Duboko učenje - Auditorne vježbe

Josip Šarić

Fakultet elektrotehnike i računarstva
Sveučilište u Zagrebu
Zagreb, Hrvatska

31. ožujka 2023.

Zadatak 1

Razmatramo klasifikacijski model s arhitekturom:

- konvolucijski sloj bez nadopunjavanja: 16 jezgara 5×5, korak 1, aktivacija ReLU;
- sažimanje maksimumom 2×2;
- konvolucijski sloj bez nadopunjavanja: 32 jezgre 5×5, korak 1, aktivacija ReLU;
- sažimanje maksimumom 2×2;
- pretvaranje u vektor;
- potpuno povezani sloj dimenzije 10 te aktivacija softmaks;

Na ulaz mreže dovodimo RGB slike veličine 28×28 piksela.

Zadatak 1

1. Napišite dimenzije ulaznih i izlaznih tenzora za svaki sloj.
2. Koliko ukupno parametara ima zadani model?
3. Procijenite memorijsko zauzeće u trenutku nakon izračuna gubitka unakrsne entropije. Postupak učenja zadanog modela provodimo s veličinom mini-grupe 8. Tip podataka je 32-bitni float. Pomoć: u memorijsko zauzeće ubrojiti sve u tome trenutku pohranjene tenzore i parametre modela.
4. Kolika su receptivna polja aktivacija u prvom sloju sažimanja odnosno aktivacija u potpuno povezanom sloju? Pomoć: receptivno polje računamo s obzirom na piksele ulazne slike.
5. Implementirajte model u Pytorchu.

Zadatak 1 - rješenje 1.1 i 1.2

Sloj	Dimenzija izlaza	Broj paramatera
Ulaz	$28 \times 28 \times 3$	0
Konv1 ($c_{out} = 16,\ k = 5 \times 5$)	$24 \times 24 \times 16$	$5 \cdot 5 \cdot 3 \cdot 16 + 16$
ReLU	$24 \times 24 \times 16$	0
Maxpool1 ($k = 2 \times 2$)	$12 \times 12 \times 16$	0
Konv2 ($c_{out} = 32,\ k = 5 \times 5$)	$8 \times 8 \times 32$	$5 \cdot 5 \cdot 16 \cdot 32 + 32$
ReLU	$8 \times 8 \times 32$	0
Maxpool2 ($k = 2 \times 2$)	$4 \times 4 \times 32$	0
Pretvaranje u vektor	512	0
Potpuno povezani (10)	10	$512 \cdot 10 + 10$
Softmaks	10	0
		19178

Zadatak 1 - rješenje 1.3

Parametri: $19178 \cdot 4 = 76712\mathbf{B}$

Zbog unatražnog prolaza moramo zapamtiti ulaze svakog sloja (osim pretvaranja u vektor) zasebno za svaki primjerak minigrupe:
$8 \cdot (28 \cdot 28 \cdot 3 + 24 \cdot 24 \cdot 16 + 24 \cdot 24 \cdot 16 + 12 \cdot 12 \cdot 16 + 8 \cdot 8 \cdot 32 + 8 \cdot 8 \cdot 32 + 512 + 10 + 10) \cdot 4 = 886912\mathbf{B}$

Ukupno: $963624\mathbf{B} \approx 0.9\mathbf{MiB}$

Zadatak 1 - rješenje 1.3

Parametri: $19178 \cdot 4 = 76712\mathbf{B}$

Zbog unatražnog prolaza moramo zapamtiti ulaze svakog sloja (osim pretvaranja u vektor) zasebno za svaki primjerak minigrupe:
$8 \cdot (28 \cdot 28 \cdot 3 + 24 \cdot 24 \cdot 16 + 24 \cdot 24 \cdot 16 + 12 \cdot 12 \cdot 16 + 8 \cdot 8 \cdot 32 + 8 \cdot 8 \cdot 32 + 512 + 10 + 10) \cdot 4 = 886912\mathbf{B}$

Ukupno: $963624\mathbf{B} \approx 0.9\mathbf{MiB}$

Zadatak 1 - rješenje 1.4

Receptivno polje prvog konvolucijskog sloja: 5×5

Receptivno polje prvog sloja sažimanja: 6×6
(jer vidi dvije susjedne aktivacije konvolucijskog sloja s korakom 1)

Receptivno polje potpuno povezanog sloja: cijela slika! (28×28)

Zadatak 1 - rješenje 1.5

```
class MyModel(nn.Module):
def __init__(self):
    super().__init__()
    self.seq = nn.Sequential(
        nn.Conv2d(3, 16, 5),
        nn.ReLU(),
        nn.MaxPool2d(kernel_size=2, stride=2),
        nn.Conv2d(16, 32, 5),
        nn.ReLU(),
        nn.MaxPool2d(kernel_size=2, stride=2),
        nn.Flatten(),
        nn.Linear(512, 10),
        nn.Softmax(dim=-1)
    )
def forward(self, x):
    return self.seq(x)
def loss(self, x, y_oh):
    probs = self.forward(x)
    return -(probs.log() * y_oh).sum(-1).mean()
```

Zadatak 2

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 x proizvoljne dimenzionalnosti možemo nadopuniti nulama sljedećim pozivom: `np.lib.pad(x, 1, mode='constant')`.

Zadatak 2 - Jakobijan izlaza po ulazu

$$\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{p}} = \begin{bmatrix} w_1 & w_2 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ w_0 & w_1 & w_2 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & w_0 & w_1 & w_2 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & w_0 & w_1 & w_2 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & w_0 & w_1 & w_2 & 0 & \dots & 0 \\ & & & & \dots & & & & \\ 0 & 0 & 0 & 0 & \dots & 0 & w_0 & w_1 & w_2 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & w_0 & w_1 \end{bmatrix} \tag{1}$$

$$\tag{2}$$

Zadatak 2 - Jakobijan gubitka po ulazu

$$\frac{\partial L}{\partial \boldsymbol{p}} = \frac{\partial L}{\partial \boldsymbol{q}} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{p}} = \left[\frac{\partial L}{\partial q_0}, \frac{\partial L}{\partial q_1}, \frac{\partial L}{\partial q_2}, \frac{\partial L}{\partial q_3}, \dots, \frac{\partial L}{\partial q_{n-1}} \right] .$$

$$\begin{bmatrix} w_1 & w_2 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ w_0 & w_1 & w_2 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & w_0 & w_1 & w_2 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & w_0 & w_1 & w_2 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & w_0 & w_1 & w_2 & 0 & \dots & 0 \\ & & & & \dots & & & & \\ 0 & 0 & 0 & 0 & \dots & 0 & w_0 & w_1 & w_2 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & w_0 & w_1 \end{bmatrix}$$

Skalarni umnošci vektora retka $\frac{\partial L}{\partial \boldsymbol{q}}$ sa stupcima matrice $\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{p}}$ podsjećaju na konvoluciju. Doista, nadopunjavanjem to i postižemo...

Zadatak 2 - Jakobijan gubitka po ulazu

$$\frac{\partial L}{\partial \boldsymbol{p}} = \left[0, \frac{\partial L}{\partial q_0}, \frac{\partial L}{\partial q_1}, \frac{\partial L}{\partial q_2}, \frac{\partial L}{\partial q_3}, ..., \frac{\partial L}{\partial q_{n-1}}, 0\right] \cdot \begin{bmatrix} w_2 & 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 \\ w_1 & w_2 & 0 & 0 & 0 & 0 & 0 & ... & 0 \\ w_0 & w_1 & w_2 & 0 & 0 & 0 & 0 & ... & 0 \\ 0 & w_0 & w_1 & w_2 & 0 & 0 & 0 & ... & 0 \\ 0 & 0 & w_0 & w_1 & w_2 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & w_0 & w_1 & w_2 & 0 & ... & 0 \\ & & & & ... & & & & \\ 0 & 0 & 0 & 0 & ... & 0 & w_0 & w_1 & w_2 \\ 0 & 0 & 0 & 0 & ... & 0 & 0 & w_0 & w_1 \\ 0 & 0 & 0 & 0 & ... & 0 & 0 & 0 & w_0 \end{bmatrix}$$

$$= \mathrm{pad}(\frac{\partial L}{\partial \boldsymbol{q}}, 1, \mathrm{val} = 0) \star \mathrm{flip}(\boldsymbol{w})$$

Jakobijan izlaza po parametrima

$$\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{w}} = \begin{bmatrix} 0 & p_0 & p_1 \\ p_0 & p_1 & p_2 \\ p_1 & p_2 & p_3 \\ p_2 & p_3 & p_4 \\ p_3 & p_4 & p_5 \\ & \dots & \\ p_{n-2} & p_{n-1} & 0 \end{bmatrix}$$

Jakobijan gubitka po parametrima

$$
\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{w}} &= \frac{\partial L}{\partial \boldsymbol{q}} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{w}} \\
&= \left[\frac{\partial L}{\partial q_0}, \frac{\partial L}{\partial q_1}, \frac{\partial L}{\partial q_2}, \frac{\partial L}{\partial q_3}, ..., \frac{\partial L}{\partial q_{n-1}}\right] \cdot \begin{bmatrix} 0 & p_0 & p_1 \\ p_0 & p_1 & p_2 \\ p_1 & p_2 & p_3 \\ p_2 & p_3 & p_4 \\ p_3 & p_4 & p_5 \\ & ... & \\ p_{n-2} & p_{n-1} & 0 \end{bmatrix}
\end{aligned}
$$

Ovaj umnožak također možemo izraziti unakrsnom korelacijom:
$\frac{\partial L}{\partial \boldsymbol{w}} = \mathrm{pad}(\boldsymbol{p}, 1, \mathrm{val} = 0) \star \frac{\partial L}{\partial \boldsymbol{q}}$

Zadatak 2 - kod

```
class Layer:
  def forward(self, p):
    self.p = p
    p2 = np.lib.pad(p, 1, 'constant')
    q = (p2[0:p.shape[0]] * self.w0 +
         p * self.w1 +
         p2[2:p.shape[0]+2] * self.w2)
    return q

  def backward_inputs(self, dq):
    dq2 = np.lib.pad(dq, 1, 'constant')
    dp = dq * self.w1 + dq2[0:dq.shape[0]] * self.w2 + dq2[2:dc
    return dp

  def backward_params(self, dq):
    dq2 = np.lib.pad(dq, 1, 'constant')
    dw0 = np.dot(self.p, dq2[2:dq.shape[0]+2])
    dw1 = np.dot(self.p, dq)
    dw2 = np.dot(self.p, dq2[0:dq.shape[0]+0])
    return np.array((dw0,dw1,dw2))
```

Zadatak 2 - kod

```
def forward_alt(self, p):
    self.p = p
    p2 = np.lib.pad(p, 1, 'constant')
    q = np.zeros_like(p)
    for i in range(len(q)):
        q[i] = p2[i] * self.w0
               + p2[i+1] * self.w1
               + p2[i+2] * self.w2
    return q
```

Zadatak 3

Razmatramo klasifikacijski model kojeg opisuju sljedeće jednadžbe:

$$\boldsymbol{h} = \text{ReLU}(\boldsymbol{W} \cdot \boldsymbol{x} + \boldsymbol{b}) \tag{3}$$

$$\boldsymbol{h}_2 = \text{ReLU}(\boldsymbol{W}_2 \cdot \boldsymbol{h} + \boldsymbol{b}_2) \tag{4}$$

$$\boldsymbol{s} = \text{softmax}(\boldsymbol{W}_3 \cdot \boldsymbol{h}_2 + \boldsymbol{b}_3 + \boldsymbol{h}) \tag{5}$$

Početne vrijednosti parametara modela su: $\boldsymbol{W} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $\boldsymbol{W}_2 = \boldsymbol{W}_3 = \mathbf{I}$, $\boldsymbol{b} = \boldsymbol{b}_2 = \boldsymbol{b}_3 = \mathbf{0}$. Na ulaz modela dovodimo podatak $\boldsymbol{x} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^\top$ koji bi se trebao klasificirati u razred $y = 0$. Gubitak je negativna log-izglednost.

1. Provedite unaprijedni prolaz i odredite predikciju modela.
2. Odredite gubitak.
3. Odredite gradijente po $\boldsymbol{W}$.

Zadatak 3 - rješenje

Unaprijedni prolaz:

$$\boldsymbol{x} = [1, 2, 3]^\mathsf{T} \tag{6}$$

$$\boldsymbol{a} = \boldsymbol{W} \cdot \boldsymbol{x} + \boldsymbol{b} = [3, 3]^\mathsf{T} \tag{7}$$

$$\boldsymbol{h} = \mathsf{ReLU}(\boldsymbol{a}) = [3, 3]^\mathsf{T} \tag{8}$$

$$\boldsymbol{a}_2 = \boldsymbol{W}_2 \cdot \boldsymbol{h} + \boldsymbol{b}_2 = [3, 3]^\mathsf{T} \tag{9}$$

$$\boldsymbol{h}_2 = \mathsf{ReLU}(\boldsymbol{a}_2) = [3, 3]^\mathsf{T} \tag{10}$$

$$\boldsymbol{s} = \boldsymbol{W}_3 \cdot \boldsymbol{h}_2 + \boldsymbol{b}_3 + \boldsymbol{h} = [6, 6]^\mathsf{T} \tag{11}$$

$$\boldsymbol{p} = \mathsf{softmax}(\boldsymbol{s}) = [0.5, 0.5]^\mathsf{T} \tag{12}$$

Gubitak:

$$L = -\ln \mathrm{p}(\underline{y} = 0 \mid \boldsymbol{x}, \boldsymbol{\theta}) = -\ln(\boldsymbol{p}_{[0]}) = \ln(2) \approx 0.693$$

Zadatak 3 - rješenje

Gradijent:

$$\frac{\partial L}{\partial \boldsymbol{s}} = \boldsymbol{p} - \mathbf{y}_{\mathrm{oh}} = [0.5, 0.5] - [1, 0] = [-0.5, 0.5] \tag{13}$$

$$\frac{\partial L}{\partial \boldsymbol{h}_2} = \frac{\partial L}{\partial \boldsymbol{s}} \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{h}_2} = \frac{\partial L}{\partial \boldsymbol{s}} \boldsymbol{W}_3 \tag{14}$$

$$= [-0.5, 0.5] \tag{15}$$

$$\frac{\partial L}{\partial \boldsymbol{a}_2} = \frac{\partial L}{\partial \boldsymbol{h}_2} \frac{\partial \boldsymbol{h}_2}{\partial \boldsymbol{a}_2} = [-0.5, 0.5] \tag{16}$$

$$\frac{\partial L}{\partial \boldsymbol{h}} = \frac{\partial L}{\partial \boldsymbol{s}} \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{h}} = \frac{\partial L}{\partial \boldsymbol{a}_2} \boldsymbol{W}_2 + \frac{\partial L}{\partial \boldsymbol{s}} \tag{17}$$

$$= [-0.5, 0.5] + [-0.5, 0.5] = [-1, 1] \tag{18}$$

$$\frac{\partial L}{\partial \boldsymbol{a}} = \frac{\partial L}{\partial \boldsymbol{h}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{a}} = [-1, 1] \tag{19}$$

Zadatak 3 - rješenje

Gradijent:

$$\frac{\partial L}{\partial \boldsymbol{W}_{[i,\cdot]}} = \frac{\partial L}{\partial \boldsymbol{a}} \frac{\partial \boldsymbol{a}}{\partial \boldsymbol{W}_{[i,\cdot]}} = \frac{\partial L}{\partial \boldsymbol{a}_{[i]}} \frac{\partial \boldsymbol{a}_{[i]}}{\partial \boldsymbol{W}_{[i,\cdot]}} \tag{20}$$

$$= \frac{\partial L}{\partial \boldsymbol{a}_{[i]}} \boldsymbol{x}^{\mathsf{T}} \tag{21}$$

$$\frac{\partial L}{\partial \boldsymbol{W}_{[1,\cdot]}} = -1 \cdot [1, 2, 3] \tag{22}$$

$$\frac{\partial L}{\partial \boldsymbol{W}_{[2,\cdot]}} = 1 \cdot [1, 2, 3] \tag{23}$$

$$\nabla_{\boldsymbol{W}} L = \begin{bmatrix} -1 & -2 & -3 \\ 1 & 2 & 3 \end{bmatrix} \tag{24}$$