

ПРИМЕНА *NEUROPH* РАДНОГ ОКВИРА У УЧЕЊУ ВЕШТАЧКИХ НЕУРОНСКИХ МРЕЖА

Милош Пејановић¹ Вукман Кораћ² Перица Штрбац³ Стефан Пејановић⁴

Резиме: У оквиру предмета Меко рачунарство (*Soft Computing*) и Интелигентни системи и технологије на АТУСС - Одсек ВИШЕР, у деловима предмета који се односе на вештачке неуронске мреже предвиђено је коришћење *Neuroph* радног оквира. Овај радни оквир служи за развој вештачких неуронских мрежа. Идеја је да студенти на прагматичан начин науче концепт функционисања вештачких неуронских мрежа коришћењем *Neuroph* радног оквира. На поменутих предметима студенти уче како креирати, обучити и тестирати вештачку неуронску мрежу. Циљ овог рада је да се прикаже један приступ учења студената у развоју и примени једне од метода меког рачунарства у оквиру вештачке интелигенције, без примене класичних алгоритама коришћењем радног оквира *Neuroph*, уз примену графичког интерфејса *NeurophStudio* и делом знања из програмског језика Java. У раду је дат илустративни пример креирања, учења и обучавања вештачке неуронске мреже за предикцију резултата спортског такмичења који служи као практичан пример студентима за разумевање концепта вештачких неуронских мрежа. У датом примеру приказани су: модел креиране вештачке неуронске мреже, скривени слојеви, резултати обучавања и тестирања.

Кључне речи: Neuroph, вештачке неуронске мреже, меко рачунарство, Java, обучавање, тестирање.

USAGE OF NEUROPH FRAMEWORK IN LEARNING OF ARTIFICIAL NEURAL NETWORKS

Abstract: Within the course Soft Computing and Intelligent Systems and Technologies at ATUSS – Department of VIŠER, in the parts of the course related to artificial neural networks, the use of the Neuroph framework is envisaged. This framework serves for the development of artificial neural networks. The idea is for students to learn in a pragmatic way the concept of the functioning of artificial neural networks using the Neuroph framework. In the mentioned subjects, students learn how to create, train and test an artificial neural network. The aim of this paper is to present an approach to student learning in the application of one of the methods of soft computing within artificial intelligence, without the application of classical algorithms using the Neuroph framework, using the graphical interface *NeurophStudio* and part of knowledge of Java programming language. The paper gives an illustrative example of creating an artificial neural network for predicting the results of sports competition, which serves as a practical example for students to understand the concept of artificial neural networks. The given example shows a model of an artificial neural network created, hidden layers, training, and testing results.

Keywords: Neuroph, artificial neural networks, soft computing, Java, training, testing.

1. УВОД

Предмет Меко рачунарство (*soft computing*), који се слуша у четвртном семестру основних струковних студија Академије техничко-уметничких струковних студија на одсеку Висока школа електротехнике и рачунарства у Београду, обухвата изучавање и примену концепата меког рачунарства и појединих алгоритама машинског учења при пројектовању софтверских решења, која се делимично заснивају на тим концептима. Меко рачунарство је нова мултидисциплинарна област вештачке интелигенције која има циљ да развија интелигентне машине које ће обезбедити решавање проблема из реалног света који нису моделовани или су превише комплексни за математички модел.

¹ Предавач, АТУСС – Одсек Висока школа електротехнике и рачунарства, Војводе Степе 283, Београд, milos.pejanovic@viser.edu.rs:

² Асистент, АТУСС – Одсек Висока школа електротехнике и рачунарства, Војводе Степе 283, Београд, vkorac@viser.edu.rs:

³ Професор, АТУСС – Одсек Висока школа електротехнике и рачунарства, Војводе Степе 283, Београд, pericas@viser.edu.rs:

⁴ Дипл.инж, NewCo. – Научно-технолошки парк, Вељка Дугошевића 54, Београд.

За разлику од конвенционалног рачунарства (*hard computing*), меко рачунарство је толерантно на непрецизности, неизвесности, делимичне истине и приближне вредности [6]. Настава је оријентисана ка практичном приступу где се користе радни оквири (*frameworks*), делови кода или комплетни кодовиза дате наставне теме. Теме су: основе математичке логике, основе вештачке интелигенције и меког рачунарства, концепт вештачких неуронских мрежа, концепт фази скупова и фази система (*Fuzzy logika*) и концепт генетских алгоритама. У оквиру концепта вештачких неуронских мрежа предвиђено је коришћење *Neuroph* радног оквира.

Главни циљ је омогућити да студенти у оквиру једне до две вежбе, користећи *Neuroph* радни оквир примене концепт вештачких неуронских мрежа, решавају примере из реалног света који нису моделовани или су превише комплексни за математички модел.

На основу претходног, одабрано је да се креира *Neuroph* пројекат за предикцију резултата спортског такмичења, који обухвата: креирање, обучавање и тестирање вештачке неуронске мреже. Након више итерација студенти дефинишу оптимално решење.

2. ПРОЈЕКАТ „*PredikcijaRezultataTakmicenja*“

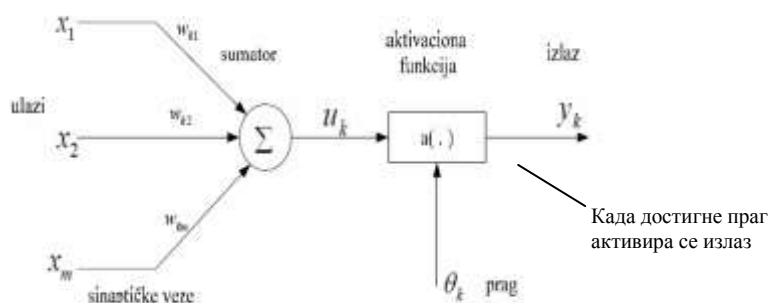
У овом примеру је креирана и обучена вештачка неуронска мрежа која предвиђа резултат између два такмичарска тима, односно исход у корист домаћег тима, гостујућег тима или нерешен резултат. За обучавајући скуп података узети су конкретни резултати такмичења Премиер лиге Енглеске из сезоне 2011/12. Због великог броја мечева, насумично је узето 106 резултата. Сваки резултат има 8 улазних и 3 излазна атрибута.

Улазни атрибути су: оцена голмана домаћег тима, оцена одбрамбених играча домаћег тима, оцена везних играча домаћег тима, оцена нападача домаћег тима, оцена голмана гостујућег тима, оцена одбрамбених играча гостујућег тима, оцена везних играча гостујућег тима и оцена нападача гостујућег тима. Излазни атрибути су: победа домаћег тима, нерешено и победа гостујућег тима.

2.1. Вештачке неуронске мреже

Вештачке неуронске мреже симулирају начин рада људског мозга при обављању задатка или неке функције. У суштини представљају масовно паралелизован дистрибуирани процесор са природном способношћу меморисања искуственог знања и обезбеђивања његовог коришћења [3]. Знање се захвата кроз процес обучавања, а меморисање се обезбеђује кроз тежине (јачине) синаптичких веза између неурона. Обучавање вештачких неуронских мрежа се обавља кроз алгоритам обучавања на тај начин да се на систематичан начин мењају синаптичке тежине у циљу достизања жељених перформанси.

Модел вештачког неурона који је приказан на слици 1, чине три основна елемента: скуп синаптичких тежина $\{w_{ij}\}$, суматор (формира тежинску суму улаза) и активациона функција (лимитира амплитуду излазног сигнала вештачког неурона). Вештачке неуронске мреже садрже слојеве неурона који су међусобно повезани синаптичким везама. Могу бити једнослојне или вишеслојне. Засноване су на подацима и дају имплицитни модел окружења уз истовремено обављање жељеног процесирања. Знање о окружењу у вештачким неуронским мрежама је кодовано кроз конкретне вредности слободних параметара добијених кроз обучавање. Вештачке неуронске мреже саме подешавају параметре док се не добије одговарајући излаз [5].



Слика 1– Модел вештачког неурона

Обучавање вештачких неуронских мрежа је процес адаптирања слободних параметара, који се обавља кроз стимулацију окружења. Процес обучавања је класификован у три категорије: обучавање са надзором или учитељем (*supervised learning*), обучавање са подстицањем (*reinforcement learning*) и самообучавање или без учитеља (*unsupervised learning*). Важно правилно је како одредити коефицијент обучавања. Велике вредности за η (коефицијент обучавања) могу да убрзају конвергенцију, али и да доведу до премашаја циља, док исувише мале вредности имају супротан ефекат. Пракса показује да се η може кретати зависно од конкретне проблема у опсегу од 0,001 до 10.

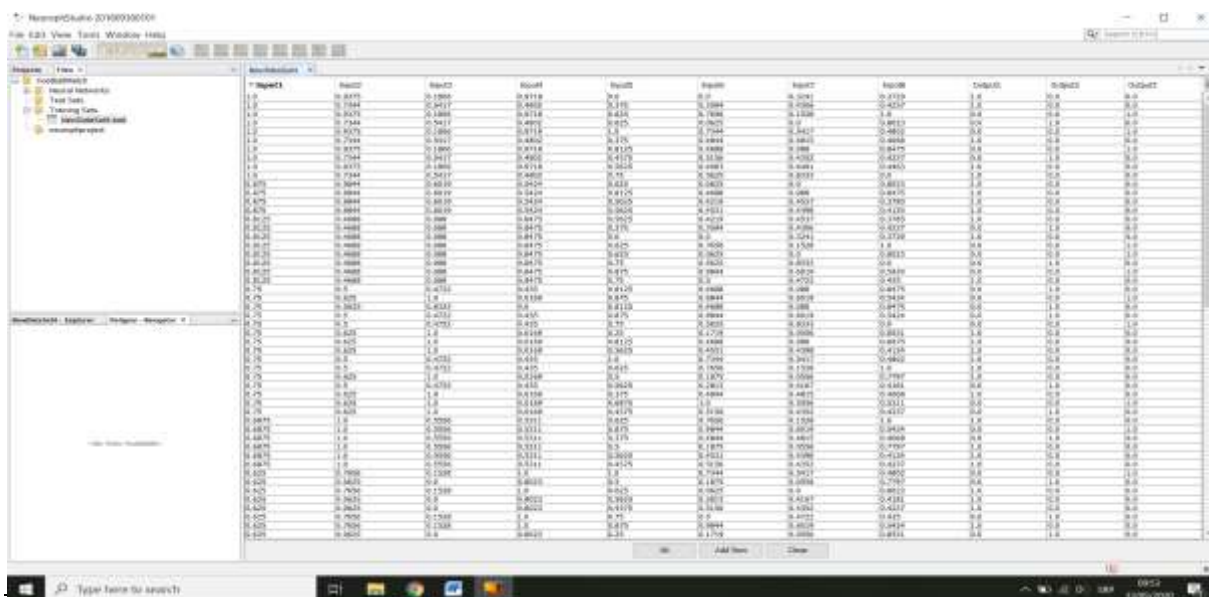
Вештачке неуронске мреже садрже улазни, излазни и опционо скривени слој неурона. Кључно питање је како одредити број неурона у скривеном слоју у креирању и примени вештачких неуронских мрежа. Егзактну анализу је тешко реализовати ради стохастичке природе већине алгоритама обучавања [3]. Практичне препоруке се односе на принцип да се проба са почетним бројем неурона у скривеном слоју који је знатно мањи од димензије улазног слоја, а ако је обучавање задовољавајуће покушати са даљим смањивањем. У супротном, инкрементално повећавати њихов број.

2.2. Neuroph радни оквир (*framework*)

Neuroph радни оквир омогућава библиотеку Java класа и апликацију (*Neuroph studio*) са графичким интерфејсом за креирање вештачких неуронских мрежа [4]. Најважнија карактеристика *Neuroph* радног оквира по којој се издваја од осталих, а која је определила коришћење у настави на АТУСС – Одсек ВИШЕР у деловима предмета који се односе на вештачке неуронске мреже, је врло интуитиван и једноставан програмски интерфејс (API) и јасна логика захваљујући доменски оријентисаном дизајну. Апликација за развој вештачких неуронских мрежа заснована на *NetBeans* платформи омогућава високо професионални кориснички интерфејс, напредне функционалности и интеграцију са Java развојним окружењем у оквиру исте апликације. То га чини јединственим алатом ове врсте зато што се у оквиру исте апликације може развити вештачка неуронска мрежа, обучити, тестирати и уградити у кориснички програм. *Neuroph* подржава основне врсте вештачких неуронских мрежа и алгоритама за обучавање, а поред тога има и специјализоване алате за примену неуронских мрежа у одређеним областима.

Један од основних принципа за разумевање вештачких неуронских мрежа од стране студената је да *Neuroph* радни оквир буде интуитиван и једноставан за коришћење како на нивоу графичког интерфејса, тако и на нивоу целокупног радног оквира [2]. Овај радни оквир управо на тај начин омогућава све кораке у креирању и обучавању

вештачке мреже кроз: припрему података (нормализацију), креирање новог пројекта, формирање података за обуку (*training set*), израду и обучавање вештачке неуронске мреже, и на крају тестирање.

The image shows a screenshot of the Neuroph studio application. The main window displays a data table with 11 columns. The first 8 columns are labeled 'Input1' through 'Input8', and the last 3 columns are labeled 'Output1', 'Output2', and 'Output3'. The table contains numerical data for each row, representing normalized input and output values. The interface includes a menu bar at the top, a toolbar, and a status bar at the bottom.

Слика 2 – Нормализовани подаци у апликацији Neuroph studio

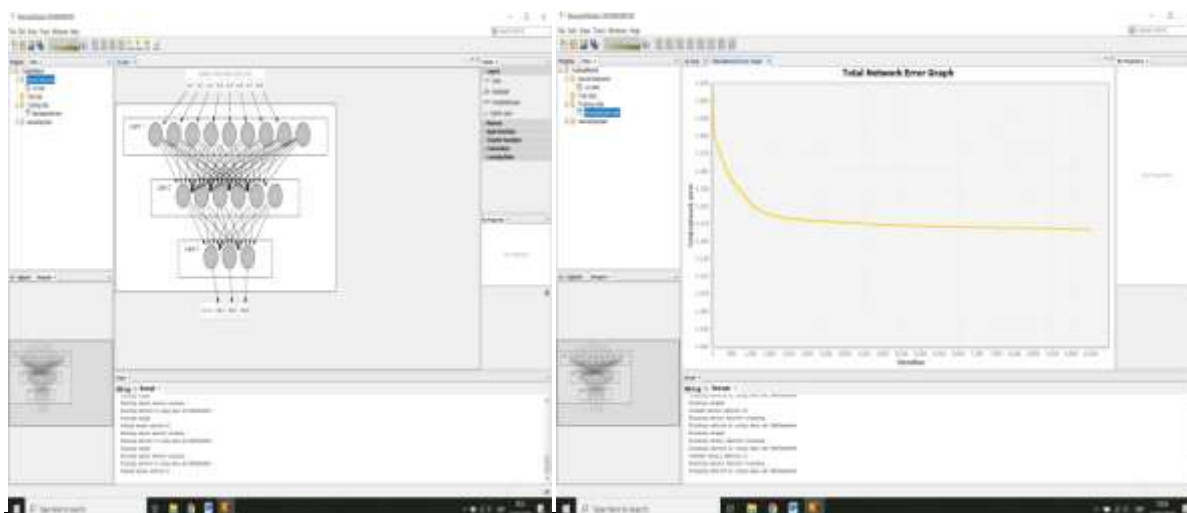
У конкретној примени *Neuroph* радног оквира у учењу вештачких неуронских мрежа реализован је студијски пример израде вештачке неуронске мреже кроз пројекат „*PredikcijaRezultataTakmicenja*”. Први корак је да се изврши припрема (прилагођавање) улазних података за обучавање и тестирање вештачке неуронске мреже. То подразумева нормализацију података, односно примену одређених техника за припрему података за обраду помоћу вештачких неуронских мрежа, која подразумева свођење вредности на интервал $[0, 1]$ или $[-1, 1]$. Ова функционалност је од суштинске важности јер обезбеђује интегрисано решење за припрему података и обучавање у *Neuroph* оквиру.

У конкретном наведеном примеру на слици 2 је приказана реализована нормализација података за 8 улаза и 3 излаза, за узорак резултата претходних такмичења на основу којих ће се обучавати вештачка неуронска мрежа.

2.3. Креирање обучавање вештачке неуронске мреже

Вештачка неуронска мрежа за предикцију резултата такмичења садржи 8 улаза и 3 излаза. За пројектовање ове мреже је важно како одредити број неурона у скривеном слоју. Применом принципа да се проба са почетним бројем неурона у скривеном слоју који је знатно мањи од димензије улазног слоја, води то тога да у првом покушају дефинишемо 5 скривених неурона. На слици 3 је приказано коришћење интерфејса *Neuroph* радног оквира за креирање и обучавање почетне варијанте наведене мреже.

Прихватљиво коначно решење вештачке неуронске мреже се добија кроз више итерација структуре мреже и варијацијом параметара обучавања. Важно је правилно одредити коефицијент обучавања η . Имајући у виду да пракса показује да се η може кретати зависно од конкретног проблема у опсегу од 0,001 до 10, за прву верзију мреже предлаже се вредност 0,2.



Слика 3 – Верзија 1 вештачке неуронске мреже

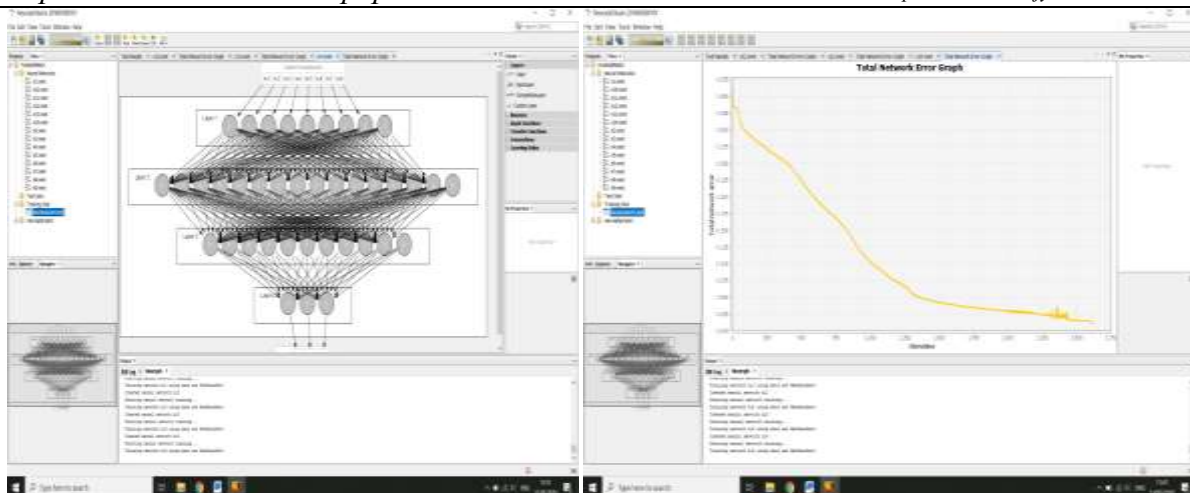
Обучавање мреже се врши по изабраном алгоритму пропацијом уназад, тако што се на улазу користи изабрани сет података и кроз одређени број итерација добија на излазу резултат са прихватљивим коефицијентом грешке. На слици 3 је приказан следећи резултат обучавања вештачке неуронске мреже: Верзија 1, број скривених неурона 5, Train (max error: 0.01, learning rate: 0.2, momentum: 0.7), Total Mean Square Error: 0.10901300460337103. Може се закључити да резултати обучавања ове мреже нису прихватљиви, односно да су укупни средњи квадрат грешке и број итерација обучавања неприхватљиви.

Имајући у виду претходне резултате, може се закључити да није довољан број скривених неурона, тако да је потребно креирати нову верзију вештачке неуронске мреже. Друга верзија вештачке неуронске мреже за предикцију резултата такмичења ће такође садржати 8 улаза и 3 излаза, али ћемо повећати број скривених неурона на 10. Резултат обучавања вештачке неуронске мреже: Верзија 2, број скривених неурона 10, Train (max error: 0.01, learning rate: 0.2, momentum: 0.7), Total Mean Square Error: 0.03338184152968076.

Може се закључити да су резултати обучавања ове мреже нешто бољи, али још нису прихватљиви, односно да су укупни средњи квадрат грешке и број итерација обучавања неприхватљиви.

Трећа верзија вештачке неуронске мреже за предикцију резултата такмичења ће такође садржати 8 улаза и 3 излаза, али ћемо додати још један слој скривених неурона. Један слој скривених неурона ће задржати 10 неурона а нови слој ће садржати 8 неурона. Резултат обучавања вештачке неуронске мреже: Верзија 3, број скривених неурона 10, 8, Train (max error: 0.01, learning rate: 0.2, momentum: 0.7), Total Mean Square Error: 0.006649497685939103. Резултати обучавања ове мреже су знатно бољи, са знатно смањеним бројем итерација, али још нису прихватљиви.

Четврта верзија вештачке неуронске мреже за предикцију резултата такмичења ће такође садржати 8 улаза и 3 излаза. У циљу побољшања резултата повећавамо број неурона у скривеним слојевима на 15 и 10.



Слика 4 – Верзија 4 вештачке неуронске мреже (одабрана)

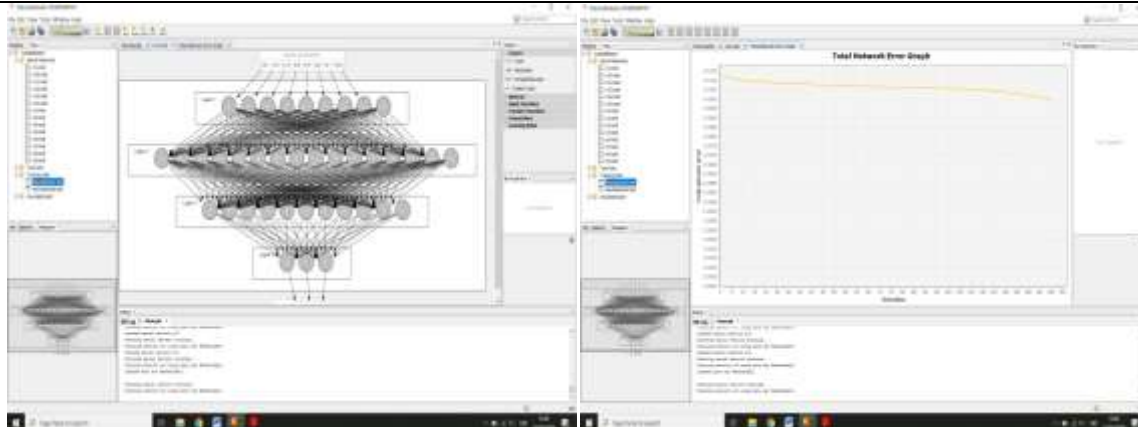
На слици 4 је приказано коришћење интерфејса *Neuroph* радног оквира за креирање и обучавање четврте варијанте мреже. Резултат обучавања вештачке неуронске мреже: Верзија 4, број скривених неурона 15,10, Train (max error: 0.01, learning rate: 0.2, momentum: 0.7), Total Mean Square Error: 0.0059850644345920585. Укупни средњи квадрат грешке и број итерација обучавања ове вештачке неуронске мреже је прихватљив, као и број итерација.

Може се закључити да се кроз више итерација структуре мреже и варијацијом параметара обучавања долази до решења са прихватљивим резултатима обучавања. Студенти целокупан процес реализују у *Neuroph* радном оквиру, где је могуће поред варијације параметара у овом примеру вршити промене и других параметара, као што су избор различитих активационих функција, метода и алгоритама обучавања. У овом случају је одабрана последња верзија вештачке неуронске мреже за наредну фазу – тестирање.

2.4. Тестирање вештачке неуронске мреже

За тестирање је одабрана верзија 4 вештачке неуронске мреже са најприхватљивијим резултатима обучавања. Тестирање се врши једноставно, тако да се у итерацијама на улаз мреже учитавају различити подаци у односу на обучавање и на излазу се анализира одступање стварних резултата од резултата обучавања.

Након тестирања, може се закључити да је мрежа успешно обучена ако је укупни средњи квадрат грешке и број итерација након тестирања приближан резултатима обучавања. На слици 5 је приказано коришћење интерфејса *Neuroph* радног оквира за тестирање одабране четврте варијанте мреже. Резултат тестирања вештачке неуронске мреже: Верзија 4, број скривених неурона 15,10, Train (max error: 0.01, learning rate: 0.2, momentum: 0.7), Total Mean Square Error: 0.005657758717512678. Разлика за укупни средњи квадрат грешке и број итерација тестирања у односу на обучавање ове вештачке неуронске мреже је прихватљива, као и број итерација.



Слика 5 – Тестирање верзије 4 вештачке неуронске мреже

Студенти могу лако разумети и закључити да ли је креирање и обучавање вештачке неуронске мреже успешно, као на пример у наведеном пројекту „PredikcijaRezultataTakmicenja”.

2.5. Апликација

```
public static void main(String[] args) {

    String datasetPutanja = "obucavanje.txt";
    int brojUlaza = 8;
    int brojSriivenihNeurona1 = 15;
    int brojSriivenihNeurona2 = 10;
    int brojIzlaza = 3;

    // Ucitavanje seta podataka
    DataSet setTreniranje = null;
    try {
        setTreniranje = TrainingSetImport.importFromFile(datasetPutanja, brojUlaza, brojSriivenihNeurona1,
        brojSriivenihNeurona2, brojIzlaza, ",");
    } catch (FileNotFoundException ex) {
        System.out.println("Fajl nije pronadjen");
    } catch (IOException | NumberFormatException ex) {
        System.out.println("Greska pri citanju formata fajla");
    }
}

// Kreiranje veštačke neuronske mreže tipa MultilayerPerceptron
MultiLayerPerceptron mreza = new MultiLayerPerceptron(TransferFunctionType.SIGMOID, 8, 15,10, 3);

// postavljanje parametara za obucavanje
MomentumBackpropagation ucenjePravila = (MomentumBackpropagation) mreza.getLearningRule();
ucenjePravila.setLearningRate(0.2);
ucenjePravila.setMomentum(0.7);
// treniranje mreze
System.out.println("Treniramo ...");
mreza.learn(setTreniranje);
System.out.println("Završeno");

// testiranje mreze
DataSet setTest = null;
try {
    setTest = TrainingSetImport.importFromFile("testiranje.txt", brojUlaza, brojSriivenihNeurona1,
    brojSriivenihNeurona2, brojIzlaza, ",");
} catch (FileNotFoundException ex) {
    System.out.println("Fajl nije pronadjen");
} catch (IOException | NumberFormatException ex) {
    System.out.println("Greska pri citanju formata fajla");
}

for (DataSetRow row : setTest.getRows()) {
    mreza.setInput(row.getInput());
    mreza.calculate();
    //Preko Arrays - vraca ref ako se ide row.getInput().toString()
    System.out.println("Ulaz je " + Arrays.toString(row.getInput()));
    System.out.println("Izlaz je " + Arrays.toString(mreza.getOutput()));
}

}
```

Слика 6 – Програмски код

Neuroph радни оквир омогућава библиотеку Java класа поред приказане апликације (*Neuroph studio*) са графичким интерфејсом за креирање вештачких неуронских мрежа [4]. То значи да студенти могу на приступачан начин развити апликацију за креирање и обучавање вештачких неуронских мрежа и уградити у друге апликације. На слици 6 је приказан пример програмског кода апликације за разматрани пројекат „*PredikcijaRezultataTakmicenja*”, који обухвата учитавање сета података за обучавање, креирање *MLP (MultiLayerPerceptron)* вештачке неуронске мреже уз помоћ истоимене класе, постављање параметара за обучавање у коришћењу алгорита *MomentumBackpropagation* и тестирање, односно коришћење мреже.

3. ЗАКЉУЧАК

У оквиру предмета Меко рачунарство на АТУСС – Одсек ВИШЕР за поглавље предмета које се односи на вештачне неуронске мреже предвиђено је коришћење *Neuroph* радног оквира. У раду је дат пример пројекта за креирање и обучавање вештачке неуронске мреже за предикцију резултата спортског такмичења који би се креирао у оквиру једне лабораторијске вежбе. Студенти се при креирању пројекта ослањају на знања из претходно реализованих предмета из објектно оријентисаног програмирања. У току реализације пројекта студенти креирају и образлажу различита решења која нису моделована или су превише комплексна за математички модел самостално користећи *Neuroph* радни оквир кроз све фазе креирања и обучавања вештачке неуронске мреже (дефинисање сета података за обучавање и тестирање мреже на основу узорка резултата претходног такмичења, више итерација у дефинисању вештачке неуронске мреже и подешавање параметара за обучавање). Предвиђено је да студенти самостално ураде и образложе свој пројекат са сопственим избором структуре мреже и сета података за обучавање на основу којег би у потпуности научили да креирају сопствене апликације.

4. ЛИТЕРАТУРА

- [1] Alma Alanis Nancy, Arana-Daniel, Carlos Lopez-Franco (2019). *Artificial Neural Networks for Engineering Applications*. USA, 1st Edition. ISBN: 9780128182482
- [2] J. Steven Perry (2018). *Create an artificial neural network using the Neuroph Java framework*. IMB Tutorial.
- [3] Eva Volna (2013). *Introduction to Soft Computing Book*. Ostrava, Czech Republic
- [4] Zoran V. Ševarec (2012.). *Software Engineering of Intelligent Systems*. Doctoral Dissertation, University of Belgrade, Faculty of Organizational Sciences.
- [5] Kurhe A.B., Satonkar S.S., Khanale P.B. and Shinde Ashok (2011). *Soft Computing and its Applications*. BIOINFO Soft Computing, Volume 1, Issue 1, 2011, pp-05-07
- [6] Lotfi A. Zadeh (1994). *Fuzzy Logic, Neural Networks and Soft Computing*. Communications of the ACM, Vol37, No 3