

Animace činnosti umělé neuronové sítě v prostředí Mathematica

Animation of The Artificial Neural Network Activity in The Mathematica Environment

Tomáš Vinkler

Bakalářská práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš VINKLER**
Osobní číslo: **A09171**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Animace činnosti umělé neuronové sítě v prostředí Mathematica**

Zásady pro vypracování:

1. Seznamte se s animačními technikami v softwaru Mathematica.
2. Popište činnost umělé neuronové sítě pro účely vytvoření animace v softwaru Mathematica.
3. Vytvořte animace s možnostmi nastavování parametrů umělých neuronových sítí.
4. Popište možnosti exportu vytvořených animací do jiných formátů.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. WOLFRAM, Stephen. The mathematica book. 5th ed. Champaign: Wolfram Media, 2003. ISBN 1579550223.
2. CHRAMCOV, Bronislav. Základy práce v prostředí Mathematica. Vyd. 2. Zlín: Univerzita Tomáše Bati, 2006. ISBN 80-7318-510-5.
3. Wolfram [online]. 2012 [cit. 2012-01-23]. Dostupné z: <http://www.wolfram.com/>
4. Dokumenty. ELKAN, spol. s r.o. ELKAN, spol. s r.o. [online]. [cit. 2012-01-23]. Dostupné z: <http://www.mathematica.cz/dokumenty.php>
5. BÍLA J.: Umělá inteligence a neuronové sítě v aplikacích, ČVUT, 1996, ISBN 80-01-01275-1.
6. ZELINKA I.: Umělá inteligence I, VUT Brno, 1998, ISBN 80-214-1163-5.

Vedoucí bakalářské práce:

Ing. Zuzana Oplatková, Ph.D.

Ústav informatiky a umělé inteligence


Datum zadání bakalářské práce:

24. února 2012

Termín odevzdání bakalářské práce:

8. června 2012

Ve Zlíně dne 24. února 2012


prof. Ing. Vladimír Vašek, CSc.
děkan




prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Hlavní náplní této bakalářské práce je popis činností umělých neuronových sítí a vytvoření jejich animací pomocí animačních technik a funkcí pro vytváření grafických výstupů softwaru Mathematica. Dále popisuje jednotlivé grafické formáty, do kterých lze tyto animace exportovat.

Klíčová slova: Mathematica, animační techniky, umělé neuronové sítě, export

ABSTRACT

The main purpose of this bachelor thesis is to describe how the neural networks work and to create it's animations using animation functions and techniques, which are built in Mathematica software. An individual graphic formats of animation export will also be described.

Keywords: Mathematica, animation techniques, artificial neural network, export

Tímto bych chtěl poděkovat Ing. Zuzaně Oplatkové, Ph.D. za cenné připomínky a rady k této práci.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST	10
1 PROGRAMOVÁNÍ V SOFTWARE MATHEMATICA	11
1.1 ZÁKLADY PROGRAMOVÁNÍ.....	12
1.2 GRAFICKÉ VÝSTUPY SOFTWARE MATHEMATICA	13
1.2.1 Funkce Plot.....	14
1.2.2 Funkce Graphics.....	15
1.2.3 Funkce Show	16
1.2.4 Další vybrané grafické výstupy	16
1.3 ANIMAČNÍ TECHNIKY SOFTWARE MATHEMATICA	17
1.3.1 Funkce Manipulate	17
1.3.2 Funkce Animate	18
1.4 EXPORT	19
1.4.1 AVI (Audio Video Interleave).....	20
1.4.2 FLV (Flash Video) a SWF (Shockwave flash).....	20
1.4.3 QuickTime.....	20
1.4.4 GIF (The Graphics Interchange Format)	21
2 UMĚLÉ NEURONOVÉ SÍTĚ	22
2.1 DĚLENÍ SÍTÍ	23
2.1.1 Podle počtu vrstev	23
2.1.2 Podle algoritmu učení.....	24
2.1.3 Podle stylu učení	24
2.2 PŘENOSOVÉ FUNKCE	24
2.2.1 Funkce perceptron	25
2.2.2 Funkce binární.....	25
2.2.3 Funkce Logistická	25
2.2.4 Funkce hyperbolický tangens	25
2.3 PRINCIP NEURONOVÉ SÍTĚ.....	25
2.4 SCHÉMA NEURONOVÉ SÍTĚ	27
2.5 TYPY NEURONOVÝCH SÍTÍ	28
2.5.1 Perceptron.....	28
2.5.2 Vícevrstvá síť s algoritmem Backpropagation	30
2.5.3 Hopfieldova síť	30
2.5.4 Síť CLN (Competitive Learning Network)	31
2.5.5 Síť LAM (Linear Associative Memory).....	32
2.5.6 Síť BAM (Bidirectional Associative Memory).....	32
2.5.7 Kohonenova síť	33
2.5.8 Síť ART (Adaptive Resonant neTwork)	34
2.6 OPTIMALIZACE TOPOLOGIE NEURONOVÉ SÍTĚ	35
2.6.1 Počet vrstev	35
2.6.2 Počet neuronů ve vrstvách.....	35

II	PRAKTICKÁ ČÁST	37
3	ANIMACE UMĚLÝCH NEURONOVÝCH SÍTÍ	38
3.1	ANIMACE JEDNODUCHÉHO PERCEPTRONU	38
3.2	ANIMACE DOPŘEDNÉ VÍCEVRSTVÉ NEURONOVÉ SÍTĚ.....	40
3.3	VIZUALIZACE TOPOLOGIE NEURONOVÝCH SÍTÍ	41
3.3.1	Dopředná vícevrstvá síť	41
3.3.2	Hopfieldova síť	43
3.3.3	Síť CLN.....	44
3.3.4	Síť BAM.....	45
3.3.5	Kohonenova síť	46
3.3.6	Síť ART.....	47
4	EXPORT VYTVOŘENÝCH ANIMACÍ	48
4.1	EXPORT ANIMACE JEDNODUCHÉHO PERCEPTRONU	48
4.1.1	Export AVI.....	49
4.1.2	Export GIF.....	50
4.2	EXPORT ANIMACE DOPŘEDNÉ VÍCEVRSTVÉ SÍTĚ	51
4.2.1	Export FLV	51
4.2.2	Export SWF.....	52
	ZÁVĚR	53
	ZÁVĚR V ANGLIČTINĚ.....	54
	SEZNAM POUŽITÉ LITERATURY.....	55
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	58
	SEZNAM OBRÁZKŮ	59
	SEZNAM PŘÍLOH.....	61

ÚVOD

Historie neuronových sítí spadá do první poloviny 20. století. Období od roku 1943 do roku 1969 je považováno za dobu vzniku tohoto oboru. Ve 40. letech minulého století fyziologové McCulloch, Pitts a Hebb ve svých pojednáních zavedli normu grafického znázorňování struktury umělých neuronových sítí. V roce 1958 vytvořil F. Rosenblatt první funkční neuronovou síť. Tato síť měla však velké nedostatky, čehož využil M. Minsky spolu s S. Papertem a publikovali knihu „Perceptron“, která upozorňovala na tyto nedostatky Rosenblattovy sítě. Díky tomu odpadl zájem o neuronové sítě až do roku 1984, kdy J. Hopfield spustil druhou vlnu zájmu tím, že konstruoval neuronové sítě zvané autoasociativní. V 80. letech vznikaly i další typy sítí jako Kohonenova síť a Grossbergova ART síť. Význam a užití neuronových sítí nabývá stále většího uplatnění.

Cílem této bakalářské práce je vytvoření animace činnosti umělé neuronové sítě v softwaru Mathematica tak, aby měl uživatel možnost nastavovat jednotlivé parametry, které tuto animaci ovlivňují.

Teoretická část se dělí na dvě kapitoly. První kapitola popisuje jak základní příkazy a funkce, tak i složité techniky pro vytváření animací v softwaru Mathematica. Dále se zabývá funkcí Export a multimediálními formáty. Druhá kapitola se věnuje samotným umělým neuronovým sítím.

Praktická část je zaměřená na možnosti ovládání vytvořených animací. Jsou zde popsány veškeré prvky jednotlivých modulů, použité funkce i algoritmy vytvořených animací. V závěru této části jsou popsány možnosti exportu u vybraných multimediálních i grafických formátů.

I. TEORETICKÁ ČÁST

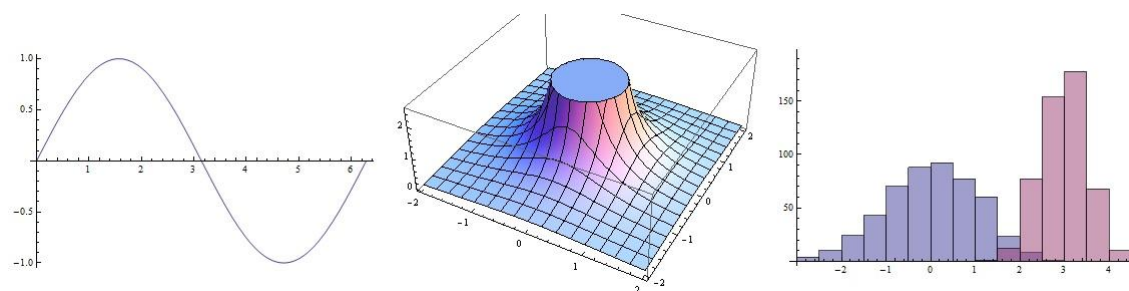
1 PROGRAMOVÁNÍ V SOFTWARE MATHEMATICA

Jedná se o nový symbolický jazyk, jehož programování umožňuje představovat data, funkce, grafy, programy a dokonce i celý dokument jako symbolický výraz. Jednoduchost kódování je založena na tom, že funkce zde může reprezentovat jak vstup, tak i výstup další funkce.

V programovacím jazyce softwaru Mathematica není zapotřebí předeclarovat typ proměnné, rozměr matice, překládat naprogramovaný kód ani přímé řízení paměti. Veškeré procesy jako seřazení, vyhledávání a zacházení s daty jsou již v programovacím jazyce zabudovány [1].

Základem programu Mathematica je **notebook** – soubor, do něhož se píše veškeré příkazy. Součástí je také nápověda systému Mathematica. Jednoduchost této nápovědy spočívá ve struktuře popisu veškerých funkcí nebo příkazů (název, syntaxe zápisu, více informací). Nápověda dále umožňuje zobrazit nejběžnější příklady dané funkce, interaktivní odkazy na funkce a příkazy, anglicky psané návody na práci s danou funkcí nebo příkazem a dodatečné informace a komentáře. S předdefinovanými příklady příkazů je možné také manipulovat (měnit je, doplňovat) a následně vyzkoušet, aniž by se změnil kód uživatelského programu. Tyto příklady je možné kopírovat do vlastních notebooků.

Mathematica umožňuje uživateli použít velké množství grafických vzorů pro vykreslení výsledků. Obsahuje 2D, 3D, vrstevnicové grafy, grafy hustoty a také ekonomické a statistické grafy (Obr. 1)



Obr. 1 - Ukázka zobrazovacích možností softwaru Mathematica

1.1 Základy programování

Mathematica umožňuje vytvářet vlastní programy pomocí operátorů, lokálních a globálních proměnných, funkcí pro vytváření cyklů, podmínek, a dalších funkcí. Základem programování jsou závorky. Mathematica využívá čtyři druhy závorek [2]:

- Kulaté `()` – uzavření matematických výrazů
- Hranaté `[]` – parametry funkcí
- Složené `{}` – sloučení více objektů
- Dvojitě hranaté `[[]]` – adresování

Relační operátory se využívají k zjištění, zda jsou si dvě čísla rovna nebo jestli je jedno menší než druhé. Veškeré příkazy, které obsahují relační operátor, se vyhodnotí jako pravdivý nebo nepravdivý. Existuje dvojí způsob zápisu těchto operátorů:

- `x==y` nebo `Equal[x,y]` – srovnání dvou čísel, pravda pro rovnost x a y
- `x===y` nebo `SameQ[x,y]` – totožnost, pravdivé když x a y jsou totožné
- `x!=y` nebo `Unequal[x,y]` – číselné srovnání, pravda pro nerovnost x a y
- `x!==y` nebo `UnsameQ[x,y]` – totožnost, pravda pokud x a y nejsou totožné
- `x<y` nebo `Less[x,y]` – pravda, pokud x je menší než y
- `x>y` nebo `Greater[x,y]` – pravda, pokud x je větší než y
- `x<=y` nebo `LessEqual[x,y]` – pravda, pokud x je menší nebo rovno y
- `x>=y` nebo `GreaterEqual[x,y]` – pravda, pokud x je větší nebo rovno y

Logické operátory se používají nejen v programování, ale i v matematických operacích.

- `And[výraz1,výraz2,...]` – logický součin
- `Or[výraz1,výraz2,...]` – logický součet
- `Nand[výraz1,výraz2,...]` – negace logického součinu
- `Nor[výraz1,výraz2,...]` – negace logického součtu
- `Xor[výraz1,výraz2,...]` – nonekvivalence (exklusive OR) [1]

Mathematica využívá různé typy podmínkových funkcí, které vyhodnocují dané výrazy jen tehdy, pokud splní určené podmínky.

- `If[test, true, false]` – na základě vyhodnocené podmínky *test* se provede *true* nebo *false*
- `Which[test1, value1, test2, value2, ...]` – vyhodnotí více podmínek *test*, vrátí příslušnou hodnotu *value*
- `Switch[expr, form1, value1, form2, value2, ...]` – vyhodnotí výraz *expr*, porovná jej se všemi hodnotami *form*. Vrací výsledek *value*, který odpovídá první nalezené rovnosti mezi výrazem *expr* a hodnotami *form*

Pro opakované vyhodnocení výrazů slouží funkce pro vytváření cyklů (smýček).

- `Do[expr, {i, imin, imax, di}]` – vyhodnotí výraz *expr* pro proměnnou *i*, přičemž *i* začíná hodnotou *i_{min}* a končí hodnotou *i_{max}* po kroku *di*
- `While[test, body]` – opakovaně vyhodnotí podmínku *test*, pokud je tato podmínka splněna, provede se tělo cyklu *body* (pokud je podmínka pravdivá, cyklus se neustále opakuje)
- `For[start, test, incr, body]` – pokud je podmínka *test* splněna, provede výrazy těla cyklu *body*, kdy v každém cyklu zvyšuje hodnotu proměnné (její počáteční hodnota je dána hodnotou *start*) o *incr*
- `Table[expr, {i, imin, imax, di}]` – generuje seznam hodnot výrazu *expr* pro *i* od *i_{min}* do *i_{max}* po kroku *di* [3]

1.2 Grafické výstupy softwaru Mathematica

Mathematica díky symbolické architektuře a dynamickému rozhraní umožňuje jedinečný a pohodlný přístup k vytváření grafů a vizualizaci informací. Dokáže okamžitě data vzít a vytvářet působivé dynamické vizualizace v nejrůznějších formátech. Mathematica obsahuje velké množství funkcí, jimiž dokáže zobrazit výsledky provedeného výpočtu [1].

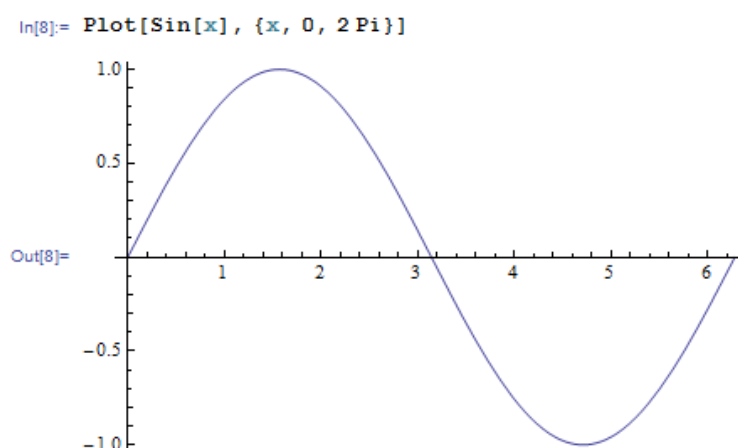
1.2.1 Funkce Plot

Funkce `Plot` slouží k vizualizaci funkce v daném intervalu (Obr. 2). Její zápis je následující:

$$\text{Plot}[f, \{x, x_{\min}, x_{\max}\}, \text{options}]$$

Parametrem funkce je vykreslovaná funkce f a nezávislá proměnná x . Dalším nezbytným parametrem je dolní mez x_{\min} a horní mez x_{\max} , mezi kterými je graf funkce vykreslován. Nápopěda umožňuje vyhledat parametr *options* této funkce, díky kterému lze tyto grafy upravovat. Mezi nejpoužívanější patří:

- *AxesOrigin* – nastavení průsečíku os kartézského součinu, do něhož je graf vykreslován, přičemž s použitím možnosti *Automatic* se program snaží vykreslit graf optimálně
- *AspectRatio* – nastavení poměru výšky a šířky v grafu, pokud není u kružnic tento parametr nastaven na *Automatic*, může se kružnice vykreslit jako elipsa
- *PlotRange* – rozsah zobrazovaného grafu, využívá se pro snadnější přechzení důležitých vlastností funkcí (např. extrémů)
- *AxesLabel* – nastavení a zobrazení jednotlivých popisků os
- *PlotStyle* – slouží k vizuálním úpravám průběhu funkce (nastavení barvy, tloušťky, atd.) [4]



Obr. 2 - Vizualizace funkce pomocí příkazu `Plot`

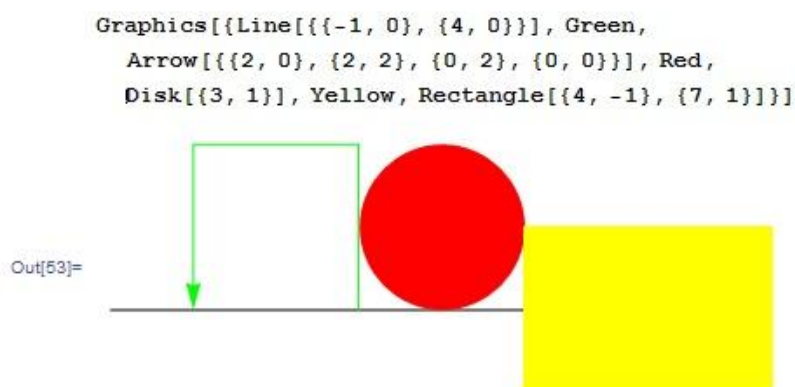
1.2.2 Funkce Graphics

Funkce slouží pro vizualizaci objektů v jednom grafickém módu (Obr. 3). Funkci lze zapsat tímto způsobem:

$$\text{Graphics}[\text{primitives}, \text{options}]$$

Parametr *options* této funkce je obdobný jako u funkce `Plot`. Za výraz *primitives* lze dosazovat řadu funkcí pro vykreslení objektů, které se mohou pomocí této funkce zobrazit. Mezi nejzákladnější patří:

- `Point[coords]` – bod o souřadnicích *coords*
- `Line[{pt1, pt2, ...}]` – čára procházející body *pt*
- `Arrow[{pt1, pt2}]` – vektor o určitém počátečním bodě *pt₁* a koncovém bodě *pt₂*
- `Circle[{x,y},r]` – kružnice se středem v bodě o souřadnicích $\{x,y\}$ a poloměru *r*
- `Disk[{x,y},r]` – kruh se středem v bodě $\{x,y\}$ a poloměru *r*
- `Rectangle[{xmin,ymin},{xmax,ymax}]` – obdélník o levém dolním a pravém horním vrcholu o souřadnicích $\{x_{min},y_{min}\}$, $\{x_{max},y_{max}\}$
- `Text[expr,coords]` – zobrazení výrazu *expr* v textovém formátu se středem v bodě *coords*



Obr. 3 - Vizualizace objektů pomocí funkce *Graphics*

Každý ze zmiňovaných objektů má své parametry, kterými mohou jakkoli upravovat svůj vzhled (barva, styl, otočení, tloušťka, atd.) [5].

1.2.3 Funkce Show

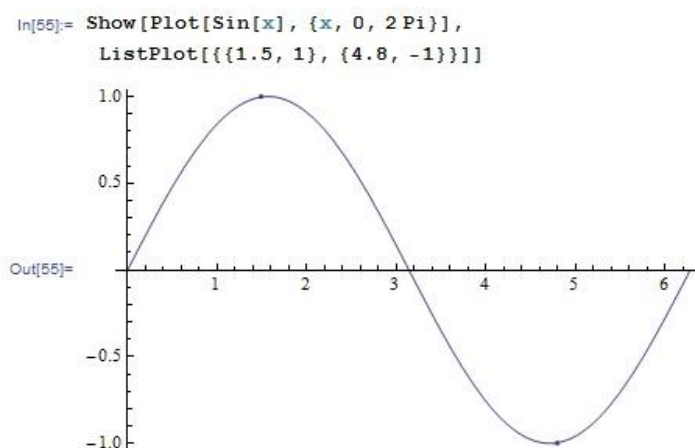
Slouží pro zobrazení několika grafických výstupů do jednoho grafu (Obr. 4). Této funkce se využívá při vizualizaci bodů křivek, grafu závislostí spolu se schématem zapojení, atd. `Show` lze zapsat dvěma způsoby, přičemž každý plní jinou funkci [2].

První zápis přidá k funkci *graphics* další možné nastavení *options*.

`Show[graphics, options]`

Druhý zápis umožňuje vykreslit kombinace funkcí *g*, demonstrované na obrázku (Obr. 4).

`Show[g1, g2, ...]`



Obr. 4 - Zobrazení grafických výstupů do jednoho grafu pomocí *Show*

1.2.4 Další vybrané grafické výstupy

Tyto 2D grafické výstupy jsou obdobné k těm, které byly uvedeny výše. Využívají se pouze pro jinak zadané funkce. Jejich parametry jsou analogické k parametrům funkce `Plot`. Zápis vybraných grafických výstupů je následující [5]:

- `ListPlot[{{x1, y1}, {x2, y2}, ...}]` – vykreslení bodů o souřadnicích $\{x, y\}$
- `ParametricPlot[{fx, fy}, {u, umin, umax}]` – Vykreslení parametricky zadaného grafu funkce jako závislost x -ové a y -ové souřadnice na parametru u : $x(u)$ a $y(u)$, kde u je z intervalu od u_{min} do u_{max} .
- `PolarPlot[r, {θ, θmin, θmax}]` – Vizualizace grafu funkce do polárních souřadnic o poloměru r a úhlu $θ$ v intervalu od $θ_{min}$ do $θ_{max}$.

Další grafické výstupy slouží pro vykreslení grafů do trojrozměrného prostoru. Zápis syntaxe se mění pouze doplněním znaky „3D” k dané funkci a přidáním do parametrů této funkce další proměnné, pro kterou je graf vykreslován. Tyto 3D grafické výstupy jsou obdobné 2D grafickým výstupům.

- `Plot3D[f, {x, xmin, xmax}, {y, ymin, ymax}]` – vykreslení funkce f o proměnných x a y v intervalu od x_{min} do x_{max} a y_{min} a y_{max}
- `RevolutionPlot3D[fz, {t, tmin, tmax}]` – jedním z využití tohoto grafického výstupu je vykreslení plochy vzniklé rotací dané funkce f_z kolem osy na daném intervalu od t_{min} do t_{max} [2,6]

1.3 Animační techniky softwaru Mathematica

Program Mathematica přinesl od verze 6.0 další dvě nové klíčové funkce v oblasti vizualizace – `Manipulate` a `Animate`. Obě tyto funkce umožňují uživateli jednoduše vytvářet plně interaktivní grafiku. Hlavní funkcí je `Manipulate`, která definuje nový směr vizualizace. Jedná se o vytváření nových grafických aplikací, které umožňují interaktivní změnu hodnot zvolených proměnných, díky čemuž tato změna probíhá prostřednictvím grafických prvků.

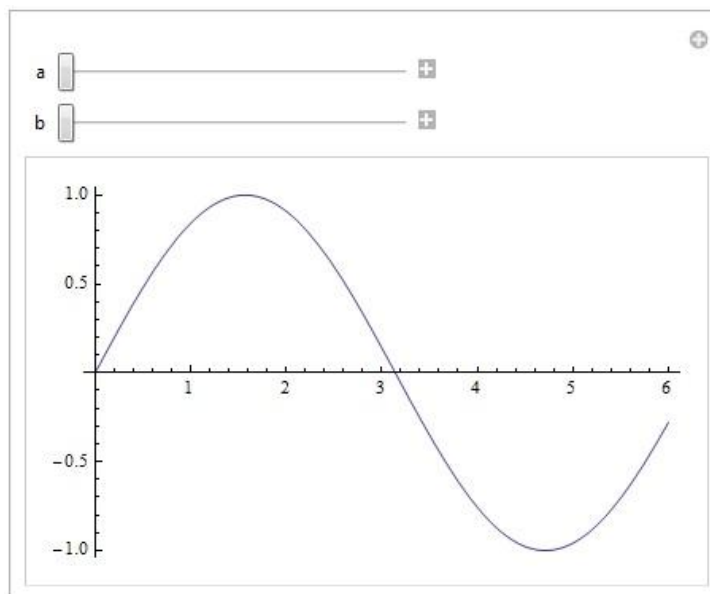
1.3.1 Funkce Manipulate

Funkce `Manipulate` vytváří dynamický modul, ve kterém přiřadí ke všem lokálním proměnným kontrolery, pomocí nichž může snadno měnit hodnoty proměnných tažením myši (Obr. 5). Pomocí standardních ovládacích prvků lze měnit parametry prováděné simulace. Mezi tyto ovládací prvky patří tlačítka, rolovací seznamy, posuvníky a další standardní vybavení grafických aplikací. Novější verze programu přinesly další užitečné ovládací prvky `Locator` (bod, se kterým může uživatel pohybovat) a `Slider2D` (pohyblivý bod). Zápis této funkce je:

$$\text{Manipulate}[\text{expr}, \{u, u_{min}, u_{max}\}]$$

Funkce obsahuje několik parametrů. Prvním parametrem je výraz `expr`, který bude zobrazen společně s ovládacími prvky. Mezi tyto výrazy patří nejčastěji funkce pro vykreslení grafů (`Plot`, `Plot3D`, atd.) nebo funkce `Graphics` sloužící pro zobrazení několika objektů do jednoho grafického módu. Dalšími parametry je definice

proměnných, které obsahují tři hlavní prvky - název u , minimální hodnotu u_{min} a maximální hodnotu u_{max} . Kombinací vytvořených funkcí a funkce `Manipulate` získáme na základě parametrů, které byly nastaveny uživatelem, dynamicky vykreslovaný objekt (Obr. 5) [7].



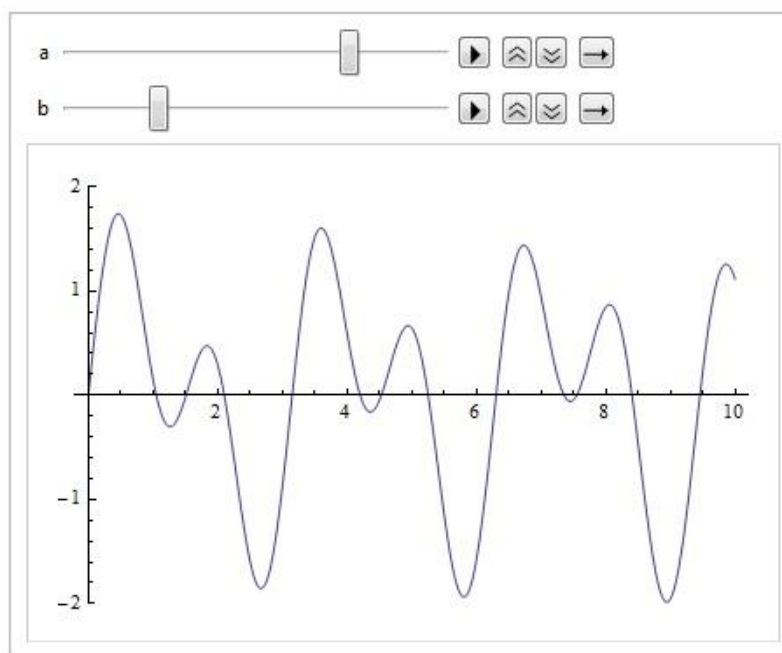
Obr. 5 – Funkce `Manipulate`

1.3.2 Funkce `Animate`

Funkce `Animate` taktéž dynamicky mění hodnoty parametrů, ale narozdíl od `Manipulate` neposkytuje interaktivní zásahy. Zajišťuje pouze dosazení předem daných posloupností hodnot za zvolené proměnné. V kombinaci s vykreslovacími funkcemi umožňuje vytvářet animace (Obr. 6).

$$\text{Animate}[\text{expr}, \{u, u_{min}, u_{max}\}]$$

Prvním parametrem stejně jako u funkce `Manipulate` je výraz expr , kterým může být opět funkce pro vykreslování grafů. Dalším parametrem je seznam proměnných (proměnná u v intervalu od u_{min} do u_{max}), které animaci ovlivňují a jsou obsaženy ve funkci expr [8].



Obr. 6 – Funkce Animate

1.4 Export

Mathematica umožňuje exportovat obsah notebooku až do 116 různých formátů, mezi které patří kromě běžných bitmapových a vektorových formátů také formáty vědecké (FITS, HDF, atd.), formáty popisující 3D modely (např. 3DS) a webové formáty. Zápis této funkce je následující:

```
Export["cesta/název.format", funkce]
```

Prvního parametr určuje, kam se má daná funkce exportovat. Formát exportu je libovolný. Uživatel si může vybírat ze 116 možných formátů. Druhým parametrem je myšlena funkce, která se má exportovat (data, grafický výstup, atd.). Funkci lze zapsat i tímto způsobem:

```
Export["cesta/název.format", funkce, element]
```

Jedná se o zápis, který má navíc třetí parametr *element*. Tento parametr slouží pro rozšíření nastavení exportu (rozlišení, komprese, atd.). Animace, které mají parametry simulované funkce dány na svém začátku a neumožňují interaktivní zásahy je možno exportovat ve video formátech (AVI, FLV, atd.) [9].

1.4.1 AVI (Audio Video Interleave)

AVI (Audio Video Interleave) je jeden z nejstarších a nejrozšířenější kontejnerů pro ukládání videa, vynalezen společností Microsoft v roce 1992 jako součást multimediální technologie Video for Windows. AVI ukládá jak video, tak i audio data do jednoho souboru, což umožňuje synchronní audio-video přehrávání.

Formát AVI má značné nedostatky při použití nejnovějších kodeků (např. XviD nebo DivX), kdy dochází k značnému zvětšení velikosti souboru. Proto je dnes tento formát považován za zastaralý, ovšem díky své kompatibilitě a spoustou programů pro jeho přehrávání a editaci zůstal stále rozšířeným [10].

1.4.2 FLV (Flash Video) a SWF (Shockwave flash)

Flash Video (FLV, F4V) je formát multimediálního kontejneru používaný k přehrávání videa přes Internet. Obsah flash videa může být vložen do souborů SWF. Existují dva různé formáty souborů videa známé jako Flash Video - FLV a F4V. Audio a video data v souborech FLV jsou kódovány stejným způsobem, jakým jsou do souborů SWF. Oba formáty jsou podporovány aplikací Adobe Flash Player a dnes i nejnovějším Adobe Systems. FLV byl původně vyvinut společností Macromedia. Formát FLV je podporován většinou operačních systémů a prohlížečů prostřednictvím volně dostupných Plutonů [11].

1.4.3 QuickTime

Jedná se o multimediální prostředí vyvinuté společností Apple Inc., které umožňuje práci s mnoha formáty digitálního obrazu, klipů, animace, textu i hudby.

Formát souboru QuickTime (přípona .mov) je multimediální kontejner obsahující jednu nebo více datových stop, přičemž každá stopa ukládá jeden typ dat (zvuk, video, efekty nebo text) a obsahuje digitálně zakódovaný mediální tok (zakódován pomocí speciálního kodeku) či odkazy na jinou datovou stopu (umístěnou v jiném souboru). Veškeré datové stopy jsou hieraticky uspořádány v tzv. atomech, kde jeden atom může být rodičem ostatních atomů, obsahovat data nebo je editovat [12].

1.4.4 GIF (The Graphics Interchange Format)

GIF byl vytvořen společností CompuServe. Dodnes patří mezi nejpoužívanější formáty pro zobrazování webové grafiky. Formát GIF dokáže dosáhnout dobrého kompresního poměru, zejména když jde o motivy s malým počtem barevných odstínů (tlačítka, ikony, atd.) – bezeztrátová komprese. Formát podporuje osmibitovou grafiku (maximální počet barev je 256).

Soubot GIF může obsahovat i více obrázků, které jsou pomocí vhodného softwaru zobrazovány jeden po druhém. Společně s daty obrázků jsou v souboru uloženy i informace, které říkají, jak rychle se mají obrázky střídát (rychlost animace) [13].

2 UMĚLÉ NEURONOVÉ SÍTĚ

Umělá neuronová síť (UNS) je struktura, která je určena pro rozšířené paralelní zpracování dat. Síť se skládá z umělých neuronů, jejichž základem jsou biologické neurony. Neuron má libovolný počet vstupů. Všechny neurony sítě jsou propojeny a předávají si signály, které dále transformují pomocí přenosových funkcí.

Funkcí UNS je transformace f vstupního signálu u na výstupní signál $y = f(u)$. Formálně je umělá neuronová síť definována jako orientovaný graf s dynamicky ohodnocenými hranami a vrcholy, nebo-li uspořádanou pěticí $[V, E, \varepsilon, w, y]$, kde:

V je množina neuronů (vrcholů),

E množina hran (synapsí),

ε zobrazení incidence hran s vrcholy ($\varepsilon : E \rightarrow V \times V$),

w dynamické ohodnocení hran ($w : \varepsilon(E) \times T \rightarrow R$),

y dynamické ohodnocení vrcholů ($y : V \times t \rightarrow R$),

T, t časové osy [14].

Nejrozšířenějším modelem neuronu je tzv. formální neuron (Obr. 7), který zpracovává údaje podle vztahu:

$$y = f\left(\sum_{i=1}^N w_i x_i + \Theta\right) \quad (1)$$

kde:

y je výstup (aktivita neuronu),

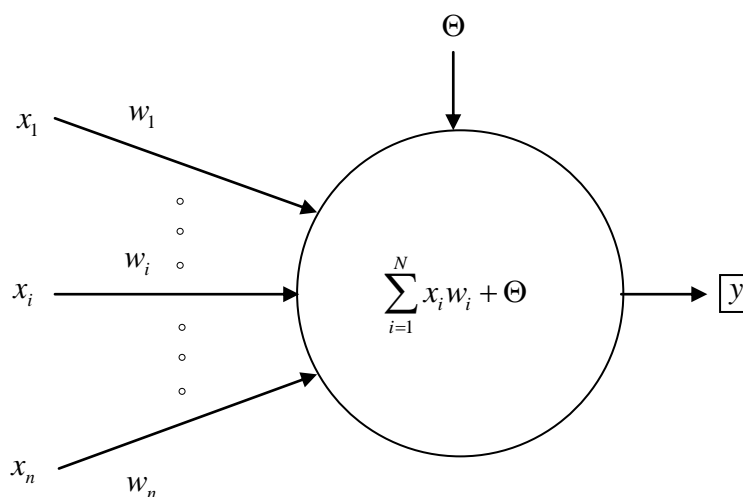
x_i i -tý vstup neuronu,

w_i hodnota i -té synaptické váhy,

f nelineární přenosová funkce neuronu,

Θ prahová hodnota.

Pro každý vstup x_i představuje w_i příslušnou synoptickou váhu, nebo-li lokální paměť a spojením všech neuronů dostáváme celkovou paměť sítě. Pro získání požadované hodnoty na výstupu je potřeba najít odpovídající hodnotu vektoru synoptických vah. Práhová hodnota Θ určuje, kdy bude neuron aktivní. Neuron bude aktivní tehdy, pokud bude vážená suma všech vstupů vyšší než práhová hodnota [15].



Obr. 7 – Model McCulloch-Pittsova neuronu

2.1 Dělení sítí

Všechny stavební jednotky (neurony) jsou mezi sebou různě propojeny, mohou obsahovat různé přenosové funkce a učící algoritmus. To vše určuje typ sítě.

2.1.1 Podle počtu vrstev

Jedná se o dělení vrstev podle toho, z kolika vrstev se UNS skládá. Existují sítě jednovrstvé, dvouvrstvé, třívrstvé a vícevrstvé. Speciální sítě jako např. Hopfieldova, Kohonenova nebo ART síť, které mají vlastní speciální učící algoritmus a topologii, spadají pod sítě s jednou nebo dvěma vrstvami. Třívrstvé a vícevrstvé sítě naopak používají klasickou vícevrstvou síť s algoritmem Backpropagation. Pro topologii UNS platí pravidlo, které říká, že každý neuron bývá spojen s každým neuronem ve vyšší vrstvě (Obr. 10). Zvláštním případem je Hopfieltova síť (Obr. 13), ve které je spojen právě každý neuron se všemi ostatními. Každé spojení neuronu má své váhy, které vyjadřují, jaký význam má spoj pro daný neuron.

2.1.2 Podle algoritmu učení

Učení probíhá tak, že se nastavují jednotlivé synaptické váhy a prahy. Na počátku se nastaví startovací hodnoty (mohou být náhodně zvolené). Poté se na vstup přivede tzv. trénovací vstup a neuronová síť poskytne výstup a odezvu. Učení UNS se dělí na učení s učitelem a bez učitele.

Učení s učitelem – Jedná se o snahu sítě přizpůsobit svou odezvu na vstupní informace tak, aby se výstup z této sítě co nejvíce podobal požadovanému originálu. Poté se změní nastavení synaptických vah tak, aby se rozdíl mezi aktuální a požadovanou výstupní hodnotou snížil. Tento postup se provádí pomocí učicího algoritmu. Učení probíhá dokud nebude splněna určitá podmínka.

Učení bez učitele – Síť vychází z informací, které jsou součástí vstupních vektorů, nenachází se zde žádné rozhodovací kritérium.

2.1.3 Podle stylu učení

Principem tohoto učení je přístup k zajištění vah sítě. Při zjišťování výpočtem se jedná o deterministické učení. Pokud je použit generátor náhodných čísel pro získání vah, pak se jedná o stochastický styl učení, který se obvykle používá jen při „startu“ sítě [15,16].

2.2 Přenosové funkce

Přenosová funkce udává, jaká bude odezva výstupu na vstupní podnět. Funkce mohou být lineární, spojitě nebo diskrétní. Pro všechny přenosové funkce platí, že jejich hodnota je v intervalu od -1 až 1.

Výběr funkce závisí na typu řešené úlohy, na náročnosti výpočtů a času vyhrazeného trénování těchto sítí. Mezi nejpoužívanější funkce patří:

- Perceptron
- Binární
- Logistická
- Hyperbolický tangens

2.2.1 Funkce perceptron

Funkce perceptron je lineární funkce, která řeší pouze problémy lineárně separabilní. Její zápis je

$$f(a) = a \quad \forall a > 0 \quad \text{jinak} \quad a = 0 \quad (2)$$

2.2.2 Funkce binární

Jedná se o binární (skokovou) funkci, která nabývá pouze hodnot 0 nebo 1. její zápis je

$$f(a) = 1 \quad \forall a > 0 \quad \text{jinak} \quad a = 0 \quad (3)$$

2.2.3 Funkce Logistická

Funkce logistická (nazývaná také logistická sigmoida) patří mezi nejpoužívanější přenosové funkce. Funkce vznikla aproximací přenosové funkce biologického neuronu. Její zápis je

$$f(a) = \frac{1}{1 + e^{-a}} \quad (4)$$

2.2.4 Funkce hyperbolický tangens

Obdobou logistické funkce je funkce hyperbolický tangens, s tím rozdílem, že může nabývat hodnot -1 až 1 [16,17]. Její zápis je

$$\text{Tanh}(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \quad (5)$$

2.3 Princip neuronové sítě

Neuronové sítě se mohou nacházet ve dvou fázích. V aktivací (vybavovací) fázi vykonává svou činnost, v adaptační (učicí) se síť učí. Při opakovaném střídání těchto fází se jedná o vlastní učení. Důležitou součástí adaptační fáze je trénování množina, která se skládá z vektorů obsahujících informace o daném problému pro učení. Při učení s učitelem obsahuje trénování množina vektor vstupní a výstupní, pokud jde o učení

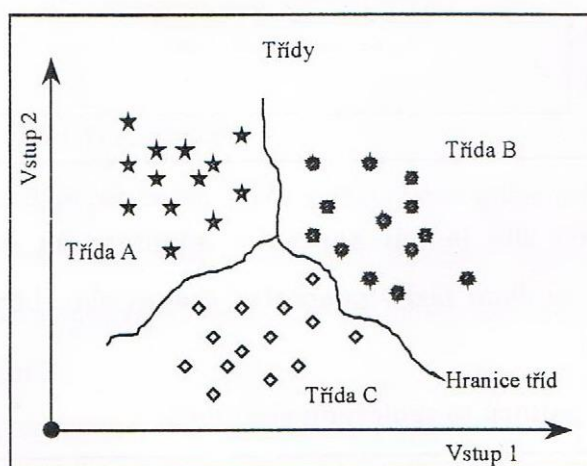
bez učitele, obsahuje pouze vstupní vektory. Fáze aktivační se používá jen tehdy, pokud je síť naučena.

Aktivační fáze je proces, ve kterém se vstupní vektor informací přepočítá přes všechny spoje s jejich váhami až na výstup. Na výstupu vznikne odezva sítě na vstupní vektor. Během učení dojde k porovnání tohoto vektoru s požadovaným vektorem a do paměti se uloží rozdíl mezi oběma vektory (lokální chyba).

Adaptační fáze je proces, kde se přepočítávají váhy spojů směrem od výstupu ke vstupu. Účelem fáze je minimalizovat lokální chybu, nebo-li snaha docílit co největší podobnosti vstupní odezvy s výstupním (originálním) vektorem.

Dalším krokem je opět aktivační fáze. Dojde k získání další lokální chyby, která se přičte k předešlé. Tímto postupem se prochází celá trénovací množina. Projití celé této množiny znamená dokončení jedné epochy, kde součet všech lokálních chyb v jedné epoše se nazývá globální chyba. Celý proces je dokončen tehdy, až globální odchylka bude menší než námi požadovaná chyba.

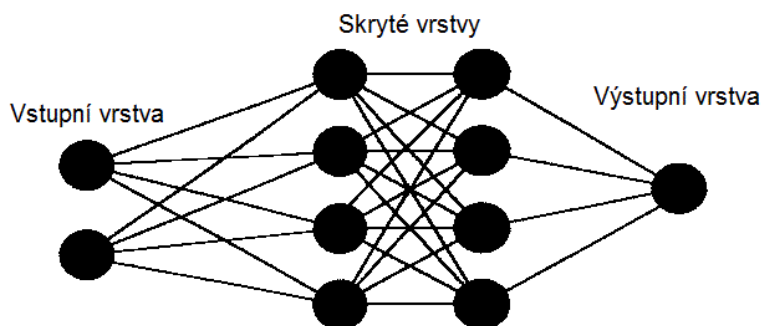
Svým způsobem se jedná o neustálé předávání informací ze vstupu na výstup a naopak, přičemž se na tzv. vstupním prostoru vytváří shluky bodů, které představují jednotlivé třídy (Obr. 8). Aby se síť naučila správným odezvám na dané podněty, závisí na velkém množství okolností a to na množství vektorů, velikostí vektorů, přípravě trénovací množiny a jiných [16].



Obr. 8 – Nelineární hranice mezi třídami [16]

2.4 Schéma neuronové sítě

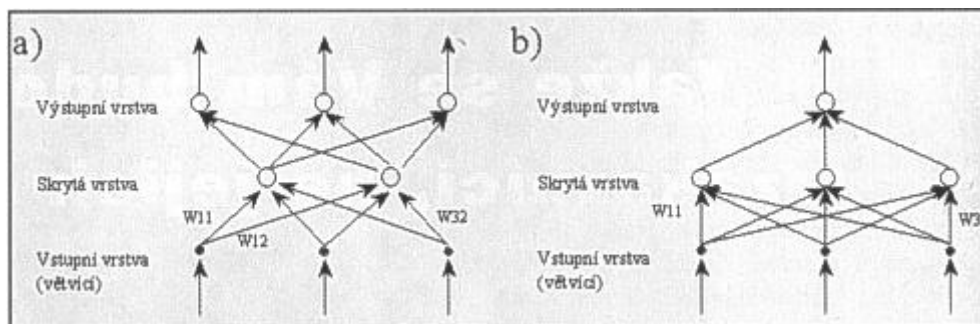
Neuronová síť se skládá z několika vrstev, kde každá vrstva se skládá z určitého počtu neuronů (Obr. 9).



Obr. 9 – Vícevrstvá síť

Sítě mohou být jednovrstvé (Hopfieldova a Kohonenova) nebo vícevrstvé. Vlastní funkce neuronové sítě je transformační funkce, která přiřazuje jinému vstupnímu obrazu obraz výstupní. Matematický důkaz této funkce nebyl dlouho k dispozici. Nakonec bylo zjištěno, že k aproximaci funkce neuronovou sítí stačí, aby obsahovala minimálně tři vrstvy s odpovídajícím počtem neuronů v každé vrstvě. Bohužel nebylo zjištěno, kolik je zapotřebí v těchto vrstvách neuronů, aby bylo řešení daného problému optimální.

U vícevrstvných sítí je první vrstva vždy větvičí, což znamená, že vstupní vektory pouze šíří hodnoty do další vrstvy. Pokud má síť více vstupů, mluvíme o vstupních (výstupních) vektorech informací (Obr. 10). Počet neuronů v jednotlivých vrstvách záleží na řešeném problému. Často je možné počet neuronů ve vrstvách odhadnout nebo vypočítat, ale ne vždy dojdeme ke správným výsledkům [16,18].

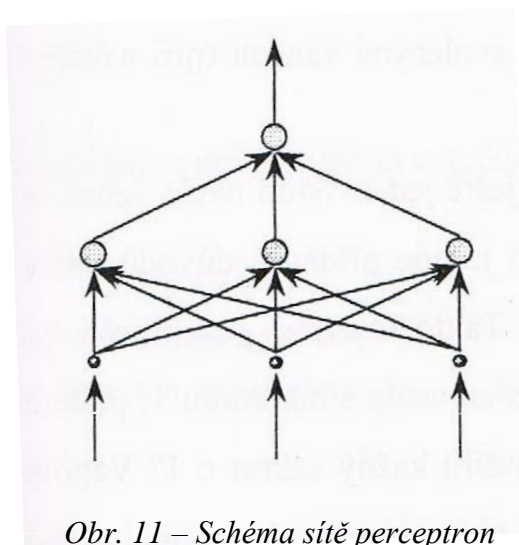


Obr. 10 – Různé typy neuronových sítí [16]

2.5 Typy neuronových sítí

2.5.1 Perceptron

Nejjednodušším modelem dopředné neuronové sítě je perceptron, který se skládá pouze z jednoho neuronu. Jeho užití je velmi omezené, používá se jen na množiny, které jsou lineárně separovatelné. Pro širší možnosti použití byl tento model rozšířen na vícevrstevný perceptron.



Obr. 11 – Schéma sítě perceptron

Typickým perceptronem (Obr. 11) je síť s jedním pracovním (výstupním) neuronem spojeným s n vstupy (x_1, x_2, \dots, x_n) , kde každý spoj má váhovou hodnotu (w_1, w_2, \dots, w_n) . Váhové hodnoty se přizpůsobují tak, aby rozdíl mezi skutečným a požadovaným výstupem byl co nejmenší. Kromě klasických vstupů mají neurony v učící vrstvě navíc další vstup nastavený na hodnotu 1, který slouží pro nastavování prahu neuronu. Takovéto rozšíření jednotkového vstupu je chápáno jako rozšíření každého vstupního vektoru o jeden vstup s hodnotou 1 – rozšířený vektor [16].

Perceptron je jednovrstevná neuronová síť, s dopředným šířením a učením s učitelem. Celkový podnět, označován jako potenciál neuronu, udává součet:

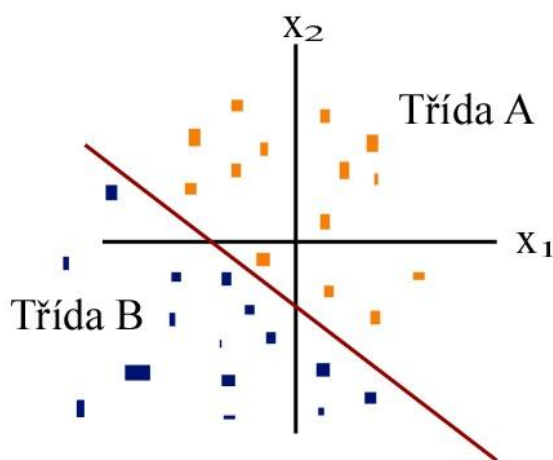
$$\xi = \sum_{i=1}^N w_i x_i + \Theta = \sum_{i=0}^N w_i x_i \quad (6)$$

x_i je i -tý vstup neuronu,

w_i hodnota i -té synaptické váhy,

Θ prahová hodnota.

Na potenciál reaguje perceptron výstupní odezvou $y = f(\xi)$, kde f je přenosová funkce. Jedná-li se o přenosovou funkci skokovou, pak funguje perceptron jako binární klasifikátor. Tato funkce je realizována neuronem (označována také jako TLU – Threshold Logica Unit – prahová logická jednotka) od McCullocha-Pittse s N vstupy, může nabývat hodnot 1 (třída A) nebo 0 (třída B).



Obr. 12 – Klasifikátor pro lineárně separovatelné obrazy [20]

Jednou z metod pro výpočet nových vah sítě je metoda fixního přírůstku. Její zápis je:

$$w(k+1) = \begin{cases} w(k) + c \cdot x(k) & \text{jestli } w(k) \cdot x(k) \leq 0 \\ w(k) & \text{jestli } w(k) \cdot x(k) > 0 \end{cases} \quad (7)$$

kde:

$w(k+1)$ jsou nové váhy,

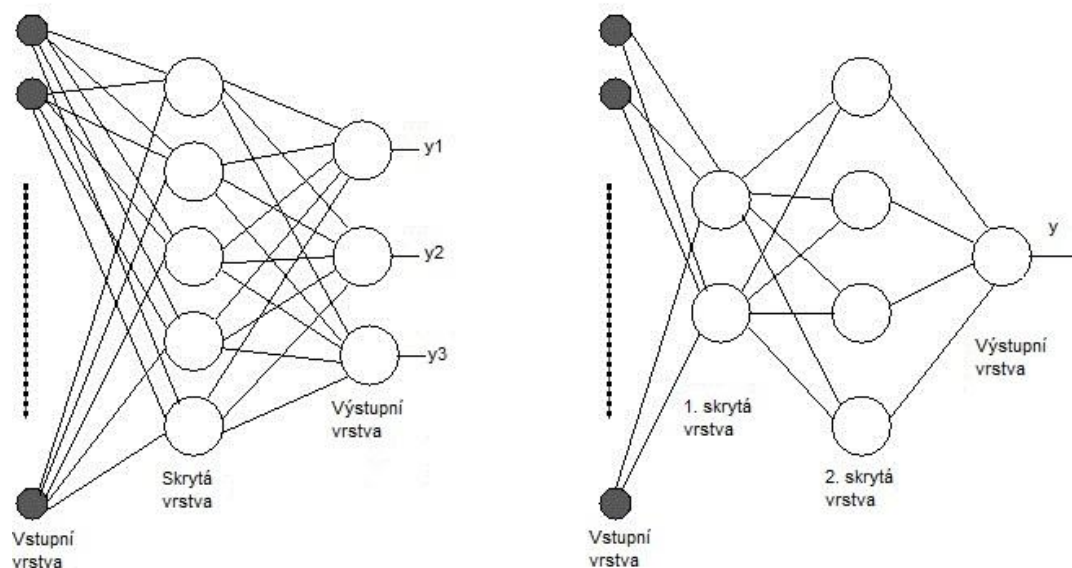
$w(k)$ aktuální váhy,

c korekční přírůstek (učicí faktor),

$x(k)$ vstupní vektor [19].

2.5.2 Vícevrstvá síť s algoritmem Backpropagation

Jedná se o síť s dopředným šířením s učitelem (Obr. 13). Síť se skládá ze vstupní, skryté a výstupní vrstvy. Aktivační funkcí této sítě je nejčastěji logická sigmoida. Použití vícevrstevných sítí je velmi široké a spolu s učícím algoritmem Backpropagation (algoritmus zpětného šíření chyby) je zastoupeno ve více jak 80% aplikací neuronových sítí v praxi.

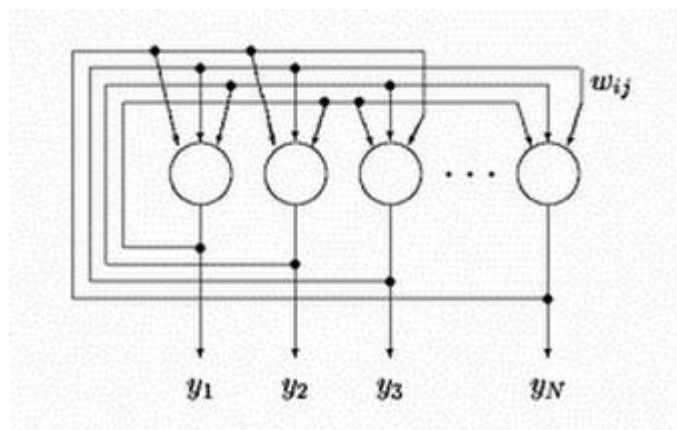


Obr. 13 – Typy vícevrstevných neuronových sítí [22]

Backpropagation je iterační proces, jehož cílem je dosáhnout stavu úplného naučení. Výběr vzoru z trénovací množiny může být náhodný i sekvenční. Časovému intervalu, během kterého se síti předloží každý vzor z trénovací množiny alespoň jednou, se nazývá epocha. Těchto epoch je potřeba k naučení sítě stovky až tisíce. Samotné učení lze ukončit dosažením určité minimální hodnoty chyby nebo překročením maximálního počtu epoch [21,16].

2.5.3 Hopfieldova síť

Jedná se o jednovrstevnou rekurentní síť s pevnými váhami (Obr. 14) vynalezenou Johnem Hopfieldem. Přenosovou funkcí je používána funkce skoková, kde vstupní vektory nabývají hodnot 1 a 0 (unipolární binární) nebo 1 a -1 (bipolární binární).

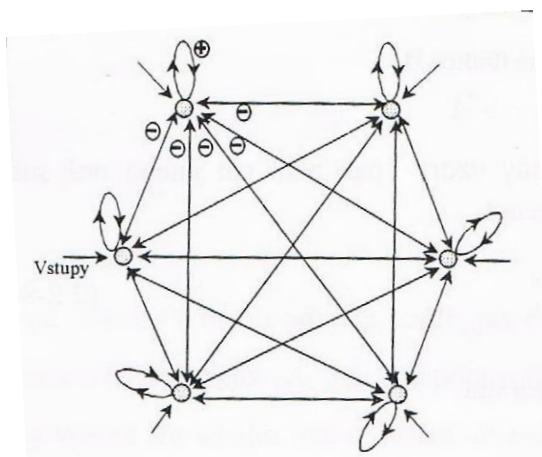


Obr. 14 – Obecné schéma Hopfieldovy sítě [24]

Nejčastěji se Hopfieldova síť používá ke klasifikaci a optimalizaci. Síť obsahuje navzájem propojené neurony, kde váhy mezi dvěma neurony jsou totožné [23].

2.5.4 Síť CLN (Competitive Learning Network)

Síť CLN (nazývána také síť „soutěživých“ neuronů) vychází z Hopfieldovy sítě (Obr. 14). Principem je soutěž neuronů, kde každý z nich se může stát vítězem, který bere vše. Prakticky to znamená, že po stabilizaci sítě může být aktivní pouze jeden neuron. Síť se používá pro rozdělení tříd mezi vstupními daty při učení bez učitele.



Obr. 15 – Schéma sítě CLN [16]

Topologie sítě CLN (Obr. 15) je totožná s Hopfieldovou sítí s tím rozdílem, že všechny neurony jsou navíc propojeny samy se sebou. Principem je snaha neuronů potlačit ostatní a přitom sebe co nejvíce aktivovat.

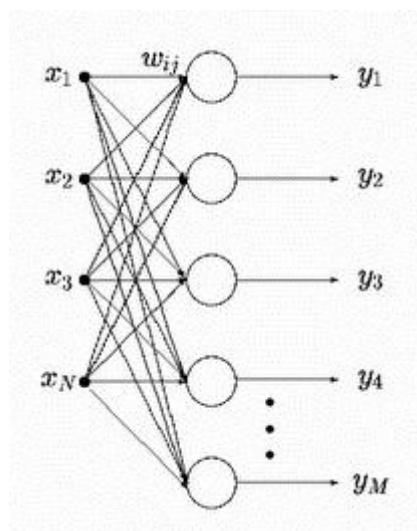
Vlastní učení se skládá z fáze soutěžení a odměny. Ve fázi soutěžení dochází ke zjištění, který z neuronů byl aktivován pro daný vstupní vektor a ve fázi odměny jsou

aktivovanému neuronu změněny jeho váhy tak, aby se daný vstupní vektor co nejlépe posunul ke středu třídy, do které patří.

CLN sítě se využívají nejčastěji v oblasti práce s daty (komprese, klasifikace, atd.). Princip lze použít i pro vícevrstvé sítě tak, že v první skryté vrstvě zůstane po soutěžení několik nenulových neuronů. Výstup z této vrstvy se stane vstupem další vrstvy, kde se opět provede soutěžení, dokud nedojde až k poslední výstupní vrstvě, ve které je aktivován pouze jeden neuron [16].

2.5.5 Sít' LAM (Linear Associative Memory)

Sít' LAM je jednovrstvá síť s dopředným šířením a pevnými vahami. Vstupní vektory mohou být binární, popř. reálné. U neuronových sítí typu LAM probíhá vybavení příslušné informace na základě její částečné znalosti (asociace). Topologie sítě je zobrazena na obrázku (Obr. 16).

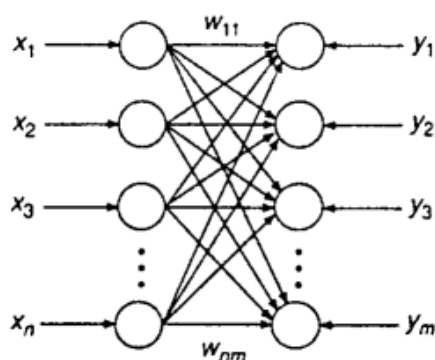


Obr. 16 – Schéma sítě LAM [26]

Sít' LAM se používá například jako hetero či auto asociativní paměť nebo při rekonstrukci poškozených obrazů [25].

2.5.6 Sít' BAM (Bidirectional Associative Memory)

Sít' BAM (obousměrná asociativní paměť) je dvouvrstvá rekurentní neuronová síť s pevnými vahami a schopností heteroasociace, což je schopnost přiřazovat příbuzné obrazy k obrazům vstupním. Schéma sítě je zobrazeno na obrázku (Obr. 17) [16].

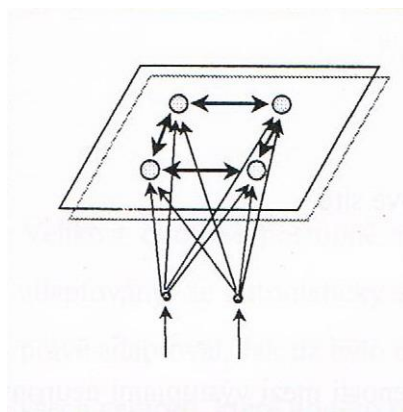


Obr. 17 – Schéma sítě BAM [28]

V aktivním režimu sítě BAM si neurony obou vrstev předávají mezi sebou signál, dokud všechny neurony nedosáhnou rovnovážného stavu, nebo-li není nalezen příslušný vhodný vektor. Neurony v této síti jsou stejné jako v Hopfieldově síti (Obr. 14) [23,27].

2.5.7 Kohonenova síť

Kohonenova neuronová síť patří do skupiny samoorganizujících neuronových sítí s učením bez učitele a pracuje na principu vyhledávání závislostí v datech v trénování množině (Obr. 18).



Obr. 18 – Kohonenova síť [16]

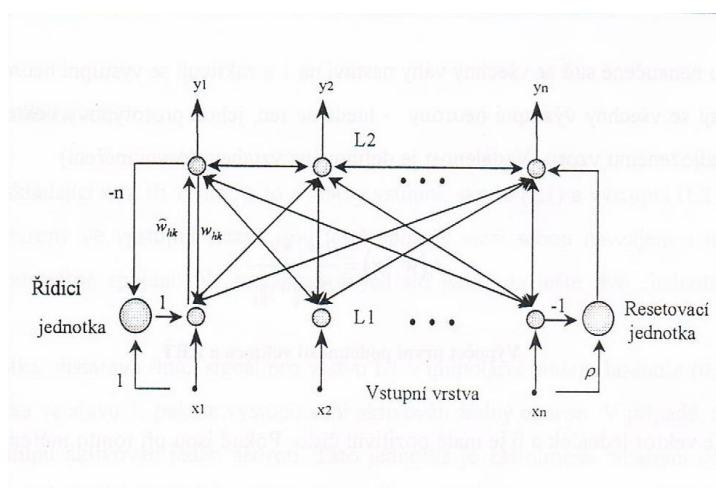
Základní návrh sítě předpokládá v roli neuronů logické prahové elementy uspořádané v jednorozměrné (popř. dvourozměrné) struktuře s propojením sousedních neuronů. V procesu učení formují výstupy sítě pouze neurony, které mají nejvyšší hodnotu aktivační funkce. Během učení se nastavují váhy prvků dosahujících nejvyšších hodnot aktivační funkce a prvky v jejich okolí. Jelikož po naučení sítě odpovídají vzájemně

sousední neurony podobnými odezvami na vstupní signál, je Kohonenova síť vysvětlována jako jedna z forem asociativních sítí.

2.5.8 Síť ART (Adaptive Resonant neTwork)

Síť ART sestrojili S. Grossberg a G.A. Carpenter z důvodu prověření některých závěrů teorie adaptivní resonance. Teorie je založena na základě předpokladu, že váhy neuronů se nastavují z podmínek resonancí vstupních signálů, které jsou považovány za časové řady. Přenosová funkce sítě nebyla blíže formulována.

Principem sítě ART je porovnávání vstupního vektoru s tzv. prototypovými vektory, které jsou uchovány v síti, dokud není nalezen prototypový vektor, který se nachází dostatečně blízko vstupnímu vektoru. Prototypovými vektory jsou tedy myšleny váhy jednotlivých spojů mezi skrytou a výstupní vrstvou (Obr. 19).



Obr. 19 – Schéma sítě ART [16]

Síť se skládá z vstupní vrstvy, skryté L1 a výstupní L2 (Obr. 19). Neurony ve výstupní vrstvě jsou navíc propojeny mezi sebou a mezi vrstvami L1 a L2 existuje obousměrné spojení. Další součástí sítě je řídicí a resetovací jednotka.

Řídicí jednotka, která může nabývat hodnot 0 a 1, obstarává signál pro vrstvu L1 tak, že ve stavu 1 není na výstup aktivován žádný neuron. V opačném případě je aktivován jeden neuron [16,23].

2.6 Optimalizace topologie neuronové sítě

Snaha najít optimální počet neuronů a vrstev vzhledem k řešenému problému se nazývá optimalizace topologie sítě, přičemž pojem **topologie sítě** znamená způsob, jakým jsou mezi sebou propojeny neurony, počet vstupů a výstupů sítě, typ přenosové funkce ve vrstvách, atd.

Optimalizace topologie sítě je problém, který nelze univerzálně řešit. Největší vliv na strukturu mají váhy a jejich množství, které ovlivňují celkový chod sítě. Tento problém se týká nejčastěji vícevrstevných sítí, které narozdíl od např. Hopfieldovy nebo Kohonenovy sítě nemají strukturu již danou definováním problému (Hopfieldova síť má předem daný počet pixelů = neuronů).

2.6.1 Počet vrstev

Neuronová síť se skládá ze vstupní, výstupní a skryté vrstvy. Sítě mají nejčastěji 3 nebo 4 vrstvy. Síť se třemi vrstvami má méně vah, což znamená kratší dobu učení. Vývoj globální chyby je rychlejší a stabilnější. Nevýhodou je, že se může setkat s funkcí, jejíž aproximace je horší než u čtyřvrstvé sítě. Síť se čtyřmi vrstvami je tedy výhodnější při aproximaci funkcí. Tato síť má však delší dobu učení a méně stabilnější vývoj globální chyby, jelikož má větší členitost chybové plochy.

2.6.2 Počet neuronů ve vrstvách

Počet neuronů ve vstupní a výstupní vrstvě je dán problémem a tím, jak jej chceme řešit. Pro počet neuronů ve skrytých vrstvách platí obecně pravidlo čím méně, tím lépe. Zjistit počet neuronů ve skrytých vrstvách lze docílit **pevným nastavením** nebo **adaptivním nastavením**.

Pevné nastavení lze nastavit počet neuronů pomocí univerzálních vzorců, které mají za následek vytváření sítí pyramidového tvaru. Ovšem tyto vzorce jsou obecné a také nepřesné. Pro dvouvrstvou síť platí vztah

$$N_{skryt.} = \sqrt{N_{vstup} \cdot N_{výstup}} \quad (8)$$

Počet neuronů v první a druhé skryté vrstvě u třívrstevných sítí je dán vztahy

$$N_{1.skryta} = N_{výstup} \cdot \left(\sqrt[3]{\frac{N_{vstup}}{N_{výstup}}} \right)^2 \quad N_{2.skryta} = N_{výstup} \cdot \left(\sqrt[3]{\frac{N_{vstup}}{N_{výstup}}} \right) \quad (9)$$

Adaptivní nastavení je algoritmus, pomocí kterého se přidávají nebo odebírají neurony ve vrstvě. Principem algoritmu je cyklické kontrolování velikosti trénování a testovací chyby. Na jejich základě dochází k přidání neuronu (následuje opět trénování) nebo ukončení činnosti algoritmu [16].

II. PRAKTICKÁ ČÁST

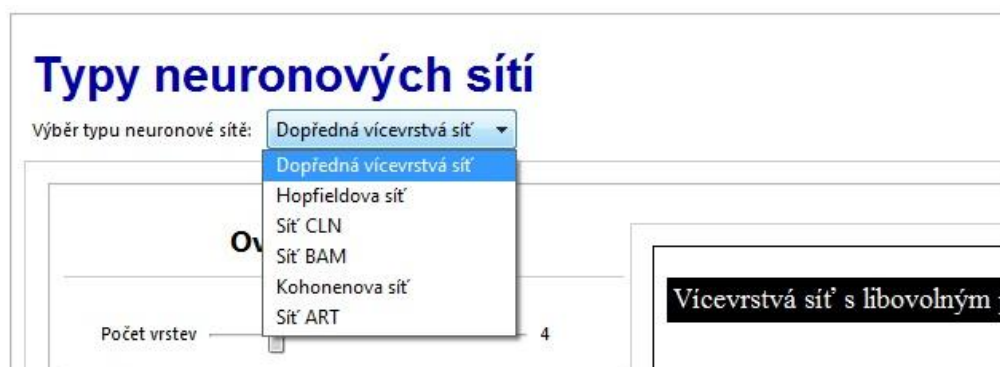
3 ANIMACE UMĚLÝCH NEURONOVÝCH SÍTÍ

V této části jsou popsány možnosti a ovládání jednotlivých modulů, které zobrazují animace a typy umělých neuronových sítí. Základní použité funkce pro vytvoření animací byly popsány v teoretické části.

Program byl navržen tak, aby měl uživatel možnost rychlého přepínání mezi jednotlivými animacemi nebo typy neuronových sítí pomocí výběrového (Obr. 20) nebo rozbalovacího menu (Obr. 21) a nemusel tak hledat a spouštět jednotlivé programy odděleně.



Obr. 20 – Možnost volby typu neuronové sítě pomocí výběrového menu

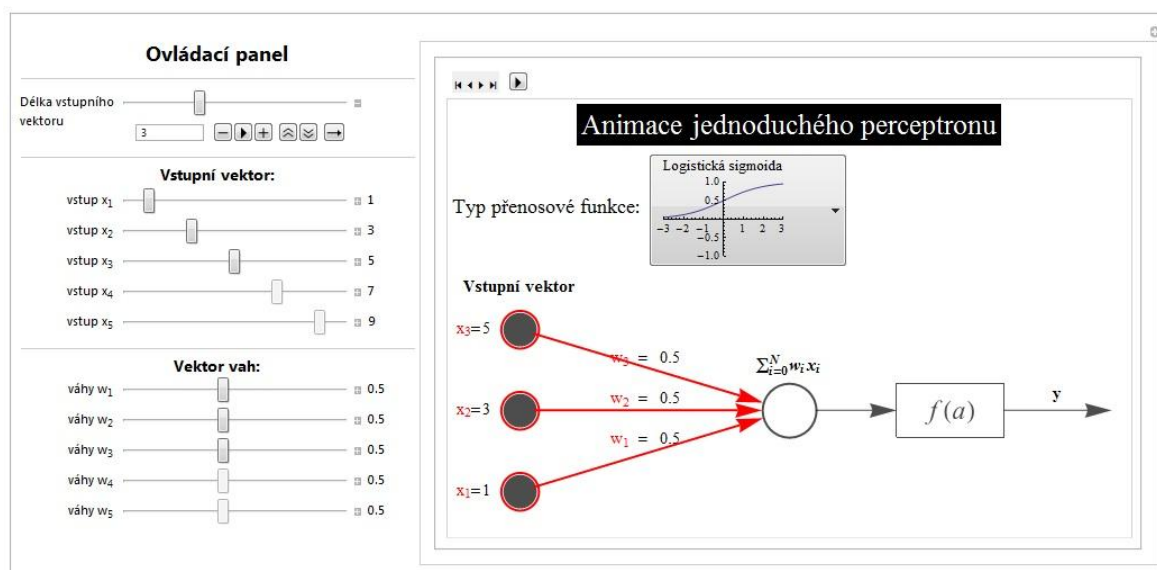


Obr. 21 – Možnost volby typu neuronové sítě pomocí rozbalovacího menu

3.1 Animace jednoduchého perceptronu

Tento modul slouží k animaci jednoduchého perceptronu včetně výpočtů potřebných pro získání hodnoty výstupu. Modul se skládá z ovládacího panelu, ve kterém má uživatel možnost nastavit parametr, který dokáže ovlivňovat animace i výpočty (délka vstupního vektoru) nebo parametry, které zasahují pouze do výpočtů (hodnoty vstupního vektoru nebo vektoru vah). Druhou částí modulu je grafický objekt, který se mění v závislosti na uživatelem zvolených parametrech. Součástí objektu je skupina tlačítek, pomocí kterých

lze animaci ovládat ručně, nebo přehrát automaticky. Další možností je výběr typu přenosové funkce v podobě rozbalovacího menu. Při otevření tohoto menu má uživatel možnost vidět již vykreslené grafy všech možných přenosových funkcí. Poslední a nejdůležitější částí grafického objektu je struktura jednoduchého perceptronu. Spolu s veškerými popiskami jsou v průběhu animace dynamicky zobrazovány i parametry zvolené uživatelem (Obr. 22).

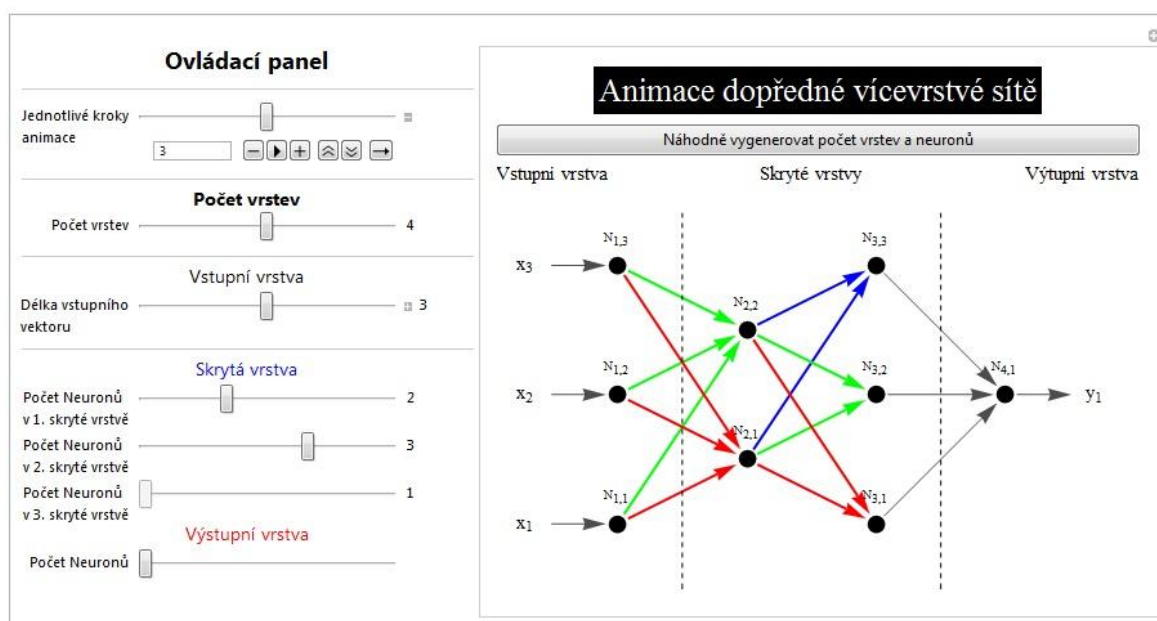


Obr. 22 – Modul Animace jednoduchého perceptronu

Celá animace byla vytvořena pomocí funkce `SlideView[{expr1, expr2,...}]` představující objekt, ve kterém jsou výrazy *expr* stanoveny tak, aby byly zobrazovány jeden po druhém. Výrazy *expr* jsou zde zastoupeny funkcemi `Graphics`, které tvoří jednotlivé kroky animace. V rozbalovacím menu, sloužící pro výběr přenosové funkce, bylo využito dynamické proměnné pro zobrazení průběhu funkce přímo do možností výběru tohoto menu. V průběhu animace dochází k výpočtům, které na sebe navazují. Ve třetím kroku animace se vypočítá suma pomocí vztahu (6), která je v dalším kroku obsažena v rovnici (2) pro výpočet přenosové funkce – perceptron, skoková funkce (3), logistická sigmoida (4), hyperbolický tangens (5) nebo funkce RBF. Program byl vytvořen tak, aby se při jakékoli změně vstupních parametrů uživatelem vrátila animace zpět na začátek.

3.2 Animace dopředné vícevrstvé neuronové sítě

V tomto modulu je zobrazena pouze animace dopředné vícevrstvé neuronové sítě. Uživatel má možnost nastavit pomocí skupiny posuvníků v ovládacím panelu jednotlivé parametry (kroky animace, počet vrstev v síti, délku vstupního vektoru nebo počet neuronů ve vrstvách). Součástí grafického objektu je navíc tlačítko, pomocí kterého je možné náhodně vygenerovat počet vrstev a neuronů v síti. Celý modul je znázorněn na obrázku (Obr. 23).



Obr. 23 – Modul Animace dopředné vícevrstvé sítě

Narozdíl od předešlé animace, byla tato animace vytvořena využitím klasického posuvníku. Ten má možnost pomocí příkazu Appearance→"Open" zobrazit skupinu tlačítek pro ovládání animace. Během animace se postupně zvýrazňují vektory, které vstupují do jednotlivých neuronů (každý neuron má odlišnou barvu skupiny vektorů, které do něj vstupují), kde vstupní a výstupní vektory jsou vždy označeny červenou barvou. Při změně kteréhokoli parametru dochází k přepočítávání souřadnic všech bodů v síti (souřadnice vektorů a popisek vycházejí ze souřadnic bodů) pomocí funkcí pro vytváření cyklů. Tlačítko pro náhodné generování parametrů slouží také pro nastavení animace na začátek.

3.3 Vizualizace topologie neuronových sítí

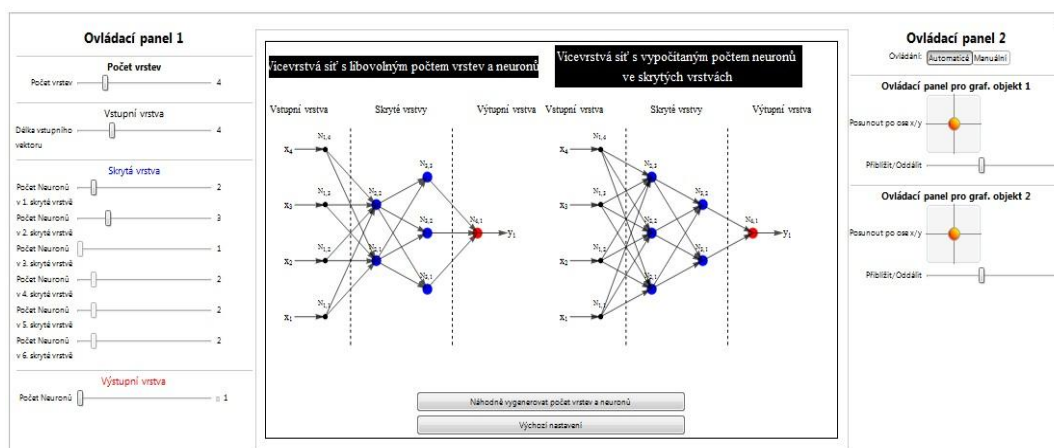
Uživatel má na výběr z šesti možností, kde každá představuje jiný typ sítě (každá z těchto sítí byla popsána v teoretické části).

- Vícevrstvá neuronová síť
- Hopfieldova síť
- Síť CLN
- Síť BAM
- Kohonenova síť
- Síť ART

3.3.1 Dopředná vícevrstvá síť

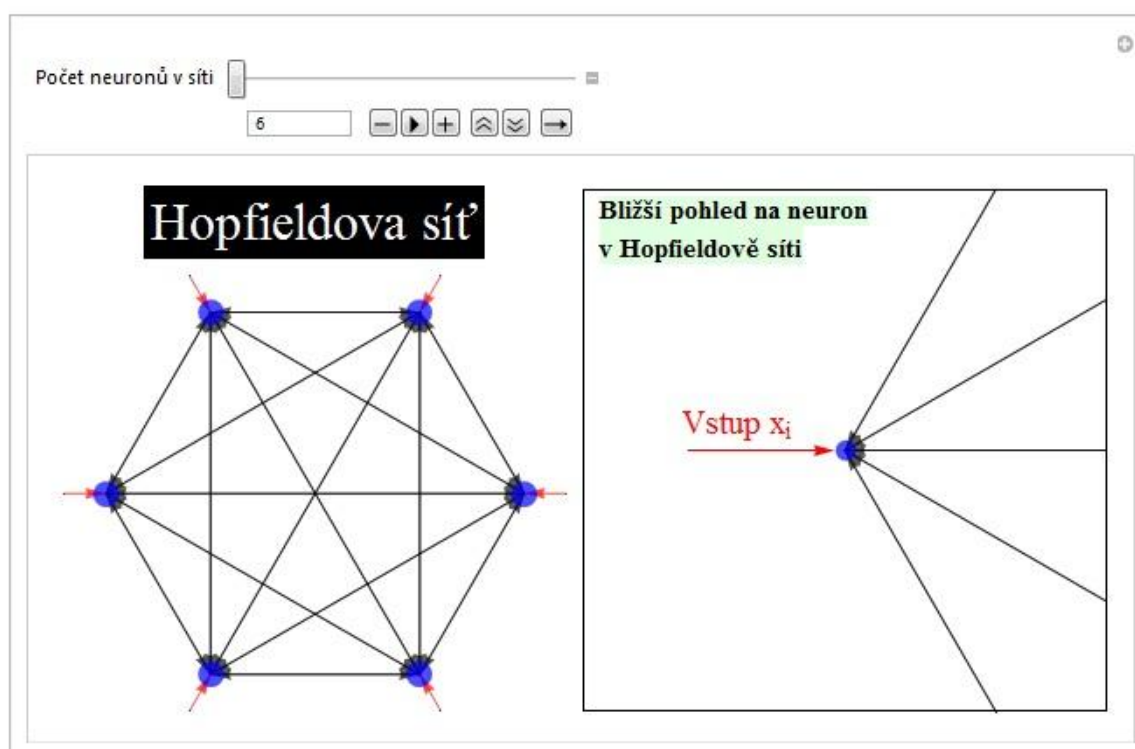
Výstupem tohoto modulu jsou dva grafické objekty, které na základě zvoleného nastavení zobrazují topologii vícevrstvé neuronové sítě. První grafický objekt slouží k vizualizaci neuronové sítě s libovolným počtem vrstev a neuronů v nich, jež lze nastavit pomocí skupiny posuvníků (s přibývajícím zvoleným počtem vrstev se aktivují posuvníky pro nastavení počtu neuronů ve skrytých vrstvách) nebo náhodně vygenerovat tlačítkem v levé části modulu, zatímco druhý zobrazuje síť s vypočítaným množstvím neuronů ve skrytých vrstvách na základě zvoleného počtu vrstev a neuronů ve vstupní a výstupní vrstvě pomocí vztahů (8) a (9). V obou případech představují černé body vstupní neurony, modré skryté a červené výstupní. Jednotlivé vrstvy jsou odděleny svislou přerušovanou čarou.

Při zvolení velkého množství vrstev a neuronů docházelo k nepřehlednému zobrazení topologie sítě. Z tohoto důvodu byl přidán ovládací panel (zobrazen v pravé části modulu), který slouží k přibližování, oddalování a posouvání pohledu na zobrazovanou síť. Součástí je také tlačítko, které vrací pohled do původního stavu (Obr. 24).



3.3.2 Hopfieldova síť

Následující jednoduchý modul slouží k vizualizaci topologie Hopfieldovy neuronové sítě (Obr. 25). Zobrazovány jsou opět dva grafické objekty. První grafický objekt poskytuje pohled na celou síť, kdežto druhý pouze na jeden neuron. V horní části modulu se nachází pouze jeden posuvník, kterým lze nastavovat počet neuronů v síti.

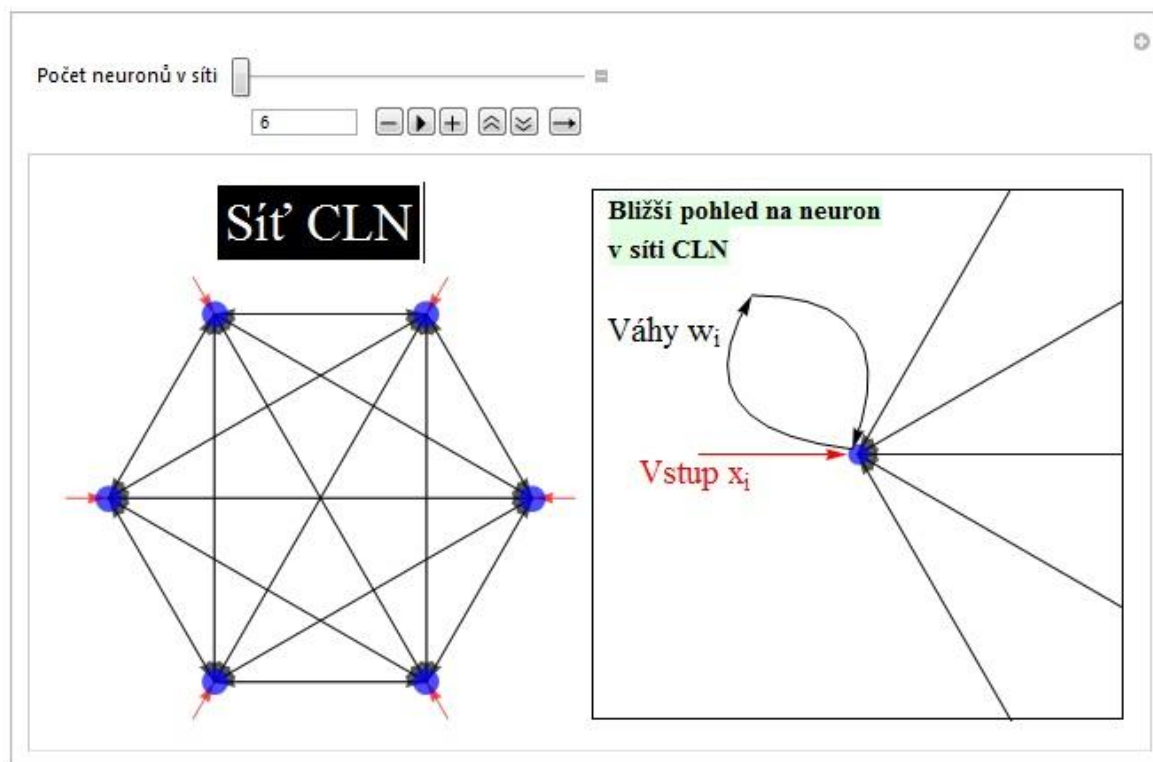


Obr. 25 – Modul Hopfieldova síť

V tomto případě se jedná o kruhové uspořádání neuronů. S využitím jednotkové kružnice, která vznikla přenesením souřadnic $[x, y]$ z grafu goniometrických funkcí na kružnici o poloměru 1, byly vypočítány v závislosti na zvoleném počtu neuronů v síti veškeré potřebné souřadnice bodů (neuronů). Na rozdíl od vícevrstvé neuronové sítě, ve které jsou propojeny pouze neurony v sousedních vrstvách, je v této síti propojen každý neuron s každým. Vstupní vektory byly vytvořeny díky pomocné kružnici, která slouží jako množina počátečních souřadnic vektorů.

3.3.3 Síť CLN

V tomto modulu je demonstrována síť CLN (Obr. 26), jejíž topologie je obdobná Hopfieldově síti. Stejně tak se skládá z dvou grafických objektů a jednoho posuvníku.

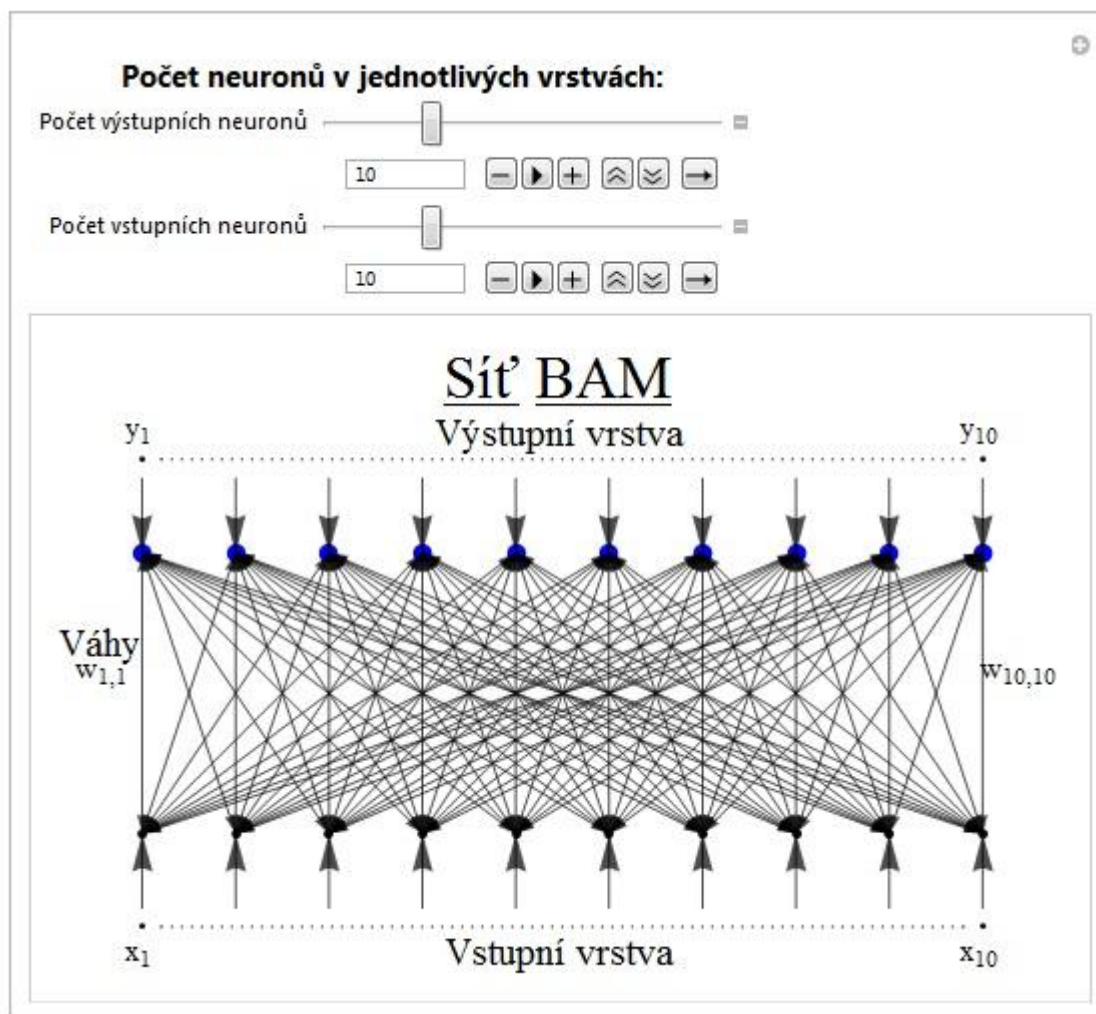


Obr. 26 – Modul Síť CLN

Výpočty souřadnic byly získány obdobným způsobem jako v předcházejícím modulu. Na rozdíl od Hopfieldovy sítě, tato síť obsahuje navíc další počet vektorů závislých na zvoleném množství neuronů v síti.

3.3.4 Síť BAM

Další modul slouží k vizualizaci dvouvrstvé sítě BAM. Grafický výstup modulu (Obr. 27) zobrazuje dvě navzájem propojené vrstvy – vstupní a výstupní. K nastavení počtu neuronů v obou vrstvách slouží dva posuvníky v horní části modulu. Stejně jako v ostatních případech vedlo velké množství neuronů ve vrstvách k nepřehlednosti struktury sítě, proto byl jejich počet omezen maximální hodnotou 30 (počet neuronů ve vrstvě).

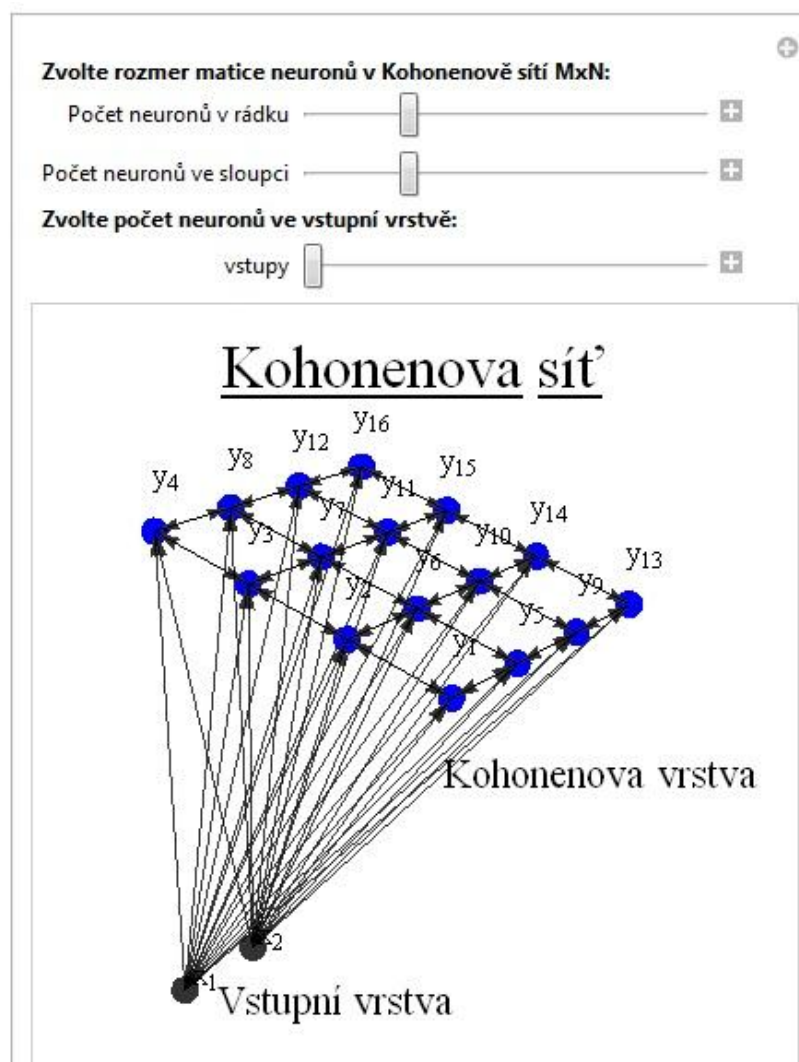


Obr. 27 – Modul Síť BAM

Síť byla vytvořena na základě generování dvou množin bodů, které představují neurony ve vstupní a výstupní vrstvě. Pomocí jednoduchých příkazů byly tyto množiny navzájem propojeny vektory (váhy neuronů). Žádné souřadnice prvků grafického objektu nebyly nastaveny na pevnou hodnotu, nebo-li jejich souřadnice byly určeny na základě vypočítaného posunutí od již vygenerovaných množin bodů.

3.3.5 Kohonenova síť

Tento modul byl zaměřen na prostorovou vizualizaci struktury Kohonenovy sítě (Obr. 28). Grafický objekt zobrazuje dvě vrstvy. Vstupní vrstvu reprezentují černé body (neurony) a výstupní Kohonenova vrstva. Součástí modulu jsou tři posuvníky, přičemž prvníma dvěma lze nastavit počet neuronů v Kohonenově (výstupní) vrstvě a třetím počet neuronů ve vstupní vrstvě. Uchopením a tažením myši lze celou síť otáčet v prostoru.

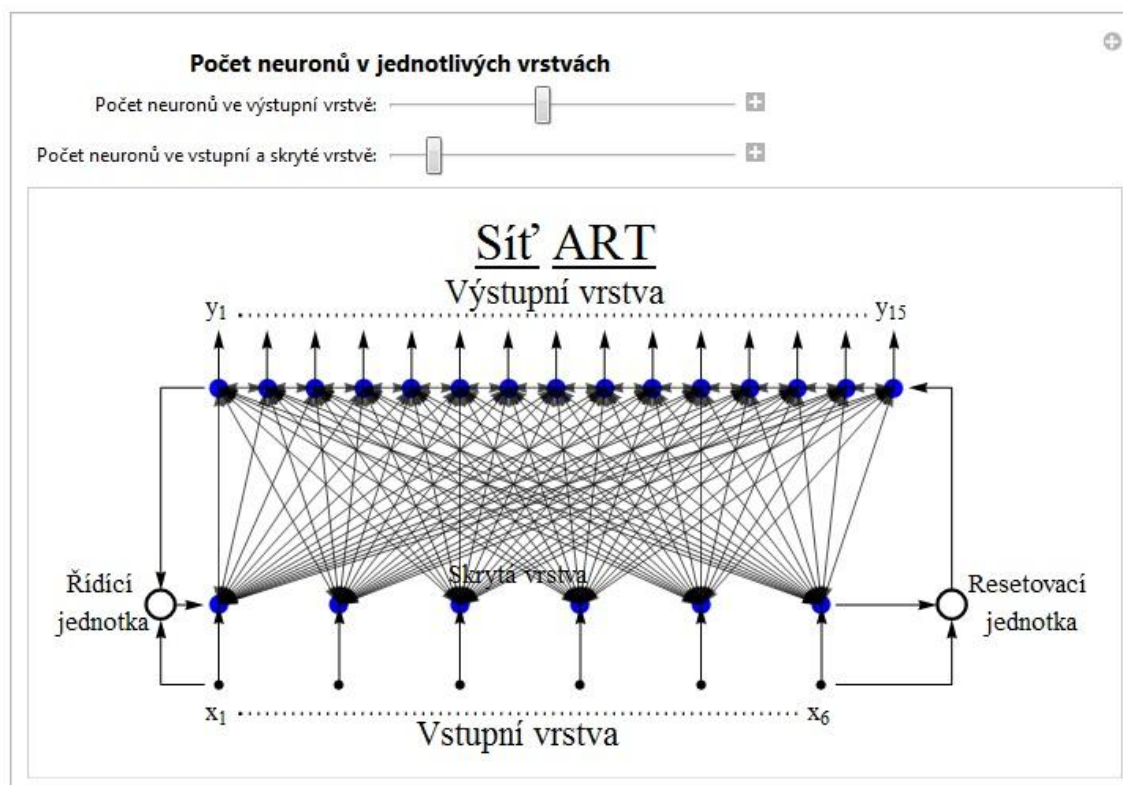


Obr. 28 – Modul Kohonenova síť

Princip vizualizace této sítě je podobný jako u ostatních sítí s tím rozdílem, že tato síť byla vymodelována v 3D prostoru (Obr. 28). Každý prvek (neuron) vstupní množiny je opět spojen s každým prvkem množiny výstupní, kde tyto spoje představují váhy. V Kohonenově vrstvě jsou propojeny pouze sousední neurony.

3.3.6 Sít' ART

Následuje modul, který zobrazuje neuronovou síť typu ART (Obr. 29). Topologii této sítě je možné měnit dvojicí posuvníků, které představují počet neuronů v jednotlivých vrstvách. Grafický objekt zobrazuje tři vrstvy (vstupní, výstupní a skrytou), propojení mezi nimi (mezi skrytou a výstupní vrstvou existují obousměrné spoje), výstupní vektory a řídicí a resetovací jednotku. Počet neuronů ve vstupní a skryté vrstvě je v síti ART stejný.



Obr. 29 – Modul Sít' ART

Sít' ART obsahuje navíc řídicí a resetovací jednotku. Zatímco souřadnice řídicí jednotky se při nastavování počtu neuronů v jakékoliv síti nemění, souřadnice resetovací jednotky se mění v závislosti, která ze dvou vrstev obsahuje více neuronů. Jestliže výstupní vrstva obsahuje více neuronů, dochází k vypočítání hodnoty posuvu (rozdíl x -ových souřadnic posledních bodů výstupní a vstupní, popř. skryté vrstvy), která je poté přičtena k x -ovým souřadnicím počátečních bodů vektorů směřujících do resetovací jednotky. V opačném případě jde o obdobný postup při výpočtu posuvu (rozdíl x -ových souřadnic posledních bodů vstupní a výstupní vrstvy).

4 EXPORT VYTVOŘENÝCH ANIMACÍ

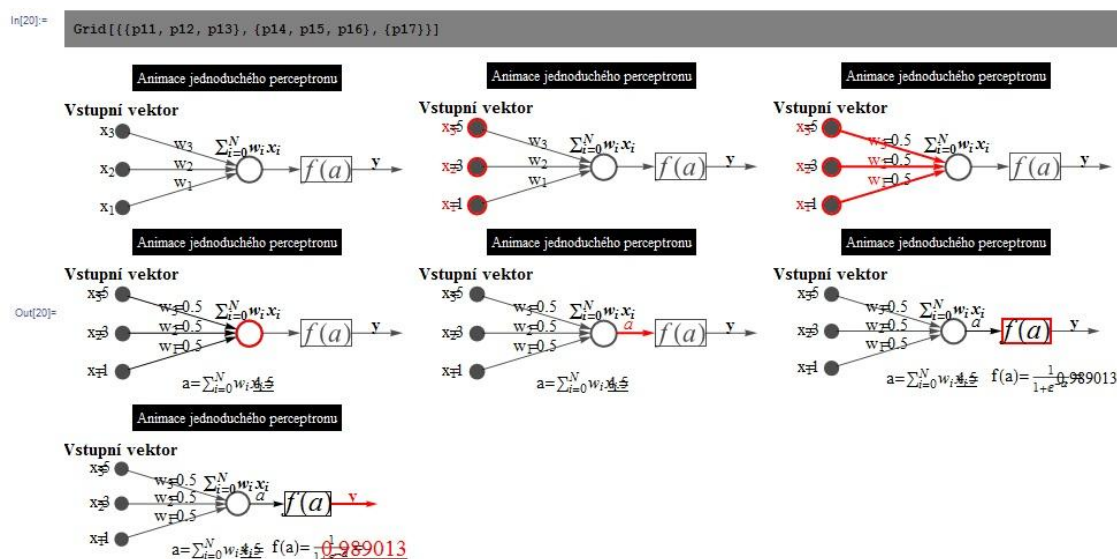
Cílem této části bakalářské práce je exportovat vytvořené animace do multimediálních popřípadě grafických formátů, které Mathematica podporuje. Pro export byly použity tyto formáty: AVI, SWF, FLV a GIF.

Jedním z parametrů funkce Export jsou elementy - skupina příkazů, pomocí které lze měnit vlastnosti exportovaného objektu (rozměr, pozadí, atd.) v závislosti na zvoleném výstupním formátu.

4.1 Export animace jednoduchého perceptronu

První ukázkou je export animace jednoduchého perceptronu do multimediálního formátu AVI a grafického formátu GIF. Jak již bylo zmíněno v teoretické části, funkce Export převede data (druhý parametr) do formátu odpovídající příponě v prvním parametru.

V tomto případě jsem použil již vytvořený program pro animaci jednoduchého perceptronu (Obr. 7), která se skládá ze sedmi kroků p11 až p17 demonstrováných na obrázku (Obr. 30).



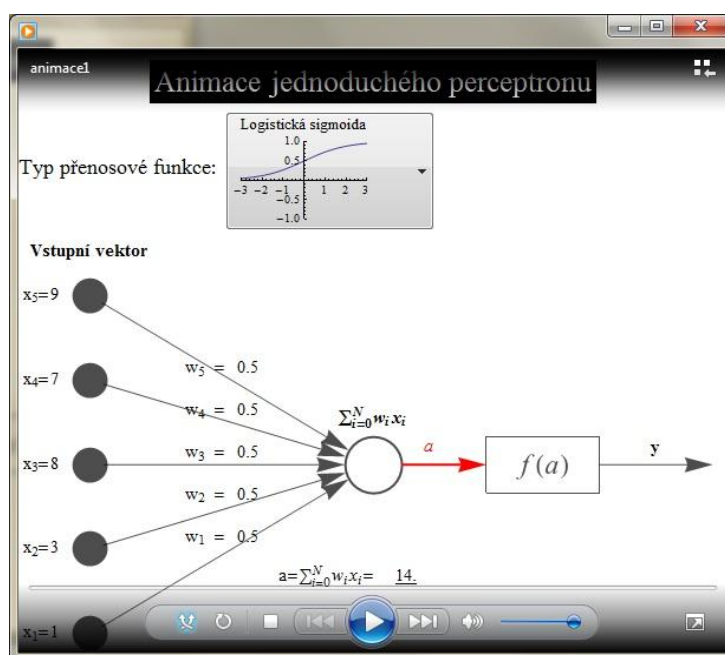
Obr. 30 – Zobrazení jednotlivých kroků animace

4.1.1 Export AVI

Převedení animace do multimediálního formátu AVI jsem docílil následujícím způsobem:

```
Export["animace1.avi", {p11,p12,p13,p14,p15,p16,p17},
"FrameRate"→1]
```

Pro zobrazení výsledného exportu jsem použil multimediální přehrávač Windows Media Player (součástí operačního systému Windows 7) (Obr. 31).



Obr. 31 – Export do multimediálního formátu AVI

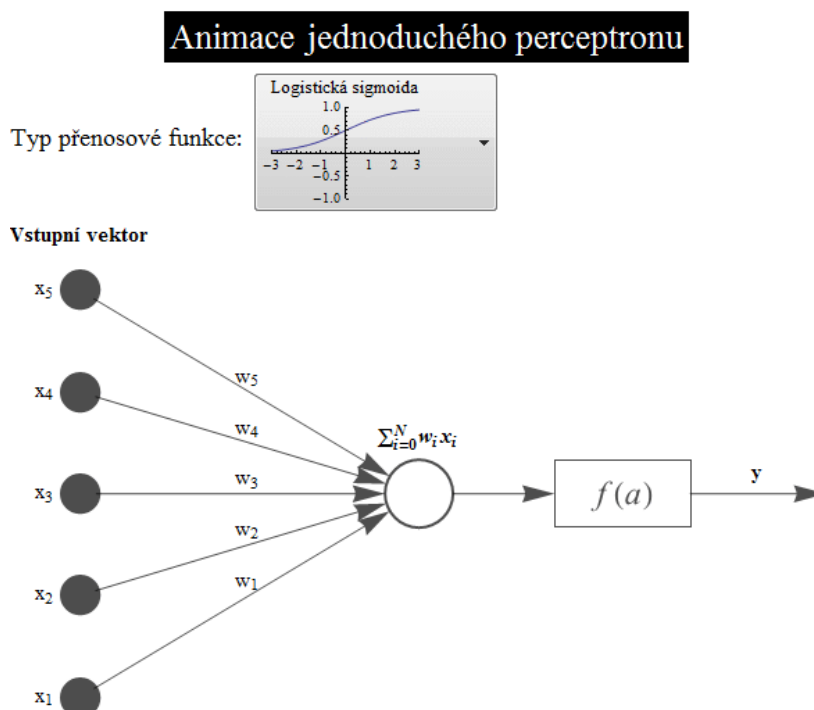
Pokud uživatel nezadá do prvního parametru celou cestu, je soubor automaticky exportován do systémových dokumentů. V případě absence třetího parametru, nebo-li elementů, jsou všechny možné elementy pro tento formát nastaveny na výchozí hodnotu. Formát AVI má možnost využít pouze základní elementy, kterými jsou: Background - nastavení pozadí animace, ImageSize - rozlišení obrazu (pokud je tento element nastaven na Automatic, dochází k vytváření snímků, jejichž rozměry pixelů jsou násobky 4) a FrameRate - počet zobrazovaných snímků za sekundu (rychlost animace).

Výchozí hodnota elementu FrameRate je 30, počet snímků v této animaci je 7. To znamená, že exportovaná animace se přehrálá za necelou čtvrtinu sekundy. Proto jsem byl donucen přidat element FrameRate s hodnotou 7 (doba zobrazení každého snímku je jedna sekunda).

4.1.2 Export GIF

Animace je možné převést i do grafických formátu, jako je GIF. Zápis funkce `Export` je obdobný jako u formátu AVI. Liší se pouze možnostmi volby elementů. Ukázka tohoto exportu byla demonstrována opět na animaci jednoduchého perceptronu. Animace převedena do grafického formátu GIF je zobrazen na obrázku (Obr. 32).

```
Export["animace2.gif", {p11,p12,p13,p14,p15,p16,p17},
"DisplayDurations"→1]
```



Obr. 32 – Export do grafického formátu GIF

Kromě základních elementů jako jsou `Background` a `ImageSize`, má tento formát možnost využít i další elementy: `TransparentColor` – mění zvolenou barvu pixelů na barvu průhlednou (alfa kanál), `DisplayDurations` – zobrazení doby trvání každého snímku v sekundách, `Interlaced` – ukládání obrázků v prokládané formě, což umožňuje postupné vykreslování, `AnimationRepetitions` – určuje počet přehrání animace (pokud je nastaven na hodnotu `Infinity`, animace se neustále opakuje).

Pokud jsem nenastavil v zápisu funkce žádný element, docházelo k rychlému přehrávání animace. To způsoboval element `DisplayDurations`, jehož výchozí hodnota `Automatic` pouze odhaduje dobu trvání jednoho snímku. Proto byl tento element nastaven na hodnotu 1 (každý snímek je zobrazen jednu sekundu).

4.2 Export animace dopředné vícevrstvé sítě

Zatímco animace jednoduchého perceptronu byla vytvořena na základě sedmi daných grafických objektů (Obr. 30), animace dopředné vícevrstvé sítě obsahuje pouze jeden grafický objekt, který se mění v závislosti se zadanými vstupními parametry. Díky tomu jsou zápisy funkce Export těchto animací odlišné.

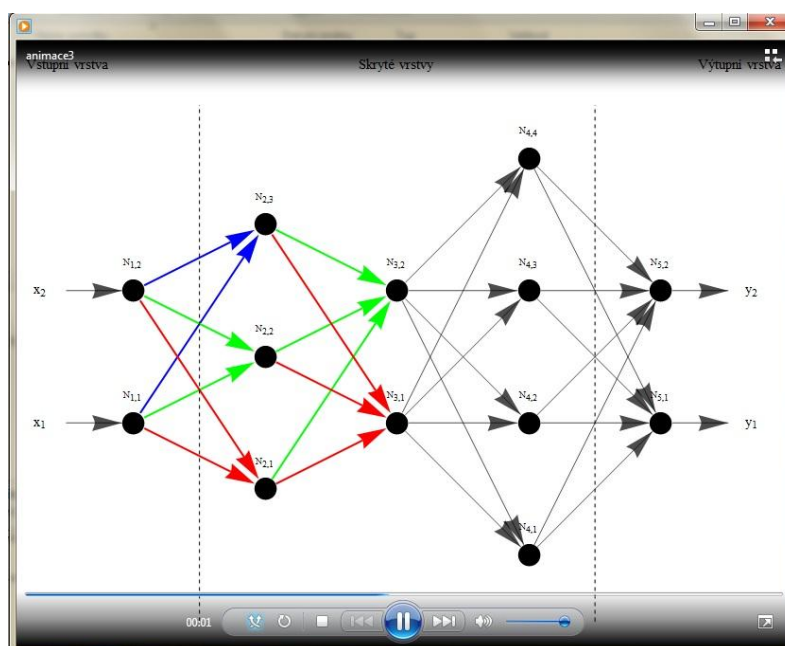
Pro export animace dopředné vícevrstvé sítě jsem využil multimediální formáty FLV a SWF.

4.2.1 Export FLV

Export animace do multimediálního formátu FLV jsem provedl následovně:

```
Export["animace3.flv",
  Manipulate[
    VicVrst[5,2,3,2,4,2,anim],
    {{anim,1," "},1,6,1},
    Bookmarks->{"start":>(anim=1), "end":>(anim=6)}
  ],"ControlAppearance"->None]
```

V tomto zápisu funkce Export je nejdůležitějším prvkem proměnná anim, která ovládá celou animaci. Funkce Manipulate přehrává animaci dopředu a poté i zpět. Proto jsem využil příkazu Bookmarks, který umožňuje nastavení začátku a konce animace. Exportovaná animace do formátu FLV je zobrazena na obrázku (Obr. 33).



Obr. 33 – Export do multimediálního formátu FLV

Tento formát využívá i elementy `CompressionMethod` a `VideoEncoding` pro kompresi a kódování dat, `AnimationDuration` pro nastavení doby animace v sekundách, `ControlAppearance` pro určení způsobu zobrazování ovládacích prvků v exportovaném souboru a `RepeatAnimation` pro nastavení nekonečného opakování animace. Kromě těchto elementů obsahuje formát FLV i základní elementy `Background` a `ImageSize`.

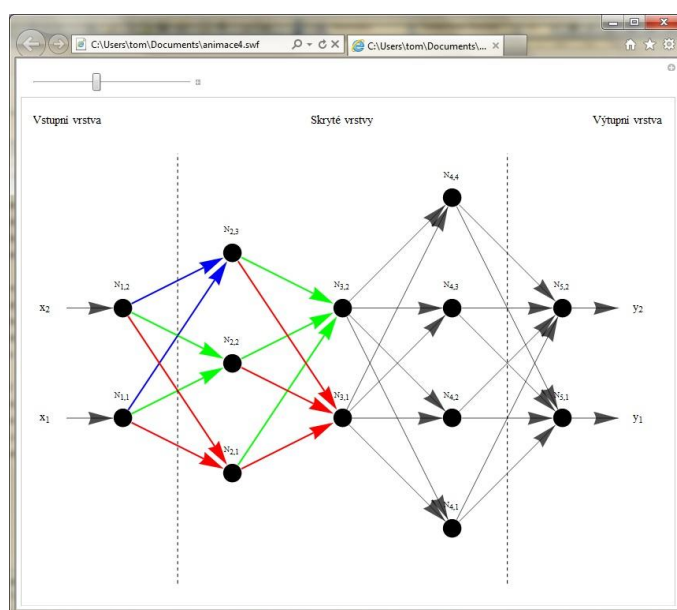
V tomto exportu jsem využil elementu `ControlAppearance` nastaveného na `None`, který v animaci ukryje ovládací prvek.

4.2.2 Export SWF

Tento multimediální formát je analogický formátu FLV. Zápis funkce se liší pouze příponou v prvním parametru.

```
Export["animace4.swf",
  Manipulate[
    VicVrst[5,2,3,2,4,2,anim],
    {{anim,1," "},1,6,1},
    Bookmarks[{"start"→(anim=1), "end"→(anim=6)}],
    "FrameRate"→10,"ControlAppearance"→Automatic,
    "RepeatAnimation"→True]
```

Výslednou animaci jsem zobrazil pomocí internetového prohlížeče Internet Explorer a plug-inu pro podporu flash animací – Adobe Flash Player 11 (Obr. 34).



Obr. 34 – Export do formátu SWF

ZÁVĚR

Hlavním cílem této práce bylo vytvoření animace činnosti umělých neuronových sítí. Proto nebylo potřeba zaměřovat se v první kapitole teoretické části na popis funkcí, které nebyly použity při vytváření animací. Nejprve byly popsány důležité základní funkce pro tvorbu algoritmů a větvení programu. V další části byl uveden výčet funkcí, které slouží k vytváření grafických objektů. U každé z těchto funkcí byl uveden jak zápis, tak i obrázek s příkladným použitím. Velký důraz byl kladen na funkce pro vytváření animací, u kterých byly podrobně rozepsány jejich možnosti i parametry. Nechybělo také seznámení s funkcí Export a formáty, do kterých lze animace exportovat. V poslední kapitole teoretické části o umělých neuronových sítích byly opět zmíněny pouze potřebné informace týkající se této problematiky, ze kterých se vycházelo při vytváření animací.

Praktická část se zabývá animací a vizualizací vybraných typů neuronových sítí, které jsou zobrazeny v tzv. modulech. U jednotlivých modulů bylo popsáno jednak jejich ovládání, tak i postup při jejich tvorbě. Stejně tak byla řešena kapitola, která se zabývá vizualizací topologií jednotlivých typů neuronových sítí. Rozdíl od předešle kapitoly spočívá pouze v tom, že v těchto modulech nemá možnost uživatel spustit animaci přenosu signálu mezi neurony. Tyto moduly slouží pouze pro vizualizaci vybraných typů neuronových sítí včetně nastavování různých parametrů. V závěru praktické části byly detailně popsány veškeré možnosti a nastavení formátů, do kterých bylo možné vytvořené animace exportovat.

Veškeré animace byly vytvořeny na základě důležitých vztahů a informací týkajících se problematiky neuronových sítí a znalostí potřebných funkcí softwaru Mathematica, které byly popsány v teoretické části. Vybrané animace byly poté exportovány do multimediálních (popřípadě grafických) formátů. Problém se vyskytl pouze při exportu do formátu QuickTime (přípona .mov). Tento formát je určen především pro operační systémy Mac, proto měly starší i novější verze operačního systému Windows problém s absencí některých z knihoven tohoto formátu.

ZÁVĚR V ANGLIČTINĚ

The main purpose of this bachelor thesis is to describe how the neural networks work. Hence there was no need to focus on function's description, which weren't use in the first theoretical part. In the first step, the basic functions for algorithm creating and code branching were described. In the next part, the graphics objects making functions were summarized. There were also given accurate description, picture and example how to use for each of this function. The emphasis was placed on creating animations functions with a detailed description of it's options and parameters. We didn't forget to acquaint listeners with export function and it's corresponding formats. In the last part of this bachelor thesis, necessary informations about problems with animations creating were mentioned again.

The practical part deals with animation and visualization of selected neural networks, which are displayed in the modules. The individual modules have been described their control and process of their creation. In this way, the chapter solved, which deals with the visualization of the topology of individual neural networks. The difference from the previous chapter is only in the fact that these modules the user has no way to run an animation of signal transmission between neurons. These modules are used for visualization of selected neural networks, including the setting of various parameters. In conclusion, the practical part were described in detail all the options and settings formats in which this animation could be exported.

All animations were created on the basis of important relationships and information related to the issue of neural networks and knowledge necessary features of the software Mathematica, which were described in the theoretical part. Then were selected animations exported to the multimedia (or graphic) formats. The problem occurred only when exporting to QuickTime format (.mov). This format is designed primarily for the Mac operating systém. Therefore, older and newer versions of Windows had a problem with the absence of libraries of this format.

SEZNAM POUŽITÉ LITERATURY

- [1] CHRAMCOV, Bronislav. *Základy práce v prostředí Mathematica*. Zlín: Univerzita Tomáše Bati Academia centrum, 2006. 2. ISBN 80-7318-510-5
- [2] Seznámení se systémem Mathematica 7: Prostředí M7, definice proměnných. *Mathematica - fórum* [online]. [cit. 2012-06-08]. Dostupné z: http://www.mathematica-forum.cz/materialy/lekce/lekce_01.nb
- [3] Loops and Control Structures. *Wolfram Mathematica Documentation center* [online]. [cit. 2012-06-08]. Dostupné z: <http://reference.wolfram.com/mathematica/tutorial/LoopsAndControlStructures.html>
- [4] Plot. *Wolfram Mathematica Documentation center* [online]. [cit. 2012-06-08]. Dostupné z: <http://reference.wolfram.com/mathematica/ref/Plot.html>
- [5] The Structure of Graphics. *Wolfram Mathematica Documentation center* [online]. [cit. 2012-06-08]. Dostupné z: <http://reference.wolfram.com/mathematica/tutorial/TheStructureOfGraphics.html>
- [6] Plotting Lists of Data. *Wolfram Mathematica Documentation center* [online]. [cit. 2012-06-08]. Dostupné z: <http://reference.wolfram.com/mathematica/tutorial/PlottingListsOfData.html>
- [7] Introduction to Manipulate. *Wolfram Mathematica Documentation center* [online]. [cit. 2012-06-08]. Dostupné z: <http://reference.wolfram.com/mathematica/tutorial/IntroductionToManipulate.html>
- [8] How to: Create Animations. *Wolfram Mathematica Documentation center* [online]. [cit. 2012-06-08]. Dostupné z: <http://reference.wolfram.com/mathematica/howto/CreateAnimations.html>
- [9] Exporting Graphics and Sounds. *Wolfram Mathematica Documentation center* [online]. [cit. 2012-06-08]. Dostupné z: <http://reference.wolfram.com/mathematica/tutorial/ExportingGraphicsAndSounds.html>

- [10] Audio Video Interleave. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-06-08]. Dostupné z: http://cs.wikipedia.org/wiki/Audio_Video_Interleave
- [11] What is FLV format?. *Digiarty software* [online]. [cit. 2012-06-08]. Dostupné z: <http://www.winxdvd.com/resource/flv.htm>
- [12] QuickTime. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-06-08]. Dostupné z: <http://cs.wikipedia.org/wiki/QuickTime#Historie>
- [13] GIF, JPEG a PNG - jak a kdy je použít?. Interval.cz [online]. [cit. 2012-06-08]. Dostupné z: <http://interval.cz/clanky/gif-jpeg-a-png-jak-a-kdy-je-pouzit/>
- [14] TUČKOVÁ, Jana. Úvod do teorie a aplikací umělých neuronových sítí. Vyd. 1. Praha : Vydavatelství ČVUT, 2003. 103 s. ISBN 80-010-2800-3.
- [15] Úvod do neuronových sítí. Neuronové sítě [online]. [cit. 2012-06-08]. Dostupné z: http://cyber.felk.cvut.cz/gerstner/biolab/bio_web/teach/FunBio/neuron/neursite.html
- [16] ZELINKA, Ivan. UMĚLÁ INTELIGENCE I: Neuronové sítě a genetické algoritmy. VUT Brno: VUTIMUM, 1998. ISBN 80-214-1163-5.
- [17] Neuronová síť. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-06-08]. Dostupné z: http://cs.wikipedia.org/wiki/Neuronov%C3%A1_s%C3%AD%C5%A5
- [18] Úvod do neuronových sítí. AUTOMA: Časopis pro automatizační techniku [online]. [cit. 2012-06-08]. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=30255
- [19] Perceptron. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-06-08]. Dostupné z: <http://cs.wikipedia.org/wiki/Perceptron>
- [20] Klasifikátor pro lineárně separovatelné obrazy. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-06-08]. Dostupné z: <http://upload.wikimedia.org/wikipedia/commons/d/d5/LinearneSeparovatelne.png>

- [21] Multilayer Neural Network. [online]. [cit. 2012-06-08]. Dostupné z: <http://www.teco.uni-karlsruhe.de/~albrecht/neuro/html/node18.html>
- [22] Examples of Multilayer Neural Network Architectures. [online]. [cit. 2012-06-08]. Dostupné z: <http://www.teco.uni-karlsruhe.de/~albrecht/neuro/html/img4.gif>
- [23] BÍLA, Jiří. Umělá inteligence a neuronové sítě v aplikacích. Vyd. 1. Praha: Vydavatelství ČVUT, 1995, 115 s. ISBN 80-010-1275-1.
- [24] Hopfieldova síť. [online]. [cit. 2012-06-08]. Dostupné z: http://avari.cz/uir/gfx/hopf_1.png
- [25] Lineární asociativní paměť. Neuronové sítě [online]. [cit. 2012-06-08]. Dostupné z: <http://avari.cz/uir/index.php?pg=lam>
- [26] Lineární asociativní paměť (LAM). Neuronové sítě [online]. [cit. 2012-06-08]. Dostupné z: http://avari.cz/uir/gfx/lam_1.png
- [27] Obousměrná asociativní paměť. Neuronové sítě [online]. [cit. 2012-06-08]. Dostupné z: http://moon.felk.cvut.cz/~pjv/Jak/_neur/n566/bam.html
- [28] BAM. Neuronové sítě [online]. [cit. 2012-06-08]. Dostupné z: http://moon.felk.cvut.cz/~pjv/Jak/_neur/n566/obr01.gif

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AVI	Audio Video Interleave.
FLV	Flash Video.
GIF	The Graphics Interchange Format.
UNS	Umělé neuronové síť.
TLU	Threshold Logica Unit.
CLN	Competetive Learning Network.
LAM	Linear Associative Memory.
BAM	Bidirectional Associative Memory.
ART	Adaptive Resonant neTwork

SEZNAM OBRÁZKŮ

<i>Obr. 1 - Ukázka zobrazovacích možností softwaru Mathematica</i>	11
<i>Obr. 2 - Vizualizace funkce pomocí příkazu Plot</i>	14
<i>Obr. 3 - Vizualizace objektů pomocí funkce Graphics</i>	15
<i>Obr. 4 - Zobrazení grafických výstupů do jednoho grafu pomocí Show</i>	16
<i>Obr. 5 - Funkce Manipulate.....</i>	18
<i>Obr. 6 - Funkce Animate.....</i>	19
<i>Obr. 7 - Model McCulloch-Pittsova neuronu</i>	23
<i>Obr. 8 - Nelineární hranice mezi třídami</i>	26
<i>Obr. 9 - Vícevrstvá síť.....</i>	27
<i>Obr. 10 - Různé typy neuronových sítí.....</i>	27
<i>Obr. 11 - Schéma sítě perceptron</i>	28
<i>Obr. 12 - Klasifikátor pro lineárně separovatelné obrazy</i>	29
<i>Obr. 13 - Typy vícevrstevných neuronových sítí.....</i>	30
<i>Obr. 14 - Obecné schéma Hopfieldovy sítě</i>	31
<i>Obr. 15 - Schéma sítě CLN.....</i>	31
<i>Obr. 16 - Schéma sítě LAM.....</i>	32
<i>Obr. 17 - Schéma sítě BAM.....</i>	33
<i>Obr. 18 - Kohonenova síť</i>	33
<i>Obr. 19 - Schéma sítě ART.....</i>	34
<i>Obr. 20 – Výběr pomocí výběrového menu.....</i>	38
<i>Obr. 21 - Výběr pomocí rozbalovacího menu.....</i>	38
<i>Obr. 22 - Modul Animace jednoduchého perceptronu</i>	39
<i>Obr. 23 - Modul Animace dopředné vícevrstvé sítě.....</i>	40
<i>Obr. 24 - Modul Vícevrstvá neuronová síť</i>	42

<i>Obr. 25 - Modul Hopfieldova síť</i>	43
<i>Obr. 26 - Modul Síť CLN</i>	44
<i>Obr. 27 - Modul Síť BAM</i>	45
<i>Obr. 28 - Modul Kohonenova síť</i>	46
<i>Obr. 29 - Modul Síť ART.....</i>	47
<i>Obr. 30 - Zobrazení jednotlivých kroků animace</i>	48
<i>Obr. 31 - Export do multimediálního formátu AVI.....</i>	49
<i>Obr. 32 - Export do grafického formátu GIF</i>	50
<i>Obr. 33 - Export do multimediálního formátu FLV.....</i>	51
<i>Obr. 34 - Export do multimedialního formátu SWF</i>	52

SEZNAM PŘÍLOH

CD se zdrojovými kódy vytvořených animací a animace exportované do multimediálních formátů

