Machine Learning Crash Course

- **Example**
  - Detecting spam/phishing emails
    - We use our experience to classify emails as spam or not.

- **ML**
  - Input: classified emails
  - Generate representation vector for each email
    - How do you decide what information to represent?
  - Classification algorithm finds a hyper-plane that separates spams from non-spams
    - Non-linear separations are much harder

- **Logistic Regression**
  - Simplest classification algorithm
  - For 2D vectors, the separation is a line (ax+by+c=0)
    - In general a hyperplane $a1x1 + a2x2 + ... = 0$
  - Start with a=b=0
  - Cost function measures error with cost = 0 or 1
  - Iteratively update a and b
  - Inequations ax+by>0 (resp.<0) perform the classification
    - Sigmoid function gives certainty that classification is correct
    - I.e., $1/(1+e^{-(ax+by+c)})$
      - On the line gives certainty 0.5

- **Deep learning**
  - Neural network with multiple layers
    - Basic NN: input, function and output

- **Number of features**
  - Too few or many features can lead to under- or overfitting
  - Overfitting = works only for your training set

- **Feature normalization**
  - All features should have same scale
    - Eg from 0 to 1
  - Usual approach: subtract mean and divide by SD
    - Assumes normal distribution
    - Alternative: scale to max-min

- **Outliers**
  - Often removed

- **Supervised ML**

- **Unsupervised learning**

- **Neural Networks**
  - Input nodes plus bias node feed into hidden nodes/layers
  - Training
    - Use sigmoid function as "activation function" to combine inputs and weights
      - $1/(1 + e^{-(x0 * theta0 + x1 * theta1 + ...)})$
      - Alternative is $tanh(x) = 2 * sigmoid(2x) - 1$
    - Search for weights that get close to optimum values
    - Use Newton's method?
      - Take derivatives to select new weights and iterate
  - Recurrent NN
    - Good for sequences

Nevena Milojkovic crash course 2018-08-10

Third session