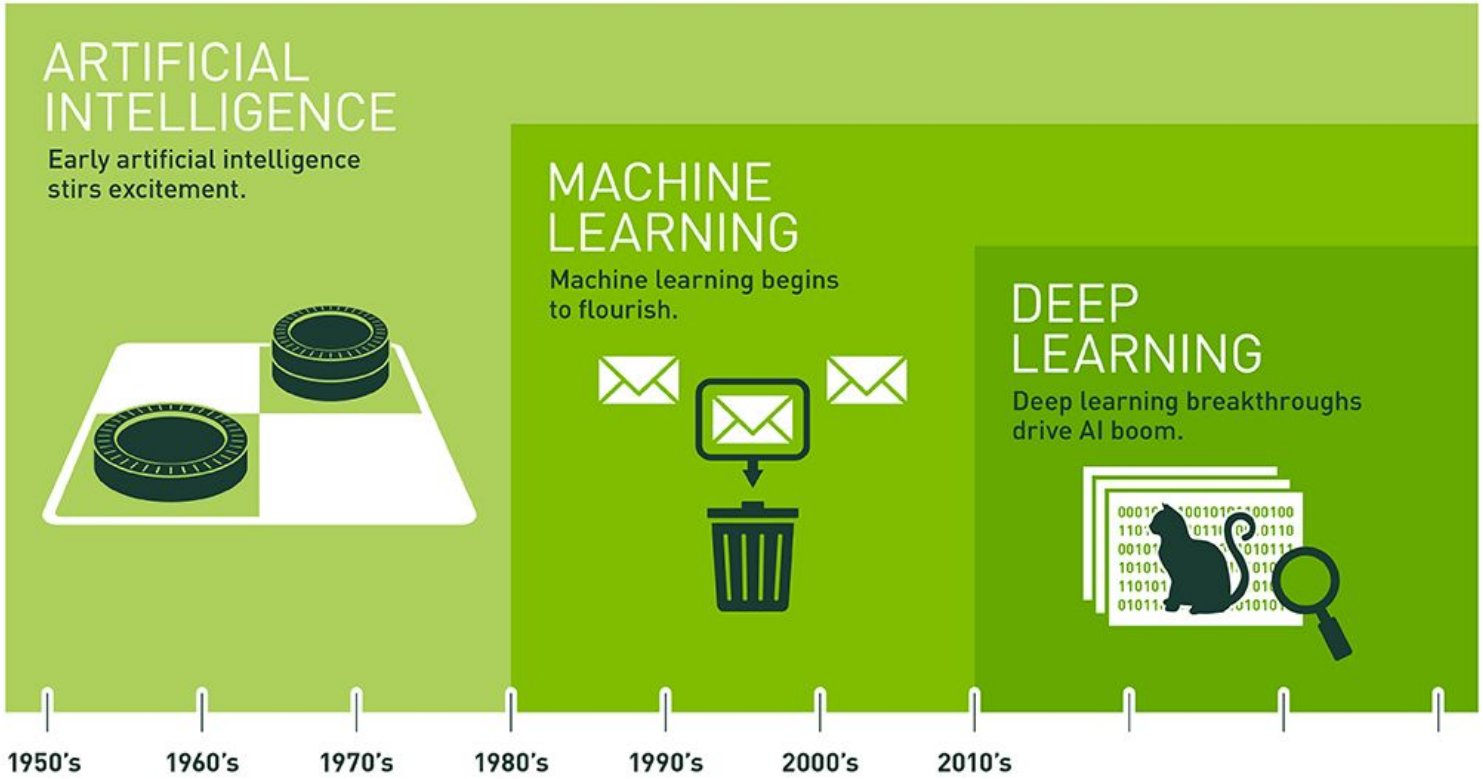
Machine learning primer

Yu (Andy) Huang, Ph.D.
Senior Scientist, Soterix Medical Inc.

November 1, 2022

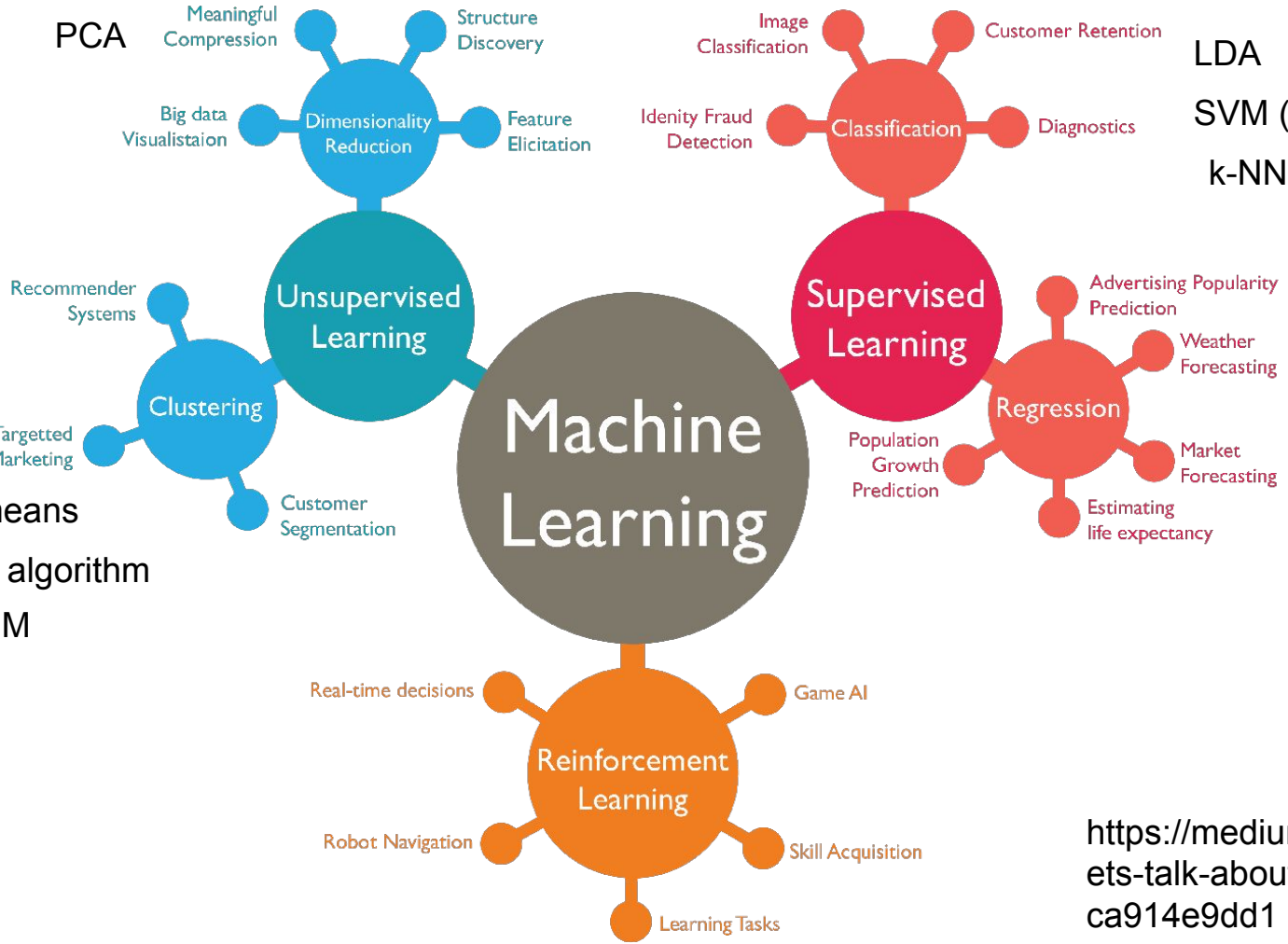
Disclosure

My background is not machine learning, or computer science, or statistics.



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

https://medium.com/@lizziedotdev/lets-talk-about-machine-learning-ddca914e9dd1



PCA

LDA

SVM (kernel methods)

k-NN

Linear regression

Logistic regression

Deep neural nets

k-means

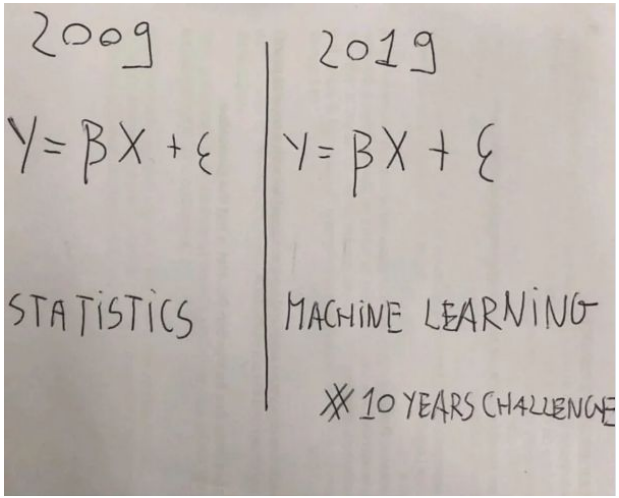
EM algorithm

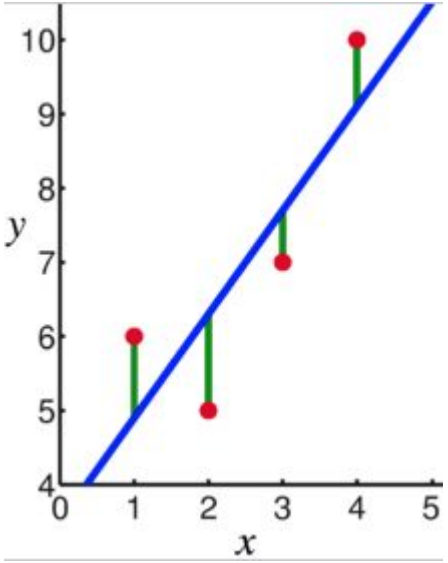
GMM

https://medium.com/@lizziedotdev/lets-talk-about-machine-learning-ddca914e9dd1

Supervised learning

Linear regression

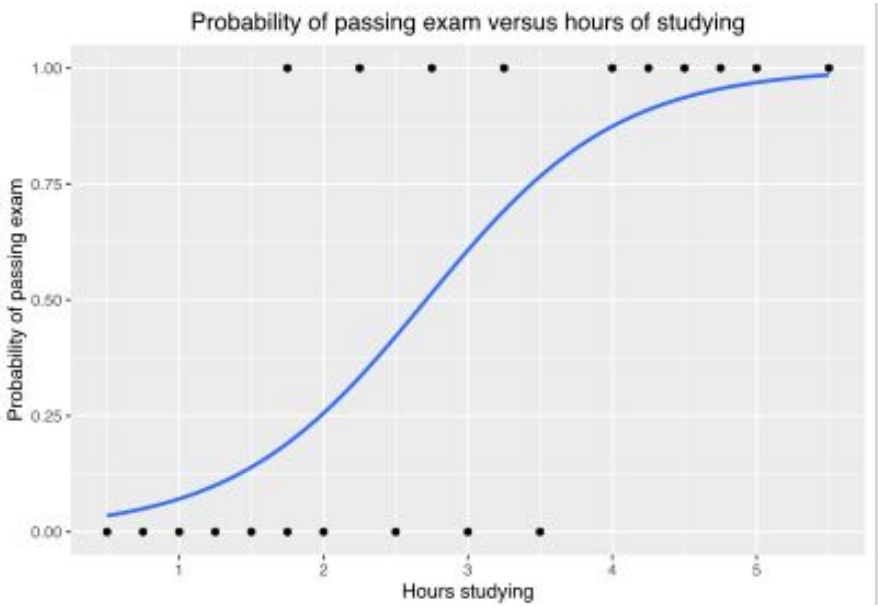




$y = b + ax$

Usually solved by least square

Logistic regression

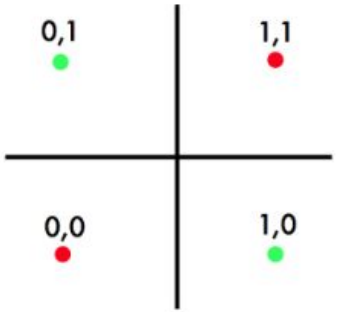


$y = 1/(1+\exp(-b-ax))$

Logit (log odds) is the linear term:
$\log(y/(1-y)) = b+ax$

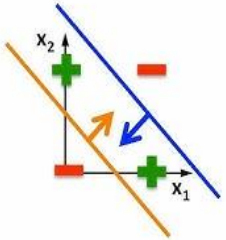
Also solved by least square
(iteratively reweighted least squares)

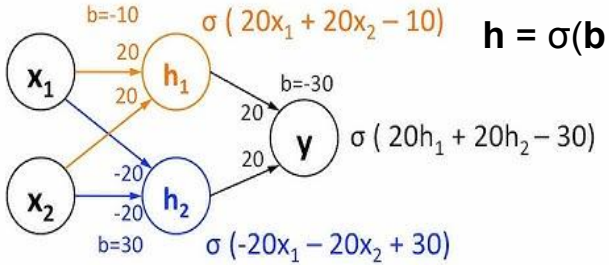
Neural networks



XOR

Linear classifiers cannot solve this



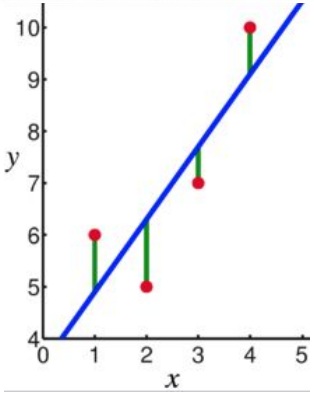


h = σ(b + Wx)

σ(20*0 + 20*0 – 10) ≈ 0
σ(20*1 + 20*1 – 10) ≈ 1
σ(20*0 + 20*1 – 10) ≈ 1
σ(20*1 + 20*0 – 10) ≈ 1

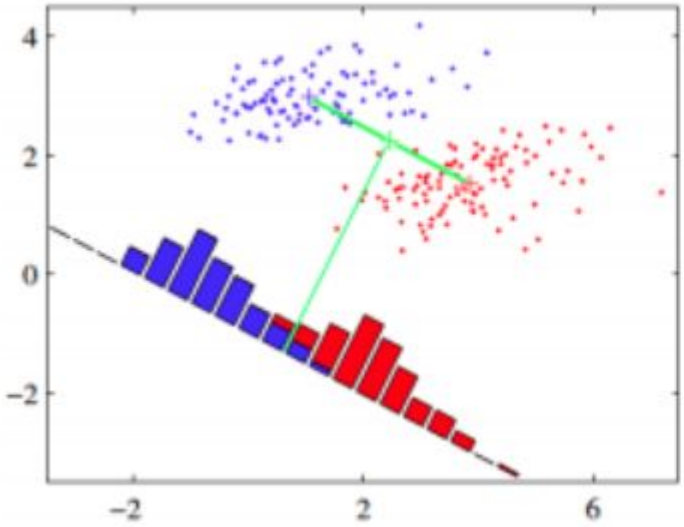
σ (-20*0 – 20*0 + 30) ≈ 1
σ (-20*1 – 20*1 + 30) ≈ 0
σ (-20*0 – 20*1 + 30) ≈ 1
σ (-20*1 – 20*0 + 30) ≈ 1

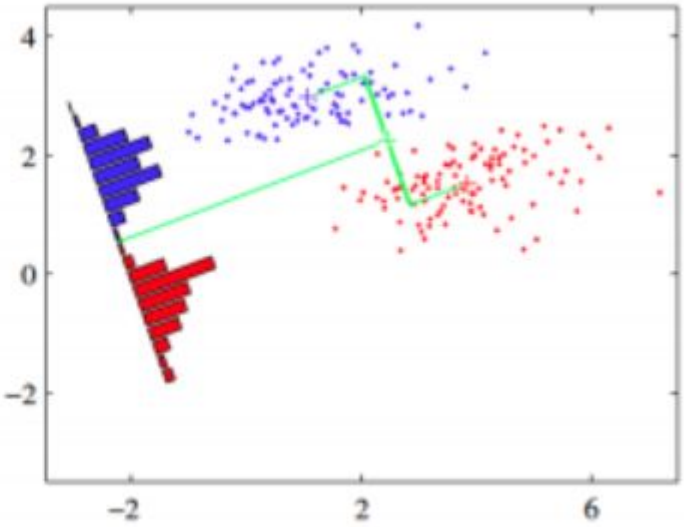
σ (20*0 + 20*1 – 30) ≈ 0
σ (20*1 + 20*0 – 30) ≈ 0
σ (20*1 + 20*1 – 30) ≈ 1
σ (20*1 + 20*1 – 30) ≈ 1



y = b + ax

Linear discriminant analysis (LDA, Fisher's LDA)

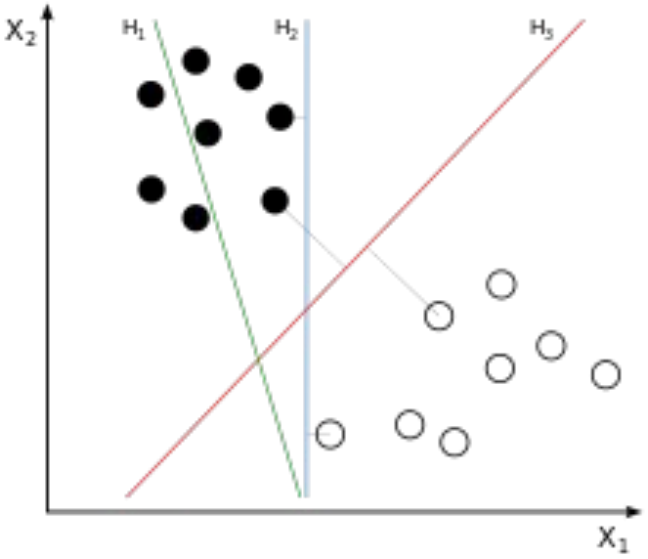


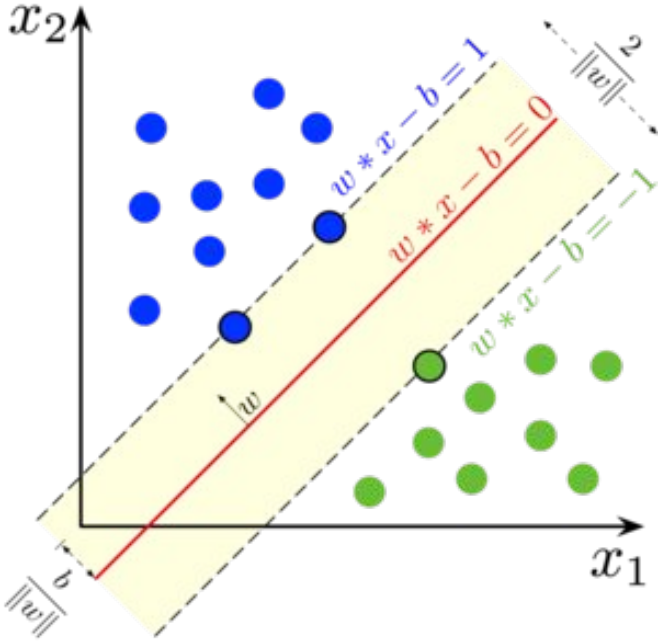


$$S = \frac{\sigma^2_{\text{between}}}{\sigma^2_{\text{within}}} = \frac{(\vec{w}\cdot\vec{\mu}_1 - \vec{w}\cdot\vec{\mu}_0)^2}{\vec{w}^T\Sigma_1\vec{w} + \vec{w}^T\Sigma_0\vec{w}} = \frac{(\vec{w}\cdot(\vec{\mu}_1 - \vec{\mu}_0))^2}{\vec{w}^T(\Sigma_0 + \Sigma_1)\vec{w}}$$

w: orthogonal to the decision boundary

Support Vector Machines (SVM)

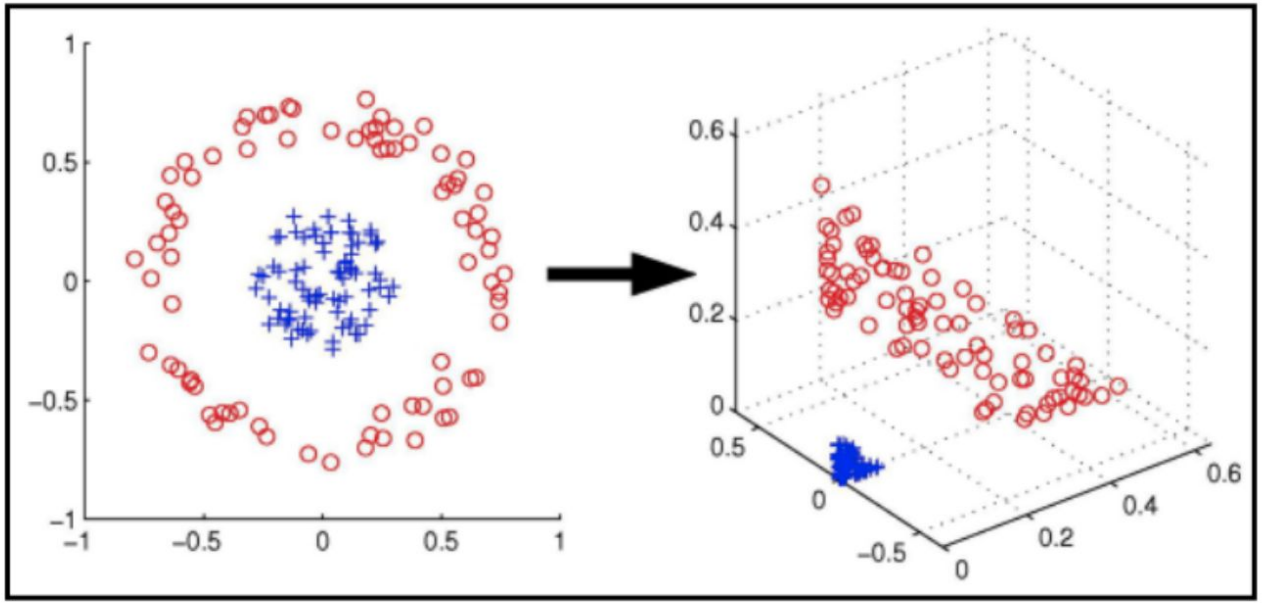




Classifier with max-margin

$$\lambda\|\mathbf{w}\|^2 + \left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i - b)\right)\right]$$

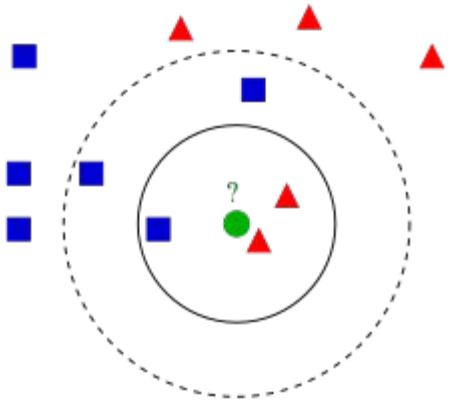
SVM with kernel tricks



$$\lambda\|\mathbf{w}\|^2 + \left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i - b)\right)\right]$$

Kernel function applied to input data first

k nearest neighbors (k-NN)



3-NN: red triangle
5-NN: blue square

Unsupervised learning

k-means clustering

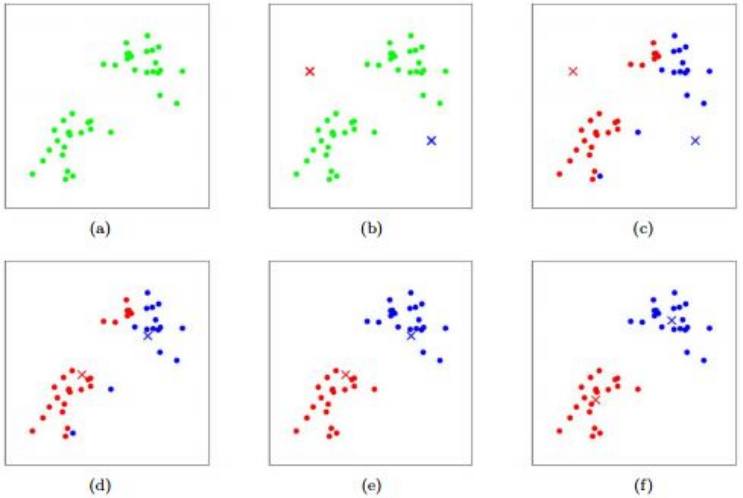
$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

Assignment step:

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \ \forall j, 1 \leq j \leq k \right\},$$

Update step:

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$



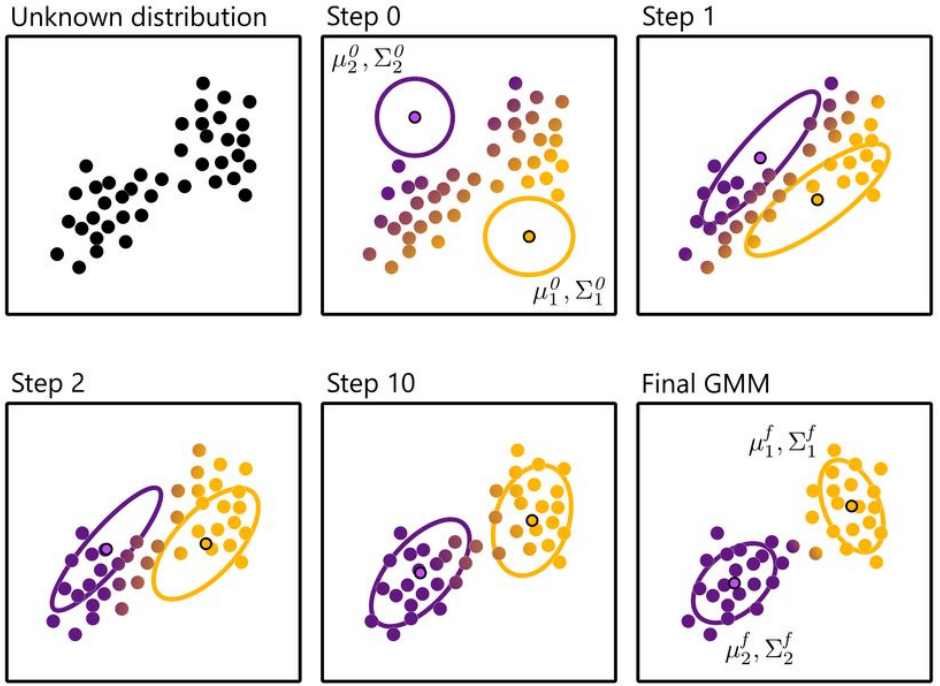
Expectation-maximization (EM) algo and Gaussian mixture model (GMM)

Expectation step:

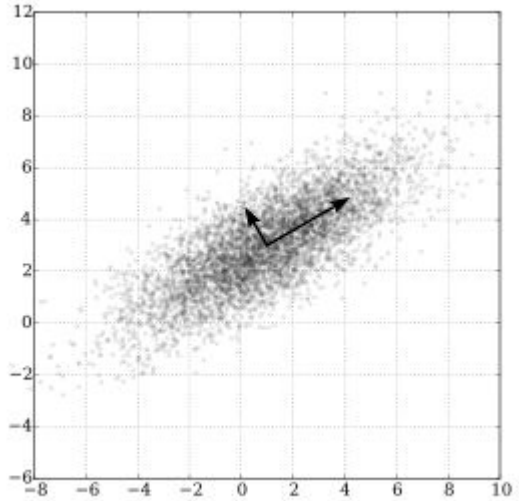
$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) = \mathrm{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)}}\left[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})\right]$$

Maximization step:

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)})$$



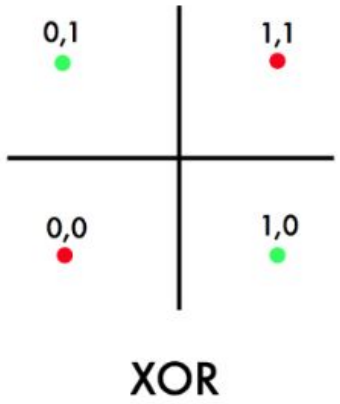
Principal component analysis (PCA)



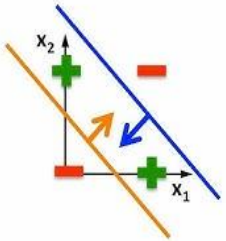
$$\mathbf{w}_{(1)} = \arg\max \left\{ \frac{\mathbf{w}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{X}\mathbf{w}}{\mathbf{w}^{\mathsf{T}}\mathbf{w}} \right\}$$

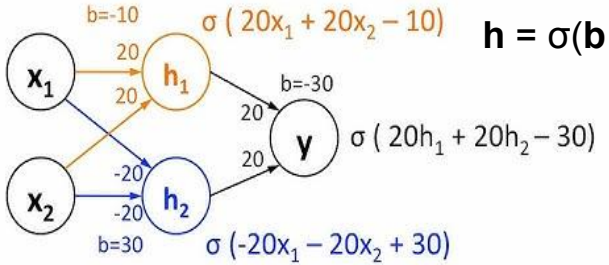
Deep learning

Neural networks



Linear classifiers cannot solve this



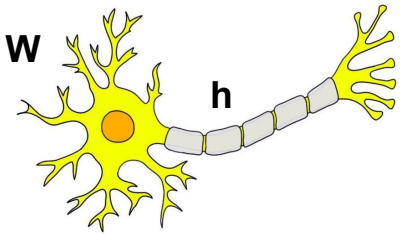


h = σ(b + Wx)

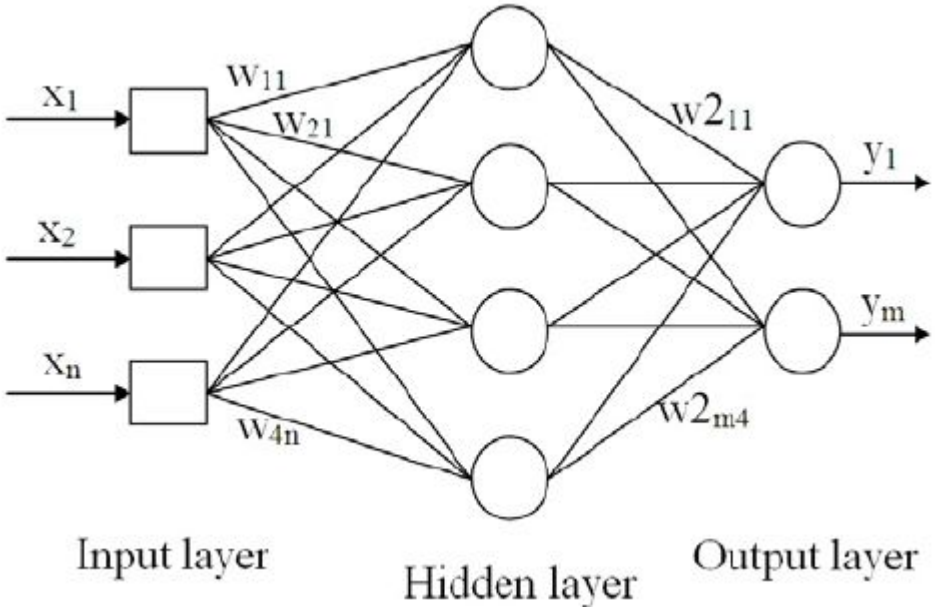
σ(20*0 + 20*0 – 10) ≈ 0
σ(20*1 + 20*1 – 10) ≈ 1
σ(20*0 + 20*1 – 10) ≈ 1
σ(20*1 + 20*0 – 10) ≈ 1

σ (-20*0 – 20*0 + 30) ≈ 1
σ (-20*1 – 20*1 + 30) ≈ 0
σ (-20*0 – 20*1 + 30) ≈ 1
σ (-20*1 – 20*0 + 30) ≈ 1

σ (20*0 + 20*1 – 30) ≈ 0
σ (20*1 + 20*0 – 30) ≈ 0
σ (20*1 + 20*1 – 30) ≈ 1
σ (20*1 + 20*1 – 30) ≈ 1



Neural networks - Multi-layer Perceptron (MLP)

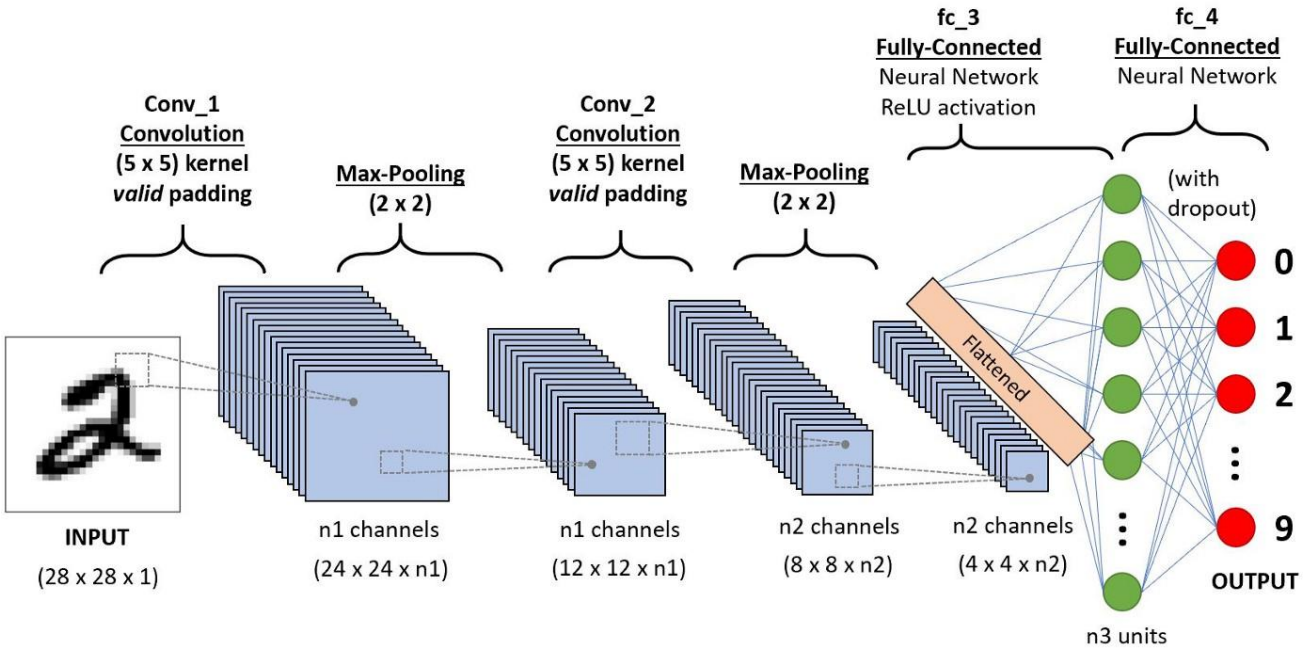


$$\mathbf{y} = \sigma_2(\mathbf{b}_2 + \mathbf{W}_2(\sigma_1(\mathbf{b}_1 + \mathbf{W}_1\mathbf{x}))$$

σ: activation function (ReLu, softmax, etc.)

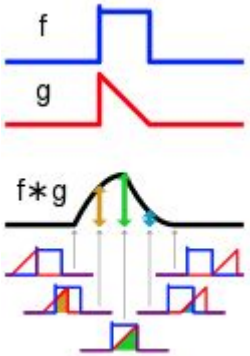
Trained by backpropagation and gradient descent

Neural networks - Convolutional Neural Network (CNN)



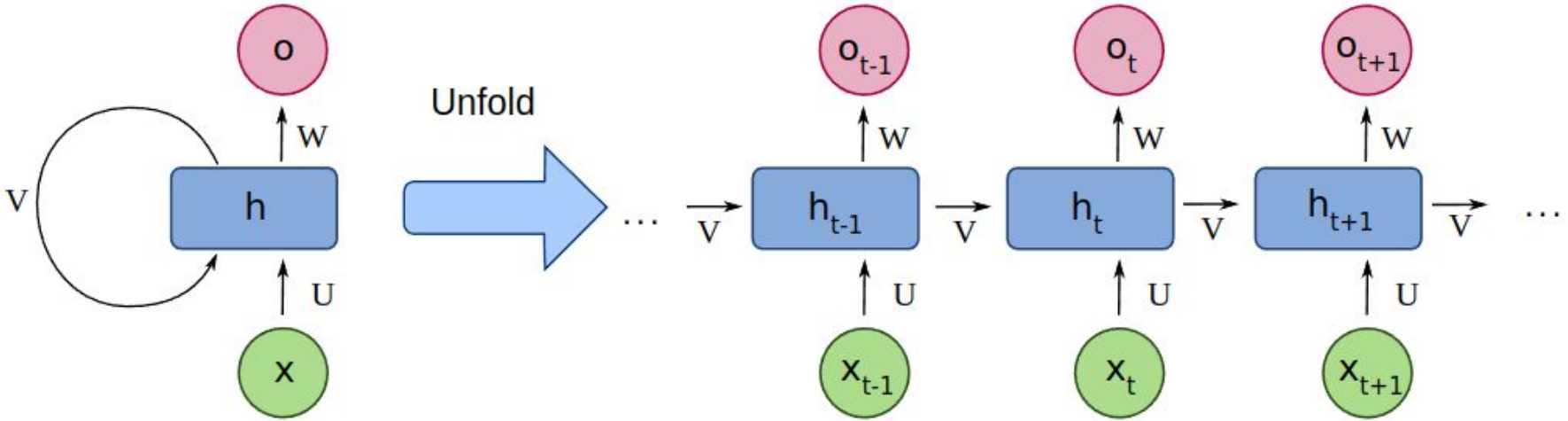
MLP with convolution

$\sigma_1(\mathbf{b}_1 + \text{conv}(\mathbf{W}_1, \mathbf{x}))$

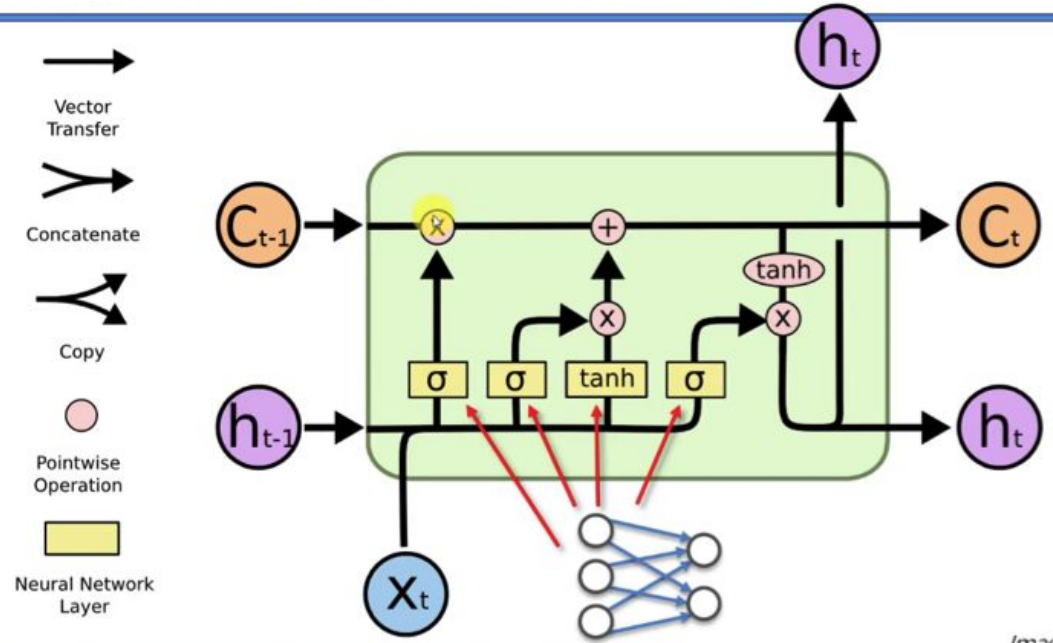


Convolution as weight sharing to reduce the size of the model

Neural networks - Recurrent Neural Network (RNN)



Neural networks - Long Short-term Memory (LSTM)

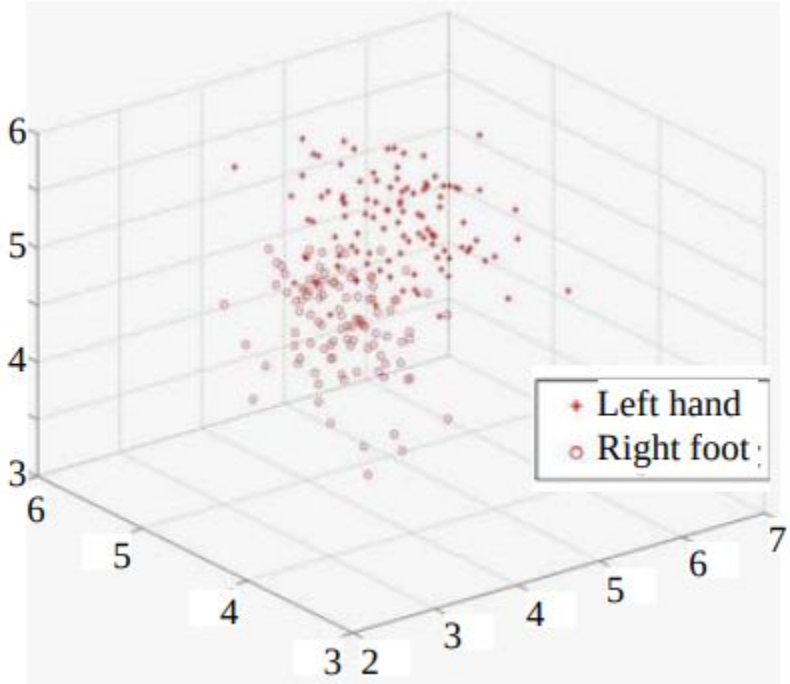


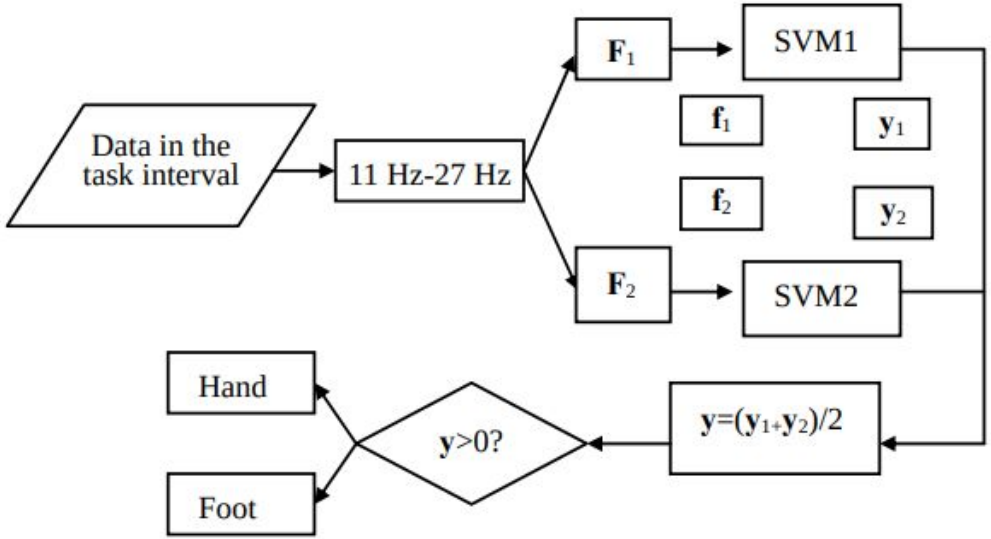
Resolve long-dependence in training RNN

Forget gate; memory gate

Examples

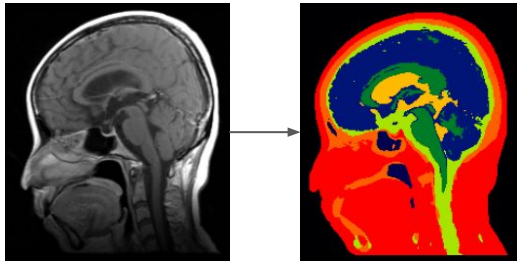
PCA + SVM in classifying EEG trials





Huang et al, 2009

EM + GMM in segmentation of head MRIs



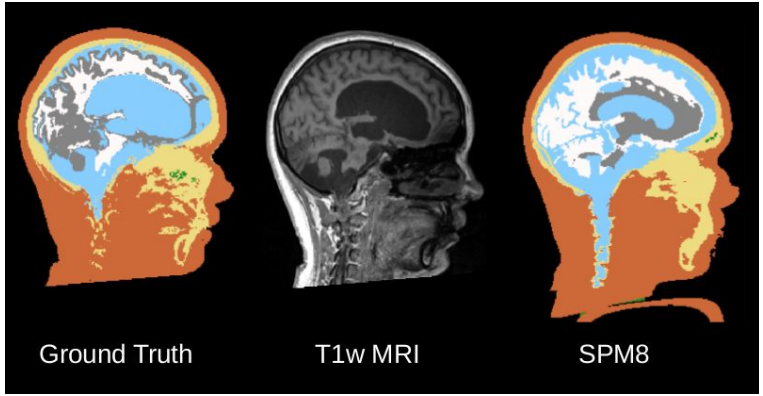
$$q(\mathbf{x}_i) = \frac{P(y_i|\mathbf{x}_i,\boldsymbol{\theta}) \exp\left[\frac{1}{2}\beta \sum_{j\in\mathcal{N}_i} \sum_{\mathbf{x}_j} q(\mathbf{x}_j)\mathbf{x}_i^T \mathbf{J}_{ij}\mathbf{x}_j + \mathbf{h}_i^T \mathbf{x}_i\right]}{\sum_{\mathbf{x}_i} P(y_i|\mathbf{x}_i,\boldsymbol{\theta}) \exp\left[\frac{1}{2}\beta \sum_{j\in\mathcal{N}_i} \sum_{\mathbf{x}_j} q(\mathbf{x}_j)\mathbf{x}_i^T \mathbf{J}_{ij}\mathbf{x}_j + \mathbf{h}_i^T \mathbf{x}_i\right]}.$$

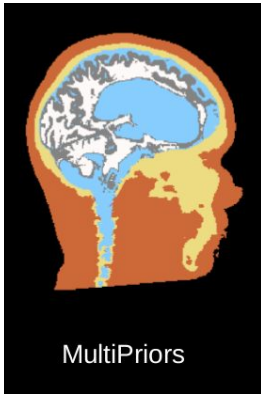
$$\boldsymbol{\mu}_x = \frac{\sum_{i=1}^{N} q(x_i = x)\mathbf{y}_i}{\sum_{i=1}^{N} q(x_i = x)},$$

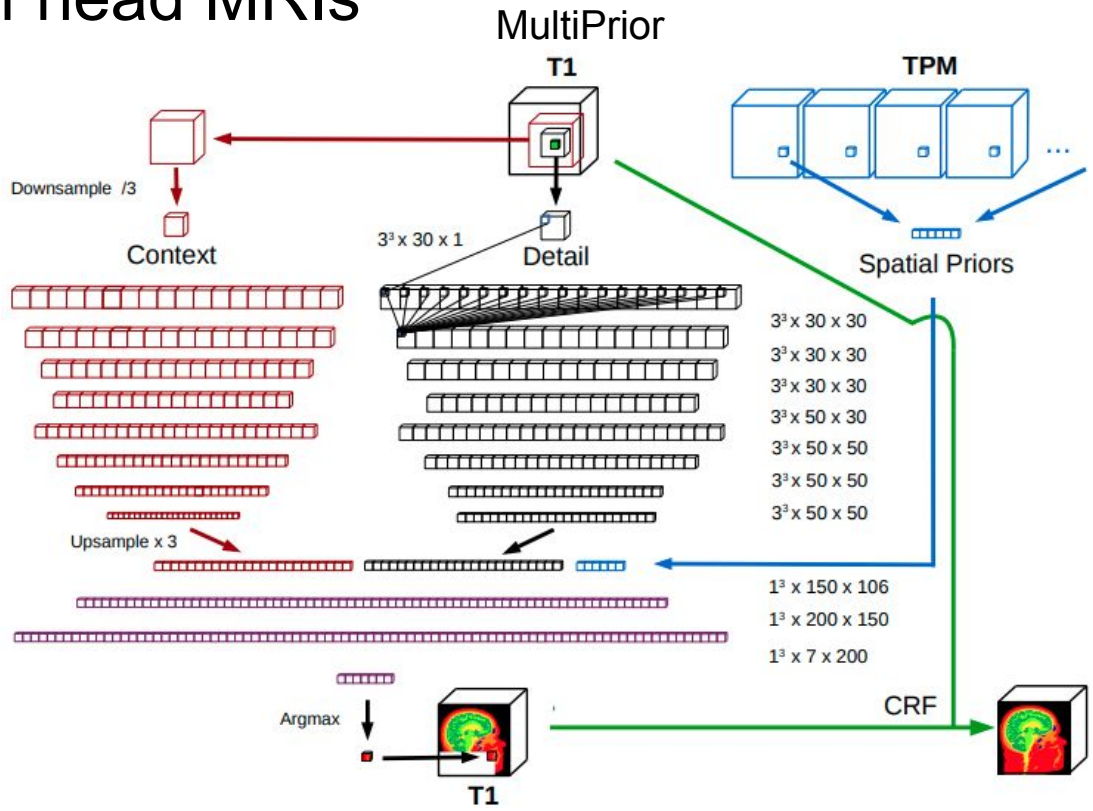
$$\boldsymbol{\Sigma}_x = \frac{\sum_{i=1}^{N} q(x_i = x)(\mathbf{y}_i - \boldsymbol{\mu}_x)(\mathbf{y}_i - \boldsymbol{\mu}_x)^T}{\sum_{i=1}^{N} q(x_i = x)}$$

Huang & Parra, 2015

CNN in segmentation of head MRIs

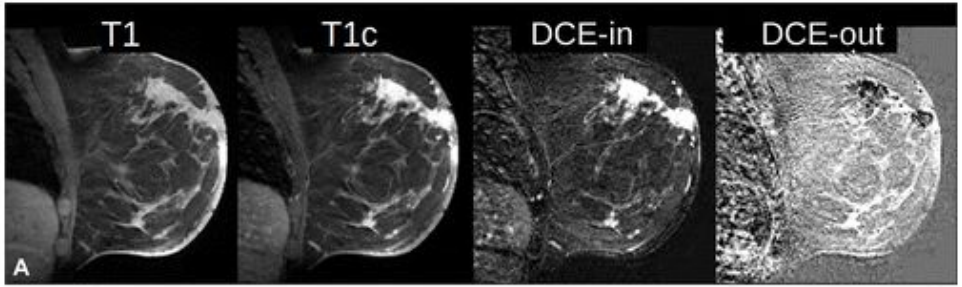






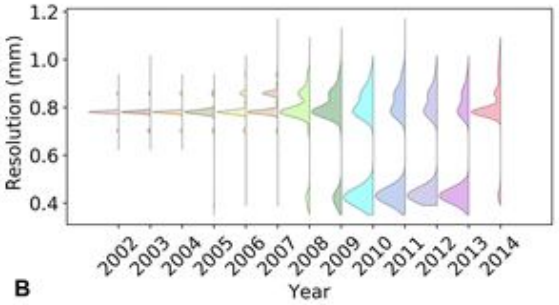
Hirsch, Huang, et al, 2021, *JMI*
https://github.com/lkshrsch/MultiPrior_Brain

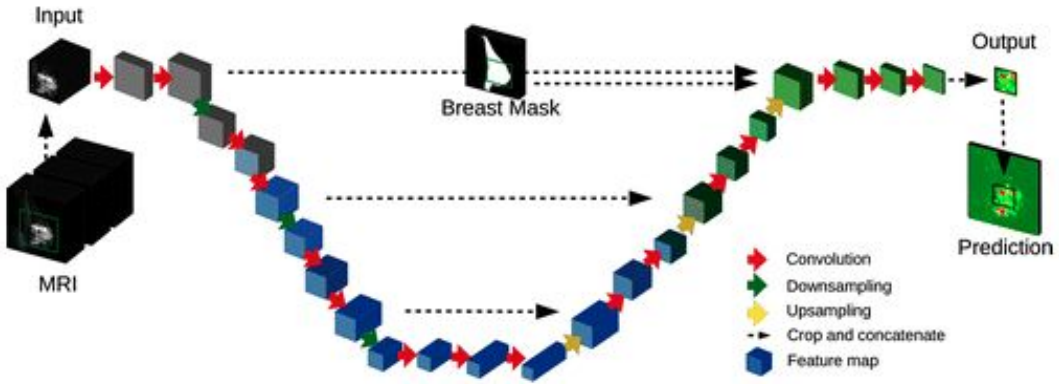
CNN in segmentation of breast MRIs



~64,000 MR scans
10TB data;
Training N=62,663
Testing N=250 (x4)

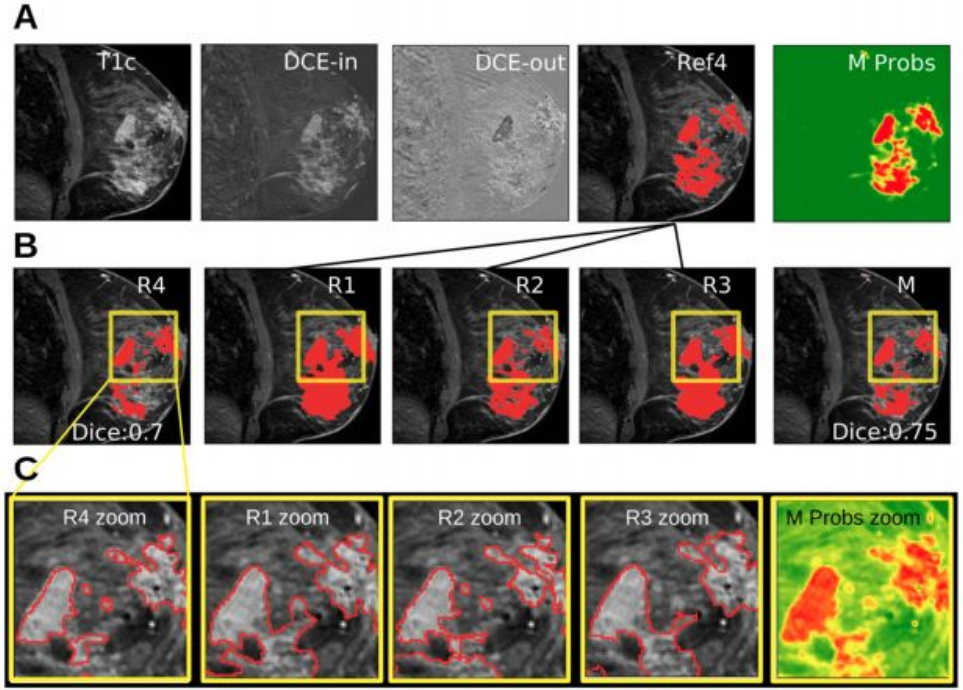
Elizabeth Sutton

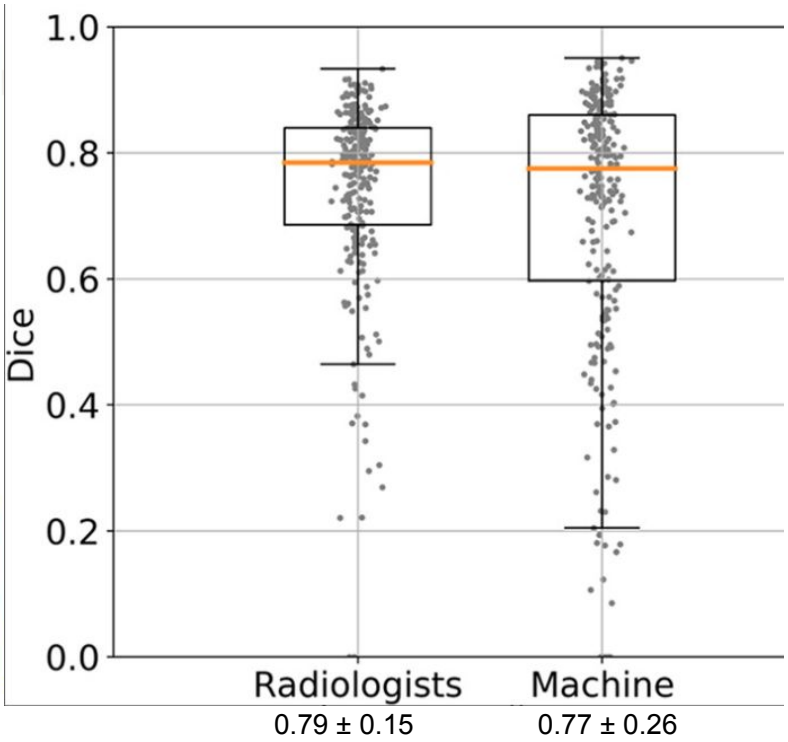




Hirsch, Huang, et al, 2021, *Radiology-AI*
https://github.com/lkshrsch/Segmentation_breast_cancer_MRI

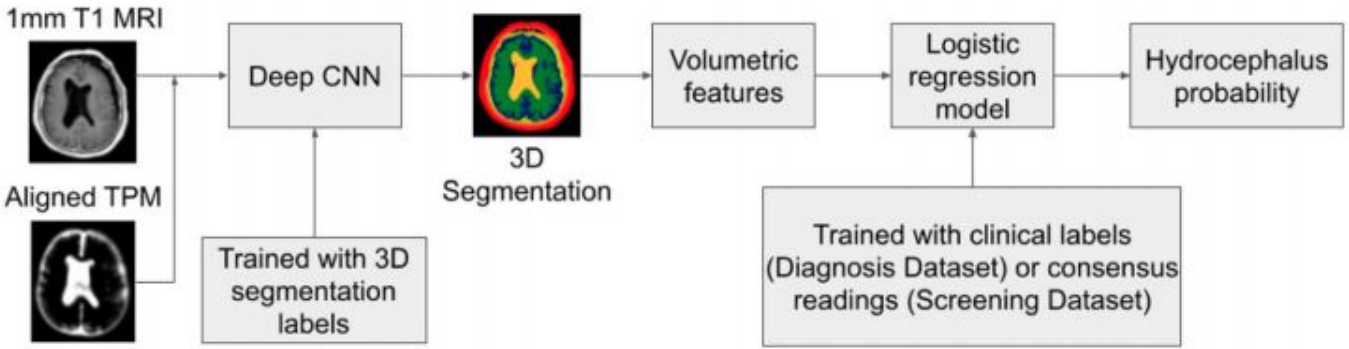
CNN in segmentation of breast MRIs



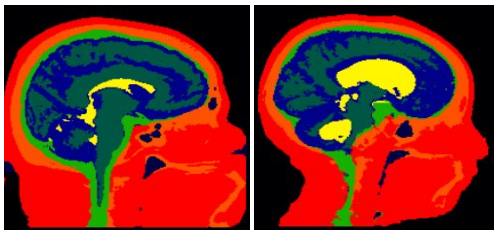


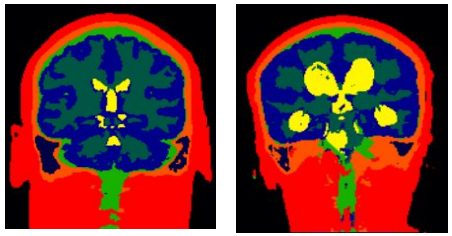
Hirsch, Huang, et al, 2021, *Radiology-AI*
https://github.com/lkshrsch/Segmentation_breast_cancer_MRI

CNN + logistic regression in diagnosis of hydrocephalus requiring treatment



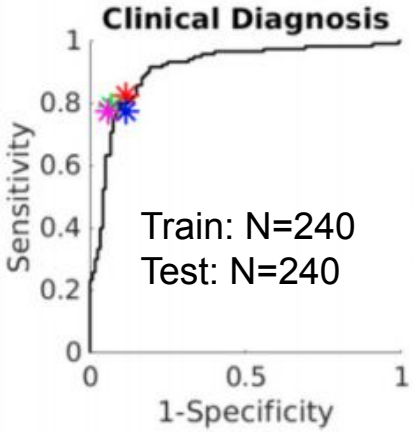
Robert Young

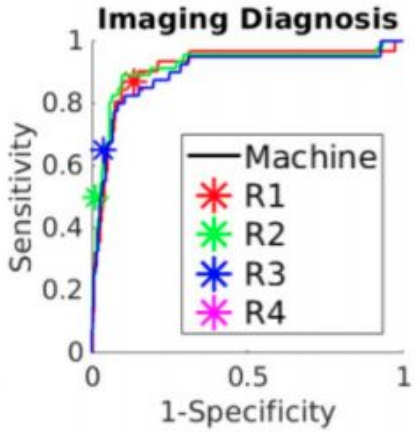




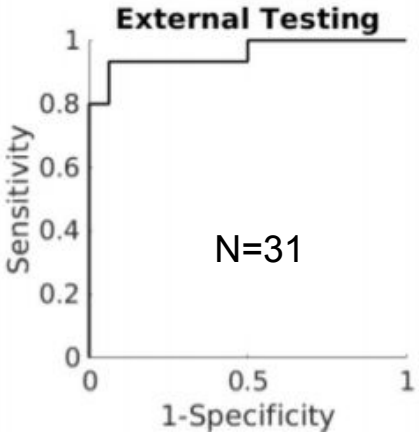
Huang, et al, 2022, *JDI*
https://github.com/andypotatohy/hydroDetector

CNN + logistic regression in diagnosis of hydrocephalus requiring treatment



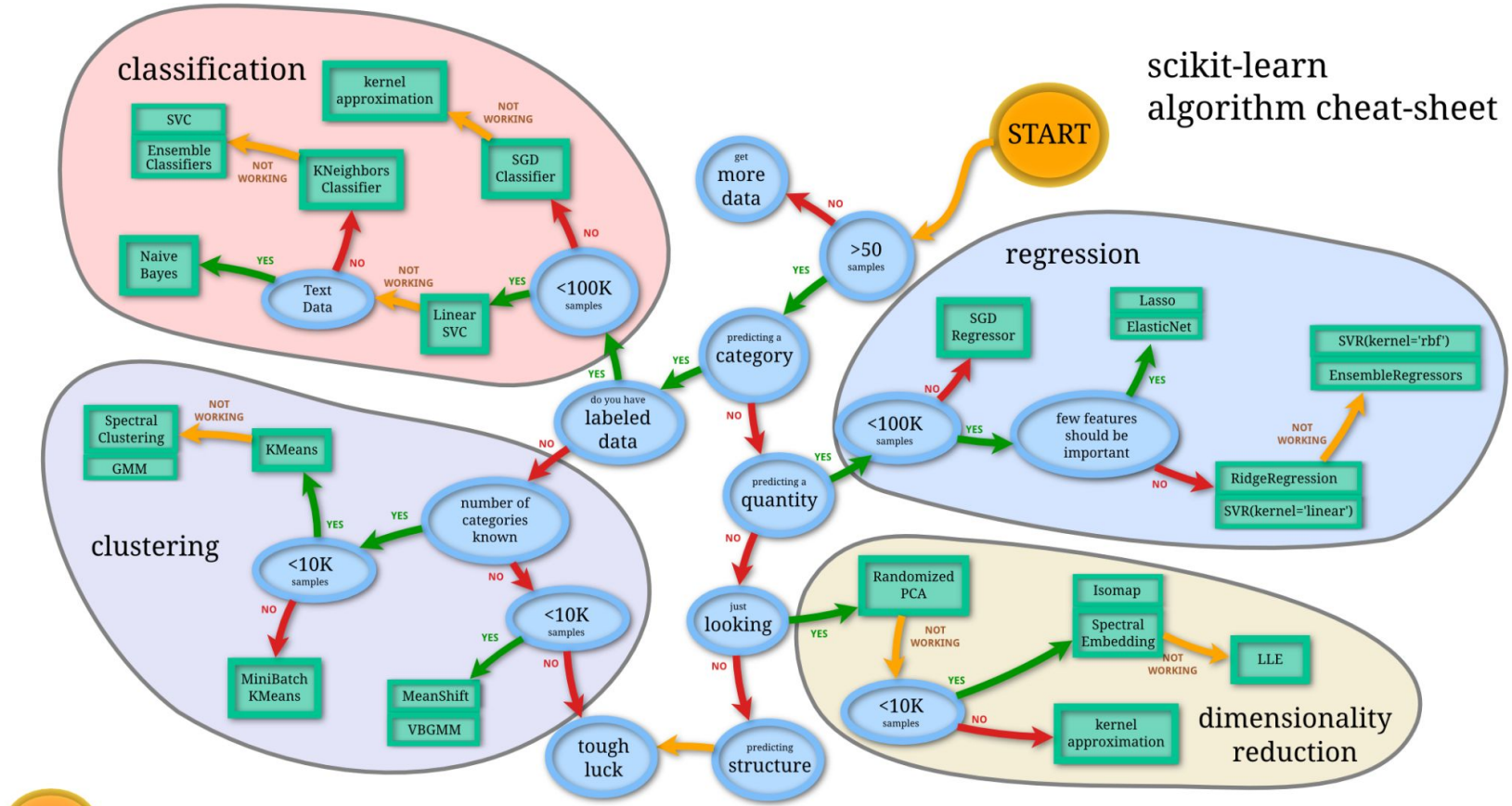


N=451,
Leave-one-out



Henry Rusinek

Huang, et al, 2022, *JDI*
https://github.com/andypotatohy/hydroDetector



https://medium.com/@lizziedotdev/lets-talk-about-machine-learning-ddca914e9dd1

Further readings

- Classic machine learning:
  - Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006
  - Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification*, John Wiley & Sons, Nov 9, 2012
- Deep learning:
  - Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, MIT Press, 2016
  - François Chollet, *Deep Learning with Python*, Manning; 2nd edition, December 21, 2021

Q & A