

Metode i primjene vještačke inteligencije

Haris Dajic

November 2019

1 Učitavanje dataset-a

1. Koja naredba se koristi za utvrđivanje oblika strukture xTrain? Ispišite izlaz ove naredbe i objasnite značenje svih vrijednosti.
Koristi se naredba 'shape'. Izlaz ove naredbe je (50000, 32, 32, 3), gdje 50000 označava broj slika, 32x32 jeste dimenzija slika, a 3 označava broj kanala.
2. Koliko MB zauzima dataset u memoriji? Objasnite kako ste došli do tog zaključka.
Veličina slike 32x32, pomnožena sa brojem kanala (3) i brojem slika (50 000).

$$32 * 32 * 3 * 50000 = 153600000$$

Kada to pretvorimo u MB, dobijemo 146.484375 MB. Iste informacije smo mogli dobiti i pozivom naredbe nbytes.

3. Odgovorite na ista pitanja za testni skup podataka.
Koristi se ista naredba kao i kod skupa za treniranje, samo što je kod testnog skupa broj slika 10 000. Koristeći naredbu 'nbytes' dobijemo vrijednost 30720000 B, kada ovo konvertujemo u MB dobijemo 29.296875.

2 Prikazivanje slika prije treniranja

1. Koja biblioteka i koje naredbe se koriste za prikazivanje slika?
Biblioteka koja se koristi jeste **matplotlib**, a naredbe koje se koriste za prikazivanje slika jesu **subplot** i **imshow**.
2. Koje slike će biti prikazane?
Slike koje će biti prikazane iz skupa xTrain jesu slike sa indeksima 10 i 11. Takoer i slike iz skupa xTest će biti prikazane sa indeksima 10 i 11. Te slike pripadaju sledećim klasama: deer, horse, plane i truck.
3. Koja naredba se koristi za snimanje slike na drive?
Koristi se naredba **savefig**.
4. Kako se određuje baseDir?
Određuje se na osnovu lokacije fajla kojeg smo dobili pomoću sljedećeg poziva:

```
os.path.dirname(os.path.abspath('__file__')) + '/'
```

.

5. Zašto se koristi [np.array(0)] i da li se informacija o klasi može prikazati bez ove klase?
Obzirom da yTrain i yTest su matrice čiji su elementi nizovi sa po jednim elementom, iz tog razloga kada pristupimo jednom od ova dva objekta, on nam vraća niz pa iz tog razloga moramo označiti kojem tačno elementu pristupamo. Umjesto [np.array(0)] mogli smo jednostavno napisati [0].

3 Organizacija podataka

1. Zašto je dobro koristiti skup za validaciju? Koji parametri se podešavaju ovim skupom?
Vrši se provjera koje metode za klasifikaciju su najbolje, na primjer ako koristimo random forest, da li je onda bolje koristiti sa 40 ili 50 stabala.
2. Koliko elemenata imaju sada skupovi za treniranje, testiranje i validaciju?
Skup za treniranje ima 49 000 elemenata, 10 000 za testiranje i 1000 za validaciju.
3. Šta radi naredba np.squeeze()? Odgovor pronaći u dokumentaciji Python-a i detaljno obrazložiti za konkretan primjer.
Naredba 'squeeze' uklanja jednodimenzionalne unose iz niza.

4. Koji je ukupni efekat prethodnih naredbi na organizaciju dataset-a?
Uklonjeni su nepotrebni nizovi i promijenjen je tip podataka u float.
5. Koliko MB memorije je sada potrebno za sve slike iz dataset-a?
Potrebno je isto memorije, jer je isti broj slika.

4 Pretprocesiranje slika

1. Zašto je potrebno napraviti normalizaciju podataka?
Kako bi se transformirali podaci, tako da se svi podaci mogli postaviti na istu skalu. Tipičan postupak normalizacije skalira numeričke vrijednosti da budu u vrijednostima između 0 i 1. Mogućnost postojanja situacije u kojoj bi jedba težina bila mnogo veća od ostalih je jedan od razloga uvođenja normalizacije.
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.



5 Oblikovanje slika i dodavanje dimenzije za otklon (bias)

1. Detaljno objasnite na koji način je izvršeno oblikovanje skupova podataka, tj. koje naredbe su korištene?
 - `numpy.reshape` - mijenja oblik niza, bez mijenjanja njegovih podataka. Novi oblik bi trebao biti kompatibilan prijašnjem.
 - `numpy.hstack` - funkcija se koristi za slaganje redoslijeda ulaznih nizova vodoravno kako bi se napravio jedan niz.
 - `numpy.ones` - kreira niz sa zadanom dimenzijom i tipom popunjen jedinicama.

```
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))])
```

2. Koji oblik sada imaju podaci u skupovima? Napisati kod koji ispisuje ove informacije (tri print naredbe).

```
print(xTrain.shape) (49000, 3073)
print(xTest.shape) (10000, 3073)
print(xVal.shape) (1000, 3073)
```
3. Koja naredba se koristi za dodavanje dodatne dimenzije u skupovima? Kako se ona koristi?
 Koristi se naredba `np.expand(array, axis)`. Array označava niz kojem se želi povećati dimezionalnost, a axis osa po kojoj se želi dodati.
4. Koji oblik sada imaju podaci u skupovima? Napisati kod koji ispisuje ove informacije (tri print naredbe).

```
print(xTrain.shape) (49000, 3073, 1)
print(xTest.shape) (10000, 3073, 1)
print(xVal.shape) (1000, 3073, 1)
```

6 SVM klasifikator

1. Objasnite šta se pokušava prikazati sa iskazom if. Detaljno objasnite postupak.
2. Isprintajte nekoliko (5) nasumičnih vrijednosti iz matrice W. Sta možete zaključiti na osnovu ovih informacija?

```
print(classifier.W[15])
print(classifier.W[25])
print(classifier.W[35])
print(classifier.W[45])
print(classifier.W[55])
```

Dobijene vrijednosti:

```
-0.00645513, -0.00158665, 0.00070405, -0.01910093, -0.0105467, -0.00112344, 0.004326, 0.00781659, 0.0110238,
0.0147575
-0.00205957, -0.00812182, -0.0028092, -0.01159192, 0.00368118, -0.01084964, 0.00193525, -0.0129366, 0.00940983,
-0.0064835
-0.00776758, 0.02460273, -0.01013406, -0.00994739, 0.00344531, 0.00318486, 0.00221194, 0.01222155, -0.02062943,
-0.0130985
0.012275, -0.01062976, -0.00155651, 0.00293953, 0.01366348, -0.00353692, 0.00067406, 0.00777851, -0.00547198,
-0.01546757
-0.01391245, 0.00585021, -0.00091048, -0.0006698, -0.01071149, 0.00537002, -0.01478887, 0.00445001, 0.01283774,
-0.00180221
```

Svaki red matrice W ima 10 kolona, koje predstavljaju klase.

3. Koja naredba se koristi za pokretanje treniranja klasifikatora?
 Naredba za pokretanje treniranja klasifikatora:

```
classifier.train(self, x, y, lr = 1e - 3, reg = 1e - 5, iter = 100, batchSize = 200, verbose = False)
```

4. Objasnite sve parametre ove naredbe.

- x - podaci za treniranje
- y - izlazni podaci
- lr - tipa float koji označava stepen učenja za optimizaciju
- reg - tipa float i označava jačinu regularizacije

- iter - ukupan broj iteracija
- batchSize - broj primjera u svakom batch izvršenju
- verbose - ispis gubitaka i tačnosti treniranja u slučaju da je 'true'

5. Izmjerite vrijeme potrebno da se istrenira klasifikator i isprintajte ga. Vrijeme potrebno da se istrenira klasifikator: 10.62455736512047.

7 Izračunavanje tačnosti klasifikatora

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).

```
print("xTrain: ", classifier.calAccuracy(xTrain, yTrain))
print("xVal: ", classifier.calAccuracy(xVal, yVal))
print("xTest: ", classifier.calAccuracy(xTest, yTest))
```

```
xTrain: 36.028161568515645
xVal: 36.71
xTest: 35.3
```

2. Grafički prikazati vrijednosti matrice W (kao na početku sekcije 3), a zatim i isprintati nekoliko (5) nasumičnih vrijednosti. Sta 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.

Regression(reg) Learning rate(lr)	500	1500	7500	12000	skup
0.5e-7	36.70204081632653 36.11 35.7	37.37755102040816 36.36 37.1	36.416326530612245 35.5 35.9	35.542857142857144 34.980000000000004 36.811005050901104	treniranje validacija testiranje
1e-7	36.9734693877551 35.690000000000005 37.8	38.16326530612245 36.76 38.099999999999994	36.28775510204082 35.61 35.800000000000004	35.502040816326534 34.55 35.4	treniranje validacija testiranje
2e-7	36.463265306122445 34.8 36.800000000000004	37.46530612244898 35.35 37.6	35.91428571428571 35.2 35.4	34.58571428571428 34.21 35.2	treniranje validacija testiranje
6e-7	36.09795918367347 35.451689999999995 35.7	33.96122448979592 32.39 33.5	32.897959183673464 32.38 33.199999999999996	28.954693877551017 28.430000000000003 28.2	treniranje validacija testiranje

4. Na osnovu prethodnih mjerenja, zaključiti koji skup parametara daje najveću tačnost. Najveću tačnost daje skup za treniranje.

8 Softmax klasifikator

1. Ispod koda za SVM dodajte kod za treniranje Softmax klasifikatora i napravite izmjene specifične za ovaj klasifikator (potrebne su minorne izmjene koda).

```
numClasses = np.max(yTrain) + 1
#classifier = Svm(xTrain.shape[1], numClasses)
classifier = Softmax(xTrain.shape[1], numClasses)
```

- Prije treniranja isprintajte nekoliko (5) nasumičnih vrijednosti iz matrice W. Sta možete zaključiti na osnovu ovih informacija, tj. u odnosu na informacije dobijene nakon inicijalizacije matrice W kod SVM-a?
0.00645513, -0.00120465, 0.00078905, -0.01530093, -0.0105467, -0.00112344, 0.004326, 0.00781659, -0.0110238, 0.0147575
0.00205957, 0.00812182, -0.0028092, -0.01159192, 0.00368118, -0.01084964, 0.00193525, 0.0127366, 0.00980983, -0.0064235
-0.00776752, 0.02460273, -0.01013406, -0.00994739, 0.00344531, 0.00408486, 0.00221194, 0.01222155, -0.03062943, -0.0130985
0.012275, 0.01062976, -0.00155651, 0.00293953, 0.01366348, -0.00353692, 0.00060286, 0.00572851, 0.00447198, -0.01546757
-0.01391245, 0.00585021, -0.00091048, -0.0006698, -0.01071875, 0.00537021, -0.01478887, 0.00445001, 0.01283774, -0.00180568

Vrijednosti se ne razlikuju puno od vrijednosti koje dobijemo koristeći 'svm', jer na isti način dobijamo vrijednosti matrice W.

- Pokrenuti treniranje klasifikatora i izračunari vrijeme treniranja, te ga prikazati na ekranu i u izvještaju.
Vrijeme treniranja: 8.69527458558213.
- Definisati 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).

```
print("xTrain: ", classifier.calAccuracy(xTrain, yTrain))
print("xVal: ", classifier.calAccuracy(xVal, yVal))
print("xTest: ", classifier.calAccuracy(xTest, yTest))
```

```
xTrain: 33.5195174625
xVal: 33.54
xTest: 33.7
```

- Grafički prikazati vrijednosti matrice W, a zatim i isprintati nekoliko (5) nasumičnih vrijednosti. Sta možete zaključiti na osnovu ovih informacija, tj. u odnosu na informacije prikazane u prethodnoj sekciji (nakon treniranja)?
- Uporedite grafički prikaz matrice W nakon treniranja kod SVM-a i Softmax-a. Koje zaključke možete izvesti?
- Napraviti testiranja na osnovu parametara prikazanih u tabeli 2. i izmjeriti tačnosti (3) za svaki test.

Regression(reg) Learning rate(lr)	500	1500	7500	12000	skup
0.5e-7	34.27551020408163 34.08 36.0	35.46530612244898 35.1 36.4	34.33061224489796 32.9 35.800000000000004	31.60612244897959 31.8 33.6	treniranje validacija testiranje
2e-7	36.96530612244898 35.92 37.2	36.6265306122449 36.290000000000006 37.5	33.702152512151597 33.742318762380002 35.1	31.9539387755102 31.73 33.5	treniranje validacija testiranje
4e-7	37.614285714285714 36.35 37.400000000000006	37.932653061224485 36.410000000000004 37.6	33.29183673469388 33.13 35.3	32.538775510204076 32.72 34.3	treniranje validacija testiranje
8e-7	36.09795918367347 35.379999999999995 35.6	36.18775510204082 36.18 36.4	32.50204081632653 32.24 34.6	31.620204081632654 30.51 31.6	treniranje validacija testiranje

8. Na osnovu prethodnih mjerenja, zaključiti koji skup parametara daje najveću tačnost.
Kao i prošli put, u većini slučajeva skup za testiranje ima najveću tačnost.
9. Na osnovu svih mjerenja za SVM i Softmax, koji klasifikator i sa kojim parametrima je imao najveću tačnost?
- SOFTMAX:
Ir 4e-7 i reg: 1500
Treniranje: 37.932653061224485
Validacija: 36.410000000000004
Testiranje: 37.6
 - SVM:
Ir: 1e-7 i reg: 1500
Treniranje: 38.16326530612245
Validacija: 36.76
Testiranje: 38.099999999999994
10. Uporediti vrijeme izvršenja ova dva klasifikatora. Generalno, koji je brži?

Softmax: 8.6952748558213 Svm: 10.62455736512047

Vidimo da je softmax brži.