SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

IZVJEŠTAJ

Parsevalove mreže

Ivan Grubišić
Voditelj: Siniša Šegvić

Zagreb, siječanj 2018.

Sadržaj

1. Uvod **1**

1.1. Rizik kod nadziranog učenja 1

1.2. Neprijateljski primjeri 1

2. Parsevalove mreže **4**

2.1. Ograničavanje neprijateljskog rizika Lipschitzovom konstantom 4

2.2. Lipschitzova konstanta neuronske mreže 6

2.3. Parsevalove mreže 7

2.4. Rezultati 8

3. Programsko ostvarenje i rezultati **11**

3.1. Razlike u odnosu na modele koje su autori koristili 11

3.2. Rezultati. 12

4. Literatura **16**

A. Konfiguracija rezidualnih mreža **18**

B. Ostali rezultati **19**

1. Uvod

Model nadziranog strojnog učenja može se prikazati funkcijom $h : \mathcal{X} \times \Theta \to \mathcal{Y}$, gdje je $\mathcal{X}$ skup kojemu pripadaju primjeri, Θ skup mogućih parametara, a $\mathcal{Y}$ skup mogućih oznaka. Nadalje ćemo razmatrati klasifikacijske modele gdje su $\mathcal{X}$ i Θ vektorski prostori, a $\mathcal{Y}$ konačan skup $\{1..C\}$.

1.1. Rizik kod nadziranog učenja

Cilj algoritma nadziranog strojnog učenja je po parametrima modela θ minimizirati rizik $R(\boldsymbol{\theta})$ nad razdiobom označenih primjera $\mathcal{D}$. Uz odabir odgovarajuće funkcije gubitka L, rizik je ovako definiran:

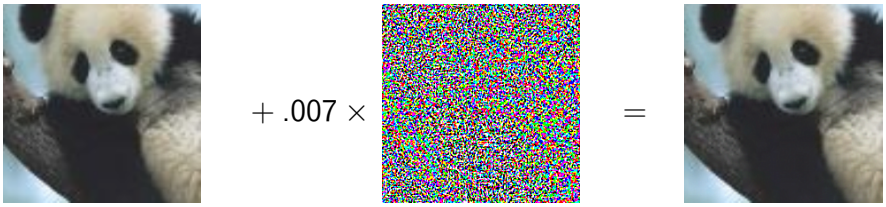
$$R(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[L(h(\boldsymbol{x};\theta), y) \right] . \tag{1.1}$$

U problemima koji se rješavaju strojnim učenjem stvarna razdioba $\mathcal{D}$ nije poznata i dostupan je samo konačan broj primjera. Moguće je minimizirati samo procjenu rizika na temelju dostupnih podataka – empirijski rizik.

1.2. Neprijateljski primjeri

I za najbolje klasifikacijske modele koji postižu jako dobru generalizaciju na prirodnim podacima moguće je pronaći primjere takve da se u ljudskoj percepciji malo razlikuju od originalnih prirodnih i lako se prepoznaju, ali da ih model potpuno krivo klasificira [11, 4]. Na slici 1.1 prikazano je generiranje neprijateljskog primjera malom izmjenom izvorne slike.

Pronalaženje neprijateljskih primjera Neka je $d : \mathcal{X} \times \mathcal{X}$ funkcija udaljenosti u ulaznom prostoru. Za svaki primjer $\boldsymbol{x}$ može se definirati susjedstvo



$\boldsymbol{x}$
panda (0.577)

$\operatorname{sgn} \nabla_{\boldsymbol{x}} L(h(\boldsymbol{x}; \theta), y)$

$\tilde{\boldsymbol{x}}$
gibon (0.993)

Slika 1.1: Prilagođeni prikaz dobivanja neprijateljskog primjera FGSM-om iz [4]. Nakošene riječi predstavljaju razrede, a brojevi u zagradama vjerojatnosti koje neuronska mreža dodjeljuje razredima.

$B_\epsilon(\boldsymbol{x}) = \{\tilde{\boldsymbol{x}} : d(\tilde{\boldsymbol{x}}, \boldsymbol{x}) \leq \epsilon\}$. Pronalaženje neprijateljskih primjera se može definirati kao optimizacijski problem [11, 3, 9] pronalaženja primjera $\tilde{\boldsymbol{x}}$ koji maksimizira gubitak uz ograničenje da se nalazi u susjedstvu B_ϵ prirodnog primjera $\boldsymbol{x}$:

$$\tilde{\boldsymbol{x}} = \underset{\tilde{\boldsymbol{x}} \in B_\epsilon(\boldsymbol{x})}{\arg\max}\, L(h(\tilde{\boldsymbol{x}}; \theta), y). \tag{1.2}$$

Za funkciju udaljenosti d se obično uzima neka p-norma razlike [4, 3, 9]. Npr. za ∞-normu je $B_\epsilon(\boldsymbol{x}) = \{\tilde{\boldsymbol{x}} : \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\infty \leq \epsilon\}$.

Neprijateljski primjeri mogu se pronaći iterativnim optimizacijskim postupcima prvog reda uz održavanje ograničenja susjedstva. Pokazuje se da je neprijateljske primjere moguće pronaći već samo jednim korakom u smjeru predznaka gradijenta po svakoj dimenziji. Jedna vrsta takvog napada je "*fast gradient sign method*" (FGSM) [4]. Neprijateljskom primjeru koji se pronalazi FGSM-om odgovara sljedeći izraz:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \epsilon \operatorname{sgn} \nabla_{\boldsymbol{x}} L(h(\boldsymbol{x}; \theta), y). \tag{1.3}$$

U [9] definiran je iterativni postupak koji se temelji na FGSM-u i autori ga nazivaju *projected gradient descent* (PGD):

$$\tilde{\boldsymbol{x}} \leftarrow \Pi_{B_\epsilon(\boldsymbol{x})}(\tilde{\boldsymbol{x}} + \alpha \operatorname{sgn} \nabla_{\tilde{\boldsymbol{x}}} L(h(\tilde{\boldsymbol{x}}; \theta), y)). \tag{1.4}$$

α je veličina koraka optimizacije, a $\Pi_{B_\epsilon(\boldsymbol{x})}$ ovdje predstavlja projekciju na zatvorenu ϵ-kuglu oko prirodnog primjera $\boldsymbol{x}$ uz ∞-normu. Npr. projekcijom vektora $\boldsymbol{v}$ na susjedstvo vektora $\boldsymbol{x}$, $\Pi_{B_\epsilon(\boldsymbol{x})}(\boldsymbol{v}) = \arg\min_{\boldsymbol{v}' \in B_\epsilon(\boldsymbol{x})} \|\boldsymbol{v}' - \boldsymbol{v}\|_\infty$, svakoj komponenti v_i dodjeljuje se najbliža vrijednost unutar intervala $[x_i - \epsilon, x_i + \epsilon]$.

Mogući su i napadi bez uvida u strukturu modela, npr. genetskim algoritmom. Također, pokazuje se da su neprijateljski primjeri u velikoj mjeri prenosivi između različitih modela [11, 4, 8].

Neprijateljski rizik. Može se definirati oblik rizika koji se može nazvati *neprijateljskim rizikom* [9]:

$$\widetilde{R}(\boldsymbol{\theta}) = \widetilde{R}(\boldsymbol{\theta}; d, \epsilon) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\widetilde{\boldsymbol{x}}\in B_\epsilon(\boldsymbol{x})} L(h(\widetilde{\boldsymbol{x}}; \theta), y) \right]. \tag{1.5}$$

Mali neprijateljski rizik predstavlja dobru lokalnu generalizaciju u susjedstvu prirodnih primjera.

Učenje s neprijateljskim primjerima Trenutno najuspješniji pristup za postizanje otpornosti na neprijateljske primjere je učenje s neprijateljskim primjerima (engl. *adversarial training*). Kod učenja s neprijateljskim primjerima skup za učenje se proširuje neprijateljskim primjerima koji se tijekom učenja prilagođavaju parametrima mreže [4, 7].

U nedavno objavljenom radu [9] eksperimentalno je pokazano da je učenjem s neprijateljskim primjerima dobivenim PGD-om (jednadžba 1.4) moguće postići dobru otpornost na neprijateljske primjere. Također, rezultati pokazuju da je za postizanje otpornosti na neprijateljske primjere uz održavanje dobre generalizacije potreban značajno veći kapacitet mreže.

2. Parsevalove mreže

U [3] autori predstavljaju tzv. *Parsevalove mreže*. Kod njih se kontrolira Lipschitzova konstanta svih slojeva i cijele mreže tako da ne bude veća od 1 i vrši se posebna vrsta regularizacije nad matricama težina. Motivacija je postizanje otpornosti na neprijateljske primjere kod dubokih neuronskih mreža. Kao najvažnija značajka Parsevalovih mreža ističe se to da se matrice težina linearnih i konvolucijskih slojeva ograničavaju tako da približno odgovaraju Parsevalovim uskim okvirima, tj. da budu ortogonalne matrice poopćene na nekvadratne matrice. Prema autorima, takve mreže postižu bolju otpornost na naprijateljske primjere generirane FGSM-om od odgovarajućih mreža koje nisu Parsevalove, brže se uče i njihov kapacitet se bolje iskorištava.

2.1. Ograničavanje neprijateljskog rizika Lipschitzovom konstantom

Lipschitzova konstanta funkcije f, ako postoji, definirana je ovako:

$$\Lambda = \sup_{\boldsymbol{x} \neq \widetilde{\boldsymbol{x}}} \frac{\|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}})\|}{\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\|}. \tag{2.1}$$

Za funkcije za koje je Lipschitzova konstanta definirana kaže se da su Lipschitz-kontinuirane.

Klasifikator je funkcija $\hat{h} : \mathbb{R}^D \times \Theta \to \mathcal{Y}$, gdje je D dimenzija ulaznog prostora, Θ prostor parametara, a $\mathcal{Y} = \{0..C\}$ uz to da je C broj razreda. Neuronske mreža za klasifikaciju su obično funkcije h koje ulaz preslikavaju u vektor dimenzije C koji predstavlja kategoričku razdiobu razreda i vrijedi $\hat{h}(\boldsymbol{x}; \boldsymbol{\theta}) = \arg\max_y h(\boldsymbol{x}; \boldsymbol{\theta})_y$. U daljnjem razmatranju mreža će se nekad predstavljati funkcijom g kojoj su kodomena realni vektori koji predstavljaju kategoričke logite, tj. vrijedi $h(\boldsymbol{x}; \boldsymbol{\theta}) = \text{softmax}(g(\boldsymbol{x}; \boldsymbol{\theta}))$.

Neuronska mreža se može prikazati kao usmjereni aciklički računski graf $G = (\mathcal{N}, \mathcal{E})$ gdje je svaki čvor $n \in \mathcal{N}$ funkcija svoje djece:

$$n(\boldsymbol{x}) = f^{(n)}(\boldsymbol{\theta}^{(n)}, (n'(\boldsymbol{x}))_{n':(n,n')\in\mathcal{E}}). \tag{2.2}$$

Funkcija $h(\boldsymbol{x}) = h(\boldsymbol{x}; \boldsymbol{\theta})$ koju ostvaruje mreža je korijen toga grafa. U nastavku će $n' \lessdot n$ označavati da je n roditelj od n', tj. $(n, n') \in \mathcal{E}$.

Za gubitak klasifikatora koji kao izlaz daje kategoričku razdiobu obično se koristi gubitak unakrsne entropije:

$$L(h(\boldsymbol{x}; \boldsymbol{\theta}), y) = -\ln h(\boldsymbol{x}; \boldsymbol{\theta})_y \tag{2.3}$$

Gubitak se preko funkcije g može izraziti ovako:

$$L(h(\boldsymbol{x}; \boldsymbol{\theta}), y) = -g(\boldsymbol{x}; \boldsymbol{\theta})_y + \ln \sum_{y'\in\mathcal{Y}} \exp g(\boldsymbol{x}; \boldsymbol{\theta})_{y'}. \tag{2.4}$$

Radi jednostavnijeg zapisa definiramo funkciju ℓ tako da uvijek vrijedi $\ell(g(\boldsymbol{x}; \boldsymbol{\theta}), y) = L(h(\boldsymbol{x}; \boldsymbol{\theta}), y)$. Možemo Uz zadanu p-normu pretpostaviti da postoji λ_p takav da

$$\forall \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^C, \forall y \in \mathcal{Y}, |\ell(\boldsymbol{z}, y) - \ell(\boldsymbol{z}', y)| \leq \lambda_p \|\boldsymbol{z} - \boldsymbol{z}'\|_p. \tag{2.5}$$

Vrijedi $\frac{\partial \ell(\boldsymbol{z},y)}{\partial z_{y'}} = -[\![y' = y]\!] + \text{softmax}(\boldsymbol{z})_{y'}$ i u najgorem slučaju se za $\frac{\partial \ell(\boldsymbol{z},y)}{\partial \boldsymbol{z}}$ može dobiti vektor redak kojemu je jedan element –1, jedan 1, a svi drugi 0. Njegova 2-norma je $\sqrt{2}$, a ∞-norma 2. Zato za odgovarajuće Lipschitzove konstante gubitka s obzirom na logite vrijedi $\lambda_2 = \sqrt{2}$ i $\lambda_\infty = 2$.

Uz neku p-normu, neka je λ_p Lipschitzova konstanta funkcije gubitka s obzirom na logite, a Λ_p Lipschitzova konstanta funkcije $g(\boldsymbol{x})$, tj. izlaza sloja logita s obzirom na ulaz mreže. Za svaki p i $\epsilon > 0$ iz izraza 2.5 i definicije rizika $R(\boldsymbol{\theta})$ i neprijateljskog rizika $\widetilde{R}(\boldsymbol{\theta}) = \widetilde{R}(\boldsymbol{\theta}, p, \epsilon)$ može se pokazati da vrijedi

$$\widetilde{R}(\boldsymbol{\theta}) \leq R(\boldsymbol{\theta}) + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\widetilde{\boldsymbol{x}}\in B_\epsilon(\boldsymbol{x})} |\ell(z(\boldsymbol{x}; \boldsymbol{\theta}), y) - \ell(z(\boldsymbol{x}; \boldsymbol{\theta}), y)| \right] \tag{2.6}$$

$$\leq R(\boldsymbol{\theta}) + \epsilon\lambda_p\Lambda_p. \tag{2.7}$$

Budući da uvijek vrijedi $R(\boldsymbol{\theta}) \leq \widetilde{R}(\boldsymbol{\theta})$, slijedi

$$0 \leq \widetilde{R}(\boldsymbol{\theta}) - R(\boldsymbol{\theta}) \leq \lambda_p\Lambda_p\epsilon. \tag{2.8}$$

Može se vidjeti da smanjivanje Lipschitzove konstante samo po sebi nije dovoljno za poboljšanje otpornosti na neprijateljske primjere. Promjenom Lipschitzove

konstante općenito se može utjecati i na $\widetilde{R}(\boldsymbol{\theta})$ i na $R(\boldsymbol{\theta})$. Npr. uvođenjem množenja logita s nekim skalarom $\alpha \in [0, 1)$ prije primjene softmax-a, tako da bude $h(\boldsymbol{x}) = \text{softmax}(\alpha g(\boldsymbol{x}))$, smanjuje se Lipschitzova konstanta mreže i razlika $\widetilde{R}(\boldsymbol{\theta}) - R(\boldsymbol{\theta})$ postaje manja, ali time se samo povećava entropija izlazne razdiobe i ne utječe se na rezultat klasifikacije $\arg\max_y h(\boldsymbol{x})_y$. U slučaju kada $\alpha = 0$, dobiva se $\widetilde{R}(\boldsymbol{\theta}) = R(\boldsymbol{\theta})$, ali izlazna razdioba vjerojatnosti $h(\boldsymbol{x}; \boldsymbol{\theta})$ postaje uniformna. Ozbiljniji primjeri smanjivanja Lipschitzove konstante su L_1 i L_2 regularizacija za koje se isto pokazalo da nemaju značajan utjecaj na otpornost na neprijateljske primjere [11, 4].

Autori u članku o Parsevalovim mrežama iskazuju još jedan argument koji ima veze s kontroliranjem Lipschitzove konstante referencirajući [12], ali to neće biti obrađeno u ovom izvještaju.

2.2. Lipschitzova konstanta neuronske mreže

Uz definiranje $\Lambda_p^{(n,n')}$ kao Lipschitzove konstante čvora n s obzirom na njemu ulazni čvor n', može se pokazati da vrijedi

$$\Lambda_p^{(n)} = \sum_{n' \leqslant n} \Lambda_p^{(n,n')} \Lambda_p^{(n')}. \tag{2.9}$$

Sloj linearnog preslikavanja. Kod linearnih slojeva kod kojih $n(\boldsymbol{x}) = \boldsymbol{W}^{(n)} n'(\boldsymbol{x})$ Lipschitzova konstanta s obzirom na ulaz $n'(\boldsymbol{x})$ odgovara matričnoj normi matrice parametara $\boldsymbol{W}^{(n)}$ pa je

$$\Lambda_p^{(n)} \leq \|\boldsymbol{W}^{(n)}\|_p \Lambda_p^{(n')}. \tag{2.10}$$

Za afine transformacije vrijedi isto jer pomak ne utječe na Lipschitzovu konstantu.

Konvolucijski sloj. Razmatramo konvolucijski sloj s d prostornih dimenzija i jednom dimenzijom značajki. Tada konvolucijska jezgra, kojoj su sve dimenzije jednake k (radi jednostavnijeg zapisa) i koja na svakom položaju po prostornim dimenzijama ulaza računa jedan element izlaza na temelju ulaza sa semantičkom dimenzijom H, sadrži $k^d H$ elemenata. Konvolucijskom sloju koji daje izlaz sa semantičkom dimenzijom D ima složenu jezgru koja se sastoji od D takvih (pod)jezgri. Može se definirati operator razmatanja U koji polje dimenzija $H \times a_1 \times \cdots \times a_d$ pretvara u matricu dimenzija $k^d H \times \prod_{i \in \{1..d\}} a_i$ tako da se za svaki položaj jezgre svi elementi koje ona pokriva kopiraju u jedan vektor stupac [2].

Ako se svaka podjezgra razmota u vektor redak i cijela jezgra prikaže kao matrica $\boldsymbol{W}^{(n)}$ dimenzija $D \times k^d H$, onda se konvolucija $n(\boldsymbol{x}) = \mathbf{w}^{(n)} *_d n'(\boldsymbol{x})$ može prikazati kao matrično množenje koje kao rezultat daje $\boldsymbol{W}^{(n)} U(n'(\boldsymbol{x}))$, što je matrica dimenzija $D \times \prod_{i \in \{1..d\}} a_i$ koja se opet može prikazati kao polje dimenzija $D \times a_1 \times \cdots \times a_d$. Budući da je U linearan operator koji svaki element ulaza kopira k^d puta, vrijedi $\|U(n'(\boldsymbol{x})) - U(n'(\widetilde{\boldsymbol{x}}))\|_2^2 \leq k^d \|n'(\boldsymbol{x}) - n'(\widetilde{\boldsymbol{x}})\|_2^2$ i dobiva se

$$\Lambda_2^{(n)} \leq k^{d/2} \|\boldsymbol{W}^{(n)}\|_2 \Lambda_2^{(n')}, \tag{2.11}$$

$$\Lambda_\infty^{(n)} \leq \|\boldsymbol{W}^{(n)}\|_\infty \Lambda_\infty^{(n')}. \tag{2.12}$$

Aktivacijski sloj. Kod aktivacijskih slojeva Lipschitzova konstanta s obzirom na ulaz čvora odgovara Lipschitzovoj konstanti prijenosne funkcije $\lambda_p^{(n)}$ pa vrijedi

$$\Lambda_p^{(n)} \leq \lambda_p^{(n)} \Lambda_p^{(n')}. \tag{2.13}$$

Linearna kombinacija izlaza različitih slojeva. Kod slojeva linearne kombinacije vrijedi $n(\boldsymbol{x}) = \sum_{n' \leqslant n} \alpha^{(n,n')} n'(\boldsymbol{x})$, gdje su $\alpha^{(n,n')}$ skalari. Za Lipschitzovu konstantu vrijedi

$$\Lambda_p^{(n)} \leq \sum_{n' \leqslant n} \alpha^{(n,n')} \Lambda_p^{(n')}. \tag{2.14}$$

Poseban slučaj takvog čvora je čvor zbrajanja kakav se javlja kod preskočnih veza u rezidualnim mrežama [5]. Za takav čvor je

$$\Lambda_p^{(n)} \leq \sum_{n' \leqslant n} \Lambda_p^{(n')}. \tag{2.15}$$

2.3. Parsevalove mreže

Kako bi se osiguralo ograničenje Lipschitzove konstante kroz cijelu mrežu, autori predlažu održavanje redaka matrica težina ortonormalnima i zamjenu zbrojeva kod preskočnih veza konveksnim kombinacijama. Održavanje takvih ograničenja autori nazivaju *Parsevalovom regularizacijom*.

Ortogonalnost matrica težina Neka Za matricu težina[1] $\boldsymbol{W}^\mathsf{T} \in \mathbb{R}^{D \times H}$ s ulaznom dimenzijom H i izlaznom dimenzijom D cilj je održavati ograničenje da su

[1] Radi usklađenosti s oznakama autora, transformacijska matrica se ovdje označava s $\boldsymbol{W}^\mathsf{T}$ umjesto s $\boldsymbol{W}$ i izlaz se računa kao $\boldsymbol{y} = \boldsymbol{W}^\mathsf{T} \boldsymbol{x}$.

singularne vrijednosti blizu 1, tj. ako $D < H$, da vrijedi $\boldsymbol{W}^\mathsf{T}\boldsymbol{W} \approx \boldsymbol{I}_D$. Takvoj matrici je lako odrediti vlastite vrijednosti i ograničiti spektralnu normu kao i operatorsku ∞-normu. Iz izraza 2.11 slijedi da je kod konvolucijskih slojeva matricu težina potrebno podijeliti s k^2 kako bi se održalo ograničenje Lipschitzove konstante, tj. da bude $\Lambda_2^{(n)} \leq \Lambda_2^{(n')}$. Kako bi se održavalo ograničanje ortogonalnosti, autori prvo za sve matrice težina $\boldsymbol{W}$ definiraju regularizacijski gubitak:

$$R_\beta(\boldsymbol{W}) = \frac{\beta}{2}\|\boldsymbol{W}^\mathsf{T}\boldsymbol{W} - \boldsymbol{I}\|_2^2 \tag{2.16}$$

i koriste postupak prvog reda kako bi se održavao blizu 0. Za gradijent tog gubitka po matrici težina vrijedi $\nabla_{\boldsymbol{W}} R_\beta(\boldsymbol{W}) = \beta(\boldsymbol{W}\boldsymbol{W}^\mathsf{T} - \boldsymbol{I})\boldsymbol{W}$. U Parsevalovim mrežama se nakon svakog ažuriranja težina radi efikasnosti primjenjuje jedan korak gradijentnog spusta po ovom gubitku:

$$\boldsymbol{W} \leftarrow (1+\beta)\boldsymbol{W} - \beta\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\boldsymbol{W}. \tag{2.17}$$

Ograničenje konveksnosti linearne kombinacije Kod Parsevalovih mreža slojevi koji obavljaju linearnu kombinaciju ulaznih čvorova, kao što je zbrajanje kod preskočnih veza, zamjenjuju se slojevima koji obavljaju konveksnu kombinaciju. Neka je $\boldsymbol{a}$ vektor koji sadrži sve koeficijente iz jednadžbe (2.14). Koeficijenti se ograničavaju se tako da vrijedi $\sum_i \alpha_i = 1$ i $\forall i\, \alpha_i \in [0,1]$. Tako se osigurava da Lipschitzove konstanta sloja s obzirom na ulaz ne bude veća od najveće Lipschitzove konstante među ulaznim čvorovima. α su parametri koji se uče i ograničenje se ostvaruje projekcijom koeficijenata nakon svakog koraka gradijentnog spusta. Općeniti postupak objašnjen je u članku. U slučaju preskočnih veza zbroj se sastoji od dvaju pribrojnika i on se zamjenjuje konveksnom kombinacijom $n(\boldsymbol{x}) = \alpha n'(\boldsymbol{x}) + (1-\alpha)n''(\boldsymbol{x})$, gdje su n' i n'' djeca čvora n. Ograničavanje koeficijenta je onda jednostavno: $\alpha \leftarrow \min\{\max\{\alpha, 0\}, 1\}$.

2.4. Rezultati

Autori su eksperimentalno evaluirali postupak s rezidualnim mrežama [5, 6, 13] na skupovima podataka MNIST, CIFAR-10, CIFAR-100 i SVHN i s mrežama s potpuno povezanim slojevima na skupovima MNIST i CIFAR-10. Ovdje će samo biti kratko opisani rezultati analize rezidualne mreže WRN-28-10 [13] i odgovarajuće Parsevalove mreže na skupu podataka CIFAR-10.

Konfiguracija mreža i hiperparametri prikazani su u tablici A.1. Korišten je isti postupak učenja i proširivanja podataka kao u [13]. Razlika između Parsevalove mreže i obične mreže WRN-28-10 je u ograničavanju matrica težina i konveksnim kombinacijama umjesto zbrojeva kod preskočnih veza. Grupna normalizacija utječe na Lipschitzovu konstantu, ali autori Parsevalovih mreža nisu naveli jesu li mijenjali nešto u vezi nje.

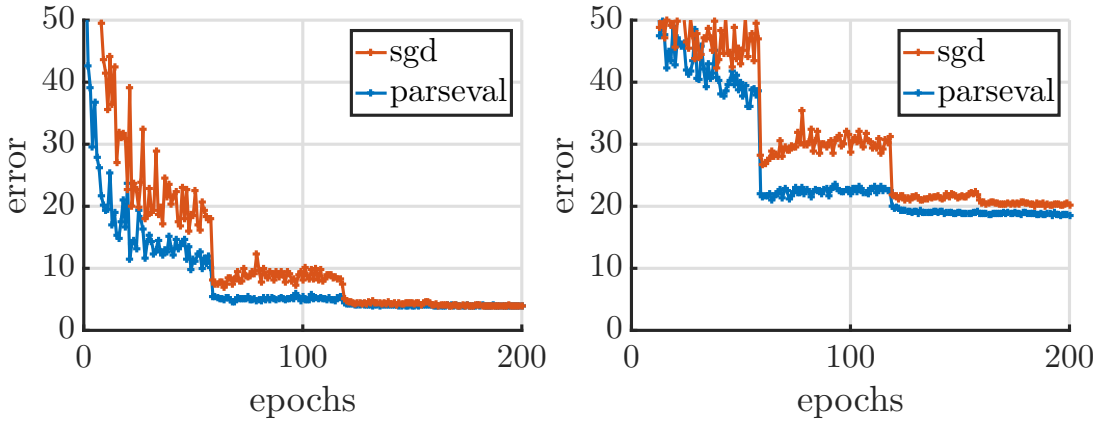
Autori zaključuju da se kod Parsevalovih mreža poboljšava otpornost na neprijateljske primjere, ubrzava učenje i bolje iskorištva kapacitet mreže. U tablici 2.1 prikazane su točnosti klasifikacije neprijateljskih primjera mreže WRN-28-10 ovisno o skupu podataka i vrsti regularizacije. Omjer signala i šuma u decibelima u tablici je za ulaz $\boldsymbol{x}$ i njegov pomak $\boldsymbol{\delta}$ definiran ovako: $\mathrm{SNR}_{\mathrm{dB}}(\boldsymbol{x}, \boldsymbol{\delta}) = 20 \log_{10} \frac{\|\boldsymbol{x}\|_2}{\|\boldsymbol{\delta}\|_2}$.

	Model	**Bez šuma**	$\varepsilon \approx 50$	$\varepsilon \approx 45$	$\varepsilon \approx 40$	$\varepsilon \approx 33$
CIFAR-10	Vanilla	95.63	90.16	85.97	76.62	67.21
	Parseval(OC)	95.82	91.85	88.56	78.79	61.38
	Parseval	**96.28**	**93.03**	**90.40**	**81.76**	**69.10**
	Vanilla	95.49	91.17	88.90	86.75	84.87
	Parseval(OC)	95.59	92.31	90.00	**87.02**	**85.23**
	Parseval	**96.08**	**92.51**	**90.05**	86.89	84.53
CIFAR-100	Vanilla	79.70	65.76	57.27	44.62	34.49
	Parseval(OC)	81.07	70.33	63.78	49.97	32.99
	Parseval	**80.72**	**72.43**	**66.41**	**55.41**	**41.19**
	Vanilla	79.23	67.06	62.53	56.71	51.78
	Parseval(OC)	**80.34**	69.27	62.93	53.21	**52.60**
	Parseval	80.19	**73.41**	**67.16**	**58.86**	39.56
SVHN	Vanilla	**98.38**	97.04	95.18	92.71	88.11
	Parseval(OC)	97.91	**97.55**	**96.35**	**93.73**	**89.09**
	Parseval	98.13	97.86	96.19	93.55	88.47

Tablica 2.1: Točnost klasifikacije mreže WRN-28-10 na skupovima CIFAR-10 i CIFAR-100 s različitim regularizacijama. ε predstavlja omjer signala i šuma u decibelima. Za $\varepsilon = 30$, čovjek može prepoznati da je dodan neprijateljski šum. Za svaki skup podataka prva 3 retka predstavljaju rezultate za učenje bez neprijateljskih primjera, a donja 3 retka za učenje s neprijatljskim primjerima. *Parseval(OC)* označava mrežu na kojoj se ne koriste konveksne kombinacije, nego samo ograničenja ortogonalnosti. Tablica je preuzeta iz članka.

Na slici 2.1 prikazane su krivulje učenja koje su dobili autori s Parsevalovom i

neizmijenjenom rezidualnom mrežom na skupovima podataka CIFAR-10 i CIFAR-100.



Slika 2.1: Krivulje koje pokazuju ovisnost klasifikacijske pogreške o broju završenih epoha koje su autori dobili učenjem obične (narančasto) i Parsevalove (plavo) mreže WRN-28-10 na skupovima CIFAR-10 (lijevo) i CIFAR-100 (desno). Slika je preuzeta iz članka.

3. Programsko ostvarenje i rezultati

Autori Parsevalovih mreža [3] nisu objavili izvorni kod i za projektni zadatak je bilo potrebno implementirati i evaluirati Parsevalovu rezidualnu mrežu. Pythonu je ostvarena biblioteka koja omogućuje definiranje općenite široke rezidualne mreže [5, 6, 13] u TensorFlow-u [1]. Uz običnu rezidualnu mrežu implementirana je i odgovarajuća Parsevalova mreža. Izvorni kod je dostupan ovdje: https://github.com/Ivan1248/Parseval-networks.

Osim TensorFlowa, NumPyja, Matplotliba i drugih standardnih biblioteka korištena je i biblioteka CleverHans [10] za generiranje neprijateljskih primjera.

Za sve ekperimente je korištena rezidualna mreža koja bi trebala biti što sličnija mreži WRN-28-10 [13] i njoj odgovarajuća Parsevalova mreža. Podskupovi skupa podataka CIFAR-10 korišteni su za učenje i evaluaciju.

3.1. Razlike u odnosu na modele koje su autori koristili

Ostvarene su rezidualne mreže koje bi trebale odgovarati Parsevalovoj i običnoj rezidualnoj mreži WRN-28-10. Poznate razlike u odnosu na modele koje su autori koristili su:

1. Za inicijalizaciju težina koristi se uniformna razdioba s varijancom $2/(k_1 k_2 H + D)$, dok su autori članka vjerojatno kao i [13] koristili normalnu razdiobu s varijancom $2/k_1 k_2 H$, gdje su k_1 i k_2 prostorne dimenzije jezgre, H semantička dimenzija ulaza, a D semantička dimenzija izlaza konvolucijskog sloja.

2. Za grupnu normalizaciju se procjene očekivanja i varijance računaju kao eksponencijalni pokretni prosjek s ažuriranjem oblika $\hat{\mu} \leftarrow 0.95\hat{\mu} + 0.05\mu_b$ koji

se ažurira nakon svake mini-grupe b tijekom učenja. Nije poznato koji su faktor autori koristili.

3. U blokovima u kojima se smanjuju prostorne, a povećava semantička dimenzija, preskočna veza zamjenjuje se sažimanjem usrednjavanjem po prostornim dimenzijama i dopunjavanjem nulama u semantičkoj dimenziji kao kod [5, 6]. Kod [13] se za to koristi konvolucija 1×1 s korakom 2. U [6] analizirane su neproširene rezidualne mreže i prema rezultatima se čini da dopunjavanje nulama daje malo bolje rezultate, ali nije poznato s koliko je značajna ta razlika.

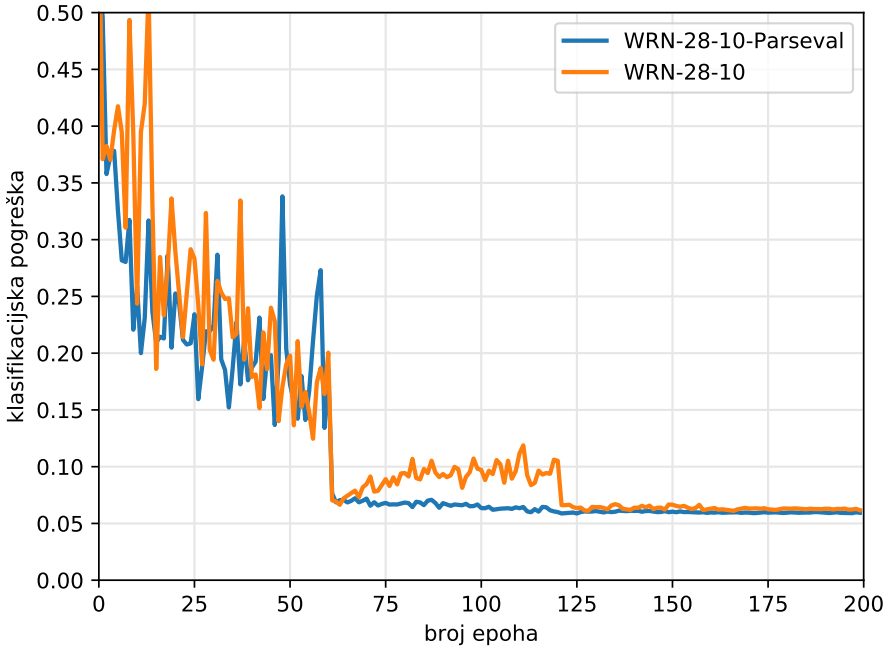
U članku ne piše kako su autori inicijalizirali koeficijente α slojeva konveksne kombinacije. Oni se ovdje inicijaliziraju brojem 0.99. Bilo bi dobro analizirati utjecaj početnih vrijednosti koeficijenata α na učenje. Zasad se na temelju ne jako temeljitih eksperimenata čini da prije 20. epohe kod mreže WRN-28-1 razdioba koeficijenata kroz slojeve postane slična uniformnoj, ali malo koncentriranije oko početne vrijednosti. Primjer slučajnog uzorka naučenih koeficijenata je u tablici B.1.

Dobivene točnosti klasifikacije su otprilike 2 postotna boda lošiji za obje mreže. Navedene razlike ne izgledaju kao da bi trebale imati značajan utjecati na točnost klasifikacije. Moguće je i da je negdje drugdje napravljena greška u implementaciji.

3.2. Rezultati.

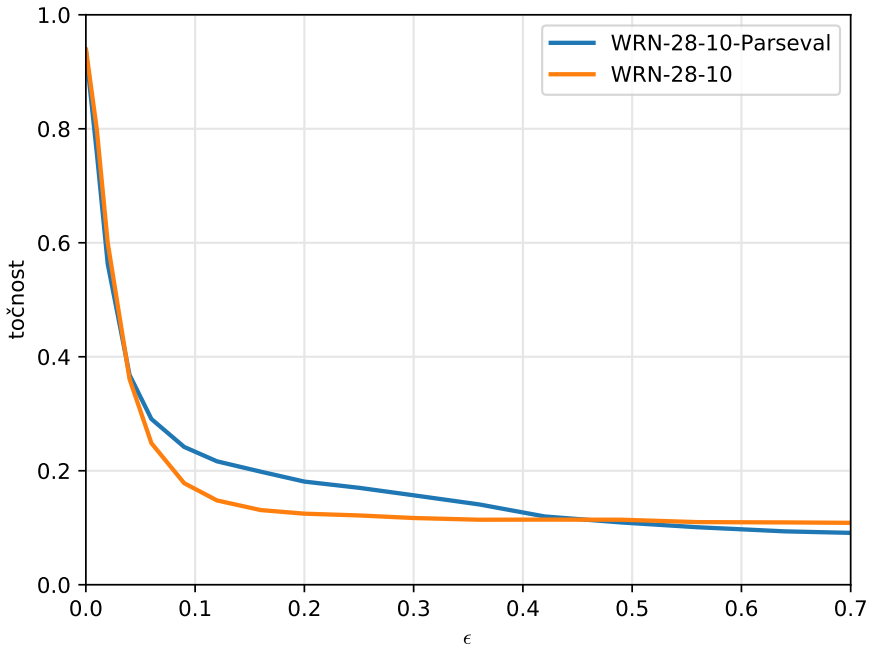
Brzina učenja i rezultati evaluacije Na slici 3.1 su grafovi ovisnosti klasifikacijske pogreške o broju završenih epoha jednog mjerenja za Parsevalovu rezidualnu mrežu i jednog za običnu rezidualnu mrežu WRN-28-10 na podskupu za testiranje skupa CIFAR-10. Krivulje su konvergirale na točnost oko 0.94. Stopa učenja je do 60. epohe 0.1, a nakon 60., 120. i 160. se smanjuje množenjem s 0.2. Glavni uzrok velikih oscilacija krivulja kod početne velike stope učenja je vjerojatno sporo ažuriranje pokretnog prosjeka statistika grupne normalizacije u odnosu na brzinu promjene parametara.

Otpornost na neprijateljske primjere Na slici 3.2 prikazane su krivulje koje pokazuju otpornost modela na neprijateljske primjere ovisno o ∞-normi pomaka dobivenog FGSM-om. Na slici 3.3 su prikazani primjeri neprijateljskih primjera



Slika 3.1: Ovisnost klasifikacijske pogreške o broju završenih epoha za Parsevalovu (plavo) i običnu (narančasto) rezidualnu mrežu. Svaka mreža se učila na uniji skupa za učenje i skupa za validaciju, a evaluirala na skupu za testiranje. Svaka krivulja je rezultat jednog mjerenja.

dobivenih za različite ϵ.



Slika 3.2: Krivulje ovisnosti točnosti o iznosu ∞-norme šuma generiranog algoritmom FGSM za mreže iz slike 3.1. Šum se dodaje normaliziranim slikama is podskupa za testiranje skupa CIFAR-10. Za dobivanje grafa su radi bržeg generiranja neprijateljskih primjera i evaluacije korištene 16384 slike iz skupa za testiranje.

Slika 3.3: Neprijateljski primjeri dobiveni FGSM-om za $\epsilon = 0, 0.02, 0.05, 0.2, 0.5, 1$ redom odozgo prema dolje.

4. Literatura

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, i Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

[2] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, i Evan Shelhamer. cudnn: Efficient primitives for deep learning. *CoRR*, abs/1410.0759, 2014. URL http://arxiv.org/abs/1410.0759.

[3] Moustapha Cissé, Piotr Bojanowski, Edouard Grave, Yann Dauphin, i Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. U *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, stranice 854–863, 2017. URL http://proceedings.mlr.press/v70/cisse17a.html.

[4] Ian J. Goodfellow, Jonathon Shlens, i Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014. URL http://arxiv.org/abs/1412.6572.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, i Jian Sun. Deep Residual Learning for Image Recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, i Jian Sun. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016. URL http://arxiv.org/abs/1603.05027.

[7] Alexey Kurakin, Ian J. Goodfellow, i Samy Bengio. Adversarial machine learning at scale. *CoRR*, abs/1611.01236, 2016. URL http://arxiv.org/abs/1611.01236.

[8] Yanpei Liu, Xinyun Chen, Chang Liu, i Dawn Song. Delving into transferable adversarial examples and black-box attacks. *CoRR*, abs/1611.02770, 2016. URL http://arxiv.org/abs/1611.02770.

[9] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, i Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *CoRR*, abs/1706.06083, 2017. URL http://arxiv.org/abs/1706.06083.

[10] Nicolas Papernot, Ian Goodfellow, Ryan Sheatsley, Reuben Feinman, i Patrick McDaniel. cleverhans v1.0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2016.

[11] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, i Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. URL http://arxiv.org/abs/1312.6199.

[12] Huan Xu i Shie Mannor. Robustness and generalization. *CoRR*, abs/1005.2243, 2010. URL http://arxiv.org/abs/1005.2243.

[13] Sergey Zagoruyko i Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. URL http://arxiv.org/abs/1605.07146.

Dodatak A
Konfiguracija rezidualnih mreža

Ime hiperparametra	**Vrijednost**	**Komentar**
Konvolucijski sloj		
inicijalizacija težina	$U(-M, M)$, $M = \sqrt{\frac{6}{k_0 k_1 H + D}}$	k_0 i k_1 su prostorne dimenzije jezgre, H semantička dimenzija ulaza, a D semantička dimenzija izlaza
inicijalna vrijednost pomaka	**0.05**	
Grupna normalizacija		
stopa eksp. pokr. prosjeka	0.95	
inicijalni pomak	**0**	
inicijalno skaliranje	**1**	
Konveksna kombinacija (Parsevalova mreža)		
koeficijent α	0.99	$(\boldsymbol{x}, \boldsymbol{r}) \mapsto \alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{r}$, $\alpha \in [0, 1]$
Rezidualni blok		
broj konvolucijskih slojeva	2	
prostorne dimenzije jezgri	3	
dropout	0.3	
način povećanja broja značajki	identitet s nadopunjavanjem nulama	
faktor proširenja k [13]	10	(semantička dimenzija blokova prve grupe je $16k$)
Rezidualna mreža		
duljine grupa	$(4, 4, 4)$	(3 grupe s po 4 bloka)
L_2 regularizacija	$5 \cdot 10^{-4}$	
stopa učenja	0.1 na početku, u 60., 120. i 160. epohi se smanjuje množenjem s 0.2	
veličina mini-grupe	128	
Rezidualna mreža (Parsevalova mreža)		
β (od R_β)	10^{-4}	

Tablica A.1: Hiperparametri testiranih rezidualnih mreža.

Dodatak B
Ostali rezultati

Blok	α
g0/b0	0.532
g0/b1	0.708
g0/b2	0.704
g0/b3	0.713
g1/b0	0.272
g1/b1	0.481
g1/b2	0.771
g1/b3	0.682
g2/b0	0.203
g2/b1	0.418
g2/b3	0.012
g2/b2	0.286

Tablica B.1: Slučajan uzorak koeficijenata slojeva konveksne kombinacije nakon oko 22 epohe kod mreže WRN-28-1 uz početne vrijednosti 0.99. "gx/by" označava y-ti blok x-te grupe.