

Основне академске студије  
Информациони инжењеринг

Основи рачунарске интелигенције

# Библиотека Keras за језик Python

(материјали за вежбе)

1. Увод
2. Класификација
3. Конструкција неуронске мреже
4. Обучавање и евалуација модела

## Библиотека *Keras*

посвећена прављењу неуронских мрежа  
развијена с фокусом на брзо експериментисање  
поседује *API* ка другим библиотекама

*TensorFlow*, *Theano* (до верзије 2.3), *CNTK* (до верзије 2.3)...  
могуће коришћење *CPU*, *GPU* и *TPU*

## Библиотека *Keras*

### ОСНОВНИ ПОДАЦИ

развијена је као део истраживања на пројекту *ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System)*

водећи аутор Франсоа Шоле

настанак 2015. године

слободна за употребу

актуелна верзија 2.4.0

Интернет сајт

<https://keras.io/>

1. Увод
- 2. Класификација**
3. Конструкција неуронске мреже
4. Обучавање и евалуација модела

## Обрада података

учитавање података

обрада недостајућих вредности

у зависности од присуства недостајућих вредности

обрада излазних вредности

замена скаларне вредности циљног категоријског обележја вектором с јединицом на одговарајућој позицији и нулама на осталим позицијама (енгл. *one-hot encoding*)

може бити погодно када има више од две вредности циљног категоријског обележја

пример

могуће скаларне вредности циљног обележја

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

одговарајући вектор за скаларну вредност 9

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

подела података

скуп за обучавање

скуп за тестирање

## Подаци

скуп података о резултатима хемијских анализа вина узгајаних у истој регији у Италији добијених од три различите сорте

постоји 178 примерака у скупу и 14 обележја (класа, алкохол, јабучна киселина, пепео, алкалност пепела, магнезијум, укупни феноли, флаваноиди, нефлаваноидни феноли, проантоцијанини, интезитет боје, нијанса, *OD280/OD315* разблажених вина, пролин)

извор скупа података

скуп података *Wine Data Set*

матична Интернет страна

<https://archive.ics.uci.edu/ml/datasets/Wine>

подаци доступни у *CSV* формату

скуп података је креиран 1. 7. 1991. године

за власника скупа података су назначени *Forina, M. et al, PARVUS*

за донатора скупа података је назначен *Stefan Aeberhard stefan@coral.cs.jcu.edu.au*

1. Увод
2. Класификација
- 3. Конструкција неуронске мреже**
4. Обучавање и евалуација модела

# Конструкција неуронске мреже

## Главне структуре података у библиотеци *Keras*

модел (енгл. *model*)

структура изграђена од повезаних слојева

„усмерени ациклични граф слојева”

слој (енгл. *layer*)

главна структура за трансформацију вредности

подршка за разноврсне слојеве

густо повезани слој

конволуциони слојеви

обједињавајући слојеви

рекурентни слојеви

...

## Модел

постоје три начина да се креира модел помоћу библиотеке *Keras*

### *Sequential model*

једноставан, листа слојева

### *Functional API*

једноставан

функционалан API који подржава произвољне архитектуре модела  
препоручује се за употребу

### *Model subclassing*

имплементација од нуле препуштена кориснику

препоручује се за употребу уколико постоји потреба за комплексним случајем  
коришћења (најчешће за истраживања)

# Конструкција неуронске мреже

## Слој – густо повезани слој (енгл. *dense layer*)

### основни тип слоја

#### параметри

матрица тежина (енгл. *kernel*)

вектор прагова (енгл. *bias*)

### имплементација помоћу класе *Dense*

#### главна подешавања при формирању слоја

*units* – димензије слоја (димензије излаза)

*activation* – функција активације

#### примери

*elu, relu, selu, sigmoid, softmax, tanh...*

*use\_bias* – употреба прага

*input\_shape* – димензије улаза (у случају првог слоја)

# Конструкција неуронске мреже

## Додатна конфигурирања

### критеријумска функција (енгл. *loss function*)

рачуна вредност коју треба минимизовати током обучавања

*categorical\_crossentropy, binary\_crossentropy, sparse\_categorical\_crossentropy, mean\_squared\_error, mean\_absolute\_error...*

### оптимизатор

алгоритам на основу ког се подешавају хиперпараметри

*SGD, RMSprop, Adam, Adadelta, Adagrad...*

### метрика

функција која се користи за оцену перформанси модела

слична критеријумској функцији, али се не користи приликом обучавања

*accuracy, categorical\_cross\_entropy, mean\_squared\_error, root\_mean\_squared\_error, AUC, precision, recall...*

# Конструкција неуронске мреже

## Пример конструкције неуронске мреже

```
nm = Sequential()  
nm.add(Dense(6, activation="tanh",  
            kernel_initializer="zeros",  
            input_shape=(ulaz_dimenzija,)))  
...  
nm.compile(loss="binary_crossentropy",  
           optimizer="adam",  
           metrics=["binary_accuracy"])
```

1. Увод
2. Класификација
3. Конструкција неуронске мреже
- 4. Обучавање и евалуација модела**

## Поступак обучавања модела

главна подешавања

подаци за коришћење у обучавању

величину беча

*batch\_size*

број епоха

*epochs*

пример

```
nm.fit(podaci_obuka_ulazi,  
        podaci_obuka_oznake,  
        batch_size=b,  
        epochs=e)
```

# Обучавање и евалуација модела

## Поступак евалуације модела

главна подешавања

подаци за коришћење у евалуацији

пример

```
ev = nm.evaluate(podaci_test_ulazi,  
                 podaci_test_oznake,  
                 verbose=0)
```

## Поступак примене модела

пример

```
izlazi_test = nm.predict(podaci_test_ulazi)
```

## Задатак 1.

Проширити скуп података тако да буде уведено ново циљно обележје о сортама код којег постоје само две могуће класе.

Формирати неуронску мрежу која обухвата барем два густо повезана слоја.

Обучити и евалуирати неуронску мрежу за ново циљно обележје.

Испитати какве излазе генерише неуронска мрежа.

Експериментисати с различитим подешавањима неуронске мреже и поступка обучавања.

## Задатак 2.

Као циљно обележје искористити полазно обележје о сортама код којег постоје три могуће класе, уз евентуалне пратеће трансформације.

Формирати неуронску мрежу која обухвата барем два густо повезана слоја.

Обучити и евалуирати неуронску мрежу.

Испитати какве излазе генерише неуронска мрежа.

Експериментисати с различитим подешавањима неуронске мреже и поступка обучавања.

## Основна литература

Keras. Keras: The Python deep learning API. Internet: <https://keras.io/>

Keras. Getting started. Internet: [https://keras.io/getting\\_started/](https://keras.io/getting_started/)

Keras. Keras API reference. Internet: <https://keras.io/api/>