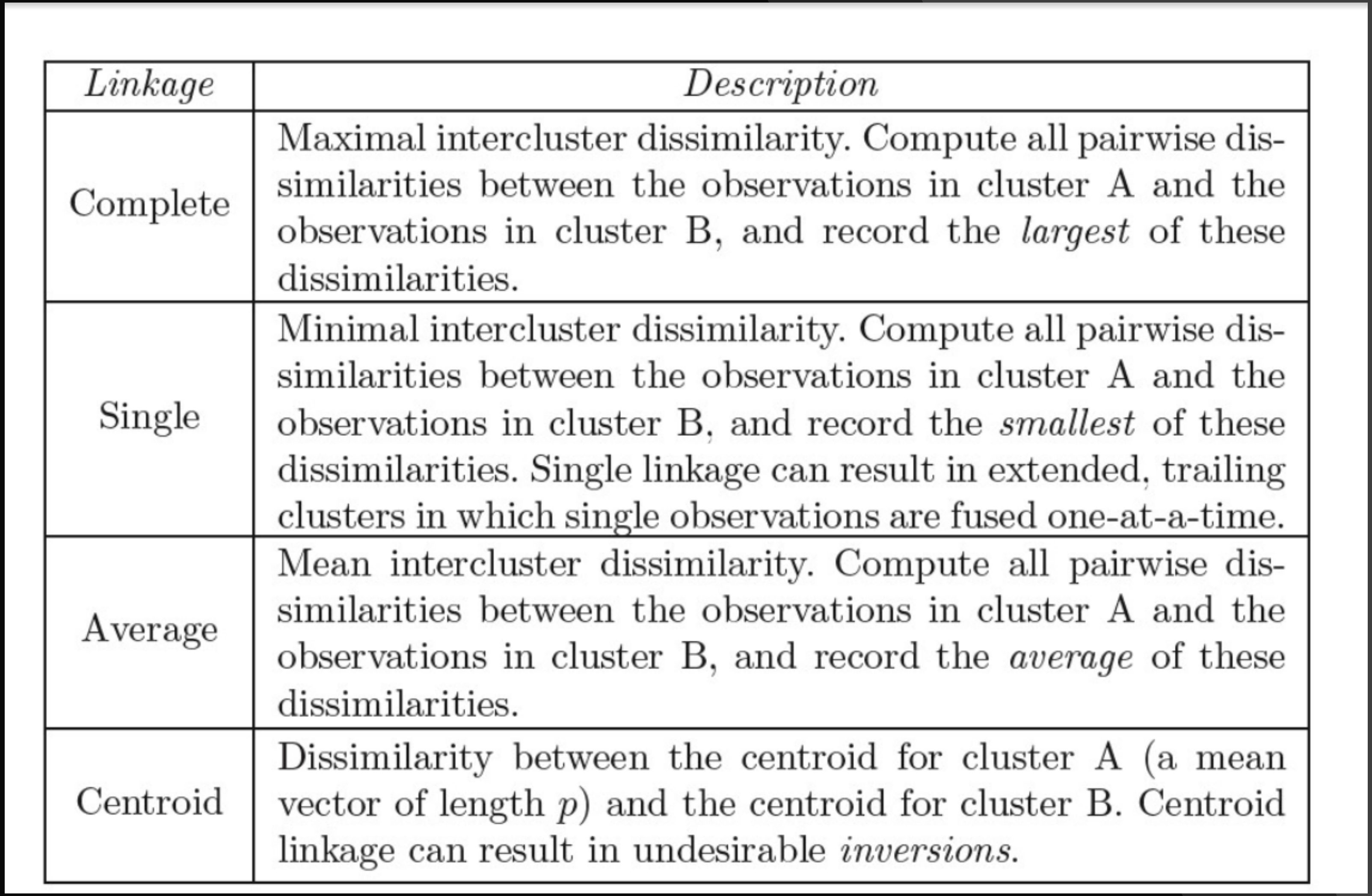
Dissimilarity between two clusters

Group1 Observation	variable1	variable2
1	a1	a2
2	b1	b2

Group2 Observation	variable1	variable2
1	c1	c2
2	d1	d2
3	e1	e2

- How to measure the dissimilarity between the two clusters?
- We need to extend the concept of dissimilarity to a pair of groups of observations.
- The extension is achieved by developing the notion of linkage.
- Details next page..

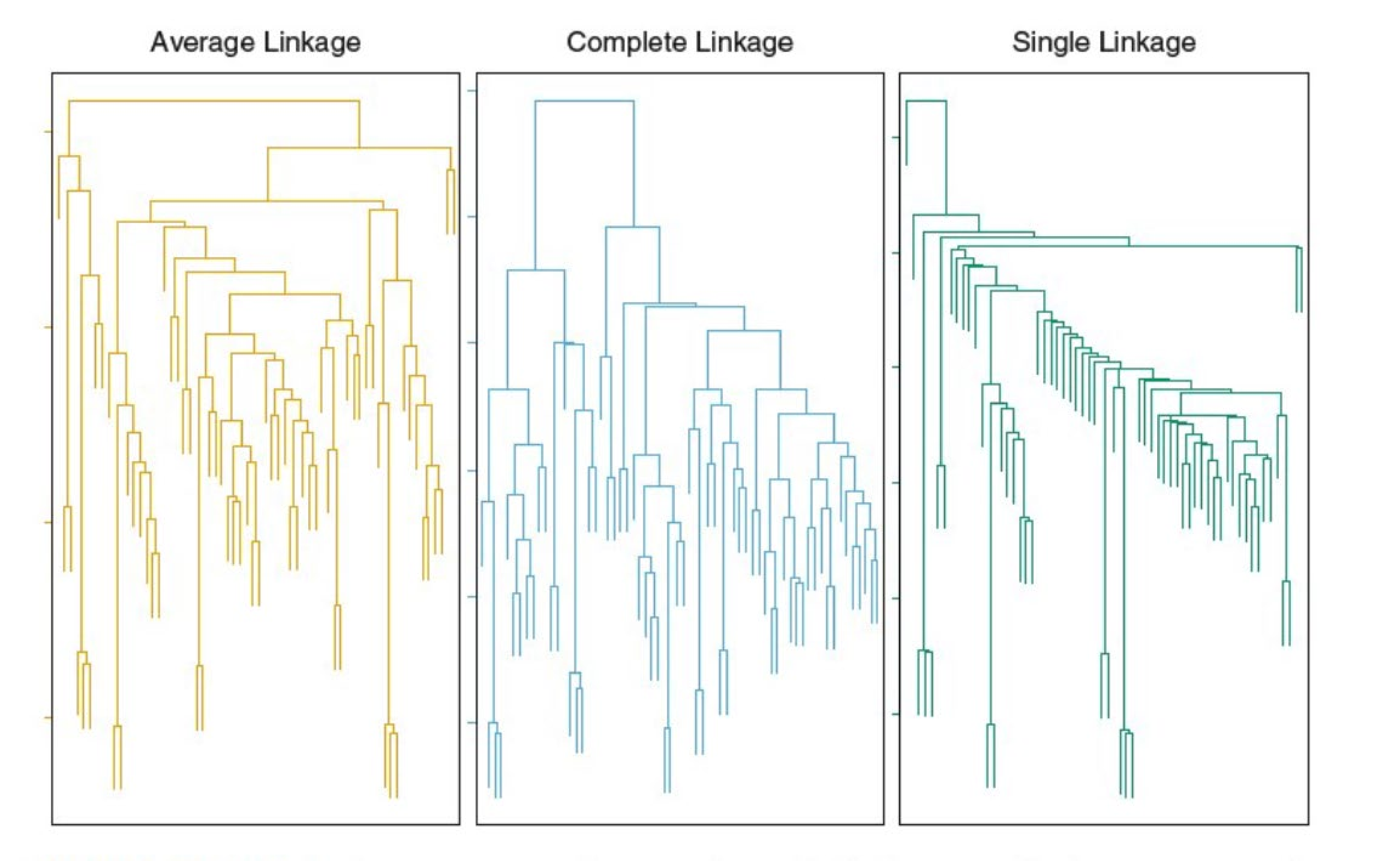
Dissimilarity between two clusters



Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.

Resulting clusters using different types of linkage



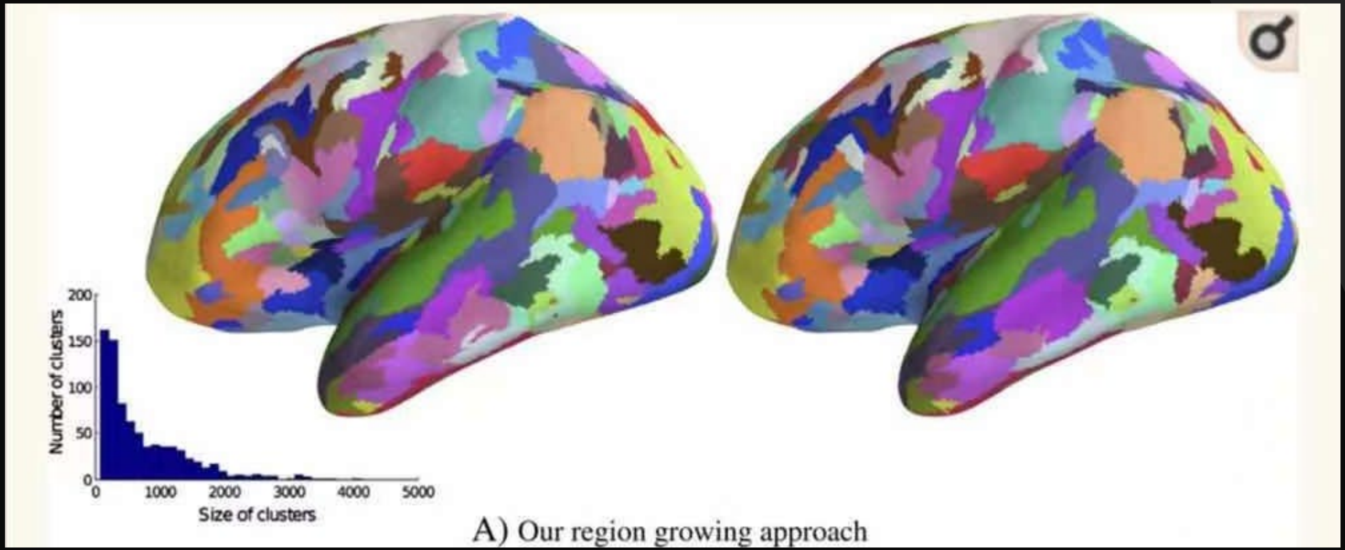
James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.

Paper to be reviewed

Blumensath T, Jbabdi S, Glasser M F, Van Essen D C, Ugurbil K, Behrens T E. Spatially constrained hierarchical parcellation of the brain with resting-state fMRI. Neuroimage 2013;76:313–24.

In this paper, the authors introduce a computational strategy to partition the cerebral cortex into disjoint, spatially contiguous and functionally homogeneous parcels (to conduct a cortical parcellation).

Cortical parcellations provide fundamental maps for functional neuroimaging and provide first abstractions on which many models of brain function are based.



A) Our region growing approach

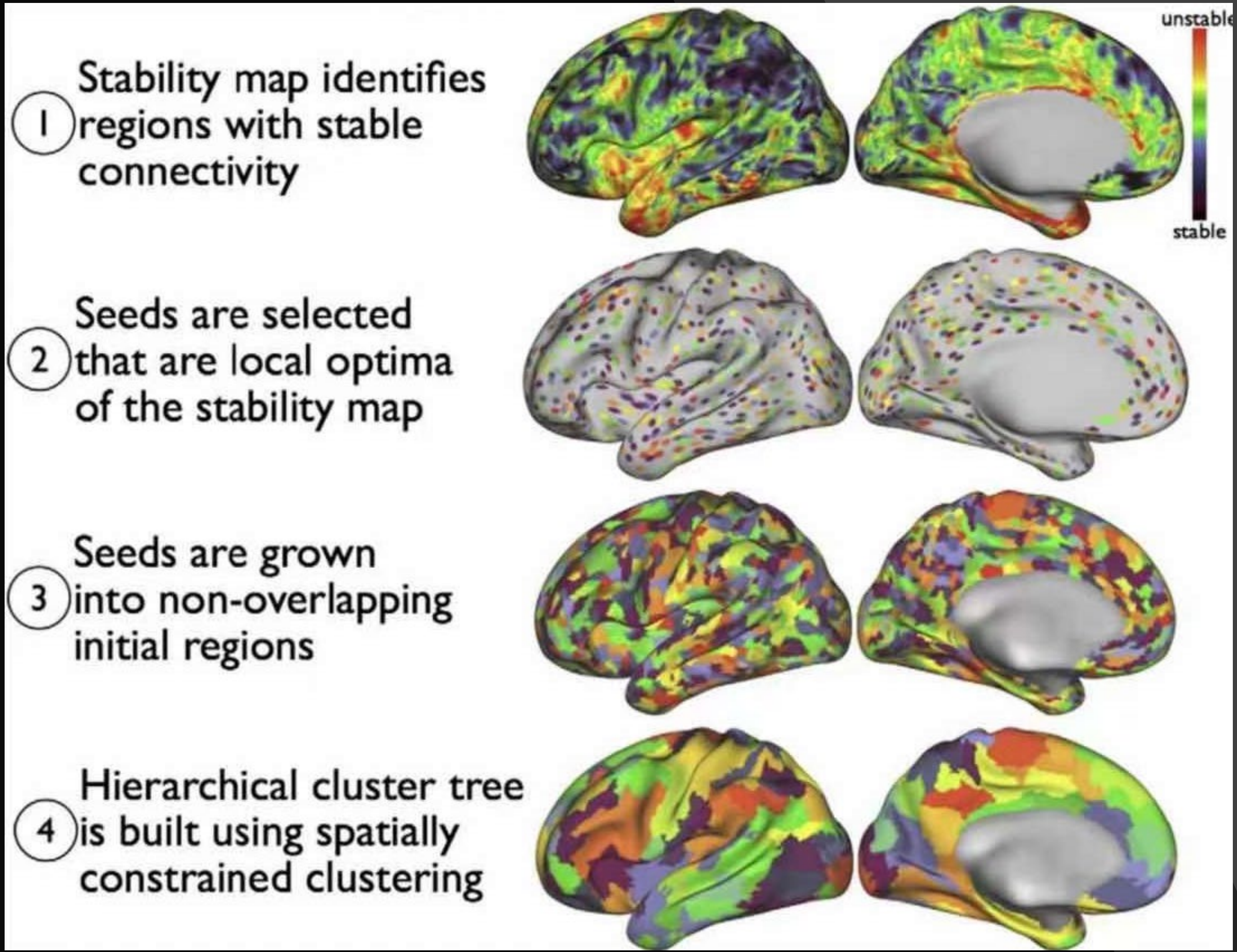
The goal:

- Properties of fMRI data acquired during “rest” (rs-fMRI) can provide both functional homogeneity and functional connectivity (Biswal et al., 1995)

- The authors aim to develop a robust and fully automated technique that uses rs-fMRI data to produce reliable parcellations of the entire human cerebral cortex.

- These parcellations are envisaged to delineate fundamental functional subdivisions of the brain and this reflect subject specific brain organization

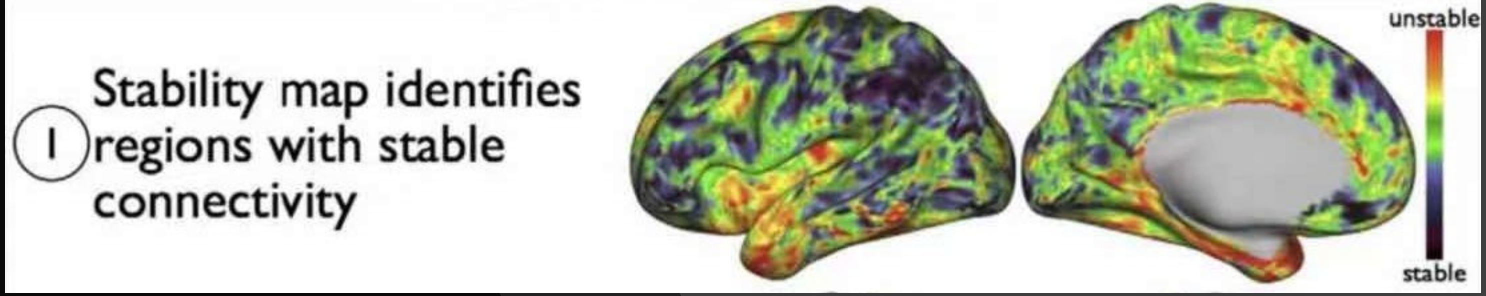
Steps:



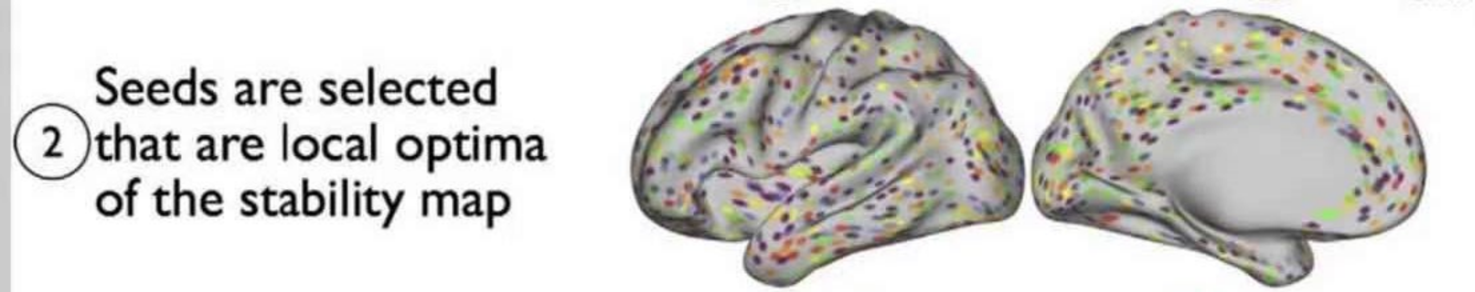
1. Stability map identifies regions with stable connectivity
2. Seeds are selected that are local optima of the stability map
3. Seeds are grown into non-overlapping initial regions
4. Hierarchical cluster tree is built using spatially constrained clustering

Blumensath T, Jbabdi S, Glasser M F, Van Essen D C, Ugurbil K, Behrens T E. Spatially constrained hierarchical parcellation of the brain with resting-state fMRI. Neuroimage 2013;76:313–24.

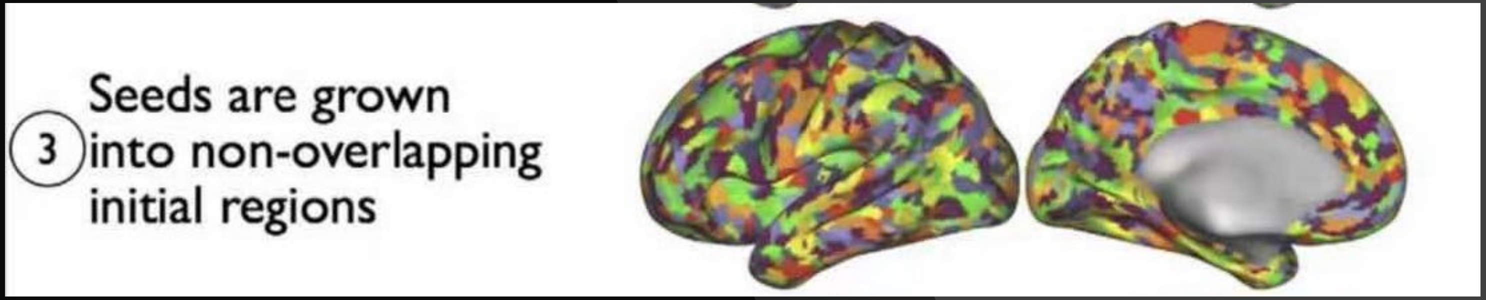
Steps 1 and 2:



- In this paper, stability map is calculated as the root mean square error between all time series in an ROI and the ROI's mean time series.
- We then apply a surface-based Gaussian kernel smoother.
- Seed locations were then identified as the local minima of the stability map
- The seed locations are estimations of cortical locations that lie inside functionally homogeneous brain regions. Their BOLD time series are representative of local BOLD activity


2
Seeds are selected that are local optima of the stability map

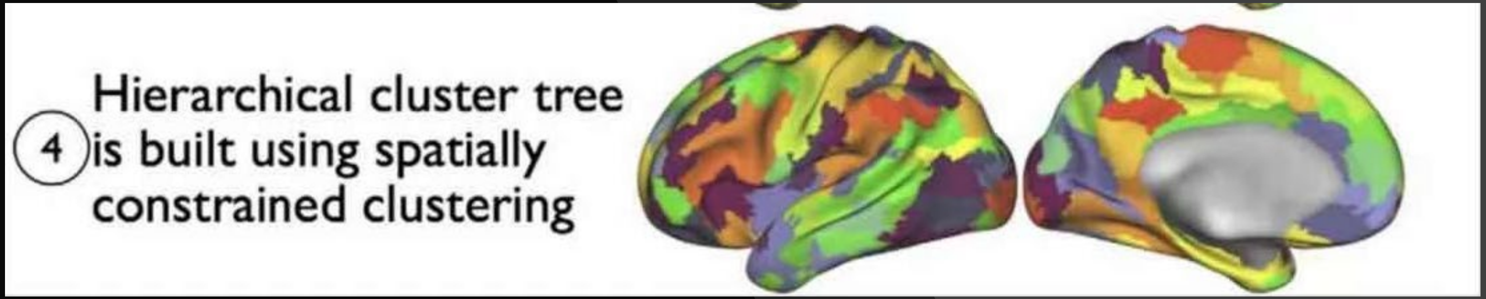
Steps 3:



3 Seeds are grown into non-overlapping initial regions

- Each seed is grown into a initial cluster by an iterative process
- Neighboring vertices(voxels) are attached to a cluster if the correlation between their time-courses and the region’s time-course exceeds 0.9 times the maximal correlation between all region time-courses and associated neighborhood vertex time-courses.
- If a vertex neighbors more than one region, then it is assigned to the region to which it is most correlated.

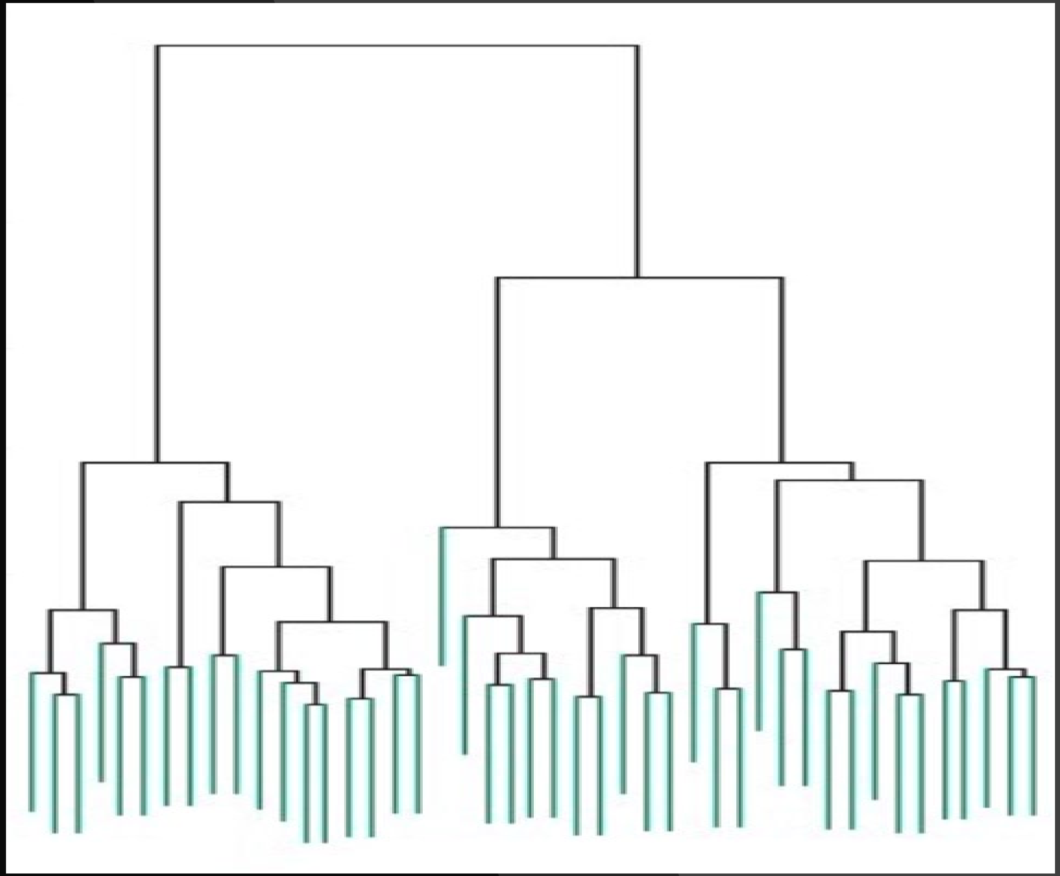
Steps 4:



4 Hierarchical cluster tree is built using spatially constrained clustering

- Now that we have a single parcel for each seed, a one-to-one relationship.
- Now we use hierarchical clustering introduced previously to further cluster the given parcels.
- We calculated initial cluster similarity using the correlation between the mean time courses extracted from a 3mm radius ROI centered on the region’s seed vertex.

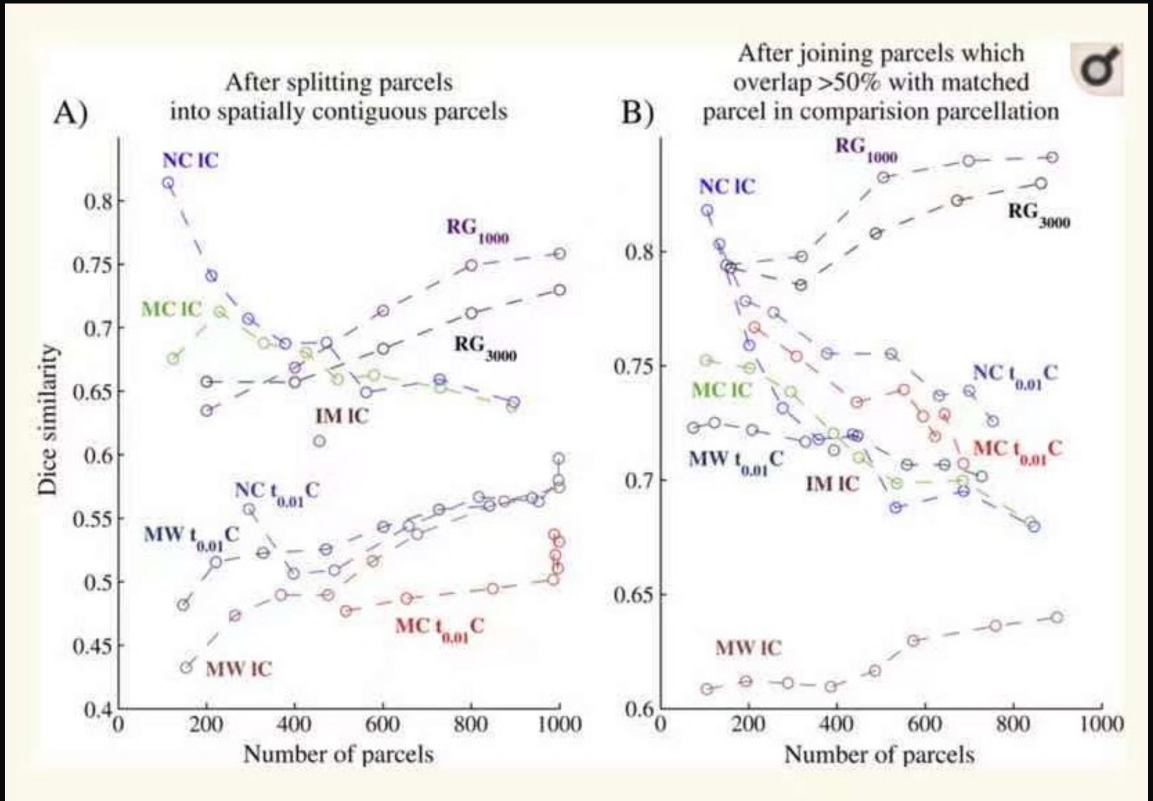
Note: distance measure here is not Euclidean distance

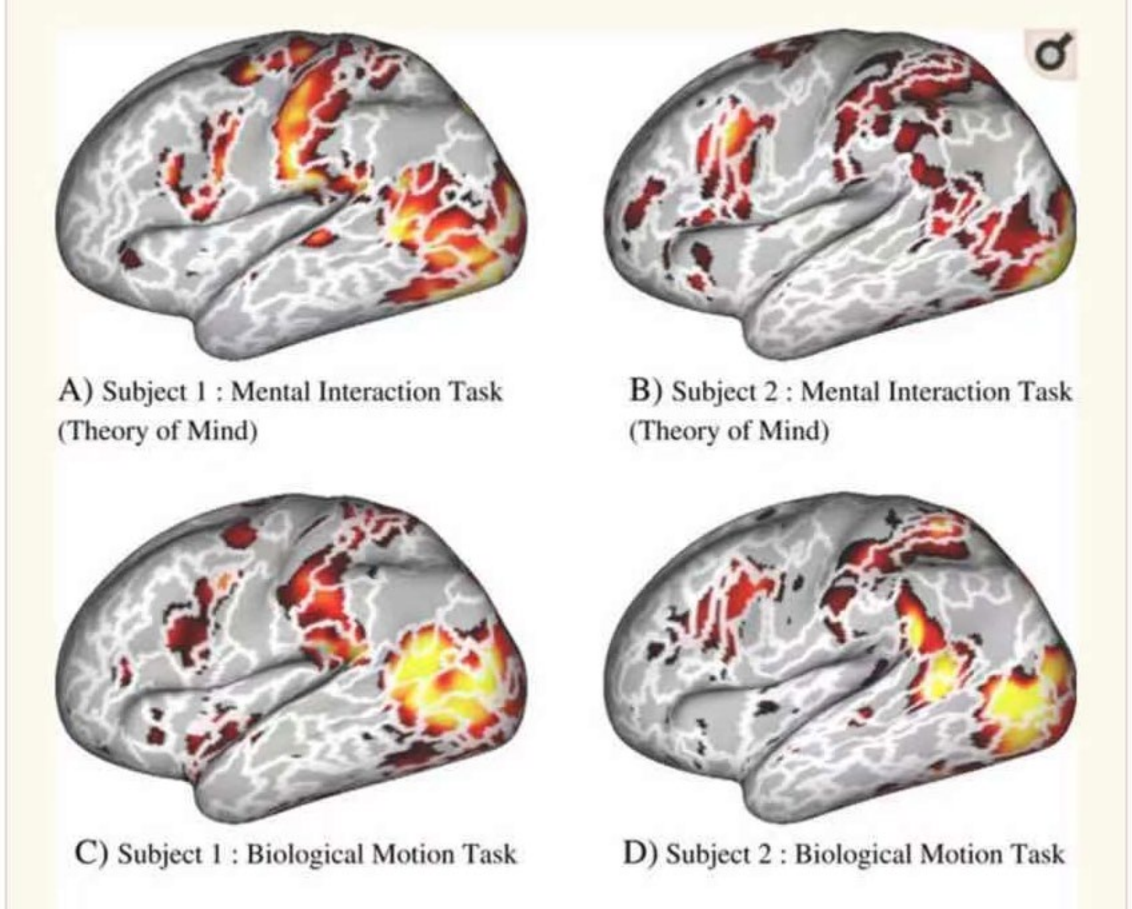


James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.

Performance

- High replicability
- Parcellation borders align with task activations





A) Subject 1 : Mental Interaction Task (Theory of Mind)

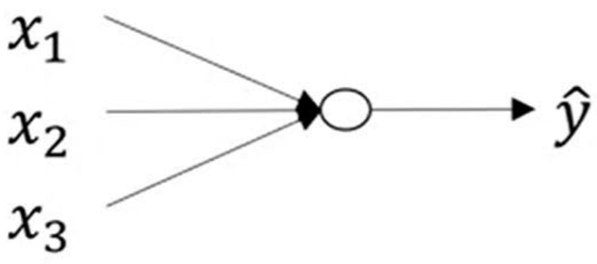
B) Subject 2 : Mental Interaction Task (Theory of Mind)

C) Subject 1 : Biological Motion Task

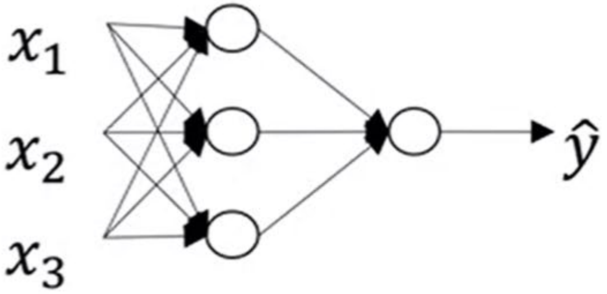
D) Subject 2 : Biological Motion Task

Artificial Neural Network

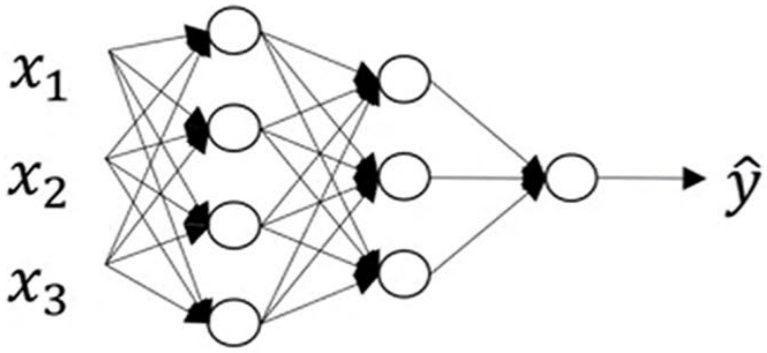
From Shallow to Deep Neural Network



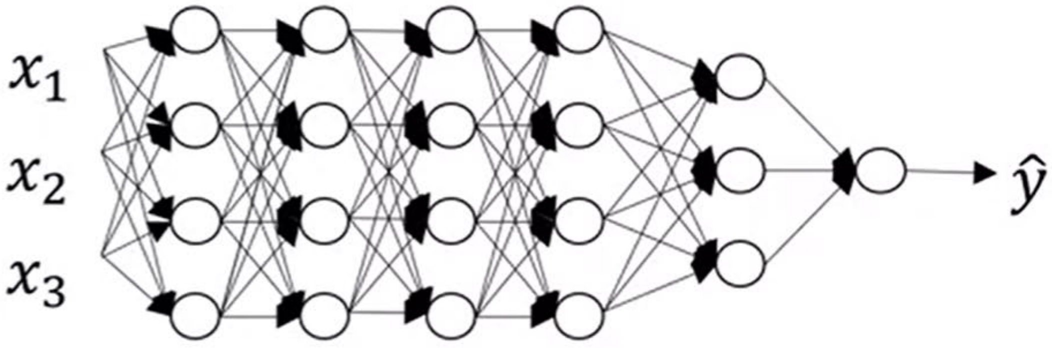
logistic regression



1 hidden layer

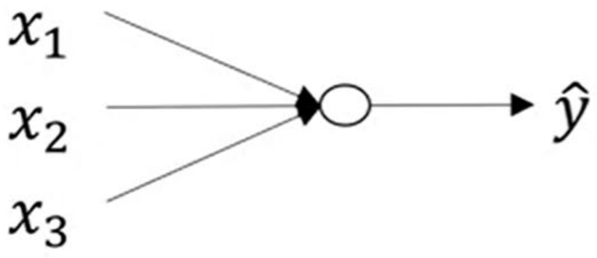


2 hidden layers

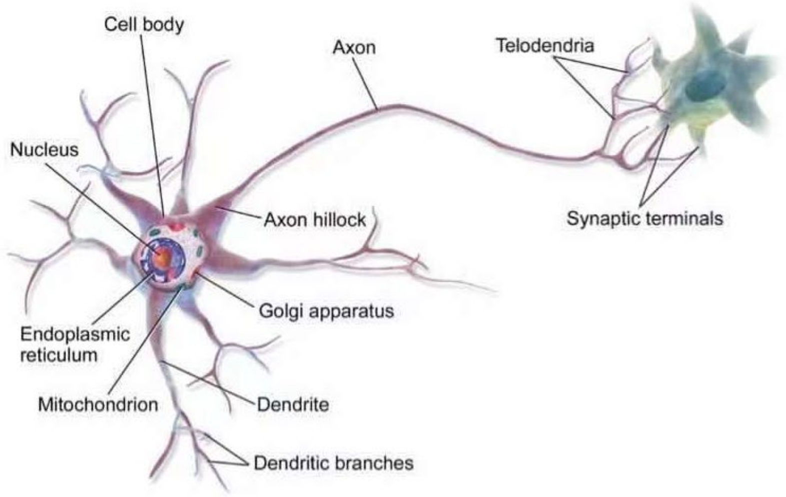


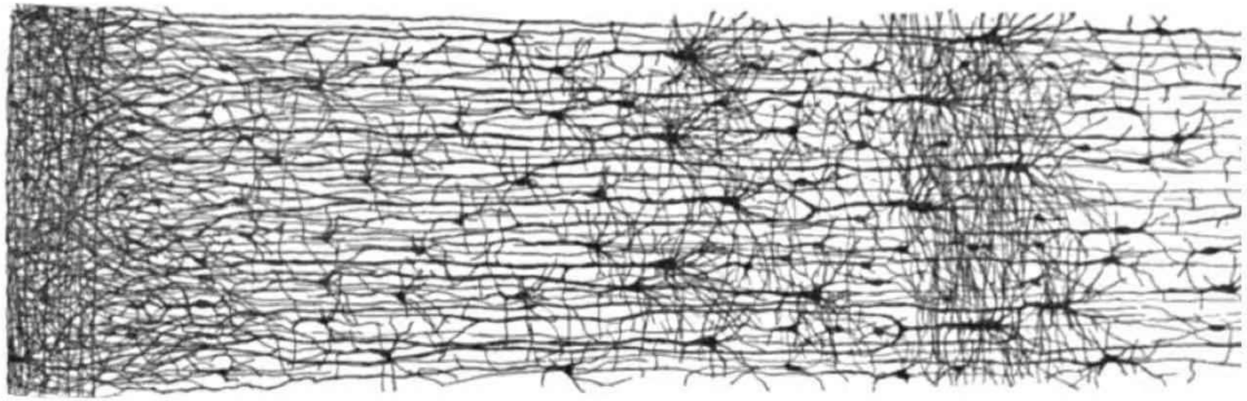
5 hidden layers

The Logistic Regression



logistic regression





Given x , want $\hat{y} = P(y = 1 \mid x)$,

$$\hat{y} = \sigma(w^T x + b)$$

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

Note:

If z is large, $\sigma(z) \approx \frac{1}{1+0}$.

If z is large negative, $\sigma(z) \approx \frac{1}{1+\infty} = 0$

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

The Logistic Regression

$\hat{y}^{(i)} = \sigma(w^T x^{(i)} + b)$, where $\sigma\left(z^{(i)}\right) = \frac{1}{1+e^{-z^{(i)}}}$

Given our dataset {($x^{(1)}, y^{(1)}$), ($x^{(2)}, y^{(2)}$)…,($x^{(m)}, y^{(m)}$)}, to estimate the parameters w and b such that $\hat{y}^{(i)} \approx y^{(i)}$,

Loss function:

$$L(\hat{y}, y) = -(y * \ln(\hat{y}) + (1 - y) * \ln(1 - \hat{y}))$$

Intuitive justification:
If $y = 1$: $L(\hat{y}, y) = -\ln(\hat{y})$ we want $-\ln(\hat{y})$ large, equivalently, want $\hat{y}$ large(as close to 1 as possible)
If $y = 0$: $L(\hat{y}, y) = -\ln(1 - \hat{y})$ we want $-\ln(1 - \hat{y})$ large, equivalently, want $1 - \hat{y}$ large, equivalently, $\hat{y}$ small (as close to 0 as possible)

For each observation in the dataset, we can calculate a loss: $L\left(\hat{y}^{(i)}, y^{(i)}\right)$

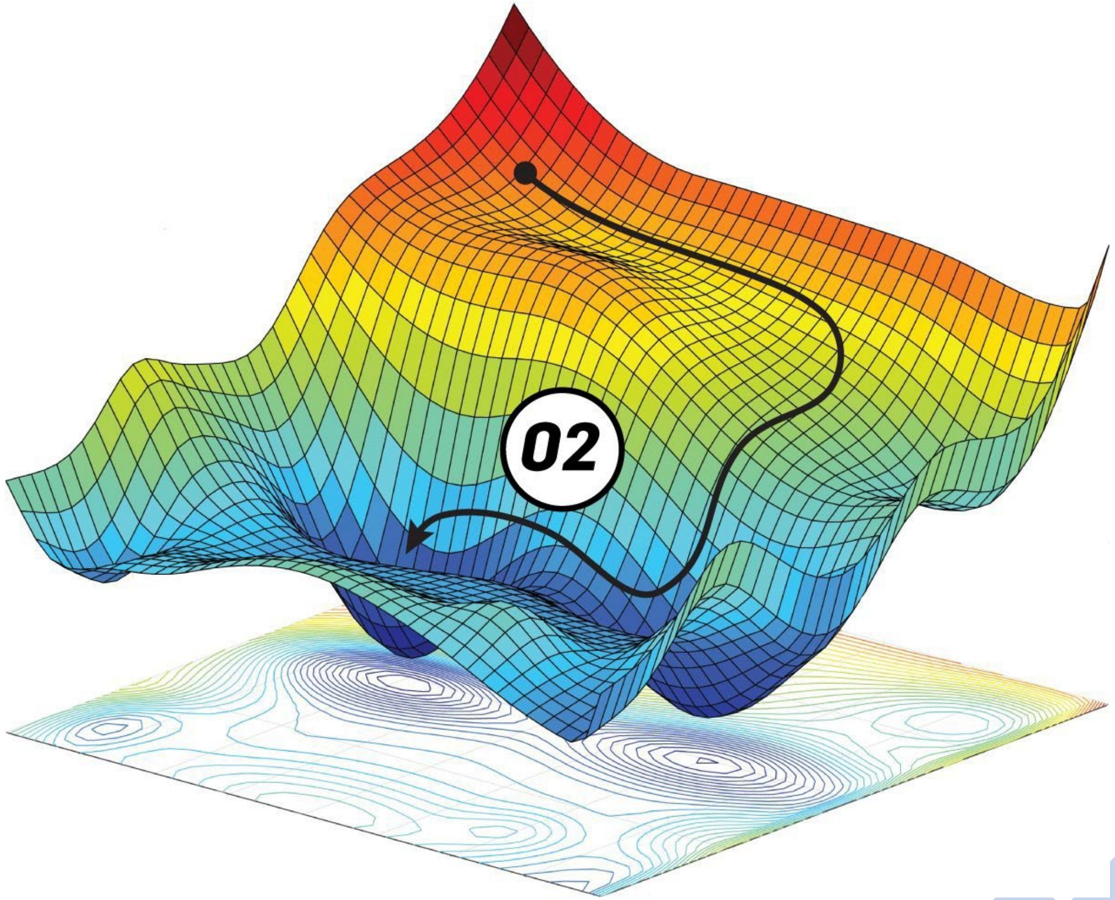
Sum across the entire dataset, we have the **cost function:**

$$\boldsymbol{J(w, b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} * ln\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) * ln\left(1 - \hat{y}^{(i)}\right)\right)}$$

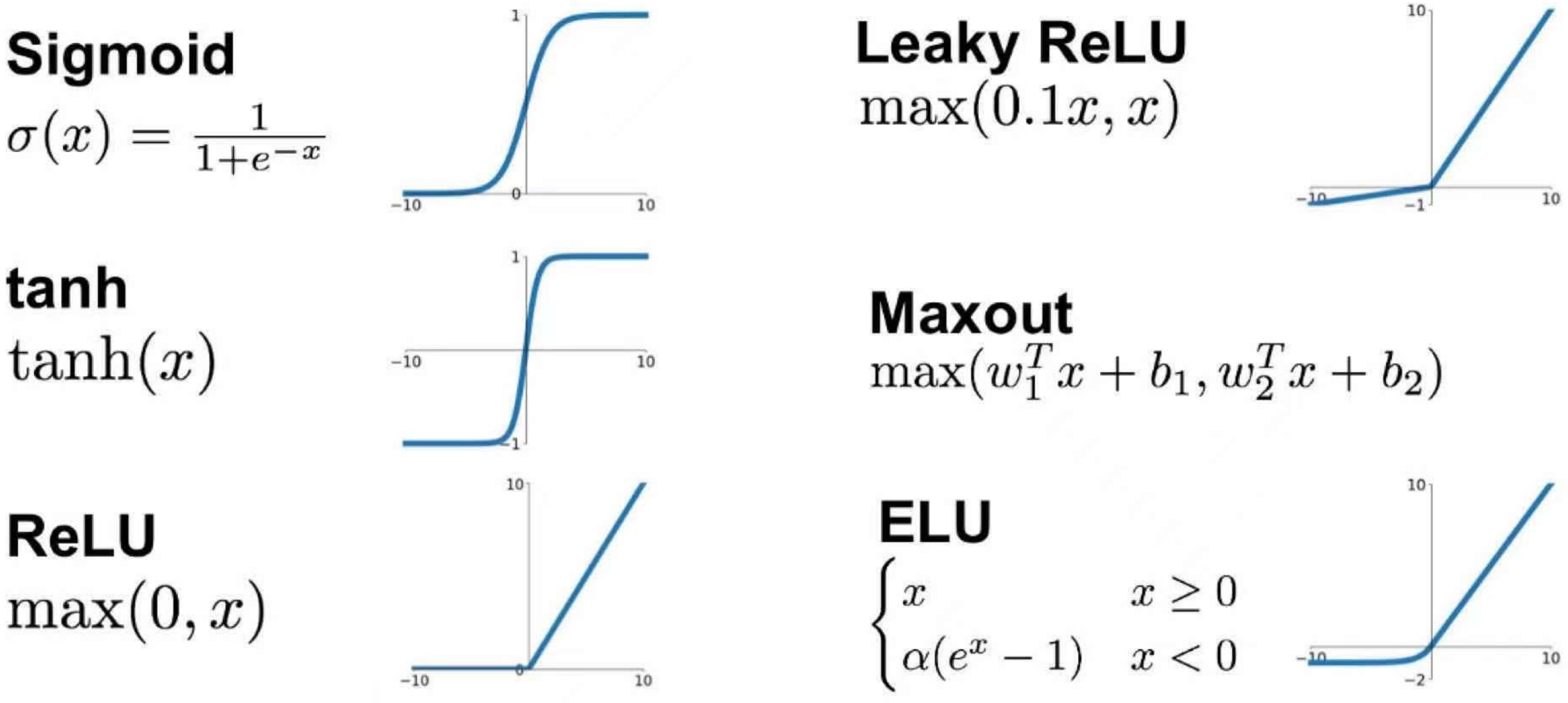
Gradient Descent, an optimizing algorithm

Minimize the cost function:

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} * ln\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) * ln\left(1 - \hat{y}^{(i)}\right)\right)$$



The Logistic Regression



Sigmoid

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

tanh

$\tanh(x)$

ReLU

$\max(0, x)$

Leaky ReLU

$\max(0.1x, x)$

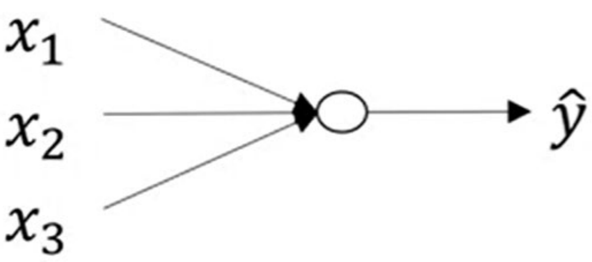
Maxout

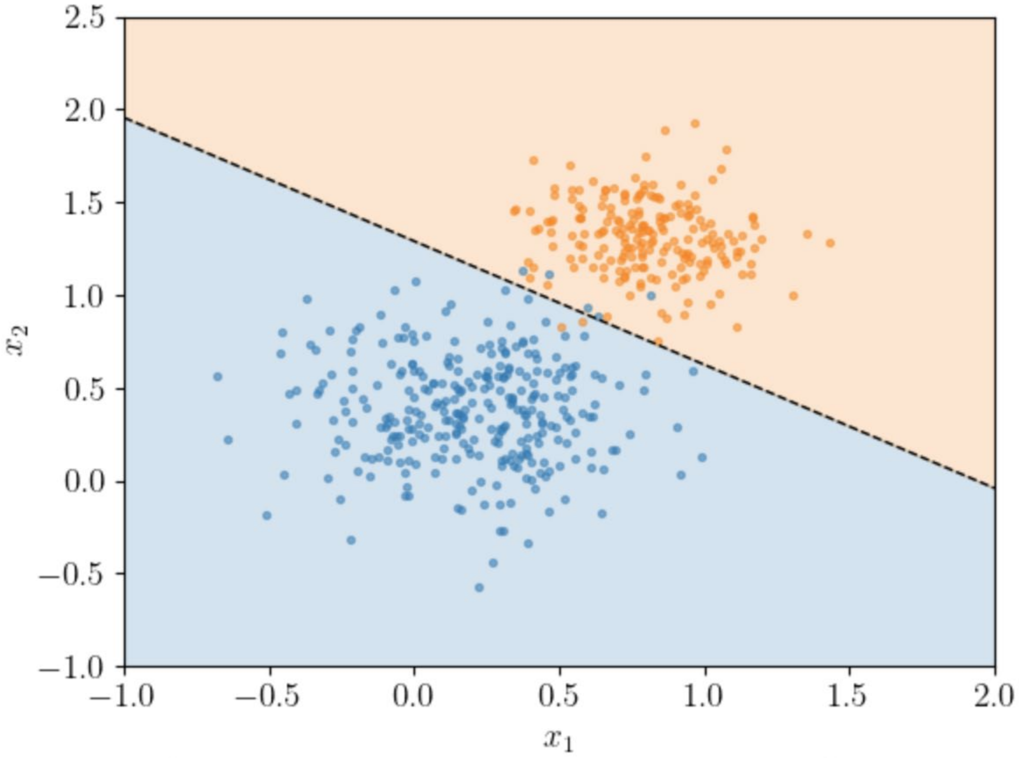
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

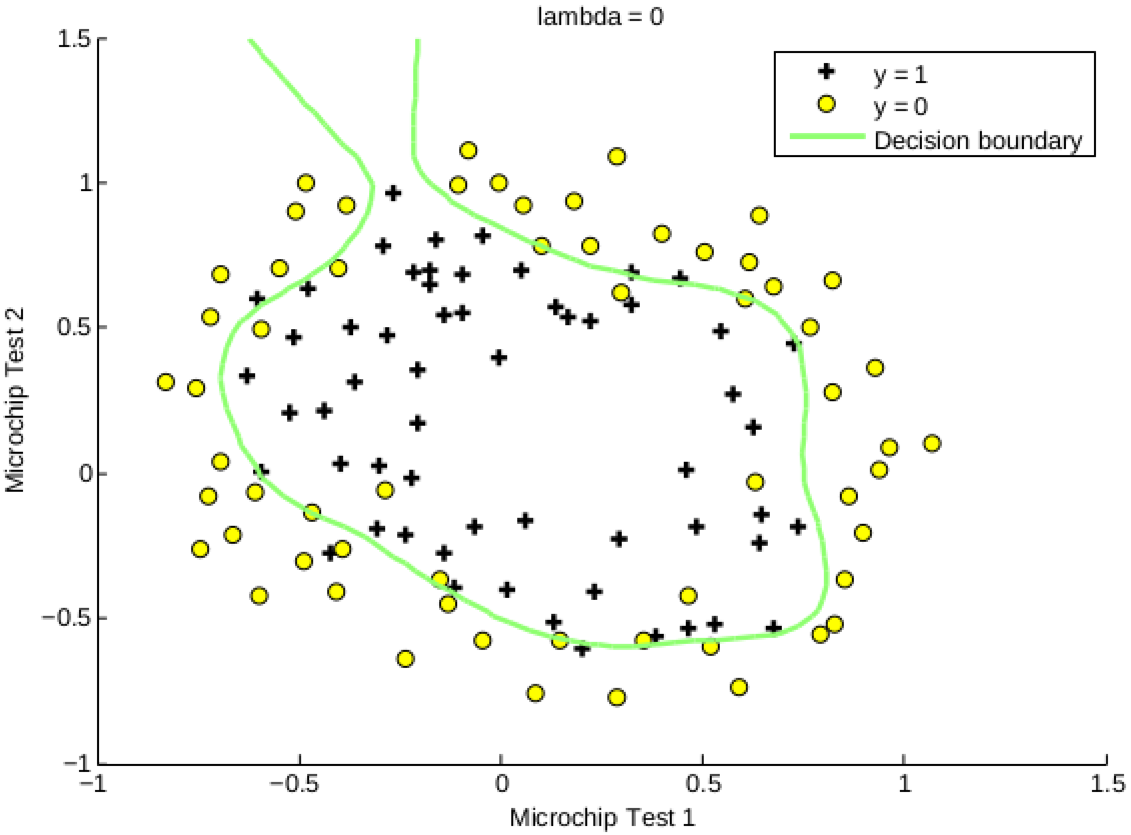
The Logistic Regression decision boundary

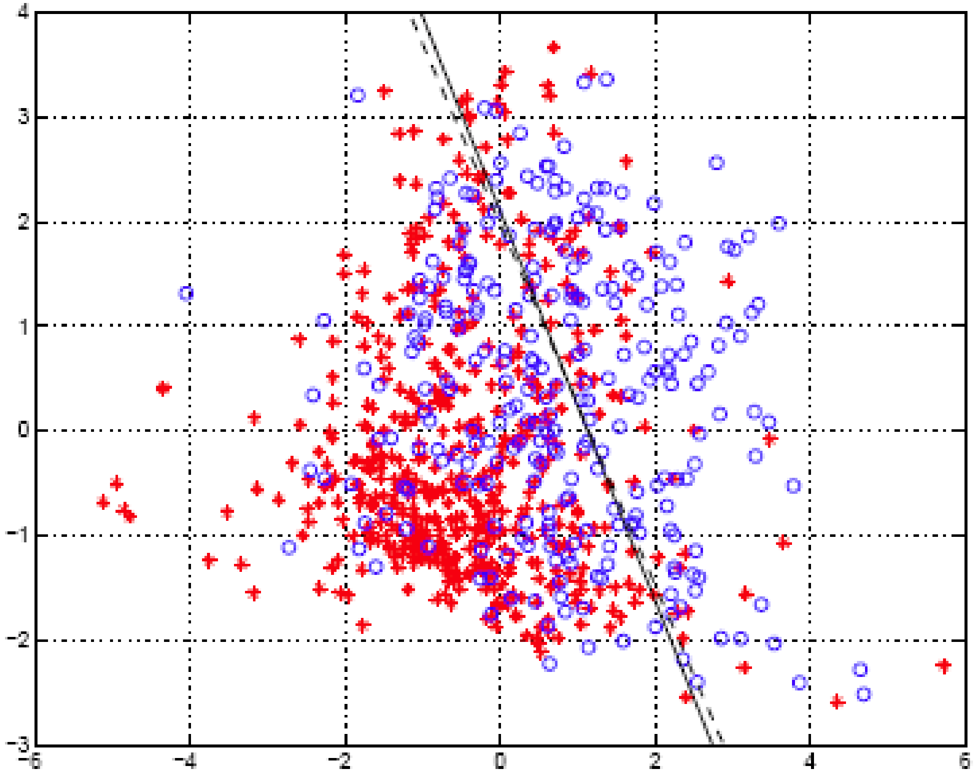




Linearly separable data

Why do we need hidden layers?

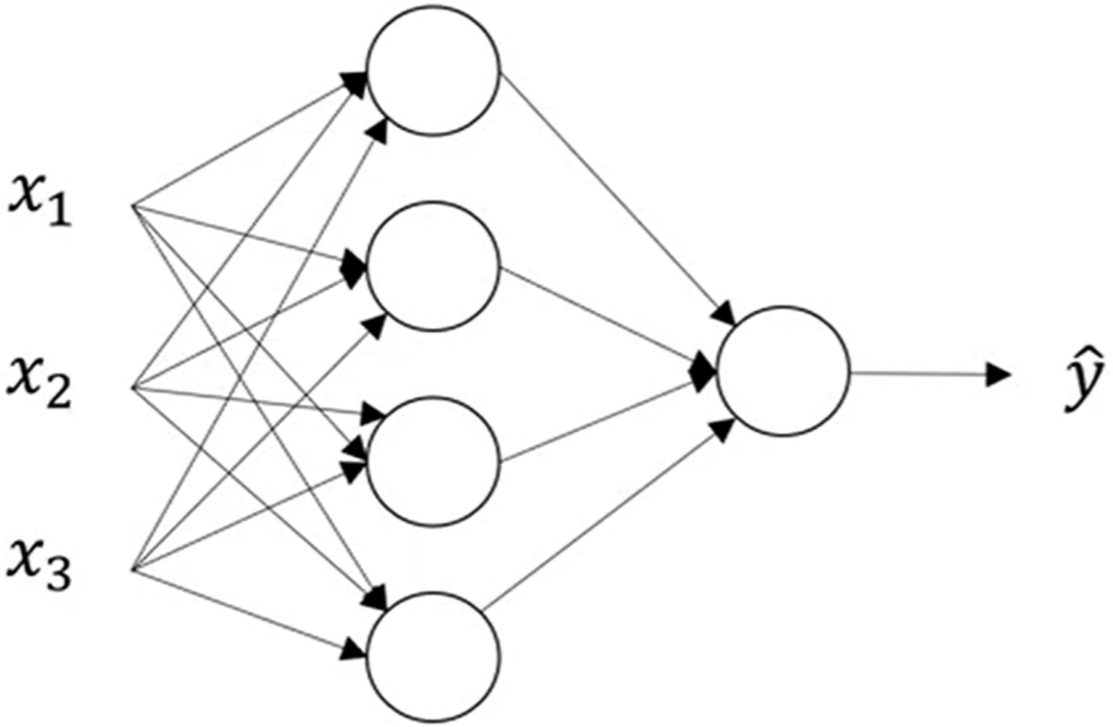




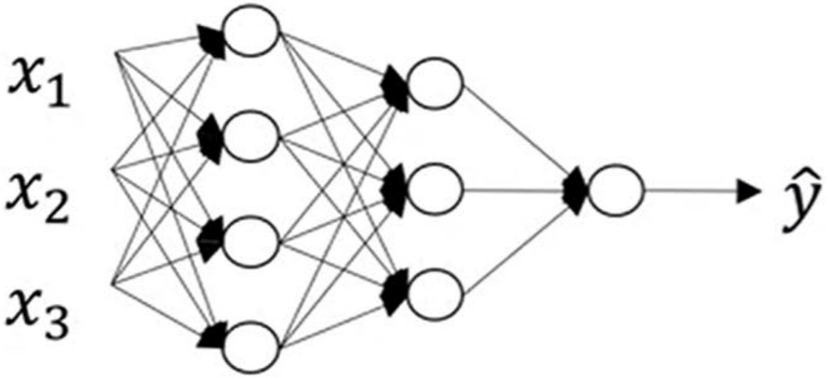
Pictures from Stack overflow

Shallow Neural Network

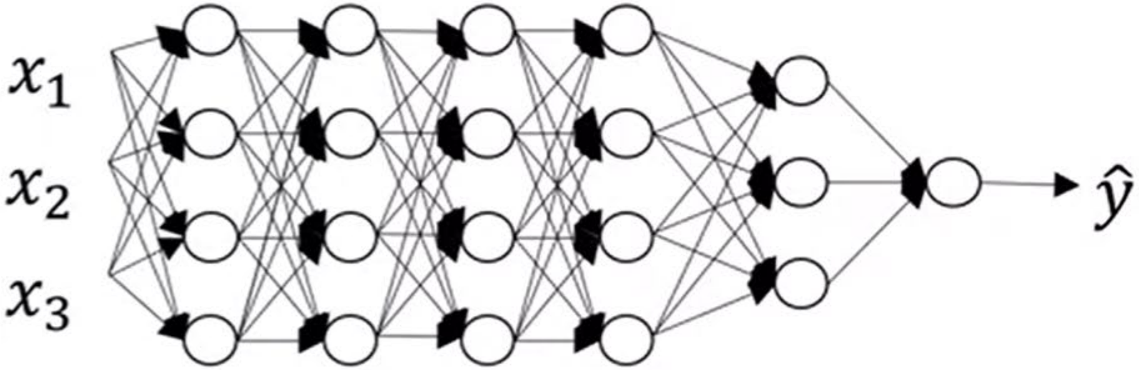
How many parameters?



Deep Neural Network



2 hidden layers

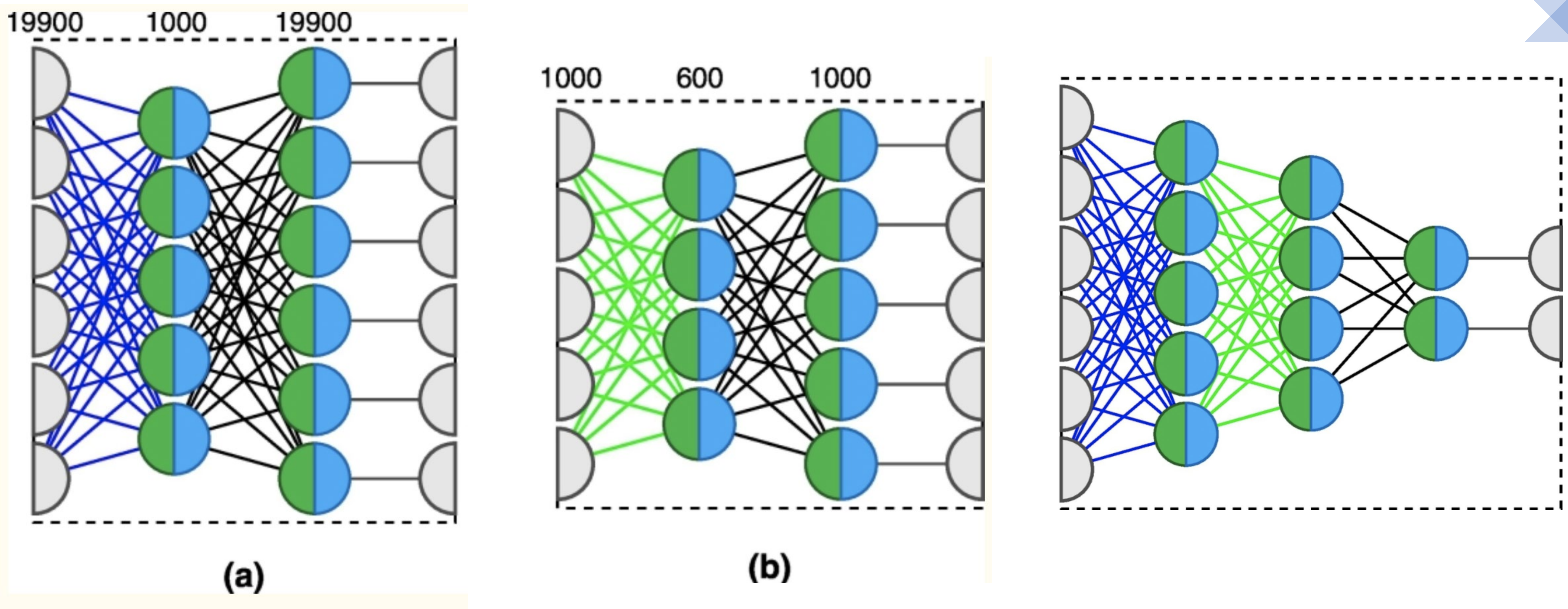


5 hidden layers

Feature selection

- Functional connectivity was used to classify subjects as ASD or TC
- Functional connectivity provides an index of the level of co-activation of brain regions based on the time series of rs-fMRI brain imaging data
- Number of features: $S = \frac{(N-1)N}{2}$
- N is the number of correlated brain regions
- The CC200 ROI atlas was used, and thus we have 19900 features

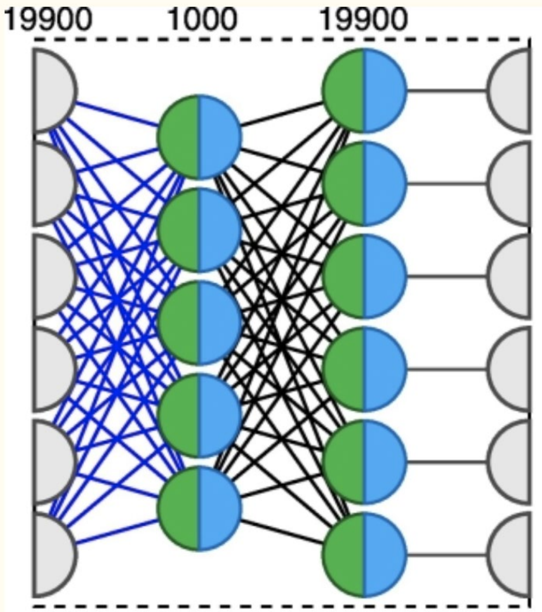
Multilayer Perceptron; Transfer learning is applied..



Two stacked denoising autoencoders for the unsupervised pretraining stage..

Heinsfeld, Anibal Sólon, et al. "Identification of autism spectrum disorder using deep learning and the ABIDE dataset." *NeuroImage: Clinical* 17 (2018): 16-23.

Denoising autoencoder architecture



(a)

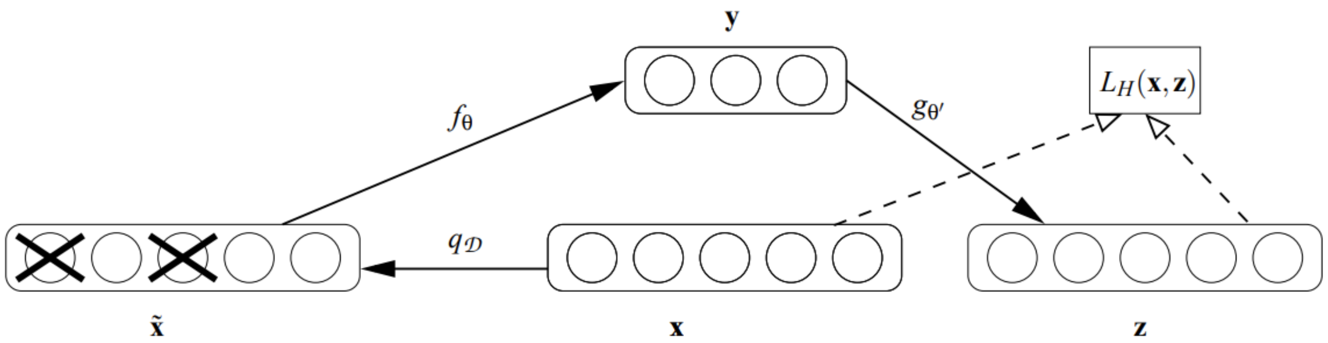
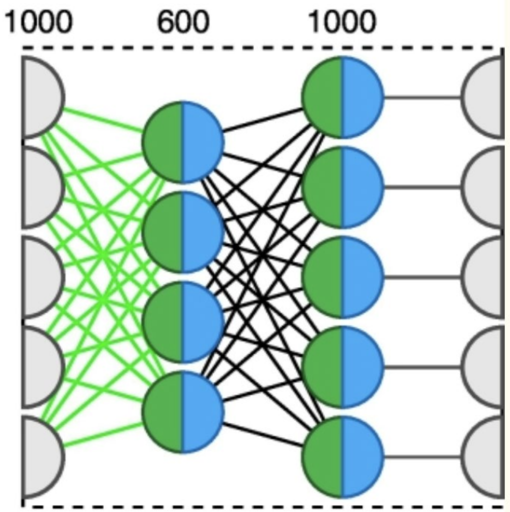


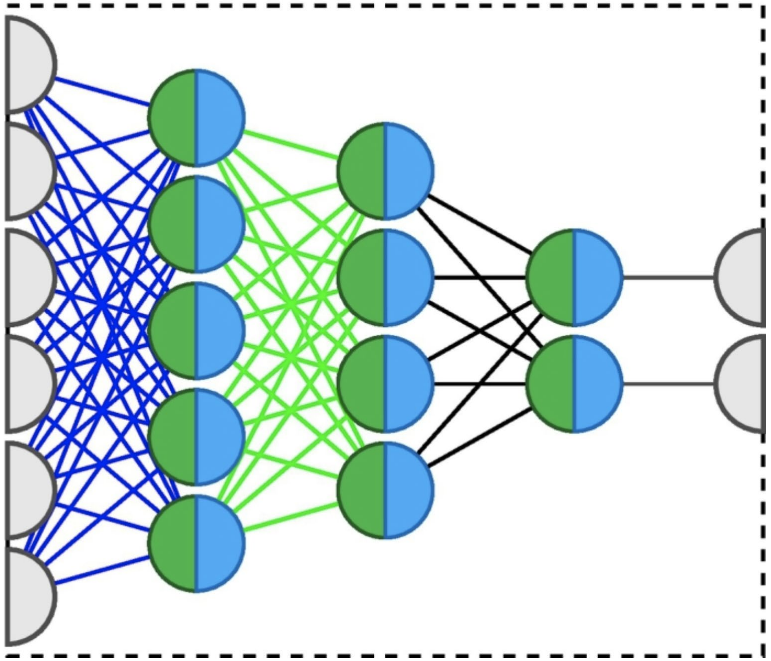
Figure 1: The denoising autoencoder architecture. An example $\mathbf{x}$ is stochastically corrupted (via $q_\mathcal{D}$) to $\tilde{\mathbf{x}}$. The autoencoder then maps it to $\mathbf{y}$ (via encoder f_θ) and attempts to reconstruct $\mathbf{x}$ via decoder $g_{\theta'}$, producing reconstruction $\mathbf{z}$. Reconstruction error is measured by loss $L_H(\mathbf{x}, \mathbf{z})$.



(b)

Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research* 11.12 (2010).

Two stacked denoising autoencoders for the unsupervised pretraining stage..



- The output layer contains two output units: each unit represents the probability of an input to be from an ASD or a TC subject. This type of output is called one-hot: during fine-tuning only one of the outputs is expected to have an activation value of 1 (and the others, 0); the output is obtained applying a SoftMax function.
- SoftMax functions normalize the output distribution, so outputs denote complementary probabilities of being one class (i.e., a sum one of probabilities of being ASD or TC; for example, an output of probability of being ASD: 80%, and of being TC: 20%).

Heinsfeld, Anibal Sólon, et al. "Identification of autism spectrum disorder using deep learning and the ABIDE dataset." *NeuroImage: Clinical* 17 (2018): 16-23.

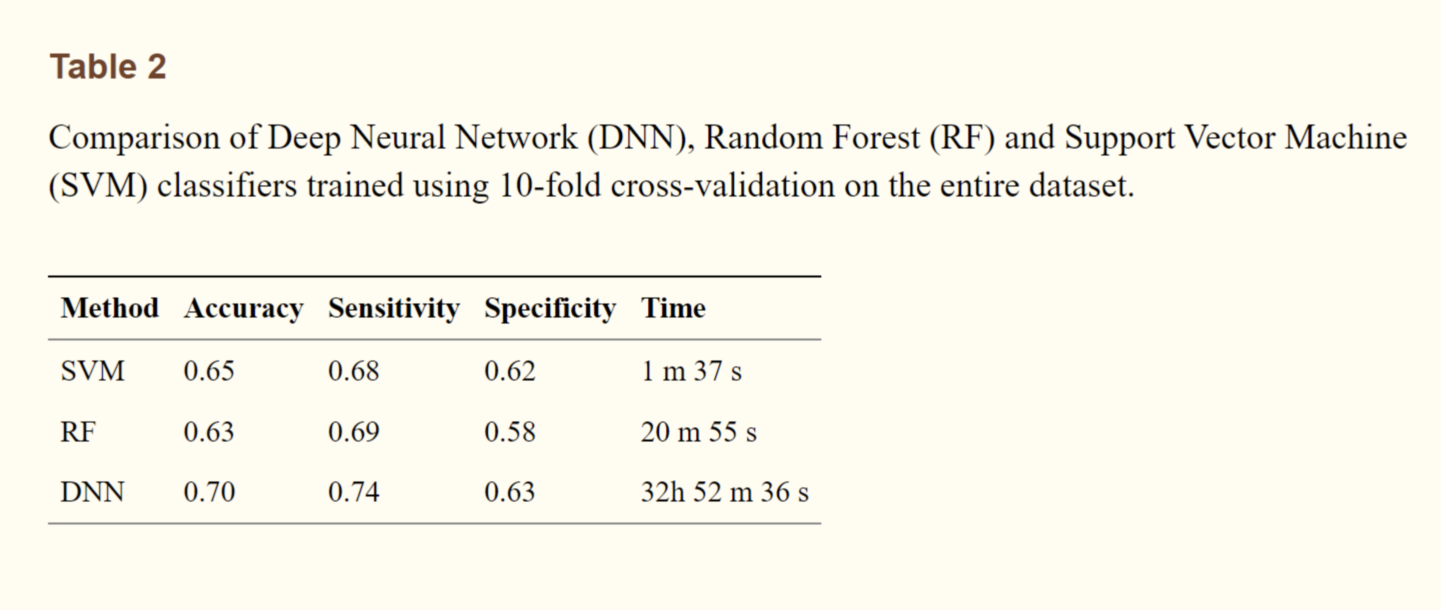


Table 2

Comparison of Deep Neural Network (DNN), Random Forest (RF) and Support Vector Machine (SVM) classifiers trained using 10-fold cross-validation on the entire dataset.

Method	Accuracy	Sensitivity	Specificity	Time
SVM	0.65	0.68	0.62	1 m 37 s
RF	0.63	0.69	0.58	20 m 55 s
DNN	0.70	0.74	0.63	32h 52 m 36 s

Note:

Sensitivity: how many autistic people are correctly identified as having the condition

Specificity: how many typical controls are identified as not having the condition

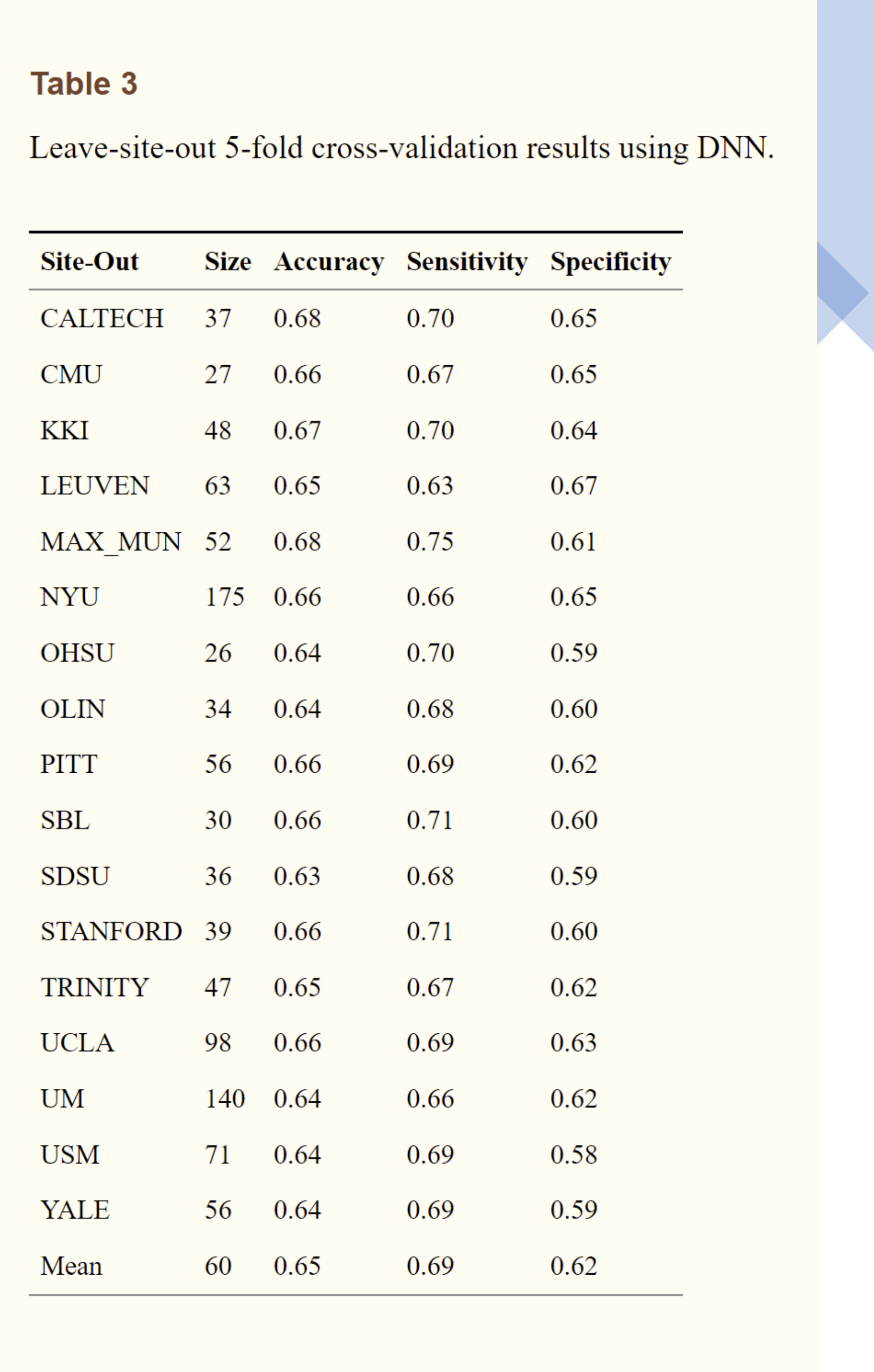


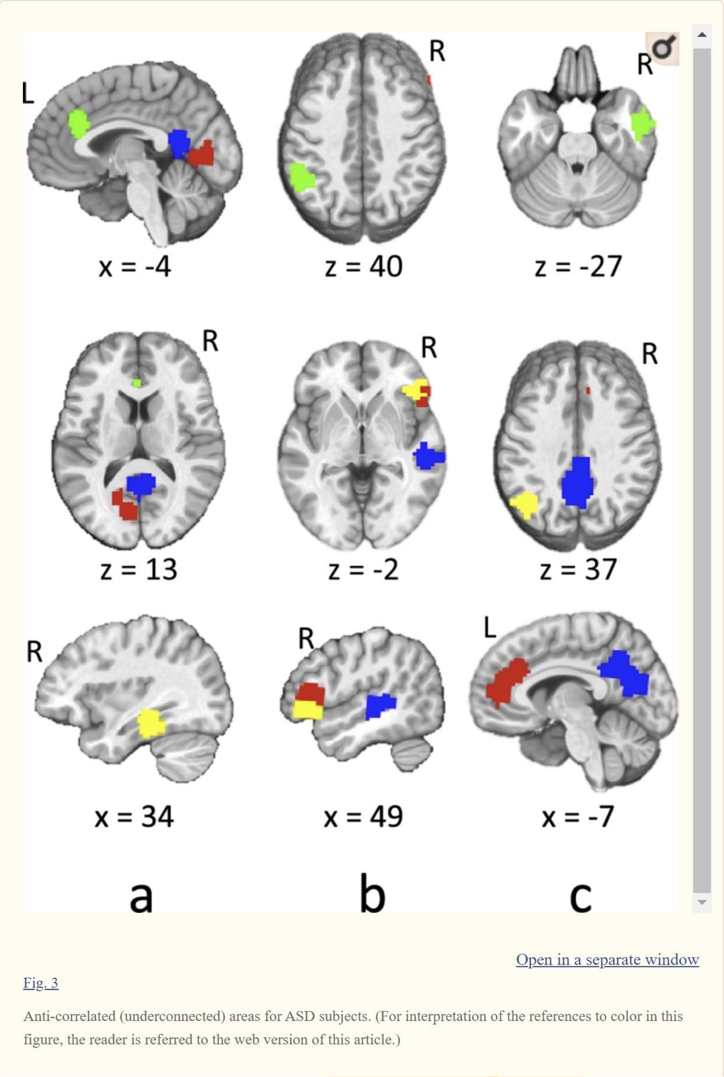
Table 3

Leave-site-out 5-fold cross-validation results using DNN.

Site-Out	Size	Accuracy	Sensitivity	Specificity
CALTECH	37	0.68	0.70	0.65
CMU	27	0.66	0.67	0.65
KKI	48	0.67	0.70	0.64
LEUVEN	63	0.65	0.63	0.67
MAX_MUN	52	0.68	0.75	0.61
NYU	175	0.66	0.66	0.65
OHSU	26	0.64	0.70	0.59
OLIN	34	0.64	0.68	0.60
PITT	56	0.66	0.69	0.62
SBL	30	0.66	0.71	0.60
SDSU	36	0.63	0.68	0.59
STANFORD	39	0.66	0.71	0.60
TRINITY	47	0.65	0.67	0.62
UCLA	98	0.66	0.69	0.63
UM	140	0.64	0.66	0.62
USM	71	0.64	0.69	0.58
YALE	56	0.64	0.69	0.59
Mean	60	0.65	0.69	0.62

Heinsfeld, Anibal Sólon, et al. "Identification of autism spectrum disorder using deep learning and the ABIDE dataset." *NeuroImage: Clinical* 17 (2018): 16-23.

Neural patterns: connectivity in the autistic brain

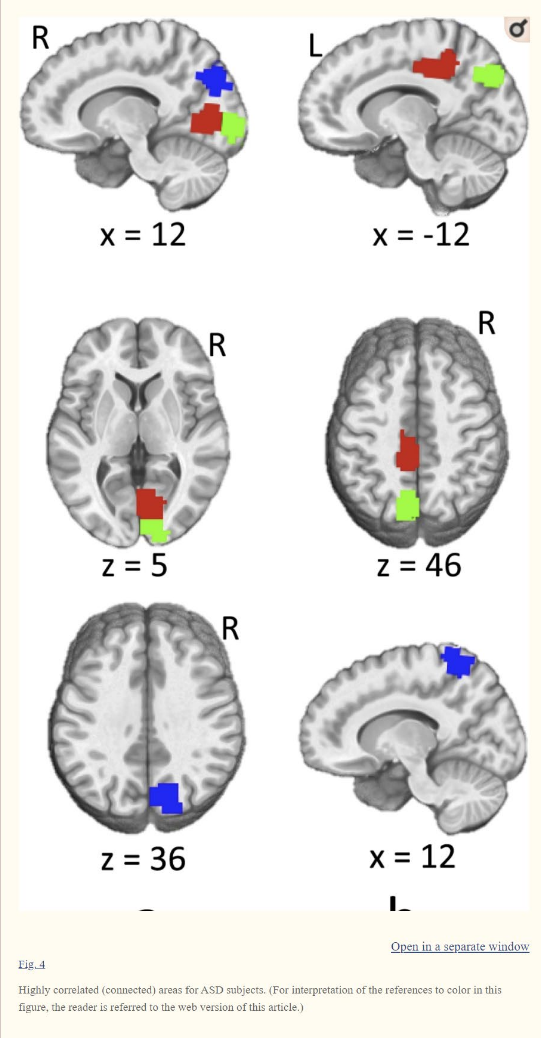


Open in a separate window

Fig. 3

Anti-correlated (underconnected) areas for ASD subjects. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Anti-correlated (underconnected) areas for ASD subjects.



Open in a separate window

Fig. 4

Highly correlated (connected) areas for ASD subjects. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Highly correlated (connected) areas for ASD subjects.

Heinsfeld, Anibal Sólon, et al. "Identification of autism spectrum disorder using deep learning and the ABIDE dataset." *NeuroImage: Clinical* 17 (2018): 16-23.

Neural patterns: connectivity in the autistic brain

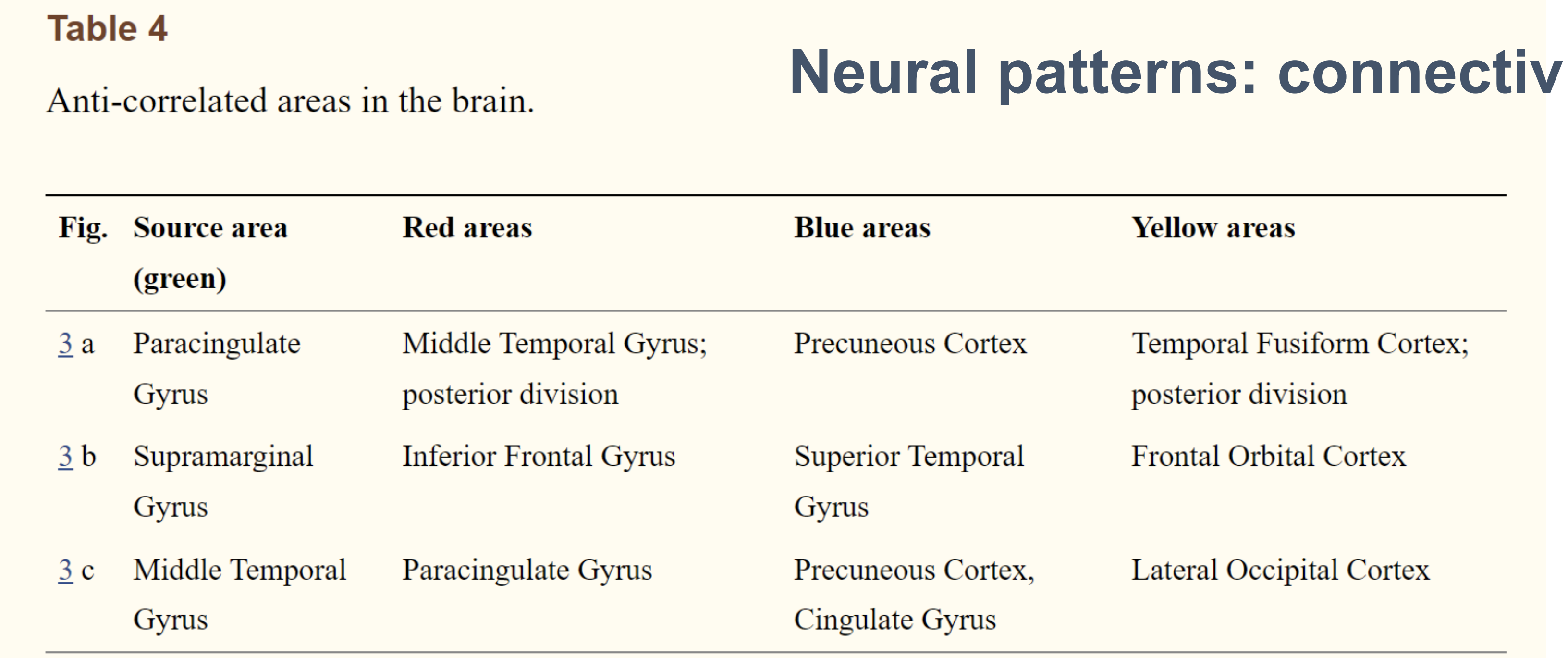


Table 4

Anti-correlated areas in the brain.

Fig.	Source area (green)	Red areas	Blue areas	Yellow areas
3 a	Paracingulate Gyrus	Middle Temporal Gyrus; posterior division	Precuneous Cortex	Temporal Fusiform Cortex; posterior division
3 b	Supramarginal Gyrus	Inferior Frontal Gyrus	Superior Temporal Gyrus	Frontal Orbital Cortex
3 c	Middle Temporal Gyrus	Paracingulate Gyrus	Precuneous Cortex, Cingulate Gyrus	Lateral Occipital Cortex

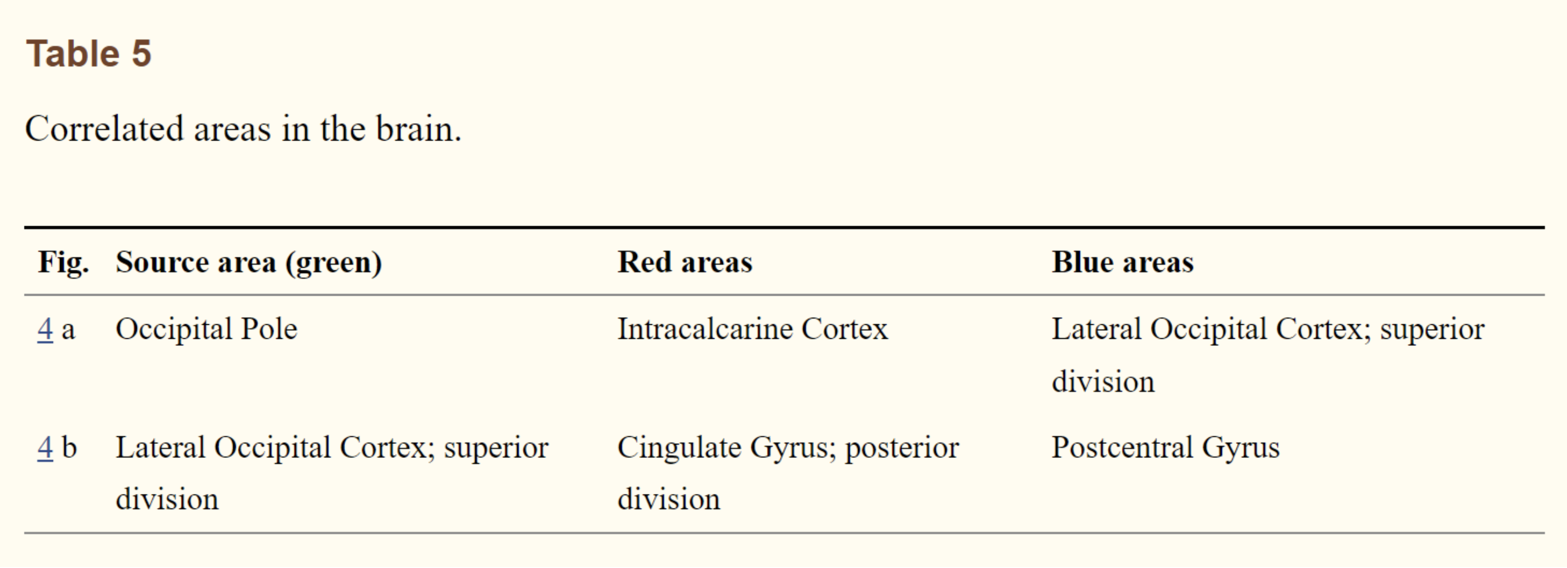


Table 5

Correlated areas in the brain.

Fig.	Source area (green)	Red areas	Blue areas
4 a	Occipital Pole	Intracalcarine Cortex	Lateral Occipital Cortex; superior division
4 b	Lateral Occipital Cortex; superior division	Cingulate Gyrus; posterior division	Postcentral Gyrus

Heinsfeld, Anibal Sólon, et al. "Identification of autism spectrum disorder using deep learning and the ABIDE dataset." *NeuroImage: Clinical* 17 (2018): 16-23.

Concluding Remarks

- We have seen that machine learning offers an extremely versatile set of tools that can handle both fundamental problems in rs-fMRI, such as generating brain parcellations on which other researches rely upon, and specific problems such as diagnosing certain disorders.
- Machine learning techniques are very flexible, and can be relatively easily implemented with a moderate level of programming skills
- As the quantity of rs-fMRI is becoming increasingly abundant, the opportunities associated with machine learning are limitless.

Reference

- Blumensath T, Jbabdi S, Glasser M F, Van Essen D C, Ugurbil K, Behrens T E. Spatially constrained hierarchical parcellation of the brain with resting-state fMRI. Neuroimage 2013;76:313–24.
- Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.
- Heinsfeld, Anibal Sólon, et al. "Identification of autism spectrum disorder using deep learning and the ABIDE dataset." *NeuroImage: Clinical* 17 (2018): 16-23.
- James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.
- Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research* 11.12 (2010).