

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ROZPOZNÁVANIE OVOCIA A ZELENINY
POMOCOU HLBOKÝCH NEURÓNOVÝCH SIETÍ
BAKALÁRSKA PRÁCA

2021
FILIP LAJČIN

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ROZPOZNÁVANIE OVOCIA A ZELENINY
POMOCOU HLBOKÝCH NEURÓNOVÝCH SIETÍ
BAKALÁRSKA PRÁCA

Študijný program: Aplikovaná informatika
Študijný odbor: Informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: Mgr. Marek Šuppa

Bratislava, 2021
Filip Lajčin



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Filip Lajčin
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Rozpoznávanie ovocia a zeleniny pomocou hlbokých neurónových sietí
Fruit and Vegetable Recognition Using Deep Neural Networks

Anotácia: Schopnosť rozpoznať druh ovocia a zeleniny na obrázkoch rôznych typov je základným predpokladom pre mnohé praktické aplikácie, od navigačných systémov pre industriálne roboty až po informačné systémy založené na webovej platforme. Existujúci výskum v oblasti spracovania obrazu naznačuje, že použitie hlbokých neurónových sietí by mohlo priniesť efektívne a kvalitné riešenie tejto úlohy.

Cieľ:

1. Zmapujte existujúci výskum v oblasti rozpoznávania ovocia a zeleniny z obrázkov, ako aj použitia neurónových sietí pre rozpoznávanie obrazu
2. Na základe existujúceho výskumu v oblasti rozpoznávania ovocia a zeleniny z obrázkov navrhnete novú metódu pre riešenie tejto úlohy pomocou hlbokých neurónových sietí
3. Navrhnutý model otestujte a vyhodnoďte jeho úspešnosť na verejne dostupných datasetoch

Literatúra: Mureşan, Horea, and Mihai Oltean. "Fruit recognition from images using deep learning." arXiv preprint arXiv:1712.00580 (2017).
Duong, Linh T., et al. "Automated fruit recognition using EfficientNet and MixNet." Computers and Electronics in Agriculture 171 (2020): 105326.

Vedúci: Mgr. Marek Šuppa
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Spôsob sprístupnenia elektronickej verzie práce:
bez obmedzenia

Dátum zadania: 15.10.2021

Dátum schválenia: 11.11.2021

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Pod'akovanie: Tu môžete pod'akovať školiteľovi, prípadne ďalším osobám, ktoré vám s prácou nejako pomohli, poradili, poskytli dáta a podobne.

Abstrakt

Slovenský abstrakt v rozsahu 100-500 slov, jeden odstavec. Abstrakt stručne sumarizuje výsledky práce. Mal by byť pochopiteľný pre bežného informatika. Nemal by teda využívať skratky, termíny alebo označenie zavedené v práci, okrem tých, ktoré sú všeobecne známe.

Kľúčové slová: jedno, druhé, tretie (prípadne štvrté, piate)

Abstract

Abstract in the English language (translation of the abstract in the Slovak language).

Keywords:

Obsah

Úvod	1
1 Východiská	3
1.1 Rozpoznávanie obrazu	3
1.2 Strojové učenie	4
1.3 Neurónové siete	6
1.3.1 Učenie neurónovej siete s učiteľom	7
1.3.2 Konvolučné neurónové siete	10
1.4 Súčasný stav	11
1.5 Nástroje na tvorbu modelov neurónových sietí	11
2 Koncept riešenia	13
2.1 Koncept riešenia	13
3 Implementácia	15
3.1 Implementácia	15
4 Výsledky	17
4.1 Výsledky	17
Záver	19
Príloha A	23
Príloha B	25

Zoznam obrázkov

1.1	Ukážka fotografií z datasetu Fruits-360	4
1.2	Overfitting modelu.	5
1.3	Vizualizácia matematického modelu neurónu. Modrou farbou je označený axon neurónu z predchádzajúceho neurónu, červenou synapsie neurónu na obrázku, zelenou je označený axon neurónu.	6
1.4	Efekt zmeny biasu a váh na krivku sigmoidy.	7
1.7	Vizualizácia minimalizácie chybovej funkcie pomocou metódy SGD - stochastic gradient descent.	9
1.8	Vizualizácia konvolúcie medzi vstupnou vrstvou a neurónom v skrytej vrstve	10
1.9	Architektúra vrstiev konvolučných neurónových sietí.	11
1.10	Ukážka augmentácie obrázkov.	12

Zoznam tabuliek

Úvod

Metódy strojového učenia sú známe už od 60. rokov 20. storočia, avšak iba v posledných dvoch desaťročiach sa ich praktické použitie dramaticky spopularizovalo. Vďaka môžeme najmä technologickému pokroku v oblasti výpočetnej techniky a takisto dostupnosti veľkého objemu dát zo širokého spektra domén.

Strojové učenie umožňuje riešenie problémov, ktoré nie sú riešiteľné použitím klasických postupov a algoritmov. Medzi tieto problémy patrí okrem iného aj rozpoznávanie vzorov, kde sa pozorovaním a analýzou podobností... TODO

Kapitola 1

Východiská

V tejto kapitole riešime teoretický základ, ktorý je potrebný na porozumenie konceptov, s ktorými pracujeme v praktickej časti práce. Predstavíme neurónové siete a ich použitie v oblasti rozpoznávania obrazu. Ďalej popisujeme knižnice, ktoré boli použité na tvorbu modelu neurónovej siete.

1.1 Rozpoznávanie obrazu

Pojem počítačové videnie, do ktorej rozpoznávanie obrazu zapadá, je známy už od roku 1966, kedy sa výskumníci z MIT počas experimentu známeho pod menom Letný projekt videnia (The Summer Vision Project) pokúšali o napodobenie ľudského vizuálneho vnímania. Z päťdesiatročného projektu sa nakoniec stalo úsilie tisícov výskumníkov po celom svete umožniť počítačom rozumieť dvojdimenziálnym reprezentáciám nášho sveta.

Ďalší výskum v tomto období sa venoval najmä extrahovaniu informácií o 3D objektoch z fotografií. Vtedy vznikali prvé algoritmy na detekciu hrán objektov, segmentáciu obrazu, či algoritmy na rekonštrukciu 3D objektov. Práve v 60. rokoch minulého storočia začala byť oblasť umelej inteligencie akademickou disciplínou a na rozpoznávanie obrazu sa čoraz viac používali práve metódy z tejto oblasti. V 21. storočí sa situácia nezmenila a najpopulárnejšie postupy na porozumenie a rozpoznávanie obrazu sú práve tie s použitím umelej inteligencie, napríklad hĺbkové neurónové siete.

K úspechu použitia neurónových sietí prispelo viacero faktorov. Výpočetná sila, ktorá je potrebná na tréning neurónových sietí s veľkým počtom parametrov je dnes veľmi dostupná a natrénovať model jednoduchej siete zvládne aj notebook strednej kategórie. Ďalším podstatným faktorom je dostupnosť dát, konkrétne pri rozpoznávaní obrazu je to dostupnosť fotografií, na ktorých je možné model natrénovať.

Pri tvorbe modelu neurónovej siete na rozpoznávanie ovocia a zeleniny z fotografií použijeme aj my verejne dostupný dataset Fruits-360, ktorý obsahuje cez 90000 obráz-

kov 131 druhov ovocia a zeleniny. Všeobecne je možné povedať, že je dnes možné nájsť datasety zo širokého spektra domén.



Obr. 1.1: Ukážka fotografií z datasetu Fruits-360

1.2 Strojové učenie

Medzi problémy, ktoré strojové učenie rieši patria napríklad klasifikácia, regresia a zhľukovanie. Klasifikácia je problém, kedy algoritmus zaradi objekty zo vstupu do jednej z predom daných tried. Regresia je typ problému, kedy algoritmus na základe závislosti medzi viacerými premennými vypočíta nejaký výstup na základe vstupných dát. Medzi praktické využitie tohto algoritmu je napríklad výpočet ceny nehnuteľnosti na základe jej parametrov. Ďalší problém, ktorý strojové učenie dokáže riešiť je zhľukovanie. Zhľukovanie je podobné klasifikácii, avšak pred jeho riešením nepoznáme počet, ani názov tried, do ktorých algoritmus objekty zo vstupnej množiny zaraďuje.

Pri strojovom učení existujú tri prístupy, ktorými sa tieto problémy riešia.

Učenie s učiteľom sa bežne používa pri klasifikácii, respektíve regresii, keďže počas trénovania vieme postupne upravovať váhy, na základe ktorých sa vypočíta objektom z trénovacej množiny výsledná trieda, respektíve hodnota. Tento koncept sa nazýva zmenšovanie chyby, bližšie sa s ním zoznámime v kapitole 1.3.

Učenie bez učiteľa sa používa napríklad pri problémoch typu zhľukovanie. Keďže predom nepoznáme správny výsledok algoritmu, nedokážeme model pri učení upravovať na základe výpočtu chyby.

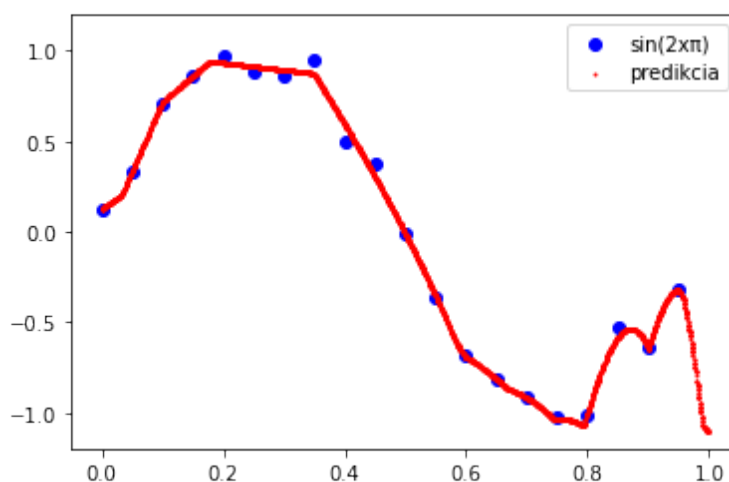
Učenie s posilňovaním funguje na princípe odmeňovania. Model sa teda rozhoduje na základe jeho skúseností a podľa akcie, ktorú si zvolí sa následne učí podľa odmeny - negatívneho, či pozitívneho hodnotenia. Tento spôsob učenia sa často používa v hrách, či multiagentových systémoch.

V tejto práci sa venujeme pridelovaniu tried obrázkom ovocia a zeleniny, teda ide o klasifikačný problém. Používať pri trénovaní teda budeme algoritmus učenia s učiteľom. Cieľom algoritmu teda je pre každý obrázok x_i z množiny obrázkov $x_i, x_i + 1, \dots, x_n$ vyprodukovať identifikátor triedy, do ktorej patrí. Klasifikačných tried môže byť ľubovoľne veľa - v spomenutom datasete Fruits-360 je množina obrázkov rozdelená do

131 druhov ovocia a zeleniny, teda do 131 disjunktných tried. Tento algoritmus teda môže byť reprezentovaný ako funkcia $f(x_i)$, ktorej výsledkom je trieda, do ktorej obrázok x_i zapadá. Definícia funkcie f spočiatku nie je známa, ale zistí sa tréňovaním na tréňovacej množine.

Všeobecne platí, že kvalita modelu zodpovedá kvalite a množstvu tréňovacej množiny. Čím viac kvalitných dát máme, tým lepší výsledok môžeme očakávať. Aj keď chyba, ktorú po natréňovaní modelu z tréňovacej množiny popisuje kvalitu modelu, nemusí byť vždy dôveryhodná. Opätovné tréňovanie na rovnakej množine objektov môže mať za následok, že je model pretréňovaný, čo znamená, že vie spoľahlivo určiť výsledok pre objekty z tréňovacej množiny, avšak pre iné objekty nepodáva rovnaký výkon. Tento dej sa označuje ako overfitting.

Overfitting si môžeme ilustrovať na jednoduchom regresnom probléme. Na obrázku 1.3 vidíme rovnomerne distribuované body po horizontálnej osi. Na body sme aplikovali funkciu $\sin(2\pi x)$ a následne sme ich hodnotu upravili o šum normálneho rozdelenia (Gaussova distribúcia). Naším cieľom je aproximovať funkciu, ktorú reprezentujú body.

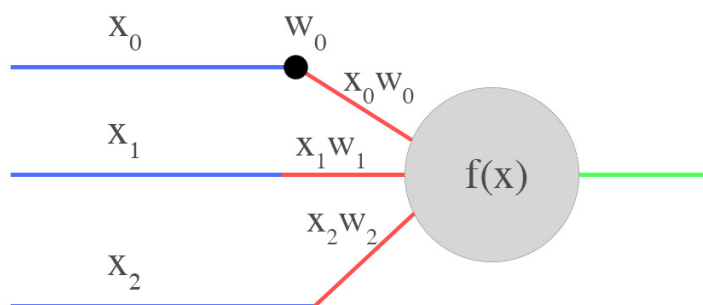


Obr. 1.2: Overfitting modelu.

Ako môžeme vidieť na obrázku, predikcia modelu je pre tréňovaciu množinu správna, avšak nie je dostatočne všeobecná. Cieľom nášho modelu bude však vedieť klasifikovať obrázky ovocia pre akékoľvek fotografie ovocia a zeleniny, teda jeho generalizačná schopnosť musí byť dostatočná. Schopnosť zovšeobecňovať na natréňovanom modeli môžeme merať pomocou merania chyby na validačnej množine objektov. Validačná množina je množina objektov, ktoré pochádzajú z tréňovacej množiny, avšak slúžia na kontrolu overfittingu pri tréňovaní modelu. Ak je chyba tréňovacej množiny dostatočne nižšia od chyby na validačnej množine, vieme, že model overfittuje a nebude podávať dobré výsledky na neznámych objektoch.

1.3 Neurónové siete

Neurónové siete boli pôvodne inšpirované biologickými nervovými systémami a matematický model neurónu sa v správaní veľmi podobá tomu biologickému. Neuróny prijímajú signály, ktoré spracujú a vyšľú upravený signál ďalej, buď do ďalších neurónov, alebo na výstup. Spôsob, akým neurón upraví prijatý signál je závislý od hodnôt zvaných váhy. Váhy korešpondujú so synaptickými silami biologického neurónu, ktoré určujú silu a smer zmeny signálu.



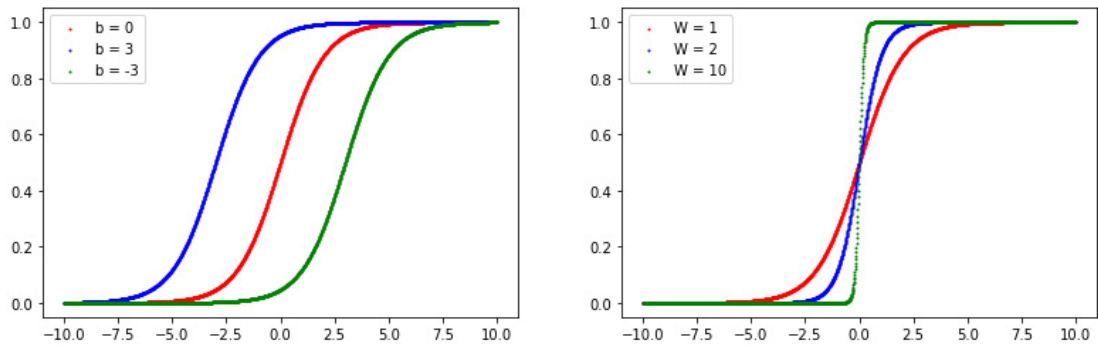
Obr. 1.3: Vizualizácia matematického modelu neurónu. Modrou farbou je označený axon neurónu z predchádzajúceho neurónu, červenou synapsie neurónu na obrázku, zelenou je označený axon neurónu.

Hodnoty váh W neurónu vopred nie sú známe a práve aj tieto hodnoty zisťujeme tréňovaním modelu neurónovej siete. V neuróne sa hodnoty váh násobia s hodnotou vstupu, čo znamená, že priamo ovplyvňujú výstup neurónu. Na sumu vynásobených váh so vstupom sa nakoniec aplikuje funkcia f , ktorá sa nazýva aktivačná. Ide o nelineárnu funkciu, ktorá ovplyvňuje silu výstupu neurónu. Okrem sumy váh vynásobených so vstupom má rolu na výstup neurónu aj "bias" b .

$$f\left(\sum_{i=1}^N Wx_i + b\right)$$

Výpočet výstupu neurónu.

Zatiaľ čo váhy upravujú stúpanie funkcie, bias posúva hodnoty funkcie po horizontálnej osi. Jeho prítomnosť je tu kritická, keďže hodnota váh je naviazaná na hodnotu vstupu a nemá plnú kontrolu nad hodnotou, ktorá je argumentom aktivačnej funkcie. Podľa voľby aktivačnej funkcie by sa teda mohlo stať, že by mal neurón nulový (žiadny) výstup, čo však vieme zamedziť biasom, ktorý vie krivku funkcie posunúť.



Obr. 1.4: Efekt zmeny biasu a váh na krivku sigmoidy.

Medzi populárne aktivačné funkcie patria napríklad sigmoida, hyperbolický tangens a ReLU.

Sigmoida stlačí jej argument do hodnoty v intervale $(0, 1)$. Jej hlavnou nevýhodou je, že aktivácie sigmoidy blízko okraju jej oboru hodnôt majú gradient blízky nule. Táto vlastnosť je nežiadúca pri spätnej úprave váh (viac v podkapitole 1.3.1), pretože zapríčiňuje pomalé učenie siete.

Hyperbolický tangens (\tanh) je podobný sigmoide, v podstate sa od sigmoidy líši zdvojnásobeným intervalom oboru hodnôt $(0, 2)$, ktorý je nakoniec znížený o 1, čiže stred oboru hodnôt má hodnotu 0.

Najpopulárnejšou aktivačnou funkciou posledných rokov je Rectified Linear Unit (ReLU) funkcia, ktorej definícia je veľmi jednoduchá, čo značne napomáha jej výpočtovej náročnosti. Ďalšou výhodou ReLU je, že umožňuje rýchle učenie, keďže narozdiel od sigmoidy sa gradient pri vysokých hodnotách neznižuje, ale od nuly zostáva rovnaký. Nevýhodou ReLU funkcie je, že pri vysokej rýchlosti učenia sa môžu váhy na neurónoch nastaviť na hodnoty, pri ktorých bude výstup neurónu 0. To znamená, že sa pri tréňovaní zbytočne stráca výpočetná sila.

$\frac{1}{1+e^{-x}}$	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	$\max(0, x)$
(a) Sigmoida	(b) Tanh	(c) ReLU

Definície populárnych nelineárnych aktivačných funkcií.

1.3.1 Učenie neurónovej siete s učiteľom

Na učenie neurónovej siete je potrebné zadať ukazovateľ, ktorý nám povie, ako veľmi sa výsledok líši od skutočného výsledku. Tento ukazovateľ sa nazýva stratová funkcia. Ako s aktivačnou funkciou neurónu, je opäť možné si vybrať z viacerých. Medzi populárne voľby pre regresné problémy patria napríklad L2 alebo MSQ (mean square

error) a L1, alebo MAE (mean absolute error). Medzi najpoužívanéjšie stratové funkcie pre klasifikačné problémy patria cross entropy loss (využívajúca softmax funkciu), alebo SVM (support vector machines).

$$\begin{array}{ll}
 -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) & \\
 \text{(a) Cross=entropy} & \text{- soft-} \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + \Delta) \\
 \text{max} & \text{(b) Multiclass SVM}
 \end{array}$$

Definície populárnych stratových funkcií pre klasifikačné problémy.

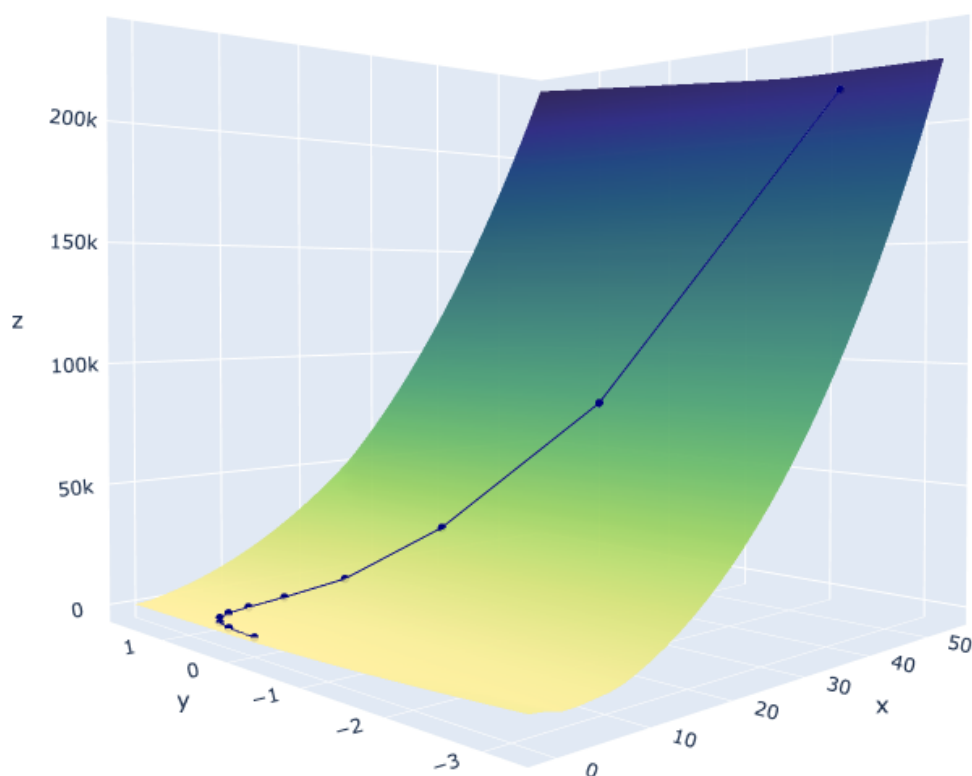
Pri tréovaní neurónovej siete sa na jej výstupe vypočíta chyba pomocou chybovej funkcie. Aby model podával čo najlepší výsledok, je zrejmé, že je potrebné chybu minimalizovať, teda maximalizovať presnosť modelu. Existuje viacero mechanizmov, ktorým sa minimalizuje chyba. Metódy, ktoré zahŕňajú náhodné vyhľadávanie vynecháme, keďže sa v praxi často nepoužívajú a nepodávajú dobré výsledky.

Keďže ukazovateľ, ktorý nám popisuje chybu vyhodnotenia modelu, je funkcia, môžeme na jej minimalizáciu použiť deriváciu. Pri funkciách s jedným parametrom dostaneme pri výpočte derivácie sklon funkcie v určitom bode, teda nám derivácia hovorí, ako funkcia reaguje na zmenu jej parametra. Keďže funkciu chceme minimalizovať, môžeme sklon použiť na nasmerovanie parametra tým smerom, ktorým výsledok chybovej funkcie znížime. Keďže chybová funkcia nebýva v neurónových sieťach jednorozmerná, používame parciálne derivácie a teda hľadáme gradient chybovej funkcie. Analogicky chceme nasledovať smer gradientu k nižším hodnotám chybovej funkcie - tejto metóde hovoríme gradient descent.

Algoritmus, ktorý použitím gradient descent metódy hľadá minimum chybovej funkcie sa všeobecne označuje ako optimalizátor. V praxi sa najčastejšie používajú algoritmy využívajúce SGD, teda stochastic gradient descent. Od gradient descentu sa líši tým, že na minimalizáciu chybovej funkcie nepoužíva všetky dáta, ale iba ich podmnožinu, čím vie znateľne rýchlejšie konvergovať k minimu chybovej funkcie. V roku 2015 sa na prestížnej konferencii pre výskumníkov hĺbkového učenia - ICLR 2015, prvýkrát prezentoval algoritmus známy pod názvom Adam - Adaptive Moment Estimation. V súčasnosti je veľmi často používaný a odporúčaný expertami z oblasti hĺbkového učenia kvôli jeho rýchlosti a efektívnosti využívania zdrojov.

Používané optimalizátory súťažia okrem iného aj v tom, ako sa im darí nachádzať globálne minimum chybovej funkcie. V prípade, že algoritmus narazí na lokálne minimum, musí použiť mechanizmy na zamedzenie zotrvania v mieste lokálneho minima a odraziť sa smerom k tomu globálnemu. Medzi tieto mechanizmy patria rôzne techniky ako simulované žihanie, či genetické algoritmy.

Pri každej iterácii optimalizátora upravujeme váhy a bias hodnoty neurónov po-



Obr. 1.7: Vizualizácia minimalizácie chybovej funkcie pomocou metódy SGD - stochastic gradient descent.

mocou metódy spätného šírenia - backpropagation. Ide o metódu, použitím ktorej sa dramaticky zefektívni učenie neurónovej siete, keďže gradienty, pomocou ktorých upravujeme váhy a biasy, sa vypočítajú iba raz pre všetky neuróny - na výstupe neurónovej siete. Následne sa váhy upravujú, smerom k vstupu, využitím vypočítaných lokálnych gradientov.

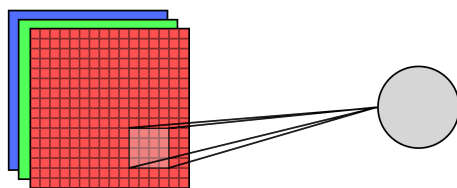
Ďalším významným činiteľom pri učení neurónovej siete je rýchlosť učenia. Ide o parameter, ktorý je potrebné určiť vopred, keďže ho neurónová sieť neoptimalizuje. Parametre takéhoto typu nazývame hyperparametre a patrí medzi ne aj celková architektúra neurónovej siete, ktorú takisto musíme vytvoriť my pred jej učením. Rýchlosť učenia je jednoducho parameter, ktorý určuje, ako veľmi sa budú meniť hodnoty váh a biasov v neurónoch. Jej hodnota teda zodpovedá aj času, ktorý trvá natrénovanie neurónovej siete. Príliš vysoké hodnoty riskujú zacyklenie optimalizátora v okolí lokálneho minima, pretože hodnoty, ktorými sa upravujú váhy a biasy sú príliš vysoké. Nižšie hodnoty majú zväčša lepšie výsledky, avšak predlžujú potrebný čas na natrénovanie siete. Často sa používa technika postupného zmenšovania rýchlosti učenia (learning rate decay), ktorá umožňuje jemnejšie úpravy váh a biasov na konci tréningového in-

tervalu siete, keďže vtedy sa nachádzame k globálnemu minimu chybovej funkcie oveľa bližšie, než na začiatku tréovania. Tým dosiahneme rýchle učenie na začiatku tréovania, avšak presnejší pohyb optimalizátora po chybovej funkcii na konci tréovacieho intervalu.

1.3.2 Konvolučné neurónové siete

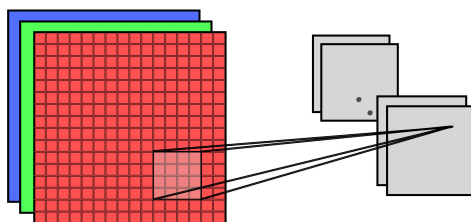
Konvolučné neurónové siete nie sú oproti obyčajným neurónovým sieťam príliš odlišné. Algoritmus učenia siete a toku dát po sieťi je totožný, v čom sa však líšia, je architektúra jednotlivých vrstiev a ich prepojenie.

Ak by sme chceli natrénovať model na klasifikáciu obrázkov pomocou bežnej neurónovej siete, komplikácie by sa nám ukázali už pri množstve parametrov, ktoré by sa sieť musela naučiť. Dôvodom je, že obrázky, aj malého rozmeru, so sebou nesú pomerne veľké množstvo dát. Farebný obrázok o rozmere 100 x 100 si môžeme predstaviť ako trojrozmernú maticu s rozmerom 100 x 100 x 3. Počet jej prvkov je 30000, čo znamená 30000 parametrov pre jeden neurón, ktoré sa sieť musí naučiť. Pri väčšom rozmere siete tento počet, samozrejme, rýchlo rastie. Tento problém konvolučné neurónové siete riešia práve s využitím konvolúcie na predchádzajúcich vrstvách. To znamená, že neuróny v skrytých vrstvách nie sú prepojené s každým neurónom v predchádzajúcej vrstve, ale používajú okno (v ang. filter/kernel), ktoré skupinu neurónov z predchádzajúcej vrstvy interpretuje ako imaginárny neurón, ktorého výstupnú aktiváciu dostávajú ako vstup. Týchto filtrov môže byť viac, čo znamená, že vrstva v konvolučnej neurónovej sieťi už netvorí stĺpec neurónov, ale trojrozmerný útvar, ktorého hĺbka je určená počtom filtrov.



Obr. 1.8: Vizualizácia konvolúcie medzi vstupnou vrstvou a neurónom v skrytej vrstve

Konvolučné neurónové siete si našli obľubu aj pri klasifikácii obrázkov, keďže ponúkajú efektívne učenie siete s dobrými výsledkami. Argumentom pre ich použitie v praxi môže byť aj pohľad do rebríčka modelov použitých na klasifikáciu, lokalizáciu a detekciu objektov v obrázkoch v prestížnej celosvetovej súťaži ILSVRC (ImageNet Large Scale Visual Recognition Challenge), kde nemalá časť najlepších modelov je založená na architektúre konvolučných neurónových sietí.



Obr. 1.9: Architektúra vrstiev konvolučných neurónových sietí.

1.4 Súčasný stav

Ako použitie umelej inteligencie a neurónových sietí vyzerá v praxi môžeme vidieť aj v našom každodennom živote. Počas pandémie ochorenia COVID-19 sa vo verejných priestranstvách začali umiestňovať zariadenia na rýchle meranie telesnej teploty ľudí bez potreby telesného kontaktu, či personálu. Výskum vakcín proti infekcii koronavírusom bol dramaticky urýchlený použitím umelej inteligencie. Zariadenia z obrazu vedeli detekovať človeka a na základe lokalizácie vyhradeného miesta na tele, určeného pre meranie telesnej teploty vedeli meranie uskutočniť automaticky, rýchlo a spoľahlivo. Pri protestoch v Honk Kongu aktivisti zhadzujú stĺpy so zariadeniami slúžiacimi na rozpoznávanie tvárí, ktoré umožňujú vláde získať detailný prehľad o životoch občanov. Prekladanie textu do obrovského množstva cudzích jazykov sa stalo možnosťou, ktorú máme k dispozícii v zariadeniach dostupných pre väčšinu obyvateľstva Zeme.

Výsledky použitia neurónových sietí teda zažívame už teraz každý deň, otázkou však už desiatky rokov zostáva - dokážu počítače naozaj myslieť? Otázka, ktorú si už v roku 1980 položil filozof John Searle, zostáva dnes sporom mnohých diskusií. Využitím neurónových sietí vieme zatiaľ iba simulovať ľudskú inteligenciu, avšak skutočné porozumenie problému, ktorý počítače riešia, je niečo, k čomu dosiaľ výskum v oblasti umelej inteligencie zatiaľ nedospel.

1.5 Nástroje na tvorbu modelov neurónových sietí

Popularizáciou neurónových sietí sa takisto spopularizovali nástroje na tvorbu modelov neurónových sietí rôznych typov. Vývoju vďaka nadšencom z oblasti, akademikom zaoberajúcim sa výskumom umelej inteligencie, či aj rôznym spoločnostiam, ktorých praktické využívanie týchto nástrojov umožnilo rapidný vývoj. Výsledok práce je vytvorený v programovacom jazyku Python, teda nástroje, ktoré používame sú s Pythonom kompatibilné.

Medzi najpopulárnejšie nástroje patrí Tensorflow s rozšírením s názvom Keras. Vďaka obrovskému počtu používateľov je o týchto nástrojoch na internete množstvo

informácií o budovaní modelov neurónových sietí. Umožňujú hárddverovú akceleráciu učenia siete, vďaka čomu je s použitím grafických kariet učenie siete násobne rýchlejšie oproti výpočtom na procesore. Tensorflow a Keras umožňujú tvorbu širokého spektra sietí na použitie v rôznych odvetviach. Medzi časté účely využitia týchto knižníc patria aj riešenia klasifikačných problémov, medzi ktoré patrí aj problém, ktorý riešime v tejto práci. Knižnice obsahujú množstvo užitočných častí, vďaka ktorým je veľmi jednoduché pracovať s obrázkami, spracovávať ich, či rozdeľovať do trénovacej a validačnej množiny. Veľmi jednoduchá je takisto augmentácia dát, ktorá jednoduchými transformáciami obrázkov vie niekoľkonásobne zväčšiť dataset, na ktorom sa sieť trénuje. Teda jednoducho vieme zamedziť overfittingu siete a zlepšiť jej schopnosť generalizovať.



Obr. 1.10: Ukážka augmentácie obrázkov.

Kapitola 2

Koncept riešenia

V tejto kapitole si povieme niečo o jadre práce a o jej členení. V zdrojovom kóde v súbore `kapitola.tex` nájdeme ukážky použitých príkazov LaTeXu potrebných na písanie nadpisov a podnadpisov a číslovaných a nečíslovaných zoznamov. Zvyšok textu tejto kapitoly je prebratý zo smernice o záverečných prácach [1, článok 5].

Jadro je hlavná časť školského diela a člení sa na kapitoly, podkapitoly, odseky a pod., ktoré sa vzostupne číslujú.

2.1 Koncept riešenia

Členenie jadra školského diela je určené typom školského diela. Vo vedeckých a odborných prácach má jadro spravidla tieto hlavné časti:

Kapitola 3

Implementácia

V tejto kapitole si povieme niečo o jadre práce a o jej členení. V zdrojovom kóde v súbore `kapitola.tex` nájdeme ukážky použitých príkazov LaTeXu potrebných na písanie nadpisov a podnadpisov a číslovaných a nečíslovaných zoznamov. Zvyšok textu tejto kapitoly je prebratý zo smernice o záverečných prácach [1, článok 5].

Jadro je hlavná časť školského diela a člení sa na kapitoly, podkapitoly, odseky a pod., ktoré sa vzostupne číslujú.

3.1 Implementácia

Členenie jadra školského diela je určené typom školského diela. Vo vedeckých a odborných prácach má jadro spravidla tieto hlavné časti:

Kapitola 4

Výsledky

V tejto kapitole si povieme niečo o jadre práce a o jej členení. V zdrojovom kóde v súbore `kapitola.tex` nájdeme ukážku použitých príkazov LaTeXu potrebných na písanie nadpisov a podnadpisov a číslovaných a nečíslovaných zoznamov. Zvyšok textu tejto kapitoly je prebratý zo smernice o záverečných prácach [1, článok 5].

Jadro je hlavná časť školského diela a člení sa na kapitoly, podkapitoly, odseky a pod., ktoré sa vzostupne číslujú.

4.1 Výsledky

Členenie jadra školského diela je určené typom školského diela. Vo vedeckých a odborných prácach má jadro spravidla tieto hlavné časti:

Záver

Na záver už len odporúčania k samotnej kapitole Záver v bakalárskej práci podľa smernice [1]: „V závere je potrebné v stručnosti zhrnúť dosiahnuté výsledky vo vzťahu k stanoveným cieľom. Rozsah záveru je minimálne dve strany. Záver ako kapitola sa nečísluje.“

Všimnite si správne písanie slovenských úvodzoviek okolo predchádzajúceho citátu, ktoré sme dosiahli príkazmi `\glqq` a `\grqq`.

V informatických prácach niekedy býva záver kratší ako dve strany, ale stále by to mal byť rozumne dlhý text, v rozsahu aspoň jednej strany. Okrem dosiahnutých cieľov sa zvyknú rozoberať aj otvorené problémy a námety na ďalšiu prácu v oblasti.

Abstrakt, úvod a záver práce obsahujú podobné informácie. Abstrakt je kratší text, ktorý má pomôcť čitateľovi sa rozhodnúť, či vôbec prácu chce čítať. Úvod má umožniť zorientovať sa v práci skôr než ju začne čítať a záver sumarizuje najdôležitejšie veci po tom, ako prácu prečítal, môže sa teda viac zamerať na detaily a využívať pojmy zavedené v práci.

Literatúra

- [1] Univerzita Komenského v Bratislave. Vnútorňý predpis č. 7/2018, Úplné znenie vnútorného predpisu č. 12/2013 Smernice rektora Univerzity Komenského v Bratislave o základných náležitostiach záverečných prác, rigorózných prác a habilitačných prác, kontrole ich originality, uchovávaní a sprístupňovaní na Univerzite Komenského v Bratislave v znení dodatku č. 1 a dodatku č. 2 smernica rektora Univerzity Komenského v Bratislave o základných náležitostiach záverečných prác, rigorózných prác a habilitačných prác, kontrole ich originality, uchovávaní a sprístupňovaní na Univerzite Komenského v Bratislave, 2013. [Citované 2020-10-19] Dostupné z https://uniba.sk/fileadmin/ruk/legislativa/2018/Vp_2018_07.pdf.

Príloha A: obsah elektronickej prílohy

V elektronickej prílohe priloženej k práci sa nachádza zdrojový kód programu a súbory s výsledkami experimentov. Zdrojový kód je zverejnený aj na stránke <http://mojadresa.com/>.

Ak uznáte za vhodné, môžete tu aj podrobnejšie rozpísať obsah tejto prílohy, prípadne poskytnúť návod na inštaláciu programu. Alternatívou je tieto informácie zahrnúť do samotnej prílohy, alebo ich uviesť na oboch miestach.

Príloha B: Používateľská príručka

V tejto prílohe uvádzame používateľskú príručku k nášmu softvéru. Tu by ďalej pokračoval text príručky. V práci nie je potrebné uvádzať používateľskú príručku, pokiaľ je používanie softvéru intuitívne alebo ak výsledkom práce nie je ucelený softvér určený pre používateľov.

V prílohách môžete uviesť aj ďalšie materiály, ktoré by mohli pôsobiť rušivo v hlavnom texte, ako napríklad rozsiahle tabuľky a podobne. Materiály, ktoré sú príliš dlhé na ich tlač, odovzdajte len v electronickej prílohe.