

**NEUROPH - SOFTVERSKI FREJMVORK OTVORENOG KODA
ZA RAZVOJ NEURONSKIH MREŽA**
**NEUROPH - AN OPEN SOURCE SOFTWARE FRAMEWORK
FOR NEURAL NETWORK DEVELOPMENT**

Zoran Ševarac

REZIME: Neuroph je softverski frejmvork otvorenog koda za razvoj neuronskih mreža, napisan u programskom jeziku Java. Sadrži biblioteku Java klasa i integrisano grafičko razvojno okruženje za razvoj neuronskih mreža. Zahvaljujući jednostavnoj i jasnoj strukturi (dizajnu softvera), frejmvork se može brzo savladati, i relativno jednostavno modifikovati, što ga čini veoma pogodnim za primenu u istraživanju i edukaciji. Razvijen je na Fakultetu organizacionih nauka u Beogradu, gde se koristi u nastavi i istraživačkim projektima. U ovom radu dati su pregled glavnih karakteristika softvera, primeri primene i poređenje sa drugim sličnim rešenjima.

KLJUČNE REČI: neuronske mreže, softver, frejmvork, Java, otvoreni kod

ABSTRACT: Neuroph is an open source Java neural network framework, developed at the Faculty of Organisational Sciences, University of Belgrade. It provides a set of Java classes and a GUI tool for working with neural networks. It has a clear structure and, is easy to learn and modify, which makes Neuroph very suitable for research and educational applications. It is being used for research and teaching neural networks at Faculty of organisational sciences. This paper gives an overview of the Neuroph project, its features, design, application examples and comparison with other similar solutions

KEY WORDS: neural networks, software, framework, Java, open source

1. UVOD

Softverski frejmvork za neuronske mreže omogućava softversku implementaciju i primenu raznih vrsta neuronskih mreža. Najznačajniji projekti otvorenog koda iz ove oblasti su Joone¹, Encog² i Neuroph³ [Wikipedia, 2011]. Joone je prvi veliki framework iz ove oblasti, i na neki način je kreirao osnovu i postavio standarde za one koji su razvijeni kasnije. Iako se ne razvija aktivno, pomenut je jer je na neki način uspostavio standarde u ovoj oblasti u pogledu skupa funkcionalnosti i opšteg modela frejmvorka ove vrste. Encog je frejmvork visokih performansi sa izuzetnom podrškom za paralelno procesiranje. Nastao je kao prateći softver za popularnu knjigu o programiranju neuronskih mreža u Java-i, i praktično se razvio iz Joone-a. Neuroph je nastao sa ciljem da pruži jednostavan programski interfejs (API) za neuronske mreže u duhu programskog jezika Java, i da pojednostavi upotrebu neuronskih mreža. Nastao je kao akademski projekat u okviru Laboratorije za veštačku inteligenciju na Fakultetu organizacionih nauka u Beogradu. U ovom radu dat je pregled glavnih funkcionalnosti i karakteristika Neuroph frejmvorka, primer primene i poređenje sa Encog frejmvorkom.

2. GLAVNE FUNKCIONALNOSTI NEUROPH FREJMVORKA

Neuroph frejmvork podržava oko 13 vrsta različitih arhitektura neuronskih mreža i njihovih varijacija, i 11 algoritama za učenje neuronskih mreža. Pored toga ima razne dodatne

¹ <http://sourceforge.net/projects/joone/>

² <http://www.heatonresearch.com/encog>

³ <http://neuroph.sourceforge.net/>

funkcionalnosti i alate koji olakšavaju rad sa neuronskim mrežama.

Podržane arhitekture neuronskih mreža: Adaline, Perceptron, Multi Layer Perceptron, Hopfield, Bidirectional Associative Memory, Kohonen, Instar, Outstar, Competitive Network, MaxNet, RBF Network, Neuro Fuzzy Perceptron, and Hebbian Network [Gupta et. al., 2003].

Podržani algoritmi za učenje neuronskih mreža: LMS, Perceptron Learning, Delta Rule, Backpropagation, Resilient Propagation, Dynamic Backpropagation, Hebbian Learning, Anti Hebbian Learning, Oja Learning, Instar Learning, and Outstar Learning [Gupta et. al., 2003].

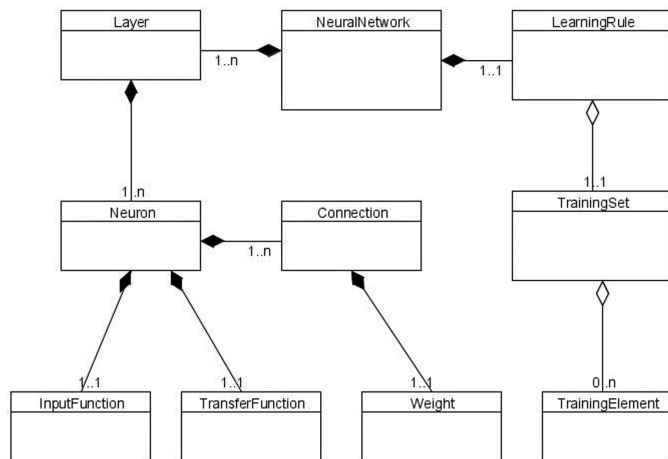
Dodatne funkcionalnosti: normalizacija podataka, algoritmi za inicijalizaciju težinskih koeficijenata, ulazno/izlazni adapteri za fajlove, baze podataka i URL adrese, sistem plugin-ova za proširenja, plugin-ovi za prepoznavanje slika i teksta i jednostavnu podršku za testiranje performansi (*micro-benchmarking*).

Integrisano grafičko razvojno okruženje. *Neuroph Studio* je integrisano razvojno okruženje za neuronske mreže zasnovano na NetBeans platformi. Zasnovano je na sistemu *wizard-a*, i omogućava kreiranje, trening i testiranje neuronskih mreža. Zahvaljujući NetBeans platformi, aplikacija se može u potpunosti integrisati sa NetBeans Java razvojnim okruženjem, što omogućava razvoj i korišćenje neuronskih mreža u okviru iste aplikacije, čime se pojednostavljuje i ubrzava ceo proces.

3. DIZAJN NEUROPH FREJMVORKA

Dizajn Neuroph-a u potpunosti prati osnovne koncepte domenskog modela neuronskih mreža. Osnovne klase

obezbeđuju opštu strukturu, logiku i komponente koje su zajedničke za sve vrste neuronskih mreža. Konkretno vrste neuronskih mreža se kreiraju nasleđivanjem osnovnih klasa i redefinisanjem ili dodavanjem odgovarajućih funkcionalnosti. Na taj način može se kreirati veliki broj neuronskih mreža i iskoristiti veliki deo već postojećih funkcionalnosti (*reusability*). Na slici 1 prikazan je dijagram osnovnih klasa Neuroph frejmvorka.



Slika 1. – Dijagram osnovnih klasa Neuroph frejmvorka

- **Neuron**, osnovni element procesiranja u neuronskoj mreži, sadrži veze ka drugim neuronima (*Connection*), i logiku za izračunavanje (*InputFunction* i *TransferFunction*).
- **Connection**, veza između dva neurona, sadrži reference ka neuronima koje povezuje i odgovarajući težinski koeficijent te veze (*Weight* objekat)
- **Weight**, predstavlja težinski koeficijent koji se dodeljuje vezi između dva neurona. Ovo je na neki način ključna klasa, jer se učenje kod neuronskih mreža zasniva na podešavanju težinskih koeficijenata veza.
- **InputFunction**, predstavlja ulaznu funkciju koja na osnovu ulaza dobijenih iz drugih neurona i težinskih koeficijenata ulaznih veza računa ukupni ulaz za određeni neuron.
- **TransferFunction**, aktivaciona funkcija ili funkcija transfera uzima ukupni ulaz određenog neurona i računa izlaz neurona.
- **Layer**, gradivni element neuronske mreže koji predstavlja sloj/ kolekciju/skup neurona.
- **NeuralNetwork**, opšta osnovna klasa za neuronske mreže. Specifične vrste neuronskih mreža se kreiraju nasleđivanjem ove klase i redefinisanjem odgovarajućih metoda.
- **LearningRule**, osnovna klasa za sve algoritme za učenje. Konkretni algoritmi za učenje se kreiraju nasleđivanjem ove klase i dodavanjem odgovarajuće logike algoritma u izvedenim klasama.
- **TrainingSet**, skup elemenata za trening sadrži skup podataka koje neuronska mreža treba da nauči.
- **TrainingElement**, predstavlja jedan element iz skupa podataka koje neuronska mreža treba da nauči.

4. KRATKA ISTORIJA I RAZVOJ NEUROPH PROJEKTA

Neuroph je nastao kao rezultat studentskog projekta u okviru predmeta Inteligentni sistemi, na Fakultetu organizacionih nauka. Kreiran je kako bi se ispitali i razjasnili detalji u vezi sa implementacijom raznih algoritama za učenje neuronskih mreža. Dalji razvoj nastavljen je kroz diplomski rad, magistarsku tezu i doktorsku disertaciju. Prva javna verzija objavljena je 2008. godine kao projekat otvorenog koda na najvećem portalu za razvoj softvera otvorenog koda SourceForge. Od tada, Neuroph je objavio ukupno 13 verzija, pri čemu su prilikom svakog objavljivanja uvedena određena poboljšanja u vidu dodavanja novih funkcionalnosti ili poboljšavanja postojećih. Detaljne informacije o svim objavljenim verzijama su dostupne na sajtu projekta. Tokom postojanja projekat je imao oko 45000 preuzimanja, u proseku 45 dnevno, i sa prosečnim ciklusom objavljivanja 144 dana (tabela 1). Podaci dati u tabeli 1., dostupni su sa strane za preuzimanje Neuroph-a na SourceForge-u.

Tabela 1. – Statistika Neuroph projekta po verzijama

Verzija	Datum objavljivanja	Životni vek verzije	Broj preuzimanja	Prosečni dnevni broj preuzimanja
2.1	4/7/2009	87	1174	13
2.2	7/3/2009	153	2223	15
2.3	12/3/2009	187	7873	42
2.4	6/8/2010	240	8551	36
2.5	2/3/2011	127	11481	90
2.5.1b	6/10/2011	65	5185	80
2.5.1RC2	8/14/2011	150	9289	62
Prosečno		144	6539	45
Ukupno		1009	45776	

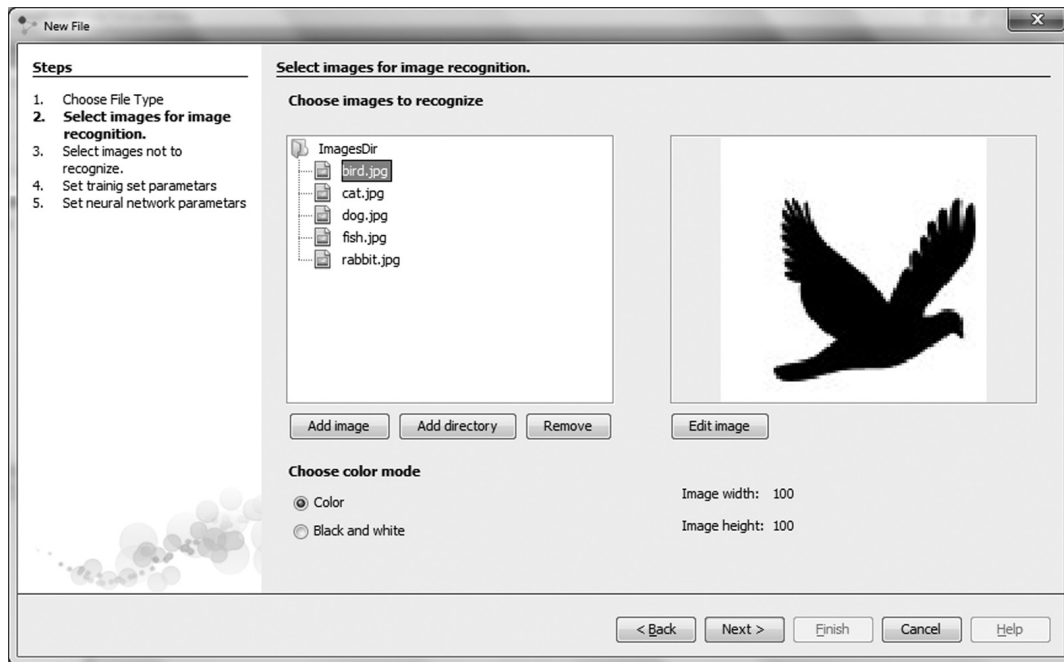
5. PRIMER PRIMENE

U ovoj sekciji dat je primer koji demonstrira primenu specijalizovanog *wizard-a* za kreiranje neuronskih mreža za prepoznavanje slika, u okviru razvojnog okruženja NeurophStudio. U primeru je obrađen problem prepoznavanja crno-belih silueta životinja. Problem prepoznavanja slika (i prepoznavanja uopšte) predstavlja tipičan problem za primenu neuronskih mreža. Skup podataka za trening čini 5 pojedinačnih slika sa siluetama životinja koje treba prepoznati: pas, mačka, zec, riba i ptica (slika 2). Sve slike su dimenzija 100 × 100 piksela (tačaka) i od RGB informacija o boji svakog piksela (tačke) formira se ulazni vektor za neuronsku mrežu.



Slika 2. – Slike za prepoznavanje

U ovom primeru korišćeni su neuronska mreža tipa višeslojni perceptron (*Multi Layer Perceptron*) i *Backpropagation*



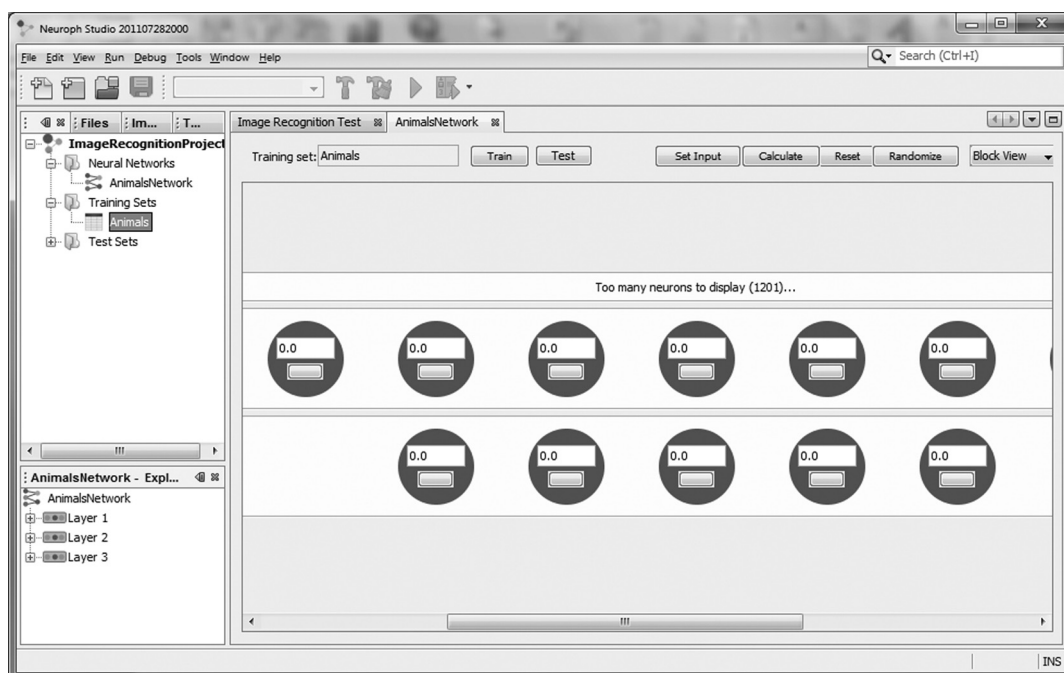
Slika 3. – Wizard za prepoznavanje slika – korak 1: izbor slika za prepoznavanje

algoritam za učenje. Kreiranje neuronske mreže i generisanje skupa podataka za trening na osnovu slika, vrši se pomoću specijalizovanog *wizard*-a aplikacije *Neuroph Studio*, koji se pokreće iz glavnog menija aplikacije *File > New > Neuroph > Image Recognition*. U prvom koraku *wizard*-a biraju se slike za prepoznavanje (slika 3). Dodavanje slika se vrši pomoću dugmeta *Add Image*, a moguće je i dodavanje celog direktorijuma sa slikama pomoću *Add directory*.

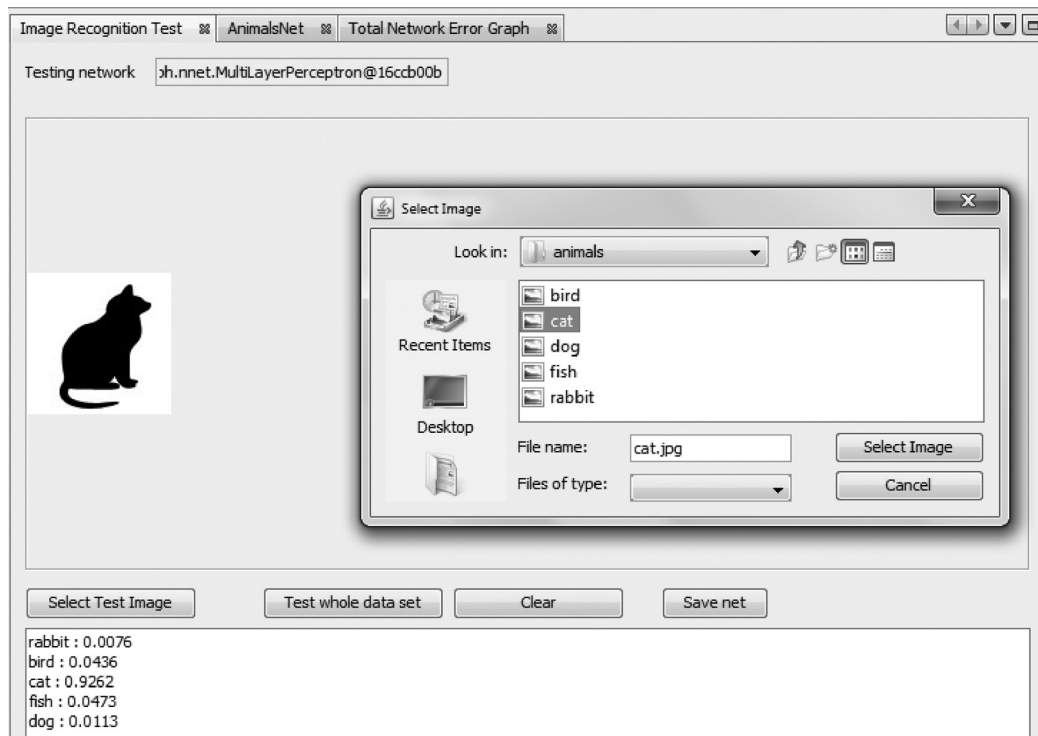
U ostalim koracima *wizard*-a, zadaju se slike koje pomažu da se izbegne pogrešno prepoznavanje, i parametri neuronske mreže (broj skrivenih slojeva, neurona i funkcija

transfera). Nakon prolaska kroz *wizard*, automatski se kreiraju skup podataka za trening od izabranih slika, i neuronska mreža sa zadatim podešavanjima (slika 4). Sama logika za kreiranje podataka za trening na osnovu slika i neuronske mreže za prepoznavanje slika implementirani su u okviru *Neuroph* frejmworka koji obezbeđuje jednostavan API za ove operacije, dok *wizard* u okviru aplikacije *Neuroph Studio* obezbeđuje grafički interfejs za unos slika i podešavanja.

Nakon kreiranja skupa podataka za trening i neuronske mreže, pokreće se trening mreže klikom na dugme *Train*, nakon čega se otvara dijalog za podešavanje parametara



Slika 4. – Kreirana neuronska mreža za prepoznavanje slika



Slika 5. – Prozor za testiranje neuronske mreže za prepoznavanje slika

treninga. Eksperimentalnim putem, određene su vrednosti parametara koje daju dobre rezultate za nekoliko neuronskih mreža sa različitim brojem skrivenih neurona (tabela 2).

Broj skrivenih neurona	Learning Rate	Momentum	Broj iteracija
20	0.2	0.5	214
20	0.2	0.7	423
20	0.5	0.5	72
20	0.7	0.5	beskonačno
30	0.2	0.5	302

Tabela 2. – Rezultati treninga sa različitim podešavanjima neuronske mreže i algoritma za trening

Nakon treninga neuronske mreže, u posebnom prozoru za testiranje vrši se provera prepoznavanja slika (slika 5). Nakon izbora slike, u posebnom polju ispisuju se vrednosti sa izlaza neuronske mreže, tj. rezultati prepoznavanja. Svakom izlaznom neuronu dodeljena je oznaka tj. naziv slike, i neuron sa najvećom izlaznom vrednošću predstavlja prepoznatu sliku. U primeru na slici 5. Prepoznata je slika mačke sa verovatnoćom 0.9, dok je verovatnoća da je na slici zec 0.007.

KORIŠĆENJE ISTRENIRANE NEURONSKE MREŽE ZA PREPOZNAVANJE SLIKA

Neuronska mreža istrenirana za prepoznavanje slika može se sačuvati kao serijalizovani Java objekat, i kasnije koristiti u drugim Java aplikacijama. Na listingu je dat Java programski kod sa objašnjenjima u vidu komentara koji demonstrira korišćenje neuronske mreže za prepoznavanje slika.

```
// učitaj snimljenu neuronsku mrežu
NeuralNetwork nnet = NeuralNetwork.
load("MyImageRecognition.nnet");

// uzmi plugin za prepoznavanje slika
iz neuronske mreže
ImageRecognitionPlugin imageRecognition
= (ImageRecognitionPlugin)nnet.getPlugin(
ImageRecognitionPlugin.class);

try {
// samo prepoznavanje slika se vrši
ovde pomoću metode recognizeImage
imageRecognition.recognizeImage(new
File("someImage.jpg"));

// ispiši rezultate prepoznavanja
System.out.println(output.
toString());
} catch(IOException ioe) {
ioe.printStackTrace();
}
```

6. POREĐENJE SA DRUGIM REŠENJIMA

U ovoj sekciji dato je kratko poređenje dva najpopularnija Java frejmvorka otvorenog koda: Neuroph i Encog. Uporednim poređenjem podržanih arhitektura za ova dva frejmvorka, utvrđeno je da Encog podržava nekoliko arhitektura koje nisu podržane u Neuroph-u i to: *Adaptive Resonance Theory*, *Boltzmann Machine*, *Counterpropagation Neural Network*, *Elman*

Network, Jordan Network, Neuroevolution of Augmenting Topologies (NEAT) and Recurrent Self Organizing Map.

Encog takođe podržava i nekoliko algoritama za učenje neuronskih mreža, koji nisu podržani u Neuroph-u: *Scaled Conjugate Gradient, Manhattan Update Rule Propagation, Levenberg Marquardt, Genetic Algorithm Training.*

Ipak, podržane arhitekture neuronskih mreža i algoritmi za učenje nisu glavna prednost Encog-a, jer se oni relativno lako mogu implementirati pomoću Neuroph-a, zahvaljujući njegovoj fleksibilnosti i proširivosti. Glavna prednost Encog-a, su visoke performanse koje su dobijene zahvaljujući podršci za paralelno procesiranje i korišćenje GPU (*Graphical Processing Unit*) procesora. Takođe treba napomenuti, da je implementacija Encog-a optimizovana za visoke performanse. To u praksi znači korišćenje nizova i matrica, što za posledicu ima manju čitljivost koda, što otežava razumevanje i modifikaciju. S druge strane Neuroph ima jasniji javni programski interfejs (*public API*), veoma intuitivni dizajn i čitljivu implementaciju. U praksi ovo znači da je Encog možda bolje rešenje za primene koje zahtevaju visoke performanse, ali da je Neuroph pogodniji za korisnike koji žele da eksperimentišu sa postojećim ili razvijaju nove vrste neuronskih mreža. Zahvaljujući jasnom jednostavnom dizajnu, i čitljivom kodu Neuroph je pogodniji za edukativnu primenu. Prvo testiranje performansi Java frejmworka za neuronske mreže koje javno objavljeno⁴, pokazalo je da je Encog 2.4 bio 10 puta brži od Neuroph-a 2.4. Sa sledećom verzijom 2.5, Neuroph je povećao performanse 2 puta, ali je i dalje bio značajno sporiji od Encog-a. Sa verzijom 2.5, Neuroph je uveo i implementaciju zasnovanu na matricama, koja se po performansama približila Encog-u na jednom procesoru, i uveo je ideju o implementaciji posebnog sloja u arhitekturi koji je optimizovan za visoke performanse⁵. Kao rezultat toga Neuroph i Encog projekat su objavili saradnju⁶, i omogućili integraciju brzog procesiranja iz Encog-a u Neuroph za neke vrste neuronskih mreža. Glavna ideja ove integracije je zadržati intuitivan programski interfejs Neuroph-a, i koristiti visoke performanse Encog-a. Oba projekta su nastavila nezavisan razvoj, ali sa idejom da razmenjuju tehnologije i održavaju interoperabilnost u daljem razvoju. Još jedna stvar koju treba imati u vidu prilikom poređenja je grafičko razvojno okruženje. Encog ima osnovno razvojno okruženje sa grafičkim interfejsom pod nazivom EncogWorkbench, dok Neuroph ima razvojno okruženje zasnovano na industrijskim standardima koje definiše NetBeans platforma, otvoreno za proširenja i integraciju sa drugim alatima.

7. ZAKLJUČAK

U ovom radu opisan je Neuroph, Java frejmwork otvorenog koda za razvoj neuronskih mreža. Zahvaljujući funkcionalnostima koje ima, fleksibilnosti, jednostavnosti korišćenja, i licenci otvorenog koda veoma je pogodan za istraživačku i edukativnu primenu. Neuroph ima stabilnu petogodišnju istoriju razvoja, i veliki broj korisnika i preuzimanja za projekat ovog tipa.

⁴ <http://www.codeproject.com/KB/recipes/benchmark-neuroph-encog.aspx>

⁵ http://neuroph.sourceforge.net/improving_performance.html

⁶ <http://neuroph.sourceforge.net/NeurophEncogCollaboration.html>

Neuroph je takođe poznat po saradnji sa drugim projektima otvorenog koda, posebno NetBeans i Encog projektima. Iako su performanse Neuroph-a iza drugih sličnih rešenja, njegove glavne prednosti su fleksibilnost (jednostavan za izmene i proširenja), jednostavnost korišćenja (jednostavan za učenje i korišćenje), što ga čini pogodnim za edukativnu i istraživačku primenu. U sekciji literatura navedeno je 6 istraživačkih projekata koji koriste Neuroph, poznatih autoru, a izvesno je i da ih ima više.

Dalji razvoj Neuroph-a ići će u sledećim pravcima:

1. Razvoj novih arhitektura i algoritama za učenje neuronskih mreža
2. Pобољшanje performansi
3. Razvoj dodatnih funkcionalnosti za automatizaciju procedure treninga i testiranja neuronskih mreža u okviru razvojnog okruženja Neuroph Studio.

Imajući u vidu dosadašnji razvoj Neuroph projekta, prihvaćenost od strane korisnika i dalju perspektivu razvoja u budućnosti, Neuroph definitivno zaslužuje da bude razmotren kao opcija od strane istraživača i nastavnika koji predaju neuronske mreže u okviru svojih predmeta. Neuroph softver sa kompletnom dokumentacijom i izvornim kodom je dostupan za preuzimanje sa Neuroph sajta⁷.

LITERATURA:

- [1] Andre Christoffer Andersen, Autonomous Neural Development and Pruning, Department of Bioengineering, University of California San Diego, La Jolla, CA 92122, 2010. Available electronically via: <http://itspiren.no/2010/12/autonomous-neural-development-and-pruning/>
- [2] C. Abhishek, V. P. Kumar, H. Vitta, P. R. Srivastava, Test Effort Estimation Using Neural Network, Journal of Software Engineering and Applications, 3:331-340, 2010. doi:10.4236/jsea.2010.34038
- [3] Alexandre Maciel and Edson Carvalho, FIVE - Framework for an Integrated Voice Environment, 17th International Conference on Systems, Signals and Image Processing, 2010.
- [4] Citation. (2011, December 21). In Wikipedia, The Free Encyclopedia. Retrieved 12.21.2011., from <http://en.wikipedia.org/wiki/JOONE>
- [5] Fisher, R.A., The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188, 1936.
- [6] Gupta, M., Jin, L., Homa, N., Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory, Wiley-IEEE Press, 2003.
- [7] Jia Zhu, Gabriel Fung and Liwei Wang, Efficient name disambiguation in digital libraries, Proceedings of the 12th international conference on Web-age information management, 430-441, Springer-Verlag, Berlin, ISBN: 978-3-642-23534-4, 2011.
- [8] Jan Boltzen, Jorge Sanchez, Peter Olsson, Marcus Johansson, Backgammon AI Player, 2010 [ONLINE]: <http://trac.assembla.com/BackgammonAI/export/20/trunc/docs/proposal.pdf>



Zoran Ševarec

⁷ <http://neuroph.sourceforge.net/download.html>