

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



**Diplomová práce**

**Klasifikace pomocí neuronových sítí a vektorových strojů**

**Anton Lesniak**

© 2019 ČZU v Praze

# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Anton Lesniak

Informatika

---

Název práce

**Klasifikace pomocí neuronových sítí a vektorových strojů**

Název anglicky

**Classification with neural networks and vector machines**

### Cíle práce

V práci bude porovnána efektivita klasifikačních algoritmů založených na neuronových sítích a klasifikačních algoritmů založených na vektorových strojích (support vector machines).

### Metodika

V práci bude porovnána efektivita klasifikačních algoritmů založených na neuronových sítích a klasifikačních algoritmů založených na vektorových strojích (support vector machines). Diplomant se zaměří především na případ, kdy klasifikované objekty jsou popsány mnoha parametry a k dispozici jsou pouze relativně malé učební soubory. Pro práci použije open source software přístupný na webu a data z veřejného úložiště dat Kalifornské univerzity, která jsou určena pro testování rozhodovacích modelů.

**Doporučený rozsah práce**

60

**Klíčová slova**

neuronové sítě, vektorové stroje

**Doporučené zdroje informací**

BISHOP, C M. *Pattern recognition and machine learning*. [New York]: Springer, 2006. ISBN 0-387-31073-8.

**Předběžný termín obhajoby**

2018/19 ZS – PEF (únor 2019)

**Vedoucí práce**

doc. Ing. Arnošt Veselý, CSc.

**Garantující pracoviště**

Katedra informačního inženýrství

Elektronicky schváleno dne  
7. 3. 2018

---

**Ing. Martin Pelikán, Ph.D.**  
Vedoucí katedry

Elektronicky schváleno dne  
7. 3. 2018

---

**Ing. Martin Pelikán, Ph.D.**  
Děkan

V Praze dne 27. 03. 2019

### **Čestné prohlášení**

Prohlašuji, že svou diplomovou práci " Klasifikace pomocí neuronových sítí a vektorových strojů" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne

---

### **Poděkování**

Rád bych vyjádřil své poděkování doc. Ing. Arnoštu Veselému, CSc., vedoucímu mé diplomové práce za odborné poradenství a cenné rady. Dále bych taky poděkoval mé rodině za podporu.

# Klasifikace pomocí neuronových sítí a vektorových strojů

## Abstrakt

Umělá neuronová síť je jeden z výpočetních modelů používaných v umělé inteligenci. Jejím vzorem je chování odpovídajících biologických struktur. Umělá neuronová síť je struktura určená pro distribuované paralelní zpracování dat.

Skládá se z umělých (nebo také formálních) neuronů, jejichž předobrazem je biologický neuron. Neurony jsou vzájemně propojeny a navzájem si předávají signály a transformují je pomocí určitých přenosových funkcí. Neuron má libovolný počet vstupů, ale pouze jeden výstup.

Neuronové sítě se používají mimo jiné i pro rozpoznávání a kompresi obrazů nebo zvuků, předvídání vývoje časových řad (např. burzovních indexů), někdy dokonce k filtrování spamu. V lékařství slouží k prohlubování znalostí o fungování nervových soustav živých organismů. Například perceptronová síť vznikla původně jako simulace fyziologického modelu rozpoznávání vzorů na sítnici lidského oka.

**Klíčová slova:** vektorový stroj, strojové učení, neuronová síť, neuron, umělá inteligence, klasifikace

# Classification with neural networks and vector machines

## Abstract

Artificial neural networks (ANNs) or connectionist systems are computing systems inspired by the biological neural networks that constitute animal brains. Such systems learn (progressively improve performance) to do tasks by considering examples, generally without task-specific programming. For example, in image recognition, they might learn to identify images that contain cats by analysing example images that have been tagged "cat" or "no cat" and using the analytic results to identify cats in untagged images. They have found most use in applications difficult to express in a traditional computer algorithm using rule-based programming.

**Keywords:** vector machine, machine learning, artificial neural network, neuron, artificial intelligence, classification

# Obsah

<b>1 Úvod.....</b>	<b>10</b>
<b>2 Cíl práce a metodika .....</b>	<b>12</b>
<b>2.1 Cíl práce .....</b>	<b>12</b>
<b>2.2 Metodika .....</b>	<b>12</b>
<b>3 Teoretická východiska .....</b>	<b>13</b>
<b>3.1 Historie .....</b>	<b>13</b>
<b>3.2 Umělá inteligence .....</b>	<b>16</b>
<b>3.3 Strojové učení .....</b>	<b>17</b>
3.3.1 Úlohy strojového učení .....	18
3.3.2 Modely strojového učení .....	19
3.3.2.1 Rozhodovací stromy .....	19
3.3.2.2 Neuronové sítě .....	20
3.3.2.3 Podpůrné vektorové stroje .....	21
3.3.3 Strojové učení a MATLAB.....	22
3.3.3.1 Učení a predikce .....	23
<b>3.4 Oblastí použití.....</b>	<b>24</b>
3.4.1 Ekonomika o podnikání .....	24
3.4.2 Služby bezpečnosti .....	25
3.4.3 Zdravotnictví.....	26
3.4.4 Počítačové hry.....	26
3.4.5 Počítačové vidění .....	28
<b>4 Praktická část .....</b>	<b>30</b>
<b>4.1 Neuronové sítě .....</b>	<b>30</b>
4.1.1 Model neuronových sítí .....	30
4.1.1.1 Perceptron.....	30
4.1.1.2 Aktivační funkce .....	31
4.1.2 Učení neuronové sítě .....	33
4.1.2.1 Informace o sadě dat.....	33
4.1.2.2 Vložení dat do neuronové sítě: .....	35
4.1.2.3 Proces učení.....	35
4.1.2.4 Analýza.....	37
4.1.2.5 Testování neuronové sítě .....	42
<b>4.2 Podpůrné vektorové stroje .....</b>	<b>43</b>
4.2.1 Klasifikační model SVM .....	43



4.2.2	Učení SVM .....	45
4.2.2.1	Kontrola modelů SVM .....	46
4.2.2.2	Analýza dat .....	47
4.2.3	Testování SVM .....	51
<b>5</b>	<b>Výsledky a diskuse .....</b>	<b>53</b>
<b>5.1</b>	<b>Efektivita modelů na 300 množstvích dat .....</b>	<b>53</b>
5.1.1	Testování neuronové sítě .....	53
5.1.2	Testování SVM .....	55
<b>5.2</b>	<b>Efektivita modelů na 400 množstvích dat .....</b>	<b>56</b>
5.2.1	Testování neuronové sítě .....	56
5.2.2	Testování SVM .....	58
<b>6</b>	<b>Závěr.....</b>	<b>59</b>
<b>7</b>	<b>Citovaná literatura.....</b>	<b>60</b>

# 1 Úvod

Umělé neuronové sítě jsou matematické modely, které provádějí paralelní výpočty pomocí množství vzájemných jednoduchých procesů. Každý z těchto procesů neboli neuronů v biologickém systému pracuje pouze na přijímání a odevzdávání signálů, což je mnohem snazší než libovolná vyčíslování na personálním počítači. Po vzájemném propojení jednotlivých elementů a sestavení dostatečně velké sítě signály se řízenou interakcí transformují pro řešení nejtěžších problémů moderní vědy.

První propojení a popsání fyziologických procesů v mozku pomocí matematických modelů začalo ještě v polovině 20. století, ale velký rozvoj začal pouze v 90. letech, když už byly k dispozici hlubší teoretické znalosti a taky dokonalé výpočetní stroje. «Umělé» sítě se tak nazývají proto, aby zdůraznit, že to není reálný biologický proces, ale jen pokus simulovat mozek v číselné podobě pro objasnění jedné z hlavních tajemství lidského života.

V současné době umělé neuronové sítě reprezentují pouze zjednodušený model lidského mozku, hlavním jejích cílem je zpracování a vyčíslování konkrétních matematických úkolů hodně užitečných v současném světě. Čím lepe budeme rozumět zásadám fungování umělých neuronových sítí, tím globálnější budou výsledky v neurofyziologii, v ekonomice (prognózování, modelování) nebo v jiných oblastech lidského života.

Ale velké údaje zatěžují výuku neuronových, a to díky obrovskému množství zpracovávaných dat a následným požadavkům na paměť a výpočet. Dalším problémem v klasifikaci je datová nelinearita, která charakterizuje funkci překrývání různých tříd, což uděluje úkol separace tříd obtížnější. Zejména z těchto důvodů k výběru vhodné síťové architektury se začátkem 21. století byly preferovány podpůrné vektorové stroje (SVM), které ukázaly být v mnoha ohledech lepší než neuronové sítě.

SVM nabízí principiální přístup k problémům strojového učení díky svým matematickým základům v teorii statistického učení. SVM konstruuje řešení jako vážený součet podpůrných vektorů, které jsou pouze podmnožinou tréninkového vstupu. Podobně jako neuronové sítě SVM minimalizuje konkrétní funkci nákladů na chyby založené na souboru údajů o školení a spoléhá na empirický model rizika. Kromě toho SVM používá

minimalizaci strukturálních rizik a ukládá dodatečné omezení na problém optimalizace, což nutí krok k nalezení modelu, který bude nakonec lépe zobecněn.

V diplomové práci budou shrnuty výhody a řešení problém efektivity obou modelů v konkrétních situacích s použitím různého množství malých dat.

## 2 Cíl práce a metodika

### 2.1 Cíl práce

V práci bude porovnána efektivita klasifikačních algoritmů založených na neuronových sítích a klasifikačních algoritmů založených na vektorových strojích (support vector machines). Ve spojitosti s tím, že jsou to nástroje strojového učení, lze formulovat dílčí cíle:

- Vysvětlení teoretického principu strojového učení
- Popsaní praktického využití strojového učení
- Vyhledávání potřebných dat pro testování modelů
- Zvolení nejvhodnějšího softwaru pro další práci z data
- Vytvoření klasifikačních modelů neuronových sítí a podpůrných vektorových strojů v prostředí použitého softwaru
- Porovnání efektivity dvou klasifikačních algoritmů
- Shrnutí výsledků a závěru

### 2.2 Metodika

Metodika dané diplomové práce je založena především na zpracování dostupných informačních zdrojů v literární i internetově podobě jak v českém, tak v anglickém jazykách v oblasti umělé inteligence, strojového učení a algoritmech neuronových sítí a vektorových strojů, jejich rozbor a třídění. V práci jsou vysvětleny jak teoretické principy, tak praktická vyjádření daných nástrojů. Pro umělé neuronové sítě jsou hlavně popsány rozdíly a shody s biologickým systémem. Následně budou objasněny principy klasifikaci a porovnání jí s jinými matematickými algoritmy. Současně budou navrženy metodické pokyny pro učení a testování klasifikačních modelů neuronových sítí a vektorových strojů na otevřené platformě MATLAB přístupné na webu s využitím veřejně dostupných dat Kalifornské univerzity, kdy klasifikované objekty jsou popsány mnoha parametry a k dispozici jsou pouze relativně malé učební soubory. Dále bude provedená analýza výsledků na základě teoretických poznatků a formulování závěru diplomové práce, výhod a nevýhod používání daných algoritmů, jejich efektivity v praktickém využití.

## 3 Teoretická východiska

### 3.1 Historie

První elektronický počítač se objevil již během druhé světové války ve Spojených státech amerických. Nazýval se ENIAC (*Electronic Numerical Integrator And Calculator*) a byl vyprojektován v roce 1943 pro potřeby Americké armády. Fungoval až 1000krát rychleji než jeho mechanické předchůdce. Jeho nevýhodou bylo ale to, že zaujímal obrovské prostory a ochlazoval se pomocí leteckých motorů. (1)

Výzkumníci Warren McCullock a Walter Pitts publikovali svůj první koncept zjednodušené mozkové buňky v roce 1943. Ten se nazývá McCullock-Pitts (MCP) neuron. Takovou nervovou buňku popsali jako jednoduchou logickou bránu s binárními výstupy. (2)

Alan Turing, britský matematik a zakladatel moderní informatiky v roce 1950 představil tzv. Turingův test, který se pokouší ověřit, zda se daný stroj nebo systém chová inteligentně. Vzhledem k tomu, že inteligence je obtížně ověřitelná koncepce, porovnává inteligenci s člověkem. Počítačovým systémům to trvalo dlouho, než tento test zvládly. Poprvé to bylo až v roce 2014. (3)

V roce 1950 Von Neumann, Eckert a Mauchly navrhli nástupce ENIAC zvaného EDVAC (*Electronic Discrete Variable Automatic Computer*), už to byl programovatelný počítač. Von Neumann takzve vynalezl, že všechna data a pokyny se mohou být uloženy ve stejné jednotce a nazval jí «paměť». V témže roce matematik Norbert Wiener přišel na to, že všechny děje a rozhodnutí v mozku fungují na principu zpětné vazby. (4)

Poté Newell a Simon v roce 1956 vynalezli takzvaný „myslící stroj“ a nazvali ho „*The Logic Theorist*“. Tento počítačový program byl schopen dokázat různé matematické věty v symbolické podobě, čímž simuloval některé aspekty lidského myšlení, a proto je považován za první inteligentní program. (5)

Za otce umělé inteligence je považován John McCarthy, který v roce 1956 organizoval konferenci “*The Dartmouth summer research project on artificial intelligence.*” pro všechny, kdo se zajímá o počítačovou inteligenci. Přesně od této chvíle se tomuto oboru začalo říkat AI. V roce 1957 se testoval program “*The General Problem Solver*” (GPS),

který vytvořili ti samí lidé jako *Logic Theorist*. GPS pracoval na principu zpětné vazby a byl schopen řešit velké množství běžných problémů.

Arthur Samuel z IBM (International Business Machines) vymyslel pojem strojové učení, tj. podoblast umělé inteligence, která pomáhá strojům se učit, postupně se zdokonalovat, když pracoval v oblasti počítačových her v roce 1959. (6)

Dalším významným rokem ve vývoji AI byl 1963, kdy Massachusettský technologický institut (angl. MIT) dostal obrovský grant o hodnotě 2,2 milionů dolarů od agentury amerického ministerstva obrany(angl. ARPA) na zajištění technologické výhody oproti Sovětskému svazu. Tím projektem zvýšila rychlost vývoje a financování výzkumů v oborech spojených s AI.

V sedmdesátých letech vzniklo mnoho nových metod. Například teorie Davida Marra o rozeznání počítačovým programem obrazů na základě odstínu, barvy, tvaru a textury nebo vznik programovacího jazyka PROLOG. Nejvýznamnější novinkou byl vznik expertních systémů, které umožnily předvídat pravděpodobnost řešení konkrétní situaci.

Od osmdesátých let nejrůznější podniky jako Digital Electronics, DuPont, General Motors a Boeing začaly užívat XCON (*for eXpert CONfigurer*), expertní systém navrženy pro programování na VAX (*Virtual Address eXtension*) počítačích.

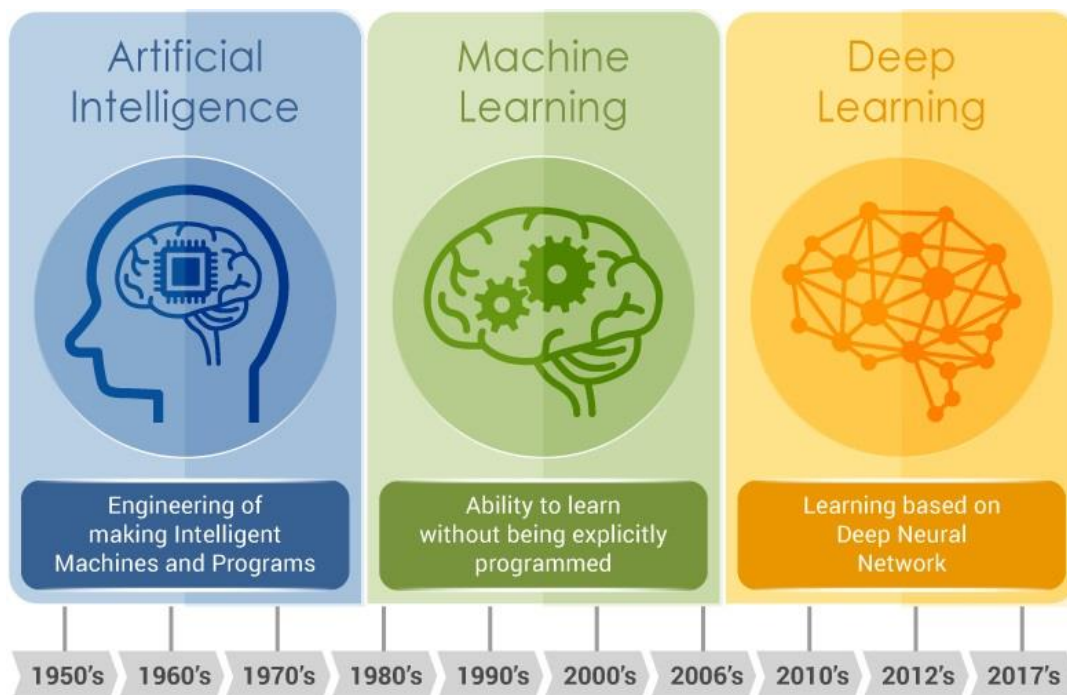
Pro další růst umělé inteligence bylo zapotřebí vyšší výpočetní síla počítačů, vylepšené algoritmy strojového učení a nahromadění dat, ke kterým dochází právě teď. (7). Strojové učení je každým rokem stále populárnější. Počet zprav, článků, výzkumu a práce v této oblasti se každým rokem zvyšuje. Je to jasně vidět, stačí jen použít statistiky "*Google Trends*". Tato statistika je sestavena pro všechny odkazy na téma „*Machine learning*“ v Internetu za posledních 15 let na celém světě. (Obrázek 1) (8)



Obrázek 1. Vyhledávání "Machine learning" na webu za posledních 15 let.

Převzato a pozměněno z: <https://trends.google.com/trends>

Se začátkem 21. století začíná vývoj hlubokého učení, ke kterému se spadájí i umělé neuronové sítě (angl. Artificial neural networks). Ačkoli první pokusy začaly někde v šedesátých letech, jejich komerční používání přišlo jen kolem roku 2010 s obecně dostupným výkonným hardwarem. (Obrázek 2). AI technologie budou i nadále rychle vyvíjet v roce 2019 a stanou se více široce dostupné díky cenové dostupnosti a zpracování velkých objemů dat. (9)



Obrázek 2. Chronologie vývoje umělé inteligence. Převzato z:  
<https://www.viatech.com/en/2018/05/history-of-artificial-intelligence>

### 3.2 Umělá inteligence

Umělá inteligence (AI) je širší pojem strojů schopných provádět úkoly způsoby, které bychom mohli považovat za „chytré“. Může napodobovat lidské inteligenci, ale to není nutná podmínka. Nezbytné procesy zahrnují učení, uplatnění získaných znalostí pomocí předem definovaných pravidel (tj. něco jako uvažování) a konečně opravování sebe sama. Je často klasifikována do jedné ze dvou základních skupin – specializovaná (angl. applied) nebo obecná (angl. general). Specializovanou AI je mnohem běžnější, jsou to systémy navržené pro virtuální asistenti. Například Alexa, Siri nebo Cortana jsou naprogramovány na specifické úkoly. Obecná AI může teoreticky zvládnout jakýkoli úkol sama, bez pomoci, a dokonce i napodobovat kognitivním schopnostem člověka. Skutečná obecná umělá inteligence ještě neexistuje. (10) (11)



### 3.3 Strojové učení

Strojové učení (angl. Machine Learning, ML) je jednou z klíčových podoblastí umělé inteligence, je dynamická adaptace stroje na nové situace a využití dostupných dat.

Strojové učení dává počítači soubor pravidel a úkolů, pak nechá ho zjistit způsob, jak tyto úkoly splnit. Stroj v podstatě začíná bez znalosti a díky pokusům a chybám přichází s vhodným řešením.

Existuje několik způsobů, jak se stroje mohou učit, jak mohou zlepšovat své algoritmy:

- **Učení s učitelem (angl. supervised learning)** – vyžaduje pomoc člověka, který zadává vstupní a výstupní data a upravuje přesnost předpokládaného strojem řešení. Učitel ručně označí nesprávně určené objekty a tím dochází k přenastavení algoritmů.
- **Učení bez učitele (angl. unsupervised learning)** – nepotřebuje dohled, protože nejde člověkem stanovit správný výstup, zkoumá data a vyspívá k výsledkům samostatně. Tento přístup se používá například pro zjištění potenciálního útočnicka v síti na základě behaviorální analýzy. Stroj nasbírá obrovská data o typickém provozu sítě a porovná je s atypickým chováním. Patří sem neuronové sítě.
- **Kombinace obou přístupů (angl. semi-supervised learning)** - dokáže zrychlit učení bez učitele. Například tím, že v předchozím případě budou člověkem označeni fiktivní útočníci a obyčejní uživatelé. Systém pak sám učiní novou analýzu chování stanovených útočnicků od běžných uživatelů a přenastaví své algoritmy.
- **Zpětnovazební učení (angl. reinforcement learning)** – stroj se učí ze zkušeností, samostatně hledá nejlepší cestu, aby dospěl k žádoucímu výsledku, ale algoritmus není trénován s existující správnou odpovědí jako při učení s učitelem. Využívá se to třeba u průmyslových strojů, které se postupně samy naučí urychlit a specifikovat prováděné operace na bázi analýzy jejich výsledku.

(6) (11)

### 3.3.1 Úlohy strojového učení

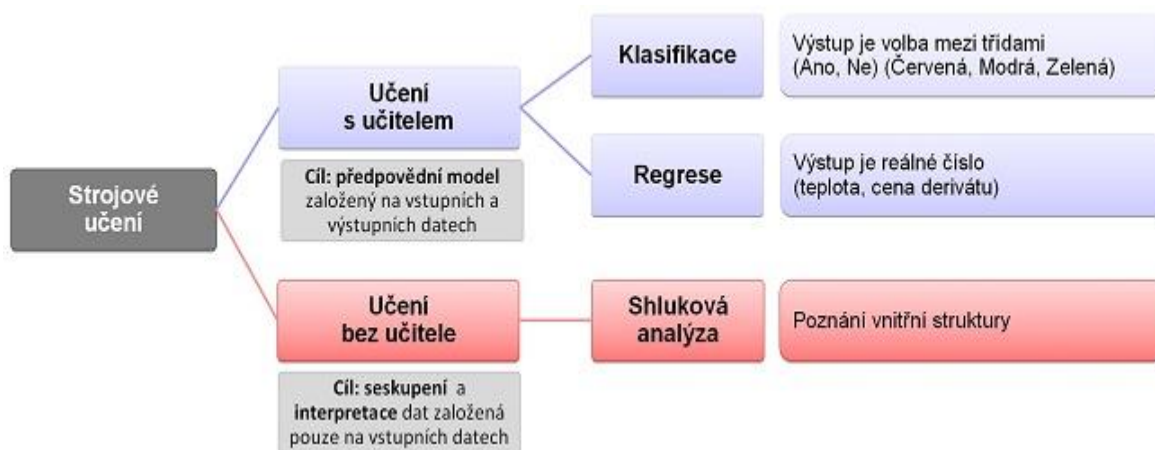
Hlavními úkoly strojového učení jsou **klasifikace**, **regrese** a **shluková analýza**.

**Klasifikace** používá matematický model k rozdělení objektů popsaných vstupními daty do skupin. Například předvídat, zda je daný e-mail spam nebo ne spam? Přidělení určitého článku zprávy do skupiny jako je sport, počasí nebo věda. Bude dnes pršet nebo ne?

**Regrese** odhaduje výstupní číselné hodnoty na základě vstupních dat. Některé příklady jsou cena domu v konkrétním městě se 3 ložnicemi a nad 2000 m<sup>2</sup>? Předpovídání finančních výsledků a ceny akcií.

**Shluková analýza** zkoumá v datech přirozené spojení na základě podobných charakteristik, je příkladem učení bez učitele, když klasifikace a regrese jsou součástí učení s učitelem. (Obrázek 3). Shluková analýza se podobně věci seskupuje společně. Můžeme se s tím setkat, když obdržíme knihu nebo filmové doporučení na základě předchozích nákupů nebo vyhledávání.

(12) (13)



Obrázek 3. Rozdělení strojového učení podle typu úlohy. Převzato z:

<https://sciencemag.cz/strojove-uceni-a-matlab>

### 3.3.2 Modely strojového učení

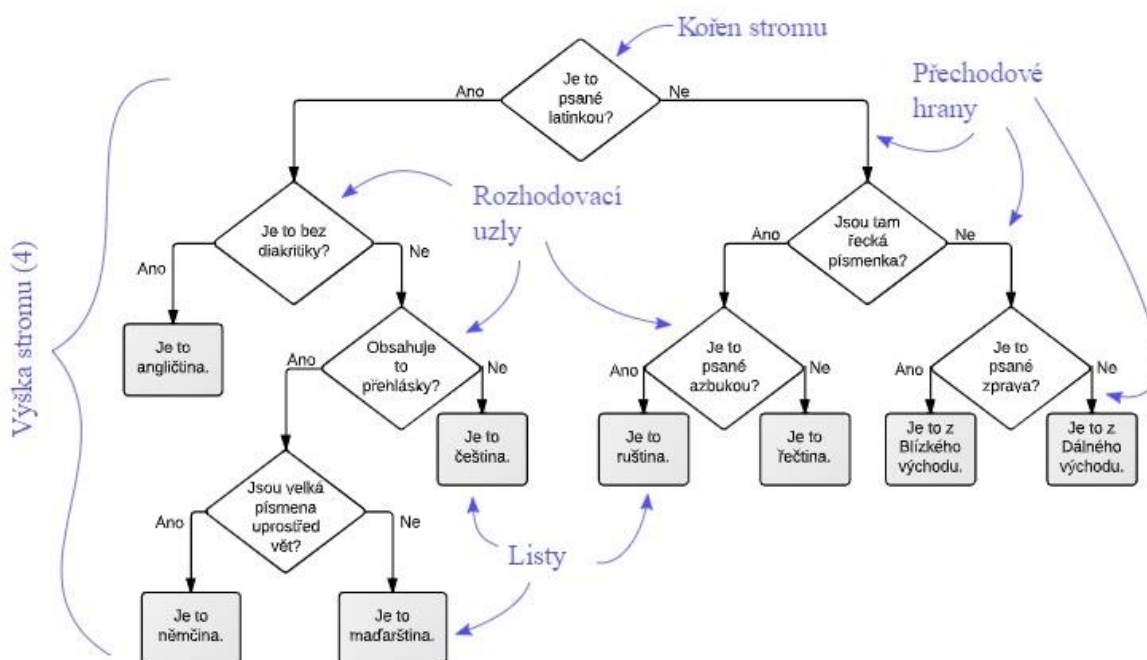
Pro konkrétní řešení je třeba vybrat správný model, každý z nich používá různé algoritmy v rámci výše uvedených úkolů.

Nejpoužívanější modely:

#### 3.3.2.1 Rozhodovací stromy

Rozhodovací stromy jsou rozšířeným a užitečným nástrojem strojového učení určeným pro úlohy klasifikace a regrese. Rozhodovací stromy mají čtyři hlavní části (Obrázek 4): kořen stromu, přechodové hrany, rozhodovací uzly a listy. Kořenový uzel je výchozím bodem stromu, přechodové hrany obsahují otázky nebo kritéria, která mají být zodpovězena.

Přechodové hrany jsou šipky spojující uzly, ukazující postup od otázky k odpovědi. Každý uzel má obvykle dva nebo více uzlů, které se od něj rozprostírají. Pokud například otázka v prvním uzlu vyžaduje odpověď „ano“ nebo „ne“, bude pro odpověď „ano“ jeden uzel listu a další uzel „ne“. (14)



Obrázek 4. Rozhodovací strom pro určení jazyka z psaného textu. Převzato z:

[https://popelka.ms.mff.cuni.cz/~lessner/mw/index.php/Učebnice/Informace/Rozhodovací\\_stromy\\_a\\_chytré\\_otázky](https://popelka.ms.mff.cuni.cz/~lessner/mw/index.php/Učebnice/Informace/Rozhodovací_stromy_a_chytré_otázky)

### 3.3.2.2 Neuronové sítě

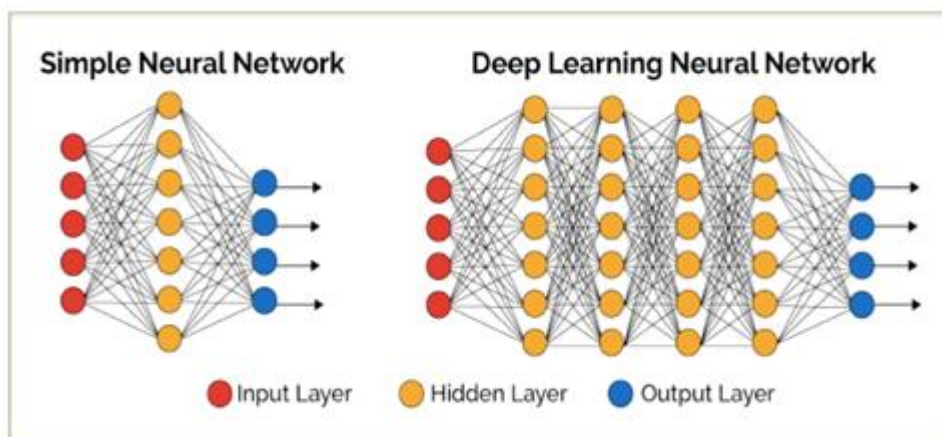
Umělá neuronová síť je nejsložitější statistický model, protože má více vrstev, který má schopnost učit se podobně jako lidský mozek. Ačkoli první pokusy začaly někde v šedesátých letech, jejich komerční expanze přišla jen kolem roku 2010 spolu s dostupným výkonným hardwarem.

Umělá neuronová síť má obvykle několik desítek až miliónů umělých neuronů, nazývaných jednotky a uspořádaných v sérii vrstev. Vstupní vrstva (angl. Input layer) přijímá různé formy informací z okolního světa. Jedná se o data, o která se síť snaží o nich dozvědět. Ze vstupní jednotky procházejí data jednou nebo více skrytými jednotkami (angl. Hidden layer). Úkolem skryté jednotky je transformovat vstup na něco, co může výstupní jednotka použít.

Většina neuronových sítí je plně propojena z jedné vrstvy do druhé. Čím vyšší je číslo, tím větší vliv má jedna jednotka na druhou, podobně lidskému mozku. Vzhledem k tomu, že data procházejí každou jednotkou, síť se o těchto datech dozví více. Na druhé straně sítě jsou výstupní jednotky (angl. Output layer), tam, kde síť reaguje na data, která byla dána a zpracována. (15)

**Hluboké učení** je podmnožina strojového učení a je založena na hlubokých neuronových sítích, které mají mnoho vrstev pro učení v několika krocích. (Obrázek 5).

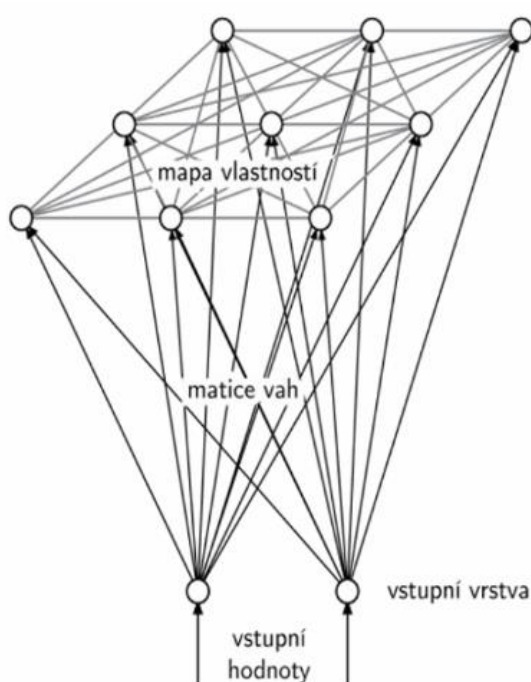
Komplexní hluboké neuronové sítě často provádějí rozpoznávání obrazu zpracováním hierarchie vlastností, kde každá vrstva hledá další složité objekty. (16)



Obrázek 5. Porovnání prosté a hluboké neuronové sítě. Převzato z:

<https://towardsdatascience.com/notes-on-artificial-intelligence-ai-machine-learning-ml-and-deep-learning-dl-for-56e51a2071c2>

**Samoorganizující se neuronová síť** se podle svého tvůrce někdy nazývá Kohonenova síť. Jednovrstvá síť, učení algoritmus v „učení bez učitele“ varianta pouze provádí analýzu vstupních dat (shluková analýza), neuronová síť obsahuje jedinou vrstvu skrytých neuronů, které jsou uspořádány do mřížky, použit pro shlukování a klasifikace. Samoorganizující se neuronové sítě skládají se ze vstupů (jejich počet je záleží na složitosti úkolu), jednotlivými váhami (Obrázek 6). Tento nekontrolovaný model neuronové sítě koncentruje data do klastrů. Podstatou je snížit vstupní soubor mapováním do menšího počtu shluků. Tyto sítě se často používají k předprocesnímu zpracování dat pro jiný typ neuronové sítě. (17)

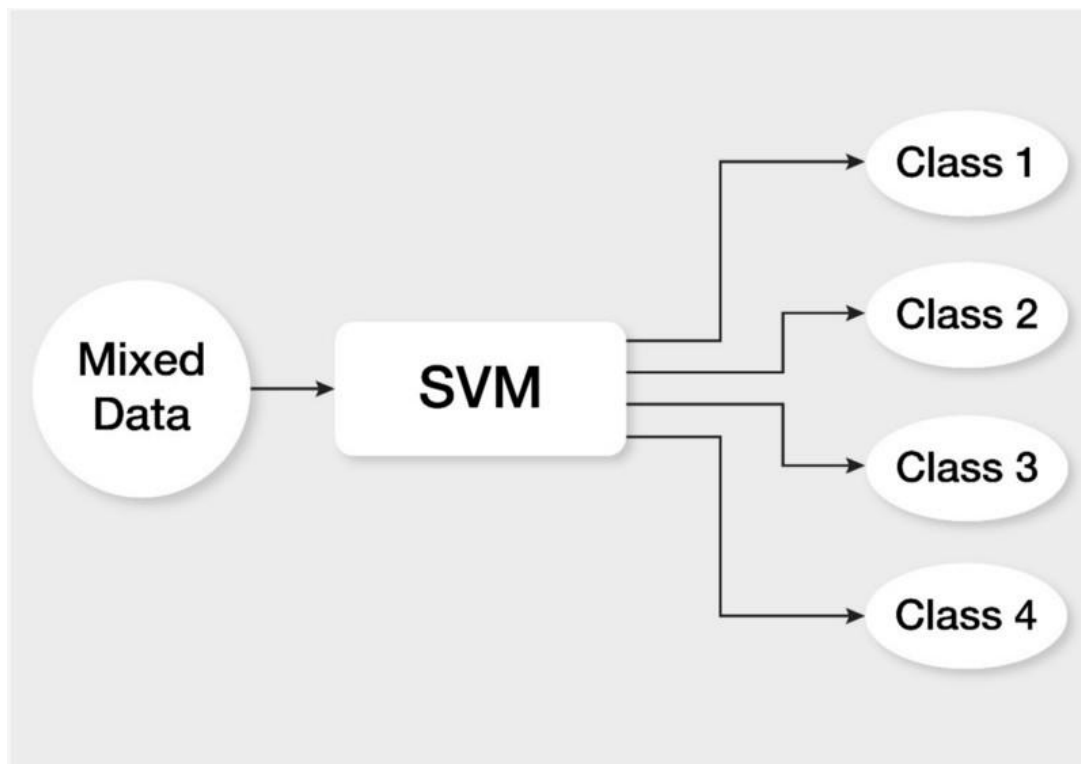


Obrázek 6. Model samoorganizující se sítě se dvěma vstupy. Převzato z:  
[https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=21471](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21471)

### 3.3.2.3 Podpůrné vektorové stroje

Podpůrné vektorové stroje (angl. Support vector machines, SVM) je model, který nabízí velmi progresivní a novou metodu strojového učení. Byl vyvinut na konci 20. a začátku 21. století V.Vapnikem a A.Červoněnkis. Tuto metodu lze použít především pro klasifikační úlohy, ale také najde aplikaci v regresním modelování a neparametrických odhadech hustoty. (18). Umí řešit lineární i nelineární problémy. Myšlenka SVM je jednoduchá:

Algoritmus vytvoří čáru (angl.hyperplane), která odděluje data do jednotlivých skupin.(Obrázek 7) (19)



Obrázek 7. Princip SVM. Převzato z: <https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989>

V praktické části této práci budou podrobněji probrány poslední dva z výše uvedených modelů strojového učení, tj. neuronové sítě a podpůrné vektorové stroje.

### 3.3.3 Strojové učení a MATLAB

MATLAB je integrované prostředí pro vědecké výpočty, modelování, vývoj algoritmů, analýzu a prezentaci dat, poskytuje funkce pro všechny hlavní úlohy strojového učení:

- **Klasifikace**

Cílem klasifikace je oddělit objekty do nespojitých tříd. Objekty jsou popsány hodnotami dat (vlastnosti objektu), které jsou vstupem do klasifikačního algoritmu, klasifikátoru. MATLAB poskytuje celou řadu klasifikačních algoritmů. Jedná se o rozhodovací stromy,

diskriminační analýzu, podpůrné vektorové stroje (SVM), Bayesův klasifikátor, neuronové sítě, fuzzy logika a další.

Práce s klasifikátory v MATLABu je velmi snadná, protože i když jsou tyto algoritmy velmi odlišné, MATLAB uděluje nad nimi jednotné rozmezí. Klasifikační model je vytvořen jako objekt v pracovním prostoru MATLAB. Pro následnou klasifikaci nových dat je použita funkce predikce bez ohledu na typ klasifikátoru. Samozřejmostí je možnost nastavení různých parametrů, které ovlivňují chování klasifikačních algoritmů.

Kromě toho, MATLAB poskytuje jasné grafické uživatelské rozhraní, jeho aplikace umožňuje vybrat data, vyberte klasifikátory, nastavit a naučit je. Dává také jasnou vizualizaci výsledků, včetně metrik pro porovnání každého klasifikátoru navzájem bez nutnosti programování. Výsledný klasifikátor pak lze snadno aplikovat na nová vstupní data.

- **Regrese**

Účelem regresní analýzy je zhodnocení kontinuální výstupní hodnoty na základě vstupních dat. MATLAB poskytuje algoritmy pro lineární regresi, nelineární regresi a generalizovanou lineární regresi (lineární regresní model, doplněný nelineární spojovací funkcí).

- **Shluková analýza**

Účelem klastrové analýzy je rozdělit data do skupin na základě podobných vlastností. MATLAB poskytuje funkce, jako je hierarchická klastrová analýza, algoritmus k-prostředků, směs Gaussových distribucí, samoorganizující se sítě, algoritmy fuzzy c-prostředků, skryté Markovovy modely a mnoho dalšího.

### 3.3.3.1 Učení a predikce

Strojové učení se obvykle skládají ze dvou fází (viz. Obrázek 8) – fáze učení, kde je určen výsledný klasifikační nebo regresní model, a fáze predikce, ve které je model používán pro získání nového výstupu z hodnocení nebo zařazování nových dat do správné kategorie.

Data se nejprve vhodně upraví a poté předají na vstup klasifikačního nebo regresního modelu. Pokud se data skládají z příliš velkého množství, můžete použít analýzu hlavních komponent (PCA) a převést vícerozměrná data do jednodušší podoby. Kromě toho není vždy nutné, aby model pracoval se všemi vzorky dat, stačí použít agregované statistiky za určité časové období nebo provést klastrovou analýzu. Vzhledem k tomu, že výsledný model je vyškolen v modifikovaných datech, z toho vyplývá, že stejná korekce dat by měla být provedena ve fázi predikce. Pokud musí být prognóza prováděna v reálném čase, musí být korekce dat dostatečně rychlá a tato skutečnost musí být zohledněna během fáze školení modelu. (12)

## FÁZE 1: UČENÍ



## FÁZE 2: PREDIKCE



Obrázek 8. Sled kroků strojového učení v prostředí MATLAB. Převzato z:

<https://sciencemag.cz/strojove-uceni-a-matlab>

## 3.4 Oblastí použití

### 3.4.1 Ekonomika o podnikání

Techniky AI se využívají v různých aspektech lidského života jak v technologických sektorech, tak i v ekonomických. Velký význam to má v průmyslových předpovědích výpadků produkce. Umělá inteligence umožňuje předvídat narušení výroby, zhoršení



kvality výrobků. Data mohou být analyzována pomocí počítačové techniky a umělé inteligence přímo na výrobních linkách, a tak předcházet chybám dříve, než k nim dojde. Takže se to dá využít pro zvýšení produktivity.

AI má umístění nejen v oblastech spojenou s robotikou. Pomocí ní je možné předpovědět poptávku po produktech v daném místě a čase a díky tomu vylepšit dodávky zákazníkům, naplánovat spotřeby financí, zásob, materiálů, pracovní síly atd. Pomocí senzorům jde najít vady v jednotlivých řetězcích a přijít na jejich původce.

Umělá inteligence se postupně objevuje na rostoucím počtu pracovišť zejména ve formě virtuálních osobních asistentů, které pomáhají zvyšovat produktivitu zaměstnanců v řadě podniků, zejména v anglicky mluvících zemích. Analytici společnosti Gartner předpovídají, že 70% organizací budou využívat nějakou formu virtuální pomoci ke zlepšení výkonnosti zaměstnanců v roce 2021. (20)

Nezáleží na tom, jestli pracujete v kanceláři za počítačem nebo někde na výrobní lince. Vždy můžete najít něco, co lze automatizovat. Jednou z nejrozzumnějších aplikací strojového učení je třídění naskenovaných dokumentů podle jejich obsahu. Umělá inteligence rozpoznává údaje každého dokumentu a podle toho je pojmenuje a přiřadí k nějaké skupině. (3)

### 3.4.2 Služby bezpečnosti

V posledních letech hrozně vzrostl počet IT (Information Technology) útoků a je jasné, že společnosti potřebují zlepšení kybernetického zabezpečení. Implementace AI softwaru a strojového učení dokáže identifikovat minulé události, odhalit anomálie, monitorovat neobvyklé síťové a uživatelské aktivity a odhadnout jaké z nich by mohly být potenciálně škodlivé. Podle průzkumu společnosti «Vanson Bourne» zveřejněného 11. října 2019 roku už 80 procent organizací používá nějakou formu umělé inteligence. Zejména firmy zabývající se finančními službami. Protože mají velmi cenná data, musí být v oblasti kybernetického zabezpečení nejlepší a mají peníze na zaplacení nových technologií. (21)

Ale nikdo si neosvojuje nové technologie strojového učení a AI tak rychle jako hackeři. Příslušný software je snadno dostupný a je jednodušší se s ním naučit pracovat, když je konkrétní cíl a cena. Poskytovatelé bezpečnostních služeb využívají strojové učení

k rozpoznávání špatného softwaru a domén. Jenže zločinci se snaží vypozerovat, co a jak bezpečnostní firmy dělají, a přizpůsobit svůj model strojového učení, tak aby vytvářel výjimky. Nejdůležitější věc, kterou nyní mohou firmy udělat, je zajistit si silné technické vybavení a pokračovat v simulacích útoků pro zlepšení předpovědí AI softwaru. (22)

Nejde jen o ochranu v kyberprostoru, ale také o ochranu lidských životů. Rostoucí investice do pokročilejších technologií, ať už kamer nebo systémů umělé inteligence, jsou nyní schopny varovat evropská a americká města o nadcházejících hrozbách. Dobře fungující systém nemůže najít útočníka pouze po spáchání trestného činu, ale může ho dokonce upozornit ještě před jeho spácháním. Jedná se především o aplikace, které rozpoznávají obličeje, ale také podle predikce činností založených na skutečných emocích vyjádřených výrazem na obličeji. To umožňuje systému včas varovat bezpečnostní síly, aby minimalizovaly škody. (3)

### 3.4.3 Zdravotnictví

Ačkoli zdravotní a zdravotní péče obecně se neustále vyvíjí, stále se objevují nová a nová onemocnění, kterým musí lidstvo zabraňovat. Vždy najdeme nemoc, kterou nemůžeme vyléčit. Jedním z největších obtíží při léčbě závažných onemocnění je pozdní diagnóza, která je někdy způsobena dřívějšími chybami při určování choroby. To však může provést pouze počítač. Příkladem je čínský robot «Xiaoyi», který absolvoval lékařský průkaz způsobilosti. Největší úspěchy však lze očekávat od spojení umělého intelektu a člověka. «BenevolentAI» pracuje na vývoji nových léčiv tímto způsobem. (3)

Technologie AI mohou být použity v diagnostice, léčbě a předvídání výsledků. Umělá neuronová síť byla nejpoužívanějším analytickým nástrojem, zatímco jiné umělé inteligentní techniky jako fuzzy expertní systémy, evoluční výpočty a hybridní inteligentní systémy byly použity v různých klinických podmínkách. Techniky umělé inteligence mají potenciál uplatnit se v téměř každé oblasti medicíny. Existuje potřeba dalších klinických studií, které by byly vhodně navrženy před tím, než tyto vznikající techniky naleznou uplatnění v reálném klinickém prostředí. (23)

### 3.4.4 Počítačové hry

Umělá inteligence dnes ovládá většinu her: od deskových her po interaktivní hrané hry. Poskytují komplexní a rozhodovací prostředí pro experimentování s AI. Schopnost her

poskytovat zajímavé a složité problémy a kreativitu učinila jednu z nejoblíbenějších a nejvýznamnějších oblastí pro výzkumné pracovníky AI.

Hry nabízejí jednu z nejvýznamnějších oblastí, které mohou zpracovávat, interpretovat a stimulovat lidské chování. Obousměrný vztah her a AI vyšel na novou cestu a lze říci, že herní průmysl do značné míry revolucionizuje způsob, jakým AI funguje:

- **Interakce:**

Interakce hráče s hrou je výhodná pro AI, protože dává přístup k algoritmu ke studiu hráče, který zažívá emocionální chování. Studium hry a lidské interakce je klíčem nejen ke studiu lidského chování, ale také k tomu, aby AI vytvořila lepší systém interakce člověk-počítač. Dále posouvá hranice AI za účelem studia a pochopení systémů interakce člověk-počítač a řešení problémů, kterým čelí její aplikace v reálném světě.

- **Rozhodování:**

To je hlavní podstata AI. AI musí být schopna činit rozhodnutí při pohledu na soupeřovu akci. Existují různé modely používané pro rozhodování ve hře. «Fine State Machine» (FSM) je jednou z mnoha metod AI používaných při rozhodování.

- **Predikce:**

Predikce zahrnuje předvídaní dalšího pohybu hráče, takže rozhodování může být prováděno na jeho základě. To se provádí pomocí metod, jako je rozpoznávání minulých událostí a náhodný odhad. Umělé neuronové sítě (ANN), podpůrné vektorové stroje (SVM) a rozhodovací stromové učení jsou algoritmy používané pro predikci. Regresní algoritmy se používají pro předvídaní chování hráče. Tento proces zahrnuje situace, jako je předpověď doby, kdy se očekává, že hráč bude na určité úrovni hry, jakou položku bude hráč vybírat dále, kde se bude pohybovat.

- **Inteligence:**

Sociální inteligence a interakce člověk-počítač jsou nejvyšším cílem AI. Tyto dvě věci existují ve hrách, a tak pomáhají při rozvoji AI. Virtuální postavy vykazující lidské

chování i inteligenci. AI se dozvěděla o inteligenci počítačů nejvíce z her než z jiných aplikací, protože poskytují virtuální platformu pro testování všech druhů algoritmů. Navíc poskytují komplikované matematické problémy, se kterými se lze vypořádat, takže učení AI není omezeno pouze na herní svět. Studium inteligence ve hrách nám nejen umožňuje vědět více o lidské inteligenci, ale také o inteligenci AI.

Hry nabízejí jak zábavu, tak interakci, množství fantazijních funkcí najednou: vizuální umění, zvukový design, grafický design, všechno to je v jediném softwaru. Hry jsou dokonalou zkouškou pro AI, protože fungují jako nejlepší aplikace počítačové kreativity. Jako výsledek, s použitím výpočetní kreativity v herním průmyslu, poskytuje způsob, jak postupovat AI. (24)

### 3.4.5 Počítačové vidění

Počítačové vidění lze definovat jako „teorie a technologie pro vytváření umělých systémů, které získávají informace z obrazů nebo jiných multimediálních dat“.

Přístup pro provádění rozpoznávání objektů lze rozdělit do třech skupin: klasifikace, identifikace a detekce:

- Při klasifikaci objektů máme několik dříve naučených objektů, které chceme rozpoznat v obraze. Například počítač klasifikuje fotografii jako portrét osoby, protože tato fotografie obsahuje obličej.
- Identifikace objektu je rozpoznání konkrétního objektu. Například, stroj identifikuje, že na obrázku jsou dva obličeje a že jeden z nich je mužský a druhý je ženský.
- Detekce objektů je schopnost rozpoznat, že je objekt v obraze. To se obvykle používá u zproplatněných silnic, kdy stroj automaticky rozpoznává dálniční známku.

Kombinace počítačového vidění se strojovým učením přinesla určité úžasné výsledky. Facebook například kombinoval počítačové vidění, strojové učení a jejich masivní sbírku fotografií, aby získal přesné výsledky v rozpoznávání obličeje. Algoritmus počítačového vidění se může naučit identifikovat obličej osoby na fotografii. Facebook může také identifikovat objekty na obrázcích pomocí stejného procesu. (25)

Téměř nepoznatelná pomoc umělé inteligenci je v nejnovějších řadách smartphonů. Jsou vybaveny AI, abychom mohli udělat kvalitní fotografie. Umělá inteligence posoudí, zda existují dostatečné světelné podmínky, aby se dosáhlo dokonalého výsledku, a pak buď

ihned pořídí fotografii nebo nejprve upraví nastavení fotoaparátu. Jedná se o poměrně jednoduchý upgrade stávajících smartphonů, ale má významný dopad na výsledné obrazy.

(3)

## 4 Praktická část

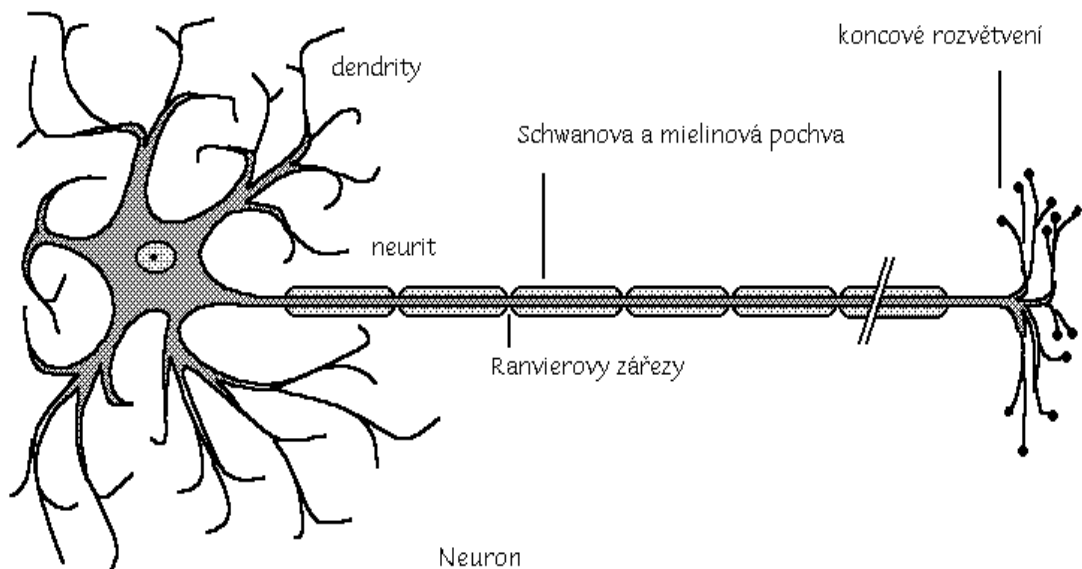
### 4.1 Neuronové sítě

#### 4.1.1 Model neuronových sítí

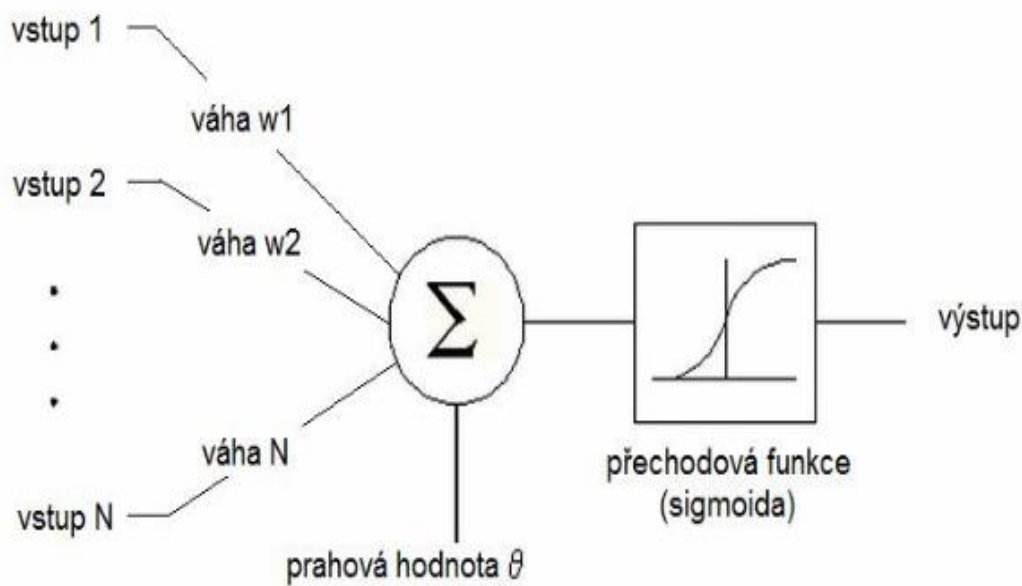
Umělá neuronová síť je složena z mnoha jednoduchých procesorů - perceptronů.

##### 4.1.1.1 Perceptron

Perceptron je základní stavební jednotka neuronové sítě, z toho vychází např. označení vícevrstvé perceptronové sítě. V dnešní době však již dochází k záměně a používá se pro něj spíše označení "neuron", protože perceptron je matematický model biologického neuronu. (Obrázek 9)



Obrázek 9. Schéma biologického neuronu. Převzato z:  
<https://cgg.mff.cuni.cz/~pepca/prg022/mucha/>



Obrázek 10. Schéma perceptronu. Převzato z:

[https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=21471](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21471)

Existuje pět hlavních složek umělého neuronu (Obrázek 10). Zleva doprava jsou to:

**Vstupní uzly.** Každý vstupní uzel je spojen s číselnou hodnotou, kterou může být libovolné reálné číslo. Příkladem může být jedna pixelová hodnota obrázku. Vstupy jsou jako podněty z vnějšího prostředí, nebo jako výstupy z jiných neuronů u vícevrstevných sítí.

**Připojení.** Každé spojení má váhu ( $w$ ), která je s ním spojena, a to může být libovolné reálné číslo. ANN běží a šíří milionkrát, aby optimalizovala tyto hodnoty „ $w$ “.

**Prahová hodnota.** Při překonání této hodnoty perceptron vede signál k výstupu ve formě **přechodové(aktivační) funkce** stejně jako potenciál u biologického neuronu.

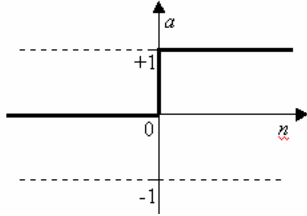
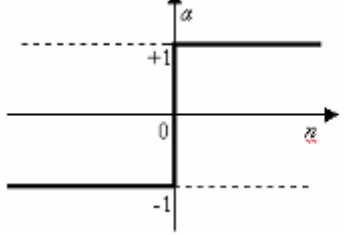
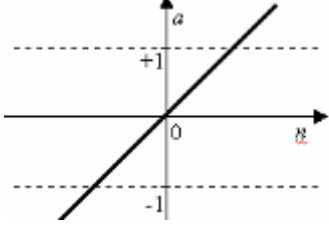
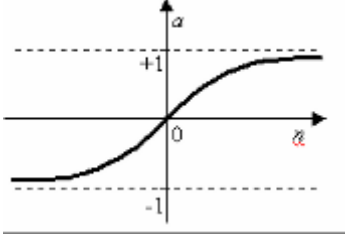
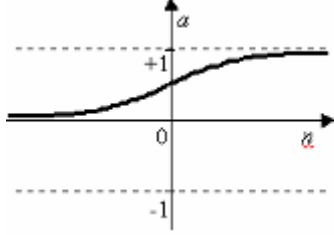
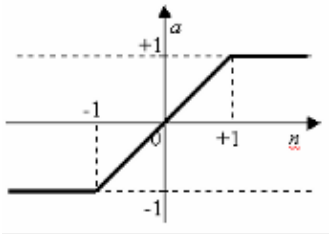
**Výstupní uzel,** který je spojen s funkcí váženého součtu vstupních uzlů.

(13) (17)

#### 4.1.1.2 Aktivační funkce

Pro neuronové sítě a jejich praktické nasazení se používá velké množství aktivačních funkcí. Aktivační funkce excitovaného neuronu moduluje jeho výstupní signál, který je

směřován na úroveň neuronové sítě. Výběr vhodné aktivační funkce neuronu ovlivňuje konvergenci výpočtu neuronové sítě. Nejpoužívanějších přechodové funkce (Tabulka 1):

Funkce jednotkového skoku ( <i>hardlim</i> ).	
Aktivační funkce signum ( <i>hardlims</i> ).	
Lineární aktivační funkce ( <i>purelin</i> ).	
Aktivační funkce hyperbolický tangens ( <i>tansig</i> ).	
Sigmoidální aktivační funkce ( <i>logsig</i> )	
Saturační přenosová funkce ( <i>satlins</i> ).	

Tabulka 1. Nejběžnější aktivační funkce neuronových sítí. Převzato z:  
[https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=21471](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21471)



#### 4.1.2 Učení neuronové sítě

Cílem výuky neuronové sítě je nastavit váhy modelu neuronové sítě tak, aby se získala správná odezva výstupního signálu na daný vstupní signál. Jak bylo uvedeno v teoretické části, používají se dvě strategie učení neuronové sítě: učení se učitelem a učení bez učitele. Výuka s učitelem byla použita v mé vlastní práci. Měřítko sítě je nastaveno tak, že výstupní signál je konzistentní, to znamená, že síť poskytuje stejnou odezvu na budící signál se stejnými nebo podobnými vstupními vektory. Někdy je vstup do neuronové sítě označován jako vstupní vektor. V tomto případě je vstupním vektorem jeden soubor obsahující hodnoty jednotlivých indikátorů stavu environmentálního modelu.

Při práci s neuronovými sítěmi je nutné rozlišovat mezi údaji o školení a ověřování a zkušebními údaji. Na začátku učení jsou váhy obvykle zadávány náhodným číslem generovaným v určitém intervalu, nebo je hodnota přiřazena všem hodnotám. Proces učení se snaží minimalizovat síťovou chybu (rozdíl mezi výstupními požadovanými hodnotami od těch, které jsou právě přijímány). Hodnota chybové funkce by se měla u funkce učení snížit. Každá neuronová síť má svůj vlastní algoritmus učení.

Před učením sítě (s učitelem) je nutné zvolit školící data, které jsou dostatečně reprezentativní pro jednotlivé stavy úlohy, kterou by měla síť provádět v rámci výpočetního procesu. Pokud během ladění sítě nastane situace, kdy síť nemá dostatečnou klasifikační schopnost, pak může přistoupit k přidání neuronu do skryté vrstvy. V procesu učení se hodnota chybové funkce snižuje, až se blíží nule. Je-li chybový stav neuronové sítě nulový, jedná se o lineární klasifikátor, což je nežádoucí stav, protože v tomto případě síť nemá obecnou možnost zobecnění a nemůže dále činit rozhodnutí týkající se jiných dat než těch, na kterých se naučila. (17)

##### 4.1.2.1 Informace o sadě dat

Datový soubor dálkového průzkumu s vysokým rozlišením (Quickbird). Tento soubor dat obsahuje některé údaje o tréninku a testování ze studie dálkového průzkumu společnosti Johnson et al. (2013), které se týkaly odhalování nemocných stromů. Existuje jen málo tréninkových vzorků pro třídu „nemocných stromů“ (74) a mnoho pro třídu „zdravé stromy“ (4265). Datový soubor se skládá ze segmentů obrazů, které obsahují spektrální

informace z multispektrálních obrazových pásem Quickbird a texturní informace z obrazové oblasti Panchromatic (Pan).

**Soubory:**

training.csv: sada dat pro učení (4339 obrazových segmentů)

testing.csv: sada testovacích dat (500 obrazových segmentů)

**Informace o parametrech:**

Column\_1: '1' (nemocné stromy), '0' (zdravé stromy)

Column\_2: Průměrná textura GLCM (Pan Pan)

Column\_3: Průměrná zelená hodnota

Column\_4: Střední červená hodnota

Column\_5: Střední hodnota NIR

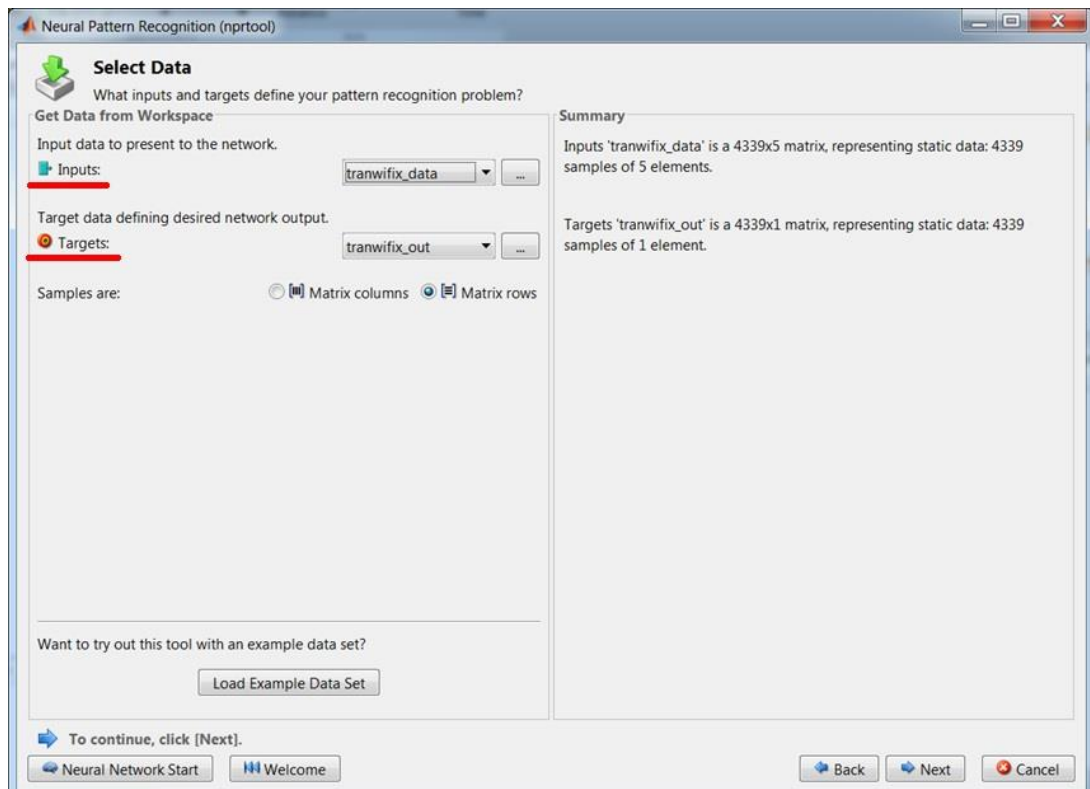
Column\_6: Standardní odchylka (Pan)

Pro učení neuronové sítě bylo náhodně vybráno pouze 200 segmentů.

(26)

#### 4.1.2.2 Vložení dat do neuronové sítě:

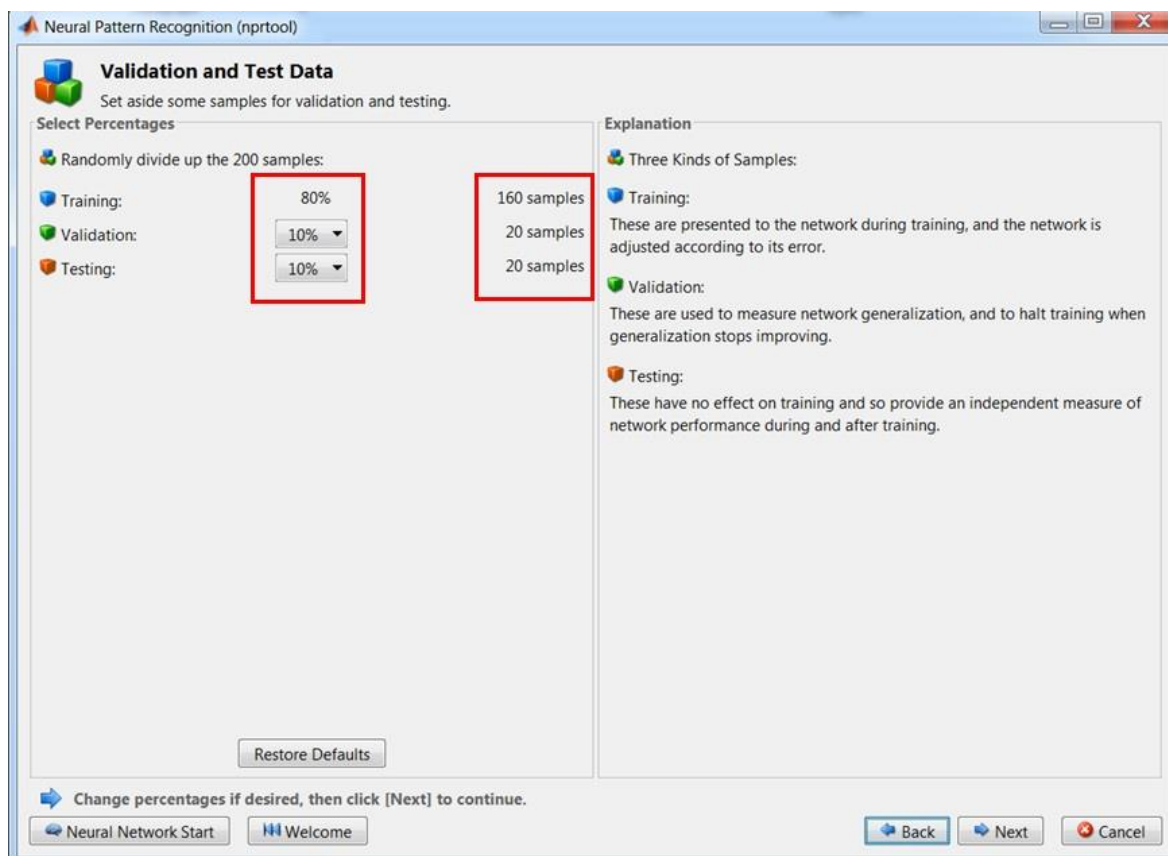
Vstupní data zadávají ve 2 souborech. *Inputs* jsou data (5 sloupců), se kterými bude pracovat neuronová síť. *Targets* jsou data, se kterými bude neuronová síť porovnávat svá rozhodnutí a následně se měnit, aby získala požadované cílové hodnoty (viz. Obrázek 11).



Obrázek 11. Vložení dat do neuronové sítě

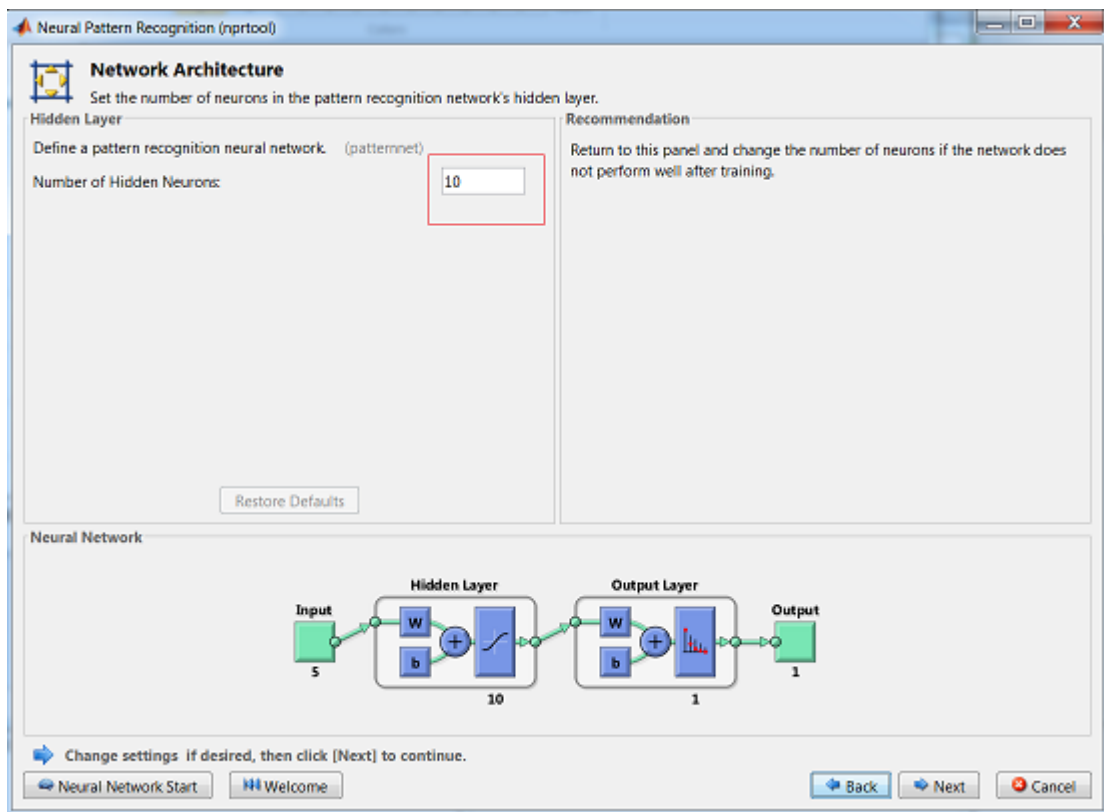
#### 4.1.2.3 Proces učení

- 1) Vybrat množství dat pro školení (*training*), kontrolu (*validation*) a testování (*testing*), (Obrázek 12). Standardní hodnoty jsou 70% 15% 15%. Pro daný soubor bylo mírně zvýšeno množství dat pro školení do 80% (160 vzorků) a sníženo zbytek do 10% (20 vzorků). To nebude mít vliv na správnost řešení, ale grafiky budou informativnější.



Obrázek 12. Výběr množství dat pro školení, kontrolu a testování

- 2) Dále lze vybrat počet skrytých vrstev neuronové sítě (Obrázek 13). To ovlivňuje efektivitu analýzy dat, ale zvyšuje dobu zpracování. Pro optimální poměr rychlosti a kvality zpracování doporučuji použít standardních 10 skrytých vrstev. *Input* – počet vstupních parametrů.

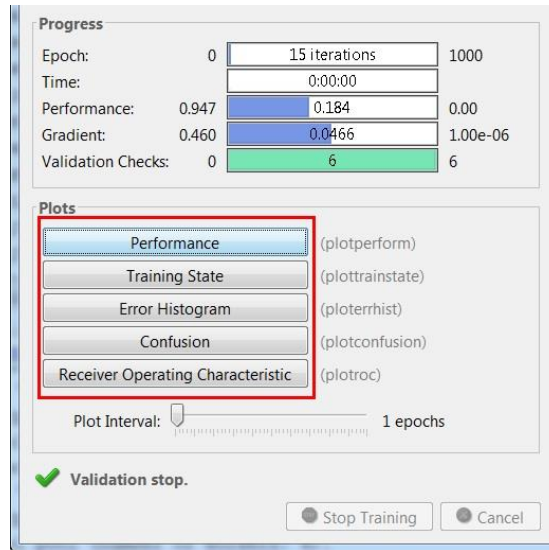


Obrázek 13. Výběr počtů vrstev neuronové sítě

3) Startovat učení. Při využití malého souboru dat učení probíhá velmi rychle.

#### 4.1.2.4 Analýza

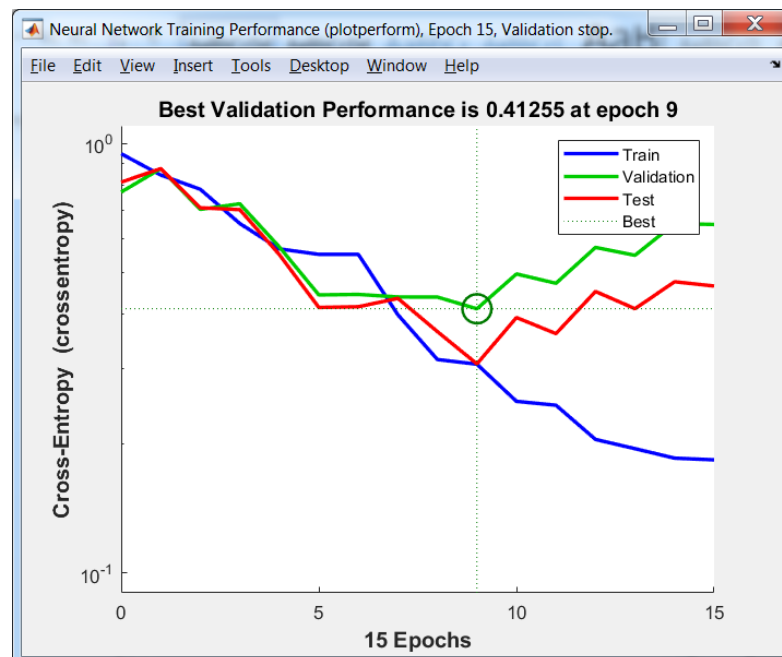
Po učení neuronové sítě budou k dispozici výsledky v různých formách, které je možné podrobněji prohlédnout (Obrázek 14):



Obrázek 14. Souhrnná analýza procesu učení neuronové sítě

- Performance

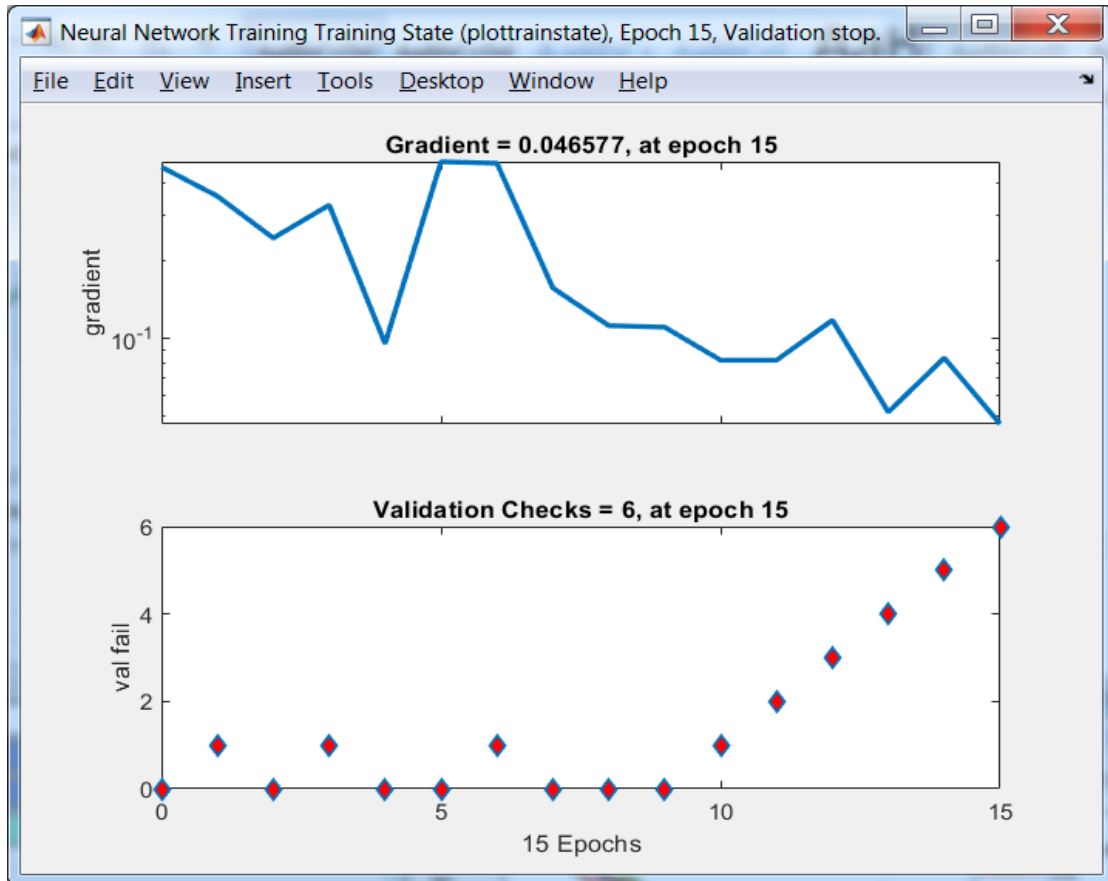
Vlastnost *the best epoch* (Obrázek 15) označuje iteraci č.9, při které výkon skenování dosáhl svého maxima. Trénink pokračoval dalších 6 iterací, než se zastavil. Toto číslo neznamená žádné závažné problémy s učením. Křivky validace a testování jsou velmi podobné.



Obrázek 15. Graf výkonosti učení

- Training state

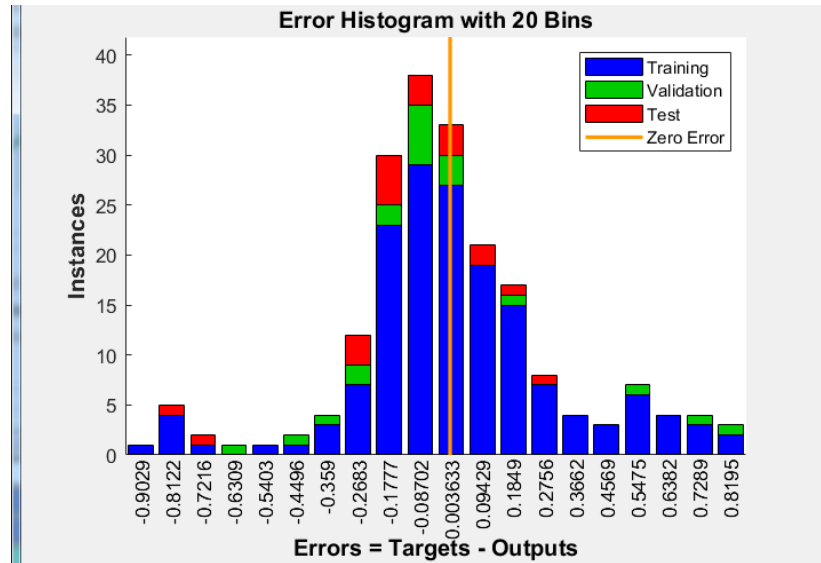
Ukazuje počet chyb při kontrole (Obrázek 16).



Obrázek 16. Grafy počtů chyb při kontrole

- Error histogram

Ukazuje, v jakém rozmezí bylo uděláno nejvíc chyb. Chyby = vstupní data – výstupní data (viz. Obrázek 17).



Obrázek 17. Histogram chyb

- Confusion matrix



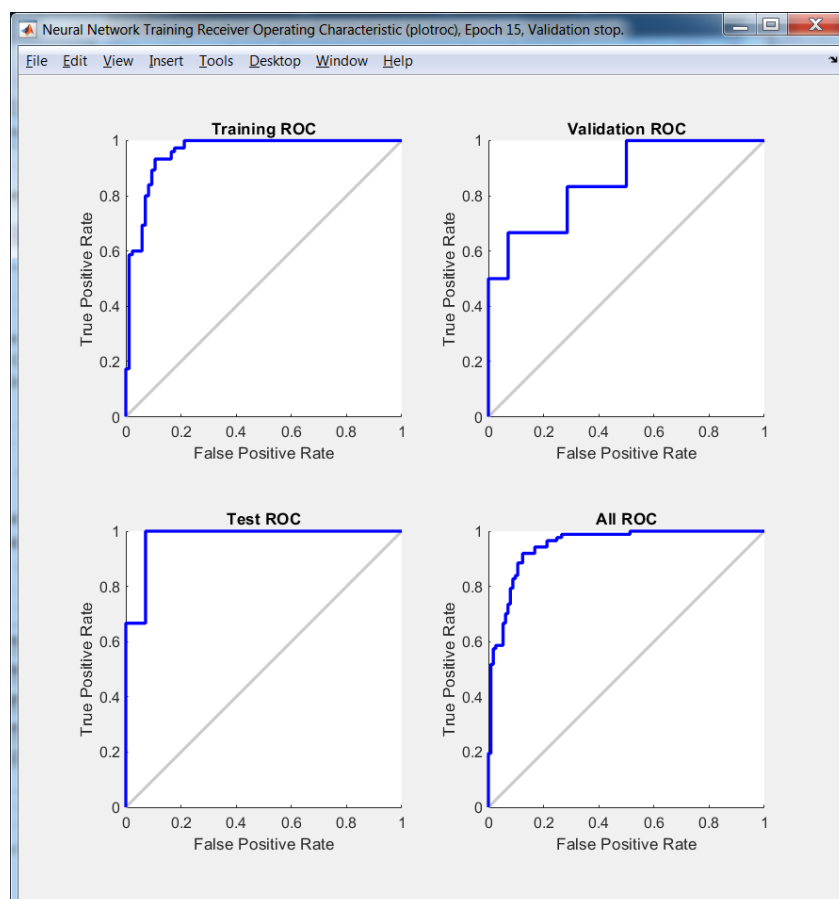
Obrázek 18. Matice zmatků pro neuronovou síť (200)



Matice zmatků (Obrázek 18) zobrazuje, kolik dat bylo správně (zelené buňky) nebo špatně (červené buňky) klasifikováno. 0 - zdravé stromy, 1 – nemocné stromy. Me zvyrazněným čtverci je ukázána procentuální přesnost této neuronové sítě – 86,0%.

- Plot ROC

Křivky ROC (receiver operating characteristic) – grafy (Obrázek 19), který umožňuje vyhodnotit kvalitu binárního členění, zobrazuje poměr mezi podílem objektů z celkového počtu nosných znaku, správně klasifikovaného jako nesoucí znak (angl. *true positive rate*, tzv. citlivost algoritmu klasifikace) a podíl objektů v celkovém počtu objektů, které nesou znak, chybně klasifikovaný jako nesoucí znak (*false positive rate*, tzv. specifická klasifikačního algoritmu).



Obrázek 19. Grafy křivek ROC pro neuronovou síť

Na konci učení lze vytvořit funkce pro vyškolenou síť a vyrobit pro ně skripta pro testování jiných dat ve stejném formátu.

#### 4.1.2.5 Testování neuronové sítě

Pro testování neuronové sítě byla použita náhodná data (Tabulka 2), kde ve druhém sloupci jsou představeny skutečné výsledky, kterých by síť musela dosáhnout, 1 – nemocné stromy, 0 – zdravé stromy.

1.	1	124.591944	179.4166667	89.08333333	208.0833333	20.35260168
2.	1	135.0244845	190.877551	104.5510204	270.6530612	14.32993157
3.	1	142.5657895	183.4166667	89.76388889	240.2638889	11.2986602
4.	0	140.8571429	175.75	79.91666667	293	6.015604708
5.	0	132.5920074	164.6829268	68.68292683	177.897561	8.123651767
6.	0	138.2404181	198.25	87.05555556	445.3888889	27.26414006

Tabulka 2. Zkušební data pro testování neuronové sítě na 200 množstvích dat

Výsledky testování jsou představeny na obrázku 20 pro 1 a na obrázku 21 pro 0. Je jasně vidět, že neuronová síť neposkytuje přesnou hodnotu, ale pouze číslo, které je blíž k 0 nebo 1. Číslo, které je okolo 0,5 udává náhodnost řešení, což je chyba.

```
>>1sim(net200e, [124.591944000000;179.416666700000;89.0833333300000;208.083333300000;20.3526016800000])
ans =
    0.4082
>>2sim(net200e, [135.024484500000;190.877551000000;104.551020400000;270.653061200000;14.3299315700000])
ans =
    0.9648
>>3sim(net200e, [142.565789500000;183.416666700000;89.7638888900000;240.263888900000;11.2986602000000])
ans =
    0.8091
```

Obrázek 20. Výsledky testování neuronové sítě (200) pro hodnotu 1

```

>>4sim(net200e, [140.857142900000;175.750000000000;79.9166666700000;293;6.01560470800000])
ans =
    0.2873
>>5sim(net200e, [132.592007400000;164.682926800000;68.6829268300000;177.897561000000;8.12365176700000])
ans =
    0.2713
>>6sim(net200e, [138.240418100000;198.250000000000;87.0555555600000;445.388888900000;27.2641400600000])
ans =
    0.0904

```

Obrázek 21. Výsledky testování neuronové sítě (200) pro hodnotu 0

## 4.2 Podpůrné vektorové stroje

Modely SVM využívají teorii Vapnik-Chervonenkisovy (VC) dimenze modelu s pravděpodobností

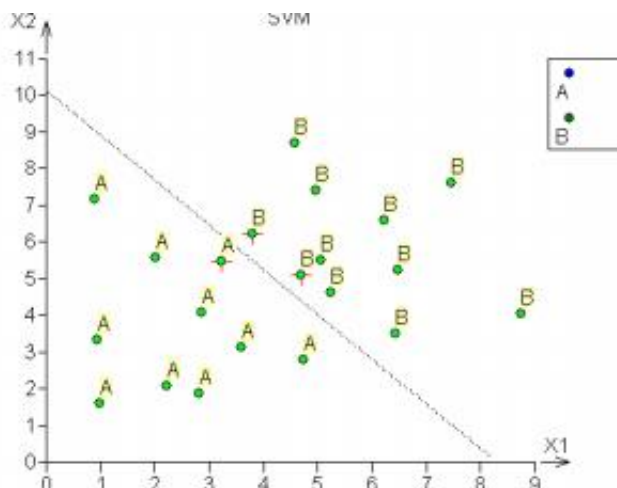
$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(\ln(2l/h) + 1) - \ln(\eta/4)}{l}}, \quad (1)$$

Kde  $R(a) \int \frac{1}{2} |y - f(x, a)| p(x, y) dx dy$  je risk (skutečná střední chyba modelu),  $l$  je počet parametrů modelu  $\alpha$  dat,  $R_{emp}(a) = \frac{1}{2} \sum_{j=1}^l |y_j - f(x, a)|$  je empirický risk a  $h$  je nezáporná celočíselná dimenze modelu. (18)

### 4.2.1 Klasifikační model SVM

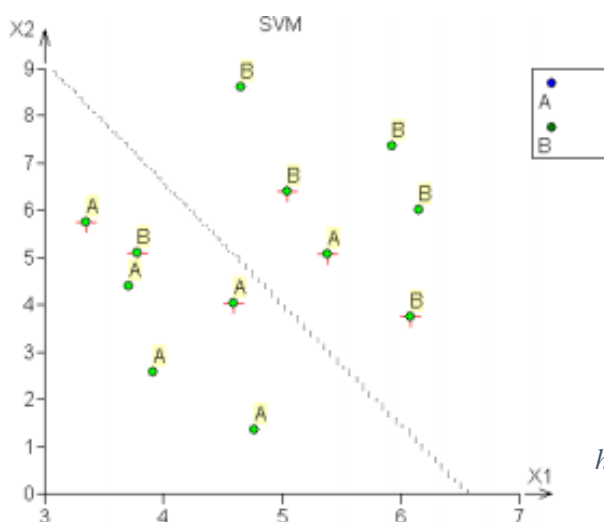
Modely SVM minimalizují dobře definovanou chybu (například nesprávnou klasifikaci pro klasifikační modely, chybu v některých regresních metrikách). Například, v lineární separabilní klasifikaci ve dvou rozměrech (dvě číselné nezávislé proměnné), je hledána přímka, která odděluje obě třídy, přičemž se udržuje maximální vzdálenost od experimentálních dat, čímž se minimalizuje riziko nesprávné klasifikace při předpovídání

nových dat (viz. Obrázek 22). Model SVM nám mimo jiné může pomoci vyhodnotit (včetně pravděpodobností), do které třídy objekt patří, na základě nových hodnot nezávislé proměnné.



Obrázek 22. Separabilní lineární model SVM. Převzato z:  
<http://www.trilobyte.cz/downloadfree/qcemanual/svm.pdf>

V neseparabilním případě se hledá přímka, která minimalizuje "přebytek" nesprávně klasifikovaných údajů. Příklad je znázorněn na obrázku 23, kde jsou minimalizovány kolmé vzdálenosti jednoho nesprávně klasifikovaného bodu B a jednoho nesprávně klasifikovaného bodu A od oddělovací čáry a je vyžadováno maximální oddělení ostatních dat. To poskytuje vysoce účinný klasifikační model. (18)



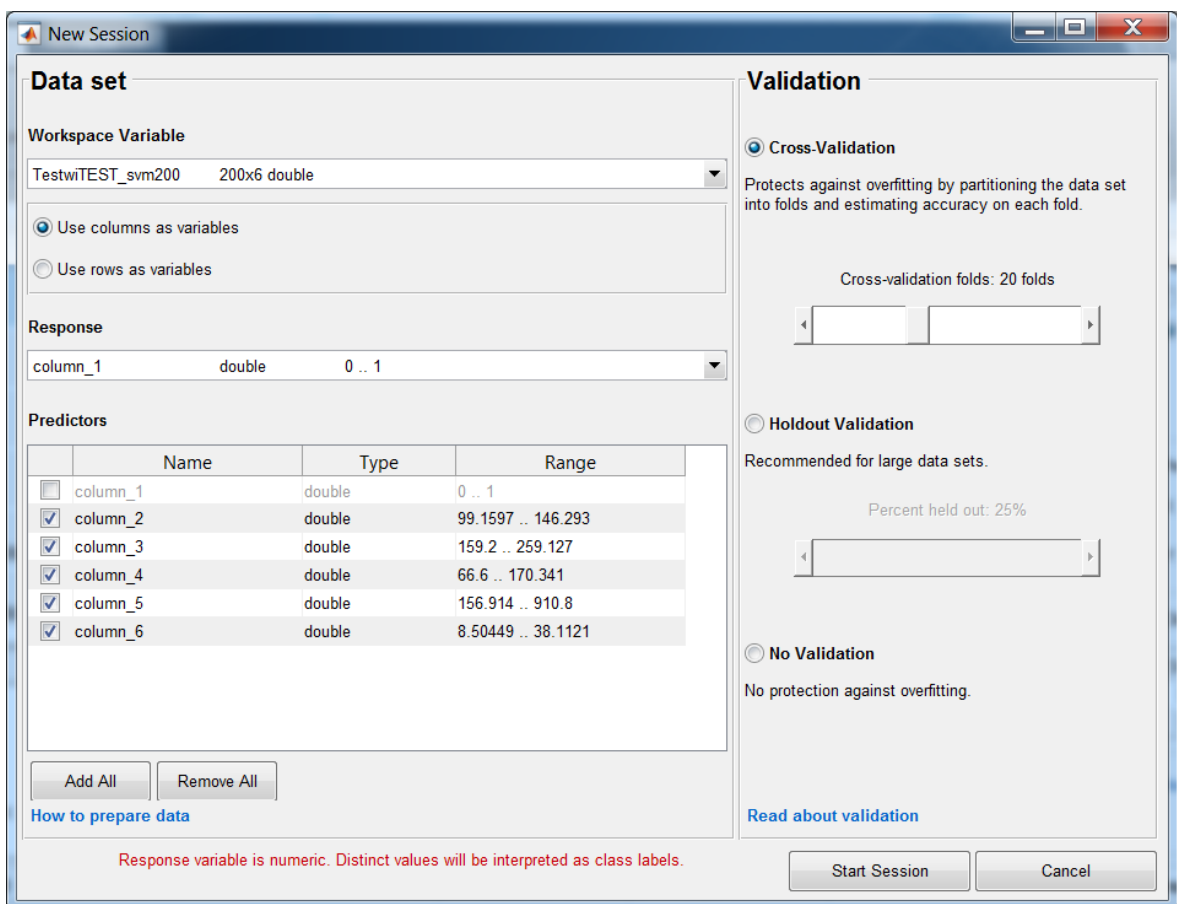
Obrázek 23. Lineární neseparabilní model SVM. Převzato z:  
<http://www.trilobyte.cz/downloadfree/qcemanual/svm.pdf>

## 4.2.2 Učení SVM

Pro klasifikaci SVM obecně se používají grafické objekty. Pro práci potřebujeme stáhnout jeden soubor, který obsahuje všechny datové vlastnosti pro každou fotografii a výsledek, který musíme získat.

Po importu souboru je nutné zvolit (viz. Obrázek 24):

- 1) Který sloupec je výsledkem (*response*), v daném případě *column\_1*.
- 2) Které prediktory (*predictors*) budou používány pro ověřování a testování, v daném případě všechny pět: *column\_2 – column\_6*
- 3) Kontrolní (*validation*) metodu, která bude zkoumat prediktivní přesnost použitých modelů. Pro daný soubor bylo vybráno křížové ověření s 20 záhyby.



Obrázek 24. Pracovní prostor pro výběr prediktorů a metody kontroly

#### 4.2.2.1 Kontrola modelů SVM

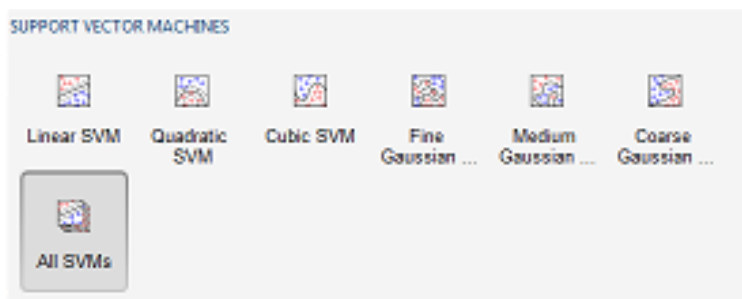
Kontrola (*validation*) odhaduje výkon modelu na nových datech ve srovnání s údaji o školení a pomáhá vybrat nejlepší model, chrání před přeplněním.

Cross-Validation (křížová kontrola): počet zvolených záhybů(folds) odpovídá počtu disjunkčních množin, do kterých se data rozdělují. Tato metoda vypočítá průměrnou chybu ve všech záhybech, poskytuje dobrý odhad prediktivní přesnosti konečného modelu vyškoleného se všemi daty, je doporučeno pro malé datové sady.

Validace Holdout: Je nutné vybrat procento dat, které je zapotřebí použít jako testovací sadu. Aplikace trénuje model na tréninkové sadě a hodnotí jeho výkon s testovací sadou. Model používaný pro validaci je založen pouze na části dat, takže doporučuje pouze pro velké datové sady. Konečný model je proškolen s kompletním souborem dat.

No validation: Žádná ochrana proti přeplnění. Aplikace používá všechna data pro školení a počítá chybovost na stejných datech. Bez testovacích dat získá nerealistický odhad výkonnosti modelu na nových datech. To znamená, že přesnost tréninkového vzorku bude pravděpodobně nereálně vysoká a prediktivní přesnost bude pravděpodobně nižší.

Dále budou provedeny testy přesnosti na všech typech SVM – *All SVMs* (Obrázek 25):



Obrázek 25. Výběr typů SVM, na kterých bude probíhat kontrola modelu

Po ukončení školení je třeba zvolit nejpřesnější model. Na obrázku 26 je vidět, že model *Linear SVM* vykazuje 91,5% přesnost a proto byla zvolena pro další testy.

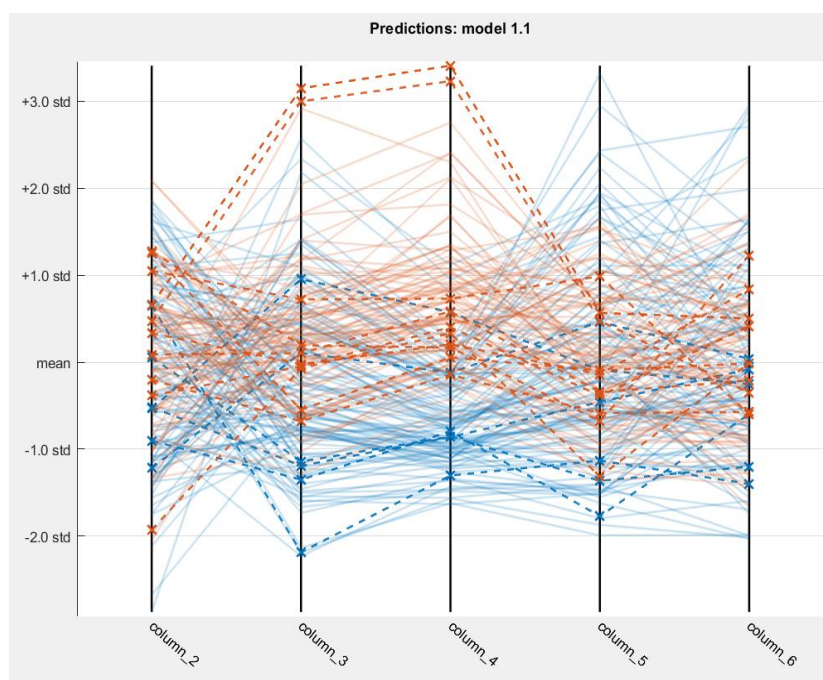
<b>1.1</b> ☆ SVM	Accuracy: <b>91.5%</b>
Last change: Linear SVM	5/5 features
<b>1.2</b> ☆ SVM	Accuracy: 91.0%
Last change: Quadratic SVM	5/5 features
<b>1.3</b> ☆ SVM	Accuracy: 87.0%
Last change: Cubic SVM	5/5 features
<b>1.4</b> ☆ SVM	Accuracy: 78.0%
Last change: Fine Gaussian SVM	5/5 features
<b>1.5</b> ☆ SVM	Accuracy: 90.5%
Last change: Medium Gaussian SVM	5/5 features
<b>1.6</b> ☆ SVM	Accuracy: 86.5%
Last change: Coarse Gaussian SVM	5/5 features

Obrázek 26. Výsledky validačního testu

#### 4.2.2.2 Analýza dat

- Graf paralelních souřadnic

Pro zkoumání objektů pro zahrnutí nebo vyloučení byl použit graf paralelních souřadnic (viz. Obrázek 27), který vizualizuje data o vysokém rozměru na jediném grafu. Tento graf může pomoci pochopit vztahy mezi objekty a identifikovat užitečné prediktory pro separaci, může vizualizovat tréninková data a nesprávně klasifikované body. Při vytváření výsledků klasifikátoru nesprávně klasifikované body ukazují přerušované čáry. Standardizace zobrazuje průměrnou hodnotu každého prediktoru na nulu a měřítko prediktorů podle jejich standardních odchylek.

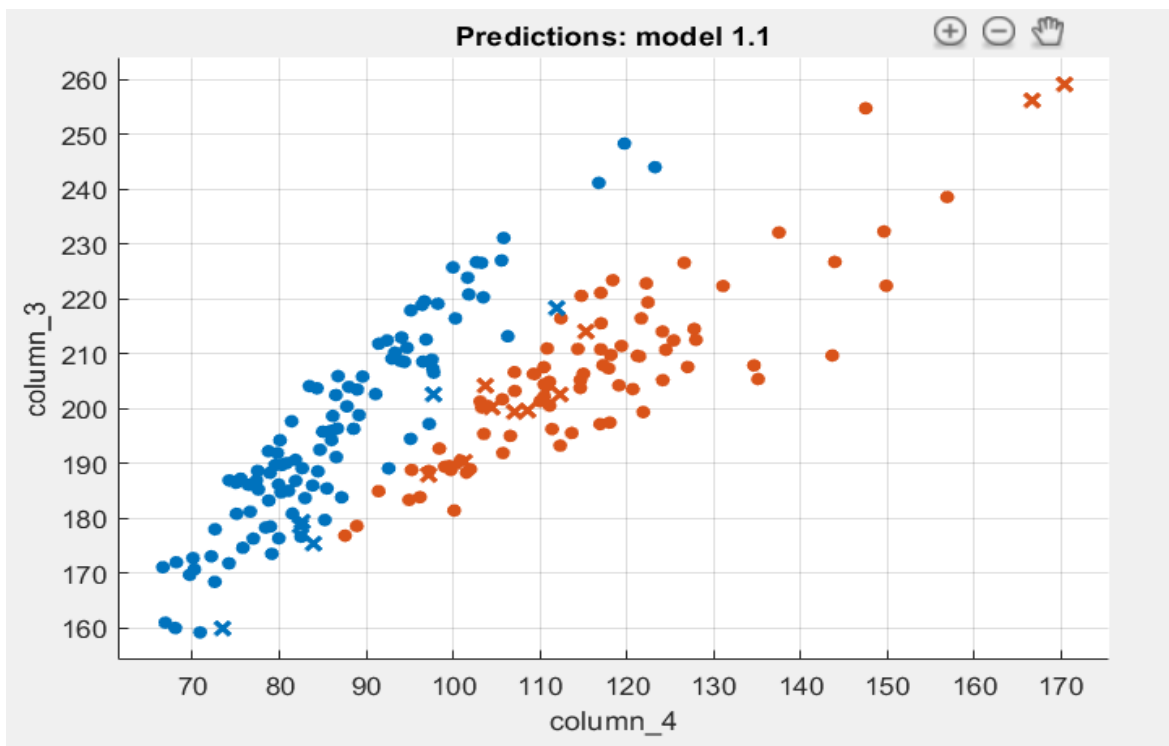


Obrázek 27. Standardizovaný graf paralelních souřadnic

- Bodový diagram

Identifikuje prediktory, které dobře oddělují třídy vynesím různých párů prediktorů na bodový graf (Obrázek 28). Graf může pomoci prozkoumat funkce, které je potřeba zahrnout nebo vyloučit. V tomto grafu vyneseny hodnoty 3. a 4. sloupce, je jasné vidět, jak jsou klasifikovány nemocné (červené tečky) a zdravé (modré tečky) stromy. Vidíme velkou akumulaci hodnot nemocných stromů v jedné části grafu. A můžeme taky pozorovat neúspěšné identifikace stromů(křížky).

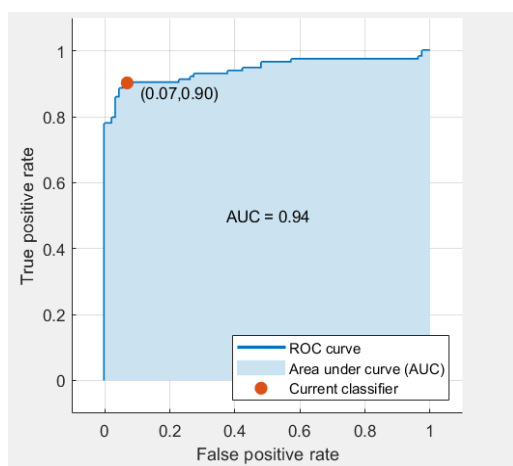




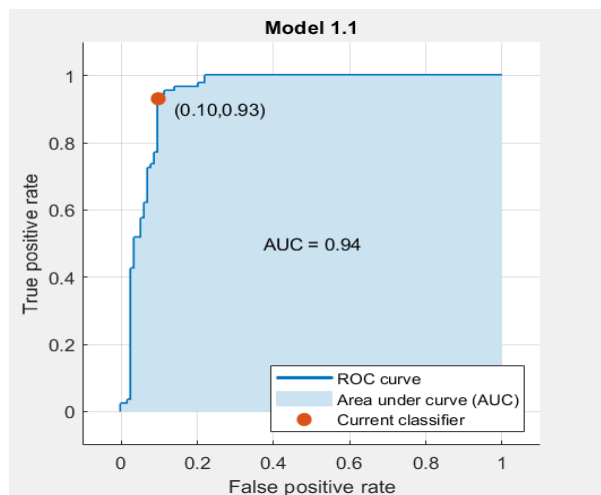
Obrázek 28. Bodový diagram prediktorů.

- Plot ROC curve

Kvantitativní interpretace ROC je dána ukazatelem AUC (angl. *area under ROC curve*, plocha pod křivkou ROC) je plocha ohraničená křivkou ROC a osou zlomku falešně pozitivních klasifikací. Čím vyšší je AUC, tím lepší je klasifikátor, hodnota 0,5 ukazuje nevhodnost zvolené klasifikační metody. (Obrázky 29 a 30).



Obrázek 29. Křivka ROC pro 0 - zdravé stromy



Obrázek 30. Křivka ROC pro 1 - nemocné stromy

- Confusion Matrix (matice zmatků)

Na grafu (Obrázek 31) řádky ukazují skutečnou třídu a sloupce zobrazují předpokládanou třídu. Diagonální buňky označují, kde jsou skutečná třída a předpokládaná třída stejné. Pokud jsou tyto buňky zelené, klasifikátor pracoval dobře a správně klasifikoval pozorování této skutečné třídy.



Obrázek 31. Matice zmatků SVM.

Po analýze dat je možné exportovat (Obrázek 32) daný model a používat na testování dat.



Obrázek 32. Ovládací panel, exportování vyškoleného modelu

#### 4.2.3 Testování SVM

Testování SVM bylo provedeno na stejných náhodných datech jako i neuronová síť (Tabulka 2).

Výsledky testování pomocí podpůrných vektorů jsou představeny na obrázku 33 pro hodnotu 1 a na obrázku 34 pro hodnotu 0. Algoritmus SVM oproti neuronovým sítím udává jako řešení přesnou celočíselnou hodnotu.

```
>>1trainedModel1200.predictFcn([124.591944000,179.416666700,89.0833333300,208.083333300,20.3526016000])  
ans =  
    1  
>>2trainedModel1200.predictFcn([135.024484500,190.877551000,104.551020400,270.6530612000,14.329931570])  
ans =  
    1  
>>3trainedModel1200.predictFcn([142.5657895000,183.416666700,89.7638888900,240.263888900,11.2986602000])  
ans =  
    1
```

Obrázek 33. Výsledky testování SVM pro hodnotu 1

```
>>4 trainedModel200.predictFcn([132.592007400,164.68292680,68.68292683000,177.897561000,8.12365176700])  
  
ans =  
  
0  
  
>>5 trainedModel200.predictFcn([138.240418100,198.250000000,87.0555555600,445.388888900,27.2641400600])  
  
ans =  
  
0  
  
>>6 trainedModel200.predictFcn([140.8571429000,175.7500000,79.91666667000,293,6.015604708000])  
  
ans =  
  
0
```

Obrázek 34. Výsledky testování SVM pro hodnotu 0

## 5 Výsledky a diskuse

Aby bylo zajisteno správné fungování a přesnost neuronové sítě a SVM je třeba zkontrolovat dané modely ještě na větším množství dat, tj, na 300 a na 400, a porovnat výsledky s výše uvedenými. Očekává se zvýšení přesnosti těchto modelů v důsledku zvýšení množství vstupních informací. Při kontrole budou použity matice zmatků a testovací data, která nejsou součástí učení.

### 5.1 Efektivita modelů na 300 množstvích dat

Pro testování na 300 množstvích dat byl použit stejný postup jako na 200 a přesnost modelů je uvedená na maticích zmatků.

#### 5.1.1 Testování neuronové sítě

Neuronová síť podle očekávání vykazovala vyšší přesnost - 89,7%. než u 200 tréninkových dat (Obrázek 35). Pokud se rozběhnete dopředu, stále to není tak efektivní jako SVM, je to vidět o něco dál v práci(Obrázek 38)



Obrázek 35. Matice zmatků neuronové sítě (300)

Pro testování byla vybrána náhodná data pouze s hodnotou  $I$ , protože neuronová síť dělá v nich největší počet chyb (Tabulka 3).

1.	1	142.5657895	183.4166667	89.76388889	240.2638889	11.2986602
2.	1	124.6110092	180.826087	93.43478261	242.6086957	9.803702433
3.	1	130.8496504	175.3333333	83.40277778	342.2777778	21.17167847

Tabulka 3. Zkušební data pro testování neuronové sítě při (300)

Z výsledků testu je vidět, že se znova objevuje chyba v parametrech č. 3 (Obrázek 36).

```
>>1 sim(net300, [142.565789500000;183.416666700000;89.7638888900000;240.263888900000;11.2986602000000])
ans =
    0.8674
>>2 sim(net300, [124.611009200000;180.826087000000;93.4347826100000;242.608695700000;9.80370243300000])
ans =
    0.9723
>>3 sim(net300, [130.849650400000;175.333333300000;83.4027777800000;342.277777800000;21.1716784700000])
ans =
    0.2739
```

Obrázek 36. Výsledky testování neuronové sítě (300)

Při kontrole parametrů č. 1.(viz. Obrázek 20), kde v předchozím testu na 200 množstvích dat vyskytla chyba, je vidět, že teď neuronová síť opravila danou chybu (Obrázek 37). To je jasný příklad toho, jak se lépe síť naučila.

```
>>1 sim(net300, [124.591944000;179.4166667000;89.0833333300;208.083333300;20.3526016800])
ans =
    0.9212
```

Obrázek 37. Opravení chyby

### 5.1.2 Testování SVM

Po učení SVM na 300 množstvích dat byla vybrána model *Quadratic SVM* s 92% přesností (Obrázek 38). SVM dosud vykazuje lepší přesnost než neuronová síť při větším objemu dat. Jediný rozdíl je, že nyní je tato přesnost dána na kvadratickém SVM (*Quadratic SVM*).

<b>1.1</b> ☆ SVM	Accuracy: 91.7%
Last change: Linear SVM	5/5 features
<b>1.2</b> ☆ SVM	Accuracy: <b>92.0%</b>
Last change: Quadratic SVM	5/5 features
<b>1.3</b> ☆ SVM	Accuracy: 89.7%
Last change: Cubic SVM	5/5 features
<b>1.4</b> ☆ SVM	Accuracy: 86.0%
Last change: Fine Gaussian SVM	5/5 features
<b>1.5</b> ☆ SVM	Accuracy: 90.3%
Last change: Medium Gaussian SVM	5/5 features
<b>1.6</b> ☆ SVM	Accuracy: 88.3%
Last change: Coarse Gaussian SVM	5/5 features

Obrázek 38. Výsledky validačního testu SVM (300)

Jak je vidět z matice zmatků (Obrázek 39), SVM taky dělá nejvíc chyb pro hodnotu 1, a proto pro testování byla zvolena stejná data jako pro neuronovou síť (Tabulka 3)



Obrázek 39. Matice zmatků SVM (300)

Na rozdíl od neuronové sítě SVM nevydává chyby při vlastním testování (Obrázek 40).

```
>>1trainedModel300.predictFcn([142.565789500,183.416666700,89.7638888900,240.263888900,11.2986602000])  
ans =  
1  
>>2trainedModel300.predictFcn([124.611009200,180.826087000,93.4347826100,242.6086957000,9.803702433000])  
ans =  
1  
>>3trainedModel300.predictFcn([130.849650400,175.333333300,83.4027777800,342.277777800,21.171678470000])  
ans =  
1
```

Obrázek 40. Výsledky testování SVM (300)

## 5.2 Efektivita modelů na 400 množstvích dat

Aby přesně pochopit, zda SVM pracuje na malých množstvích dat lépe než neuronové sítě, bylo provedeno také testování s účastí 400 dat. V průběhu tohoto testování buď potvrdíme, že SVM, i když ne moc, ale lépe zvládá klasifikaci dat o malých objemech nebo vyvrátíme toto tvrzení.

### 5.2.1 Testování neuronové sítě

Stejně jako dříve bylo provedeno učení neuronové sítě a bylo zjištěno, jakou přesnost ukazuje. Je vidět, že přesnost neuronové sítě je ještě vyšší, nyní je to 91%, což je docela dobrý výsledek (Obrázek 41).





Obrázek 41. Matice zmatků neuronové sítě (400)

Pro testování byly zvoleny parametry č. 3.(viz. Tabulka 3), kde neuronová síť udělala velkou chybu při testu na 300 množstvích dat, jak je vidět, chyba nebyla opravená (Obrázek 42). A to znamená, že síť je stále ještě není dokonale vyškolená, protože tento problém pro SVM byl zvládnutelný.

```
>> sim(net400, [130.849650400;175.333333300;83.4027777800;342.277777800;21.1716784700])
ans =
    0.1901
```

Obrázek 42. kontrola opravení chyby neuronovou sítí

## 5.2.2 Testování SVM

Bylo provedeno poslední učení na 400 datech a zkontrolováno, zda zůstane přesnost vyšší než u neuronové sítě.

1.1 ☆ SVM	Accuracy: 92.0%
Last change: Linear SVM	5/5 features
1.2 ☆ SVM	Accuracy: <b>92.3%</b>
Last change: Quadratic SVM	5/5 features
1.3 ☆ SVM	Accuracy: 89.3%
Last change: Cubic SVM	5/5 features
1.4 ☆ SVM	Accuracy: 85.8%
Last change: Fine Gaussian SVM	5/5 features
1.5 ☆ SVM	Accuracy: 91.5%
Last change: Medium Gaussian SVM	5/5 features
1.6 ☆ SVM	Accuracy: 89.5%
Last change: Coarse Gaussian SVM	5/5 features

Obrázek 43. Výsledky validačního testu pro SVM (400)

Stejně jako v případě s 300 data se ukázala jako nejpřesnější model kvadratický SVM a jeho přesnost je 92,3%, což je vyšší než u neuronové sítě.(Obrázek 43)

Pro testování byly zvoleny parametry č. 3 (Tabulka 3), tam, kde neuronová síť neopravila chybu. SVM udává správnou hodnotu (Obrázek 44).

```
>> trainedModel1400.predictFcn([130.84965040,175.333333300,83.4027777800,342.277777800,21.1716784700])
```

ans =

1

Obrázek 44. Opravení chyby pomocí SVM

## 6 Závěr

Cílem mé práce bylo klasifikovat malé objemy dat a zjistit, který algoritmus, neuronové sítě nebo SVM, vykazuje vyšší efektivitu při malém objemu dat.

Teoretická část práce je věnována nejdůležitějším událostem v oblasti strojového učení, jeho historický vývoj a současné pokroky. Byly zkoumány základní teoretické body související s umělým intelektem. V průběhu teoretické práce jsem identifikoval hlavní oblasti činnosti, ve kterých se strojové učení používá obecně a zejména umělá inteligence. Ukázalo se také, že tohle téma, je v moderní společnosti super-relevantní a její popularita roste každý rok.

Praktická část práce je věnována učení algoritmů založených na neuronových sítích a podpůrných vektorových strojích. Jako nejvhodnější prostředí pro porovnání těchto algoritmů byl vybrán otevřený software MATLAB, který poskytuje nejlepší prostředí pro klasifikační modely. V průběhu mé práce byly provedeny testy na různých objemech dat. Přesný proces učení byl ukázán pro soubor 200 tréninkových dat spolu s analýzou navrhovaných grafů. Poté byly provedeny testy dat v množství 300 resp. 400 pro porovnání efektivitu na různých, ale dosud malých objemech dat.

Podle výsledků všech provedených testů lze vyvodit závěr, že algoritmus založený na podpůrných vektorových strojích je nejlepší volbou pro klasifikaci o malém množství dat. Podle výsledků práce lze říci, že s relativně malými daty pro učení je patrný rozdíl mezi oběma algoritmy a přesnost neuronové sítě může být až o 8-10% nižší. S růstem vstupních dat se však toto číslo (rozdíl v přesnosti) postupně snižuje. Na konci práci, při provádění testů vstupních dat o velikosti 400 vyplývá, že rozdíl v přesnosti je pouze 1%. Lze předpokládat, že čím větší bude množství dat, tím lépe bude fungovat neuronová síť.

Učením podpůrných vektorových strojů se dalo zjistit, že nejpřesnějšími modely jsou lineární a kvadratické SVM. To se dá využít pro jiné možné výzkumy a testování dat v jiných objemech.

## 7 Citovaná literatura

1. **Matthew N. O. Sadiku, Clarence N. Obiozor.** Evolution of Computer Systems. *FIE 2012*. [Online] 2012. <http://fie2012.fie-conference.org/sites/fie2012.fie-conference.org/history/fie96/papers/434.pdf>.
2. **Simplilearn Solutions.** Deep Learning with TensorFlow. *Simplilearn*. [Online] 2019. [Citace: 20. 03 2019.] <https://www.simplilearn.com/what-is-perceptron-tutorial>.
3. **Špaček, Jakub.** Umělá inteligence v praxi. [Online] 11. 06 2019. [Citace: 21. 03 2019.] <https://www.procomputing.cz/2018/06/umela-inteligence-praxi/>.
4. **Rajaraman, V.** The First Electronic Computer. *Resonance*. 05 1996.
5. **Gugerty, L.** Newell and Simon's Logic Theorist: Historical Background and Impact on Cognitive Modeling. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 2012, 50, stránky 880–884.
6. **Blog.helios.** Vše, co jste chtěli vědět o strojovém učení. [Online] 7. 12 2017. [Citace: 16. 03 2019.] <https://blog.helios.eu/cz/clanky/vse-co-jste-chteli-vedet-o-strojovem-uceni/>.
7. **Tyl, Honza.** Všudypřítomná umělá inteligence bude megatrendem příštích deseti let. [Online] 18. 01 2018. [Citace: 19. 03 2019.] <https://m.systemonline.cz/business-intelligence/vsudypritomna-umela-inteligence-bude-megatrendem-pristich-deseti-let.htm>.
8. **Novák, Radek.** Umělá inteligence klepe průmyslu na dveře. [Online] 07 2018. [Citace: 10. 03 2019.] <http://www.edotace.cz/clanky/umela-inteligence-klepe-prumyslu-na-dvere>.
9. **Vacek, Ludvík.** Historie a současnost umělé inteligence. [Online] 2000. [Citace: 08. 03 2019.] <https://www.fi.muni.cz/usr/jkucera/pv109/2000/xvacek.htm>.
10. **Marr, Bernard.** What Is The Difference Between Artificial Intelligence And Machine Learning? *Forbes*. [Online] 06. 12 2016. [Citace: 19. 03 2019.]

<https://www.forbes.com/sites/bernardmarr/2016/12/06/what-is-the-difference-between-artificial-intelligence-and-machine-learning/>.

11. **Slouka, David.** AI vs ML: co je umělá inteligence a co strojové učení? [Online] 04. 03 2019. [Citace: 19. 03 2019.] <https://aiworld.cz/umela-inteligence/ai-vs-ml-co-je-umela-inteligence-a-co-strojove-uceni-288>.

12. **HUMUSOFT s. r. o.** Strojové učení a MATLAB. [Online] 2016. [Citace: 20. 03 2019.] <https://sciencemag.cz/strojove-uceni-a-matlab>.

13. **Genç, Özgür.** Notes on Artificial Intelligence, Machine Learning and Deep Learning for curious people. [Online] 2019. [Citace: 20. 03 2019.] <https://towardsdatascience.com/notes-on-artificial-intelligence-ai-machine-learning-ml-and-deep-learning-dl-for-56e51a2071c2>.

14. **SmartDraw.** Decision Tree. [Online] [Citace: 20. 03 2019.] <https://www.smartdraw.com/decision-tree/>.

15. **Marr, Bernard.** What Are Artificial Neural Networks - A Simple Explanation For Absolutely Anyone. *Forbes*. [Online] 24. 09 2018. [Citace: 21. 03 2019.] <https://www.forbes.com/sites/bernardmarr/2018/09/24/what-are-artificial-neural-networks-a-simple-explanation-for-absolutely-anyone/#237085f91245>.

16. **Branscomb, Mary.** Strojové učení ve firemním IT. [Online] 30. 12 2018. [Citace: 21. 03 2019.] <https://computerworld.cz/software/strojove-uceni-ve-firemnim-it-55114>.

17. **Neuronové sítě.** *Elektronické studijní materiály Mendelové univerzity v Brně*. [Online] [Citace: 22. 03 2019.] [https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=21471](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21471).

18. **SVM - Support Vector Machines.** *TriloByte Statistical Software*. [Online] [Citace: 22. 03 2019.] <http://www.trilobyte.cz/downloadfree/qcemanual/svm.pdf>.

19. **Pupale, Rushikesh.** Support Vector Machines(SVM)—An Overview. [Online] 16. 06 2018. [Citace: 22. 03 2019.] <https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989>.

20. Jak dopadne umělá inteligence a další technologie na naše pracovní životy? [Online] 08. 03 2019. [Citace: 20. 03 2019.] <https://computerworld.cz/analyzy-a-studie/jak-dopadne-umela-inteligence-a-dalsi-technologie-na-nase-pracovni-zivoty-55253>.
21. **Korolov, Maria.** Umělá inteligence ve službách bezpečnosti. [Online] 02. 03 2019. [Citace: 16. 03 2019.] <https://computerworld.cz/securityworld/umela-inteligence-ve-sluzbach-bezpecnosti-55241>.
22. **Korolov, Maria.** Umělá inteligence ve službách zločinu. [Online] 19. 07 2018. [Citace: 16. 03 2019.] <https://computerworld.cz/securityworld/umela-inteligence-ve-sluzbach-zlocinu-54796>.
23. **Ramesh AN, Kambhampati C, Monson JR, Drew PJ.** Artificial intelligence in medicine. *Annals of The Royal College of Surgeons of England*. 10 2004, 86(5),334-338.
24. **Misal, Disha.** The Gaming Industry Is Revolutionising Artificial Intelligence, One Win At A Time. [Online] 08. 09 2018. [Citace: 20. 03 2019.] <https://www.analyticsindiamag.com/the-gaming-industry-is-revolutionising-artificial-intelligence-one-win-at-a-time/>.
25. **Crawford, Charlie.** Introduction to Computer Vision. *Algorithmia*. [Online] 02. 03 2017. [Citace: 20. 03 2019.] <https://blog.algorithmia.com/introduction-computer-vision>.
26. **Johnson, B., Tateishi, R., Hoan, N.** A hybrid pansharpening approach and multiscale object-based image analysis for mapping diseased pine and oak trees. *International Journal of Remote Sensing*. 2013, 34 (20), stránky 6969-6982.
27. **FORBES INSIGHTS With Intel IoT.** Computer Vision: Moving Far Beyond The Visual Cortex. [Online] 2018. [Citace: 20. 03 2019.] <https://www.forbes.com/sites/insights-inteliot/2018/06/14/computer-vision-moving-far-beyond-the-visual-cortex/#29e78c272bfa>.