AIMLab.

ML IN HEALTHCARE
FACULTY OF BIOMEDICAL ENGINEERING

Contents

1. Introduction

With billions of mobile devices worldwide and the low cost of connected medical sensors, recording and transmitting medical data has become easier than ever. However, this 'wealth' of physiological data has not yet been harnessed to provide actionable clinical information. This is due to the lack of smart algorithms that can exploit the information encrypted within these 'big databases' of biomedical time series and images, take individual variability into account and generalize to different population sample.

Exploiting such data necessitates an in depth understanding of the physiology underlying the biomedical time series and images, the use of advanced digital signal processing and machine learning tools (e.g. deep learning) to recognize and extract characteristic patterns of health function, and the ability to translate these patterns into clinically actionable information for the purpose of **diagnosis, prognosis and treatment**. In particular, the creation of intelligent algorithms combined with existing and novel wearable and biosensors offer an unprecedented opportunity to improve **Human Health** by providing new intelligent patient monitoring systems in the clinical environment and for remote health monitoring.

In this course you will learn about aspects of information processing including data preprocessing, visualization, regression, dimensionality reduction (PCA, ICA), feature selection, classification (LR, SVM, Deep Learning) and their usage for decision support in the context of **biomedical engineering** and with a focus on improving **Human Health**. It will aim to train a new generation of scientists whom can perform research on large steams of data including **genomic data, sensor data and healthcare data.** The course aims to provide an overview of computer tools and machine learning techniques for processing such datasets within the context of healthcare. Each session is structured with two lectures and two hours of tutorial plus an optional third hour of "workshop". During the lectures the necessary theory and intuition will be covered and practical ("hands on") computer based tutorials and assignments will confront you with real world research question dealing with a variety of medical datasets. The lectures are divided in three parts: ML basis, popular classifiers and introduction to deep learning.

2. Course Summary

Course title:	Machine Learning in Healthcare (MLH)
Short title:	ML in Healthcare
Course ref. no.	336546
Number of credits:	3
Number of weeks: - **Weekly lectures** - **Weekly tutorials**	13 2 hours (total 26 hours) 2 hours (total 26 hours) + 1 hour optional (13 hours)
Course assessment:	Four assignments: 25% each. Three computer-based assignments and one combined with theoretical questions.
Capacity:	32 Working station Up to 85 students
Computer requirements:	Six GPU (department cluster). Software: PyCharm, jupyter notebook, Git, Atom. Libraries: Numpy, Panda, Scilearn, Keras.
Lecturer(s):	Joachim Behar (JB), PhD
Teaching assistants:	Moran Davoodi (MD), PhD candidate Yuval Ben Sason (YBS), MSc candidate
Guests Lecturers:	Anne Weill (AW), PhD, Technion-BME Danny Eytan (DE), MD-PhD, Rambam Hospital Uri Shalit (US), PhD, Technion-IE
Teaching objectives:	Students will acquire the following skills: • Python for biomedical data science. • Main classifiers, intuition and mathematical background. • Neural networks and deep learning. • Performance statistics in healthcare. • ML for diagnosis, prognosis and treatment. • Ground truth in medical data science.
Important deadlines:	Assignments: • 03/11-08/12: First assignment. It will be released right after the second tutorial of the second group and will be open for submission for 5 weeks. • 08/12-05/01: Second assignment. • 13-14/01: Time limited assignment (theory + mini project). • 17/01-28/02: Fourth assignment (deep). Submission is up to about one week after end of moed A.

Requirements:

- All students will be expected to keep a digital log book of their code and results on their GitHub account for each tutorial session.
- Label your figures clearly: parameters and units on both axes in a font large enough to be readable, with a legend describing each line and symbol you plot.

3. Syllabus

1.1 Part I: ML Basis

Week	Lecture	Subjects covered
1	#C01 Introduction	- Course objectives and settings - Introduction to ML in healthcare - Supervised, unsupervised and reinforcement learning - ML for diagnosis, prognosis and treatment - Medical data, sources, challenges and regulations
	#C02 ML concepts	- Polynomial curve fitting - Cost function - Under and overfitting - Notations
2	#C03 Data exploration and preprocessing	- Exploratory data analysis - Data visualization - Abnormality detection and handling - Features scaling
	#C04 Linear models for regression	- Intuition - Calculus proof - Probabilistic proof - Sequential learning - Cost function - Gradient descent
3	#C05 Linear models for classification	- Classification versus regression - LR hypothesis representation - LR Cost function - Gradient descent - Multiclass classification - Odds ratio - Linear discriminant analysis
	#C06 Regularization	- Overfitting - Cost function - Regularized linear regression - Regularized logistic regression - Ridge, Lasso regression - Geometrical interpretation
4	#C07 Training a Classifier I	- Evaluating a model (train, validation, test sets) - Model selection, learning curves and error analysis - Bias-variance tradeoff - Cross validation approaches - Stratification - Information leakage - Generalization performance
	#C08 Training a Classifier II	- Performance statistics - Receiver operative curve - Training the final ML model.

Commented [JB1]: Need to add an explanation on gradient descent (at least stochastic). Math oriented.

1.2 Part II: Popular classifiers

Week	Lecture	Subjects covered
5	#C09 Machine Learning in Healthcare	- ML in medical practice, - Opportunities and ethical challenges (DE)
	#C10 Random Forest	- Getting nonlinear (XOR), QDA - Random forest
6	#C11 Support vector machines	- Maximum margin classifiers - Dual representation - Kernel trick - Grid search and random search
	#C12 Support vector machines	- (Continued)
7	#C13 Principal component analysis (Unsupervised Learning)	- Blind source separation - Principal component analysis - Change of basis - Mathematical proof - PCA in machine learning
	#C14 Independent component analysis (Unsupervised Learning)	- Independent component analysis - Statistical independence versus correlation - Whitening - Beyond ICA: t-SNE
8	#C15 K-means and GMM (Unsupervised Learning)	- K-nearest neighbor - Probabilistic data analysis: GMM
	#C16 Feature selection	- Relevance and redundancy - Filters, wrappers and embedded - LASSO, mRMR
9	#C17 Causal inference	- Causal inference (US)
	#C18 Performance computing	- High Performance Computing (AW)
10	#C19 Common challenges in medical ML	- Data anonymization and sharing - Class imbalance - Multi-task classification - Dataset size - Obtaining the ground truth
	#C20 Common challenges in medical ML	- Population sample - Sensing technology - Generalization - External validation - Integration to the medical practice

Commented [JB2]: TODO. Perhaps leave for the second course in order to cover more theory in this course.

1.3 Part III: Neural networks and introduction to deep learning

Week	Lecture	Subjects covered
11	#C21 ANN I: introduction	- Revisiting logistic regression - Introduction to NN - Notations - Representation learning - Forward propagation - Backward propagation - Activation functions - Multiclass classification (softmax)
	#C22 ANN II: training a NN	- Revisiting train-validation-test split - Weight initialization - Optimization algorithms - Revisiting bias-variance tradeoff - Batch normalization
12	#C23 ANN III: hyperparameters tuning	- Grid search - Random search - Bayesian optimization - Vanishing and exploding gradient
	#C24 Deep Learning CNN	- Foundation - Convolution - CNN architecture - Striding, padding, pooling.
13	#C25 Deep Learning CNN	- (Continued)
	#C26 Examples of medical ML @Technion	- Presentation of ongoing research in the lab.

2. Mathematical notations and terminology

Some notations used in this course are adapted from the notations of the Stanford CS230 course. Reference: https://cs230.stanford.edu/files/Notation.pdf

General notations:

(i)	Example number.
m	The number of examples in the dataset.
n_x	Number of features or input samples (input size).
n_y	Number of classes (output size).
$X \in \mathbb{R}^{n_x \times m}$	Input matrix i.e. matrix with input features n_x for all examples m.[1]
$x^{(i)} \in \mathbb{R}^{n_x}$	Column vector of the i^{th} example.
$x_j^{(i)}$	Scalar value of the j^{th} feature for example i^{th}.
$Y \in \mathbb{R}^{n_y \times m}$	Target matrix i.e. matrix with targets n_y for all examples m.
$y^{(i)} \in \mathbb{R}^{n_y}$	Target label for the i^{th} example.
$\hat{y}^{(i)} \in \mathbb{R}^{n_y}$	The predicted output vector from the classifier.
$\underline{y} \in \mathbb{R}^{m}$	A vector of scalar targets for all examples m.
h	The hypothesis function.
f	Target function i.e. the function we aim to learn.
$\hat{h}$	The estimated target function using the hypothesis function h.
J	Cost function i.e. cost function for all m examples.[2]
E	Error i.e. for a single example.
$\mathcal{N}(\mu, \sigma)$	Normal distribution with mean μ and standard deviation σ.
$w \in \mathbb{R}^{n_x}$	Weights vector in linear and logistic regression.

[1] With exception of the lecture on linear regression the $X \in \mathbb{R}^{m \times n_x}$.
[2] The function that we aim to minimize or maximize is called the objective function. As we are minimizing it is often called equivalently the cost function, loss function, or error function. The term “cost function” usually refers to an optimization problem and “loss function” usually refers to parameter estimation.

Notations specific for Neural Networks:

Hyperparameters in NN:

α	Learning rate.
β	Momentum
p	Mini batch size
K	Number of iterations for gradient descent.
$n_h^{[l]}$	Number of hidden units of the l^{th} layer.
L	Number of layers in a neural network.
$g^{[l]}$	Activation function for layer l.
k	Learning rate decay
	Features scaling method
	Other model specific hyperparameters (e.g. convolution kernel width in CNN.)

NN variables:

$W^{[l]} \in \mathbb{R}^{n_h^{[l]} \times n_h^{[l-1]}}$	Weight matrix for layer l.
$w_j^{[l]} \in \mathbb{R}^{n_h^{[l]}}$	Weight vector for j^{th} activation at layer l.
$w_{jk}^{[l]} \in \mathbb{R}$	k^{th} weight coefficient for j^{th} activation at layer l i.e. element of $W^{[l]}$ at (j,k)
$b^{[l]} \in \mathbb{R}^{n_h^{[l]}}$	Bias vector at layer l.
$b_j^{[l]} \in \mathbb{R}$	j^{th} bias activation at layer l.
$a^{[l]} \in \mathbb{R}^{n_h^{[l]}}$	Activation vector at layer l.
$a_j^{[l]} \in \mathbb{R}$	j^{th} activation at layer l.

Terminology

Example	Refers to a set of features describing an observation.
Target	The label we are aiming to learn to predict.
Hypothesis class	A space of possible hypotheses for mapping inputs to outputs.
Hypothesis function	An instance of the hypothesis class that maps inputs to outputs.

Acronyms

SGD	Stochastic gradient descent.
BGD	Batch gradient descent.