Transfer learning s CNN: VGG16 a ResNet50 pro analýzu kapslové endoskopie

Lukáš Florner
flornerlukas@seznam.cz
Semestrální projekt - FIM@UHK

***Abstrakt*—Tato práce se zabývá konvolučními sítěmi (Convolutional Neural Networks (CNN)) s aplikací přeneseného učení, resp. hlubokého přeneseného učení (deep transfer learning (DTL)) a jejich schopnostmi práce s omezenými datovými sadami. Práce obsahuje teoretický úvod do problematiky (DTL) a dále se zaměřuje na dva modely - ResNet50 a VGG16. Tyto modely spouštím s různými konfiguracemi nad úlohou klasifikace obrázků kapslové endoskopie (conventional white light endoscopy (CWE)) a porovnávám nejprve tyto konfigurace v rámci modelu (pro zjištění změn v přesnosti klasifikace) a následně i modely mezi sebou. Obrázky CWE obsahují 3 různé kategorie (normal, bleeding, polypoids) a zpracování je prováděno pomocí frameworku Keras a v prostředí Google colab. Výsledky je možné použít pro výběr konkrétního modelu a konfigurace pro obdobné klasifikační úlohy.**

***Klíčová slova*—Přenesené učení; Konvoluční neuronová síť; ResNet; VGG; Porovnání**

I. INTRODUCTION/ÚVOD

A. OBSAH

Aplikace konvolučních sítí rozšiřuje možnosti a použitelnost strojového učení a umělých neurálních sítí (Artifical Neural Networks (ANN)) pro klasifikační úlohy obrazových dat. Jedná se o úlohy kdy je třeba rozlišit objekty na obrázcích a zařadit je do předem definovaných kategorií. Počty cílových kategorií mohou být různé - od 1 (resp. 2) pro binární klasifikaci, až po mnohem vyšší počty tříd (např. klasifikační úloha MNIST pracuje s ručně nakreslenými obrázky čísel, které se snaží zařadit do deseti kategorií 0-9). Oproti klasickým ANN nezpracovává CNN celá vstupní data najednou, ale pracuje s jádrem (tzv. kernel), což je matice nebo maska zvolené velikosti, jejíž jednotlivé členy představují konkrétní pixel z obrázku a barevného kanálu převedený na číslo (0-255). Nejčastěji jde o kernel 3x3, příp. 5x5, což oproti ANN představuje vyšší efektivitu (u ANN by vstup pro obrázek např. 100x100 v jednom kanálu znamenal 10000 parametrů a tomu by odpovídala i nutná hloubka sítě). Výsledné zpracování si lze představit tak, že se obrázek pokryje maticemi (sousední matice se mohou částečně překrývat - dle parametru stride) a v každé matici proběhne skalární součin s vahami sítě. Výsledkem je lehce pozměněný obrázek, ve kterém nám při hlubších a hlubších operacích začnou objevovat tzv. features, které lze chápat jako charakteristické znaky objektu (hrany, jejich vzájemná poloha, skupinový výskyt atd.). Konvoluční vrstvy lze stohovat, stejně jako vrstvy ANN. Samotné konvoluční vrstvy by nám vracely obrázek stejné velikosti, což by nebylo příliš použitelné. Z toho důvodu se mezi jednotlivé konvoluční vrstvy nebo jejich stohy vkládají vrstvy poolingu provádějící komprimaci dat (resp. snižují dimenzi vstupních dat) před vstupem do další konvoluční vrstvy, nebo stohu vrstev. V CNN se nejčastěji používá max-pooling a velikost kernelu 2x2, bez překrývání matic. Max-pooling vrstva tedy pokryje výstup z předchozí konvoluční vrstvy maticemi 2x2 a v každé matici vybere maximální hodnotu. Mimo max-pooling operací mohou CNN obsahovat i obecné pooling vrstvy, které provádí jiné komprimační operace (např. normalizaci nebo average-pooling). Díky pooling vrstvám je pak dimenze dat snáze zpracovatelná závěrečnou částí celé sítě - plně propojeným klasifikátorem. Ten obsahuje dense vrstvy, jejichž prostřednictvím se síť učí správně klasifikovat vstupní data.

Jako všechny neurální sítě i u CNN závisí kvalita výstupu na kvalitě a množství poskytnutých dat - čím větší je počet různých příkladů, tím lépe se síť naučí rozeznávat vzory a tím přesnější výsledky poskytuje. Důraz je přitom kladen nejen na slovo *počet*, ale i na slovo *různých*, neboť i v případě tisíců ale velmi podobných obrázků by se síť naučila reprezentaci vzorů, které by nemusely dopovídat očekávání. Např. po učení se na rozsáhlé obrazové databázi obličejů bez brýlí by taková síť nemusela správně klasifikovat obličej s brýlemi - jednoduše proto, že takovému obrázku nesouhlasí klíčový rys proti trénovacímu souboru (což může být důvodem, proč malé děti při prvním kontaktu s obrýleným člověkem většinou pláčou - že by přirozená dětská reakce na overfitting?). Ve zpracovávané úloze je právě množství poskytovaných dat tím nejzásadnějším problémem. Vstupních příkladů, na kterých je možné data natrénovat a validovat je málo - 308 příkladů pro kategorii bleeding, 1179 pro kategorii normal a 44 v kategorii polypoids. U přidruženého data setu (pro porovnání různých klasifikátorů), který obsahuje hned 16 kategorií, je situace ještě horší, když některé kategorie mají jen jednotky příkladů a jiné stovky. V ideálním případě by měly v každé kategorii být alespoň stovky různých obrázků a raději násobně více. I s touto omezenou sadou je ale možné pracovat a pomocnou ruku nám v tom může podat technika přeseného učení a augmentace DTL.

DTL využívá předtrénované neurální sítě, která se již naučila rozeznávat vzory na rozsáhlých obrazových databázích. V našem případě jde o modely VGG16 a ResNet50 trénované na databázi ImageNet, což je obrazová databáze obsahující v průměru 1000 obrázků v každé kategorii (tzv. synsetů), kterých je zhruba 100000. Jak je patrné, jedná se o úctyhodnou databázi a modely, které uspějí v klasifikačních úlohách by měly mít váhy nastavené ke snadnému rozeznání

podobných obrazových vzorů, i když je trénovací množina hodně omezená. Což v našem případě určitě platí. Problém ale je, že ImageNet neobsahuje obrazové předlohy z oblasti medicíny a už vůbec neobsahuje obrazová data CWE. Je tedy otázkou, do jaké míry je technika DTL schopná správně klasifikovat nikdy neviděná data, která navíc nejsou podobná žádné natrénované klasifikaci.

A závěrem ještě zmínka o augmentaci. Augmentace je technika, která provádí náhodné transformace podkladového obrázku (posunutí, natočení, přiblížení aj.), címž vytvoří nové zobrazení. V základu takový transformovaný obrázek neobsahuje nová data a nepřináší tedy nové rysy pro učení, ale obsahuje stávající data v nové souvislosti, což může při trénování pomoci lépe uchopit existující rysy.

B. CITACE

O'Shea, Keiron, and Ryan Nash. "An Introduction to Convolutional Neural Networks." ArXiv:1511.08458 [Cs], Dec. 2015. arXiv.org, http://arxiv.org/abs/1511.08458.

Martin Pilát. https://martinpilat.com/cs/prirodou-inspirovane-algoritmy/neuronove-site-konvolucni-site-zpracovani-obrazu. Accessed 7 Jan. 2021.

Weiss, Karl, et al. "A Survey of Transfer Learning." Journal of Big Data, vol. 3, no. 1, May 2016, p. 9. Springer Link, doi:10.1186/s40537-016-0043-6. ¨ ImageNet. http://www.image-net.org/about-overview. Accessed 9 Jan. 2021.

Building Powerful Image Classification Models Using Very Little Data. https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html. Accessed 9 Jan. 2021.

II. PROBLEM DEFINITION/ DEFINICE PROBLÉMU

A. OBSAH

Zadáním je kategorizace obrázků pořízených pomocí CWE. Na těchto obrázích jsou záběry střev, přičemž v základu (původní soubor pro testování a validaci) je úkolem rozdělit je do 3 kategorií:

- Normal: kategorie obsahující obrázky zdravých střev, bez poškození a podezřelých útvarů
- Bleeding: kategorie obsahující obrázky střev s patrným krvácením různého rozsahu
- Polypoid: kategorie obsahující podezřelé útvary - polypy (train = Bleeding: 184, Normal: 707, Polypoids: 26; validation = Bleeding: 62, Normal: 236, Polypoids: 9)

Tuto sadu budu označovat jako CWE3. Jak již bylo zmíněno výše, rozsahy jednotlivých kategorií jsou nevyvážené. Zdaleka nejvíc je zastoupena kategorie Normal (1179 obrázků), o něco méně kategorie Bleeding (308 obrázků) a početně nejchudší je kategorie Polypoids (44 obrázků). Kategorizaci budeme provádět ve dvou různých modelech - VGG16 a ResNet50 a zajímá nás, jaký bude rozdíl ve výsledcích zejména z hlediska přesnosti a okrajově i z hlediska rychlosti.

Mimo výše zmíněných dat CWE jsem si připravil ještě další datové sady, nad kterými jsem následně spouštěl jednotlivé algoritmy.

- CWE s vyváženými počty dat v jednotlivých kategoriích (CWE3-RED) - toho jsem docílil smazáním obrázků v početně bohatších kategoriích tak, aby výsledný poměr byl 35:35:35. Zde bylo myšlenkou pokusit se zjistit, zda s minimální ale vyváženou datovou sadou bude síť pracovat lépe.
- CWE s vyváženými počty dat v jednotlivých kategoriích (CWE3AG), ale namísto, abych data smazal, přidal jsem je do trénovací sady pomocí augmentace, tedy náhodné transformace. Testovací dataset tak obsahuje 3 kategorie se 707 obrázky v každé z nich, přičemž kategorie Normal obsahuje jen původní netransformovanou sadu (707 obrázků) a v kategoriích Bleeding a Polypoids je rozdíl mezi počtem obrázků původní sady a 707 obrázky požadovanými pro vyrovnání datasetu vytvořen náhodnými transformacemi původního datasetu.
- obrázky ručně psaných číslic 0-2, tedy jakousi obdobu úlohy MNIST, resp. CIFAR. V této datové sadě je 100 příkladů pro každou ze 3 kategorií a to jak u datasetu pro trénování, tak i pro validaci. Motivací pro sestavení této sady je vyzkoušet si hned na počátku, zda síť vůbec funguje, nebo jsou výsledky jen náhodné. Tuto sadu dále označuji jako CIFAR3. [1].

Jako řešení zadání kategorizace CWE se nám nabízí hnde několik možností:

1) CNN učená od začátku se základní sadou dat (tj. s CWE3, CWE2 apod.)
2) CNN učená od začátku s vybalancovaným datasetem
3) DTL s feature extraction a základní sadou dat
4) DTL s feature extraction s vybalancovaným datasetem
5) DTL s fine-tunning a základní sadou dat
6) DTL s fine-tunning s augmentací

Jak je patrno, jedná se o poměrně rozsáhlý projekt, který předpokládá sestavení 3 různých sítí (základní CNN a DTL pro feature extraction a fine-tunning), sítě DTL navíc ve dvou provedeních (pro VGG16 a ResNet50). Tvorba a běh tolika řešení by přitom byl jak časově, tak znalostně náročný (přeci jen jsem se s CNN dosud nesetkal a podstatnou novinkou je pro mě i Python). Proto budu u DTL pracovat jen s modelem VGG16 a teprve na vítěznou síť aplikuji ResNet50 pro vzájemné porovnání.

B. TUDY NE, PŘÁTELÉ...až na výjimky

Co nám nefunguje (a v některých případech to bylo intuitivně jasné už na začátku)...

1) CNN učená od začátku se základní sadou dat: Jak již bylo řešeno, byť je CNN mocným nástrojem, ke kvalitnímu naučení se vyžaduje velké datové sady. Pro tento případ jsem si sestavil CNN sestávající z kombinací 3 dvojic vrstev Conv2D a MaxPool2D následovaných vrstvou Flatten a v klasifikátoru Dense doplněných i vrstvou Dropout proti přeučení (několikrát

[1]Při studiu CNN jsem procházel několik různých zdrojů, včetně tutorialů třeba na medium.com, a nejednou jsem narazil na síť, která u jednoduchých klasifikací ("dogs vs. cats", CIFAR) poskytovala výslednou přesnost lehce nad 50%, což se mi nezdálo jako příliš slavný výsledek. Domnívám se, že s jednoznačnými vstupy, jako je CIFAR, by síť měla mít přesnost nad 90%, aby se dala označit za funkční

vyzkoušeno bez Dropout, s ní a s různým výpadkem a nakonec se mi jako nejvhodnější jevil Dropout s 0.5). Výsledek na CIFAR3 dosahuje 90% přesnosti. Výsledked na originálním sadě CWE3 je 76-78%. Mohlo by se zdát, že daná přesnost je poměrně vysoká s ohledem na nízké množství dat pro trénink. Bohužel je to ale zásluhou struktury základního datasetu CWE3, nikoliv učebního potenciálu konvoluční sítě. Při bližším pohledu totiž zjistíme, že kategorii Normal kategorizuje velmi dobře, ale ve všech ostatních kategoriích se mýlí a jen výjimečně správně klasifikuje alespoň Bleeding. Výsledných 78.2% (v nejlepší iteraci) je tedy dáno poměrem příkladů v Normal proti příkladům ve dvou zbývajících kategoriích[2]. Z mého pohledu je zajímavé uvědomit si, že výsledky nezávisí jen na správném sestavení sítě, ale stejnou měrou i na tom, jaká data máme k dispozici. A navíc že i solidně vypadající výsledek může tzv. klamat tělem a je třeba se mu "podívat pod pokličku".

2) CNN učená od začátku s augmentací dat pro vybalancování datasetu: Výsledky sítě s nevybalancovanými daty jsou, řekněme, mizerné. Proto mě napadlo, že by bylo zajímavé zjistit, jak by dopadla ta samá síť nakrmená početně vyrovnanými kategoriemi. Úplně první a nejjednodušší možností je smazání nadbytečných obrázků z početně bohatších kategorií. Tedy vytvoření redukovaného, ale vybalancovaného datasetu CWE3-RED. Výsledek? Nic moc. Přesnost validace nepřesahuje 40%. Nicméně při bližším pohledu už dochází ke správnému zařazení i u kategorií Bleeding a Polypoids. Jen celkově stále s velmi nízkou úspěšností. Je ale patrné, že pokud bychom měli k dispozici mnohem větší datovou sadu (tisíců obrázků) s vyrovnanými zástupci v kategoriích, mohli bychom se i s obyčejnou CNN dostat k zajímavým výsledkům.

Zajímavý jev nastal s daty CWE3AG, kde se sice přesnost sítě při validaci blížila podobným hodnotám jako u CWE3, ale při podrobnějším pohledu došlo ke správnému zařazení kategorií Bleeding a Polypoids ve více případech (a na druhé straně i k vyššímu počtu nesprávných zařazení kategorie Normal). Výsledná nejvyšší dosažená přesnost validace je 78.5% (při přesnosti trénování mezi 77.5 a 88.5%). Jak je vidět, prostá augmentace vedoucí k vyrovnání počtu příkladů v kategoriích umí zlepšit výkon sítě k poměrně zajímavým číslům a opět je to potvrzení předpokladu, že s rozsáhlejší vybalancovanou databází bychom mohli dosáhnout mnohem lepších výsledků[3].

3) DTL s feature extraction a základní sadou dat: Pro mě velmi překvapivé je i zjištění přesnosti u DTL s feature extraction. Feature extraction je, zkráceně řečeno, extrakce vah z předučeného modelu (např. z VGG16), což znamená, že se síť neučí od počátku s náhodným nastavením vah, ale využívá váhy, ve kterých jsou již propsány (naučeny) vzory z ImageNet. Výhodou je, že je běh sítě poměrně rychlý (základní model se prochází jen jednou z důvodu extrakce vzorů). Jak již ale bylo řečeno, ImageNet neobsahuje medicínská data a to je nejspíš příčinou toho, že DTL s feature extraction dosahuje jen o málo lepších výsledků, než CNN trénovaná od počátku s datasetem CWE3 (nikoliv až tolik z hlediska absolutní přesnosti, ale kvůli faktu, že zařadí správně až 16% příkladů kategorie Bleeding). V nejlepší epoše dosahuje tato síť validační přesnosti 76.5%. DTL s CWE3 většinou správně zařazuje kategorii Normal, rozpozná i část Bleeding, ale vůbec nedokáže správně zařadit kategorii Polypoids, která obsahuje příliš málo trénovacích i validačních dat. V rámci testu ale lze konstatovat, že obecně síť pracuje správně, když s CIFAR3 dosahuje přesnosti přes 99%.

4) DTL s feature extraction a vybalancovaným datasetem: Nepříliš přesvědčivé výsledky podává DTL i v případě redukovaného datasetu. V případě CWE3-RED je výsledná přesnost 44.5%. O něco lepší výsledek, než u CNN, ale stále žádná sláva. Při běhu DTL sítě s augmenotvaným datasetem CWE3AG je přesnost mnohem vyšší a to až 77%, navíc se síti běžící nad CWE3AG povede při validaci zařadit i kategorii Polypoids, i když jen v jednom případě. Tzn. s velký počtem augmenovaných dat docházíme k mírně lepším výsledkům, než při použití obyčejné CNN, nebo původní sady CWE3. Augmentace (v případě CWE3AG vlastně i jako samostatný preprocessing) tedy skutečně přináší zlepšení výkonnosti sítě.

C. ZHODNOCENÍ

V této sekci jsme si ukázali řešení, která nevedou k požadované přesnosti (z mého pohledu alespoň k 85% správně zařazených příkladů). Zajímavé byly ale zejména výsledky nad augmentovanou datovou sadou CWE3AG, která poskyuje solidní přesnost

III. NEW SOLUTION / NOVÉ ŘEŠENÍ

A. TRANSFER LEARNING s FINE TUNING na VGG16 a RESNET50

Nejprvé pár informací k VGG16 a ResNet50 a jejich rozdílům.

1) VGG16: VGG16 je konvoluční síť byla představena roku 2015, prošla velmi intenzivním učením na datových sadách ImageNet a vykázala (v té době mimořádně) vysokou přesnost a použitelnost i na malých datových sadách. VGG16 se skládá ze 13 konvolučních vrstev Conv2D, 5 vrstev MaxPooling2D umístěnými mezi stohy konvolučních vrstev a vlastního 3-vrstvého hustě propojeného klasifikátoru (sestavení modelu si lze prohlédnout pomocí funkce *summary*). Všechny konvoluční vrstvy Conv2D používají jádro velikosti 3x3 a vrstvy MaxPooling2D jádra 2x2. Všechny skryté vrstvy pracují s aktivační funkcí ReLu. Výstupní vrstva používá softmax.

2) ResNet50: ResNet50 se dostala do povědomí roku 2015, kdy tato síť vyhrála soutěž v klasifikaci ImageNet. ResNet50 je zjednodušeným modelem původní ResNet152 a obsahuje tedy již "jen"50 vrstev, ale stále jde o velmi hlubokou síť. Skládá se z konvolučních vrstev Conv2D s velikostmi jádra 1x1, 3x3 a 7x7 (různé vrstvy mají různé veikosti), jedné MaxPooling2D vrstvy s jádrem 3x3 a AvgPooling2D vrstvy

[2]S jistou jízlivotí lze poznamenat, že pokud bychom chtěli s touto sítí dosáhnout 90% přesnosti, stačilo by odmazat několik příkladů z kategorie Bleeding a Polypoids...ale za to by nám pacienti ani lékaři moc nepoděkovali

[3]Při augmentaci jsem použil jen prostorové transfomace jako natočení, zrcadlení apod. Bylo by možná zajímavé vyzkoušet třeba i změny gamma nebo barevného vyvážení, které by mohly přinést ostřejší přechody mezi oblastmi a tím možná i ještě výkonnější učení. A nebo augmentovat nejen kategorii Bleeding a Polypoids, ale i Normal pro získání násobně většího množství příkladů v každé třídě

a jádrem 7x7. Na konci (nejhlouběji) je pak umístěn vlastní hustě propojený klasifikátor. Vytváření takto hlubokých sítí je problematické kvůli jevu zvanému ztráta gradientu - čím více je vrstev, tím vyšší je pravděpodobnost, že při propagaci změny (gradientu) do hlubších a hlubších vrstev dojde kvůli neustálým přepočtům a snižování hodnoty k výpočtu tak malého gradientu, že ho síť už nebude schopna využít (delta se dostane pod sítí rozlišitelnou hodnotu). Síť pak nepracuje správně, protože každé další vrstvě se analyzovaná data jeví jako "homogenní", bez rozpoznatelných rysů. ResNet50 ale přichází s řešením problému v podobě přeskakování spojení, kdy je uložena hodnota před vstupem do konvoluční vrstvy (resp. bloku vrstev) a poté je znovu přičtěna k hodnotě z vrsvy vystupující těsně před aplikací aktivační funkce ReLu. Případně je uložená hodnota ještě přepočítána průchodem přes vlastní koncoluční a normalizační vrstvu (bez zařazené aktivační funkce), abychom dosáhli stejné dimenze u zpracované i uložené informace (protože jde o součet matic, musí být u obou zajištěna stejná dimenzionalita). Díky tomuto přeskakování se gradient neztrácí, vždy je vlastně přičtěn k výstupu z bloku, který nese informaci o právě zpracovaných features (na konci jsou ale oba sečtené výsledky hodnoceny pomocí aktivační funkce).

Fine tuning je vedle feature extraction druhou hojně používanou metodou přeneseného učení. Její výhodou by měla být vyšší přesnost učení a nevýhodou vyšší nároky na výkon (a čas) a složitější implementace. Princip fine tuningu spočívá v tom, že se učení spustí ve dvou cyklech. Nejprve běží přímo na modelu předlohy (VGG16, nebo ResNet50), jehož vrstvy jsou zamčeny, aby nedošlo k náhodné inicializaci vah na počátku a tím ztrátě předučených vzorů. Tato fáze je velmi podobná feature extraction a stejně jako u ní je model inicializován bez vlastního klasifikátoru. Ten je nahrazen naším vlastním klasifikátorem. Po dokončení prvního cyklu učení jsou některé nebo všechny vrstvy modelu odemčeny, aby síť mohla "doladit"detaily na naučených reprezentacích a výstup prvního cyklu je jí znovu protažen. Efektem by mělo být zvýšení přesnosti nad hodnoty dosažitelné ve feature extraction.

IV. IMPLEMENTATION / IMPLEMENTACE ŘEŠENÍ A TESTOVÁNÍ

A. OBSAH

Oba modely (ResNet50 i VGG16) mají totožné nastavení. Liší se jen v počtu odemykaných vrstev v druhém běhu. Pro oba modely jsou tak nastaveny následující parametry (vycházejí z vlastního testování s několika různými hodnotami):

- optimalizátor Adam, learning rate=0.0001 (testován i SDG, ale Adam poskytoval lepší výsledky, learning rate jsem testoval s hodnotami 0.1-0.000001, zvolená hodnota byla opět výsledkově nejzajímavější)
- počet epoch = 20 (otestován i vyšší počet epoch, ale ten nevedl k vyšší přesnosti, navíc implementován EarlyStopping a běh sítě díky němu v mnoha případech končí dříve než po 20 epochách)
- callback EarlyStopping - ukončení trénování, pokud se vybraný parametr (validační ztráta) nezlepší za 6 po sobě jdoucích epoch (parametr patience = 6) - mimo zkrácení doby trénování ji používáme jako prostředek proti overfittingu
- v druhém cyklu se odemykají 4 spodní vrstvy u VGG16 a 12 spodních vrstev u ResNet50 (zhruba stejný poměr odemčených vrstev k jejich celkovému počtu v modelu)

Pro zjištění funkčnosti sítě opět využiji CIFAR3. Přesnost validace u této datové sady dosahuje u VGG16 v prvním cyklu 98-99% (podobně jako u feature extraction) a ve druhém cyklu, s odemčenými vrstvami, se přesnost ještě zvýší nad 99.6%. U modelu ResNet50 je přesnot nižší zhruba o necelé 1%, což je stále validní výsledek. Obě sítě tedy můžu považovat za funkční a přistoupíme k testování s daty CWE.

1) VGG16 s CWE3: VGG16 při použití fine tuningu vykazuje vyšší přesnost než v případě feature extraction (možná to měl být předpoklad, ale oblast CNN je plná překvapení...minimálně pro mne). V nejlepší iteraci dosáhla přenost validace 87.5%, což je, s ohledem na kvalitu datasetu CWE3, velmi solidní výsledek. Navíc se modelu daří přiřadit správné kategorie pro Bleeding a Polypoids, byť zejména u Polypoids jde maximáně o 1 zásah z 9. Fine tuning tak skutečně přispěl ke zlepšení výkonu. Pro zajímavost - ve fázi prvního cyklu dosahovala přesnost v maximu okolo 77%, což je skutečně výsledek odpovídající feature extraction nad CWE3.

2) VGG16 s CWE3AG: Augmentace dat prostřednictvím rozšířené datové základy pro trénování se v případě CWE3AG ukazuje jako výhodné řešení i v případě fine tuningu. Na VGG16 dosahuje přesnosti 89% při validaci (odpovídající přesnost na tréningových datech byla v tomto případě cca 80% a obecně lze říci, že při validaci dosahuje takto sestavená síť vyšší přenosti než na augmentované trénovací sadě).

3) VGG16 s CWE16: Jako poslední přichází na řadu datová sada CWE16. Datovou sadu jsem upravil tak, aby validační i trénovací data obsahovala stejný počet kategorií a aby byly kategorie stejně pojmenovány. Bez této úpravy běh končí chybou, protože dimenze training a validation datasetu vzájemně neodpovídají (možná existuje v rámci DirectoryIteratoru nějaký parametr, kterým to lze snadno vyřešit, ale ani usilovné googlení nepřineslo potřebný výsledek). Model VGG16 s fine tuning spuštěný nad upraveným datasetem CWE16 vykazuje maximání přesnost 63% (ve feature extraction dosahovala síť maximálně 40-45% přesnosti). Počet, resp. poměr správných kategorizací v rámci tříd koreluje s poměrem příkladů v jednotlivých kategoriích, což je opět důkazem toho, jak důležitá je vyváženost trénovacích dat. Je také možné, že aplikací augmentace podle CWE3AG a/nebo zavedením různých vah pro jednotlivé třídy bych mohl dosáhnout přesnosti o něco vyšší. Vzhledem ke struktuře datasetu ale 63% hodnotím jako solidní výsledek.

4) ResNet50 s CWE3: ResNet50 spuštěný nad výchozí sadou CWE3 dosahuje přesnoti těsně pod 80%, což je překvapivý výsledek, protože jsem očekával stejný nebo o něco málo lepší výsledek, než u VGG16. Z hlediska změny ztrátové funkce a přesnosti je také ResNet50 "divočejší"- zatímco u VGG16 je změna Loss mezi epochami vyrovnaná a postupně klesá, má Loss u ResNet50, i přes celkově klesající trend, daleko větší rozptyl. To je nejspíš dáno přeskoky v

Model		LOSS (CWE16)		ACCURACY (CWE16)	
		Training	Validation	Training	Validation
ResNet50	Best val.	1.2234	1.6972	0.584	0.6231
ResNet50	Average	1.3565	1.4878	0.551	0.599
VGG16	Best val.	0.9546	1.243	0.6925	0.6326
VGG16	Average	1.2435	1.2492	0.621	0.6991

síti, které vedou k vyšší variabilitě výpočtů. Zajímavé mi také přišlo srovnání kategorizace v rámci tříd, kde VGG16 opět podává o něco lepší výsledky, tzn. ResNet50 má menší úspěšnost v kategoriích Bleeding a Polypoids než VGG16.

5) ResNet s CWE3AG: Augmentovaná sada CWE3AG vykazuje maximální přesnost 78.8%, nicméně současně s tím je zde také vyšší pravděpodobnost overfittingu - trénovací data vykazují u maximální validační přesnosti 78.8% vlastní trénovací přesnost 85.5% (tzn. model rozpoznává mnohem lépe data trénovací, než data nová). Se silně augmentovanou datovou sadou si tedy zjevně lépe poradí VGG16 než ResNet50.

6) ResNet50 s CWE16: Na datasetu se 16 kategoriemi (CWE16) vykazuje ResNet50 opět o něco nižší přesnost než VGG16, ale rozdíly nejsou až tak patrné. Také nedochází v průběhu učení k overfittingu tak silně jako u VGG16. Domnívám se tedy, že ResNet50 je citlivější k matoucím vzorům, které se snáze naučí právě na uměle upravených (augmentovaných) datech.

Skripty jsou dostupné na GitHub:

- CNN:.....
- VGG16 s feature extraction:.....
- VGG16 s fine tuning:.....
- ResNet50 s fine tuning:.....

B. CITACE

Chollet, François, and Rudolf Pecinovský. Deep learning v jazyku Python: knihovny Keras, Tensorflow. 2019. Building Powerful Image Classification Models Using Very Little Data. https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html. Accessed 9 Jan. 2021. Dwivedi, Priya. "Understanding and Coding a ResNet in Keras." Medium, 27 Mar. 2019, https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33. Theckedath, Dhananjay, and R. R. Sedamkar. "Detecting Affect States Using VGG16, ResNet50 and SE-ResNet50 Networks." SN Computer Science, vol. 1, no. 2, Mar. 2020, p. 79. DOI.org (Crossref), doi:10.1007/s42979-020-0114-9. Peng, Jie, et al. "Residual Convolutional Neural Network for Predicting Response of Transarterial Chemoembolization in Hepatocellular Carcinoma from CT Imaging." European Radiology, vol. 30, no. 1, Jan. 2020, pp. 413–24. DOI.org (Crossref), doi:10.1007/s00330-019-06318-1.

V. FINAL RESOLUTION / ZHODNOCENÍ

A. OBSAH

V této části přikládám přehledovou tabulku pro CWE16 spuštěný na obou modelech. Pro prezentaci jsem zvolil právě tento dataset, neboť bude (nejspíš) porovnáván v širší skupině.

Mimo porovnání přesnosti a ztrátových funkcí jsem sledoval i běh obou modelů nad CWE16 a spočítal průměrnou dobu běhu na epochu (za oba cykly):

- VGG16 = 66.5 s
- ResNet50 = 77 s

VI. CONCLUSIONS / ZÁVĚRY

Záměrem této seminární práce bylo porovnání dvou široce používaných modelů pro přenesené učení na medicínských datech. Výsledkem je zjištění, že na daném datasetu dosahuje lepších výsledků VGG16 než ResNet50 a navíc je i rychlejší. Naopak, na jednodušších datech (CIFAR3) je ryhlejší ResNet50 a přesnost obou modelů je srovnatelná. Možná ale mnohem důležitější "side-efect"je pro mne zjištění, jak zásadně výkon a výsledek neurální sítě ovlivňuje kvalita vstupních dat a to nejen z hlediska jejich celkového množství, ale i z hlediska vyváženosti příkladů v kategoriích a z hlediska povahy zobrazovaných informací, resp. jejich vzdálenosti od původních předloh využívaného modelu a jeho vah. Zejména je to patrné při použití nevyvážených datových sad, jako je CWE3, případně při velmi malém počtu pro ImageNet neznámých příkladů, jako byl dataset CWE3-RED. Dovedu si tedy představit, že přesnost sítě by razantně vzrostla, pokud bychom měli k dispozici dataset s řádově stovkami příkladů v každé z cílových kategorií nebo pokud bychom měli jako základní model k dispozici síť trénovanou na medicínských datech (namísto ImageNet by se jednalo spíše o jakýsi ImageMed). V takovém případě by se vítězem soutěže mohl stát i ResNet50.

Dále mě potěšilo, že jsem si vyzkoušel preprocessing v podobě vygenerování datasetu pomocí náhodných transformací. Bylo to dobré i pro uvědomění si, jak důležitá (a časově náročná) je v oblasti CNN příprava dat, jejich správné rozdělení a že výsledky sítě nezávisí jen na vyváženém množství, ale i na rozdělení dat v rámci kategorií (síť se prostě z augmentovaných dat nedokáže naučit nové vzory, jen mnoho reprezentací těch existujících). K řešení jsem se pokusil využít různá nastavení hyperparametrů, zejména u klasické CNN a DTL s feature extraction. To mi také přineslo mnoho důležitých informací o tom, jak hyperparametry sítě dokáží i při malé změně viditelně ovlivnit výsledek. A v neposlední řadě jsem rád, že jsem se seznámil s callback funkcemi, pomocí níž jsem např. generoval úspěšnost klasifikace v rámci tříd po každé epoše (s aktuálními vahami), nebo se pokusil předejít overfittingu. Z osobního hlediska pak už jen poznanemám, že mě práce s konvolučními sítěmi velmi bavila a nelituji, že jsem si vybral právě toto téma. I když jsem měl v průběhu práce mnohokrát úplně jiný pocit.

Pokud se ohlédnu zpět na všechny sestavené a otestované modely, pak vzhledem k dosaženým výsledkům je reálné nasazení problematické. Nejde ani tak o celkovou přesnost (87.5% na CWE3 s VGG16 vůbec nevypadá špatně), ale o citlivost na falešně negativní výsledky. Zařazení obrázku kategorie Bleeding jako Normal bude mít totiž pro pacienta i ošetřující personál mnohem závažnější důsledky, než falešně pozitivní výsledek v podobě zařazení obrázku kategorie Normal jako Bleeding. Ve druhém případě taková síť jen snižuje efektivitu

práce kvůli nutnosti ruční kontroly, zatímco v případě prvním může přímo ohrozit pacienta. Cílem by tedy měla být síť, která co nejlépe kategorizuje závažné příklady. Částečně by mohlo být řešením využití dobře kalibrovaných vah pro jednotlivé kategorie, nebo jiná forma preprocesingu (např. zvýšení kontrastu obrázků, odstranění šumu apod.) a zcela jistě by prospěl i mnohem rozsáhlejší trénovací dataset.

REFERENCE

[1] Chollet, François, and Rudolf Pecinovský. Deep learning v jazyku Python: knihovny Keras, Tensorflow. 2019.

[2] O'Shea, Keiron, and Ryan Nash. "An Introduction to Convolutional Neural Networks." ArXiv:1511.08458 [Cs], Dec. 2015. arXiv.org, http://arxiv.org/abs/1511.08458.

[3] Martin Pilát. https://martinpilat.com/cs/prirodou-inspirovane-algoritmy/neuronove-site-konvolucni-site-zpracovani-obrazu. Accessed 7 Jan. 2021.

[4] Weiss, Karl, et al. "A Survey of Transfer Learning." Journal of Big Data, vol. 3, no. 1, May 2016, p. 9. Springer Link, doi:10.1186/s40537-016-0043-6.

[5] ImageNet. http://www.image-net.org/about-overview. Accessed 9 Jan. 2021.

[6] Building Powerful Image Classification Models Using Very Little Data. https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html. Accessed 9 Jan. 2021.

[7] Dwivedi, Priya. "Understanding and Coding a ResNet in Keras." Medium, 27 Mar. 2019, https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33.

[8] Theckedath, Dhananjay, and R. R. Sedamkar. "Detecting Affect States Using VGG16, ResNet50 and SE-ResNet50 Networks." SN Computer Science, vol. 1, no. 2, Mar. 2020, p. 79. DOI.org (Crossref), doi:10.1007/s42979-020-0114-9.

[9] Peng, Jie, et al. "Residual Convolutional Neural Network for Predicting Response of Transarterial Chemoembolization in Hepatocellular Carcinoma from CT Imaging." European Radiology, vol. 30, no. 1, Jan. 2020, pp. 413–24. DOI.org (Crossref), doi:10.1007/s00330-019-06318-1.