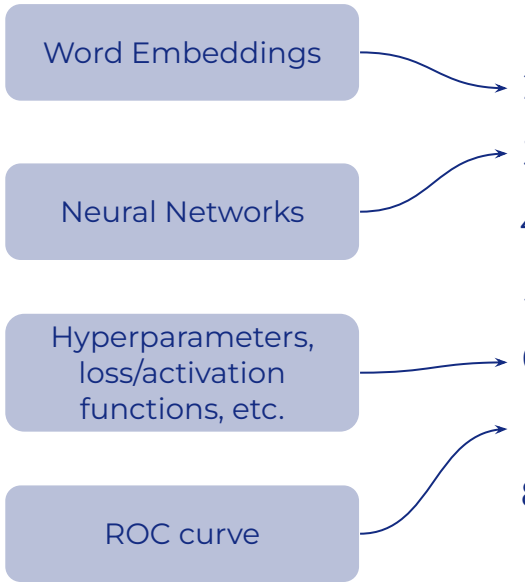
Homework 2 Tutorial

Artificial Intelligence II - Fall 22-23
Dept. of Informatics & Telecommunications
National & Kapodistrian University of Athens

Katerina Xagorari (katexagorari@athenarc.gr)

Based on previous slides by George Katsogiannis et al.

AI Project Structure - Recipe



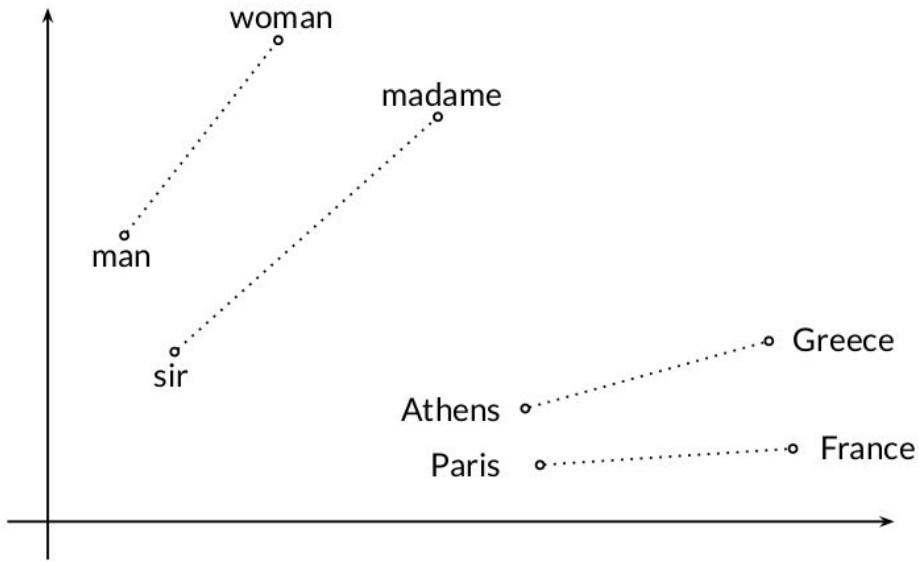
1. Explore the dataset, gain insights
2. Data Pre-processing
3. Create your model(s)
4. Train your model(s) with *training data*
5. Make predictions
6. Tune your model(s) with *validation data*
7. Evaluate your final model with *test data*
8. Present your results (e.g.: accuracy metrics, performance, competitors' models)

Word Embeddings

GloVe: Global Vectors for word representation

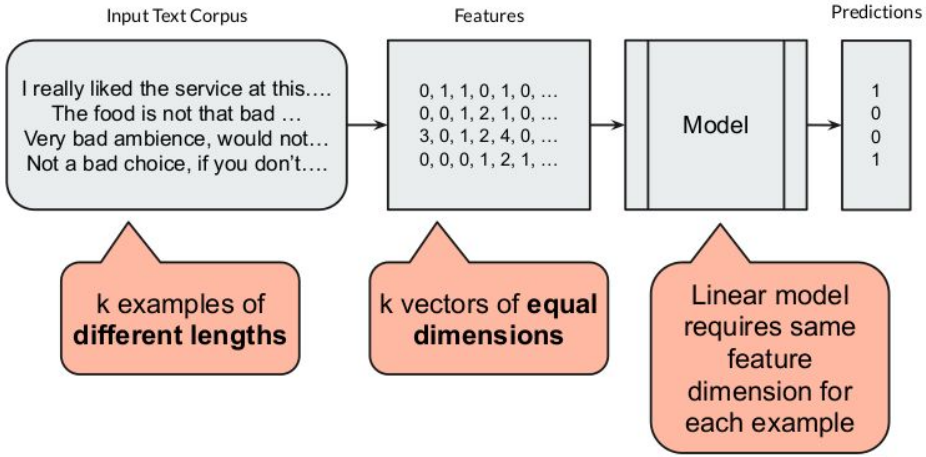
Unsupervised learning (clustering) algorithm developed by Stanford

- Create meaningful vector representations for each word
- Vectors calculated based on word co-occurrence in the training corpus
- Useful linear substructures for word relations
- Words close to each other are semantically similar
- Pre-trained vectors created from large corpuses are available for download



Using Word Embeddings with Linear Networks

- A pre-trained set of embeddings contains a vector of the same dimension for each word (usually 50, 100 or 300)
- How can we create a representation of sentences using word-embeddings, whilst keeping the same dimension for each sentence?
    - Ideas: Averaging, summation, etc.



Deep Learning Frameworks

- TensorFlow
- Keras
- Caffe2
- DL4J
- Microsoft Cognitive Toolkit (CNTK)
- Torch/ PyTorch

Deep Learning Frameworks

TensorFlow

- Open source framework for dataflow and differential programming
- One of the most preferred frameworks for Deep Learning
- Developed by the Google Brain team
- Support for Python, C++, R
- Known to be complicated
- Excellent documentation and community support

Keras

- An easy-to-use library that provides interfaces for:
  - DL layers
  - Activation functions, loss functions
  - Optimisers, and many more
- Easy to test an idea quickly with little code
- Uses TensorFlow as its backend
- Integrated in TensorFlow V2

Deep Learning Frameworks

Merged with PyTorch

- Deep Learning Framework, widely used for Computer Vision Tasks
- Support for C, C++, Python, MATLAB and various CLI
- It is considered to be one of the fastest DL Frameworks
- Big advantage is the Caffe's Model Zoo that contains easy to use pre-trained models
- Developed by Berkley

- Deep Learning for Java and all the JVM based programming languages (i.e. Scala, Kotlin, Closure, etc)
- Support Keras models loaded as **.h5** objects
- Supports distributed training using Apache Spark
- It's considered one of the fastest, alongside with Caffe2

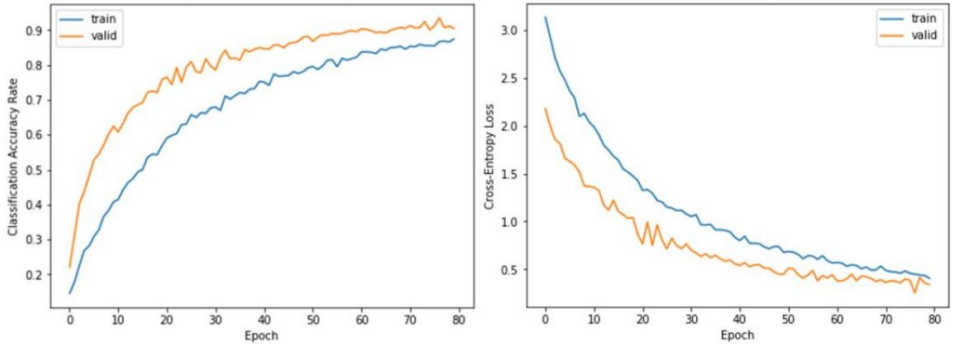
Deep Learning Frameworks

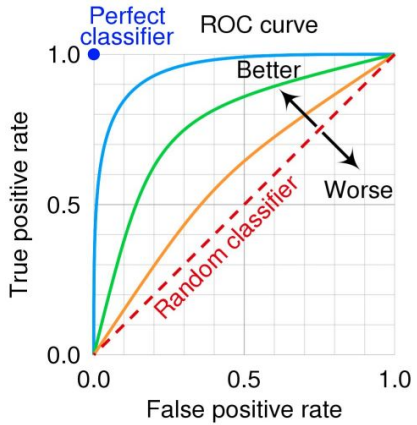
- One of the most popular in Industry and Research
- Supports Python and C++
- Easier to learn than TensorFlow
- More educative than Keras, as provides better understanding for what happens:
  - during the forward pass
  - during backward pass
  - usage of activation functions, loss functions, optimizers, etc
- Developed by Meta AI/Linux

- PyTorch and TensorFlow Comparison
- PyTorch Documentation Tutorials:
  - Building a Neural Network
  - How Automatic Differentiation works
  - Optimising (Training) a PyTorch Model ❗

Evaluating your model

- All the same things apply, as in HW1
- *Note:* Learning curves are plotted with the number of epochs/steps in the x-axis
- Plot a ROC curve (used to evaluate classifier output quality)





Demo Notebooks

Notebook 1: PyTorch Intro

Introduction to Tensors

- Torch Tensors
- Creating/Reshaping Tensors
- Operations with Tensors
- Converting Tensors to NumPy arrays
- Automatic Differentiation for tensor operations
- Examples on computing gradients with PyTorch

Notebook 2: PyTorch Example

Training a NN model with PyTorch

- GloVe Intro & Examples
- Loading & Pre-processing your data:
  - handling NaN values
  - normalising values
  - separating features and targets
  - saving values to tensors
- Creating a model
- Training your model