

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 346

**EVOLUCIJA NEURONSKIH MREŽA
ZA DETEKCIJU LICA**

Hrvoje Ban

Zagreb, lipanj 2012.

Zagreb, 5. ožujka 2012.

DIPLOMSKI ZADATAK br. 346

Pristupnik: **Hrvoje Ban**

Studij: Informacijska i komunikacijska tehnologija

Profil: Telekomunikacije i informatika

Zadatak: **Evolucija neuronskih mreža za detekciju lica**

Opis zadatka:

Opisati metode oblikovanja neuronskih mreža uz pomoć evolucijskih algoritama. Ostvariti programski sustav za oblikovanje neuronske mreže temeljen na algoritmu NEAT. Usporediti osnovnu inačicu algoritma s mogućim proširenjima s obzirom na konvergenciju i uspješnost neuronske mreže. Ispitati ostvarene algoritme na nekoliko jednostavnijih problema te za problem detekcije lica. Statistički analizirati uspješnost ostvarenih postupaka u usporedbi s postojećim rezultatima u literaturi. Radu priložiti algoritme, izvorne tekstove programa i rezultate uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

*Hvala profesoru Domagoju Jakoboviću za svu pomoć i
savjete tijekom prethodne tri godine i za iznimnu
količinu strpljenja kad su rokovi bili u pitanju ☺*

Sadržaj

Uvod	1
1 Detekcija lica na slikama	2
2 Umjetne neuronske mreže	5
2.1 Učenje neuronskih mreža.....	6
3 Evolucija neuronskih mreža	8
3.1 Neuroevolucija rastućih topologija.....	9
3.2 Algoritam RBF-NEAT.....	12
3.3 Neizravno kodiranje temeljeno na strukturi hiperkocke	13
4 Implementacija	15
4.1 Priprema slika.....	15
4.2 Učenje neuronskih mreža.....	19
5 Rezultati	22
Zaključak	36
Literatura	37
Sažetak	40
Summary	41

Uvod

Detekcija ljudskog lica na slikama zanimljiv je problem koji ima brojne primjene i početna je točka u rješavanju mnogih drugih problema. Određivanje lokacija na kojima se na slici pojavljuju ljudska lica omogućuje određivanje broja i rasporeda skupine ljudi. Nakon određivanja lokacije lica moguće je nastaviti s određivanjem emocija, prepoznavanjem identiteta osoba, lociranjem značajki lica kao što su usta ili oči. Problem detekcije lica može se dalje proširiti na problem praćenja lica čija se lokacija mijenja kroz slijed slika video zapisa.

Detekcija lica iznimno je zahtjevan problem zbog velikih različitosti koje općenito postoje u ljudskim licima, kao što su veličina očiju, nosa i usana te izrazi lica koji nastaju prilikom izražavanja emocija. Dodatni problemi nastaju zbog raznih artefakata medija koji se koristi: crno-bijele slike ili slike u boji različitih rezolucija, osvjetljenja, kontrasta i općenito kvalitete.

Ovaj rad podijeljen je u pet dijelova. U prvom poglavlju opisan je problem detekcije lica, različiti načini detekcije te je dan pregled nekoliko postojećih metoda za detekciju lica. U drugom poglavlju opisane su neuronske mreže koje se koriste kao klasifikator za detekciju lica. U trećem poglavlju opisani su evolucijski algoritmi te algoritmi koji se koriste za učenje neuronskih mreža: algoritam NEAT i modifikacije tog algoritma RBF-NEAT i HyperNEAT. U četvrtom poglavlju opisano je kako su prikupljene i obrađene slike koje su korištene za učenje te kako su algoritam NEAT i njegove modifikacije korištene za učenje neuronskih mreža. U petom poglavlju prikazani su i objašnjeni rezultati naučenih mreža.

1 Detekcija lica na slikama

Detekcija lica (engl. *face detection*) aktivno je područje istraživanja i brojne metode razvijene su tijekom prethodnih 20 godina (Hjelmås, 2001; Yang, 2002). Cilj je detekcije lica odrediti sadrži li određena slika ljudska lica te odrediti lokacije tih lica. Detekcija lica mora raditi s velikim rasponom varijacija na slikama: različiti pogledi na lice (frontalno, polu-profil, profil), izrazi lica i emocija, različite orijentacije i rotacije slike, različita osvjetljenja pod kojim je slika napravljena, osobe s bradom, brkovima ili naočalama, slike u boji i crno-bijele slike, slike loše kvalitete itd.

Lokalizacija lica (engl. *face localization*) pojednostavljeni je slučaj detekcije kada je poznato da slika sadrži točno jedno lice i cilj je odrediti lokaciju tog lica. Detekcija lica preduvjet je za mnoge druge probleme analize slika ljudskog lica. Prepoznavanje izraza lica (engl. *facial expression recognition*) nastoji prepoznati izraze sreće, tuge, straha i drugih emocija. Detekcija karakteristika lica (engl. *facial feature detection*) koristi se za otkrivanje lokacija značajki kao što su oči, obrve, nos ili usta. Metode za prepoznavanje lica (engl. *face recognition*) nastoje povezati sliku osobe s njezinim stvarnim identitetom. Praćenje lica (engl. *face tracking*) koristi se za neprekinuto praćenje lokacije i orijentacije ljudskog lica kroz slijed slika video zapisa.

Metode detekcije lica mogu se podijeliti u četiri skupine (Yang, 2002):

- Metode temeljene na znanju (engl. *knowledge-based*): ljudsko znanje o tome što čini tipično lice predstavlja se skupom programskih pravila. Pravila obično opisuju odnose položaja i veličina različitih karakteristika lica. Obično se koriste samo pri lokalizaciji lica;
- Metode temeljene na nepromjenjivim značajkama (engl. *feature invariant*): prepoznavanje značajki u strukturi lica koje su prisutne bez obzira na

orijentaciju lica, kut gledanja i uvjete osvjetljenja. Koriste se samo pri lokalizaciji lica;

- Korištenje predložaka lica (engl. *template matching*): slika se uspoređuje s jednim ili više standardnih predložaka koji opisuju cijelo lice ili samo određene izraze. Konačna odluka donosi se na temelju korelacije između slike i predložaka. Koristi se pri lokalizaciji i detekciji lica;
- Metode temeljene na izgledu (engl. *appearance-based*): konstruira se model za detekciju lica ili izraza lica učenjem na temelju reprezentativnih skupova slika koje sadrže lica i slika koje ne sadrže lica.

Slika koja se ispituje može se predstaviti vektorom, a svaki piksel slike onda se tumači kao jedna dimenzija vektora (Sung, 1996). Slike (vektori) lica, koje se koriste za učenje, grupiraju se u grozdove. Rezultat je skup višedimenzionalnih Gaussovih grozdova s lokacijom centroide i kovarijacijskom matricom. Za detekciju lica na novim slikama računa se udaljenost vektora nove slike od prije izračunatih grozdova. Dobivene udaljenosti se prosljeđuju višeslojnom perceptronu koji donosi konačnu odluku.

Jedan od prvih pokušaja korištenja neuronske mreže za detekciju lica (Rowley, 1999) primjer je uspješne implementacije metode temeljene na izgledu. Slike koje se ispituju razdvajaju se na segmente pravokutnih oblika čime je problem pojednostavljen. Za učenje mreže korišten je algoritam *backpropagation*, a za naknadno povećavanje uspješnosti detekcije korišteno je više mreža te se različitim kombiniranjem njihovih izlaza donosi konačna odluka o detekciji.

Umjesto obične neuronske mreže može se koristiti konvolucijska neuronska mreža koja se sastoji od ulaznog i nekoliko konvolucijskih slojeva (Garcia, 2002). Konvolucijski sloj sadrži više ravnina: skupina neurona kojih ima devet puta manje nego ulaznih neurona. Ulazna se slika dijeli na regije veličina 3x3 piksela i svi neuroni takve regije spajaju se na jedan konvolucijski neuron. Drugi konvolucijski sloj računa konvoluciju prvog sloja itd. Na taj se način učenjem automatski dobiva računanje određenih obilježja na slici. Zadnja dva sloja predstavljaju višeslojni perceptron koji donosi konačnu odluku na temelju izračunatih obilježja.

Wiegand (2004) je istražio mogućnost korištenja jednostavne evolucijske strategije za paralelni razvoj strukture mreže i težina između neurona. Nova generacija dobiva se kopiranjem trenutne generacije. Kopije se mutiraju mijenjanjem težina, dodavanjem novih ili brisanjem postojećih neurona te dodavanjem ili brisanjem veza.

Dubey (2011) je koristio standardni genetski algoritam za učenje mreže s fiksnom strukturom. Skup slika nad kojima se mreža uči u početku sadrži samo slike s licima. Svakih 100 generacija dobivena mreža ispituje se na skupu svih slika koje ne sadrže lice. Do 250 takvih slika koje je mreža krivo klasificirala dodaju se u skup za učenje kao negativni primjerci.

Iznimno brza detekcija može se postići korištenjem Harrovih obilježja (Viola, 2003). Zbog velikog broja mogućih obilježja koristi se skup slabih klasifikatora koji prepoznaju po jedno obilježje. Slabi klasifikatori koriste se za izradu mnogo jačeg klasifikatora pomoću algoritma AdaBoost. Za bržu detekciju koristi se kaskada jakih klasifikatora: prvo se primjenjuju klasifikatori koji se brzo izvode i odbacuju veliki postotak slika bez lica, a na kraju klasifikator koji ima visoku preciznost i koji donosi konačnu odluku.

Umjesto Harrovih obilježja mogu se koristiti anizotropni Gaussovi filtri (Meynet, 2007). Ovi se filtri također koriste za pronalaženje i računanje obilježja slike. Prvo se razvija skup slabih klasifikatora koji prepoznaju samo prag i parnost filtra. Jači klasifikator dobiva se kombinacijom slabih klasifikatora pomoću algoritma *AdaBoost*.

Strojevi s potpornim vektorima (engl. *support vector machine, SVM*) još su jedna vrsta klasifikatora koji se mogu koristiti za detekciju lica (Heisele, 2003). Stroj prima tri vrste ulaza: sliku s normaliziranim histogramom, sliku na koju je primijenjen Sobel filter za detekciju rubova i sliku konvoluiranu s tri različite maske za računanje Harr valića (engl. *wavelet*). Detekcija lica obavlja se u dva koraka: prvo se otkrivaju karakteristična obilježja kao što su oči, nos i usta te se lokacije obilježja analiziraju kako bi se odredilo odgovara li geometrijska konfiguracija tih obilježja ljudskom licu.

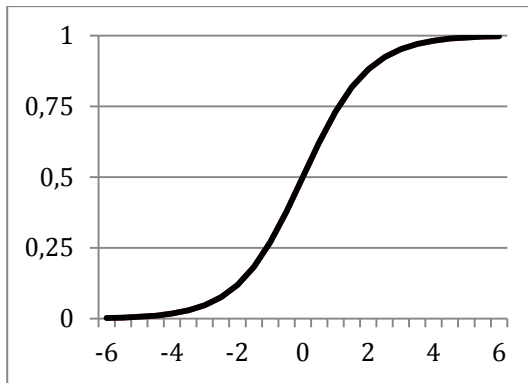
2 Umjetne neuronske mreže

Umjetne neuronske mreže (engl. *artificial neural networks*) su model računanja čiji se rad zasniva na oponašanju prirodnih neuronskih mreža. Neuronska mreža sastoji se od skupa neurona i jednosmjernih težinskih veza koje ih povezuju. Neuroni se mogu podijeliti na tri vrste: ulazni, izlazni i skriveni. Svaki neuron ima svoju aktivacijsku vrijednost: za ulazne neurone te su vrijednosti predodređene, dok se za ostale neurone aktivacijska vrijednost računa kao:

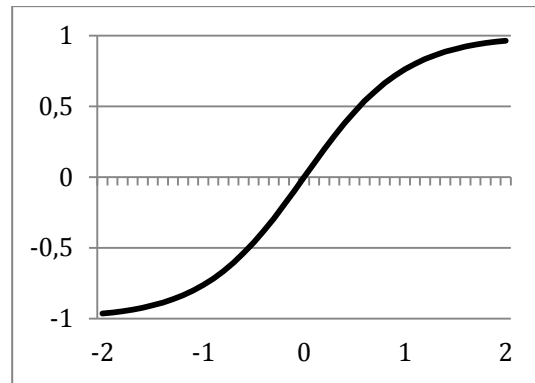
$$f_j(x) = K \left(\sum_i w_{ij} g_i(x) \right) \quad (2.1)$$

gdje je x skup aktivacijskih vrijednosti ulaznih neurona, w_{ij} težina veze između i -tog i j -tog neurona, g_i aktivacijska vrijednosti i -tog neurona, a K aktivacijska funkcija. Aktivacijske vrijednosti ulaznih neurona predstavljaju parametre računanja, dok izlazne vrijednosti predstavljaju rezultat. Neuronska mreža može se promatrati kao funkcija s višedimenzionalnom domenom (ulaznim neuronima) i slikom (izlaznim neuronima). Aktivacijske vrijednosti skrivenih neurona su međurezultati koji nisu sami po sebi važni pa se ignoriraju tj. ostaju skriveni.

Za aktivaciju se mogu koristiti praktički sve funkcije, s iznimkom algoritama učenja koji nameću određene uvjete (npr. postojanje prve derivacije). Ipak, obično se koriste funkcije koje su monotono rastuće i približno linearne u području domene oko nule. Za aktivacijske funkcije poželjno je da imaju svojstvo sažimanja, tj. mapiranja sa skupa svih realnih brojeva $\mathbb{R}$ na ograničeni podskup, najčešće $(0, 1)$ ili $(-1, 1)$, jer se time ograničava raspon vrijednosti s kojima neuroni računaju. Često korištene aktivacijske funkcije koje zadovoljavaju te uvjete (i za koje dodatno postoji prva derivacija) su sigmoidne funkcije kao što su logistička funkcija (2.2) i tangens hiperbolni (2.3).



$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$



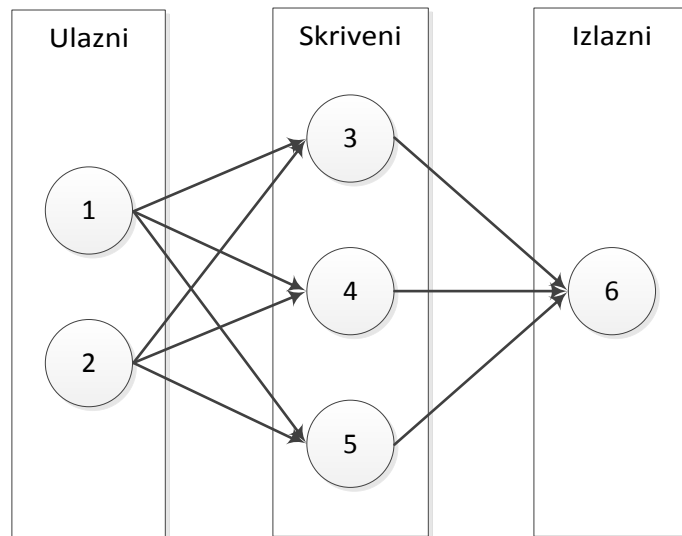
$$f(x) = \text{th}(x) \quad (2.3)$$

Dok je broj ulaznih i izlaznih problema određen problemom koji se rješava, broj skrivenih neurona, njihovu međusobnu povezanost te broj i težinu veza potrebno je zasebno odabrati. Ovisno o algoritmu koji se koristi za razvoj mreže, struktura može biti potpuno slobodna ili na određeni način ograničena.

2.1 Učenje neuronskih mreža

Algoritam *backpropagation* (Rumelhart, 1986), jedan od prvih uspješnih algoritama za razvoj neuronskih mreža, traži da se mreža podijeli na najmanje tri sloja: sloj ulaznih neurona, sloj izlaznih neurona te jedan ili više slojeva skrivenih neurona. Slika 2.1 prikazuje primjer jedne takve mreže. Neuroni jednog sloja potpuno su povezani samo s neuronima sljedećeg sloja. Učenje mreže, tj. traženje odgovarajućih težina veza provodi se iterativnim računanjem greške izlaza mreže za ulazne vrijednosti iz skupa za učenje (engl. *training set*) te propagiranjem greške unatrag kroz veze između neurona, po čemu je algoritam dobio ime. Osim skupa za učenje, koristi se još i skup za validaciju (engl. *validation set*) kojim se provjerava sposobnost generalizacije mreže tj. sprječava se da mreža jednostavno „memorira“ skup za učenje. Skup za ispitivanje (engl. *testing set*) koristi se kada je učenje završeno kako bi se izmjerile performanse neuronske mreže.

Iako noviji algoritmi za učenje neuronskih mreža nemaju ograničenja na strukturu mreža, višeslojna struktura zadržala se zbog jednostavnosti modeliranja: jedini parametri su broj slojeva skrivenih neurona te broj neurona u pojedinim slojevima.



Slika 2.1: Neuronska mreža s višeslojnom strukturom

Ako se unaprijed odredi struktura neuronske mreže tj. broj i povezanost neurona, problem učenja mreže svodi se samo na izbor težina veza. Ako se težine predstavljaju vektorom ili matricom brojeva, tada se mogu koristiti optimizacijski algoritmi koji traže optimalne vrijednosti brojeva, kao što npr. evolucijski algoritmi.

3 Evolucija neuronskih mreža

Evolucijski algoritmi (engl. *evolutionary algorithms*) su skup metaheuristika koje se koriste za rješavanje optimizacijskih problema. Metaheuristike su algoritmi koji iterativno traže bolje rješenje za zadani problem poboljšavajući postojeće rješenje ili više njih. S obzirom na to da ne uzimaju u obzir specifičnosti problema primjenjivi su za široki skup optimizacijskih problema.

Rad evolucijskih algoritama temelji se na principima prirodne evolucije: nad skupom potencijalnih rješenja (populacijom) primjenjuje se selekcija (engl. *selection*) kojom se biraju dobre jedinke (preživljavanje najjačih), te se nova rješenja dobivaju križanjem (engl. *crossover*) i mutacijom (engl. *mutation*) odabranih rješenja. Slično kao i kod prirodne evolucije, kombiniranjem i mijenjanjem postojećih rješenja moguće je dobiti nova, bolja rješenja. Primjeri evolucijskih algoritama su:

- Genetski algoritmi (engl. *genetic algorithms*) rješenja modeliraju kao genome koji mogu biti nizovi bitova, vektori i matrice brojeva, permutacije skupova i dr. Jedino je potrebno da se nad rješenjima mogu primijeniti operacije mutacije (izmjene rješenja) i križanja (generiranja novog rješenja iz dva postojeća);
- Genetsko programiranje (engl. *genetic programming*) koristi se za razvijanje funkcija i računalnih programa (Koza, 1992) koji se najčešće predstavljaju kao stablaste strukture zbog jednostavnosti implementacije operatora mutacije i križanja ili kao linearni programi koji su sličniji stvarnom strojnom kodu;
- Diferencijalna evolucija (engl. *differential evolution*) rješenja modelira kao vektore realnih brojeva (Storn, 1997). Nova rješenja dobivaju se iz postojećih dodavanjem razlike dvaju (slučajno ili ciljano) odabranih vektora;
- Optimizacija rojem čestica (engl. *particle swarm optimization*) predstavlja rješenja točkama u višedimenzionalnom prostoru (Kennedy, 1995). Gibanje

točaka kroz prostor simulira ponašanje ptica u prirodi koje svoje gibanje određuju prema gibanju ostalih ptica u jatu;

- Umjetni imunološki sustav (engl. *artificial immune system*) razvija rješenje kao što prirodni imunološki sustav razvija antitijela (Cutello, 2002). Rješenja se kopiraju te se stvara više kopija što je rješenje bolje, s tim da kopije nisu identične originalu već se malo razlikuju. Kopije koje su zbog promjena bolje od originala se zadržavaju u populaciji i dalje kopiraju.

Svi evolucijski algoritmi imaju sličan način rada. Sve se jedinke populacije ocjenjuju kako bi se odredila njihova dobrota (engl. *fitness*) tj. koliko dobro rješavaju problem. Na temelju dobrote biraju se jedinke te se ovisno o algoritmu uzima dio ili cijela populacija, a određena jedinka može biti izabrana jedan ili više puta. Izabrane jedinke se mutiraju ili se križaju kako bi se dobila nova rješenja koja se također mogu podvrgnuti operaciji mutacije. Populacija novih rješenja ili potpuno mijenja staru populaciju ili se na neki način spaja s njome, na primjer zadržavanjem samo određenog broja najboljih jedinki.

3.1 Neuroevolucija rastućih topologija

Za razliku od svih dosad spomenutih algoritama, neuroevolucija rastućih topologija (engl. *neuroevolution of augmenting topologies*, NEAT) istovremeno razvija i strukturu mreže i težine veza između neurona (Stanley, 2004). NEAT spada u neuroevolucijske genetske algoritme: svaki genom sadrži direktno kodirani opis strukture neuronske mreže. Direktno kodiranje znači da genom sadrži eksplicitne informacije o neuronima i vezama između njih, za razliku od indirektnog kodiranja koje sadrži pravila na temelju kojih je moguće konstruirati neuronsku mrežu.

S obzirom na to da je cilj algoritma NEAT otkrivanje optimalne strukture mreže, sve mreže počinju s minimalnom strukturom – mreže ne sadrže skrivene neurone, već samo potpuno povezane ulazne i izlazne neurone. U slučaju mreža s velikim brojem ulaznih ili izlaznih neurona, umjesto potpune može se početi samo s djelomičnom povezanosti neurona pri čemu algoritam onda može odrediti postoje li ulazni neuroni koji nisu potrebni ili relevantni za rješavanje problema jer NEAT neće dodavati veze koje bi ih povezale sa skrivenim i izlaznim neuronima.

Slika 3.1 prikazuje shemu genoma koji sadrži dva skupa gena: za neurone i za veze. Geni neurona definiraju koji sve ulazni, izlazni i skriveni neuroni postoje u strukturi mreže. Svakom neuronu dodijeljen je jedinstveni broj inovacije (mutacije kojom je dodan u strukturu mreže). Geni veza definiraju povezanost neurona mreže: gen veze sadrži jedinstveni broj inovacije veze, jedinstvene brojeve neurona koji su povezani, težinu veze i zastavicu koja označava je li gen aktivan (jedino će veze aktivnih gena postojati u konačnoj strukturi mreže).

Algoritam NEAT koristi dvije mutacije za proširivanje strukture mreže: dodavanje nove veze i dodavanje novog neurona. Nova veza dodaje se između dva neurona koji još nisu povezani i dodjeljuje joj se slučajno odabrana težina. Novi neuron dodaje se na mjesto postojeće veze. Postojeća se veza deaktivira i dodaju se dvije nove veze: prvoj vezi koja spaja početni neuron stare veze i novi neuron dodjeljuje se težina 1, dok se drugoj vezi koja spaja novi neuron i krajnji neuron stare veze dodjeljuje težina stare veze. Ovakav način dodavanja novog neurona i dodjeljivanja težina minimizira utjecaj mutacije na ponašanje neuronske mreže. Zbog korištenja grupiranja, mreža će imati vremena optimizirati i iskoristiti novo proširenje strukture.

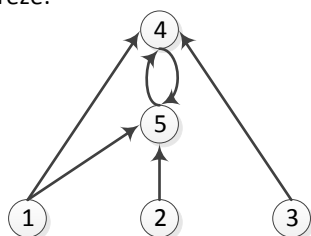
Geni čvorova:

Čvor 1	Čvor 2	Čvor 3	Čvor 4	Čvor 5
Ulazni	Ulazni	Ulazni	Izlazni	Skriveni

Geni veza:

Ulaz: 1 Izlaz: 4 Težina: 0,4 Aktivan: Da Inovacija: 1	Ulaz: 2 Izlaz: 4 Težina: -0,5 Aktivan: Ne Inovacija: 2	Ulaz: 3 Izlaz: 4 Težina: 0,5 Aktivan: Da Inovacija: 3	Ulaz: 2 Izlaz: 5 Težina: 0,2 Aktivan: Da Inovacija: 4	Ulaz: 5 Izlaz: 4 Težina: 0,4 Aktivan: Da Inovacija: 5	Ulaz: 1 Izlaz: 5 Težina: 0,6 Aktivan: Da Inovacija: 6	Ulaz: 4 Izlaz: 5 Težina: 0,6 Aktivan: Da Inovacija: 7
---	--	---	---	---	---	---

Struktura mreže:



Slika 3.1: Primjer NEAT genoma i mreže koju upisuje

Svaki gen neurona i veze ima jedinstveni broj inovacije koji se koristi za određivanje zajedničkih dijelova dviju različitih mreža. Tijekom rada algoritma pamti se skup svih

inovacija (mutacije koje mijenjaju strukturu mreže) i svakoj se dodjeljuje jedinstveni broj. Kada se prvi put dodaje nova veza između dva određena neurona, u skup inovacija dodaje se mapiranje s para brojeva inovacije neurona na broj inovacije nove veze (novi brojevi inovacije mogu se generirati npr. uvećavanjem globalnog brojača). Pri dodavanju veze između dva neurona istih brojeva inovacije kod druge mreže, za tu vezu koristiti će se isti broj inovacije. Isti postupak koristi se pri dodavanju novog neurona, pri čemu se dodaje mapiranje s broja inovacije veze koja se zamjenjuje na broj inovacije novog neurona. Početnim neuronima i vezama između njih mogu se dodijeliti proizvoljni brojevi.

Nove inovacije obično prvo smanje sposobnost mreže za rješavanje problema dok se odgovarajuće ne optimiziraju podešavanjem težina veza i povezanosti s ostatkom strukture mreže. Kako NEAT, kao i svi evolucijski algoritmi, koristi dobrotu kao pokazatelj koja rješenja treba nastaviti razvijati, mreže s novim inovacijama bile bi brzo eliminirane zbog smanjene dobrote. Zbog toga algoritam NEAT koristi grupiranje mreža po sličnosti strukture. Mreže koje dijele veliki broj zajedničkih elemenata pripadati će istoj grupi i natjecati će se samo s drugim mrežama unutar te grupe. Ako se uzme u obzir da je vjerojatnost mutacija koje mijenjaju strukturu obično vrlo mala, skoro svaka nova inovacije uzrokovati će stvaranje nove grupe unutar koje se može slobodno razvijati. Za svaku grupu prati se i vrijeme stagnacije – broj iteracija tijekom kojih se jedinkama grupe ili samo najboljoj jedinki nije poboljšala dobrota. Nakon što vrijeme stagnacije pređe unaprijed određenu granicu, nova se inovacija smatra neuspješnom i sve jedinke grupe brišu se iz populacije.

Na kraju svake iteracije jedinke nove populacije (djeca) generiraju se iz postojećih jedinki (roditelja). Djeca se generiraju iz grupa, pri čemu je broj jedinki koje se generiraju iz određene grupe proporcionalan prosječnoj dobroti članova te grupe. Generiranje nove jedinke započinje nasumičnim izborom jednog roditelja. Unaprijed određenom vjerojatnošću bira se između križanja i mutacije. U slučaju mutacije izabrani roditelj se kopira i kopija se mutira. U slučaju križanja iz grupe se nasumično bira drugi roditelj (različit od prvog) uz malu vjerojatnost biranja roditelja iz druge grupe. Kod križanja dvaju roditelja gledaju se samo geni veza (skup gena neurona može se jednostavno rekonstruirati iz gena veza) te se geni grupiraju u one koji su zajednički za oba roditelja i one koji pripadaju samo jednom. Za svaki zajednički gen,

dijete dobiva gen jednog od roditelja ili kombinaciju: za težinu se uzima aritmetička sredina težina roditeljskih gena, a za zastavicu aktivnosti preuzima se vrijednost iz jednog roditeljskog gena. Za gene koji nisu zajednički za oba roditelja preuzimaju se samo geni koji pripadaju boljem roditelju.

Nakon što su generirani, geni djeteta podvrgavaju se mutaciji. Tri moguće mutacije su dodavanje nove veze, dodavanje novog neurona i mijenjanje težina postojećih veza. Postoje različiti načini mijenjanja težina, ovisno o načinu na koji se biraju veze koje će se mijenjati i načinu mijenjanja njihovih težina:

- mijenjanje svih veza uz određenu vjerojatnost,
- mijenjanja određenog broja veza,
- mijenjanje samo novijih veza (pretpostavlja se da su za starije gene veza već pronađene optimalne vrijednosti težina),
- dodavanjem vrijednosti iz ravnomjerne raspodjele,
- dodavanjem vrijednosti iz normalne raspodjele i
- kombinacije prethodnih vrijednosti: za svaku mutaciju nasumično se odabire način biranja veza i mijenjanja njihovih težina.

3.2 Algoritam RBF-NEAT

Algoritam RBF-NEAT modifikacija je osnovne inačice algoritma NEAT koji za skrivene neurone, osim sigmoidne aktivacijske funkcije, koristi i radijalne bazne (engl. *radial basis*) funkcije (Kohl, 2009). Radijalne bazne funkcije su funkcije čije vrijednosti ovise jedino o udaljenosti od izvorišta ili neke druge točke koja se naziva centar funkcije. Primjeri takvih funkcija su Gaussova, višekvadratična, inverzna kvadratična i druge. Prilikom dodavanja novog skrivenog neurona u strukturu mreže, algoritam RBF-NEAT s određenom vjerojatnošću bira između sigmoidne i Gaussove funkcije. U slučaju Gaussove funkcije na slučajno odabrane vrijednosti postavlja se očekivana vrijednost i disperzija funkcije. Algoritam se dodatno proširuje novom mutacijom koja mijenja vrijednost ta dva parametra na sličan način kao što se mijenjaju težine veza.

3.3 Neizravno kodiranje temeljeno na strukturi hiperkocke

Neuroevolucija rastućih topologija korištenjem neizravnog kodiranja temeljenog na strukturi hiperkocke (engl. *hypercube-based neuroevolution of augmenting topologies*, HyperNEAT) je metoda razvoja neuronskih mreža s vrlo velikim brojem veza (Stanley, 2009). HyperNEAT koristi neizravno kodiranje strukture neuronskih mreža pomoću kompozicijskih mreža za generiranje uzoraka povezanosti (engl. *connective compositional pattern producing networks, connective CPPN*) koje strukturu mreže predstavljaju kao funkciju Kartezijevog koordinatnog sustava.

Rad algoritma HyperNEAT inspiriran je činjenicom da je mapiranje između biološkog genotipa i konačnog rezultata (fenotipa) također neizravno: na primjer ljudski genom od 30.000 gena kodira više od 100 bilijuna veza ljudskog mozga. Kako struktura fenotipa često sadrži ponavljajuće uzorke, za svako ponavljanje specifikacija uzorka može se dekodirati iz iste skupine gena. Brojne simetrije u anatomiji kralježnjaka, receptori vidne kore mozga te prsti ruku i nogu samo su neki od brojnih primjera ponavljanja uzoraka u prirodi.

CPPN mreže generiraju prostorne uzorke koristeći kompoziciju osnovnih funkcija. Za razliku od umjetnih neuronskih mreža koje najčešće koriste samo jednu, obično sigmoidnu, aktivacijsku funkciju, CPPN mreže koriste širi skup funkcija kao što su linearne, pilaste (engl. *sawtooth*), sinusoidne i Gaussove funkcije. Takve funkcije omogućavaju generiranje različitih korisnih uzoraka koji sadrže simetrije, nesavršene simetrije i ponavljanja. Slika 3.2 prikazuje primjer nekoliko dvodimenzionalnih slika generiranih CPPN mrežama koje demonstriraju redom: savršenu simetriju oko horizontalne osi, nesavršenu simetriju oko vertikalne osi i ponavljanje uzoraka.



Slika 3.2: Primjeri pravilnosti koje mogu generirati CPPN mreže (Stanley, 2009)

Algoritam HyperNEAT paralelno razvija dvije neuronske mreže za svaku jedinku. Prva mreža je obična neuronska mreža čija je struktura unaprijed određena i ne može se mijenjati. Svakom neuronu mreže dodijeljene su koordinate u višedimenzionalnom prostoru. Koordinate neurona obično se biraju tako da se nalaze unutar jedinične hiperkocke. Zbog jednostavnosti se često koristi trodimenzionalni prostor jer je lako vizualizirati smještaj neurona u prostoru.

Za takvu neuronsku mrežu može se koristiti klasična višeslojna struktura. U slučaju trodimenzionalne kocke, izdvoji se onoliko dvodimenzionalnih presjeka koliko ima slojeva mreža. Presjeci su paralelni s jednom ravninom koordinatnog sustava tako da svi neuroni istog sloja imaju jednu zajedničku koordinatu. Na primjer, ako su presjeci paralelni s ravninom xy , tada će se neuroni dva sloja razlikovati po z koordinati.

Druga mreža je CPPN mreža koja se razvija algoritmom NEAT s proširenim skupom aktivacijskih funkcija. CPPN mreža ima po dva ulazna neurona za svaku dimenziju prostora u kojem se nalaze neuroni prve mreže i jedan izlazni neuron. Ova se mreža koristi za računanje težina veza između neurona prve mreže tako da se za svaki par neurona njihove koordinate postavljaju za vrijednosti ulaznih neurona CPPN mreže, a izlaz mreže predstavlja vrijednost težine veze.

Za razliku od ostalih algoritama za učenje neuronskih mreža, algoritam HyperNEAT ima pristup podacima o prostornom rasporedu ulaza što mu omogućuje bolje određivanje prikladnih težina veza. Na primjer, ako su ulazi mreže vrijednosti piksela dvodimenzionalne slike, algoritmu će biti poznato koliko su pikseli međusobno udaljeni i kako su raspoređeni u odnosu na druge piksele slike. Oslanjanje na podatke o koordinatama neurona je istovremeno prednost i nedostatak algoritma. Raspored koji loše opisuje stvarne odnose između ulaznih vrijednosti imati će negativan utjecaj na performanse razvijenih mreža (Clune, 2009).

Različiti uzorci koje mogu generirati CPPN mreže omogućavaju algoritmu HyperNEAT prilagođavanje učene neuronske mreže karakteristikama ulaza. Razne simetrije i ponavljanja koja postoje među ulaznim vrijednostima mogu se izravno preslikati u vrijednosti težina veza.

4 Implementacija

Prvi korak u procesu učenja neuronske mreže za detekciju lica je prikupljanje i obrada slika koje će biti u skupovima za učenje i ispitivanje. Potrebne su dvije vrste slika: one koje sadrže ljudsko lice (za koje mreža treba dati pozitivan odgovor) i one koje ne sadrže ljudsko lice (za koje mreža treba dati negativan odgovor).

Za slike ljudskih lica korištena je baza FERET (*Face Recognition Technology*) koja sadrži više tisuća slika ljudi slikanih iz nekoliko kutova u različitim uvjetima osvjetljenja (Phillips, 1998; Phillips, 2000). Slike prikazuju ljude s neutralnim i nasmiješenim izrazima lica, ljude različitih dobnih skupina, s naočalama, bradom ili brkovima, ljude oba spola i različitih rasa. Za većinu slika dostupni su podaci o lokaciji određenih značajki lica: očiju, usta i nosa. Za proces učenja korištene su sve slike iz baze FERET osim slika ljudi s naočalama, bradom i brkovima te slika koje nemaju podatke o lokaciji značajki lica. Ukupno su za postupak učenja korištene 1.994 slike.

4.1 Priprema slika

S obzirom na to da baza FERET sadrži slike u boji kao i crno-bijele slike, sve slike pretvorene su u crno-bijeli format. Svjetlina (L) slike računa se iz komponenti za crvenu (R), zelenu (G) i plavu (B) bolju po YUV modelu (4.1) jer takav model daje vjerniji prikaz svjetline od aritmetičke sredine sve tri komponente boje.

$$L = 0,299 R + 0,587 G + 0,114 B \quad (4.1)$$

Prikupljene slike zatim su odrezane tako da sadrže samo područje lica koje omeđuje oči i usta. Ograničenje o postavljanju podataka o koordinatama tih značajki uvedeno je kako bi se taj korak mogao automatizirati. Slike su odrezane tako da razmak između očiju iznosi 50% ukupne širine slike te da oči zauzimaju gornjih 20% slike. Ti postoci

su izabrani jer tako odrezane slike sadrže sve bitne značajke ljudskog lica (oči, nos i usta). Odrezane slike potom se smanjuju na veličinu 20x20 piksela korištenjem bikubične interpolacije. Slika 4.1 prikazuje nekoliko primjera slika iz baze FERET nakon što su odrezane i pretvorene u crno-bijeli format i njihove umanjene inačice.



Slika 4.1: Primjeri korištenih slika lica iz baze FERET

Nakon odrezivanja i smanjivanja slika potrebno je riješiti problem nejednolikog osvjetljenja. Iako je moguće ostaviti neuronskoj mreži da nauči prepoznavati lica snimljena pod različitim uvjetima osvjetljenja, bolje je obradom slika eliminirati varijacije u osvjetljenju jer se time smanjuje potrebna kompleksnost mreže (broj skrivenih neurona i dodatnih veza) te postupak učenja postaje brži.

Slika lica prije obrade može se shvatiti kao dvodimenzionalni signal koji je jednak zbroju izvorne slike (s jednolikim osvjetljenjem) i šuma (varijacija osvjetljenja). Da bi se dobila izvorna slika potrebno je izračunati vrijednosti signala šuma i oduzeti ih od pošumljene slike. Varijacije osvjetljenja moguće je predstaviti linearnim modelom koji pretpostavlja da se osvjetljenje po slici mijenja konstantom brzinom pod određenim kutom (Sung, 1996). Iako ovaj model ne predstavlja uvijek najbolje rješenje (pogotovo ako osvjetljenje ima više izvora ili se varijacije ne mijenjaju konstantom brzinom), njegova je prednost što se može vrlo brzo izračunati zbog malog broja parametara i jer je greška modela često prihvatljivo mala.

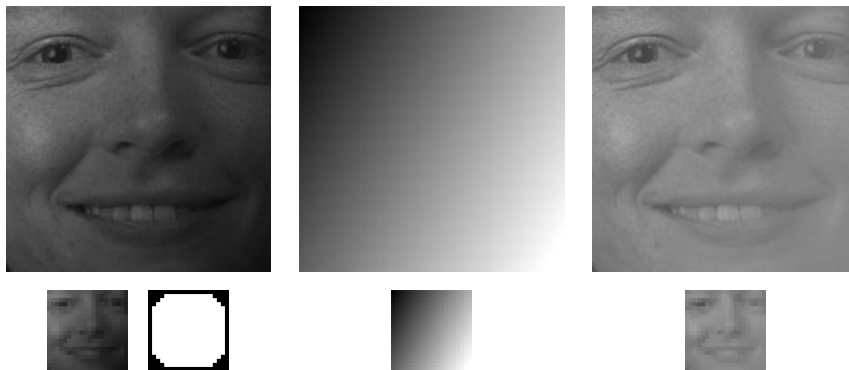
Linearni model $I(x, y)$ daje vrijednost osvjetljenja u točki slike s koordinatama (x, y) , a potpuno je određen je s tri parametra (4.2): brzinom promjene u smjeru osi x (a), brzinom promjene u smjeru osi y (b) i konstantom (c). Ovakav model odgovara plohi u trodimenzionalnom prostoru: parametri a i b određuju nagib plohe u odnosu na plohu xy , a parametar c određuje visinu plohe iznad središta koordinatnog sustava.

Kako su poznate vrijednosti svjetline pošumljene slike koja se sastoji od 400 piksela, za tri parametra modela šuma dobiva se ukupno 400 linearno nezavisnih jednadžbi. Kako je broj jednadžbi veći od broja nepoznanica (parametara) takav sustav nema jedinstveno rješenje, ali je moguće koristiti metodu najmanjih kvadrata za računanje vrijednosti parametara koji minimiziraju sumu kvadrata pogrešaka. Za ovaj konkretni problem metoda najmanjih kvadrata svodi se na rješenja matrične jednadžbe (4.3), gdje je z_i vrijednost svjetline i -tog piksela slike, a x_i i y_i koordinate tog piksela.

$$I(x, y) = [x \quad y \quad 1] \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (4.2)$$

$$\sum_i \begin{bmatrix} x_i^2 & x_i y_i & x_i \\ x_i y_i & y_i^2 & y_i \\ x_i & y_i & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \sum_i \begin{bmatrix} x_i z_i \\ y_i z_i \\ z_i \end{bmatrix} \quad (4.3)$$

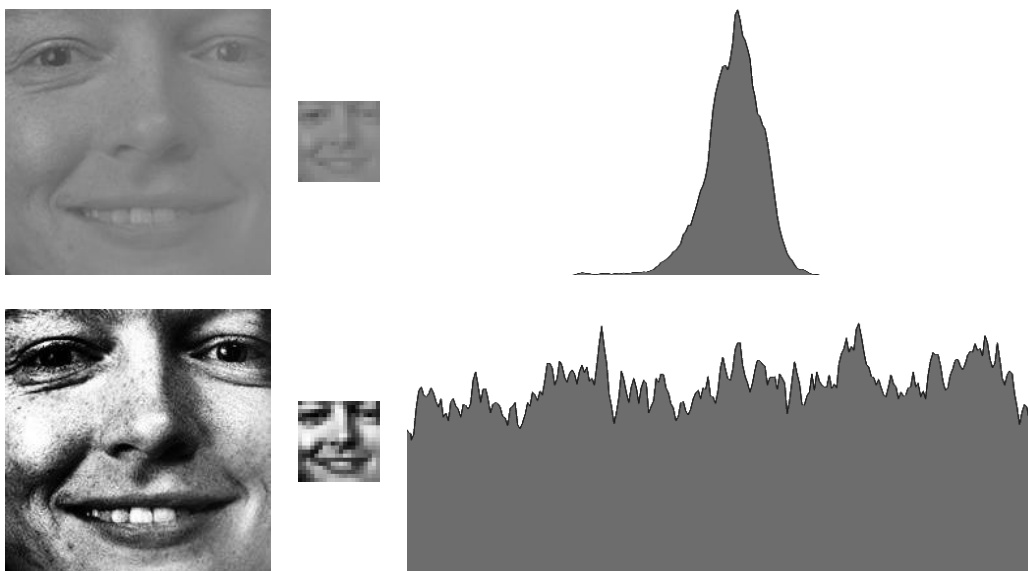
Prije računanja linearnog modela na pošumljenu sliku stavlja se maska koja prekriva rubove slike (1 piksel debljine) i kutove (3 piksela debljine). Ti dijelovi slike često sadrže artefakte poput zidova i tamnih sjena koji bi loše utjecale na računanje modela osvjetljenja. Svi pikseli koji se nalaze ispod maske se zanemaruju. Nakon računanja modela, vrijednosti šuma se oduzimaju od slike, pri čemu se ne koristi parametar konstante modela c jer je cilj ukloniti samo varijacije u osvjetljenju, a ne potpuno eliminirati osvjetljenje slike. Slika 4.2 prikazuje tri slike u punoj i umanjenoj verziji: pošumljenu sliku zajedno s maskom, izračunate vrijednosti modela varijacija osvjetljenja i konačnu sliku dobivenu uklanjanjem varijacija.



Slika 4.2: Postupak uklanjanja varijacija osvjetljenja

Zadnji korak obrade slike je normalizacija histograma. Histogram pokazuje koliko su pojedine vrijednosti svjetline zastupljene na slici. Slike iz baze FERET, pogotovo nakon

uklanjanja varijacija osvjetljenja imaju histogram malog raspona: slika sadrži svjetline iz ograničenog spektra. Normalizacija je postupak obrade slike kojim se dobiva histogram koji se rasprostire kroz cijeli raspoloživi spektar čime se ističu detalji i dobiva slika većeg kontrasta. Slika 4.3 prikazuje sliku i njezin histogram prije i poslije postupka normalizacije histograma.



Slika 4.3: Slike i pripadajući histogrami prije i poslije normalizacije

Za skup slika koje ne sadrže ljudska lica korištene su slike koje dolaze s operacijskim sustavom Windows 7. Slike prikazuje prirodu, zgrade, a također postoji i nekoliko apstraktnih slika. Svaka slika izrezana je na dijelove veličine 20x20 piksela, tako da se ti dijelovi ne preklapaju i pokrivaju cijelu sliku. Slika je potom umanjena za faktor $\sqrt{2}$ i ponovno izrezana na dijelove. Postupak se ponavlja sve dok je veličina slike veća od 20x20 piksela.

Dodatno, skup je proširen slikama koje sadrže lica koje su također iterativno umanjene i isječene, tako da neuronska mreža nauči ignorirati dijelove lica te da prepozna samo prozore koji sadrže cijelo ljudsko lice. Nad svim prikupljenim prozorima slika koje ne sadrže lica također su primijenjeni isti postupci uklanjanja varijacija osvjetljenja i normalizacije histograma.

Skup svih slika sastoji se od 1994 slike koje sadrže lice i 2039245 slika koje ne sadrže lice. Skup svih slika dijeli se na dva disjunktivna podskupa: skup za učenje i skup za ispitivanje.

4.2 Učenje neuronskih mreža

Cilj je ovog rada razvoj neuronske mreže koja će kao ulaz primiti vrijednosti svjetline prozora od 20x20 piksela, a kao izlaz treba dati odluku sadrži li taj prozor ljudsko lice. Kako su izlazi korištenih aktivacijskih funkcija u intervalu od 0 do 1, izabrano je da će izlaz 1 predstavljati odluku da prozor sadrži lice, a izlaz 0 da ne sadrži lice. Naravno, mreža nikada neće dati izlaz koji je točno 0 ili 1, već neki broj između. Tada se koristi sljedeće pravilo: izlazi iz intervala od 0 do 0,9 znače da slike ne sadrži lice, a izlazi od 0,9 do 1 znače da sadrži. Ovakva neravnomjerna raspodjela izabrana je broj slika za učenje koje sadrže lice i broj slika koje ne sadrže lice također neravnomjeran i korištenje ovakvog omjera pokazalo je bolje rezultate.

Prozor veličine 20x20 piksela znači da neuronska mreža treba imati 400 ulaznih neurona. To je dosta više od broja ulaza koji se inače pojavljuju kod problema za koje se koristi algoritam NEAT. S obzirom na to da algoritam NEAT počinje s minimalno povezanim mrežama i da koristi relativno malu vjerojatnost dodavanja novih veza, za svaki novi skriveni neuron evolucija bi zahtijevala veliki broj iteracija da ga poveže s većinom ulaznih neurona. Stoga se koristi struktura slična konvolucijskoj neuronskoj mreži (Garcia, 2004). Umjesto ulaznih koriste se konvolucijski neuroni: jedan takav neuron spojen je na više pravih ulaznih neurona. Konvolucijski neuron sadrži vektor težina koji ima onoliko elemenata koliko ima ulaznih neurona na koje je spojen. Izlaz konvolucijskog neurona jednak je težinskoj sumi vrijednosti ulaznih. Konvolucijski neuroni predstavljaju se algoritmu NEAT kao ulazni neuroni. Ovakav pristup ima dvije prednosti: drastično se smanjuje broj ulaznih neurona koje algoritam treba povezati sa skrivenim neuronima, a podjela ulaza na manje dijelove također olakšava učenje neuronskih mreža (Rowley, 1999).

Sličan princip koristi se i za algoritam HyperNEAT dodavanjem skrivenog sloja koji nije potpuno povezan s ulaznim slojem, već je svaki skriveni neuron povezan samo s ulaznim neuronima koji sadrže određeni segment slike.

Za učenje neuronskih mreža koriste se tri varijacije algoritma NEAT: izvorni NEAT, RBF-NEAT i HyperNEAT. Za svaki su algoritam ispitana četiri načina grupiranja ulaznih neurona i povezivanja s konvolucijskim neuronima. Način grupiranja neurona

određen je s dva parametra: veličinom grupe i veličinom preklapanja susjednih grupa. Na primjer, za veličinu prozora 5x5 uz preklapanja od 2 piksela prvi će konvolucijski neuron biti spojen na ulazne neurone s koordinatama od (0, 0) do (4, 4), a sljedeći će biti spojen na ulazne neurone s koordinatama od (3, 3) do (6, 6). Oba će konvolucijska neurona biti spojena na ulazne neurone u pojasu širine 2 piksela koji se nalaze na koordinatama od (3, 0) do (4, 5).

Za algoritme NEAT i RBF-NEAT koriste se sljedeća četiri grupiranja:

- veličina prozora 4x4, bez preklapanja (25 konvolucijskih neurona),
- veličina prozora 5x5, bez preklapanja (16 konvolucijskih neurona),
- veličina prozora 5x5, preklapanje 2 piksela (36 konvolucijskih neurona) i
- veličina prozora 10x10, bez preklapanja (4 konvolucijskih neurona).

Za algoritam HyperNEAT potrebno je dodatno izabrati i broj dodatnih skrivenih slojeva (osim konvolucijskog koji je uvijek prisutan). Koriste se sljedeća grupiranja:

- veličina prozora 4x4, bez preklapanja, jedan dodatni skriven sloj,
- veličina prozora 5x5, bez preklapanja, jedan dodatni skriven sloj,
- veličina prozora 5x5, preklapanje 2 piksela, jedan dodatni skriven sloj i
- veličina prozora 5x5, preklapanje 2 piksela, bez dodatnog skrivenog sloja.

Za algoritam NEAT korišteni su sljedeći parametri:

- broj jedinki populacije: 500,
- broj grupa: 25,
- raspon inicijalnih vrijednost težina: $(-5, 5)$
- vjerojatnost mutacije: 50%,
- vjerojatnost križanja: 50%,
- vjerojatnost dodavanja novog neurona: 1%,
- vjerojatnost dodavanje nove veze: 4%,
- vjerojatnost mutacije težina (HyperNEAT): 95%.
- vjerojatnost mutacije težina (NEAT i RBF-NEAT): 47,5%,
- korištene mutacije: uniformna i normalna razdioba,
- vjerojatnost mutacije parametara neurona (NEAT i RBF-NEAT): 47,5% i
- broj iteracija algoritma: 1000.

Kako skup svih slika sadrži iznimno veliki broj slika, učenje mreže nad cijelim skupom bilo bi nepraktično jer bi zahtijevalo previše vremena. S druge strane nije jednostavno izabrati podskup slika jer bi bilo potrebno odrediti slike koje su reprezentativne tj. takve da bi neuronska mreža učena samo nad njima poslije učenja mogla uspješno prepoznavati slike iz cijelog skupa.

Skup za učenje gradi se na način sličan onome koji je opisao Rowley (1999). Početni skup za učenje sastoji se od 500 slučajno odabranih slika lica i 1500 slučajno odabranih slika koje ne sadrže lica. Iako ovaj omjer slika (1:3) ne odgovara stvarnom omjeru koji je približno 1:1000, korištenje manjeg omjera, iako u teoriji nije ispravno, u praksi često daje vrlo dobre rezultate (Lawrence, 1998). Svakih 50 iteracija algoritma provodi se ispitivanje nad skupom svih slika s četiri jedinice iz četiri najbolje grupe (iz svake grupe uzima se samo najbolja jedinica). Nakon validacije u skup za učenje dodaje se novih 10 slika lica i 30 slika koje ne sadrže lica, a biraju se upravo one slike za koje četiri najbolje mreže imaju najveću grešku.

Nakon 1000 iteracija algoritma, skup za učenje sadržavati će 690 slika lica i 2070 slika koje ne sadrže lica. To znači da je za potrebe učenja iskorišteno 34,6% slika lica i 0,1% slika koje ne sadrže lica, te se sve ostale slike koriste u skupu za ispitivanje kako bi se provjerile performanse naučene neuronske mreže.

Dobrota jedinice računa se na temelju iznosa pogrešaka nad slikama iz trenutnog skupa za učenje. Pogreška je apsolutna razlika između očekivanog i dobivenog izlaza (1 za lica, inače 0). Ako je pogreška e manja od 0,5 (i samo onda) dobrota jedinice za tu sliku iznosi $(1 - e^2)$, inače je 0. Ukupna dobrota jedinice jednaka je sumi dobrota za sve slike iz skupa za učenje. Zbog ovakvog računanja dobrote, mreža s velikim brojem malih pogrešaka imati će veću dobrotu od mreže s malim brojem velikih grešaka. S obzirom na to da velike greške obično predstavljaju krivu klasifikaciju mreže, najveću dobrotu imati će mreže koje većinom pravilno klasificiraju slike, što znači da su im izlazi vrlo blizu ciljanim 0 ili 1.

5 Rezultati

Ponašanje osnovne inačice algoritma NEAT prvo je ispitano na nekoliko jednostavnih problema. Iako rješavanje tih problema nema praktičnu vrijednost, korisno je za bolje upoznavanje s radom algoritma i načinom na koji pronalazi optimalnu strukturu neuronske mreže.

Prvi problem, ekskluzivno-ili (engl. *exclusive-or*, XOR) mreža, često se koristi za ispitivanje performansi algoritama za učenje neuronskih mreža (Stanley, 2004). Cilj je razviti mrežu koja će imati izlaz 1 jedino ako je točno jedan ulaz postavljen na 1. Kako algoritam NEAT počinje s minimalnom strukturom, a XOR mreža mora imati barem jedan skriveni neuron, ovaj problem istovremeno ispituje sposobnost algoritma NEAT u otkrivanju potrebe za dodavanjem novih skrivenih neurona i određivanju ispravnih vrijednosti za težine veza.

Mreža ima tri ulazna neurona, od kojih su dva pravi ulazi, a treći (*bias*) je uvijek postavljen na 1. Dobrota jedinke računa se na temelju suma kvadratnih pogrešaka mreža za svaki od četiri moguće kombinacije ulaza. Korištena je veličina populacije od 150 jedinki i 10 grupa, a postupak učenja ponovljen je 50 puta. U prosjeku, algoritmu je potrebno 60 iteracija da pronađe rješenje koje daje ispravne izlaze za sve moguće kombinacije ulaza. Najbolje pronađeno rješenje ima u prosjeku 2,5 skrivenih neurona i 5 dodatnih veza. Iako prosječna mreža ima od jedan do dva skrivena neurona više nego što je potrebno, algoritam je u nekoliko slučajeva ipak uspio pronaći minimalno rješenje.

Drugi problem nad kojim je algoritam korišten je učenje mreža koja se ponaša kao multipleksor sa šest ulaza. Dva ulaza predstavljaju adresne bitove koji određuju koji od ostala četiri ulaza treba preslikati na izlaz multipleksora. Sedmi ulazni neuron je *bias* koji je uvijek postavljen na 1.

Za razliku od problema XOR, ovaj problem ima ukupno 64 moguće kombinacije ulaza te će mreža morati imati veći broj skrivenih neurona i dodatnih veza. Dobrota jedinke računa se na isti način, dok je veličina populacije povećana na 500 jedinki podijeljenih u 25 grupa, a postupak učenja ponovljen je 50 puta. U prosjeku, algoritmu je potrebno oko 930 iteracija da pronađe točno rješenje. Najbolja pronađena jedinka ima u prosjeku 9,7 skrivenih neurona i 32 dodatne veze.

Dok su prethodna dva problema ispitivala sposobnost algoritma u razvoju mreža koje računaju funkcije, točnije Booleove funkcije, zadnji problem ispituje sposobnost algoritma u razvoju mreža za upravljanje. Za takve probleme obično nije poznato najbolje rješenje, niti je moguće provjeriti je li pronađeno rješenje najbolje moguće, već je cilj razviti mrežu sa što je moguće boljim performansama.

Problem koji je korišten je balansiranje štapa (engl. *pole balancing*). Štap određene duljine pričvršćen je na sredinu vozila i može se slobodno rotirati oko te točke. Na vozilo u svakom trenutku djeluje sila konstantnog iznosa, dok smjer sile određuje neuronska mreža. Vozilom se može upravljati pravovremenim mijenjanjem smjera u kojem djeluje sila. Mreža ima četiri ulaza: udaljenost vozila od središnje pozicije, brzina kojom se vozilo kreće, kut štapa u odnosu na ravnotežni položaj (90° u odnosu na horizontalnu os) i kutna brzina kojom se štap rotira. Mreža mora upravljati vozilom tako da kut štapa i udaljenost vozila od početne pozicije nikada ne pređu unaprijed zadane najveće dopuštene vrijednosti.

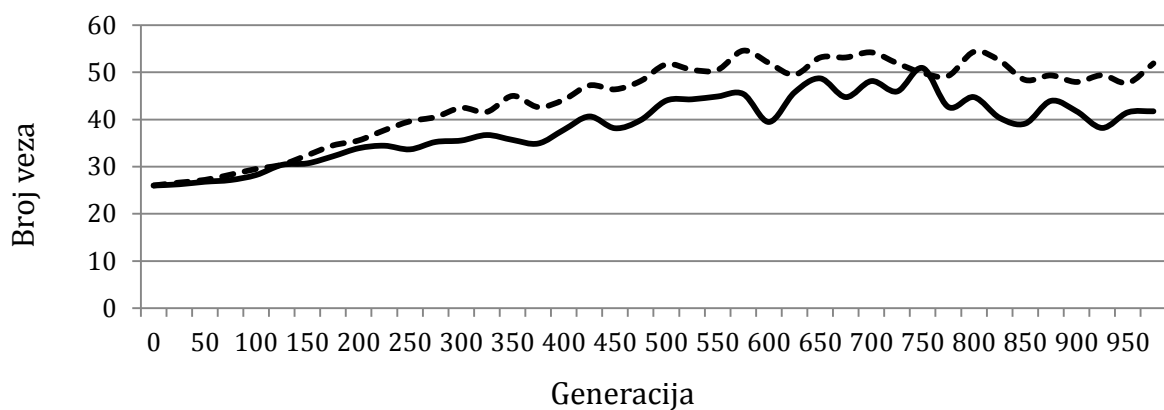
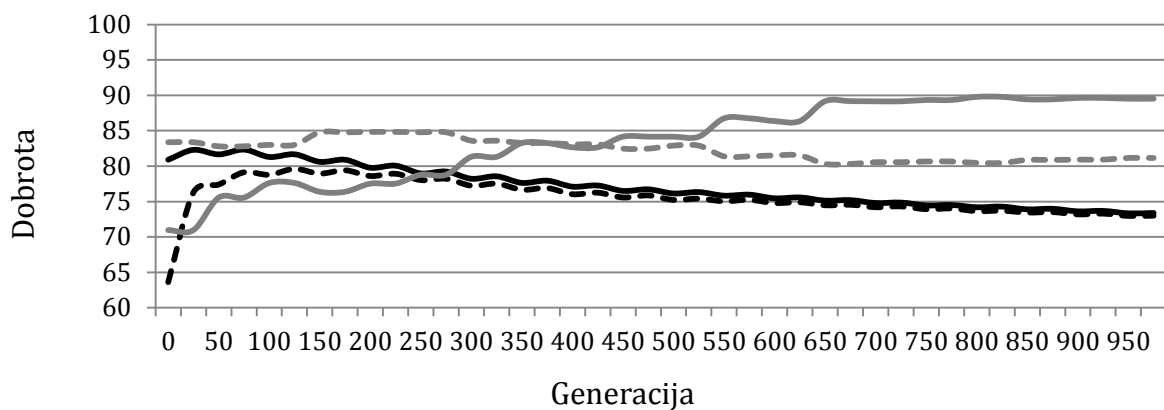
Cilj trećeg problema je razvoj mreže koja će uvijek uspjeti dovesti vozilo u neutralni položaj kada se vozilo nalazi u središnjoj poziciji, a brzina vozila i kutna brzina štapa jednake su nuli. Za ovaj problem poznato je optimalno rješenje. Za problem je korištena veličina populacije od 150 jedinki podijeljenih u 10 grupa, a postupak učenja ponovljen je 300 puta. Svaki put su udaljenost vozila od središnje pozicije i kut štapa postavljeni na slučajno izabrane vrijednosti. Početna brzina vozila i početna kutna brzina štapa uvijek su postavljeni na nulu.

Za treći je problem algoritam NEAT uspio pronaći rješenje vrlo blizu optimalnom za što mu je u prosjeku trebalo 75 iteracija. Dobivena mreža imala je u prosjeku 2,3 skrivena neurona i 4,9 dodatnih veza.

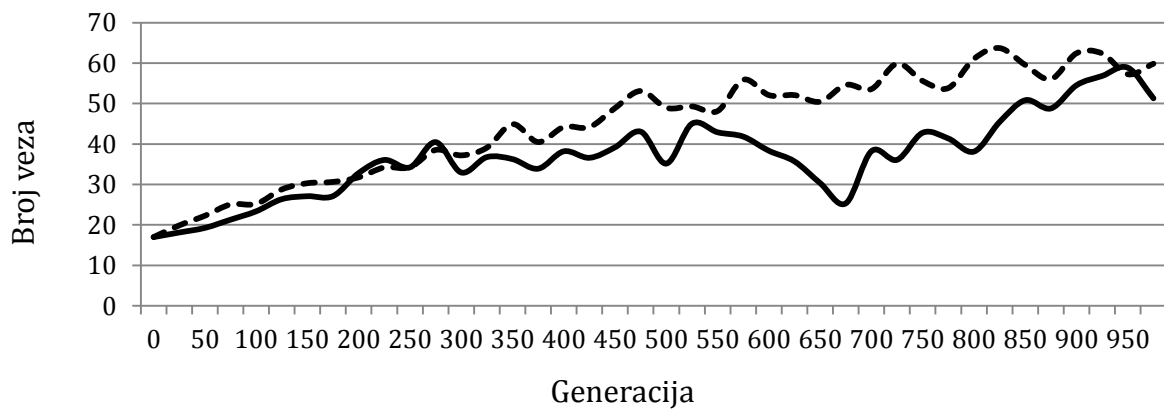
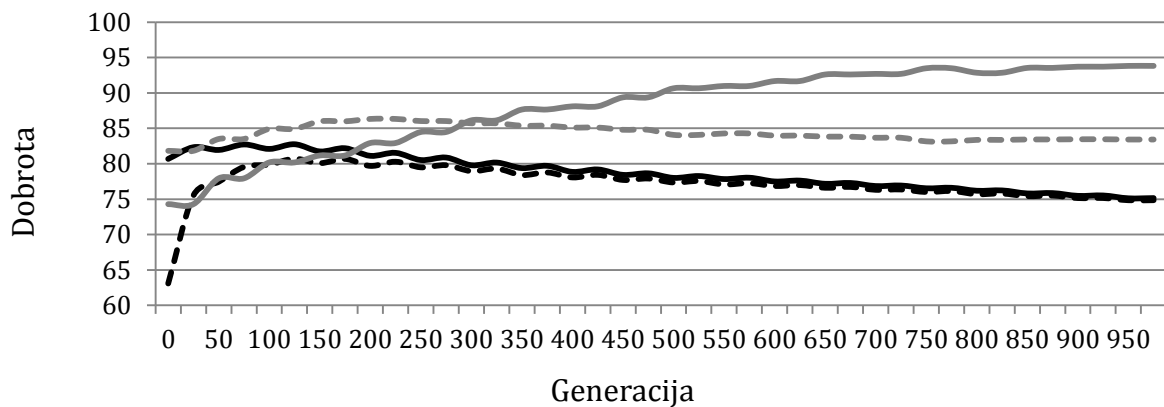
Algoritam NEAT za prethodne je probleme uspješno pronašao rješenja koja su ili optimalna ili blizu teoretski optimalnom rješenju. Osim dobrih rezultata, pronađene su mreže imale, po broju skrivenih neurona i dodatnih veza, relativno jednostavnu strukturu. Problem detekcije lica dosta je teži jer neuronske mreže nije moguće učiti nad svim mogućim kombinacijama ulaza (kao za prva dva problema) niti generirati veliki broj ispitnih slučajeva (kao za treći problem). Dok je za slike koje sadrže lica moguće koristiti dostupne baze slika, broj slika koje ne sadrže lica toliko je velik i raznolik (slike prirode, zgrada, životinja, objekata itd.) da je nemoguće izabrati reprezentativan podskup. Algoritam NEAT mora na temelju dostupnih slika „shvatiti“ koje se karakteristike tipične za ljudsko lice te naučiti neuronsku mrežu da ih raspozna. Skup dostupnih slika koje ne sadrže lica, iako zanemarive veličine u odnosu na skup svih takvih slika, nužan je za ispravan rad algoritma kako bi mogao procijeniti koliko mreža dobro detektira nepostojanje ljudskog lica na slici. Idealan bi detektor lica pronašao sva ljudska lica ignorirajući pritom sve dijelove slike koji ne sadrže lica. Najbolje što se stvarno može postići je razvoj detektora koji će prepoznati veliku većinu lica, uz prihvatljivo mali broj neispravnih detekcija.

Za svaku od 12 konfiguracija (3 različita algoritma, svaki s četiri načina grupiranja ulaznih neurona), postupak učenja potpuno je proveden 30 puta. Tijekom svakog postupka učenje za svaku iteraciju algoritma pratili su se podaci o najboljoj i srednjoj dobroti jedinki, o broju veza koje su dodane u strukturu mreža te o postocima točnih detekcija za slike koje sadrže lica i slike koje ih ne sadrže.

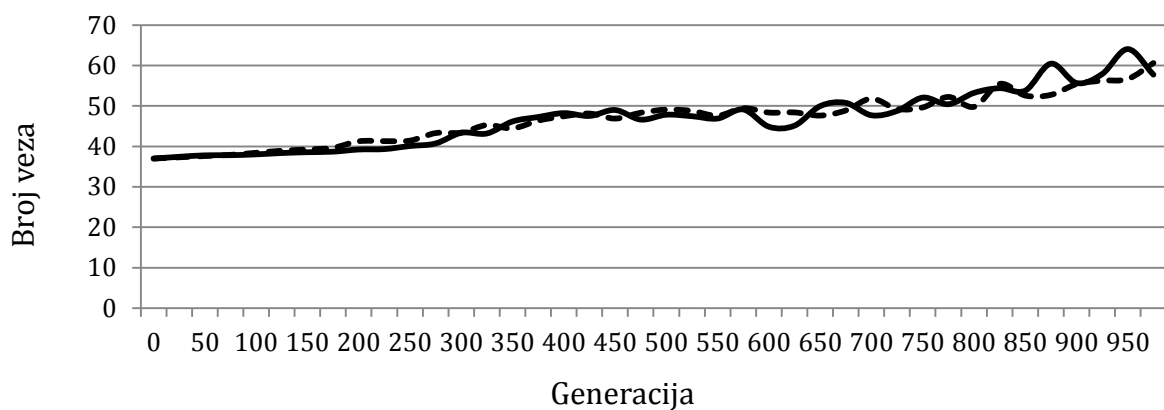
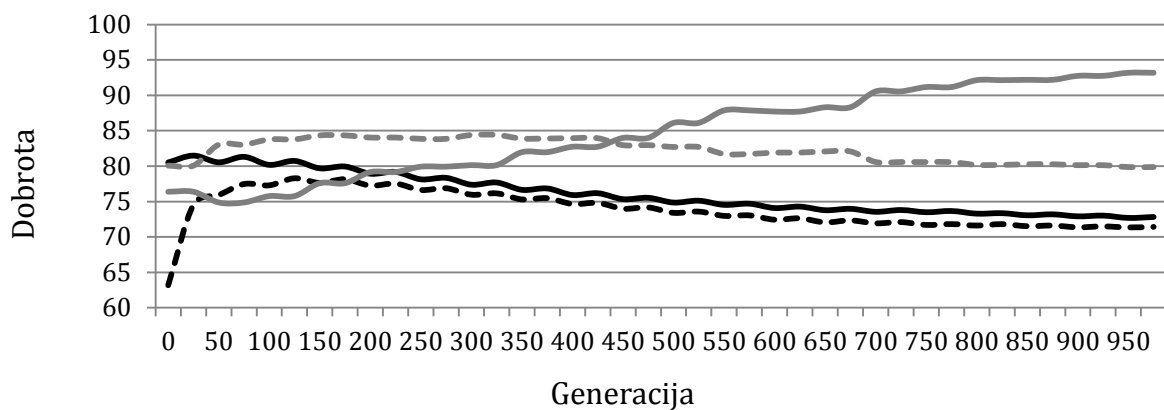
Grafovi od 5.1 do 5.12 prikazuju prosječne vrijednosti po iteracijama za dobrotu najbolje jedinice (gornji graf, puna crna linija), prosječnu dobrotu svih jedinica (gornji graf, isprekidana crna linija), postotak točno klasificiranih slika lica (gornji graf, puna siva linija), postotak točno klasificiranih slika bez lica (gornji graf, isprekidana siva linija), broj veza neuronske mreže najbolje jedinice (donji graf, puna linija) i prosječni broj veza neuronskih mreža svih jedinica (donji graf, isprekidana linija).



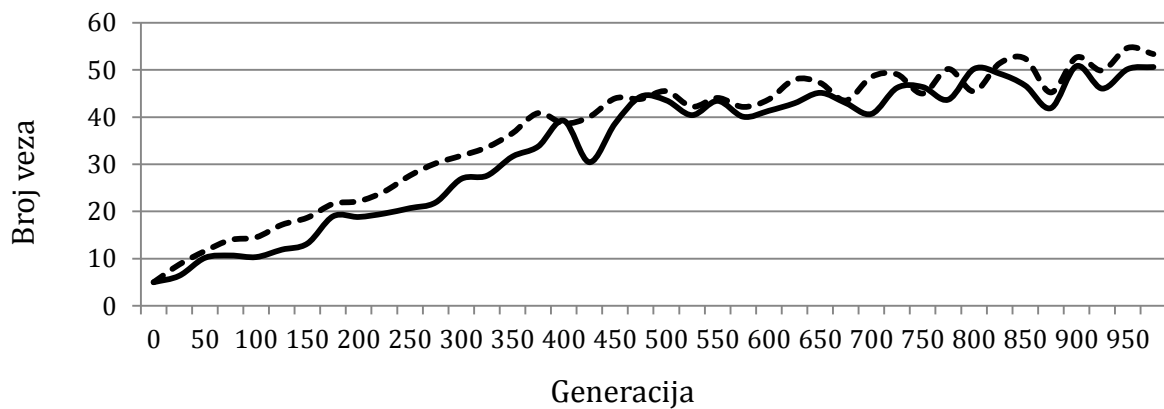
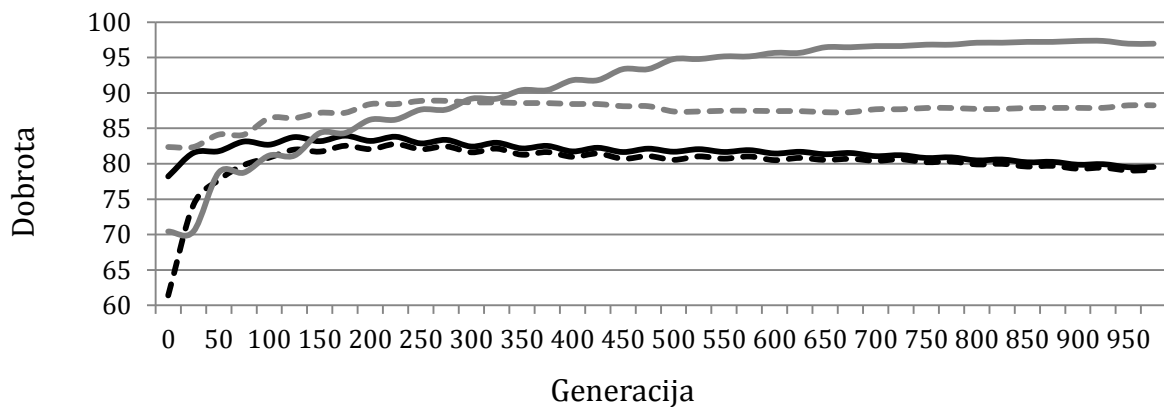
Slika 5.1: Algoritam NEAT, 4x4 bez preklapanja



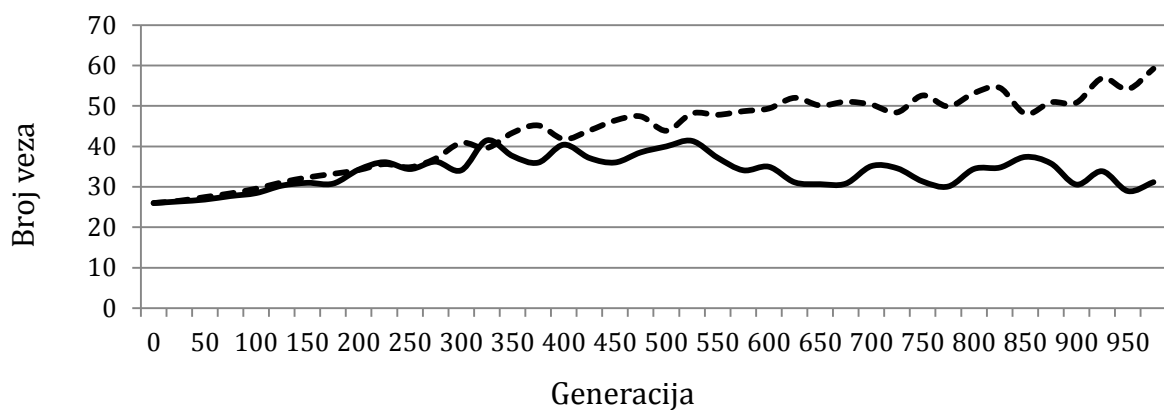
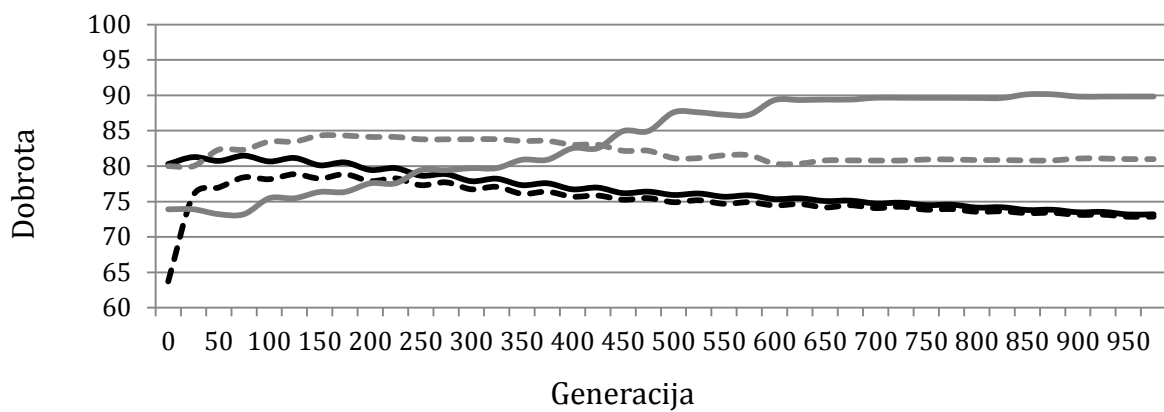
Slika 5.2: Algoritam NEAT, 5x5 bez preklapanja



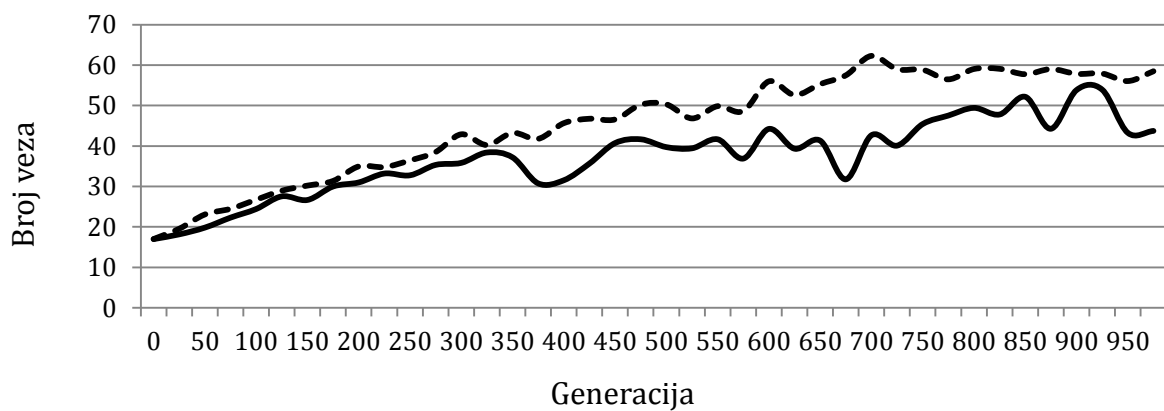
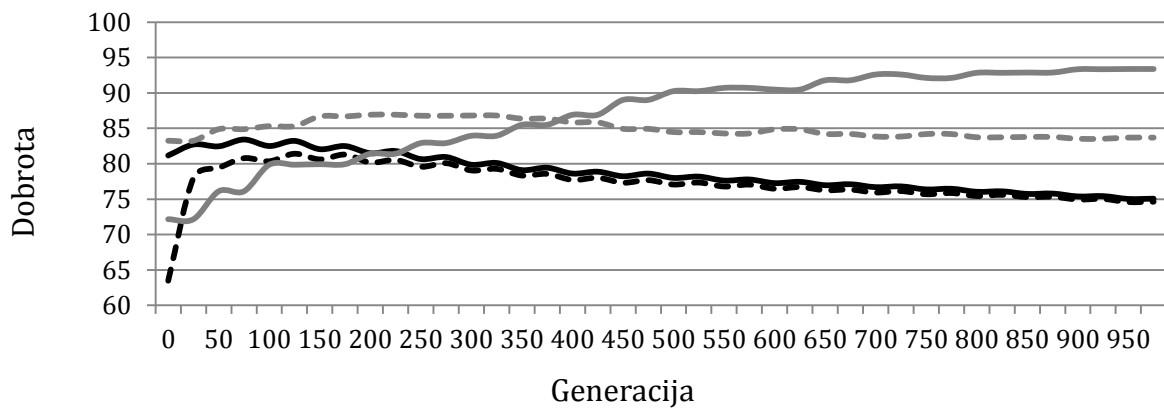
Slika 5.3: Algoritam NEAT, 5x5 s preklapanjem od 2 piksela



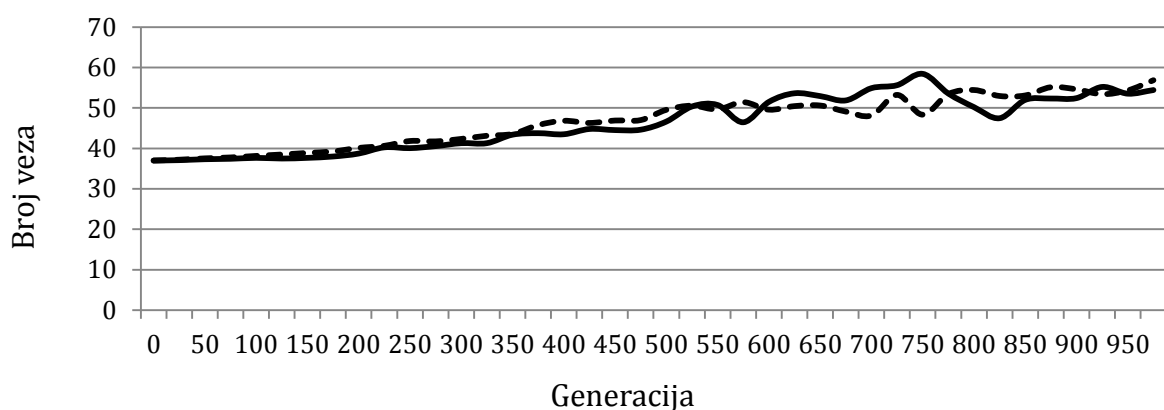
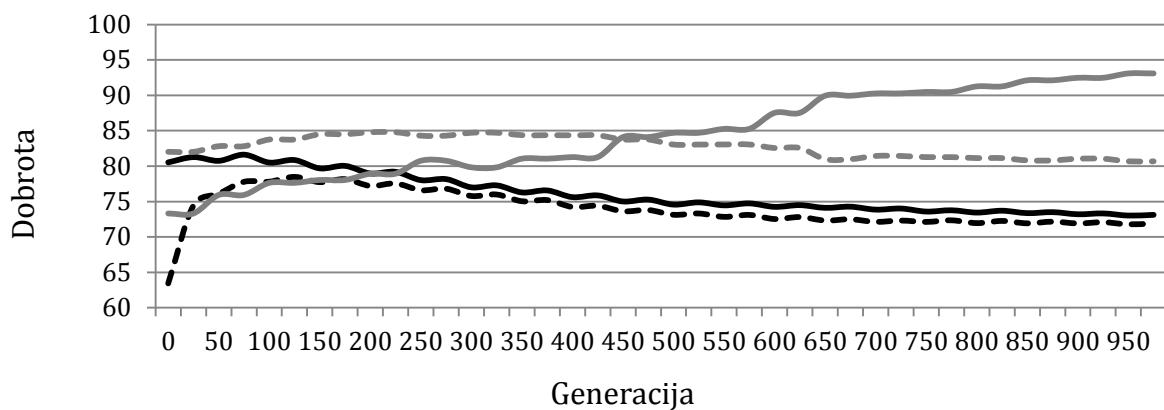
Slika 5.4: Algoritam NEAT, 10x10 bez preklapanja



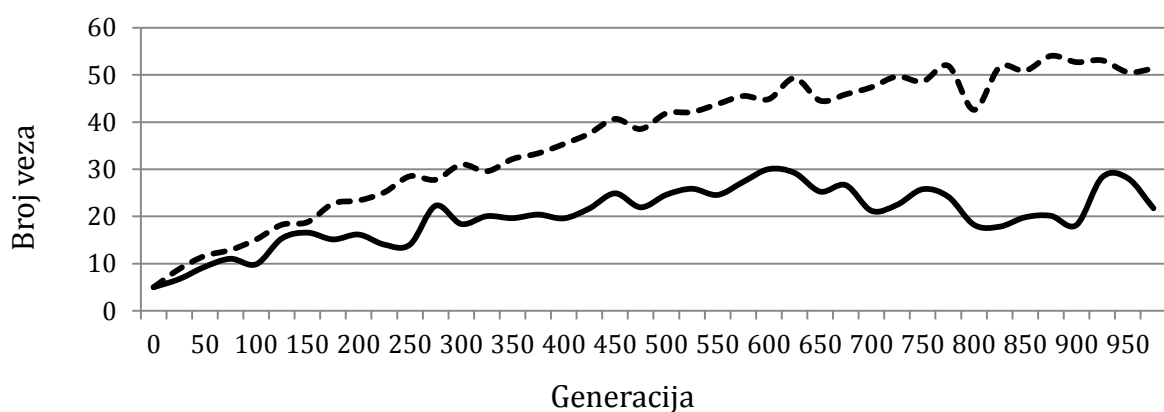
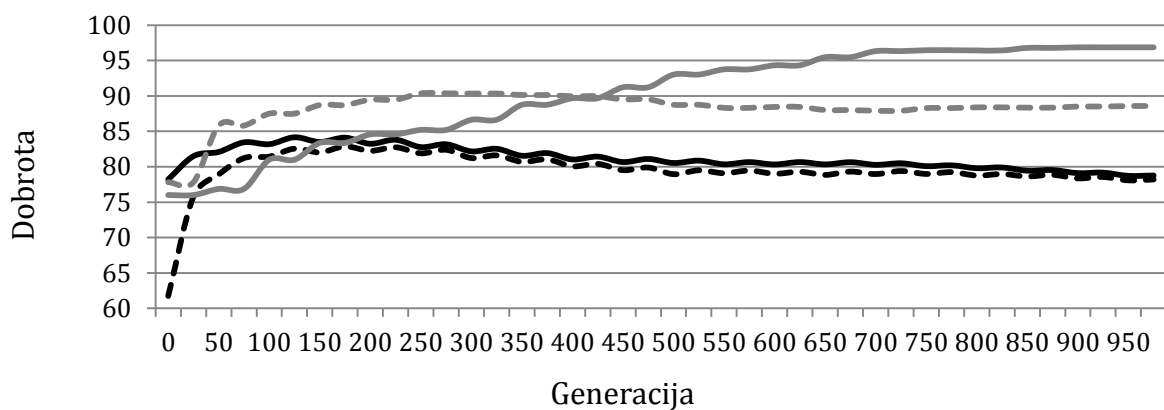
Slika 5.5: Algoritam RBF-NEAT: 4x4 bez preklapanja



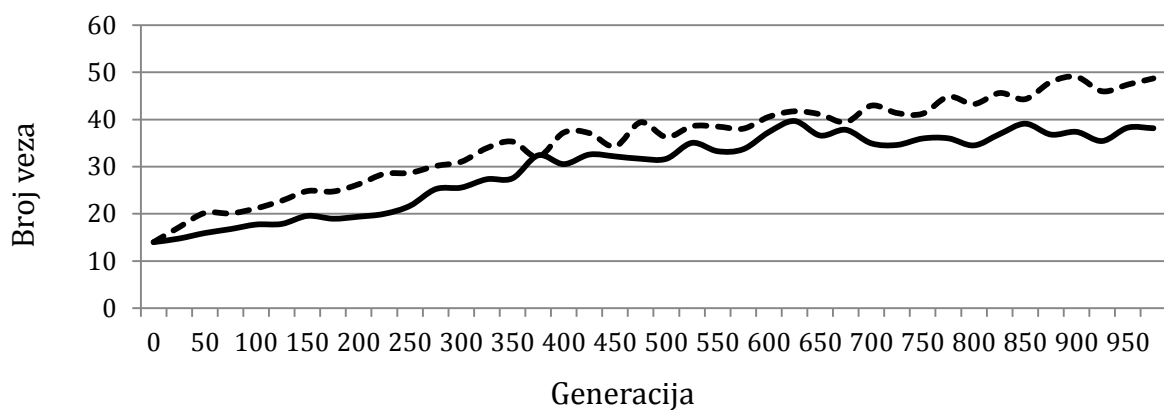
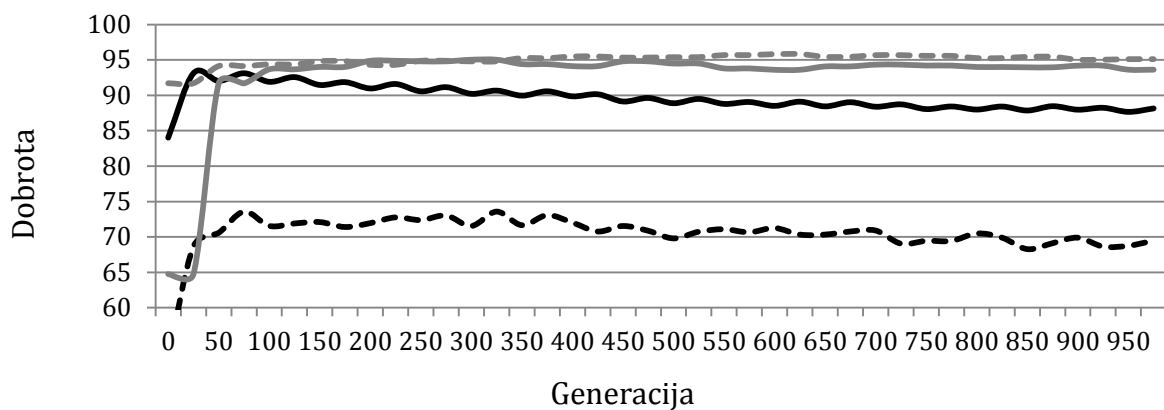
Slika 5.6: Algoritam RBF-NEAT, 5x5 bez preklapanja



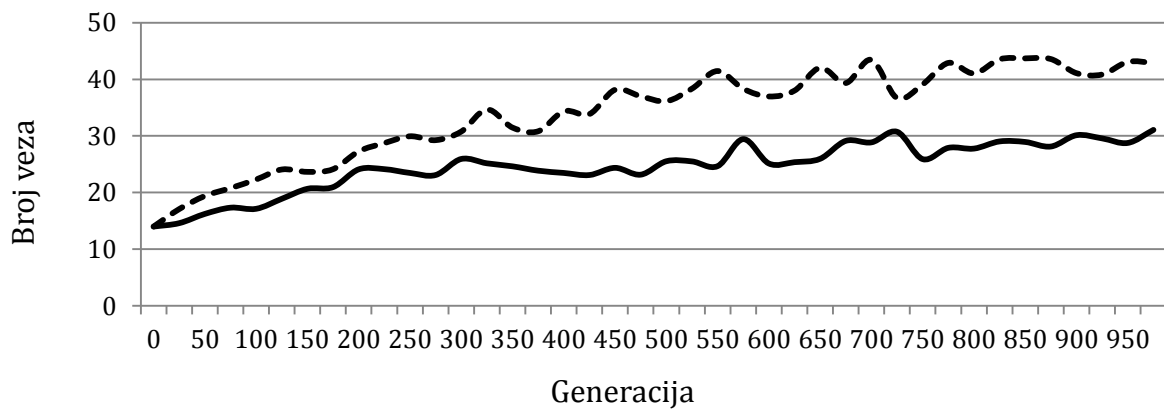
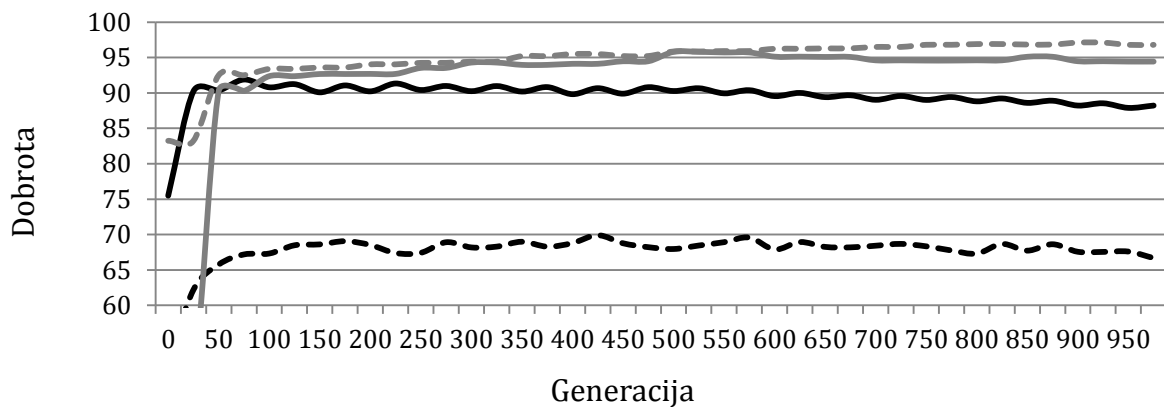
Slika 5.7: Algoritam RBF-NEAT, 5x5 uz preklapanje od 2 piksela



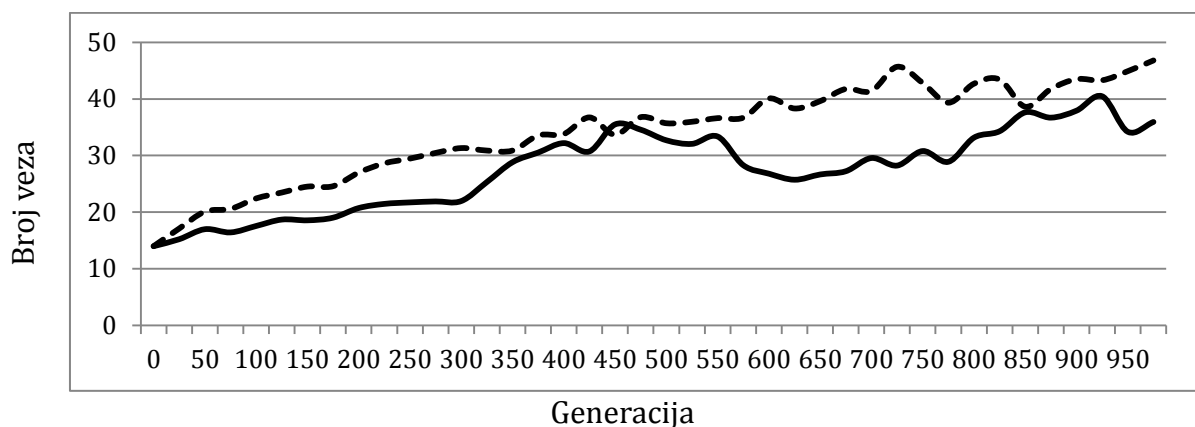
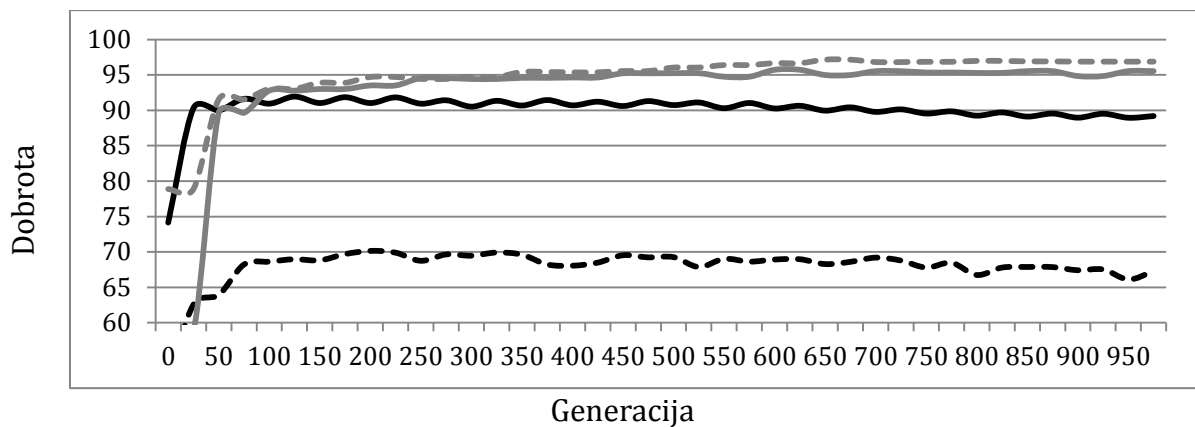
Slika 5.8: Algoritam RBF-NEAT, 10x10 bez preklapanja



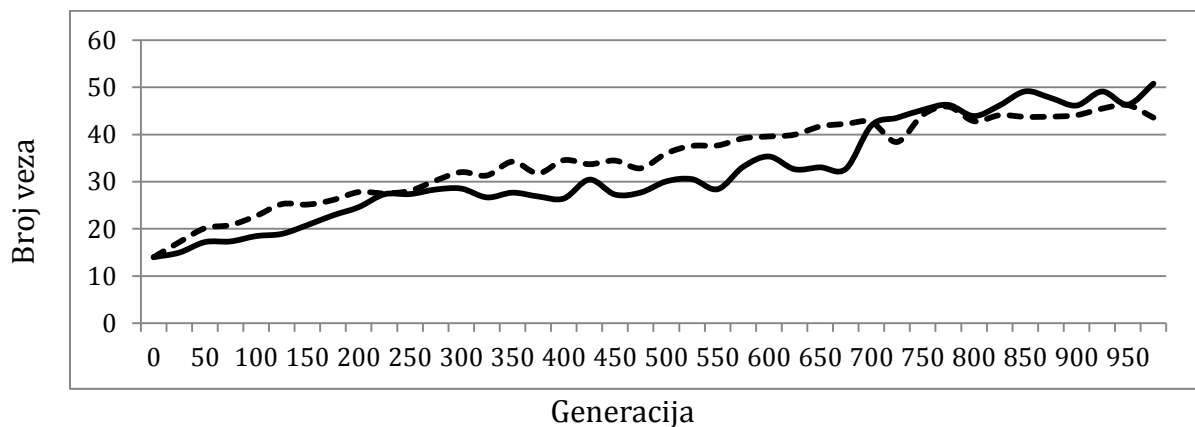
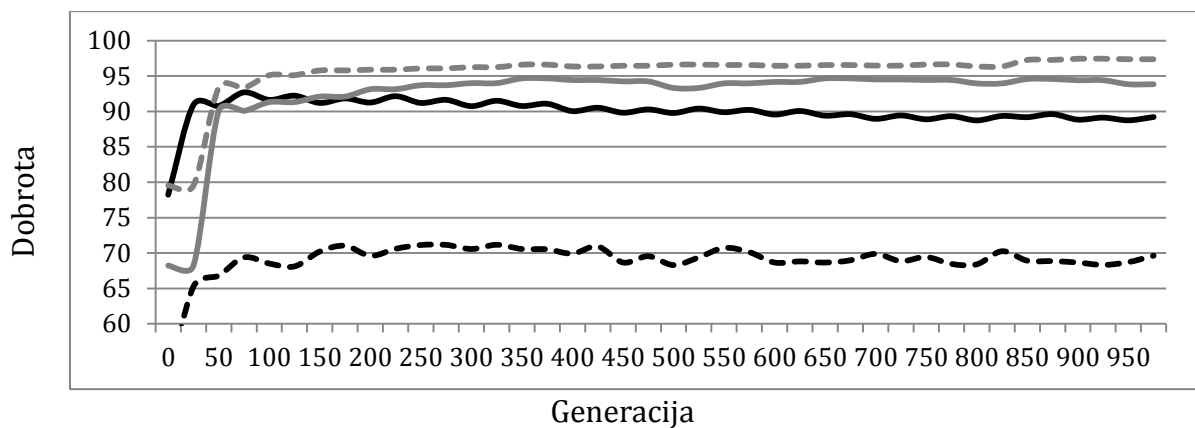
Slika 5.9: Algoritam HyperNEAT, 4x4 bez preklapanja



Slika 5.10: Algoritam HyperNEAT, 5x5 bez preklapanja



Slika 5.11: Algoritam HyperNEAT, 5x5 s preklapanjem od 2 piksela



Slika 5.12: Algoritam HyperNEAT, 5x5 s preklapanjem od 2 piksela bez skrivenog sloja

Iz grafova je vidljivo da algoritmi NEAT i RBF-NEAT imaju konzistentno ponašanje za sve četiri konfiguracije. Nakon kratkog rasta dobrote tijekom početnih 100 generacija dobrota počinje opadati kako se skup za učenje sve više proširuje novim slikama. Iako je za algoritme čiji se rad većinom temelji na slučajnim odlukama teško odrediti točan uzrok ponašanja, moguće ga je s određenom pouzdanošću pretpostaviti. Vjerojatan uzrok ovakvog ponašanja algoritama NEAT i RBF-NEAT jest brza konvergencija prema rješenju koje ima relativno dobre performanse za početni skup za učenje. Kako se taj skup proširuje, algoritmi ne uspijevaju prilagoditi jedinke novim slikama i dobrota se smanjuje. Veći broj iteracija između povećavanja skupa za učenje algoritmima bi mogao dati dovoljno vremena da uspiju razviti jedinke koje bolje klasificiraju nove slike, ali to bi znatno povećalo ionako veliki broj iteracija evolucije.

Algoritmi postižu najbolje rezultate kada se za grupiranje ulaznih neurona koristi veličina prozora 10×10 jer je tada broj ulaznih neurona s kojima algoritmi rade najmanji (samo četiri neurona). Nameće se zaključak da način na koji algoritmi NEAT i RBF-NEAT traže optimalnu strukturu mreže radi bolje što je početna struktura manja. U slučaju početnih struktura s relativno velikim brojem neurona ili veza, mutacije koje dodaju nove neurone i veze imaju dosta manji utjecaj na ponašanje neuronske mreže pa je algoritmu teže odrediti koje su mutacije korisne i treba ih zadržati, a koje može sigurno odbaciti.

Iz grafova koji prikazuju kompleksnost mreža kroz iteracije rada algoritma može se primijetiti da broj dodatnih veza ne raste konstantno već ponekad opada. To se događa kada algoritam uspije razviti jednostavnije jedinke koje postižu bolje rezultate od trenutno najboljih, ali kompleksnih jedinki. Zbog načina na koji algoritam generira sljedeću populaciju, nove jednostavnije jedinke biti će roditelji za većinu jedinki nove generacije što će smanjiti prosječnu kompleksnost cijele populacije. Konzistentnost kojom se događa takav pad kompleksnost navodi na zaključak da je broj veza kojima treba pronaći optimalne težine, čak i kad se koriste konvolucijski neuroni, i dalje prevelik za algoritam NEAT.

Kako algoritam RBF-NEAT postiže skoro iste rezultate kao izvorna inačica, može se zaključiti da korištenje Gaussove aktivacijske funkcije nije naročito korisno za rješavanje problema detekcije lica.

Za razliku od algoritama NEAT i RBF-NEAT, algoritam HyperNEAT ima vrijednosti dobrote i postotka točno klasificiranih slika s licem i bez lica koje su skoro konstantne tijekom svih iteracija evolucije. Dva su razloga toj razlici. Prvo, algoritam HyperNEAT je razvijen upravo za mreže s iznimno velikim brojem veza. Algoritam je uspješno korišten za probleme s nekoliko tisuća ulaza, tako da 400 ulaza koliko ima jedna slika ne predstavlja problem. Drugo, zbog korištenja neizravnog kodiranja broj neurona i veza s kojima algoritam zapravo radi je osjetno manji od broja s kojima su radili algoritmi NEAT i RBF-NEAT. Manja kompleksnost mreža povlači veći utjecaj raznih mutacija koje mijenjaju strukturu mreža tako da algoritam može lakše odrediti koje su mutacije stvarno potrebne. Također, manji broj veza znači da je potrebno manje vremena za pronalazak optimalnih vrijednosti težina.

Tablica 5.1: Performanse naučenih neuronskih mreža

Algoritam, grupiranje		Točno klasificiranih slika lica (postotak)		Točno klasificiranih slika bez lica (postotak)	
		prosjeak	najbolje	prosjeak	najbolje
NEAT	4x4	89,93 (2,45)	89,22	81,06 (2,11)	82,56
	5x5	94,03 (0,89)	94,48	83,12 (1,68)	82,64
	5x5 ^a	92,88 (2,70)	94,68	80,72 (2,81)	83,98
	10x10	97,39 (0,65)	97,49	88,26 (1,51)	90,19
RBF-NEAT	4x4	89,81 (2,34)	88,16	81,14 (2,19)	82,72
	5x5	93,51 (2,01)	92,78	83,72 (2,04)	85,09
	5x5 ^a	93,39 (1,79)	93,88	80,50 (3,42)	83,21
	10x10	96,68 (1,43)	96,59	88,62 (1,51)	89,49
HyperNEAT	4x4	94,09 (2,90)	95,09	95,38 (3,02)	98,85
	5x5	94,75 (2,97)	98,09	96,78 (2,63)	98,14
	5x5 ^a	95,25 (2,34)	97,44	97,31 (1,70)	97,86
	5x5 ^{ab}	94,38 (2,67)	95,44	97,45 (1,47)	99,45

^a s preklapanjem od 2 piksela

^b bez dodatnog skrivenog sloja

Tablica 5.2: Vrijednosti statističke analize neuronskih mreža

Algoritam, grupiranje		Preciznost	Odziv	Specifičnost	Točnost
NEAT	4x4	0,498	89,569	99,988	82,523
	5x5	0,541	94,684	99,994	82,977
	5x5 ^a	0,467	95,436	99,994	80,138
	10x10	0,844	97,894	99,998	88,759
RBF-NEAT	4x4	0,426	92,177	99,990	78,960
	5x5	0,485	94,233	99,993	81,106
	5x5 ^a	0,463	93,129	99,992	80,450
	10x10	1,152	97,041	99,997	91,864
HyperNEAT	4x4	1,253	92,879	99,993	92,841
	5x5	4,766	97,743	99,998	98,090
	5x5 ^a	3,181	94,283	99,994	97,191
	5x5 ^{ab}	23,066	84,804	99,985	99,709

Tablica 5.1 prikazuje prosječne i najbolje vrijednosti postotka točno klasificiranih slika s licem i bez lica najboljih pronađenih jedinki u 30 ponavljanja za svaku kombinaciju korištenog algoritma i načina grupiranja. Za svaku vrijednost u zagradama je prikazana standardna devijacija. Iznos standardne devijacije je relativno visok što znači da postupak učenja nije dovoljno napraviti samo jedanput, već se treba ponoviti više puta kako bi se dobilo što bolje rješenje. Iako takvo ponašanje nije neočekivano za algoritme čiji se rad zasniva na slučajnim odlukama, ipak predstavlja nedostatak zbog količine vremena koja je potrebna za izvođenje dovoljnog broja ponavljanja.

Za rješavanje problema detekcije lica algoritam HyperNEAT je očito bolji izbor od izvorne inačice algoritma NEAT i modificirane inačice RBF-NEAT. Osim manjeg broja neurona i veza koje algoritam treba optimizirati, neizravno kodiranje ima dodatnu prednost što može jednostavno opisati simetrije i ponavljanja koja se pojavljuju među ulaznim vrijednostima. U slučaju detekcije ljudskog lica očita je simetrija lijeve i desne strane lica te isti oblik oba oka i velika sličnost u oblicima pojedinih zuba. Dok bi

izravnim kodiranjem bilo potrebno više puta optimizirati segmente neuronske mreže za prepoznavanje tih obilježja, u slučaju neizravnih kodiranja dovoljno je optimizaciju napraviti jedanput nakon čega se taj segment može ponoviti proizvoljan broj puta.

Za svaku od 12 konfiguracija napravljena je statistička analiza najbolje dobivene mreže. Tablica 5.2 prikazuje vrijednosti dobivene statističkom analizom: preciznost, odziv, specifičnost i točnost. Preciznost (engl. *precision*) je vjerojatnost da slika koju je mreža prepoznala kao lice stvarno sadrži lice. Odziv (engl. *recall*) je vjerojatnost da će mreža ispravno klasificirati sliku koje sadrže lice. Specifičnost (engl. *specificity*) predstavlja postotak slika koje ne sadrže lice koje će mreža ispravno klasificirati. Točnost (engl. *accuracy*) je vjerojatnost da će mreža ispravno klasificirati sliku.

Za svaki algoritam postoji način grupiranja s kojim dobivena mreža ima odziv veći od 97%, što znači da će mreže prepoznati većinu lica. Sve mreže imaju vrlo nisku preciznost, što nije neočekivano jer preciznost ovisi o omjeru ispravnih i krivih pozitivnih klasifikacija, koji je relativno mali jer većina slika ne sadrži lice, pa je upravo ta vrsta krive klasifikacije najčešća. Iz istog razloga sve mreže imaju vrlo visoku specifičnost, što je očekivano jer specifičnost ovisi o omjeru ispravno i krivo klasificiranih slika bez lica, koji je visok zbog velikog broja slika koje ne sadrže lice. Vrijednost po kojoj se mreže najviše razlikuju je točnost, koja ovisi o broju ispravno klasificiranih slika. Najbolji rezultati dobiveni su opet algoritmom HyperNEAT, koji je postigao veću točnost u slučaju mreže bez dodatnog skrivenog sloja, što opet pokazuje da korišteni algoritmi mogu bolje optimizirati manje i jednostavnije mreže.

Tablica 5.3 prikazuje performanse postojećih metoda za detekciju lica na slikama. Za svaku je metodu prikazana baza slika koje je korištena u postupku učenja i ispitivanja performansi mreže, postotak točno klasificiranih slika lica i broj krivo klasificiranih slika bez lica. Da bi se metode opisane u ovom radu mogle usporediti s postojećim metodama potrebno je na temelju postotka točno prepoznatih slika bez lica izračunati ili aproksimirati broj krivo klasificiranih slika bez lica.

Mreža s najboljim rezultatima dobivena je algoritmom HyperNEAT uz grupiranje s veličinom prozora od 5x5 piksela i bez korištenja dodatnog skrivenog sloja ima postotak točno klasificiranih slika lica od 94,38% čime se smješta među bolje mreže po tom kriteriju. S obzirom na prosječnu veličinu jedne slike iz baza MIT i CMU,

postotak točno klasificiranih slika od 99,45% znači da ova mreža krivo klasificira približno 3 prozora koja ne sadrže lica. Za bazu slika MIT-23 koja sadrži 23 slike to bi značilo da bi mreža imala ukupno 69 krivo klasificiranih slika bez lica, 375 krivih klasifikacija za bazu CMU-125 i 390 krivih klasifikacija za bazu CMU-130. Toliki broj krivih klasifikacija dosta je veći od skoro svih već postojećih metoda. S druge strane, treba uzeti u obzir da u ovom radu nisu korišteni načini kombiniranja više mreža za donošenje konačne odluke koji mogu imati veliki utjecaj na smanjivanje broja krivih klasifikacija, kao što je pokazao Rowley (1999).

Tablica 5.3: Rezultati drugih radova za problem detekcije lica

Autor, godina	Baza slika	Točno klasificiranih slika lica (postotak)	Krivo klasificiranih slika bez lica
Lew, 1996	MIT-23	94,1	64
Sung, 1996	MIT-23	79,9	5
Colmenarez, 1997	CMU-130	93,9	8122
Lin, 1997	MIT-23	72,3	6
Osuna, 1997	MIT-23	74,2	20
Rowley, 1999	CMU-130	86,2	23
Feraud, 2000	CMU-130	86,0	8
Gu, 2000	MIT-23	87,1	0
Schneiderman, 2000	CMU-125	94,4	65
		90,2	110
Yang, 2000a	CMU-125	94,8	78
	MIT-20	94,1	3
	MIT-23	84,5	8
Yang, 2000b	CMU-125	92,3	82
	MIT-20	93,6	74
		89,4	3
		91,5	1

Zaključak

U ovom je radu objašnjen način korištenja neuroevolucije za razvoj neuronskih mreža za detekciju ljudskog lica na slikama. Opisana je osnovna inačica algoritma NEAT kao i modifikacije RBF-NEAT i HyperNEAT. Objašnjen je način na koji su prikupljene slike lica i slike koje ne sadrže lica i metode koje su korištene za uklanjanje artefakata i poboljšanja kvalitete slika kako bi se olakšao postupak učenja neuronskih mreža. Opisan je način grupiranja ulaznih neurona i spajanja s konvolucijskim neuronima radi smanjenja broja veza čije težine algoritmi trebaju optimizirati.

Implementirani su svi algoritmi koji su korišteni za obradu slika, učenje neuronskih mreža i ispitivanje njihovih performansi. Za svaki od tri algoritma ispitana su četiri različita načina grupiranja ulaznih neurona. Svako je ispitivanje ponovljeno 30 puta kako bi se mogla napraviti statistička analiza.

Pokazano je da algoritmi mogu bolje optimizirati mreže koje imaju manji broj ulaznih neurona, što se postiže korištenjem većeg grupiranja pri spajanju s konvolucijskim neuronima. Ispitivanje je pokazalo da algoritam RBF-NEAT nije značajno bolji od izvorne inačice algoritma NEAT što znači da problem detekcije lica nije pogodniji za korištenje radijalnih baznih aktivacijskih funkcija. Algoritam HyperNEAT koji koristi neizravno kodiranje pokazao se značajno boljim od druga dva algoritma zbog manjeg broja neurona i težina veza koje treba optimizirati i jer je ciljano razvijen za učenje mreža s vrlo velikim brojem ulaza, što je svakako slučaj za problem detekcije lica.

U usporedbi s postojećim metodama detekcije lica, razvijene mreže imaju vrlo visok postotak točno klasificiranih slika lica, ali i relativno visok broj krivo klasificiranih slika koje ne sadrže lica.

Literatura

- Clune, J., Ofria, C., Pennock, R. T. *The Sensitivity of HyperNEAT to Different Geometric Representations of a Problem*. Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, Montreal, 2009, str. 675–682
- Colmenarez, A., Huang, T. *Face Detection with Information-Based Maximum Discrimination*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, 1997, str. 782–787
- Cutello, V., Nicosia, G. *An Immunological Approach to Combinatorial Optimization Problems*. Proceedings of the 8th Ibero-American Conference on AI: Advances in Artificial Intelligence, Seville, 2002, str. 361–370
- Dubey, D. *Face Detection Using Genetic Algorithm and Neural Network*. International Journal of Science and Advanced Technology. 1, 6 (2011), str. 104–109
- Feraud, R., Bernier, O., Viallet, J. E., Collobert, M. *A Fast and Accurate Face Detector for Indexation of Face Images*. Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, 2000, str. 77–82
- Garcia, C., Delakis, M. *Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence. 26, 11 (2004), str. 1408–1423
- Gu, Q., Li, S. Z. *Combining Feature Optimization into Neural Network Based Face Detection*. Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, 2000, str. 814–817
- Heisele, B., Serre, T., Prentice, S., Poggio, T. *Hierarchical Classification and Feature Reduction for Fast Face Detection with Support Vector Machines*. Pattern Recognition. 36, 9 (2003), str. 2007–2017

- Hjelmås, E., Low B. K. *Face Detection: A Survey*. Computer Vision and Image Understanding. 83, 3 (2001), str. 236–274
- Kennedy, J. *Particle Swarm Optimization*. Proceedings of the IEEE International Conference on Neural Networks, Washington, 1995, str. 1942–1948
- Kohl, N., Miikkulainen, R. *Evolving Neural Networks for Strategic Decision-Making Problems*. Neural Networks. 22, 3 (2009), str. 326–337
- Koza, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Prvo izdanje. Cambridge: MIT Press, 1992
- Lawrence, S., Burns, I., Back, A., Tsoi, A. C., Giles, C. L. *Neural Network Classification and Prior Class Probabilities*. Neural Networks: Tricks of the Trade. Prvo izdanje. Berlin: Springer-Verlag, 1998, str. 299–314
- Lew, M. S., Huijsmans, N. *Information Theory and Face Detection*. Proceedings of the 13th International Conference on Pattern Recognition, Vienna, 1996, str. 601–605
- Lin, S.-H., Kung, S.-Y., Lin, L.-J. *Face Recognition/Detection by Probabilistic Decision-Based Neural Network*. IEEE Transactions on Neural Networks. 8, 1 (1997), str. 114–132
- Meynet, J., Popovici, V., Thiran, J.-P. *Face Detection with Boosted Gaussian Features*. Pattern Recognition. 40, 8 (2007), str. 2283–2291
- Osuna, E., Freund, R., Girosit, F. *Training Support Vector Machines: an Application to Face Detection*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997, str. 130–136
- Phillips, P. J., Wechsler, H., Huang, J., Rauss P. *The FERET Database and Evaluation Procedure for Face Recognition Algorithms*. Image and Vision Computing, 16, 5 (1998), str. 295–306
- Phillips, P. J., Moon, H., Rizvi, S. A., Rauss, P. J. *The FERET Evaluation Methodology for Face Recognition Algorithms*. IEEE Transactions on Pattern Analysis and Machine Intelligence. 22, 10 (2000), str. 1090–1104
- Rowley, H. A. *Neural Network-Based Face Detection*. Doktorski rad. Carnegie Mellon University, 1999

- Rumelhart, D. E., Hinton G. E., Williams, R. J. *Learning representations by back-propagating errors*. Nature. 9, 323 (1986), str. 533–536
- Schneiderman, H., Kanade, T. *A Statistical Model for 3D Object Detection Applied to Faces and Cars*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, 2000, str. 746–751
- Stanley, K. O. *Efficient Evolution of Neural Networks through Complexification*. Doktorski rad. The University of Texas at Austin, 2004
- Stanley, K. O., D'Ambrosio, D., Gauci, J. *Hypercube-Based Indirect Encoding for Evolving Large-Scale Neural Networks*. Artificial Life. 15, 2 (2009), str. 185–212
- Storn, R., Price, K. *Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*. Journal of Global Optimization. 11, 4 (1997), str. 341–359
- Sung, K.-K. *Learning and Example Selection for Object and Pattern Detection*. Doktorski rad. Massachusetts Institute of Technology, 1996
- Sung, K.-K., Poggio, T. *Example-Based Learning for View-Based Human Face Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence. 20, 1 (1998), str. 39–51
- Viola, P., Jones, M. J., *Robust Real-Time Face Detection*. International Journal of Computer Vision. 57, 2 (2004), str. 137–154
- Wiegand, S., Igel, C., Handmann, U. *Evolutionary Optimization of Neural Networks for Face Detection*. Proceedings of the European Symposium on Artificial Neural Networks, Bruges, 2004, str. 139–144
- Yang, M.-H., Roth, D., Ahuja, N. *A SNoW-Based Face Detector*. Advances in Neural Information Processing Systems, 2000a, str. 855–861
- Yang, M.-H., Abuja, N., Kriegman, D. *Face Detection Using Mixtures of Linear Subspaces*. Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, Washington, 2000b, str. 70–76
- Yang, M.-H., Kriegman, D. J., Ahuja, N. *Detecting Faces in Images: A Survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence. 24, 1 (2002), str. 34–58

Sažetak

Evolucija neuronskih mreža za detekciju lica

U ovom radu opisan je način rješavanja problema detekcije lica korištenjem umjetnih neuronskih mreža. Za učenje mreža korišten je neuroevolucijski algoritam NEAT i njegove dvije modifikacije RBF-NEAT i HyperNEAT. Opisi su načini obrade slika kako bi bile pogodnije za postupak učenja: eliminacija varijacija osvjetljenja i normalizacija histograma. Kako bi se smanjio broj veza čije težine algoritmi trebaju optimizirati korišteno je grupiranje više ulaznih neurona i spajanje s konvolucijskim neuronima. Svaki algoritam ispitan je s četiri načina grupiranja neurona i postupak učenja ponovljen je više puta. Pokazano je da algoritmi bolje optimiziraju mreže s manjim brojem grupa ulaznih neurona te da algoritam HyperNEAT postiže najbolje rezultate.

Ključne riječi

detekcija lica, neuronske mreže, konvolucijske neuronske mreže, evolucijski algoritmi, neuroevolucija, NEAT, RBF-NEAT, HyperNEAT,

Summary

Evolving neural networks for face detection

The way of solving face detection problem using neural networks is described in this thesis. Neuroevolutionary algorithm NEAT and its two modifications RBF-NEAT and HyperNEAT were used for training neural networks. Images used for training were processed to improve their quality by eliminating variations in lighting and by normalizing their histograms. In order to reduce the number of connections that the algorithms will need to optimize, multiple input neurons were grouped and connected to convolution neurons. Every algorithm was tested with four different ways of grouping neurons and learning process was repeated several times. Results showed that algorithms were able to better optimize neural networks with a fewer number of input neuron groups and that the algorithm HyperNEAT was able to produce better networks than the other two.

Keywords

face detection, neural networks, convolutional neural networks, evolutionary algorithms, neuroevolution, NEAT, RBF-NEAT, HyperNEAT