

ВЕБ АПЛИКАЦИЈА ЗА ВИЗУЕЛИЗАЦИЈУ ОБУЧАВАЊА НЕУРОНСКЕ МРЕЖЕ У РЕАЛНОМ ВРЕМЕНУ**A WEB APPLICATION FOR REAL-TIME VISUALIZATION OF NEURAL NETWORK TRAINING**

Дане Милишић, Факултет техничких наука, Нови Сад

Област – Примењене рачунарске науке и информатика

Кратак садржај – Овај рад се бави развојем веб апликације за визуелизацију обучавања неуронске мреже у реалном времену. Циљ овог рада је истражити да ли је могуће уз помоћ актуелних веб технологија, попут Риект ЈС библиотеке, реализовати алат за праћење обучавања неуронске мреже у реалном времену. Решење имплементирано у овом раду има могућност коришћења различитих библиотека за испуњавање истих захтева, како би се могло извршити тестирање њихових перформанси и одабрати најбоље решење за специфичне захтеве.

Кључне речи: ЈаваСкрипт, Риект ЈС, Сокет ИО, Пајтон, Неуронске мреже, Веб апликација

Abstract – This paper deals with development of web application for real-time visualization of neural network training. Goal of this paper is researching if it is possible to develop tool for tracking of neural network training in real-time, using current web technology, such as React.JS. Application developed in this research has different solutions for same problems. Therefore, this application can be used for testing different solutions and selecting the best one for different segments of application.

Keywords: JavaScript, React.JS, Socket.IO, Python, Neural networks, Web application

1. УВОД

Тренутно, колико је аутору познато, не постоји алат за визуелизацију и праћење рада неуронске мреже који је реализован као веб апликација. Проблем који би решио такав алат огледа се у томе да у случају да корисник жели да сагледа структуру или да прати обучавање неке велике неуронске мреже, морао би модел те неуронске мреже да поседује на својој машини. Такође, корисник не би био приморан да поседује разне технологије које захтевају алати развијени као десктоп апликација, али би се такође и заобишао процес у ком корисник треба да пронађе адекватан модел.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Дину Драган, ванредни професор.

Највећи потенцијални проблем на који се наилази приликом реализације решења веб апликације за визуелизацију обучавања неуронске мреже у реалном времену, јесу нешто слабије перформансе најпопуларнијих веб технологија, када је реч о обради великог броја података. Из претходно поменутог разлога, решење описано у овом раду садржи више решења за различите сегменте у апликацији, како би се омогућило тестирање различитих технологија.

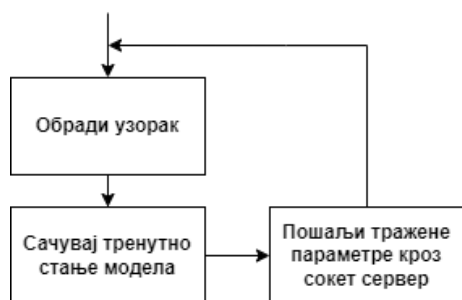
Рад има следећу структуру. У другом поглављу се описује развој веб апликације за визуелизацију неуронске мреже у реалном времену, где су представљене коришћене технологије и структура апликације. У трећем поглављу представљена је имплементација апликације, пропраћена сликама корисничког интерфејса. У четвртном поглављу представљена је анализа перформанси и употребљивости система. Пето поглавље представља закључак, док је у шестом наведена литература. На самом крају рада налази се биографија аутора.

2. РАЗВОЈ ВЕБ АПЛИКАЦИЈЕ ЗА ВИЗУЕЛИЗАЦИЈУ ОБУЧАВАЊА НЕУРОНСКЕ МРЕЖЕ У РЕАЛНОМ ВРЕМЕНУ

Задатак овог рада, представљен у наслову, може се поделити у три целине, а то су: визуелизација у веб апликацији [1], комуникација у реалном времену и обучавање неуронске мреже. Идеја је да поменуте три целине буду развијене као три апликације које нису директно зависне једне од других. С обзиром на то да је неопходно реализовати комуникацију у реалном времену између веб апликације и серверске апликације са неуронском мрежом, потребно је креирати сокет (енгл. *socket*) сервер. Сокет сервер има улогу да отвара сокете између клијената или клијента и сервера, који омогућава комуникацију у реалном времену, и за разлику од традиционалне комуникације преко ХТТП (енгл. *Hypertext Transfer Protocol*) захтева, овај вид комуникације остаје отворен. Развој сокет сервера је реализован уз помоћ Сокет ИО (енгл. *Socket.IO*) технологије [2], која осим што омогућава развој сервера и клијената, такође и омогућава комуникацију између апликација које су развијене у различитим технологијама. Комуникација преко сокет сервера се огледа у томе да се клијенти првенствено повежу на

сервер, а затим док неки клијенти емитују догађај, који носи неку поруку, други клијенти се претплаћују на такав догађај. Када клијент емитује неки догађај, сервер тај догађај пролеђује клијентима који су претплаћени на њега.

Развој неуронске мреже представља велик посао, међутим уз помоћ библиотеке Тенсорфлоу (енгл. *Tensorflow*) [3] у склопу програмског језика Пајтон (енгл. *Python*) [4] процес је у великој мери поједностављен. Неуронска мрежа, такође, мора бити адаптирана тако да врши комуникацију уз помоћ сокет сервера, али и да складишти податке, након сваког узорка приликом обучавања. Начин на који се неуронска мрежа обучава јесте да након сваког обрађеног узорка, чув тренутно стање свих тежина и бајес (енгл. *bias*) вредности у моделу, а затим одређене вредности шаље корисницима који су се на исте претплатили (слика 1).



Слика 1. Процес обучавања неуронске мреже уз комуникацију и чување стања

Како би се реализовала визуелизација у веб апликацији која подржава промене у реалном времену, неопходно је употребити неку библиотеку која користи виртуелни ДОМ (енгл. *Document Object Model*). Пример такве библиотеке је Риект ЈС (енгл. *React.js*) [5], која такође подржава интеграцију са другим технологијама, попут библиотеке за изградњу графова и графикана [6].

Кориснички део апликације пре свега има задатак да представи структуру неуронске мреже уз помоћ графа, који такође треба да подржава интеракцију са корисником. Неке од библиотека које омогућавају интеграцију оваквих врста графова у веб апликацију су Ситоскејп ЈС (енгл. *Cytoscape.js*) [7] и Сигма ЈС (енгл. *Sigma.js*) [8]. Интеракција са графом подразумева зум и пан, који су подразумевано имплементирани, филтер податак који се приказује, али и дефиницију различитих догађаја. Филтер се односи на податке који су приказани на графу и захтева да се после примене учита граф са новим подацима. Поновно учитавање се лако реализује у Риект ЈС библиотеци, тако што се подаци чувају у стањима, а променом стања све компоненте које су повезане са истим се поново учитавају. Такође, коришћењем Ситоскејп ЈС библиотеке могуће је дефинисати акције које прате одређене догађаје и на тај начин се омогућава директна интеракција са корисником. Специфично у овој апликацији, кориснику треба омогућити да притиском на одређену грану или чвор на графу добије приказ промена тежине, тј. бајес вредности кроз време, за одабрани елемент на графу.

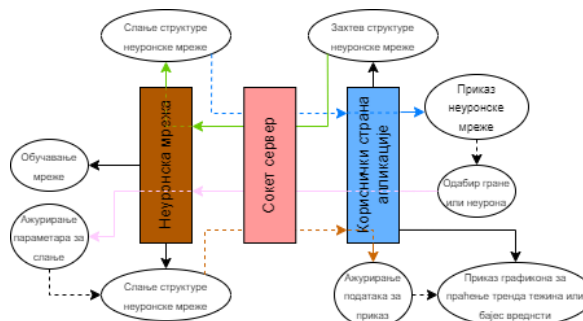
Како би се приказала промена поменутих вредности кроз време, неопходно је у апликацију интегрисати линијски графикон. Неке од библиотека које подржавају линијске графиконе су Чарт ЈС (енгл. *Chart.js*) [9], Ричартс (енгл. *Recharts*) [10] и Виктори (енгл. *Victory*) [11]. С обзиром на то да подаци на графикону подлежу константним променама, неопходно их је као и у случају са графом дефинисати уз помоћ стања у склопу Риект ЈС библиотеке. Такође, ствара се потреба за процесом који ће у позадини да очекује новопристигле податке и исте додаје у поменуто стање, како би били приказани на графикону (слика 2).



Слика 2. Дијаграм процеса који преузима податке из сокета и смешта их на графикон

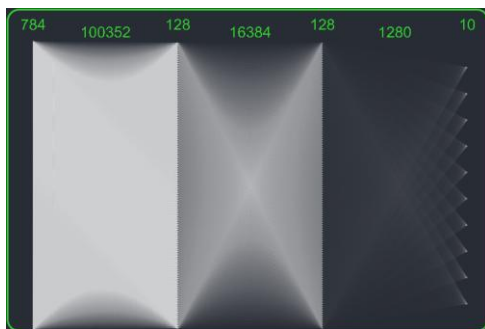
3. РЕШЕЊЕ

Решење веб апликације за визуелизацију обучавања неуронске мреже у реалном времену се састоји из три сегмента, као што је то описано у претходном поглављу. Кориснички део апликације и неуронска мрежа комуницирају у реалном времену уз помоћ сокета сервера, док у међувремену обе стране омогућавају извршавање својих функционалних захтева (слика 3).



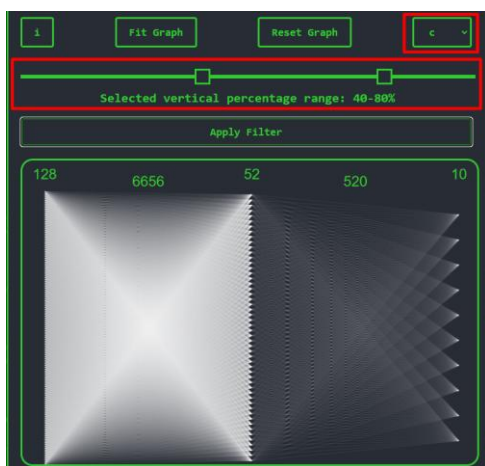
Слика 3. Дијаграм функционалних захтева и комуникације у склопу система

Приказ неуронске мреже имплементиран је уз помоћ Ситоскејп библиотеке. Неуронска мрежа је приказана кроз граф, који такође садржи и лателе које информишу корисника о броју неурона и грана у сваком слоју неуронске мреже (слика 4).



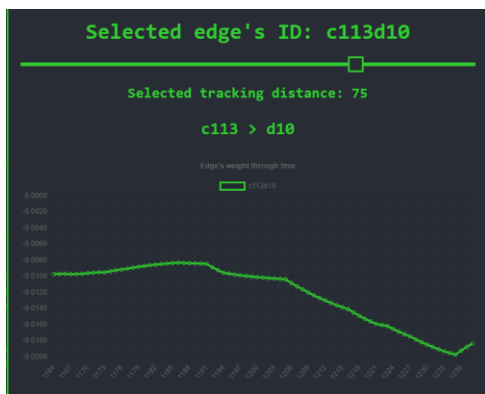
Слика 4. Приказ неуронске мреже на корисничкој страни апликације

Кориснику је омогућена интеракција са графом у виду зума, пана и филтрирања података. Корисник може да одабере слој који ће бити приказан као централни, а уз њега ће се, ако постоје, приказати претходни и наредни слој. Такође, у склопу одабраног слоја, може да бира који део ће бити приказан. Филтрирање ажурира и латенце за приказ броја неурона и грана у сваком слоју (слика 5).



Слика 5. Филтрирање података који ће се приказати на графу

Корисник, такође, има могућност да одабере грану или неурон за коју ће пратити тренд промене тежине, тј. бајес вредности. Тренд је приказан преко линијског графикона, који је такође подложен променама у реалном времену које одговарају обучавању неуронске мреже (Слика 6).



Слика 6. Приказ тренда промене тежине за одабрану грану

4. АНАЛИЗА ПЕРФОРМАНСИ И УПОТРЕБЉИВОСТИ СИСТЕМА ЗА ВИЗУЕЛИЗАЦИЈУ

Анализа перформанси овог система подразумева испитивање како се систем понаша када су у оптицају велике количине података. Када је у питању обучавање неуронске мреже узорак по узорак са праћењем долази до значајног успоравања. Обучавање неуронске мреже на уобичајен начин са 60 хиљада података у три епохе траје свега 10 до 15 секунди, међутим када се то промени на обучавање узорак по узорак уз праћење, брзина обучавања износи 150 узорака по минути.

Када је реч о перформансама сокет сервера, очекивано, комуникација се извршава у реалном времену.

Део апликације, чије су перформансе највише погођене растом броја података јесте корисничка страна. Пре свега учитавање структуре неуронске мреже се знатно успорава са растом броја неурона и грана, које, такође, утиче и на време одзива (Табела 1).

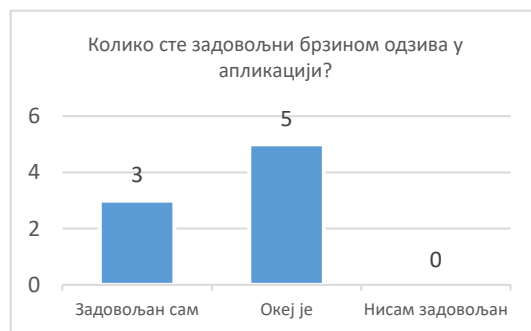
Број елемената	Време учитавања	Време одзива
≈ 1000	100ms	real-time
≈ 10000	1000ms	100ms
≈ 100000	8000ms	3000ms

Табела 1. Време учитавања и одзива у односу на број елемената

Када је реч о анализи употребљивости, пре свега је спроведено тестирање са корисницима, који су по завршетку тестирања попуњавали анкету, где су имали прилику да оцене апликацију и њену употребљивост. Оцене за неке од најзначајнијих сегмената приказане су на сликама 7 и 8.



Слика 7. Расподела оцена квалитета интеракције у апликацији



Слика 8. Однос оцена у вези са брзином одзива у апликацији

Тakoђе, корисници који су имали искуства са алатима за визуелизацију неуронских мрежа су одговарали на

питање да ли је могуће да овакав алат замени постојеће, који су развијени као десктоп апликације и одговори су били подељени. Два корисника су се изјаснили да би радије наставили са коришћењем постојећих алата, док је један корисник рекао како види перспективу у оваквом типу алата.

5. ЗАКЉУЧАК

У склопу овог рада представљена је веб апликација за визуелизацију обучавања неуронске мреже у реалном времену, која је развијена у три дела: неуронска мрежа, сокет сервер и кориснички интерфејс. Сокет сервер је развијен у ЈаваСкрипт програмском језику, кошћењем Сокет ИО технологије за комуникацију у реалном времену. Неуронска мрежа је развијена у Пајтон програмском језику користећи Тензерфлоу библиотеку, али и Сокет ИО Пајтон клијент библиотеку како би се омогућила конекција на сокет сервер. Кориснички интерфејс је развијен у склопу библиотеке Риект ЈС, а коришћене су и библиотеке Ситоскејп, Чарт ЈС, Ричартс и Виктори за визуелизацију података.

Главна мотивација за развијање оваквог алата као веб апликације јесте омогућавање лакше и шире примене алата, без нужне потребе да се неуронска мрежа налази на истој машини где и алат, што такође утиче на то да корисникова машина не мора бити натпросечних перформанси.

Апликација је развијена на такав начин да корисницима пружа могућност бирања различитих алата за визуелизацију, како би се подстакло тестирање различитих алата и на тај начин, у перспективи, дошло до идеалног алата сачињеног од више мањих.

Приликом развоја апликације, највећи непревазиђени изазов јесте нешто слабије перформансе корисничког интерфејса, кад су у оптицају веће количине података које се требају приказати. Стога, први наредни кораци у развоју овог алата били би пре свега тражење библиотеке за визуелизацију графа која је боље оптимизована, а вероватно и замене главне библиотеке Риект ЈС, за неку са бољим перформансама, као што су Вју (енгл. *Vue*) или Лептос (енгл. *Leptos*).

6. ЛИТЕРАТУРА

- [1] Fundamentals of Data Visualization,
урл: <https://clauswilke.com/dataviz/>
[Пристапљено 15.9.2024.]
- [2] Socket.IO – Official website,
урл: <https://socket.io/>
[Пристапљено 15.9.2024.]
- [3] TensorFlow – Official website,
урл: <https://www.tensorflow.org/>
[Пристапљено 15.9.2024.]
- [4] Python – Official website,
урл: <https://www.python.org/>
[Пристапљено 15.9.2024.]

- [5] React – Official website,
урл: <https://react.dev/>
[Пристапљено 15.9.2024.]
- [6] Vizzu – 40 types of Data Visualization Charts and Graphs,
урл: <https://www.vizzu.io/blog/data-visualization-types>
[Пристапљено 15.9.2024.]
- [7] Cytoscape.js – Official website,
урл: <https://js.cytoscape.org/>
[Пристапљено 15.9.2024.]
- [8] Sigma.js – Official website,
урл: <https://www.sigmapjs.org/>
[Пристапљено 15.9.2024.]
- [9] Chart.js – Official website,
урл: <https://www.chartjs.org/docs/latest/>
[Пристапљено 15.9.2024.]
- [10] Recharts – Official website,
урл: <https://recharts.org>
[Пристапљено 15.9.2024.]
- [11] Victory – Official website,
урл: <https://commerce.nearform.com/open-source/victory/docs/victory-chart/>
[Пристапљено 15.9.2024.]

Биографија:



Дане Милишић рођен је 9. априла 2000. године у Новом Саду. Завршио је основну школу „Доситеј Обрадовић“ у Новом Саду, а затим друштвено-језички смер гимназије „Исидора Секулић“ у Новом Саду. Школске 2019/2020. године уписује основне академске студије Информационог инжењеринга на Факултету техничких наука у Новом

Саду. Положио је све испите прописане планом и програмом, а потом и дипломирао у септембру 2023. године, са просечном оценом студија 8.88. Школске 2023/2024. године уписује мастер академске студије Примењених рачунарских наука и информатике на Факултету техничких наука у Новом Саду. Положио је све испите прописане планом и програмом на поменутиим студијама у склопу модула Мултимедија и рачунарске игре.

контакт: milisic@uns.ac.rs