

Taxonomie výpočetních modelů neuronových sítí: Od subregulárních jazyků k super-turingovským výpočtům

Jiří Šíma*
Ústav informatiky AV ČR,
Pod Vodárenskou věží 2, 182 07 Praha 8
sima@cs.cas.cz

1 Úvod

1.1 Výpočetní teorie neuronových sítí

Tzv. (*umělé*) *neuronové sítě* představují velmi zjednodušené matematické modely nervových systémů živých organismů. Ve snaze pochopit a modelovat biologické a kognitivní funkce lidského mozku se tyto modely stále zdokonalují, aby věrněji odrážely současné neurofyziologické poznatky (Gerstner, Kistler, 2002). Na druhé straně tohoto úsilí stojí inženýři, kteří modely neuronových sítí modifikují, aby je mohli využít pro řešení praktických úloh z umělé inteligence (Haykin, 1999). Je zajímavé, že již konstruktéři prvních počítačů chtěli napodobit lidskou inteligenci, a proto se při své práci inspirovali funkcí mozku (von Neumann, 1951, 1956). Např. první neuropočítač (Minsky, 1954) byl zkonstruován v roce 1951 nedlouho poté, co byl vyvinut první konvenční počítač von neumannovské architektury. Neuronové sítě se dále profilovaly jako samostatný informatický obor. Hlavní předností neuronových sítí je jejich schopnost učit se z dat, resp. zobecňovat požadovanou funkci z příkladů. Neuronové sítě jako netradiční výpočetní prostředky inspirované přírodou byly úspěšně aplikovány v oblastech, kde je návrh exaktních algoritmů problematický, např. rozpoznávání obrazců, řízení, predikce, rozhodování, analýza a transformace signálů, detekce chyb, expertní systémy, apod. Současné komerční programy pro analýzu a zpracování dat (např. *Matlab*, *Statistica*) obvykle obsahují moduly pro práci s neuronovými sítěmi, které představují velmi účinný nástroj v mnoha praktických aplikacích strojového učení.

Úvod do oboru neuronových sítí lze najít v 9. kapitole prvního dílu této publikace, zatímco s pokročilejšími technikami se lze seznámit v 7. a 9. kapitole čtvrtého dílu. Cílem této kapitoly je prezentovat taxonomii formálních výpočetních modelů neuronových sítí v rámci jednotného formalismu spolu s přehledem jejich výpočetních vlastností. Vycházíme při tom z přehledové práce (Šíma, Orponen, 2003c). Výpočetní potenciál a limity konvenčních počítačů jsou studovány např. pomocí Turingových strojů, u kterých je příp. omezena doba výpočtu či velikost paměti (srovnej se 7. kapitolou třetího dílu této publikace). Analogicky bylo v posledních 15 letech dosaženo mnoho fundamentálních výsledků týkajících se schopnosti neuronových sítí implementovat univerzální výpočty (Anthony 2001; Floréen, Orponen, 1994; Orponen, 1994; Parberry, 1994; Roychowdhury a kol., 1994; Siegelmann 1999a; Siu a kol., 1995; Šíma, Neruda, 1996; Wiedermann, 1994). Výpočetní síla neuronových sítí se obvykle studuje tak, že se různé modely sítí porovnávají nejen mezi

*Tato práce vznikla za částečné podpory projektů 1ET100300517 programu Informační společnost, GA ČR P202/10/1333 a výzkumného záměru AV0Z10300504.

sebou, ale především s tradičními výpočetními modely a popisnými prostředky jako jsou konečné automaty, regulární výrazy, gramatiky, Turingovy stroje, booleovské obvody apod. Cílem tohoto přístupu je zjistit, co lze v principu nebo efektivně spočítat pomocí jednotlivých modelů neuronových sítí, resp. jak implementovat požadovanou funkci optimálním způsobem. Hlavním technickým nástrojem tohoto výzkumu je výpočetní teorie složitosti (Balcázar a kol., 1995; Papadimitriou, 1994), příp. formální teorie jazyků (Hopcroft, Ullman, 1979; Chytil, 1984), proto v příloze připomínáme technické definice některých složitostních tříd, i když předpokládáme, že čtenář je seznámen se základními abstraktními výpočetními modely jako jsou konečné automaty, Turingovy stroje apod.

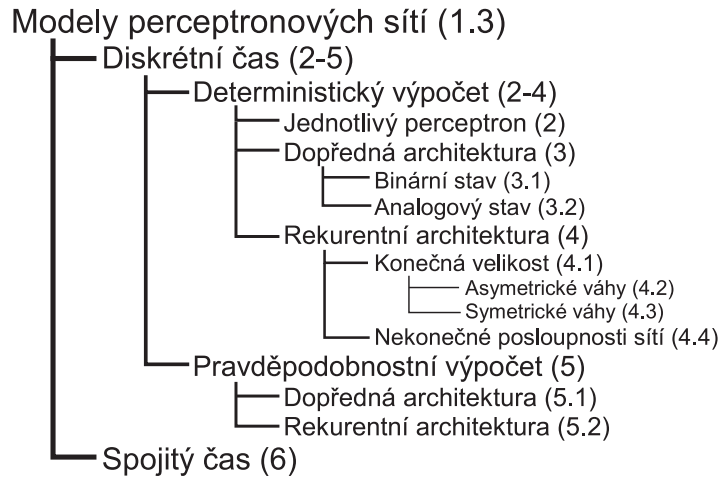
Hluboký teoretický vhled a porozumění principům a možnostem neurovýpočtů je předpokladem pro volbu adekvátního modelu a parametrů neuronové sítě při řešení daného problému nebo pro návrh efektivnějších učících algoritmů. Vzhledem k tomu, že modely neuronových sítí jsou inspirovány neurofyzilogickými poznatky, teoretické výsledky mohou kromě praktického využití prohloubit naše chápání výpočetních aspektů mentálních procesů. Navíc neuronové sítě obohacují tradiční repertoár formálních výpočetních modelů, které jsou obvykle parametrizovány časem a prostorem (časová a paměťová složitost), o nové zdroje efektivního počítání jako jsou např. *analogové stavy* (Siegelmann, 1999a), *spojitý čas* (Orponen, 1997c), *energie* (Šíma, 2003; Uchizawa a kol., 2006) apod. Protože např. tradiční Turingův stroj tyto parametry nezná a příslušné modely neuronových sítí mnohdy formálně implementují super-turingovské výpočty, neuronové sítě v tomto případě aspirují na to, aby se staly referenčními modely při studiu těchto alternativních výpočetních zdrojů.

Výpočetní aspekty neuronových sítí lze studovat ve třech hlavních směrech: efektivní vytváření, resp. adaptace reprezentace neuronové sítě (*složitost učení a generalizace*), její paměťová náročnost (*deskriptivní složitost*) a efektivní odezva neuronové sítě na její vstup (*výpočetní síla*). V této kapitole se zaměříme pouze na deskriptivní složitost a výpočetní sílu neuronových sítí, zatímco úplně opomineme důležité výsledky o složitosti učení, které by zasluhovaly samostatný přehled (Anthony, Bartlett, 1999; Roychowdhury a kol., 1994; Šíma, 2002; Vidyasagar, 1997). Dále se omezíme jen na *digitální* výpočty, jejichž vstupy a výstupy jsou ve své podstatě diskrétní veličiny, i když jsou často kódovány pomocí analogových stavů neuronů a vlastní výpočet neuronové sítě může pracovat s reálnými čísly. Naopak výpočetní schopnosti a efektivita modelů s *analogovými* vstupy se studují v teorii aproximace funkcí pomocí neuronových sítí, kterou se zabývá 8. kapitola čtvrtého dílu této publikace.

1.2 Taxonomie formálních výpočetních modelů neuronových sítí

Nejprve si připomeneme obecnou definici modelu neuronové sítě, která ve své neurčitosti pokrývá většinu publikovaných a používaných modelů a kterou budeme dále konkretizovat při specifikaci jednotlivých modelů. *Neuronová síť* se skládá z s jednoduchých výpočetních *jednotek*, resp. *neuronů*, z množiny $V = \{1, \dots, s\}$, kde počet neuronů $s = |V|$ se nazývá *velikost* sítě. Některé z těchto jednotek mohou sloužit jako externí vstupy nebo výstupy, proto předpokládáme, že síť má n *vstupních* a m *výstupních* neuronů. Tyto jednotky jsou hustě propojeny do orientovaného grafu, který reprezentuje *architekturu* (resp. *topologii*) sítě, v níž je každá hrana (i, j) vedoucí z neuronu i do j ohodnocena reálnou (*synaptickou*) *váhou* $w(i, j) = w_{ji} \in \mathbf{R}$. Absence spoje v architektuře odpovídá nulové váze mezi příslušnými neurony a naopak.

Výpočetní dynamika neuronové sítě určuje vývoj reálného *stavu* (*výstupu*) $y_j^{(t)} \in \mathbf{R}$ pro každý neuron $j \in V$ jako funkci času $t \geq 0$. Souhrnně je tak definován *stav sítě* $\mathbf{y}^{(t)} = (y_1^{(t)}, \dots, y_s^{(t)}) \in \mathbf{R}^s$ v každém časovém okamžiku $t \geq 0$. Na začátku



Obrázek 1: Taxonomie modelů perceptronových sítí.

výpočtu se síť nachází v *počátečním stavu* $\mathbf{y}^{(0)}$, který může kódovat externí vstup pomocí počátečních stavů vstupních neuronů. Stav sítě je typicky aktualizován tak, že podmnnožina neuronů načte své vstupy od incidentních neuronů přes příslušné zvážené spoje a transformuje je na své aktuální výstupy. Globální výstup sítě je odečten ze stavů výstupních neuronů na konci nebo někdy i v průběhu výpočtu. Jak již bylo řečeno, budeme se zabývat digitálními výpočty, a proto předpokládáme *binární* externí vstupní a výstupní hodnoty.

Výpočetní modely neuronových sítí lze klasifikovat podle typu architektury na dopředné (acyklické) nebo rekurentní (cyklické), podle časové dynamiky na diskrétní nebo spojitě síť, podle typu výpočtu na deterministické nebo pravděpodobnostní síť, podle domény stavů neuronů na binární nebo analogové síť, podle omezení na váhové parametry na symetrické nebo asymetrické síť, podle typu vstupního protokolu na konečné síť nebo nekonečné posloupnosti sítí atd. Kombinací těchto omezení dostáváme bohatou taxonomii modelů neuronových sítí, které mají různou výpočetní sílu. V našem přehledu se zaměříme především na tradiční perceptronové síť a v závěru jen zmíníme modely, které využívají jiných typů jednotek. Kromě RBF (Radial Basis Function) a WTA (Winner-Take-All) jednotek se jedná o biologicky více plauzibilní pulzní (spiking) neurony, které používají časového kódování informace a jsou v poslední době předmětem intenzivního výzkumu. Struktura této kapitoly částečně sleduje taxonomii modelů perceptronových sítí, která je znázorněna na obrázku 1, kde jsou také vyznačeny odkazy na příslušné odstavce.

1.3 Perceptronové síť

Nejpopulárnějším modelem neuronových sítí jsou perceptronové síť. Jak je zvykem, termín *perceptron* je zde chápán obecněji než v původní práci Rosenblatta (1958) a pokrývá i typy jednotek, které z něj byly později odvozeny, např. sigmoïdální jednotky. V této kapitole se zaměříme zejména na výpočetní aspekty *diskrétní* perceptronové sítě, která aktualizuje svůj stav jen v diskrétních časových krocích $t = 1, 2, \dots$. V čase $t \geq 0$ se nejprve vyhodnotí *excitace* (*vnitřní potenciál*) každého neuronu $j \in V$ jako zvážený součet jeho vstupů:

$$\xi_j^{(t)} = \sum_{i=0}^s w_{ji} y_i^{(t)}. \quad (1)$$

Uvedená suma navíc obsahuje absolutní člen, tzv. *bias* $w_{j0} \in \mathbf{R}$, který lze chápat jako váhu přidaného formálního jednotkového vstupu $y_0^{(t)} \equiv 1$. Připomeňme, že některé váhy w_{ji} ($1 \leq i \leq s$) ve vzorci (1) mohou být nulové, což značí absenci spoje z neuronu i do neuronu j . V následujícím časovém kroku $t + 1$ neurony $j \in \alpha_{t+1}$ z vybrané podmnožiny $\alpha_{t+1} \subseteq V$ počítají svůj nový výstup $y_j^{(t+1)}$ pomocí *aktivační (přenosové) funkce* $\sigma : \mathbf{R} \rightarrow \mathbf{R}$:

$$y_j^{(t+1)} = \sigma \left(\xi_j^{(t)} \right) \quad \text{pro } j \in \alpha_{t+1}, \quad (2)$$

zatímco zbývající jednotky $j \notin \alpha_{t+1}$ své stavy nemění, tj. $y_j^{(t+1)} = y_j^{(t)}$ pro $j \notin \alpha_{t+1}$. Tímto způsobem je určen nový stav sítě $\mathbf{y}^{(t+1)}$ v čase $t + 1$.

Perceptronové sítě s *binárními stavy* $y_j \in \{0, 1\}$ (zkráceně *binární sítě*) obvykle používají *Heavisideovu* aktivační funkci

$$\sigma(\xi) = \begin{cases} 1 & \text{pro } \xi \geq 0 \\ 0 & \text{pro } \xi < 0 \end{cases} \quad (3)$$

a odpovídající jednotky se také nazývají *prahová hradla*. V případě potřeby je možné binární stavové hodnoty $\{0, 1\}$ nahradit bipolárními $\{-1, 1\}$ nebo dokonce obecnějšími diskretními doménami, což zásadním způsobem nemění velikost váhových parametrů (Parberry, 1994). Váhy binárního perceptronu lze při zachování jeho funkce také upravit tak, aby excitace (1) byla nenulová pro všechny vstupy.

Na druhou stranu sítě s *analogovými stavy* (zkráceně *analogové sítě*) aproximují Heavisideovu funkci (3) nějakou spojitou sigmoidální aktivační funkcí, např. *saturovanou lineární* funkcí

$$\sigma(\xi) = \begin{cases} 1 & \text{pro } \xi \geq 1 \\ \xi & \text{pro } 0 < \xi < 1 \\ 0 & \text{pro } \xi \leq 0 \end{cases} \quad (4)$$

nebo *logistickou (standardní) sigmoidou*

$$\sigma(\xi) = \frac{1}{1 + e^{-\xi}} \quad (5)$$

a odpovídající jednotky se také nazývají *sigmoidální hradla*. Tedy výstupy ze sigmoidálních hradel jsou obecně reálná čísla, např. $y_j \in [0, 1]$. Pro potřeby výpočtu booleovských funkcí jsou analogové stavy y_j výstupních neuronů j interpretovány jako binární výstupy 0, resp. 1, se *separací* $\varepsilon > 0$, jestliže $y_j \leq h - \varepsilon$, resp. $y_j \geq h + \varepsilon$, pro danou pevnou prahovou hodnotu $h \in \mathbf{R}$.

2 Jednotlivý perceptron

Jednotlivý perceptron s n externími binárními vstupy zřejmě počítá booleovskou funkci $f : \{0, 1\}^n \rightarrow \{0, 1\}$ n proměnných. Je všeobecně známo, že ne všechny booleovské funkce lze implementovat pomocí jednoho perceptronu. Prominentním příkladem je parita (XOR), která se nabývá 1, právě když je počet vstupních bitů s hodnotou 1 lichý (Minsky, Papert, 1969). Naopak booleovské funkce, které lze počítat jedním perceptronem, se nazývají *lineární prahové funkce* (Muroga, 1971). Tato důležitá třída funkcí obsahuje základní logické funkce AND, OR, NOT a je uzavřená na negaci funkce a negaci vstupních proměnných, tj. negaci libovolné booleovské lineární prahové funkce s příp. negací některých jejích proměnných lze opět počítat pomocí jednoho perceptronu. Počet booleovských lineárních prahových funkcí n proměnných je řádově $2^{\Theta(n^2)}$. Horní odhad $2^{n^2 - n \log_2 n + O(n)}$ pochází již od

Schläfliho (1901), zatímco přesnější přiléhavý dolní odhad $2^{n^2 - n \log_2 n - O(n)}$ byl dokázán o dost později (Kahn a kol., 1995). To znamená, že jen velmi malý zlomek ze všech 2^{2^n} booleovských funkcí je možné počítat pomocí jednoho perceptronu. Navíc je obecně algoritmicky těžké (co-NP-úplné) rozhodnout, zda konkrétní booleovská funkce daná v disjunktivní nebo konjunktivní normální formě je lineární prahová (Hegedüs, Megiddo, 1996).

Také byla zkoumána deskriptivní složitost booleovských lineárních prahových funkcí. Je známo, že každou takovou funkci lze implementovat perceptronem, který má jen *celočíslné* váhy (Minsky, Papert, 1969), a počet bitů, které jsou potřeba (Håstad, 1994) a stačí (Muroga a kol., 1961) na reprezentaci jedné celočíselné váhy perceptronu s n vstupy, je řádově $\Theta(n \log n)$. Tyto odhady mají velký význam pro hardwarovou příp. softwarovou implementaci perceptronu.

3 Dopředné síť

Architekturu *dopředné (vrstvené, vícevrstvé, acyklické)* síť tvoří acyklický graf. To znamená, že jednotky v dopředné síti lze vždy jednoznačně seskupit do nejkratší posloupnosti $d+1$ po dvou disjunktích *vrstev* $\alpha_0, \dots, \alpha_d \subseteq V$ tak, aby neurony v libovolné vrstvě α_t byly napojeny jen na neurony v následujících vrstvách α_u , $u > t$. Obvykle se první, tzv. *vstupní vrstva* α_0 , která se skládá z n externích vstupů, nepočítá do počtu vrstev a velikosti sítě. Podobně poslední, tzv. *výstupní vrstva* α_d , je tvořena m výstupními neurony. Skryté jednotky se pak nachází v mezilehlých, tzv. *skrytých vrstvách* $\alpha_1, \dots, \alpha_{d-1}$. Počet vrstev d vyjma vstupní vrstvy se nazývá *hloubka sítě* (např. dvouvrstvé síť mají hloubku 2). Výpočet sítě postupuje vrstvou po vrstvě směrem od vstupní vrstvy přes skryté vrstvy až k výstupní vrstvě. Na počátku jsou stavy vstupních neuronů nastaveny na externí vstup. V obecném kroku předpokládáme, že výstupy všech neuronů jsou určeny až do jisté vrstvy $t < d$ včetně. Stavy neuronů v následující vrstvě α_{t+1} jsou vypočteny paralelně podle vzorců (1) a (2). Nakonec stavy výstupních neuronů z α_d představují výsledek výpočtu. Tímto způsobem je vyhodnocena tzv. *funkce sítě* $\mathbf{f} : \{0, 1\}^n \rightarrow \{0, 1\}^m$ v paralelním čase, který se rovná hloubce sítě d .

Shora popsaný model dopředné neuronové sítě splývá s modelem *booleovských prahových obvodů* (dále jen *obvodů*), jejichž výpočetní jednotky se obvykle nazývají (prahová) hradla. Výpočetní aspekty obvodů jsou předmětem intenzivního výzkumu v teorii složitosti booleovských obvodů (Vollmer, 1999; Wegener, 1987). Jednotlivý obvod C s n vstupy počítá (vektorovou) booleovskou funkci \mathbf{f} , která závisí na pevném počtu proměnných n . Pro potřeby univerzálních výpočtů nad vstupy libovolné délky se používá *nekonečná posloupnost (rodina)* obvodů $\{C_n\}$, která obsahuje jeden obvod C_n pro každou délku vstupu $n \geq 0$. V tomto kontextu jsou pak míry složitosti obvodu jako velikost $S(n)$, hloubka $D(n)$, maximální váha $W(n)$ atd. vyjádřeny jako funkce délky vstupu n . Například pro posloupnost obvodů *polynomiální velikosti* je funkce $S(n)$ omezena nějakým polynomem $O(n^c)$ (kde c je konstanta) nebo posloupnost obvodů *konstantní hloubky* splňuje $D(n) \leq c$ apod.

Nekonečné posloupnosti obvodů mohou být podle shora uvedené definice obecně *neuniformní* v tom smyslu, že neexistuje jejich konečný popis, který by dával do souvislosti obvodu pro různé délky vstupu. To znamená, že takové nekonečné posloupnosti obvodů mohou implementovat funkce, které nelze počítat pomocí konečně popsaného algoritmu (např. Turingova stroje), a tedy mají „super-turingovské“ výpočetní schopnosti, což je triviální důsledek jejich definice. Tomu se lze vyhnout tak, že předpokládáme tzv. *uniformní* posloupnost obvodů, pro kterou existuje konečný algoritmus (pevný Turingův stroj), který pro každou délku vstupu $n \geq 0$ zkonstruuje příslušný obvod C_n z této posloupnosti. Je zřejmé, že uniformní posloupnosti obvodů jsou výpočetně ekvivalentní Turingovým strojům.

3.1 Binární dopředné sítě

V tomto odstavci se budeme zabývat výpočetními vlastnostmi binárních dopředných sítí s Heavisideovou aktivační funkcí (3).

3.1.1 Výpočetní univerzalita:

Nejjednodušší případ tohoto modelu, jednotlivý perceptron, jsme již uvažovali v odstavci 2, ze kterého je zřejmé, že prahová hradla musí být zapojena do větších obvodů, aby byla schopna implementovat všechny booleovské funkce. Konkrétně pro libovolnou vektorovou booleovskou funkci $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ lze vždy zkonstruovat prahový obvod hloubky 4, který počítá f s použitím řádově

$$s = \Theta \left(\sqrt{\frac{m2^n}{n - \log m}} \right) \quad (6)$$

hradel (Lupanov, 1972). Tato konstrukce je optimální v tom smyslu, že existují funkce, které dokazatelně vyžadují uvedený počet hradel i v případě, že povolíme neomezenou hloubku obvodu (Horne, Hush, 1994). Odpovídající dolní (Nechiporuk, 1964) a horní (Lupanov, 1961) odhad byl nejprve stanoven pro sítě s jedním výstupem a souhlasí se vzorcem (6) pro $m = 1$. Navíc Cover (1968) dokázal, že počet spojů, který je v nejhorším případě potřeba v prahovém obvodu počítajícím libovolnou booleovskou funkci, lze vyjádřit jako čtverec formule (6). Tyto výsledky znamenají, že (neuniformní) nekonečné posloupnosti prahových obvodů konstantní hloubky s exponenciálním počtem hradel (vzhledem k počtu vstupů) jsou schopny počítat libovolnou vstupně-výstupní funkci v konstantním paralelním čase.

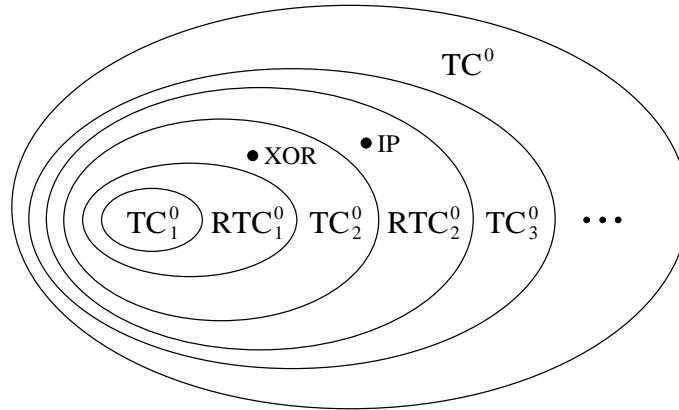
3.1.2 Polynomiální váhy:

V booleovské složitosti se kladou další omezení na parametry obvodů. Například v prahových obvodech jsou celočíselné váhové parametry obvykle omezeny polynomelem $O(n^c)$ (pro nějakou konstantu c) vzhledem k počtu vstupů obvodu n , což odpovídá $O(\log n)$ bitům na reprezentaci jedné váhy. Polynomiální váhy zřejmě snižují výpočetní sílu prahových obvodů, protože již z odstavce 2 víme, že perceptron vyžaduje v nejhorším případě exponenciální váhy. Nicméně Goldmann a kol. (1992) dokázali, že neomezené váhy v prahových obvodech lze principiálně vždy (za cenu jen polynomiálního nárůstu velikosti obvodu) nahradit polynomiálními váhami, pokud zvýšíme hloubku obvodu o jednu vrstvu. Goldmann a Karpinski (1998) dále navrhli explicitní konstrukci takového obvodu s polynomiálními váhami, u něhož polynomiální nárůst velikosti nezávisí na hloubce obvodu. Tedy u vícevrstvé perceptronové sítě lze vždy předpokládat polynomiální váhy, pokud navíc připustíme jeden výpočetní krok.

Jiná konvence předpokládá, že hodnoty booleovských proměnných vstupují do obvodu vždy spolu se svými negacemi. Za tohoto předpokladu lze každý prahový obvod upravit tak, aby obsahoval jen *kladné* váhy, zatímco jeho velikost se nejvýše zdvojnásobí a jeho hloubka a velikost vah se nezmění (Hajnal a kol., 1993).

3.1.3 Omezený vstupní stupeň hradel:

Jiné omezení se týká maximálního *vstupního stupně* hradel v obvodu, tj. maximálního počtu vstupů u jednotlivých neuronů. Obvody s omezeným vstupním stupněm modelují konvenční technologii obvodů, u kterých je počet vstupů hradla omezený. Neomezený vstupní stupeň je naopak typický pro hustě propojené neuronové sítě, které díky tomu mohou být výpočetně efektivnější. Například každý obvod velikosti s a hloubky d s maximálním vstupním stupněm 2 je možné implementovat pomocí



Obrázek 2: Možná TC^0 -hierarchie se separací malých hloubek.

prahového obvodu s neomezeným vstupním stupněm velikosti $O(s^2)$ hradel, jehož hloubka je řádově $O(d/\log \log s)$ (Chandra a kol., 1984). V neuronových sítích je tak možné v rámci polynomiální velikosti dosáhnout zrychlení paralelního výpočtu o faktor $O(\log \log n)$.

3.1.4 Polynomiální velikost a konstantní hloubka:

Univerzální konstrukci obvodu exponenciální velikosti (6) nelze prakticky realizovat pro větší počty vstupů. Na druhou stranu o žádné prakticky „zajímavé“ funkce (např. ze složitostní třídy NP) nebylo zatím dokázáno, že by ke své implementaci potřebovala více jak lineární počet hradel obvodu s neomezenou hloubkou, i když intuitivně nejsou funkce vyžadující velké obvody žádnou vzácností (Wegener, 1987). Nicméně mnoho důležitých funkcí je možné počítat pomocí prahových obvodů polynomiální velikosti a konstantní hloubky (viz odstavce 3.1.5 a 3.1.6), a tedy je lze prakticky implementovat. Navíc jsou tyto výsledky ve shodě s neurofyziologickým pozorováním, že komplikované funkce jsou v mozku realizovány s využitím jen několika málo vrstev neuronů.

Třída všech funkcí, které lze počítat pomocí prahových obvodů polynomiální velikosti a hloubky $d \geq 1$ s polynomiálními váhami, se obvykle značí TC_d^0 (alternativní značení je \widehat{LT}_d , kde LT_d bez stříšky odkazuje na odpovídající třídu s neomezenými váhami). Složitostní třída $TC^0 = \bigcup_{d \geq 1} TC_d^0$ pak obsahuje všechny funkce, které lze pomocí takových obvodů počítat v konstantním paralelním čase. Zřejmě

$$TC_1^0 \subseteq TC_2^0 \subseteq TC_3^0 \subseteq \dots \quad (7)$$

je možná TC^0 -hierarchie, která je znázorněna na obrázku 2 spolu s odpovídajícími pravděpodobnostními třídami složitosti RTC_d^0 , kterými se budeme zabývat níže v odstavci 5.1. Například třída TC_1^0 , která obsahuje všechny funkce, které lze počítat jednotlivými perceptrony s polynomiálními váhami, je jistě vlastní podtřídou třídy TC_2^0 , tj. $TC_1^0 \subsetneq TC_2^0$, protože např. paritu (XOR) nelze počítat jedním perceptronem, ale lze ji jednoduše implementovat malým obvodem hloubky 2.

Méně známý výsledek je netriviální separace $TC_2^0 \subsetneq TC_3^0$ (Hajnal a kol., 1993), která má velký význam v praktických aplikacích dopředných neuronových sítí, kde se obvykle rozhoduje o použití dvouvrstvé nebo třívrstvé sítě. Příkladem funkce, která náleží do $TC_3^0 \setminus TC_2^0$ je *booleovský skalární součin* $IP : \{0, 1\}^{2k} \rightarrow \{0, 1\}$ ($k \geq 1$), který je definován jako

$$IP(x_1, \dots, x_k, x'_1, \dots, x'_k) = \bigoplus_{i=1}^k AND(x_i, x'_i), \quad (8)$$

kde \oplus značí paritu k bitů, která se nabývá 1, právě když je počet vstupních bitů s hodnotou 1 liché. Tento výsledek znamená, že pro perceptronové sítě polynomiální velikosti s polynomiálními váhami mají třívrstvé sítě prokazatelně větší výpočetní sílu než dvouvrstvé. Odpovídající separace pro neomezené váhy je doposud otevřený problém.

Separace TC^0 -hierarchie pro hloubky větší než 3 není známá a představuje jeden z největších otevřených problémů v současné teorii složitosti. V této chvíli je dokonce myslitelné (i když nepravděpodobné), že by všechny funkce ve třídě NP mohly být počítány pomocí prahových obvodů hloubky 3 s polynomiálním počtem hradel. V takovém případě by se TC^0 -hierarchie zhroutila v hloubce 3, tj. $TC^0 \subseteq TC_3^0$.

3.1.5 Symetrické booleovské funkce:

Důležitou třídou booleovských funkcí jsou *symetrické* funkce, jejichž hodnoty nezávisí na pořadí vstupních proměnných, tj. hodnota symetrické funkce závisí jen na počtu vstupních bitů s hodnotou 1. Typickým příkladem symetrické funkce je parita. Je známo, že libovolnou symetrickou funkci n proměnných je možné implementovat pomocí prahového obvodu velikosti $O(\sqrt{n})$ hradel a hloubky 3 s polynomiálními váhami (Siu a kol., 1991). Navíc velikost takového obvodu je téměř optimální, protože v nejhorším případě musí obvod obsahovat aspoň $\Omega(\sqrt{n/\log n})$ hradel, i když povolíme neomezenou hloubku a váhy. Pro hloubku 2 O'Neil (1971) dokázal přiléhavou dolní mez $\Omega(\sqrt{n})$.

Tento výsledek také dosvědčuje, že prahové obvody jsou výpočetně silnější než tzv. *AND-OR obvody*, jejichž hradla počítají jen základní logické funkce AND, OR a NOT. Bylo totiž dokázáno (Furst a kol., 1984; Yao, 1985; Håstad, 1989), že paritu nelze počítat pomocí AND-OR obvodů polynomiální velikosti a konstantní hloubky (odpovídajících složitostní třídě označované AC^0). To vedlo k domněnce $AC^0 \subseteq TC_3^0$ slabšího kolapsu TC^0 -hierarchie. Částečný výsledek v tomto směru dosáhl Allender (1989), když dokázal, že funkce z AC^0 lze počítat pomocí prahových obvodů hloubky 3 a subexponenciální velikosti $n^{\log^c n}$ pro nějakou konstantu c .

3.1.6 Aritmetické funkce:

Výpočetní sílu prahových obvodů polynomiální velikosti a konstantní hloubky budeme dále ilustrovat na jejich schopnosti implementovat základní aritmetické funkce (Razborov 1992; Roychowdhury a kol., 1994; Siu a kol., 1995). Například s použitím výsledku Goldmanna a Karpinského (1998) je možné zkonstruovat dvouvrstvé perceptronové sítě polynomiální velikosti s polynomiálními váhami pro porovnání a pro součet dvou (resp. dokonce n) n -bitových binárních čísel (Alon, Bruck, 1994, resp. Siu, Roychowdhury, 1994). Nebo součin a podíl dvou n -bitových binárních čísel, n -tou mocninu (Siu, Roychowdhury, 1994) a třídění n takových čísel (Siu a kol., 1993) je možné implementovat pomocí třívrstevných sítí.

Z výsledku Hajnala a kol. (1993) vyplývá, že pro polynomiální velikost a váhy není možné hloubku těchto aritmetických obvodů zmenšit (Hofmeister, Pudlák, 1992; Wegener, 1993; Siu a kol., 1993) kromě obvodu pro mocninu, kde je tato otázka otevřená. Na druhou stranu mohou být polynomiální velikosti obvodů, které mají optimální hloubku, příliš velké na to, aby se daly prakticky realizovat. V takovém případě lze přidáním vrstvy snížit počet hradel na rozumnou míru. Například třívrstvou sítí nepřijatelné polynomiální velikosti $O(n^{27} \log n)$, která násobí dvě n -bitová čísla, je možné nahradit ekvivalentním čtyřvrstevným obvodem s $O(n^2)$ prahovými hradly (Siu, Roychowdhury, 1994). V hloubce 4 lze pak také realizovat součin n n -bitových binárních čísel (Siu, Roychowdhury, 1994).

Uvedené výsledky o optimální hloubce aritmetických prahových obvodů polynomiální velikosti a vah jsou shrnuty v tabulce 1 (Razborov, 1992; Hofmeister,

funkce	dolní odhad	horní odhad
porovnání	2	2
součet	2	2
násobný součet	2	2
součin	3	3
dělení	3	3
mocnina	2	3
třídění	3	3
násobný součin	3	4

Tabulka 1: Počet vrstev dopředné sítě polynomiální velikosti a vah počítající aritmetické funkce.

1994). Vyplývá z nich, že libovolnou analytickou funkci je možné s velkou přesností aproximovat pomocí perceptronové sítě polynomiální velikosti a vah s využitím jen malého konstantního počtu vrstev, pokud je reálná hodnota argumentu této funkce kódována binárně pomocí n vstupních bitů sítě (Reif, Tate, 1992).

3.1.7 Celková délka propojení:

Teprve nedávno byla zavedena nová složitostní míra dopředných sítí, tzv. *celková délka propojení (total wire length)* (Legenstein, Maass, 2002, 2005). Tato míra se snaží postihnout některé důležité aspekty VLSI implementace speciálních obvodů, např. neuronových sítí určených pro efektivní zpracování velkého počtu paralelních senzorických vstupů. V jednom z možných modelů se předpokládá, že hradla jsou umístěna v průnikových bodech dvoudimenzionální mřížky, která má jednotkovou vzdálenost mezi sousedními průnikovými body. Hradla mohou být spojena v rovině libovolně pomocí vodičů, které se mohou křížit. Celková délka propojení je pak definována jako minimální hodnota součtu délek vodičů v obvodu vzhledem ke všem možným umístěním hradel. Obecné (prahové) hradlo s k vstupy je zde modelováno jako mikroobvod (s jednotkovým časem výpočtu) pokrývající k průnikových bodů mřížky, které jsou spojeny vodičem libovolným způsobem.

Architektura dopředné sítě, která by měla optimální (např. téměř lineární) celkovou délku propojení, se musí zásadně lišit od běžně navrhovaných obvodů, protože úplné propojení mezi dvěma dvoudimenzionálními vrstvami s lineárním počtem hradel $O(n)$ vyžaduje celkovou délku propojení $\Omega(n^{2.5})$. Například booleovskou funkci $P_{LR}^k : \{0, 1\}^{2k} \rightarrow \{0, 1\}$ ($k \geq 2$) definovanou jako $P_{LR}^k(x_1, \dots, x_k, x'_1, \dots, x'_k) = 1$, právě když existuje $1 \leq i < j \leq k$ takové, že $x_i = x'_j = 1$, která je prototypem detektoru jednoduchého globálního vzoru, lze počítat pomocí dvouvrstvé sítě s $2 \log_2 k + 1$ prahovými hradly a celkovou délkou propojení $O(k \log k)$ (Legenstein, Maass, 2002).

3.2 Analogové dopředné sítě

V tomto odstavci porovnáme výpočetní vlastnosti binárních dopředných neuronových sítí a analogových acyklických sítí s aktivační funkcí logistickou sigmoidou (5), i když následující výsledky jsou platné i pro obecnější třídy sigmoidálních hradel. Tyto analogové sítě jsou pravděpodobně nejčastěji používaným výpočetním modelem v praktických aplikacích neuronových sítí, např. při učení pomocí algoritmu backpropagation (Rumelhart a kol., 1986). Booleovské prahové obvody je zřejmě možné implementovat pomocí analogových dopředných sítí stejné topologie tak, že zvýšíme absolutní hodnotu vah (vynásobením nenulové excitace (1) dostatečně

velkým kladným číslem) v případě, že je třeba přesnější aproximace Heavisideovy funkce (3) pomocí logistické sigmoidy (5) (Maass a kol., 1991).

3.2.1 Konstantní velikost:

Na druhou stranu použitím reálných hodnot stavů neuronů lze dosáhnout efektivnějších implementací funkcí pomocí malých analogových sítí konstantní velikosti. Například booleovskou funkci $F_k : \{0, 1\}^{2k} \rightarrow \{0, 1\}$ ($k \geq 1$) definovanou jako $F_k(x_1, \dots, x_k, x'_1, \dots, x'_k) = \text{Majority}(x_1, \dots, x_k) \oplus \text{Majority}(x'_1, \dots, x'_k)$, kde $\text{Majority}(x_1, \dots, x_k) = 1$, právě když $\sum_{i=1}^k x_i \geq \frac{k}{2}$, a \oplus značí funkci XOR, lze počítat pomocí dvouvrstvé sítě skládající se z pěti analogových jednotek s omezenými váhami a s výstupní separací $\varepsilon = \Omega(k^{-2})$ (resp. $\varepsilon = \Omega(1)$, pokud povolíme polynomiální váhy). Funkci F_k však nelze implementovat pomocí dvouvrstvé binární sítě s konstantním počtem prahových hradel, i když povolíme neomezené váhy (Maass a kol., 1991). Tento výsledek dosvědčuje, že analogové dopředné sítě jsou o trochu silnějším výpočetním prostředkem než jejich binární protějšky.

Podobný výsledek, který však nezávisí na hloubce obvodu, byl dokázán pro unární druhou mocninu $SQ_k : \{0, 1\}^{k^2+k} \rightarrow \{0, 1\}$ ($k \geq 1$) definovanou jako $SQ_k(x_1, \dots, x_k, z_1, \dots, z_{k^2}) = 1$, právě když $(\sum_{i=1}^k x_i)^2 \geq \sum_{i=1}^{k^2} z_i$. Tuto funkci je možné počítat jen pomocí dvou analogových jednotek s polynomiálními váhami a separací $\varepsilon = \Omega(1)$, zatímco libovolná binární dopředná síť, která počítá SQ_k , má velikost aspoň $\Omega(\log k)$ prahových hradel i pro neomezenou hloubku a váhy (DasGupta, Schnitger, 1996). Tedy velikost dopředné sítě lze někdy zmenšit o logaritmický faktor, když nahradíme prahová hradla sigmoidálními.

3.2.2 Polynomiální velikost:

Složitostní třídu TC_d^0 je možné zobecnit pro analogové sítě tak, že $TC_d^0(\sigma)$ ($d \geq 1$) obsahuje všechny funkce, které lze počítat analogovými dopřednými sítěmi polynomiální velikosti a hloubky d , s polynomiálními váhami, aktivací funkcí (5) a separací $\varepsilon = \Omega(1)$. Ukazuje se, že na této úrovni analogové a binární třídy splývají, tj. $TC_d^0(\sigma) = TC_d^0$ pro každé $d \geq 1$ (Maass a kol., 1991). Jinými slovy analogové a binární dopředné sítě polynomiální velikosti s daným počtem vrstev mají stejnou výpočetní sílu, a proto i v případě analogových sítí si třívrstvé sítě zachovávají výpočetní převahu nad dvouvrstvými. Tato výpočetní ekvivalence analogových a binárních dopředných sítí polynomiální velikosti platí i pro neomezenou hloubku a exponenciální váhy, pokud povolíme vzrůst počtu vrstev u binární sítě o konstantní faktor (DasGupta, Schnitger, 1993).

4 Rekurentní neuronové sítě

Architekturu *rekurentních (cyklických)* neuronových sítí tvoří obecně cyklický graf, a proto výpočty těchto sítí nemusí skončit po konečném počtu kroků. Pro rekurentní síť lze uvažovat různé výpočetní módy (režimy) v závislosti na volbě množin aktualizovaných jednotek α_t ve vzorci (2). Rekurentní síť pracuje v *sekvenčním* módu, jestliže v každém časovém okamžiku $t \geq 1$ nejvýše jedna jednotka aktualizuje svůj stav podle (2), tj. $|\alpha_t| \leq 1$. Abychom formálně vyloučili dlouhé konstantní mezivýpočty, kdy jsou aktualizovány jen ty jednotky, které ve skutečnosti nemění své výstupy, zavádí se koncept tzv. *produktivního* výpočtu délky t^* , v jehož průběhu se v každém časovém kroku $1 \leq t \leq t^*$ aktualizuje aspoň jedna jednotka $j \in \alpha_t$ tak, že $y_j^{(t)} \neq y_j^{(t-1)}$. Produktivní výpočet rekurentní sítě *skončí, konverguje* nebo *dosáhne stabilního stavu* $\mathbf{y}^{(t^*)}$ v čase $t^* \geq 0$, jestliže $\mathbf{y}^{(t^*)} = \mathbf{y}^{(t^*+k)}$ pro všechna $k \geq 1$ (resp. pro analogové sítě platí aspoň $\|\mathbf{y}^{(t^*)} - \mathbf{y}^{(t^*+k)}\| \leq \varepsilon$ pro nějakou malou konstantu

$0 \leq \varepsilon < 1$). Obvykle se předpokládá *systematická* volba α_t , např. $\alpha_{\tau s+j} = \{j\}$ pro $j = 1, \dots, s$, kde $\tau = 0, 1, 2, \dots$ představuje diskrétní *makroskopický čas*. V rámci jednoho makroskopického kroku jsou aktualizovány všechny jednotky v síti (obecně mohou některé z nich aktualizovat svůj výstup vícekrát). Naproti tomu v *paralelním* režimu výpočtu existuje časový okamžik $t \geq 1$, ve kterém aspoň dva neurony současně přepočítávají svůj nový výstup, tj. $|\alpha_t| \geq 2$. Typicky se uvažuje *plně paralelní* mód, ve kterém jsou v každém časovém kroku aktualizovány všechny jednotky, tj. $\alpha_t = V$ pro všechna $t \geq 1$.

Při *asynchronních* výpočtech je volba aktualizací množiny α_t náhodná, takže každá jednotka v síti se de facto nezávisle rozhoduje, kdy bude aktualizovat svůj stav. Oproti tomu při *synchronních* výpočtech jsou množiny α_t deterministicky určeny předem pro každý časový okamžik t . Pro binární stavy bylo ukázáno (Orponen, 1997b), že zdánlivě slabší asynchronní model má v podstatě stejnou výpočetní sílu jako jeho systematický synchronní protějšek v sekvenčním i paralelním módu dokonce i v případě symetrických sítí (viz odstavec 4.3). Proto se dále většinou omezíme na synchronní model.

Pro účely univerzálních výpočtů nad vstupy libovolné délky byly pro rekurentní síť zavedeny různé vstupní protokoly. V odstavci 4.1 popíšeme vstupní protokol pro *konečné* síť s konečným pevným počtem jednotek, zatímco v odstavci 4.4 se budeme zabývat nekonečnými posloupnostmi cyklických sítí, které obsahují jednu síť pro každou délku vstupu.

4.1 Konečné neuronové akceptory jazyků

Výpočetní síla rekurentních sítí se podobně jako u tradičních výpočetních modelů studuje tak, že se síť používají jako akceptory jazyků $L \subseteq \{0, 1\}^* = \cup_{n \geq 0} \{0, 1\}^n$ nad binární abecedou. Jazyk L odpovídá danému rozhodovacímu problému v tom smyslu, že L se skládá ze vstupních instancí, pro které je odpověď na tento problém „ano”. Například rozhodovací problém, zda je daná booleovská formule splnitelná, je reprezentovaný jazykem SAT obsahujícím všechny splnitelné booleovské formule. Říkáme, že jazyk $L \subseteq \{0, 1\}^*$ s charakteristickou funkcí $c_L : \{0, 1\}^* \rightarrow \{0, 1\}$ (tj. slovo $\mathbf{x} \in \{0, 1\}^*$ náleží do jazyka L , právě když $c_L(\mathbf{x}) = 1$) je *přijímán* (rozpoznávan) výpočetním modelem M (resp. odpovídající rozhodovací problém je řešitelný pomocí M), jestliže M parciálně implementuje c_L ve smyslu, že výpočet M nemusí skončit pro vstupy $\mathbf{x} \notin L$. Dále M *rozhoduje* L , jestliže M implementuje c_L totálně, např. Turingovy stroje rozhodují *rekurzivní jazyky*.

V případě konečných sítí, tzv. *neuronových akceptorů*, pracujících v plně paralelním režimu je vstupní slovo (řetězec) $\mathbf{x} = x_1 \dots x_n \in \{0, 1\}^n$ libovolné délky $n \geq 0$ prezentováno síti sekvenčně bit po bitu prostřednictvím vstupního neuronu $in \in V$. Stav této vstupní jednotky je externě nastaven na příslušné bity v předepsaných časových krocích bez ohledu na vliv zbývajících neuronů v síti. Výstupní neuron $out \in V$ následně signalizuje, jestli vstupní slovo \mathbf{x} náleží do příslušného jazyka L .

Speciálně v případě *binárních neuronových akceptorů*, založených na neuronových sítích s binárními stavy, se vstupní bity x_1, \dots, x_n předkládají síti sekvenčně s *periodou* $p \geq 1$ diskrétních kroků, tj. $y_{in}^{(p(i-1))} = x_i$ pro $i = 1, \dots, n$, a výstupní neuron rozhoduje každý prefix vstupního slova on-line s možným časovým zpožděním $k \geq 1$:

$$y_{out}^{(p(i-1)+k+1)} = \begin{cases} 1 & \text{jestliže } x_1 \dots x_i \in L \\ 0 & \text{jestliže } x_1 \dots x_i \notin L. \end{cases} \quad (9)$$

Pro binární síť se dá ukázat, že konstantní časové zpoždění k je možné snížit na jednotkové za cenu jen lineárního nárůstu velikosti sítě (Šíma, Wiedermann, 1998).

Doména vah	Deterministické sítě		Doména pravděpodobností	Pravděpodobnostní sítě	
	neomezený čas	polynomiální čas		neomezený čas	polynomiální čas
celočíslné	regulární	regulární	reálné	regulární	regulární
racionální	rekurzivní	P	racionální	rekurzivní	BPP
			reálné	libovolné	Pref-BPP/log
reálné	libovolné	P/poly	reálné	libovolné	P/poly

Tabulka 2: Výpočetní síla deterministických a pravděpodobnostních analogových sítí se saturovanou lineární aktivační funkcí pracujících v diskretním čase.

V případě analogových sítí obvykle $p = 1$ a konec vstupního slova je detekován *validačním* vstupním neuronem $ival \in V$, např.

$$y_{ival}^{(t)} = \begin{cases} 1 & \text{pro } t = 0, \dots, n-1 \\ 0 & \text{pro } t \geq n. \end{cases} \quad (10)$$

Výstupní neuron pak po nějakém čase $T(n) \geq n$, který závisí na délce vstupu n , hlásí výsledek rozpoznávání

$$y_{out}^{(t)} = \begin{cases} 1 & \text{jestliže } \mathbf{x} \in L \text{ a } t = T(n) \\ 0 & \text{jinak.} \end{cases} \quad (11)$$

Príslušný okamžik je opět detekován *validačním* výstupním neuronem $oval \in V$:

$$y_{oval}^{(t)} = \begin{cases} 1 & \text{pro } t = T(n) \\ 0 & \text{pro } t \neq T(n). \end{cases} \quad (12)$$

V analogových sítích je možný alternativní vstupní protokol (Siegelmann, Sontag, 1995), při kterém se vstupní slovo \mathbf{x} libovolné délky n zakóduje do reálného počátečního stavu vstupního neuronu, např. $y_{in}^{(0)} = \sum_{i=1}^n (2x_i + 1)/4^i$.

4.2 Konečné asymetrické rekurentní sítě

Výpočetní síla konečných rekurentních neuronových sítí s obecně asymetrickými váhami a saturovanou lineární aktivační funkcí (4) závisí na popisné složitosti vah. Příslušné výsledky jsou shrnuty v tabulce 2 (Siegelmann, 1994) včetně srovnání s pravděpodobnostními rekurentními sítěmi, kterými se budeme zabývat v odstavci 5.2.

4.2.1 Binární sítě:

Pro celočíselné váhy tyto modely splývají s konečnými binárními rekurentními sítěmi s Heavisideovou aktivační funkcí (3). Jelikož binární neuronové akceptory složené z prahových hradel mají jen konečný počet globální stavů, jejich výpočetní síla odpovídá konečným automatům rozpoznávajícím regulární jazyky (Kleene, 1956; viz také Minsky, 1967), proto se jim někdy říká *neuromaty*. Při studiu efektivity neuronových implementací konečných automatů byly uvažovány jemnější míry složitosti (Alon a kol., 1991). Například daný deterministický konečný automat s q stavy je možné simulovat neuromatem velikosti $O(\sqrt{q})$ s konstantní periodou $p = 4$ předkládání vstupních bitů (Horne, Hush, 1996; Indyk, 1995). Velikost $\Omega(\sqrt{q})$ simulujícího neuromatu nelze v nejhorším případě snížit, pokud je perioda nejvýše $p = O(\log q)$ (Horne, Hush, 1996) nebo pokud předpokládáme polynomiální váhy (Indyk, 1995). Velikost neuromatů můžeme také porovnat s délkou regulárních výrazů, které mají větší popisnou sílu než deterministické konečné automaty. Libovolný regulární jazyk

popsaný regulárním výrazem délky ℓ lze rozpoznat neuromatem optimální velikosti $\Theta(\ell)$ (Šíma, Wiedermann, 1998). Na druhou stranu lze sestrojít neuromat lineární velikosti rozpoznávající regulární jazyk, který lze prokazatelně popsat jen regulárním výrazem exponenciální délky, což potvrzuje velkou deskriptivní schopnost neuromatů.

4.2.2 Analogové sítě s racionálními váhami:

Konečné analogové rekurentní sítě s racionálními váhami a saturovanou lineární aktivační funkcí (4) mohou simulovat Turingův stroj dokonce tak, že každý přechod Turingova stroje je realizován v jednom časovém kroku sítě (Siegelmann, Sontag, 1995). Aplikací tohoto výsledku na univerzální Turingův stroj dostáváme, že libovolnou funkci vyčíslitelnou na Turingově stroji v čase $T(n)$ lze počítat pomocí *fixní* analogové rekurentní sítě jen s 886 jednotkami v čase $O(T(n))$. Velikost této univerzální sítě je možné dále redukovat jen na 25 neuronů za cenu nárůstu času simulace na $O(n^2T(n))$ (Indyk, 1995). Z uvedeného vyplývá, že výpočty konečných analogových sítí s racionálními váhami v polynomiálním čase odpovídají známé složitostní třídě P. Turingovská univerzalita byla dokázána i pro obecnější třídu aktivačních funkcí (Koiran, 1996) včetně logistické sigmoidy (5) (Kilian, Siegelmann, 1996), i když v tomto případě známé simulace vyžadují exponenciální časovou režii k realizaci jednoho přechodu Turingova stroje.

4.2.3 Analogové sítě s reálnými váhami:

Konečné analogové rekurentní sítě s obecnými reálnými váhami a saturovanou lineární aktivační funkcí (4) mohou teoreticky implementovat „super-turingovské“ výpočty díky neomezenému množství informace, kterou lze zakódovat do jejich reálných vah s neomezenou přesností. Výpočetní síla takových analogových neuronových akceptorů pracujících v čase $T(n)$ je stejná jako u nekonečných (neuniformních) posloupností prahových obvodů, jejichž velikost je polynomiální vzhledem k $T(n)$ (Siegelmann, Sontag, 1994). To znamená, že výpočty těchto sítí v polynomiálním čase odpovídají neuniformní třídě složitosti P/poly a v exponenciálním čase pomocí nich lze počítat *libovolnou* vstupně-výstupní funkci. Polynomiální výpočty konečných analogových neuronových sítí s rostoucí kolmogorovskou složitostí (objem informace) reálných vah dokonce definují vlastní hierarchii neuniformních složitostních tříd mezi P a P/poly (Balcázar a kol., 1997). Pokud se omezíme jen na rekurzivní (algoritmicky definované) váhy, analogové sítě mohou v polynomiálním čase rozhodnout právě jazyky z rekurzivní části P/poly, která je ostře větší než P. To znamená, že tento model vykazuje vyšší efektivitu i bez použití nerekurzivních vah.

4.2.4 Analogový šum:

Všechny předchozí výsledky týkající se analogových výpočtů konečných rekurentních neuronových sítí předpokládají operace nad reálnými čísly s neomezenou přesností. Nicméně pokud jsou výstupy jednotlivých analogových neuronů vystaveny libovolnému analogovému šumu, pak tyto sítě nejsou výpočetně silnější než neuromaty (Casey, 1996; Maass, Orponen, 1998). Dokonce pro běžná pravděpodobnostní rozdělení šumu, která jsou nenulová na dostatečně velké části stavového prostoru, nejsou tyto sítě schopné rozpoznat všechny regulární jazyky (Maass, Sontag, 1999; Siegelmann a kol., 2000), ale rozpoznávají spolehlivě tzv. *definitní jazyky* (Rabin, 1963). Jazyk L je definitní, jestliže existují dvě konečné množiny slov L_1 a L_2 takové, že $\mathbf{w} \in L$, právě když buď $\mathbf{w} \in L_1$, nebo $\mathbf{w} = \mathbf{v}_1\mathbf{v}_2$ pro nějaký prefix \mathbf{v}_1 a $\mathbf{v}_2 \in L_2$. Na druhou stranu pokud je úroveň šumu omezená, pak lze spolehlivě

rozpoznat všechny regulární jazyky a horní odhady na velikost příslušných neuroamatů zmiňované v odstavci 4.2.1 jsou platné pro velkou třídu aktivačních funkcí σ , které mají různé konečné limity v nevlastních bodech (Maass, Orponen, 1998; Siegelmann, 1996; Šíma, 1997).

4.2.5 Problém zastavení:

Pro konečné rekurentní sítě byla analyzována složitost různých důležitých rozhodovacích problémů. Například otázka, zda má daná binární síť aspoň jeden stabilní stav, je NP-úplný problém (Alon, 1985; Godbeer a kol., 1988; Lipscomb, 1987; Porat, 1989). Nebo problém zastavení sítě, tj. zda rekurentní síť skončí svůj výpočet nad daným vstupem, je pro binární stavy PSPACE-úplný (Floréen, Orponen, 1994; Lepley, Miller, 1983; Porat, 1989). Tato otázka je algoritmicky nerozhodnutelná pro analogové sítě s racionálními váhami a 25 jednotkami, což je důsledkem turin-govské výpočetní univerzality takových sítí (Indyk, 1995). Ještě poznamenejme, že výpočty rekurentních sítí velikosti s , které se vždy zastaví v čase nejvýše t^* , lze zřejmě rozvinout do obvodu velikosti st^* a hloubky t^* (Savage, 1972). To znamená, že dopředné a *konvergentní* rekurentní sítě mají principiálně stejnou výpočetní sílu (Goldschlager, Parberry, 1986).

4.3 Konečné symetrické rekurentní sítě

Důležitým speciálním případem rekurentních sítí jsou *symetrické*, resp. *Hopfieldovy* sítě, jejichž váhy splňují $w(i, j) = w(j, i)$ pro každé $i, j \in V$, a tedy odpovídající architekturu tvoří neorientovaný graf.

4.3.1 Konvergence:

Dynamika symetrických (Hopfieldových) sítí je podřízena tzv. *Ljapunovově*, resp. *energetické* funkci E , definované na stavovém prostoru sítě. Tato funkce je omezená a klesá podél trajektorie každého produktivního výpočtu. Z existence takové funkce vyplývá, že síť vždy konverguje k nějakému stabilnímu stavu, který odpovídá lokálnímu minimu E . Binární symetrické sítě, které splňují $w(j, j) \geq 0$ pro každé $j \in V$ (takové sítě se někdy nazývají *semijednoduché*), vždy konvergují v sekvencním režimu z libovolného počátečního stavu (Hopfield, 1982). Tento výsledek je možné dokázat např. pomocí energetické funkce

$$E(\mathbf{y}) = - \sum_{j=1}^s y_j \left(w_{j0} + \frac{1}{2} \sum_{i=1, i \neq j}^s w_{ji} y_i + w_{jj} y_j \right). \quad (13)$$

Na druhou stranu paralelní výpočty binárních Hopfieldových sítí (včetně nesemi-jednoduchých) buď dosáhnou stabilního stavu, např. když je kvadratická forma (13) negativně definitní (Goles, 1987; Goles-Chacc a kol., 1985), nebo nakonec střídají dva různé stavy (Bruck, Goodman, 1988; Goles, Olivos 1981a; Goles, 1987; Poljak, Šíra, 1983; Tehuente, 1986). S použitím Ljapunovovy funkce (13) rozšířené o aditivní člen $\sum_{j=1}^s \int_0^{y_j} \sigma^{-1}(y) dy$ byly uvedené výsledky dále zobecněny pro analogové symetrické sítě s aktivační funkcí σ . Konkrétně bylo za slabého předpokladu platnosti jisté hypotézy ukázáno, že tyto sítě konvergují k pevnému bodu nebo k limitnímu cyklu délky nejvýše dvou paralelních stavových aktualizací (Fogelman-Soulié a kol., 1989; Koiran, 1994).

4.3.2 Čas konvergence:

Čas konvergence, resp. doba výpočtu Hopfieldovy sítě je počet aktualizací diskrétních kroků potřebných k tomu, aby síť dosáhla stabilního stavu (pro analogové

sítě v rámci dané přesnosti). U symetrických sítí s s binárními neurony platí triviální horní odhad 2^s kroků, protože síť má v tomto případě jen 2^s různých stavů. V nejhorším případě může být čas konvergence opravdu exponenciální v sekvenčním (Haken, Luby, 1988) i paralelním (Goles, Olivos, 1981b) režimu. Příkladem takové symetrické sítě je sekvenční, resp. paralelní, implementace binárního čítače, který prochází většinu stavového prostoru sítě a konverguje po $\Omega(2^{s/8})$ asynchronních sekvenčních aktualizacích (Haken, 1989), resp. po $\Omega(2^{s/3})$ plně paralelních krocích (Goles, Martínez, 1989, 1990). Na druhou stranu byla pro binární Hopfieldovy sítě dokázána za rozumných předpokladů velmi rychlá konvergence v průměrném případě po řádově $O(\log \log s)$ paralelních krocích (Komlós, Paturi, 1988). Nicméně uvedené odhady neberou v úvahu velikost vah. Z definice energetické funkce (13) vyplývá horní mez $O(W)$ na čas konvergence binárních symetrických sítí pracujících v sekvenčním (Fogelman a kol., 1983; Goles, 1985; Goles-Chacc a kol., 1985; Goles, Martínez, 1990), resp. paralelním režimu (Goles, 1987), kde $W = \sum_{j,i \in V} |w_{ji}|$ je celková váha sítě. Např. Floréen (1991) vyjádřil tento horní odhad přesněji pro celočíselné váhy. Tedy binární Hopfieldovy sítě s polynomiálními váhami konvergují v polynomiálním čase. Pro analogové Hopfieldovy sítě je možné čas konvergence vyjádřit vzhledem k počtu bitů potřebných k reprezentaci jejich vah a dá se zkonstruovat analogová symetrická síť, jejíž výpočet skončí později než výpočet libovolné binární Hopfieldovy sítě se stejnou velikostí reprezentace (Šíma a kol., 2000).

4.3.3 Stabilní stavy:

Při použití Hopfieldovy sítě jako asociativní paměti odpovídají stabilní stavy uloženým vzorům, proto je stanovení počtu stabilních stavů v síti důležitým problémem. Bylo ukázáno, že binární Hopfieldova síť velikosti s , jejíž váhy (kromě $w_{jj} = w_{j0} = 0$, $j \in V$) jsou nezávisle identicky distribuované gaussovské náhodné veličiny s nulovou střední hodnotou, má v průměrném případě asymptoticky $1.05 \times 2^{0.2874s}$ stabilních stavů (McEliece a kol., 1987; Tanaka, Edwards, 1980). Avšak problém, zda má daná binární symetrická síť např. (pokud je nesemijednoduchá) aspoň jeden (Floréen, Orponen, 1989), dva (Lipscomb, 1987) nebo tři (Floréen, Orponen, 1989) stabilní stavy, je NP-úplný. Ve skutečnosti je určení přesného počtu stabilních stavů v dané binární Hopfieldově síti algoritmicky těžký problém (Floréen, Orponen, 1989; Lipscomb, 1987). Také např. problém výpočtu *poloměru atrakce* daného stabilního stavu, což je největší počet bitů stabilního stavu, které můžeme změnit, aby příslušná Hopfieldova síť v sekvenčním nebo paralelním režimu vždy konvergovala zpátky k tomuto stavu, je NP-těžký a nelze jej rozumně aproximovat v polynomiálním čase, pokud $P \neq NP$ (Floréen, Orponen, 1993).

4.3.4 Problém minimální energie:

Hopfieldovy sítě se také používají pro heuristické řešení těžkých kombinatorických optimalizačních problémů (Hopfield, Tank, 1985). Účelová funkce optimalizačního problému je v tomto případě zakódována do energetické funkce Hopfieldovy sítě, která je minimalizována v průběhu jejího výpočtu. Proto se problém nalezení stavu Hopfieldovy sítě s minimální energií nebo energií menší než je předepsaná hodnota těší zvláštní pozornosti. Nicméně tento problém je obecně NP-úplný pro binární (Barahona, 1982; Bertoni, Campadelli, 1994) i analogové (Šíma, a kol., 2000) Hopfieldovy sítě, i když pro binární sítě s planární topologií je řešitelný v polynomiálním čase (Barahona, 1982). Navíc pro binární symetrické sítě s váhou W existuje polynomiální aproximační algoritmus, který je založený na výkonném aproximačním algoritmu pro problém maximálního řezu v grafu (Goemans, Williamson, 1995; Mahajan, Ramesh, 1999) a řeší problém minimální energie s garantovanou absolutní chybou menší než $0.243W$ (Šíma a kol., 2000). Pro $W = O(s^2)$, tj. např. pro Hop-

fieldovy sítě s s binárními neurony a konstantními váhami, tento výsledek odpovídá dolní mezi $\Omega(s^{2-\varepsilon})$ pro absolutní chybu, kterou nelze garantovat žádným polynomiálním aproximačním algoritmem pro každé $\varepsilon > 0$, pokud $P \neq NP$ (Bertoni, Campadelli, 1994).

4.3.5 Výpočetní síla:

Nakonec podáme stručný přehled výsledků o výpočetní síle Hopfieldových sítí (srovnej s odpovídajícími výsledky pro asymetrické sítě prezentovanými v odstavci 4.2). V případě binárních stavů neuronů jsou symetrické sítě schopné simulovat libovolné *konvergentní* asymetrické sítě za cenu jen lineárního nárůstu času a velikosti simulující Hopfieldovy sítě (Orponen, 1996; Šíma a kol., 2000). Tato efektivní obrácená implikace slavné Hopfieldovy konvergenční věty (viz odstavec 4.3.1) tedy ukazuje, že pro symetrické binární sítě platí ekvivalence

“konvergence \equiv symetrie vah”,

tj. nejen že všechny binární symetrické sítě konvergují, ale všechny konvergentní výpočty libovolné binární asymetrické neuronové sítě mohou být efektivně implementovány pomocí Hopfieldovy sítě. Tento výsledek byl již dříve znám pro dopředné sítě, které je možné implementovat pomocí symetrických rekurentních sítí se stejnou architekturou (Parberry, 1994). Na druhou stranu symetrické neuromaty s předem neomezenou délkou vstupního slova jsou slabším výpočetním prostředkem než konečné automaty a rozpoznávají jen vlastní podtřídu regulárních jazyků, které se nazývají *Hopfieldovy jazyky* (Šíma, Wiedermann, 1998). Dá se ukázat, že regulární jazyk $L \subseteq \{0, 1\}^*$ je Hopfieldův, právě když pro každý prefix a sufix $\mathbf{v}_1, \mathbf{v}_2 \in \{0, 1\}^*$ a dvoubitový řetězec $\mathbf{x} \in \{0, 1\}^2$ existuje k_0 takové, že buď $\mathbf{v}_1 \mathbf{x}^k \mathbf{v}_2 \in L$ pro všechna $k \geq k_0$, nebo $\mathbf{v}_1 \mathbf{x}^k \mathbf{v}_2 \notin L$ pro všechna $k \geq k_0$.

Také analogové symetrické neuronové akceptory jsou schopné spolehlivě rozpoznat Hopfieldovy jazyky (Šíma, 1997), což představuje dolní odhad na jejich výpočetní sílu. Díky jejich konvergenčním vlastnostem (viz odstavec 4.3.1) je však nepravděpodobné, že by konečné analogové Hopfieldovy sítě byly schopné simulovat libovolný Turingův stroj. Nicméně, pokud k analogové symetrické síti pracující v plně paralelním režimu přidáme externí oscilátor, který generuje posloupnost bitů obsahující nekonečně mnoho tříbitových podřetězců tvaru $bx\bar{b} \in \{0, 1\}^3$, kde $b \neq \bar{b}$, pak tento model může simulovat libovolnou asymetrickou rekurentní síť se stejnou kolmogorovskou složitostí reálných vah. To znamená, že výsledky o výpočetní síle deterministických asymetrických sítí prezentované v tabulce 2 jsou platné i pro Hopfieldovy sítě s přidaným oscilátorem, např. pro racionální váhy jsou turingovsky univerzální. Tím dostáváme úplnou charakterizaci výpočetní síly konečných analogových neuronových sítí s racionálními váhami ve tvaru

“Turingův stroj \equiv asymetrická síť \equiv symetrická síť + oscilátor”

spolu s nutnou a postačující podmínkou, kterou musí externí oscilátor splňovat, aby tato ekvivalence byla platná.

4.4 Nekonečné posloupnosti binárních rekurentních sítí

V odstavci 4.1 jsme se zabývali konečnými neuronovými akceptory, které vstupy s neomezenou délkou zpracovávají bit po bitu. Podobně jako u obvodů (odstavec 3) lze použít alternativní vstupní protokol, při kterém předpokládáme nekonečnou posloupnost (rodinu) $\{N_n\}$ binárních rekurentních sítí, která obsahuje jednu síť N_n pro každou délku vstupu $n \geq 0$. Tedy pro výpočty nad n -bitovými binárními vstupy $\mathbf{x} \in \{0, 1\}^n$ se používá síť N_n , u níž jsou stavy n vstupních neuronů na začátku výpočtu (počáteční stav) nastaveny na vstup \mathbf{x} . Pro rozpoznání jazyka $L \subseteq \{0, 1\}^*$

se pro vstupní slovo $\mathbf{x} \in \{0, 1\}^n$ použije příslušná síť N_n , která poté, co konverguje v čase t^* , signalizuje prostřednictvím svého jediného výstupního neuronu *out*, jestli \mathbf{x} náleží do L , tj. $y_{out}^{(t^*)} = 1$, právě když $\mathbf{x} \in L$. Opět definujeme *velikost* $S(n)$ jako počet jednotek v N_n . Posloupnosti binárních rekurentních sítí polynomiální velikosti rozpoznávají právě jazyky ze složitostní třídy PSPACE/poly (Lepley, Miller, 1983). Ekvivalence mezi konvergentními asymetrickými a symetrickými sítěmi (odstavce 4.3.5) implikuje stejnou výpočetní sílu pro rodiny Hopfieldových sítí polynomiální velikosti, které také rozpoznávají PSPACE/poly (Orponen, 1996). Pokud navíc Hopfieldovy sítě v těchto posloupnostech mají *polynomiální* váhy (vzhledem k délce vstupu n), pak je jejich výpočetní síla charakterizována třídou P/poly.

5 Pravděpodobnostní neuronové sítě

V literatuře se setkáváme s různými stochastickými variantami diskrétních perceptronových sítí, jejichž výpočetní vlastnosti byly analyzovány. Definujeme referenční model *pravděpodobnostních (stochastických) sítí* tak, že k příslušné deterministické síti přidáme náhodné binární vstupní jednotky $i \in \Pi$, jejichž stavy v čase reprezentují nezávislé a identicky distribuované binární posloupnosti. Tj. pro všechny diskrétní časové okamžiky $t \geq 0$ je pravděpodobnost, že $y_i^{(t)} = 1$ pro náhodný vstupní neuron $i \in \Pi$, rovna dané hodnotě p_i ($0 \leq p_i \leq 1$), a tedy $y_i^{(t)} = 0$ nastává s pravděpodobností $1 - p_i$. Tento referenční model pravděpodobnostních sítí je obvykle srovnatelný pomocí vzájemných polynomiálních simulací s jinými typy stochastických neuronových sítí, např. se sítěmi s chybovými stavy či spoji (Siegelmann, 1999b; von Neumann, 1956) nebo Boltzmannovými stroji (Ackley a kol., 1985; Parberry, 1994; Parberry, Schnitger, 1989) apod. Při studiu výpočetní síly se pravděpodobnostní sítě používají k rozpoznávání jazyků podobně jako jejich deterministické protějšky. Jazyk $L \subseteq \{0, 1\}^*$ je ε -rozpoznáván pravděpodobnostní neuronovou sítí, jestliže pravděpodobnost chyby sítě je nejvýše ε , kde $0 \leq \varepsilon < 1/2$, tj. pravděpodobnost, že síť odpoví 1 na vstup $\mathbf{x} \in \{0, 1\}^*$, je aspoň $1 - \varepsilon$ pro $\mathbf{x} \in L$ a nejvýše ε pro $\mathbf{x} \notin L$. Uvedenou symetrii v pravděpodobnosti chyby přijetí a zamítnutí vstupního slova lze vždy docílit přidáním několika náhodných vstupních jednotek (Hajnal a kol., 1993).

5.1 Pravděpodobnostní dopředné sítě

Pro jednoduchost nejprve uvažujme pravděpodobnostní binární dopředné sítě (pravděpodobnostní prahové obvody). Rozpoznávání jazyků pomocí posloupností pravděpodobnostních prahových obvodů s jedním výstupem, které mají velkou pravděpodobnost chyby (např. $\varepsilon = 0.4$), není příliš spolehlivé, avšak pravděpodobnost chyby lze libovolně snížit paralelním opakováním výpočtu. Každý jazyk, který je ε -rozpoznáván ($0 < \varepsilon < 1/2$) pravděpodobnostní dopřednou sítí hloubky d vrstev a velikosti s jednotek, může být λ -rozpoznáván stochastickým prahovým obvodem hloubky $d + 1$ a velikosti $\lceil 2 \log_{4\varepsilon(1-\varepsilon)} \lambda \rceil s + 1$ hradel pro libovolné $0 < \lambda < \varepsilon$. Proto lze každou pravděpodobnostní binární dopřednou síť dokonce nahradit přijatelně velkým ekvivalentním *deterministickým* prahovým obvodem. Konkrétně Parberry a Schnitger (1989) dokázali, že pro každý jazyk $L \subseteq \{0, 1\}^n$, který je ε -rozpoznáván ($1/4 < \varepsilon < 1/2$) pravděpodobnostní dopřednou sítí s n vstupy, hloubky d a velikosti s , existuje (deterministický) prahový obvod hloubky $d + 1$ a velikosti

$$\left\lceil \frac{8\varepsilon \ln 2}{(1 - 2\varepsilon)^2} + 1 \right\rceil ns + 1 \quad (14)$$

rozpoznávající L .

Složitostní třídu TC_d^0 ($d \geq 1$) pro posloupnosti prahových obvodů (viz odstavec 3.1.4) lze zobecnit pro stochastické sítě tak, že odpovídající pravděpodobnostní třída RTC_d^0 obsahuje všechny jazyky, které jsou $\varepsilon(n)$ -rozpoznávány posloupnostmi pravděpodobnostních prahových obvodů polynomiální velikosti, hloubky $d \geq 1$ a s polynomiálními váhami, kde pravděpodobnost chyby pro každou délku vstupu $n \geq 0$ je dána reálnou posloupností $\varepsilon(n) = 1/2 - 1/n^{O(1)}$. Například jazyk IP , který obsahuje vstupy, pro které má booleovský skalární součin (8) hodnotu 1, náleží do třídy RTC_2^0 (Hajnal a kol., 1993). To znamená, že pravděpodobnostní dopředné sítě mohou být efektivnější než jejich deterministické protějšky, protože $IP \notin TC_2^0$ (viz odstavec 3.1.4). Na druhou stranu aspoň u *neuniformních* posloupností pravděpodobnostních prahových obvodů lze ušetřit nejvýše jednu vrstvu, neboť v tomto případě platí $RTC_d^0 \subseteq TC_{d+1}^0$ pro každé $d \geq 1$ (Hajnal a kol., 1993), jak je načrtnuto na obrázku 2.

5.2 Pravděpodobnostní rekurentní sítě

Výpočetní síla konečných rekurentních sítí se saturovanou lineární aktivační funkcí (4) byla analyzována také pro pravděpodobnostní sítě (Siegelmann, 1999b) a výsledky jsou shrnuty a porovnány s deterministickými modely v tabulce 2. Pro celočíselné váhy binární pravděpodobnostní sítě ε -rozpoznávají regulární jazyky. Analogové pravděpodobnostní sítě s racionálními váhami v polynomiálním čase ε -rozpoznávají právě jazyky z neuniformní třídy složitosti $\text{Pref-BPP}/\log$ (viz příloha). Tato slabá superturingovská výpočetní schopnost je důsledkem toho, že pravděpodobnosti p_i u náhodných vstupů $i \in \Pi$ jsou *libovolná* reálná čísla. Pokud se omezíme na racionální hodnoty pravděpodobností, pak je výpočetní síla analogových pravděpodobnostních sítí s racionálními váhami v polynomiálním čase charakterizována rekurzivní složitostní třídou BPP. Nakonec pravděpodobnostní rekurentní sítě s obecnými reálnými váhami pracující v čase $T(n)$ lze simulovat pomocí odpovídajících deterministických sítí v čase $nT(n) + n^2$. To znamená, že polynomiální výpočty těchto sítí jsou také charakterizovány složitostní třídou P/poly.

6 Spojitý čas

Analogový stav neuronové sítě $\mathbf{y}(t) \in \mathbf{R}^s$ pracující ve spojitém čase (zkráceně spojitě sítě) je definován pro každý reálný časový okamžik $t \geq 0$ obvykle jako řešení soustavy s diferenciálních rovnic, ve které každá rovnice odpovídá jedné jednotce ve spojitě sítě. Počáteční podmínky této soustavy jsou dány počátečním stavem sítě $\mathbf{y}(0)$. Výpočetní dynamika spojitě sítě může být zadána např. soustavou

$$\frac{dy_j}{dt}(t) = -y_j(t) + \sigma(\xi_j(t)) = -y_j(t) + \sigma\left(\sum_{i=0}^s w_{ji}y_i(t)\right) \quad (15)$$

pro $j = 1, \dots, s$, kde excitace $\xi_j(t)$ jednotky j je definována vzorcem (1) a σ je saturovaná lineární aktivační funkce (4). Pomocí Ljapunovovy funkce

$$E(\mathbf{y}) = -\sum_{j=1}^s y_j \left(w_{j0} + \frac{1}{2} \sum_{i=1}^s w_{ji}y_i \right) + \sum_{j=1}^s \int_0^{y_j} \sigma^{-1}(y) dy \quad (16)$$

lze ukázat, že spojitá symetrická síť ($w_{ji} = w_{ij}$) s výpočetní dynamikou (15) konverguje z libovolného počátečního stavu $\mathbf{y}(0)$ ke stabilnímu stavu, pro který platí $dy_j/dt = 0$, resp. $y_j(t) = \sigma(\xi_j(t))$, pro všechny $j = 1, \dots, s$ (Cohen, Grossberg, 1983; Hopfield, 1984). V nejhorsím případě však může být čas konvergence exponenciální vzhledem k velikosti sítě (Šíma, Orponen, 2003b). Z hlediska výpočetní

síly lze pro danou binární asymetrickou rekurentní neuronovou síť pracující v diskrétním čase zkonstruovat spojitou symetrickou síť, která ji simuluje za cenu jen lineárního nárůstu velikosti sítě (Šíma, Orponen, 2003a). To znamená, že posloupnosti spojitých (symetrických) sítí polynomiální velikosti rozpoznávají aspoň jazyky z třídy PSPACE/poly.

7 Závěr

V této kapitole jsme podali systematický přehled výsledků týkající se složitostně teoretických vlastností modelů neuronových sítí, které implementují digitální výpočty. Z prostorových důvodů jsme se omezili na tradiční perceptronové jednotky, protože jejich výpočetní vlastnosti jsou nejlépe prozkoumány. Jiné typy jednotek byly z tohoto hlediska studovány teprve nedávno a jejich analýza ještě není úplná. Nicméně se ukazuje, že RBF sítě mají srovnatelnou výpočetní sílu jako perceptrony (Šorel, Šíma, 2004; Friedrichs, Schmitt, 2005), zatímco WTA (Winner-Take-All) hradla mohou některé funkce implementovat efektivněji (Maass, 2000). Také sítě spiking neuronů mají nepatrně vyšší efektivitu než perceptronové sítě (Maass, 1996, 1997). Navíc spiking neurony kromě toho, že věrněji modelují biologické neurony, používají frekvenční kódování stavů, jehož význam pro efektivní počítání nepochybně zasluhuje hlubší studium.

V našem přehledu jsme se dále především zaměřili na modely neuronových sítí pracující v diskrétním čase a o spojitých sítích zatím víme jen to, že jsou výpočetně aspoň tak silné jako jejich diskrétní protějšky. To je způsobeno tím, že nemáme k dispozici adekvátní teoretické nástroje (např. složitostní míry, redukce, univerzální výpočet) pro „přirozené“ spojitě výpočty (Maass a kol., 2002; Moore, 1998; Orponen 1997a). Spojité neuronové sítě mohou v tomto ohledu posloužit jako referenční modely pro vývoj takových nástrojů (Ben-Hur a kol., 2002; Gori, Meer, 2002).

Analýza dopředných perceptronových sítí ukázala, že mnohé důležité funkce je možné implementovat pomocí jen několika vrstev a rozumného počtu neuronů. Velmi důležitým výsledkem v tomto kontextu je, že dvě vrstvy perceptronů nejsou dostatečné pro efektivní implementaci jistých funkcí, které vyžadují třívrstvé sítě. Naproti tomu výpočetní síla rekurentních sítí byla elegantně charakterizovaná pomocí deskriptivní složitosti vah, jak je shrnuto v tabulce 2, která navíc ukazuje, že stochasticita hraje u modelů neuronových sítí podobnou úlohu jako u Turingových strojů. Analogové stavy rekurentních sítí mohou teoreticky kódovat nekonečné množství informace, což není z hlediska praktické implementace realistické, nicméně analogové neuronové sítě s omezenou přesností reálných parametrů mohou být oproti binárním sítím efektivnějším výpočetním prostředkem. Pro binární sítě pak umíme ukázat, že symetrie vah a konvergence sítě znamená *de facto* totéž.

Jak bylo ukázáno v tomto přehledu, naše poznání výpočetní síly neuronových sítí je z hlediska jejich schopnosti implementovat univerzální výpočty celkem uspokojivé. Důležitou oblastí pro další výzkum je analýza výpočetních vlastností neuronových sítí pro specializované výpočty z pohledu jejich aplikací.

Literatura

- Ackley D. H., Hinton G. E., Sejnowski T. J.: *A learning algorithm for Boltzmann machines*. Cognitive Science, vol. 9, no. 1, 1985, s. 147–169.
- Allender E.: *A note on the power of threshold circuits*. In Proceedings of the FOCS'89 Thirty Annual Symposium on Foundations of Computer Science, IEEE Computer Society, New York 1989, s. 580–584.

- Alon N.: *Asynchronous threshold networks*. Graphs and Combinatorics, vol. 1, 1985, s. 305–310.
- Alon N., Bruck J.: *Explicit construction of depth-2 majority circuits for comparison and addition*. SIAM Journal on Discrete Mathematics, vol. 7, no. 1, 1994, s. 1–8.
- Alon N., Dewdney A. K., Ott T. J.: *Efficient simulation of finite automata by neural nets*. Journal of the ACM, vol. 38, no. 2, 1991, s. 495–514.
- Anthony M.: *Discrete Mathematics of Neural Networks: Selected Topics*. Society for Industrial and Applied Mathematics, Philadelphia PA 2001.
- Anthony M., Bartlett P. L.: *Neural Network Learning: Theoretical Foundations*. Cambridge University, Cambridge UK 1999.
- Balcázar J. L., Díaz J., Gabarró J.: *Structural Complexity I* (2nd edition). Springer-Verlag, Berlin 1995.
- Balcázar J. L., Gavaldà R., Siegelmann H. T.: *Computational power of neural networks: A characterization in terms of Kolmogorov complexity*. IEEE Transactions on Information Theory, vol. 43, no. 4, 1997, s. 1175–1183.
- Barahona F.: *On the computational complexity of Ising spin glass models*. Journal of Physics A: Mathematical and General, vol. 15, no. 10, 1982, s. 3241–3253.
- Ben-Hur A., Siegelmann H. T., Fishman S.: *A theory of complexity for continuous time systems*. Journal of Complexity, vol. 18, no. 1, 2002, s. 51–86.
- Bertoni A., Campadelli P.: *On the approximability of the energy function*. In Proceedings of the ICANN'94 Fourth International Conference on Artificial Neural Networks, Springer-Verlag, Berlin 1994, s. 1157–1160.
- Bruck J., Goodman J. W.: *A generalized convergence theorem for neural networks*. IEEE Transactions on Information Theory, vol. 34, no. 5, 1988, s. 1089–1092.
- Casey M.: *The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction*. Neural Computation, vol. 8, no. 6, 1996, s. 1135–1178.
- Cohen M. A., Grossberg S.: *Absolute stability of global pattern formation and parallel memory storage by competitive neural networks*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 13, no. 5, 1983, s. 815–826.
- Cover T. M.: *Capacity problems for linear machines*. In Kanal L. (Ed): Pattern Recognition, Thompson Book Co., Washington DC 1968, s. 283–289.
- DasGupta B., Schmitger G.: *The power of approximating: A comparison of activation functions*. In Hanson S. J., Cowan J. D., Giles C. L. (Eds): Advances in Neural Information Processing Systems (NIPS'92), vol. 5, Morgan Kaufmann, San Mateo 1993, s. 615–622.
- DasGupta B., Schmitger G.: *Analog versus discrete neural networks*. Neural Computation, vol. 8, no. 4, 1996, s. 805–818.
- Floréen P.: *Worst-case convergence times for Hopfield memories*. IEEE Transactions on Neural Networks, vol. 2, no. 5, 1991, s. 533–535.
- Floréen P., Orponen P.: *On the computational complexity of analyzing Hopfield nets*. Complex Systems, vol. 3, no. 6, 1989, s. 577–587.
- Floréen P., Orponen P.: *Attraction radii in Hopfield nets are hard to compute*. Neural Computation, vol. 5, no. 5, 1993, s. 812–821.
- Floréen P., Orponen P.: *Complexity issues in discrete Hopfield networks*. Research Report, no. A-1994-4, Department of Computer Science, University of Helsinki 1994.
- Fogelman F., Goles E., Weisbuch G.: *Transient length in sequential iteration of threshold functions*. Discrete Applied Mathematics, vol. 6, no. 1, 1983, s. 95–98.

- Fogelman-Soulié F., Mejia C., Goles E., Martínez S.: *Energy functions in neural networks with continuous local functions*. Complex Systems, vol. 3, no. 3, 1989, s. 269–293.
- Friedrichs F., Schmitt M.: *On the power of Boolean computations in generalized RBF neural networks*. Neurocomputing, vol. 63, 2005, s. 483–498.
- Furst M., Saxe J. B., Sipser M.: *Parity, circuits and the polynomial-time hierarchy*. Mathematical Systems Theory, vol. 17, no. 1, 1984, s. 13–27.
- Gerstner W., Kistler W. M.: *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, Cambridge UK 2002.
- Godbeer G. H., Lipscomb J., Luby M.: *On the computational complexity of finding stable state vectors in connectionist models (Hopfield nets)*. Technical Report, no. 208/88, Department of Computer Science, University of Toronto 1988.
- Goemans M. X., Williamson D. P.: *Improved approximate algorithms for maximum cut and satisfiability problems using semidefinite programming*. Journal of the ACM, vol. 42, no. 6, 1995, s. 1115–1145.
- Goldmann M., Håstad J., Razborov A.: *Majority gates vs. general weighted threshold gates*. Computational Complexity, vol. 2, no. 4, 1992, s. 277–300.
- Goldmann M., Karpinski M.: *Simulating threshold circuits by majority circuits*. SIAM Journal on Computing, vol. 27, no. 1, 1998, s. 230–246.
- Goldschlager L. M., Parberry I.: *On the construction of parallel computers from various bases of Boolean functions*. Theoretical Computer Science, vol. 43, no. 1, 1986, s. 43–48.
- Goles E.: *Dynamics of positive automata networks*. Theoretical Computer Science, vol. 41, no. 1, 1985, s. 19–32.
- Goles E.: *Lyapunov functions associated to automata networks*. In Fogelman-Soulié F., Robert Y., Tchente M. (Eds): Automata Networks in Computer Science—Theory and Applications, Manchester University Press, Manchester 1987, s. 58–81.
- Goles-Chacc E., Fogelman-Soulié F., Pellegrin D.: *Decreasing energy functions as a tool for studying threshold networks*. Discrete Applied Mathematics, vol. 12, no. 3, 1985, s. 261–277.
- Goles E., Martínez S.: *Exponential transient classes of symmetric neural networks for synchronous and sequential updating*. Complex Systems, vol. 3, no. 6, 1989, s. 589–597.
- Goles E., Martínez S.: *Neural and Automata Networks: Dynamical Behavior and Applications*. Kluwer Academic Publishers, Dordrecht 1990.
- Goles E., Olivos, J.: *Comportement périodique des fonctions à seuil binaires et applications*. Discrete Applied Mathematics, vol. 3, no. 2, 1981a, s. 93–105.
- Goles E., Olivos, J.: *The convergence of symmetric threshold automata*. Information and Control, vol. 51, no. 2, 1981b, s. 98–104.
- Gori M., Meer K.: *A step towards a complexity theory for analog systems*. Mathematical Logic Quarterly, vol. 48, no. 1, 2002, s. 45–58.
- Hajnal A., Maass W., Pudlák P., Szegedy M., Turán G.: *Threshold circuits of bounded depth*. Journal of Computer and System Sciences, vol. 46, no. 2, 1993, s. 129–154.
- Haken A.: *Connectionist networks that need exponential time to stabilize*. Unpublished manuscript, Department of Computer Science, University of Toronto 1989.
- Haken A., Luby M.: *Steepest descent can take exponential time for symmetric connectionist networks*. Complex Systems, vol. 2, no. 2, 1988, s. 191–196.
- Håstad J.: *Almost optimal lower bounds for small depth circuits*. In Micali S. (Ed): Advances in Computing Research, Randomness and Computation, vol. 5, JAI Press Inc., Greenwich CT 1989, s. 143–170.

- Håstad J.: *On the size of weights for threshold gates*. SIAM Journal on Discrete Mathematics, vol. 7, no. 3, 1994, s. 484–492.
- Haykin S.: *Neural Networks: A Comprehensive Foundation* (2nd edition). Prentice-Hall, Upper Saddle River NJ 1999.
- Hegedüs T., Megiddo N.: *On the geometric separability of Boolean functions*. Discrete Applied Mathematics, vol. 66, no. 3, 1996, s. 205–218.
- Hofmeister T.: *Depth-efficient threshold circuits for arithmetic functions*. In Roychowdhury V. P., Siu K.-Y., Orłitsky A. (Eds): *Theoretical Advances in Neural Computation and Learning*, Kluwer Academic Publishers, Boston 1994, s. 37–84.
- Hofmeister T., Pudlák P.: *A proof that division is not in TC_2^0* . Research Report, no. 447, Dortmund University 1992.
- Hopcroft J. E., Ullman J. D.: *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading MA 1979.
- Hopfield J. J.: *Neural networks and physical systems with emergent collective computational abilities*. Proceedings of the National Academy of Sciences USA, vol. 79, 1982, s. 2554–2558.
- Hopfield J. J.: *Neurons with graded response have collective computational properties like those of two-state neurons*. Proceedings of the National Academy of Sciences USA, vol. 81, 1984, s. 3088–3092.
- Hopfield J. J., Tank D. W.: *“Neural” computation of decision in optimization problems*. Biological Cybernetics, vol. 52, no. 3, 1985, s. 141–152.
- Horne B. G., Hush D. R.: *On the node complexity of neural networks*. Neural Networks, vol. 7, no. 9, 1994, s. 1413–1426.
- Horne B. G., Hush D. R.: *Bounds on the complexity of recurrent neural network implementations of finite state machines*. Neural Networks, vol. 9, no. 2, 1996, s. 243–252.
- Chandra A. K., Stockmeyer L. J., Vishkin U.: *Constant depth reducibility*. SIAM Journal on Computing, vol. 13, no. 2, 1984, s. 423–439.
- Chytil M.: *Automaty a gramatiky*. Matematický seminář, vol. 19, SNTL, Praha 1984.
- Indyk P.: *Optimal simulation of automata by neural nets*. In Proceedings of the STACS’95 Twelve Annual Symposium on Theoretical Aspects of Computer Science, LNCS 900, Springer-Verlag, Berlin 1995, s. 337–348.
- Kahn J., Komlós J., Szemerédi E.: *On the probability that a random $\{\pm 1\}^n$ -matrix is singular*. Journal of the American Mathematical Society, vol. 8, no. 1, 1995, s. 223–240.
- Kilian J., Siegelmann H. T.: *The dynamic universality of sigmoidal neural networks*. Information and Computation, vol. 128, no. 1, 1996, s. 48–56.
- Kleene S. C.: *Representation of events in nerve nets and finite automata*. In Shannon C. E., McCarthy J. (Eds): *Automata Studies*, Annals of Mathematics Studies, vol. 34, Princeton University Press, Princeton NJ 1956, s. 3–41.
- Koiran P.: *Dynamics of discrete time, continuous state Hopfield networks*. Neural Computation, vol. 6, no. 3, 1994, s. 459–468.
- Koiran P.: *A family of universal recurrent networks*. Theoretical Computer Science, vol. 168, no. 2, 1996, s. 473–480.
- Komlós J., Paturi R.: *Convergence results in an associative memory model*. Neural Networks, vol. 1, no. 3, 1988, s. 239–250.
- Legenstein R. A., Maass W.: *Neural circuits for pattern recognition with small total wire length*. Theoretical Computer Science, vol. 287, no. 1, 2002, s. 239–249.
- Legenstein R. A., Maass W.: *Wire length as a circuit complexity measure*. Journal of Computer and System Sciences, vol. 70, no. 1, 2005, s. 53–72.

- Lepley M., Miller G.: *Computational power for networks of threshold devices in asynchronous environment*. Technical Report, Department of Mathematics, MIT, Cambridge MA 1983.
- Lipscomb J.: *On the computational complexity of finding a connectionist model's stable state vectors*. M.Sc. thesis, Department of Computer Science, University of Toronto 1987.
- Lupanov O. B.: *Implementing the algebra of logic functions in terms of bounded depth formulas in the basis $+, *, -$* . Soviet Physics Doklady, vol. 6, no. 2, 1961, s. 107–108.
- Lupanov O. B.: *Circuits using threshold elements*. Soviet Physics Doklady, vol. 17, no. 2, 1972, s. 91–93.
- Maass W.: *Lower bounds for the computational power of networks of spiking neurons*. Neural Computation, vol. 8, no. 1, 1996, s. 1–40.
- Maass W.: *Networks of spiking neurons: the third generation of neural network models*. Neural Networks, vol. 10, no. 9, 1997, s. 1659–1671.
- Maass W.: *On the computational power of Winner-Take-All*. Neural Computation, vol. 12, no. 11, 2000, s. 2519–2536.
- Maass W., Natschläger T., Markram H.: *Real-time computing without stable states: A new framework for neural computation based on perturbations*. Neural Computation, vol. 14, no. 11, 2002, s. 2531–2560.
- Maass W., Orponen P.: *On the effect of analog noise in discrete-time analog computations*. Neural Computation, vol. 10, no. 5, 1998, s. 1071–1095.
- Maass W., Schnitger G., Sontag E. D.: *On the computational power of sigmoid versus Boolean threshold circuits*. In Proceedings of the FOCS'91 Thirty-Second Annual Symposium on Foundations of Computer Science, IEEE Press, New York 1991, s. 767–776.
- Maass W., Sontag E. D.: *Analog neural nets with Gaussian or other common noise distribution cannot recognize arbitrary regular languages*. Neural Computation, vol. 11, no. 3, 1999, s. 771–782.
- Mahajan S., Ramesh, H.: *Derandomizing approximation algorithms based on semidefinite programming*. SIAM Journal on Computing, vol. 28, no. 5, 1999, s. 1641–1663.
- McEliece R. J., Posner E. C., Rodemich E. R., Venkatesh S. S.: *The capacity of the Hopfield associative memory*. IEEE Transactions on Information Theory, vol. 33, no. 4, 1987, s. 461–482.
- Minsky M. L.: *Theory of neural-analog reinforcement systems and its application to the brain-model problem*. Ph.D. Thesis, Princeton University, Princeton NJ 1954.
- Minsky M. L.: *Computation: Finite and Infinite Machines*. Prentice Hall, Englewood Cliffs NJ 1967.
- Minsky M. L., Papert S. A.: *Perceptrons*. The MIT Press, Cambridge MA 1969.
- Moore C.: *Finite-dimensional analog computers: flows, maps, and recurrent neural networks*. In Proceedings of the First International Conference on Unconventional Models of Computation, Springer-Verlag, Berlin 1998, s. 59–71.
- Muroga S.: *Threshold Logic and its Applications*. New York: Wiley-Interscience, New York 1971.
- Muroga S., Toda I., Takasu S.: *Theory of majority decision elements*. Journal of the Franklin Institute, vol. 271, 1961, s. 376–418.
- Nechiporuk E. I.: *The synthesis of networks from threshold elements*. Problemy Kibernetiki, vol. 11, 1964, s. 49–62.
- O'Neil P. E.: (1971). *Hyperplane cuts of an n -cube*. Discrete Mathematics, vol. 1, no. 2, 1971, s. 193–195.

- Orponen P.: *Computational complexity of neural networks: A survey*. Nordic Journal of Computing, vol. 1, no. 1, 1994, s. 94–110.
- Orponen P.: *The computational power of discrete Hopfield nets with hidden units*. Neural Computation, vol. 8, no. 2, 1996, s. 403–415.
- Orponen P.: *A survey of continuous-time computation theory*. In Du D.-Z., Ko K.-I. (Eds): *Advances in Algorithms, Languages, and Complexity*, Kluwer Academic Publishers, Dordrecht 1997a, s. 209–224.
- Orponen P.: *Computing with truly asynchronous threshold logic networks*. Theoretical Computer Science, vol. 174, no. 1-2, 1997b, s. 123–136.
- Orponen P.: *The computational power of continuous time neural networks*. In Proceedings of the SOFSEM'97 Twenty-Seventh Seminar on Current Trends in Theory and Practice of Informatics, LNCS 1338, Springer-Verlag, Berlin 1997c, s. 86–103.
- Papadimitriou C. H.: *Computational Complexity*. Addison-Wesley, Reading MA 1994.
- Parberry I.: *Circuit Complexity and Neural Networks*. The MIT Press, Cambridge MA 1994.
- Parberry I., Schnitger G.: *Relating Boltzmann machines to conventional models of computation*. Neural Networks, vol. 2, no. 1, 1989, s. 56–67.
- Poljak S., Sůra M.: *On periodical behaviour in societies with symmetric influences*. Combinatorica, vol. 3, no. 1, 1983, s. 119–121.
- Porat S.: *Stability and looping in connectionist models with asymmetric weights*. Biological Cybernetics, vol. 60, 1989, s. 335–344.
- Rabin M.: *Probabilistic automata*. Information and Control, vol. 6, no. 3, 1963, s. 230–245.
- Razborov A. A.: *On small depth threshold circuits*. In Proceedings of the SWAT'92 Third Scandinavian Workshop on Algorithm Theory, LNCS 621, Springer-Verlag, Berlin 1992, s. 42–52.
- Reif J. H., Tate S. R.: *On threshold circuits and polynomial computations*. SIAM Journal on Computing, vol. 21, no. 5, 1992, s. 896–908.
- Rosenblatt F.: *The perceptron: A probabilistic model for information storage and organization in the brain*. Psychological Review, vol. 65, no. 6, 1958, s. 386–408.
- Roychowdhury V. P., Siu K.-Y., Orlitsky A. (Eds): *Theoretical Advances in Neural Computation and Learning*. Kluwer Academic Publishers, Boston 1994.
- Rumelhart D. E., Hinton G. E., Williams R. J.: *Learning representations by back-propagating errors*. Nature, vol. 323, 1986, s. 533–536.
- Savage J. E.: *Computational work and time on finite machines*. Journal of the ACM, vol. 19, no. 4, 1972, s. 660–674.
- Schläfli L: *Theorie der vielfachen Kontinuität*. Zürcher & Furrer, Zürich 1901.
- Siegelmann H. T.: *On the computational power of probabilistic and faulty neural networks*. In Proceedings of the ICALP'94 Twenty-First International Colloquium on Automata, Languages, and Programming, LNCS 820, Springer-Verlag, Berlin 1994, s. 23–34.
- Siegelmann H. T.: *Recurrent neural networks and finite automata*. Journal of Computational Intelligence, vol. 12, no. 4, 1996, s. 567–574.
- Siegelmann H. T.: *Neural Networks and Analog Computation: Beyond the Turing Limit*. Birkhäuser, Boston 1999a.
- Siegelmann H. T.: *Stochastic analog networks and computational complexity*. Journal of Complexity, vol. 15, no. 4, 1999b, s. 451–475.

- Siegelmann H. T., Roitershtein A., Ben-Hur A.: *Noisy neural networks and generalizations*. In Solla S. A., Leen T. K., Müller K.-R. (Eds): *Advances in Neural Information Processing Systems (NIPS'99)*, vol. 12, The MIT Press, Cambridge MA 2000, s. 335–341.
- Siegelmann H. T., Sontag E. D.: *Analog computation via neural networks*. *Theoretical Computer Science*, vol. 131, no. 2, 1994, s. 331–360.
- Siegelmann H. T., Sontag E. D.: *Computational power of neural networks*. *Journal of Computer System Science*, vol. 50, no. 1, 1995, s. 132–150.
- Siu K.-Y., Bruck J., Kailath T., Hofmeister T.: *Depth efficient neural networks for division and related problems*. *IEEE Transactions on Information Theory*, vol. 39, no. 3, 1993, s. 946–956.
- Siu K.-Y., Roychowdhury V. P.: *On optimal depth threshold circuits for multiplication and related problems*. *SIAM Journal on Discrete Mathematics*, vol. 7, no. 2, 1994, s. 284–292.
- Siu K.-Y., Roychowdhury V. P., Kailath T.: *Depth-size tradeoffs for neural computation*. *IEEE Transactions on Computers*, vol. 40, no. 12, 1991, s. 1402–1412.
- Siu K.-Y., Roychowdhury V. P., Kailath T.: *Discrete Neural Computation: A Theoretical Foundation*. Prentice Hall, Englewood Cliffs NJ 1995.
- Šíma J.: *Analog stable simulation of discrete neural networks*. *Neural Network World*, vol. 7, no. 6, 1997, s. 679–686.
- Šíma J.: *Training a single sigmoidal neuron is hard*. *Neural Computation*, vol. 14, no. 11, 2002, s. 2709–2728.
- Šíma J.: *Energy-based computation with symmetric Hopfield nets*. In Ablameyko S., Gori M., Goras L., Piuri V. (Eds): *Limitations and Future Trends in Neural Computation*, NATO Science Series: Computer & Systems Sciences, vol. 186, IOS Press, Amsterdam 2003, s. 45–69.
- Šíma J., Neruda R.: *Teoretické otázky neuronových sítí*. Matfyzpress, Praha 1996.
- Šíma J., Orponen P.: *Continuous-time symmetric Hopfield nets are computationally universal*. *Neural Computation*, vol. 15, no. 3, 2003a, s. 693–733.
- Šíma J., Orponen P.: *Exponential transients in continuous-time Liapunov Systems*. *Theoretical Computer Science*, vol. 306, no. 1-3, 2003b, s. 353–372.
- Šíma J., Orponen P.: *General-purpose computation with neural networks: A survey of complexity theoretic results*. *Neural Computation*, vol. 15, no. 12, 2003c, s. 2727–2778.
- Šíma J., Orponen P., Antti-Poika T.: *On the computational complexity of binary and analog symmetric Hopfield nets*. *Neural Computation*, vol. 12, no. 12, 2000, s. 2965–2989.
- Šíma J., Wiedermann J.: *Theory of neuromata*. *Journal of the ACM*, vol. 45, no. 1, 1998, s. 155–178.
- Šorel M., Šíma J.: *Robust RBF finite automata*. *Neurocomputing*, vol. 62, 2004, s. 93–110.
- Tanaka F., Edwards S. F.: *Analytic theory of the ground state properties of a spin glass: I. Ising spin glass*. *Journal of Physics F: Metal Physics*, vol. 10, 1980, s. 2769–2778.
- Tchuente M.: *Sequential simulation of parallel iterations and applications*. *Theoretical Computer Science*, vol. 48, no. 2-3, 1986, s. 135–144.
- Uchizawa K., Douglas R., Maass W.: *Energy complexity and entropy of threshold circuits*. In *Proceedings of the ICALP 2006 Thirty-Third International Colloquium on Automata, Languages and Programming, Part I, LNCS 4051*, Springer-Verlag, Berlin 2006, s. 631–642.
- Vidyasagar M.: *A Theory of Learning and Generalization (With Applications to Neural Networks and Control Systems)*. Springer-Verlag, London 1997.

- Vollmer H.: *Introduction to Circuit Complexity*. Springer-Verlag, Berlin 1999.
- von Neumann J.: *The general and logical theory of automata*. In Jeffress L. A. (Ed): *Cerebral Mechanisms in Behavior*, Wiley, New York 1951, s. 1–41.
- von Neumann J.: *Probabilistic logics and the synthesis of reliable organisms from unreliable components*. In Shannon C. E., McCarthy J. (Eds): *Automata Studies*, Annals of Mathematics Studies, vol. 34, Princeton University Press, Princeton NJ 1956, s. 43–98.
- Wegener I.: *The Complexity of Boolean Functions*. Wiley/Teubner, Chichester 1987.
- Wegener I.: *Optimal lower bounds on the depth of polynomial-size threshold circuits for some arithmetic functions*. Information Processing Letters, vol. 46, no. 2, 1993, s. 85–87.
- Wiedermann J.: *Complexity issues in discrete neurocomputing*. Neural Network World, vol. 4, no. 1, 1994, s. 99–119.
- Yao A. C.-C.: *Separating the polynomial-time hierarchy by oracles*. In Proceedings of the FOCS'85 Twenty-Sixth Annual Symposium on the Foundations of Computer Science, IEEE Computer Society, New York 1985, s. 420–428.

Příloha

$g = \mathbf{O}(f(\mathbf{n}))$: $(\exists \varepsilon > 0) (\exists n_0) (\forall n \geq n_0) g(n) < \varepsilon \cdot f(n)$.

$g = \mathbf{\Omega}(f(\mathbf{n}))$: $(\exists \varepsilon > 0) (\exists n_0) (\forall n \geq n_0) g(n) > \varepsilon \cdot f(n)$.

$g = \mathbf{\Theta}(f(\mathbf{n}))$: $g = O(f(n))$ a $g = \Omega(f(n))$.

P: Třída jazyků rozpoznatelných (deterministickým) Turingovým strojem v polynomiálním čase, tj. během prvních $T(n) = O(n^c)$ výpočetních kroků stroje pro nějakou pevnou konstantu c , kde n je délka vstupu (např. počet bitů jeho binární reprezentace). Třída P obsahuje problémy, které jsou považovány za výpočetně zvládnutelné.

NP: Třída jazyků rozpoznatelných nedeterministickým Turingovým strojem v polynomiálním čase. Nedeterministický Turingův stroj si může v každém výpočetním kroku zvolit svoji další akci obecně z množiny několika (bez újmy na obecnosti dvou) možností. Z toho vyplývá, že pro daný vstup neexistuje obecně jen jeden (deterministický) výpočet (cesta), ale celá množina (strom) možných výpočtů. Pro přijetí vstupního slova nedeterministickým Turingovým strojem musí existovat aspoň jeden přijímající výpočet v této množině. Jinými slovy třída NP obsahuje ty problémy, jejichž řešení lze nejprve uhádnout (odpovídá nedeterministické volbě přijímajícího výpočtu z množiny možných výpočtů) a pak ověřit jeho správnost (zda tento výpočet je opravdu přijímající) v polynomiálním čase. Například jazyk splnitelných booleovských formulí SAT je v NP, protože pro danou booleovskou formuli lze uhádnout ohodnocení příslušných proměnných a dosazením ověřit v polynomiálním čase, zda toto ohodnocení formuli splňuje.

A je NP-těžký: Každý problém z NP lze převést (redukovat) na A v polynomiálním čase, tj. pro každý B v NP existuje funkce f , kterou je možné počítat v polynomiálním čase a která transformuje vstup x pro problém B na nějaký vstup $f(x)$ pro problém A tak, že $x \in B$, právě když $f(x) \in A$. Tedy vyřešením jediného NP-těžkého problému v polynomiálním čase bychom uměli řešit všechny problémy v NP v polynomiálním čase (tj. $P=NP$).

A je NP-úplný: A je NP-těžký a A je v NP. Tedy třída NP-úplných problémů obsahuje nejtěžší problémy v NP, o kterých se předpokládá, že nejsou výpočetně zvládnutelné (tj. $P \neq NP$). Například problém splnitelných booleovských formulí SAT je NP-úplný.

- co-NP:** Třída jazyků, jejichž doplňky jsou rozpoznatelné nedeterministickým Turingovým strojem v polynomiálním čase.
- A je co-NP-úplný:** A je co-NP a každý problém z co-NP lze převést na A v polynomiálním čase.
- PSPACE:** Třída jazyků rozpoznatelných (deterministickým) Turingovým strojem pracujícím v polynomiálním prostoru, tj. s použitím jen $S(n) = O(n^c)$ políček pásky pro nějakou pevnou konstantu c , kde n je délka vstupu. Platí $P \subseteq NP \subseteq PSPACE$ a $P \subseteq co-NP \subseteq PSPACE$, ale doposud nebylo dokázáno, že tyto inkluze jsou ostré, i když se to očekává.
- A je PSPACE-úplný:** A je v PSPACE a každý problém z PSPACE lze převést na A v polynomiálním čase. Množina pravdivých kvantifikovaných booleovských formulí (bez volných proměnných) QBF je příkladem PSPACE-úplného problému.
- BPP:** Třída jazyků rozpoznatelných pravděpodobnostním Turingovým strojem v polynomiálním čase s pravděpodobností chyby shora omezenou nějakou kladnou konstantou $\varepsilon < 1/2$. Pravděpodobnostní Turingův stroj je nedeterministický Turingův stroj, který může v každém výpočetním kroku volit z právě dvou různých možných akcí. Navíc pro daný vstup je počet kroků ve všech možných výpočtech stejný a každý takový výpočet skončí v koncovém stavu, ve kterém vstup buď přijímá, nebo zamítá. Pro libovolný vstup x je pravděpodobnost chyby definována jako podíl počtu výpočtů nad x s chybnou odpovědí k celkovému počtu výpočtů nad x . Problémy z třídy BPP jsou považovány za algoritmicky zvládnutelné problémy a platí $P \subseteq BPP \subseteq PSPACE$, ale doposud neumíme dokázat, že tyto inkluze jsou ostré.
- P/poly:** Třída jazyků rozpoznatelných Turingovým strojem v polynomiálním čase s polynomiální poradní (advice) funkcí. Poradní funkce f , která nemusí být rekurzivní, poskytuje Turingovu stroji na začátku výpočtu externí radu ve tvaru řetězce $f(n)$, která závisí jen na délce vstupu n . Pro délku tohoto řetězce u polynomiálních poradních funkcí platí $|f(n)| = O(n^c)$ pro nějakou pevnou konstantu c .
- PSPACE/poly:** Třída jazyků rozpoznatelných Turingovým strojem s polynomiálním prostorem a polynomiální poradní funkcí.
- Pref-BPP/log:** Třída jazyků rozpoznatelných pravděpodobnostním Turingovým strojem v polynomiálním čase s omezenou pravděpodobností chyby a logaritmickou poradní funkcí f ($|f(n)| = O(\log n)$), která je uzavřená na prefixy, tj. $f(n_1)$ je prefixem $f(n_2)$ pro každé $n_1 < n_2$.