Документација за проектот “Wine quality testing” по вештачка интелигенција

Изработил: Антонио Крпачовски 231024

Ментор: д-р Соња Гиевска

1. Вовед

Во рамки на предметот вештачка интелигенција, го развив проектот „Класификација на квалитет на вино со невронска мрежа оптимизирана со генетски алгоритам“. Целта на проектот беше да се изгради систем што автоматски ќе проценува дали примерок на вино е “добар“ или “лош“ врз основа на неговите хемиски својства.

Проектот опфаќа неколку клучни концепти од обработка на податоци како и вештачка интелигенција, вклучувајки подготовка на податочно множество, имплементација на невронска мрежа како и генетски алгоритам за опцимизација на тежините на невронската мрежа.

Целта на овој проект беше да се запознаам со примената на генетскиот алгоритам и да експериментирам со различни параметри и архитектури на генетскиот алгоритам и невронската мрежа, со цел да се анализира како тие влијаат врз точноста и стабилноста на моделот.

2. Користени технологии и алатки

2.1. Python

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

2.2. PyGAD

За генетскиот алгоритам, ја користам библиотеката 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.

2.3. Невронска мрежа

За невронска мрежа најпрво започнав со користење на однапред готова библиотека PyTorch, но поради потешкотии при поврзување на PyTorch невронската мрежа со PyGAD генетскиот алгоритам, одлучив да се обидам да напишам свој код за невронска мрежа.

Невронската мрежа е составена од еден влезен слој, n 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). Бидејќи задачата е бинарна класификација (“good” или “bad”), сигмоидната функција е идеална, затоа што излезната вредност ≥ 0.5 се класифицира како “good“, а < 0.5 како “bad“.

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.

2.4. PyGame

Инспириран од проекти кои ги најдов на интернет како и проекти од други колеги одлучив да се обидам да направам мој сопствен кориснички интерфејс за проектот.

Како цел за корисничкиот интерфејс сакав да може корисникот да се обиде да ги внесе карактеристиките на едно вино, отпосле со помош на невронската мрежа да се евалуира дали тоа вино е добро или лошо.

За овој интефејс користев PyGame, кој овозможи лесно цртање на полиња за внес на податоци, копчиња и динамичко прикажување на резултатите.

За оптимизација на PyGame беше користена генеративна вештачка интелигенција за да ги подобрам визуелните елементи како и да дадам по жив изглед на апликацијата.

3. Тестирање

Во делот за тестирање сакав да видам колкаво влијание имаат одредени атрибути, како што се: бројот на неврони во скриениот слој, бројот на скриени слоеви, процентот на гени кој мутираат како и бројот на единки. За дадените експерименти кодот се извршува 20 пати за да се добијат просечните резултати.

3.1. Тестирање на бројот на скриени слоеви и неврони

За оваа цел направив нов код за невронска мрежа кој е многу сличен на оригиналниот но, има два скриени слоеви наместо еден. За наредниот експеримент константо е дека бројот на генерации е 500, бројот на единки кој се зачувуваат е 5, процентот на мутација е 15% и бројот на единки е 50.

3.1.1. Со 5 неврони во скриениот слој:

Со еден скриен слој, невронската мрежа достигнува просечна точност од околку 76.5% (76,36% - 76,93%). Најголемо зголемување на точност има во првите 70 генерации, а отпосле 190-та генерација зголемувањата се минимални и незначајни.

Со два скриени слоја, невронската мрежа повторно достигнува просечна точност од околу 76.5% (76,18%-77.03%). Најголемо зголемување на точност има во првите 40 генерации, а отпосле 350-та генерација зголемувањата се минимални и незначајни.

3.1.2. Со 3 неврони во скриениот слој:

Со еден скриен слој, невронската мрежа достигнува просечна точност од околку 76.4% (76,36% - 77,13%). Најголемо зголемување на точност има во првите 70 генерации, а отпосле 230-та генерација зголемувањата се минимални и незначајни.

Со два скриени слоја, невронската мрежа повторно достигнува точност од околу 76.5% (76,38%-76.63%). Најголемо зголемување на точност има во првите 50 генерации, а отпосле 400-та генерација зголемувањата се минимални и незначајни.

3.1.3. Со 10 неврони во скриениот слој:

Со еден скриен слој, невронската мрежа достигнува просечна точност од околку 77.6% (76,78% - 78,43%). Најголемо зголемување на точност има во првите 10 генерации, а отпосле 100-та генерација зголемувањата се минимални и незначајни.

Со два скриени слоја, невронската мрежа повторно достигнува точност од околу 77.4% (76,66%-78.63%). Најголемо зголемување на точност има во првите 20 генерации, а отпосле 100-та генерација зголемувањата се минимални и незначајни.

3.1.4. Резултати од првиот експеримент

Точноста на невронската мрежа се зголемува при зголемување на бројот на неврони. Точноста исто така се зголемува при зголемување на бројот на слоеви, но не дотолку значително како со бројот на неврони.

3.2. Тестирање на процентот на мутација

За наредниот експеримент константо е дека бројот на генерации е 500, бројот на единки кој се зачувуваат е 5, бројот на неврони во скриениот слој е 5, бројот на скирени слоеви е 1 и бројот на единки е 50.

3.2.1. Со 15% процент на мутација:

Со 15% процент на мутација, невронската мрежа достигнува просечна точност од околку 76.5% (76,36% - 76,94%). Најголемо зголемување на точност има во првите 70 генерации, а отпосле 200-та генерација зголемувањата се минимални и незначајни.

3.2.2. Со 30% процент на мутација:

Со 30% процент на мутација, невронската мрежа достигнува просечна точност од околку 76.3% (75,73%-78.13%). Најголемо зголемување на точност има во првите 90 генерации, а отпосле 400-та генерација зголемувањата се минимални и незначајни.

3.2.3. Со 50% процент на мутација:

Со 50% процент на мутација, невронската мрежа достигнува просечна точност од околку 76.9% (76,33%-78.43%). Најголемо зголемување на точност има во првите 100 генерации, а отпосле 350-та генерација зголемувањата се минимални и незначајни.

3.2.4. Резултати од вториот експеримент

Со зголемување на процентот на мутација невронската мрежа станува многу по хаотична. Има шанса да се добијат многу по високи точности, но исто така има шанса точноста драстично да падне иако според резултатите изгледа како да треба да се стабилизира.

3.3. Тестирање на бројот на единки

За наредниот експеримент константо е дека бројот на генерации е 500, бројот на единки кој се зачувуваат е 5, бројот на неврони во скриениот слој е 5, бројот на скирени слоеви е 1 и процентот на мутација е 15%.

3.3.1. Со 50 единки:

Со 50 единки, невронската мрежа достигнува просечна точност од околку 76.6% (76,25% - 77,34%). Најголемо зголемување на точност има во првите 70 генерации, а отпосле 200-та генерација зголемувањата се минимални и незначајни.

3.3.2. Со 100 единки:

Со 100 единки, невронската мрежа достигнува просечна точност од околку 77.4% (77,22%-77.92%). Најголемо зголемување на точност има во првите 40 генерации, а отпосле 190-та генерација зголемувањата се минимални и незначајни.

3.3.3. Со 200 единки:

Со 200 единки, невронската мрежа достигнува просечна точност од околку 77.5% (76,82%-77.91%). Најголемо зголемување на точност има во првите 20 генерации, а отпосле 120-та генерација зголемувањата се минимални и незначајни.

3.3.4. Резултати од третиот експеримент

Со зголемување на бројот на единки алгоритмот станува многу по стабилен. Многу побрзо се стигнува до стабилна точност, но по цена на побавно извршување на кодот.

4. Заклучок

Проектот ми помогна да научам како да го користам генетскиот алгоритам и како да го применувам во реални проекти. Научив за што точно се користи овој алгоритам и како неговите параметри влијаат на резултатите.

Генетскиот алгоритам ретко дава совршени решенија, но секогаш обезбедува доволно добри решенија кои можат да се земат предвид при оптимизација. Поради тоа, генетскиот алгоритам се користи главно како метод за оптимизација на постоечки системи, а не како самостоен алгоритам за решавање на сите проблеми.

5. 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=PLQVvvaa0QuDcjD5BAw2DxE6OF2tius3V3

Користени библиотеки:

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