

KOMPRESIJA SLIKA HIBRIDIZACIJOM NEURONSKIH MREŽA I GENETSKOG ALGORITMA

Strahinja Mitrić

Matematički fakultet Univerziteta u Beogradu

171/2015

strahinjamitric123@gmail.com

Nikola Mandić

Matematički fakultet Univerziteta u Beogradu

291/2015

mandinikola@gmail.com

Lazar Jovanović

Matematički fakultet Univerziteta u Beogradu

163/2015

jlazar1996@gmail.com

Kratak pregled

Ovaj rad bavi se konstruisanjem jednog, relativno, manjeg autoenkodera za kompresovanje slika, iz određenog skupa međusobno sličnih podataka, uz korišćenje genetskog algoritma za pronalaženje što boljih parametara učenja.

1 Uvod

"Što ne mogu da rekreiram, to i ne razumem."

— Ričard Fajman

Uz sve veći porast primena mašinskog učenja postavlja se pitanje šta to naši programi uče i da li se to može smatrati istim učenjem kao što je ljudsko. Autoenkoderske mreže, vođene gore-citiranom idejom koriste nam da to i ispitamo i uverimo se u mogućnosti neuronskih mreža da generišu, pročišćavaju od šuma i kompresuju visokodimenzionalne podatke, a njihove varijacije u poslednje vreme pokazuju zapanjujuće rezultate na zadacima kreiranja agenata za igranje igara ([3]).

Ideja autoenkodera prvi put pomenuta je 1986. u članku Rumelhalta, Hiltone i Vilijsa koji se bavio detaljnom analizom propagacije unazad [1]. Narednih godina ideja se ponavljala u više naučnih radova. Članak iz 1989. Baldija i Hornika - "Neural networks and principal component analysis: 'Learning from examples without local minima'" [4] bliže upoznače čitaonice sa autoenkoderima, već tada ih opisavši kao 'uobičajen trik'. Značajan je takođe i Hintonov i Zemelov rad iz 1994. koji opisuje nove ciljeve treniranja autoenkodera koji im omogućuje i učenje nelinearnih, faktorijskih reprezentacija [2]. Svakako, teško je ikome dodeliti titulu "oca" ove računarske grane jer je literatura raznolika i terminologija je evoluirala tokom vremena. Značajno je primetiti da su sve trenutno korišćene varijacije autoenkodera definisane u poslednjih 10 godina.

1.1 Opis autoenkodera

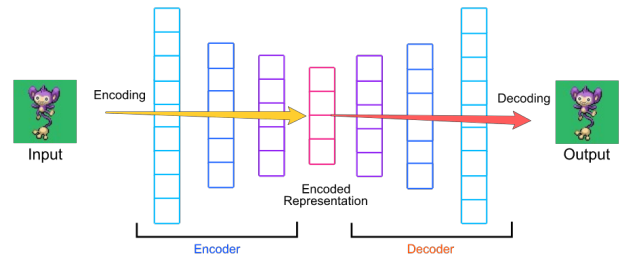
Autoenkoder se sastoji iz dva dela: enkodera i dekodera koji se mogu definisati kao funkcije ϕ i ψ takve da:

$$\phi : \mathcal{X} \rightarrow \mathcal{F}$$

$$\psi : \mathcal{F} \rightarrow \mathcal{X}$$

$$\phi, \psi = \arg \min_{\phi, \psi} \|X - (\psi \circ \phi)X\|^2$$

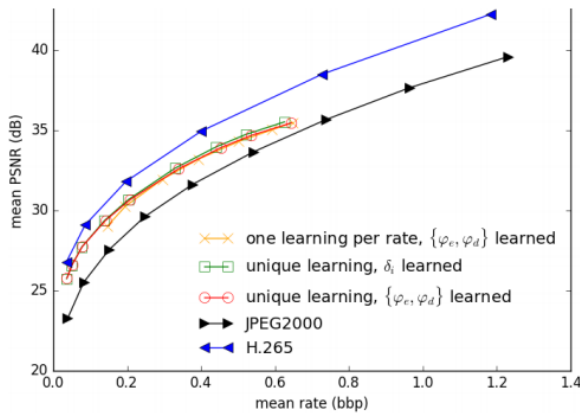
Enkoder uči da kompresuje podatke (tj. smanji im dimenzionalnost), dekode da dekompresuje (odnosno, povraća originalne podatke, sa što manjim gubitkom). Nakon treniranja, razdvajamo ih i koristimo zasebno. Savremene arhitekture umesto konkretnih podataka rade nad raspodelama iz kojih ti podaci dolaze i uče da generišu uzorke iz istih. "DVAE" autoenkodera (Disentangled Variational AutoEncoders) se razlikuju po tome što pretpostavljaju redundantnost u raspodeli koju uče, te minimizuju njen domen.



Slika 1: Uopšten prikaz autoenkodera

2 Povezani radovi

Na našu sreću radova na ovu temu bilo je pregršt u poslednjih nekoliko godina i imali smo dobru ideju o tome šta treba izvesti. Sa druge strane, nijedan (na koji smo naišli) nije koristio biblioteku koju smo mi namerili da koristimo, te nam je bilo prilično nejasno kako njeno korišćenje uklopiti u ceo projekat. Većina radova od nekoliko prethodnih godina pokazala je da se ovaj zadatak može uspešno izvesti na *lokalizovanom skupu podataka* i čak po rezultatima prevazići trenutne kompresije poput JPEG-a [6]. Zajedničko za sve radove koje smo našli, sem dobrih rezultata, bilo je korišćenje mnoštva konvolutivnih slojeva te je naša prvobitna ideja bila da se pozabavimo baš time uz minimalne varijacije na već ispitane arhitekture mreže. Rad na koji smo se najviše oslanjali na početku bio je "Creation of deep convolutional auto-encoder in Caffe" [7] od V. Turčenka i A. Lučaka.



Slika 2: Poređenje rešenja iz [6] sa JPEG kompresijom

3 O skupu podataka

Skup podataka korišćen za trening i test našeg modela je *UTKFace* [5], koji se sastoji od malo više od 24.000 slika dimenzija 200x200 piksela. Predprocesiranje se sastojalo od toga da su slike isečene i centrirane tako da su lica na istim pozicijama i manje-više zauzimaju istu veličinu prostora na svakoj slici. Skup je vrlo raznovrstan i obuhvata lica širokog raspona godina (preciznije - od 0 do 116) i različite etničke pripadnosti. Dodatna otežavajuća okolnost je da su slike slikane pod različitim osvetljenjem i da su subjekti na njima sa različitim facijalnim ekspresijama. Slike su slučajnim izborom podeljene u dva skupa, trening skup i test skup. Trening skup se sastoji od 18.912 slika (oko 80% skupa), dok test skup sadži 4.795 slika (oko 20%).

Na slici 3 se mogu videti nekoliko slika iz skupa podataka na kojima se jasno vidi raznovrsnost skupa počevši od broja godina, preko rezolucije i osvetljenja do boje kože.



Slika 3: Prikaz raznovrsnosti skupa podataka

4 Metod

4.1 Pregled

Inicijalna ideja, kao što je rečeno, je bilo korišćenje konvolutivne neuralne mreže (CNN) za kompresovanje pomenutih slika lica. Kod konvolutivnih mreža, ulazna slika se obilazi filterima u konvolutivnim slojevima kako bi dobila *mapu karakteristika* sa slike. Sve karakteristike sa mape su zatim kombinovane

kroz potpuno povezani sloj i kroz nekoliko ovakvih iteracija sa manjim slojevima dolazi se do odgovarajućeg kodiranja u zavisnosti od izlaza Euklidske greške (ovde zapisane kao "l2"). Dva glavna razloga zašto smo na početku odabrali konvolutivne mreže su:

- Korišćenjem konvolutivnih mreža odvajamo se od detalja na nivou piksela i prelazimo na učenje karakteristika na slici.
- Trenutno su među najpopularnijim tehnikama mašinskog učenja i njihovo razumevanje značilo bi nam u daljem napretku iz oblasti mašinskog učenja.

Kako bismo ostvarili cilj ovog rada implementirali smo: (1) konvolutivni autoenkoder prikazan na slici 4, i (2) tradicionalan autoenkoder prikazan na slici 6. Za svaki od ovih modela, podešavali smo parametre: *learning rate*, *momentum*, i *weight decay*. Genetski algoritam primenjen je samo na drugom modelu.

4.2 Genetski algoritam

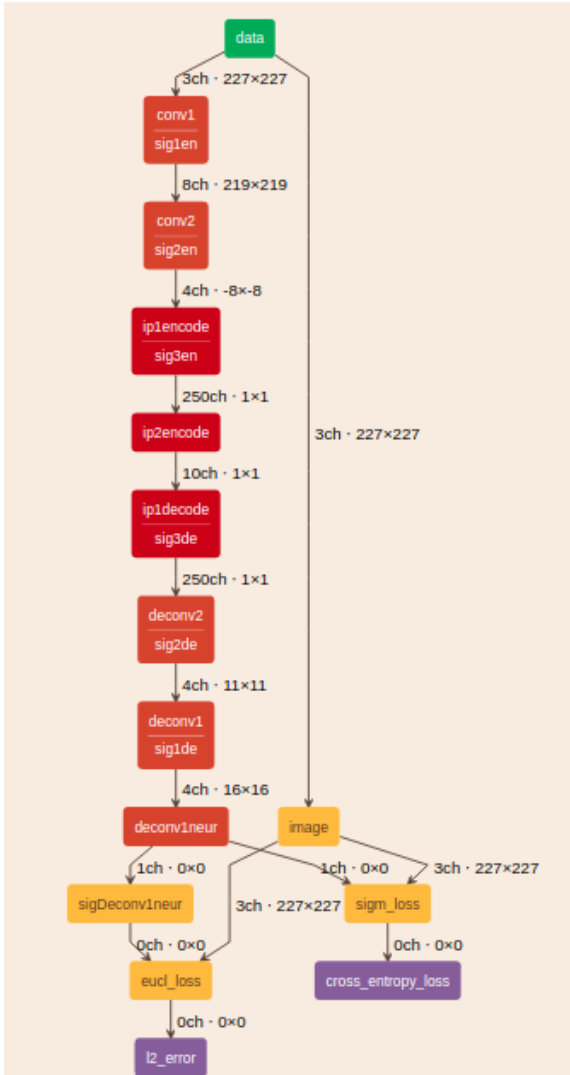
Genetski algoritam koristi parametre učenja autoenkodera kao hromosome. Njihove vrednosti uzima iz intervala koji su nam se empirijski pokazali kao održivi za učenje (gubitak ne ide u beskonačnost ili NaN, što je čest slučaj za korišćenu biblioteku). Funkcija greške, l2, naučene mreže preslikava se u funkciju prilagođenosti. Reprodukcija se obavlja ukrštanjem po svakom parametru učenja pojedinačno. Za odabir jedinki za reprodukciju koristi turnirsku selekciju i prethodnu generaciju potpuno menja novom. Veoma je važno naglasiti da je evolucija ove vrste jedinki veoma spora pošto je pri kreiranju jedinke neophodno istrenirati neuronsku mrežu do određenog broja iteracija.

4.3 Konvolutivna neuronska mreža

U skladu sa svojim osobinama, a i trenutnim trendom u mašinskom učenju konvolutivne mreže postavile su se kao prva opcija kojom bi smo pokušali da rešimo zadati problem, te smo od njih i krenuli. Inspiraciju za ovo pronalazimo u [7].

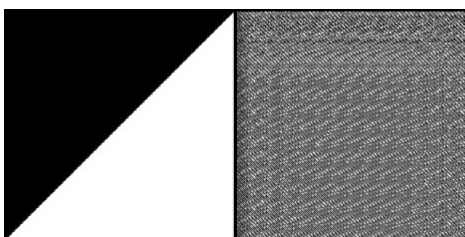
Kreirana mreža ima dva konvolutivna i dva potpuno povezana sloja u enkoderskom delu, dok se dekoderski deo sastoji iz jednog potpuno povezanog sloja i 2 dekonvolutivna sloja. U prvom konvolutivnom sloju imamo 8 filtera veličine 9x9. Drugi konvolutivni sloj ima duplo manje filtera od prvog. Prvi sloj nam daje jednodimenzionalni izlaz dimenzije 250, a drugi taj izlaz smanjuje na svega 10 brojeva (što je 5 puta više podataka na izlazu nego u originalnom radu). Zatim imamo dekoderski deo sa potpuno povezanim slojem, sa izlazom dimenzije 250, dekonvoluticiju sa

4 filtera dimenzije 12x12, dekonvoluciju sa još 4 filtera ovaj put dimenzije 17x17 i na kraju dekonvoluciju za povratak na ulaznu dimenziju slike koji se prosleđuje slojevima za računanje gubitaka (Euklidsko rastojanje i *cross-entropy-loss*).



Slika 4: CAE

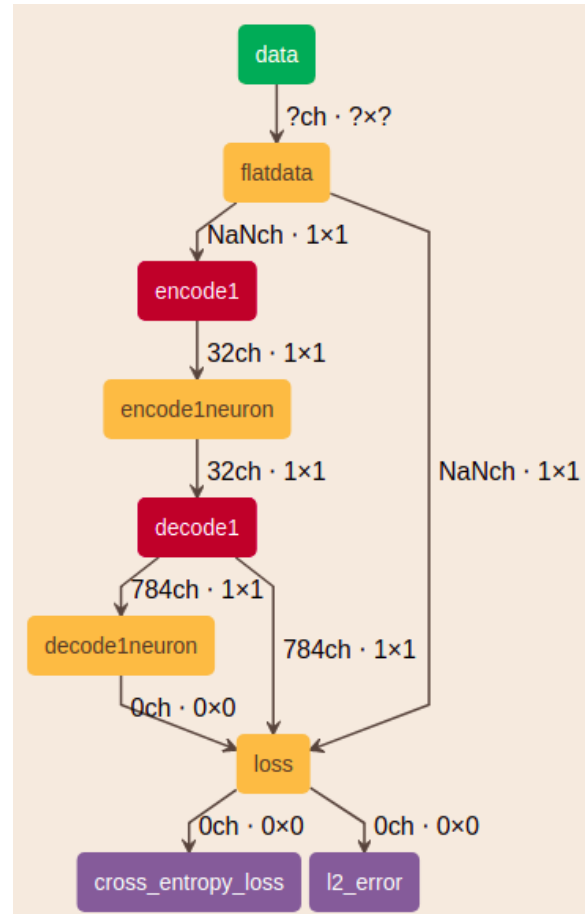
Uprkos očekivanjima, mreža se pokazala promašajem. Od originalne slike, rekonstrukcija ne bi donela ništa (osim šuma). Nakon velikog broja izmena parametara mreže, menjanja arhitektura i povećanja iteracija učenja rezultati nisu bili nimalo bolji, te smo morali pronaći bolje rešenje, što u skladu sa našim znanjem, što u skladu sa raspoloživim hardverom. Dalje istraživanje navelo nas je da se vratimo nekoliko koraka unazad i krenemo od jednostavnijih autoenkodera.



Slika 5: Rezultat konvolutivne mreže - 50000 iteracija

4.4 Jednostavni autoenkoder

Ovaj autoenkoder sastoji se iz svega jednog skrivenog sloja koji predstavlja kompresovanu sliku. Skup podataka korišćen za treniranje je MNIST [8]. Dimenzija ulaza je 28x28. Mreža se pokazala vrlo dobro - značajno bolje od mreže sa konvolutivnim slojevima.



Slika 6: Jednostavni autoenkoder



Slika 7: Ulazne slike



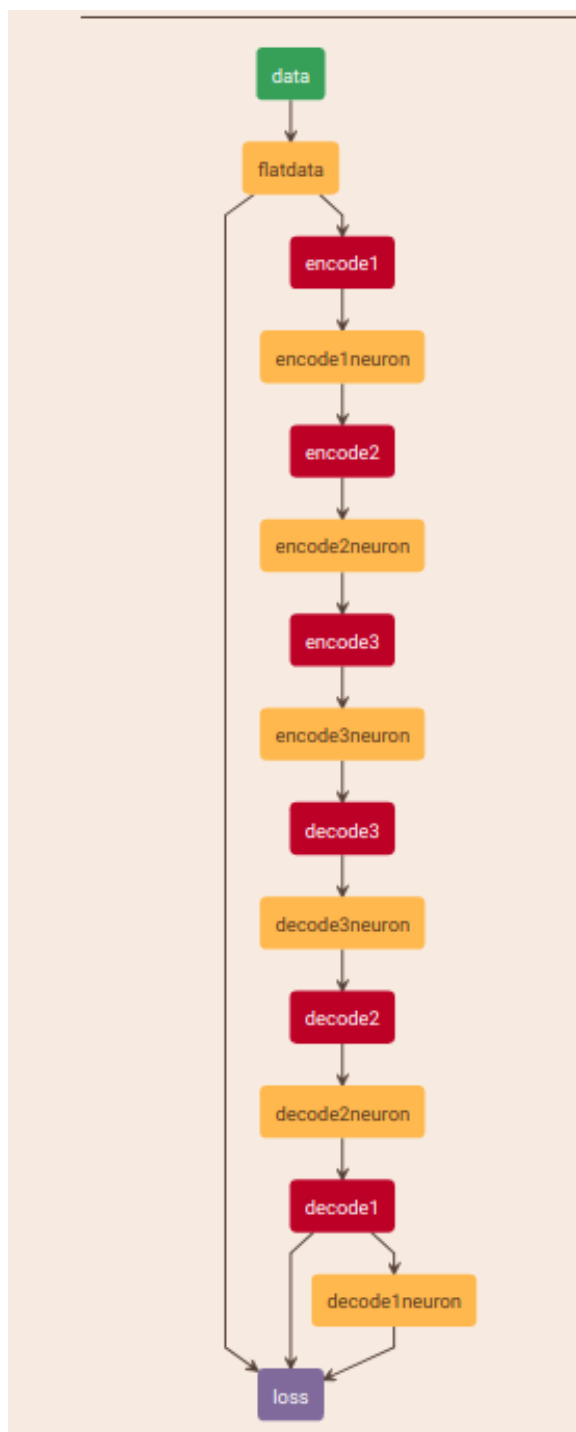
Slika 8: Izlazne slike

Korišćena je *AdaDelta* optimizacija. Broj iteracija je 50.000. Greška je kao i ranije računata Euklidskim rastojanjem po svakom pikselu.

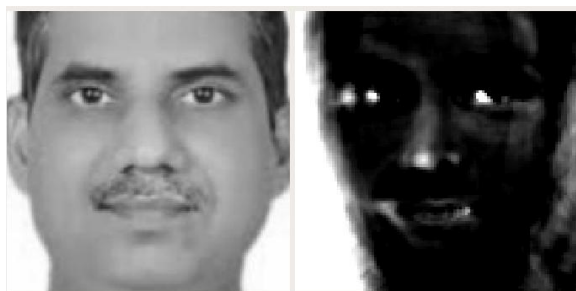
4.5 Duboki autoenkoder

Uspesna (de)kompresija slika rukom pisanih cifara nas je ohrabrila da odemo korak dalje. Za skup podataka sada uzimamo lica, dok mreži dodajemo još unutrašnjih slojeva radi efikasnije kompresije. Dimenzija ulaza sa prethodnih 28x28 raste na čak 100.000. Ova mreža ima 3 potpuno povezana sloja za kompresiju i 3 potpuno povezana za dekompresiju. Dimenzije se kreću 100x100 -> 2048 -> 1024 -> 512 -> 1024

-> 2048 -> 100x100, greška se i ovde računa preko Euklidskog rastojanja i *cross entropy loss* sloja.



Slika 9: DubokiAutoenkoder



Slika 10: Ulaz i izlaz iz dubokog autoenkodera - 200000 iteracija



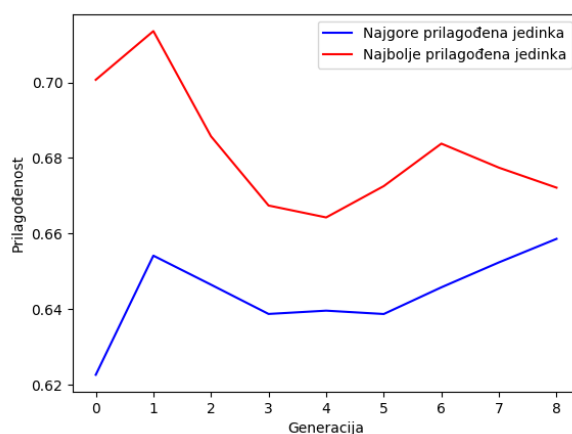
Slika 11: Ulaz i izlaz iz dubokog autoenkodera - 400000 iteracija



Slika 12: Ulaz i izlaz iz dubokog autoenkodera - 858675 iteracija

5 Rezultat

Genetski algoritam je nakon 40 sati izvršavanja stigao svega do 9. iteracije. U svakoj generaciji nalazi se 15 jedinki od kojih je svaka jedinka istrenirana mreža sa 50.000 iteracija. Prilagodjenost jedinki obrnuto je proporcionalna grešci kompresije. Rezultate koje možemo videti na slici 13, navode nas na zaključak da broj generacija nije dovoljan da bismo imali konvergenciju.



Slika 13: Prilagođenost jedinki kroz generacije
 Interesantna je činjenica da prilagođenost najbolje jedinice u generaciji opada, dok prilagođenost najgore jedinice u generaciji raste. Ova pojava i nije čudna kada se uzme u obzir veličina generacije i broj iteracija jer sam algoritam nije mogao ni da da naznake konvergencije usled prekratkog izvršavanja. Uprkos ovome, odlučujemo se da uzmemo najpri-

lagođeniju jedinku iz poslednje iteracije i da koristeći njene parametre učenja obučimo mrežu na čak 850.000 iteracija.

6 Zaključak

Rezultati koje smo dobili navode nas na dva zaključka. Prvi, tehnika ocenjivanja greške pomoću *Euclidskog rastojanja* nije dovoljno dobra. Ideja kojom bismo išli u dalji razvoj trenutnog rada bi bila da iskoristimo *Sijamsku mrežu*, koja se koristi za prepoznavanje sličnosti slika, kako bi nam vratila razliku između dve slike i tu razliku iskoristili kao vrednost greške (odnosno kao funkciju prilagođenosti u genetskom algoritmu). Drugi zaključak koji se nameće je nedostatak odgovarajućeg hardvera. Izračunavanje jedne generacije u genetskom algoritmu se izvršavalo čak 5 sati! Rezultati bi bili daleko interesantniji kada bismo imali veći broj generacija.

Samim odabirom teme za ovaj rad, očekivali smo da će ostvarivanje zacrtanih ciljeva biti teško izvodljivo, kako zbog postojanja samo osnovnog znanja s naše strane o (konvolutivnim) neuronskim mrežama i generalno mašinskom učenju, zatim manjka iskustva, ali i nedostatka adekvatnog hardvera koji bi mogao da se uhvati u koštac sa složenim problemom kao što je ovaj. Nakon pribavljanja hardvera koji je omogućio kakvo-takvo treniranje mreže, naše znanje nije bilo dovoljno da se na istom u razumnom vremenskom roku dobiju bilo kakvi rezultati koji bi mogli biti opisani sa barem "vrlo dobri". Mreža obrađuje slike toliko da se može reći da je na originalnoj slici bilo lice, neke konture lica se u većini slučajeva takođe očuvaju, ali se svakako ne može reći da se posao obavlja u potpunosti. Za dalje razvijanje ovog projekta svakako bi bilo neophodno obezbediti značajno brži hardver i nastaviti u pravcu boljeg razumevanja i razvijanja konvolutivnih neuronskih mreža, jer je

jasno da u slučaju poznavanja njihovih mogućnosti i kako se one koriste - rezultati mogu biti značajno bolji.

Reference

- [1] R. J. W. D. E. Rumelhart, G. E. Hinton. Learning internal representations by error propagation. 1986. URL <https://dl.acm.org/citation.cfm?id=104293>.
- [2] R. S. Z. Geoffrey E. Hinton. Autoencoders, minimum description length and helmholtz free energy. 2001. URL <https://www.cs.toronto.edu/~hinton/absps/cvq.pdf>.
- [3] A. R. L. M. C. B. A. P. M. B. C. B. A. L. Irina Higgins, Arka Pal. Darla: Improving zero-shot transfer in reinforcement learning. 2018. URL <https://arxiv.org/pdf/1707.08475.pdf>.
- [4] K. H. Pierre Baldi. Neural networks and principal component analysis: Learning from examples without local minima. 1986. URL <https://www.researchgate.net/publication/222438485>.
- [5] A. Research. Utkface. URL <https://goo.gl/QyXzw8>.
- [6] R. Thierry Dumas, Aline and C. Guillemot. Autoencoder based image compression: Can the learning be quantization independent? 2018. URL <https://arxiv.org/pdf/1802.09371.pdf>.
- [7] A. L. Volodymyr Turchenko. Creation of a deep convolutional auto-encoder in caffe. 2015. URL <https://arxiv.org/pdf/1512.01596.pdf>.
- [8] N. C. C. G. L. N. Y. C. J. B. M. R. R. Yann LeCun, Courant Institute. Mnist. URL <http://yann.lecun.com/exdb/mnist/>.