Odhalování steganografie pomocí neuronových sítí

Steganography detection by means of neural networks

Bc. Jiří Hološka

Diplomová práce
2008

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

*** nascanované zadání str. 1 ***

*** nascanované zadání str. 2 ***

ABSTRAKT

Kryptografie je v dnešní době velmi frekventované slovo. Na bezpečnost přenášených zpráv je dlouhodobě kladen velký důraz a z tohoto důvodu mají specialisté mnoho práce s vymýšlením nových, stále dokonalejších a účinnějších metod šifrování. Vzhledem ke skutečnosti, že užívání či spíše zneužívaní kryptografie k zastření kriminální činnosti má vzestupný trend, specializované kryptoanalytické týmy mají neméně práce s odhalováním a dešifrování zpráv sloužících k výměně informací či řízení činností s kriminálním podtextem. Steganografie je doplňující metoda vedoucí k lepšímu zabezpečení zpráv. Tato metoda jde ruku v ruce s kryptografií a z toho důvodu odhalení takto zabezpečené zprávy není jednoduchou záležitostí. Tato diplomová práce pojednává o modelu detekování zpráv ukrytých pomocí programu OutGuess, využitím neuronových sítí jako klasifikátoru. Model neuronových sítí je velmi pružný ve schopnosti učit se obtížným a rozdílným problémům. Výsledky této práce představují model schopný téměř se stoprocentní účinností detekovat zprávy ukryté pomocí programu OutGuess.

Klíčová slova: Steganografie, kryptografie, neuronové sítě

ABSTRACT

Cryptography is one of very much spoken word nowadays. Security of messages transfer is very important and specialists have a work to think a new cryptography up. Cryptography, on the other hand, is used by jailbirds, so cryptanalysts have also very important job to detect and reveal and then decode the coded messages. Steganography is an additional method leading to better secure up messages which goes hand by hand with cryptography, therefore reveal of such a message is not easy. This thesis deals with neural network models which are able to detect steganography content coded by a program OutGuess using neural networks like taxonomist. Neural networks are methods which are very flexible in learning to different and difficult problems. Results in this paper show that used model had almost 100 % success in revealing steganography by means of OutGuess.

Keywords: Steganography, Cryptography, Neural networks

Prohlašuji, že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně, …….
22. května 2008 Podpis diplomanta

Poděkování

V úvodu této práce bych rád poděkoval Ing. Zuzaně Oplatkové PhD. za její odborné konzultace, připomínky a trpělivost.

Ve Zlíně, 22. května 2008

Jiří Hološka

OBSAH

ÚVOD

V dnešní době si život bez počítačů již pomalu ani neumíme představit, při jejich využívání se dříve nebo později dostaneme do situace, kdy bezpečná výměna informací bude klíčovým faktorem k úspěchu a tady přichází na scénu šifrování dat spolu s utajováním přenosu dat jako takových. Bohužel na druhou stranu výhody a možnosti bezpečné komunikace přitahují pozornost lidí využívajících těchto metod k zakrytí kriminální činnosti, tato situace vede ke známému „začarovanému kruhu“, který představují kryptografové se svou snahou o vytvoření nové, nerozluštitelné šifry a kryptoanalytikové, kteří stejně neúnavně vynalézají metody vedoucí k prolomení takto zabezpečené komunikace. Naprosto identická situace panuje v případě steganografie .

Ve své diplomové práci jsem se zaměřil právě na steganografii - ve světě zabezpečování informací poněkud nedoceněnou menší sestřičku kryptologie. Mezi své cíle jsem si vytyčil nenásilnou formou nastínit historii kryptologie, steganografie, kryptografie a využít prvky kryptoanalýzy spolu s možnostmi klasifikace neuronových sítí. Výsledkem mého snažení by měl být funkční model detekce skrytých informací vložených pomocí programu OutGuess 0.2 do standardních a na první pohled naprosto běžných obrazových informací.

Pokud bych chtěl být pouze šedivý a fádní, použil bych ke klasifikaci statistickou analýzu, která se pro tyto účely běžně požívá, bohužel její nasazení je poměrně komplikované a s nejistými výsledky. Jelikož jsem se rozhodl pojmout řešení kreativněji, netrvalo dlouho a dostal jsem se do mrtvého bodu, ze kterého se mi podařilo pohnout až po prozkoumání temných zákoutí umělé inteligence. Konkrétně jsem v této práci využil neuronové sítě s dopředným šířením chyby a síť RBF.

TEORETICKÁ ČÁST

STEGANOGRAFIE VS. KRYPTOLOGIE

Využívání steganografie k předávání informací je obecně velice efektivní, zejména z toho důvodu, že neznalou osobu zpravidla ani nenapadne, že drží v rukou informaci, která je určená pouze pro určitou skupinu osob či jednotlivce a to i přesto, že je tato zpráva předávána veřejně dostupným médiem. Za druhé světové války byly běžně využívány radiogramy, které obsahovaly klíčová slova, ale pro nezasvěcené posluchače tyto zprávy představovaly pouze přání k narozeninám, svátku, sňatku atd. Pouze osoby, případně skupiny osob, které byly v předstihu informovány o významu vysílaného textu, získávali ze standardního radiového vysílání životně důležité informace či pokyny. Snad neznámější radiogram byl vysílán pátého června 1944, kdy po 21. hodině rozhlasová stanice BBC odvysílala neúplný verš z Verlainovy básně Óda na podzim: "... blessent mon coeur d'une langueur monotonne" ("... zraňují mé srdce monotónní touhou"). Odvysílání tohoto neúplného verše bylo signálem, že odbojové organizace ve Francii mají začít s narušováním německých spojů, vyhazováním mostů a s přepady německých hlídek, neboť ke spojenecké invazi dojde do 48 hodin. Stručně vyjádřeno, klíčová slova se vztahovala na pár desítek povelů, které byly dopředu domluveny a vhodně dosazovány do jinak nezajímavých zpráv.

Úkolem steganografie je skrýt samotnou existenci zprávy, ale zpráva samotná může být napsána nebo předána ve srozumitelné podobě. Je zřejmé, že sem patří velké množství nejrůznějších technik utajení zpráv. [1]

Velkou výhodou zprávy zabezpečené pomocí steganografie spočívá ve způsobu, jakým je se zprávou nakládáno. U steganograficky ošetřené zprávy není a nemá být na první pohled patrné, že daná zpráva obsahuje důvěrné informace. Z toho vyplývá i to, že nezúčastněný pozorovatel zprávu přejde bez povšimnutí, neboť jí nebude přikládat žádnou váhu. Zcela odlišným přístupem k zabezpečení informace přistupuje kryptografie, kde je již na první pohled zřejmá jistá úroveň důvěrnosti zprávy a tím i velká váha sdělované informace. Je to dáno psychologickým jevem udávajícího úměrnost váhy informace a jejím zabezpečením. Bezpečnost zprávy je jištěna “pouze” silou šifrovacího algoritmu a spolehlivostí přenosového media, které zajistí předání zprávy a uchrání zprávu před jakoukoliv snahou se zašifrované zprávy zmocnit .

Steganografie tedy nemá nahradit šifrování, ale v praxi se používá ke zvýšení bezpečnosti přenášené zprávy.

STEGANOGRAFIE

1.1 Historie

Steganografie je věda zabývající se ukrýváním informací v obecných zprávách a předávání skrytých vzkazů. Výraz steganografie vychází z řečtiny a skládá se z výrazů „stenos“ – kryt , „graphos“ – psaní.

Jako o prvním možném využití steganografie se mluví o Řecích Demaratusovi ze Sůs a Histiaeusovi. V 5. století před naším letopočtem Demaratus ze Sůs poslal varování o přípravě perské invaze do Řecka na tu dobu originálním způsobem. Z voskové psací tabulky nejdříve odstranil všechen vosk, poté vyryl zprávu do dřevěné podkladové desky a opětovně celou desku zakryl voskem. Odlišný způsob zvolil Histiaeus, když oholil otrokovi hlavu, zprávu mu na ní poté vytetoval a počkal až mu opět vlasy dorostou. Posléze ho vyslal na cestu do Milétu, aby zprávou pomohl odvrátit útok Peršanů. Takto předané zprávy pomohly rozhodnout bitvy u Thermopyl a Salamin.

Číňané také měli zajímavý způsob předávání tajných zpráv, tajnou zprávu napsali na hedvábí a pak ho zmačkali do malé kuličky a zalili voskem.

Běžně se používaly mnohem jednodušší metody skrývání zpráv a to využitím více či méně důmyslných skrýší jako berle, protézy, fůry s hnojem.

Jednou z možností ukrývání zpráv, patřící mezi méně technicky náročné, je používání neviditelného inkoustu. Jako neviditelný inkoust nám poslouží směs mléka, citrónové šťávy, octu a ovocné šťávy. Takto napsaný text je možné zobrazit po nahřání, kdy se díky rozkladu organických látek tyto tekutiny barví do hněda. Existují i inkousty nevyužívající výše zmíněného rozkladu organických látek, ale zobrazují se pomocí ultrafialového světla jako například kyselina salicylová. Ovšem tento způsob je závislý na používání speciálního druhu papíru.

Za II. světové války se velmi často používala například technika mikroteček. Šlo o malé nezřetelné tečky na pásku filmu, kdy teprve při mnohonásobném zvětšení pod mikroskopem bylo možno přečíst drobné písmo [2].

Také zprávy šířené novinami formou inzerátů nekrologů, rádiem v rámci relace hudby na přání byly za druhé světové války velice populární.

1.1.1 Johannes Trithemius

Slovo steganografie je také spjato s mnichem Johannesem Trithemiem (1. února 1462, Trittenheim – 13. prosince 1516, Würzburg), klášterním opatem ve Sponheimu, Trithemius byl mnohostranný učenec a humanista, známý také jako teoretik angelologie, bílé magie a čarodějnictví. Je rovněž autorem knihy Steganographia, jeho nejznámějšího díla, napsaného v r. 1499 v vydaného ve Frankfurtu r. 1606 a umístěného na Index Librorum Prohibitorum r. 1609. Tato kniha má tři svazky a řadí se do kategorie černé magie a to zejména z důvodu používání ducha (lat. spirit) a duchovních mocností pro komunikaci na dlouhé vzdálenosti. Od té doby, kdy Trithemius publikoval šifrovací klíč v prvních dvou svazcích r. 1606, vešel ve známost tím, že se začal zabývat kryptografií a steganografií. Ještě poměrně nedávno se třetí svazek považoval za dílo o magii, ale později se ukázalo, že se jedná o tzv. stegotexty s velkým kryptografickým významem. Toto dílo propůjčilo jméno moderní vědě steganografii.

Čím se tento duchovní učenec zabýval ve své době (kryptováním, šifrováním a kódováním) a co bylo považováno téměř za černou magii, tím se o pět století později začali seriózně zabývat kryptologové, vojenští a počítačoví experti. [3]

1.1.2 Polygraphia

Roku 1508 dokončuje Trithemius šest svazků díla Polygraphia, vytiskl ho v r. 1518. V pátém díle se nachází tabulka „Tabula recta“, jedná se o polyalfabetickou šifru, inspirovanou césarovou šifrou. Na rozdíl od césarovy šifry se nepoužívá jedna ale 26 abeced a každá je od té předcházející posunutá o jeden znak viz Obrázek 1. [3]

```
1 ABCDEFGHIJKLMNOPQRSTUVWXYZ
2 BCDEFGHIJKLMNOPQRSTUVWXYZA
3 CDEFGHIJKLMNOPQRSTUVWXYZAB
4 DEFGHIJKLMNOPQRSTUVWXYZABC
5 EFGHIJKLMNOPQRSTUVWXYZABCD
6 FGHIJKLMNOPQRSTUVWXYZABCDE
7 GHIJKLMNOPQRSTUVWXYZABCDEF
8 HIJKLMNOPQRSTUVWXYZABCDEFG
....
24 XYZABCDEFGHIJKLMNOPQRSTUVW
25 YZABCDEFGHIJKLMNOPQRSTUVWX
26 ZABCDEFGHIJKLMNOPQRSTUVWXY
```

Obrázek 1 – zkrácená verze „Tabula recta“

Soubor věnoval císaři Maxmiliánovi. Popularita byla tak veliká, že kniha byla přepisována a vydávána ještě celá staletí po autorově smrti. Například v r. 1561 ji přeložil Gabriel de Collanges a po něm r. 1620 ještě Dominique de Hotting. [3]

1.1.3 Vliv

Soubor díla Cipher Manuscripts byl zašifrován s použitím Trithemiovy šifry, tedy pomocí jednoduchého zaměňování znaků, které popsal ve své knize Polygraphia. Cipher Manuscripts byl použit k založení Hermetického řádu úsvitu, prvního tajného sdružení, které velmi ovlivnilo Viktoriány a moderní evropský esoterismus. [3]

1.2 Moderní steganografie

Minulé století bylo ve znamení páry a elektřiny ale také ve znamení dvou světových válek, následné studené války a celé řady lokálních ozbrojených konfliktů – tyto události a zejména potřeba efektivního zabezpečení dopravovaných zpráv byly významným hnacím motorem v oblastech steganografie a kryptologie, tyto disciplíny v závislosti na výše uvedených faktorech dosáhly do té doby nevídaného rozmachu. Dominantním znakem století současného jsou počítače a telekomunikační systémy. V závislosti na vývoji počítačů se vyvíjejí různé vědní obory a pokud se přímo nerozvíjí v závislosti na IT, tak jsou alespoň jejich vývojem částečně poznamenány, neboť IT a telekomunikační technologie významnou měrou na jedné straně usnadnily velké části lidské populace vzájemnou komunikaci, na straně druhé, tato stále se rozvíjející elektronická komunikace s sebou nese nutnost zdokonalování systémů pro ochranu přenášených zpráv, počítačová data představují pro steganografii a kryptografii nepřeberné množství způsobů uplatnění.

.

1.3 Steganografické metody

1.3.1 Metoda využití nejméně významných bitů:

Při kódování zpráv existují bity s různou důležitostí. Tyto méně významné bity je možné vyměnit za bity ukrývané zprávy bez toho, aby došlo ke zřejmé změně původní zprávy.

Například při ukrývání informace do grafického souboru GIF (Graphic Interchange Format) je obraz dán maticí pixelů neboli obrazových bodů. Každý z těchto bodů je

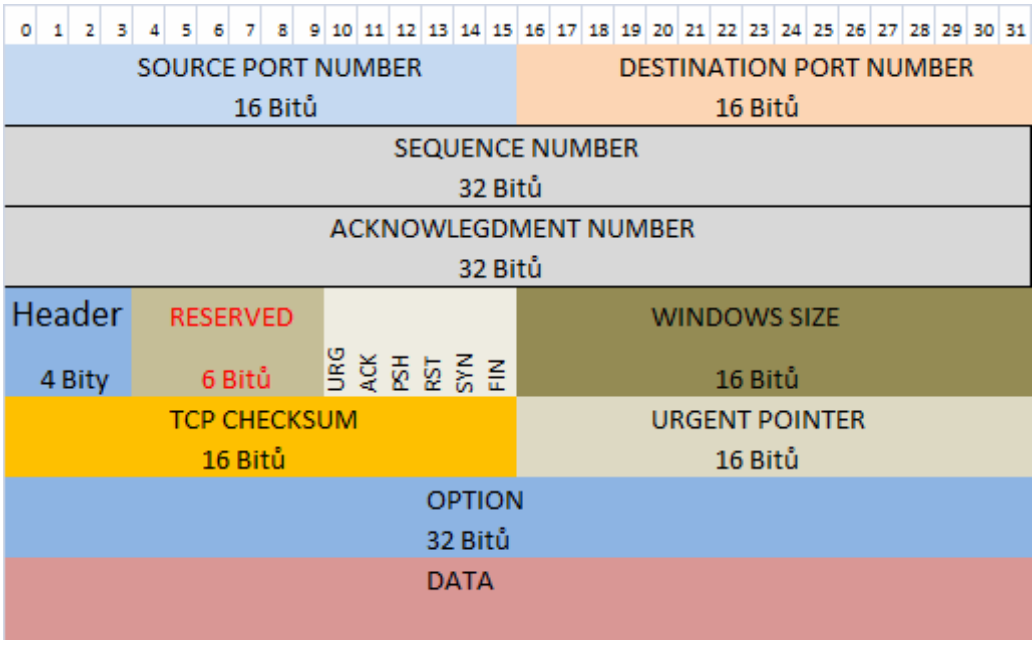
reprezentován 24 bitovým (24 bitů = 3 bajty) slovem, vyjadřující podíl červené, zelené a modré barvy v daném obrazovém bodu.

Počet všech možných barevných kombinací je 2^{24}, což se rovná 16,8 milionů. Změnou posledního bajtu se barva pixelu změní na jeden z $2^3 = 8$ navzájem sousedících odstínů. Lidské oko není schopné rozlišit 8 sousedních odstínů v paletě 16,8 milionů barev.

U grafického formátu JPEG je situace o něco složitější, protože se obrázek nekóduje bit po bitu jako u GIF nebo BMP, ale je rozdělen do bloků 8x8 obrazových bodů, který se převede na frekvenční údaje diskrétní kosinové transformace (DCT). DCT je vyjádřena maticí 8x8 frekvenčních koeficientů. Tyto koeficienty se následně dělí pomocí kvantizačního koeficientu z kvantizační matice (DQT). Hodnoty vyjádřené z DCT bývají v rozmezí od -1024 do 1023. To odpovídá 11 bitovým slovům, nejvyšší hodnota z kvantizační tabulky je okolo 10. Z toho vychází, že by při přenášení měl zabírat zhruba á bitů. Z nastíněné situace jasně vyplývá větší náročnost ukrývání informací do obrazového formátu JPEG než u GIF, BMP atd. Samozřejmě metoda nahrazování nejméně významných bitů funguje i zde, je jen náročnější a i maximální velikost skrývané zprávy musí být zákonitě menší než u předešlých formátů.

1.3.2 Metoda využití rezervních dat

Metoda využívaná u různých přenosových protokolů, kde jsou přenášeny i rezervní bity jako například u TCP protokolu, na význam těchto dat není při směrování ani při následném zpracování v koncových zařízeních brán ohled, proto se přesně hodí pro potřeby přenosu skrytých zpráv u TCP, kde je možné přenést 6 bitů v každé TCP hlavičce, jak ukazuje obrázek 2.



Obrázek 2- hlavička protokolu TCP

POUŽITÉ PROGRAMY

1.4 OutGuess

Je univerzální steganografický nástroj, který umožňuje vkládání skrytých informací do redundantních bitů vstupních dat. Druh vstupních dat není pro OutGuess vůbec důležitá, jelikož program využívá specifické ovladače pro dané grafické formáty, extrahující redundantní bity a následně je po úpravách zapíší zpět. Ve verzi, kterou jsem použil v diplomové práci jsou podporovány formáty JPEG a PNG.

OutGuess je dostupný pod BSD licencí a je ho možné používat zdarma i pro komerční účely.

Stáhnout je ho možné v podobě UNIXových zdrojových balíčků a binárního balíčku pro Windows z adresy http://www.outguess.org/download.php.

1.4.1 Podporované systémy

- OpenBSD: i386, alpha, sparc
- NetBSD: vax
- FreeBSD: i386
- Linux: i386, sparc, powerpc, alpha
- AIX: RS6000, powerpc
- Solaris: ultra sparc
- HP-UX: mips
- BeOS: works, platform unknown
- Mac OS X: powerpc
- Windows NT: i386

1.4.2 OutGuess pracuje odlišně

OutGuess u JPEG obrazů odolává statistickým výpočtům založených na frekvenční analýze. Výsledky statistických testů nejsou schopny odhalit přítomnost steganografického obsahu, protože OutGuess před samotným vkládáním dat do obrazu zjišťuje maximální velikost zprávy, která může být skryta a bude stále možné statisticky nezměnit výsledný obraz z pohledu frekvenční analýzy. Tato technika byla popsána na bezpečnostní odborné konferenci Defending Against Statistical Steganalysis Niels Provos, 10th USENIX Security Symposium. Washington, DC, August 2001.

OutGuess používá váhy ke zjištění, které bity mohou být změněny. Pomocí semínek může být ovlivněno chování těchto vah. Semínko je vloženo do dat spolu se zbytkem

vkládané zprávy. Pomocí změn semínek se OutGuess pokouší najít posloupnosti bitu minimalizující počet změn v datech, které musejí být změněny.

1.5 Steghide

Je steganografický program se schopností vkládat data do různých grafických i zvukových datových formátů. Při vkládání informací Steghide se stejně jako OutGuess snaží vyhnout modifikaci barevného schématu obrázků a tím snížit účinnost frekvenční analýzy.

V diplomové práci jsem použil aktuální verzi jež má číslo 0.5.1.

Tato verze podporuje :

Kompresi vkládaných dat

Šifrování vkládaných dat

Vkládání kontrolních součtů k ověření celistvosti extrahovaných dat

Datové formáty JPEG,BMP z obrazových a ze zvukových to jsou WAW a AU.

Steghide je uvolněn pod GNU General Public Licence (GPL), která povoluje úpravy a distribuci programu tak dlouho dokud budou tyto úpravy dostupné pod GPL.

1.6 Mathematica 5.2

Při plánování této diplomové práce jsem si zvolil vývojové prostředí Mathematica 5.2, zaměřeného na matematické výpočty a poskytujícího velké množství typů výpočtů a vizualizací z důvodů dostupnosti dvou toolboxů :

1) Digital Image Processing

Tento toolbox je plně integrován do prostředí Mathematica 5.2 a poskytuje rychlé zpracování a analýzu obrázků.

Zpracování obrázků popisují publikace :

Practical Handbook on Image Processing for Scientific and Technical Applications [4]

Digital Image Processing [5]

Bohužel, ve chvíli, kdy jsem potřeboval zpracovat více jak 200 obrázků, se použití tohoto toolboxu stalo pro mě neefektivní a raději sem se uchýlil k jednoduššímu programu JpegSnoop, jehož možnosti se sice co do obsáhlosti funkcí s toolboxem od Wolfram Researge nemohou rovnat, nicméně jeho použití v diplomovém projektu mi ušetřilo mnoho práce při vytvářením podkladů trénovacích množin.

2) Neural Net

Tento toolbox je také z dílny Wolfram Research a stejně jako Digital image processing je plně implementován do prostředí Mathematica 5.2.

Tento toolbox umí zpracovávat sítě Perceptron, Hopfieldovu síť, ale také síť s dopředným šířením chyby a síť RBF vhodné ke klasifikaci.

Klasifikace je právě typ operace, který jsem pro odhalování steganografie použil. Určitý vzorek chceme buď zařadit k neupraveným nebo „zakódovaným“ obrázkům.

1.7 JPEG snoop

JPEG snoop je program vytvořený pro potřeby zkoumání rozšiřujících informací v obrazových, video a textových souborech , aktuální verze JPEG snoopu podporuje tyto formáty:

- JPG - JPEG
- .THM - Náhledy RAW Photo / video souborů
- .AVI* - AVI videa
- .DNG - Digital Formát digitálního negativu
- .CRW, .CR2, .NEF, .ORF, .PEF - RAW Photo
- .MOV* - Videa QuickTime, QTVR (Virtual Reality / 360 Panoramic)
- .PDF – Dokumenty Adobe PDF

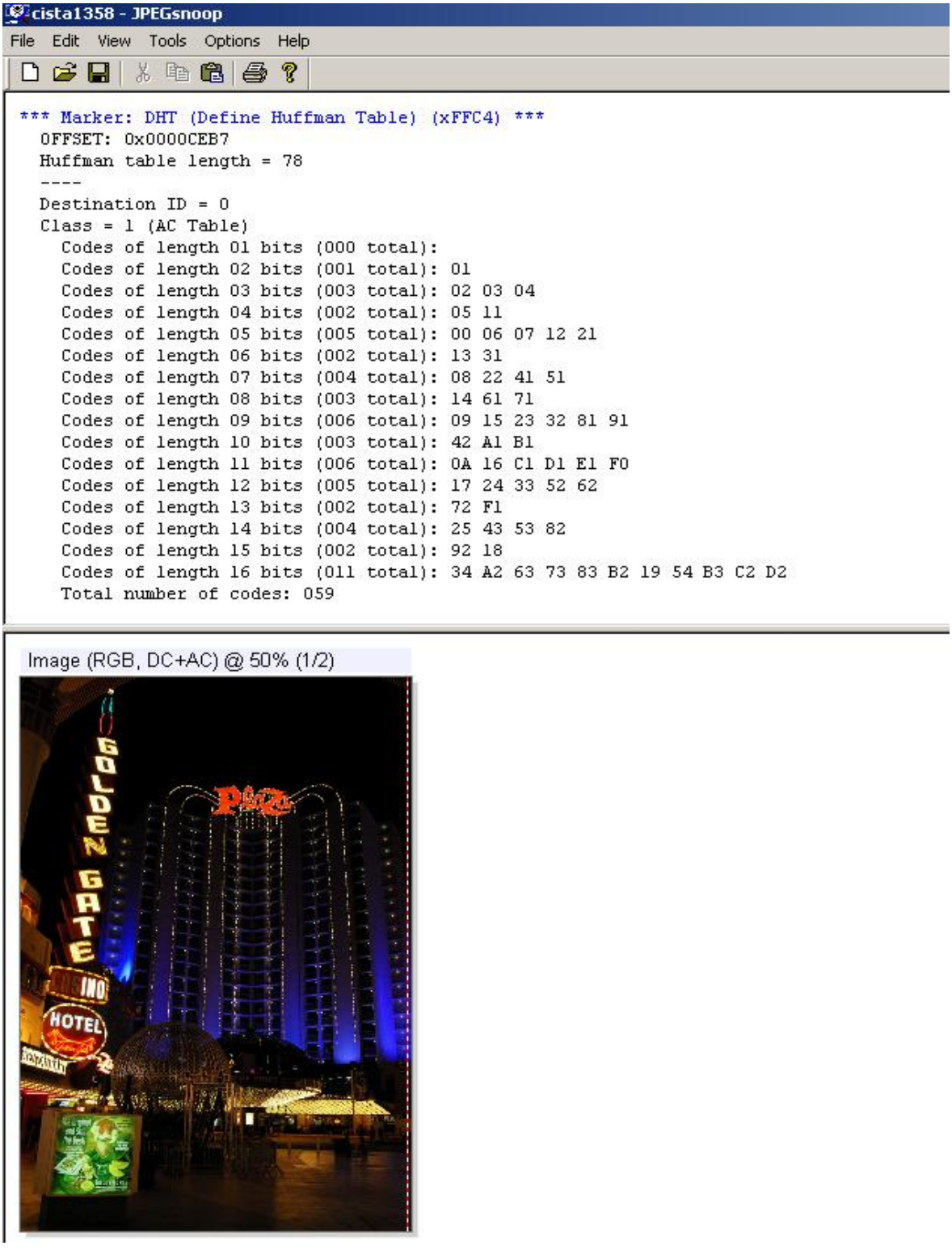
Pomocí JPEG snoopu jsme schopni zobrazit a vyextrahovat z obrazové informace data v podobě:

- Matic kvantizačních tabulek barevnosti a jasu
- Informace o redukci barevných složek
- Stanovení kvality JPEG obrazu
- EXIF metadat
- RGB histogramů
- Tabulek Huffmanova kódu

Program má grafické uživatelské prostředí, jak ukazuje obrázek 3, ale je možné použít i vstup z příkazové řádky.

Práce s programem v příkazové řádce vypadá následovně.

jpegsnoop.exe -i obrázek.jpg -o vystup.txt -nogui



Obrázek 3 - Grafické prostředí programu JPEGsnoop

METODIKA VZORKOVÁNÍ

1.8 Kvantizační tabulky

Původní návrh počítal s použitím kvantizačních tabulek (DQT), zobrazující jas a barevnost, neboť po použití steganografických programů vykazovaly zjevné změny, bohužel, jak se později ukázalo, tento postup nebyl vhodný pro získání dat potřebných pro sestavení trénovacích množin.

1.8.1 Původní vzorky DQT

Tabulka 1 – DQT neupravených obrázků

Kvantizační tabulka jasu

Destination ID=0 (Luminance)

Row #0:	1	1	1	1	1	1	2	3
Row #1:	1	1	1	1	1	1	3	4
Row #2:	1	1	1	1	1	2	4	5
Row #3:	1	1	1	1	2	3	4	5
Row #4:	1	1	2	2	3	4	5	5
Row #5:	2	3	3	4	5	5	6	5
Row #6:	2	3	3	4	5	6	6	5
Row #7:	3	2	2	3	4	4	5	5

Kvantizační tabulka barevnosti

Destination ID=1 (Chrominance)

Row #0:	1	1	1	2	5	5	5	5
Row #1:	1	1	1	3	5	5	5	5
Row #2:	1	1	2	5	5	5	5	5
Row #3:	2	3	5	5	5	5	5	5
Row #4:	5	5	5	5	5	5	5	5
Row #5:	5	5	5	5	5	5	5	5
Row #6:	5	5	5	5	5	5	5	5
Row #7:	5	5	5	5	5	5	5	5

1.8.2 Vzorky DQT po použití programu OutGuess

Tabulka 2 – DQT upravených obrázků

Kvantizační tabulka jasu

Destination ID=0 (Luminance)

Row #0:	8	6	5	8	12	20	26	31
Row #1:	6	6	7	10	13	29	30	28
Row #2:	7	7	8	12	20	29	35	28
Row #3:	7	9	11	15	26	44	40	31
Row #4:	9	11	19	28	34	55	52	39
Row #5:	12	18	28	32	41	52	57	46
Row #6:	25	32	39	44	52	61	60	51
Row #7:	36	46	48	49	56	50	52	50

Kvantizační tabulka barevnosti

Destination ID=1 (Chrominance)

Row #0:	9	9	12	24	50	50	50	50
Row #1:	9	11	13	33	50	50	50	50
Row #2:	12	13	28	50	50	50	50	50
Row #3:	24	33	50	50	50	50	50	50
Row #4:	50	50	50	50	50	50	50	50
Row #5:	50	50	50	50	50	50	50	50
Row #6:	50	50	50	50	50	50	50	50
Row #7:	50	50	50	50	50	50	50	50

1.9 Huffmanovo kódování

Huffmanovo kódování bylo navrženo roku 1952 Davidem Huffmanem. Tato metoda bere symboly reprezentovanými například hodnotami DCT a kóduje je kódem s proměnnou délkou tak, že přiřazuje dle statistické pravděpodobnosti nejkratší bitové vyjádření znakům s nejvyšším výskytem. Naopak, u znaků vyskytujících se jen zřídka, je přiřazeno delší bitové vyjádření. Tato metoda kódování byla využita Samuelem Morsem k zakódování znaků abecedy dle výskytu v anglickém jazyce a vznikla tak Morseova abeceda .

Princip metody spočívá ve vytvoření binárního stromu, jehož koncové uzly odpovídají symbolům původní abecedy, hrany jsou ohodnoceny symboly 0 a 1 a uzly jsou ohodnoceny pravděpodobností výskytu. Pravděpodobnost vnitřního uzlu je přitom rovna součtu pravděpodobností jeho následníků. Uzly řadíme do posloupnosti podle rostoucí pravděpodobnosti, v každém kroku z ní odstraníme dva uzly s nejnižší prioritou, vytvoříme z nich následníky nového uzlu a ten opět zařadíme do seznamu.

Huffmanův kód má dvě důležité vlastnosti. Jednak je kódem s minimální délkou, a také je to prefixový kód, je tedy jednoznačně dekódovatelný. Jeho problémem je to, že musíme znát rozdělení pravděpodobnosti výskytu jednotlivých symbolů. To lze nahradit odhadem, případně je možné tento odhad v průběhu komprese upřesňovat [6].

JPEG obsahuje až 4 tabulky. Vytváření těchto tabulek všeobecně dovoluje vyčíst, jak často se každý znak (DCT kódové slovo) vyskytuje v obrázku a nastavit tak příslušnou délku bitového vyjádření.

Tabulka 3 obsahuje výsledky histogramu Huffmanova kódování obsažené v JPEG souboru použitého k přípravě učících množin neuronové sítě.

Obrázky 3, 4 a 5 zobrazují 3 jednoduché grafy znázorňující frekvenční analýzu. Představující změny výskytu slov v rozmezí 1-16 bitů v jednotlivých střídavých a stejnosměrných složkách, které mi byly vodítkem při sestavování trénovacích množin pro detekci dat vložených programem OutGuess.

1.9.1 Výsledky histogramu Huffmanova kódování:

Tabulka 3 - výsledky histogramu Huffmanova kódování

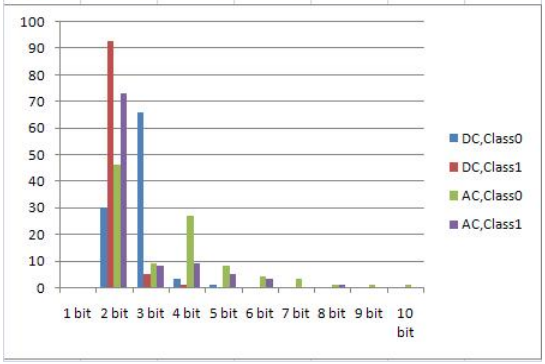
Huffman Table: (Dest ID: 0,Class: DC)		**Huffman Table: (Dest ID: 1,Class: DC)**	
# codes of length 01 bits:	0 (0%)	# codes of length 01 bits:	0 (0%)
# codes of length 02 bits:	738 (6%)	# codes of length 02 bits:	6458 (53%)
# codes of length 03 bits:	10128 (82%)	# codes of length 03 bits:	3482 (28%)
# codes of length 04 bits:	774 (6%)	# codes of length 04 bits:	1604 (13%)
# codes of length 05 bits:	408 (3%)	# codes of length 05 bits:	456 (4%)
# codes of length 06 bits:	123 (1%)	# codes of length 06 bits:	239 (2%)
# codes of length 07 bits:	72 (1%)	# codes of length 07 bits:	45 (0%)
# codes of length 08 bits:	45 (0%)	# codes of length 08 bits:	4 (0%)
# codes of length 09 bits:	0 (0%)	# codes of length 09 bits:	0 (0%)
# codes of length 10 bits:	0 (0%)	# codes of length 10 bits:	0 (0%)
# codes of length 11 bits:	0 (0%)	# codes of length 11 bits:	0 (0%)
# codes of length 12 bits:	0 (0%)	# codes of length 12 bits:	0 (0%)
# codes of length 13 bits:	0 (0%)	# codes of length 13 bits:	0 (0%)
# codes of length 14 bits:	0 (0%)	# codes of length 14 bits:	0 (0%)
# codes of length 15 bits:	0 (0%)	# codes of length 15 bits:	0 (0%)
# codes of length 16 bits:	0 (0%)	# codes of length 16 bits:	0 (0%)

Huffman Table: (Dest ID: 0,Class: AC)	
# codes of length 01 bits:	0 (0%)
# codes of length 02 bits:	254706 (56%)
# codes of length 03 bits:	45859 (10%)
# codes of length 04 bits:	67263 (15%)
# codes of length 05 bits:	42938 (9%)
# codes of length 06 bits:	16805 (4%)
# codes of length 07 bits:	12090 (3%)
# codes of length 08 bits:	6890 (2%)
# codes of length 09 bits:	4443 (1%)
# codes of length 10 bits:	2259 (0%)
# codes of length 11 bits:	627 (0%)
# codes of length 12 bits:	423 (0%)
# codes of length 13 bits:	0 (0%)
# codes of length 14 bits:	0 (0%)
# codes of length 15 bits:	28 (0%)
# codes of length 16 bits:	357 (0%)

Huffman Table: (Dest ID: 1,Class: AC)	
# codes of length 01 bits:	0 (0%)
# codes of length 02 bits:	54847 (37%)
# codes of length 03 bits:	31300 (21%)
# codes of length 04 bits:	26458 (18%)
# codes of length 05 bits:	12812 (9%)
# codes of length 06 bits:	10743 (7%)
# codes of length 07 bits:	2663 (2%)
# codes of length 08 bits:	4279 (3%)
# codes of length 09 bits:	2419 (2%)
# codes of length 10 bits:	1192 (1%)
# codes of length 11 bits:	370 (0%)
# codes of length 12 bits:	18 (0%)
# codes of length 13 bits:	0 (0%)
# codes of length 14 bits:	263 (0%)
# codes of length 15 bits:	149 (0%)
# codes of length 16 bits:	15 (0%)

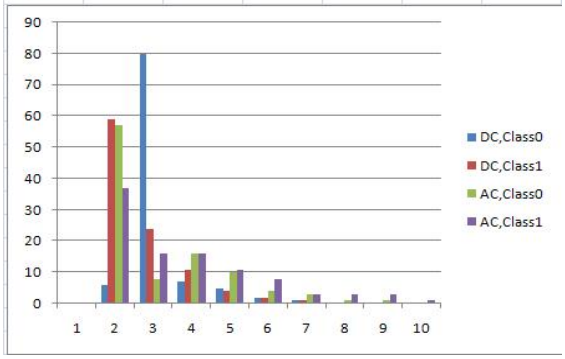
1.9.2 Grafy frekvenčních analýz

	DC,Class0	DC,Class1	AC,Class0	AC,Class1
1 bit	0	0	0	0
2 bit	30	93	46	73
3 bit	66	5	9	8
4 bit	3	1	27	9
5 bit	1	0	8	5
6 bit	0	0	4	3
7 bit	0	0	3	0
8 bit	0	0	1	1
9 bit	0	0	1	0
10 bit	0	0	1	0
11 bit	0	0	0	0
12 bit	0	0	0	0
13 bit	0	0	0	0
14 bit	0	0	0	0
15 bit	0	0	0	0
16 bit	0	0	0	0



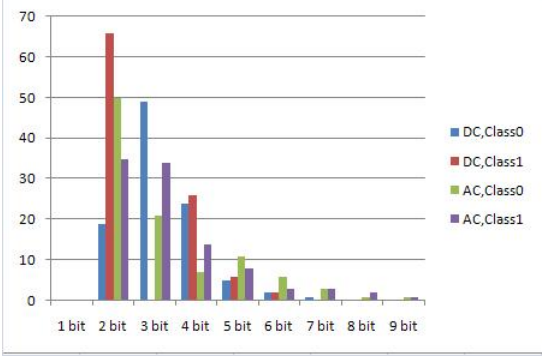
Obrázek 4 – neupravený obrázek

	DC,Class0	DC,Class1	AC,Class0	AC,Class1
1 bit	0	0	0	0
2 bit	6	59	57	37
3 bit	80	24	8	16
4 bit	7	11	16	16
5 bit	5	4	10	11
6 bit	2	2	4	8
7 bit	1	1	3	3
8 bit	0	0	1	3
9 bit	0	0	1	3
10 bit	0	0	0	1
11 bit	0	0	0	0
12 bit	0	0	0	0
13 bit	0	0	0	0
14 bit	0	0	0	0
15 bit	0	0	0	0
16 bit	0	0	0	0



Obrázek 5 – obrázek zpracovaný programem OutGuess

	DC,Class0	DC,Class1	AC,Class0	AC,Class1
1 bit	0	0	0	0
2 bit	19	66	50	35
3 bit	49	0	21	34
4 bit	24	26	7	14
5 bit	5	6	11	8
6 bit	2	2	6	3
7 bit	1	0	3	3
8 bit	0	0	1	2
9 bit	0	0	1	1
10 bit	0	0	0	0
11 bit	0	0	0	0
12 bit	0	0	0	0
13 bit	0	0	0	0
14 bit	0	0	0	0
15 bit	0	0	0	0
16 bit	0	0	0	0



Obrázek 6 - obrázek zpracovaný programem StegHide

NEURONOVÉ SÍTĚ

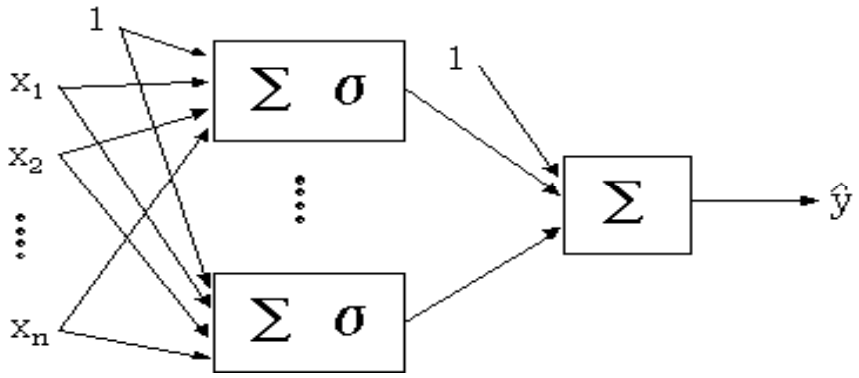
Pro steganografii jsme použili dva typy neuronových sítí – s dopředným šířením chyby a síť RBF.

1.10 Sítě s dopředným šířením chyby

Všeobecně struktura neuronu umožňuje zpracování několika vstupů a to včetně jednotlivých prahů. Při nelineárním zpracování získáváme jednotný výstup.

Sítě s dopředným šířením chyby jsou nejpopulárnější a nejčastěji používaným modelem v mnoha aplikacích. Jsou známy pod několika různými jmény jako například mnohovrstvý Perceptron.

Následující Obrázek 7 ukazuje příklad sítě s dopředným šířením chyby (feedforward net) s jednou skrytou vrstvou, vstupy x1.... xn a prahovou hodnotou 1 a výstupní hodnotou ŷ. Každá šipka v obrázku představuje parametr v síti. Síť je rozdělena do vrstev. Vstupní vrstva je složená výhradně ze vstupů sítě. Dále následuje skrytá vrstva, která obsahuje přesně nespecifikovaný počet neuronů nebo skrytých jednotek zapojených paralelně. Každý neuron provádí vážený součet vstupů, který pak prochází nelineární aktivační funkcí, někdy také nazývanou funkcí neuronovou. V literatuře se běžně uvádí dva přístupy k používání vah – buď vstupy x1 až xn jsou pouze větvící uzly, u kterých se žádné váhy neobjevují, neboť se aplikují až v následující vrstvě a nebo v případě druhého přístupu jsou už i tyto vstupy do první vrstvy neuronů ováhované. Druhý přístup používá toolbox NeuralNet pro Mathematicu.



Obrázek 7 – Příklad sítě s dopředným šířením chyby

Matematická funkce skrytého neuronu je popsána

$$\sigma\left(\sum_{j=1}^{n} w_j\, x_i + b_j\right) \tag{1}$$

Výstup ze sítě je tvořen dalším váženým součtem výstupů neuronů ve skryté vrstvě. Tento součet na výstupu se nazývá výstupní vrstva.

Neurony ve skryté vrstvě sítě mají podobnou strukturu jako u Perceptronu s tím rozdílem, že jejich aktivační funkce může být funkcí diferenciální.

Výstup této sítě je dán vzorečkem

$$\hat{g}(\theta) = g(\theta, x) = \sum_{i=1}^{nh} w_i^2\, \sigma\left(\sum_{j=1}^{n} w_{i,j}^1\, x_i + b_{j,i}^1\right) + b^2 \tag{2}$$

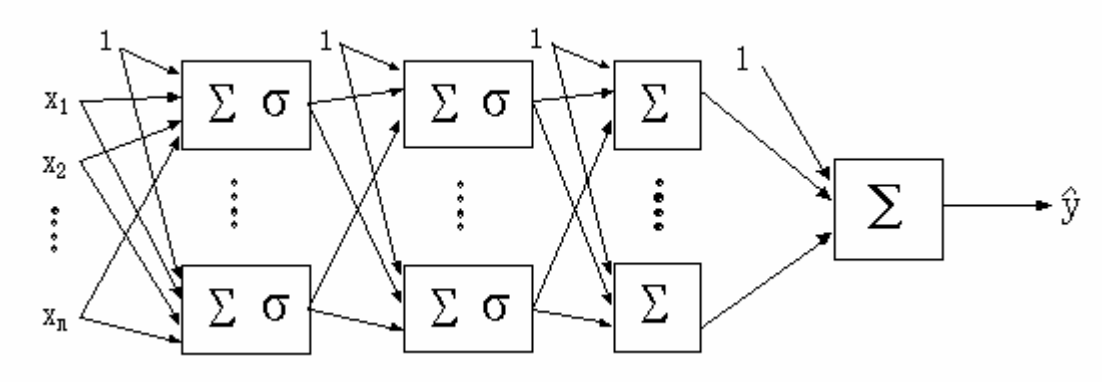
Kde *n* je počet vstupů a *nh* je počet neuronů ve skryté vrstvě. Proměnné { $w_{i,j}^1$, $b_{j,i}^1$, w_i^2, b^2 } jsou parametry síťového modelu, který je souhrně reprezentován parametrem vektoru θ. Obvykle je model neuronové sítě reprezentován dohodnutým zápisem g*(θ,x)*, ačkoliv přesná struktura neuronové sítě taková být nemusí.

Velikost vstupní a výstupní vrstvy je dána počtem vstupů a výstupů sítě, takže po specifikování sítě je potřeba určit pouze počet skrytých neuronů. Někdy se tato síť uvádí jako třívrstvá, počítá se ze vstupní, skrytou a výstupní vrstvou. Protože vstupní vrstva nic nezpracovává, je také někdy nazývána dvouvrstvou sítí. Aby se předešlo zbytečným nedorozuměním, budu tuto síť nazývat jako sít s dopředným šířením chyby a jednou skrytou vrstvou.

Trénování sítě je prováděno přírůstkovým upravováním jejích parametrů do té doby, než tréninková data ŷ(θ) uspokojivě korespondují s požadovaným výstupem. Algoritmů k učení je více - toolbox preferuje Levenberg-Marquardtův algoritmus.

Počet vrstev a počet neuronů v každé skryté vrstvě si definuje uživatel sám. Obecně platí pravidlo výběru co možná nejednodušší síťě pro řešení daného problému, bohužel toto pravidlo není příliš použitelné. Z této okolnosti vyplývá nutnost experimentování s různými návrhy sítě a porovnávání výsledků, tak abychom mohli zvolit síť, která bude nejlépe odpovídat našim požadavkům při řešení problému.

Pro většinu praktických nasazení bude síť s jednou či dvěma skrytými vrstvami dostačující. Doporučuje se začít s lineárním modelem, což je neuronová síť bez skryté vrstvy, následně přejít k síti s jednou skrytou vrstvou, která neobsahuje více jak 5-10 neuronů. Jako poslední krok by se měla zkoušet síť se dvěmy skrytými vrstvami [7].



Obrázek 8 - Příklad vícevrstvé sítě s dopředným šířením chyby

1.11 RBF síť

Radial Basis Function (RBF) je třívrstvá síť, jejíž struktura je obdobná jako u třívrstvé sítě s algoritmem Backpropagation, ale přenosová funkce výstupních neuronů musí být lineární, což nemusí být pro síť s algoritmem Backpropagation pravda a přenosové funkce skrytých neuronů jsou tzv. Radial Basis Functions, odtud i název sítě. Jejich charakteristickým znakem je, že buď monotónně klesají nebo rostou směrem od svého středového bodu.

Kromě vstupní vrstvy, která slouží jen pro předání hodnot, má RBF síť vrstvu RBF (skrytá vrstva) a vrstvu výstupní tvořenou Perceptrony. Mezi jednotlivými vrstvami se zpravidla používá úplné propojení. Výpočet vnitřního potenciálu pak řeší rovnice (3).

$$\phi = \sqrt{\sum_{i=1}^{n} (x_i - c_i)^2} \tag{3}$$

Jinými slovy: Vnitřní potenciál se počítá jako euklidovská vzdálenost vstupního vektoru x od c dělená šířkou b.

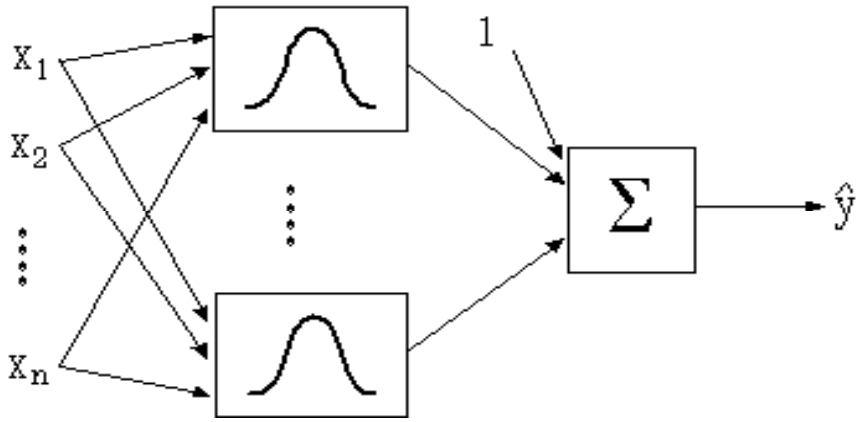
Pro RBF neurony se používá Eukleidovská metrika, narozdíl od Perceptronů, kde se používá skalární součin. Vektor C={c1,...,cn} označujeme jako prototyp, protože reprezentuje jistou podmnožinu vstupních dat ve tvaru shluku. Jako aktivační funkce se nejčastěji používá Gaussova funkce a multikvadratická funkce (ale i jiné, viz dále).

1.11.1 RBF síť jako klasifikátor

V tomto případě můžeme využít jak spojitých, tak nespojitých výstupních funkcí RBF neuronů. Nespojité zařazují vstupní vektor do naučené množiny (shluku), spojité mohou informovat do jaké míry patří vektor do této množiny. Základní pravidla pro stavbu sítě jsou:

- Výstup RBF neuronu je napojen pouze na jeden výstupní neuron.
- Na výstupní neuron může být napojeno více RBF neuronů.
- Průnik sféry vlivu RBF neuronů náležejících více různým kategoriím musí být prázdný.
- RBF neurony téže kategorie musí reprezentovat vzory dané kategorie s minimální chybou.

Pokud používáme pro klasifikaci RBF síť se spojitými výstupními funkcemi RBF neuronů, musí být váhy a práh výstupních neuronů nastaven tak, aby tyto neurony realizovaly prahovou logickou funkci OR. Obvykle se RBF síť učí řádově rychleji, než obyčejná dopředná neuronová síť. Bianchiniho kolektiv v roce 1995 uvedl, že kdykoli je vstupní prostor oddělitelný pomocí sfér (hypersfér), chybová funkce minimalizuje výskyt místních minim. To je velká výhoda oproti klasickým vícevrstvým neuronovým sítím. RBF síť je zpravidla pomalejší při použití většího počtu uzlů [8].



Obrázek 9 – Příklad sítě RBF

PRAKTICKÁ ČÁST

PRAXE

1.12 Outguess

OutGuess je napsán pro práci v příkazové řádce neobsahuje tedy žádné grafické uživatelské prostředí.

1.12.1 Vkládání dat

```
$ outguess -k "heslo" -d zprava.txt vstupniobraz.jpg vystupniobraz.jpg

Reading vstupniobraz.jpg....
JPEG compression quality set to 75
Extracting usable bits: 40059 bits
Correctable message size: 21194 bits, 52.91%
Encoded zprava.txt: 14712 bits, 1839 bytes
Finding best embedding...
   0:  7467(50.6%)[50.8%], bias  8137(1.09), saved:   -13, total: 18.64%
   1:  7311(49.6%)[49.7%], bias  8079(1.11), saved:     5, total: 18.25%
   4:  7250(49.2%)[49.3%], bias  7906(1.09), saved:    13, total: 18.10%
  59:  7225(49.0%)[49.1%], bias  7889(1.09), saved:    16, total: 18.04%
59, 7225: Embedding data: 14712 in 40059
Bits embedded: 14744, changed: 7225(49.0%)[49.1%], bias: 7889, tot: 40032,
skip: 25288
Foiling statistics: corrections: 2590, failed: 1, offset: 122.585494 +- 239.664983
Total bits changed: 15114 (change 7225 + bias 7889)
Storing bitmap into data...
Writing foil/ vystupniobraz...
```

Seed – hodnota semínka pro výsledný řádek

Changed Bits – Počet bitů které je potřeba změnit a podíl v procentech ve vztahu k množství skrývaných dat.

Bias – celková odchylka změněných bitů. Celková odchylka je měřítkem odhalitelnosti změn.

Saved – Počet bytů které mohou být beze změny reprezentovány,základ je 50 %.

1.12.2 Získávání dat

```
$ outguess -k " heslo " -r vystupniobraz.jpg text.txt
```

```
Reading out.jpg....
Extracting usable bits:   40059 bits
Steg retrieve: seed: 7225, len: 1839
```

1.13 Steghide

Stejně jako Outguess je Steghide navržen pro práci v příkazové řádce.

1.13.1 Vkládání dat

Základní syntaxe

$ steghide embed -cf vstupniobraz.jpg -ef zprava.txt

Enter passphrase:

Re-Enter passphrase:

embedding " zprava.txt " in " vstupniobraz.jpg "... done

Rozšířená syntaxe

steghide embed -p heslo -cf vstupniobraz.jpg -ef zprava.txt -sf vystupniobraz.jpg

-p přepínač pro vložení hesla.

-cf definuje vstupní soubor, do kterého budou vkládány data.

-ef definuje soubor s vkládanou zprávou.

-sf definuje vystupní soubor, pokud tento přepínač není použit pro výstup se použije vstupní soubor.

Další parametry které lze použít

-Z vkládaná data nebudou komprimována.

-K do výstupního obrazu se nebude vkládat CRC32 kontrolní součet, tento přepínač lze použít, pokud zpráva již obsahuje nějaký druh kontrolního součtu nebo pokud potřebujete uvolněných 32 bitů použít pro vložení zprávy.

-N do výstupního obrazu se nebude ukládat jméno souboru s vkládanou zprávou, při použití tohoto přepínače bude nutné pro extrahování zprávy určit jméno, do kterého se uloží zpráva.

1.13.2 Získávání dat

$ steghide extract -sf vystupniobraz.jpg.

Enter passphrase:

wrote extracted data to " zprava.txt ".

-sf definuje soubor obsahující skrytou informaci

Další parametry, které lze použít:

-xf definuje jméno výstupního souboru, tento přepínač je nutné použít pokud se při vkládání zamezilo vložení jména souboru se zprávou .

1.14 Příprava trénovacích množin

Pro přípravu všech vstupních souborů, vložení textu do obrázku, export výpisů z programu JPEG snoop a jejich transformaci do podoby trénovacích množin jsem napsal řadu skriptů, které většinu ze zmíněných kroků plně automatizovaly.

Pro potřeby trénovacích množin jsem použil bezmála 13500 fotek, které byly uloženy v nejvyšší možné kvalitě a v následujících rozlišeních:

1. 2592x1944
2. 2048x1536
3. 3872x2592
4. 1280x1024
5. 1024x768

První tři skupiny rozlišení obrázků jsou fotky tak, jak byly uloženy fotografickým přístrojem a jsou bez jakýchkoliv dalších úprav. Pro zbylé dvě skupiny rozlišení jsem použil následující skript.

1.14.1 Skript na změnu velikosti

```
#!/bin/bash

for x in cista*.jpg

do

convert -geometry 1024x768 $x zmenseniny1024_768/$x
```

```
convert -geometry 1280x1024 $x zmenseniny1280_1024/$x
done
```

Skript načte všechny soubory odpovídající výběrovému pravidlu. V tomto případě jsou to všechny soubory mající na začátku jména „cista“ a typ souboru je jpeg. Ty pak dosazuje do příkazů jako proměnnou „**x“**.

1.14.2 Skript na zakódování textu do obrázků

1.14.2.1 outguess

```
#!/bin/bash
for x in cista*.jpg
do
dd if=/dev/urandom bs=128 count=1 |uuencode /dev/stdout >4hidden.txt
number=$RANDOM
echo $number
let vysledek=(number+number)*2
let vysledek2=1,5*vysledek
heslo=$vysledek2$vysledek$number
outguess -k "$heslo" -d 4hidden.txt $x zakodovana_$x
done
for x in zakodovana*.jpg
do
rename 's/zakodovana_cista/zakodovana/' *
done
```

1.14.2.2 steghide

```
#!/bin/bash
for x in cista*.jpg
do
dd if=/dev/urandom bs=128 count=1 |uuencode /dev/stdout > 4hidden.txt
number=$RANDOM
echo $number
let vysledek=(number+number)*2
```

```
let vysledek2=1,5*vysledek
heslo=$vysledek2$vysledek$number
steghide embed -p $heslo -cf $x -ef 4hidden.txt -sf st_zakodovana_$x
done
for x in steghide*.jpg
do
rename 's/st_zakodovana_cista/zakodovana/' *
done
```

Skript funguje obdobně jako ten předešlý jen s s tím, že se zde generuje náhodná zpráva pomocí generátoru pseudonáhodných čísel o délce 128 bitů a výsledek je zapsán do souboru 4hidden.txt, to zaručuje unikátnost zprávy vkládané do každého obrázku.

dd if=/dev/urandom bs=128 count=1 |uuencode /dev/stdout > 4hidden.txt

Dále se generuje heslo pro zabezpečení zprávy u Steghidu se zpráva šifruje pomocí algoritmu Rijndael se 128 bitovou délkou klíče. U OutGuessu není šifrovací mechanizmus znám.

$RANDOM je systémová proměnná, která při každém volání vrací náhodné celé číslo v rozmezí 0 – 32767, čímž jsem pomocí jednoduchých operací získal nové heslo pro každý soubor.

1.14.3 Skript pro extrahování dat z JPEG

FOR %%F IN (\\NAS\dokumenty\skola\UTB\Diplomka\fotky*.jpg) DO (\\NAS\dokumenty\skola\UTB\Diplomka\programky\JPEGsnoop.exe -i %%F -o %%F.txt -nogui).

Program JPEGsnoop je bohužel napsán pro systém Microsoft Windows a o portu na GNU/Linux se pouze uvažuje, z toho důvodu bylo potřeba vytvořit skript pro export výpisů právě pro příkazovou řádku MS Windows 2000/XP/Vista. Skript je natolik transparentní, že ho nebudu dále nijak popisovat.

1.14.4 Skript generující tréninkové množiny

Skript transformující export z programu JPEG snoop na trénovací množiny neuronové sítě.

#!/bin/bash

```
for x in cista*.jpg.txt
do
echo "----soubor" $x >>
cat ./$x |cut -c7-50|grep ^#|head -n16|tail -n16|head -n10|cut -c38-40|paste -s -d " " |tr -s " "   >> ./6vystup_tmp_ciste.txt
cat ./$x |cut -c7-50|grep ^#|head -n32|tail -n16|head -n10|cut -c38-40|paste -s -d " " |tr -s " " >> ./6vystup_tmp_ciste.txt
cat ./$x |cut -c7-50|grep ^#|head -n48|tail -n16|head -n10|cut -c38-40|paste -s -d " " |tr -s " " >> ./6vystup_tmp_ciste.txt
cat ./$x |cut -c7-50|grep ^#|head -n64|tail -n16|head -n10|cut -c38-40|paste -s -d " " |tr -s " " >> ./6vystup_tmp_ciste.txt
done

for x in zakodovana*.jpg.txt
do
echo "----soubor" $x
cat ./$x |cut -c7-50|grep ^#|head -n16|tail -n16|head -n10|cut -c38-40|paste -s -d " " |tr -s " " >> ./6vystup_tmp_zakodovane.txt
cat ./$x |cut -c7-50|grep ^#|head -n32|tail -n16|head -n10|cut -c38-40|paste -s -d " " |tr -s " " >> ./6vystup_tmp_zakodovane.txt
cat ./$x |cut -c7-50|grep ^#|head -n48|tail -n16|head -n10|cut -c38-40|paste -s -d " " |tr -s " " >> ./6vystup_tmp_zakodovane.txt
cat ./$x |cut -c7-50|grep ^#|head -n64|tail -n16|head -n10|cut -c38-40|paste -s -d " " |tr -s " " >> ./6vystup_tmp_zakodovane.txt
done

y=`cat ./6vystup_tmp_ciste.txt |wc -l`
x=4
until [ "$x" -eq "$y" ]
do
cat ./6vystup_tmp_ciste.txt |head -n$x|tail -n4|paste -s -d " " |tr -s " "|awk {'print "{"$0"}" '}|sed -e 's/{ /{/'|sed -e 's/ /,/g' >> ./7vystup_zkraceny_ciste.txt
cat ./6vystup_tmp_zakodovane.txt |head -n$x|tail -n4|paste -s -d " " |tr -s " "|awk {'print "{"$0"}" '}|sed -e 's/{ /{/'|sed -e 's/ /,/g' >> ./7vystup_zkraceny_zakodovane.txt
x=$(($x+4))
done
exit 0
```

1.15 Trénovací množiny

Vzorek trénovací množiny pěti původních obrázků

{0,3,63,15,10,5,3,2,0,0,0,48,22,12,8,6,4,1,0,0,0,50,11,17,10,4,4,2,1,1,0,39,21,18,10,6,2,2,1,0}

{0,1,39,20,20,14,5,1,0,0,0,32,21,21,15,8,2,0,0,0,0,50,9,18,10,4,4,2,1,1,0,36,20,20,12,7,2,2,1,1}

{0,4,70,15,8,3,1,0,0,0,0,46,22,18,9,3,1,0,0,0,0,49,10,18,10,4,4,2,1,1,0,38,20,20,10,6,2,2,1,0}

{0,4,77,11,5,2,1,0,0,0,0,48,24,17,6,2,1,0,0,0,0,52,5,18,11,6,3,2,1,1,0,41,22,19,8,6,1,2,1,0}

{0,1,32,18,19,17,10,2,0,0,0,23,17,19,17,12,7,4,1,0,0,46,12,18,10,3,5,2,1,1,0,34,19,20,15,6,2,2,1,0}

Vzorek trénovací množiny pěti obrázků obsahující skrytou informaci

{0,2,50,18,13,7,5,4,1,0,0,54,13,8,8,8,7,2,0,0,0,52,12,17,10,2,3,2,1,1,0,38,12,19,15,8,2,2,2,1}

{0,0,25,16,20,19,14,5,1,0,0,24,16,19,17,14,7,2,0,0,0,47,12,18,11,3,4,2,1,1,0,34,13,20,18,8,3,2,2,1}

{0,4,60,17,12,5,1,0,0,0,0,46,18,16,12,6,2,0,0,0,0,49,13,18,10,3,3,1,1,0,0,40,13,19,14,7,2,2,2,1}

{0,4,68,14,8,4,2,1,0,0,0,47,23,12,7,5,4,1,0,0,0,57,6,18,10,4,2,1,1,0,0,44,15,19,11,7,1,2,1,0}

{0,1,19,15,20,23,7,6,0,0,0,14,12,9,20,16,12,8,2,0,0,39,13,19,13,2,7,3,1,1,0,32,16,19,17,7,3,2,3,1}

1.16 Porovnání obrázků

Na obrázcích (Obrázek 10, Obrázek 11, Obrázek 12) jsem se rozhodl demonstrovat účinnost steganografických programů a nabídl jsem možnost porovnat vzorky obrázků programů OutGuess a Steghide s původním obrázkem.

Obrázek 10 – obrázek bez skryté informace

Obrázek 11 – obrázek obsahující text vložený programem OutGuess

Obrázek 12 – obrázek obsahující text vložený programem Steghide

1.17 Použití neuronových sítí

Nejprve jsem testoval grafiku upravenou programem OutGuess, pak Steghide i OutGuess dohromady, ale kvůli neuspokojivým výsledkům, byl Steghide testován také samostatně. Použil jsem síť s dopředným šířením chyby se dvěma skrytými vrstvami, s jednou skrytou vrstvou a síť RBF. Trénovací množina byla složena z 1000 vzorků od každého rozlišení, dohromady tedy čistá trénovací množina měla 3000 vzorků. Na testování byl použit zbytek, tedy 366 vzorků od každého rozlišení, dohromady 1098. Totéž platilo pro jednotlivé zakódované – OutGuess pro učení – 3000 vzorků, pro testování 1098, Steghide také.

V případě učení obou najednou, tedy čistých vzorků bylo 3000 a upravených nespecifikovanou technikou 6000.

Délka vektoru byla 40, tedy i počet vstupů do neuronové sítě byl 40.

Klasifikace probíhala do dvou tříd – čisté (hodnota 0 na výstupu ze sítě) a upravené (hodnota 1).

Nastavení neuronové sítě dle vzorců [9]:

pro síť se dvěma skrytými vrstvami

$$N_{skryt1} = N_{vystup}\left(\sqrt[3]{\frac{N_{vstup}}{N_{vystup}}}\right)^2 \tag{4}$$

$$N_{skryta2} = N_{vystup}\sqrt[3]{\frac{N_{vstup}}{N_{vystup}}} \tag{5}$$

kde Nskryta1 je počet neuronů ve skryté vrstvě 1

Nskryta2 je počet neuronů ve skryté vrstvě 2

Nvystup je počet neuronů ve výstupní vrstvě, tedy 1

Nvstup je počet neuronů ve vstupní vrstvě, tedy 40

Nskryt1 = 12

Nskryt2 = 4

pro síť s jednou skrytou vrstvou

$$N_{skryt} = \sqrt{N_{vystup} N_{vstup}} \tag{6}$$

kde Nskryta je počet neuronů ve skryté vrstvě

Nvystup je počet neuronů ve výstupní vrstvě, tedy 1

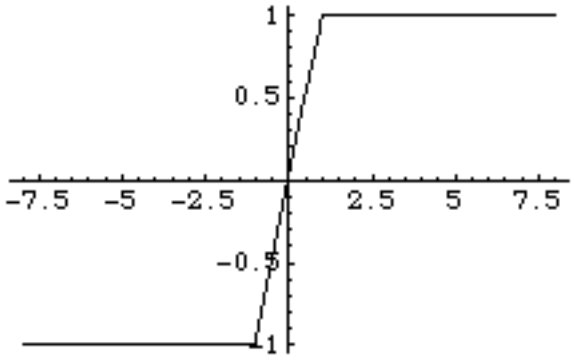
Nvstup je počet neuronů ve vstupní vrstvě, tedy 40

Nskryt = 7

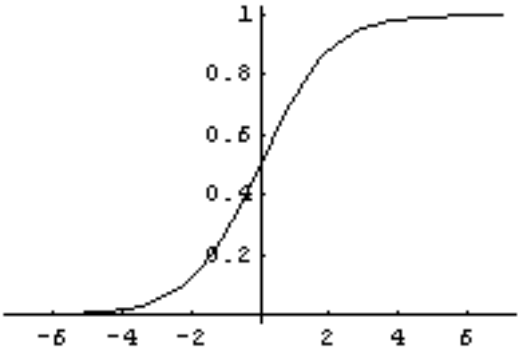
pro síť RBF jsem zvolil počet neuronů ve skryté vrstvě 15.

Počet trénovacích epoch = 30

Typ přenosové funkce v neuronech skryté vrstvy – lineární saturovaný, ve výstupním neuronu je přenosová funkce sigmoida – viz Obrázek 13 a Obrázek 14.

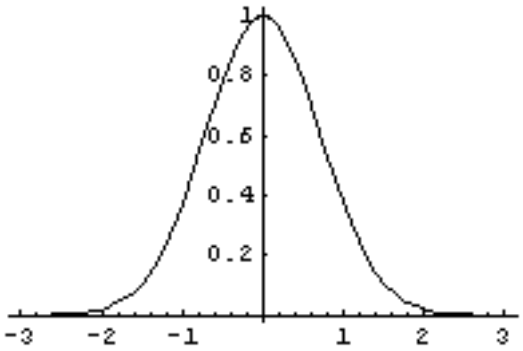


Obrázek 13 - lineární saturovaná sigmoida



Obrázek 14 - sigmoida

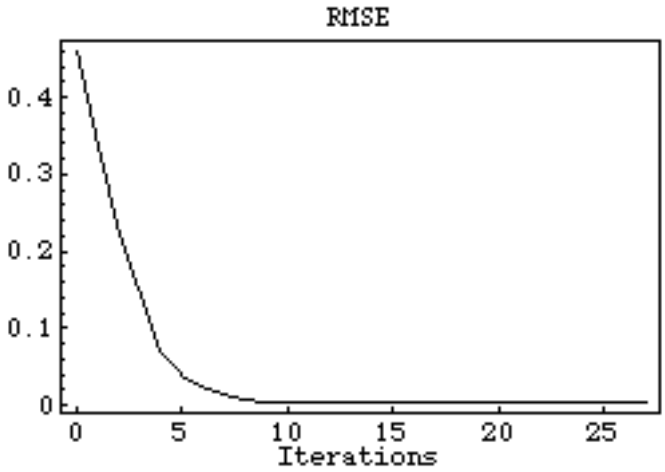
u RBF jsem použil defaultní funkci (Obrázek 15) ve skryté vrstvě a také sigmoidu ve výstupním neuronu.



Obrázek 15 – RBF funkce

1.17.1 OutGuess se sítí se dvěma skrytými vrstvami

Obrázek 16 zobrazuje vývoj chyby při učení neuronové sítě. Osa x – učící epochy, osa y RMSE (root-mean-square error) = odmocnina z průměru čtvercových odchylek u každého prvku trénovací množiny.



Obrázek 16 - vývoj chyby při učení sítě se dvěma skrytými vrstvami

Nejdůležitějším parametrem každé naučené sítě jsou její váhy. Jak takové váhy vypadají jsou na následujících obrázcích.

První vrstva vah (Obrázek 17) je matice 41 x 12 (tedy 40 vstupu plus jeden práh do každého neuronu ve vyšší vrstvě)

(Druhy) Out[26]//MatrixForm=

0.0367688	-0.0166827	0.0254287	0.133869	-0.0638958	0.0313052	-0.0303499	0.0063854	-0.213778	0.0139874	0.0594525	-0.0270772
0.0559227	-0.0963852	-0.0280846	-0.0517141	-0.0621568	0.0390861	0.0311706	0.00156727	0.0200302	-0.0280143	-0.044168	-0.0341996
-0.0265766	0.02887	-0.0116843	0.0443538	-0.0541425	-0.0229879	0.00247773	0.0373212	0.0219364	-0.107326	0.018625	-0.0359746
0.0993253	0.0190071	-0.0105247	-0.044716	-0.00994164	0.0425993	-0.0258793	-0.0826721	0.0107614	-0.144976	0.00526972	-0.0554836
-0.0341998	0.00418326	-0.0309483	0.0139706	0.0335112	-0.0246969	0.054686	0.282043	-0.0817446	0.110918	-0.0800116	0.0786588
-0.0586006	-0.0448435	0.0307238	0.084197	-0.014419	-0.0385493	-0.0407401	0.117081	0.359507	-0.0151382	0.00159483	-0.186677
0.016203	-0.0271378	-0.00510436	0.0120654	0.0274319	-0.103189	0.0259394	-0.102256	0.49585	0.02347	0.0253717	-0.181578
0.0428871	-0.0376918	0.0480081	0.0139847	-0.012835	0.041248	0.036443	-0.111523	0.128844	0.0696418	-0.0218848	-0.0560741
0.0370134	0.011798	0.0524135	0.0170643	-0.0420696	0.0131823	-0.0299383	-0.0143287	-0.0295226	-0.015837	0.000895141	0.0285925
-0.00504602	-0.0628893	0.00853424	-0.0224808	-0.0485642	-0.0167945	0.0215925	-0.0483094	0.00142015	-0.0140977	-0.0369493	0.0726425
0.0260871	0.0349348	-0.00486808	0.0395205	0.0147878	-0.100691	-0.0187877	0.00749369	-0.0401853	0.0872781	0.0125392	-0.054208
0.0151425	0.00313351	-0.0324189	0.0165891	0.0241382	-0.029299	0.0194257	0.0995708	-0.037664	0.061231	-0.0437033	0.00118679
0.0332278	-0.0487199	0.0246772	0.00455346	-0.122726	0.0268962	-0.0370782	-0.0909657	-0.0229738	0.0919951	-0.0694188	-0.0164294
0.0553922	0.00248706	0.0421586	0.0225917	-0.0694221	-0.0444609	0.0250696	0.115338	0.00640911	-0.153175	-0.0116711	0.129331
-0.0437688	0.0323631	0.0129541	0.0476959	-0.0640106	0.09969	0.050287	0.0697533	-0.12162	-0.184638	0.0527529	-0.0342135
0.00665038	-0.0071616	0.0659364	0.041581	-0.00778014	-0.0106477	-0.0175503	0.0422408	0.0981937	0.0331683	-0.0648904	0.138944
-0.0141209	0.050869	-0.00767595	0.00493566	-0.0370825	-0.0796935	-0.00903824	0.00585558	0.255006	0.0452901	0.00839546	-0.0364457
-0.0326673	0.0624454	0.00885618	-0.00535248	0.0443987	0.0148291	-0.0307701	-0.0430833	0.0429392	0.0379401	-0.0563589	0.000740509
0.0369481	0.0383843	0.0156599	-0.0131601	0.011928	0.0314927	-0.0285939	0.0399758	0.0228645	-0.0414845	0.0160195	0.0483882
-0.00226936	-0.0255967	-0.061818	-0.00329403	0.0189092	0.0452739	0.0444688	-0.072125	-0.0125367	-0.0300021	0.0391952	-0.0110593
0.0184182	-0.00536849	0.0655889	-0.0590071	0.0716227	0.15073	0.0846388	-0.0451447	-0.0698588	0.0121269	-0.043734	0.00501102
0.0297331	-0.00979258	-0.0364994	0.137143	-0.106154	0.0951263	0.0113571	0.105853	-0.00413302	0.0397427	0.00275139	0.0384196
-0.0592227	0.0204439	0.00423892	0.0277516	-0.0693796	0.0099365	0.0143691	-0.140292	-0.00437923	-0.13896	0.0163613	0.00351554
-0.0220461	0.0309065	0.00256779	0.0791029	-0.0841328	-0.00463346	-0.0415503	-0.0115312	-0.0180008	-0.0893404	0.0306058	0.145039
-0.0258425	-0.0381892	-0.0184693	0.00739482	-0.0466915	-0.102451	-0.0324961	0.273082	0.0558765	-0.0173581	0.0557103	-0.0433396
0.0490164	0.0461585	-0.0140266	0.0623676	0.0280579	-0.0460741	-0.0523389	0.0508171	0.0123258	0.0187174	-0.0272581	0.00710263
-0.0457337	-0.0112915	-0.0410291	0.0357804	0.0686562	0.048729	-0.00839779	-0.0552303	0.351099	0.0141914	0.0154951	-0.195498
0.0341767	-0.0129778	-0.022428	-0.00742368	0.0486754	0.188763	-0.036232	-0.130939	0.163924	0.0520046	0.0355271	0.0873057
0.00935602	0.0514505	0.0294993	0.0191849	0.0174788	-0.0378008	-0.0407423	-0.0146559	-0.0788124	0.0174105	0.0175561	0.094391
-0.0444386	0.0908607	-0.0105544	0.0535314	-0.0365436	0.0418839	-0.0323504	0.0696298	-0.0747322	0.0687874	-0.0199226	0.18192
0.0860028	0.046864	0.0104248	0.182157	-0.0324755	-0.0245867	0.0400792	-0.0552788	0.0709842	0.179758	0.0307644	0.0985764
-0.0269369	0.0144958	-0.0149556	-0.0610296	-0.050225	0.0239664	-0.0266358	-0.201055	0.0552362	0.151179	0.0243367	0.0579758
0.0349719	0.0392094	-0.022835	-0.0543415	-0.0498841	0.00442501	0.0360199	-0.0752335	0.0415237	0.036223	-0.0399956	-0.08125
-0.00271315	0.0726485	0.033352	0.0276581	-0.0360416	-0.00969834	0.00922448	0.0516545	0.0826064	0.0346019	-0.0622818	0.146542
0.0453485	0.0228344	0.0266098	0.0505199	-0.00486204	-0.0759401	0.0207916	0.0524283	0.152747	0.00657596	0.0271445	-0.115741
-0.0370635	-0.026654	0.0247059	0.0439251	0.0308552	-0.0106706	-0.0348458	-0.100831	0.0595338	0.0749704	-0.0451806	0.147092
-0.0296158	0.0591607	-0.031312	0.00724961	0.0293772	0.0753015	-0.0493679	-0.011507	0.0948511	-0.145077	0.0113211	-0.134927
-0.04682	-0.0130013	0.0205082	-0.0353329	-0.0287405	-0.0469849	0.0247022	0.117057	0.00632681	-0.0559656	0.0244169	-0.0203724
0.0113357	0.0508884	-0.00714122	-0.0303356	0.028117	0.0663434	-0.0357398	0.03859	-0.0816615	-0.0672801	0.00243118	0.112003
-0.000517234	0.0452642	0.0279603	0.0258332	-0.0223341	-0.0351065	-0.0437508	-0.107761	-0.257032	0.0642032	-0.024888	0.231481
-10.3308	-9.99085	-4.12086	-4.60406	9.76117	-3.0144	-2.56398	-5.3382	-6.23418	-1.43016	9.64009	-5.1386

Obrázek 17 - První vrstva vah

Druhá vrstva vah (Obrázek 18) má matici 13 x 4 (12 neuronů ve skryté vrstvě plus práh do každého neuronu ve skryté vrstvě 2)

(Druhy) Out[27]//MatrixForm=

2.50439	1.10237	0.210116	1.78377
1.05457	-0.879703	0.253388	0.542074
-0.57794	-0.060545	-1.46358	1.6246
0.742042	0.598753	-1.01865	-1.11878
0.636629	0.0586152	0.0935107	0.0377781
-2.08165	0.835681	-1.37281	0.733439
-1.40205	1.03495	0.984319	1.358
-0.613871	0.401051	-1.55765	-1.18696
-1.81907	2.45892	3.93452	-1.48456
0.239052	-0.623083	1.3077	1.68504
1.66371	1.80791	-1.45756	-0.157794
1.33978	-2.32455	3.20958	0.958776
-2.69805	0.410733	0.199867	1.18289

Obrázek 18 - Druhá vrstva vah

Třetí vrstva vah (Obrázek 19) má matici 5 x 1 (4 neurony plus jeden práh do výstupního neuronu)

(Druhy) Out[28]//MatrixForm=

$$\begin{pmatrix} 3.02044 \\ -3.30274 \\ 1.31571 \\ -0.121381 \\ -0.991697 \end{pmatrix}$$

Obrázek 19 – Třetí vrstva vah

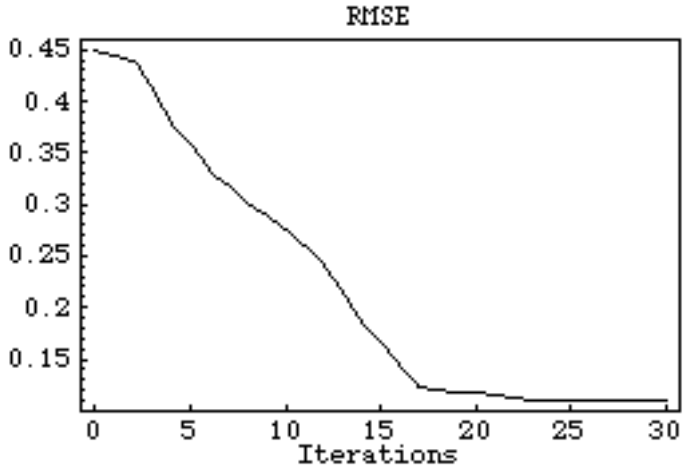
Při prohledávání odpovědí aplikované sítě na testovací množiny bylo objeveno, že v případě čisté množiny se síť v jednom případě netrefila, resp. odpověď sítě byla v rozmezí 0.5 až 1, přesně 0.975671. U zakódované množiny byla 100 procentní úspěšnost.

Tabulka 4 – Výsledky detekce programu OutGuess se sítí se dvěma skrytými vrstvami

	Počet špatně zařazených případů (chyb)	Procentuální chybovost	Procentuální úspěšnost (100 – procentuální chybovost)
Čistá	1	0.0910747	99.9089
Zakódovaná	0	0	100

1.17.2 OutGuess se sítí s jednou skrytou vrstvou

Obrázek 20 zobrazuje vývoj chyby při učení neuronové sítě s jednou skrytou vrstvou. Osa x – učící epochy, osa y RMSE (root-mean-square error) = odmocnina z průměru čtvercových odchylek u každého prvku trénovací množiny.



Obrázek 20 - vývoj chyby při učení sítě s jednou skrytou vrstvou

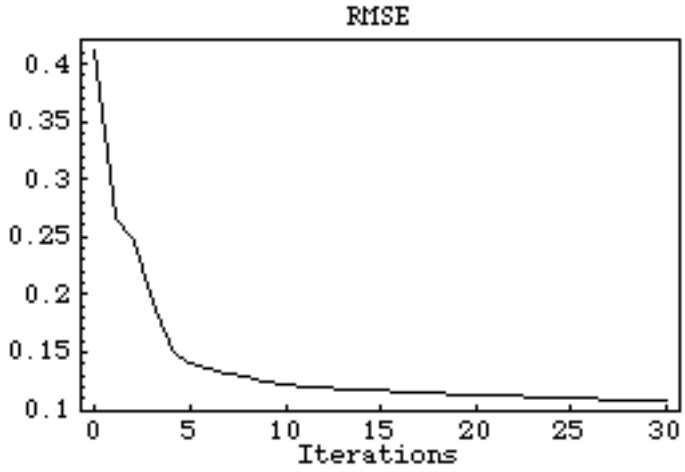
Při prohledávání odpovědí aplikované sítě na testovací množiny bylo objeveno, že v případě čisté množiny se síť v osmi případech netrefila, resp. odpověď sítě byla v rozmezí 0.5 až 1, přesně to byly následující hodnoty {0.697057, 0.721635, 0.634685, 0.891021, 0.543743, 0.586112, 0.566472, 0.622812}. U zakódované množiny byla 100 procentní úspěšnost.

Tabulka 5 - Výsledky detekce programu OutGuess se sítí s jednou skrytou vrstvou.

	Počet špatně zařazených případů (chyb)	Procentuální chybovost	Procentuální úspěšnost (100 – procentuální chybovost)
Čistá	8	0.728597	99.2714
Zakódovaná	0	0	100

1.17.3 OutGuess se sítí RBF

Obrázek 21 zobrazuje opět vývoj chyby při učení neuronové sítě. Osa x – učící epochy, osa y RMSE (root-mean-square error) = odmocnina z průměru čtvercových odchylek u každého prvku trénovací množiny.



Obrázek 21 - vývoj chyby při učení RBF sítě

Při prohledávání odpovědí aplikované sítě na testovací množiny bylo objeveno, že v případě čisté množiny síť v devíti případech provedla nekorektní klasifikaci, resp. odpověď sítě byla v rozmezí 0.5 až 1. Hodnoty v tomto případě byly následující {0.690727, 0.672197, 0.598242, 0.537289, 0.850413, 0.555159, 0.570557, 0.568398, 0.51539}. U zakódované množiny se objevil jeden případ špatně zařazeného obrázku s hodnotou {0.479888}.

Tabulka 6 - Výsledky detekce programu OutGuess se sítí RBF

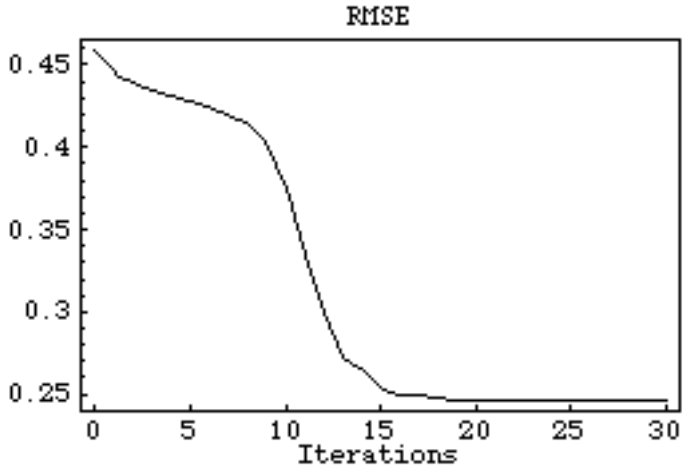
	Počet špatně zařazených případů (chyb)	Procentuální chybovost	Procentuální úspěšnost (100 – procentuální chybovost)
Čistá	9	0.819672	99.1803
Zakódovaná	1	0.0910747	99.9089

1.17.4 Minizávěr pro odhalování steganografie u programu OutGuess

Síť se dvěma skrytými vrstvami se chovala nejlépe. Dosáhla 99.9 procentuální úspěšnosti klasifikace, dále pak je nutno zdůraznit, že síť dosáhla naučeného stavu již při deseti iteracích. Vzhledem k této úspěšnosti, další testy budou provedeny s OutGuess i Steghide dohromady.

1.17.5 OutGuess i Steghide se sítí se dvěma skrytými vrstvami

Obrázek 22 zobrazuje vývoj chyby při učení neuronové sítě. Osa x – učící epochy, osa y RMSE (root-mean-square error) = odmocnina z průměru čtvercových odchylek u každého prvku trénovací množiny.



Obrázek 22 - vývoj chyby při učení sítě se dvěma skrytými vrstvami

Tabulka 7 - Výsledky detekce programů OutGuess i Steghide se sítí se dvěma skrytými vrstvami

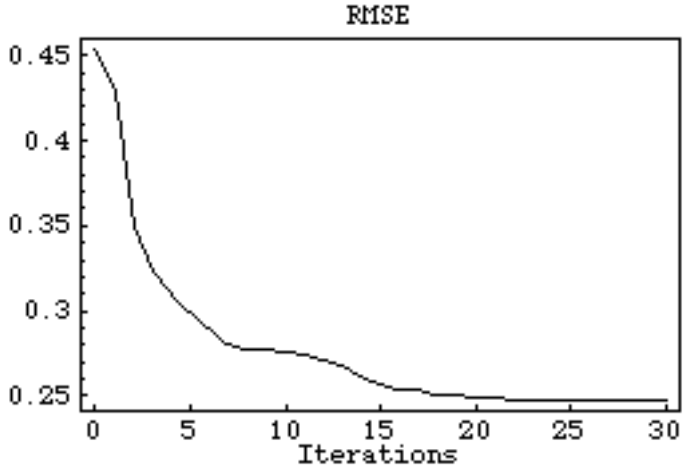
	Počet špatně zařazených případů (chyb)	Procentuální chybovost	Procentuální úspěšnost (100 – procentuální chybovost)
Čistá	360	32.7869	67.2131
Zakódovaná OutGuess	0	0	100

Zakódovaná Steghide	10	0.910747	99.0893

Při prohledávání odpovědí aplikované sítě na testovací množiny bylo objeveno, že v případě čisté množiny síť v 360 z 1098 případů případech provedla nekorektní klasifikaci, resp. odpověď sítě byla v rozmezí 0.5 až 1. U zakódované množiny máme tentokrát zvlášť testovací množinu vytvořenou programem OutGuess, kde síť provedla korektní klasifikaci ve 100 procentech správně a množinu zakódovanou programem Steghide, kde síť v 10 případech provedla nekorektní klasifikaci, neboli odpověď byla v rozmezí 0 – 0.5.

1.17.6 OutGuess i Steghide se sítí s jednou skrytou vrstvou

Obrázek 23 zobrazuje vývoj chyby při učení neuronové sítě. Osa x – učící epochy, osa y RMSE (root-mean-square error) = odmocnina z průměru čtvercových odchylek u každého prvku trénovací množiny.



Obrázek 23 - vývoj chyby při učení sítě s jednou skrytou vrstvou

Při prohledávání odpovědí aplikované sítě na testovací množiny bylo objeveno, že v případě čisté množiny síť v 363 z 1098 případů provedla nekorektní klasifikaci, resp. odpověď sítě byla v rozmezí 0.5 až 1. U zakódované množiny máme tentokrát zvlášť

testovací množinu vytvořenou programem OutGuess, kde síť provedla korektní klasifikaci ve 100 procentech správně a množinu zakódovanou programem Steghide, kde síť v 9 případech provedla nekorektní klasifikaci, neboli odpověď byla v rozmezí 0 – 0.5.

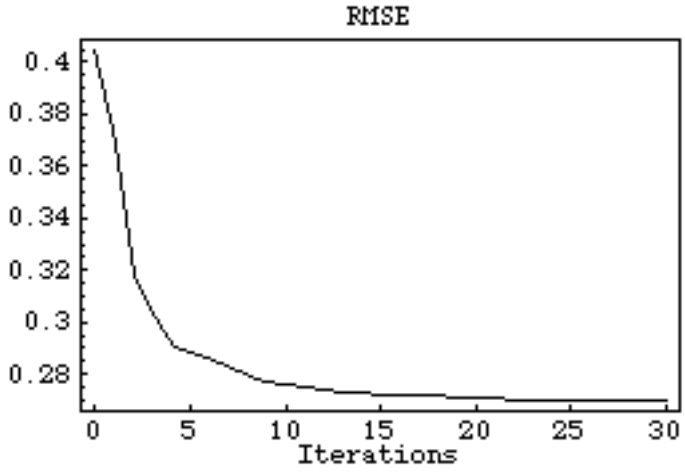
Tabulka 8 - Výsledky detekce programů OutGuess i Steghide se sítí s jednou skrytou vrstvou

	Počet špatně zařazených případů (chyb)	Procentuální chybovost	Procentuální úspěšnost (100 – procentuální chybovost)
Čistá	363	33.0601	66.9399
Zakódovaná OutGuess	0	0	100
Zakódovaná Steghide	9	0.819672	99.1803

1.17.7 OutGuess i Steghide se sítí RBF

Obrázek 24 zobrazuje vývoj chyby při učení neuronové sítě. Osa x – učící epochy, osa y RMSE (root-mean-square error) = odmocnina z průměru čtvercových odchylek u každého prvku trénovací množiny.

Při prohledávání odpovědí aplikované sítě na testovací množiny bylo objeveno, že v případě čisté množiny síť v 254 z 1098 případů provedla nekorektní klasifikaci, resp. odpověď sítě byla v rozmezí 0.5 až 1. U zakódované množiny máme tentokrát zvlášť testovací množinu vytvořenou programem OutGuess, kde síť provedla korektní klasifikaci ve 100 procentech případů a množinu zakódovanou programem Steghide, kde síť ve 120 případech provedla nekorektní klasifikaci, neboli odpověď byla v rozmezí 0 – 0.5.



Obrázek 24 - vývoj chyby při učení RBF sítě

Tabulka 9 - Výsledky detekce programů OutGuess i Steghide se sítí RBF

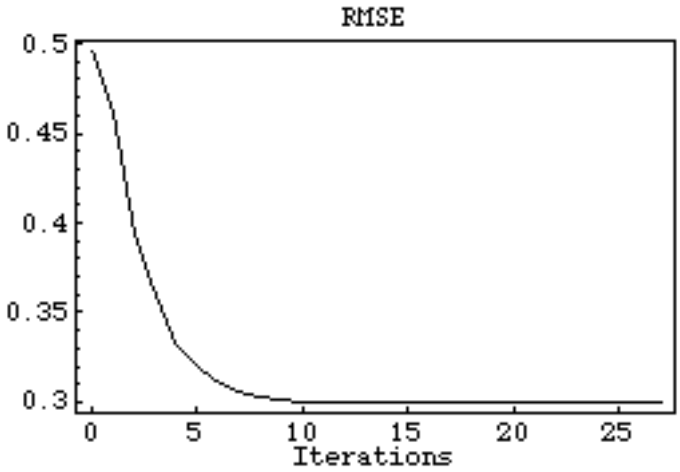
	Počet špatně zařazených případů (chyb)	Procentuální chybovost	Procentuální úspěšnost (100 – procentuální chybovost)
Čistá	254	23.133	76.867
Zakódovaná OutGuess	0	0	100
Zakódovaná Steghide	120	10.929	89.071

1.17.8 Minizávěr ke klasifikaci s použitím obou programů

V porovnání s detekcí samotného programu OutGuess, se zde objevily problémy a neúspěch při detekci čistých i zakódovaných obrázku pomocí Steghide. Proto následující testy byly provedeny pouze na trénovací množině připravené programem Steghide.

1.17.9 Steghide se sítí se dvěma skrytými vrstvami

Obrázek 25 zobrazuje vývoj chyby při učení neuronové sítě. Osa x – učící epochy, osa y RMSE (root-mean-square error) = odmocnina z průměru čtvercových odchylek u každého prvku trénovací množiny.



Obrázek 25 - vývoj chyby při učení sítě se dvěma skrytými vrstvami

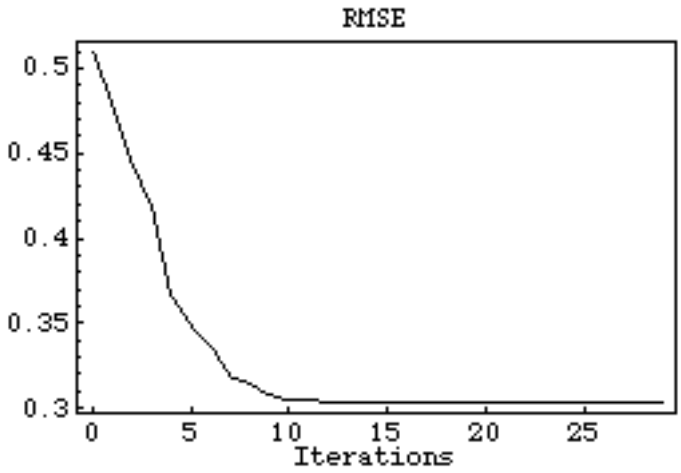
Při prohledávání odpovědí aplikované sítě na testovací množiny bylo objeveno, že v případě čisté množiny se síť ve 130 případech provedla nekorektní klasifikaci, resp. odpověď sítě byla v rozmezí 0.5 až 1. U zakódované množiny se objevilo 244 případů špatně zařazeného obrázku, tedy v rozmezí 0 – 0.5.

Tabulka 10 - Výsledky detekce programu Steghide se sítí se dvěma skrytými vrstvami

	Počet špatně zařazených případů (chyb)	Procentuální chybovost	Procentuální úspěšnost (100 – procentuální chybovost)
Čistá	130	11.8397	88.1603
Zakódovaná	244	22.2222	77.7778

1.17.10 Steghide se sítí s jednou skrytou vrstvou

Obrázek 26 zobrazuje vývoj chyby při učení neuronové sítě. Osa x – učící epochy, osa y RMSE (root-mean-square error) = odmocnina z průměru čtvercových odchylek u každého prvku trénovací množiny.



Obrázek 26 - vývoj chyby při učení sítě s jednou skrytou vrstvou

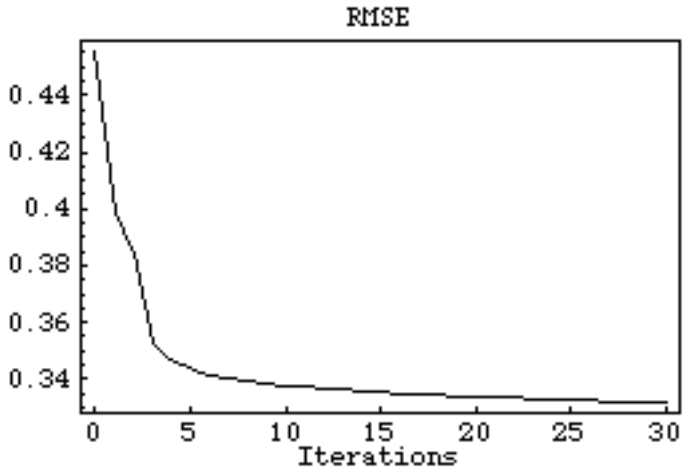
Při prohledávání odpovědí aplikované sítě na testovací množiny bylo objeveno, že v případě čisté množiny se síť v 83 případech provedla nekorektní klasifikaci, resp. odpověď sítě byla v rozmezí 0.5 až 1. U zakódované množiny se objevilo 284 případů špatně zařazeného obrázku .

Tabulka 11 - Výsledky detekce programu Steghide se sítí s jednou skrytou vrstvou

	Počet špatně zařazených případů (chyb)	Procentuální chybovost	Procentuální úspěšnost (100 – procentuální chybovost)
Čistá	83	7.5592	92.4408
Zakódovaná	284	25.8652	74.1348

1.17.11Steghide se sítí RBF

Obrázek 27 zobrazuje vývoj chyby při učení neuronové sítě. Osa x – učící epochy, osa y RMSE (root-mean-square error) = odmocnina z průměru čtvercových odchylek u každého prvku trénovací množiny.



Obrázek 27 - vývoj chyby při učení RBF sítě

Při prohledávání odpovědí aplikované sítě na testovací množiny bylo objeveno, že v případě čisté množiny se síť v 304 případech provedla nekorektní klasifikaci, resp. odpověď sítě byla v rozmezí 0.5 až 1. U zakódované množiny v 67 případech, neboli v rozmezí 0 – 0.5.

Tabulka 12 - Výsledky detekce programu Steghide se sítí RBF

	Počet špatně zařazených případů (chyb)	Procentuální chybovost	Procentuální úspěšnost (100 – procentuální chybovost)
Čistá	304	27.6867	72.3133
Zakódovaná	67	6.102	93.898

1.17.12 Minizávěr ke programu Steghide

Jak se ukázalo, tak problém nebyl v tom, že by neuronová síť nedokázala klasifikovat prvky zakódované oběma programy najednou, ale že se problémy objevily při detekci programu Steghide samotného. Síť s jednou skrytou vrstvou dosáhla nejlepší procentuální úspěšnosti, nicméně zde byla vysoká chybovost, která se pohybovala okolo 7,5 % u neupravených obrázků a téměř 26 % u obrázků upravených programem Steghide.

1.17.13 Shrnutí testování

V rámci testování obrázků zpracovaných programem OutGuess se dosáhlo velmi dobrých výsledků. Celková úspěšnost klasifikace 99.9% při odhalování mě dovedla k myšlence, zařadit do diplomové práce i program Steghide, který pro ukládání využívá jiný algoritmus a vyzkoušet, jak se neuronová síť bude chovat, pokud ji budeme učit podle postupů vyzkoušených s OutGuessem. Výsledky jasně ukazují nedostatky, nicméně dosažená úspěšnost klasifikace 76% u neupravených a téměř 90% u upravených obrázků jasně naznačuje možnosti využívání neuronových sítích pro klasifikaci. Pevně věřím, že pokud bychom trénovací množiny upravili výhradně jen pro potřeby detekce programu Steghide, výsledná úspěšnost by se zvýšila.

ZÁVĚR

Na začátku mého diplomového projektu jsem absolvoval předměty Metody umělé inteligence pod vedením doc. Ing. Ivana Zelinky, Ph.D. , Ing. Zuzana Oplatkové Ph.D. a Počítačová grafika pod vedením Ing. Pavla Pokorného, Ph.D.

Zprvu jsem měl jednoduchý nápad, jak vytvořit detektor zpráv obsahující steganografický obsah, postupem času jak sem se dostával hlouběji do zkoumaného problému, jsem mnohdy uvíznul na mrtvém bodě, který mě vetšinou donutil zahodit veškeré mé výsledky a začít znovu. To platí zejména v případě trénovacích množin pro neuronové sítě. Trvalo mi pár týdnů zkoumání analýz grafického formátu JPEG, než jsem překonal počáteční smůlu a získal indície vedoucí k zajímavým výsledkům, které jsou detailněji popsány v kapitole 4.2.2. Od této chvíle jsem mohl zahájit plnohodnotnou spolupráci s doktorkou Oplatkovou a práce se stala naší zábavou, alespoň tedy tak, jak nám náš volný čas strávený nad projektem dovolil.

V diplomové práci bylo testováno bezmála 13500 fotek získaných z různých fotografických přístrojů a s různým rozlišením obrazu. Pro hromadné vložení textu do obrázků byla použita sadu skriptů napsaných pro operační systém GNU/ Linux, jejichž součástí byl i generátor zpráv generující zprávy o velikosti 128 bitů. K analýze obrázků posloužil program JPEGsnoop, výsledky byly následně transformovány do trénovacích množin neuronových sítí skripty napsaných opět pro GNU / Linux. Při testech byly použity tři druhy sítí - RBF, síť s dopředným šířením chyby a jednou skrytou vrstvou a síť s dopředným šířením chyby a dvěma skrytými vrstvami. Poslední zmíněná sít během testů vykazovala nejlepší výsledky.

Tento projekt byl jedním z nejkomplexnějších, které jsem kdy řešil. Bylo nezbytné začít od nuly v oblastech zpracování digitálních obrázků a klasifikace pomocí neuronových sítí a krok po kroku se přibližovat cílům této diplomové práce. Během zpracování diplomové práce jsem získal mnoho teoretických i praktických znalostí ve výše zmíněných oblastech a jako bonus beru znalosti základů kryptoanalýzy, které bych rád využil při svých dalších projektech.

Věřím že i doktorka Oplatková z naší spolupráce získala nové zkušeností, které jí pomohou při řešení jejích dalších projektů.

CONCLUSION

On a beginning of this master thesis I had passed subjects like Methods of artificial intelligence under leadership of doc. Ing. Ivan Zelinka, Ph.D. , Ing. Zuzana Oplatková Ph.D. and Computer graphics under leadership of Ing. Pavel Pokorný, Ph.D. At the beginning I had a simple idea how to make easy detector of steganographical content by neural networks but by the time as I got into problems deeply I stayed stuck at dead ends. What made me drop all my research and start all over. It stands good especially for part where I trained sets for neural networks after weeks of examination various kinds of JPEG analysis. Since I broke the bad luck and found a glue leading to interesting results as it is shown in chapter 4.2.2. I could start full cooperation in research with doctor Oplatková and work become to be fun at least as our free time let us enjoy that cooperation.

For the purpose of this master thesis nearly 13500 photos were tested which were gained form various digital cameras in various sizes. For batch embedding script designed for operation system GNU/ Linux have been used, part of this script is generator of 128 bit long messages. A program JPEGsnoop was used for image analysis, its results were transformed by another GNU/ Linux script in to the training sets for neural networks. Three types of neural networks have been used during tests - RBF, feedforward neural network with one hidden layer and feedforward neural network with two hidden layers. Last mentioned network had the best results of revealing steganography on tested samples.

This project was the most complex problem I have ever sold. It was necessary start from zero in fields of digital image processing, classification in neural networks and go step by step to meet goals of this thesis. During work I got a lot of theoretical and practical experience in above mentioned field and as a bonus I got into practical cryptoanalysis what I would be happy to use in my future projects.

I trust that doctor Oplatkova is looking same way on our cooperation and she also got usable experience for her own future projects.

SEZNAM POUŽITÉ LITERATURY

[1] VONDRUŠKA, Pavel . Steganografie. Crypto-World : Informační sešit GCUCMP. 2000, č. 9, s. 3-6. Dostupný z WWW: <http://crypto-world.info/casop2/crypto09_00.pdf>.

[2] Steganografie [online]. 2008 [cit. 2008-03-20]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Steganografie>.

[3] Johannes Trithemius [online]. 2008 [cit. 2008-03-20]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Johannes_Trithemius>.

[4] Bernd J., Practical Handbook on Image Processing for Scientific and Technical Applications, CRC 2004, ISBN: 978-0849319006

[5] Pratt W., Digital Image Processing 2007, Wiley, 2007, 4th ed., ISBN:978-0-471-76777-0

[6] BENEŠ, Miroslav. Komprese : Huffmanovo kódování [online]. [cit. 2008-02-15]. Dostupný z WWW:
<http://www.cs.vsb.cz/benes/vyuka/pte/texty/komprese/ch02s02.html>.

[7] Neural Network Theory : Feedforward neural networks. Mathematica : Help

[8] MINDEK, Marian. Radial Basis Function network [online]. 2004 , 15.2.2004 [cit. 2008-04-21]. Dostupný z WWW: <http://homen.vsb.cz/~min038/rbf/>.

[9] Zelinka Ivan. Umělá inteligence I. Volume 1. Zlín : Vutium, Brno, 1998. 126 p. ISBN 80-214-1163-5.

[10] SINGH, Simon. Kniha kódů a šifer : Utajování od starého Egypta po kvantovou kryptografii. P. Koupský, D. Eckhardtová. 1. vyd. Praha : Argo a Dokořán, 2003. 382 s. ISBN 80-86569-18-7.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

DQT	Quantization table
RBF	Radial Basis Function
RMSE	Root-mean-square error
JPEG	Joint Photographic Experts Group
GNU	*GNU's Not Unix*

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: ODPOVĚĎ NA TESTOVACÍ MNOŽINU NEUPRAVENÝCH OBRÁZKŮ

{{0.00279318},{0.245338},{0.00279318},{0.00279318},{0.00234807},{0.000201563},{0.00279318},{0.00134665},{0.00121051},{0.000774347},{0.000910099},{0.000327617},{0.000329529},{0.000217677},{0.000889137},{0.00051044},{0.000201563},{0.00095764},{0.00279318},{0.000201563},{0.000515987},{0.00279318},{0.00279318},{0.000568955},{0.00130732},{0.000201563},{0.00115027},{0.000385874},{0.000201563},{0.000643702},{0.000355676},{0.000385874},{0.00279318},{0.00028563},{0.000994138},{0.000201563},{0.00279318},{0.00271894},{0.000524749},{0.00174913},{0.00254934},{0.00279318},{0.000323323},{0.000201563},{0.00279318},{0.000413048},{0.000201563},{0.000415841},{0.00230993},{0.000371008},{0.00279318},{0.000201563},{0.00279318},{0.000657117},{0.00279318},{0.00279318},{0.000346986},{0.00216068},{0.00279318},{0.00279318},{0.00279318},{0.00022066},{0.000798243},{0.00205401},{0.00279318},{0.000374674},{0.00279318},{0.00128522},{0.000296167},{0.000538575},{0.00225915},{0.00279318},{0.000201563},{0.00279318},{0.000201563},{0.000385874},{0.00279318},{0.00279318},{0.00279318},{0.00279318},{0.00279318},{0.000789381},{0.000201563},{0.000302771},{0.0021357},{0.000598717},{0.000201563},{0.00143309},{0.00279318},{0.000201563},{0.00279318},{0.000907766},{0.00279318},{0.00279318},{0.000201563},{0.00239156},{0.00276319},{0.00279318},{0.000233389},{0.00242373},{0.000201563},{0.00279318},{0.000823929},{0.000201563},{0.000201563},{0.00112113},{0.000201563},{0.00279318},{0.00207876},{0.00118202},{0.000926131},{0.00279318},{0.000201563},{0.000209694},{0.00279318},{0.00279318},{0.000201563},{0.00279318},{0.00279318},{0.000201563},{0.00276319},{0.00279318},{0.000824643},{0.00135569},{0.00279318},{0.000941441},{0.00279318},{0.00238662},{0.00279318},{0.000576993},{0.000705522},{0.000201563},{0.00279318},{0.00279318},{0.000385874},{0.00279318},{0.00133045},{0.000201563},{0.00279318},{0.00044115},{0.00279318},{0.000201563},{0.000761509},{0.00118548},{0.000201563},{0.00204353},{0.00279318},{0.000244451},{0.000283991},{0.00070624},{0.00279318},{0.000201563},{0.000384345},{0.000978503},{0.00059724},{0.000385874},{0.000201563},{0.00279318},{0.000354891},{0.00279318},{0.00279318},{0.00279318},{0.000555722},{0.00259939},{0.000980551},{0.00193588},{0.00279318},{0.000841595},{0.000705055},{0.00279318},{0.000201563},{0.00279318},{0.00075923},{0.00081669},{0.000858239},{0.00279318},{0.00279318},{0.000385874},{0.000631555},{0.00279318},{0.000201563},{0.000201563},{0

.000422525},{0.00197901},{0.00279318},{0.000591962},{0.000201563},{0.000323979}
,{0.00139039},{0.000928589},{0.000611603},{0.000201563},{0.000201563},{0.000642
271},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.0
00409675},{0.00161764},{0.00279318},{0.000201563},{0.000201563},{0.00279318},{0
.000201563},{0.000377769},{0.00041474},{0.000201563},{0.00279318},{0.00279318},
{0.000201563},{0.000201563},{0.000201563},{0.000432682},{0.000982595},{0.000252
386},{0.000201563},{0.000201563},{0.00130724},{0.000201563},{0.00257782},{0.002
79318},{0.00192658},{0.000222562},{0.000201563},{0.00149915},{0.00038306},{0.00
0201563},{0.000201563},{0.000201563},{0.000201563},{0.000967315},{0.000470302},
{0.000209084},{0.00279318},{0.00279318},{0.000201563},{0.000201563},{0.00027948
2},{0.00204446},{0.00279318},{0.00115473},{0.000201563},{0.00249364},{0.0002015
63},{0.00157885},{0.000201563},{0.000744613},{0.00274206},{0.000201563},{0.0027
9318},{0.00279318},{0.00237495},{0.00110485},{0.00279318},{0.000201563},{0.0003
92254},{0.000298071},{0.00210701},{0.00184871},{0.000322521},{0.000351169},{0.0
0279318},{0.00107039},{0.00242056},{0.000385874},{0.000201563},{0.000201563},{0
.00055755},{0.000738729},{0.00128335},{0.000201563},{0.000201563},{0.00279318},
{0.000201563},{0.000201563},{0.00279318},{0.00279318},{0.0020775},{0.00279318},
{0.000201563},{0.000201563},{0.000370193},{0.00279318},{0.00279318},{0.00279318
},{0.00279318},{0.000906095},{0.00127375},{0.000201563},{0.000201563},{0.000201
563},{0.000315019},{0.000201563},{0.000201563},{0.000201563},{0.00279318},{0.00
106132},{0.00236967},{0.0025744},{0.00279318},{0.000653393},{0.000201563},{0.00
0310059},{0.000201563},{0.00279318},{0.000201563},{0.000201563},{0.00279318},{0
.000201563},{0.000201563},{0.00125247},{0.00123741},{0.000761163},{0.000247641}
,{0.000304947},{0.00279318},{0.00279318},{0.0023642},{0.000657941},{0.000385874
},{0.000304477},{0.000664309},{0.000201563},{0.000201563},{0.00274569},{0.00020
1563},{0.00279318},{0.000655909},{0.00033354},{0.000201563},{0.00115876},{0.001
28865},{0.000937902},{0.000201563},{0.000201563},{0.00021695},{0.000201563},{0.
000201563},{0.000300364},{0.00113795},{0.000201563},{0.0022528},{0.000201563},{
0.00279318},{0.00103505},{0.00020953},{0.000201563},{0.000201563},{0.000853256}
,{0.000201563},{0.00279318},{0.000201563},{0.00279318},{0.00279318},{0.00020156
3},{0.00279318},{0.00042874},{0.00279318},{0.00164672},{0.000201563},{0.0005523
83},{0.000337253},{0.00279318},{0.000385874},{0.00116162},{0.000158125},{0.0006
27435},{0.000844812},{0.000201563},{0.000158125},{0.000627435},{0.000200613},{0

.000201563},{0.000201563},{0.000201563},{0.000201563},{0.00054354},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000244658},{0.000201563},{0.000201563},{0.00045587},{0.000201563},{0.000201563},{0.000307324},{0.000201563},{0.000627435},{0.000703403},{0.000201563},{0.000201563},{0.000261586},{0.000201563},{0.000575422},{0.000201563},{0.000201563},{0.000336683},{0.00024761},{0.000554849},{0.000215084},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000238812},{0.000201563},{0.000201563},{0.000201563},{0.000353855},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000544412},{0.000201563},{0.000201563},{0.000201563},{0.000158125},{0.00033465},{0.000198415},{0.000556221},{0.000627435},{0.000158125},{0.000201563},{0.000217458},{0.00279318},{0.000627435},{0.000201563},{0.000201563},{0.000627435},{0.00279318},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.000409337},{0.000627435},{0.0021748},{0.000627435},{0.000627435},{0.000218815},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000282165},{0.000627435},{0.000627435},{0.000201563},{0.000627435},{0.000201563},{0.00111447},{0.000201563},{0.00043474},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000186447},{0.000201563},{0.000391476},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.000197613},{0.000627435},{0.00124762},{0.000627435},{0.128133},{0.000627435},{0.000201563},{0.00091895},{0.00279318},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.000490309},{0.000201563},{0.000744613},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000353722},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.00022159},{0.000201563},{0.00279318},{0.000280221},{0.000201563},{0.000242021},{0.000201563},{0.00051638},{0.000627435},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.00200477},{0.000460835},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.00279318},{0.000312329},{0.00279318},{0.000541644},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000281826},{0.000201563},{0.000367607},{0.000201563},{0.000627435},{0.000232198},{0.000201563},{0.000889914},{0.000744613},{0.00029802

},{0.000983448},{0.000201563},{0.000201563},{0.000310767},{0.000627435},{0.0002
01563},{0.00261408},{0.000201563},{0.00279318},{0.000201563},{0.000201563},{0.0
00627435},{0.000201563},{0.000329773},{0.000201563},{0.000201563},{0.000201563
},{0.000201563},{0.00279318},{0.000201563},{0.000201563},{0.000201563},{0.00033
7111},{0.000201563},{0.000201563},{0.000201563},{0.000258546},{0.000201563},{0.
000201563},{0.000754571},{0.000359976},{0.000201563},{0.000201563},{0.00020156
3},{0.000201563},{0.000379622},{0.000201563},{0.000201563},{0.000201563},{0.000
627435},{0.000627435},{0.000201563},{0.000201563},{0.00279318},{0.000201563},{0
.000627435},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.00020156
3},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000
201563},{0.000201563},{0.00206213},{0.00173854},{0.000627435},{0.000316446},{0.
000627435},{0.00279318},{0.000201563},{0.000201563},{0.000201563},{0.000201563
},{0.00045095},{0.0016373},{0.311097},{0.000201563},{0.000744613},{0.000204709},
{0.00125019},{0.000255183},{0.000201563},{0.000201563},{0.000627435},{0.0002015
63},{0.000440826},{0.975671},{0.000201563},{0.000201563},{0.000201563},{0.00062
7435},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.
000201563},{0.000201563},{0.000201563},{0.000744613},{0.000275678},{0.00020156
3},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000
557222},{0.000221468},{0.000627435},{0.000201563},{0.000201563},{0.000744613},{
0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.0002015
63},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.00
0201563},{0.000627435},{0.000627435},{0.000627435},{0.000201563},{0.000201563},
{0.000266516},{0.000201563},{0.00108165},{0.000627435},{0.000201563},{0.0002015
63},{0.000201563},{0.000336707},{0.00107977},{0.000201563},{0.000235782},{0.000
201563},{0.000201563},{0.000627435},{0.00279318},{0.00065417},{0.000201563},{0.
000201563},{0.000201563},{0.000201563},{0.000235132},{0.000201563},{0.00020156
3},{0.000201563},{0.00279318},{0.000627435},{0.000201563},{0.000627435},{0.0006
27435},{0.000412796},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0
.000627435},{0.00279318},{0.000201563},{0.000201563},{0.000201563},{0.000300644
},{0.00279318},{0.000348443},{0.000201563},{0.000627435},{0.000201563},{0.00020
1563},{0.000158125},{0.000201563},{0.000201563},{0.000201563},{0.000158125},{0.
000627435},{0.000172673},{0.000201563},{0.000247033},{0.000201563},{0.00020156
3},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000

201563},{0.000201563},{0.000201563},{0.000507795},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.000627435},{0.000272904},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.000408954},{0.000201563},{0.000201563},{0.000511925},{0.000627435},{0.000201563},{0.000201563},{0.00279318},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000228575},{0.000201563},{0.000201563},{0.000627435},{0.000168705},{0.000627435},{0.000174268},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000308235},{0.000535042},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000213188},{0.000472919},{0.000201563},{0.000513813},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000816754},{0.00022273},{0.000627435},{0.00021864},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000186315},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.00279318},{0.000201563},{0.000201563},{0.000201563},{0.000614543},{0.000201563},{0.000201563},{0.000201563},{0.00131378},{0.000627435},{0.000158125},{0.000201563},{0.000201563},{0.000201563},{0.000303491},{0.000201563},{0.000547864},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000477419},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.00049811},{0.000249063},{0.000261111},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000255036},{0.000638321},{0.000201563},{0.00321291},{0.000201563},{0.000627435},{0.000231527},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.000201563},{0.00279318},{0.000443751},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000537329},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.00237135},{0.000218388},{0.000201563},{0.000201563},{0.000201563},{0.00157703},{0.000201563},{0.000627435},{0.000620136},{0.000201563},{0.000201563},{0.

000201563},{0.000220355},{0.000201563},{0.000627435},{0.000339029},{0.00020156
3},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000
201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{
0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.00219759},{0.00020156
3},{0.000627435},{0.000201563},{0.000201563},{0.00279318},{0.000295784},{0.0002
01563},{0.000201563},{0.000201563},{0.000959689},{0.000627435},{0.000201563},{0
.000201563},{0.00279318},{0.000201563},{0.000373933},{0.000201563},{0.000201563
},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000201563},{0.0002
01563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.00045101},{0.
00150364},{0.000201563},{0.000744613},{0.000201563},{0.000201563},{0.000201563
},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.0002
01563},{0.000201563},{0.000201563},{0.000201563},{0.000213989},{0.000201563},{0
.000201563},{0.000228669},{0.000627435},{0.000201563},{0.000201563},{0.00074461
3},{0.000627435},{0.000201563},{0.000363331},{0.000627435},{0.000201563},{0.000
201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{0.000201563},{
0.000201563},{0.000744613},{0.00279318},{0.000627435},{0.000201563},{0.00020156
3},{0.000293345},{0.000201563},{0.000201563},{0.000627435},{0.000201563},{0.000
627435},{0.000201563},{0.000201563},{0.000744613},{0.000627435},{0.000201563},{
0.000201563},{0.000201563},{0.000494182},{0.000627435},{0.000232945},{0.0002015
63},{0.000201563},{0.000627435},{0.000621922},{0.000201563},{0.000201563},{0.00
0627435},{0.000627435},{0.000201563},{0.000201563},{0.00217515},{0.000201563},{
0.000783328},{0.00057055},{0.000201563},{0.000201563},{0.000201563},{0.00038660
9},{0.000201563},{0.000201563},{0.000373341},{0.000201563},{0.00021486},{0.0003
77869},{0.00107089},{0.000437473},{0.000201563},{0.000201563},{0.000201563},{0.
000293345},{0.000201563},{0.000201563},{0.0010878},{0.000744613},{0.00279318},{
0.000627435},{0.000627435},{0.000627435},{0.000627435},{0.000201563},{0.0002015
63},{0.000201563},{0.000201563},{0.000201563},{0.000627435},{0.00165461},{0.000
201563},{0.000201563},{0.000201563},{0.000201563},{0.00279318},{0.000201563},{0
.000201563},{0.000627435},{0.000201563}}

PŘÍLOHA P II: ODPOVĚĎ NA TESTOVACÍ MNOŽINU ZPRACOVANÝCH PROGRAMEM OUTGUESS

{{0.998832},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998843},{0.998832},{0.998852},{0.998832},{0.998852},{0.99885},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998832},{0.998852},{0.998832},{0.998852},{0.998832},{0.998852},{0.998852},{0.998832},{0.998843},{0.998552},{0.998832},{0.998844},{0.998852},{0.998832},{0.998852},{0.99884},{0.998832},{0.998839},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998838},{0.998832},{0.998832},{0.998852},{0.998852},{0.998852},{0.998847},{0.998832},{0.998832},{0.998852},{0.998852},{0.998843},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998832},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998851},{0.998852},{0.998844},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998832},{0.998832},{0.997584},{0.998852},{0.998852},{0.998852},{0.998852},{0.998843},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998839},{0.998852},{0.998852},{0.998852},{0.998852},{0.998846},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998835},{0.998852},{0.998852},{0.998852},{0.998546},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998848},{0.998852},{0.998852},{0.998852},{0.99232},{0.998852},{0.998852},{0.998852},{0.998832},{0.998832},{0.998842},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.997917},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998848},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998832},{0.998852},{0.998852},{0.998401},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998848},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998

852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852
},{0.998852},{0.998852},{0.998842},{0.998852},{0.998852},{0.998852},{0.998852},{0
.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998
852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998844},{0.998852},{0.998847
},{0.998844},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0
.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998
852},{0.998852},{0.998852},{0.99884},{0.998852},{0.998832},{0.998852},{0.998832},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998851},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998844},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998828},{0.998848},{0.998852},{0.998852},{0.9
98842},{0.998852},{0.998852},{0.998832},{0.998852},{0.998832},{0.998852},{0.9988
52},{0.998842},{0.998852},{0.998852},{0.998846},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998832},{0.998832},{0.9
98852},{0.998832},{0.998833},{0.998842},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998832},{0.998832},{0.998841},{0.998852},
{0.998852},{0.998852},{0.998834},{0.998852},{0.998832},{0.998852},{0.998832},{0.9
98852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.9988
41},{0.998832},{0.998852},{0.998832},{0.998852},{0.998832},{0.998852},{0.998852},
{0.998852},{0.998676},{0.998832},{0.998836},{0.998832},{0.998852},{0.998852},{0.9
98852},{0.998845},{0.998852},{0.998852},{0.998842},{0.998852},{0.998832},{0.9984
78},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998843},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998838},{0.9
98852},{0.998852},{0.998852},{0.998682},{0.998852},{0.998852},{0.998852},{0.9988

52},{0.998832},{0.998836},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998569},{0.998852},{0.9
98843},{0.998832},{0.812321},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998851},{0.998832},{0.9
98852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998832},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.993146},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998054},{0.998852},{0.998843},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998812},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
32},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998845},{0.9979
24},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998843},{0.998852},{0.997311},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.9
98852},{0.998852},{0.997758},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998837},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998839},{0.9988

52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
32},{0.998852},{0.998832},{0.998832},{0.998852},{0.998841},{0.998832},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998843},{0.998852},{0.998838},{0.9
98832},{0.998832},{0.998832},{0.998832},{0.998852},{0.998833},{0.998832},{0.9988
41},{0.998852},{0.998852},{0.998852},{0.99885},{0.998852},{0.998832},{0.998852},{
0.998852},{0.998832},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.99
8852},{0.998832},{0.998852},{0.998832},{0.998852},{0.99884},{0.998852},{0.998832
},{0.998852},{0.998852},{0.998852},{0.998832},{0.998832},{0.998852},{0.998833},{0
.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998
832},{0.998832},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998847
},{0.998849},{0.998852},{0.998832},{0.998832},{0.998852},{0.998852},{0.998852},{0
.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998
852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998684
},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998832},{0.998833},{0
.998832},{0.998832},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998
852},{0.998613},{0.998852},{0.998852},{0.998832},{0.587459},{0.998852},{0.998852
},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0
.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998
852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852
},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0
.998852},{0.998852},{0.998832},{0.998832},{0.998852},{0.998852},{0.998852},{0.998
852},{0.998852},{0.998832},{0.998837},{0.998852},{0.998852},{0.998852},{0.998852
},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998833},{0
.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998
852},{0.998852},{0.998852},{0.998812},{0.998852},{0.998277},{0.998852},{0.998852
},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0
.998852},{0.998852},{0.998836},{0.998852},{0.998852},{0.998852},{0.998832},{0.998
852},{0.998844},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852
},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0
.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998851},{0.998852},{0.998
852},{0.998852},{0.998852},{0.998852},{0.998825},{0.998852},{0.998852},{0.998852

},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0
.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998
846},{0.998852},{0.998852},{0.998852},{0.99884},{0.998852},{0.998852},{0.998852},
{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998852},{0.998843},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998833},{0.998852},{0.998832},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.998852},{0.9
98852},{0.998843},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.9988
32},{0.997908},{0.998852},{0.998484},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.988954},{0.998832},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998832},{0.9
98852},{0.998852},{0.998852},{0.998832},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9988
52},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},
{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.998852},{0.9
98852},{0.998852},{0.998852},{0.998852},{0.998852}}