Homework 2 Tutorial

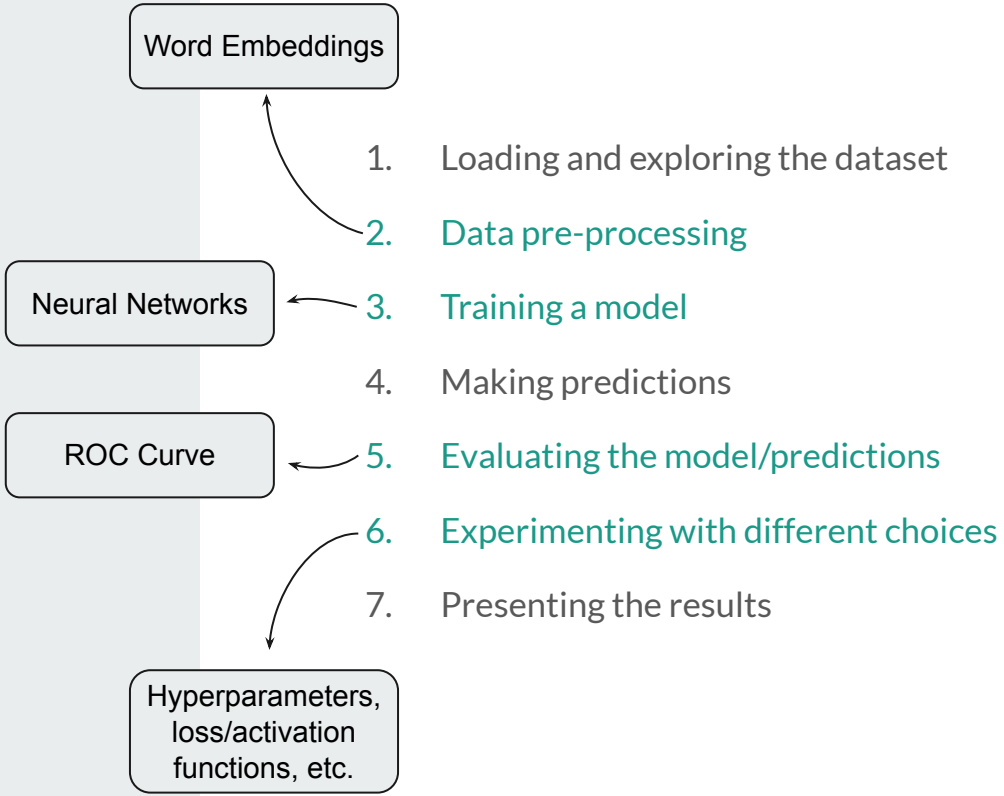
Artificial Intelligence II - Fall 21-22
Dept. of Informatics & Telecommunications
National & Kapodistrian University of Athens

George Katsogiannis (katso@athenarc.gr)

Based on previous slides by George Mandilaras

Structuring a NLP Project

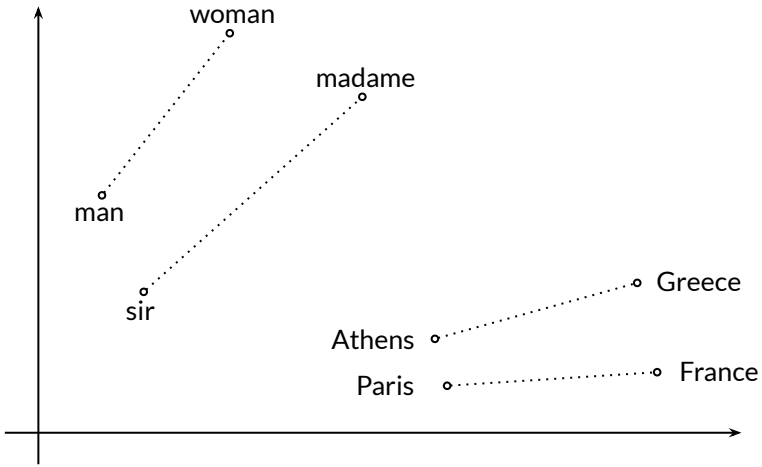
What is new for this HW?



1. Loading and exploring the dataset
2. Data pre-processing
3. Training a model
4. Making predictions
5. Evaluating the model/predictions
6. Experimenting with different choices
7. Presenting the results

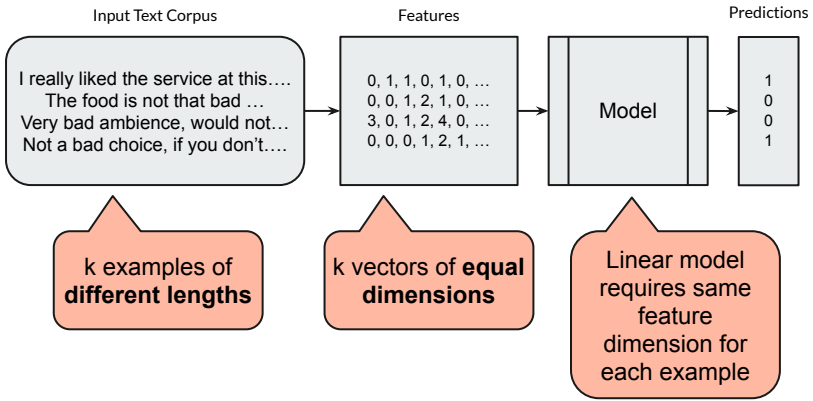
Word Embeddings: GloVe

- 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 vectors of the same dimension for each word (usually 50, 100 or 300)
- How can we create a representations of sentences using word embeddings, keeping the same dimension for each sentence?
    - Ideas: Averaging, summation, etc.



Deep Learning Frameworks

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

Deep Learning Frameworks

Caffe2:

- 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

Theano:

- It is a scientific computing library
- It was developed by the Université de Montréal in 2007
- Python
- Good integration with NumPy and SciPy
- Does not support execution in multiple GPUs
- The developers have announced that they will no longer actively maintain or develop Theano any more

Deep Learning Frameworks

DL4J:

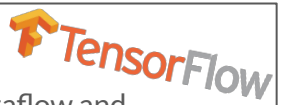
- Deep Learning for Java and all the JVM based programming languages (i.e. Scala, Kotlin, Closure, etc)
- Support Keras models loaded as .h5 objects
- Supports distributed training using Apache Spark
- It's considered one of the fastest, alongside with Caffe2

Microsoft Cognitive Toolkit (CNTK):

- Support for Python, C# and C++
- Facilitates the implementation of complex models like GANs
- Good performance and Scalability

Deep Learning Frameworks

Tensorflow:



- Open source framework for dataflow and differential programming
- One of the most prefered 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
  - Optimizers, and many more
- Easy to test an idea quickly with little code
- Uses TensorFlow or Theano as its backend
- Integrated in TensorFlow in TensorFlow V2

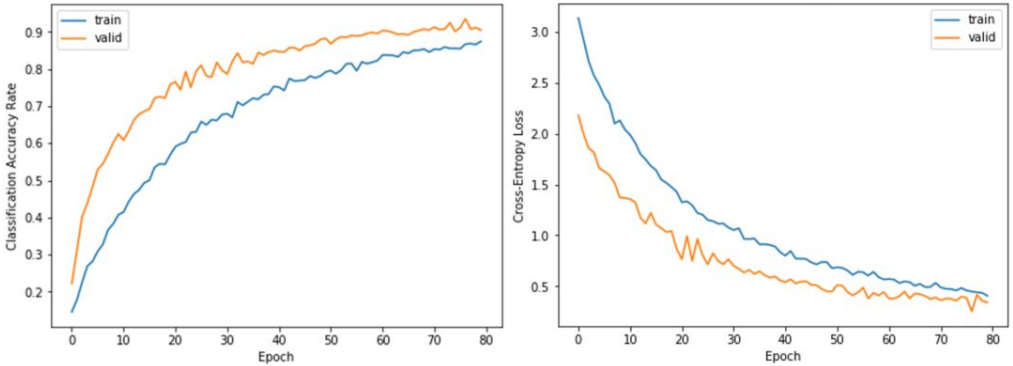
Deep Learning Frameworks

Pytorch:

- One of the most popular in Industry and Research
- Python
- Easier to learn than TensorFlow
- More educative than Keras, as provides better understanding for what happens
  - during the forward pass
  - during backward pass
  - when usaging of activation functions, loss functions, optimizers, etc

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

Evaluating the model/predictions



- All the same things apply, as in HW1
- Note: Learning curves are plotted with the number of epochs/steps in the x axis
- New: You must also plot a ROC curve
  - More about ROC/AUC