Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-0000-00000

Patrik Beka

Určovanie pútavých častí grafických rozhraní

Bakalárska práca

Vedúci práce: Ing. Márius Šajgalík

December 2016

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-0000-00000

Patrik Beka

Určovanie pútavých častí grafických rozhraní

Bakalárska práca

Študijný program: Informatika
Študijný odbor: 9.2.1 Informatika
Miesto vypracovania: Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU, Bratislava
Supervisor: Ing. Márius Šajgalík

December 2016

Anotácia

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
Študijný program: Informatika
Autor: Patrik Beka
Bakalárska práca: Určovanie pútavých častí grafických rozhraní
Vedúci práce: Ing. Márius Šajgalík
December 2016

Pútavý a graficky príťažlivý dizajn je mimoriadne dôležitou súčasťou každej webovej stránky, hoci sa to tak nemusí na prvý pohľad javiť. Niekto môže namietať, že oveľa dôležitejší je samozrejme jej obsah, avšak bez dobrého dizajnu býva často veľkou výzvou dostať sa k hľadanému obsahu. Z tohto pohľadu je dobrý dizajn dôležitý nielen pre používateľov stránok, ale rovnako aj majiteľov, pretože tí by mali byť schopní umiestňovať obsah na základe jeho dôležitosti. Nepodstatné reklamy tak, aby nepôsobili rušivo a relevantné informácie na miesta, kde ich používateľ ihneď zaregistruje.

K otestovaniu pútavosti dizajnu sa najlepšie hodí vedieť kam presne sa pozerá návštevník stránky, čo môže byť dosť zložité, keďže je k tomu väčšinou potrebná istá vzorka ľudí spolu s potrebným vybavením a spracovaním týchto dát. Nami zvolená metóda je určená k predikcii pohľadov na webstránku len z jej obrázka a to pomocou neurónových sietí, konkrétne za použitia konvolučných neurónových sietí. Tie sú schopné naučiť sa určité zákonitosti priamo z dát, následne ich aplikovať na doteraz nevidené nové dáta a s dostatočnou presnosťou predikovať pohľady na nových nevidených stránkach.

Annotation

Slovak University of Technology Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES
Degree Course: Informatika
Author: Patrik Beka
Bachelor thesis: Determination of the eye-catching parts in graphical interfaces
Supervisor: Ing. Márius Šajgalík
December 2016

Eye-catching and graphically attractive design is an extremely important part of every website, although it might not appear to you at first sight. Someone could argue that the content is of course much more important, but without a good design, required content is often getting through a big challenge. From this point of view, good design is important not only for the users of web pages, but also for the owners, because they would be able to place the contain according to its importance. The insubstantial ads should be placed on spot, which will not appeal bothering and the relevant informations should be on places, where the user notices it immediately.

To test the saliency of design, it is the best to know where exactly the visitor of page is looking mostly. This may be pretty difficult, because it requires certain sample of people, along with the proper equipment and processing of this data. Our selected method is determined to predict the views on website just from its image with a little help of the neural networks, specifically by using convolutional neural networks. These networks are able to learn certain regularities directly from the data, subsequently apply them on previously unseen data and with the sufficient accuracy predict the views on new unseen web pages.

POĎAKOVANIE

Ďakujem svojmu vedúcemu Ing. Máriusovi Šajgalíkovi za odborné rady a čas strávený pri konzultovaní tejto práce.

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

Obsah

Slovník pojmov a skratiek

MKL - multiple kernel learning. Je sada metód strojového učenia, ktoré používajú preddefinovanú sadu jadier k učeniu sa lineárnych a nelineárnych kombinácií jadier vrámci určitého algoritmu.[6]

SVM - support vector machine. Sú to modely učenia s príslušnými algoritmami učenia, ktoré analyzujú dáta použité pre klasifikáciu a regresnú analýzu. SVM model je reprezentácia príkladov bodov v priestore mapovaných tak, že príklady z rôznych kategórií sú rozdelené viditeľnou medzerou, tak širokou ako sa dá.[5]

Gradient - zmena veličiny v závislosti od inej premennej

RMSProp algoritmus - metóda, pri ktorej rýchlosť učenia závisí priamo od konkrétneho parametra.[18]

Zoznam obrázkov

1 Úvod

Návrh dobrého grafické rozhrania pre web stránky nie je vôbec jednoduchá vec. V dnešnej dobe, kedy sa informácie hľadajú už skoro len výlučne pomocou internetu nadobúda dobrý dizajn web stránok o to väčšiu dôležitosť. To, či ma ako návštevníka určitej stránky osloví jej obsah, nie je ani zďaleka tak podstatné ako to, či ma zaujme jej dizajn (výzor) a ako jednoducho sa na nej dostanem k tomu pre mňa podstatnému. Dizajn stránky je určite jedným z najdôležitejších predpokladov pre budúci návrat na stránku. Nikto nemá rád, keď jeho pozornosť od dôležitých a hľadaných informácií odtrhávajú nepodstatné reklamy, hoci si to často ani len neuvedomuje. V súčasnosti sa k určeniu kvalitných pútavých webových stránok nepoužíva také veľké množstvo nástrojov, ako by sa dalo predpokladať. Dva hlavné prístupy k určovaniu nielen pútavosti, ale aj dôležitosti rôznych blokov webov sú popísané v kapitole 2.2.

Tieto riešenia používajú aj neurónové siete, či už viac alebo menej. Tie sú nielen v súčasnosti, ale aj v posledných rokoch, na vzostupe. Budúcnosť počítačov leží v umelej inteligencii, medzi ktorú patria aj už spomínané neurónové siete. Hoci ich teoretický model a fungovanie boli popísané už pred desaťročiami, až s postupným pokrokom prišli technológie, vďaka ktorým bolo možné previesť teoretické poznatky na fungujúce prototypy, či už mnohých typov autopilotov alebo rozpoznávania tvárí. Popisu fungovania neurónových sietí ako aj ich rôznym typom sa venujem v časti 2.1.

Po rozanalyzovaní podstatného je v časti 3 popis návrhu riešenia zadaného problému, spolu s popisom získaného datasetu a návrhom neurónovej siete.

V závere je po krátkom zhrnutí práce predstavený v stručných odrážkach plán na skúškové obdobie a letný semester.

2 Analýza

V tejto časti sú postupne popísané neurónové siete, ich princíp a základné typy. Osobitnejšie sú popísané konvolučné neurónové siete pre prácu s obrázkami. Podkapitola 2.2 je venovaná existujúcim riešeniam problému určovania pútavých častí grafického rozhrania webstránky.

2.1 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, respektíve model neurónu[1]. Ten spracováva rôzne množstvo vstupov (N) a výstupov (M). V minulosti sa zvykol vyjadrovať podľa nasledovnej matematickej špecifikácie:

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

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-tym neurónom vrstvy k+1 a j-tym neurónom vrstvy k
f() - funkcia

V súčasnosti sa však používa radšej matematické vyjadrenie zobrazené nižšie. 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 pridelí 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[2] nasledovne:

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

Pre vyššie uvedené platí:

y_k - výstup k-teho neurónu

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

x_j - j-ty neurón

ϕ - funkcia

Neurónová sieť sa 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. Každá vrstva (s výnimkou výstupnej) musí ešte 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 softmax[1] (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 matimatické vyjadrenie je nižšie.

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

  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 zárove potlačenie hodnôt, ktoré sú výrazne menšie ako maximálna hodnota.

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

- **ReLU**

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

  $$f(x) = max(0, x) \tag{4}$$

  kde *x* je vstup do neurónu. Používa sa vďaka svojej jednoduchosti, keďže neobsahuje žiadne komplikované výpočty. Jej využitie je možné pozorovať napríklad pri hlbokých neurónových sieťach.

- **Softplus**

  Je v podstate aproximáciou k predošlej ReLU s matematickým vyjadrením:

  $$f(x) = ln(1 + e^x) \tag{5}$$

  Rovnako ako pri ReLU je oborom hodnôt interval (0, ∞). 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:

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

- **Tanh**

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

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

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[8]). 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 očakaváne výstupy. Učenie sa bez učeiteľa naproti tomu odvodzuje funkciu k popisu skrytej štruktúry z neoštítkovaných dát, teda bez štítkov, ktoré nám určujú očakaváne 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, taže 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 optimizačné algoritmy ako Gradient descent optimizer[13] alebo Adam optimizer[10], 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 gradientu 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 vykonania tejto aktualizácie. Týmito typmi sú:

- Dávkový gradient descent:

  Z angl. Batch gradient descent. Gradient sa počíta pre celý tréningový data-

set, 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 gradientu descent optimizéru, resp. jeho modifikácie RMSProp algoritmu. Rozdiel oproti Gradient descent optmizéru je ale v tom, že je schopný variabilne určovať rýchlosť učenia neurónovej siete.

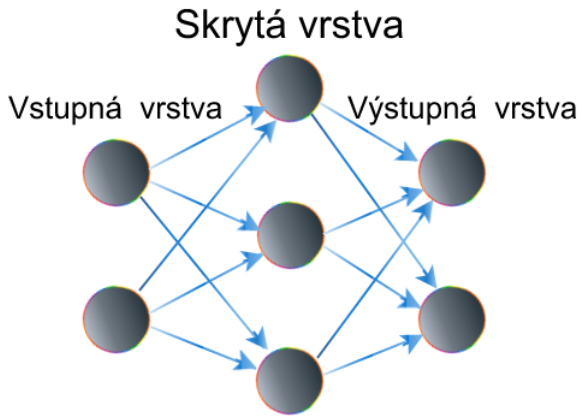
Po fáze učenia sa nasleduje validácia, po ktorej sa prejde k samotnému testovaniu neurónovej siete, kedy sa do nej posúvajú dáta tento krát bez toho, aby sieť mala prístup k štítkom. Na základe jej predikcie a štítkom k testovacím dátam sa určí jej presnosť. Na učenie, testovanie a validáciu by nemali byť použité tie isté dáta. Pomer dát k jednotlivým fázam by mal byť 80-10-10[12], 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 výpočovej sily, aby sa problém neriešil pridlhý čas.

2.1.1 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 zobraziť 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 1. 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]).



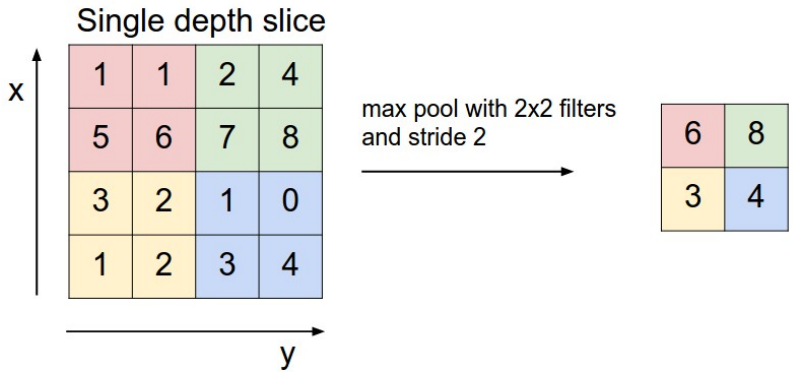
Obr. 1: *Príklad jednoduchej neurónovej siete*

Zložitejším typom sú rekurentné neurónové siete. Už z názvu vyplýva že umožňujú rekurenciu, takže 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[7], či generovaní popisu k obrázkom[9], 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.

2.1.2 Konvolučné neurónové siete

Základ tejto siete tvorí vstupná konvolučná vrstva[2] s konvolučným filtrom, ten býva väčšinou malý (3x3, 5x5). Vstup tejto vrstvy musí byť v tvare *m* x *m* x *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[11]), ktorá tieto filtre rozvzorkuje. To prebieha nezávisle na každom získanom filtry 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 prejdenie 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 obrázku nižšie je vidieť výsledok popisovaného postupu na jednoduchom príklade.

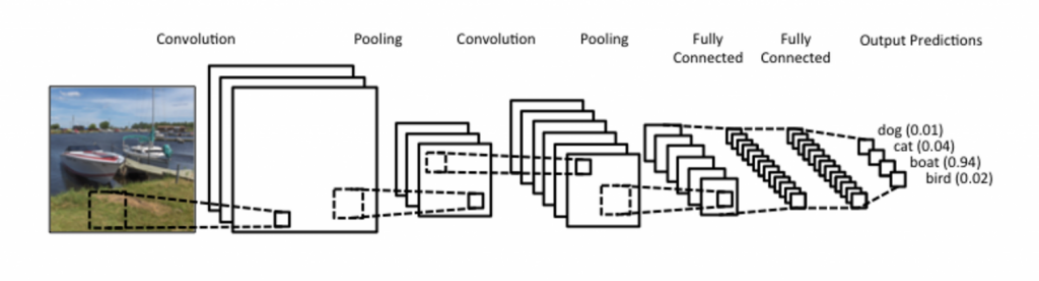


Obr. 2: *Príklad vrstvy združovania, pri ktorom sa rozvzorkuje výstup z konvolučnej vrstvy o veľkosti 4x4, filtrom 2x2, s krokom veľkosti 2 za použitia operácie MAX*

Takýchto konvolučných vrstiev s vrstvami združovania môže byť aj viac, ne-

[2] http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

musia 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.



Obr. 3: *Príklad konvolučnej neurónovej siete*

Využite tohto typu je v podstate všade, kde sa jedná o rozpoznávanie obrázkov. Či už ide o automatické vyznačenie tvárí pre označenie na facebooku, autonómne vozidlá, ktoré sa vedia riadiť sami (autopilot) alebo triedenie uhoriek na farmách v Japonsku[3]. Na tento konkrétny softvér bol použitý príklad kódu jednoduchej konvolučnej siete z tutoriálu[4] pre TensorFlow (knižnice pre prácu s neurónovými sieťami), s modifikáciou konvolučnej a združovacej vrstvy tak, aby bola sieť uspôsobená počtu tried uhoriek (10) a ich formátu obrázkov.

Z mnohých pokusov o autonómnu jazdu stoja za zmienku hlavne tie od Tesly a Google. Prototyp autonómneho systému vozidla od Google-u Dave-2[3] 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 YUV[5] úrovní a posunutý do neurónovej siete.

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

[4]https://www.tensorflow.org/versions/0.6.0/tutorials/mnist/pros/index.html

[5]farebný priestor používaný vo video aplikáciách

2.2 Existujúce riešenia

V nasledujúcej časti sú opísané niektoré z existujúcich riešení problému určenia pútavých častí grafického rozhrania webovej stránky. V podstate existujú 2 prístupy, z ktorých jeden vychádza z HTML kódu stránku a druhý len z jej obrázku, screenshot-u.

2.2.1 Riešenia na báze segmentácie stránok

Väčšina riešení podobného problému na začiatku vyhádza z HTML kódu stránky, ktorú na je ho základe rozdelí tzv. segmentovaním na hlavné časti (segmenty, bloky) stránky. Na toto rozdelenie sa najčastejšie používajú tieto algoritmy:

- Segmentácia založená na DOM (dokumentový objektový model, z angl. document object model):
  HTML dokument je reprezentovaný ako DOM strom. Tagy môžu reprezentovať jeden blok stránky, napr. P-paragraf, TABLE-tabulka, atď. I keď tento strom dokáže veľmi presne reprezentovať štruktúru HTML dokumentu (stránky), často nie je dostatočne presný pri rozdelení sémanticky (vizuálne) rôznych blokov.
- Segmentácia založená na lokácii (z angl. location-based segmentation):
  Stránka sa rozdelí na 5 hlavných častí: stred, vľavo, vpravo, hore, dolu. Problémom je však, že nie vždy sa toto rozdelenie hodí na každú stránku a v prípade, že je stránka príliš dlhá (scroll-ovateľná) sa časti, ktoré boli na začiatku napr. dolu, posunú smerom hore a rozdelenie stránky sa tým zmení.
- VIPS[14] - segmentácia stránky založená na pohľade (z angl. vision-based page segmentation):
  VIPS je v podstate kombinácia predchádzajúcich dvoch algoritmov. Delí stránku aj na základe farby, veľkosti blokov atď. V prvom rade vyberie vhodné uzly z DOM stromu a nájde medzi nimi separátory, ktoré naznačujú horizontálne a vertikálne línie webovej stránky. Na základe DOM stromu a separátorov sa vytvorí sémantický strom stránky, v ktorom je každý segment stránky reprezentovaný ako samostatný uzol v strome, koreňom stromu je samotná stránka. Spojitosť segmentácie stránky je kontrolovaná prededifinovaným stupňom koherencie (predefined degree of coherence (PDOC)) –

každému uzlu je daný určitý stupeň súvislosti, čo zabezpečuje, že VIPS dokáže efektívne držať obsah pokope, zatiaľ čo sémanticky odlišné bloky od seba.

Segmentácia potom končí v momente, kedy by pokračovanie v nej zničilo sémantickú integritu blokov. Bloky sú ďalej očíslované od 1 po 4 podľa dôležitosti (1 – nepodstatné informácie ako reklamy, 4 – najdôležitejšia časť stránky, hlavný obsah). Z toho sa potom zostrojí model dôležitosti blokov, čo je vlastne funkcia mapovania každého bloku a jeho dôležitosti zobrazená ako:

$$\{ \text{block features} \} \rightarrow \text{block importance}$$

Na odhad dôležitosti blokov sa potom nahliada ako na regresný problém a na jeho riešenie je použitá neurálna sieť, v ktorej sú bloky reprezentované ako dvojica (x, y), kde x je reprezentácia bloku a y je dôležitosť (berie sa ako reálne číslo). Typ siete je RBF (radial basis function) a vyhľadávacou technikou je štandardný gradient descent. Sieť zahŕňa 3 vrstvy, každú s inou funkciou. Prvú (vstupnú) vrstvu tvoria zdrojové uzly, druhá je jediná skrytá vrstva a zabezpečuje nonlineárnu transformáciu zo vstupného priestoru do skrytého. Obsahuje RBF neuróny, ktoré kalkulujú vstup skrytej vrstvy kombináciou vážených vstupov a odhadov. Tretia (výstupná) vrstva dodáva dôležitosť blokov do reprezentácie blokov aplikovanej v prvej (vstupnej) vrstve.

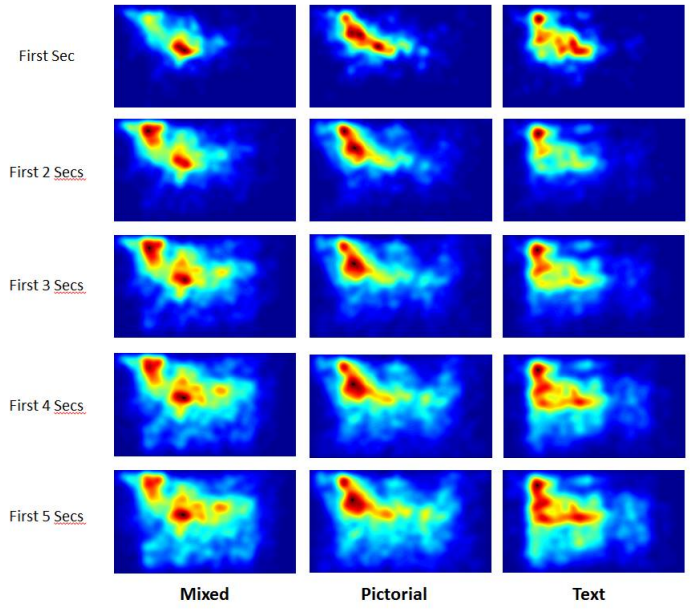
Výhodou tohto riešenia je rozdelenie stránky podľa logických blokov, nevýhodu vidím v začiatočnom ohodnotení blokov ľuďmi podľa dôležitosti, pretože mi to príde príliš individuálne. Tiež rozdelenie stránky do stromu s jednotlivými blokmi a časťami je v závislosti od stránky dosť pamäťovo náročná operácia.

2.2.2 Riešenia vychádzajúce z obrázku stránky

Tento typ riešenia používa k určeniu dôležitých (pútavých) častí stránky jej screenshot v kombinácii s neurónovou sieťou. Autori na začiatku rozdelili ich dataset do niekoľkých kategórií podľa obsahu stránky, t. j. ilustrované (prevládajú obrázky), textové (prevažne blogy, články, atď.) a mix predošlých dvoch kategórií. V každej z nich bolo približne 50 obrázkov s rozlíšením 1360x768. Na obrázky web

stránok potom pozeralo 11 subjektov a ich sekvencie pohľadov boli zaznamenané pomocou MATLABU s Psychtoolbox[4] a systémom Eyelink 1000.

Na predikciu pohľadov[16] na web stránky bola použitá metóda MKL (multiple kernel learning - kombinuje viacero jadier SVM[6] miesto jedného). MKL bolo trénované ako binárny regresný problém, z datasetu bolo použitých 119 obrázkov na trénovanie, zvyšných 30 bolo použitých na finálne testovanie. Zo získaných výsledkov bolo zistené, že človeka mimo iné upúta pri pohľade na web stránku v prvom rade ľudská tvár, oči a celkovo horná časť tela, či neprimerane veľké logo. Taktiež sa dá predpovedať, že človek sa začne pozerať v prvom rade najprv do strednej oblasti a oblasti viac vľavo hore od stredu. Na obrázku nižšie je možné vidieť vizualizáciu pohľadu na rôzne typy stránok v prvých 5 sekundách formou teplotných máp.



Obr. 4: *Vizualizácia pohľadov v prvých 5 sekundách. Vľavo mix stránok, v strede prevažne ilustrované stránky a vpravo textové.*

Za výhodu tohto riešenia oproti predošlému pokladám hlavne vygenerovanie heat mapy, ktorá vypovie o dôležitých častiach stránky ďaleko viac ako očíslované

[6]Support vector machine - modely učenia s príslušnými algoritmami učenia, ktoré analyzujú dáta použité pre klasifikáciu a regresnú analýzu[5]

HTML bloky. Taktiež zistenia, že človek si ako prvé všíma napríklad ľudské tváre, sú pre dizajnéra web stránky určite velké plus oproti dôležitosti informácií na základe ich umiestnenia.

3 Návrh

V nasledujúcich riadkoch tejto kapitoly je rozobratý postupný návrh neurónovej siete určenej k riešieniu zadaného problému, spolu s potrebným spracovaním poskytnutého datasetu.

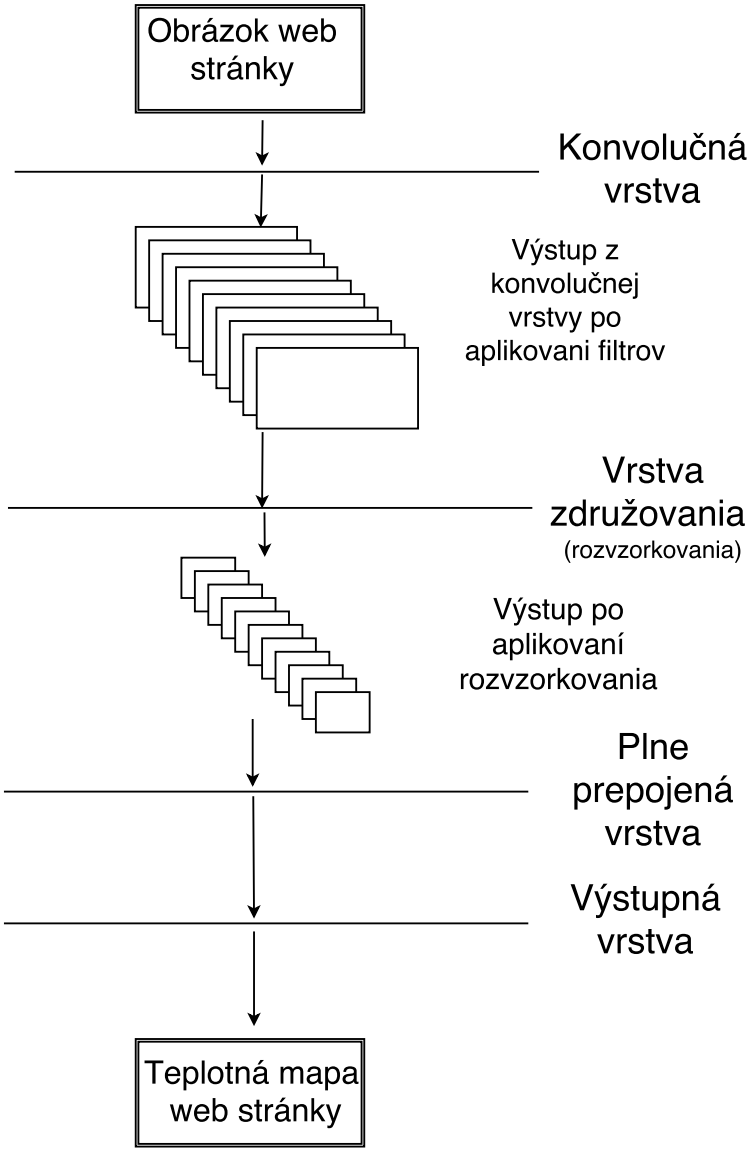
3.1 Dataset

Dataset sa skladá z obrázkov niekoľkých webstránok a pohľadov skupiny ľudí na ne. Nakoľko celý dataset obsahuje okrem údajov pre nás potrebných aj údaje z množstva iných, pre tento problém nerelevantných úloh, je nutné ho najprv prerieďiť a odfiltrovať všetko nepotrebné. Taktiež aj obrázky webstránok je nutné jemne upraviť, zmenšiť veľkosť nakoľko majú príliš veľké rozlíšenie pre neskoršie spracovanie neurónovou sieťou. Dataset je ďalej nutné rozdeliť v pomere 80-10-10 pre jednotlivé fázy spracovania v neurónovej sieti (trénovanie, validácia a testovanie).

Vstupné dáta do neurónovej siete tvoria obrázky, štítkami sú v tomto prípade finálne teplotné mapy vytvorené pomocou sekvencie pohľadov a Gaussovho normálneho rozdelenia.

3.2 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. Sieť by mala pozostávať z konvolučnej a združovacej vrstvy (vrstiev) pre spracovanie obrázku, nasledovaných plne prepojenou vrstvou (vrstvami) s aktivačnou funkciou ReLU, pretože je rýchla bez komplikovaných výpočtov. Za nimi nasleduje už len výstupná vrstva, samozrejme bez aktivačnej funkcie, ktorá nie je potrebná. Celá architektúra je načrtnutá na schéme nižšie.



Obr. 5: *Grafická schéma neurónovej siete*

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 vrste združovania s použitím operácie MAX, okna (filtra) s veľkosťou 2x2 a posunom s veľkosťou 2. Z plne prepojenej vrstvy do výstupnej vrstvy sa nakoniec vykoná predikcia výslednej teplotnej mapy.

K trénovaniu sa použije Adam optimizér kvôli variabilnej rýchlosti učenia, spolu s náhodným generátorom dát, ktorý bude z datasetu určenému pre fázu trénovania náhodne vyberať dávky dát (batch-e). Tým sa zabezpečí simulácia náhodnosti vstupných dát a teda sa sieť bude schopná lepšie učiť.

4 Záver a plán ďalšej práce

Vypracovaná prvá časť bakalárskeho projektu obsahuje analýzu nosnej časti pripravovaného riešenia, neurónových sietí. K ich správnemu 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 problematikou určovania pútavých častí grafický rozhraní.

Súčasťou odovzdávanej časti je taktiež aj prvotný návrh neurónovej siete k riešeniu tejto problematiky, spolu s popisom datasetu a jeho spracovania.

Plán a rozdelenie zvyšnej práce je zhruba nasledovný:
Skúškové obdobie

- refaktorizácia a „upratanie”doterajšej práce (hlavne kódu)
- implementácia návrhu
- mať funkčný prototyp schopný predikcie
- otestovať rôzne možnosti učenia (meniť veľkosti batch-ov, rýchlosť,...)
- v januári prekonzulovať dosiahnuté výsledky spolu s ďalším postupom

Letný semester

- refaktorizácia a optimalizácia riešenia
- postupné dopisovanie práce a dokumentácie

Vyššie uvedený plán je iba orientačný, určite sa bude meniť, či už v závislosti od vzniknutých problémov alebo množstva času nutného k splneniu jednotlivých bodov plánu a vypracovaniu častí práce.

Literatúra

[1] Kvasnička V. a kol. *Úvod do teórie neurónových sietí*. Iris, 1997.

[2] Martin Anthony. *Discrete mathematics of neural networks: selected topics*, volume 8. Siam, 2001.

[3] 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.

[4] David H Brainard. The psychophysics toolbox. *Spatial vision*, 10:433–436, 1997.

[5] Cortes C. and Vapnik V. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[6] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(Jul):2211–2268, 2011.

[7] 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.

[8] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Unsupervised learning. In *The elements of statistical learning*, pages 485–585. Springer, 2009.

[9] 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.

[10] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] Fei-Fei Li, Andrej Karpathy, and J Johnson. Cs231n: Convolutional neural networks for visual recognition, 2015.

[12] 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.

[13] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[14] Ji-Rong Wen Ruihua Song, Haifeng Liu and Wei-Ying Ma. Learning block importance models for web pages. In *Proceedings of the 13th international conference on World Wide Web (WWW '04)*, pages 203–211, New York, USA, 2004.

[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] Chengyao Shen and Qi Zhao. *Webpage Saliency*, pages 33–46. Springer International Publishing, Cham, 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.

A.1 TensorFlow

TensorFlow 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 frontendu (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.