

Predmet: “Metode i primjena vještačke inteligencije”

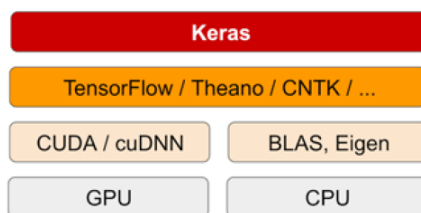
Laboratorijska vježba 1: Osnove rada sa Keras API-jem

1 Cilj vježbe

Upoznavanje sa osnovnim informacijama u vezi sa instalacijom i korištenjem *Keras* API-ja. Predstavljanje “Hello World” primjera iz oblasti mašinskog učenja. U vježbi će biti korišteni neki termini koji nisu *a priori* poznati, ali će biti objašnjeni u nastavku predavanja i vježbi.

2 Uvod

Keras¹ je API na visokom nivou apstrakcije namijenjen za brzo i efikasno razvijanje i eksperimentiranje sa neuronskim mrežama. Napisan u *Python*-u i može se pokretati preko *TensorFlow*-a, *CNTK*-a ili *Theano*-a. Trenutno aktuelna verzija Keras-a je 2.3.0 i podržava *TensorFlow* 2.0.



2.1 Instalacija *TensorFlow*-a

Prije korištenja *Keras*-a potrebno je odlučiti koji od ponuđenih backend-a će se koristiti: *TensorFlow*, *CNTK* ili *Theano*. S obzirom da korištenu opremu, na ovom kursu će se koristiti *TensorFlow*².

Za instalaciju *TensorFlow*-a potrebno je pokrenuti naredbe navedene ispod. U slučaju da nemate specijalnu GPU karticu koja može izvršavati tensor operacije (a vjerovatno nemate), koristit ćete *TensorFlow* za CPU (tada zanemarite zadnju naredbu).

```
# Requires the latest pip
pip install --upgrade pip
```

```
# Current stable release for CPU-only
pip install tensorflow
```

```
# (Optional) Or preview build for CPU/GPU (unstable)
pip install tf-nightly
```

2.2 Instalacija *Keras*-a

Zatim je potrebno instalirati *Keras* API. Keras će po *default*-u koristiti *TensorFlow* biblioteku za manipuliranje tensorima. Za podešavanje *backend*-a pogledajte stranicu: <https://keras.io/backend/>.

Naredbe navedene ispod koriste se za Linux i Mac okruženje. U slučaju da koristite Windows okruženje pokrenite naredbe bez “sudo”.

```
# Instalacija preko PyPI
sudo pip install keras
```

Za instalaciju iz izvornog koda možete koristiti i GitHub (opcionalno):

¹Keras (κέρας) means horn in Greek. It is a reference to a literary image from ancient Greek and Latin literature, first found in the *Odyssey*, where dream spirits (Oneiroi, singular Oneiros) are divided between those who deceive men with false visions, who arrive to Earth through a gate of ivory, and those who announce a future that will come to pass, who arrive through a gate of horn. It's a play on the words κέρας (horn) / ὑφαίνω (fulfill), and ἰβήφας (ivory) / ἐλεφαίρειον (deceive).

²<https://www.tensorflow.org/install/>

```
git clone https://github.com/keras-team/keras.git
cd keras
sudo python setup.py install
```

3 Primjer: Klasifikacija rukom pisanih cifri

U prvom primjeru klasificirati ćemo skup slika dimenzija 28×28 , koje su snimljene u nijansama sive boje (grayscale). Broj kategorija je unaprijed određen, jer znamo da postoji 10 cifara (od 0 do 9). Koristit će se MNIST³ skup podataka (dataset). U dataset-u postoji 60,000 slika za treniranje i 10,000 slika za testiranje.

3.1 Učitavanje podataka

Pošto je ovaj skup podataka već pripremljen u *Keras*-u i kako još uvijek niste radili tehnike za pripremanje podataka za rad sa metoda mašinskog učenja i neuronskim mrežama, za prvu vježbu koristiti će se već pripremljen dataset iz Keras API-ja. U nekoj od sljedećih vježbi naučiti ćete tehnike za pripremu podataka prije njihovog korištenja u neuronskoj mreži.

Podaci se učitavaju na sljedeći način:

```
import keras
from keras.datasets import mnist
```

```
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

Model će se trenirati na osnovu podataka učitanih u varijable `train_images` i `train_labels`. A zatim će se model testirati na osnovu podataka učitanih u varijable `test_images` i `test_labels`. Slike su kodirane kao Numpy⁴ nizovi, dok su labele definisane kao niz cifara i predstavljane su vrijednostima od 0 do 9. Svakoј slici dodjeljena je jedna labela.

Sljedeće naredbe daju neke informacije o podacima za treniranje i testiranje. Pokrenite ih i samostalno zaključite o kojim informacijama je riječ.

```
train_images.shape
len(train_labels)
train_labels
```

Odgovorite na sljedeća pitanja:

1. Koliko slika ima u skupu za treniranje?
2. Koja je rezolucija slika?
3. Kojeg tipa su osnovne informacije unutar slika? i koliko bita zauzimaju u memoriji?
4. Koliko prostora će čitav skup zauzeti u memoriji?
5. Odgovorite na ista pitanja za testni skup podataka.

Kada su podaci spremni za treniranje, sljedeći korak je definisanje i izgradnja modela neuronske mreže.

3.2 Definisanje neuronske mreže

Zadata arhitektura neuronske mreže treba da nauči vezu između slika i labela. Neuronska mreža zatim treba da donese neke zaključke o testnom skupu slika. Obzirom da su labele poznate za testni skup podataka, moguće je donijeti neki zaključak o tačnosti predviđanja neuronske mreže. Na već postojeći kod potrebno je učitati najjednostavniju arhitekturu mreže. Vodite računa da neke naredbe dodajete u zaglavlje datoteke, a druge ispod već postojećeg koda.

```
from keras import models
from keras import layers
```

```
network = models.Sequential()
network.add(layers.Dense(512, activation='relu', input_shape=(28 * 28,)))
network.add(layers.Dense(10, activation='softmax'))
```

³NIST je skraćena od *National Institute of Standards and Technology*

⁴<https://numpy.org/>

Za ovaj primjer koristit će se `Sequential()` model⁵. Osnovna komponenta svake neuronske mreže su nivoi (*layers*), koji se mogu posmatrati kao “filteri” podataka. Zadatak nivoa je da ekstrahuje reprezentaciju podataka koje dobije na ulazu u mrežu. Današnje deep learning mreže se obično sastoje od niza nivoa, a zadatak nivoa je da “destiluju podatke” koje dobiju na ulazu.

U datom primjeru, mreža se sastoji od dva `Dense()` nivoa, koji su potpuno konektovani (*fully connected*). Zadnji nivo je “softmax” nivo koji će kao izlaz dati 10 vrijednosti koje predstavljaju vjerovatnoće da je riječ o nekoj od 10 cifri.

Svaka mreža dodatno mora imati još tri dijela, koji su dio “koraka kompajliranja”:

1. Funkcija gubitka (*loss function*): omogućava mjerenje kvalitete izlaznih rezultata koji se kasnije mogu koristiti za poboljšavanje određenih parametara u neuronskoj mreži.
2. Optimizator: mehanizam putem kojeg će se koeficijenti mreže ažurirati na osnovu ulaznih podataka i funkcije gubitka.
3. Metrika za praćenje u toku treninga i testiranja: postoje razne metrike pomoću kojih možemo pratiti rezultate mreže. Za konkretan primjer koristit će se tačnost, a kasnije će se koristiti i druge metrike.

Osnovne funkcije ovih dijelova su detaljno objašnjene na predavanjima. U ovom primjeru koristit ćete predefinisane vrijednosti, a u samostalno vrijeme možete isprobati druge parametre.

```
network.compile(optimizer='rmsprop',
                 loss='categorical_crossentropy',
                 metrics=['accuracy'])
```

Prije treniranja potrebno je promijeniti oblik podataka u oblik koji mreža očekuje, a to su vrijednosti u intervalu od $[0, 1]$. To će se postići sljedećim kodom:

```
train_images = train_images.reshape((60000, 28 * 28))
train_images = train_images.astype('float32') / 255

test_images = test_images.reshape((10000, 28 * 28))
test_images = test_images.astype('float32') / 255
```

Odgovorite na sljedeća pitanja:

1. Šta radi naredba `reshape`? A šta `astype`?
2. Koje vrijednosti su sada snimljene u `train_images` i `test_images`?
3. Koliko bita sada zauzima jedan pixel u memoriji?

Pošto su podaci pripremljeni, ostalo je još da se pripreme labelle.⁶

```
from keras.utils import to_categorical

train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
```

3.3 Treniranje mreže

Mreža se trenira pozivanjem funkcije `fit` nad mrežom koju smo definisali.

```
network.fit(train_images, train_labels, epochs=5, batch_size=128)
```

Odgovorite na sljedeća pitanja:

1. Šta je epoha?
2. Šta određuje parametar `batch_size`?
3. Koje parametre vidite u konzoli u toku procesa treniranja?
4. Koja je tačnost i vrijednost funkcije gubitka?

⁵Više informacija možete naći na stranici <https://keras.io/getting-started/sequential-model-guide/>

⁶Ovaj dio će biti objašnjen na predavanjima kada za to dođe vrijeme.

3.4 Evaluacija mreže

Kada je mreža istrenirana sa skupom podataka, potrebno je izvršiti evaluaciju mreže nad testnim podacima.

```
test_loss , test_acc = network.evaluate(test_images , test_labels)
```

Odgovorite na sljedeća pitanja:

1. Koja je tačnost i vrijednost funkcije gubitka u skupu testnih podataka?

4 Samostalni zadatak

Uzmite papir i olovku i napišite 10 proizvoljnih brojeva, a zatim to isto ponovite na način da zamolite još dvije osobe da isto to učine. Slikajte ili skenirajte taj papir, obradite ga sa nekim alatom za slike i napravite svoj mali skup podataka (svaka cifra predstavljena sa 28×28 piksela) koji ćete unijeti u postojeći program i provjeriti rad mreže. Koja je tačnost nad datim skupom podataka?

Samostalni zadatak nije obavezan, ali predstavlja interesantan eksperiment koji testira vaše poznavanje obrađenog primjera i Python-a.