

Анализирање на квалитет со невронски мрежки оптимизирани со генетски алгоритам

- Проект по вештачка интелигенција-

Факултет за информатички науки и компјутерско инженерство

Универзитет “Свети Кирил и Методиј”, Скопје

Изработил: Антонио Крпачовски 231024

Ментор: д-р Соња Гиевска

Декември 2025, Скопје

Содржина

Вовед	3
Методологија.....	3
Анализа на архитектурата	5
Анализа на генетскиот алгоритам	5
Потешкотии и предизвици	7
Заклучок	7
Користена литература и инспирации.....	8

Апстракт

Во овој технички извештај е прикажан систем за класификација на квалитетот на вино користејќи невронска мрежа оптимизирана со генетски алгоритам. Целта на проектот е да се развие модел кој врз основа на хемиски карактеристики на виното автоматски одредува дали примерокот е „добар“ или „лош“. Проектот ги опфаќа фазите на обработка на податочно множество, имплементација на сопствена невронска мрежа, интеграција со генетски алгоритам и развој на кориснички интерфејс со PyGame. Дополнително, извршени се неколку експерименти за анализа на влијанието на генетски параметри врз точноста на моделот. Резултатите покажуваат дека бројот на неврони, единки и процентот на мутација значително ја менуваат стабилноста и перформансите на моделот.

Вовед

Во рамки на предметот Вештачка интелигенција беше развиен проект со цел да се изгради систем што ќе го класифицира квалитетот на вино врз основа на неговите хемиски својства. Проектот „Wine quality testing“ користи невронска мрежа оптимизирана со генетски алгоритам .

Мотивацијата зад проектот произлегува од потребата да се истражи примената на генетските алгоритми во практични сценарија, како и нивната ефективност во оптимизирање на невронски мрежи. Дополнително, проектот претставува можност за експериментирање со различни архитектури, параметри и техники на оптимизација, со цел анализа на нивното влијание врз точноста, стабилноста и конвергенцијата на системот.

Методологија

Проектот е имплементиран во програмскиот јазик Python , поради неговата едноставност, читливост и голем број библиотеки за генетски алгоритам и обработка на податоци.

За имплементација на генетскиот алгоритам е користена PyGAD, која овозможува едноставно дефинирање на атрибути на генетскиот алгоритам, како што се селекција, кросовер и мутација. Оваа библиотека исто така овозможува лесно менување и еволуција на геномите според дефинирани критериуми кое што значително го поедноставува процесот на оптимизација на тежините на невронската мрежа.

За невронска мрежа првично се започна со користење на однапред готова библиотека PyTorch, но поради потешкотии при поврзување на PyTorch невронската мрежа со PyGAD генетскиот алгоритам, беше изработена сопствена имплементација на невронска мрежа.

Невронската мрежа е составена од еден влезен слој, п скриени слоеви и еден излезен слој. Невронската мрежа служи како модел кој ја предвидува категоријата на производот врз основа на неговите карактеристики.

Секоја врска во мрежата има своја тежина, а токму овие тежини се оптимизираат со помош на генетскиот алгоритам. Наместо традиционален тренинг со градиентски методи, проектот го користи генетскиот алгоритам за да ги пронајде најдобрите вредности на тежините што резултираат со највисока точност.

Како активациска функција во скриениот и во излезниот слој е користена сигмоидната функција, дефинирана како:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Оваа функција ги трансформира влезните вредности во опсегот (0, 1).

За поврзување на невронската мрежа и генетскиот алгоритам беше користена фитнес функција, која ја пресметува точноста на мрежата за даден геном. При пишувањето на оваа функција беше искористена помош од генеративна вештачка интелигенција.

Инспирирано од проекти пронајдени на интернет како и проекти од други колеги, изработен е кориснички интерфејс за проектот.

Корисничкиот интерфејс овозможува корисникот да ги внесе карактеристиките на едно вино, а отпосле со помош на невронската мрежа да се евалуира дали тоа вино е добро или лошо.

За овој интерфејс користено е PyGame, кој овозможува лесно цртање на полиња за внес на податоци, копчиња и динамичко прикажување на резултатите.

За оптимизација на PyGame беше користена генеративна вештачка интелигенција подобрување за на визуелните елементи.

За визуелизација и графичка анализа на резултатите беше употребена библиотеката matplotlib.

Анализа на архитектурата

Невронската мрежа која е користена заеваулирање на податочното множество се состои од:

- 12 влезни неврони
- еден или два скриени слоеви
- еден излезен неврон

Користената е сигмоидна активациска функција, која е соодветна за бинарна класификација. Сигмоидната функција секогаш враќа резултати од 0-1, па затоа за резултати ≥ 0.5 сметаме дека виното е “добро”, а за резултати < 0.5 сметаме дека виното е “лошо”.

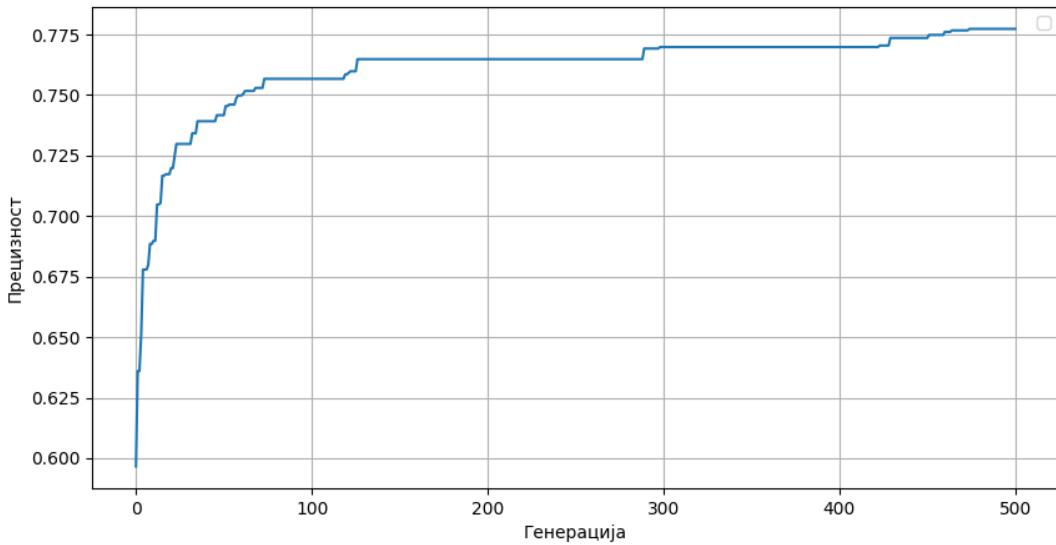
Анализа на генетскиот алгоритам

Генетскиот алгоритам оптимизира околу ($W \times \text{biases}$) параметри, каде W е бројот на тежини. Бројот на параметри расте експоненцијално со бројот на неврони. Клучни параметри на генетскиот алгоритам се:

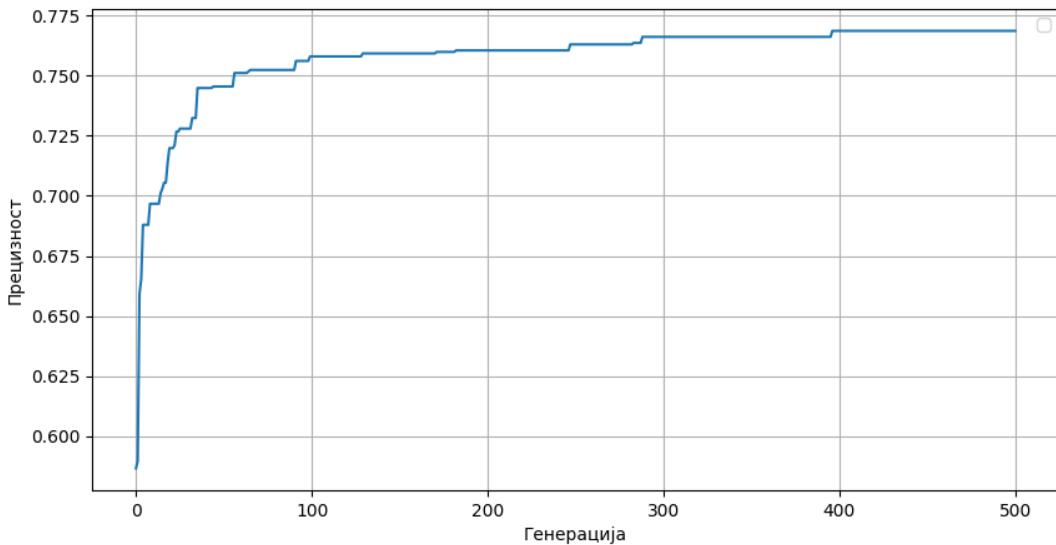
- **Selection method**
 - PyGAD стандардно користи "tournament" или "rank" селекција. Ова обезбедува стабилност и избегнување на прерана конвергенција.
- **Mutation**
 - Мутацијата е најважниот оператор. Техничката анализа покажува дека премногу мал процент ($< 10\%$) значи премногу бавно менување на геномите, додека преголем процент ($> 40\%$) покажува нестабилност и губење на добри решенија, поради тоа оптималната вредност за мутација е меѓу 10–20%.
- **Population size**
 - Поголемата популација ја стабилизира еволуцијата, дава побавна конвергенција, подобро го покрива просторот на можни тежини. Исто така со поголема популација расте сложеноста на алгоритмот:

$$O(\text{population_size} \times \text{fitness_evaluations})$$

Техничка анализа на резултатите покажува дека моделот конвергира брзо во првите 20–80 генерации. Откако ќе се постигне точност од ~76–78%, моделот почнува да стагнира. Генетскиот алгоритам не гарантира глобален оптимум, но обезбедува стабилни решенија.



Слика 1: перформансите на невронска мрежа со еден скриен слој оптимизирана со генетски алгоритам



Слика 2: перформансите на невронска мрежа со два скриени слоеви оптимизирана со генетски алгоритам

Потенцијални подобрувања можат да бидат направени со повеќе скриени слоеви во мрежата, користење адаптивна мутација, хибриден пристап со генетски алгоритам за иницијализација и backpropagation за подобрување на тежините на невронската мрежа.

Потешкотии и предизвици

Во текот на развојот на системот се соочив со повеќе технички предизвици кои бараа внимателен пристап и иновативни решенија.

Првичната идеја беше да се користи PyTorch како основа за невронската мрежа, но несовпаѓањето помеѓу PyTorch и PyGAD библиотеките, како и неуспешни обиди за создавање геноми во PyTorch, доведе до одлуката да напишам сопствен код за невронската мрежа.

Првичните резултати беа субоптимални поради лошата конфигурација на параметрите на генетскиот алгоритам. Сепак, врз основа на стекнатото искуство и PyGAD документацијата, беа пронајдени параметри кои овозможуваат значително подобрување на оптимизација на тежините на невронската мрежа.

Пишувачкото на фитнес функцијата за PyGAD исто така беше голем предизвик поради начинот на кој треба да се обработат влезните и излезните податоци, но со помош на PyGAD документацијата и генеративна вештачка интелигенција, беше направено функција која конзистентно генерира влезни и излезни податоци соодветни за генетскиот алгоритам.

Заклучок

Проектот овозможи практично запознавање со генетските алгоритми и нивната примена како метод за оптимизација на невронски мрежи. Беше покажано дека, иако генетските алгоритми ретко даваат совршени решенија, тие конзистентно генерираат доволно добри и употребливи модели. Генетскиот алгоритам се покажува како добар метод за оптимизација кога традиционалните методи не се применливи.

Користена литература и инспирации

Инспирации за невронска мрежа:

- <https://github.com/ygutgutia/Snake-Game-Genetic-Algorithm>
- https://www.youtube.com/watch?v=Wo5dMEP_BbI&list=PLQVvaa0QuDcjD5BAw2DxE6OF2tius3V3

Користени библиотеки:

- <https://pygad.readthedocs.io/en/latest/>
- <https://www.pygame.org/docs/>
- https://pygame.readthedocs.io/en/latest/1_intro/intro.html

Слични трудови од кои е земено инспирација:

- I. Jordanov, “Конструкција на портфолио со граф невронски мрежи,” дипломска работа, под менторство на проф. д-р Мирослав Мирчев, ФИНКИ Универзитет „Св. Кирил и Методиј“, Скопје.
- <https://ceur-ws.org/Vol-3312/paper13.pdf>
- <https://www.mdpi.com/2076-3417/11/2/744>