Pripremni materijali za Nedelju informatike

Organizatori Nedelje informatike

1. april 2018

Sadržaj

Glava 1

O ovom dokumentu

Pripremа za predavanja

Svesni smo da predznanje polaznika Nedelje informatike dosta varira u nekim oblastima. Koliko smo mogli, potrudili smo se da predavanja učinimo pristupačnim svima koji su uspeli da prođu naš "ulazni test".

Ipak, pojedina predavanja zahtevaju poznavanje nekih matematičkih pojmova koje možda nisu svi videli do sada, a njihovo uvođenje i objašnjavanje bi nam oduzelo previše vremena koje bismo mogli da iskoristimo za daleko zanimljivije stvari. Radionice bi bilo teško realizovati ukoliko biste po prvi put čuli ili videli za neke stvari koje ćemo na njima koristiti. Ovi materijali postoje da bi vas spremili i ne bi trebalo da vam njihovo čitanje oduzme previše vremena.

Trudili smo se da pojasnimo ove pojmove što je moguće jednostavnije. Ukoliko je nešto nejasno, postoji dosta materijala o stvarima koje su ovde napisane koje možete da nađete i *online* jako lako. Ukoliko još uvek imate problema, obratite nam se!

Raspored Nedelje

Raspored predavanja i radionica na Nedelji informatike je fleksibilan, ali ovo je trenutni plan:

- Ponedeljak
  - **Uvod u paralelno procesiranje** – Miloš Stanojević
  - **Formalna verifikacija programa** – Andrej Ivašković
  - **Modelovanje širenja bolesti** – Vuk Vuković
- Utorak
  - **Emission security** – Dimitrije Erdeljan
  - **Kako videti *n* dimenzija?** – Nikola Jovanović
  - **Od NAND-a do procesora** – Lazar Mitrović
- Sreda

  - **Učenje sa pojačavanjem** – Petar Veličković
  - **OpenAI Gym radionica** – Filip Vesović, Vladimir Milenković

- Četvrtak
  - **Više o segmentnim stablima** – Kosta Grujčić
  - **Simulacija fizičkih interakcija u dvodimenzionalnom prostoru** – David Davidović
  - **Programirati znači dokazati** – Andrej Ivašković

- Petak
  - **3D grafika** – Marko Stanojević
  - **Regularni izrazi kroz primere** – Mina Šekularac
  - **The next frontier of Machine Learning** – Jovana Mitrović

Predavanja traju po **sat vremena**. Između predavanja će biti pauze od oko 15 minuta.

Svim danima će predavanja biti održana u **Matematičkoj gimnaziji** (Kraljice Natalije 37). Tada ćemo sve vreme biti u jednom od kabineta na trećem spratu i predavanja će početi u **14:00** ovim danima.

Imajte u vidu ovaj raspored pri čitanju ovih materijala: do ponedeljka pročitajte sve što je relevantno za "Računarsko modelovanje širenja bolesti", do srede sve što je relevantno za "Učenje sa pojačavanjem" itd.

Autori materijala

Ove materijale su sastavili organizatori Nedelje informatike, i to konkretno:

1. **Računarsko modelovanje širenja bolesti** – Vuk Vuković
2. **Kako videti *n* dimenzija?** – Nikola Jovanović
3. **Učenje sa pojačavanjem** – Petar Veličković
4. **Simulacija fizičkih interakcija u dvodimenzionalnom prostoru** – David Davidović

Glava 2

Pripremni materijali

Računarsko modelovanje širenja bolesti

Uvod

Svrha ovog predavanja će biti da vam prikaže na koji način se računari mogu primeniti na modelovanje realnih problema iz svakodnevnice. Takođe, plan je da naučite i osnove Python biblioteka *numpy* i *matplotlib* kako biste u budućnosti mogli da brzo i elegantno rešavate razne probleme: od obrade rezultata laboratorijskih vežbi na fakultetu do maturskih ili istraživačkih radova.

Python se izdvaja kao odličan jezik za rešavanje ovakvih problema uzimajući u obzir njegov visok nivo apstrakcije i mogućnost pisanja koncizno koda sa velikom slobodom. U daljem tekstu biće smatrano da je čitalac upoznat sa osnovama jezika Python, tj. njegovom sintaksom, petljama, funkcijama, primitivnim i apstraktnim tipovima podataka (int, float, string, boolean, liste, skupovi, rečnici) kao i njihovim operacijama.

Pored instalacije samog Python-a, biblioteke koje su nam potrebne su numpy, matplotlib i scipy. U zavisnosti od operativnog sistema koji koristite, pomenute stvari se relativno jednostavno instaliraju prateći uputstva sa zvaničnih sajtova. Takođe, Python zajedno sa svim potrebnim bibliotekama je moguće instalirati i koristiti iz okruženja Spyder u okviru paketa Anaconda.

Numpy

Numpy je Python biblioteka koja se najčešće koristi u naučne svrhe i pre svega omogućava efikasnu upotrebu više-dimenzionih nizova u programskom jeziku Python.

Za početak, potrebno je dodati modul *numpy*.

```
import numpy as np
```

Nadalje, prilikom korišćenja funkcija iz ovog modula pozivi će se odvijati pomoću skraćenice np.

Osnovni objekat u radu sa *numpy* bibliotekom je homogeni multidimenzioni niz koji predstavlja tabelu elemenata (najčešće brojeva) koji su svi **istog tipa**.

Za početak, pogledajmo načine na koje se mogu definisati *numpy* nizovi.

```
# Niz od 3 elementa
np.array([5,7,8])
```

```
# Matrica dimenzija 2x3
np.array([[1., 0., 0.], [0., 1., 2.]])
# Nula matrica dimenzija 3x4
np.zeros((3,4))
# Višedimenzioni niz 2x4x3 ispunjen jedinicama
np.ones((2,4,3))
# Matrica dimnezija 2x3 ispunjena slučajnim
# vrednostima od 0 do 1
np.random((2,3))
```

Veoma korisne funkcije ***arange*** i ***linspace*** nam omogućavaju stvaranje ekvidistantnih sekvenci u određenim granicama.

```
# niz brojeva od 10 do 30 (bez 30) sa korakom 0.5
np.arange(10, 30, 0.5)
# niz od 1000 ekvidistantnih tačaka od 0 do 10
np.linspace(0, 10, 1000)
```

Najpogodnija stvar koju *numpy* omogućava jeste primena funkcija i aritmetičkih izraza na njegove objekte.

```
A = np.array([[1,1], [0,1]])
B = np.array([[2,0], [3,4]])

# razlika matrica A i B
C = A-B
# proizvod pojedinačnih elemenata
D = A*B
# matrični proizvod
E = np.dot(A, B)

x = np.linspace(0, 2*np.pi, 100)
t = np.arange(0, 10, 100)

# vrednosti funkcije 2sin(x) zadate nizom x
y = 2*np.sin(x)
# vrednosti funkcije e^(-x^2*t) zadate nizovima x i t
g = np.exp(-x**2*t)
# Maksimalna vrednost niza
M = max(y)
# Suma niza
S = sum(y)
```

Matplotlib

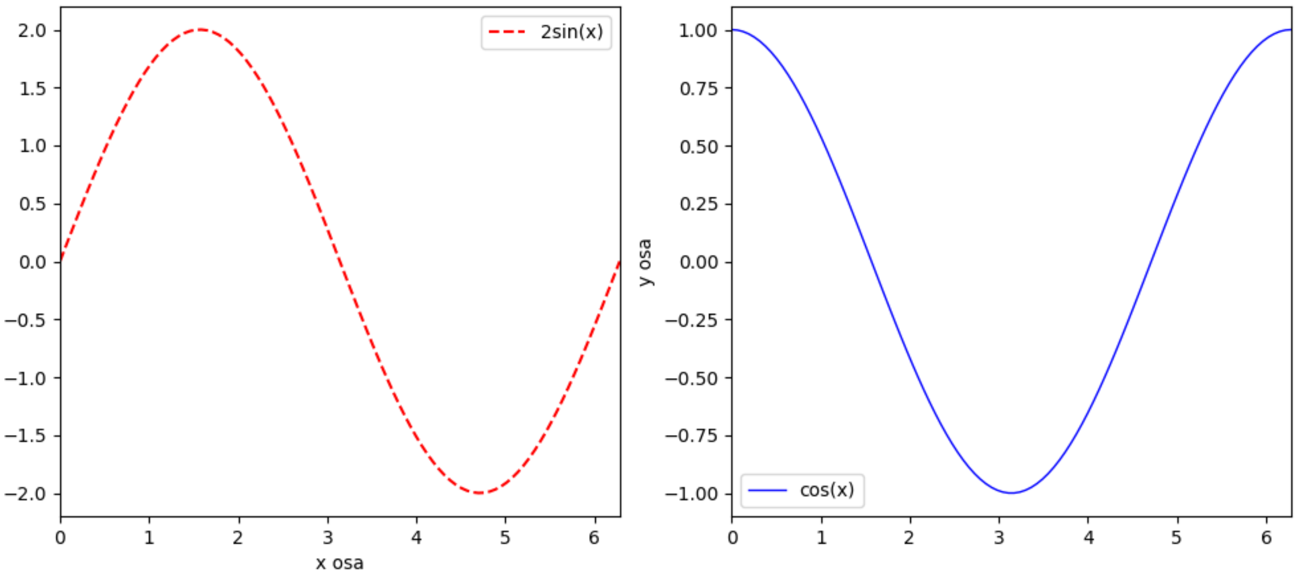
Biblioteka *matplotlib* omogućava iscrtavanje grafika, histograma, scatterplot i 3D grafika u Python-u. Najčešće se koristi u uskoj vezi sa *numpy* bibliotekom.

Modul ove biblioteke se uvozi na sledeći način (sa najčešćom skraćenicom plt).

```
import matplotlib.pyplot as plt
```

Svi grafici se crtaju na platnu koje se naziva ***figure***. Opciono je moguće podesiti broj platna, dimenzije u inčima, kao i rezoluciju. Na jedno platno se može dodati više grafika funkcijom ***subplot***. Prvi argument ove funkcije predstavlja broj redova, drugi broj kolona, a treći redni broj grafika brojeći sa leva na desno kroz redove. Funkcijom ***plot*** se iscrtava grafik tako što se prosleđuju liste vrednosti na x i y osi, kao i dodatni opcioni argumenti koji opisuju izgled grafika: debljina, boja i stil linije (-, :, -., –), naziv koji se prikazuje u legendi. Legenda se uključuje funkcijom ***legend***, dok se njena lokacija može podesiti argumentom ***loc***. Moguće je eksplicitno ograničiti opseg x i y ose funkcijama ***xlim*** i ***ylim***. Nazivi x i y osa se navode funkcijama ***xlabel*** i ***ylabel***. Finalno, grafik se prikazuje funkcijom ***show***. Sve opisane funkcije oslikane su u kodu i grafiku u sledećem primeru.

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0,2*np.pi,100)
y1 = 2*np.sin(x)
y2 = np.cos(x)

plt.figure(num=1,figsize=(12,5),dpi=100)

plt.subplot(1,2,1)
plt.plot(x,y1,color="red",linewidth=1.5,linestyle="--",label="2sin(x)")
plt.legend(loc=1)
plt.xlim(0,2*np.pi)
plt.ylim(-2.2,2.2)
plt.xlabel("x osa")

plt.subplot(1,2,2)
plt.plot(x,y2,color="blue",linewidth=1.0,linestyle="-",label="cos(x)")
plt.legend(loc=3)
plt.xlim(0,2*np.pi)
plt.ylabel("y osa")

plt.show()
```



Diferencijalne jednačine

Prilikom opisivanja i modelovanja određenog problema, mi u stvari opisujemo njegovu dinamiku tj. promenu parametara sistema u vremenu. Samim tim, modeli se najčešće sastoje od diferencijalnih jednačina.

Od posebnog značaja su nelinearne diferencijalne jednačine čija su analitička rešenja retko moguća. Srećom, korišćenjem numeričkih metoda, ovakve jednačine se mogu rešiti na određenom intervalu koji se posmatra. U tu svrhu ćemo koristiti *odeint* integrator iz Python biblioteke *scipy*.

Postupak rešavanja diferencijalnih jednačina primenom numerčkih integratora svodi se na tri koraka.

1. Svođenje diferencijalnih jednačina na jednačine prvog reda i predstavljanje u obliku:
$$\frac{dx}{dt} = f(x, t)$$
2. Predstavljanje diferencijalnih jednačina pomoću Python funkcija kojima se redom prosleđuju zavisna promenljiva, vreme i konstante. Primer:
```
def dxdt(x,t,a,k):
    return e**(-k*t) - a*x;
```
3. Definisanje početnih uslova, vremenskog intervala koji se posmatra, kao i konstanti i poziv funkcije odgovarajućeg integratora. Primer:
```
from scipy import integrate
sol = integrate.odeint(dxdt, x0, t, args=(a,k))
```
$dxdt$ je funkcija definisana u koraku 2, $x0$ predstavlja početni uslov, t vreme, a zatim se redom navode arugmenti u vidu tuple-a. Dobijena rešenja zatim dalje možemo obrađivati i grafički predstaviti.

Odeint integrator rešava diferencijalne jednačine prvog reda. Međutim, i diferencijalne jednačine višeg reda se mogu svesti na sistem diferencijalnih jednačina prvog reda koje ovaj integrator takođe može rešiti. Sledeći primer oslikava pomenuti postupak za diferencijalnu jednačinu drugog reda.

$$\frac{d^2x}{dt^2} + Ax = 0$$

Uvodimo smene:

$$y_1 = x,\ y_2 = \frac{dx}{dt}$$

Sada se početna diferencijalna jednačina drugog reda svodi na sistem dve diferencijalne jednačine prvog reda.

$$\frac{dy_1}{dt} = \frac{dx}{dt} = y_2$$
$$\frac{dy_2}{dt} = \frac{d^2x}{dt^2} = -Ax = -Ay_1$$

Primer za vežbu

Blok vezan za horizontalnu oprugu je otklonjen iz ravnotežnog položaja na rastojanje $x = 5m$ i održava se u stanju mirovanja. Masa bloka je $1kg$, koeficijent elastičnosti opruge je $0.5N/m$, a sistem se nalazi u vakuumu. Iscrtati grafik:

1. zavisnosti položaja bloka od vremena

2. zavisnosti potencijalne i kinetičke energije bloka od pozicije bloka

Rešenje odrediti u vremenskom intervalu od 0 do 20 sekundi, definisanom u 1000 tačaka.

Na osnovu 2. Njutnovog zakona sledi sledeća jednačina:

$$ma = -kx$$

$$m\frac{d^2x}{dt^2} = -kx$$

$$\frac{d^2x}{dt^2} + \frac{k}{m}x = 0$$

Korišćenjem smena iz prethodnog odeljka dobijamo sistem linearnih jednačina prvog reda koji se integratoru prosleđuju u vidu liste. Takođe, rešenja jednačine i se nalaze u koloni $i-1$ liste koja je vraćena kao rešenje. Primetiti da smena y_2 fizički gledano predstavlja brzinu bloka.

$$y_1 = x,\; y_2 = \frac{dx}{dt}$$

$$\frac{dy_1}{dt} = y_2,\; \frac{dy_2}{dt} = -\frac{k}{m}y_1$$

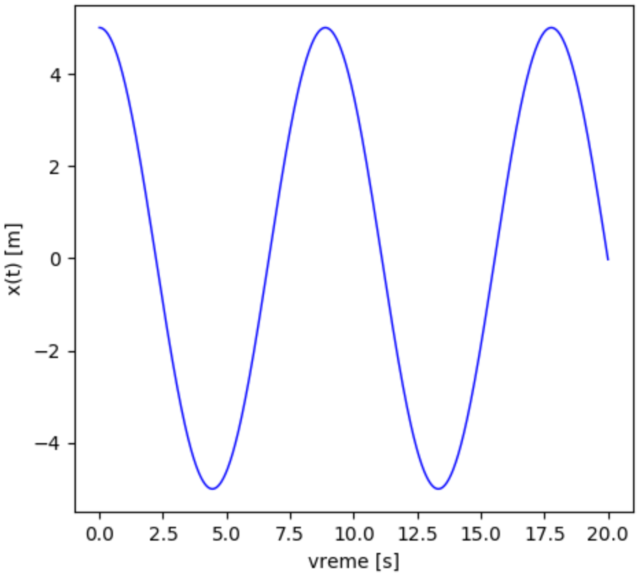
Takođe važe sledeće relacije za kinetičku i potencijalnu energiju.

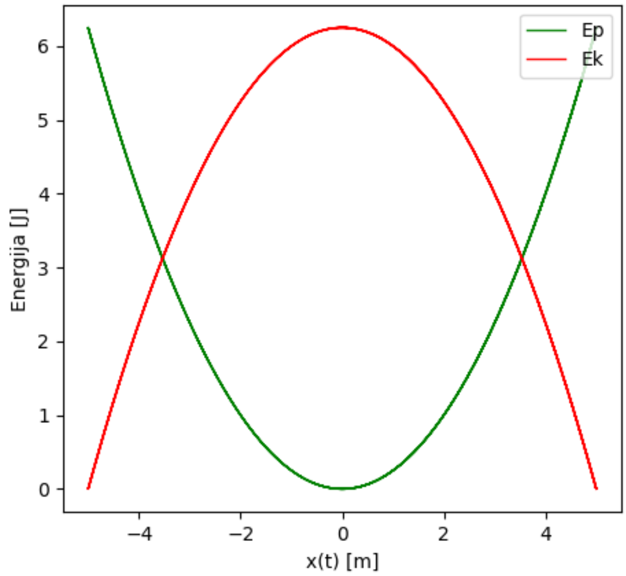
$$E_k = k\frac{x^2}{2}$$

$$E_p = m\frac{v^2}{2}$$

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy import integrate

def LHO(y,t,k,m):
  y1,y2=y
  return [y2, -k/m*y1]

k,m=0.5,1
x0,v0=5,0
t=np.linspace(0,20,1000)

sol = integrate.odeint(LHO, [x0,v0], t, args=(k,m))
x = sol[:,0]
v = sol[:,1]
Ep = k*x**2/2
Ek = m*v**2/2

plt.figure(1)
plt.plot(t,x,color="blue",linewidth=1.0,linestyle="-")
plt.xlabel('vreme [s]')
plt.ylabel('x(t) [m]')
plt.show()

plt.figure(2)
```

```
plt.plot(x,Ep,color="green",linewidth=1.0,linestyle="-",label="Ep")
plt.plot(x,Ek,color="red",linewidth=1.0,linestyle="-",label="Ek")
plt.legend(loc=1)
plt.xlabel('x(t) [m]')
plt.ylabel('Energija [J]')
plt.show()
```





Kako videti *n* dimenzija?

Za razumevanje ovog predavanja **neće biti neophodno** posebno predznanje. Svi potencijalno nepoznati pojmovi (uključujući i one navedene u nastavku) će biti objašnjeni u skladu sa vremenom koje je na raspolaganju. Ipak, poželjno je da pre samog predavanja usvojite osnovne koncepte iz domena neuralnih mreža i verovatnoće, makar na intuitivnom nivou. Ovo može da značajno olakša razumevanje samog predavanja, a naročito matematičkih osnova algoritama koji će biti prezentovani. U pitanju su sledeći pojmovi:

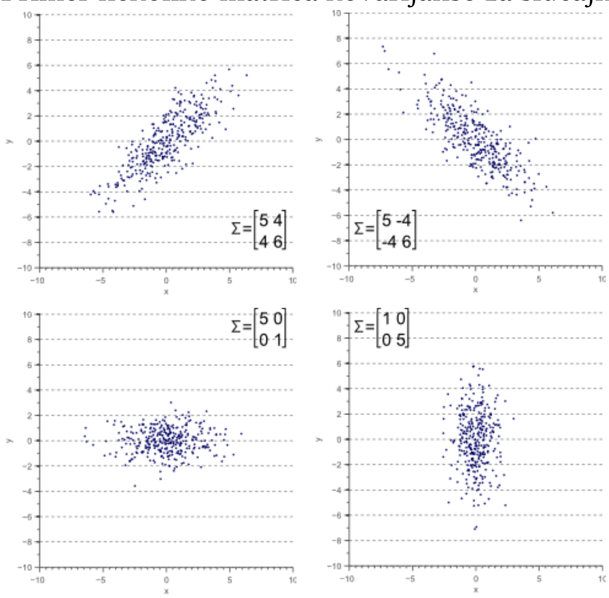
- Višeslojni perceptron, aktivaciona funkcija, funkcija gubitka, gradijentni spust – preporučujemo prošlogodišnje predavanje o neuralnim mrežama (videti slajdove 8–16):

  `http://csnedelja.mg.edu.rs/static/resources/v3.0/sre1_neuralne_pv.pdf`

- Matematičko očekivanje, varijansa, standardna devijacija – preporučujemo prošlogodišnje pripremne materijale (poglavlje 2.2.3):

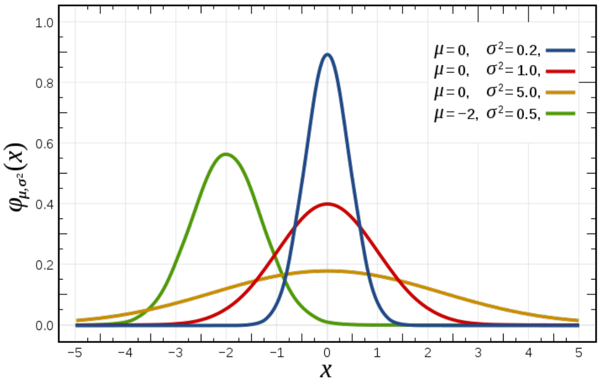
  `http://csnedelja.mg.edu.rs/static/resources/v3.0/pripremni_materijali.pdf`

- Kovarijansa:
  - Mera linearne povezanosti između dve slučajne promenljive X i Y, označava se sa $Cov(X, Y)$.
  - Ako velike vrednosti jedne promenljive odgovaraju velikim vrednostima druge, kovarijansa je pozitivna. U suprotnom, ako velike vrednosti jedne promenljive odgovaraju malim vrednostima druge, kovarijansa je negativna. Za nezavisne promenljive, ovaj broj je jednak nuli.
  - Formalno: $Cov(X, Y) = \mathbb{E}((X - \mathbb{E}(X))(Y - \mathbb{E}(Y)))$, gde $\mathbb{E}$ označava matematičko očekivanje. Primetimo da je $Cov(X, X) = Var(X)$, gde je Var oznaka za varijansu.
- Matrica kovarijanse:
  - Definišemo je za slučajan vektor $X = (X_1, \ldots, X_n)$ (vektor slučajnih promenljivih) i označavamo je sa Σ.
  - U i-tom redu i j-toj koloni ove matrice nalazi se kovarijansa promenljivih X_i i X_j. Kako je $Cov(X, Y) = Cov(Y, X)$ matrica je simetrična u odnosu na glavnu dijagonalu.
  - Kao što smo varijansu mogli da shvatimo kao meru „razbacanosti” neke slučajne promenljive, tako matrica kovarijanse u potpunosti opisuje „razbacanost” slučajnog vektora.
  - Primer nekoliko matrica kovarijanse za slučajni vektor (X, Y):



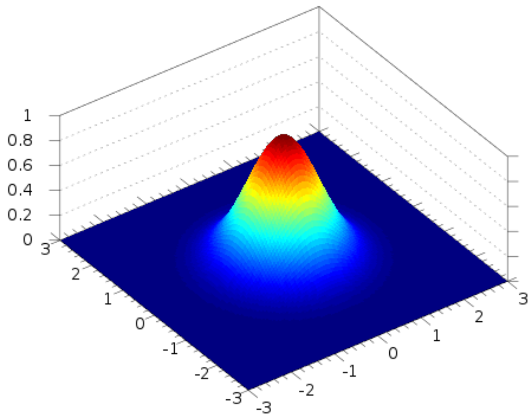
    Primetimo da brojevi na dijagonali predstavljaju varijanse promenljivih X i Y, dok $\Sigma_{12} = \Sigma_{21}$ označava njihovu linearnu povezanost.
- Normalna (Gausova) raspodela
  - Jedna od najvažnijih raspodela koja ima oblik zvona (tačnije, njena funkcija gustine).
  - Ako slučajna promenljiva X ima normalnu raspodelu pišemo $X \sim \mathcal{N}(\mu, \sigma^2)$, pri čemu parametar μ označava matematičko očekivanje X, a σ^2 varijansu.

– Prikaz normalne raspodele za različite parametre:



Primetimo da je najveća verovatnoća da promenljiva X uzme vrednost blisku μ tj. u „telu” raspodele, a da je ta verovatnoća manja za vrednosti ka „repovima” raspodele.

– Normalna raspodela može biti uopštena na više dimenzija. Prikaz jedne dvodimenzionalne raspodele za slučajni vektor (X, Y):



Učenje sa pojačavanjem

Predavanje o učenju sa pojačavanjem na ovogodišnjoj Nedelji informatike će biti izuzetno ambiciozno. Želja je da se za sat vremena pokriju svi neophodni detalji da biste mogli maksimalno da iskorististite *OpenAI Gym* radionicu koja sledi kasnije. Zbog toga, toplo **preporučujemo** da pre samog predavanja pregledate sledeće materijale (ili ih se podsetite):

- Slajdovi sa predavanja **“Veštačka inteligencija i igre”** (od Andreja Ivaškovića) sa prve Nedelje informatike. Na ovom predavanju su pokrivene osnove veštačke inteligencije i učenja sa pojačavanjem (bez uvođenja elemenata verovatnoće), sa algoritmom Q-učenja. Iako se na ovogodišnjem predavanju nećemo baviti tom klasom algoritama, ovi slajdovi i dalje predstavljaju odličan uvod.

  `http://csnedelja.mg.edu.rs/static/resources/v1.0/AI.pdf`

- Pripremni materijali za prošlogodišnje predavanje o **neuralnim mrežama**. Osnovni pojmovi o gradijentima, verovatnoći i statistici su i dalje izuzetno relevantni za ovo predavanje, dok se algoritam propagacije unazad i dalje može ostaviti za posle predavanja.

`http://csnedelja.mg.edu.rs/static/resources/v3.0/pripremni_materijali.pdf`

- Slajdovi sa prošlogodišnjeg predavanja o neuralnim mrežama mogu poslužiti kao dodatni kontekst—iako neće biti preterano relevantni za sadržaj ovog predavanja, mogu vam poslužiti za radionicu.

  `http://csnedelja.mg.edu.rs/static/resources/v3.0/sre1_neuralne_pv.pdf`

Simulacija fizičkih interakcija u dvodimenzionalnom prostoru

Prezentacija **Simulacija fizičkih interakcija u dvodimenzionalnom prostoru** je namenjena upoznavanju sa osnovnim pojmovima i tehnikama, sa strane računarskih nauka, koje omogućavaju *uverljivo* oponašanje fizičkih (preciznije, mehaničkih) zakona između *krutih tela*.

Akcenat nije na numeričkoj i fizičkoj tačnosti samih simulacija, već na postizanju uverljivosti za ljudsko oko. Ovakav pristup je koristan za, na primer, video-igre, ali (naravno) nije zadovoljavajući za, na primer, naučne eksperimente. Nećemo zalaziti u vrlo kompleksne dubine svih fizičkih svojstava koje je potrebno pokriti za kompletan fizički *engine*, već će nam fokus biti na osnovnim, generalnim konceptima koji se po potrebi mogu nadograđivati.

Stoga, predznanje koje se od vas očekuje se svodi na koncepte sa kojima ste tokom školovanja već upoznati, a to su:

1. **Vektori u ravni.** Očekuje se osnovno poznavanje pojma vektora i vektorskog računa. S obzirom na to da problemu fizičkih interakcija prilazimo sa strane računarskih nauka, vektore ćemo predstavljati i sa njima baratati kao uređenim parovima (x, y), koji predstavljaju dužinu projekcije vektora na x i y osu, respektivno. Nad ovom repezentacijom možemo definisati poznate operacije:

   - **Negacija vektora** $\vec{v} = (x, y)$ je jednostavno $-\vec{v} = (-x, -y)$.
   - **Zbir vektora** $\vec{v_1} = (x_1, y_1)$ **i** $\vec{v_2} = (x_2, y_2)$ je $\vec{v_1} + \vec{v_2} = (x_1 + x_2, y_1 + y_2)$.
   - **Intenzitet vektora** $\vec{v} = (x, y)$ je $|\vec{v}| = \sqrt{x^2 + y^2}$. Uglavnom, po konvenciji, izostavljanje strelice u oznaci vektora (dakle, v umesto $\vec{v}$) se takođe smatra oznakom za intenzitet.
   - **Skalarni proizvod vektora** $\vec{v_1} = (x_1, y_1)$ **i** $\vec{v_2} = (x_2, y_2)$ definiše se kao $\vec{v_1} \cdot \vec{v_2} = x_1x_2 + y_1y_2$. Prisetimo se da je intenzitet skalarnog proizvoda jednak $|\vec{v_1}||\vec{v_2}| \cos\theta$, gde je θ ugao koji zaklapaju $\vec{v_1}$ i $\vec{v_2}$.

2. **Njutnovska mehanika krutih tela.** Kao što je spomenuto, s obzirom na to da ne zalazimo u dubinu sâme fizike, osnovni pojmovi brzine, ubrzanja, sile, energije i impulsa, s kojima bi svi trebalo da ste već upoznati, će biti dovoljno predznanje. Podsetimo ih se:

- **Brzina tela** je mera promene pozicije tela u jedinici vremena. Brzinu ćemo predstavljati kao vektor $\vec{V} = (V_x, V_y)$, gde su V_x i V_y promena x- i y-koordinate po jedinici vremena, respektivno. Odatle direktno sledi da je pomeraj objekta koji se kreće brzinom $\vec{V}$ za vreme Δt jednak $\Delta\vec{x} = \vec{V}\Delta t$.
- **Ubrzanje tela** je mera promene brzine tela u jedinici vremena. Ubrzanje ćemo takođe predstavljati kao vektor $\vec{a} = (a_x, a_y)$, gde su a_x i a_y promena x- i y-intenziteta brzine po jedinici vremena, respektivno. Analogno gornjem, brzina koju će telo koje ubrzava ubrzanjem $\vec{a}$ dobiti za vreme Δt jednaka je $\Delta\vec{V} = \vec{a}\Delta t$, dok je pomeraj za isto vreme jednak $\Delta x = \dfrac{\vec{a}\Delta t^2}{2}$.
- **Silom** se smatra bilo koja interakcija koja utiče na kretanje tela ukoliko se ne spreči. **Drugi Njutnov zakon** povezuje silu i ubrzanje kao $F = ma$, gde je F intenzitet sile koja deluje na telo, m masa tela, a a ubrzanje tela.
- **Impuls** tela se definiše kao proizvod njegove mase i brzine. Impuls je vektorska veličina i označava se sa $\vec{p} = m\vec{v}$, gde je m masa tela a $\vec{v}$ brzina tela.
- **Energija** je, grubo rečeno, fizičko svojstvo koje je neophodno preneti na telo kako bi se na njemu izvršio *rad*. Striktnu definiciju ovih pojmova je izuzetno teško dati bez konteksta, pri čemu je takav poduhvat van obima ovih pripremnih materijala; pretpostavlja se da polaznici mogu prihvatiti ove pojmove "bez dokaza", s obzirom na to da su sa njima već upoznati u ranijem školovanju.
- **Kinetička energija** je energija koju telo poseduje usled kretanja. Kinetička energija za telo mase m, koje se kreće konstantnom brzinom V jednaka je $E_k = \dfrac{mV^2}{2}$.
- **Elastični sudar** je sudar dva tela u kome je njihova ukupna kinetička energija posle sudara istovetna sa njihovom ukupnom kinetičkom energijom pre sudara (odnosno, nema gubitka energije). Slično, **zakon održanja impulsa** nam, grubo rečeno, govori da će ukupan impuls posle sudara dva tela biti istovetan. Posmatrajmo dva tela *u jednoj dimenziji* (u cilju jednostavnosti), masa m_1 i m_2 respektivno, koji su pre sudara imali brzine $\vec{V_1}$ i $\vec{V_2}$. Označimo sa $\vec{V_1'}$ i $\vec{V_2'}$ brzine ovih tela nakon sudara. Na osnovu istovetnosti ukupnog impulsa, imamo:

$$m_1\vec{V_1} + m_2\vec{V_2} = m_1\vec{V_1'} + m_2\vec{V_2'}$$

Na osnovu istovetnosti ukupne kinetičke enrgije, imamo:

$$\frac{m_1V_1^2}{2} + \frac{m_2V_2^2}{2} = \frac{m_1V_1'^2}{2} + \frac{m_2V_2'^2}{2}$$

Rešavanjem ovog sistema dobijamo jedino netrivijalno rešenje:

$$V_1' = \frac{V_1(m_1 - m_2) + 2m_2V_2}{m_1 + m_2} \tag{2.1}$$

$$V_2' = \frac{V_2(m_1 - m_2) + 2m_1V_1}{m_1 + m_2} \tag{2.2}$$

U ovom slučaju, smerovi brzina ova dva tela će se samo obrnuti, dok će pravci ostati isti. Ako posmatramo dvodimenzionalni umesto jednodimenzionalni prostor, ovaj problem je nešto kompleksniji, s obzirom na to da će se i pravci brzina promeniti u zavisnosti od tačke sudara i normale na površinu: zamislite dve bilijarske kugle koje se sudaraju pod raznim uglovima i u raznim tačkama, i ovo će biti intuitivno jasno (ali ne manje teže formalno izračunati). Ovaj slučaj obradićemo na samom predavanju.