

## Neuronové sítě

Umělá inteligence je jedním z nejmladších a nejrychleji se rozvíjejících vědních odvětví. Tímto číslem zahajujeme seriál článků, v nichž se seznámíte s několika jeho nejzajímavějšími oblastmi, a to s neuronovými sítěmi, genetickými algoritmy, diferenciální evolucí, fuzzy logikou, počítačovým viděním a robotikou.

### Umělá inteligence: hrozba, či naděje? (1)

Zařazení umělé inteligence jako obooru je vcelku obtížné. Lze na ni pohlížet jako na matematickou disciplínu s aplikacemi, nebo také jako na technický obor. Smyslem našeho seriálu však není posoudit, kam patří, ale podat o této perspektivní a pro nezasvěceného stále ještě tajemné disciplíně alespoň základní informace.

Závratný růst tohoto obooru uspíšilo mnoho faktorů, jako např. zvyšování požadavků v oblastech automatizovaného řízení, průzkumu mořských hlubin, vulkánů či planet a řada dalších činností, kde je přítomnost člověka z technických či zdravotních důvodů vyloučena.

Jednou z nezbytných vlastností systému nasazeného v takových oblastech je schopnost "učit se", tedy přizpůsobovat své reakce dříve získaným zkušenostem. Není proto divu, že se vědci nechali inspirovat "matkou přírodotu" a pokusili se za tím účelem alespoň částečně napodobit mozkovou činnost vyšších živočichů. Právě této disciplíně se proto věnujeme nejdříve.

## Neuronové sítě

### Historie

Vznik neuronových sítí spadá do první poloviny 20. století, kdy Američan W. S. McCulloch publikoval první práce o neuronech a jejich modelech. Ve 40. letech vypracoval se svým studentem W. Pittsem model neuronu, který se prakticky používá dodnes. Na základě těchto výsledků vytvořil v roce 1958 F. Rosenblatt první funkční perceptronovou síť. Měla však jeden velký nedostatek – byla schopna řešit pouze problémy, které byly tzv. lineárně separabilní. Tohoto faktu "využil" Rosenblattův bývalý spolužák M. Minsky spolu s S. Papertem k publikování knihy "Perceptron" (1969), která upozorňovala na tuto slabost Rosenblattovy sítě. Vzhledem k dobrému matematickému zpracování knihy a vědecké popularitě obou autorů pak zájem o neuronové sítě poměrně rychle opadl, prakticky až na nulu, a v tomto stavu setrval asi do poloviny 80. let.

Jen díky několika průkopníkům, kteří se nedali odradit, došlo v polovině 80. let k renesanci neuronových sítí. Jejich význam se zvětšuje každým dnem a používají se stále více. V těchto letech vznikla práce D. Rumelharta, G. Hintona a R. Wiliamse "Learning Internal Representation by Error Propagation". Pojednává o vícevrstvých sítích, schopných řešit i problémy, které nejsou lineárně separabilní. V 80. letech vznikly i další typy sítí jako Hopfieldova síť, Kohonenova síť a Grossbergova ART síť. Uplatnění neuronových sítí pak nabývalo stále většího významu, podobně jako tomu tehdy bylo s PC.

### Co je neuronová síť

Neuronové sítě jsou v podstatě jednoduché matematické algoritmy, které simulují činnost základních stavebních kamenů a struktur lidského mozku – biologických neuronů a jejich vzájemných spojů. Základním schématem je síť složená z primárních jednotek – neuronů. Neuron, základní stavební jednotku neuronových sítí, si lze představit jako "černou krabičku", která seče všechny vstupy (ohodnocené tzv. vahami) a takto získané číslo použije jako argument přenosové funkce. Funkční hodnota vystupující z neuronu pak vstupuje do další vyšší vrstvy nebo vystupuje ven ze sítě. Taková struktura má schopnost "učit se" na daném problému, který je obsažen v tzv. trénovací množině (což není nic jiného než množina vektorů popisující daný problém).

Neuronové sítě mohou mít jednu, dvě nebo tři vrstvy. Více vrstev už nepřináší viditelné zlepšení kvalit sítě, ale prodlužuje čas potřebný k učení. U vícevrstvých sítí platí, že první vrstva je vždy větvící, což znamená, že neurony ve vstupní vrstvě pouze distribuuují vstupní hodnoty do další vrstvy.

Vzhledem k tomu, že se obecně jedná o vícebodový vstup do sítě, mluvíme o vstupních, resp. výstupních vektorech informací. Potřebný počet neuronů v jednotlivých vrstvách je variabilní a

záleží na řešeném problému – určit jej však zdaleka není triviální záležitost. (V dnešní době sice existují vzorce pro výpočet neuronů v každé vrstvě, ale ani tyto vzorce nejsou všeňkem.)

Mnohem vhodnější způsob, jak určit počet neuronů, je použít síť, která si sama tento počet mění podle vývoje globální chyby. Každému vstupu do neuronu je přiřazena tzv. váha  $W_{xy}$ . To je bezrozměrné číslo, které určuje, jaký význam má daný vstup pro příslušný neuron (ne pro síť či problém).

Schopnost učení neuronových sítí spočívá právě v možnosti měnit všechny váhy v síti podle vhodných algoritmů – na rozdíl od sítí biologických, kde je tato schopnost založena na možnosti tvorby nových spojů mezi neurony. Fyzicky jsou tedy obě schopnosti učení založeny na rozdílných principech, z hlediska logiky však nikoliv. V případě vzniku nového spoje (vstupu) u biologického neuronu je to stejné, jako když je v technické síti spoj mezi dvěma neuronami původně ohodnocen vahou 0 (a tudíž jako vstup pro neuron, do něhož vstupuje, neexistuje) a v okamžiku, kdy se váha změní na nenulové číslo, se daný spoj zviditelní – vznikne.

#### Jak síť funguje

Nově vytvořenou, ale také jakoukoliv “nevyučenou” neuronovou síť lze považovat za jakéhosi technického novorozence, který nic neumí. Neumí rozeznávat, klasifikovat, atd. Aby se síť mohla používat, musí se potřebné dovednosti naučit podobně jako kterýkoliv živý tvor (množství informací a délka učení se samozřejmě nedají porovnávat). Proto byly vyvinuty algoritmy, pomocí nichž se síť dokáže naučit správným reakcím na danou množinu informací.

Takový algoritmus se obvykle dělí na dvě fáze: na fázi aktivační (vybavovací) a adaptační (učící), které ke své činnosti potřebují už zmíněnou trénovací množinu. Tou je skupina vektorů obsahujících informace o daném problému. Jsou-li v ní dvojice vektorů vstup – výstup, říkáme, že učíme síť “s učitelem”; pokud trénovací množina obsahuje jen vstupní vektory, mluvíme o učení “bez učitele”.

Cyklické střídání obou fází představuje vlastní učení, při kterém – obrazně řečeno – dochází k “přelévání” informací ze vstupu na výstup a naopak. Pokud používáme jen fázi aktivační, pak mluvíme o vybavování (tuto fázi používáme samostatně jen tehdy, když je síť naučena).

Představme si nyní, že máme několik tříd dopravních prostředků, např. třídu kol, aut, lodí atd. Z každé třídy vybereme množinu reprezentativních zástupců (vzorů pro učení) a každý z nich popíšeme vhodným číselným způsobem ve formě vektorů. Pro každou množinu vektorů jedné třídy vytvoříme vzorový vektor, který bude zastupovat tuto třídu z reálného světa.

V tomto okamžiku tedy máme skupiny vektorů popisující jednotlivé členy a ve formě vektorů také jim příslušející představitele tříd. (Například máme vektor popisující vybrané členy z třídy kol a vektor, který říká “já jsem třída kol”.) V procesu učení se pak učící algoritmus snaží najít takovou kombinaci vah, které umožní přiřazení vektoru třídy jejím členům. Jinak řečeno, hledá se taková kombinace vah, že pokud ve vybavovací fázi položíme na vstup vektor popisující např. závodní kolo, pak by se na výstupu měl objevit vektor, který říká “já jsem třída kol” nebo, chcete-li, “daný vstupní vektor patří do třídy kol”.

To, zda se naše síť naučí správným odezvám na dané podněty, závisí na více okolnostech – na množství vektorů a jejich velikosti, topologii sítě, odlišnosti charakteristických vlastností jednotlivých tříd, přípravě trénovací množiny a jiných. Povšimněme si také, že schopnost sítě zařazovat jednotlivé vstupní členy do známých tříd je v podstatě dána tím, že síť počítá “vzdálenost” daného člena od členů již přiřazených a na základě toho usuzuje, do jaké třídy daný vektor patří.

#### Rozdíl mezi PC a neuronovou sítí

I když neuronové sítě “běží” na PC, nelze oba prostředky ztotožňovat. Při používání klasického PC musíme pro řešení problému vytvořit program, do něhož obvykle zahrneme ve formě podmínek a rozhodovacích instrukcí veškeré dostupné informace. Co se však stane, je-li nás program postaven před problém, který sice patří do třídy známých problémů, ale je dost odlišný? Obvykle je takový vstup ignorován, nebo v lepším případě je obsluha alespoň upozorněna, že se vyskytl neznámý případ a byl “odložen bokem”. Pak musí opět nastoupit programátor a program upravit.

Použijeme-li neuronovou síť, jsme na tom lépe. Nemusíme vymýšlet žádný algoritmus a (při vhodné konfiguraci a dobré zvládnutém učení) naše neuronová síť zareaguje správně a novou

informaci s velkou pravděpodobností zařadí do správné třídy. Podstatné je, že přitom není třeba žádné úpravy sítě.

Několik nejdůležitějších rozdílů mezi klasickým PC a neuronovou sítí vidíte v připojené tabulce. Samozřejmě, nic na světě není neměnné, a jak vzrůstá počet nových informací, je pravděpodobné, že vzniknou i nové třídy informací, které se naše síť musí doučit a případně bude nutno pozměnit i konfiguraci. Ale i to se může obejít bez přítomnosti člověka – vraťme se pro názornost k našemu příkladu.

Představte si dva programátory, kteří se neznají a oba dostanou stejný úkol: navrhnut algoritmus, který má podle vstupních informací rozlišit, do jaké třídy patří daný dopravní prostředek. Kolo do třídy kol, auto do třídy aut, letadlo do třídy letadel atd.

“Klasický” programátor nejspíš vytvoří kriteriální filtry typu “jestliže má objekt dvě kola, řetězový převod na zadní kolo, řídítka, sedátko pro jednu osobu atd., pak jej zařaď do třídy kol” a podobně to udělá pro všechny možné třídy. (Není snad třeba zdůrazňovat, že chytřejší programátor bude popis dělat v proporcích a ne v absolutních rozměrech – i malé dětské kolo je kolo.)

Co se však stane, jestliže má takový program vyhodnotit kolo z minulého století (obří přední kolo s malíčkým vzadu a navíc bez převodu – šlapky byly přímo na předním kole) nebo dětské kolo s pomocnými kolečky či nějaký nový futuristický model bicyklu? Pravděpodobně takový předmět vyřadí a programátor bude muset dělat úpravy, což stojí jak čas, tak peníze. (A navíc, jak tvrdí Murphysho zákony, pokud odstraňujeme chybu v programu, obvykle se dopustíme dalších dvou.) Snad je už zbytečné zdůrazňovat, že (dobře navržená) neuronová síť by s velkou pravděpodobností zařadila nový objekt (kolo) do správné třídy.

#### Využití

Využití neuronových sítí je opravdu široké a nabývá čím dál tím více na významu. Lze je použít například pro

I identifikaci radarových či sonarových signálů,

I predikci chování dynamických systémů,

I klasifikaci,

I optimalizaci,

I filtraci

a v mnoha dalších úkolech. Pro ilustraci nyní uvedeme několik ukázkových příkladů.

#### Predikce

Predikce je činnost postavená na různých matematických algoritmech. Jejím cílem je co nejpřesněji odhadnout budoucí stav daného dynamického systému na základě současného stavu, historie jeho chování a jeho matematického popisu.

Potřeba predikovat budoucí děje je stará jako lidstvo samo a v posledních desetiletích je navíc mnohem silnější díky našim větším dovednostem v oblasti matematiky a fyziky. Dalším odvětvím, které čím dál více vyžaduje znalost budoucího vývoje, je ekonomika (informace, jaké budou zítra či za týden ceny akcii nebo třeba ropy, mohou mít cenu zlata).

V regulační technice zase často potřebujeme vědět, jaký bude stav regulované soustavy v blízké budoucnosti, meteorologové musejí předpovídat počasí, a osoby závislé na sluneční aktivitě zase zajímá, kdy se dají očekávat nepříznivá maxima či odchyly od normálu. V tomto výčtu by se jistě dalo pokračovat.

Neuronové sítě mají oproti standardním predikčním technikám tu výhodu, že ke své činnosti nepotřebují model a navíc jsou tolerantní k šumu v dané časové řadě (i při zašuměné řadě jsou schopny poskytnout rozumné výsledky v porovnání s klasickými metodami).

Jako příklad uvedeme dnes už klasický problém – predikci sluneční aktivity. Jako výchozí informace byla použita časová řada relativního čísla slunečních skvrn v rozmezí let 1900 až 1992. Sluneční skvrny jsou v podstatě vyvěrající proudy žhavého plynu, které se na povrchu ochladí a jsou viditelné ve formě tmavších skvrn. Počet skvrn se zakresluje do grafu sluneční aktivity a jeho budoucí průběh nám může v mnohém pomoci (biopředpovědi, plán kosmických misí s lidskou posádkou atd.).

Sluneční aktivity se opakují v cyklech několika typů (11letý, 80letý, ...) s různými odchylkami, které se dají stěží klasicky předpovědět (mimo jiné také proto byla vyzkoušena neuronová síť).

Časová řada sluneční aktivity byla při pokusu rozdělena na dvě části: na řadu pro trénování (1900 – 1965) a řadu pro otestování naučené sítě (1965 – 1992). Vlastní síť se skládala ze tří

vrstev s pyramidální topologií, tzn. že počet neuronů na vstupu byl největší (36) a směrem k výstupu se zmenšoval. Jako přenosová funkce byla použita logistická funkce ve všech neuronech a vrstvách. Výstupní neuron byl jen jeden a na jeho výstupu byla predikovaná data pro budoucí měsíční průměr aktivity.

To napovídá o složení trénovací množiny: 36 prvků ve vstupním vektoru a jeden výstupní – ten byl vlastně 37. v pořadí od začátku vektoru. Posun vektoru byl jeden měsíc, což při řadě znázorňující měsíční průměry znamená posun o jeden člen. Tato poslední hodnota byla při učení položena na výstup, a tímto postupem se síť naučila predikovat – přiřazovat novou budoucí hodnotu na základě posledních známých 36.

Vlastní učení zahrnovalo 10 000 epoch v trvání 10 minut na síti vytvořené v jazyce C++. Pak byl proces učení zastaven a druhá část řady (1965 – 1992) byla použita pro test, jak umí síť predikovat. Postupně byly na vstup síti pokládány jednotlivé vektory o 36 prvcích a výstupní hodnota byla zanášena do grafu, kde ji lze porovnat s originálním průběhem. Předpovězené hodnoty a originální průběh se kryjí velmi dobře a zajímavé rovněž je, že síť dokázala předpovědět i "zub" na sestupu po prvním předpovězeném maximu, který se v historii této řady nevyskytoval. Problematiku snad dostatečně dokreslí připojené obrázky.

#### Potlačení šumu

Potlačení šumu je další oblastí, kde lze využít neuronové sítě. Také to, že neuronová síť dokáže rekonstruovat poškozené vzory, není nic jiného než potlačování šumu (obvykle si totiž pod tímto pojmem představujeme jen odstraňování šumu z časové řady, zatímco např. na možnost opravy vadných pixelů v obrázku se zapomíná).

V mnoha technických aplikacích se pro potlačování šumu používají různé filtry, které mohou být poměrně slušně nahrazeny neuronovou sítí. Pokud je neuronová síť dobře naučena, pak je signál ze zašuměného vzoru úspěšně vyfiltrován a šum silně potlačen. Jako příklad si uveďme dnes velice moderní disciplínu rozpoznávání písma.

Byla použita vícevrstvá síť naučená na tvar písmene H. Do tohoto tvaru byl postupně přidáván šum v několika úrovních. Síť začala dělat první chyby až při znečištění 50 % (viz obrázky), což dost převyšuje odhadní schopnosti člověka. Nutno podotknout, že úspěšnost odhadu sítě také záleží na tom, jaké písmeno použijeme. Pokud jsou si dvě písmena podobná, jako např. O a U, lze i při nízkém stupni zašumění očekávat chybné odhady a naopak. To ale samozřejmě platí i pro lidský mozek, jehož odhad při vztřustingajícím stupni zašumění klesá více než u neuronové sítě.  
Ivan Zelinka

Autor:

Ivan Zelinka

Rubrika:

Magazín

Vydání:

729755 - 729785