

Duboko učenje

provjera znanja 3. laboratorijske vježbe

1. Razmatramo običnu povratnu neuronsku mrežu.
 - (a) Odredite broj parametara jednoslojne mreže kao funkciju veličine skrivenog stanja h i dimenzije ulazne vektorske reprezentacije e . Koristeći navedenu formulu, odredite broj parametara za mrežu s skrivenom reprezentacijom veličine 10 i dimenzijom ulazne vektorske reprezentacije 5.
 - (b) Proširite prethodnu formulu da funkcionira i za višeslojne povratne neuronske mreže. Odredite broj parametara dvoslojne mreže s istim hiperparametrima kao i u prethodnom podzadatku.
 - (c) Implementirajte unaprijedni prolaz jednoslojne i jednosmjerne obične povratne neuronske mreže kao metodu “forward” podrazreda “nn.Module” u Pytorchu koristeći isključivo osnovne operacije matričnog množenja, zbrajanja i nelinearnosti. Pretpostavite da su ulazi x metode “forward” niz vektorskih reprezentacija dimenzija $T \times B \times E$
2. Iduća pitanja tiču se pojašnjenja implementacijskih detalja laboratorijske vježbe.
 - (a) Kao komponentu vježbe trebalo je implementirati tzv. “collate” funkciju. Objasnite zašto nam je one nužna u problemima analize teksta.
 - (b) Kao komponentu vježbe trebalo je implementirati razred vokabulara, koji je mapirao riječi na njihove indekse u riječniku. Za te riječi smo potom učitali prednaučene vektorske reprezentacije. Što se događa s vektorskim reprezentacijama riječi koje se ne pojavljuju u datoteci s prednaučenim reprezentacijama? Što su to posebni znakovi i čemu služe? Koje smo posebne znakove koristili u okviru vježbe i kako se inicijaliziraju njihove reprezentacije?
3. Razmatramo zadatak odgovaranja na pitanja (engl. *Question Answering, QA*) u kojemu na ulazu imamo dva niza riječi. Za dobivanje gustih vektora riječi oba niza ćemo koristiti istu ugradbenu matricu (za koju možete pretpostaviti da je već inicijalizirana *embeddings*), dok ćemo za svaki niz koristiti zasebnu LSTM mrežu. Skrivena stanja iz zadnjeg vremenskog koraka obje LSTM mreže konkatenirajte te predajte u linearni sloj s jednim izlazom.

Nadopunite konstruktor te implementirajte unaprijedni prolaz klasifikatora sličnosti rečenica temeljenog na dvije višeslojne LSTM mreže. Pri implementaciji ne morate koristiti razred “PackedSequence”.

```
class QAClassifier(nn.Module):
    def __init__(self, embeddings, ...):
        super(QAClassifier, self).__init__()
        self.embeddings = embeddings
        self.rnn_q = nn.LSTM(...)
        self.rnn_a = nn.LSTM(...)
        self.decoder = nn.Linear(...)

    def forward(self, q, a):
        '''Performs forward pass
        Assume both Q and A are TxB sequences of word indices
        '''
        pass
```