



Софийски университет „Св. Кл. Охридски“
Факултет по математика и информатика

Регресия с размита невронна мрежа и извличане на правила от медицински записи

Курсов проект по
Размити множества

Николай Димов Паев
Факултетен номер: 2MI3400435
Магистърска програма: Изкуствен Интелект

Съдържание

Въведение.....	2
Данни.....	2
Предварителна обработка на данните.....	2
Нормализация на данните.....	3
Създаване на база правила.....	4
Откриване на броя и вида на правилата чрез клъстеризация.....	4
Научаване на параметрите чрез backpropagation в размитата мрежа.....	6
Резултати след обучението и сравнение с линейна регресия.....	7
Визуализация като размити правила и тълкуване.....	7
Заключение.....	8
Използвана литература и други източници.....	8
Връзка към хранилище с изходния код.....	8

Въведение

Медицинската наука се характеризира с неточности и субективно експертно знание. По тази причина използването на размита логика изглежда като добър апарат за моделиране на явления в този домейн. В настоящия проект ще бъде имплементирана и тествана размита невронна мрежа, чиито извод е еквивалентен на извод от типа Такаги-Сугено [1]. Невронната мрежа ще бъде използвана за регресионна задача за предсказване на зрение на пациент след корекция.

Данни

Налични са 6800 амбулаторни листове на пациенти прегледани от кабинет по очни болести в периода от 10 години. Данните са в xml формат и от тях са извлечени следните полета:

- дата на посещение
- тип на посещение
- рождена дата на пациента
- пол на пациента
- статус на здравно осигуряване
- код на основна диагноза
- код на допълнителни диагнози
- кодове на извършени процедури
- анамнеза
- обективен статус

С цел запазване на анонимността на пациентите, други данни не са извлечени.

Предварителна обработка на данните

Полето анамнеза е свободен текст, макар да съдържа важна информация то е трудно за автоматична обработка.

Полето обективен статус съдържа информация за зрението на пациента. Въпреки, че то също е въведено като текст, то е в известен смисъл структурирано. Например началото му винаги има вида:

ВОД=<нативно зрение>с<корекция диоптри сфера>дсф<корекция диоптри цилиндър>дц<градус на диоптрите цилиндър>гр.=<коригирано зрение>

Тук ВОД означава visus oculi dextra (в превод: зрение на дясно око). По аналогичен начин е изписано и за лявото око.

Зрението е записано като дробно число в интервала $[0, 1] \cup \{1/\sim, \text{zero}\}$. Зрение 1 означава добро зрение, зрение близо до 0 означава слабо зрение. Диоптрите сфера и цилиндър са в интервала $[-\infty, +\infty]$, като стойностите са през 0,25. Градусите са в $[0, 360]$.

Не всички полета от зрението присъстват в записа. Например хора, които имат сериозно увредено зрение, не са изследвани за очила, затова при тях липсват полетата с корекция. Обратно някои, които се оплакват, че се нуждаят от очила, не са изследвани за нативно зрение, а директно са коригирани и при тях не присъства полето за нативно зрение. Освен това наличието на диоптри цилиндър също не е задължително. В такива случаи ги приемаме за 0. Например:

```
...  
'anamnesis': 'Нуждае се от нови очила за четене....',  
'h_state': 'ВОД=с+0.4дсф=1.0 ВОС=с+0.5дсф=1.0  
...
```

Пример 1. Поле анамнеза и поле обективно състояние на пациент с изпуснати данни.

Съществува зависимост между коригираното зрение, диоптрите за корекция и нативното зрение. В случай, когато полето за нативно зрение не присъства, то е автоматично попълнено чрез използване на алгоритъма kNN (с $k = 2$) на базата на примери с всички полета.

Настоящия проект се интересува само от

- възрастта на пациента
- наличие на заболяване - булева променлива в $\{0, 1\}$. Под заболяване се разбира наличие на сериозно очно заболяване като катаракта (старческо перде), глаукома и др. Късогледство, далекогледство и други подобни нарушения на зрението не са счетени за заболявания.
- зрение преди корекция (нативно зрение)
- зрение след корекция

Под корекция се разбира поставяне на очила.

Нормализация на данните

Данните са приведени в $[0, 1]$. Възрастта е приведена в този интервал чрез делене на 100. Зрението няма нужда да се нормализира.

Създаване на база правила

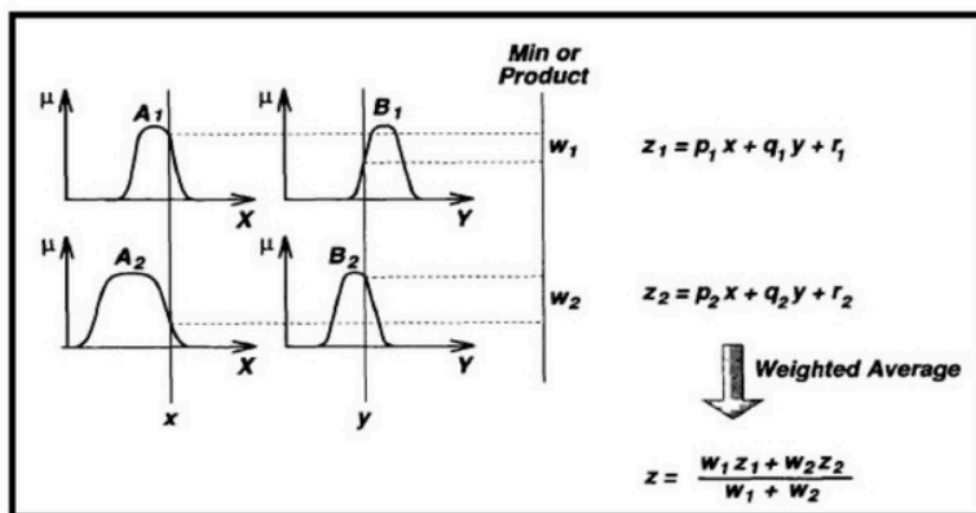
Ще бъде създадена база правила от вида:

Ако възрастта на пациента е А, той има наличие В на заболяване и зрението му преди корекция е С, то зрението му след корекция е D.

За функция на принадлежност на множествата е избрана гаусовата функция:

$$e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Базата от правила ще позволява изводи от типа Такаги-Сугено. По този начин зрението D ще бъде автоматично дефъзифицирано.



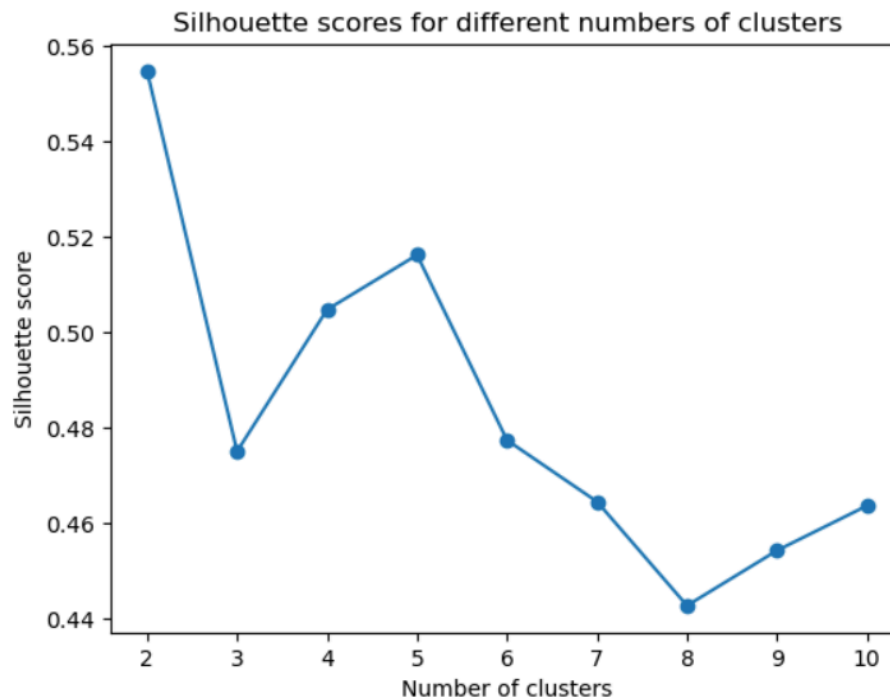
Фиг. 1. Извод от типа Такаги-Сугено.

Коефициентите p , q и r за всяко правило, както и параметрите на функцията на принадлежност за всяко множество ще бъдат научени чрез backpropagation. Изображението е от [2]

Откриване на броя и вида на правилата чрез клъстеризация

Преди да се тренира невронната мрежа трябва да се избере нейната архитектура - да се установят броя на правилата, както и вида им: за всяко правило, кои размити множества ще бъдат антецедентите му. Тези хиперпараметри ще бъдат получени с клъстеризация върху данните.

Върху нормализираните данни е приложен алгоритъм k-means. Използвана е имплементацията от библиотеката Scikit learn [3]. За определяне на k е използвана метриката silhouette score. Най-висок такъв резултат се получава при $k = 2$.



Фиг 2. silhouette score при различно k

Получените клъстери имат следните центроиди:

Центроид 1 [7.23266129e-01, 1.00000000e+00, 3.95987903e-01, 5.61580645e-01]

Центроид 2: [4.82674294e-01, 1.77635684e-15, 6.86199499e-01, 9.05352163e-01]

Тук първата координата съответства на нормализираната възраст, втората на наличието на патология, третата и четвъртата съответно на зрение преди и след корекция.

За по-добра визуализация са представени в таблицата:

	Възраст	Патология	Зрение преди корекция	Зрение след корекция
Центроид 1	72	1	0.39	0.56
Центроид 2	48	0	0.68	0.9

Със броя на клъстерите ще бъде инициализиран броя на правилата на мрежата.

Със стойностите на атрибутите на центроидите ще бъдат инициализирани параметрите μ на гаусовата функция за принадлежност на множествата на antecedентите.

Параметрите σ ще бъдат инициализирани с 0.2. Тази стойност е избрана експериментално, защото твърде висока и твърде ниска води до изчезване на градиента и бавна сходимост при обучение.

Множествата В съответстващи на полето патология заемат стойности от 0 и 1, т.е. те са размити синглетони. Представянето им като такива обаче не е диференцируемо и ще попречи на научаването на параметрите. По тази причина те са представени със същата гаусова функция на принадлежност.

Научаване на параметрите чрез backpropagation в размитата мрежа

Параметрите на функциите на принадлежност на множествата А, В и С, както и теглата за приноса на всяка променлива във резултата на всяко правило ще бъдат научени чрез backpropagation върху еквивалентна размита невронна мрежа.

```
def forward(parameters, x):
    antecedent_sets_mu, antecedent_sets_sigma, rules_bias, rules_parameters = parameters
    sum_firing_strengths = 0
    sum_rule_results = 0
    for rule_index, rule in enumerate(self.rules_antecedent_sets):
        firing_strength = self.t_norm(
            np.array([
                gaussian(x[antecedent_index],
                    antecedent_sets_mu[antecedent_index][antecedent_set_index],
                    antecedent_sets_sigma[antecedent_index][antecedent_set_index]
                )
            ])
            for antecedent_index, antecedent_set_index in enumerate(rule))
        sum_firing_strengths = sum_firing_strengths + firing_strength

        rule_result = firing_strength*(np.dot(x, rules_parameters[rule_index]) + rules_bias[rule_index])
        sum_rule_results = sum_rule_results + rule_result
    return sum_rule_results/sum_firing_strengths
```

Фиг. 3 Код за изчисление на правата посока в размитата невронна мрежа.

Невронната мрежа работи в правата си посока така:

Изчислява се степента на принадлежност на всеки от antecedentите във всяко правило. Степента на стрелба на всяко правило се получава като се прилага t-норма (минимум) върху степените на принадлежност на неговите antecedenti.

Изчислява се резултата от всяко правило, като скалярно произведение на antecedentите и параметрите на правилото + отклонение. Резултатът от базата е претеглено по силата на стрелба средно на резултатите на всяко правило.

```
idx = np.arange(len(X), dtype='int32')
np.random.shuffle(idx)
i = 0
# training loop
for e in range(epoch):
    for b in range(0, len(idx), gradient_accumulation_step):
        batch_X = [ X[i] for i in idx[b:min(b+gradient_accumulation_step, len(idx))]]
        batch_Y = [ Y[i] for i in idx[b:min(b+gradient_accumulation_step, len(idx))]]
        def loss(parameters, t):
            return -logprob(parameters, batch_X, batch_Y)

        def callback(params, t, g):
            print("Iteration {} Epoch {} log likelihood {}".format(i, e, -loss(params, t)))

        i += 1
        self.parameters = sgd(grad(loss), self.parameters, step_size=step_size, num_iters=1, callback=callback)
```

Фиг. 4. Алгоритъм за обучение

Обучението е извършено на 10 епохи със стохастично спускане по градиента на функцията на загубата. За такава е избрана квадратична грешка. Степента на обучение е 0.0001. Прилага се също и L2 регуляризация с коефициент 0.1. Стъпката за акумулиране на градиента е 32. За съставяне на изчислителния граф на градиента е използвана библиотеката autograd на Python. [5]

Резултати след обучението и сравнение с линейна регресия

За оценка на регресионни модели често се използва R squared.

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}},$$

$$= 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}.$$

Фиг 5. Формула на R squared

Тази оценка пресмята 1 - квадратичната грешка на модела върху квадратичното отклонение на целевата променлива от средното ѝ.

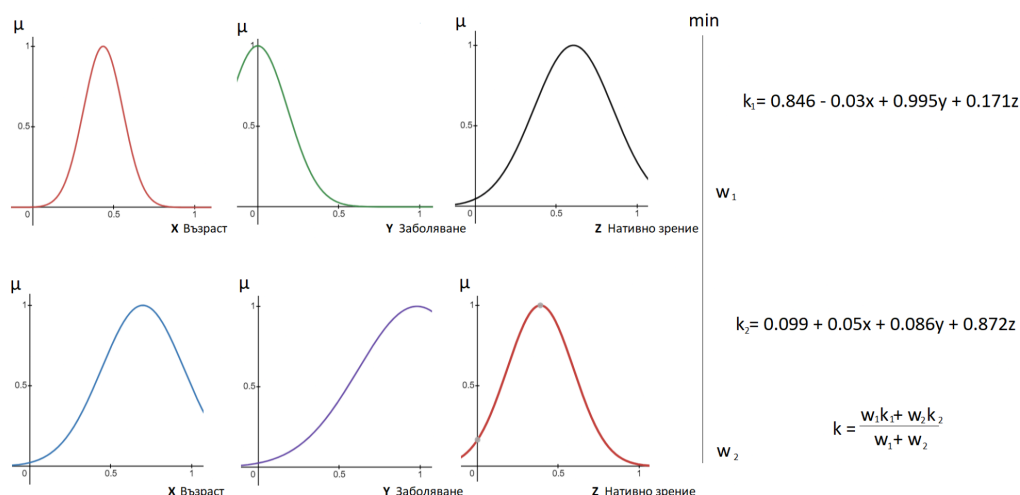
За сравнение на резултатите на размитата мрежа е обучена линейна регресия. Използвана е имплементацията от библиотеката Scikit Learn [5]. Резултатите са представени в таблицата:

	Линейна регресия	FNN модел
R squared	0.519	0.605

Таблица 1: метрика R squared върху резултати от размитата невронна мрежа и линейна регресия

Размитата мрежа дава малко по-висока стойност на R squared от линейната регресия. Най-вероятно това се дължи на факта, че размитата мрежа моделира нелинейна функция. Въпреки това този резултат не е висок.

Визуализация като размити правила и тълкуване



Фиг 6. Еквивалентно представяне на мрежата като база от правила

Предимство на размитите невронни мрежи е, че те са еквивалентни на база от правила, което улеснява разбирането им. На Фиг 6. е показана еквивалентната база от размити правила.

Наблюдаваните правила могат да се тълкуват така:

Първото правило е за пациенти, които нямат очно заболяване.

Забелязва се, че коефициентът пред z е нисък, което означава, че нативното зрение не оказва влияние на стойността на зрението след корекция. Това е очаквано, тъй като често хората, които имат нужда от очила виждат слабо без тях и перфектно след коригиране с правилни лещи.

Второто правило е за пациенти, страдащи от очно заболяване.

Интересно е, че параметърът x и в двете правила има коефициент близък до 0, което означава, че почти не участва в пресмятането на зрението. Това може да се тълкува като пример, че други фактори оказват по-голямо влияние върху зрението от възрастта.

Заклучение

В настоящия проект беше имплементирана и приложена размита невронна мрежа върху регресионна задача от медицински домейн. Въпреки, че не са постигнати високи резултати предвид малкото данни и сравнително ниското им качество, беше демонстрирано предимство на размития модел пред линейната регресия от гледна точка на резултати и разбираемост.

Използвана литература и други източници

[1] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control.

[2] Визуализация на Такаги-Сугено модел от

<https://researchhubs.com/post/engineering/fuzzy-system/takagi-sugeno-fuzzy-model.html>

[3] K-means <https://scikit-learn.org/stable/modules/clustering.html#k-means>

[4] MODELING AN INTELLIGENT SYSTEM FOR THE ESTIMATION OF TECHNICAL STATE OF CONSTRUCTION STRUCTURES S. Terenchuk , A. Pashko, Taras Shevchenko, B. Yeremenko

[5] Документация на autograd <https://autograd.readthedocs.io/en/latest/>

[6] Linear regression

[https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.htm](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html)
!

Връзка към хранилище с изходния код

<https://github.com/NikolayDPaev/medical-fuzzy-rules-extraction>