



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská



Robustní strojové učení a adversariální vzorky

Robust machine learning and adversarial examples

Bakalářská práce

Autor: **Pavel Jakš**
Vedoucí práce: **Mgr. Lukáš Adam, Ph.D.**
Akademický rok: 2021/2022

- Zadání práce -

- Zadání práce (zadní strana) -

Poděkování:

Chtěl bych zde poděkovat především svému školiteli - panu doktoru Adamovi - za pečlivost, ochotu, vstřícnost a odborné i lidské zázemí při vedení mé bakalářské práce.

Čestné prohlášení:

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechnu použitou literaturu.

V Praze dne 7. července 2022

Pavel Jakš

Robustní strojové učení a adversariální vzorky

Obor: Matematická informatika

Vedoucí práce: Mgr. Lukáš Adam, Ph.D., Katedra počítačů, Fakulta elektrotechnická, České vysoké učení technické v Praze, Karlovo náměstí 13, 121 35, Praha 2

[illegible]

Robust machine learning and adversarial examples

[illegible]

Key words: keywords in alphabetical order separated by commas

Obsah

Úvod	11
1 Neuronové sítě	13
1.1 Hluboká dopředná neuronová síť	13
1.2 Konvoluční sítě	14
1.2.1 Konvoluce	14
1.2.2 Pooling	15
2 Učení neuronové sítě	17
2.1 Účelové funkce	17
2.1.1 Střední kvadratická chyba	17
2.1.2 Ztráta křížové entropie	17
2.2 Algoritmus zpětného šíření chyby	17
2.3 Algoritmy učení	17
2.3.1 Gradientní sestup	17
2.3.2 Stochastický gradientní sestup	17
2.3.3 Adam	17
3 Adversariální vzorky	19
3.1 Metody generování adversariálních vzorků	19
3.1.1 FGSM	19
3.1.2 Iterativní FGSM	19
4 Robustní učení neuronové sítě	21
Závěr	23

Úvod

Pojem neuronové sítě představuje výpočetní jednotku, která svou univerzálností nachází uplatnění v mnoha disciplínách.

Kapitola 1

Neuronové sítě

Princip fungování neuronové sítě spočívá v poskládání celku z dílčích výpočetních jednotek - umělých neuronů. Takovýto neuron je standardně funkcí více proměnných, jehož výstup je proměnná jediná. Typickým modelem umělého neuronu je funkce definovaná předpisem

$$f(x_1, \dots, x_n) = \sigma\left(\sum_{i=1}^n w_i x_i + b\right), \quad (1.1)$$

kde n je počet vstupujících proměnných, w_i jsou tzv. váhy (w z anglického slova weight), b je práh (b z anglického slova bias), σ označuje tzv. aktivační funkci.

Roli vstupujících proměnných mohou hrát např. hodnoty RGB pixelů barevných obrázků, je-li aplikací klasifikace obrázků, nebo výstupy jiných neuronů. Pod pojmem váha se skrývá míra ovlivnění výstupu neuronu daným vstupem. Je-li váha u nějakého vstupu vysoká, pak je výstup citlivější na daný vstup. Prah pro změnu určuje posunutí citlivosti neuronu na všechny vstupy jako celku.

Poslední, avšak velmi důležitou charakteristikou tohoto modelu neuronu je aktivační funkce. Za aktivační funkci lze vzít libovolnou funkci $f: \mathbb{R} \rightarrow \mathbb{R}$, existuje však základní sada:

- Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$
- ReLU: $f(x) = \max(0, x)$
- LeakyReLU: $f(x) = \max(0, x) + \alpha * \min(x, 0)$, kde $\alpha \in \mathbb{R}^+$
- Tanh: $f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Tyto funkce lze doplnit o jejich mírné modifikace. Moderní doporučenou praxí je užívat ReLU jako aktivační funkci a (1.1) jako model neuronu dle [1].

1.1 Hluboká dopředná neuronová síť

Je-li pojem umělého neuronu objasněn, lze se přesunout k jeho užití v neuronových sítích. Základní myšlenkou těchto sítí je vhodné poskládání umělých neuronů do vrstev, které dohromady tvoří síť neuronů. Taková vrstva je potom trojího druhu - vstupní, výstupní a skrytá. *Vstupní vrstva* je množina umělých neuronů, které mají za vstup výstupy problému, jehož je neuronová síť řešením. Za vstup si lze představit matici černobílých pixelů, které představují obrázek číslíce, kterou je cíl klasifikovat. *Výstupní vrstva* sestává z neuronů, které mají za vstup výstupy neuronů předchozí vrstvy. Výstupem této vrstvy pak bude řešení daného problému - například klasifikace číslíce. Posledním druhem vrstvy je *vrstva*

skrytá. Takováto vrstva má za vstupy výstupy vrstvy předcházející a její výstupy slouží jako vstupy pro vrstvu nadcházející. Má-li neuronová síť tuto architekturu, hovoří se o *dopředné neuronové síti*. Má-li navíc alespoň jednu skrytou vrstvu, lze mluvit o *hluboké dopředné neuronové síti*.

Se znalostí pojmu vrstvy neuronů lze přistoupit k poznámce o tzv. *softmax funkci*. Jedná se o vektorovou funkci $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, kde

$$f(x_1, \dots, x_n)_i = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}.$$

Její užití je nasnadě: Výstup této funkce lze totiž interpretovat jako diskrétní pravděpodobnostní distribuci, a proto ji lze užít jako aktivační funkci výstupní vrstvy, je-li cílem dané neuronové sítě klasifikace vstupu do kategorií. Potom $i_m = \operatorname{argmax}\{f(x_1, \dots, x_n)_i | i = 1, \dots, n\}$ představuje index predikované kategorie a příslušné maximum jistotu predikce.

Další poznámka se bude věnovat zjednodušení zápisu akce vrstvy na vstup. Podle modelu neuronu v (1.1) se akce jednoho neuronu na vstup sestává z násobení, následného sčítání, přičtení prahu a aplikací aktivační funkce. Tato procedura nastává pro každý neuron ve vrstvě. Tak lze sestavit z jednotlivých vah $w_i^{(j)}$ (i -tá váha j -tého neuronu ve vrstvě) matici \mathbb{A} , jejímiž prvky jsou právě ony váhy $(\mathbb{A})_{j,i} = w_i^{(j)}$, z prahů pak vektor b , kde $b_i = b^{(i)}$ - práh i -tého neuronu ve vrstvě. Potom při ztotožnění aktivační funkce σ s její vektorovou variantou, za označení vektoru vstupu vrstvy jako x_{in} a výstupu vrstvy jako x_{out} lze psát:

$$x_{out} = \sigma(\mathbb{A}x_{in} + b) \quad (1.2)$$

Tedy stěžejní operací se stává maticové násobení, respektive násobení vektoru maticí zprava.

Při tomto si lze povšimnout, že takováto neuronová síť má řadu parametrů, o kterých není jasné jak je správně nastavit. Některé parametry (například váhy a prahy) se nastavují během učení neuronové sítě, čemuž je věnována samostatná kapitola. Potom tu jsou parametry, jejichž charakter je poněkud odlišný. Jedná se o ty parametry, které zůstávají během života neuronové sítě netknuté. Jako příklad lze uvést počet neuronů ve skryté vrstvě, který se promítne v rozměrech matice vah či dimenzionalitě výstupu vrstvy. Takovýmto parametrům je přisuzován název hyper-parametry.

1.2 Konvoluční síť

Konvoluční síť nebo též *konvoluční neuronové síť* přináší svou architekturou nové možnosti zpracování dat se specifickou strukturou, do které patří například časové řady, obrázky nebo videa. Středobodem konvolučních sítí je, jak již název napovídá, operace *konvoluce*. Ta nahrazuje maticové násobení, kterým lze reprezentovat operace ve výše popsaném modelu hluboké dopředné sítě.

Další nedílnou součástí konvolučních sítí je tzv. *pooling*. Spolu s konvolucí tak tvoří mocný nástroj, který ve formě konvolučních a pooling vrstev hlubokých neuronových sítí přináší například invarianci sítě vůči malému posunutí [1].

1.2.1 Konvoluce

Operace *konvoluce* je ve vší obecnosti operace mezi dvěma funkcemi f a g . Necht' jsou dále pro jednoduchost f a g reálné funkce jedné reálné proměnné. Potom operací konvoluce, jež se standardně značí $*$, vzniká nová funkce:

$$(f * g)(t) = \int_{-\infty}^{+\infty} f(x)g(t-x)dx,$$

a to za předpokladu, že integrál na pravé straně konverguje.

1.2.2 Pooling

Kapitola 2

Učení neuronové sítě

2.1 Účelové funkce

2.1.1 Střední kvadratická chyba

2.1.2 Ztráta křížové entropie

2.2 Algoritmus zpětného šíření chyby

2.3 Algoritmy učení

2.3.1 Gradientní sestup

2.3.2 Stochastický gradientní sestup

2.3.3 Adam

Kapitola 3

Adversariální vzorky

3.1 Metody generování adversariálních vzorků

3.1.1 FGSM

3.1.2 Iterativní FGSM

Kapitola 4

Robustní učení neuronové sítě

Závěr

Text závěru....

Literatura

- [1] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*. MIT Press, 2016.
- [2] I. Goodfellow, J. Shlens, C. Szegedy, *Explaining and Harnessing Adversarial Examples*. In 'International Conference on Learning Representations', ICLR 2015.
- [3] J. Nocedal, S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.