

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

tmp

Patrik Beka

Modelovanie ľudskej vizuálnej pozornosti metódami počítačového videnia a umelej inteligencie

Diplomová práca

Študijný program: Inteligentné softvérové systémy

Študijný odbor: Inteligentné softvérové systémy

Miesto vypracovania: Ústav počítačového inžinierstva a aplikovanej informatiky,
FIIT STU, Bratislava

Supervisor: doc. Ing. Vanda Benešová, PhD.

Máj 2018

ČESTNÉ PREHLÁSENIE

Čestne vyhlasujem, že som bakalársku prácu vypracoval samostatne, na základe konzultácií a štúdia odbornej literatúry, ktorej zoznam som uviedol na príslušnom mieste.

.....

Patrik Beka

POĎAKOVANIE

Anotácia

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijný program: Inteligentné softvérové systémy

Autor: Patrik Beka

Diplomová práca: Modelovanie ľudskej vizuálnej pozornosti metódami počítačového videnia a umelej inteligencie

Vedúci práce: doc. Ing. Vanda Benešová, PhD.

Máj 2018

Annotation

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: Intelligent Software Systems

Author: Patrik Beka

Master thesis: The modeling of human visual attention using computer vision and artificial intelligence

Supervisor: doc. Ing. Vanda Benešová, PhD.

May 2018

Obsah

1	Úvod	1
2	Vizuálna pozornosť	3
2.1	Bottom-up spracovanie scény	3
2.2	Top-down spracovanie scény	3
2.3	Existujúce modely vizuálnej pozornosti	3
3	Neurónové siete	5
3.0.1	Typy neurónových sietí	10
3.0.2	Konvolučné neurónové siete	11
4	Problémová oblasť	13
4.1	Metriky používané na ohodnotenie modelov vizuálnej pozornosti .	13
5	Návrh	15
5.1	Prvotné experimenty	15
5.2	Návrh neurónovej siete	15
5.3	Dataset	17
6	Zhrnutie	19
	Literatúra	21
A	Technická dokumentácia	i
A.1	Hardvér	i
A.2	Používané knižnice	i
A.2.1	TensorFlow	i
A.2.2	Matplotlib	ii

Zoznam obrázkov

1	Itti-ho hierarchický model vizuálnej pozornosti	4
2	Jednoduchá neurónová sieť	10
3	Vrstva združovania - príklad vzorkovania	12
4	Konvolučná neurónová sieť	12
5	Návrh neurónovej siete	16

1 Úvod

Vizuálna výraznosť zhora nadol (z angl. top-down) je veľmi dôležitou súčasťou vizuálnej pozornosti, nakoľko sa pri nej berie do úvahy sémantický kontext pozorovanej scény. To je niečo, čo zohráva kľúčovú úlohu pri snahe výskumníkov umožniť počítačom vnímať a porozumieť obrázkom rovnakým spôsobom ako ľudia. V posledných rokoch sa ako prelomovým v tejto doméne javí nevídaný pokrok v oblasti umelej inteligencie, konkrétne neurónových sietí. Ich charakteristickým znakom je využitie veľkého množstva hierarchickým vrstiev pre spracovanie nelineárnych informácií. Vďaka tomu sa objavili nové možnosti detekcie salientných častí skúmanej scény na základe jej sémantického kontextu.

2 Vizuálna pozornosť

Termín vizuálna pozornosť možno definovať ako súbor všetkých faktorov, ktoré ovplyvňujú naše mechanizmy výberu podstatných častí v scéne, nezáležiac od toho, aké tieto mechanizmy sú. Tento pojem je často zamieňaný s vizuálnou pútavosťou (výraznosťou?), avšak tieto dva termíny nevyjadrujú úplne to isté. Presnejšou definíciou vizuálnej pútavosti (z angl. visual saliency) je, že sa jedná o značne subjektívnu perceptuálnu vlastnosť, vďaka ktorej niektoré veci vo svete (v scéne) vyčnievajú v porovnaní so svojimi susedmi kvôli ich vlastnostiam ako farba, jas, kontrast či orientácia a preto okamžite upútajú pozornosť.[7]

2.1 Bottom-up spracovanie scény

Vizuálne stimuly (i?), ktoré upútajú pozornosť automaticky, mimovoľne, sa nazývajú bottom-up stimuly (i?).

2.2 Top-down spracovanie scény

Tá sa spolu s pútavosťou (výraznosťou) delí na dva typy a to:

- pozornosť „zdola nahor“ (z angl. bottom-up) - riadená čisto iba pútavými stimulmi, ktoré sú v podstate známkou toho, že „táto lokácia je značne odlišná od svojho okolia a stojí za pozornosť“. Je rýchla a nevedomá.
- pozornosť „zhora nadol“ (z angl. top-down) - riadená tzv. predvídateľnými mechanizmami, ako napríklad hľadáním niečoho konkrétneho na webovej stránke. Je pomalšia, vedomá a ovplyvnená našou pamäťou.

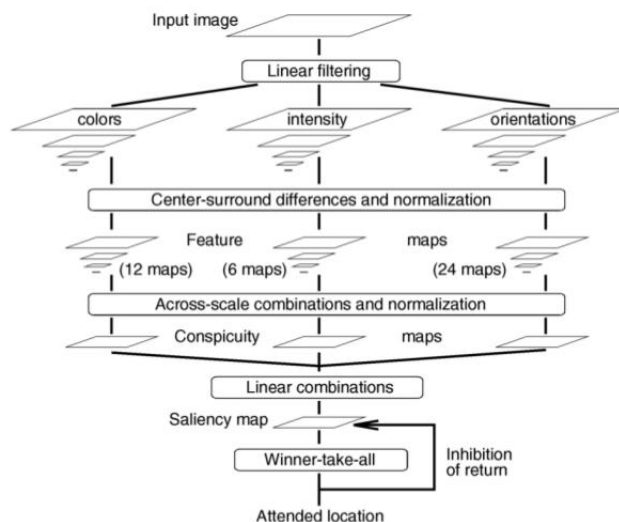
2.3 Existujúce modely vizuálnej pozornosti

Existujúce modely vizuálnej pozornosti[12] možno rozdeliť nasledovne:

- hierarchické - využívajú hierarchické rozkladanie príznakov
- Bayesove - využívajú kombináciu výraznosti s predchádzajúcimi znalosťami

- rozhodovaco-teoretické - využívajú diskriminačnú teóriu výraznosti
- informaticko-teoretické - využívajú maximalizáciu informácie z daného prostredia
- grafické - predikcia výraznosti je založená na grafových algoritmoch
- vzorovo klasifikačné - využívajú strojové učenie zo vzorov s výraznými črtami

Jedným z najznámejších modelov vizuálnej pozornosti je Itti-ho hierarchický model[4]. Je to biologicky inšpirovaný bottom-up model, ktorý využíva hierarchické rozloženie vlastností a ich kombináciu do výslednej mapy výraznosti (z angl. saliency map). Ako je vidieť na obrázku nižšie, z obrázka sa vytvoria 3 typy máp a to podľa farby, intenzity a orientácie, ktorých kombináciou sa dosiahne už spomenutá mapa výraznosti.



Obr. 1: Itti-ho hierarchický model vizuálnej pozornosti[4]

3 Neurónové siete

Neurónová sieť je abstraktný výpočtovový model založený na princípe reálnych biologických neurosystémov. Základnou stavebnou jednotkou je tak rovnako ako u neurónových sietí živočíchov neurón, resp. model neurónu[1]. Ten spracováva rôzne množstvo vstupov (N) a výstupov (M). V minulosti sa zvykol vyjadrovať podľa nasledovnej matematickej špecifikácie:

$$o_i^{k+1} = f \left(\sum_{j=1}^N w_{ij}^k * o_j^k - \theta_i^{k+1} \right) \quad (1)$$

Pre vyššie uvedené platí:

$0 < i \leq M$

$0 < j \leq N$

o_i^{k+1} - výstupná hodnota i-teho neurónu patriaceho k+1 vrstve

k - číslo vrstvy

θ_{ij}^k - prah stimulácie i-teho neurónu k+1 vrstvy

w_{ij}^k - váha medzi i-tým neurónom vrstvy k+1 a j-tým neurónom vrstvy k

f() - funkcia

V súčasnosti sa však používa radšej matematické vyjadrenie zobrazené v rovnici 2. Vypustil sa z neho prah stimulácie neurónu, miesto ktorého sa používa tzv. predsudok (z angl. bias), čo je niečo ako predpokladaná hodnota (náš chýbajúci prah stimulácie) neurónu. Tá sa časom samozrejme mení.

Predpokladajme, že máme $m+1$ vstupov so signálmi od x_0 po x_m a váhami od w_0 po w_m . Obvykle sa vstupu x_0 prideli hodnota +1, čím sa stane predsudkom vstupu s $w_{k0} = b_k$. To necháva potom iba m vstupov do neurónu, od x_1 do x_m . Samotný výstup z k-teho neurónu je potom matematicky vyjadrený nasledujúcou rovnicou:

$$y_k = \phi \left(\sum_{j=0}^m w_{kj} * x_j \right) \quad (2)$$

Pre vyššie uvedené platí:

y_k - výstup k-teho neurónu

w_{kj} - váha j-teho neurónu spojeného s k-tým neurónom na ďalšej vrstve

x_j - j-ty neurón

ϕ - funkcia

Neurónová sieť sa môže skladať z viacerých vrstiev, na ktorých sú umiestnené neuróny. Prvá vrstva sa nazýva vstupná, posledná výstupná. Medzi nimi môže byť ľubovoľný počet skrytých vrstiev rôzneho typu. Každá vrstva (s výnimkou výstupnej) by mala ešte navyše obsahovať aktivačnú funkciu - matematické vyjadrenie použité k aproximácii vplyvu na neurón. V našom prípade sa jedná o neurón umelý a funkciu, ktorá definuje výstup neurónu pre vstup alebo sadu vstupov. Aktivačných funkcií existuje niekoľko typov, každá vhodná na iný typ úloh. Niektoré z nich sú popísané nižšie.

Aktivačné funkcie

- **Softmax**

Funkcia softmax¹ (inak aj normalizovaná exponenciálna funkcia) normalizuje daný n dimenzionálny vektor tak, že upraví jeho hodnoty do rozsahu (0,1), pričom ich súčet bude rovný 1. Jej matematické vyjadrenie je nižšie.

$$S_{vec_j} = \frac{e^{vec_j}}{\sum_{i=1}^n e^{vec_i}} \quad (3)$$

Pre vyššie uvedené vyjadrenie platí:

$$\forall j \in 1..n$$

vec - konkrétny vector

Keď si ako príklad vezmeme jednoduchý vektor [1, 2, 3], výsledok po aplikovaní softmaxu bude [0.09, 0.24, 0.67]. Ako môžeme vidieť, funkcia sa väčšinou používa na zvýraznenie väčších hodnôt a zároveň potlačenie hodnôt, ktoré sú výrazne menšie ako maximálna hodnota.

- **ReLU**

¹<http://eli.thegreenplace.net/2016/the-softmax-function-and-its-derivative/>

Upravená lineárna jednotka (z angl. rectified linear unit) je funkcia v tvare:

$$f(x) = \max(0, x) \quad (4)$$

kde x je vstup do neurónu. Používa sa vďaka svojej jednoduchosti, keďže neobsahuje žiadne komplikované výpočty, čoho dôsledkom je aj jej značná rýchlosť. Jej využitie je možné pozorovať napríklad pri hlbokých neurónových sieťach.

- **Softplus**

Je v podstate aproximáciou k predošlej ReLU s matematickým vyjadrením:

$$f(x) = \ln(1 + e^x) \quad (5)$$

Rovnako ako pri ReLU je oborom hodnôt interval $(0, \infty)$. Jej využitie je napríklad pri rozoznávaní reči.

- **Sigmoid**

Táto funkcia sa používa hlavne keď je potrebné pracovať s pravdepodobnosťami, keďže jej výstup tvorí interval $(0, 1)$. Jej matematické vyjadrenie je nasledovné:

$$S(t) = \frac{t}{1 + e^{-t}} \quad (6)$$

- **Tanh**

Hyperbolický tangens. Často sa používa v rovnakých prípadoch ako Sigmoid, keďže matematicky sa dá vyjadriť aj za použitia Sigmoidu. Jeho vzorec je nasledovný:

$$\tanh(x) = \frac{\cosh(x)}{\sinh(x)} = \text{Sigmoid}(2x) - \text{Sigmoid}(-2x) = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (7)$$

Učenie sa

Základným prvkom toho, aby bola neurónová sieť schopná riešiť úlohy je učenie sa. Existujú viaceré typy učenia sa neurónovej siete, za zmienku stojí

napríklad učenie sa s učiteľom (z angl. supervised learning) a učenie sa bez učiteľa (z angl. unsupervised learning[6]). Hlavným rozdielom medzi nimi je, že učenie s učiteľom musí prebiehať na predpripravenom datasete, ktorý musí obsahovať nejaké testovacie vstupné dáta (pre ktoré chceme vypočítať výstupnú funkciu) a takzvané štítky (z angl. labels), ktoré sú v podstate naše očakávané výstupy. Učenie sa bez učiteľa naproti tomu odvodzuje funkciu k popisu skrytej štruktúry z neoštiekovaných dát, teda bez štítkov, ktoré nám určujú očakávané výstupy. Nie je tu teda žiadna chyba ani signál k ohodnoteniu potenciálneho riešenia.

Príkladom učenia s učiteľom môže byť napríklad jednoduchá neurónová sieť, ktorá má riešiť funkciu XOR[1], kedy potrebujeme reprezentovať vstupné dáta ako dvojicu núl a jednotiek. Štítkami sú v tomto prípade očakávané výstupy, takže napríklad pre vstup (dvojicu) [0,1] je štítkom 1. Takto pripravený dataset pre učenie sa by mal byť veľmi rozsiahly aby sa dosiahla maximálna presnosť. Ďalej je potrebné použiť niektorý z algoritmov učenia. Široko používaným je na takýto typ úloh algoritmus učenia spätného šírenia (z angl. backpropagation). Tento algoritmus sa snaží minimalizovať chybu pri učení a to tak, že najprv sa vypočíta chyba na poslednej (výstupnej) vrstve. Tá sa potom šíri späť k vstupnej vrstve a aktualizujú sa váhy jednotlivých neurónov. V kombinácii s algoritmami učenia sa používajú optimačné algoritmy ako Gradient descent optimizer[14] alebo Adam optimizer[9], popísané nižšie. Tieto algoritmy sú určené k nájdeniu minima funkcie medzi váhami.

Gradient descent optimizer

Je to iteratívny algoritmus používaný k nájdeniu lokálneho minima funkcie, kedy podniká kroky k nájdeniu záporného gradientu² funkcie v aktuálnom bode. To je využívané pri určovaní rýchlosti učenia sa neurónovej siete.

Existujú 3 hlavné varianty gradient descent optimizéru, ktoré počítajú sklon (gradient) funkcie. Delia sa hlavne podľa množstva dát určenému k spracovaniu, kedy sa robí kompromis medzi presnosťou aktualizácie parametra a časom, ktorý je potrebný na vykonanie tejto aktualizácie. Týmito typmi sú:

- Dávkový gradient descent:

² zmena veličiny v závislosti od inej premennej

Z angl. Batch gradient descent. Gradient sa počíta pre celý tréningový dataset, takže pre jednu aktualizáciu je potrebné ho prejsť celý a preto môže byť veľmi pomalý.

- Stochaistický gradient descent:

Tento typ je presným kontrastom voči dávkovému gradient descentu. Aktualizácia sa uskutočňuje pre každú vzorku z tréningového datasetu.

- Mini-dávkový gradient descent:

Je kompromisom medzi predošlými dvomi typmi. Aktualizácia prebieha pre malú dávku (batch) z datasetu o veľkosti n vzoriek.

Adam optimizer

V podstate vychádza priamo zo Stochaistického gradient descent optimizéru, resp. jeho modifikácie RMSProp algoritmu[18]. Rozdiel oproti Gradient descent optimizéru je ale v tom, že je schopný variabilne určovať rýchlosť učenia neurónovej siete.

Ftrl optimizer

Vychádza z algoritmu učenia FTRL-Proximal[11], celým názvom Nasleduj regularizovaného vodcu (z angl. Follow The (Proximal) Regularized Leader). Tento algoritmus je bez regularizácie v podstate identický s gradient descentom, avšak používa alternatívnu reprezentáciu koeficientov váh a tak môže byť regularizácia implementovaná efektívnejšie.

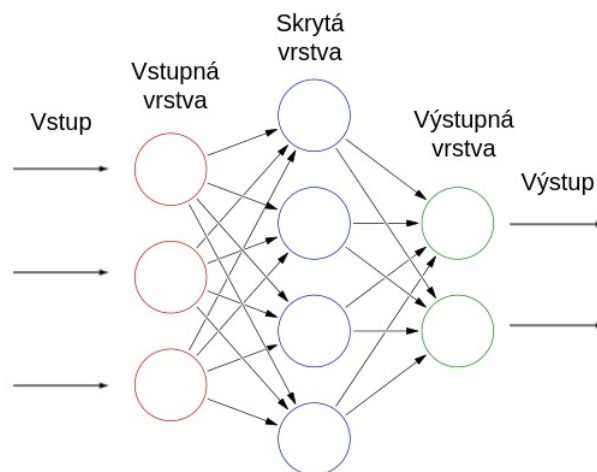
Po fáze učenia sa nasleduje validácia, pri ktorej sieť nemá prístup k štítokom. Na záver sa prejde k samotnému testovaniu neurónovej siete, kedy sa do nej posúvajú dáta rovnako bez toho, aby sieť mala prístup k štítokom. Na základe jej predikcie a štítokom k testovacím dátam sa určí jej presnosť. Na učenie, testovanie a validáciu by nemali byť použité tie isté dáta. Pomer dát k jednotlivým fázam by mal byť 80-10-10[13], čiže 80% dát je určených na učenie sa, 10% na validáciu a 10% na samotné otestovanie predikcií modelu siete.

Aj keď neurónové siete dokážu efektívne riešiť veľké množstvo úloh, problémom stále zostáva mať k dispozícii dostatok dát k učeniu neurónovej siete ešte pred riešením úloh. Taktiež je potrebné mať dostatok výpočtovej sily, aby sa problém neriešil prídlhý čas, a dostatok pamäte, keďže neurónové siete jej potrebujú značné množstvo.

3.0.1 Typy neurónových sietí

Neurónové siete majú niekoľko typov, ktoré sa rozlišujú hlavne podľa spôsobu prepojenia neurónov, ale aj podľa typu úloh, na ktoré sú určené, či podľa počtu vrstiev neurónov alebo štýlu učenia.

Najjednoduchší typ možno zobrazit' ako jednu vstupnú vrstvu, jednu skrytú a jednu výstupnú, neuróny sú tu poprepájané z n -tej vrstvy do $n+1$ vrstvy, ako je možné vidieť na obrázku 2. Tento typ sa nazýva dopredná neurónová sieť (z angl. feedforward neural network) a môže mať aj viac ako len jednu skrytú vrstvu. Používa sa hlavne ak sa jedná o predikciu nelineárnej funkcie (napríklad carbon-13 NMR chemické posuny alkánov[17]).



Obr. 2: Príklad jednoduchej neurónovej siete

Zložitejším typom sú rekurentné neurónové siete. Už z názvu vyplýva, že jednou z vecí, ktoré umožňujú, je rekurenciu. Vďaka nej prepojenia neurónov už nie sú jednosmerné len z jednej vrstvy na druhú, ale umožňuje prepojiť neuróny

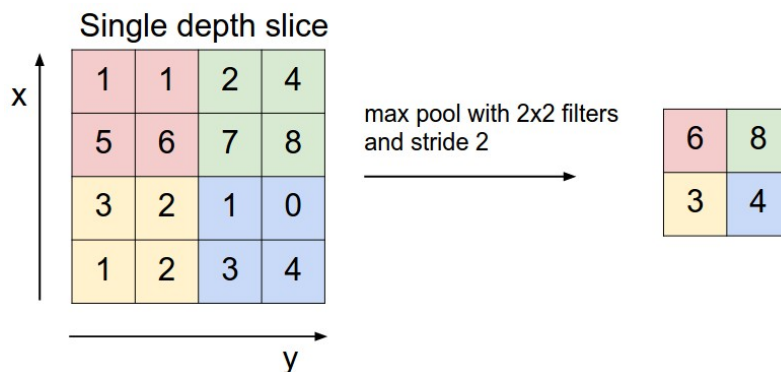
akokoľvek a tak vytvárať napríklad slučky či cykly. To dovoľuje zachytiť aj dynamické časovo obmedzené správanie a používať kontext z minulosti (avšak len niekoľko krokov dozadu), teda použiť niečo ako krátkodobú „pamäť“. Na rozdiel od doprednej neurónovej siete je možné spracovať aj ľubovoľnú sekvenciu vstupov. V praxi to znamená, že keď chceme napríklad predikovať ďalšie slovo vo vete, je dobré vedieť, ktoré slová boli pred ním. Tento typ sietí sa používa napríklad pri rozpoznávaní reči[15] alebo písma[5], či generovaní popisu k obrázkom[8], kedy však funguje v kombinácii s konvolučnou neurónovou sieťou (z angl. convolutional neural network). Tá je použitá na klasifikáciu obrázkov a rozoznávanie objektov, rekurentná sieť je použitá iba na výsledné generovanie jednoduchého popisu. Konvolučná neurónová sieť je ďalším typom neurónovej siete, ktorá sa používa pri práci s obrázkami (rozpoznávanie objektov, atď.). Podrobnejšie je tento typ popísaný nižšie, nakoľko je to typ, s ktorým budeme pracovať aj neskôr.

3.0.2 Konvolučné neurónové siete

Základ tejto siete tvorí vstupná konvolučná vrstva³ s konvolučným filtrom, ten býva väčšinou malý (3x3, 5x5). Vstup tejto vrstvy musí byť v tvare $m \times m \times r$, kde m je šírka a výška obrázku, r je počet farebných kanálov. Napríklad pre RGB obrázok je $r=3$ (červená, zelená, modrá). Konvolučným filtrom sa prejde celý obrázok a výstupom z tejto vrstvy je niekoľko filtrov. Tie sa potom spracúvajú v ďalšie vrstve združovania (z angl. pooling layer[10]), ktorá tieto filtre rozvzorkuje. To prebieha nezávisle na každom získanom filtri z konvolučnej vrstvy. Rozvzorkovanie v podstate znamená, že sa zmení veľkosť filtrov použitím operácie MAX. Najbežnejšou formou spomínanej vrstvy je verzia s oknom (filtrom) o veľkosti 2x2 aplikovaným s krokom veľkosti 2. Toto sa dá jednoducho vysvetliť ako prejdienie každého výstupu z konvolučnej vrstvy oknom uvedenej veľkosti postupne po 2 políčkach na šírku aj výšku, pričom z každej štvorice v okne sa získa MAX operáciou maximum, s ktorým sa pracuje ďalej. Na obrázku⁴ 3 je vidieť výsledok popisovaného postupu na jednoduchom príklade.

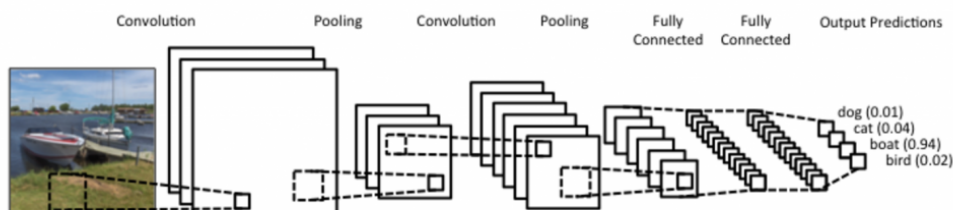
³<http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/>

⁴<http://cs231n.github.io/convolutional-networks/>



Obr. 3: Príklad vrstvy združovania, pri ktorom sa rozvzorkuje výstup z konvolučnej vrstvy o veľkosti 4x4, filtrom 2x2, s krokom veľkosti 2 za použitia operácie MAX

Takýchto konvolučných vrstiev s vrstvami združovania môže byť aj viac, nemusia ani nutne nasledovať po sebe. Po týchto vrstvách nasleduje plne prepojená vrstva alebo vrstvy (z angl. fully-connected layers), čo je vrstva, v ktorej majú neuróny plné spojenie so všetkými aktiváciami v predošlej vrstve, rovnako ako pri bežných neurónových sieťach. Aktivačnou funkciou neurónov na tejto vrstve býva väčšinou ReLU. Po plne prepojenej vrstve (vrstvách) už nasleduje iba výstupná vrstva. Na obrázku⁵ nižšie je jednoduchý náčrt vyššie popísané konvolučnej neurónovej siete.



Obr. 4: Príklad konvolučnej neurónovej siete

Využitie tohto typu je v podstate všade, kde sa jedná o rozpoznávanie obrázkov. Či už ide o automatické vyznačenie tvárí pre označenie na facebooku, autonómne

⁵<https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

vozidlá, ktoré sa vedia riadiť sami (autopilot) alebo triedenie uhoriek na farmách v Japonsku⁶. Na tento konkrétny softvér bol použitý príklad kódu jednoduchkej konvolučnej siete z tutoriálu⁷ pre TensorFlow (knihnica pre prácu s neurónovými sieťami), s modifikáciou konvolučnej a združovacej vrstvy tak, aby bola sieť uspokojená počtu tried uhoriek (10) a ich formátu obrázkov.

Z mnohých pokusov o autonómnú jazdu stoja za zmienku hlavne tie od Tesly a Google. Prototyp autonómneho systému vozidla od Google-u Dave-2[2] využíva model neurónovej siete s 9 vrstvami, jednu normalizačnú, 5 konvolučných a 3 plne prepojené vrstvy. Kamerami spracovaný obraz okolia s frekvenciou 10 snímkov za sekundu (tak nízky počet preto, aby sa predišlo veľkému množstvu príliš podobných obrázkov) je po jednom snímku rozdelený do YUV⁸ úrovni a posunutý do neurónovej siete.

4 Problémová oblasť

4.1 Metriky používané na ohodnotenie modelov vizuálnej pozornosti

Tradične sa tieto modely evaluujú vzhľadom na pohyb očí, resp. samotné fixácie. K tomu slúži značný počet rôzne fungujúcich metrík[3], najpoužívanejšie sú:

- NSS - Normalizovaná cesta pútavosti (z angl. Normalized Scanpath Saliency). Využíva priemer hodnôt pútavosti na n fixácií v normalizovanej mape podľa nasledovného vzorca:

$$\frac{1}{n} \sum_{i=1}^n \frac{s(x_h^i, y_h^i) - \mu_s}{\sigma_s} \quad (8)$$

- AUC - Oblasť pod ROC krivkou (z angl. Area Under the ROC Curve). Ľudské fixácie sú považované za pozitívnu sadu a niektoré body na obrázku

⁶<https://cloud.google.com/blog/big-data/2016/08/how-a-japanese-cucumber-farmer-is-using-deep-learning-and-tensorflow>

⁷<https://www.tensorflow.org/versions/0.6.0/tutorials/mnist/pros/index.html>

⁸farebný priestor používaný vo video aplikáciách

sú vybrané ako negatívna sada. K mape pútavosti je potom pristupované ako k binárnemu klasifikátoru na separáciu pozitívnych vzorkov od negatívnych. Presnosť podľa tejto metriky je daná nasledovne:

- 0.90 - 1 = výborná
 - 0.80 - 0.90 = dobrá
 - 0.70 - 0.80 = priemerná
 - 0.60 - 0.70 = slabá
 - 0.50 - 0.60 = veľmi slabá
- sAUC - Zamiešaná oblasť pod ROC krivkou (z angl. shuffled Area Under the ROC Curve) je mierna modifikácia vyššie uvedenej metriky, kedy ako negatívna sada nie sú vybrané len niektoré body, ale všetky body, ktoré nie sú ľudskými fixáciami, sú považované za negatívne. Určenie presnosti na základe hodnôt je rovnaké ako pri AUC.
 - CC - Korelačný koeficient, určuje prakticky podobnosť v tomto prípade dvoch máp výraznosti, kde jedna je výsledok modelu vizuálnej pozornosti a druhá je reálna mapa vypočítaná z fixácií.

$$CC(s, h) = \frac{cov(s, h)}{\sigma_s \sigma_h} \quad (9)$$

5 Návrh

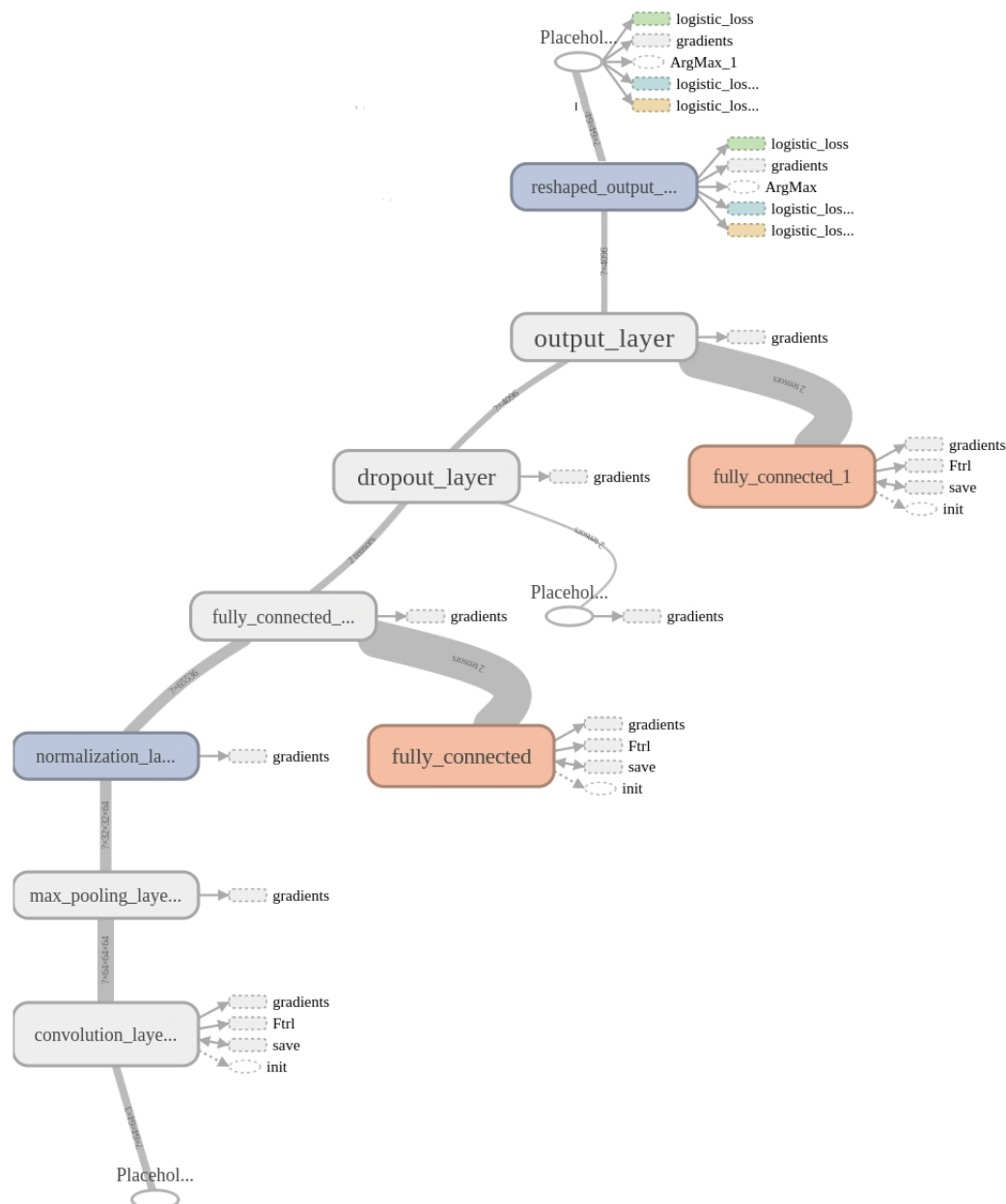
V nasledujúcich riadkoch tejto kapitoly je rozobratý postupný návrh neurónovej siete určenej k riešeniu zadaného problému, spolu s dvomi typmi datasetov, s ktorými sme pracovali, v časti 5.3.

5.1 Prvotné experimenty

5.2 Návrh neurónovej siete

K predikcii pohľadov na webové stránky je vhodné použiť konvolučnú neurónovú sieť, keďže webstránky sú vo forme obrázkov. Neurónová sieť bude predikovať priamo výslednú teplotnú mapu pohľadov (mapu výraznosti). Sieť by mala pozostávať z konvolučnej a združovacej vrstvy (vrstiev) pre spracovanie obrázku, nasledovaných normalizačnou vrstvou, plne prepojenou vrstvou a vrstvou výpadku (z angl. dropout layer). Za nimi nasleduje už len výstupná vrstva. Ako aktivačnú funkciu sme vybrali sigmoid, nakoľko sa bude predikovať mapa výraznosti, t. j. v podstate pravdepodobnosť pohľadu. Celá architektúra je načrtnutá na schéme na obrázku 5 vytvorenej pomocou nástroja TensorBoard⁹.

⁹https://www.tensorflow.org/get_started/summaries_and_tensorboard/



Obr. 5: Grafická schéma neurónovej siete, zdola vstup (obrázok web stránky), vrstvy neurónovej siete, výstup (predikovaná mapa výraznosti)

Na konvolučnej vrstve sa použije konvolučný filter o veľkosti 5x5, ktorým sa

prejde vstupný obrázok. Výstupom z nej budú mapy (obrázky) po aplikovaní filtra. Tieto dáta budú spracovávané vo vrstve združovania s použitím operácie MAX, okna (filtra) s veľkosťou 2x2 a posunom s veľkosťou 2. Výstup všetkých filtrov je zlúčený do jednej širokej vrstvy a to normalizačnej. Z nej nasledujú prepojenia do plne prepojenej vrstvy, za ktorou nasleduje vrstva výpadku[16]. Tá prakticky obsahuje hodnotu (od 0 do 1) určujúcu koľko neurónov bude dočasne „odstránených“ z modelu spolu so všetkými spojeniami, ako vstupnými tak aj výstupnými. To by malo zabrániť pretrénovaniu siete. Za vrstvou výpadku nasleduje už len výstupná vrstva, ktorá je nakoniec ešte aj transformovaná do 2D matice reprezentujúcej predikovanú mapu výraznosti.

Aktivačná funkcia sigmoid bude použitá iba na prvej konvolučnej vrstve a na plne prepojenej vrstve. K trénovaniu sa použije FTRL optimizér spolu s náhodným generátorom dát, ktorý bude z datasetu určenému pre fázu trénovania náhodne vyberať dáta na učenie. Tým sa zabezpečí simulácia náhodnosti vstupných dát a teda sa sieť bude schopná lepšie učiť.

5.3 Dataset

6 Zhrnutie

Vypracovaná prvá časť bakalárskeho projektu obsahuje analýzu nosnej časti pripravovaného riešenia, neurónových sietí. K ich správne mu popisu, princípom a fungovaniu bolo nutné prečítať a naštudovať značné množstvo hlavne odborných článkov. Spolu s popisom existujúcich riešení v danej oblasti práca v aktuálnom stave poskytuje dostatok informácií k oboznámeniu sa s popisovanou problematikou.

Súčasťou odovzdávanej časti sú taktiež aj popisy prvotných experimentov spolu s návrhom neurónovej siete a datasetom použitým počas v rámci experimentovania.

Literatúra

- [1] Kvasnička V. a kol. *Úvod do teórie neurónových sietí*. Iris, 1997.
- [2] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.
- [3] Ali Borji, Hamed R Tavakoli, Dicky N Sihite, and Laurent Itti. Analysis of scores, datasets, and models in visual saliency prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 921–928, 2013.
- [4] Renwu Gao, Faisal Shafait, Seiichi Uchida, and Yaokai Feng. A hierarchical visual saliency model for character detection in natural scenes. In *International Workshop on Camera-Based Document Analysis and Recognition*, pages 18–29. Springer, 2013.
- [5] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2009.
- [6] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Unsupervised learning. In *The elements of statistical learning*, pages 485–585. Springer, 2009.
- [7] Laurent Itti. Visual salience. *Scholarpedia*, 2(9):3327, 2007.
- [8] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.
- [9] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

- [10] Fei-Fei Li, Andrej Karpathy, and J Johnson. Cs231n: Convolutional neural networks for visual recognition, 2015.
- [11] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013.
- [12] Patrik Polatsek. Saliency maps. Prezentácia, 2015.
- [13] David MW Powers. Roc-concert: Roc-based measurement of consistency and certainty. In *Engineering and Technology (S-CET), 2012 Spring Congress on*, pages 1–4. IEEE, 2012.
- [14] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [15] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH*, pages 338–342, 2014.
- [16] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [17] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39:43–62, 1997.
- [18] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012.

A Technická dokumentácia

V nasledujúcej časti sa nachádza technická dokumentácia, spolu s doteraz použitými technológiami. Jednotlivo sú popísané najdôležitejšie použité knižnice a funkcionality rozdelená v procedúrach.

A.1 Hardvér

Trénovanie modelu neurónovej siete prebiehalo na fakultnom serveri s grafickou kartou Nvidia GeForce GTX 980 Ti 6gb a RAM pamäťou o veľkosti 24gb, avšak samotné riešenie nie je viazané na určitý typ hardvéru, stačí len mať dostatok pamäte na vytvorenie modelu a spracovanie datasetu.

A.2 Použité knižnice

A.2.1 TensorFlow

TensorFlow¹⁰ je open-source softvérová knižnica, ktorá pre numerické výpočty používa graf dátového toku, kde uzly grafu reprezentujú matematické operácie a hrany multidimenzionálne dátové polia, tzv. tenzory. Graf je možné skonštruovať použitím jazykov s podporou frontend-u (C++ a Python).

Flexibilná architektúra umožňuje vykonávať výpočty na CPU alebo GPU (nepomerne rýchlejšie) na serveroch, desktopových počítačoch či dokonca aj mobilných zariadeniach. Pôvodne bol TensorFlow vyvinutý výzkumníkmi a inžiniermi v Google-i pre strojové učenie a hlboké učenie, avšak jeho využitie je oveľa širšie. Momentálne používa TensorFlow veľké množstvo programov, napríklad Google vyhľadávač, prekladač alebo YouTube.

Centrálnou jednotkou v TensorFlow-e je tenzor. Ten pozostáva z hodnôt sfomovaných v pole akejkoľvek veľkosti či dimenzie. Jadro TensorFlow-u tvorí výpočtový graf, čo je vlastne séria TensorFlow operácií, ktoré sú do grafu pridané ako uzly.

Klientske programy interagujú s TensorFlow-om vytvorením tzv. *Session*. Jej primárne podporovanou operáciou je *Run*, ktorá v podstate funguje ako *init*, keďže

¹⁰<https://www.tensorflow.org/>

do jej zavolania sú hodnoty všetkých premenných TensorFlow-u neinicializované.

Pre potreby strojového učenia je dostupné značné množstvo funkcií od nastavenia optimizérov, cez funkcie počítajúce chybu či presnosť až po tie zabezpečujúce samotné učenie.

A.2.2 Matplotlib

Ďalšia open-source knižnica¹¹, pre zmenu určená pre 2D vykresľovanie nielen rôznych grafov. Poskytuje obrovské množstvo možností vizualizácií dát, ktoré ocení nielen hárka vedcov. Jeho základnou súčasťou je modul pyplot, ktorý prakticky implementuje vykresľovaciu funkcionálnu Matlabu.

¹¹<https://matplotlib.org/>