

Po malé exkurzi do problematiky neuronových sítí v minulém čísle Chipu se dnes pustíme do další kapitoly umělé inteligence, vědního oboru, který možná v příštím století ovlivní budoucnost lidstva víc, než si dnes vůbec dokážeme představit.

Umělá inteligence: hrozba, či naděje? (2)

Určitě není náhodou, že se vědci na své cestě za umělou inteligencí obracejí pro poučení k živé přírodě. Jak jsme viděli minule, je tomu tak v případě neuronových sítí a – jak uvidíme dále – také nejrůznější teorie týkající se evoluce umělých systémů vycházejí z nejmodernějších poznatků biologů o vývoji živých organismů. Zdroj inspirace ostatně prozrazují i názvy zmíněných disciplín, o nichž si něco povíme v tomto článku: genetické algoritmy a diferenciální evoluce.

Historie

Genetické algoritmy byly odvozeny na základě biologické genetiky a Darwinovy teorie evoluce, ovlivňující vývoj všeho živého na této planetě. Při matematickém vyjádření principů genetického křížení se také přihlédlo k již existující terminologii v oblasti biologické genetiky, což vedlo a vede k používání biologických termínů v počítačové vědě, která se na první pohled jeví být velmi vzdálena vlastní genetiky. Tak se stalo, že se dnes v počítačovém světě můžeme setkat i s takovýmito “cizorodými” výrazy:

Chromozom – v přírodě je to část molekuly DNA, která je stočena do záhybů. V PC aplikacích jde o vektor popisující jeden či více parametrů problému, který má být řešen.

Geny jsou jednotlivé části chromozomu. Kompletní genetický popis organismu je tzv. **genotyp**. V souvislosti s genotypem se ještě uvádí i tzv. **fenotyp**, který je v podstatě fyzickým popisem genotypu. V počítačové interpretaci potom například binárnímu vyjádření genotypu 0101 odpovídá jako fenotyp jeho dekadická hodnota, tedy 5.

Allela – obecné označení pro množinu jistých hodnot, kterých mohou geny nabývat.

Prvopočátky genetických algoritmů sahají teprve do sedmdesátých let tohoto století, kdy se výzkumník J. Holland rozhodl využít výhody genetické dědičnosti v technických aplikacích. Až díky jemu a skupině jeho studentů vešla problematika genetických algoritmů ve známost.

Na dalším šíření se pak podíleli např. D. E. Goldberg, L. Davis a jiní. V dnešní době je používání genetických algoritmů poměrně rozšířené v mnoha oblastech vědy a techniky, například pro výpočet optimálního tvaru některých částí letadla, navrhování optimální struktury neuronové sítě, určování trajektorie “končetin” robota ap.

Genetické algoritmy se řadí díky své podstatě do třídy tzv. *evolučních algoritmů* – dnes už spolu s tzv. diferenciální evolucí, což je v této oblasti novinka. **Diferenciální evoluce** (na vývoji a ověření tohoto algoritmu pracují také autoři tohoto příspěvku) má v porovnání s genetickými algoritmy další výhodu v tom, že dokáže vyřešit daný problém mnohem rychleji a přesněji, a navíc může pracovat s libovolným typem čísla (integer, float, binary) nebo jejich kombinacemi, což u genetických algoritmů přináší jisté problémy. Vzhledem k tomu, že struktura algoritmu diferenciální evoluce je dost podobná genetickým algoritmům, uvedeme zde jen příklad jejího použití (v závěru článku).

Principy

V přírodě hraje genetika důležitou roli. Při rozmnožování člověka musí mít potomek 46 chromozomů. Od každého z rodičů dostane polovinu – výsledná kombinace pak určuje pohlaví a ostatní vlastnosti. Potomek sdílí po svých rodičích genovou výbavu a s tím částečně i jejich schopnosti. Vzhledem k tomu, že v přírodě přežívají jen ti nejschopnější, umírají nevyhovující jedinci velmi rychle, dříve než stačí předat nevyhovující genetickou výbavu svým potomkům.

Zde je ovšem na místě podotknout, že jejich výbava byla nevýhodná jen pro aktuální okolní prostředí a v jiném prostředí by mohli být úspěšní. Vezměme si např. různé druhy vyhynulých pravěkých zvířat, která zmizela jen proto, že se nestačila přizpůsobit okolním podmínkám. Kdyby se podmínky nezměnily, asi by žila dodnes.

Kritériem pro přežití druhu je mimo jiné také to, zda se jedná o rozmnožování bezpohlavní či pohlavní (i s tím se setkáme v genetických algoritmech v počítačovém prostředí). Na první pohled by se mohlo zdát, že bezpohlavní rozmnožování je jednodušší a méně náročné. Při pohlavním rozmnožování totiž musí jedinci daného druhu vynaložit hodně energie na to, aby našli vhodného partnera a obstáli v konkurenci ostatních (to jistě i my lidé můžeme potvrdit), což činí tento způsob na první pohled náročným a nevhodným. Jak je tedy možné, že je na světě rozšířeno převážně pohlavní rozmnožování?

Odpověď je jednoduchá. Představte si, že ve stejném životním prostředí existují dva druhy (A a B), přičemž A se rozmnožuje nepohlavně (dělením) a B pohlavně. V případě A dostane nový jedinec kompletní genetickou kopii od svého rodiče i s případnou mutací. V případě B je zapotřebí ke vzniku nového jedince dvou rodičů. Každý z nich mu předá pouze část své genetické výbavy (je to ovlivněno způsobem dělení genů), díky čemuž se potomek od rodičů liší. Takto vzniklý jedinec dokáže díky nové genetické výbavě lépe vzdorovat okolním podmínkám, což může po dlouhé době vést ke kvalitativním skokům ve srovnání s druhem A (který pak bude druhem B nejspíš vytlačen).

Na stejném principu jako náš druh B pracují i genetické algoritmy v počítačových aplikacích. Náhodně se vygeneruje množina chromozomů, z níž se vybírají dvojice (rodiče) na základě toho, jak dobře splňují tzv. *kritériální funkci*. Potomci jsou tvořeni "polovinou" z každého rodiče a nahrazují své rodiče. Stejně jako v přírodě, tak i v technických aplikacích hrají důležitou roli náhodné změny – mutace. To se v technice obvykle řeší pomocí generátoru náhodných čísel.

Při používání genetických algoritmů v technických aplikacích se setkáme ještě s dalšími pojmy, např.:

Objektivní funkce – je to funkce, kterou chceme minimalizovat (v případě neuronových sítí je to globální chyba sítě atd.).

Vhodnost – číslo, které udává vhodnost nového potomka z hlediska kritériální funkce.

Představuje v podstatě matematický popis životního prostředí daného jedince (obvykle to bývá převrácená hodnota objektivní funkce – čím je větší, tím je potomek vhodnější pro dané okolní podmínky).

Schéma – množina genů v chromozomu, které mají jisté specifické hodnoty. V “neuronových” aplikacích se takováto množina chápe jako skupina genů, která je schopna se za určitých podmínek chovat “jako jeden muž” a vytvořit žádaný efekt.

Obsah genu je variabilní. Nejjednodušší je, známe-li obsah genu v binární podobě, např. 01011101, avšak lze použít i dekadický tvar.

Genetický algoritmus

Vlastní algoritmus genetické optimalizace je cyklus, v němž se vytvářejí noví potomci, kteří se použijí jako rodiče v dalším cyklu. Po každém cyklu se vyhodnocuje vhodnost potomka a na základě toho se buď pokračuje v hodnocení, nebo se končí. Podobně jako u neuronových sítí, kde se cyklus nazývá *epocha*, má i zde výše zmíněný cyklus svůj název, a to (jak se dalo očekávat) **generace**.

Vlastní cyklus neboli schéma genetického algoritmu se skládá z těchto kroků:

1. navržení genetické struktury
2. inicializace
3. ohodnocení
 - 3.1. konverze genotypu na fenotyp
 - 3.2. ohodnocení objektivní funkce
 - 3.3. konverze objektivní funkce na vhodnost
 - 3.4. konverze vhodnosti na selekci rodičů
4. volba rodičů
5. reprodukce
6. mutace
7. zpět na 3.

Popišme si nyní jednotlivé kroky tohoto algoritmu podrobněji:

Navržení genetické struktury

Při návrhu genetické struktury je důležité, jak budou reprezentovány jednotlivé alely a jak budou v jednotlivých chromozomech rozmístěny (reprezentace alel a jejich rozmístění ovlivňují výkonnost příslušného algoritmu).

Při **reprezentaci** se obvykle vychází ze dvou možností. První je vyjádření pomocí *binárních hodnot*, což je dost rozšířený přístup. Pak je chromozom tvořen řetězem alel a fenotyp je počítán jako dekadická hodnota binárního genu – řetězce. Tento přístup má mimo jiné dvě výhody. Jednak je binární

kódování vlastní všem PC, a algoritmy jsou tudíž poněkud jednodušší, jednak jsou teoremy, které pojednávají o různých algoritmech, lépe dokazatelné.

V případě použití *dekadických číslic* je to trochu složitější. Lze použít různé přístupy – např. že každá alela bude nabývat jen dekadických parametrů.

Další nezanedbatelnou věcí je **pozice genů** v chromozomu. Standardní metoda dělení chromozomů (ne vždy používaná, jak uvidíme později) je dělení na dvě poloviny, z nichž každá se daruje jednomu potomku. Pokud jsou geny zastupující důležité kooperující parametry, daleko od sebe, dochází při jejich dělení k znehodnocení jejich celkové účinnosti.

Představme si, že oba rodiče mají geny, jejichž struktura popisuje “chytrost”. Ty jsou v chromozomu rozházeny a při jejich dělení půl na půl může dojít k jejich odtržení, a tak výsledkem jsou průměrně chytří potomci. Pokud by však byly geny blízko sebe, pak by mohl jeden potomek obdržet “chytře” geny a druhý bohužel jen ten zbytek. Měli bychom pak dva potomky, jeden by byl asi geniální a ten druhý hloupoučký, což je základním principem přirozeného výběru druhů – chytrý přežije a hloupý zahyne (alespoň v přírodě to tak platí). Je tedy určitě vhodné umístit důležité geny tak, aby pravděpodobnost jejich rozdělení byla co nejmenší.

Inicializace

To je v podstatě primitivní proces, který náhodným způsobem (pomocí generátoru náhodných čísel) založí populaci prvotních rodičů.

Ohodnocení

V tomto kroku se postupně provedou následující operace:

Konverze genotypu na fenotyp je pouhý přepočítání, např. při použití binárních genů na dekadickou hodnotu (máme-li chromozom 00001100, jeho fenotyp je 12).

Ohodnocení objektivní funkce – zde se počítá výsledek objektivní funkce (v případě neuronových sítí chyba za jednu epochu). Obvykle se předpokládá, že daná funkce je ovlivňována všemi geny zhruba stejně významně. Pokud by se vyskytl nějaký gen či skupina genů s vysokou “prioritou” ovlivňování, pak by genetický algoritmus měl horší výkonnost.

Konverze objektivní funkce na vhodnost se skládá ze dvou operací. Objektivní funkci musíme nejprve zkonvertovat na tzv. *hrubou vhodnost* (ta se dá počítat mnoha způsoby). Takto získanou hrubou vhodnost musíme ještě zkonvertovat na tzv. *jemnou vhodnost*, a to z následujícího důvodu: Pokud bychom použili jen hrubou vhodnost, z prvotní vygenerované populace by vyšli jako vítězové jen ti potomci, jejichž vhodnost významně převyšuje vhodnost ostatních, a tak by došlo k tvorbě pouze silných potomků. To však v konečném důsledku znamená konec přirozeného výběru, protože by pak nebylo z čeho vybírat...

Úpravou hrubé vhodnosti na jemnou se dosáhne toho, že příliš “schopným” jedincům je vhodnost snížena, zatímco průměrné hodnoty zůstávají nedotčeny.

Konverze vhodnosti na selekci rodičů pak (číselnou hodnotou) určí, který jedinec (a kolikrát) bude vybrán k vytvoření potomka.

Volba rodičů

Jakmile provedeme selekci rodičů, můžeme přistoupit k jejich výběru. V dřevních dobách genetických algoritmů se tento problém řešil tak, že se vytvořila "ruleta" s pozicemi, jejichž pravděpodobnosti byly úměrné vhodnosti, a pak se přistoupilo k náhodnému výběru. Vzhledem k příliš velké roli náhody však docházelo k degradaci účinnosti vlastního genetického algoritmu.

Mnohem lepší přístup je proto vytvořit pole, které se nejprve zaplňuje rodiči s nejvyšší selekcí (jedinci se selekcí 3 budou ve 3 políčkách), poté rodiči s menší selekcí atd. až po jedince se selekcí menší než 1, kteří se do pole vybírají náhodně. Z tohoto pole se pak budou v dalším kroku vytvářet noví potomci – budoucí rodiče.

Reprodukce

Při reprodukci konečně dochází k dělení rodičovských chromozomů a vytváření potomků jejich spojením. Jednoduše řečeno, každý rodič se na tom zúčastní polovinu své genetické výbavy. Jak už bylo dříve řečeno, je dobré mít skupiny vhodných genů pohromadě a tím snížit pravděpodobnost jejich rozdělení. Ta se snižuje také tak, že se náhodně vygeneruje pozice "přestřihnutí" chromozomu. Od této pozice se v případě dělení půl na půl odečte na každou stranu stejný počet genů tak, jako by genový řetězec byl uzavřený prsten. Tuto metodu se doporučuje ještě střídavě kombinovat s dělením na třetiny.

Mutace

Mutace jsou nezbytnou součástí genetických algoritmů. Díky jim lze mnohdy najít nové jedince, kteří lépe vyhovují okolním podmínkám, a tak zkvalitnit jak genetický proces, tak množinu jedinců – budoucích rodičů. Volba míry a pravděpodobnosti mutace genů je velmi ošemetný problém, který při necitlivém řešení může způsobit i katastrofu v genetické množině.

V podstatě se dají zvolit dvě základní cesty, jak k mutaci přistupovat. První cestou jsou mutace s malým účinkem, ale s vysokou četností výskytu. Vzhledem k tomu, že tento typ mutací přináší zpravidla menší změny, než jaké jsou požadovány (je potřeba nový materiál v genetické množině), je tento přístup nevhodný. Aby byl vliv mutací na průběh genetického algoritmu výrazný, je potřeba volit menší četnost jejich výskytu, ale s větším účinkem (druhá metoda).

Vlastní mutace se provádí v cyklu pomocí generátoru náhodných čísel tak, že se náhodné číslo porovnává s požadovanou četností výskytu mutací. Pokud je náhodné číslo menší, pak dojde k *mutaci genu* (součin již vygenerovaného náhodného čísla a dalšího náhodného čísla se přičte k již existujícímu genu – gen zmutuje). To lze samozřejmě dále modifikovat.

Povšimněme si, že genetické algoritmy souvisejí s neuronovými sítěmi – a to prostřednictvím jejich vah. Každý spoj má svou váhu (alelu), přičemž za geny lze považovat vektory vah vstupujících do každého neuronu. Při použití genetických algoritmů pro optimální start neuronové sítě jde o vytvoření chromozomů (náhodné nastavení vah, lze použít také simulované "žhání") a jejich následné "šlechtění".

V oblasti genetických algoritmů samozřejmě existuje ještě mnoho dalších přístupů k řešení mutací, výběru rodičů atd.

Ukázkové příklady

Pojďme však již od šedé teorie k “zelenému stromu života”. Pokud jste si při čtení předchozího textu neustále kladli otázku “K čemu je to všechno vlastně dobré?”, snad vám na ni alespoň částečně odpovědí následující příklady.

Hledání extrému funkce

Jako příklad použití popsaných postupů uvedeme hledání globálního extrému komplikovanější funkce. Výraz *globální extrém* zní čistě matematicky a může budit dojem, že jde o akademický problém, nicméně opak je pravdou. S hledáním extrémů funkcí se setkáváme každý den, např. když musíme zařídit více věcí na různých místech a snažíme se tato místa obejít po nejkratší dráze (problém obchodního cestujícího), když se snažíme nalézt nejlepší poměr mezi cenou výrobku a jeho kvalitou atd.

Pro nalezení extrému (v tomto případě globálního) funkce na obr. 2 byl použit **genetický algoritmus** podle výše popsaných pravidel. Červené body představují nejlepší nalezenou hodnotu za příslušnou generaci a zelený bod tu nejlepší (extrém byl nalezen již ve čtvrté generaci).

Zde je nutno podotknout, že s hledáním extrému na této funkci by klasické metody měly problém vzhledem k existenci několika minim a maxim okolo hlavního maxima.

Pro mnohem komplikovanější funkce, jaké jsou např. na obr. 3, je lepší použít **diferenciální evoluci**, která na takových funkcích vykazuje mnohem lepší výsledky. Obě funkce jsou pro názornost počítány pro 3D, nicméně je možné použít i pro dimenze 100, 500, 1000. Tak vysoké dimenze jsou sice nezobrazitelné, ale jsou dobré pro otestování výkonnosti daného algoritmu.

Optimální tloušťka tlakové nádoby

Diferenciální evoluci jsme použili také k vyšlechtění tlakové nádoby podle obr. 4. Z tabulky porovnávacích výsledky z různých zdrojů (pro nedostatek místa ji přetiskujeme bez vysvětlivek) plyne, že diferenciální evoluce byla nejuspěšnější.

Evoluční algoritmy lze však použít ještě daleko šířeji, prakticky k řešení každého problému – problémem obchodního cestujícího počínaje, přes např. identifikaci parametrů pro výpočet korelační dimenze (používá se při rekonstrukci chaotického atraktoru) a nalezením optimální struktury neuronové sítě konče.

Jouni Lampinen (Jouni.Lampinen@UWasa.Fi),

Ivan Zelinka (Zelinka@zlin.vutbr.cz)

Jouni Lampinen pracuje na University of Vaasa (Dept. of Information Technology) ve Finsku.

Ing. Ivan Zelinka je doktorandem na katedře automatizace FT VUT ve Zlíně.