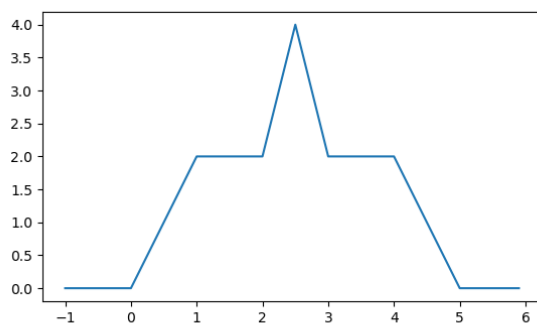


Zavod za elektroniku, mikroelektroniku,
računalne i inteligentne sustave

Duboko učenje

međuispit

1. Konstruirajte dvorazinski regresijski model koji kombinira affine transformacije i zglobnicu da bi aproksimirao funkciju zadanu slikom.



2. Napišite jednadžbu gubitka negativne log-izglednosti za grupu od N podataka. Izvedite izraz za gradijente tog gubitka s obzirom na ulaz softmaxa u svim podacima grupe.

Unakrsnu entropiju predikcije modela u podatku \mathbf{x}_i s obzirom na željenu distribuciju \mathbf{y}_i možemo izraziti jednadžbom: $\mathcal{L}_{CE} = -\sum_{j=1}^C y_{ij} \log p(Y = y_{ij} | \mathbf{x}_i)$. Odredite gradijent gubitka \mathcal{L}_{CE} s obzirom na ulaz softmaxa u podatku \mathbf{x}_i .

3. Razmatramo klasifikacijski duboki model koji podatak \mathbf{x} preslikava u distribuciju \mathbf{p} :

$$\begin{aligned} \mathbf{h}_A &= \text{ReLU}(\mathbf{W}_A \cdot \mathbf{x} + \mathbf{b}_A) \\ \mathbf{s} &= \mathbf{W}_B \cdot \mathbf{h}_A + \mathbf{b}_B \\ \mathbf{p} &= \text{softmax}(\mathbf{s}) \end{aligned}$$

- (a) Odredite izlaze mreže ako je na ulazu zadan podatak (2,4), a početni parametri su: $\mathbf{W}_A = \begin{bmatrix} -1 & 1 \end{bmatrix}$, $\mathbf{b}_A = [-1]$, $\mathbf{W}_B = \begin{bmatrix} 1 & -1 \end{bmatrix} \cdot \ln(2)$, $\mathbf{b}_B = \begin{bmatrix} 1 & 1 \end{bmatrix}$
- (b) odredite gubitak negativne log-izglednosti ako je podatak označen distribucijom $\mathbf{y} = (0,1)$;
- (c) odredite gradijente gubitka negativne log-izglednosti s obzirom na težine \mathbf{W}_A i \mathbf{W}_B ;
- (d) odredite parcijalnu derivaciju $\partial \mathbf{s} / \partial \text{vec}(\mathbf{W}_B)$ i komentirajte svrsishodnost njenog korištenja u praksi; napomena: $\text{vec}(\mathbf{W}_B)$ označava vektor sa svim elementima tenzora \mathbf{W}_B ;
- (e) komentirajte zalihost modela: koji dio modela bi se mogao izostaviti bez utjecaja na kapacitet?

4. Razmatramo konvolucijski duboki model koji 2D tenzor \mathbf{x} preslikava u vjerojatnost p :

```

$$\begin{aligned} \mathbf{h}_A &= \text{ReLU}(\text{conv}(\mathbf{x}, \mathbf{w}_A) + \mathbf{b}_A) \\ \mathbf{h}_B &= \text{conv}(\mathbf{h}_A, \mathbf{w}_B) + \mathbf{b}_B \\ s &= \text{GMP}(\mathbf{h}_B) \\ p &= \sigma(s \cdot \ln(2)) \end{aligned}$$

```

Pri tome ReLU označava zglobnicu, conv – konvoluciju bez nadopunjavanja, a GMP – globalno sažimanje maksimalnom vrijednošću. Tenzor kojeg proizvodi GMP ima prostorne dimenzije 1x1, dok broj kanala izlaza odgovara broju kanala ulaza.

Tenzor \mathbf{w}_A ima oblik 2x1x3x3. Prve dvije dimenzije odgovaraju broju kanala izlaza i ulaza. Posljednje dvije dimenzije odgovaraju retcima i stupcima po kojima provodimo konvoluciju i sažimanje. Početne vrijednosti elemenata \mathbf{w}_A su: $\mathbf{w}_{A0010}=-1$, $\mathbf{w}_{A0011}=1$, $\mathbf{w}_{A1021}=-1$, $\mathbf{w}_{A1011}=1$, a svi ostali elementi su 0.

Tenzor \mathbf{w}_B ima oblik 1x2x1x1 te jednaku organizaciju kao i \mathbf{w}_A . Početne vrijednosti elemenata \mathbf{w}_B su: $\mathbf{w}_{B0000} = -1$, $\mathbf{w}_{B0100} = 1$, a svi ostali elementi su 0.

Početne vrijednosti svih elemenata pomaka \mathbf{b}_A i \mathbf{b}_B su 0.

Ulazni tenzor \mathbf{x} ima oblik 4x4, $\mathbf{x}_{11}=\mathbf{x}_{12}=\mathbf{x}_{22}=1$, a svi ostali elementi su 0. Točan razred podatka $y=1$.

Odredite izlaz modela te izračunajte gradijente negativne log-izglednosti s obzirom na elemente konvolucijske jezgre \mathbf{w}_B .

Svi indeksi počinju od nule.

BONUS: izračunajte gradijente negativne log-izglednosti s obzirom na elemente konvolucijske jezgre \mathbf{w}_A .

5. Oblikujte funkciju koja implementira optimizacijski algoritam ADAM. Napišite ispitni kod koji uz pomoć te funkcije pronalazi minimum funkcije $x^4 - 4x$ tijekom 10000 iteracija, počevši iz $x=-2$. Zadatak riješite prema sljedećim uputama.

- Neka vas ne zbuni što se nigdje ne spominju podatci: njih smo izostavili kako bi zadatak bio jednostavniji;
- tražena funkcija treba raditi samo za skalarne funkcije skalarne varijable;
- neka argumenti funkcije budu:
 - početna vrijednost parametra (`theta`),
 - funkcija koja računa gradijent (`fgrad`), te
 - broj iteracija (`n`);
- neka povratna vrijednost funkcije bude konačna vrijednost parametra;
- nemojte komplicirati: naša implementacija ima 20 redaka Pythona od čega 5 redaka otpada na inicijalizaciju hiperparametara.

6. Razmatramo sloj dubokog modela koji ulazni vektor \mathbf{p} transformira u izlazni vektor \mathbf{q} koristeći skalarne parametre w_0 , w_1 i w_2 , a može se opisati jednadžbom: $q_i = w_0 \cdot p_{i-1} + w_1 \cdot p_i + w_2 \cdot p_{i+1}$.

Odredite Jakobijan gubitka po ulazu i Jakobijan gubitka po parametrima ako je poznat Jakobijan gubitka po izlazu.

Napišite kod koji bi omogućio uklapanje sloja u duboki model proizvoljne složenosti. Sloj izrazite razredom koji implementira sučelje Layer iz prvog zadatka druge laboratorijske vježbe te implementira sljedeće metode: `forward(self, inputs)`, `backward_inputs(self, grads)`, te `backward_parameters(self, grads)`.

Izvedba treba osigurati da dimenzionalnost izlaznog vektora bude jednaka dimenzionalnosti ulaznog vektora.

Pomoć: numpyjev tenzor \mathbf{x} proizvoljne dimenzionalnosti možemo nadopuniti nulama sljedećim pozivom: `np.lib.pad(x, 1, mode='constant')`.