VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AKTIVNÍ UČENÍ S NEURONOVÝMI SÍTĚMI

ACTIVE LEARNING AND NEURAL NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **ŠTĚPÁN BENEŠ**
AUTHOR

VEDOUCÍ PRÁCE **Ing. MICHAL HRADIŠ, Ph.D.**
SUPERVISOR

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Beneš Štěpán**
Obor: Informační technologie
Téma: **Aktivní učení s neuronovými sítěmi**
Active Learning with Neural Networks
Kategorie: Zpracování obrazu

Pokyny:

1. Prostudujte základy konvolučních neuronových sítí a aktivního učení.
2. Vytvořte si přehled o současných metodách využívajích aktivní učení a neuronové sítě.
3. Vyberte konkrétní metody a aplikace vhodné pro experimenty.
4. Implementujte navržené metody a proveďte experimenty nad vhodnou datovou sadou.
5. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
6. Vytvořte stručné video prezentující vaši práci, její cíle a výsledky.

Literatura:

- Krizhevsky, A., Sutskever, I. and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012
- Hu et al.: Discriminative Deep Metric Learning for Face Verification in the Wild. CVPR 2014.

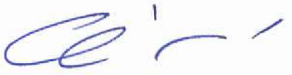
Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Hradiš Michal, Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2017
Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
L.S. 612 66 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Práce se věnuje problematice propojení aktivního učení a konvolučních neuronových sítí při rozpoznávání obrazu. Cílem je pozorovat chování vybraných strategií aktivního učení v širším spektru podmínek. Nejprve se v práci nachází teoretický úvod do problematiky aktivního učení, následně je věnován prostor motivaci a obtížím spojení aktivního učení s neuronovými sítěmi. Samotné chování vybraných strategií při kontinuálním učení je pak pozorováno pomocí několika experimentů, testujících závislost výkonu na obtížnosti datasetu, kvalitě trénovaného modelu, trénovacích epochách, velikosti přidávané sady vzorků, spolehlivosti anotátora a použití techniky pseudo-označování. Výsledky ukazují závislost kontinuálního aktivního učení na obtížnosti datasetu a počtu trénovacích iterací, dále pak odolnost strategií na rozumnou míru chybovosti anotátora. Benefity z pseudo-označování jsou úzce spjaty s dostatečnou kvalitou modelu. Konečně, tradiční strategie aktivního učení mohou v několika případech konkurovat strategiím šitým na míru pro konvoluční sítě.

Abstract

The topic of this thesis is the combination of active learning strategies used in conjunction with deep convolutional networks in image recognition tasks. The goal is to observe the behaviour of selected active learning strategies in a wider array of conditions. The first section of the thesis is dedicated to the theory of active learning, followed by the motivation and challenges of combining them with convolutional neural networks. The goal of this thesis is achieved by a series of experiments, in which the behaviour of active learning strategies is tested for dependencies on the difficulty of the dataset, quality of the learning model, number of training epochs, the size of a batch of samples added in each iteration, the oracle's consistency and the usage of pseudo-labeling technique. The results show the dependency of continuous active learning on the number of training epochs in each iteration and the difficulty of a given dataset. Chosen strategies also seem somewhat resistant to the oracle's faults. The benefits of using pseudo-labeling come hand in hand with the quality of the learning model. Finally, traditional active learning strategies have shown in some cases that they are capable of keeping the pace with modern, tailored strategies.

Klíčová slova

aktivní učení, konvoluční neuronové sítě, rozpoznávání obrazu, strojové učení, umělá inteligence

Keywords

active learning, convolutional neural networks, image recognition, machine learning, artificial intelligence

Citace

BENEŠ, Štěpán. *Aktivní učení s neuronovými sítěmi.* Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Hradiš, Ph.D.

Aktivní učení s neuronovými sítěmi

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Hradiše, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Štěpán Beneš
13. května 2018

Poděkování

Poděkovat chci hlavně svému vedoucímu, Ing. Michalu Hradišovi Ph.D., za skvělé a vždy pozitivní vedení na mnohdy trnité cestě, za trpělivost, ochotu a nakažlivý zápal pro řešené téma. Dále pak svým oddaným přátelům Duc Vu Nguyenovi a Adamu Zugárkovi za postřehy a jazykovou korekturu.

Obsah

Kapitola 1

Úvod

Tato práce se zabývá skloubením konvolučních neuronových sítí a techniky aktivního učení pro řešení problému rozpoznávání obrazu. Hluboké a aktivní učení jsou důležité pilíře strojového učení, které však doposud spíše nezávisle koexistovaly. Hlavně kvůli problému škálovatelnosti a adaptace tradičních metod aktivního učení pro práci s architekturami obsahujícími mnoho parametrů, jako jsou hluboké neuronové sítě. V posledních letech se však udělaly na tomto poli značné pokroky, motivované potenciálem aktivního učení citelně redukovat velikost trénovací sady a tím pádem i čas a peníze věnované označování trénovacích vzorků.

Cílem mé práce je pomocí široké škály experimentů pozorovat účinnost a chování novodobých i tradičnějších strategií aktivního učení ve spojení s neuronovými sítěmi napříč odlišnými parametry.

V úvodní, teoretické, části své práce se věnuji popsání principu aktivního učení a jeho uplatnění v praktických problémech. Zároveň podrobněji rozebírám metody aktivního učení, především ty, které jsem zvolil pro své experimenty. Následuje využití aktivního učení konkrétně s konvolučními neuronovými sítěmi, motivace pro tento postup, i unikátní problémy, jež toto spojení přináší. Na závěr zmiňuji stávající *state-of-the-art* řešení, i směry, kterými se současný výzkum tohoto tématu udává.

Následující kapitola je již o praktické stránce mé práce. Svůj prostor zde mimo samotné implementace a využitých nástrojů dostává popis mnou zvolených datasetů a modelů, jež se na nich budou trénovat. Podstatnější část tvoří samotné experimenty, předpoklady a dílčí závěry.

V samotném závěru pak rekapituluji dosažené poznatky a zamýšlím se nad možnými budoucími experimenty a oblastmi zájmu.

Kapitola 2

Aktivní učení, i v kombinaci s neuronovými sítěmi

2.1 Samotné aktivní učení

Aktivní učení je disciplínou strojového učení, obecněji také umělé inteligence. Klíčovou myšlenkou tohoto přístupu je nechat učící se algoritmus vybírat si data, na kterých se bude trénovat. Takto „zvědavý“ učící se algoritmus pak podá lepší výkon s menším trénovacím úsilím. Tato vlastnost je obzvlášť žádoucí při využití strojového učení s učitelem (*supervised learning*), kdy pro získání rozumného výsledku je potřeba velkého množství označených trénovacích vzorků. Označené vzorky jsou v některých případech k dispozici snadno a lacině, například při hodnocení filmů počtem hvězdiček nebo označení nežádaných zpráv za „spam“. Učící se systémy pak na základě těchto příznaků a hodnocení dokáží lépe filtrovat poštu či doporučovat filmy a knihy. V těchto případech sám uživatel poskytuje zdarma dostatek označených vzorků, ale v mnoha sofistikovanějších scénářích a problémech strojového učení s učitelem jsou kvalitně označené vzorky obtížně získatelné nebo je potřeba velkého časového či finančního úsilí k jejich označení [43].

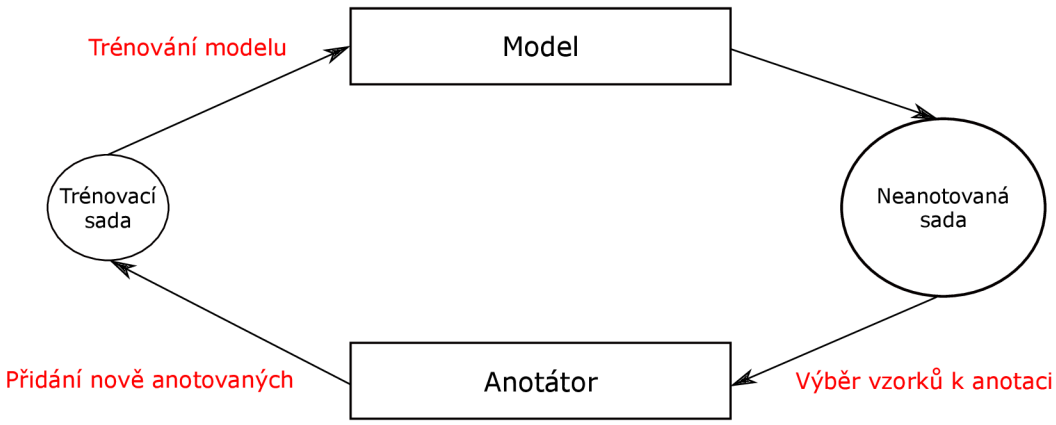
Kupříkladu:

- **Rozpoznávání řeči**. Přesné značení mluveného projevu je velmi časově náročné a vyžaduje spolupráci zkušených lingvistů. Anotace na úrovni slov může trvat až desetinásobek doby trvání nahrávky a anotaci fonémů dokonce až čtyřistanásobek (pro minutovou nahrávku tedy trvá značení slov deset minut a fonémů skoro sedm hodin) [60]. Dialekty a řídce používané jazyky tento problém umocňují.

- **Extrakce informace**. Kvalitní systémy pro extrakci informace musí být trénovány na detailně anotovaných dokumentech. Uživatelé vyznačují entity a vztahy, jež jsou středem zájmu (například jména osob a organizací, zda osoba pracuje pro nějakou společnost apod.). Avšak i pro jednoduché novinové články může lokalizace a vyznačení těchto údajů trvat třicet minut i více [45]. Anotace dat jiných znalostních domén může vyžadovat dodatečnou expertízu, například pro účely extrakce biomedicínských informací je zapotřebí asistence při anotaci od doktorů biologie.

- **Klasifikace a filtrování**. Učení klasifikace dokumentů, obrazových a zvukových dat či videí vyžaduje od uživatele označení každého z tisíců, ba i desetitisíců vzorků. Tento proces může být nesmírně zdlouhavý a často vede k chybám v anotaci z důvodu únavy.

Smyslem aktivního učení je překonat tuto překážku pomocí *dotazů* v podobě neoznačených vzorků, pro které chce získat nejčastěji od lidského anotátora správné označení. Tímhle způsobem se aktivně učící se algoritmus snaží dosáhnout nejvyšší možné přesnosti za využití co nejméně označených dat, tím pádem se minimalizuje úsilí vynaložené anotaci. Ve spoustě moderních problému strojového učení, kdy není obtížné získat samotná data, ale jejich označení s sebou nese velkou časovou nebo finanční zátěž, lze snadno odůvodnit využití a rozvoj technik aktivního učení [43].

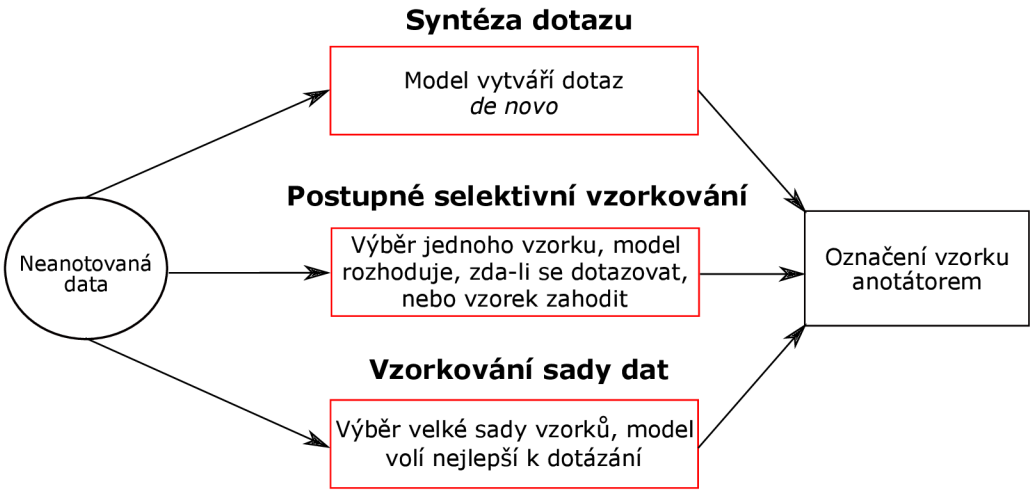


Obrázek 2.1: Cyklus aktivního učení.

Obrázek 2.1 představuje jednoduchý náčrt cyklu aktivního učení. Učící se model začíná trénovat s malou sadou označených vzorků, dotazuje se anotátora o označení jednoho či více vhodně zvolených vzorků z větší, neoznačené, sady a následně nově anotované vzorky přidává do své trénovací sady. Po epizodě trénování opět model žádá další informativní vzorky.

2.1.1 Varianty pokládání dotazů

Existuje několik rozdílných přístupů, jak učící se model může formulovat dotazy. Tři hlavní varianty jsou „Syntéza dotazu“ (*Membership query synthesis*), „Postupné selektivní vzorkování“ (*Stream-based selective sampling*) a „Vzorkování sady dat“ (*Pool-based sampling*). Obrázek 2.2 nastiňuje jejich podstatu a rozdíly, podrobnější popis následuje. Na těchto třech variantách stojí podstatná část dosavadního výzkumu aktivního učení. Všechny tři předpokládají výsledný dotaz v podobě neoznačené instance předané anotátorovi k označení, existují však i alternativní přístupy, přiblíženy v [43].



Obrázek 2.2: Diagram tří hlavních variant pokládání dotazu.

Syntéza dotazu

Jeden z prvních scénářů, jak formulovat dotaz. Učící se model může dotazovat anotaci jakéhokoliv neoznačeného vzorku ze vstupních dat, ale především se předpokládá, že model bude vyžadovat anotaci jím vygenerovaného umělého vzorku. Efektivní syntéza dotazů často zlepšuje učenlivost modelu jak v oborech konečných problémů [1] [2], tak v regresních úkolech strojového učení, například při odhadu souřadnic pozice robotického ramena [11].

Tento přístup je přijatelný pro mnoho praktických problémů, avšak značení často velmi náhodných generovaných dotazů se může projevit jako velmi obtížné pro lidského anotátora. Příkladem budiž práce [30], kdy byla syntéza dotazů použita v tandemu s lidskými anotátory pro klasifikaci rukou psaných znaků. Nečekaným problémem se projevila skutečnost, že většina modelem vygenerovaných dotazů se skládala z umělých, hybridních, znaků bez sémantického významu. Obdobné obtíže lze předpokládat i při zpracování přirozené řeči. Tyto limitace daly za vznik již zmíněným dalším variantám formulování dotazu.

Nicméně, práce kolektivu kolem R. Kinga [25] [24] popisují inovativní a slibný způsob, jak syntézu dotazů využít vskutku efektivně k řešení skutečného problému. Využívají „robotického vědce“, jenž dokáže vykonat sérii autonomních biologických experimentů pro nalezení metabolických cest kvasinky *Saccharomyces cerevisiae*. V tomto případě neoznačený vzorek představuje směs chemických roztoků, které tvoří živnou půdu, a zároveň konkrétní kvasinková mutace. Označením vzorku je pak skutečnost, zda-li se mutaci v této živné půdě dařilo. Všechny experimenty jsou autonomně syntetizovány a fyzicky provedeny laboratorním robotem. Tenhle přístup k využití aktivního učení vyústil v trojnásobné snížení ceny experimentálního materiálu oproti nasazení experimentů s nejnižší cenou, a až stonásobné snížení v případě náhodně generovaných experimentů.

V doménách, kdy označení vzorků neposkytují lidští anotátoři, ale výsledky experimentů, jako ve zmíněném případě, se může syntéza dotazu jevit jako slibný směr pro automatizované bádání [43].

Postupné selektivní vzorkování

Alternativou syntézy dotazu je postupné selektivní vzorkování neoznačených dat. Klíčovým předpokladem je, že neoznačená data lze získat snadno a levně. Následně jsou data ze vstupní distribuce vzorkována a učící se model rozhoduje, zda-li pro vybraný vzorek bude chtít znát jeho označení od anotátora, nebo se jej zbaví. Zpravidla se k dotázání vybírá právě jeden vzorek a postupně se prochází celá vstupní distribuce. Pro uniformní distribuci vstupních dat se může tato metoda svým chováním podobat první zmíněné metodě. Pokud je vstupní distribuce neuniformní či neznámá, je stále zaručena smysluplnost dotazů, jelikož vycházejí ze skutečného rozložení distribuce [10] [9].

Rozhodnutí, zda-li se pro vybraný vzorek dotazovat, či ne, lze uskutečnit několika způsoby. Jedním z nich je ohodnotit vzorky pomocí nějaké „míry informativnosti“ nebo „dotazovací strategie“ (o nich více v další podkapitole) a následně provést vážené náhodné rozhodnutí takovým způsobem, že více informativní vzorky mají větší šanci na zvolení k dotázání [13]. Dalším přístupem je výpočet explicitního *regionu nejistoty* [9], čili části vstupních dat, jimiž si je učící se model nejistý, a dotazovat pouze tuto část vstupních dat. Jednoduchou implementací může být existence hranice míry informativnosti, která tento region definuje. Vzorky, které po vyhodnocení libovolnou strategií tuto hranici překonají, jsou dotazovány. Kompletní výpočet tohoto regionu netriviálními metodami je však časově náročný [36]. V praxi jsou tím pádem používány pouze aproximace [47] [14].

Tento přístup byl studován na několika reálných úkolech, včetně značení slovních druhů [13], extrakce informace [57], časování senzorů [26] a rozlišování významů slov [18].

Vzorkování sady dat

V mnoha praktických učících problémech je k dispozici velká kolekce neoznačených dat. Tato skutečnost stála za vznikem další varianty formulování dotazů [32], jež předpokládá existenci malé sady označených dat $\mathcal{L}$ a podstatně větší sady dat neoznačených, $\mathcal{U}$. Dotazy jsou selektivně brány ze sady $\mathcal{U}$, o níž se obyčejně předpokládá, že zůstává beze změny, avšak nemusí to být pravidlem. Vzorky jsou typicky vybírány „hladově“, s ohledem na míru informativnosti, podle které jsou všechny vzorky sady $\mathcal{U}$ ohodnoceny. Diagram na straně 4 popisuje aktivní učení s využitím právě této metody.

Tento přístup byl studován a našel uplatnění v mnoha reálných problémech, jako například klasifikace textu [32] [22], extrakce informace [44], klasifikace obrazu [59], klasifikace videa [56], rozpoznávání řeči [52], i při diagnóze rakoviny [33] a mnoha dalších.

Hlavní rozdíl mezi postupným selektivním vzorkováním a vzorkováním sady dat spočívá v tom, že první zmíněný přístup prochází neoznačená vstupní data sekvenčně a pro každý vzorek rozhoduje, jestli požádá o jeho anotaci, kdežto druhý zmíněný přístup na základě hodnotících kriterií seřadí celou kolekci dat a posléze vybírá nejlepší vzorek k dotázání. Vzorkování sady dat je v současnosti populárnější variantou, ale existují scénáře, kdy postupné selektivní vzorkování může být vhodnější, například při práci s omezenou pamětí či výpočetní silou, jako v případě mobilních a vestavěných zařízení [43].

2.1.2 Strategie výběru vhodných vzorků

Všechny varianty pokládání dotazů potřebují způsob, jak ohodnotit informativnost neoznačených vzorků. Za tímto účelem bylo postupem času navrženo mnoho strategií (*query strategies*). Tato sekce je věnována vybraným strategiím a jejich principům.

Strategie pracující s nejistotou modelu

Jedná se o nejjednodušší a nejčastěji používanou strategii výběru vzorků vhodných k označení. Anglicky *uncertainty sampling* [32], podstatou je výběr takových vzorků, o kterých si je učící se model nejméně jistý, jak by je sám označil. Pro probabilistické modely je tento přístup poměrně přímočarý, mějme kupříkladu model učící se binární klasifikaci, v takovém případě model pomocí této strategie vybere k označení ty vzorky, u nichž si je jistý o jejich označení s pravděpodobností okolo 0.5 [32] [31]. Pro klasifikaci do vícero tříd existují tři rozdílné metody, jak na nejistotu modelu o označení vzorků nahlížet [43].

První z nich, metoda „nejnižší jistoty“ (*least confident*), jednoduše dotazuje takový vzorek, u něhož si je model nejméně jistý jeho označením:

$$x^*_{LC} = \underset{x}{argmax}\ 1 - P_\theta(\hat{y}|x)$$

Kde x^*_{LC} představuje nejinformativnější vzorek a $\hat{y} = \underset{y}{argmax}\ P_\theta(y|x)$ označení vzorku třídou, do které podle modelu θ nejpravděpodobněji spadá. Další interpretací může být, že model vybírá pro značení takový vzorek, u kterého si je nejvíce jist, že by jeho označení sám pokazil. Tahle metoda je populární například u statistických sekvenčních modelů při úkolech extrakce informace [12] [44].

Úskalím takové metody je však skutečnost, že bere v potaz při rozhodování pouze informaci o nejpravděpodobnějším označení vzorku podle mínění modelu, a tím pádem „zahazuje“ informace o rozložení pravděpodobností zbývajících tříd. Proto vznikla další metoda, zvaná anglicky *margin sampling* [40]:

$$x^*_{MS} = \underset{x}{argmin}\ P_\theta(\hat{y_1}|x) - P_\theta(\hat{y_2}|x)$$

$\hat{y_1}$ a $\hat{y_2}$ představují první respektive druhou nejpravděpodobnější třídu do které dle modelu vzorek spadá. Tato metoda se pokouší o jistou korekci nedostatků metody předcházející tím, že bere v potaz při výběru vzorku i druhou nejvyšší třídní pravděpodobnost. Intuice je taková, že v případě velkého rozdílu mezi první a druhou třídní pravděpodobností má model solidní jistotu, jak vzorek označit. V případě malého rozdílu pak modelu dotázat skutečné označení vzorku pomůže lépe tyto dvě třídy rozlišovat. Při úkolech obsahujících velmi mnoho tříd však i tato metoda ignoruje rozložení pravděpodobnosti drtivé většiny tříd [43].

Třetí, obecnější a možná i nejpopulárnější metoda, využívá k vyjádření míry nejistoty entropii [48]:

$$x^*_{EN} = \underset{x}{argmax}\ -\sum_i P_\theta(y_i|x) \log P_\theta(y_i|x)$$

Entropie je důležitým pojmem teorie informace. Udává míru informace potřebné pro zakódování distribuce. Jako taková je často brána jako míra nejistoty či nečistoty v oboru strojového učení. Pro binární klasifikaci se chová identicky jako ostatní dvě metody, avšak pro klasifikaci vícero tříd či práci s komplexnějšími strukturovanými vzorky, jako jsou sekvence [44] a stromy [23], se nejsnadněji zobecňuje [43].

Oproti předchozím metodám metoda pomocí entropie neupřednostňuje takové vzorky, u kterých je pouze jedna třída velmi nepravděpodobná, protože si je model jist, že do téhle třídy vzorek nepatří. První dvě zmíněné metody po takových vzorcích nemají problém sáhnout, pokud si model není mezi zbývajícími třídami jist. Empirické srovnání těchto metod [29] [41] [44] vyvodilo smíšené závěry, předpokládá se však, že výhodnost jednotlivých metod záleží na zvolené úloze. Intuitivně se však metoda entropie jeví vhodná v situacích, kdy je úkolem minimalizovat logaritmickou ztrátu, kdežto zbývající dvě metody jsou užitečnější při snižování klasifikační chyby, jelikož lépe umožňují modelu rozlišovat jednotlivé třídy [43].

Strategie „výboru modelů“

[47] přichází s další strategií posouzení vhodnosti vzorků k označení a to pomocí výboru modelů (*query-by-committee*). Tento přístup zahrnuje výbor $\mathcal{C} = \{\theta^{(1)}, \ldots, \theta^{(C)}\}$ modelů, které jsou trénovány na stejné označené sadě dat $\mathcal{L}$, ale představují odlišné hypotézy. Každý člen tohoto výboru může hlasovat, jak by označil vzorky kandidující na vybrání k označení anotátorem. Nejinformativnějším vzorkem je pak takový, u kterého jsou modely nejvíce v rozporu. Fundamentální premisou tohoto přístupu je minimalizování prostoru verzí, jenž je souborem hypotéz, jež jsou konzistentní se současnou označenou datovou sadou $\mathcal{L}$ [43].

Neexistuje žádná literaturou doporučená velikost výboru modelů, tento parametr zůstává závislý na aplikaci a druhu modelů. Avšak i malé výbory, obsahující dva nebo tři modely, se ukázaly jako velmi efektivní v praxi [47] [44] [35]. Pro určení míry rozporu mezi členy výboru se ujaly dva hlavní přístupy.

Prvním z nich je entropie hlasů (*vote entropy*) [13]:

$$x^*_{VE} = \underset{x}{argmax} \; -\sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C}$$

$V(y_i)$ představuje počet hlasů, které třídě dle predikce modelů vzorek náleží a $\mathcal{C}$ je výbor modelů. Na tento přístup lze nahlížet jako na generalizaci strategie pracující s nejistotou modelu pomocí míry entropie.

Další navrženou metodou pro měření rozporu modelů je průměrná *Kullback-Leiblerova divergence* [35]:

$$x^*_{KL} = \underset{x}{argmax} \; \frac{1}{C} \sum_{c=1}^{C} D(P_{\theta^{(c)}} \| P_{\mathcal{C}})$$

Kde:

$$D(P_{\theta^{(c)}} \| P_{\mathcal{C}}) = \sum_i P_{\theta^{(c)}}(y_i|x) \log \frac{P_{\theta^{(c)}}(y_i|x)}{P_{\mathcal{C}}(y_i|x)}$$

Zde $\theta^{(c)}$ představuje jednotlivý model z výboru a $\mathcal{C}$ celý výbor modelů. $P_{\mathcal{C}}(y_i|x) = \frac{1}{C} \sum_{c=1}^{C} P_{\theta^{(c)}}(y_i|x)$ je pak modely „dohodnutá“ pravděpodobnost, že y_i je správné označení. Kullback-Leiblerova divergence [28] představuje míru odlišnosti dvou pravděpodobnostních rozložení. Takto použita vede k zvažování jako nejinformativnějšího vzorku takového, u nějž má jakýkoliv člen výboru nejvyšší průměrný rozdíl distribuce tříd oproti výborem usnesené distribuci [43].

Očekávaná změna modelu

Další možnou strategií, jak zvolit vzorek k označení, je vybrat takový, který by přispěl k největší změně současného modelu. Příkladem pracujícím na tomto principu budiž metoda s názvem „očekávaná délka gradientu“ (*expected gradient length*) pracující s diskriminativními probabilistickými modely [46] [44].

Teoreticky lze tuto metodu aplikovat na jakýkoliv scénář, kde je použito trénování pomocí gradientu. Jelikož diskriminativní probabilistické modely k optimalizaci běžně gradient využívají, změnu modelu lze posoudit délkou trénovacího gradientu (například vektoru použitého k úpravě hodnot parametrů). Učící se model by tedy měl dotazovat takový vzorek x, který by po označení a přidání do trénovací sady $\mathcal{L}$ způsobil nejrozsáhlejší trénovací gradient. Budiž $\nabla\ell_\theta(\mathcal{L})$ gradientem funkce ℓ vzhledem k parametrům modelu θ. Dále nechť je $\nabla\ell_\theta(\mathcal{L}\cup\langle x,y\rangle)$ gradientem novým, který by byl získán přidáním dvojice $\langle x,y\rangle$ do označené trénovací sady $\mathcal{L}$. Jelikož model nezná předem skutečné označení vzorku y, musíme délku gradientu spočítat jako předpoklad skrze všechna možná označení [43]:

$$x^*_{EGL} = \underset{x}{argmax} \sum_i P_\theta(y_i|x)\left\|\nabla\ell_\theta(\mathcal{L}\cup\langle x,y\rangle)\right\|$$

Metoda preferuje takové vzorky, které nejvíce ovlivní parametry modelu, nezávisle na třídě, jaké přísluší. Na praktických úlohách byla ověřena funkčnost tohoto řešení, avšak může být výpočetně velmi náročné, obzvlášť pokud je prostor rysů a počet tříd poměrně rozsáhlý. Dále může tuto metodu zmást pokud nejsou rysy patřičně škálované. Informativnost vzorku může být nadhodnocena jen proto, že hodnota nějakého rysu může být neobyčejně vysoká a způsobí velký trénovací gradient. Regularizace parametrů pomáhá tento vedlejší účinek do jisté míry eliminovat [19] [7].

Očekávaná redukce chyby

Strategie podobná výše uvedené, avšak namísto změny modelu měří potenciální redukci chybovosti modelu. Myšlenkou je odhadnout budoucí chybu modelu trénovaného na $\mathcal{L}\cup\langle x,y\rangle$ při jeho použití na zbytku neoznačené sady $\mathcal{U}$ (která v tomto případě plní roli validační sady) a dotazovat takový vzorek, pro který je očekávaná budoucí chyba minimální. Minimalizovat lze buď očekávanou 0/1-ztrátu [43]:

$$x^*_{0/1} = \underset{x}{argmin} \sum_i P_\theta(y_i|x)\left(\sum_{u=1}^{U} 1 - P_{\theta^{+\langle x|y_i\rangle}}(\hat{y}|x^{(u)})\right)$$

$\theta^{+\langle x|y_i\rangle}$ představuje nový model, poté co byl znovu trénován s dvojicí $\langle x|y_i\rangle$ již přidanou do trénovací sady $\mathcal{L}$. Stejně jako v případě metody očekávané délky gradientu ani zde není známa skutečná třída vzorků, je tedy potřeba využít aproximaci skrze všemi možnými označeními. Úmyslem redukce 0/1-ztráty je snížit počet chybných predikcí.

Druhým, méně svazujícím přístupem je minimalizovat očekávanou logaritmickou ztrátu:

$$x^*_{log} = \underset{x}{argmin} \sum_i P_\theta(y_i|x)\left(-\sum_{u=1}^{U}\sum_j P_{\theta^{+\langle x|y_i\rangle}}(y_j|x^{(u)}) \log P_{\theta^{+\langle x|y_i\rangle}}(y_j|x^{(u)})\right)$$

Což je ekvivalentem redukce očekávané entropie nad $\mathcal{U}$. Jinými slovy, účelem je maximalizace očekávaného zisku informace dotazem x [43].

Výhodou této strategie je skoro úplná optimálnost a zároveň nezávislost na druhu modelu. Potřebuje pouze vhodnou ztrátovou funkci a způsob, jak odhadnout pravděpodobnosti třídních příslušností vzorků. Příkladem úspěšného použití budiž s naivním Bayesovským klasifikátorem [39], Gaussovskými náhodnými poli [61], logisitckou regresí [21] a podpůrnými vektorovými stroji [37]. Ve většině případů je však tato strategie také zdaleka výpočetně nejnáročnější. Nejenže je potřeba odhadnout očekávanou budoucí chybu nad $\mathcal{U}$ pro každý vzorek, ale každý model musí být znovu trénován pro každé možné označení vzorku. Pro neparametrické modely, jako jsou Gaussovská náhodná pole, je tato strategie vcelku efektivní a praktická [61]. Pro drtivou většinu ostatních druhů modelů je však výpočetní cena drastická. Z toho důvodu našla tato strategie uplatnění především v jednoduchých binárních klasifikačních úlohách [43].

Strategie pracující s váženou hustotou rozložení vzorků

Hlavní výhodou předchozí strategie je skutečnost, že se zaměřuje na vstupní prostor jako celek a je mnohem méně náchylná k výběru anomálních vzorků oproti jednodušším strategiím. Anomálními vzorky si sice model může být nejistý, ale po jejich dotázání modelu nejspíše výrazně nepomůžou, jelikož nejsou dostatečně reprezentativními s ohledem na vzorky ostatní. Z tohoto poznatku čerpají [44] při tvorbě strategií, které pracují s váženou hustotou rozložení vzorků ve vstupním prostoru. Klíčovou myšlenkou je, že informativní vzorky jsou nejenom ty, u kterých si je model nejistý, ale především pak ty, které jsou reprezentativní vzhledem k pravé distribuci vstupních dat (vyskytují se v hustě obsazených regionech vstupního prostoru). Obecnou metodou, jak tohoto docílit, budiž [43]:

$$x^*_{ID} = \underset{x}{argmax}\ \phi_A(x) \times \left(\frac{1}{U} \sum_{u=1}^{U} \text{sim}(x, x^{(u)}) \right)^{\beta}$$

$\phi_A(x)$ představuje informativnost vzorku x získanou nějakou základovou strategií vybírání vzorků A, například strategií využívající nejistoty či výborem modelů. Druhá půle rovnice zvažuje informativnost vzorku x pomocí jeho průměrné podobnosti ostatním vzorkům vstupní distribuce (aproximované sadou $\mathcal{U}$). Parametr β řídí důležitost tohoto dílčího výsledku. Variantou může být nejdříve rozdělit sadu $\mathcal{U}$ na shluky (*clustery*) a počítat průměrnou podobnost pro vzorky nacházející se ve stejném shluku [43].

O zformulování tohoto přístupu jako samostatné strategie se postarala práce [44], avšak náznaky využití hustoty rozložení a reprezentativnosti se vyskytovaly i v předcházejících pracích. Kupříkladu [35] kombinují výbor modelů s váženou hustotou rozložení při klasifikaci textu. [38] navrhli přístup založený na hustotě rozložení, který nejdříve vstupní data shlukuje a snaží se předejít dotazování anomálií tím, že předává informace o označení všem vzorkům ve stejném shluku. Obdobně [55] využívají shlukování pro tvorbu sady dotazů pro dávkové aktivní učení s podpůrnými vektorovými stroji. Všechny tyto práce vyzdvihují tuto strategii jako výhodnější než alternativy nepracující s mírou reprezentativnosti vzorků. Ba co více, práce [44] poukazuje, že v případě předvýpočtu hustoty rozložení a následného uložení do cache lze výběr vzorku podstatně urychlit.

2.1.3 Problematické oblasti aktivního učení v praktickém nasazení

Donedávna hlavní otázkou bádání na poli aktivního učení bylo, zda-li se pomocí dotazů dá model trénovat na menší datové sadě. Odpovědí, při dodržení několika předpokladů, je ano. Ovšem předpoklady, jako přítomnost pouze jednoho anotátora, který má navíc vždy pravdu, nebo uniformní či žádné ceny za označení vzorků, v praktických problémech často neplatí [43]. Ve finální podkapitole o aktivním učení věnuji prostor několika překážkám praktického nasazení aktivního učení, z nichž některé jsou esenciální pro skloubení aktivního učení a konvolučních neuronových sítí a bude jím věnován prostor v další kapitole i v mých experimentech.

Dávkové aktivní učení

Ve většině případů nasazení aktivního učení jsou vzorky označeny a přidávány sériově, po jednom. Někdy však trénování modelu může být časově náročné nebo drahé. Sériové přidávání může být také neefektivní v případě, že máme k dispozici několik anotátorů pracujících paralelně. Oproti tomu dávkové aktivní učení (*batch-mode active learning*) dovoluje modelu formulovat dotaz jako skupinu vzorků k označení, což je podstatně výhodnější při použití několika paralelně pracujících anotátorů či v případě dlouhé doby trénovaní modelu [43].

Výzvou tohoto přístupu je, jak optimálně složit skupinu vzorků $\mathcal{Q}$ k dotazování. Slepě vybírat n nejlepších dle strategie, která bere v potaz pouze jednotlivé vzorky, a nebere v potaz jistý překryv informativnosti již vybraných vzorků, příliš dobře nefunguje. Pro podpůrné vektorové stroje několika algoritmů pro dávkové aktivní učení již navrženo bylo, například [5] ve své práci explicitně zahrnuje míru rozmanitosti při tvorbě skupiny $\mathcal{Q}$, [55] zase skládá $\mathcal{Q}$ z center shluků vzorků, jež leží nejblíže klasifikační hranici. Ve většině případů je využito „hladových“ heuristik, aby došlo k zajištění jak rozmanitosti skupiny vzorků, tak informativnosti. Povětšinou takto konstruované skupiny vzorků přináší zlepšení oproti náhodnému výběru, či výběru n nejlepších vzorků nezávisle na složení skupiny [43].

Dávkové aktivní učení je výhodné ve spojení s hlubokými konvolučními neuronovými sítěmi, jejichž trénování zabírá podstatné množství času. Avšak jelikož se jedná o poměrně nový přístup využití aktivního učení, mnoho metod, jak vybrat vhodnou skupinku vzorků k označení, prozatím neexistuje a jsou předmětem usilovného bádání [16]. O tomto spojení a mnou zvolených metodách pro experimenty rozepisuji více v podkapitole 2.2.

Nepřesný anotátor

Dalším častým předpokladem ve většině prací na poli aktivního učení je vysoká kvalita označení vzorků anotátorem. I když se o značení vzorků mohou starat lidští experti, hned z několika důvodů nemusí být tato označení vždy spolehlivá. Hlavními důvody nepřesných označení může být jednak únava či ztráta soustředění lidského anotátora po delší době anotace, jednak prostý fakt, že některá data jsou implicitně obtížná na správné označení pro lidské i mechanické anotátory [43].

Hlavní otázkou zůstává, jak nejlépe připravit model pro práci s mylnými anotátory. Konkrétněji, kdy by učící se model měl vyžadovat nové, potenciálně mylně označené vzorky, a kdy opakované označení vzorků již označených, o kterých má pochybnosti. Práce [49] se tomuto problému věnuje pomocí několika heuristik, jež berou v potaz odhady nejistot jak modelu, tak anotátorů, a výsledkem je, že pomocí opakovaného dotazování lze zlepšit výkon aktivně učícího se modelu. Použité heuristiky však neberou ohled na možnou časem proměnlivou chybovost anotátora. Jak se model může vypořádat s nekonzistentní chybovostí

anotátora v čase zůstává otevřenou otázkou, stejně jako vliv finanční motivace na výkon anotátorů, nebo co v případě, pokud data budou inherentně obtížná na označení nezávisle na anotátorovi, a opakované žádosti o přeoznačení nebudou efektivní [43].

Ve svých experimentech zkoumám pouze prostý vliv různých úrovní nepřesností anotátora na výkon učící se konvoluční neuronové sítě, a zda-li použitím pseudooznačení lze tento vliv alespoň částečně zvrátit.

Volba a změna modelu

Výsledná trénovací sada vybudovaná aktivním učením vychází z distribuce, která je implicitně spjata s druhem učícího se modelu použitého pro výběr dotazů. Tato skutečnost může být problematická, chceme-li vybudovanou trénovací sadu použít pro učení jiného druhu modelu, nebo pokud od začátku neznáme vhodný druh modelu pro nasazení k řešení konkrétního problému [43]. Naštěstí to tak nemusí být úplně vždy, například práce [31] poukazuje, že rozhodovací stromy stále dokážou benefitovat z trénovací sady vytvořené pomocí naivních Bayesovských klasifikátorů používajících aktivní učení pracující s nejistotou. Dále [51] předvádí, že informace extrahované výborem MaxEnt modelů mohou být efektivně použity k trénování podmíněných pravděpodobnostních těles (*conditional random fields*), a vedou k úspoře ceny v porovnání s náhodným vzorkováním.

Avšak [3] na problém ztráty efektivity při znovupoužití trénovací sady jiným druhem modelu již narazili, a navrhli řešení v podobě heterogenního výboru různých druhů modelů a zároveň semi-automatického označování vzorků, vedoucímu k snížení potřebné práce od lidských anotátorů. Obdobně [34] použili aktivní učení s heterogenním výborem modelů neuronových sítí a rozhodovacích stromů na řešení problémů, u kterých vhodnější typ modelu není předem znám. Výsledkem byla obecně informativnější finální trénovací sada.

Ohledně této limitace aktivního učení je potřeba dalšího výzkumu, obecně však platí, je-li vhodná třída modelu pro konkrétní problém známa předem, lze aktivní učení bezpečně využít. Jinak může být lepší variantou alespoň zpočátku používat náhodné vzorkování, případně využít výbor rozličných modelů pro výběr dotazů [43].

Ukončovací kritéria

Finální důležitou otázkou aktivního učení a jeho nasazení v praktických problémech, kterou zde zmíním, je vědět, kdy učení ukončit. Jednou ukončovací podmínkou může být stav, kdy cena získávání dalších označených vzorků je vyšší, než cena chyb, kterých se současný model může dopustit. Druhým pohledem na věc je pozorovat, zda-li se učící se model stále výrazně zlepšuje. Pakliže ne, získávání dalších dat je nejspíše mrhání zdrojů a je lepší učení ukončit. Jelikož principem aktivního učení je zlepšení přesnosti učícího se modelu a zároveň snížení nákladů na získávání dat, je nasnadě přemýšlet o metodě, jak by model sám mohl rozhodnout o ukončení svého učení [43].

Několik takových metod bylo vskutku navrženo [53] [50] [4]. Jsou si velmi podobné a pracují s premisou, že model obsahuje jistou míru stability. Aktivní učení přestává být užitečné v momentě, kdy tato míra stability začne degradovat. Dle [43] tyto metody můžou najít uplatnění v několika případech, mnohem častěji se však jako skutečná ukončovací kritéria prokážou finanční či externí faktory.

2.2 Aktivní učení ve spojení s neuronovými sítěmi

Aktivní a hluboké učení představují dva důležité pilíře strojového učení, doposud však spíše nezávisle koexistovaly, kvůli složitosti jejich skloubení. Hlavními překážkami jsou škálovatelnost a adaptabilita běžných metod aktivního učení na architektury obsahující ohromné množství parametrů, jako jsou neuronové sítě. Další obtíží je celkové potřebné množství trénovacích iterací, jež zůstává výpočetně náročné i s nasazením GPU [16]. Většina prací na poli aktivního učení taktéž počítá s přidáváním vzorků po jednom. To ovšem v případě neuronových sítí není v drtivé většině scénářů přijatelné, jelikož jeden vzorek pravděpodobně nebude mít z důvodu lokální optimalizace na přesnost dostatečný vliv, a jelikož by bylo potřeba příliš mnoha trénovacích iterací. Je proto esenciální používání dávkového aktivního učení [42].

Podívejme se v tomto ohledu na strategie výběru vzorků zmíněné v podkapitole 2.1.2. Strategie pracující s nejistotou modelu jsou dostatečně jednoduché a výpočetně nenáročné, aby se dali použít v kombinaci s trénováním hlubokých neuronových sítí. Jejich nedostatkem je však konstrukce informativní skupiny vzorků pro dotazování, jelikož berou v potaz jen individuální neoznačené vzorky nezávisle na ostatních. Výsledná skupina vzorků pak nemusí být dostatečně reprezentativní s ohledem na celý vstupní dataset a velmi pravděpodobně obsahuje vzorky informativností podobné jiným, vzniká „překryv“ [42]. Strategie na bázi očekávané změny modelu nebo očekávané redukce chyby jsou v kombinaci s hlubokým učením pro svou výpočetní náročnost naprosto nepoužitelné. Obdobně nasazení výboru modelů, alespoň ve své původní podstatě. Práce [15] představuje inovativní využití úplné konvoluční neuronové sítě jako učícího se modelu a výboru parciálních konvolučních neuronových sítí pro výběr skupiny vzorků pro dotázání. Jako nejslibnější se jeví metody pracující s celkovým rozložením vstupní sady, hledající reprezentativní a zároveň informativní vzorky [42].

Alternativním řešením problémů skloubení aktivního učení a neuronových sítí jsou různé podpůrné techniky, jimiž lze kompenzovat nedostatky strategií aktivního učení. Příkladem budiž zmíněné parciální konvoluční neuronové sítě [15], nebo technika pseudo-označování vzorků učícím se modelem [54].

Současný výzkum kombinace aktivního učení a hlubokého učení se uchyluje dvěma směry, a to hledáním nových vhodných strategií aktivního učení, nebo podpůrných technik, s jejichž pomocí stávající strategie aktivního učení mohou být užitečné. V této podkapitole přiblížím právě zmíněné pseudo-označování a několik strategií navržených přímo pro kombinaci aktivního a hlubokého učení.

2.2.1 Inovativní strategie výběru vzorků

Zajímavou novou strategii přináší [16], založenou na variační Bayesovské inferenci pro konvoluční neuronové sítě, použité pro výběr vhodných vzorků k označení, v tandemu s odhadem maximální podobnosti (*maximum likelihood estimator*) pro formulování priorní a posteriorní distribuce parametrů sítě. Nabízí efektivní a škálovatelné řešení pro neuronové sítě, pracující s dávkovým dotazováním vzorků.

Pro své experimenty jsem však zvolil jinou nově navrženou strategii, reprezentující současný state-of-the-art, jež je dílem [42]. Základní premisou je redefinování problému aktivního učení jako problému *core-set selection*, čili hledání takových bodů vstupního prostoru, na nichž trénovaný model podá obdobný výkon, jako model trénovaný na všech bodech vstupního prostoru. Ztrátová funkce se v takovém případě skládá tradičně z trénovací chyby, generalizační chyby a navíc i z *core-set* ztráty. Ta představuje rozdíl mezi průměrnou empirickou ztrátou nad sadou dat, pro které máme k dispozici označení, a průměrnou empirickou ztrátou nad celou sadou, včetně dat neoznačených. V tomto pohledu na problém aktivního učení představuje právě *core-set* ztráta kritickou složku celkové ztráty, a je definována následovně:

$$\min_{s^1:|s^1|\leq b} \left| \frac{1}{n} \sum_{i\in[n]} l(x_i, y_i; A_{s^0\cup s^1}) - \frac{1}{|s^0 + s^1|} \sum_{j\in s^0\cup s^1} l(x_j, y_j; A_{s^0\cup s^1}) \right|$$

Kde $l(x_i, y_i; A)$ představuje ztrátovou funkci, A učící se algoritmus se sadou dat s, a x, y velkou kolekci dat a jejich označení. Laicky řečeno, máme-li počáteční malou označenou sadu s^0 a rozpočet pro pokládání dotazů b, snažíme se najít takové vzorky k dotázání s^1, aby výkon modelu po trénování na menší označené sadě byl co nejblíže výkonu modelu trénovaného na celé datové sadě. Tento optimalizační úkol však nelze přímo spočítat, jelikož nemáme k dispozici všechna označení vzorků. [42] proto přichází s horní mezí, již lze optimalizovat, v podobě teorému, jehož výsledný tvar je:

$$\left| \frac{1}{n} \sum_{i\in[n]} l(x_i, y_i; A_s) - \frac{1}{|s|} \sum_{j\in s} l(x_j, y_j; A_s) \right| \leq \delta(\lambda^l + \lambda^\mu LC) + \sqrt{\frac{L^2 \log(1/\gamma)}{2n}}$$

Ten tvrdí, že množina s je δ pokrytím množiny s^*. Což znamená, že množina kružnic o poloměru δ se středy v bodech množiny s dokáže pokrýt celou množinu s^*. Celou *core-set* ztrátu lze ohraničit pomocí poloměru pokrytí a výrazu blížícího se nule v závislosti pouze na parametru n. S touto horní mezí již lze použít aktivní učení, problémem k řešení se stává $\min_{s^1:|s^1\leq b|} \delta_{s^0\cup s^1}$. Ten je ekvivalentem tzv. *K-Center* problému, též nazývaného *min-max facility location problem*. Intuitivněji jej lze popsat takto; je potřeba vybrat b center tak, že největší vzdálenost mezi každým bodem vstupního prostoru a jeho nejbližším centrem je co nejmenší. Formálně:

$$\min_{s^1:|s^1|\leq b} \max_i \min_{j\in s^1\cup s^0} \Delta(x_i, x_j)$$

Řešení je naneštěstí NP-Hard, lze však získat efektivní, 2 - *OPT* řešení pomocí „hladového“ přístupu, popsaného algoritmem 1.

Algorithm 1 K-Center Greedy

Input: data x_i, počáteční menší označená sada s^0, rozpočet b

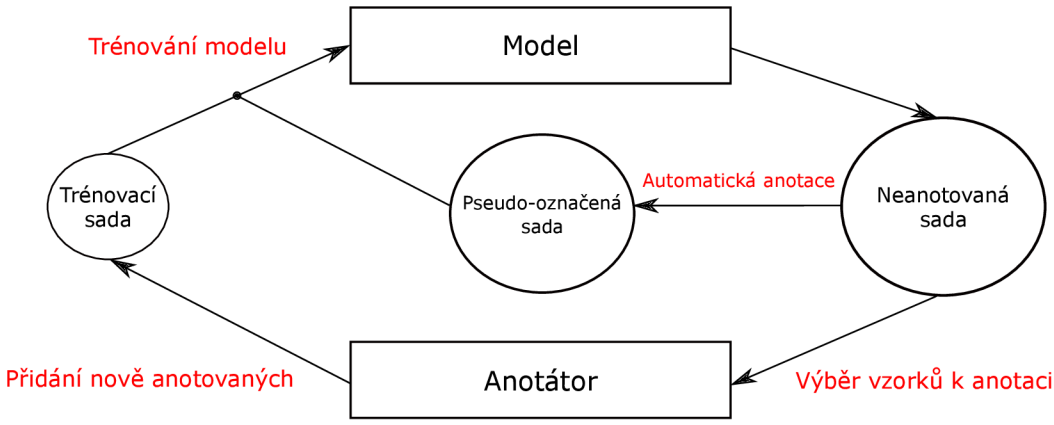
Inicializace $s = s^0$
repeat
  $u = \text{argmax}_{i \in [n] \setminus s} \min_{j \in s} \Delta(x_i, x_j)$
  $s = s \cup \{u\}$
until $|s| = b + |s^0|$

Output: s $\setminus$ s^0

Po každé trénovací iteraci jsou takto vybrané vzorky přidávány do trénovací sady s^0. [42] dále toto řešení vylepšuje na *Robust K-Center*, pro své experimenty se však spokojím s *Greedy* variantou. Teorie za tímto řešením je zde uvedena ve velké zkratce, popisující především principy, pro hlubší pochopení doporučuji přímo citovanou práci.

2.2.2 Pseudo-označování

S nápadem pseudo-označování vzorků přichází [54]. Kombinaci aktivního učení založeného na nejistotě modelu a konvoluční neuronové sítě doplňují o přidávání vzorků, u nichž si je model sám velmi jist jejich správným označením. Po každé trénovací iteraci model prohledá vstupní neoznačenou sadu a vzorkům, u nichž jeho spočtená míra jistoty přesáhne zvolenou hranici, přiřadí označení třídy, do které podle něj vzorek patří. Na takto modelem označených vzorcích je pak model následně trénován společně s tradičně aktivním učením budovanou trénovací sadou. Anotátorem označené, pro model nejisté, vzorky podstatně více zlepšují celkový výkon učícího se modelu, pseudo-označené vzorky pak pomáhají modelu spolehlivěji rozpoznávat význačné rysy.



Obrázek 2.3: Cyklus aktivního učení, doplněný o automatické pseudo-označování vzorků, u nichž si je model dostatečně jist jejich označením.

Konkrétní využité strategie aktivního učení v případě [54] jsou všechny tři odnože aktivního učení s pomocí nejistoty modelu, zmíněné v 2.1.2. Výběr vzorků pro pseudo-označování

je realizován pomocí entropie, použité k stanovení míry jistoty modelu (obdobně jako při vybírání nejistých vzorků pro anotaci). Vzorky s entropií nižší, než je hranice δ, obdrží pseudo-označení y_i:

$$j^* = \underset{j}{argmax}\, p(y_i = j | x_i; W)$$

$$y_i = \begin{cases} j^*, & en_i \leq \delta \\ \text{jinak neoznačovat} \end{cases}$$

Pro spolehlivost pseudo-označených vzorků je potřeba zvolit dostatečně velikou hodnotu hranice δ. S postupem učení modelu by pro statickou hranici zákonitě bylo vybíráno čím dál víc vzorků, o nichž by si byl model velmi jist jejich označením. Což ovšem dle [54] může vést k vyšší míře nepřesných pseudo-označení, navrhují proto postupem času snižující se hranici, díky níž zůstane zachována určitá úroveň potřebné jistoty modelu pro automatické označování. Po každé trénovací iteraci se tedy od hranice δ odečte tzv. *decay rate*.

Celkově lze tento přístup shrnout následujícím algoritmem:

Algorithm 2 Aktivní učení s pseudo-označováním

Input: neoznačené vzorky D^U, počáteční označené vzorky D^L, velikost skupiny nejistých vzorků pro dotázání k anotaci K, hranice pro výběr modelem si jistých vzorků δ, decay rate dr, počet iterací T

1: Inicializace parametrů $\mathcal{W}$ neuronové sítě pomocí D^L
2: **repeat**
3: Přidej K nově anotovaných vzorků do sady D^L pomocí zvolené strategie akt. učení
4: Získej vzorky, u nichž si je model velmi jist jejich označením, a přidej je do sady D^H
5: Trénuj model na $D^H \cup D^L$
6: Odečti od hranice δ decay rate dr
7: **until** není překročen počet iterací T

Output: parametry $\mathcal{W}$ neuronové sítě

S pomocí pseudo-označování se [54] povedlo získat nadmíru dobré výsledky i s užitím strategií aktivního učení, které nejsou považovány za příliš vhodné pro nasazení v dávkovém aktivním učení, a tedy pro kombinaci s konvolučními neuronovými sítěmi. Otevřená však zůstává otázka volby vhodné hranice δ pro různé druhy datasetů.

Kapitola 3

Implementace a experimenty

V praktické části své práce provádím experimenty v oboru klasifikace obrazu s různými strategiemi aktivního učení v kombinaci s konvolučními neuronovými sítěmi. Zkoumám jejich účinnost na rozličných datových sadách, efekt nepřesností anotátora a pozitivní efekt pseudo-označování při použití rozdílných parametrů.

3.1 Implementace

K provedení experimentů jsem stvořil v jazyce Python skript. Pro modelování a trénování konvoluční neuronové sítě využívám frameworku TensorFlow [20] a knihovny Keras [8]. Skript obsahuje všechny tři strategie aktivního učení pracující s nejistotou modelu (zmíněné v 2.1.2), náhodné vzorkování, strategii K-Center greedy (popsanou v 2.2.1), simulaci lidského anotátora s volitelnou chybovostí a pseudo-označování vzorků.

Skript je navržen tak, aby se dal jednoduše rozšířit o další strategie aktivního učení, či o další podpůrné metody podobného ražení, jako pseudo-označování. Lze pomocí vstupních parametrů ovlivnit počet trénovacích iterací modelu, velikost skupiny vzorků k anotaci, míru přesnosti anotátora i rozpočet aktivního učení, čili kolik neoznačených vzorků bude celkem přidáno po anotaci do trénovací sady. Podrobnější popis ovládání a struktury skriptu je k dispozici v příloze B.

3.2 Datasety

Pro experimenty jsem zvolil celkem čtyři datasety pro obrazovou klasifikaci, lišící se počtem tříd, obtížností a počtem dat. Úmyslem bylo mít dostatečně diverzní datasety pro zkoumání výkonu rozdílných strategií v rozdílných podmínkách. Přehled mnou zvolených datasetů poskytuje následující tabulka:

Dataset	Počet tříd	Velikost trénovací sady	Velikost testovací sady
Fashion MNIST	10	60 000	10 000
Cifar-10	10	100 000	20 000
Výběr z ImageNet	5	4 041	1 011
Výběr z VGGFace2	77	21 852	5 464

Fashion MNIST [58] dataset představuje obtížností jednoduchou sadu 28×28 grayscale obrázků s dostatečným počtem trénovacích i testovacích dat. Cifar-10 [27] je již obtížnější dataset 32×32 barevných obrázků, stále ovšem s rozumným počtem trénovacích a testovacích dat. Výběr z ImageNet je mnou vybraných pět tříd obrázků (lenochodů, masožravců, viaduktů, katedrál a květinových záhonů) z rozsáhlé databáze ImageNet [17], upravených na rozměry 48×48. Reprezentuje obtížný dataset jen několika tříd s nedostatkem dat, kdy na jednu třídu je dohromady pouze zhruba tisíc trénovacích a testovacích vzorků dohromady. Výběr z VGGFace2 je mnou vybraných sedmdesát sedm tříd obrázků z datasetu VGGFace2 [6], upravených na rozměry 64×64. Reprezentuje další obtížný dataset obsahující mnoho tříd a ne zcela ideální počet dat pro každou třídu.

Skript lze snadno rozšířit o další datasety, stačí je dodat ve formátu *.hdf5*, rozdělené na trénovací a testovací data a označení.

3.3 Modely

Na všech datasetech experimentuji se dvěma modely konvoluční neuronové sítě. Opět za účelem pozorování výkonu aktivního učení v širším spektru podmínek. Jeden z modelů je poněkud jednodušší síť, druhý naopak podstatně hlubší. Popis vrstev a jejich parametrů nabízí následující tabulky:

Typ vrstvy	Rozměry filtru / Střída	Počet filtrů / Neuronů
Konvoluční	3×3 / 1	64
Konvoluční	3×3 / 1	64
MaxPooling	2×2 / 2	
Dropout (25%)		
Konvoluční	3×3 / 1	96
Konvoluční	3×3 / 1	96
MaxPooling	2×2 / 2	
Dropout (25%)		
Plně propojená		128
Dropout (50%)		
Plně propojená		128
Dropout (50%)		
SoftMax		

První, jednodušší model konvoluční neuronové sítě.

Typ vrstvy	Rozměry filtru / Střída	Počet filtrů / Neuronů
Konvoluční	3×3 / 1	64
Konvoluční	3×3 / 1	64
Konvoluční	3×3 / 1	64
MaxPooling	2×2 / 2	
Dropout (25%)		
Konvoluční	3×3 / 1	96
Konvoluční	3×3 / 1	96
Konvoluční	3×3 / 1	96
MaxPooling	2×2 / 2	
Dropout (25%)		
Konvoluční	3×3 / 1	128
Konvoluční	3×3 / 1	128
Konvoluční	3×3 / 1	128
MaxPooling	2×2 / 2	
Dropout (25%)		
Plně propojená		512
Dropout (50%)		
Plně propojená		512
Dropout (50%)		
SoftMax		

Druhý, o poznání hlubší model konvoluční neuronové sítě.

Každou konvoluční a plně propojenou vrstvu následuje tzv. *batch-normalization* vrstva a PReLU aktivační vrstva, které jsou pro přehlednost z tabulky vynechány. Skript taktéž obsahuje funkci pro sestavení a kustomizaci konvolučních neuronových sítí, lze tedy snadno zmíněné modely rozšířit o další, i rozdílné struktury.

3.4 Experimenty

Pro experimenty jsem zvolil všechny tři varianty strategie aktivního učení pracující s nejistotou modelu. Byť jsou v dávkovém stylu aktivního učení často odsuzovány, v kombinaci například s pseudo-označováním dokážou fungovat nadmíru obstojně a i v původní podobě zůstávají stále často používané. Dále jako reprezentanta moderních strategií aktivního učení, šitých na míru pro práci s konvolučními neuronovými sítěmi, jsem zvolil metodu K-Center Greedy. Jako spodní a horní hranici pro ohodnocení výkonu jednotlivých strategií využívám náhodné vzorkování respektive pasivní učení, kdy je model trénován po obdobný počet trénovacích iterací nad celou sadou.

Všechny experimenty mají neomezený rozpočet, postupně se tedy dotazuje a značí celá neoznačená sada. Anotátor je skriptem simulován, jelikož zvolené datasety jsou celé označené, není tedy třeba lidského experta. Model začíná trénování s 10 % označených vzorků z celé sady. Zbytek označení je mu skryt a dotazováním mu je simulovaný anotátor odkrývá, vzorky se pak přidávají do trénovací sady. Po každém přidání vzorků následuje trénovací epocha nad rostoucí trénovací sadou, model se tedy učí kontinuálně, nedochází v každé iteraci k trénování od nuly.

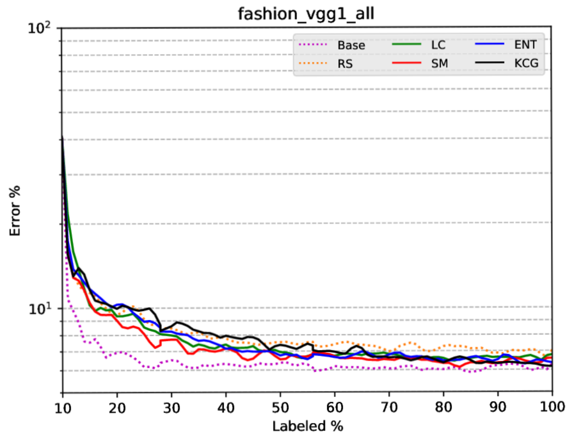
3.4.1 Experiment 1

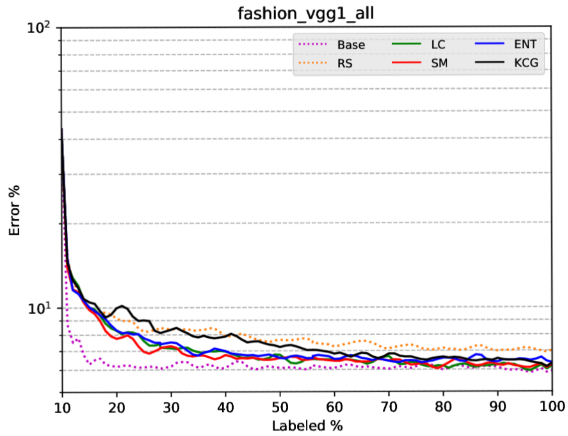
V prvním experimentu zkoumám efektivitu jednotlivých strategií. Jejich chování při kontinuálním trénovaní modelu, schopnost vybírat informativní a užitečné vzorky po celou dobu trénování, odolnost vůči přetrénování a rozdíly při větší dávce přidávaných vzorků v každé iteraci.

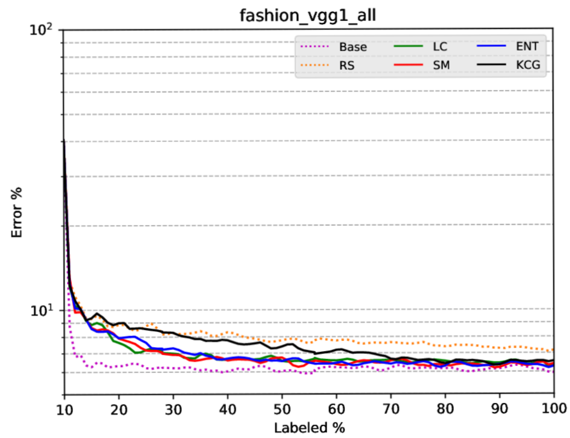
Za tímto účelem nechávám aktivní učení běžet ve čtyřech scénářích, lišících se počtem trénovacích epoch a velikostí dávky přidávaných vzorků. V prvním scénáři v každé iteraci probíhá jedna trénovací epocha a přidává se 1 % z celkové sady vzorků jako nově označené vzorky. Jelikož rozpočet algoritmu je neomezený a jelikož učení začíná s 10 % označených vzorků, proběhne devadesát iterací a trénovacích epoch, než bude označen každý vzorek sady. V dalších dvou scénářích se stále přidává po 1 %, ale trénovacích epoch v každé iteraci přibývá na dvě respektive čtyři. Ve finálním scénáři probíhá zase pouze jedna trénovací epocha v každé iteraci, vzorky jsou však přidávány po 2 %.

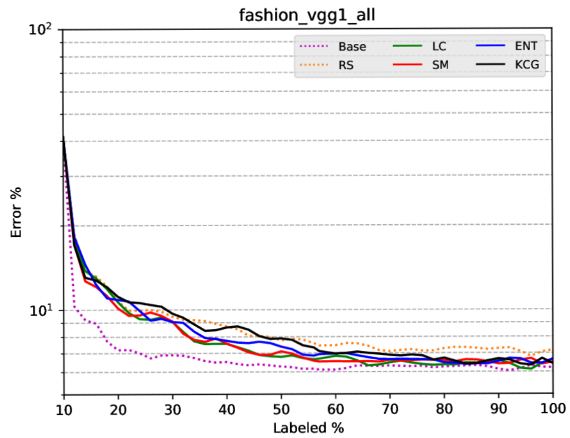
Na následujících stranách prezentuji výsledky experimentů v podobě grafů, za nimi pak následuje shrnutí pozorovaných výsledků a zjištěných poznatků. Grafy jsou rozděleny do čtveřic, dle čtyř zmíněných scénářů, na každé straně jsou pak dvě tyto čtveřice, seskupeny dle použitého datasetu a modelu, což umožňuje přehledné porovnání. V každé čtveřici jsou výsledky scénářů uskupeny zleva doprava, shora dolů, v pořadí: jedna trénovací epocha v iteraci a přidávání vzorků po 1 %; dvě trénovací epochy v iteraci; čtyři trénovací epochy v iteraci; jedna trénovací epocha a přidávání vzorků po 2 %.

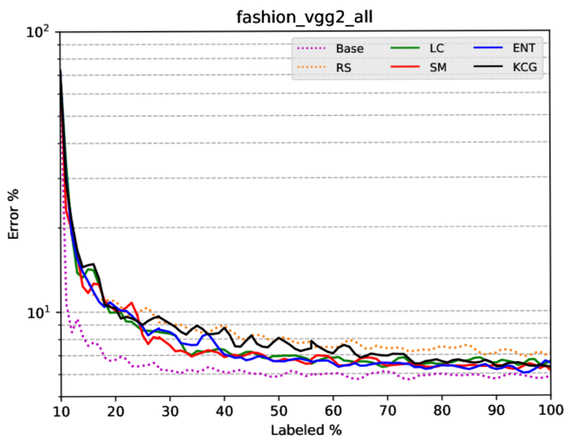
Nadpisy grafů značí použitý dataset a model. Z legendy pak *Base* a *RS* znamenají pasivní učení nad celou sadou vzorků, respektive náhodné vzorkování. Představují jisté hranice, jimiž lze strategie aktivního učení hodnotit. Ty by měly překonat náhodné vzorkování a ideálně se blížit hranici dané pasivním učením. *LC*, *SM* a *ENT* jsou tradiční strategie aktivního učení, popořadě „Least confident“ (nejnižší jistota), „Smallest margin“ či „Margin sampling“ (nejmenší rozdíl) a „Entropy“, popsané v tomto pořadí v kapitole 2.1.2. *KCG* pak znamená metodu K-Center Greedy, popsanou v 2.2.1. Osy grafů značí chybovost modelu v % na logaritmické stupnici a počet označených vzorků z celé vstupní sady, taktéž v %.

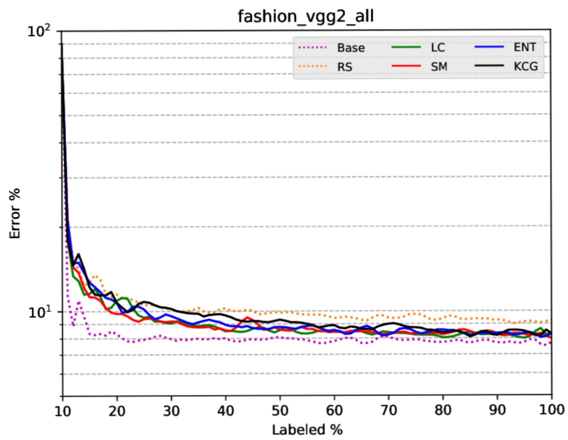

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

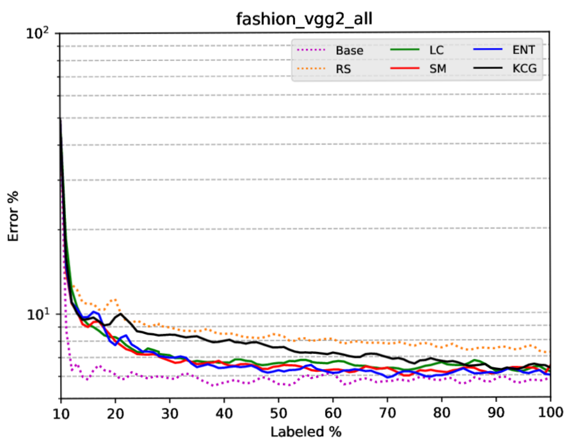

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

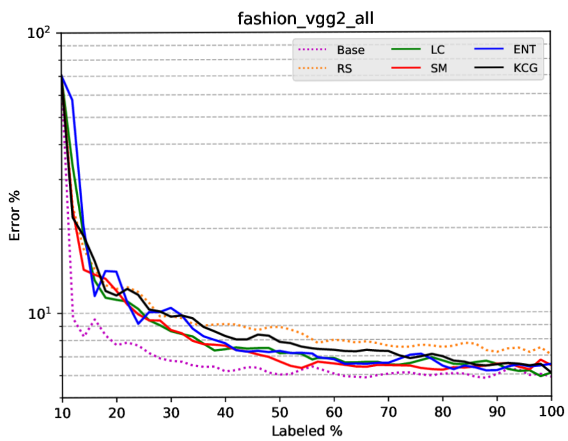

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

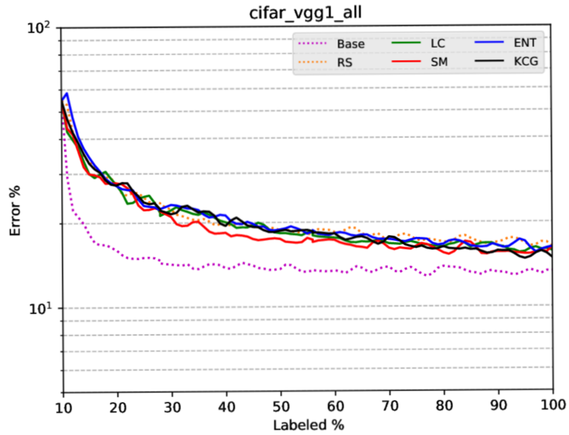

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

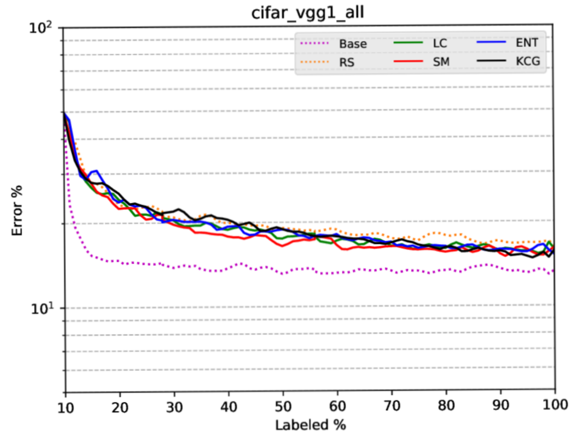

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

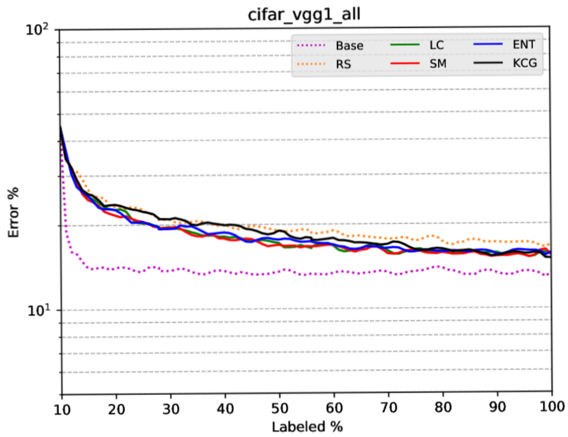

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

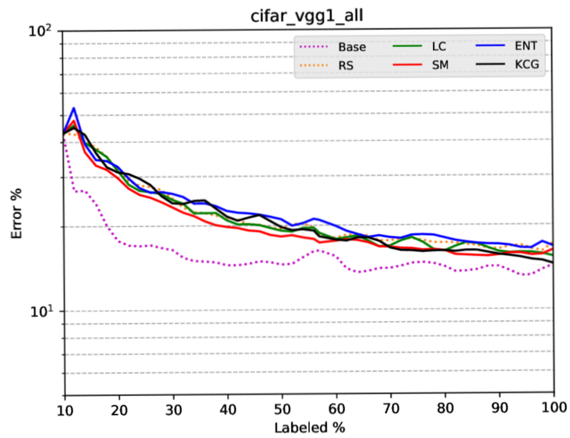

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

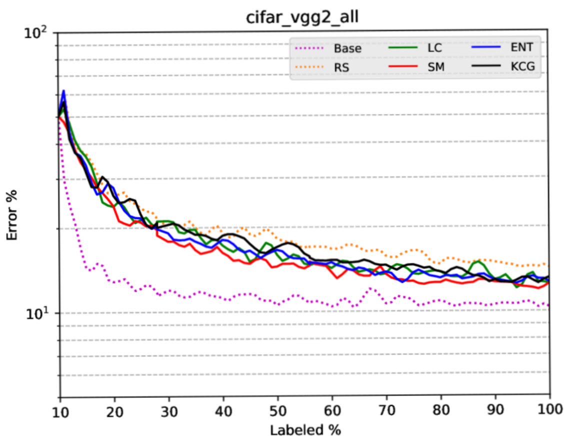

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

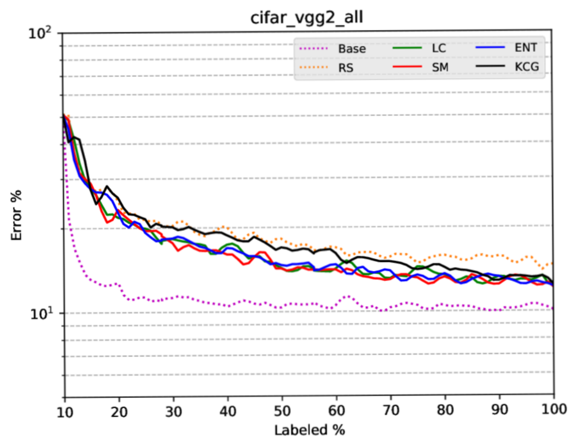

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

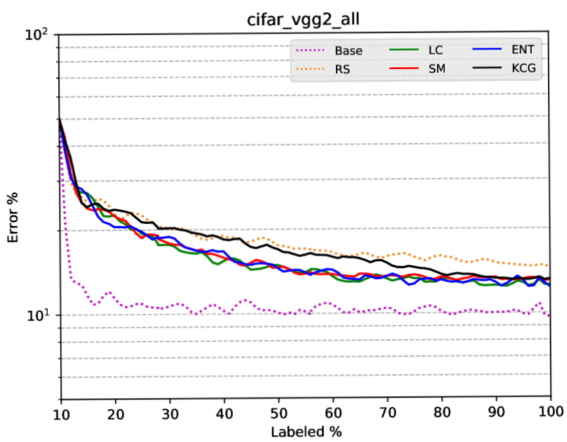

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

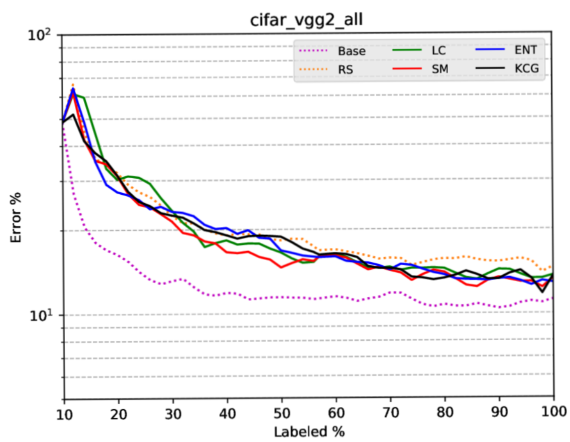

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

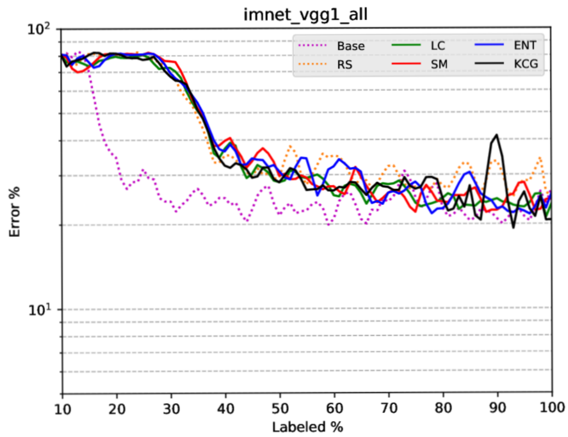

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

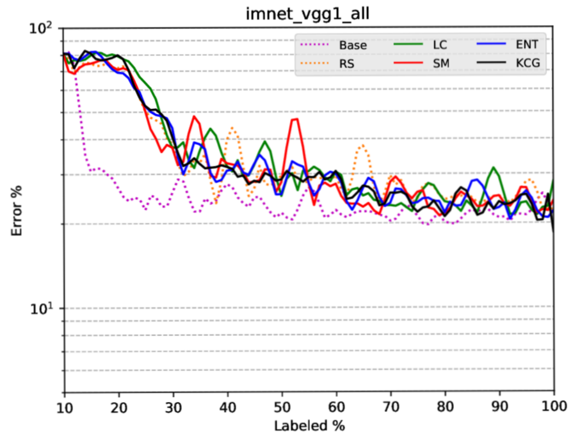

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

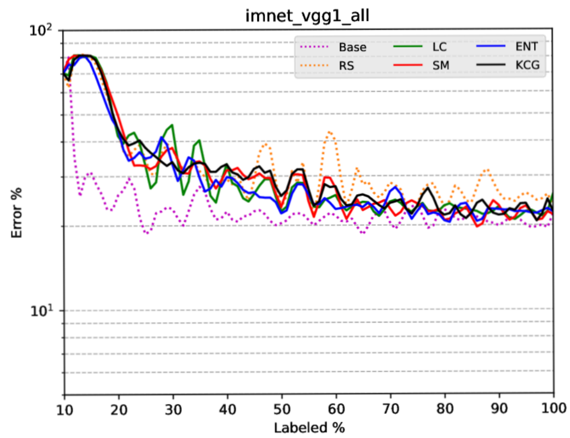

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

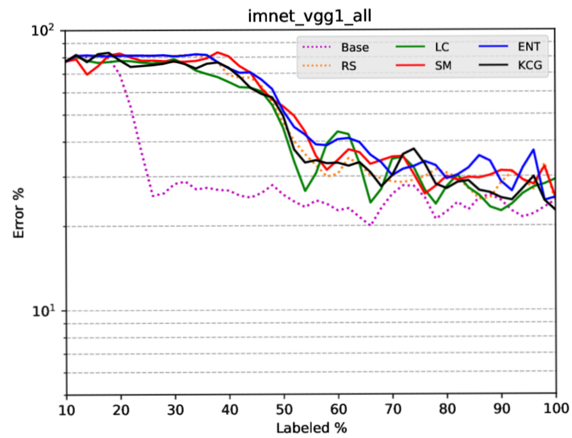

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

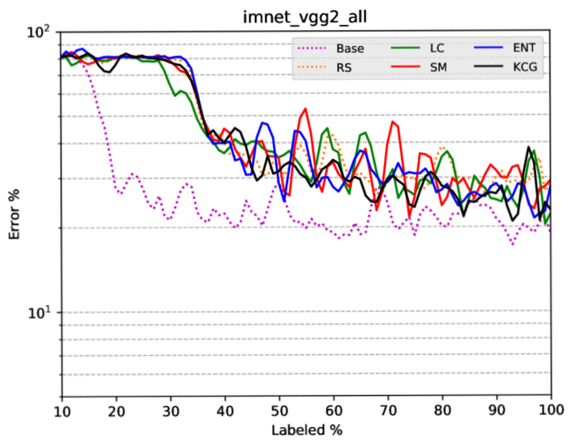

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

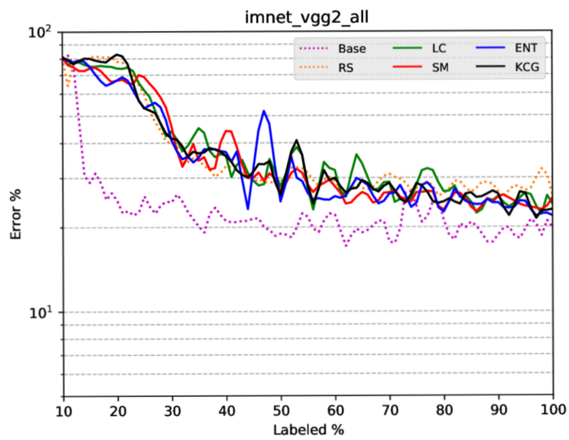

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

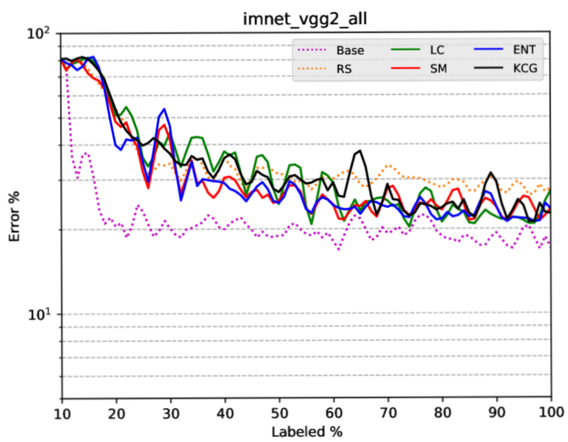

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

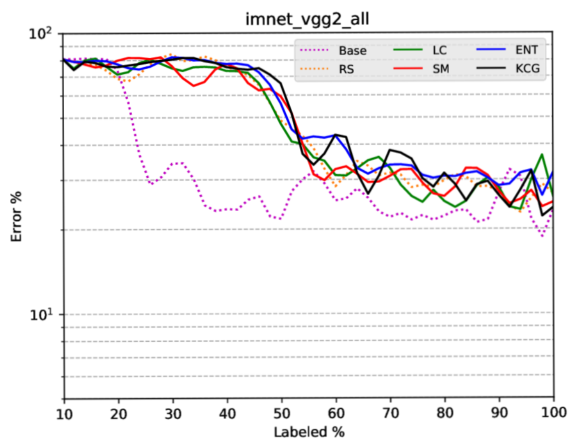

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

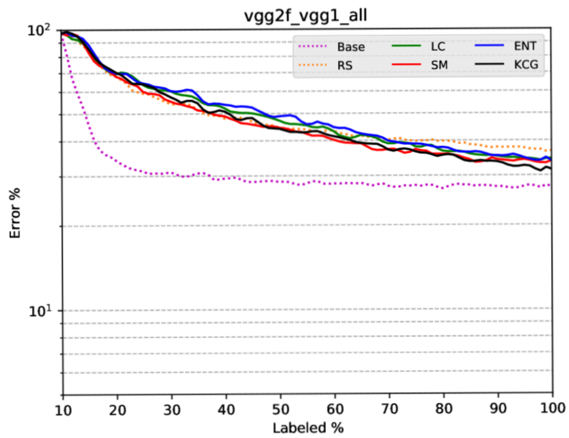

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

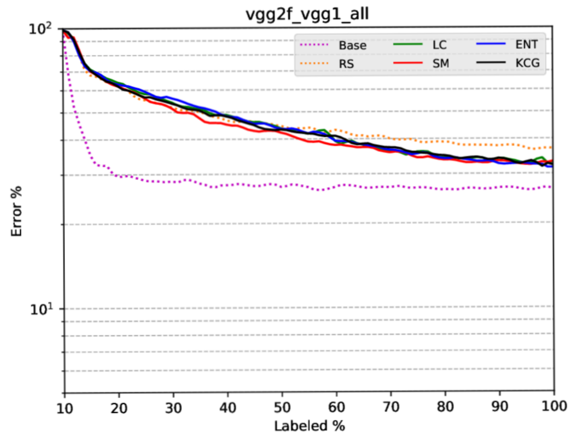

imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

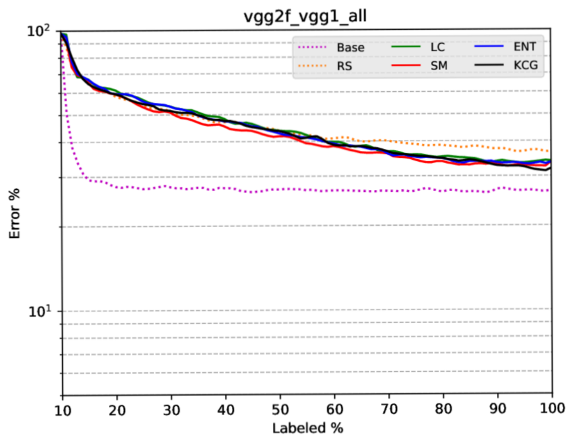

imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

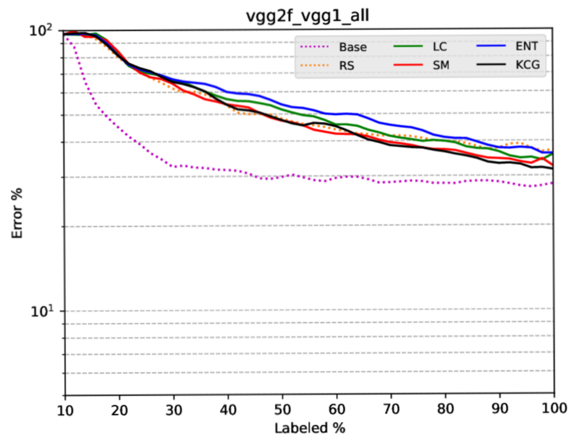

imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

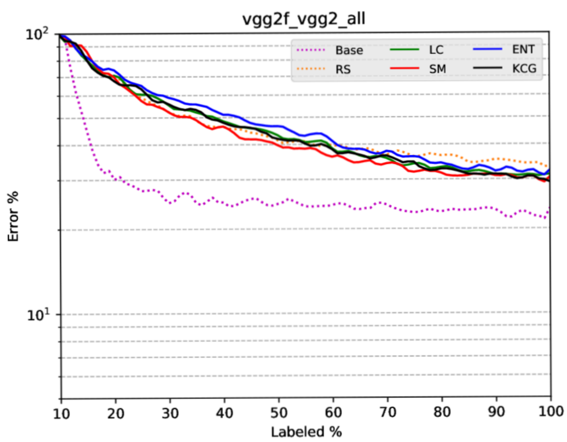

imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

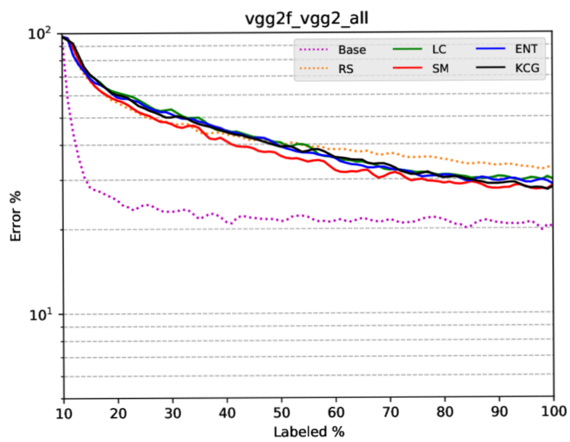

vgg2f_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

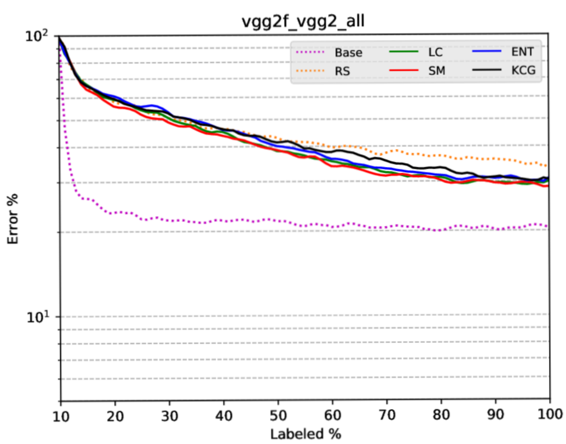

vgg2f_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

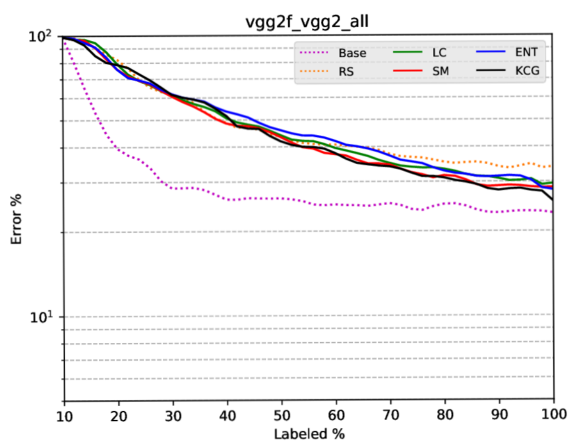

vgg2f_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

Dílčí závěry

Všechny metody jsou schopny ve všech scénářích překonat náhodné vzorkování, což může být v případě tradičních metod aktivního učení mírné překvapení, nejsou totiž dle mínění mnohých zcela vhodné pro dávkové aktivní učení. Na datasetu Fashion MNIST dokonce dokáží tyto metody zpočátku porážet metodu KCG. Ta vykazuje lepší výkon na složitějších datasetech, jako je VGGFace2 a vypadá mnohem lépe ve scénáři, kdy se přidává větší dávka vzorků v každé iteraci. Naopak ve scénářích s více trénovacími epochami o poznání pokulhává, dochází zřejmě k přetrénování.

Co se srovnání tradičních strategií aktivního učení týče, i přes rozdílné principy dokážou být velmi vyrovnané, jistý náskok si však častěji drží metoda SM, jež je schopná od začátku lépe dotazovat informativní vzorky. V průběhu učení se však většinou tyto tři metody jsou schopny srovnat. Na jednodušších datasetech, obzvlášť ve scénářích s více trénovacími epochami v jedné iteraci, jsou nesmírně vyrovnané. Na těžších datasetech však metoda SM získává větší náskok a je schopna po celou dobu konkurovat metodě KCG. Metody LC a ENT pak začínají pokulhávat a z těchto dvou se jeví jako o něco málo lepší metoda LC.

Za povšimnutí stojí naopak nezávislost výkonu metody na zvoleném modelu konvoluční neuronové sítě. Na jednom datasetu si zpravidla nezávisle na modelu strategie drží mezi sebou obdobné rozdíly.

V případě kontinuálního učení modelu platí, že čím složitější dataset, tím hůř se model trénovaný aktivním učením přibližuje výkonu modelu trénovaného pasivně. Řešením by mohl být algoritmus průběžně měnící počet trénovacích epoch a velikost dávky přidávaných vzorků, například v závislosti na dosaženém rozsahu trénovací sady a poměru přetrénování.

U datasetů lze dále pozorovat, že při dosažení určitého % velikosti trénovací sady, složené z postupně označovaných vzorků, dochází ke střetu všech strategií a následné stagnaci. Bylo by tedy v případě kontinuálního učení modelu vhodné uvažovat nad zastavujícími podmínkami, aby nedocházelo k plýtvání zdrojů.

Z výsledků datasetu výběru z IMNET lze vyčíst pouze to, že v případě příliš malého celkového počtu vzorků pro efektivní trénování konvoluční neuronové sítě se chovají všechny strategie srovnatelně nestabilně.

Shrnutí

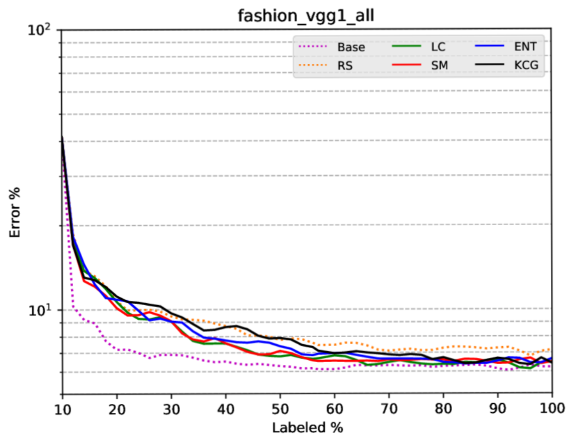
Obecně tedy lze říci, že volba strategie aktivního učení souvisí s obtížností a kvalitou datasetu. V případě jednoduchých a kvalitních datasetů lze použít i tradičnější strategie aktivního učení, především metoda SM prokazuje schopnost vybírat vhodné vzorky po celou dobu průběhu učení, rozdíly zde však nejsou tolik patrné. Podstatně výpočetně náročnější metoda KCG se hodí na složitější datasety a do scénářů, kdy lze přidávat velké množství vzorků. I zde však dokáže jednodušší metoda SM podat kvalitní výkon. U jednodušších datasetů taktéž stojí za to vzít do úvahy ukončující podmínku kontinuálního aktivního učení, jelikož ke stagnaci dochází při poměrně nízkých procentech označených vzorků z celého datasetu. U složitějších datasetů je pak potřeba hledat rovnováhu mezi počtem trénovacích epoch v iteraci a velikostí přidávané skupiny vzorků.

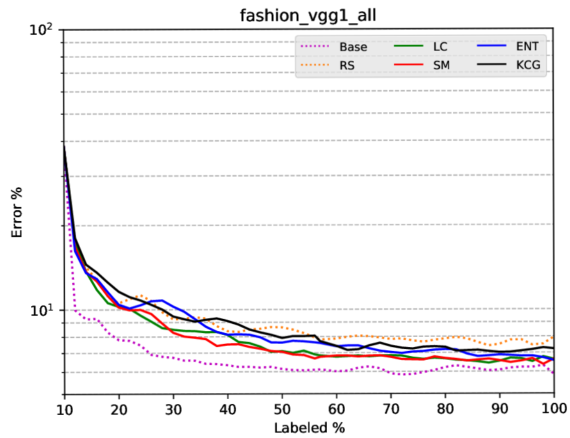
U většiny datasetů lze i kontinuální formou aktivního učení s vhodnou strategií redukovat úsilí vynaložené anotaci vzorků a dosáhnout solidních výsledků klasifikátoru.

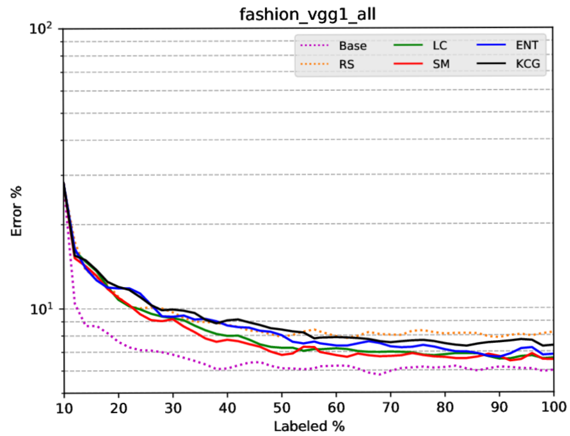
3.4.2 Experiment 2

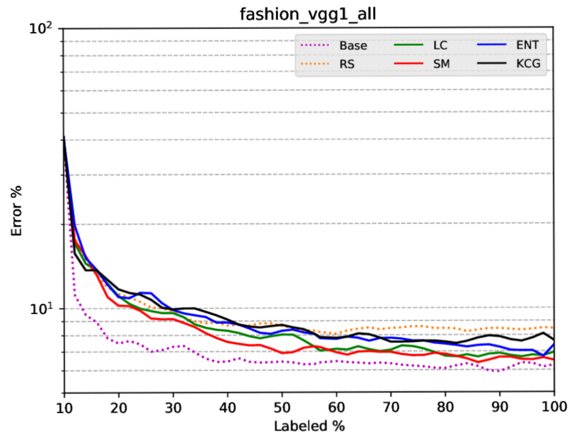
Druhým experimentem chci pozorovat odolnost jednotlivých strategií vůči omylnému anotátorovi. Anotace jsou simulovány jako v předchozím případě, tentokrát však ve třech scénářích má anotátor pravděpodobnost správného označení pouze 95, 90 a 85 %. V praxi často kvalitní anotace prezentují značnou finanční investici. Je jasné, že od určité úrovně chybovosti by bylo probíhající učení až příliš poznamenáno, zvolil jsem proto poměrně v praxi běžné a rozumné úrovně. V každé iteraci jsou přidána 2 % z celé sady vzorků a probíhá 1 trénovací epocha.

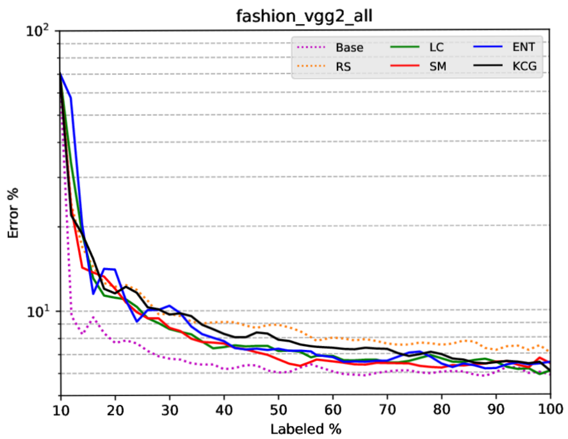
Následují opět grafy dosažených výsledků, v obdobném formátu jako v předchozím experimentu. Čtveřice zde tvoří zleva doprava, shora dolů: scénář s neomylným anotátorem pro srovnání; 95 % pravděpodobnost správného označení vzorku; 90 % pravděpodobnost správného označení vzorku; 85 % pravděpodobnost správného označení vzorku.

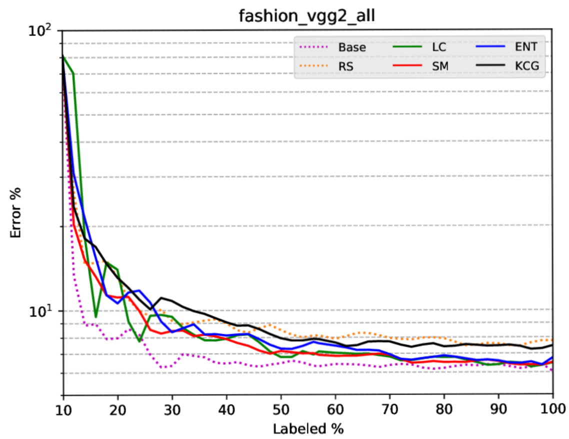

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

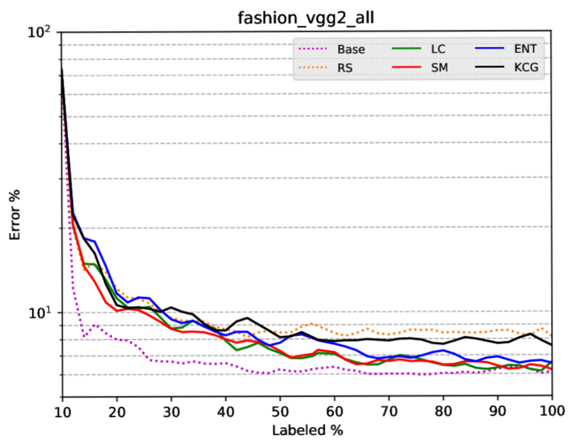

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

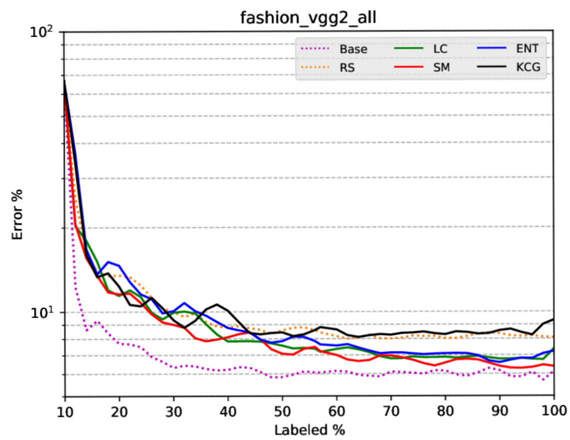

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

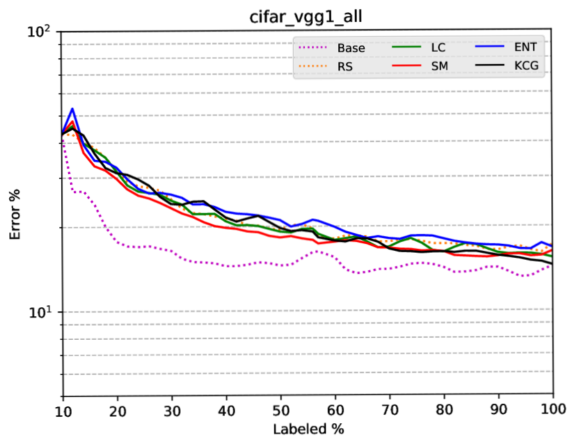

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

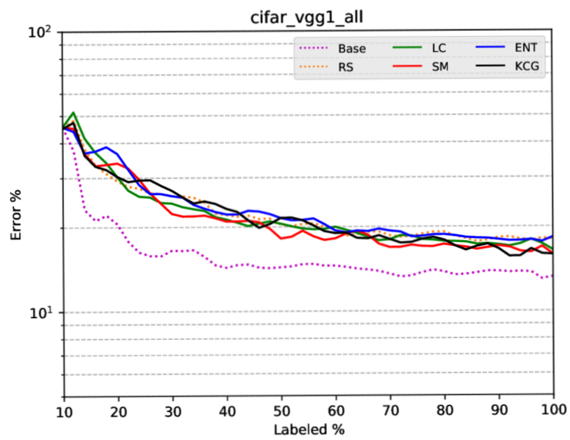

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

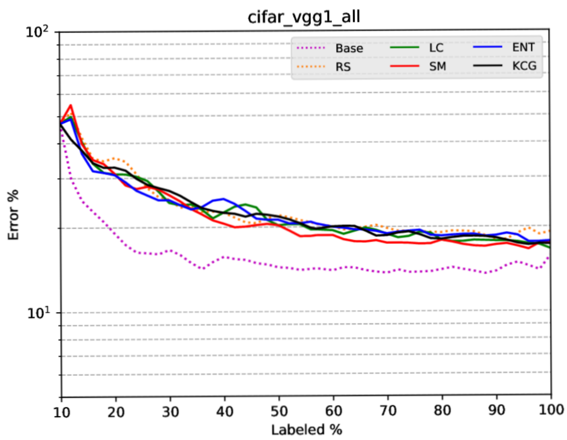

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

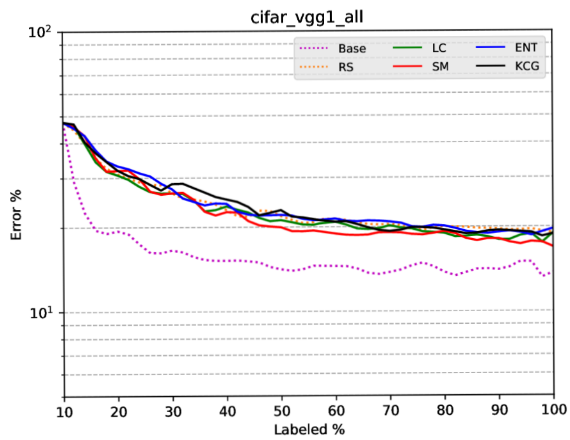

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

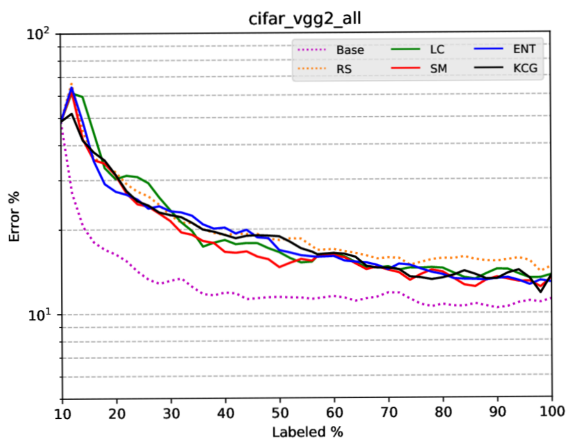

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

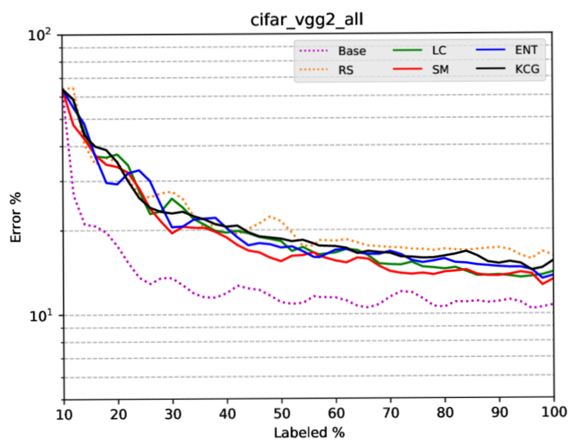

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

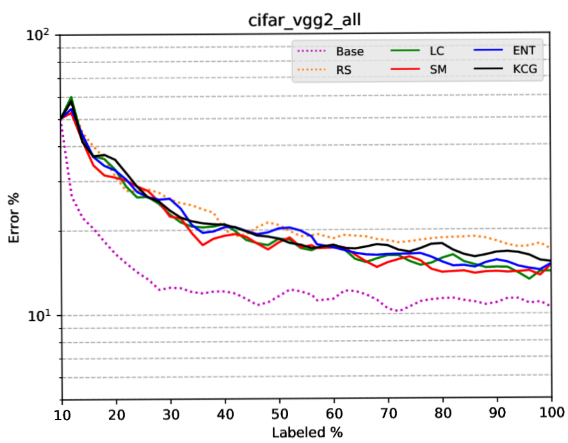

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

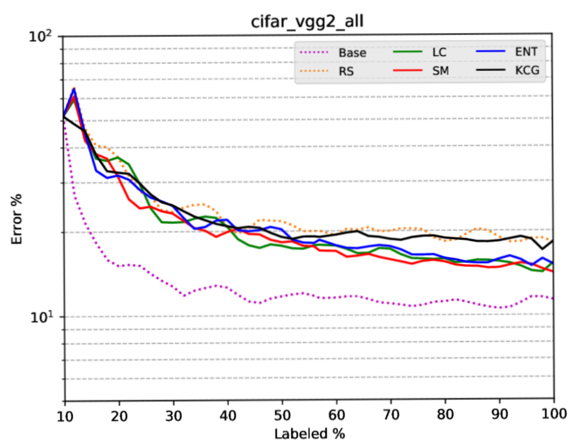

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

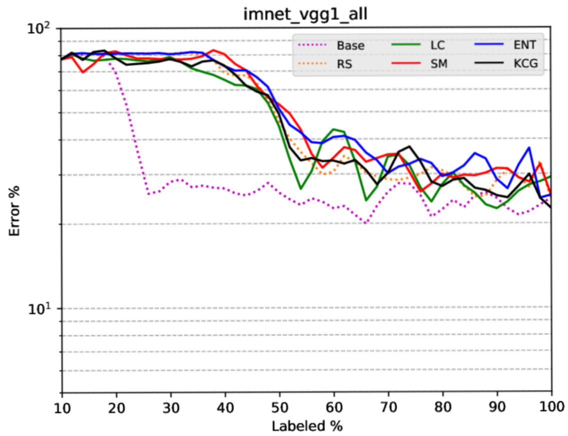

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

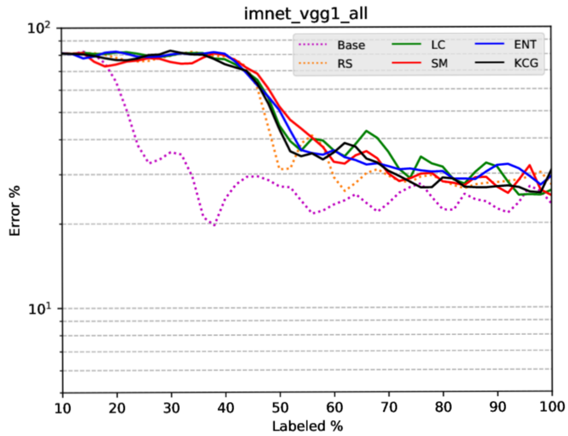

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

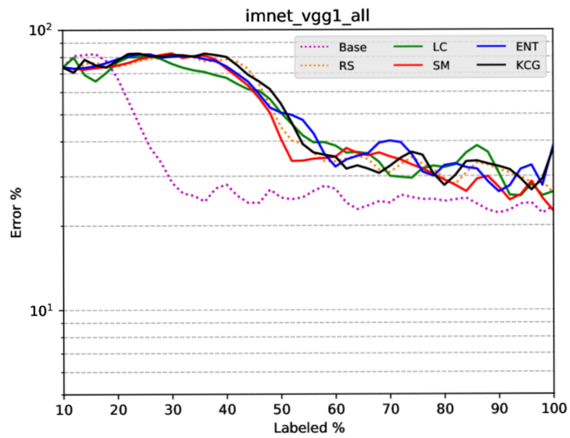

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

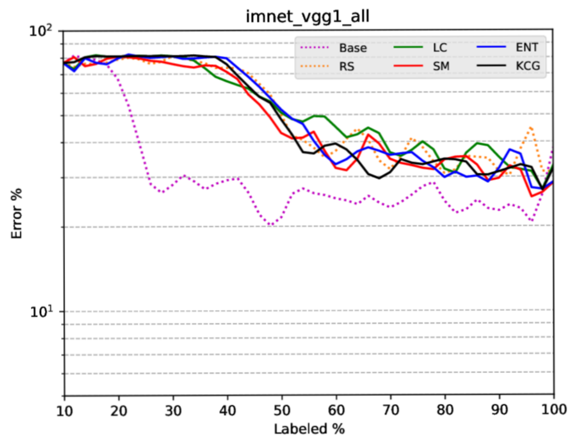

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

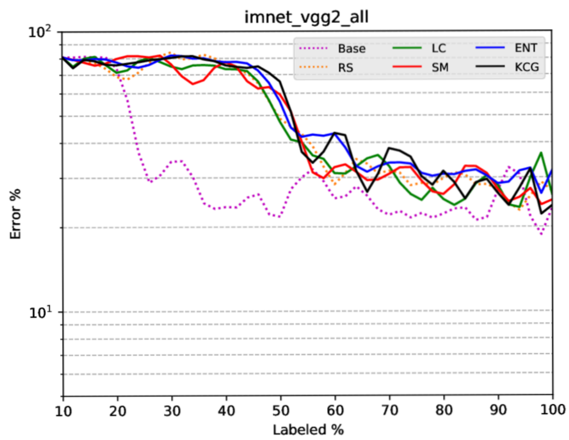

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

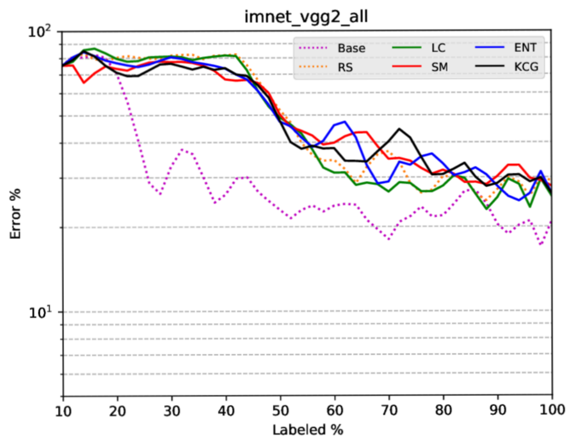

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

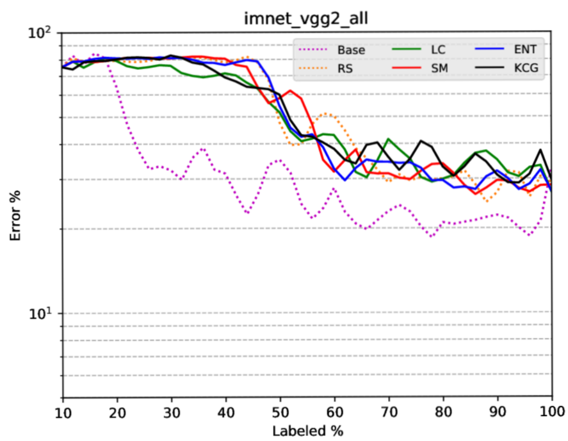

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

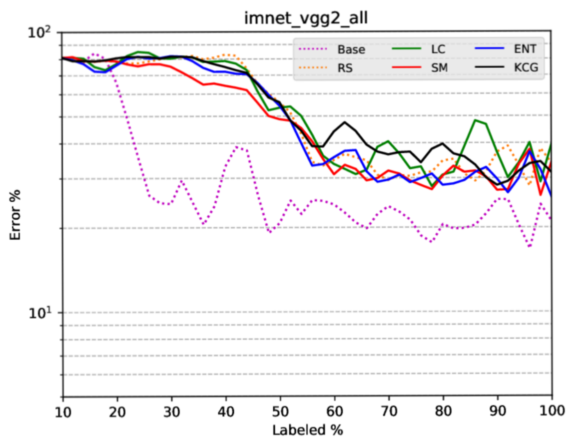

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

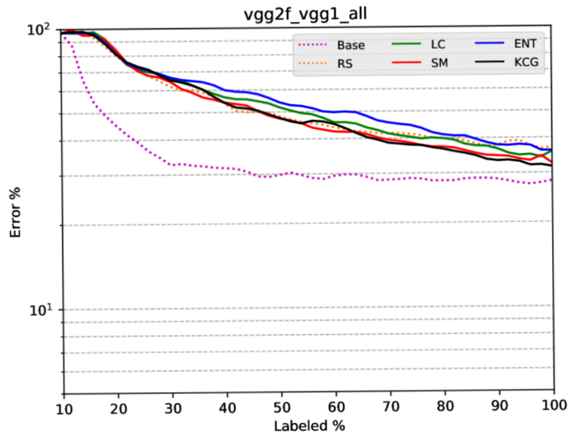

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

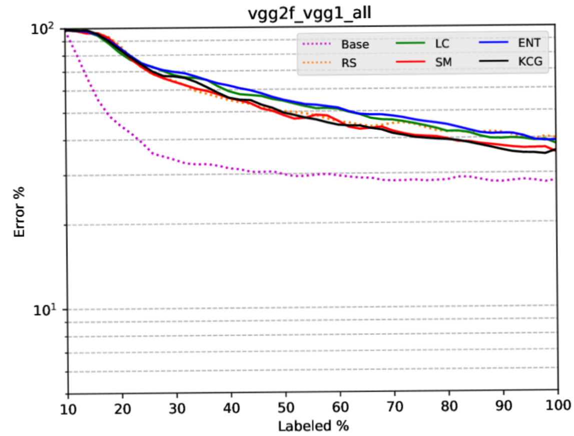

imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

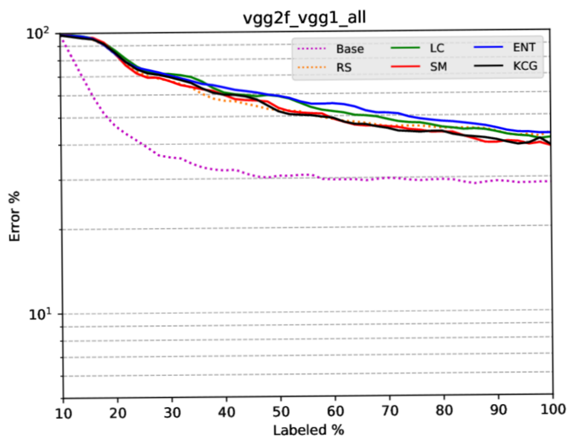

imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

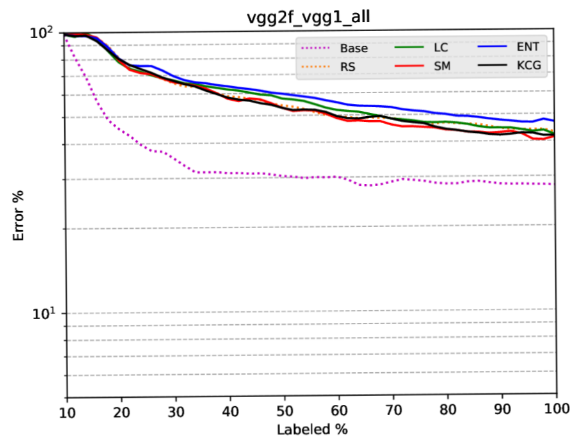

imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

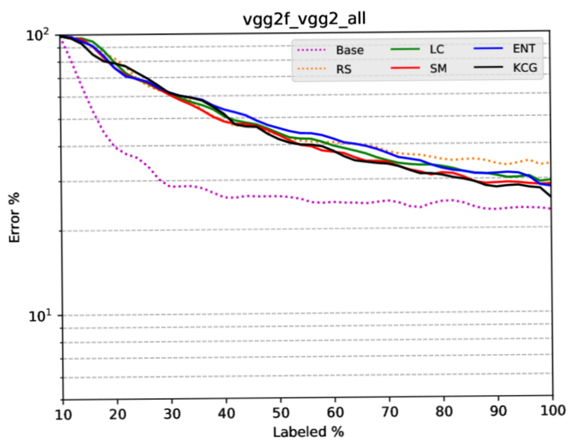

imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

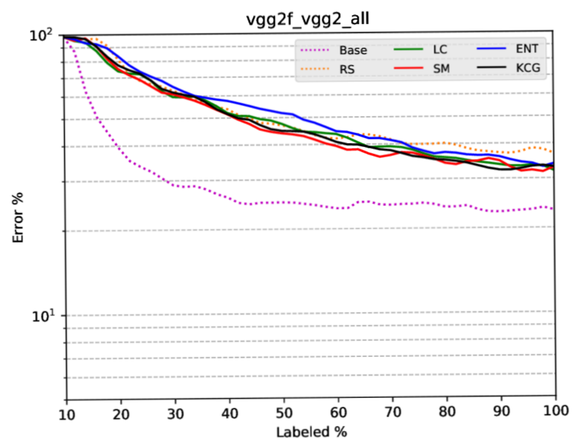

vgg2f_vgg1_all
Error %
Labeled %
Base
RS
LC
SM
ENT
KCG

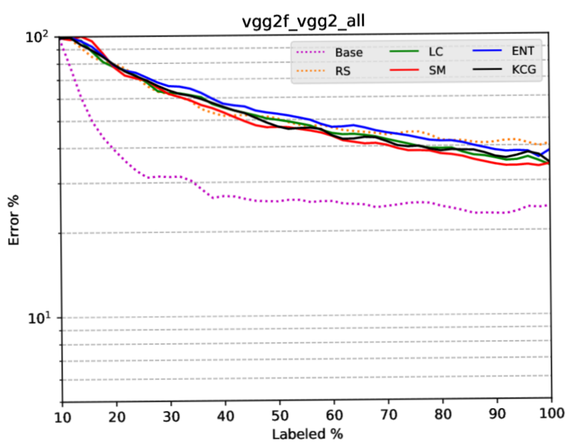

vgg2f_vgg1_all
Error %
Labeled %
Base
RS
LC
SM
ENT
KCG

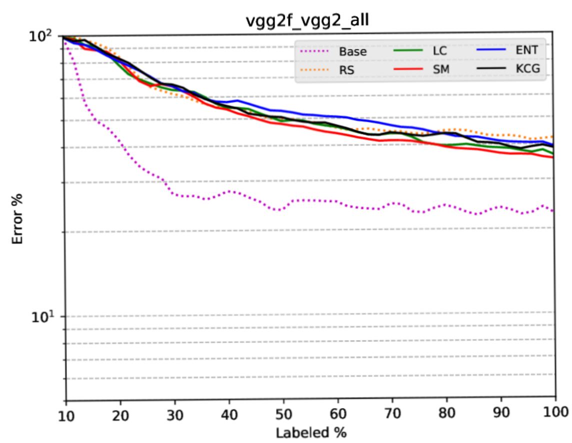

vgg2f_vgg1_all
Error %
Labeled %
Base
RS
LC
SM
ENT
KCG


vgg2f_vgg1_all
Error %
Labeled %
Base
RS
LC
SM
ENT
KCG


vgg2f_vgg2_all
Error %
Labeled %
Base
RS
LC
SM
ENT
KCG


vgg2f_vgg2_all
Error %
Labeled %
Base
RS
LC
SM
ENT
KCG


vgg2f_vgg2_all
Error %
Labeled %
Base
RS
LC
SM
ENT
KCG


vgg2f_vgg2_all
Error %
Labeled %
Base
RS
LC
SM
ENT
KCG

Dílčí závěry

Na jednodušších datasetech lze pozorovat, že i rozdíl 15 % v spolehlivosti anotátora se nepromítne do drastických celkových rozdílů ve výkonu většiny z jednotlivých strategií. Lze však stále pozorovat uniformní zhoršení v prostředních fázích, kdy je označeno 40 až 50 % všech vzorků. V náročnějších datasetech jsou však vlivy chyb anotátora velmi patrné, a to už od pouhé 90 % spolehlivosti označení.

Z principu metody SM lze u ní pozorovat horší úvodní fáze aktivního učení, než při neomylném anotátorovi. Postupně však opět zaujímá vedoucí postavení oproti metodám LC a ENT, které jsou na nepřesnosti náchylnější. Dle očekávání zdaleka nejvíce trpí na chyby anotace metoda KCG. Při neomylném označování vzorků dokázala ve většině datasetů buď vést, nebo po pomalejším startu dohnat i předehnat ostatní metody. Při 95 a 90 % spolehlivosti anotátora dokáže ještě povětšinou držet krok, a na složitějších datasetech si držet svou příčku v popředí, při nižší spolehlivosti se však hluboce propadá.

Zajímavou roli zde hraje i kvalita modelu, kdy při použití hlubší sítě trpí metoda KCG na klesající spolehlivost anotátora mnohem více, a to i na jednoduchých datasetech. Na ostatní metody nemá kvalita modelu zdaleka takový vliv.

Shrnutí

Rozdíly v celkovém výkonu strategií aktivního učení nejsou ve většině případů tak patrné. Zdaleka nejvíce na nepřesnosti anotace trpí metoda KCG. Z tradičních strategií aktivního učení nejlépe chybám odolává metoda SM, i když mívá pomalejší start. Metoda LC s ní dokáže mnohdy držet krok, není však tak vhodná pro obtížnější datasety. Metoda ENT za nimi mírně pokulhává.

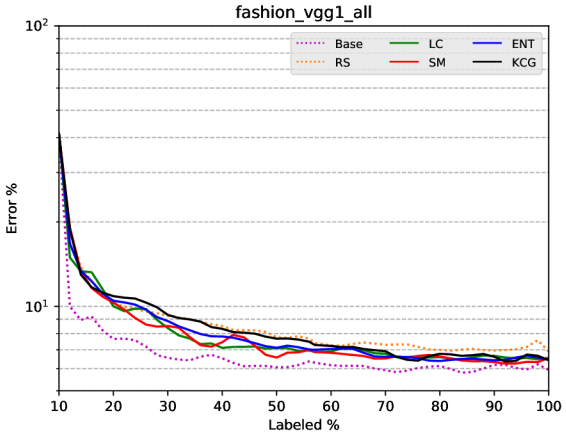
Zajímavé se mohou prokázat experimenty, od jaké chybovosti anotátora se začnou projevovat mnohem drastičtější úpadky efektivity všech strategií, nebo vliv možnosti požádat o znovuoznačení nejspíše mylně označeného vzorku.

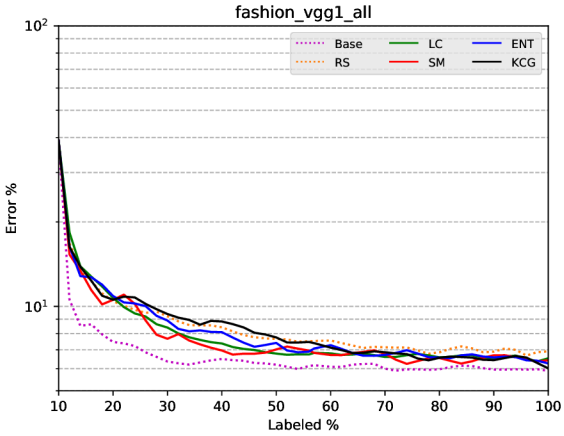
3.4.3 Experiment 3

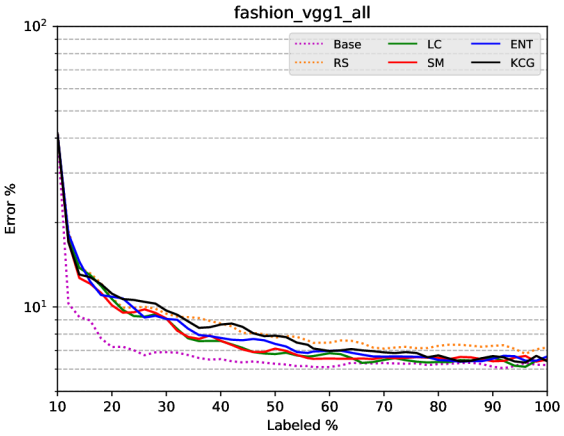
Ve třetím experimentu zkoumám vliv pseudo-označování na efektivitu strategií aktivního učení. Využívání vzorků v podstatě zadarmo pro zlepšení učení modelu je lákavá možnost, zatím však nevyzkoušená na širším spektru modelů a datasetů.

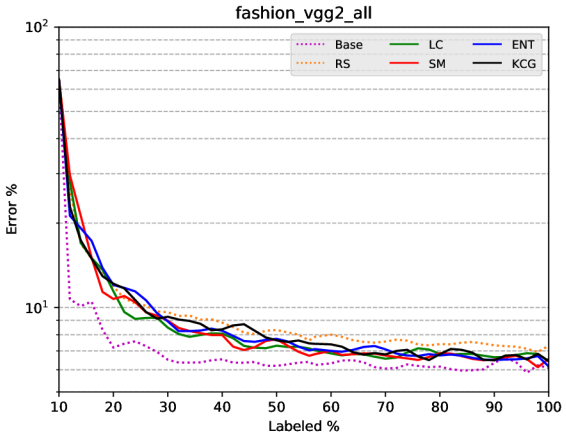
Pro určení vzorků, jejichž označením si je model jist a tím pádem se přidávají dočasně do trénovací sady, jsem zvolil entropii, obdobně jako v práci [54], ze které tento přístup vůbec čerpám. Experimentuji na dvou scénářích. V jednom je hranice pro určení důvěryhodného vzorku $\delta = 0.0005$ a decay rate $dr = 0.000033$, ve druhém $\delta = 0.005$ a decay rate pak $dr = 0.00033$. V jedné iteraci probíhá jedna trénovací epocha a vzorky jsou přidávány po 2 %.

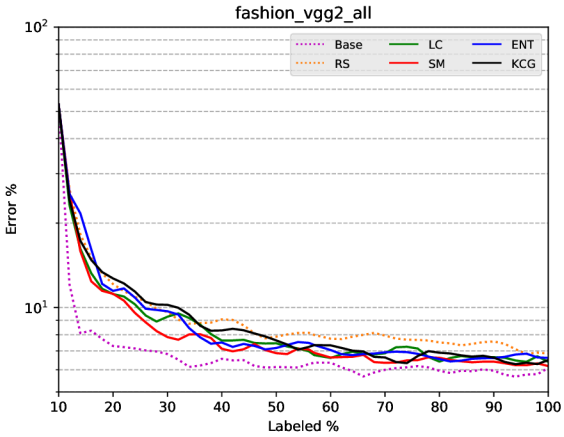
Následují opět grafy výsledků, tentokrát však po trojicích, kdy ve vrchním řádku jsou zmíněné scénáře v pořadí stejném, jako jsou představeny, a trojici uzavírá vespod pro porovnání scénář s obdobnými parametry, ale bez pseudo-označování.

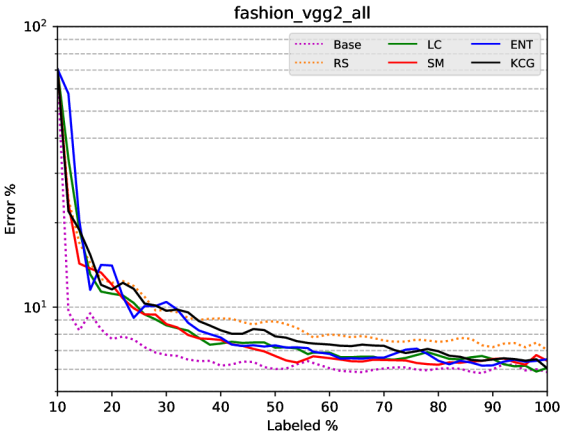

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

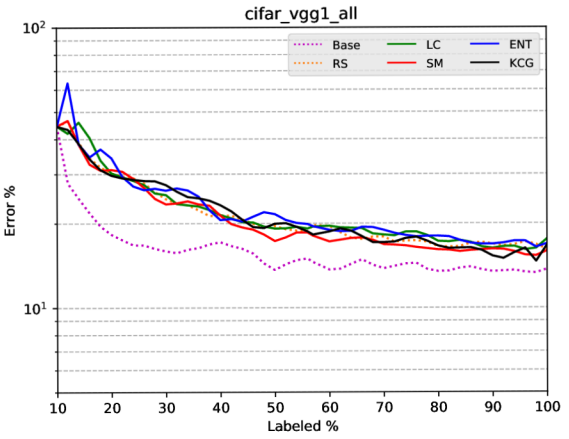

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

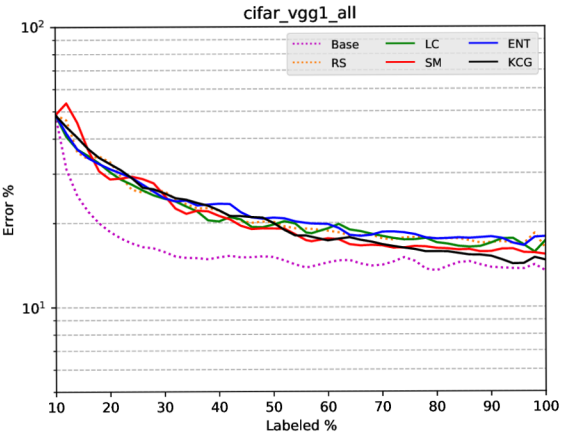

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

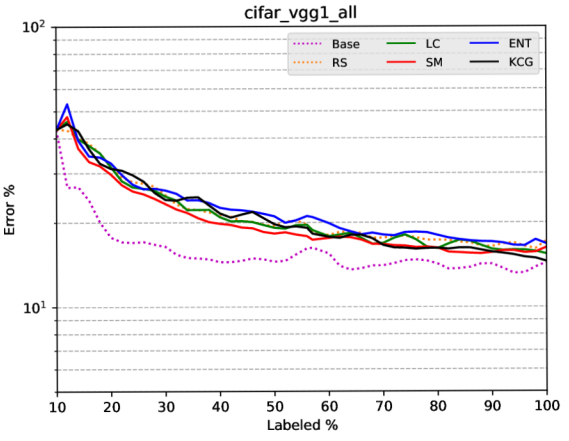

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

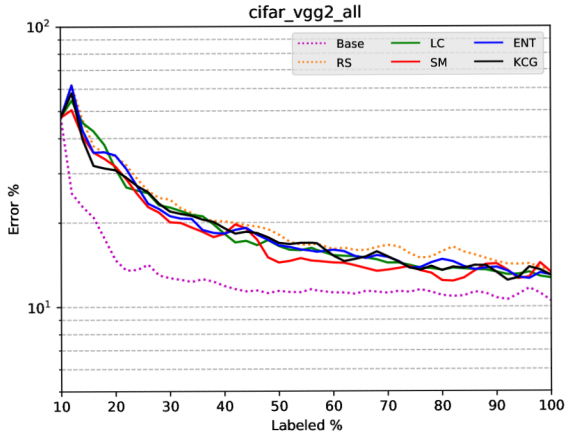

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

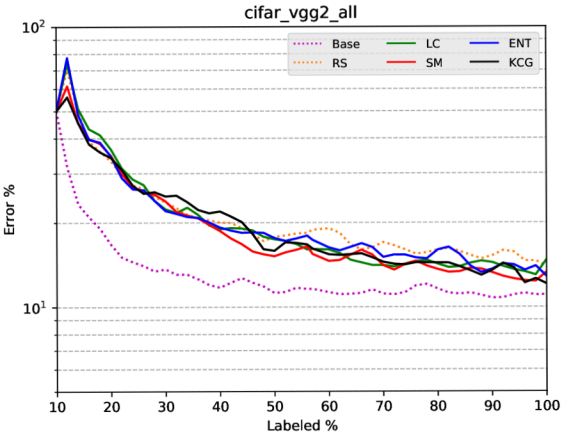

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

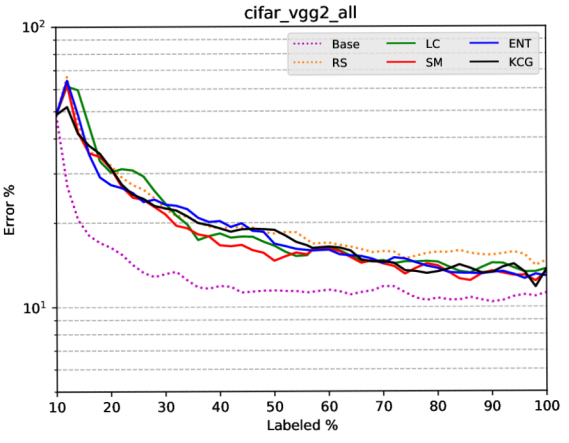

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

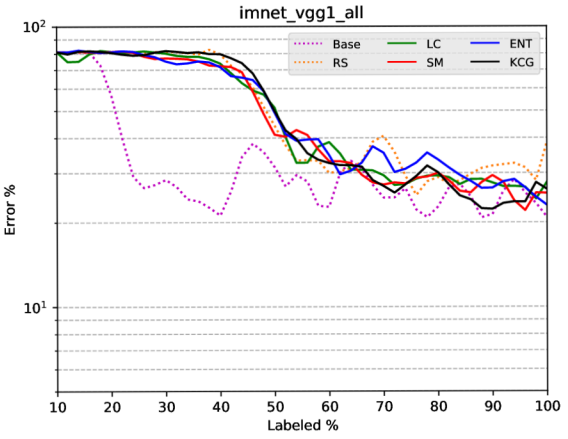

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

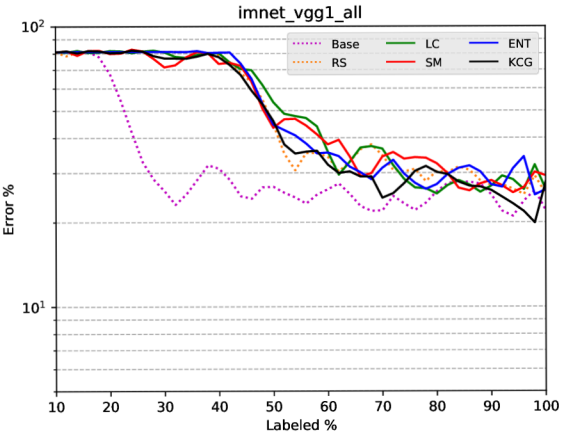

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

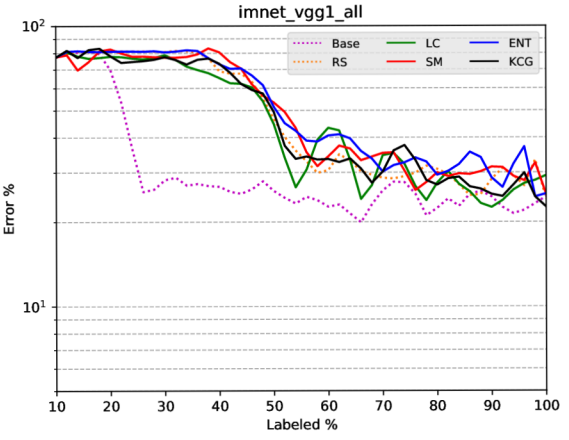

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

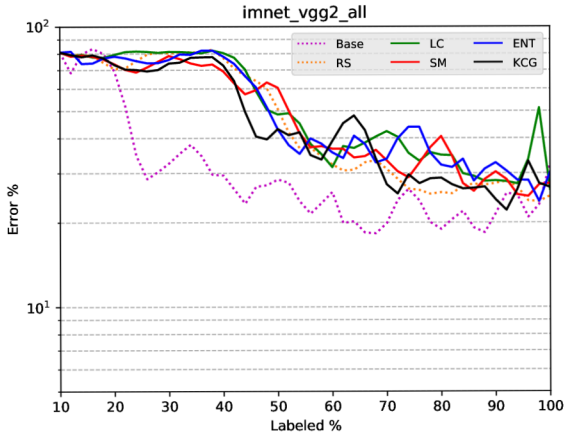

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

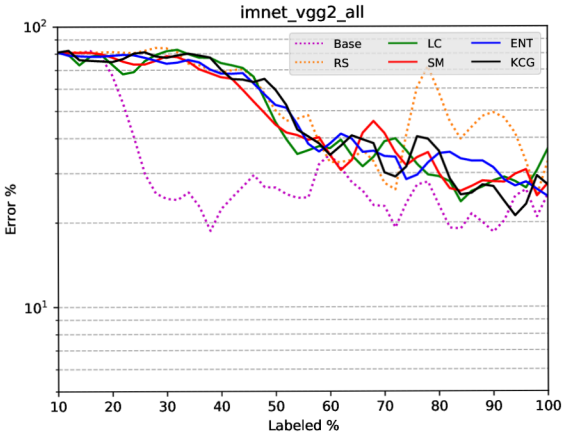

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

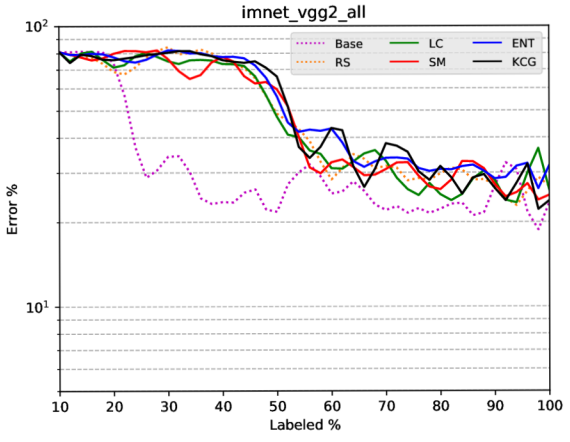

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

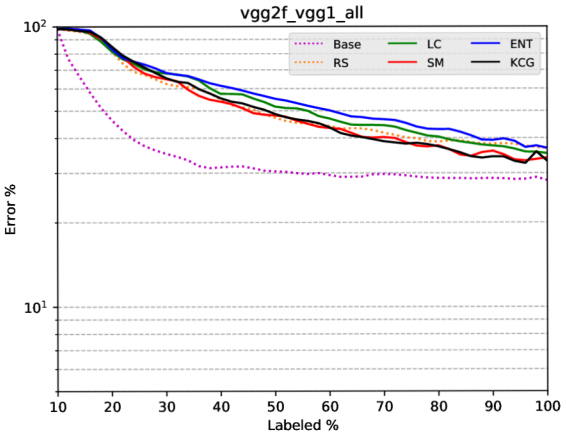

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

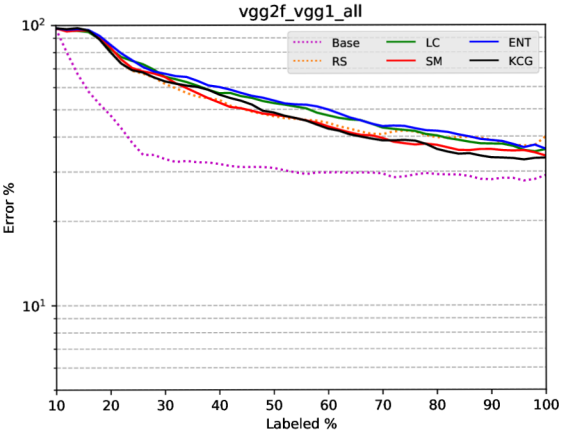

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

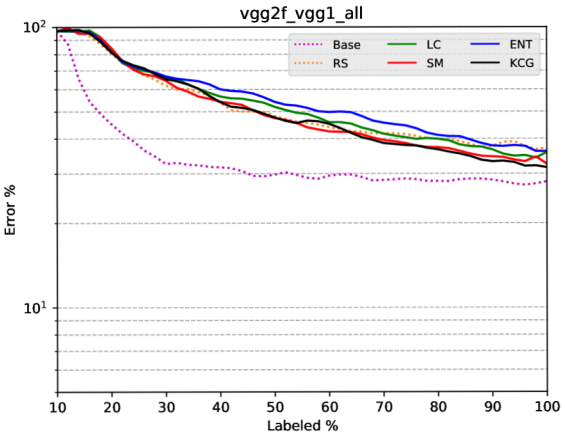

imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

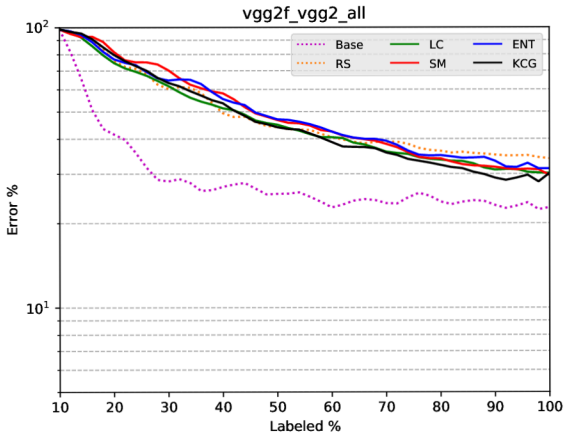

imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

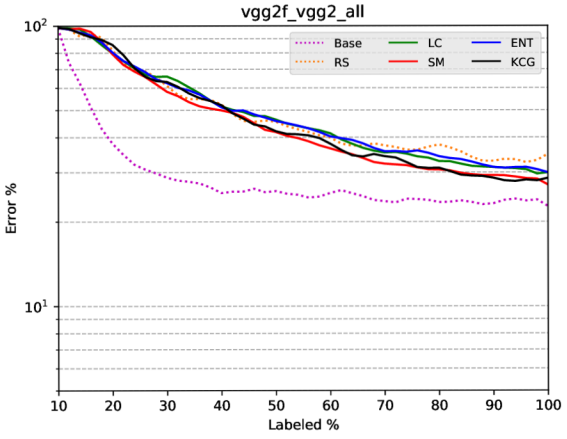

imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

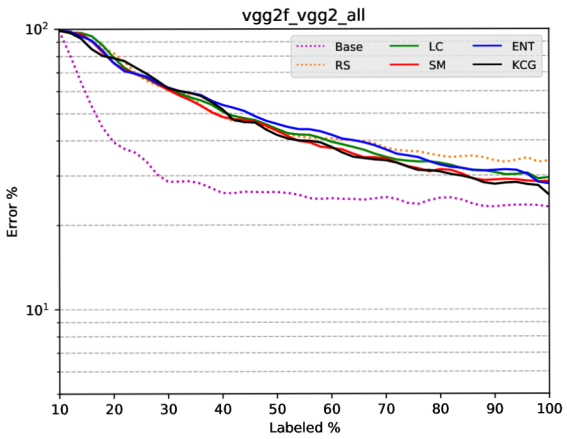

vgg2f_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

Shrnutí

Oba dva zvolené scénáře prokazují mírné zlepšení výkonu aktivního učení na jednoduchém datasetu, obzvlášť v úvodu při malé velikosti trénovací sady. Pseudo-označování taktéž pomáhá především metodě KCG stabilizovat se v prostřední fázi učení. Za povšimnutí stojí i zlepšení výkonu náhodného označování.

Naopak u obtížnějších datasetů žádný význačný vliv pozorovat nelze, vyjma nepatrného zlepšení ke konci učení při zhruba 80 až 90 % označených vzorků přidaných do trénovací sady. U složitějších datasetů též vykazuje mírně lepší výsledek scénář s mírnějším prahem pro uznání za modelem si jistý vzorek, u jednodušších datasetů přesně naopak.

Obecně lze tedy považovat efektivitu pseudo-označování za úzce spjatou s dokonalostí modelu a obtížností datasetu. Dokonalejší model více benefituje z dodatečných vzorků poskytnutých pseudo-označováním, slabší model tato technika nezachrání. Což dává smysl, dokonalejší model lépe dokáže klasifikovat zatím pro něj neoznačené vzorky a ve více případech si je dostatečně jist, aby vzorek provizorně označil. Mnou zvolené modely nedosahují takové kvality, aby si u dostatečně vlivného množství neoznačených vzorků byly v případě složitých datasetů dostatečně jisti jejich označením. K provizornímu označení tedy dojde jen v málo případech, jež nemají na trénovaní hluboké architektury znatelný vliv.

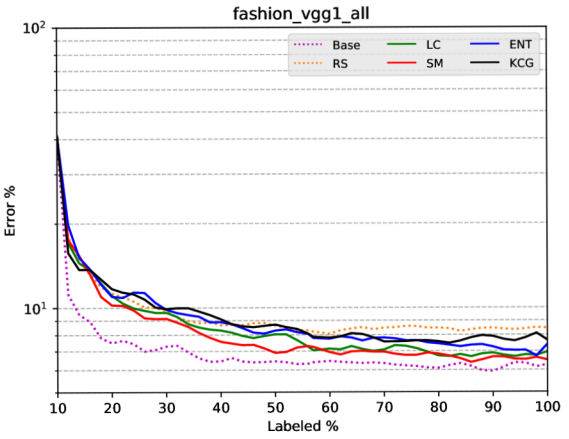
Hranice pro pseudo-označení vzorku zůstává zajímavým tématem pro další experimenty, například zda-li lze najít univerzálně vhodné hodnoty, či zda alespoň lze přiřadit vhodný řád této hranice k různým % chybovosti modelu na datasetu.

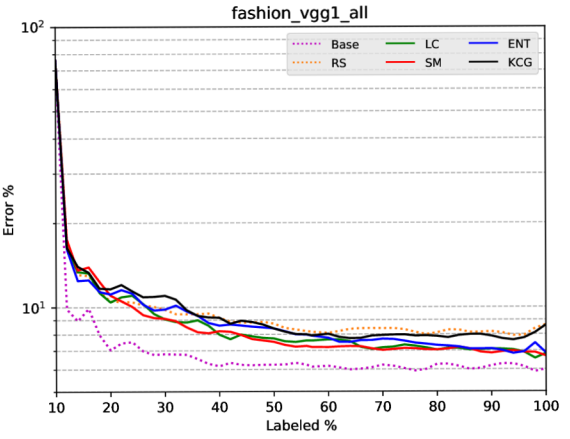
3.4.4 Experiment 4

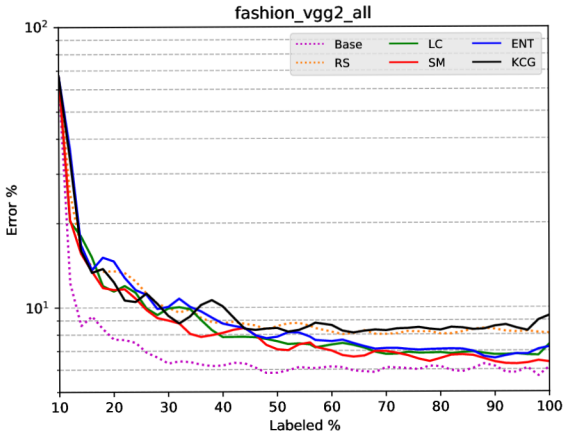
Ve finálním, stručnějším, experimentu prozkoumávám možnost zvrátit negativní vliv chyb anotátora pomocí pseudo-označování. Obavy budí varianta, kdy chyby anotátora natolik rozhodí model, že nebude schopen vybrat dostatek důvěryhodných vzorků pro účinné pseudo-označování, nebo dokonce provizorně označené vzorky rozhodí model ještě více.

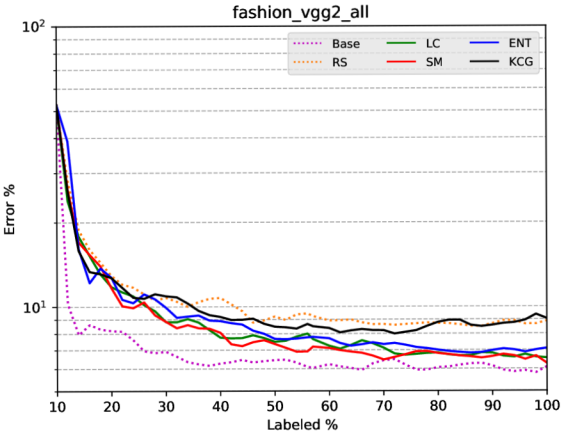
Pro experiment volím 85 % spolehlivost anotátora a parametry pseudo-označování $\delta = 0.0005$ a decay rate $dr = 0.000033$. Pro porovnání pak opět 85 % spolehlivost anotátora. V každé iteraci již tradičně probíhá jedna trénovací epocha a vzorky se přidávají v dávkách po 2 %.

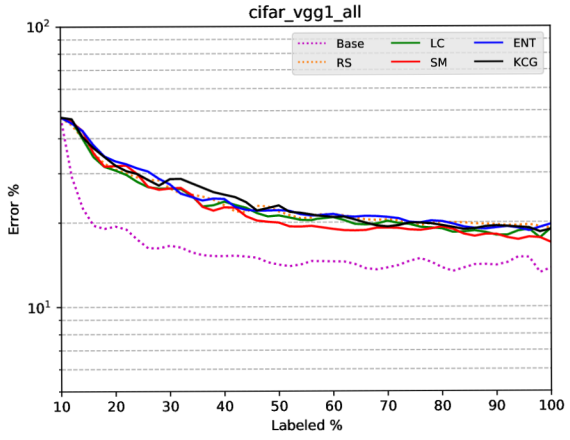
V následujících grafech je nalevo zobrazen výsledek scénáře aktivního učení se spolehlivostí anotátora 85 %, napravo pak taktéž ovšem v kombinaci s pseudo-označováním.

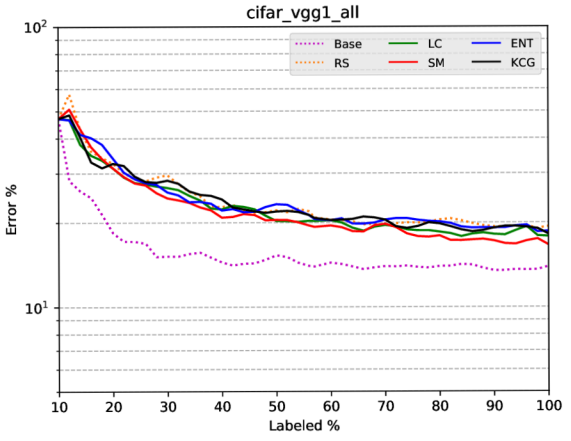

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

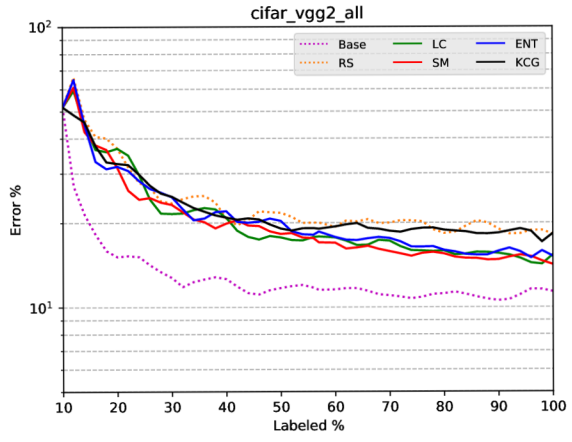

fashion_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

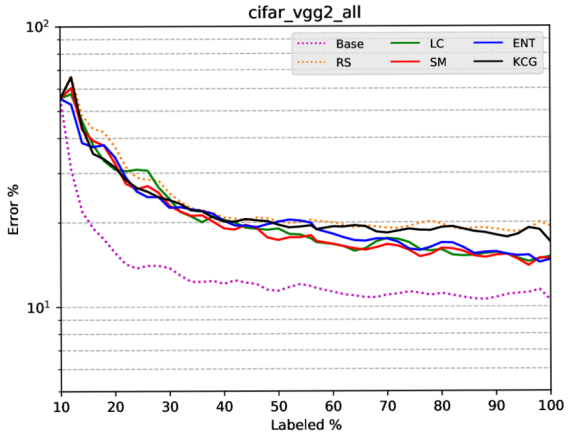

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

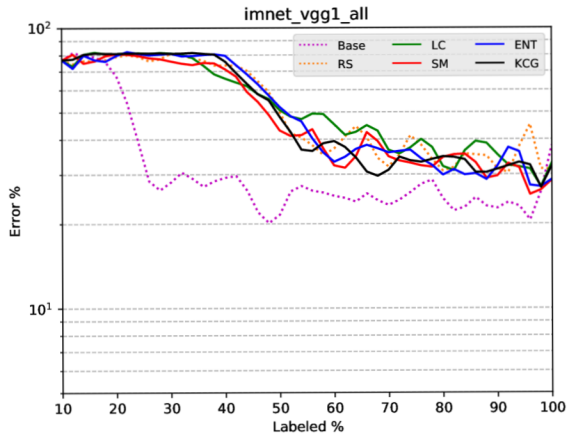

fashion_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

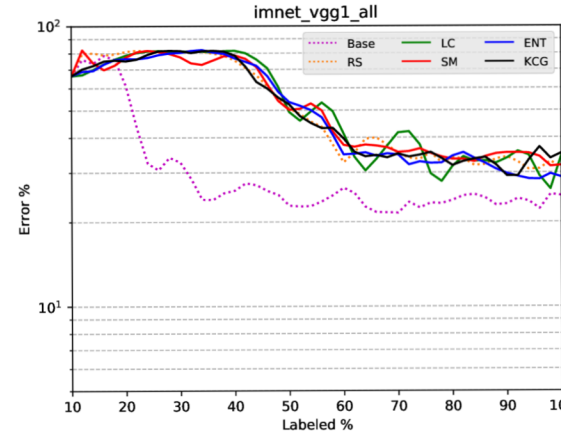

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

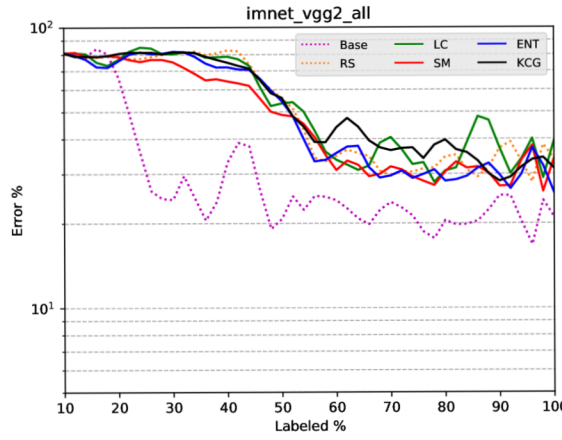

cifar_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

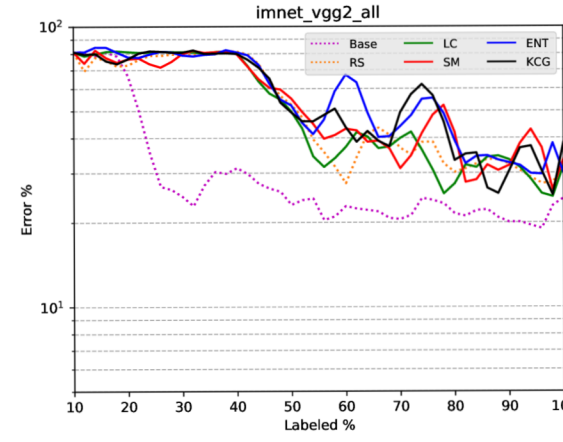

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

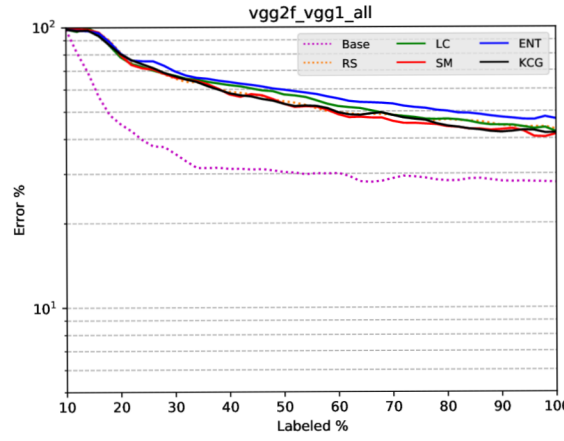

cifar_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

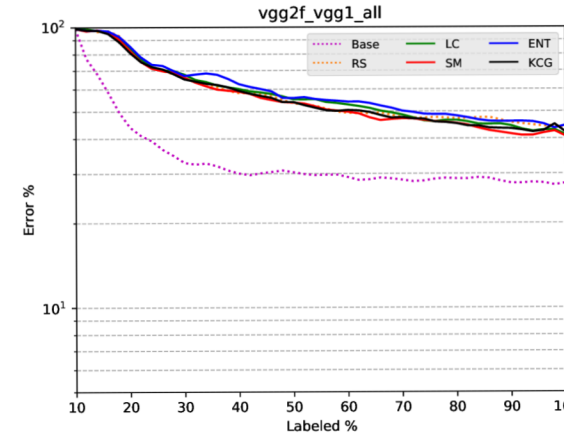

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

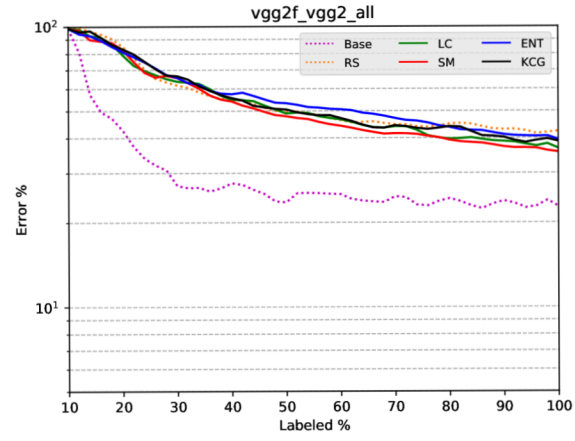

imnet_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

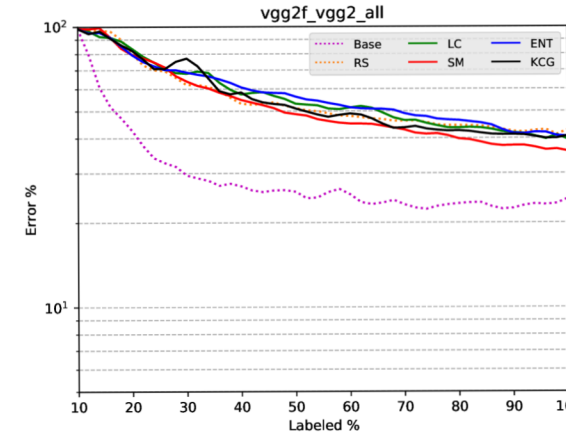

imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


imnet_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg1_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %


vgg2f_vgg2_all
Base
RS
LC
SM
ENT
KCG
Error %
Labeled %

Shrnutí

Byť pseudo-označování nedokáže výrazně vylepšit výkon strategií ovlivněný nepřesností anotátora, v jednoduchých i mírně obtížnějších datasetech lze pozorovat drobné zlepšení v úvodní fázi aktivního učení, obzvlášť v případě hlubšího modelu a metod ENT a LC.

V některých případech dochází i ke stabilizaci metody KCG. Její eventuální degradaci, pozorované již v mém druhém experimentu, však pseudo-označování zabránit nedokáže.

Pro obtížné datasety nelze pozorovat výrazné změny, v souladu s předcházejícím experimentem. Což dále umocňuje předpoklad, že pro těžení z výhod pseudo-označování je potřeba kvalitní model, adekvátní obtížnosti datasetu.

Námětem pro další experiment v téhle oblasti, mimo variant s adekvátnějším modelem, může být hledání hranice, kdy nepřesnosti anotátora dokáží zcela eliminovat benefity pseudo-označování.

Kapitola 4

Závěr

Pomocí experimentů jsem pozoroval chování vybraných strategií aktivního učení v kombinaci s konvolučními neuronovými sítěmi při kontinuálním trénování modelu. Zjištění z výsledků je hned několik.

Prvním z nich je skutečnost, že i tradiční strategie aktivního učení, mnohdy zatracované jako nevhodné pro použití s neuronovými sítěmi, dokáží dosáhnout dobrých výsledků na jednodušších datasetech a i na obtížnějších poráží náhodné vzorkování. Ze tří tradičních strategií si nejlépe vede *Margin sampling*, která dokáže držet krok na složitějších datasetech i se strategií přímo navrženou pro práci s neuronovými sítěmi *K-Center Greedy*. Oproti ní však není zdaleka tak výpočetně náročná. Ostatní dvě strategie, *Least certainty* a *Entropy sampling* dokáží na jednodušších datasetech podat skoro takový výkon, jako *Margin sampling*, na obtížnějších však výrazně ztrácí. Z těchto dvou pak podává lehce lepší výkon strategie *Least certainty*. Za zmínku stojí, že kvalita učícího se modelu v případě čistého kontinuálního učení nemá vliv na relativní výkon strategií. S kvalitnějším modelem pochopitelně všechny strategie pracují líp, mezi sebou si však drží obdobné odstupy.

Počet trénovacích epoch v jedné iteraci a velikost dávky přidávaných vzorků významně ovlivňuje výkon strategií. U strategie *K-Center Greedy* při vyšším počtu trénovacích epoch dochází k přetrénování, této strategii naopak více oproti ostatním svědčí větší počet přidávaných vzorků v každé iteraci. Celkově je tohle téma vhodné pro další experimenty, lze totiž v případě kontinuálního aktivního učení sledovat na složitějších datasetech problém přetrénování globálně na všech metodách. Ideálním řešením se jeví samoučící se algoritmus, jež dokáže dynamicky měnit počet trénovacích epoch a velikost dávky nových označených vzorků v závislosti například na současné míře přetrénování a velikosti trénovací sady, tedy kolik vzorků již bylo označeno. Z opačného pohledu, u jednodušších datasetů, kdy pomocí aktivního učení dosahuje model kvalitního výkonu srovnatelného s učením pasivním již okolo 40 či 50 % označení všech vzorků, by bylo vhodné zavést ukončovací kritéria pro šetření zdrojů.

Tradiční strategie, převážně pak *Margin sampling*, prokázali v dalších experimentech odolnost vůči omylům anotátora, alespoň při v praxi běžné chybovosti. Při chybovosti až 15 % se výkon strategií zhoršil jen o jednotky %. Naopak strategie *K-Center Greedy* na chybovost anotátora velmi trpěla. Pozoruhodné je, že mnohem horší degradaci výkonu tato strategie vykazovala při použití kvalitnějšího modelu. Pro další experimenty by stálo za to hledat hranici chybovosti, kdy se všechny strategie stávají neefektivními, a hlavně možnost zvrátit tento vliv pomocí opětovného dotazování vzorků, u kterých model tuší anotátorovu chybu.

Psuedo-označování dokáže mít na strategie prospěšný vliv, především v úvodních fázích učení, kdy je označeno jen okolo 20 či 30 % vzorků z celého datasetu. Dokáže též výkon strategií stabilizovat v pozdějších fázích. Problémem však je, že aby bylo pseudo-označování patrné, musí dosahovat model na zvoleném datasetu dostatečné kvality. Jinak není schopen provizorně označit tolik vzorků, aby to na výsledek mělo znatelný vliv. Pro další experimenty by bylo zajímavým cílem zjistit vhodné řády hranice jistoty modelu pro pseudo-označení vzhledem k jeho dosažené klasifikační kvalitě. Dalším tématem může být hledání nových vhodných strategií pro posouzení vzorků pro pseudo-označení.

Literatura

[1] Angluin, D.: Queries and concept learning. *Machine Learning*, ročník 2, 1988: s. 319–342.

[2] Angluin, D.: Queries revisited. In *Proceedings of the International Conference on Algorithmic Learning Theory*, Springer-Verlag, 2001, s. 12–31.

[3] Baldridge, J.; Osborne, M.: Active learning and the total cost of annotation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, ACL Press, 2004, s. 9–16.

[4] Bloodgood, M.; Shanker, V.: A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, ACL Press, 2009, s. 39–47.

[5] Brinker, K.: Incorporating diversity in active learning with support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, AAAI Press, 2003, str. 59–66.

[6] Cao, Q.; Shen, L.; Xie, W.; aj.: *VGGFace2: A dataset for recognising faces across pose and age*. [Online; navštíveno 18.04.2018].
URL https://www.robots.ox.ac.uk/~vgg/data/vgg_face2/vggface2.pdf

[7] Chen, S. F.; Rosenfeld, R.: A survey of smoothing techniques for ME models. *IEEE Transactions on Speech and Audio Processing*, ročník 8, č. 1, 2000: str. 37–50.

[8] Chollet, F.: *Keras: The Python Deep Learning library*. [Online; navštíveno 18.04.2018].
URL https://keras.io/

[9] Cohn, D.; Atlas, L.; Ladner, R.: Improving generalization with active learning. *Machine Learning*, ročník 15, č. 2, 1994: s. 201–221.

[10] Cohn, D.; Atlas, L.; Ladner, R.; aj.: Training connectionist networks with queries and selective sampling. In *Advances in Neural Information Processing Systems (NIPS)*, Morgan Kaufmann, 1990.

[11] Cohn, D.; Ghahramani, Z.; Jordan, M. I.: Active learning with statistical models. *Journal of Artificial Intelligence Research*, ročník 4, 1996: s. 129–145.

[12] Culotta, A.; McCallum, A.: Reducing labeling effort for stuctured prediction tasks. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, AAAI Press, 2005, s. 746–751.

[13] Dagan, I.; Engelson, S.: Committee-based sampling for training probabilistic classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, Morgan Kaufmann, 1995, s. 150–157.

[14] Dasgupta, S.; Hsu, D. J.: Hierarchical sampling for active learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, ACM Press, 2008, s. 208–215.

[15] Ducoffe, M.; Precioso, F.: Active learning strategy for CNN combining batchwise Dropout and Query-By-Committee. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, ESANN, 2017.

[16] Ducoffe, M.; Precioso, F.: Introducing Active Learning for Cnn under the Light of Variational Inference. 2017.

[17] Fei-Fei, L.; Li, K.: *ImageNet*. [Online; navštíveno 18.04.2018].
URL http://www.image-net.org/

[18] Fujii, A.; Tokunaga, T.; Inui, K.; aj.: Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, ročník 24, č. 4, 1998: str. 573–597.

[19] Goodman, J.: Exponential priors for maximum entropy models. In *Proceedings of Human Language Technology and the North American Association for Computational Linguistics (HLT-NAACL)*, ACL Press, 2004, str. 305–312.

[20] GoogleBrain: *TensorFlow: An open source machine learning framework for everyone*. [Online; navštíveno 18.04.2018].
URL https://www.tensorflow.org/

[21] Guo, Y.; Greiner, R.: Optimistic active learning using mutual information. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, AAAI Press, 2007, s. 823–829.

[22] Hoi, S. C. H.; Jin, R.; Lyu, M. R.: Large-scale text categorization by batch mode active learning. In *Proceedings of the International Conference on the World Wide Web*, ACM Press, 2006, s. 633–642.

[23] Hwa, R.: Sample selection for statistical parsing. *Computational Linguistics*, ročník 30, č. 3, 2004: s. 73–77.

[24] King, R.; Rowland, J.; Oliver, S.; aj.: The automation of science. In *Science*, 324(5923), 2009, s. 85–89.

[25] King, R.; Whelan, K.; Jones, F.; aj.: Functional genomic hypothesis generation and experimentation by a robot scientist. In *Nature*, 427(6971), 2004, str. 247–252.

[26] Krishnamurthy, V.: Algorithms for optimal scheduling and management of hidden markov model sensors. *IEEE Transactions on Signal Processing*, ročník 50, č. 6, 2002: str. 1382–1397.

[27] Krizhevsky, A.; Nair, V.; Hinton, G.: *The CIFAR-10 dataset*. [Online; navštíveno 18.04.2018].
URL https://www.cs.toronto.edu/~kriz/cifar.html

[28] Kullback, S.; Leibler, R. A.: On information and sufficiency. *Annals of Mathematical Statistics*, ročník 22, 1951: str. 79–86.

[29] Körner, C.; Wrobel, S.: Multi-class ensemble-based active learning. In *Proceedings of the European Conference on Machine Learning (ECML)*, Springer, 2006, str. 687–694.

[30] Lang, K.; Baum, E.: Query learning can work poorly when a human oracle is used. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, IEEE Press, 1992, s. 335–340.

[31] Lewis, D.; Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, Morgan Kaufmann, 1994, s. 148–156.

[32] Lewis, D.; Gale, W.: A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM/Springer, 1994, s. 3–12.

[33] Liu, Y.: Active learning with support vector machine applied to gene expression data for cancer classification. *Journal of Chemical Information and Computer Sciences*, ročník 44, 2004: str. 1936–1941.

[34] Lu, Z.; Bongard, J.: Exploiting multiple classifier types with active learning. In *In Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*, ACM Press, 2009, s. 1905–1906.

[35] McCallum, A.; Nigam, K.: Employing EM in pool-based active learning for text classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, Morgan Kaufmann, 1998, s. 359–367.

[36] Mitchell, T.: Generalization as search. *Artificial Intelligence*, ročník 18, 1982: s. 203–226.

[37] Moskovitch, R.; Nissim, N.; Stopel, D.; aj.: Improving the detection of unknown computer worms activity using active learning. In *Proceedings of the German Conference on AI*, Springer, 2007, s. 489–493.

[38] Nguyen, H. T.; Smeulders, A.: Active learning using pre-clustering. In *Proceedings of the International Conference on Machine Learning (ICML)*, ACM Press, 2004, str. 79–86.

[39] Roy, N.; McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the International Conference on Machine Learning (ICML)*, Morgan Kaufmann, 2001, str. 441–448.

[40] Scheffer, T.; Decomain, C.; Wrobel, S.: Active hidden Markov models for information extraction. In *Proceedings of the International Conference on Advances in Intelligent Data Analysis (CAIDA)*, Springer-Verlag, 2001, s. 309–318.

[41] Schein, A. I.; Ungar, L. H.: Active learning for logistic regression: An evaluation. *Machine Learning*, ročník 68, č. 3, 2007: str. 235–265.

[42] Sener, O.; Savarese, S.: Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *Proceedings of the International Conference on Learning Representations*, 2018.

[43] Settles, B.: Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[44] Settles, B.; Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, ACL Press, 2008, str. 1069–1078.

[45] Settles, B.; Craven, M.; Friedland, L.: Active learning with real annotation costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, 2008, s. 1–10.

[46] Settles, B.; Craven, M.; Ray, S.: Multiple-instance active learning. In *Advances in Neural Information Processing Systems (NIPS)*, ročník 20, MIT Press, 2008, str. 1289–1296.

[47] Seung, H. S.; Opper, M.; Sompolinsky, H.: Query by committee. In *Proceedings of the ACM Workshop on Computational Learning Theory*, 1992, str. 287–294.

[48] Shannon, C. E.: A mathematical theory of communication. *Bell System Technical Journal*, ročník 27, 1948: s. 379–423,623–656.

[49] Sheng, V. S.; Provost, F.; Ipeirotis, P. G.: Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, ACM Press, 2008.

[50] Tomanek, K.; Olsson, F.: A web survey on the use of active learning to support annotation of text data. In *Proceedings of the NAACL HLT Workshop on Active Learning for Natural Language Processing*, ACL Press, 2009, s. 45–48.

[51] Tomanek, K.; Wermter, J.; Hahn, U.: An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, ACL Press, 2007, s. 486–495.

[52] Tür, G.; Hakkani-Tür, D.; Schapire, R. E.: Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, ročník 45, č. 2, 2005: s. 171–186.

[53] Vlachos, A.: A stopping criterion for active learning. *Computer Speech and Language*, ročník 22, č. 3, 2008: str. 295–312.

[54] Wang, K.; Zhang, D.; Li, Y.; aj.: Cost-Effective Active Learning for Deep Image Classification. *IEEE Transactions on Circuits and Systems for Video Technology*, ročník 27, č. 12, 2017: s. 2591–2600.

[55] Xu, Z.; Akella, R.; Zhang, Y.: Incorporating diversity and density in active learning for relevance feedback. In *Proceedings of the European Conference on IR Research (ECIR)*, Springer-Verlag, 2007, str. 246–257.

[56] Yan, R.; Yang, J.; Hauptmann, A.: Automatically labeling video data using multi-class active learning. In *Proceedings of the International Conference on Computer Vision*, IEEE Press, 2003, s. 516–523.

[57] Yu, H.: SVM selective sampling for ranking with application to data retrieval. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, ACM Press, 2005, str. 354–363.

[58] Zalando: *Fashion MNIST: An MNIST-like dataset of 70,000 28x28 labeled fashion images*. [Online; navštíveno 18.04.2018].
URL https://www.kaggle.com/zalando-research/fashionmnist

[59] Zhang, C.; Chen, T.: An active learning framework for content based information retrieval. *IEEE Transactions on Multimedia*, ročník 4, č. 2, 2002: s. 260–268.

[60] Zhu, X.: *Semi-Supervised Learning with Graphs*. Dizertační práce, Carnegie Mellon University, 2005.

[61] Zhu, X.; Lafferty, J.; Ghahramani, Z.: Combining active learning and semisupervised learning using Gaussian fields and harmonic functions. In *Proceedings of the ICML Workshop on the Continuum from Labeled to Unlabeled Data*, 2003, s. 58–65.

Příloha A

Obsah DVD

Na přiloženém DVD jsou k dispozici všechny zdrojové soubory, včetně komprimovaných datasetů. Dále také toto *pdf*, i zdrojové soubory, ze kterých je sestaveno. Součástí je též krátká videoprezentace mé práce a výsledků. Umístění popisuje následující adresářová struktura:

- **script/** Adresář obsahující mnou vytvořený skript a datasety .
- **tex/** Zdrojové soubory pro sestavení tohoto dokumentu.
- **pdf/** Obě dvě požadované verze výsledného dokumentu.
- **videoprezentace/** Umístění krátké videoprezentace o výsledcích mé práce.

Příloha B

Detailnější popis skriptu

Skript `exps.py` (případně `exps_no_tf.py`, v situaci, kdy není k dispozici GPU pro urychlení výpočtů) spouští celé aktivní učení a je ovládán následujícími parametry, v daném pořadí:

Dataset Cesta k požadovanému datasetu v souboru formátu *.hdf5*. Ty se na přiloženém médiu nachází v adresáři `script/dataset/název-datasetu/`. Příkladem tedy může být *./dataset/imnet/imnet.hdf5*. Přiložené médium obsahuje datasety *fashion, cifar, imnet, vgg2f*, popsané v podkapitole 3.2. V elektronické podobě odevzdání z důvodu velikostního omezení datasety k dispozici nejsou.

Model Volba modelu konvoluční neuronové sítě. Buďto *vgg1* pro jednodušší síť, nebo *vgg2* pro hlubší síť. Obě jsou podrobněji popsány v 3.3. O tvorbu modelů se starají podprogramy `model_selector` a `vgg` v adresáři `script/models/`.

Strategie aktivního učení Volba strategie aktivního učení pro výběr vzorků k anotaci. Lze zadat *all* pro provedení experimentu opakovaně s každou dostupnou strategií, nebo *lc, sm, ent, rs, kcg* pro provedení experimentu pouze s jednou strategií. Jednotlivé zkratky reprezentují popořadě strategii nejnižší jistoty, nejmenšího rozdílu, použití entropie, náhodné vzorkování a K-Center Greedy. Implementace všech strategií se nachází v adresáři `script/methods/`. Lze též zadat hodnotu parametru *base*, kdy se provede trénování modelu pasivním způsobem a výsledek slouží pro porovnání výkonu strategií aktivního učení.

Rozpočet Hodnota v %, vyjadřující kolik vzorků z neoznačené sady lze celkem dotazovat. V podstatě určuje počet iterací aktivního učení. 100 % znamená, že postupně bude dotazována a označena celá neoznačená sada vzorků.

Počet trénovacích epoch Počet trénovacích epoch modelu po každém přidání nově anotovaných vzorků do trénovací sady.

Počet vzorků Velikost sady vzorků v % z celé neoznačené sady, kterou strategie aktivního učení vybírá pro anotaci v každé iteraci.

Přesnost anotátora Přesnost anotátora v %. Hodnoty nižší než 100 způsobí chybovost. Pomocí generátoru náhodných čísel se určí, zda-li se vzorku přiřadí jiné, chybné označení. Tuto funkčnost implementuje podprogram `oracle` v adresáři `script/methods/`.

Pseudo-označování Tento parametr ovlivňuje použití pseudo-označování v průběhu učení modelu. Hodnotou *yes* pseudo-označování bude použito, hodnotou *no* naopak ne. Pseudo-označování implementuje podprogram `pseudo_labeling` v adresáři `script/methods/`.

Výstupní soubor Cesta pro vytvoření výstupního souboru formátu *.npz*, obsahujícího údaje o proběhlém experimentu.

Příkladem spuštění může být:
`python exps.py ./dataset/fashion/fashion.hdf5 vgg2 all 100 1 2 100 no ./results/exp1`.

Pomocný skript `look_at_this_graph.py` vykresluje grafy z dat v souborech formátu *.npz*. Stačí mu k tomu pouze cesta k danému souboru jako argument.

Instalace

Skripty jsou snadno přenositelné, vyžadují pouze přítomnost Pythonu a instalaci několika knihoven, jejichž výčet se nachází v souboru `script/requirements.txt`. Ty lze jednoduše nainstalovat příkazem `pip install -r requirements.txt`.