

Neuronové sítě

Neuronové sítě jsou jednou z oblastí, které se v poslední době věnuje zvýšená pozornost. Je to zapříčiněno hlavně tím, že mají mnoho vlastností, které je možno s úspěchem využít v řadě oborů.

V tomto článku se dozvíte něco o historii, o využití, o umělém neuronu, o sítích neuronů a jejich speciálních případech jako jsou *vrstevnaté neuronové sítě* a *Kohonenovy mapy*.

Historie

Za počátek historie neuronových sítí se považuje rok 1943, kdy Warrena McCullocha a Waltera Pittse vytvořili první model neuronu.

V roce 1949 vydal Donald Hebb knihu *Organization of Behavior*, ve které bylo navrženo učící pravidlo samostatného neuronu (dnes známé jako *Hebbovské učení*). Velkou smůlou pro neuronové sítě byla diskreditační kampaň vedená Martinem Minským (mimo jiné konstruktor prvního neuropočítače) a Seymourem Papertem. Tato kampaň vyvrcholila roku 1969, kdy byl knižně vydán článek *Perceptron*, který zdůrazňoval neschopnost neuronu naučit se řešit problém XOR.

Až do roku 1982 se v této oblasti nedělo nic mimořádného a nebyly ani vypisovány granty. Jedině za podmínky mezioborovosti. V této době vydal fyzik John Hopfield dva články. Ale teprve až roku 1986 byl publikován článek o učícím algoritmu *Back-Propagation*. Tento algoritmus byl určen pro učení vrstevnatých neuronových sítí. Byl "objeven" vícekrát, protože předtím byl publikován v letech 1969, 1974, 1985, ale pokaždé "zapadl".

Velkým úspěchem byl systém *NETtalk*, který umožňoval převod anglického psaného textu do mluvené angličtiny a dosahoval lepších výsledků než *DECTalk*, na jehož vývoji se strávilo několik let - převážně formulováním složitých pravidel.

Využití neuronových sítí

Stručně řečeno se neuronové sítě používají všude tam, kde nám nevadí případná chyba (úspěšnost neuronových sítí je cca 95%) a kde přesný algoritmus je náročný, ať už na finanční prostředky (hardware, experti) nebo čas (nutnost rychlých rozhodnutí).

Existuje několik typů neuronových sítí. A každý z nich je určen pro něco jiného. Zatímco samostatný neuron umí rozlišit prvky do dvou (lineárně separabilních) skupin, vrstevnaté neuronové sítě umí rozdělit prvky do libovolného množství skupin a tyto skupiny ani nemusí být lineárně separabilní. Kohonenova síť je určena spíše pro odstraňování šumů, pro rozložení obrazu na více částí, pro výběr reprezentantů (např. pro zmenšování počtu barev či naopak obarvování černobílých filmů), pro kompresi, ...

Nejvíce se v současné době využívají vrstevnaté neuronové sítě. Jedním z využití je již zmiňovaný *NETtalk*, který se skládal z 203 vstupních neuronů (29×7), 80 skrytých a 26 výstupních neuronů.

Další možností jsou predikce v letecké dopravě (Systém AMT, který optimalizuje rezervace letenek). V první fázi systém předpovídá poptávku po volných místech (zde je použita ona neuronová síť) a ve druhé fázi navrhuje rozložení letů (to již tradičními způsoby).

Jinou možností je filtrace EKG signálu. Americká firma Hecht-Nielsen Neurocomputer vyvinula síť (50-12-1). Jako vstup slouží 5120 údajů, které se postupně předkládají. Existují systémy, které na základě EKG křivky umožňují odhad nemoci. Další využití v lékařství je např. při podávání antibiotik, kdy neuronová síť pracovala 4× přesněji než lékaři (např. nepředepisovala antibiotika, na které mohl být pacient alergický). Lékařům sloužila pouze jako konzultant.

Asi nejlákavější je využití ve finančnictví. Predikuje se změna kurzů měn, změna kurzů akcií. Zatímco při použití klasických metod je úspěšnost cca 55%, při použití neuronových sítí se dosahuje až 75%. Existují práce, které dokázaly (při využití i jiných informací než pouhých hodnot kurzu) predikovat s úspěšností cca 85%. Dále se predikuje spojení či krach podniků.

Jiným využitím ve finančnictví může být testování, zda daný klient bude schopen splácet půjčku, nebo dokonce navrhnout výšky půjčky a doby splácení. Zajímavý nápad je také využít dat ze Španělska v době krize bank (1977-85), kdy zbankrotovalo 58 ze 108 bank a provést analýzu v současnosti.

Za zmínku stojí také situace, kdy se neuronová síť nainstalovala a učila se reagovat na nějaké podněty od "učitele - Experta". Poté, co byla dostatečně naučena, dokázala reagovat správně i na neznámé podněty (využití např. v atomových elektrárnách, pivovarech, pokusy při řízení automobilů, kdy síť reagovala podobně jako "její" řidič - jezdila agresivně plyn, brzda, nebo naopak opatrně ...).

Často používané typy sítí

Obecné poznámky

Síť je určena několika vlastnostmi. Svoji strukturou (propojením jednotlivých neuronů), způsobem vybavování a způsobem učení.

Obecně můžeme říci, že struktura může být roztodivná, ale my se budeme zabývat pouze necyklickými grafy.

Způsob vybavování neuronu je také podobný. Na vstup se předloží nějaké hodnoty. Neuron v okamžiku, kdy mají všechny potřebné údaje, vypočtou svůj výstup a pošlou jej dál. Výstupem celé sítě je výstup výstupních neuronů.

Učení sítě se děje např. za pomoci "učitele", kdy tento kárá a nebo naopak chválí za špatný, či dobrý výsledek. Znamená to, že ke každému vstupu musí existovat požadovaný výstup.

Samotné učení probíhá tak, že se síť nejdříve náhodně nainicializuje. Poté se postupně předkládají vzory. Pokud síť určí správný výstup, nic se neděje, pokud určí špatný, upraví se váhy jednotlivých neuronů. Pokud jsme s celkovým výsledkem spokojeni (např. jsme předložili již všechny vzory a byly spočteny správně), učení ukončíme, jinak předložíme další vzor.

Neuron

Neuron je základní jednotka neuronových sítí. Formálně má neuron n vstupů x_1, x_2, \dots, x_n , které jsou ohodnoceny vahami w_1, w_2, \dots, w_n , které určují jejich propustnost. Každý neuron má práh θ . Vnitřním potenciálem neuronu chápeme:

$$\xi = \sum_{i=1}^n w_i x_i$$

V závislosti na hodnotě ξ a θ určujeme výstup neuronu. Neuron může mít několik kombinací výstupu. Např. $\{0,1\}$, $\{-1,1\}$, nebo $\langle -1, 1 \rangle$, $\langle 0, 1 \rangle$, ale i mnoho jiných.

Výstup se často počítá pomocí sigmoidy

$$y = \frac{1}{1 + e^{-\xi}},$$

jelikož má jednoduchou derivaci.

Jak již bylo řečeno, umí samostatný neuron rozdělit prvky množiny na dvě disjunktí množiny (jsou-li lineárně separabilní) tím, že n -rozměrným prostorem (n je počet vstupů) proloží nadrovinu.

Učení neuronu. Učení neuronu probíhá v 5 krocích:

- 1. krok** Inicializace vah na malé náhodné hodnoty

2. krok Předložení vstupu $X = x_1 x_2 \dots x_n$ a odpovídajícího výstupu $d = -1$ nebo $d = 1$.

3. krok Výpočet aktuálního výstupu

$$y = f \left(\sum_{i=1}^n w_i x_i - \theta \right),$$

kde f je nějaká funkce – např. sigmoida.

4. krok Upravení vah, pokud výstup y je různý od očekávaného výstupu s

$$w_i(t+1) = w_i(t) + \eta \cdot |d - y| \cdot x_i(t),$$

kde $w_i(t)$ je hodnota váhy i -tého vstupu v čase t a η je tzv. parametr učení.

5. krok Ukončení, nebo opakování od 2. kroku, pokud nejsme spokojeni s výsledkem.

Tomuto učení se také říká *perceptronové* nebo *hebbovské*.

Problém XOR . Za zmínku zde určitě stojí tzv. problém XOR, který na dlouhou dobu zasadil neuronovým sítím těžkou ránu. Jestliže si nakreslíte rozmístění hodnot funkce XOR, zjistíte, že není možno tyto čtyři body odseparovat. Neuron se tedy v takovém případě neustále učí, ale nikdy se danou věc nenaučí.

Vrstevnatá síť

Vrstevnatá neuronová síť je síť, kde jsou neurony rozděleny do několika vrstev. Spojené jsou neurony mezi dvěma sousedními vrstvami, ale nejsou spojeny mezi sebou ve vrstvě. První vrstvě se říká *vstupní*, poslední se říká *výstupní* a ostatním se říká *skryté*.

Často se topologie vrstevnaté neuronové sítě zapisuje zkráceně. Např. 203-80-26 znamená, že síť má na vstupu 203 neuronů, jednu skrytou vrstvu s 80 neurony a ve výstupní vrstvě 26 neuronů.

Existuje věta, která hovoří, že na vyřešení libovolného problému stačí síť s dvěma skrytými vrstvami – v praxi se ale často používá vrstev méně. Obecně se doporučuje používat síť s menším počtem vrstev, jelikož se rychleji učí. Naopak síť s více vrstvami umí mnohem lépe zobecňovat.

Algoritmus vybavování byl naznačen již dříve. Neurony ve vstupní vrstvě předají signál do první skryté vrstvy. Neurony v této první skryté vrstvě spočítají svůj výstup a pošlou jej jako vstup neuronům ve druhé vrstvě. A tak se pokračuje dále.

Učení Back-Propagation . Toto učení je založeno na tzv. gradientní metodě. Existuje spousta vylepšení tohoto způsobu, z nichž většina pracuje rychleji. Jelikož se jedná o učení s učitelem, můžeme určit míru chyby – chybovou funkci.

$$E = \frac{1}{2} \sum_j (d_j - y_j)^2,$$

což je kvadrát vzdálenosti mezi očekávaným a skutečným výstupem; y_j znamená očekávaný výstup j -tého vzoru.

Když provedeme parciální derivace (divný to výraz v tomto jinak krásném textu) podle vah, dostaneme

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial \xi_j} \frac{\partial \xi_j}{\partial w_{ij}}$$

Použijeme-li sigmoidu, získáme

$$\delta_j = f'(\xi_j)(d_j - y_j) = (1 - y_j)y_j(d_j - y_j),$$

kde δ je velikost změny váhy.

Ona parciální derivace nám vlastně říká, kterým směrem je nejbližší údolí.

Samotné učení tedy můžeme stručně popsat v 5 krocích:

1. krok Inicializace všech vah náhodnou malou hodnotou.

2. krok Předložení nového vzoru $X = x_1 x_2 \dots x_n$ a odpovídajícího výstupu $D = d_1 d_2 \dots d_m$. Trénovací vzory vybíráme náhodně.

3. krok Výpočet aktuálních výstupů sítě.

4. krok Adaptace vah.

Novou hodnotu váhy spočteme jako

$$w_{ij}^s(t+1) = w_{ij}^s(t) + \eta \delta_j^{s+1} y_i^s,$$

kde $w_{ij}^s(t)$ znamená váhu mezi i -tým a j -tým neuronem v s -té vrstvě v t -tém kroku adaptace. η je parametr učení. δ_j^s jsou chyby neuronů v uvažované vrstvě a ξ_i^s je buď vstup neuronu uvažované vrstvy nebo hodnoty vstupu.

Jedná-li se o poslední vrstvu, potom $s = M$ a

$$\delta_j^M = y_j(1 - y_j)(d - y_j).$$

V ostatních vrstvách platí:

$$\delta_j^s = y_j^{s+1}(1 - y_j^{s+1}) \sum_{k=1}^{p^{s+1}} \delta_k^{s+1} w_{jk}^{s+1},$$

kde $s = M - 1, \dots, 1$ a k prochází všemi neurony v $s + 1$ vrstvě, tedy ty, které jsou pod j -tou vrstvou a tedy blíže k výstupu.

5. krok Opakování procesu učení. Pokud je celková chyba větší než ε , pokračujeme krokem 2. Jinak skončíme.

Kohonenovy sítě

Jsou sice zajímavé, ale nestihl jsem se k nim dostat a asi se k nim nestihnu dostat ani na přednášce.