SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1966

Optimizirane izlazne funkcije klasifikatora temeljenog na umjetnim neuronskim mrežama u domeni implementacijskih napada na kriptografske uređaje

Juraj Fulir

Zagreb, svibanj 2019.

Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu `\izvornik`.

ZAHVALA'n'STUFF

SADRŽAJ

1. Uvod

TODO: Opis problema

2. Implementacijski napadi na kriptografske uređaje

2.1. Side-channel napadi

TODO: Postoji nekoliko vrsta.
TODO: Ovdje se obrađuje DPA.

2.2. Izvedba napada

TODO: Uštekaj uređaj, osciloskop na to i to mjesto i snimaj
TODO: Provjeri mogućnosti i zaključi najvjerojatniju
TODO: Problem netraktabilnosti postupka -> neuralke <3

2.3. DPA skupovi podataka

TODO: Ima HW i ovaj pravi
TODO: Nabaci i PCA redukcije i statistike iz jn
TODO: Mjere dobrote klasifikacije
TODO: Ne zaboravi referencu na stranicu!

3. Klasifikator temeljen na umjetnim neuronskim mrežama

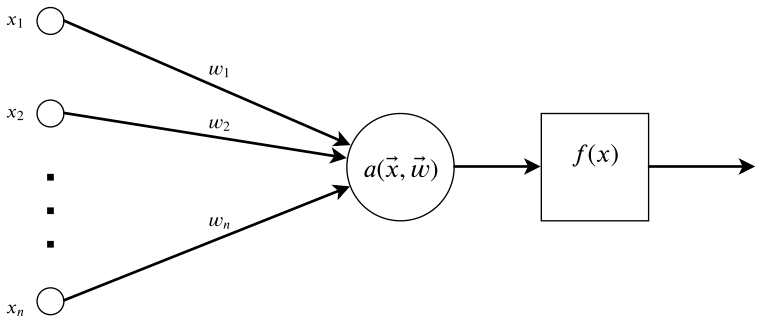
3.1. Umjetne neuronske mreže

Umjetne neuronske mreže (nadalje „neuronske mreže“) koristimo za modeliranje višedimenzijske funkcije ili distribucije kojom se aproksimira rješenje zadanog problema iz konačnog broja primjera. Vrlo su moćan alat za savladavanje teških zadataka u raznim područjima, često dostižući ljudske performanse na zadanom problemu. Danas su vrlo raširene u raznim područjima od kojih su samo neka: računalni vid (Krizhevsky et al., 2012; Redmon et al., 2016), prirodna obrada jezika (Mikolov et al., 2013; Kim, 2014) i podržano učenje (Mnih et al., 2013; Fang et al., 2017).

3.1.1. Građa

Neuronske mreže građene su od međusobno povezanih jedinica, tzv. neurona, modeliranih prema pojednostavljenom modelu biološkog neurona. Neuron očitava ulazne značajke sustava ili izlaze drugih neurona te ažurira svoje unutarnje stanje (aktivaciju) i stvara odziv (izlaz). Utjecaj ulaza na neuron vrednuje se težinama (engl. *weights*) koje definiraju kako se neuron ponaša u ovisnosti o pojedinim ulazima. Aktivacijski prag neurona (engl. *bias*) određuje jedinstvenu osjetljivost neurona na jačinu podražaja. Težine i prag neurona nazivamo parametrima neurona.
TODO: Što sve biolozi vele o neuronima? https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3812748/

Način na koji iz ulaza gradimo unutarnje stanje (aktivaciju) neurona opisujemo aktivacijskom funkcijom. Najpopularnije aktivacijske funkcije jesu afina funkcija i unakrsna korelacija. Afina funkcija je skalarni produkt vektora ulaza s vektorom težina neurona uz dodatak vrijednosti praga. Parametri neurona definiraju nagib i pomak ravnine u prostoru ulaza koja opisuje aktivaciju neurona. Primijenjuje se kada se ulazi



Slika 3.1: Prikazani osnovni dijelovi neurona su težine dendrita (w_i), aktivacijska funkcija ($a(\vec{x}, \vec{w})$) i izlazna funkcija ($f(x)$). Prag neurona nije prikazan zbog jednostavnosti dijagrama.

u model mogu zapisati vektorom značajki čiji raspored nije bitan.

$$f(\vec{x}; \underline{W}, \vec{b}) = \underline{W}^T \cdot \vec{x} + \vec{b} \tag{3.1}$$

TODO: Spomeni distance based aktivacije (ANFIS?)
TODO: Spomeni i složenije metode: (Lin et al., 2014)

Pretvorbu aktivacije neurona u izlazni signal opisujemo izlaznom funkcijom koja se detaljnije obrađuje u poglavlju 3.2. Aktivacijska i izlazna funkcija definiraju prijenosnu funkciju koja ujedno opisuje ponašanje cijelog neurona (Duch i Jankowski, 1999). U praksi se pojam aktivacijske i prijenosne vrlo često ekvivalentno koristi na mjestu pojma izlazne funkcije, no ovaj rad se drži prethodno navedene i jasnije notacije.

$$t(x) = (f \circ a)(x) = f(a(x)) \tag{3.2}$$

Povezivanjem neurona gradi se arhitektura mreže koja određuje kako podatci i gradijenti teku kroz mrežu, a time utječu na brzinu učenja i inferencije neuronske mreže. Najčešće se koriste slojevite unaprijedne arhitekture zbog jednostavnosti izvedbe. Unaprijedne arhitekture propuštaju podatke samo u jednom smjeru odnosno već izračunati neuroni se ne izračunavaju ponovno, što je posebno pogodno za optimizaciju širenjem unatrag, detaljnije opisanu u poglavlju 3.1.4. Slojevite arhitekture omogućuju paralelizaciju izvođenja operacija na grafičkim karticama što značajno ubrzava postupke učenja i inferencije. Pri definiciji slojevite arhitekture najčešće je dovoljno navesti samo redoslijed slojeva, no ponekad je potrebno definirati i način povezivanja slojeva npr. pri uporabi preskočnih veza (Srivastava et al., 2015; He et al., 2016; Huang et al., 2017). Prvi sloj služi za postavljanje ulaza mreže i nazivamo ga ulaznim slojem mreže. Posljednji sloj mreže služi nam za ekstrakciju izlaza te mjerenje kakvoće mreže i nazivamo ga izlaznim slojem mreže. Svi slojevi između ulaznog i

izlaznog sloja nazivaju se skrivenim slojevima.

Potpuno povezana arhitektura je najjednostavnija arhitektura za zadatak klasifikacije. Svaki neuron u potpuno povezanom sloju aktivira se pomoću svih izlaza iz prethodnog sloja. Za naučeni potpuno povezani sloj kažemo da vrši ekstrakciju značajki iz prethodnog sloja. Geometrijski gledano, svaki neuron vrši mapiranje značajki iz dimenzije prethodnog sloja u novu dimenziju s ciljem modeliranja boljih značajki.

Unakrsna korelacija, za razliku od afine funkcije, koristi informaciju o susjednosti ulaznih značajki. Ulaz za takav model je definiran n-dimenzijskim tenzorom, a neuron uzima samo jedan pod-tenzor ulaza (vidljiva regija) i nad njime računa skalarni produkt s n-dimenzijskim tenzorom težina (jezgrom). Kada su ulazi slike u boji ulazni tenzor ima 3 dimenzije (visina, širina, RGB kanali) pa stoga i jezgra ima 3 dimenzije, no znantno manje visine i širine. Unakrsna korelacija prozvana je konvolucijom jer radi na istom principu, a jedina razlika je da se elementi jezgre indeksiraju zrcaljeno po obje osi. S obzirom da se parametri jezgre uče automatski, nije nam bitno definirati orijentaciju jezgre.

$$f(\underline{X}; \underline{W}) = \underline{X} \circledast \underline{W} \tag{3.3}$$

TODO: Raspisat konvoluciju po elementima?

3.1.2. Postupak optimizacije umjetne neuronske mreže

Optimizacijom težina neuronska mreža prilagođava se danom zadatku, odnosno kažemo da mreža 'uči'. Optimizaciju težina najčešće izvodimo gradijentnim spustom, uz pretpostavku derivabilnosti svih komponenata neuronske mreže. Kada ta pretpostavka ne vrijedi koriste se algoritmi pretrage poput evolucijskih algoritama. U ovom radu optimizacija se vrši gradijentnim spustom.

Inicijalizacija težina

* važnost dobre inicijalizacije [slika dobre i loše inicijalizacije] * Xavier

3.1.3. Funkcija gubitka

3.1.4. Optimizacija širenjem unatrag

Optimizator

* geometrijski značaj optimizacije [slika podnaučena, generalizira, prenaučena]

Perturbacije ulaznih podataka

3.1.5. Svojstva

* Svojstva kompresije, generalizacije, univerzalne aproksimacije

3.1.6. Problemi

* Problem odabira arhitekture, hiperparametara i optimizacije * Problem pretreniranosti + adversarial primjeri

Arhitektura, prijenosne funkcije i težine definiraju neuronsku mrežu te njihov odabir značajno utječe na performanse neuronske mreže. Učenje

Derivabilne neuronske mreže optimiziraju se optimizatorom koji određuje kako se mijenjaju težine.Za ugađanje težina najčešće se koristi gradijentni spust, uz pretpostavku derivabilnosti čitave neuronske mreže. Kada pretpostavka ne vrijedi najčešće se koriste evolucijski algoritmi.

3.2. Izlazne funkcije

TODO: Bitka za odabir izlazne fje (nađi onaj rad di pljuje po sigmoidi i relu (elu rad?))
TODO: Usporedbe funkcije i derivacije
TODO: Navedene su funkcije koje su razmatrane

Ispravljena linearna jedinica (ReLU)

(engl. *Rectified linear unit*) * bez i sa cutoff

$$f(x) = \begin{cases} x, & \text{ako } x > 0 \\ 0, & \text{inače} \end{cases} \qquad f'(x) = \begin{cases} 1, & \text{ako } x > 0 \\ 0, & \text{inače} \end{cases} \tag{3.4}$$

Propusna ispravljena linearna jedinica (LReLU)

(engl. *Leaky ReLU*)

$$f(x) = \begin{cases} x, & \text{ako } x > 0 \\ \alpha x, & \text{inače} \end{cases} \qquad f'(x) = \begin{cases} 1, & \text{ako } x > 0 \\ \alpha, & \text{inače} \end{cases} \tag{3.5}$$

Ispravljena linearna jedinica s pragom (ThReLU)

(engl. *Thresholded ReLU*)

$$f(x) = \begin{cases} x, & \text{ako } x > \theta \\ 0, & \text{inače} \end{cases} \qquad f'(x) = \begin{cases} 1, & \text{ako } x > \theta \\ 0, & \text{inače} \end{cases} \tag{3.6}$$

(RReLU)

Eksponencijalno-linearna jedinica (ELU)

(engl. *Exponential linear unit*)

$$f(x) = \begin{cases} x, & \text{ako } x > 0 \\ \alpha(e^x - 1), & \text{inače} \end{cases} \qquad f'(x) = \begin{cases} 1, & \text{ako } x > 0 \\ \alpha e^x, & \text{inače} \end{cases} \tag{3.7}$$

Skalirana eksponencijalno-linearna jedinica (SELU)

(engl. *Scaled exponential linear unit*)

$$f(x) = \lambda \begin{cases} x, & \text{ako } x > 0 \\ \alpha(e^x - 1), & \text{inače} \end{cases} \qquad f'(x) = \lambda \begin{cases} 1, & \text{ako } x > 0 \\ \alpha e^x, & \text{inače} \end{cases} \tag{3.8}$$

(GELU)

Sigmoida (σ)

(engl. *Sigmoid*)

$$f(x) = \frac{1}{1 + e^{-x}} \qquad f'(x) = \sigma(x)(1 - \sigma(x)) \tag{3.9}$$

Tvrda sigmoida

(engl. *Hard sigmoid*)

$$f(x) = min(1,\ max(0,\ 0.2x + 0.5)) \qquad f'(x) = \begin{cases} 0.2, & \text{ako } x \in [-2.5, 2.5] \\ 0, & \text{inače} \end{cases} \tag{3.10}$$

Swish

$$f(x) = x\sigma(\beta x) \qquad \begin{aligned} f'(x) &= \sigma(\beta x) + \beta x \cdot \sigma(\beta x)(1 - \sigma(\beta x)) \\ &= \frac{e^x \cdot (e^x + x + 1)}{(e^x + 1)^2} \end{aligned} \tag{3.11}$$

Tangens hiperbolni (tanh)

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad f'(x) = 1 - tanh^2(x) \tag{3.12}$$

Tvrdi tangens hiperbolni

(engl. *Hard tanh*)

$$f(x) = \begin{cases} -1, & \text{ako } x < 1 \\ x, & \text{ako } x \in [-1, 1] \\ 1, & \text{inače} \end{cases} \qquad f'(x) = \begin{cases} 0, & \text{ako } x < 1 \\ 1, & \text{ako } x \in [-1, 1] \\ 0, & \text{inače} \end{cases} \tag{3.13}$$

Racionalna aproksimacija tanh

(engl. *Rational tanh*) ??????????????????

$$f(x) = 1.7159 \cdot tanh(\frac{2}{3}x), \quad \text{gdje } tanh(x) \approx sgn(x)(1 - \frac{1}{1 + |x| + x^2 + 1.41645 \cdot x^4}) f'(x) = \tag{3.14}$$

Ispravljeni tanh

(engl. *Rectified tanh*)

Softmax

$$f(\vec{x}) = \frac{e^{\vec{x}}}{\sum_i e^{\vec{x_i}}} \qquad f'(x) = \frac{e^x}{1 + e^x} \tag{3.15}$$

Softplus

$$f(x) = log(1 + e^x) \qquad f'(x) = \frac{e^x}{1 + e^x} \tag{3.16}$$

Softsign

$$f(x) = \frac{x}{1 + |x|} \qquad f'(x) = \frac{1}{(1 + |x|)^2} \tag{3.17}$$

Sinus (sin)

$$f(x) = sin(x) \qquad f'(x) = cos(x) \tag{3.18}$$

Kosinus (cos)

$$f(x) = cos(x) \qquad f'(x) = -sin(x) \tag{3.19}$$

Parabola x^2

$$f(x) = x^2 \qquad f'(x) = 2x \tag{3.20}$$

Kubna parabola x^3

$$f(x) = x^3 \qquad f'(x) = 3x^2 \tag{3.21}$$

Gauss

$$f(x) = e^{-x^2} \qquad f'(x) = -2x \cdot f(x) \tag{3.22}$$

4. Optimizacija simboličkom regresijom (tehnički genetskim programiranjem...)

4.1. Simbolička regresija

* Opis i svojstva SR * Utjecaj i brojnost parametara u GA (moš linkat i svoj završni rad :P)

4.2. Taboo evolucijski algoritam

* Problem konvergencije i stohastičnosti GP-a * EA oplemenjen taboo listom iz algoritma Taboo pretraživanja

4.3. Korišteni čvorovi i operatori (prostor pretraživanja)

* Popis čvorova * Popis operatora (un/bin)

5. Implementacija

???

6. Rezultati

6.1. 9class

6.1.1. Uobičajene izlazne funkcije

* Opis postupka pretrage * Tablica * Komentar

6.1.2. Utjecaj parametra veličine taboo liste

* Tablica * Komentar

6.2. 256class

6.2.1. Uobičajene izlazne funkcije

* Opis postupka pretrage * Tablica * Komentar

6.2.2. Utjecaj parametra veličine taboo liste

* Tablica * Komentar

7. Buduća istraživanja

* Primjena CNN na vremenskim uzorcima po uzoru na onaj rad * Ispitivanje učinkovitosti korištene optimizacije na ostalim problemima * Paralelna evolucija arhitekture i aktivacijskih fja

8. Zaključak

* Radi/Ne radi. * Pronađene zanimljivosti. * Pouka za doma.

LITERATURA

Włodzisław Duch i Norbert Jankowski. Survey of neural transfer functions. *Neural Computing Surveys*, 2(1):163–212, 1999. URL ftp://ftp.icsi.berkeley.edu/pub/ai/jagota/vol2_6.pdf.

Meng Fang, Yuan Li, i Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. U *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, stranice 595–605, Copenhagen, Denmark, Rujan 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1063. URL https://www.aclweb.org/anthology/D17-1063.

K. He, X. Zhang, S. Ren, i J. Sun. Deep residual learning for image recognition. U *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, stranice 770–778, June 2016. doi: 10.1109/CVPR.2016.90.

G. Huang, Z. Liu, L. v. d. Maaten, i K. Q. Weinberger. Densely connected convolutional networks. U *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, stranice 2261–2269, July 2017. doi: 10.1109/CVPR.2017.243.

Yoon Kim. Convolutional neural networks for sentence classification. U *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, stranice 1746–1751, Doha, Qatar, Listopad 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL https://www.aclweb.org/anthology/D14-1181.

Alex Krizhevsky, Ilya Sutskever, i Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. U F. Pereira, C. J. C. Burges, L. Bottou, i K. Q. Weinberger, urednici, *Advances in Neural Information Processing Systems 25*, stranice 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-network pdf.

Min Lin, Qiang Chen, i Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2014.

Tomas Mikolov, Kai Chen, Greg S. Corrado, i Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. URL `http://arxiv.org/abs/1301.3781`.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, i Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. URL `http://arxiv.org/abs/1312.5602`.

J. Redmon, S. Divvala, R. Girshick, i A. Farhadi. You only look once: Unified, real-time object detection. U *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, stranice 779–788, June 2016. doi: 10.1109/CVPR.2016.91.

Rupesh Kumar Srivastava, Klaus Greff, i Jürgen Schmidhuber. Highway networks. *CoRR*, abs/1505.00387, 2015.

Optimizirane izlazne funkcije klasifikatora temeljenog na umjetnim neuronskim mrežama u domeni implementacijskih napada na kriptografske uređaje

Sažetak

Proučiti postojeće metode u izgradnji izlaznih funkcija u umjetnim neuronskim mrežama. Posebnu pažnju posvetiti evolucijskim algoritmima simboličke regresije za izgradnju ciljanih funkcija. Ustanoviti moguće nedostatke postojećih algoritama ili mogućnost poboljšanja. Primijeniti evoluirane izlazne funkcije u homogenoj ili heterogenoj umjetnoj neuronskoj mreži na skupovima DPAv2 i DPAv4 te odrediti mjere kvalitete izgrađenog klasifikatora: točnost, preciznost, odziv te F mjere. Usporediti učinkovitost ostvarenih postupaka s postojećim rješenjima iz literature. Radu priložiti izvorne tekstove programa, dobivene rezultate uz potrebna objašnjenja i korištenu literaturu.

Ključne riječi: Ključne riječi, odvojene zarezima.

Title

Abstract

Abstract.

Keywords: Keywords.