SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Autoenkoderi

Renato Jurišić
Voditelj: *izv. prof. dr. sc. Zoran Kalafatić*

Zagreb, siječanj 2023.

SADRŽAJ

1. Uvod

Autoenkoderi su neuronske mreže kojima modeliramo funkciju identita. Naizgled, imaju jednostavan zadatak. Ono što dobiju na ulazu trebaju proslijediti na izlaz. Taj zadatak je zapravo trivijalan, pa se uvodi dodatan zahtjev kako bi otežali stvari. Mreža prvo treba smanjiti dimenziju podataka, tj. prvo treba naučiti kompresirati podatke, pa zatim pomoću tih kompresiranih podataka treba rekonstruirati originalni ulaz. Zbog tog uvjeta mreža ne može trivijalno samo prepisati svoje ulaze na izlaze, nego treba naći neku valjanu i efikasnu reprezentaciju podataka. Autoenkoderi su sposobni samostalno (ne nadziranim učenjem) naučiti guste reprezentacije podataka, zbog čega se mogu koristiti za smanjenje dimenzijalnosti, učenje značajki te generativno modeliranje.

Kompresirani podaci u među koraku, između enkodiranja i dekodiranja, su tipično manje dimenzije od originalnih podataka zbog čega smanjenje dimenzijalnosti dolazi prirodno iz autoenkodera. Npr. ako je ulaz vektor dimenzije 100 onda će kodirani vektor biti neke manje dimenzije npr. 10.

Dobiveno kodiranje sadrži samo najbitnije uzorke i informacije iz podataka. Drugim riječima, potrebno je pronaći značajke koje najbolje opisuju podatke. Npr. ako radimo enkoder za slike, onda mreža mora naučiti prepoznati oblike i strukture lica koje bi se kasnije mogle koristiti za sistem koji prepoznaje lica.

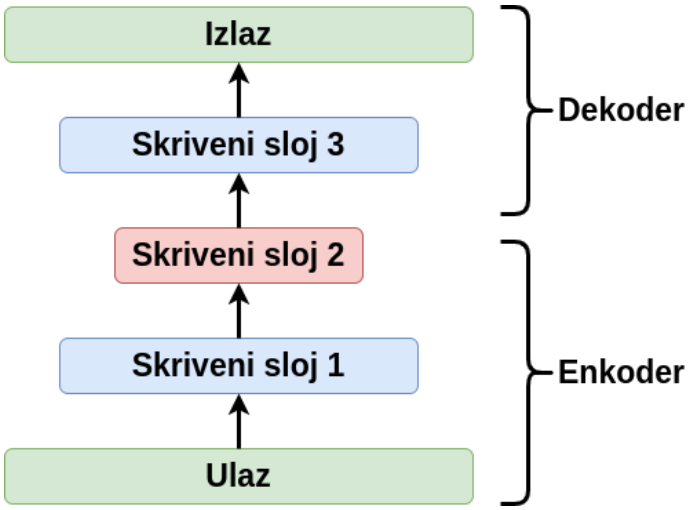
Autoenkoderi se mogu koristiti i kako bi se dobili generativni modeli, gdje je cilj naučiti model koji može generirati nove podatke koji su slični podacima na kojima je model treniran. Npr. model koji je učen na slikama s psima, mogao bi naučiti generirati nove slika pasa koji su realistične i izgledaju kao da bi mogle biti iz stvarnog života.

2. Naslagani autoenkoderi

Naslagani autoenkoder podijeljen je u dva djela. Prvi dio čini niz slojeva naslagani na način da je svaki sljedeći niz manje dimenzijalnosti od prethodnog. Taj prvi dio naziva se enkoder. Takvo ime je dobio jer jer njegova zadaća zapravo naučiti efikasan kod, tj. reprezentaciju ulaznih podataka.

Drugi dio naziva se dekoder. Po imenu se može zaključiti da će njegova zadaća biti upravo suprotna od enkodera. On će trebati iz dobivenog koda rekonstruirati početne ulaze. Dekoder se također sastoji od niza slojeva, ali ovaj puta su naslagani tako da je svaki sljedeći sloj veće dimenzije od prethodnog. Cilj je da svaki sloj radi inverznu operaciju od njemu pripadnog sloja u enkoderu.

Kako su enkoder i dekoder sebi međusobno inverzni, arhitektura autoenkodera je simetrična s obzirom na srednji sloj (Slika 2.1). Izgled podsjeća na pješčani sat. Autoenkoder može imati proizvoljno mnogo skrivenih slojeva, ali dodavanjem dodatnih slojeva povećavamo kompleksnost značajki, tj. kodiranja. Stoga se mora pripaziti da se autoenkoder ne učini pre snažnim, zbog čega će loše generalizirati.



Slika 2.1: Arhitektura autoenkodera

3. Implementacija

U svrhu demonstracije rada autoenkodera treniran je jedan autoenkoder na "Fashion MNIST" skupu podataka. "Fashion MNIST" skup podataka sastoji se od 70000 slika veličine 28x28. Slike predstavljaju odjevne predmete i grupirane su u 10 klasa. Kako se radi sa slikama prirodno se postavlja da koristimo konvolucijske slojeve koji su namijenjeni izvlačenju značajki iz slika. Autoenkoder koji ima konvolucijske slojeve naziva se konvolucijski autonekoder.

3.1. Model

Enkoder konvolucijskog autoenkodera sastoji se od niza konvlucijski slojeva, nelinearnih aktivacijskih funkcija te slojeva sažimanja. Enkoder tipično smanjuje širinu i visinu slike, a povećava dubinu. Dekoder onda mora raditi obrnuto. Smanjuje dubinu, a povećava širinu i visinu. Takve transformacije mogu se raditi "transponiranim konvolucijskim slojevima".

Ovisno o odabiru funkcije pogreške, ulazi i izlazi moraju biti jednakih dimenzija. Zbog toga je potrebno pažljivo birati parametre poput veličine kernela, iskoraka (engl. stride), nadopunjavanja (engl. padding) itd. U konačnici je dizajniran model sa slike (Slika 3.1). Kako se radi o slikama malih dimenzija model je relativno plitak, tj. niske složenosti.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 16, 28, 28]	160
ReLU-2	[-1, 16, 28, 28]	0
MaxPool2d-3	[-1, 16, 14, 14]	0
Conv2d-4	[-1, 32, 14, 14]	4,640
ReLU-5	[-1, 32, 14, 14]	0
MaxPool2d-6	[-1, 32, 7, 7]	0
Conv2d-7	[-1, 64, 7, 7]	18,496
ReLU-8	[-1, 64, 7, 7]	0
MaxPool2d-9	[-1, 64, 3, 3]	0
ConvTranspose2d-10	[-1, 32, 7, 7]	18,464
ReLU-11	[-1, 32, 7, 7]	0
ConvTranspose2d-12	[-1, 16, 14, 14]	2,064
ReLU-13	[-1, 16, 14, 14]	0
ConvTranspose2d-14	[-1, 1, 28, 28]	65
ReLU-15	[-1, 1, 28, 28]	0

Total params: 43,889
Trainable params: 43,889
Non-trainable params: 0

Slika 3.1: Konvolucijski autoenkoder

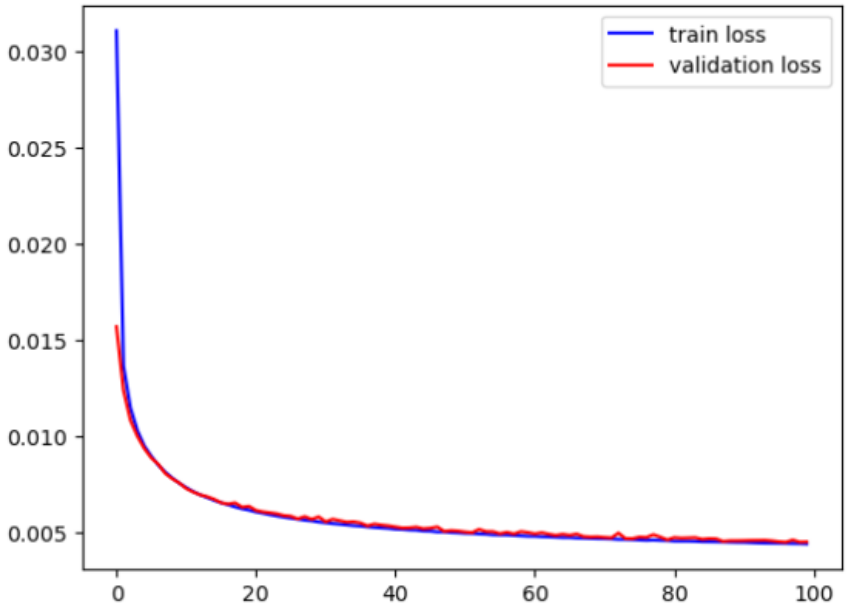
3.2. Treniranje

Za funkciju gubitka odabrana je prosječna kvadratna pogreška (Jednadžba 3.1). Dakle, za svaki piksel računa se kvadrat razlike originalnog piksela i rekonstruiranog piksela, zatim se to sve zbroj i podijeli s brojem piksela. Iz ovog načina računanja funkcije pogreške proizlazi uvjet da ulaz i izlaz mora biti iste dimenzije. Svaki ulazni piksel mora imati pripadajući izlazni piksel kako bi se mogla izračunati njihova razlika.

$$L = \frac{1}{28 \cdot 28N} \sum_{i=1}^{N} \sum_{j=1,k=1}^{28} (x_{j,k}^{i} - y_{j,k}^{i})^2 \tag{3.1}$$

Za optimizator je odabran optimizator "Adam" s predpostavljenim vrijednostima u radnom okviru PyTorch (Contributors (2022)). U jednom koraku uzeto je 128 slika (engl. batch size) te je model treniran 100 epoha.

Nije se radila optimizacija hiperparametara jer nije zapažena prenaučenost, a dobiveni model radi zadovoljavajuće dobro. Nakon 100 epoha greška na validacijskom skupu podataka bila je zanemarivo veća od greške na skupu za učenje (Slika 3.2).



Slika 3.2: Funkcija pogreške na skupu za učenje i skupu za provjeru tijekom epoha

3.3. Rezultati

Rekonstruirane slike su prepoznatljive minus dodatan šum u slikama (Slika 3.3). Prema potrebi, model bi se mogao dodatno ojačati tako da ga učinimo dubljim ili da povećamo dimenziju enkodiranja ili da treniramo model više epoha.

Slika 3.3: Rekonstrukcija podataka. Originalni primjeri (prvi red). Rekonstruirani primjeri (drugi red)

4. Primjene

4.1. Smanjenje dimenzijalnosti

Autoenkoderi se mogu koristiti za smanjenje dimenzijalnosti tako da naučene značajke skrivenih slojeva interpretiramo kao da su kompresirana reprezentacija ulaza. Ti skriveni slojevi su tipično manje dimenzije nego ulazni slojevi, pa kažemo da smo smanjili dimenziju podataka.

Npr. ako imamo podatke dimenzije 100. Onda možemo trenirati autoenkoder čiji skriveni sloj ima dimenziju 10 i zatim interpretirati izlaze iz tog skrivenog sloja kao kompresiranu reprezentaciju ulaza. Time se efektivno dobije mapiranje podataka iz dimenzije 100 u dimenziju 10.

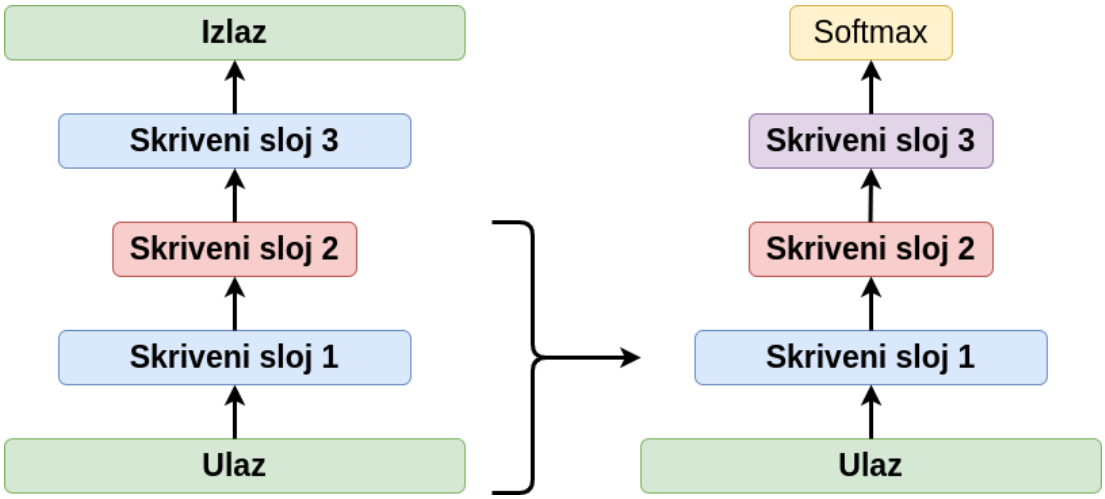
Kod koji proizvede autoenkoder je tipično bolji od kodova koji se dobiju nekim drugim tehnikama smanjenja dimenzijalnosti kao što je PCA. Razlog tomu je što autoenkoder uči kod kako bi rekonstruirao podatke, pa taj kod sadrži najbitnije značajke i uzorke, tj. sadrži više informacija.

4.2. Prijenos učenja

Nije rijetka pojava da u praksi imamo puno neoznačenih, a malo označenih podatka. Za označavanje podataka potrebno je uložiti ljudske sate i novac kako bi oznake bile kvalitetne. Svejedno, i da to napravimo možemo dobiti svega par tisuća označenih primjera. S druge strane, neoznačenih primjera ima gomila. Potrebno je samo implementirati neku skriptu koja će skidati podatke s internata.

Kada se nađemo u takvim situacijama korisno je pretražiti dostupne modele koji rješavaju sličan problem kao naš kako bi mogli iskoristiti donje slojeve u našem modelu. Donji slojevi će biti sposobni prepoznati značajke koje su nama potrebne i samo na njih dodamo još par novih slojeva kako bi uštimali mrežu za naš problem.

Ako ne uspijemo pronaći sličan model, onda se može trenirati autoenkoder. Zato jer autoenkoder gradi efikasno kodiranje podataka naučiti će prepoznavat značajke koje su najbitnije. Onda se donji slojevi enkodera mogu iskoristiti za prijenos učenja (Slika 4.1). Ljepota ove metode je u tome što je učenje autoenkodera nenadzirano, pa možemo iskoritit gomilu ne označenih podataka koji su nam na raspolaganju. A zatim konačan model dodatno uštimamo označenim podacima.



Slika 4.1: Prijenos učenja

4.3. Generativni modeli

Autoenkodri se mogu koristiti za generiranje novih primjera koji su slični primjerima na kojim je model treniran. Generativni autoenkoderi baziraju se na predpostavci da primjeri dolaze iz neke pred definirane distribucije (npr. Gaussove distribucije) i onda kodiranje pokušava naučiti parametre te distribucije (npr. kod Gaussove je to srednju vrijednost μ i standardu devijaciju σ).

Zatim se nakon treniranja mogu semplirati neke točke iz naučenog prostora i njih provući kroz dekoder. Tako možemo dobiti nove primjere.

5. Zaključak

Autoenkoder je neuronska mreža koja je namijenja aproksimaciji funkcije identiteta s uvjetom smanjenja dimenzijalnosti podataka. Arhitektura ima oblik "pješčanog sata" te se sastoji se od dva dijela, enkodera i dekodera, koji su međusobno simetrični jer obavljaju inverzne operacija. Enkoder uči kompresirati podatake i sposoban je pronaći vrlo efikasan kod podataka, a dekoder onda dekompresira dobiveni kod. Dobiveni kod se kasnije može upotrijebiti za druge zadatke kao što su smanjenje dimenzijalnosti, prijenos učenja ili generiranje novih podataka.

Autoenkoderi su jako prilagodljiv i univerzalan alat zbog čega se primjenjuju preko različitih domena kao što su računalni vid ili procesiranje i analiza jezika. Dakle, autoenkoderi su moćan alat koji nam može pomoći naučiti značajke i uzorke iz podataka koje onda možemo iskoristiti za rješavanje niza zadataka.

6. Literatura

PyTorch Contributors. Pytorch documentation. `https://pytorch.org/docs/stable/`, 2022. Accessed: 2023-01-01.

Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2019.

OpenAI. ChatGPT. `https://chat.openai.com/chat`, 2022. Accessed: 2023-01-01.

Autoenkoderi

Sažetak

U ovom radu opisana je ideja iza autoenkodera i njihova arhitektura. Napravljena je implementacija autoenkodera za "Fashion MNIST" skup podataka te su demonstrirane njegove rekostrukcije slika. Napravljen je sažetak trenutnih primjena autoenkodera na probleme u industriji.