Computational Neuroscience Laboratory

Machine Learning II

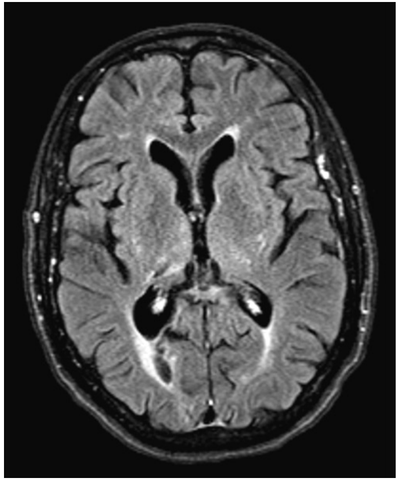
Autumn 2023

Session 6 – 10/12/2023

Today…

- Recap of previous ML session
- Gradient descent and neural networks
- Feature selection
- Performance Metrics
- Hypothesis testing of ML algorithms
- Projects

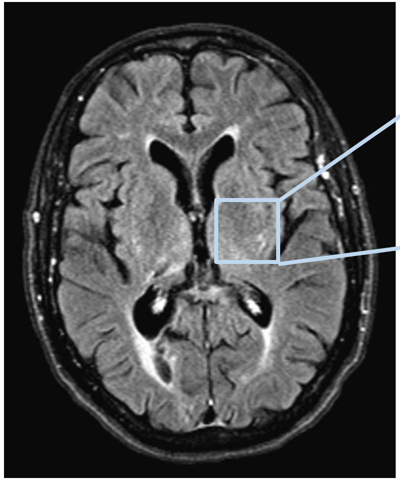
Classification: A core task



(assume given set of discrete labels)
{AD, MCI, PD, ASD, ...}

→ Alzheimer's disease

The Problem: Semantic Gap



```
[[105 112 108 111 104  99 106  99  96 103 112 119 104  97  93  87]
 [ 91  98 102 106 104  79  98 103  99 105 123 136 110 105  94  85]
 [ 76  85  90 105 128 105  87  96  95  99 115 112 106 103  99  85]
 [ 99  81  81  93 120 131 127 100  95  98 102  99  96  93 101  94]
 [106  91  61  64  69  91  88  85 101 107 109  98  75  84  96  95]
 [114 108  85  55  55  69  64  54  64  87 112 129  98  74  84  91]
 [133 137 147 103  65  81  80  65  52  54  74  84 102  93  85  82]
 [128 137 144 140 109  95  86  70  62  65  63  63  60  73  86 101]
 [125 133 148 137 119 121 117  94  65  79  80  65  54  64  72  98]
 [127 125 131 147 133 127 126 131 111  96  89  75  61  64  72  84]
 [115 114 109 123 150 148 131 118 113 109 100  92  74  65  72  78]
 [ 89  93  90  97 108 147 131 118 113 114 113 109 106  95  77  80]
 [ 63  77  86  81  77  79 102 123 117 115 117 125 125 130 115  87]
 [ 62  65  82  89  78  71  80 101 124 126 119 101 107 114 131 119]
 [ 63  65  75  88  89  71  62  81 120 138 135 105  81  98 110 118]
 [ 87  65  71  87 106  95  69  45  76 130 126 107  92  94 105 112]
 [118  97  82  86 117 123 116  66  41  51  95  93  89  95 102 107]
 [164 146 112  80  82 120 124 104  76  48  45  66  88 101 102 109]
 [157 170 157 120  93  86 114 132 112  97  69  55  70  82  99  94]
 [130 128 134 161 139 100 109 118 121 134 114  87  65  53  69  86]
 [128 112  96 117 150 144 120 115 104 107 102  93  87  81  72  79]
 [123 107  96  86  83 112 153 149 122 109 104  75  80 107 112  99]
 [122 121 102  80  82  86  94 117 145 148 153 102  58  78  92 107]
 [122 164 148 103  71  56  78  83  93 103 119 139 102  61  69  84]]
```

What the computer sees

An image is just a big grid of numbers between [0, 255]:

e.g. 800 x 600 x 3
(3 channels RGB)

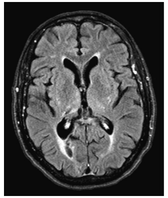
Different Challenges

- 3D Volumetric data / 4D Volume+Time / …
- Scanner and protocol difference
- Intrasubject variabilities of brain structure and function
- Brain disorders/diseases have different pathologies and development patterns
- Data imperfection (motion, blur, missing data, …)

Linear Classification

Parametric Approach

Image



Array of **32x32x32** numbers (32768 numbers total)

$\rightarrow$ f(**x**,**W**) $\rightarrow$ **10** numbers giving class scores

$\uparrow$

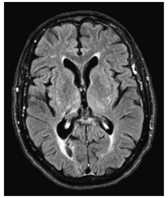
W

parameters or weights

Parametric Approach: Linear Classifier

$$f(x,W) = Wx$$

Image



Array of **32x32x32** numbers (32768 numbers total)

$\xrightarrow{\quad}$ f(**x**,**W**) $\xrightarrow{\quad}$ **10** numbers giving class scores

W

parameters or weights

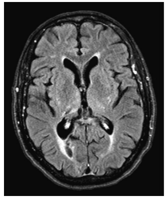
Parametric Approach: Linear Classifier

32768x1

$$f(x,W) = \boxed{W}\boxed{x}$$

10x1 10x32768

Image



Array of **32x32x32** numbers (32768 numbers total)

$\rightarrow$ f(**x**,**W**) $\rightarrow$ **10** numbers giving class scores

W

parameters or weights

Parametric Approach: Linear Classifier

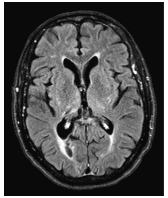
32768x1

$$\boxed{f(x,W) = \boxed{W}\boxed{x} + \boxed{b}}$$

10x1

10x1 10x32768

Image



Array of **32x32x32** numbers (32768 numbers total)

$f(\mathbf{x},\mathbf{W})$

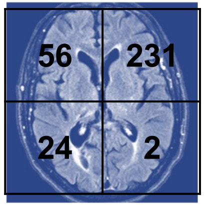
10 numbers giving class scores

W

parameters or weights

Example with an image with 4 pixels, and 3 classes (AD/MCI/NC)

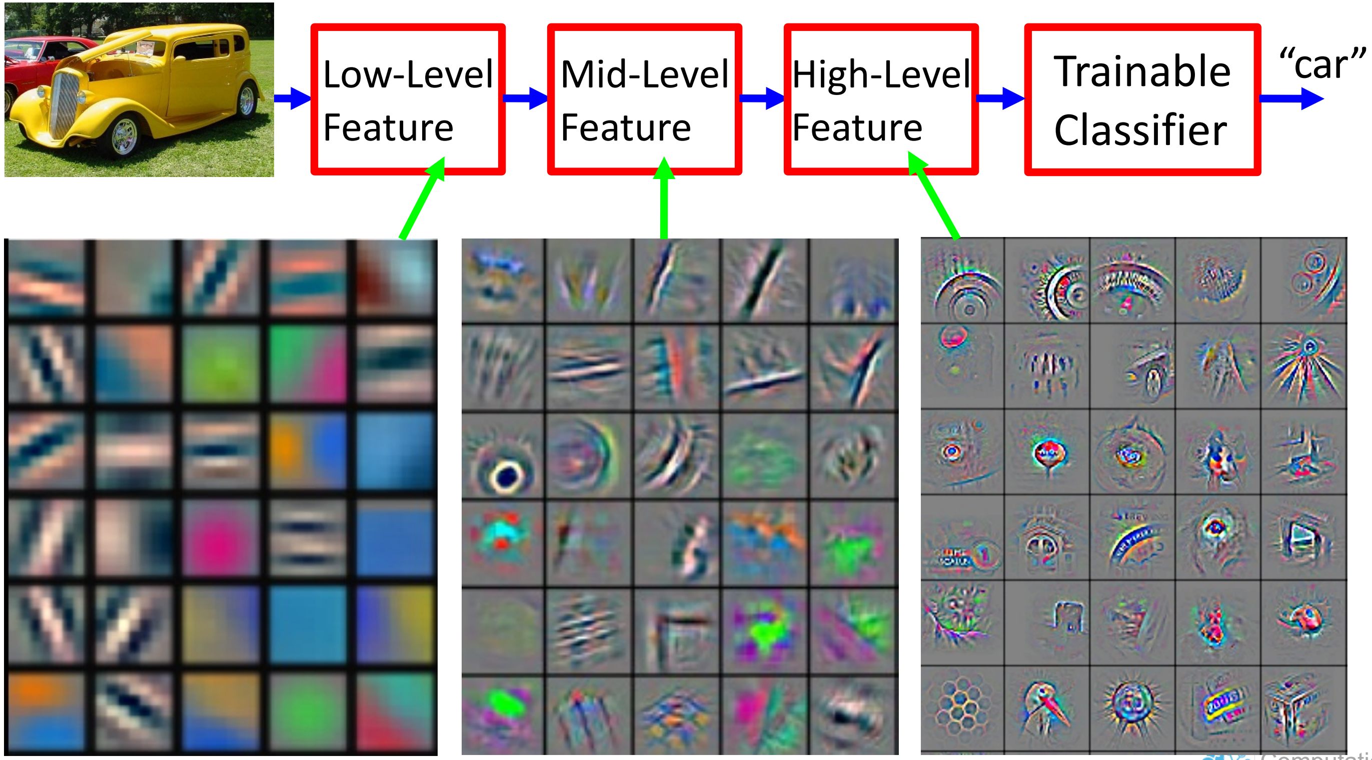
Stretch pixels into column



Input image

$$\underbrace{\begin{bmatrix} 0.2 & -0.5 & 0.1 & 2.0 \\ 1.5 & 1.3 & 2.1 & 0.0 \\ 0 & 0.25 & 0.2 & -0.3 \end{bmatrix}}_{W} \begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix} + \underbrace{\begin{bmatrix} 1.1 \\ 3.2 \\ -1.2 \end{bmatrix}}_{b} = \begin{array}{c} \begin{bmatrix} -96.8 \\ 437.9 \\ 61.95 \end{bmatrix} \end{array} \begin{array}{l} \text{AD score} \\ \text{MCI score} \\ \text{NC score} \end{array}$$

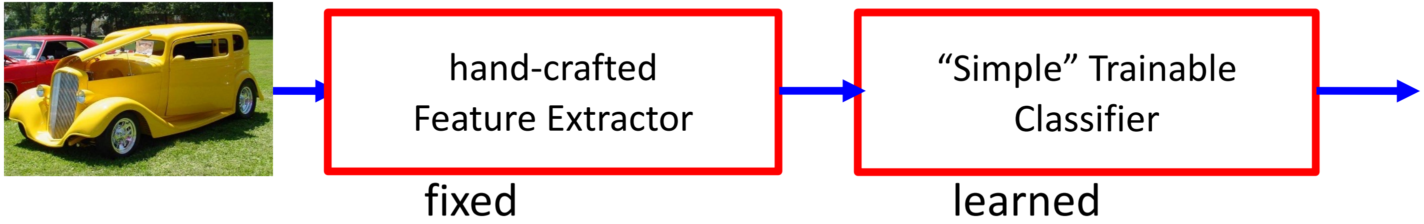
Deep Learning = Hierarchical Compositionality



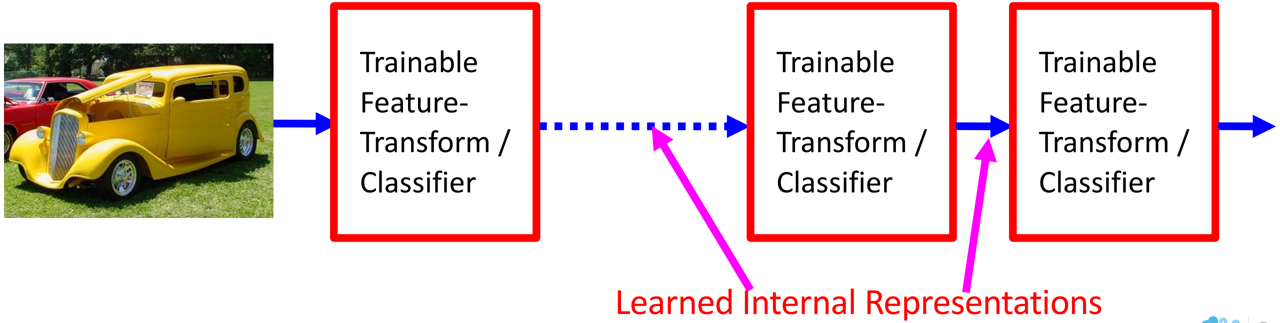
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

“Shallow” vs. Deep Learning

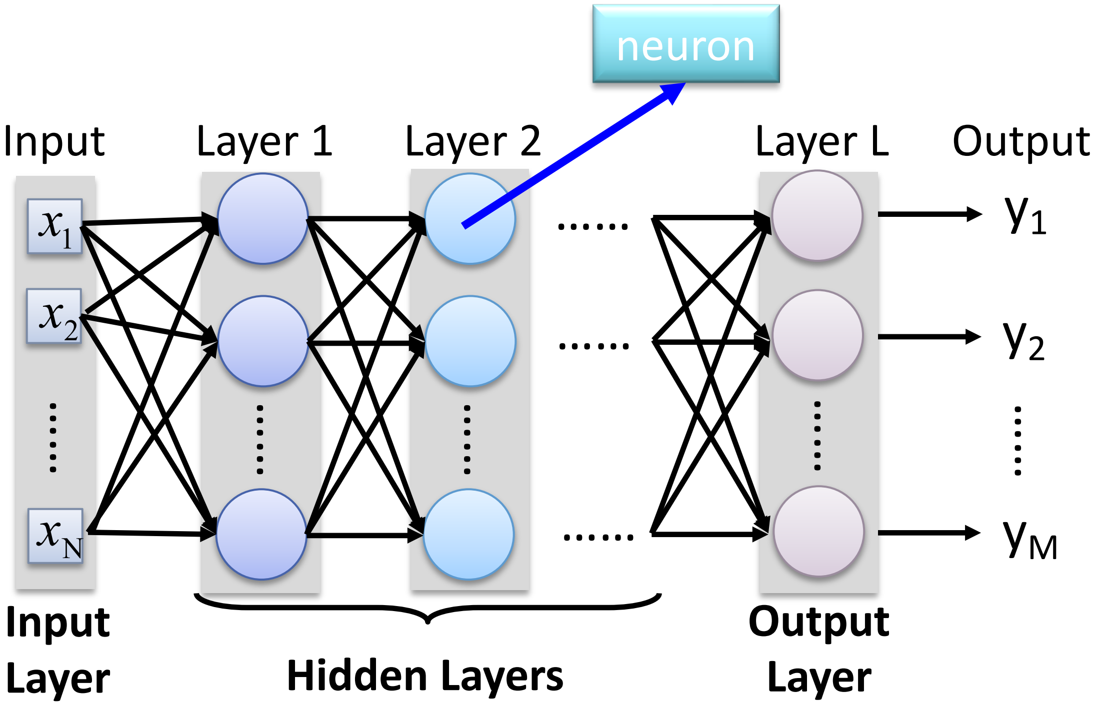
- “Shallow” models



- Deep models



Neural Network

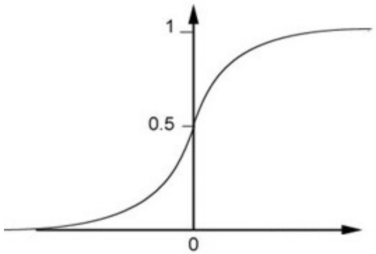


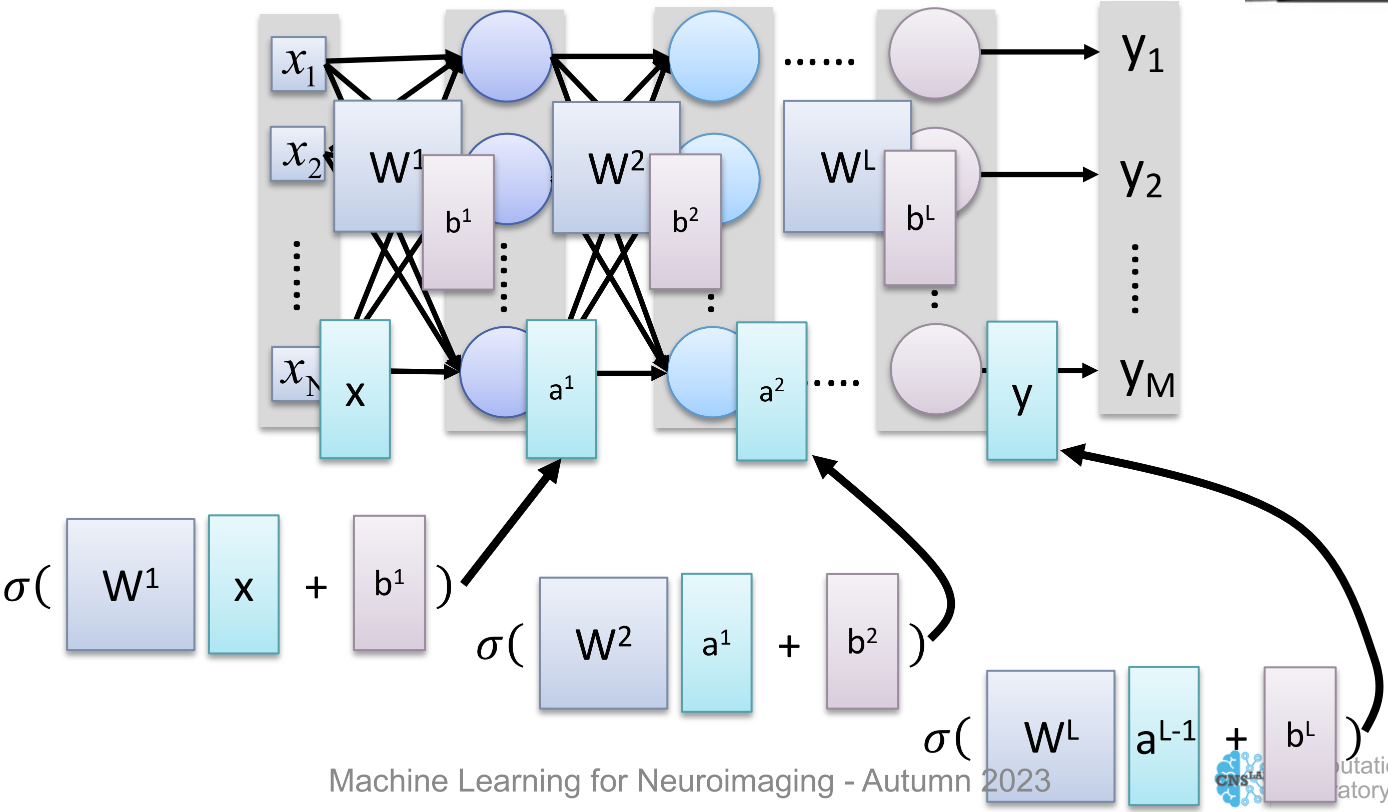
Deep means many hidden layers

Neural Network

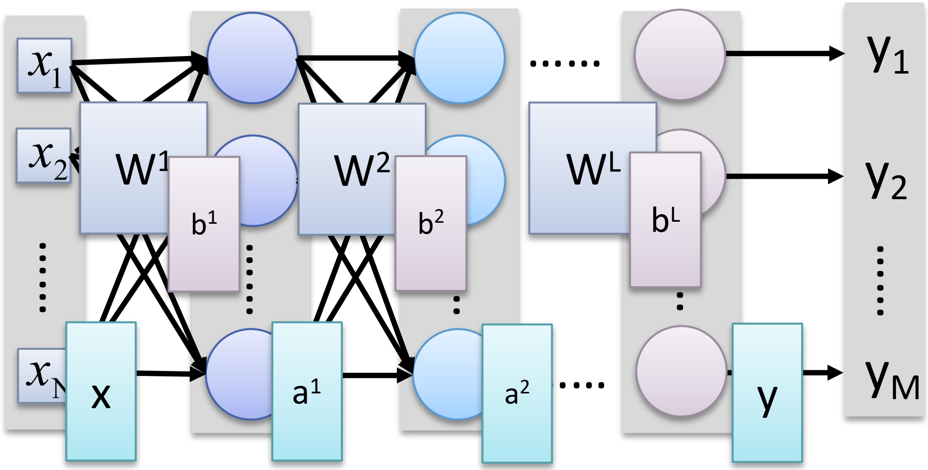
Sigmoid Function

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





Neural Network



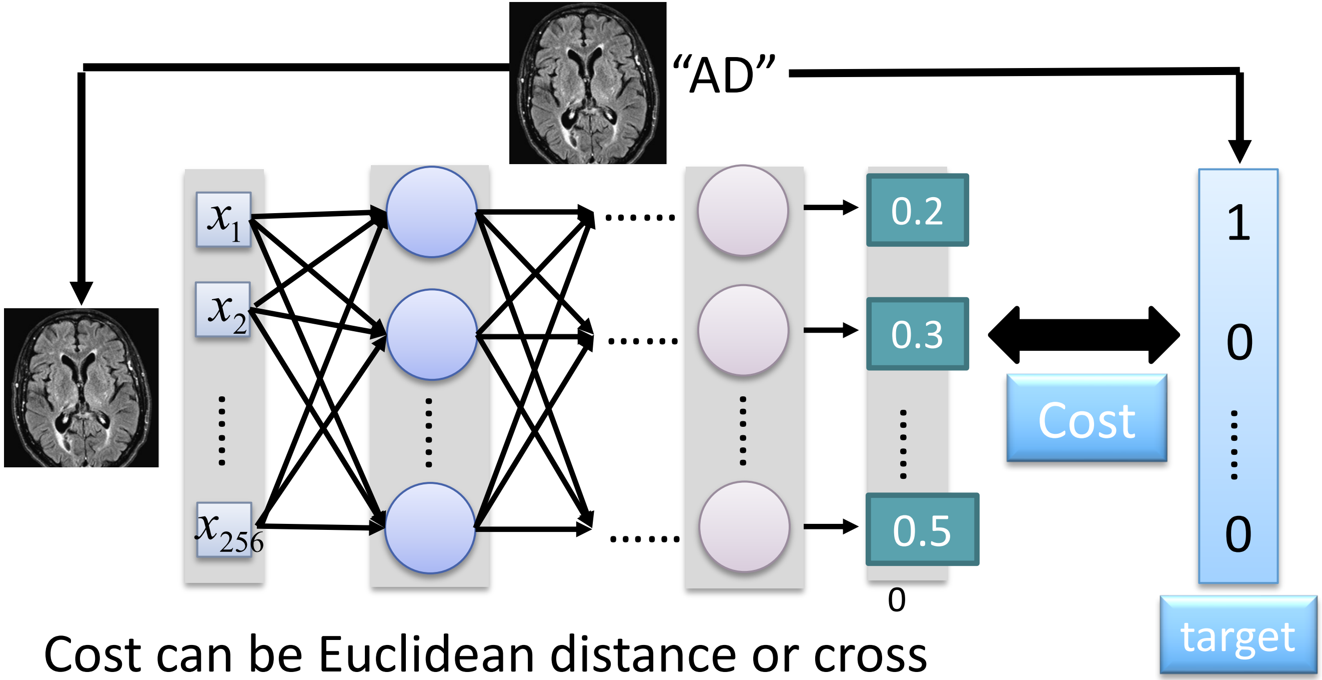
$$y = f(x)$$

Using parallel computing techniques to speed up matrix operation

$$= \sigma(W^L \cdots \sigma(W^2 \sigma(W^1 x + b^1) + b^2) \cdots + b^L)$$

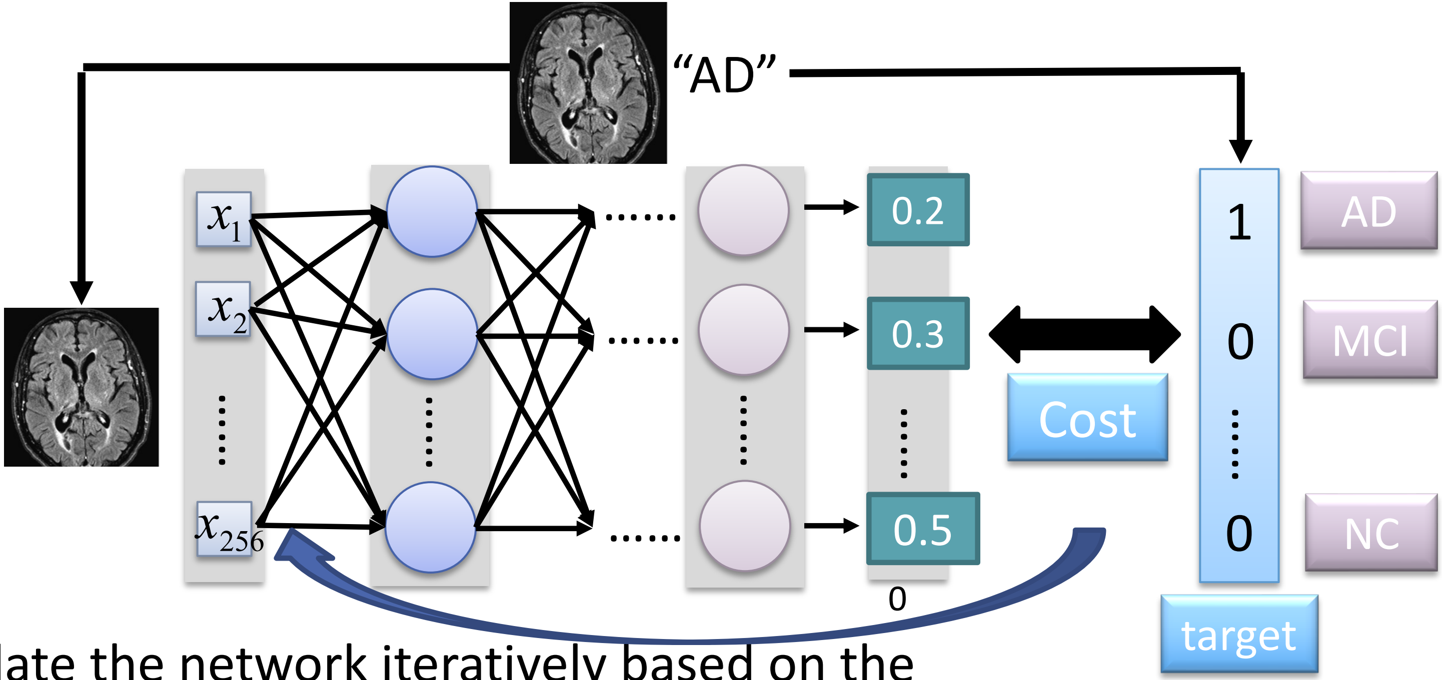
Cost

Given a set of network parameters, each example has a cost value.



Cost can be Euclidean distance or cross entropy of the network output and target

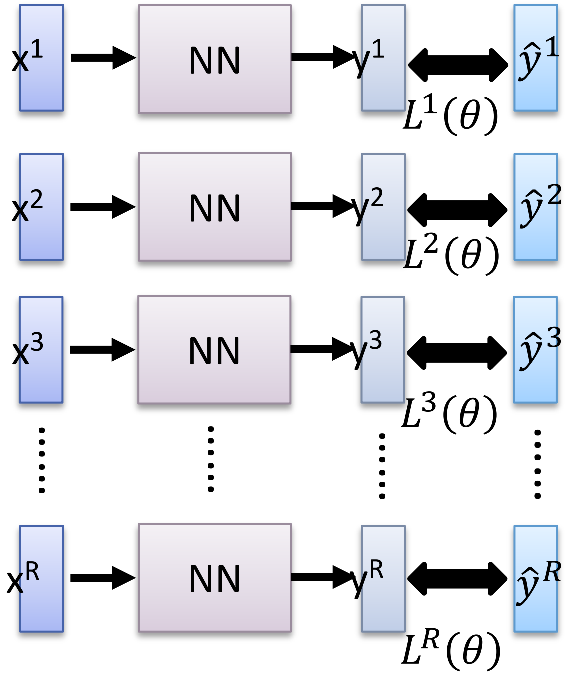
Training



Update the network iteratively based on the gradient of the cost until the network converges

Total Cost

For all training data …



Total Cost:

$$C(\theta) = \sum_{r=1}^{R} L^r(\theta)$$

How bad the network parameters θ is on this task

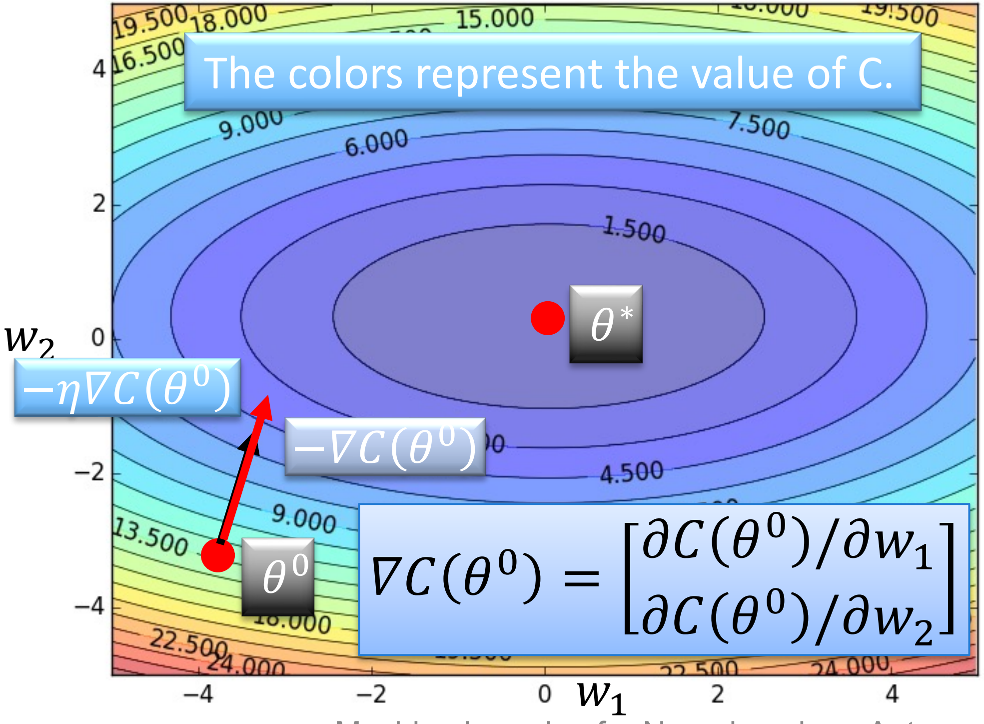
Find the network parameters θ^* that minimize this value

Gradient Descent

Assume there are only two parameters w_1 and w_2 in a network.

$$\theta = \{w_1, w_2\}$$

Error Surface



Randomly pick a starting point θ^0

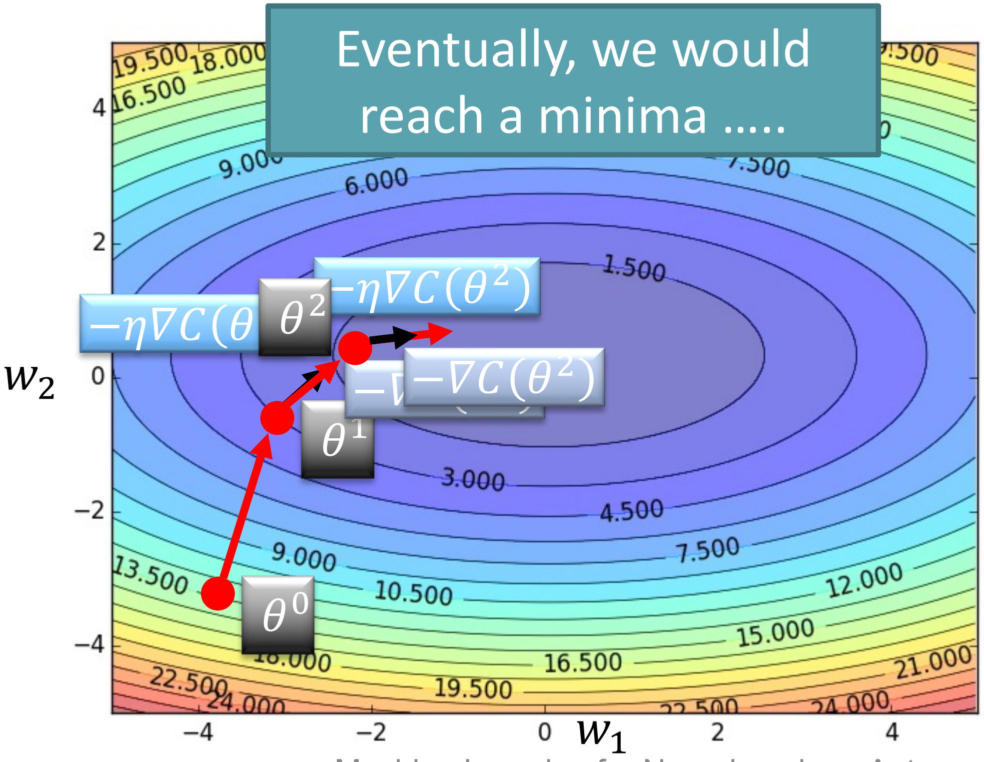
Compute the negative gradient at θ^0

$\Rightarrow -\nabla C(\theta^0)$

Times the learning rate η

$\Rightarrow -\eta \nabla C(\theta^0)$

Gradient Descent



Randomly pick a starting point θ^0

Compute the negative gradient at θ^0

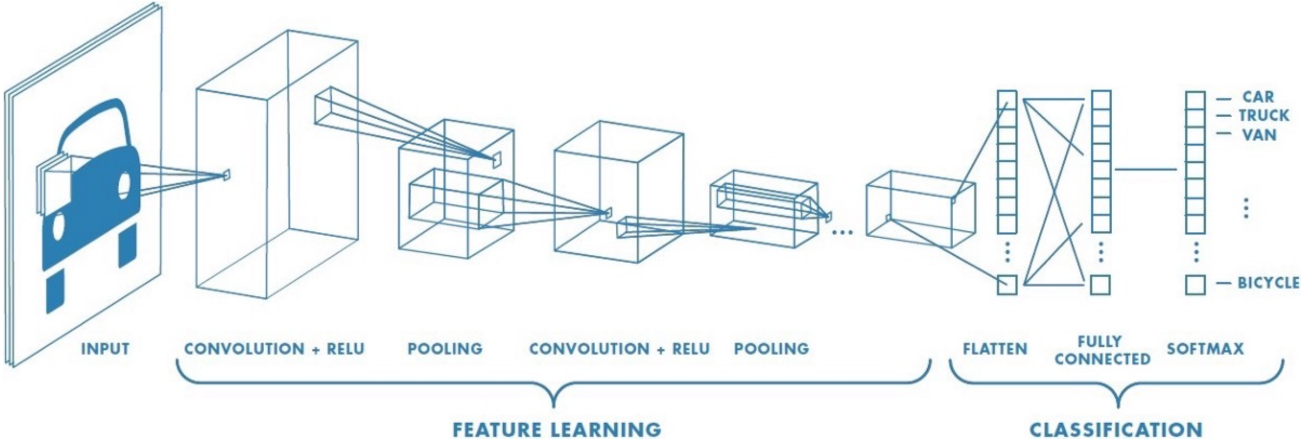
$\Rightarrow -\nabla C(\theta^0)$

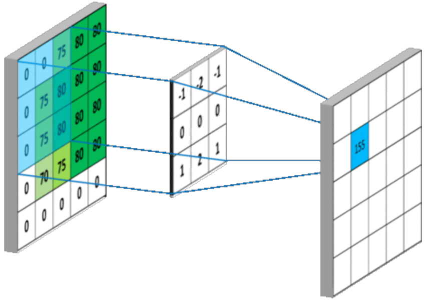
Times the learning rate η

$\Rightarrow -\eta\nabla C(\theta^0)$

Reading assignment

- Convolutional Neural Networks (CNN)
  - https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
  - https://www.kaggle.com/code/shivamb/3d-convolutions-understanding-use-case



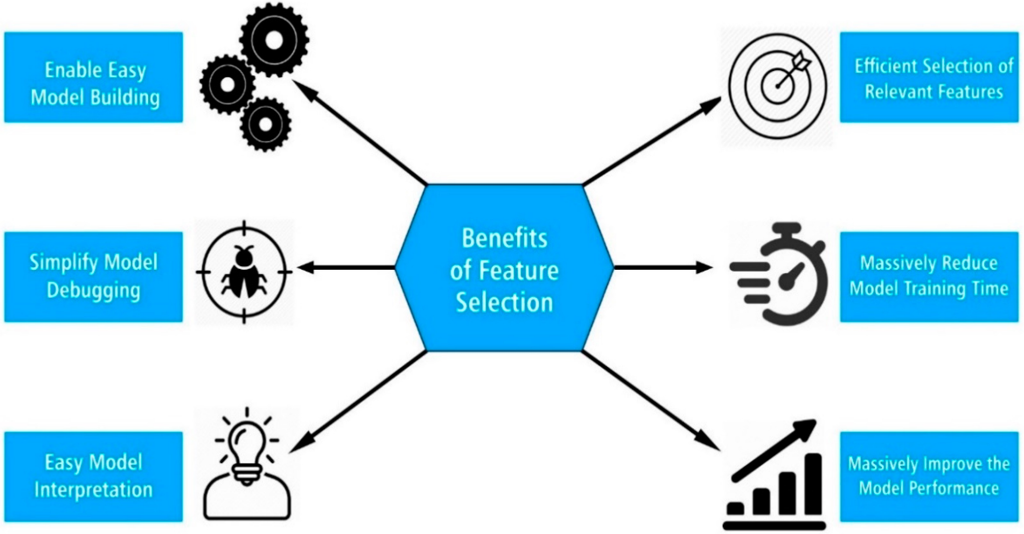


Today…

- Recap of previous ML session
- Gradient descent and neural networks
- Feature selection
- Performance Metrics
- Hypothesis testing of ML algorithms
- Projects

Why Feature Selection

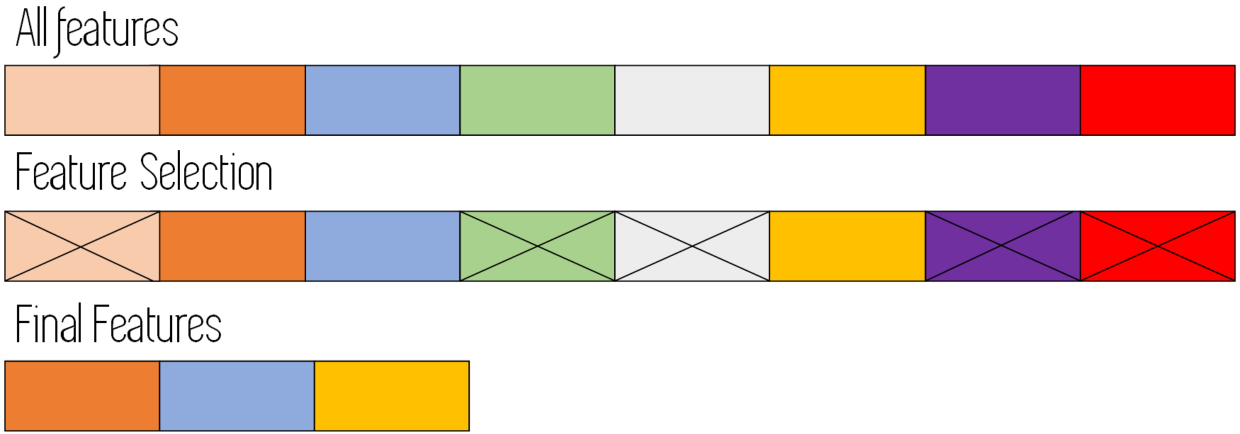
- In machine learning, one of the key challenges is to **select the right set of features** as inputs to a model.
- The features that are used to train a model will have a huge influence on the achieved performance.
- Irrelevant or partially relevant features can negatively impact the performance of a model.



heavy.ai

What is Feature Selection?

- Feature selection is the process of selecting a subset of relevant features used to train a machine learning model.



Why is feature selection important?

- Enhanced generalization by reducing overfitting
- Reduces training times
- Increase model interpretability
- Variable redundancy
- Reduces prediction time

Methods

- **Filter Methods** (fast, no/min feature interaction)
  - In this method, the selection of features is done independently of a machine learning algorithm. This method relies on the characteristics of the data to filter features based on a given metric.
  - Examples:
    - Chi-square test
    - Pearson Correlation
    - Mutual Information
    - **Minimum Redundancy-Maximum Relevance (mRMR)**

Minimum Redundancy-Maximum Relevance

- X is the selected feature subset
- x_i, x_j : feature in X
- C is the class labels
- *Rel*: relevance between X and c
- *Red*: redundancy within X

Objective Function:

$$Rel = \sum_{x_i \in X} I(x_i; C)$$

$$Red = \sum_{\substack{x_i, x_j \in X, \\ and\ i \neq j}} I(x_i; x_j)$$

$$\begin{aligned} \mathrm{I(X;Y)} &= \mathrm{H(X) - H(X|Y) = H(Y) - H(Y|X)} \\ &= \sum_{x \epsilon X, y \epsilon Y} p(x,y) log_2 \frac{p(x,y)}{p(x)p(y)} \end{aligned}$$

Minimum Redundancy-Maximum Relevance

- S is the feature subset, Ω is the pool of all candidate features, the minimum redundancy condition is:

$$\min_{s \subset \Omega} \frac{1}{|S|^2} \sum_{i,j \in S} I(x_i, x_j)$$

where |S| is the number of features in S.

- For classes $c=(c_i,....c_k)$ the maximum relevance condition maximizes the total relevance of all features in S:

$$\max_{s \subset \Omega} \frac{1}{|S|} \sum_{i \in S} I(x_i, c)$$

H.C. Peng, F.H. Long, and C. Ding, Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, 2005, pp. 1226–1238.

Minimum Redundancy-Maximum Relevance

- The mRMR feature set optimizes these two conditions simultaneously, either in quotient form:

$$\max_{s\subset\Omega}\left\{\frac{\sum_i I(x_i,c)}{\frac{1}{|S|}\sum_{i,j\in S} I(x_i,x_j)}\right\}$$

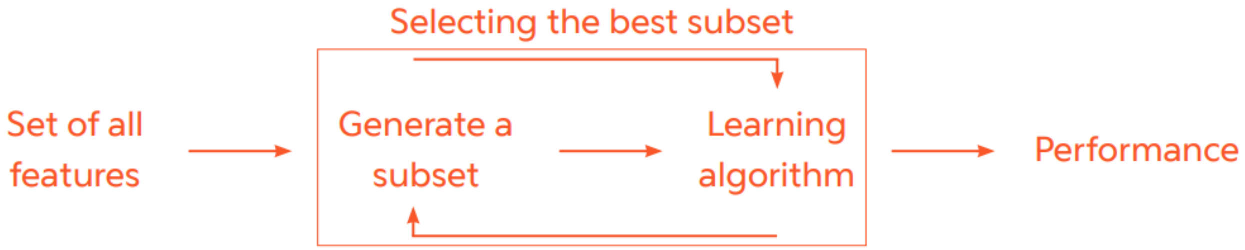
or in difference form:

$$\max_{s\subset\Omega}\left\{\sum_i I(x_i,c)-\frac{1}{|S|}\sum_{i,j\in S} I(x_i,x_j)\right\}$$

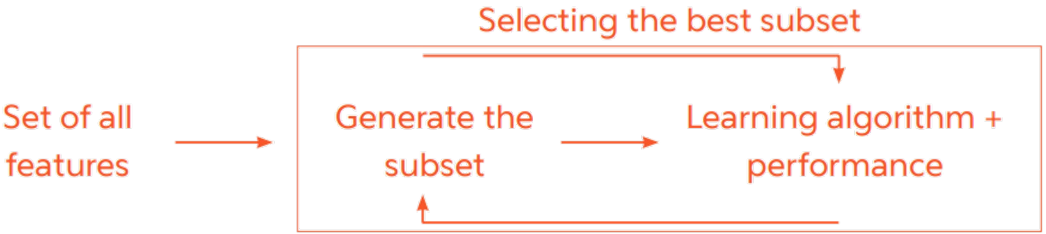
H.C. Peng, F.H. Long, and C. Ding, Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, 2005, pp. 1226–1238.

Methods

- Filter Methods (fast, no/min feature interaction)
- **Wrapper Methods** (slow, more accurate, considers feature interaction)
  - In this method, feature selection is based on a search criteria where a model is initially trained on a subset of features. Based on the inferences drawn from the previous model, we decide to either add or remove features.
  - Examples:
    - Recursive Feature Elimination (greedy)



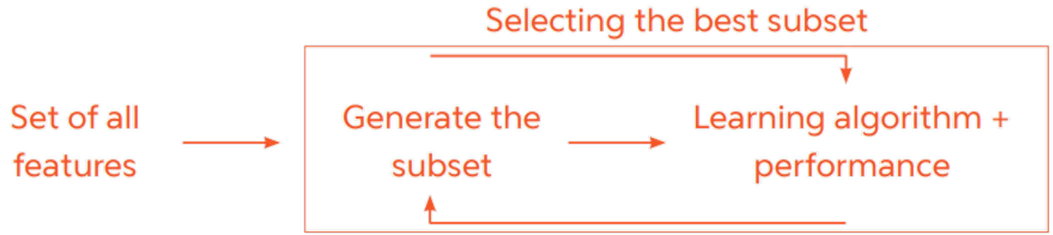
Methods



- Filter Methods (fast, no/min feature interaction)
- Wrapper Methods (slow, more accurate, considers feature interaction)
- **Embedded Methods** (more reliable feature estimates, reduce overfitting, robust to outliers, higher computational costs)
  - Embedded methods use the qualities of both the filter and wrapper methods. With this method, feature selection is embedded within the ML algorithm.
    - **LASSO regularization:** Lasso uses L1 regularization/penalty. It shrinks some parameters or feature coefficients to zero. It uses logistic regression to train a model with L1 penalty term to evaluate the coefficients of different variables and remove those variables with zero coefficients.

$$\sum_{i=1}^{n}\Big(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j\Big)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Methods



- Filter Methods (fast, no/min feature interaction)
- Wrapper Methods (slow, more accurate, considers feature interaction)
- **Embedded Methods** (more reliable feature estimates, reduce overfitting, robust to outliers, higher computational costs)
  - Lasso regularization.
  - Tree based random forest.
  - XGBoost
  - LightGBM
  - CatBoost…

Reading assignment 5

- An Introduction to Variable and Feature Selection, Journal of Machine Learning Research 2003
  - https://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf
- Make sure you understand
  - mRMR
  - LASSO feature selection

Today…

- Recap of previous ML session
- Gradient descent and neural networks
- Feature selection
- Performance Metrics
- Hypothesis testing of ML algorithms
- Projects

Which Classifier is better?

Almost as many answers as there are performance measures! (e.g., UCI Breast Cancer)

Algo	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC	Info S
NB	71.7	.4534	.44	.16	.53	.44	.48	.7	48.11
C4.5	75.5	.4324	.27	.04	.74	.27	.4	.59	34.28
3NN	72.4	.5101	.32	.1	.56	.32	.41	.63	43.37
Ripp	71	.4494	.37	.14	.52	.37	.43	.6	22.34
SVM	69.6	.5515	.33	.15	.48	.33	.39	.59	54.89
Bagg	67.8	.4518	.17	.1	.4	.17	.23	.63	11.30
Boost	70.3	.4329	.42	.18	.5	.42	.46	.7	34.48
RanF	69.23	.47	.33	.15	.48	.33	.39	.63	20.78

This and following slides courtesy of Nathalie Japkowicz, University of Ottawa

Which Classifier is better?
Ranking the results

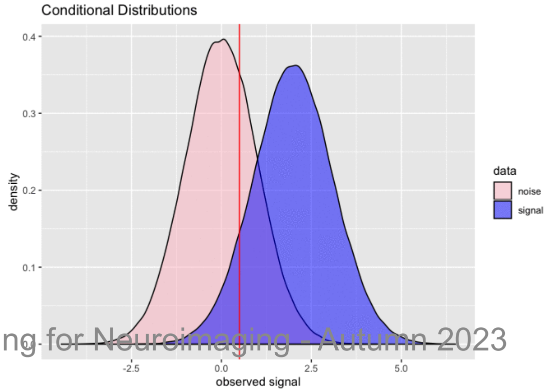
Algo	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC	Info S
NB	3	5	1	7	3	1	1	1	2
C4.5	1	1	7	1	1	7	5	7	5
3NN	2	7	6	2	2	6	4	3	3
Ripp	4	3	3	4	4	3	3	6	6
SVM	6	8	4	5	5	4	6	7	1
Bagg	8	4	8	2	8	8	8	3	8
Boost	5	2	2	8	7	2	2	1	4
RanF	7	6	4	5	5	4	7	3	7

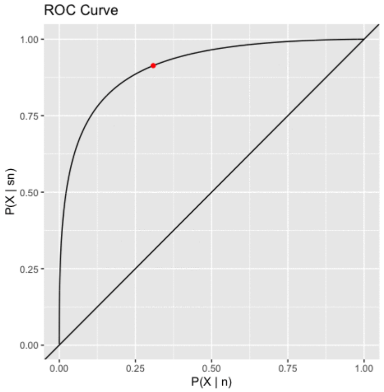
A Few Confusion Matrix-Based Performance Measures

True class → Hypothesized | class V	Pos	Neg
Yes	TP	FP
No	FN	TN
	P=TP+FN	N=FP+TN

A Confusion Matrix

- **Accuracy** = (TP+TN)/(P+N)
- **Precision** = TP/(TP+FP)
- **Recall/TP rate** = TP/P
- **FP Rate** = FP/N
- **ROC Analysis** moves the threshold between the positive and negative class from a small FP rate to a large one. It plots the value of the Recall against that of the FP Rate at each FP Rate considered.





Issues with Accuracy

True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

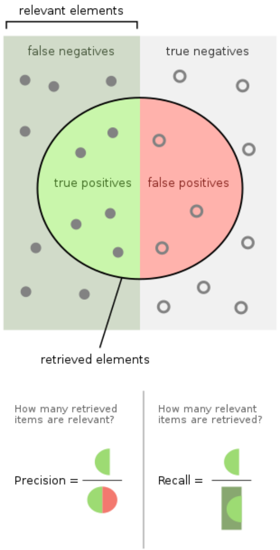
True class →	Pos	Neg
Yes	400	300
No	100	200
	P=500	N=500

- Both classifiers obtain 60% accuracy
- They exhibit very different behaviours:
  - On the left: weak positive recognition rate/strong negative recognition rate
  - On the right: strong positive recognition rate/weak negative recognition rate

Issues with Precision/Recall

True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

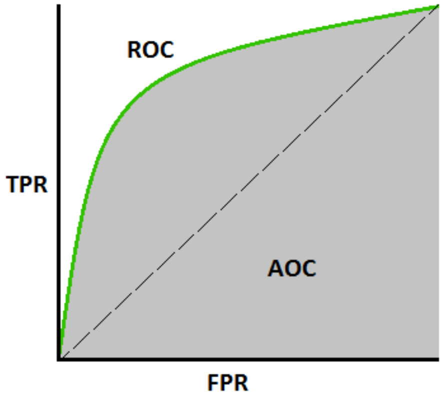
True class →	Pos	Neg
Yes	200	100
No	300	0
	P=500	N=100



- Both classifiers obtain the same precision and recall values of 66.7% and 40% (Note: the data sets are different)
- They exhibit very different behaviors:
  - Same positive recognition rate
  - Extremely different negative recognition rate: strong on the left / nil on the right
- Note: Accuracy has no problem catching this!

Is the AUC the answer?

- Many researchers have now adopted the AUC (the area under the ROC Curve).
- The principal advantage of the AUC is that it is **more robust** than Accuracy in class imbalanced situations.
- Indeed, given a 95% imbalance (in favour of the negative class, say), the accuracy of the default classifier that issues "negative" all the time will be 95%, whereas a more interesting classifier that actually deals with the issue, is likely to obtain a worse score.
- The AUC takes the **class distribution** into consideration.



RMSE

- The Root-Mean Squared Error (RMSE) is usually used for regression, but can also be used with probabilistic classifiers. The formula for the RMSE is:

$$\text{RMSE}(f) = \text{sqrt}\left(1/m \; \Sigma_{i=1}^{m}(f(x_i) - y_i)^2 \right)$$

where m is the number of test examples, $f(x_i)$, the classifier's probabilistic output on x_i and y_i the actual label.

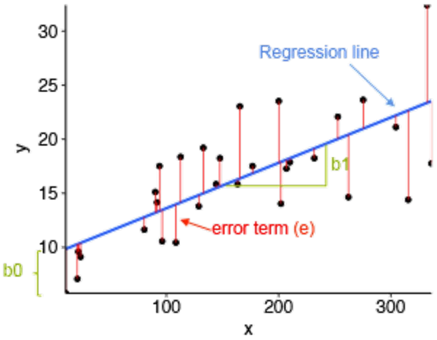
RMSE(f) RMSE(f)

ID	$f(x_i)$	y_i	$(f(x_i) - y_i)^2$
1	.95	1	.0025
2	.6	0	.36
3	.8	1	.04
4	.75	0	.5625
5	.9	1	.01

$$\text{RMSE}(f) = \text{sqrt}(1/5 * (.0025+.36+.04+.5625+.01))$$
$$= \text{sqrt}(0.975/5) = 0.4416$$

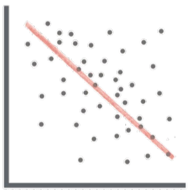
Performance Metric for Regression Tasks

- Error
  - RMSE
  - MAE

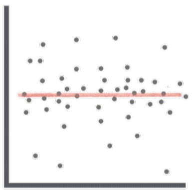


- Correlation
  - Squared Correlation (R^2)

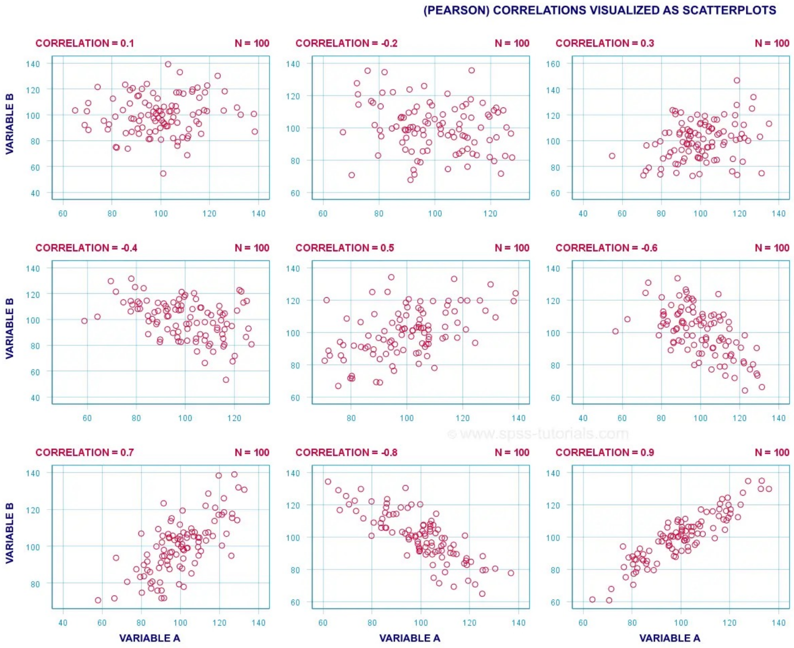
Positive Correlation



Negative Correlation

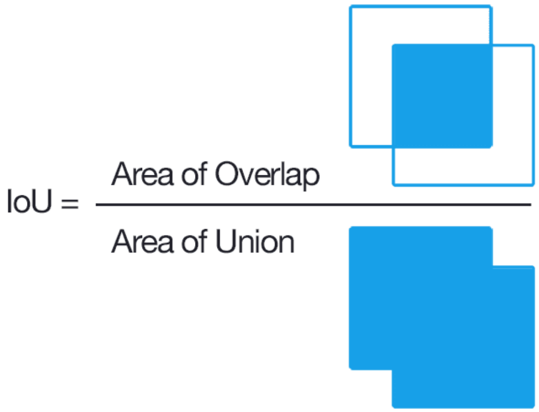


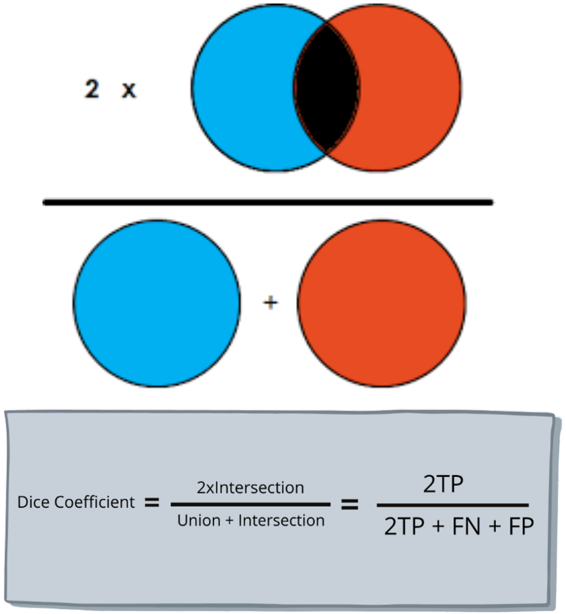
No Correlation



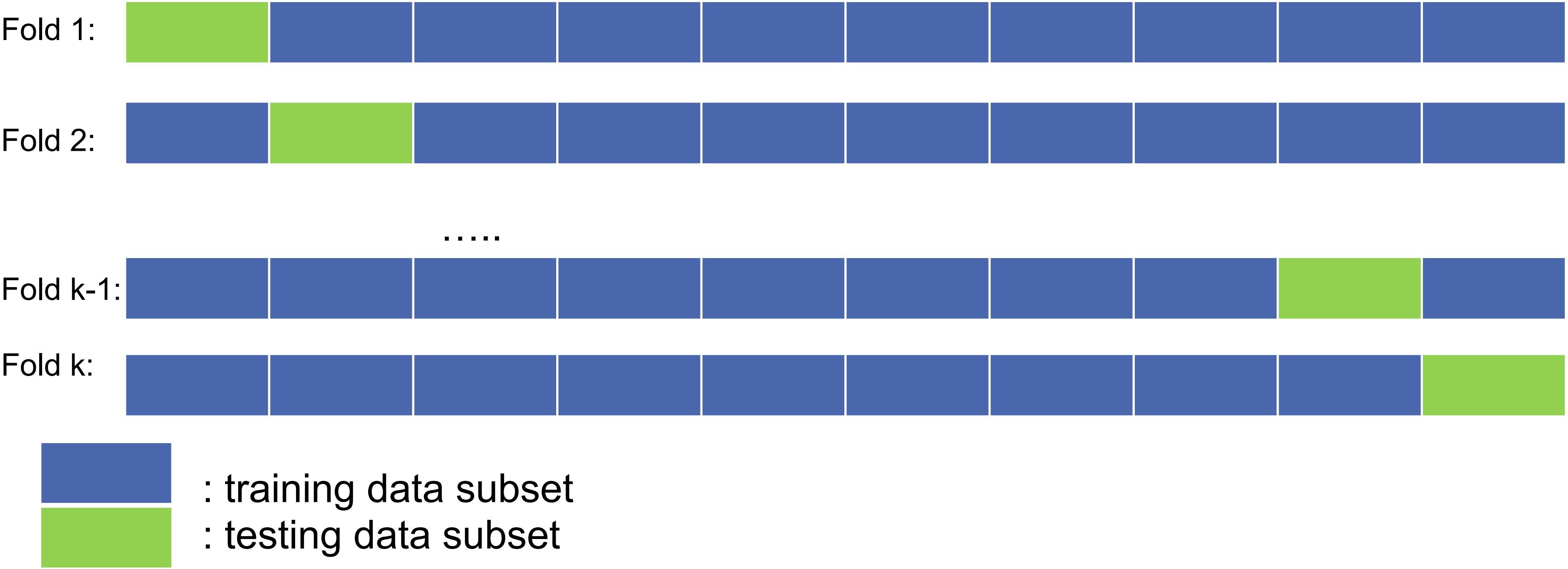
Question

In a segmentation task, what are the most popular performance measures?





k-fold Cross-Validation



In Cross-Validation, the data set is divided into k folds and at each iteration, a different fold is reserved for testing and all the others, used for training the classifiers.

Projects

- Start discussing your project ideas with us
- Email us
- Office hours and meetings?

- Human Connectome Project (HCP)
- Alzheimer's Disease Neuroimaging Initiative: ADNI
- Parkinson's Progression Markers Initiative (PPMI)
- OASIS Brains - Open Access Series of Imaging Studies
- ABIDE - Autism Brain Imaging Data Exchange International Neuroimaging Data-sharing Initiative
- Multimodal Brain Tumor Segmentation Challenge 2020 (BraTS)
- Dataset of your interest

Project Proposal

A **2-page document (including references)** with the following sections

1. Problem statement (clear input/output)
2. Motivation (why the problem is important)
3. Prior work
   a) Key challenges of the problem
   b) Challenges taken care of by the prior work, what is remained
4. Contribution (if any)
5. Technical Details
6. Datasets and Performance Metrics
7. Experiments Plan
8. Team Members