

Reševanje igre Sudoku z umetno inteligenco

Mentor: dr. Bojan Ploj

Avtor: Simon Plazar, 3. a

Program: SSI, tehnik
računalništva

Ptuj, 9. 2. 2020

Zahvala

Zahvaljujem se mentorju, dr. Bojanu Ploju, ki mi je bil pri izdelavi naloge v veliko pomoč s svojimi nasveti in predlogi ter mi pomagal rešiti problem, ki sem si ga zastavil ob začetku izdelave raziskovalne naloge.

Kazalo

1.	UVOD.....	8
2.	TEORETIČNI DEL.....	8
3.	METODE IN MATERIALI	9
3.1	Sudoku	9
3.2	Program Weka	9
3.3	Sudoku 4 x 4	9
3.4	Sudoku 9 x 9	10
3.5	Tehnike preslikave	11
3.5.1.	Sudoku 4 x 4.....	11
3.5.2.	Sudoku 9 x 9.....	14
4.	REZULTATI IN RAZPRAVA.....	19
5.	ZAKLJUČEK.....	19
6.	PRILOGE.....	20
7.	VIRI IN LITERATURA	28

Kazalo slik

SLIKA 1 PRESLIKAVA PREKO KRIŽA (4 x 4).....	11
SLIKA 2 PRESLIKAVA PREKO DIAGONALE (4 x 4).....	12
SLIKA 3 MENJAVA VRSTIC (4 x 4)	12
SLIKA 4 MENJAVA STOLPCEV (4 x 4)	13
SLIKA 5 MENJAVA STOLPCEV 1. (9 x 9)	14
SLIKA 6 MENJAVA STOLPCEV 2. (9 x 9)	14
SLIKA 7 MENJAVA VRSTIC 1. (9 x 9)	15
SLIKA 8 MENJAVA VRSTIC 2. (9 x 9)	15
SLIKA 9 MENJAVA VRSTNEGA REDA ŠTEVILK V STOLPCU KVADRATOV.....	16
SLIKA 10 MENJAVA VRSTNEGA REDA ŠTEVILK V VRSTICI KVADRATOV.....	16
SLIKA 11 PRESLIKAVA PREKO KRIŽA (9 x 9).....	17
SLIKA 12 PRESLIKAVA PREKO DIAGONALE (9 x 9).....	18

Povzetek

Danes je umetna inteligenca (UI) prisotna v mnogih pametnih napravah. Uporablja se vse od reševanja raznolikih problemov, do igranja iger, kot sta Go in Šah. Zanimalo me je, če je UI zmožen rešiti igro Sudoku. Uporabljal sem program Weka, ki mi je omogočal uporabo različnih vrst umetne inteligence. Naučil sem se grajenja arff datoteke, ki je potrebna za vnašanje podatkov v program Weka. Arff datoteko sem napolnil s številnimi učnimi primeri, ki so sestavljeni iz nerešenega in rešenega primera igre. S pomočjo Excel dokumenta sem pridobil veliko število učnih primerov v kratkem času. S tehnikami preslikave sem dobil iz enega učnega primera igre veliko učnih primerov. Raziskovalno nalogo sem začel z manjšo različico igre velikosti 4 x 4, kasneje pa napredoval na večjo različico velikosti 9 x 9. Ugotovil sem, da število učnih primerov pomembno vpliva na rezultat, ter da ni vsaka UI primerna za vsak problem, ki ga poskušamo rešiti.

Ključne besede: Umetna inteligenca, Sudoku, Weka, Arff dokument, Excel, učni primer, reševanje, igra.

Summary

Today, artificial intelligence (AI) is present in almost all of our smart devices. It is used for everything from problem solving to playing board games such as Go and Chess. I was wondering if AI is able to solve a game of Sudoku. I used a program named Weka, that allowed me to use various types of artificial intelligence. I learned how to build an arff file which allowed me to enter data into the Weka program. I filled the Arff file with a large number of learning examples. Each learning example consisted of an unsolved and solved game example. I have acquired a large number of teaching examples in a short time, using mapping techniques, in an Excel document. I started my research assignment with a smaller version of the game with the size 4 x 4, and later progressed to the larger version of the game size 9 x 9. In my research, I found out, that the number of learning examples has a big impact on the results and that not every AI is appropriate for every problem we try to solve.

Key words: Artificial intelligence, Sudoku, Weka, Arff file, Excel, learning example, solving, game.

1. UVOD

Umetna inteligenca (UI) je imela zadnja leta velik vpliv na razvoj računalništva. UI obravnava naprave, ki zaznavajo vplive iz okolja in se samostojno odzivajo na njih. Splošno je znano dejstvo, da ni inteligence brez učenja, zato inteligentni sistemi med učenjem prilagajajo svoje delovanje tako, da povečujejo uspešnost. Delovati skušajo na enak način kot človek, za kar uporabljajo modele umetnih nevronske mreže in različne druge načine iskanja rešitev z metodami, ki večinoma temeljijo na statistiki in verjetnosti. Ljubitelji znanstvene fantastike ste že zagotovo naleteli na UI. Znani tovrstni filmi so Terminator, Lara Croft, Matrica, Odiseja 2001 in mnogi drugi. Eno izmed najpomembnejših področij umetne inteligence je strojno učenje. To je postopek, ki omogoča inteligentnim sistemom v kopici podatkov poiskati pravila oziroma znanje. To znanje omogoča da inteligentni sistem učinkovito odgovarja tudi na nova vprašanja, s katerimi se med učenjem ni srečal. UI si med učenjem ne zapomni podatkov, ampak najde postopek kako priti do pravilnega odgovora. To osvoji s pridobivanjem znanja z izkušnjami. Dandanes so mnogi programi, opremljeni z UI – zdravnikom pomagajo pri postavljanju diagnoz, zavarovalnicam pri odkrivanju goljufij, veleblagovnicam povedo kateri izdelki na policah naj stojijo skupaj, premagajo najboljše igralce miselnih iger... Leta 2017 je program AlphaGo Master, ki ga je razvilo podjetje DeepMind Technologies, premagal takratnega najboljšega igralca Ke Jie-ja v miselni igri Go. Z raziskovalno nalogo sem želel ugotoviti, ali je umetna inteligenca zmožna rešiti Sudoku. V raziskovalni nalogi sem zajel nekaj pomembnih dosežkov UI v zadnjih letih, kratek opis igre sudoku in programa Weka, ki sem ga uporabljal za svoje testiranje in potek, kako sem izvajal poskuse različnih vrst UI ter moje ugotovitve raziskave. (Univerza v Ljubljani, 2015)

2. TEORETIČNI DEL

UI se uporablja pri napovedovanju (npr. vremena, dobička, prodaje, itd.) strojnem prevajanju jezikov (Google Translate) in različnih internetnih iskalnikih (Google, Najdi.si, Yahoo, Internet Explorer, itd.). (Wikipedija, 2018) Uporablja se tudi pri namiznih igrah, kot so šah, Go, poker, dama, itd. Omenim lahko dva izmed bolj znanih primerov uporabe umetne inteligence. Prvi primer je zmaga IBM-ovega šahovskega programa Deep Blue, ki je leta 1997 premagal šahovskega velemejstra in večkratnega svetovnega prvaka Garija Kimoviča Kasparova. Ta rezultat je ljudem dokazal, da so lahko inteligentne naprave uspešnejše od ljudi, kar je svet šokiralo. Drugi primer je podobna zgodba, ki se je zgodila leta 2017, ko je naprava z imenom AlphaGo, ki ga je ustvaril DeepMind Technologies, v namizni igri GO premagala takratnega svetovnega prvaka Ke Jie-ja z rezultatom 3:0. Svet je nato presenetila nova različica AlphaGo Zero, ki je svojo predhodnico AlphaGo premagala z rezultatom 100:0. (Ploj, 2019, str. 10)

3. METODE IN MATERIALI

3.1 Sudoku

Sudoku je logična igra, katere cilj je zapolniti kvadratno mrežo, po navadi velikosti 9 x 9, s števili od 1 do 9. Vsako število se lahko pojavi samo enkrat v vsakem stolpcu, vsaki vrstici in vsakem manjšem kvadratu velikosti 3 x 3. V mreži so nekatera števila že podana. Za rešitev igre je potreben logičen razmislek in malo potrpežljivosti.

3.2 Program Weka

Weka je odprtokodna programska oprema za strojno učenje, ki vsebuje grafični uporabniški vmesnik (GUI). Uporablja se za poučevanje, raziskave in industrijske aplikacije, vsebuje množico vgrajenih orodij za naloge, potrebne strojnega učenja. Ta program sem uporabil z namenom, da bi ugotovil ali je UI zmožna rešiti Sudoku. Program omogoča uporabo različnih vrst UI. Z njim sem ugotovil, kateri algoritem je najboljši za reševanje igre Sudoku.

Za vnos učnih podatkov je služila datoteka arff. To datoteko lahko napišemo sami z navadnim unicode urejevalcem besedila. Sestavljena je iz vrstic z ukazi. Da lahko izvedemo strojno učenje potrebujemo učne primere. Ti so sestavljeni iz začetnih podatkov (nerešen Sudoku) in rešitve (rešen Sudoku). Potrebujemo dovolj učnih primerov, da se UI lahko nauči, kako rešiti Sudoku. Najaviti moramo toliko atributov, kot je podatkov v učnem primeru. Program sam izračuna kje se konča en vnos in kje se začne drugi. Paziti moramo, da ne najavimo premalo ali preveč atributov, saj potem program ne interpretira vnosov pravilno.

3.3 Sudoku 4 x 4

UI potrebuje veliko učnih primerov. Da bi ustvarili učni primer, bi morali ročno vpisati vse številke, ločene z vejico, nerešenega in rešenega primera igre Sudoku. Ker bi to bil zelo obsežen proces, sem ustvaril Excel datoteko, ki mi je olajšala delo. Začel sem s različico igre, velikosti 4 x 4. Vanj lahko vpisujemo števila od 1 do 4. Ima enaka pravila, le da je igra manjša. Tehnike preslikave, ki sem jih ugotovil tukaj, sem kasneje lahko uporabil pri večji različici igre. Cilj Excel dokumenta je bil, da s čim manj pisanja dobim čim več različic igre. S tehnikami preslikave lahko dobim iz ene igre Sudoku veliko več primerov, ki jih lahko vključim v arff dokument.

V Excelu sem priredil prostor, kjer sem lahko vpisal Sudoku. Ob njem sem postavil celice, ki bi izračunale vsoto vseh števil v vrstici in stolpcu. S tem sem lahko preveril, ali je rešen Sudoku pravilen ali ne. To me je zaščitilo pred morebitnimi napakami pri prenašanju igre iz interneta/revije v Excel dokument. Nato sem začel vpisovati formule, da sem s tehnikami preslikave dobil različice tega primera.

To vse sem ponovil za nerešen Sudoku. Ko sem bil zadovoljen s številom različic, sem ustvaril sistem pretvarjanja različic v učne primere. To sem izvedel tako, da sem števila pretvoril v znake, ki sem jih potem lahko združil, ločene z vejico, v niz znakov. Združil sem nerešen in

rešen Sudoku. Dobil sem učni primer. Za vsako različico, ki sem jo dobil s preslikavo, sem dobil en učni primer. Učne primere sem nato tudi združil skupaj v dolg niz znakov. Posamezne učne primere sem ločil z »X«, kar mi je omogočilo, da sem lahko razbral posamezne učne primere med seboj. V arff dokumentu so učni primeri ločeni med seboj s pomikom v novo vrstico, česar pa nisem mogel ponazoriti v Excelu, saj sem med prenosom iz Excela v arff dokument uporabil internetni brskalnik, ki mi je pretvoril vse znake v nizu znakov v unicode tekst, ki ga lahko prenesem v arff dokument. »X« sem uporabil kot znak, da se moram pomakniti v novo vrsto. S tem sem lahko niz znakov z malo dela pretvoril v delujoče učne primere.

Ko sem napolnil arff datoteko s primeri, sem začel uporabljati program Weka in njegove funkcije. Pod zavihkom »classify« sem imel možnost izbirati načine dela. Uporabljal sem različne arff datoteke različne težavnostne stopnje. Del učnih podatkov sem uporabil za učenje, preostanek pa za preizkušanje. Učenje sem izvajal z različnimi pametnimi napravami. Program izpiše, koliko časa je potreboval za učenje in pravilnost UI pri preverjanju/testiranju. Ko sem prišel do primera, kjer UI pri preizkušanju ni naredila napake pri ugotavljanju števil v polju, sem nadaljeval z različico igre velikosti 9 x 9.

3.4 Sudoku 9 x 9

Proces grajenja Excel datoteke za Sudoku velikosti 9 x 9, je bil enak kot pri igri velikosti 4 x 4. Priredil sem prostor, tokrat nekoliko večji, za vnos igre iz interneta/revije. Uporabil sem prilagojene tehnike preslikave. Tehnik preslikave je bilo pri tej velikosti igre veliko več. Z izkušnjami, ki sem jih pridobil z grajenjem Excel datoteke za Sudoku velikosti 4 x 4, sem v kratkem času pripravil Excel dokument za Sudoku 9 x 9. Posamezne učne primere, ločene z »X«, sem združil v dolg niz znakov. Ta niz znakov sem imel shranjen v eni celici v Excel dokumentu. Ugotovil sem, da ima posamezna celica omejitev, koliko znakov lahko vsebuje: 32767 znakov. Vsota vseh znakov mojih vnosov je znašala 42768. Zaradi velikega števila znakov nisem mogel več uporabiti brskalnika kot vmesnika za pretvorbo vseh znakov v unicode tekst. Niz znakov sem ločil na dva dela. Dobil sem dve celici, ki sta vsebovali vse vnose dobljenih različic igre. Ti dve celici sem prenesel na nov list. To mi je omogočilo, da sem list izvozil iz Excel dokumenta in ga shranil kot unicode tekstovni dokument. Besedilna datoteka je vseboval dva dolga niza znakov, obdana z narekovaji. Vsebinsko sem prekopiral in prilepil v novo arff datoteko, prirejeno za Sudoku 9 x 9, in odstranil narekovaje. »X« sem zamenjal s pomiki v novo vrsto. Ko sem imel dovolj učnih primerov, sem novo arff datoteko odprl v weki.

Ko sem zagnal program Weka z novo arff datoteko, se je zataknilo. Program ni napredoval. Po nekaj minutah se je program porušil. V spletu sem poiskal razlog. Ugotovil sem, da program Weka ni imel rezerviranega dovolj polnilnika za tako velik poskus. Program Weka je zelo star in že dolgo ni bil posodobljen. Da sem lahko rešil svoj problem, sem moral program zagnati preko ukaznega okna (CMD) in tam rezervirati dovolj pomnilnika. Privzeta velikost pomnilnika je bila 256 MB, jaz pa sem jo spremenil na 1 GB.

Ko sem ponovno zagnal program s privzeto UI, je program deloval. Sprva so bili rezultati slabi. Začel sem raziskovati, katere pametne naprave so najprimernejše za reševanje igre Sudoku. Ustvaril sem štiri arff datoteke. Prva je vsebovala lahke, druga je vsebovala težje, tretja je vsebovala težke in četrta najtežje primere igre Sudoku. S tem sem želel ugotoviti, kako zahtevne različice igre Sudoku lahko reši posamezna pametna naprava. Vse datoteke so vsebovale enako število učnih primerov.

3.5 Tehnike preslikave

Za lažje razumevanje si lahko predstavljamo posamezen Sudoku kot mrežo točk. V vsakem stolpcu je bila posamezna črka, ki je predstavljala tisti stolpec, vsako vrstico pa je predstavljala številka v vrstnem redu.

3.5.1. Sudoku 4 x 4

Tehnike, ki sem jih odkril, so naslednje:

1. Zrcaljenje/Preslikanje preko križa:

A									B
	A1	B1	C1	D1		D1	C1	B1	A1
	A2	B2	C2	D2		D2	C2	B2	A2
	A3	B3	C3	D3		D3	C3	B3	A3
	A4	B4	C4	D4		D4	C4	B4	A4
	A4	B4	C4	D4		D4	C4	B4	A4
	A3	B3	C3	D3		D3	C3	B3	A3
	A2	B2	C2	D2		D2	C2	B2	A2
	A1	B1	C1	D1		D1	C1	B1	A1
C									D

Slika 1 Preslikava preko križa (4 x 4)

Vir: Lasten

Pri tej tehniki sem dobil tri nove različice igre. Na vogalih so zapisana ločna imena posameznih različic (A, B, C, D). Osnovni Sudoku (A) je preslikan preko osi, ponazorjene z rumenimi celicami. **Slika 1** prikazuje štiri mreže. Mrežo, označeno s črko »B«, sem dobil s preslikanjem mreže »A« preko navpične osi. Mrežo, označeno s črko »C«, sem dobil s preslikanjem mreže »A« preko vodoravne osi. Mrežo, označeno s črko »D«, pa sem dobil s preslikanjem mreže »A« preko izhodišča osi (rumena celica na sredini križa).

2. Zrcaljenje/Preslikanje preko diagonale:

		A(/)					A(\)			
	D4	D3	D2	D1		A1	A2	A3	A4	
	C4	C3	C2	C1		B1	B2	B3	B4	
	B4	B3	B2	B1		C1	C2	C3	C4	
	A4	A3	A2	A1		D1	D2	D3	D4	

Slika 2 Preslikava preko diagonale (4 x 4)

Vir: Lasten

Pri tej tehniki sem polja zrcalil preko točk, ki ležijo na diagonali. Te so ponazorjene z rumenimi celicami. Uporabil sem dve različici te tehnike. **Slika 2** prikazuje dve mreži, preslikani preko diagonale. V levi mreži sem zamenjal celice »D4« z »A1«, »D2« z »C1«, »B4« z »A3«, itd. V desni mreži sem zamenjal celice »A4« z »D1«, »C4« z »D3«, »A2« z »B1«, itd.

3. Menjava vrstic:

	A(-)1-2switch					A(-)3-4switch					A(-)half switch			
	A2	B2	C2	D2		A1	B1	C1	D1		A3	B3	C3	D3
	A1	B1	C1	D1		A2	B2	C2	D2		A4	B4	C4	D4
	A3	B3	C3	D3		A4	B4	C4	D4		A1	B1	C1	D1
	A4	B4	C4	D4		A3	B3	C3	D3		A2	B2	C2	D2

Slika 3 Menjava vrstic (4 x 4)

Vir: Lasten

Da sem dobil nove različice, sem menjal posamezne vrstice igre. Vrstice, ki sem jih zamenjal, sem ponazoril z drugačno barvo. **Slika 3** prikazuje tri mreže, kjer sem zamenjal različne vrstice. Pri levi mreži sem zamenjal vrstici »1« in »2«, pri sredinski mreži sem zamenjal vrstici »3« in »4«, pri desni mreži pa sem zamenjal pare vrstic »1 in 2« z »3 in 4«.

4. Menjava stolpcev:

	A()1-2switch				A()3-4switch				A()half switch			
B1	A1	C1	D1	A1	B1	D1	C1	C1	D1	A1	B1	
B2	A2	C2	D2	A2	B2	D2	C2	C2	D2	A2	B2	
B3	A3	C3	D3	A3	B3	D3	C3	C3	D3	A3	B3	
B4	A4	C4	D4	A4	B4	D4	C4	C4	D4	A4	B4	

Slika 4 Menjava stolpcev (4 x 4)

Vir: Lasten

Da sem dobil nove različice, sem menjal posamezne stolpce igre. Stolpce, ki sem jih zamenjal, sem ponazoril z drugačno barvo. **Slika 4** prikazuje tri mreže, kjer sem zamenjal različne stolpce. Pri levi mreži sem zamenjal stolpca »A« in »B«, pri sredinski mreži sem zamenjal stolpca »C« in »D«, pri desni mreži pa sem zamenjal pare stolpcev »A in B« z »C in D«.

3.5.2. Sudoku 9 x 9

Tehnike, ki sem jih uporabil:

1. Menjava stolpcev kvadratov:

1	a1	b1	c1	d1	e1	f1	g1	h1	i1
2	a2	b2	c2	d2	e2	f2	g2	h2	i2
3	a3	b3	c3	d3	e3	f3	g3	h3	i3
4	a4	b4	c4	d4	e4	f4	g4	h4	i4
5	a5	b5	c5	d5	e5	f5	g5	h5	i5
6	a6	b6	c6	d6	e6	f6	g6	h6	i6
7	a7	b7	c7	d7	e7	f7	g7	h7	i7
8	a8	b8	c8	d8	e8	f8	g8	h8	i8
9	a9	b9	c9	d9	e9	f9	g9	h9	i9
	1	2	3						

1	a1	b1	c1	g1	h1	i1	d1	e1	f1
2	a2	b2	c2	g2	h2	i2	d2	e2	f2
3	a3	b3	c3	g3	h3	i3	d3	e3	f3
4	a4	b4	c4	g4	h4	i4	d4	e4	f4
5	a5	b5	c5	g5	h5	i5	d5	e5	f5
6	a6	b6	c6	g6	h6	i6	d6	e6	f6
7	a7	b7	c7	g7	h7	i7	d7	e7	f7
8	a8	b8	c8	g8	h8	i8	d8	e8	f8
9	a9	b9	c9	g9	h9	i9	d9	e9	f9
	1	3	2						

1	d1	e1	f1	a1	b1	c1	g1	h1	i1
2	d2	e2	f2	a2	b2	c2	g2	h2	i2
3	d3	e3	f3	a3	b3	c3	g3	h3	i3
4	d4	e4	f4	a4	b4	c4	g4	h4	i4
5	d5	e5	f5	a5	b5	c5	g5	h5	i5
6	d6	e6	f6	a6	b6	c6	g6	h6	i6
7	d7	e7	f7	a7	b7	c7	g7	h7	i7
8	d8	e8	f8	a8	b8	c8	g8	h8	i8
9	d9	e9	f9	a9	b9	c9	g9	h9	i9
	2	1	3						

Slika 5 Menjava stolpcev 1. (9 x 9)

Vir: Lasten

1	d1	e1	f1	g1	h1	i1	a1	b1	c1
2	d2	e2	f2	g2	h2	i2	a2	b2	c2
3	d3	e3	f3	g3	h3	i3	a3	b3	c3
4	d4	e4	f4	g4	h4	i4	a4	b4	c4
5	d5	e5	f5	g5	h5	i5	a5	b5	c5
6	d6	e6	f6	g6	h6	i6	a6	b6	c6
7	d7	e7	f7	g7	h7	i7	a7	b7	c7
8	d8	e8	f8	g8	h8	i8	a8	b8	c8
9	d9	e9	f9	g9	h9	i9	a9	b9	c9
	2	3	1						

1	g1	h1	i1	a1	b1	c1	d1	e1	f1
2	g2	h2	i2	a2	b2	c2	d2	e2	f2
3	g3	h3	i3	a3	b3	c3	d3	e3	f3
4	g4	h4	i4	a4	b4	c4	d4	e4	f4
5	g5	h5	i5	a5	b5	c5	d5	e5	f5
6	g6	h6	i6	a6	b6	c6	d6	e6	f6
7	g7	h7	i7	a7	b7	c7	d7	e7	f7
8	g8	h8	i8	a8	b8	c8	d8	e8	f8
9	g9	h9	i9	a9	b9	c9	d9	e9	f9
	3	1	2						

1	g1	h1	i1	d1	e1	f1	a1	b1	c1
2	g2	h2	i2	d2	e2	f2	a2	b2	c2
3	g3	h3	i3	d3	e3	f3	a3	b3	c3
4	g4	h4	i4	d4	e4	f4	a4	b4	c4
5	g5	h5	i5	d5	e5	f5	a5	b5	c5
6	g6	h6	i6	d6	e6	f6	a6	b6	c6
7	g7	h7	i7	d7	e7	f7	a7	b7	c7
8	g8	h8	i8	d8	e8	f8	a8	b8	c8
9	g9	h9	i9	d9	e9	f9	a9	b9	c9
	3	2	1						

Slika 6 Menjava stolpcev 2. (9 x 9)

Vir: Lasten

Menjamo lahko stolpce kvadratov, velikosti 3 x 9 (širina, višina). Stolpce, ki sem jih zamenjal, sem ponazoril z drugačno barvo. Oznako stolpcev sem zapisal pod posameznim stolpcem. **Slika 5** prikazuje tri mreže, kjer sem zamenjal različne stolpce. Leva mreža je prvotna mreža. Pri sredinski mreži sem zamenjal stolpca »2« in »3«, pri desni mreži pa sem zamenjal stolpca »1« in »2«. **Slika 6** prikazuje tri mreže, kjer sem tudi zamenjal različne stolpce. Pri levi mreži sem zamenjal stolpec »1« z »2«, stolpec »2« z »3«, in stolpec »3« z »1«. Pri sredinski mreži sem zamenjal stolpec »1« z »3«, stolpec »2« z »1«, in stolpec »3« z »2«. Pri desni mreži pa sem zamenjal stolpec »1« z »3«, stolpec »3« z »1«, stolpec »2« pa sem ohranil v prvotni obliki.

2. Menjava vrstic kvadratkov:

	a1	b1	c1	d1	e1	f1	g1	h1	i1	1	a1	b1	c1	d1	e1	f1	g1	h1	i1		a4	b4	c4	d4	e4	f4	g4	h4	i4	2
	a2	b2	c2	d2	e2	f2	g2	h2	i2		a2	b2	c2	d2	e2	f2	g2	h2	i2		a5	b5	c5	d5	e5	f5	g5	h5	i5	
1	a3	b3	c3	d3	e3	f3	g3	h3	i3		a3	b3	c3	d3	e3	f3	g3	h3	i3		a6	b6	c6	d6	e6	f6	g6	h6	i6	
	a4	b4	c4	d4	e4	f4	g4	h4	i4	3	a7	b7	c7	d7	e7	f7	g7	h7	i7		a1	b1	c1	d1	e1	f1	g1	h1	i1	
	a5	b5	c5	d5	e5	f5	g5	h5	i5		a8	b8	c8	d8	e8	f8	g8	h8	i8		a2	b2	c2	d2	e2	f2	g2	h2	i2	
2	a6	b6	c6	d6	e6	f6	g6	h6	i6		a9	b9	c9	d9	e9	f9	g9	h9	i9		a3	b3	c3	d3	e3	f3	g3	h3	i3	1
	a7	b7	c7	d7	e7	f7	g7	h7	i7	2	a4	b4	c4	d4	e4	f4	g4	h4	i4		a7	b7	c7	d7	e7	f7	g7	h7	i7	
	a8	b8	c8	d8	e8	f8	g8	h8	i8		a5	b5	c5	d5	e5	f5	g5	h5	i5		a8	b8	c8	d8	e8	f8	g8	h8	i8	
3	a9	b9	c9	d9	e9	f9	g9	h9	i9		a6	b6	c6	d6	e6	f6	g6	h6	i6	3	a9	b9	c9	d9	e9	f9	g9	h9	i9	
	1	2	3	4	5	6	7	8	9		1	2	3	4	5	6	7	8	9		1	2	3	4	5	6	7	8	9	

Slika 7 Menjava vrstic 1. (9 x 9)

Vir: Lasten

	a4	b4	c4	d4	e4	f4	g4	h4	i4		a7	b7	c7	d7	e7	f7	g7	h7	i7		a7	b7	c7	d7	e7	f7	g7	h7	i7	3
	a5	b5	c5	d5	e5	f5	g5	h5	i5		a8	b8	c8	d8	e8	f8	g8	h8	i8		a8	b8	c8	d8	e8	f8	g8	h8	i8	
2	a6	b6	c6	d6	e6	f6	g6	h6	i6		a9	b9	c9	d9	e9	f9	g9	h9	i9		a9	b9	c9	d9	e9	f9	g9	h9	i9	
	a7	b7	c7	d7	e7	f7	g7	h7	i7	3	a1	b1	c1	d1	e1	f1	g1	h1	i1		a4	b4	c4	d4	e4	f4	g4	h4	i4	2
	a8	b8	c8	d8	e8	f8	g8	h8	i8		a2	b2	c2	d2	e2	f2	g2	h2	i2		a5	b5	c5	d5	e5	f5	g5	h5	i5	
3	a9	b9	c9	d9	e9	f9	g9	h9	i9		a3	b3	c3	d3	e3	f3	g3	h3	i3	1	a6	b6	c6	d6	e6	f6	g6	h6	i6	
	a1	b1	c1	d1	e1	f1	g1	h1	i1	2	a4	b4	c4	d4	e4	f4	g4	h4	i4		a1	b1	c1	d1	e1	f1	g1	h1	i1	
	a2	b2	c2	d2	e2	f2	g2	h2	i2		a5	b5	c5	d5	e5	f5	g5	h5	i5		a2	b2	c2	d2	e2	f2	g2	h2	i2	
1	a3	b3	c3	d3	e3	f3	g3	h3	i3		a6	b6	c6	d6	e6	f6	g6	h6	i6	2	a3	b3	c3	d3	e3	f3	g3	h3	i3	1
	1	2	3	4	5	6	7	8	9		1	2	3	4	5	6	7	8	9		1	2	3	4	5	6	7	8	9	

Slika 8 Menjava vrstic 2. (9 x 9)

Vir: Lasten

Menjamo lahko vrstice kvadratov, velikosti 9 x 3 (širina, višina). Vrstice, ki sem jih zamenjal, sem ponazoril z drugačno barvo. Oznako vrstic sem zapisal levo od posamezne vrstice. **Slika 7** prikazuje tri mreže, kjer sem zamenjal različne vrstice. Leva mreža je prvotna mreža. Pri sredinski mreži sem zamenjal vrstici »2« in »3«, pri desni mreži pa sem zamenjal vrstici »1« in »2«. **Slika 8** prikazuje tri mreže, kjer sem tudi zamenjal različne vrstice. Pri levi mreži sem zamenjal vrstico »1« z »2«, vrstico »2« z »3«, in vrstico »3« z »1«. Pri sredinski mreži sem zamenjal vrstico »1« z »3«, vrstico »2« z »1«, in vrstico »3« z »2«. Pri desni mreži pa sem zamenjal vrstico »1« z »3«, vrstico »3« z »1«, vrstico »2« pa sem ohranil v prvotni obliki.

Naslednje tehnike sem uporabil na novih različicah prejšnjih tehnik.

3. Menjava vrstnega reda števil v posameznem stolpcu kvadratov velikosti 3 x 9 (širina, višina):

1	a1	b1	c1	d1	e1	f1	i1	h1	g1	1	a1	b1	c1	f1	e1	d1	g1	h1	i1	1	c1	b1	a1	d1	e1	f1	g1	h1	i1
2	a2	b2	c2	d2	e2	f2	i2	h2	g2	2	a2	b2	c2	f2	e2	d2	g2	h2	i2	2	c2	b2	a2	d2	e2	f2	g2	h2	i2
3	a3	b3	c3	d3	e3	f3	i3	h3	g3	3	a3	b3	c3	f3	e3	d3	g3	h3	i3	3	c3	b3	a3	d3	e3	f3	g3	h3	i3
4	a4	b4	c4	d4	e4	f4	i4	h4	g4	4	a4	b4	c4	f4	e4	d4	g4	h4	i4	4	c4	b4	a4	d4	e4	f4	g4	h4	i4
5	a5	b5	c5	d5	e5	f5	i5	h5	g5	5	a5	b5	c5	f5	e5	d5	g5	h5	i5	5	c5	b5	a5	d5	e5	f5	g5	h5	i5
6	a6	b6	c6	d6	e6	f6	i6	h6	g6	6	a6	b6	c6	f6	e6	d6	g6	h6	i6	6	c6	b6	a6	d6	e6	f6	g6	h6	i6
7	a7	b7	c7	d7	e7	f7	i7	h7	g7	7	a7	b7	c7	f7	e7	d7	g7	h7	i7	7	c7	b7	a7	d7	e7	f7	g7	h7	i7
8	a8	b8	c8	d8	e8	f8	i8	h8	g8	8	a8	b8	c8	f8	e8	d8	g8	h8	i8	8	c8	b8	a8	d8	e8	f8	g8	h8	i8
9	a9	b9	c9	d9	e9	f9	i9	h9	g9	9	a9	b9	c9	f9	e9	d9	g9	h9	i9	9	c9	b9	a9	d9	e9	f9	g9	h9	i9
	1	2	3	4	5	6	9	8	7		1	2	3	6	5	4	7	8	9		3	2	1	4	5	6	7	8	9

Slika 9 Menjava vrstnega reda števil v stolpcu kvadratov

Vir: Lasten

Pri tej tehniki sem zamenjal posamezne stolpce v vsakem stolpcu kvadratov. Možne bi bile tudi druge kombinacije. Oznako stolpcev sem zapisal pod posameznim stolpcem. **Slika 9** prikazuje tri mreže, kjer sem zamenjal različne stolpce. Pri levi mreži sem zamenjal stolpca »7« in »9«. Pri sredinski mreži sem zamenjal stolpca »4« in »6«. Pri desni mreži pa sem zamenjal stolpca »1« in »3«.

4. Menjava vrstnega reda števil v posamezni vrstici kvadratov velikosti 9 x 3 (širina, višina):

3	a3	b3	c3	d3	e3	f3	g3	h3	i3	1	a1	b1	c1	d1	e1	f1	g1	h1	i1	1	a1	b1	c1	d1	e1	f1	g1	h1	i1
2	a2	b2	c2	d2	e2	f2	g2	h2	i2	2	a2	b2	c2	d2	e2	f2	g2	h2	i2	2	a2	b2	c2	d2	e2	f2	g2	h2	i2
1	a1	b1	c1	d1	e1	f1	g1	h1	i1	3	a3	b3	c3	d3	e3	f3	g3	h3	i3	3	a3	b3	c3	d3	e3	f3	g3	h3	i3
4	a4	b4	c4	d4	e4	f4	g4	h4	i4	6	a6	b6	c6	d6	e6	f6	g6	h6	i6	4	a4	b4	c4	d4	e4	f4	g4	h4	i4
5	a5	b5	c5	d5	e5	f5	g5	h5	i5	5	a5	b5	c5	d5	e5	f5	g5	h5	i5	5	a5	b5	c5	d5	e5	f5	g5	h5	i5
6	a6	b6	c6	d6	e6	f6	g6	h6	i6	4	a4	b4	c4	d4	e4	f4	g4	h4	i4	6	a6	b6	c6	d6	e6	f6	g6	h6	i6
7	a7	b7	c7	d7	e7	f7	g7	h7	i7	7	a7	b7	c7	d7	e7	f7	g7	h7	i7	9	a9	b9	c9	d9	e9	f9	g9	h9	i9
8	a8	b8	c8	d8	e8	f8	g8	h8	i8	8	a8	b8	c8	d8	e8	f8	g8	h8	i8	8	a8	b8	c8	d8	e8	f8	g8	h8	i8
9	a9	b9	c9	d9	e9	f9	g9	h9	i9	9	a9	b9	c9	d9	e9	f9	g9	h9	i9	7	a7	b7	c7	d7	e7	f7	g7	h7	i7
	1	2	3	4	5	6	7	8	9		1	2	3	4	5	6	7	8	9		1	2	3	4	5	6	7	8	9

Slika 10 Menjava vrstnega reda števil v vrstici kvadratov

Vir: Lasten

Pri tej tehniki sem zamenjal posamezne vrstice v vsaki vrstici kvadratov. Možne bi bile tudi druge kombinacije. Oznako vrstic sem zapisal levo od posamezne vrstice. **Slika 10** prikazuje tri mreže, kjer sem zamenjal različne vrstice. Pri levi mreži sem zamenjal vrstici »1« in »3«. Pri sredinski mreži sem zamenjal vrstici »4« in »6«. Pri desni mreži pa sem zamenjal vrstici »7« in »9«.

5. Zrcaljenje/Preslikanje preko križa:

A																				B	
1	a1	b1	c1	d1	e1	f1	g1	h1	i1		1	i1	h1	g1	f1	e1	d1	c1	b1	a1	
2	a2	b2	c2	d2	e2	f2	g2	h2	i2		2	i2	h2	g2	f2	e2	d2	c2	b2	a2	
3	a3	b3	c3	d3	e3	f3	g3	h3	i3		3	i3	h3	g3	f3	e3	d3	c3	b3	a3	
4	a4	b4	c4	d4	e4	f4	g4	h4	i4		4	i4	h4	g4	f4	e4	d4	c4	b4	a4	
5	a5	b5	c1	d5	e5	f5	g5	h5	i5		5	i5	h5	g5	f5	e5	d5	c5	b5	a5	
6	a6	b6	c2	d6	e6	f6	g6	h6	i6		6	i6	h6	g6	f6	e6	d6	c6	b6	a6	
7	a7	b7	c3	d7	e7	f7	g7	h7	i7		7	i7	h7	g7	f7	e7	d7	c7	b7	a7	
8	a8	b8	c4	d8	e8	f8	g8	h8	i8		8	i8	h8	g8	f8	e8	d8	c8	b8	a8	
9	a9	b9	c1	d9	e9	f9	g9	h9	i9		9	i9	h9	g9	f9	e9	d9	c9	b9	a9	
		1	2	3	4	5	6	7	8	9			9	8	7	6	5	4	3	2	1
C		1	2	3	4	5	6	7	8	9		9	8	7	6	5	4	3	2	1	D

Slika 11 Preslikava preko križa (9 x 9)

Vir: Lasten

Pri tej tehniki sem dobil tri nove različice igre. Na vogalih so zapisana ločna imena posameznih različic (A, B, C, D). Osnovni Sudoku (A) je preslikan preko osi, ponazorjene s rumenimi celicami. **Slika 11** prikazuje štiri mreže. Mrežo, označeno s črko »B«, sem dobil s preslikanjem mreže »A« preko navpične osi. Mrežo, označeno s črko »C«, sem dobil s preslikanjem mreže »A« preko vodoravne osi. Mrežo, označeno s črko »D«, pa sem dobil s preslikanjem mreže »A« preko izhodišča osi (rumena celica na sredini križa).

6. Zrcaljenje/Preslikanje preko diagonale:

i9	i8	i7	i6	i5	i4	i3	i2	i1			a1	a2	a3	a4	a5	a6	a7	a8	a9
h9	h8	h7	h6	h5	h4	h3	h2	h1			b1	b2	b3	b4	b5	b6	b7	b8	b9
g9	g8	g7	g6	g5	g4	g3	g2	g1			c1	c2	c3	c4	c5	c6	c7	c8	c9
f9	f8	f7	f6	f5	f4	f3	f2	f1			d1	d2	d3	d4	d5	d6	d7	d8	d9
e9	e8	e7	e6	e5	e4	e3	e2	e1			e1	e2	e3	e4	e5	e6	e7	e8	e9
d9	d8	d7	d6	d5	d4	d3	d2	d1			f1	f2	f3	f4	f5	f6	f7	f8	f9
c9	c8	c7	c6	c5	c4	c3	c2	c1			g1	g2	g3	g4	g5	g6	g7	g8	g9
b9	b8	b7	b6	b5	b4	b3	b2	b1			h1	h2	h3	h4	h5	h6	h7	h8	h9
a9	a8	a7	a6	a5	a4	a3	a2	a1			i1	i2	i3	i4	i5	i6	i7	i8	i9

Slika 12 Preslikava preko diagonale (9 x 9)

Vir: Lasten

Pri tej tehniki sem polja zrcalil preko točk, ki ležijo na diagonali. Te so ponazorjene z rumenimi celicami. Uporabil sem dve različici te tehnike. **Slika 12** prikazuje dve mreži, preslikani preko diagonale. V levi mreži sem zamenjal celice »i9« z »a1«, »i2« z »h1«, »b9« z »a8«, itd. V desni mreži sem zamenjal celice »i1« z »a9«, »b1« z »a2«, »i8« z »h9«, itd.

4. REZULTATI IN RAZPRAVA

Pri raziskovanju, ali je UI zmožna rešiti Sudoku velikosti 4 x 4, sem ugotovil, da je med mojim testiranjem izstopala pametna naprava z imenom »SimpleLogistic«. Rešila je veliko število iger v kratkem času, ni pa vedno našla pravilne rešitve (**Priloga 1**). Pametna naprava je 64 učnih primerov pravilno rešila, 11 pa nepravilno. Potrebovala je 0.19 sekund za treniranje/učenje iz 221 primerov in 0.01 sekund, da je rešila 75 primerov za testiranje. Uporabil sem tudi pametno napravo z imenom »MultilayerPerceptron« (MLP). Ta je bila počasnejša, a je bila skoraj vedno pravilna (**Priloga 2**). Imela je boljše rezultate kot »SimpleLogistic«. Naprava je rešila 75 učnih primerov pravilno in nobenega nepravilno. Potrebovala je štirinajst sekund za treniranje/učenje iz 221 primerov in 0.01 sekund, da je rešila 75 primerov za testiranje. Pri reševanju igre velikosti 9 x 9 sem najprej uporabil ti dve funkciji. Ugotovil sem, da naprave, kot so »SimpleLogistic« in »MultilayerPerceptron«, porabijo zelo veliko pomnilnika za svoje delovanje. To je odvisno tudi od primera. Pri reševanju igre velikosti 4 x 4 je bilo posameznih atributov v posameznem učnem primeru veliko manj, kot pri igri velikosti 9 x 9. Sklepal sem, da ti dve funkciji nista najprimernejši za reševanje igre Sudoku. Ugotovil sem, da je odločitveno drevo primernejše za ta tip naloge (**Priloga 3**). Naprava je rešila 673 preizkusov pravilno in nobenega nepravilno. Potrebovala je 76 sekund za treniranje/učenje iz 1307 primerov in 0.01 sekund, da je rešila 673 primerov za testiranje. Pri testiranju sem ugotovil, da število učnih primerov pomembno vpliva na rezultate. UI je z zadostnim številom učnih primerov zmožna z lahkoto rešiti Sudoku katere koli težavnostne stopnje. Za učenje/treniranje najtežjih primerov igre UI porabi samo 1.06 sekund več, kot za najlažje primere igre Sudoku.

5. ZAKLJUČEK

V zadnjih letih je UI pomembno prispevala k razvoju znanosti. Dan danes je prisotna v skoraj vseh pametnih napravah, od avtonomnih avtomobilov do pametnih telefonov, ki imajo sposobnost zaznavanja in razumevanja govora. Zanimalo me je, če je z UI možno rešiti Sudoku. Pred začetkom raziskovanja sem sam rešil nekaj iger Sudoku velikosti 9 x 9 v reviji. Za vsako igro sem potreboval veliko časa. Večkrat sem popravljaval vrednosti, ki sem jih vpisoval v polja. Raziskovanje sem začel z lažjo različico igre Sudoku, velikosti 4 x 4, ki sem jo lahko z lahkoto sam rešil. Z urejevalnikom tabel sem ustvaril veliko število učnih primerov in jih shranjeval v posebno datoteko tipa arff. Za strojno učenje sem uporabljal odprtokodni program Weka, ki mi je omogočal uporabo umetne inteligence. Ko sem dobil prve dobre rezultate z uporabo UI pri tej velikosti igre, sem nadaljeval z večjo različico igre. Ob izdelavi naloge sem ugotovil, da UI potrebuje zelo kratek čas, da reši tudi najtežje primere igre Sudoku. Primerjal sem rezultate UI pri arff datoteki z lahkimi in najtežjimi primeri igre. Ugotovil sem, da število učnih primerov pomembno vpliva na rezultate UI.

6. PRILOGE

1. Priloga (Izsek)

Ker je celoten zapis dolg več strani, sem priložil samo glavne informacije iz celotnega zapisa.

=== Run information ===

Scheme: weka.classifiers.functions.SimpleLogistic -I 0 -M 500 -H 50 -W 0.0

Relation: sudoku4x4

Instances: 221

Attributes: 32

polje1

polje2

polje3

polje4

polje5

polje6

polje7

polje8

polje9

polje10

polje11

polje12

polje13

polje14

polje15

polje16

result1

result2

result3

result4

result5

result6

result7

result8
 result9
 result10
 result11
 result12
 result13
 result14
 result15
 result16

Test mode: split 66.0% train, remainder test

=== Classifier model (full training set) ===

-//-

Time taken to build model: 0.19 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.01 seconds

=== Summary ===

Correctly Classified Instances	64	85.3333 %
Incorrectly Classified Instances	11	14.6667 %
Kappa statistic	0.7265	
Mean absolute error	0.0742	
Root mean squared error	0.2259	
Relative absolute error	33.9783 %	
Root relative squared error	66.6419 %	
Total Number of Instances	75	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
---------	---------	-----------	--------	-----------	-----	----------	----------	-------

0.750	0.015	0.857	0.750	0.800	0.780	0.991	0.911	1
0.400	0.043	0.400	0.400	0.400	0.357	0.869	0.526	2
0.571	0.015	0.800	0.571	0.667	0.649	0.945	0.699	3
0.750	0.000	1.000	0.750	0.857	0.853	0.950	0.856	4
0.979	0.214	0.885	0.979	0.929	0.802	0.989	0.995	0
Weighted Avg.	0.853	0.140	0.854	0.853	0.848	0.761	0.973	0.912

=== Confusion Matrix ===

a b c d e <-- classified as

6 1 0 0 1 | a = 1

0 2 1 0 2 | b = 2

0 1 4 0 2 | c = 3

0 1 0 6 1 | d = 4

Vir: Program Weka

2. Priloga (Izsek)

Ker je celoten zapis dolg več strani, sem priložil samo glavne informacije iz celotnega zapisa.

=== Run information ===

Scheme: weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Relation: sudoku4x4

Instances: 221

Attributes: 32

polje1

polje2

polje3

polje4

polje5

polje6

polje7

polje8

polje9

polje10

polje11

polje12

polje13

polje14

polje15

polje16

result1

result2

result3

result4

result5

result6

result7

result8

result9
 result10
 result11
 result12
 result13
 result14
 result15
 result16

Test mode: split 66.0% train, remainder test

=== Classifier model (full training set) ===

-//-

Time taken to build model: 14.37 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.01 seconds

=== Summary ===

Correctly Classified Instances	75	100	%
Incorrectly Classified Instances	0	0	%
Kappa statistic	1		
Mean absolute error	0.0065		
Root mean squared error	0.0135		
Relative absolute error	1.727	%	
Root relative squared error	3.1087	%	
Total Number of Instances	75		

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	2

	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	3
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	4
Weighted Avg.	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

=== Confusion Matrix ===

a b c d <-- classified as

18 0 0 0 | a = 1

0 18 0 0 | b = 2

0 0 20 0 | c = 3

0 0 0 19 | d = 4

Vir: Program Weka

3. Priloga (Izsek)

Ker je celoten zapis dolg več kot dvesto strani, sem priložil samo glavne informacije iz celotnega zapisa.

=== Run information ===

Scheme: weka.classifiers.trees.LMT -l -1 -M 15 -W 0.0

Relation: sudoku

Instances: 1980

Attributes: 162

[list of attributes omitted]

Test mode: split 66.0% train, remainder test

=== Classifier model (full training set) ===

-//-

Time taken to build model: 75.86 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.09 seconds

=== Summary ===

Correctly Classified Instances	673	100 %
Incorrectly Classified Instances	0	0 %
Kappa statistic	1	
Mean absolute error	0.002	
Root mean squared error	0.0145	
Relative absolute error	1.0281 %	
Root relative squared error	4.6094 %	
Total Number of Instances	673	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	2
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	3
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	4
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	5
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	6
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	7
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	8
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	9
Weighted Avg.	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000

=== Confusion Matrix ===

```

a b c d e f g h i <-- classified as
98 0 0 0 0 0 0 0 0 | a = 1
0 91 0 0 0 0 0 0 0 | b = 2
0 0 81 0 0 0 0 0 0 | c = 3
0 0 0 60 0 0 0 0 0 | d = 4
0 0 0 0 78 0 0 0 0 | e = 5
0 0 0 0 0 64 0 0 0 | f = 6
0 0 0 0 0 0 69 0 0 | g = 7
0 0 0 0 0 0 0 61 0 | h = 8
0 0 0 0 0 0 0 0 71 | i = 9

```

Vir: Program Weka

7. VIRI IN LITERATURA

- Ploj, B. (2019). *Bionska umetna inteligenca: Knjiga o napravah, ki se učijo in se samostojno odločajo, pri čemer lahko presežejo tudi človeštvo*. Ptuj: Visoka šola na Ptuju. Pridobljeno 29. maj 2020 s http://www2.scptuj.si/~boploj/Bionska_umetna_inteligenca.pdf
- Deep Mind (2020). *Alpha Go*. Pridobljeno 29. maj 2020 s <https://deepmind.com/research/case-studies/alphago-the-story-so-far>
- Wikipedija (2020). *Umetna inteligenca*. Pridobljeno 28. februar 2020 s https://sl.wikipedia.org/wiki/Umetna_inteligenca
- Wikipedija (2018). *Strojno učenje*. Pridobljeno 28. februar 2020 s https://sl.wikipedia.org/wiki/Strojno_u%C4%8Denje
- Wikipedija (2020). *AlphaGo*. Pridobljeno 5. marec 2020 s <https://en.wikipedia.org/wiki/AlphaGo>
- Wikipedija (2019). *Master (software)*. Pridobljeno 5. marec 2020 s [https://en.wikipedia.org/wiki/Master_\(software\)](https://en.wikipedia.org/wiki/Master_(software))
- Wikipedija (2018). *Umetna inteligenca v igrah*. Pridobljeno 7. marec 2020 s https://sl.wikipedia.org/wiki/Umetna_inteligenca_v_igrah
- University of Waikato (2019). *WEKA*. Pridobljeno 7. marec 2020 s <https://www.cs.waikato.ac.nz/ml/weka/>
- Univerza v Ljubljani (2015). *Umetna inteligenca*. Pridobljeno 29. maj 2020 s <https://fri.uni-lj.si/sl/predmet/63720>
- Wikipedia (2020). *DeepMind*. Pridobljeno 29. maj 2020 s <https://en.wikipedia.org/wiki/DeepMind>
- Ploj, B. (2018). *It is time for change: New Deep Learning Algorithms beyond Backpropagation*. Pridobljeno 30. maj 2020 s https://www.researchgate.net/publication/322617800_New_Deep_Learning_Algorithms_beyond_Backpropagation_IBM_Developers_UnConference_2018_Zurich
- Ploj, B. (2017). *Umetna inteligenca – Nevronske mreže (3. del)*. Pridobljeno 30. maj 2020 s https://www.researchgate.net/publication/315897957_Umetna_inteligenca_-_Nevronske_mreze_3del
- Ploj, B., Zorman, M., Harb, R. (2013). *Border Pairs Method—constructive MLP learning classification algorithm*. Pridobljeno 30. maj 2020 s https://www.researchgate.net/publication/249011199_Border_Pairs_Method-constructive_MLP_learning_classification_algorithm It is time for change: