

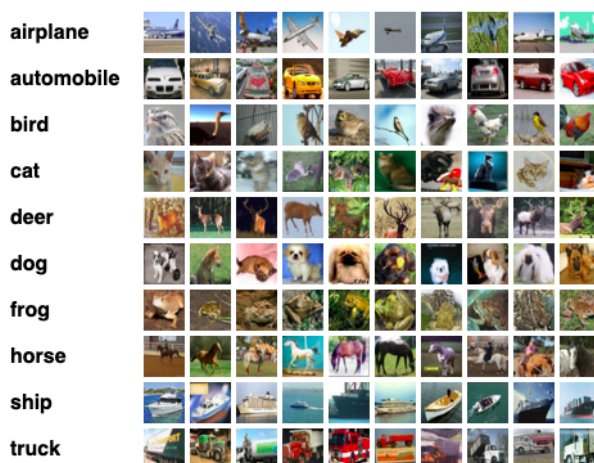


1 Cilj vježbe

Upoznavanje sa principima rada dva klasifikatora iz oblasti mašinskog učenja. Na predavanjima ste se upoznali sa dvije metode za klasifikaciju: 1) SVM i 2) Softmax. U vježbi je potrebno upoznati se sa implementacijom ovih metoda i njihovom primjenom na CIFAR10¹ dataset-u. Zatim je potrebno izvršiti niz testova u svrhu donošenja nekih bitnih zaključaka u vezi karakteristika ovih klasifikatora.

2 Dataset: CIFAR10

CIFAR-10 se sastoji od 60000 slika u boji. Rezolucija svake slike je 32x32, i slike su klasificirane u 10 klasa. Svaka klasa ima 6000 slika. Dataset je podijeljen na 50000 slika za treniranje i 10000 slika za testiranje. Sve klase su potpuno međusobno isključive (nema preklapanja među klasama). Primjeri nasumično izabranih slika prikazani su na slici 1.



Slika 1: Prikaz 10 klasa iz CIFAR10 *dataset-a*, kao i nekoliko primjera nasumično izabranih slika iz *dataset-a*.

2.1 Učitavanje dataset-a

Dataset je integralni dio Keras biblioteke, pa će se za potrebe ove vježbe dataset samo učitati. Neophodne biblioteke za ovaj dataset su: Numpy, scipy, Keras, TensorFlow i matplotlib.

```
import os
import time
import numpy as np

# Library for plotting the output and saving it to a file
import matplotlib as mpl
mpl.use('Agg')
import matplotlib.pyplot as plt

# Load the CIFAR10 dataset
from keras.datasets import cifar10
baseDir = os.path.dirname(os.path.abspath('__file__')) + '/'
```

¹Autori ovog dataset-a su Alex Krizhevsky, Vinod Nair i Geoffrey Hinton. Više informacija možete naći na <https://www.cs.toronto.edu/~kriz/cifar.html>.

```
classNames = ['plane', 'car', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship',
              , 'truck']

(xTrain, yTrain), (xTest, yTest) = cifar10.load_data()
```

Odgovorite na sljedeća pitanja:

1. Koja naredba se koristi za utvrđivanje oblika strukture xTrain? Ispišite izlaz ove naredbe i objasnite značenje svih vrijednosti.
2. Koliko MB zauzima dataset u memoriji? Obrazložite kako ste došli do tog zaključka.
3. Odgovorite na ista pitanja za testni skup podataka.

2.2 Prikazivanje slika prije treniranja

Sljedeće naredbe se mogu koristiti za vizualizaciju nekih slika prije reorganizacije podataka i treniranja.

```
plt.subplot(221)
plt.imshow(xTrain[10])
plt.axis('off')
plt.title(classNames[yTrain[10][np.array(0)]])
plt.subplot(222)
plt.imshow(xTrain[11])
plt.axis('off')
plt.title(classNames[yTrain[11][np.array(0)]])
plt.subplot(223)
plt.imshow(xTest[10])
plt.axis('off')
plt.title(classNames[yTest[10][np.array(0)]])
plt.subplot(224)
plt.imshow(xTest[11])
plt.axis('off')
plt.title(classNames[yTest[11][np.array(0)]])
plt.savefig(baseDir+'svm0.png')
plt.clf()
```

Odgovorite na sljedeća pitanja:

1. Koja biblioteka i koje naredbe se koriste za prikazivanje slika?
2. Koje slike će biti prikazane?
3. Koja naredba se koristi za snimanje slike na *drive*?
4. Kako se određuje baseDir?
5. Zašto se koristi `[np.array(0)]` i da li se informacija o klasi može prikazati bez ove naredbe?

2.3 Organizacija podataka

Funkcija `cifar10.load_data()` očekuje da se podaci dijele na dva skupa: skup za treniranje i skup za testiranje. Međutim, praksa je pokazala da je dataset dobro podijeliti na tri skupa, tj. dobro je imati i skup za validaciju. Razlozi za ovo su objašnjeni na predavanjima. Sljedeći python kod se može koristiti za reorganizaciju podataka.

```
xVal = xTrain[49000:, :].astype(np.float)
yVal = np.squeeze(yTrain[49000:, :])
xTrain = xTrain[:49000, :].astype(np.float)
yTrain = np.squeeze(yTrain[:49000, :])
yTest = np.squeeze(yTest)
xTest = xTest.astype(np.float)
```

Odgovorite na sljedeća pitanja:

1. Zašto je dobro koristiti skup za validaciju? Koji parametri se podešavaju ovim skupom?

2. Koliko elemenata imaju sada skupovi za treniranje, testiranje i validaciju?
3. Šta radi naredba `np.squeeze()`? Odgovor pronaći u dokumentaciji Python-a i detaljno obrazložiti za konkretni primjer.
4. Koji je ukupni efekat prethodnih naredbi na organizaciju dataset-a?
5. Koliko MB memorije je sada potrebno za sve slike iz dataset-a?

2.4 Pretprocesiranje slika

Sljedeći kod se koristi za normalizaciju slika u skupu podataka.

```
meanImage = np.mean(xTrain, axis=0)
xTrain -= meanImage
xVal -= meanImage
xTest -= meanImage
```

Odgovorite na sljedeća pitanja:

1. Zašto je potrebno napraviti normalizaciju podataka?
2. Kako se realizuje normalizacija?
3. Prikazati 4 slike iz skupa za treniranje nakon primjene normalizacije. Koristiti kod dat u sekciji 2.2. Obradite pažnju da je oblik podataka sada izmijenjen, pa je potrebno napraviti neke (privremene) promjene na odabranim slikama.

2.5 Oblikovanje slika i dodavanje dimenzije za otklon (bias)

Sljedeći kod se koristi za oblikovanje slika i dodavanje dimenzije za otklon (bias).

```
xTrain = np.reshape(xTrain, (xTrain.shape[0], -1))
xVal = np.reshape(xVal, (xVal.shape[0], -1))
xTest = np.reshape(xTest, (xTest.shape[0], -1))

xTrain = np.hstack([xTrain, np.ones((xTrain.shape[0], 1))])
xVal = np.hstack([xVal, np.ones((xVal.shape[0], 1))])
xTest = np.hstack([xTest, np.ones((xTest.shape[0], 1))])
```

Odgovorite na sljedeća pitanja:

1. Detaljno objasnite na koji način je izvršeno oblikovanje skupova podataka, tj. koje naredbe su korištene?
2. Koji oblik sada imaju podaci u skupovima? Napisati kod koji ispisuje ove informacije (tri print naredbe).
3. Koja naredba se koristi za dodavanje dodatne dimenzije u skupovima? Kako se ona koristi?
4. Koji oblik sada imaju podaci u skupovima? Napisati kod koji ispisuje ove informacije (tri print naredbe).

3 SVM klasifikator

Sada kad su podaci pripremljeni za treniranje, potrebno je primijeniti SVM klasifikator. Za potrebe ove vježbe koristit će se gotova funkcija iz biblioteke `svm`.

Prije treniranja klasifikatora, prikazat ćemo matricu koeficijenata W u svrhu poređenja sa istom matricom nakon treniranja.

```
from svm import Svm
numClasses = np.max(yTrain) + 1

classifier = Svm(xTrain.shape[1], numClasses)

if classifier.W is not None:
    tmpW = classifier.W[:-1, :]
    tmpW = tmpW.reshape(32, 32, 3, 10)
    tmpWMin, tmpWMax = np.min(tmpW), np.max(tmpW)
    for i in range(numClasses):
        plt.subplot(2, 5, i+1)
        plt.title(classesName[i])
        wPlot = 255.0 * (tmpW[:, :, :, i].squeeze() - tmpWMin) / (tmpWMax - tmpWMin)
        plt.imshow(wPlot.astype('uint8'))
    plt.savefig(baseDir+'svm1.png')
    plt.clf()

classifier.train(xTrain, yTrain, lr=1e-7, reg=5e4, iter=1500, verbose=True)
```

Odgovorite na sljedeća pitanja:

1. Objasnite šta se pokušava prikazati sa iskazom `if`. Detaljno objasnite postupak.
2. Isprintajte nekoliko (5) nasumičnih vrijednosti iz matrice W . Šta možete zaključiti na osnovu ovih informacija?
3. Koja naredba se koristi za pokretanje treniranja klasifikatora?
4. Objasnite sve parametre ove naredbe.
5. Izmjerite vrijeme potrebno da se istrenira klasifikator i isprintajte ga.

3.1 Izračunavanje tačnosti klasifikatora

Za izračunavanje tačnosti klasifikatora koristi se naredba `classifier.calAccuracy()`, pri čemu se kao parametri šalju: 1) skup podataka (npr. `xTrain`) i 2) labela (npr. `yTrain`).

Odgovorite na sljedeća pitanja:

1. Definirati tri poziva funkcije nad tri skupa podataka i izračunati tačnost za sva tri skupa podataka. Prikazati dobijene vrijednosti (na ekranu i u izvještaju).
2. Grafički prikazati vrijednosti matrice W (kao na početku sekcije 3), a zatim i isprintati nekoliko (5) nasumičnih vrijednosti. Šta možete zaključiti na osnovu ovih informacija, tj. u odnosu na informacije prikazane u prethodnoj sekciji?
3. Napraviti testiranja na osnovu parametara prikazanih u tabeli 1. i izmjeriti tačnosti (3) za svaki test.
4. Na osnovu prethodnih mjerenja, zaključiti koji skup parametara daje najveću tačnost.

4 Softmax klasifikator

Za potrebe ove vježbe koristit će se gotova funkcija iz biblioteke `softmax` (`from softmax import Softmax`). Kod za korištenje ovog klasifikatora se razlikuje samo u učitavanju biblioteke `softmax` i pozivu funkcije `Softmax()` umjesto `Svm()`.

Odgovorite na sljedeća pitanja:

| Regression (reg) Learning rate (lr) | 500 | 1500 | 7500 | 12000 | skup |
|--|-----|------|------|-------|--|
| 0.5e-7 | | | | | treniranje validacija testiranje |
| 1e-7 | | | | | treniranje validacija testiranje |
| 2e-7 | | | | | treniranje validacija testiranje |
| 6e-7 | | | | | treniranje validacija testiranje |

Tabela 1: Mjerenja za zadatak 3.1.3.

| Regression (reg) Learning rate (lr) | 500 | 1500 | 7500 | 12000 | skup |
|--|-----|------|------|-------|--|
| 0.5e-7 | | | | | treniranje validacija testiranje |
| 2e-7 | | | | | treniranje validacija testiranje |
| 4e-7 | | | | | treniranje validacija testiranje |
| 8e-7 | | | | | treniranje validacija testiranje |

Tabela 2: Mjerenja za zadatak 4.6.

1. Ispod koda za SVM dodajte kod za treniranje Softmax klasifikatora i napravite izmjene specifične za ovaj klasifikator (potrebne su minorne izmjene koda).
2. Prije treniranja isprintajte nekoliko (5) nasumičnih vrijednosti iz matrice W . Šta možete zaključiti na osnovu ovih informacija, tj. u odnosu na informacije dobijene nakon inicijalizacije matrice W kod SVM-a?
3. Pokrenuti treniranje klasifikatora i izračunari vrijeme treniranja, te ga prikazati na ekranu i u izvještaju.
4. Definirati tri poziva funkcije `classifier.calAccuracy()` nad tri skupa podataka i izračunati tačnost za sva tri skupa podataka. Prikazati dobijene vrijednosti (na ekranu i u izvještaju).
5. Grafički prikazati vrijednosti matrice W , a zatim i isprintati nekoliko (5) nasumičnih vrijednosti. Šta možete zaključiti na osnovu ovih informacija, tj. u odnosu na informacije prikazane u prethodnoj sekciji (nakon treniranja)?
6. Uporedite grafički prikaz matrice W nakon treniranja kod SVM-a i Softmax-a. Koje zaključke možete izvesti?
7. Napraviti testiranja na osnovu parametara prikazanih u tabeli 2. i izmjeriti tačnosti (3) za svaki test.
8. Na osnovu prethodnih mjerenja, zaključiti koji skup parametara daje najveću tačnost.
9. Na osnovu svih mjerenja za SVM i Softmax, koji klasifikator i sa kojim parametrima je imao najveću tačnost?
10. Uporediti vrijeme izvršenja ova dva klasifikatora. Generalno, koji je brži?