

Deep Learning

Übung WS 23/24

Judith Nester (nester@uni-koeln.de)

16-11-2023

Recap

» List Comprehension

- Define lists by specifying a pattern

- `[x*2 for x in l1 if x < 10]`

» Functions

- Weakly typed arguments
- Named arguments, default values
- Return values, None and NoneType

» Input/Output

- Stream-oriented
- Open file, work with stream, close file

Today

Exception Handling

Python Packages

Types of tasks

Classification

Exercise

Section 1

Exception Handling

Introduction

» Program errors

- Type errors: Variables have the wrong type (i.e., we try to multiply two strings or add a string and an integer)
- IO issues: Disk full, file/directory doesn't exist
- Parse errors: Input isn't how we expect it to be
- Users errors: A user enters something unexpected
- ...

Introduction

- » Program errors
 - Type errors: Variables have the wrong type (i.e., we try to multiply two strings or add a string and an integer)
 - IO issues: Disk full, file/directory doesn't exist
 - Parse errors: Input isn't how we expect it to be
 - Users errors: A user enters something unexpected
 - ...
- » Error prevention: Verify before doing
 - Some errors cannot be anticipated
- » Exception handling: Do, and specify behavior if error occurs

Exception Handling

- » Anytime, code can ›throw an exception‹ (Python: `raise`)
- » The exception can be caught by other code
- » Documentation: <https://docs.python.org/3/tutorial/errors.html>

Exception Handling

- » Anytime, code can ›throw an exception‹ (Python: raise)
- » The exception can be caught by other code
- » Documentation: <https://docs.python.org/3/tutorial/errors.html>

Example

```
1 l1 = [1,2,3,4,5,6]
2
3 try:
4     userInput = input()
5     intInput = int(userInput) # may raise a TypeError
6     print(l1[intInput])      # may raise an IndexError
7 except TypeError:
8     print("Please enter an integer number!")
9 except IndexError:
10    print("Doesn't exist")
```


Built-In Errors

- » A number of error types are built into the standard library
- » They are also used by library functions
- » <https://docs.python.org/3/library/exceptions.html#Exception>

Built-In Errors

- » A number of error types are built into the standard library
- » They are also used by library functions
- » <https://docs.python.org/3/library/exceptions.html#Exception>
- » Feel free to use them!

Define Your Own Exceptions

- » Exceptions are classes that inherit from `BaseException`
- » Own exceptions should be derived from `Exception`

```
1 class MyException(Exception):  
2     pass  
3  
4 if a > len(l):  
5     raise MyException
```

Exception Payload

It's possible to pass information through exceptions (e.g., the file name that doesn't exist)

Exception Payload

It's possible to pass information through exceptions (e.g., the file name that doesn't exist)

```
1 def openAndRead(filename):
2     fo = open(filename)
3     content = ""
4     for l in fo.readlines():
5         content.append(l)
6     fo.close()
7     return content
8
9 try:
10     content = openAndRead("filename.txt")
11 except OSError as err:
12     print("File " + err.filename + " does not exist.")
```

Section 2

Python Packages

Python Packages

- » Central repository for python libraries: <https://pypi.org>
- » Installation and local management: `pip`
 - Contained in python installation from python.org
 - Run on command line: `pip CMD/pip3 CMD/py -m pip CMD`
 - Documentation: <https://pip.pypa.io/en/stable/>

Python Packages

- » Central repository for python libraries: `https://pypi.org`
- » Installation and local management: `pip`
 - Contained in python installation from `python.org`
 - Run on command line: `pip CMD/pip3 CMD/py -m pip CMD`
 - Documentation: `https://pip.pypa.io/en/stable/`
- » CMD: Various commands
 - Most important: `install LIBRARY`

Python Packages

- » Central repository for python libraries: <https://pypi.org>
- » Installation and local management: `pip`
 - Contained in python installation from python.org
 - Run on command line: `pip CMD/pip3 CMD/py -m pip CMD`
 - Documentation: <https://pip.pypa.io/en/stable/>
- » CMD: Various commands
 - Most important: `install LIBRARY`

Example

```
1 pip install numpy
```

Section 3

Types of tasks

Natural Language Processing

- » Natural Language Processing (NLP) can be located between linguistics, computer science, data science, but also psychology and neuroscience
 - Our focus is of course on the Deep Learning part of NLP
- » There are many different tasks that are handled with NLP
 - Question Answering, Summerization, Sentiment Analysis, ...
 - Good overview of the current state of NLP research: <http://nlpprogress.com/>
 - Take it with a grain of salt, I'm not sure how often this repo is updated

Task types

- » Many ML/DL/NLP tasks are structurally similar
- » Structurally similar: The same system can be used, all differences can be encoded in the training data

Task types

- » Many ML/DL/NLP tasks are structurally similar
- » Structurally similar: The same system can be used, all differences can be encoded in the training data

Example

- » Part of speech tagging: Each token gets a label
 - Labels: NN, VBZ, DET, ADJA, ADJD, ...
- » Named entity recognition: Each token gets a label
 - O, B-PER, I-PER, B-LOC, I-LOC, ...

Task types

- » Many ML/DL/NLP tasks are structurally similar
- » Structurally similar: The same system can be used, all differences can be encoded in the training data

Example

- » Part of speech tagging: Each token gets a label
 - Labels: NN, VBZ, DET, ADJA, ADJD, ...
 - » Named entity recognition: Each token gets a label
 - O, B-PER, I-PER, B-LOC, I-LOC, ...
-
- » Two important task types for NLP
 - Text classification: An entire text is classified (e.g., genre, sentiment, ...)
 - Sequence labeling: Each individual word is classified (e.g., pos-tagging, ...)

Task types

Text classification

- » Texts belong to a class of texts

Task types

Text classification

- » Texts belong to a class of texts

Example

- » Customer reviews → sentiment
- » Novel → genre (fiction, non-fiction, ...)
- » Posting → \pm hate speech
- » E-mail → spam, not spam, really important

Task types

Sequence labeling

- » Words (or sequences of words) belong to classes
 - Sequence labeling: Classification + sequential dependency between classes

Task types

Sequence labeling

- » Words (or sequences of words) belong to classes
 - Sequence labeling: Classification + sequential dependency between classes

Example

- » Words → part of speech (noun, verb, adjective, ...)
- » Words → proper noun
- » Paragraphs → \pm narrative scene

Section 4

Classification

Introduction

- » Assigning *classes* to *objects/instances/items*
 - Words → parts of speech
 - Texts → genres
 - Sentences → polarity (positive/negative)

Introduction

- » Assigning *classes* to *objects/instances/items*
 - Words → parts of speech
 - Texts → genres
 - Sentences → polarity (positive/negative)
- » Many different models/algorithms available (all with variants):
 - Decision trees
 - Support vector machines
 - Naïve Bayes
 - Bayesian networks
 - ...

Introduction

- » Assigning *classes* to *objects/instances/items*
 - Words → parts of speech
 - Texts → genres
 - Sentences → polarity (positive/negative)
- » Many different models/algorithms available (all with variants):
 - Decision trees
 - Support vector machines
 - Naïve Bayes
 - Bayesian networks
 - ...
 - (Artificial) neural networks (a.k.a. »deep learning«)

Classification

Target classes

Classes: A finite set of categories

Examples

- » Parts of speech: Noun, verb, adjective, ...
 - E.g., STTS tagset
- » Genres: Abenteuerroman, Bildungsroman, Kriminalroman, ...
 - Many novels fall in multiple classes

Classification

Target classes

Classes: A finite set of categories

Examples

- » Parts of speech: Noun, verb, adjective, ...
 - E.g., STTS tagset
- » Genres: Abenteuerroman, Bildungsroman, Kriminalroman, ...
 - Many novels fall in multiple classes

Important first step: Clearly identify classes and problem properties

Two Parts

Prediction Model

How do we make predictions on data instances?
(e.g., how do we assign a part of speech tag for a word?)

Learning Algorithm

How do we create a prediction model, given annotated data?
(e.g. how do we create rules for assigning a part of speech tag for a word?)

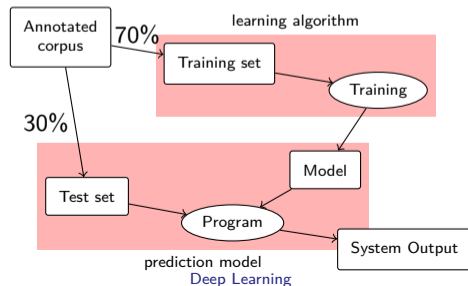
Two Parts

Prediction Model

How do we make predictions on data instances?
(e.g., how do we assign a part of speech tag for a word?)

Learning Algorithm

How do we create a prediction model, given annotated data?
(e.g. how do we create rules for assigning a part of speech tag for a word?)



Section 5

Exercise

Exercise 04

<https://github.com/IDH-Cologne-Deep-Learning-Uebung/exercise-04>