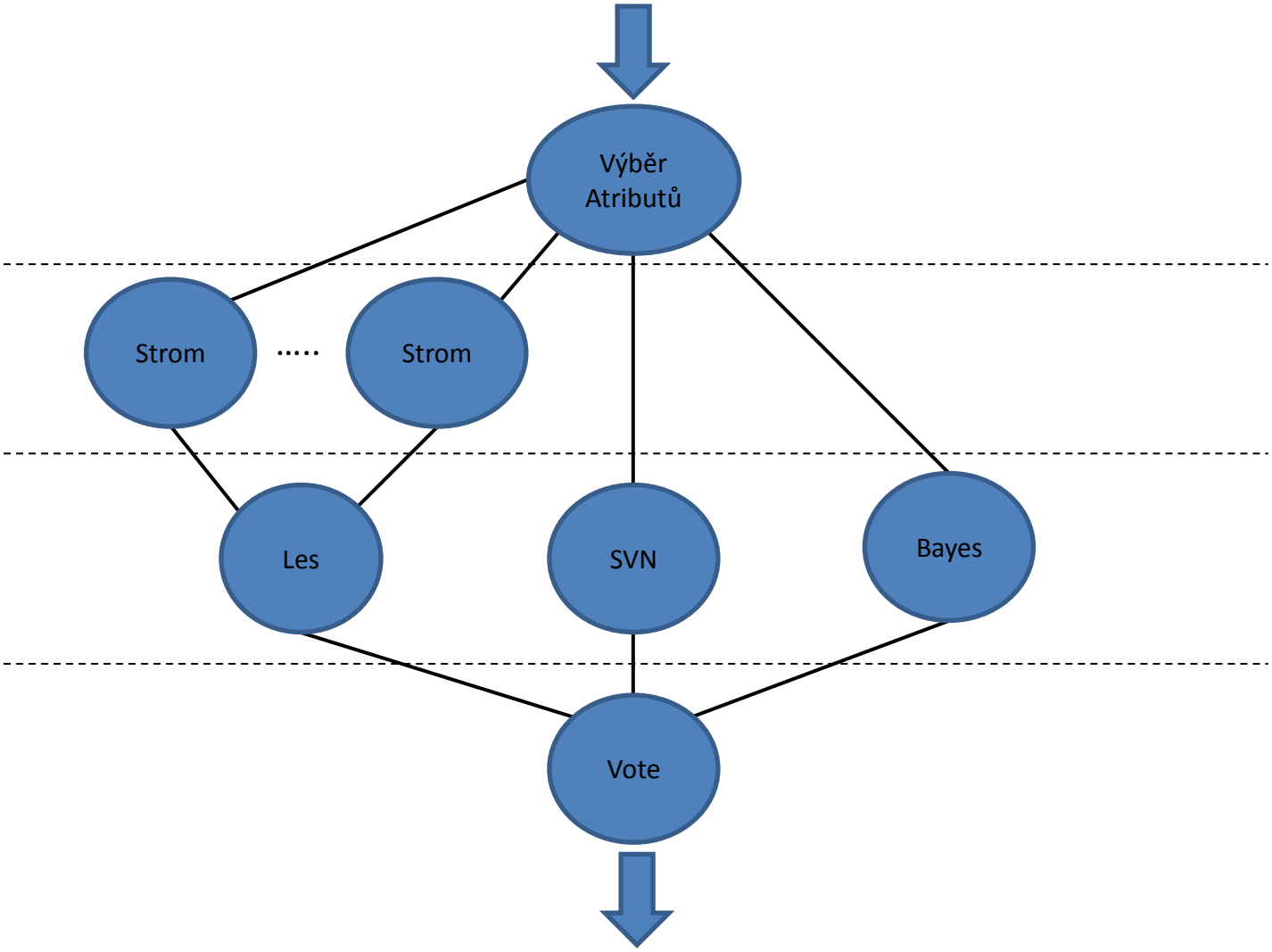
Deep Learning

Úvod

V referátu se věnuji pokročilejším metodám strojového učení. Konkrétně souboru metod deep learning. Jedná se o přiblížení strojového učení k umělé inteligenci jako takové. Jeho použití je široké, ale hlavně se využívá na úkoly spojené právě s umělou inteligencí, jako je počítačové vidění, zpracování přirozeného jazyka nebo například rozpoznávání audiovizuálních informací. Základní principy jsou známy již od osmdesátých let, ale kvůli nízké efektivitě, bylo od této ideji odpuštěno. Rozmach deep learningu nastal až v posledních letech a to mimo jiné díky zlepšování výpočetní síly počítačů a paralelizaci výpočtů. Hlavně však šlo o nalezení efektivních algoritmů pro učení. Síla deep learningu je ve stylu učení. Lze využít učení bez učitele, což znamená, že počítači dovolujeme „přemýšlet“. Například máme model, který umí rozpoznávat dopravní prostředky jako auta, motorky, vlaky. Pak ale narazí na jízdní kolo, počítač neví, že na obrázku je jízdní kolo a bude se ho snažit přiřadit k nejpodobnější kategorii. Pokud ale bude příkladů kola více, počítač může vytvořit novou kategorii pro kola a říct, že sice neví co to je, ale jsou to ty stejné věci. Člověk poté může specifikovat kategorii pouhým pojmenováním.

Idea deep learningu

Podíváme-li se na deep learning povrchně, můžeme na něj nahlížet jako na kompozici funkcí. Výstup první funkce, jde přímo na vstup druhé funkce a tak dále. Vezměme si klasický rozhodovací strom. Ten nám dává funkci, která má hloubku jedna, protože výpočet je řešen na úrovni vstupních dat. Hloubka stromu jako takového nás v tomto kontextu nezajímá. Takových stromů můžeme mít více, mohou být vytvořeny například pomocí jiných kritérií. Všechny tyto stromy jsou stále na stejné vrstvě paralelně. Jako další vrstvu můžeme použít funkci, která pro svůj výpočet používá ansámbl stromů, nebo také les. Například výpočet proběhne na všech stromech, a ty pak mezi sebou hlasují o výsledku. Toto už nám dává dvě vrstvy, protože v první potřebujeme rozhodovací stromy a v druhé jde výstup ze stromů do algoritmu lesa a z něj jde upravený výstup. Algoritmů strojového učení můžeme použít více. A to jak už zmiňované stromy, nebo ansámbly stromů, Bayesův algoritmus nebo třeba SVM. Tyto algoritmy mohou paralelně pracovat a jejich výstup můžeme poslat na další vrstvu, na meta-učící algoritmus, například Vote. Ten, podobně jako les, využívá ansámbl algoritmů, které hlasují. Získaly jsme tak tří vrstvou architekturu. Samozřejmě, abychom dostaly ještě lepší výsledek, můžeme předřadit algoritmus pro výběr atributů, který může zúžit počet vstupních algoritmů a odfiltrovat případný šum, který by mohl vzniknout nadbytečnými a zavádějícími informacemi. Výsledná architektura má hloubku čtyři. Samotná skladba architektury je v praxi dána konkrétním problémem, aby byl získán co nejlepší výsledek učení.



Víše můžeme vidět hlubokou architekturu, složenou z běžných algoritmů strojového učení znázorněnou jako graf.

Hluboké neuronové sítě

Nejčastějším a hlavním případem použití deep learningu, je využití v neuronových sítích. Neuronové sítě se dle definice řadí do vrstev. U nejjednodušší neuronové sítě je to vstupní vrstva, skrytá vrstva a výstupní vrstva. Obecně, neurony jsou vzájemně propojeny na vrstvu o jedna vyšší a o jedna nižší, ne však na své vlastní vrstvě. Klasicky, jsou v síti jedna až dvě skryté vrstvy, protože to stačí na řešení všech funkcí. Je přirozené, že s vícevrstvými sítěmi se experimentovalo i v minulosti, jenže nebylo efektivní. Kromě velmi vysokých výpočetních nároku, pro které v době prvních experimentů jednoduše nebyla k dispozici výpočetní síla. Vícevrstvé sítě vykazovaly mnohem větší míru chybovosti, než sítě s jednou nebo dvěma skrytými vrstvami. Problém byl v samotném algoritmu zpětné propagace. Síť špatně konvergovala a dostávala se do lokálního minima chyby. Zdálo se, jakoby zpětná propagace „vyšuměla“ při průchodu více vrstvami. Chyba se prostě nemohla promítnout na váhy neuronů vyšších vrstev, protože změna konvergovala někde uprostřed sítě. Jedinou výjimku z tohoto pravidla tvoří konvoluční neuronové sítě, které mají speciálně upravené skryté vrstvy. Propojení mezi vrstvami je u těchto sítí přesně definováno. Konvoluční neuronové sítě

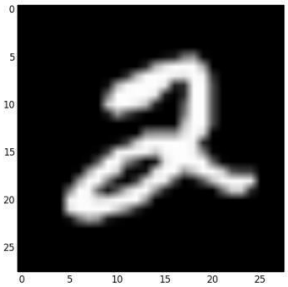
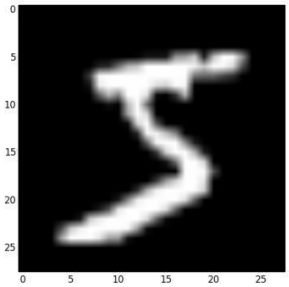
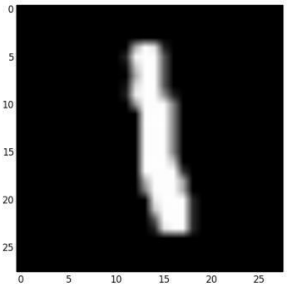
se využívají zejména pro klasifikaci obrázků, a to hlavně protože se snaží modelovat zpracování obrazu, které probíhá v mozku.

Řešením problému se ukázalo být prosté před trénování sítě učením bez učitele (unsupervised). Toto před učení dokázalo zefektivnit celý proces a vypořádat se s problémem vysoké chybovosti. Před trénování pomáhá lépe nastavit výchozí váhy v jednotlivých neuronech. Výchozí váhy pak lépe odpovídají vstupním datům a změny na váhách se lépe provádějí. Síť si jakoby vytvoří představu o datech, které do ní budou vstupovat a přizpůsobí se jim. Dochází tak k restrikci funkce. Díky ní, síť rychleji a lépe konverguje než při klasickém přístupu, tedy náhodně zvoleným vahám. V praxi se toto přeučení realizuje spuštěním modelu na podmnožině trénovacích dat, aniž by bylo známo, do které třídy patří.

Vyvstává ale otázka, jsou-li vícevrstvé neuronové sítě potřeba, když už síť se dvěma skrytými vrstvami dokáže aproximovat jakoukoliv funkci. Důvodem vícevrstvých sítí je možnost zachycení vyšší abstrakce. Obecně lze odvozovat abstrakci z abstrakce z abstrakce, a tak dále. To je právě ta síla hlubokých neuronových sítí, možnost zachytit tyto, pro člověka nezřetelné asociace. Další obrovskou výhodou je i možnost využít natrénovaný model ke generování místo klasifikování. Postup klasifikace si můžeme představit, jako přistup shora dolů, od obrázku k textovému popisu. Obrázek je postupně rozkouskován podle častých vzorů a častých vzorů z častých vzorů, kde ke každému vzoru je právě jeden neuron, až se dojde k poslední vrstvě, která obsahuje právě neurony určující třídu pro obrázek. Chceme-li generovat, potřebujeme už hotový model, z něj můžeme jít směrem zdola nahoru. Tedy začít u neuronu třídy, kterou chceme generovat a pak postupně jít po neuronech, které se aktivují. Tím tvoříme postupně z častých vzorů finální obrázek.

Hluboké neuronové sítě prakticky

Abych srovnal deep learning s klasickými algoritmy strojového učení, zvolil jsem datovou množinu MNIST. Jedná se o množinu rukou psaných číslic. Což znamená klasifikaci do desíti kategorií. Obrázky číslic jsou velké 28x28 pixelů, tedy 784 atributů pro každý obrázek. Obrázky jsou barevně normalizované, atributy díky tomu nabývají hodnot pouze na množině $(0, 1) \in \mathbb{R}$. Datová množina je rozdělena na tři podmnožiny. Tréninkovou, obsahující 50000 příkladů. Validační, obsahující 10000 příkladů. A testovací, obsahující také 10000 příkladů.

Jako softwarové nástroje pro deep learning jsem postupně prozkoumal:

- **DeepLearning Toolbox** - V Matlabu, kromě pár praktických ukázek naprosto chybí jakákoliv dokumentace k nástroji. Pro praktické použití nevhodné.

- **Theano** - Pythonovská knihovna Montrealské univerzity. Velmi zajímavé tutoriály k nejrůznějším algoritmům deep learningu. Velmi silný, ale dle mého názoru složitý nástroj na implementaci. Široká komunita příznivců. Paralelizace přes GPU.

- **Caffe** - Framework pro deep learning. Jednotlivé sítě jsou vytvářeny pouze konfiguračními soubory, nikoliv kódem jako takovým. Pěkný a přehledný nástroj, podporující výpočet přes GPU. Pouze pro Linux nebo Mac, port na Windows je nefunkční.

- **DeepLearning4j** - Javovská knihovna, která je stále ve vývoji. Přehledný kód jak pro vytváření modelu, tak pro jeho evaluaci. Také podporuje paralelní výpočet na GPU. Rozrůstající se komunita uživatelů.

Pro klasické algoritmy jsem využil nástroj Weka. Výsledky jsem shrnul do následující tabulky:

Algoritmus	Správně klasifikované	Čas
Rozhodovací Strom	88.66%	4.2m
Ansámbl Stromů (Les)	93.26%	10.2s
Bayesův Algoritmus	69.65%	45.4s
Convolutional Net (3L)	99.08%	8.15h
Deep Belief Net (3L)	98.63%	15.31 + 7.44 = 22.75h

Poslední dva algoritmy, jsou algoritmy deep learningu. Lze vidět, že mají nejlepší procento správně klasifikovaných, ale za cenu obrovského výpočetního času. Výpočetní čas u Deep belief network je součtem času před trénování a následného vlastního trénování.