

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ПЕТРА ВЕЛИКОГО»**

**Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий**

Дисциплина «Гибридные интеллектуальные системы и мягкие вычисления»

ОТЧЕТ

Лабораторная работа №1

на тему:

«Нечеткая нейронная сеть для классификации паттернов»

Выполнил:

студент группы 3540901/02001

Дроздов Никита Дмитриевич

«__» _____ 2021г., _____
(подпись)

Проверила:

Бендерская Елена Николаевна

«__» _____ 2021г., _____
(подпись)

Санкт-Петербург 2021

Оглавление

ПОСТАНОВКА ЗАДАЧИ	3
ОПИСАНИЕ ТЕОРЕТИЧЕСКОЙ БАЗЫ ИССЛЕДОВАНИЯ.....	3
НЕЧЕТКАЯ НЕЙРОННАЯ СЕТЬ	4
НЕЙРОННАЯ СЕТЬ С РАДИАЛЬНОЙ БАЗИСНОЙ ФУНКЦИЕЙ	4
АРХИТЕКТУРА НЕЧЕТКОЙ НЕЙРОННОЙ СЕТИ И АЛГОРИТМ ОБУЧЕНИЯ	5
Нечеткая кластеризация с максимальным подсчетом и алгоритмом отсечения..	7
АЛГОРИТМ ОБРЕЗКИ.....	8
ТЕСТИРОВАНИЕ FNN.....	9
РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ.....	9
ВЫВОД	13

Постановка задачи

В предлагаемой работе рассматривается задача классификации шаблонов с помощью нечеткой нейронной сети. Рассматриваемая сеть является модифицированной версией нейронной сети с радиальной базисной функцией (RBFNN). FNN использует контролируемую нечеткую кластеризацию и алгоритм отсечения для определения точного количества кластеров с правильным центроидом и шириной для формирования узлов обработки в скрытом слое. Эти кластеры представляют собой нечеткие гиперсферы множеств (FSH), которые определяются функцией нечеткой принадлежности. Обучение между скрытым слоем и выходным слоем, которое выполняется с использованием алгоритма LMS в RBFNN, избегается, а выход определяется с помощью операции нечеткого объединения. Функция нечеткой принадлежности защищает сгруппированные шаблоны, что обеспечивает 100% точность для набора данных, используемого во время обучения. В отличие от других алгоритмов кластеризации, используемых для построения скрытого слоя RBFNN, предлагаемый алгоритм кластеризации не зависит от параметров настройки и быстро обучается, и извлекается. Таким образом, FNN сокращает время вычислений, гарантирует 100% точность для любого обучающего набора и обеспечивает превосходную и сопоставимую точность распознавания для наборов данных с точным количеством FSH в скрытом слое. Следовательно, предлагаемую FNN можно использовать для классификации паттернов.

Описание теоретической базы исследования

Распознавание образов в последние годы стало активной областью исследований в различных областях. Для линейно разделяемых шаблонов классификации используется однослойная нейронная сеть, тогда как для неразделимых шаблонов используется многоуровневая сеть, такая как нейронная сеть с обратным распространением ошибок (EBPNN). EBPNN включает три уровня: входной, скрытый и выходной. Определение количества нейронов скрытого слоя - главный недостаток EBPNN. Чтобы преодолеть этот недостаток ввели нейронную сеть с радиальной базисной функцией. Алгоритмы контролируемой или неконтролируемой кластеризации используются для определения соответствующего количества нейронов в скрытом слое.

Таким образом, нечеткая нейронная сеть была предложена путем интеграции алгоритмов нечеткой кластеризации для определения количества скрытых нейронов нейронной сети с радиальной базисной функцией. Нечеткая логика предоставляет альтернативный способ представления лингвистических и субъективных атрибутов реального мира в вычислениях. Нечеткие множества обеспечивают значение принадлежности к элементам, принадлежащим универсуму дискурса, в то время как четкие множества дают двоичный результат. Значение принадлежности зависит от нечеткой функции принадлежности. Для

создания нечеткого множества используются различные функции принадлежности.

Нечеткая нейронная сеть

Нечеткая нейронная сеть — это сплав ИНС и нечеткой логики. Эти сети стали популярными, поскольку получаемый результат не четкий, а нечеткий. Благодаря этому частичная принадлежность к определенному классу помогает лучше классифицировать. Значение нечеткого членства и операции с нечетким множеством играют важную роль в работе нечетких нейронных сетей. Ввиду преимуществ нечеткой логики и ИНС, нечеткая нейронная сеть с минимальным максимальным значением (FMNN). FMNN основан на накоплении нечетких гипербоксов. Обучение FMNN осуществляется путем создания гипербоксов, принадлежащих разным классам. Алгоритм обучения состоит из двух частей: расширения и сжатия. В процессе расширения размер гипербоксов определяется коэффициентом расширения. Процесс сжатия удаляет перекрытие между гиперблоками других классов.

Позже эта концепция была расширена для построения гипербоксов с использованием помеченных и немеченых данных. Нечеткая нейронная сеть с компенсаторным нейроном (FMCN) предложена Нандедкармом и Бисвасом.

Предлагаемая FNN использует алгоритм отсечения после нечеткой кластеризации для оптимизации нейронов в скрытом слое и операция нечеткого объединения для поиска выходных данных сети. Этот классификатор обеспечивает улучшение предыдущих классификаторов.

Нейронная сеть с радиальной базисной функцией

Модель RBFNN использует радиальную базисную функцию в качестве своей функции активации для скрытых нейронов. Это сеть прямого распространения. RBFNN представляет собой трехуровневую сеть (Рисунок 1). Входной уровень не является обрабатывающим уровнем и принимает входные шаблоны. Нейроны во входном слое равны общему количеству функций, присутствующих в шаблоне. Скрытый слой создается путем кластеризации, и количество нейронов равно количеству кластеров. Выходной слой — это слой классов, в котором количество нейронов равно количеству классов.

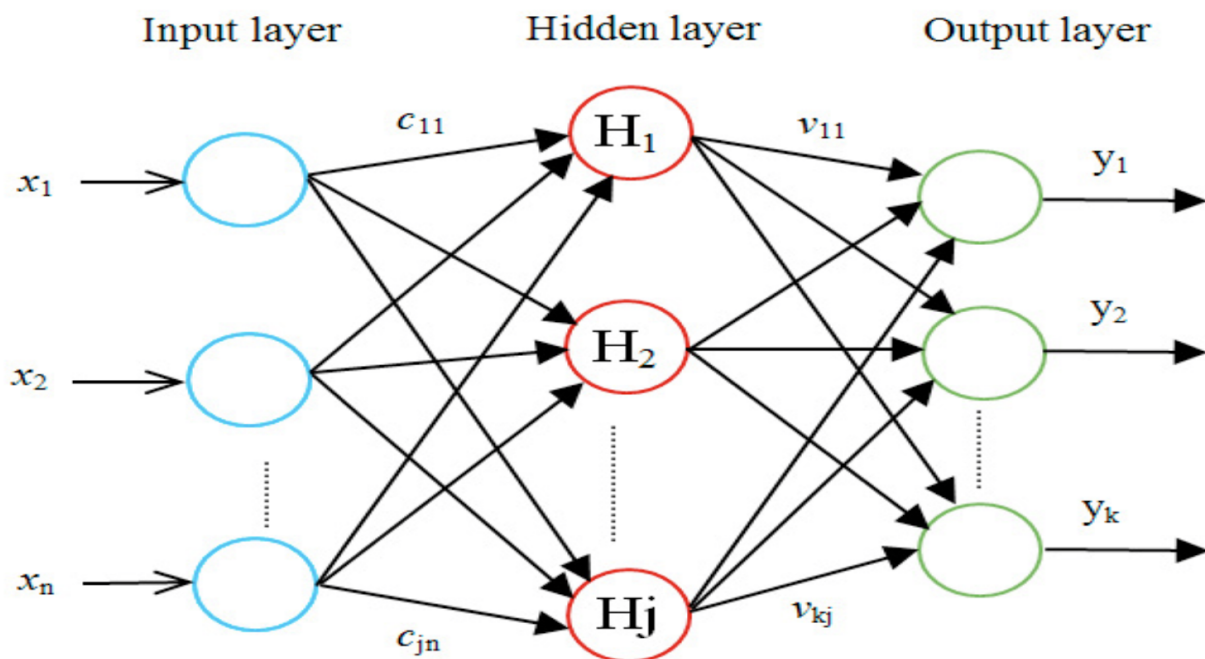


Рисунок 1 - Архитектура нейронной сети с радиальной базисной функцией

Связи между входными слоями и скрытым слоем являются весами, относящимися к центроиду кластеров, которые определяются во время кластеризации. Веса между скрытыми слоями и выходным слоем определяются алгоритмом наименьших средних квадратов (LMS).

Ключевой вопрос с RBFNN заключается в определении центров и радиусов кластеров, а также общего количества узлов обработки в скрытом слое. Параметры центров и радиусов кластеров оцениваются с помощью контролируемой или неконтролируемой кластеризации. Самым простым методом является алгоритм кластеризации К-средних.

Архитектура нечеткой нейронной сети и алгоритм обучения

Архитектура нечеткой нейронной сети показана на рисунке 2.

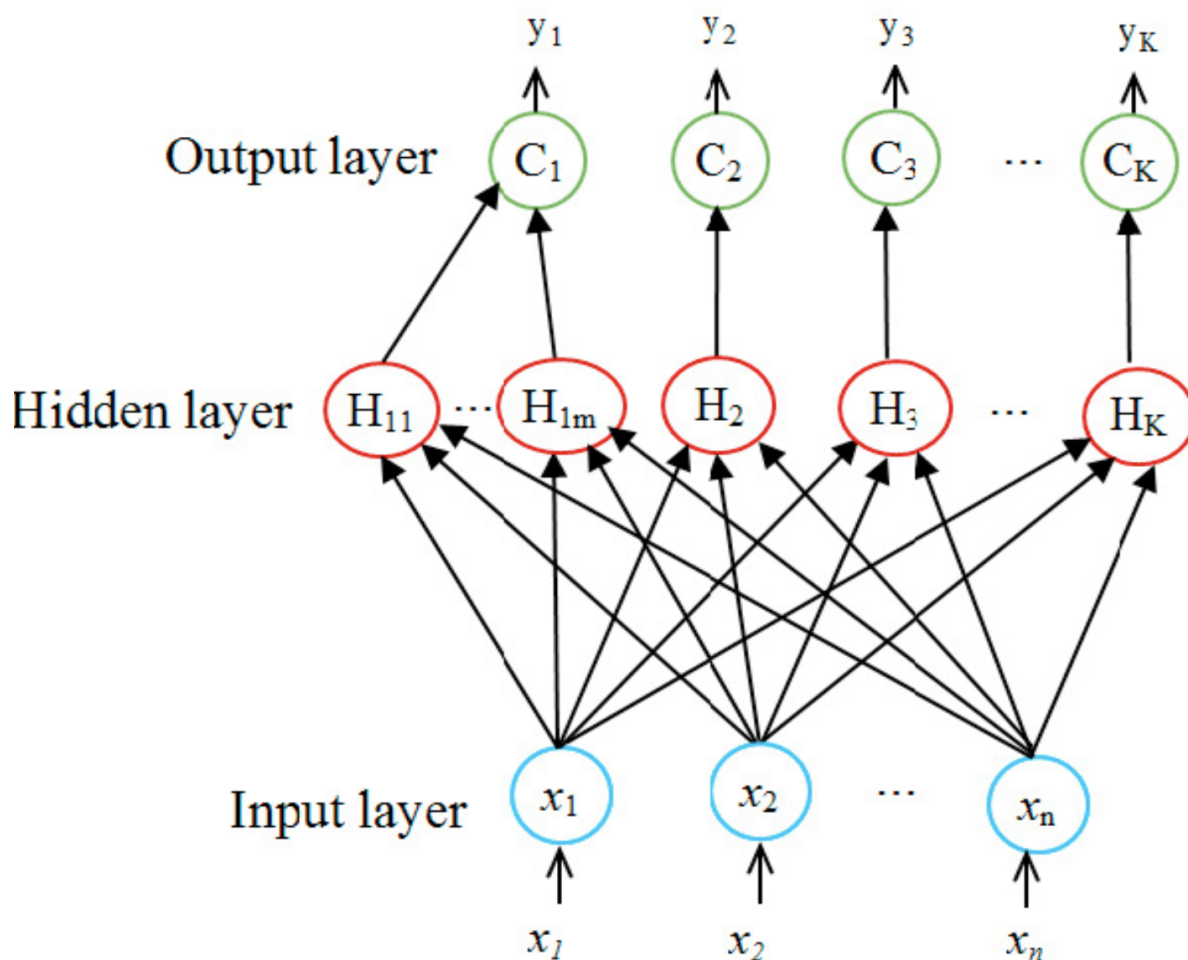


Рисунок 2 - Архитектура нечеткой нейронной сети

Выход каждого скрытого нейрона или FSH определяется нечеткой функцией принадлежности. Также, исходя из представленного рисунка, можно сказать, что существует частичная связь между выходным и скрытым слоями, поскольку FSH, созданные во время кластеризации для этого класса, подключены только к узлу класса. Выход узла класса определяется операцией нечеткого объединения.

Из рисунка видно, что класс 1 имеет m кластеров, в то время как другие классы имеют только один кластер, который может варьироваться в зависимости от приложения. Обучение предлагаемой архитектуры состоит из двух этапов:

а) Создание FSH на скрытом уровне FNN путем выполнения нечеткой кластеризации с максимальным счетчиком. В этом процессе выберите шаблон как центроид, который объединяет максимальное количество шаблонов своего собственного класса с использованием нечеткой функции принадлежности. После завершения процесса кластеризации выполните алгоритм сокращения для уменьшения кластеров с одним шаблоном.

Нечеткая функция принадлежности задается формулой

$$f(l, r_j) = \begin{cases} 1 & l \leq r_j \\ r_j/l & \text{otherwise} \end{cases}$$

где l - евклидово расстояние между входным шаблоном и центроидом j -го FSH, а r_j - радиус FSH.

б) Выходной уровень создается путем создания узлов класса, связанных с соответствующими FSH из скрытого уровня, которые создаются во время кластеризации для этого класса.

Нечеткая кластеризация с максимальным подсчетом и алгоритмом отсечения

Пусть V - обучающая выборка, состоящая из N обучающих пар. Шаблон ввода и желаемый результат представлены как $\{P_n, d_k\}$, где $n = 1, 2, \dots, N$ и $k = 1, 2, 3, 4, \dots, K$, для K классов. Пусть t_k - общее количество паттернов, принадлежащих классу s_k . Процесс кластеризации выполняется в два этапа: вначале формируется возможное количество кластеров, а затем используется алгоритм сокращения для оптимизации количества кластеров за счет уменьшения кластера с одним шаблоном.

В этом процессе кластеры или FSH формируются для каждого класса с учетом индивидуальных шаблонов классов наряду с другими шаблонами классов.

Шаги:

1) Пусть $X = X_1, X_2, \dots, X_{t_k}$ будет общим количеством шаблонов класса k и $Y = Y_1, Y_2, \dots, Y_N - t_k$ - образец другого класса из обучающей выборки V ;

2) Для каждого шаблона определяется максимальное количество шаблонов, которое он может объединить (Рисунок 3);

3) Шаблон, который объединяет максимальное количество шаблонов, будет выбран в качестве центроида, а расстояние между центроидом и самым дальним шаблоном в сгруппированных шаблонах будет радиусом. Если он объединяет только один узор, радиус оптимизируется до половины;

4) Вышеупомянутые шаги будут повторяться до тех пор, пока все шаблоны этого класса не будут сгруппированы;

5) Шаги с 1 по 4 будут повторяться для всех классов.

Algorithm 1 Count the number of clustered patterns

Input: X^k, Y **Output:** Count

```
for  $i := 1 \rightarrow t_k$  do
  for  $j := 1 \rightarrow N - t_k$  do
     $d(j) \leftarrow [\|X_i - Y_j\|]$ 
  end for
   $r \leftarrow \min(d)$ 
   $n \leftarrow 0$ 
  for  $q := 1 \rightarrow t_k$  do
     $l \leftarrow [\|X_i - X_q\|]$ 
    if  $l < r$  then
       $num \leftarrow num + 1$ 
    end if
  end for
   $\text{Count}(i) \leftarrow num$ 
end for
```

Рисунок 3 - Алгоритм подсчета количества сгруппированных паттернов

Алгоритм обрезки

Предлагаемый алгоритм удаляет отдельные кластеры шаблона, если эти кластеры замаскированы кластерами их собственных классов. Пусть $Q_k = Q_{k1}, Q_{k2}, \dots, Q_{kn}$, будет набором, представляющим n центроидов кластера, объединяющих более одного шаблона для класса k с радиусом, хранящимся в $R_k = r_{1k}, r_{2k}, \dots, r_{nk}$ соответственно.

Где $k = 1, 2, 3, 4, \dots, K$, для классов K . Пусть $S_k = S_{1k}, S_{2k}, \dots, S_{mk}$, будет набором, представляющим m центроидов кластера, объединяющих только один шаблон для класса k с радиусом, хранящимся в $W_k = w_{k1}, w_{k2}, \dots, w_{km}$ соответственно. Следующие шаги выполняются для обрезки кластеров по единому шаблону.

Шаги:

1) Вычислить значение принадлежности Sp_j , где $j = 1, 2, \dots, m$, по отношению к существующим кластерам в Q_k с радиусами в R_k для $k = 1, 2, \dots, K$ вдоль с кластерами в S_k , имеющими соответствующий радиус в W_k , для $k = 1, 2, \dots, K$ и $k \neq p$;

2) Если значение принадлежности для S_j , где $j = 1, 2, \dots, m$, является максимальным для любого из кластеров в Q для $k = p$, то отсечь кластер, удалив его из S_p и соответствующий радиус из W_k ;

3) Повторите шаги 1 и 2 для всех классов, т.е. $p \neq K$.

Тестирование FNN

1) Применить узор из набора данных к выходному слою;

2) Используя функцию принадлежности, вычислить значение принадлежности входного шаблона для каждого FSH в скрытом слое, учитывая его центроид и радиус;

3) Определить выходные данные для каждого класса, выполнив операцию объединения, учитывая значения принадлежности созданных FSH для этого класса;

4) Считается, что входной шаблон принадлежит к тому классу, который дает максимальное значение членства.

Результаты экспериментов

Оценим эффективность FNN с помощью эксперимента. Представим работу классификатора

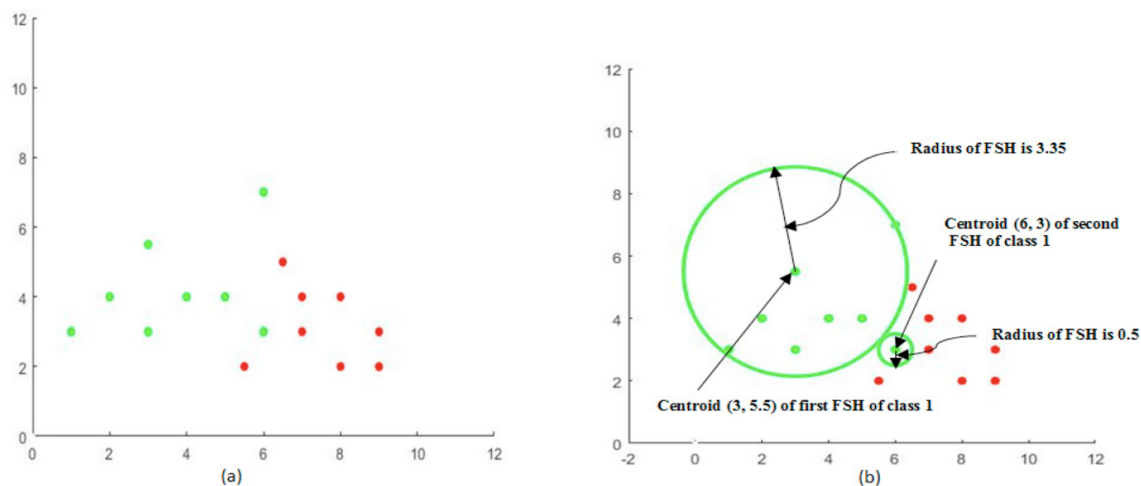


Рисунок 4 - а) диаграмма рассеяния б) FSH для класса 1

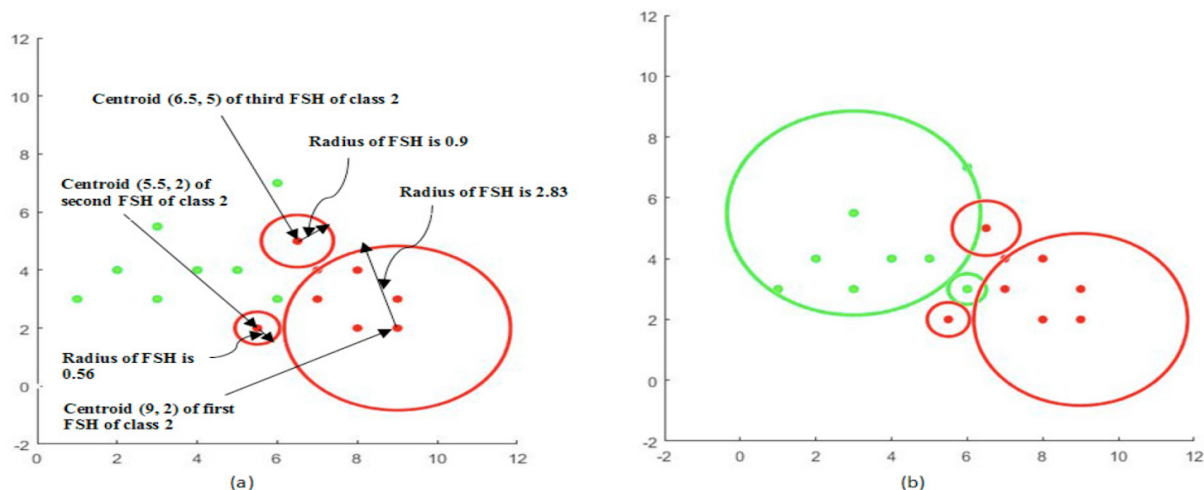


Рисунок 5 - а) FSH для класса 2 б) FSH для классов 1 и 2

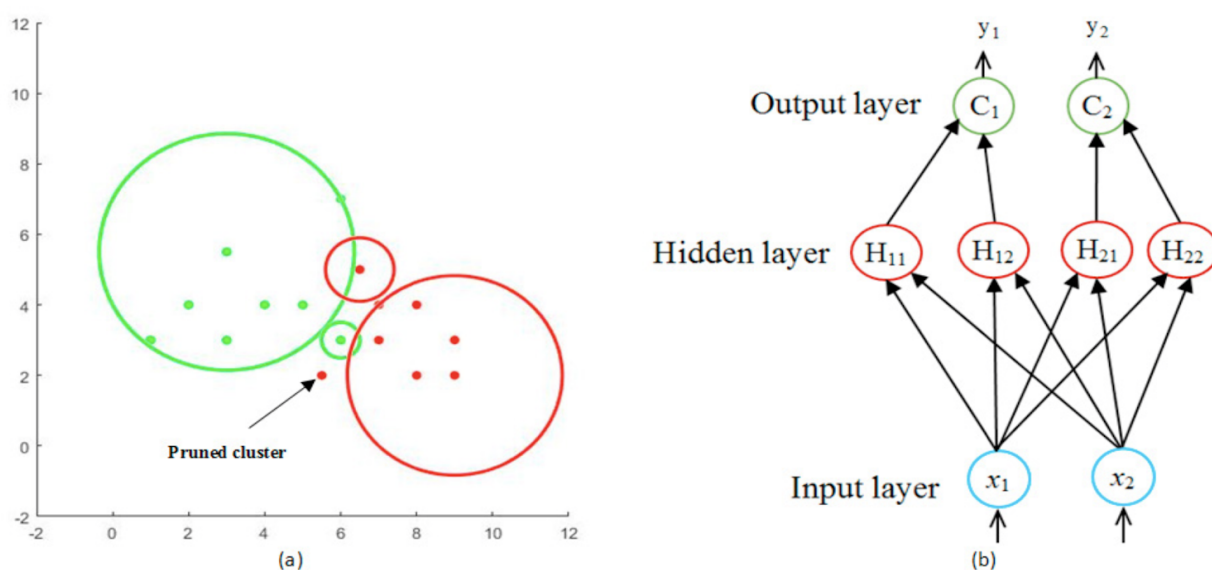


Рисунок 6 - а) конечный результат б) Структура нейронной сети

Также было проведено тестирование алгоритма для автоматизации процесса валидации уязвимостей, найденных автоматически сканерами безопасности.

Как один из способов подтверждения уязвимости, может использоваться сравнение с некоторыми аналогичными уязвимостями, о которых заранее известно, что они имеются в похожем веб-приложении или в том же веб-приложении, но более ранней версии. Для этого может быть сформирована база данных «эталонных» уязвимостей, содержащая характеристики и описание реальных, ранее найденных уязвимостей. Тогда эксперт-аналитик сможет сделать дополнительные выводы и подтвердить или опровергнуть новые кандидаты в уязвимости, опираясь на их сравнение с эталонами по некоторым правилам. Однако подобная рутинная работа также нуждается в дополнительных инструментах и процедурах, позволяющих выполнить такой анализ: сравнивать кандидаты с эталонными уязвимостями и отсеивать очевидные false positives.

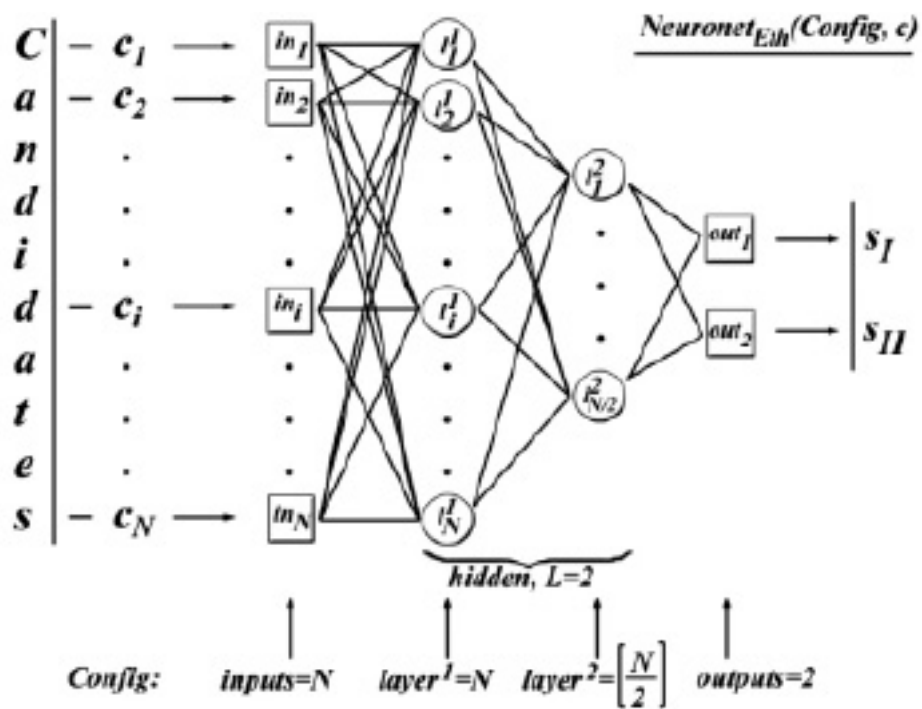


Рисунок 7 - структура нейронной сети

Этапы работы программы:

1) Этап обучения

- Инициализация объектов программы значениями от пользователя;
- Обработка входных данных и подготовка нейросети к обучению;
- Обучение нейронной сети на заданных эталонах.

2) Этап классификации

- Инициализация объектов программы значениями от пользователя;
- Обработка входных данных и подготовка нейросети к их анализу;
- Анализ нейронной сетью векторов признаков кандидатов.

Входные данные с векторами признаков эталонов и кандидатов задаются в виде обычных текстовых файлов с табуляцией в качестве разделителя значений.

Таблица 1 - эталоны

Input1	Input2	Input3	1st class_output	2nd_class_output
0.1	0.2	Min	Min	Max
0.2	0.3	Low	Min	Max
0.3	0.4	Med	Min	Max
0.4	0.5	Med	Max	Min
0.5	0.6	High	Max	Min
0.6	0.7	Max	Max	Min

Таблица 2 - кандидаты

Input1	Input2	Input3
0.12	0.32	Min

0.32	0.35	Low
0.54	0.57	Med
0.65	0.68	High
0.76	0.79	Max

В результате выполнения был получен репорт-файл:

```

Classification results for candidates vectors:

Header: [input1 input2 input3]      [1st_class_output 2nd_class_output]
-----
Input: ['0.12', '0.32', 'Min']      Output: ['Min', 'Max']
Input: ['0.32', '0.35', 'Low']      Output: ['Low', 'High']
Input: ['0.54', '0.57', 'Med']      Output: ['Max', 'Min']
Input: ['0.65', '0.68', 'High']     Output: ['Max', 'Min']
Input: ['0.76', '0.79', 'Max']      Output: ['Max', 'Min']

```

Рисунок 8 - репорт-файл

Если проанализировать данные из таблицы 2, то можно с высокой степенью уверенности утверждать, что человек-эксперт, опираясь только на данные из таблицы 1, выдал бы аналогичные результаты классификации.

Также был проведен второй эксперимент с другими входными данными.

Таблица 3 - кандидаты второго эксперимента

Input1	Input2	Input3
0.03	0.12	Low
0.11	0.21	Min
0.31	0.43	Med
0.38	0.47	Med
0.52	0.57	Med

В результате выполнения получили следующий репорт файл:

```

Classification results for candidates vectors:

Header: [input1 input2 input3]      [1st_class_output 2nd_class_output]
-----
Input: ['0.03', '0.12', 'Low']      Output: ['Min', 'Max']
Input: ['0.11', '0.21', 'Min']      Output: ['Min', 'Max']
Input: ['0.31', '0.43', 'Med']      Output: ['Min', 'Max']
Input: ['0.38', '0.47', 'Med']      Output: ['Max', 'Min']
Input: ['0.52', '0.57', 'Med']      Output: ['Max', 'Min']

```

Рисунок 9 - репорт файл второго эксперимента

Исходя из полученных данных в результате второго эксперимента, можно сделать аналогичный вывод, выводу из первого эксперимента.

Вывод

FSH в скрытом слое классификатора FNN создаются на основе нечеткой кластеризации. Процесс создания FSH выполняется в два этапа. Во-первых, FSH создаются путем выбора FSH, которые группируют больше шаблонов своего класса с использованием нечеткой функции принадлежности, а на втором этапе FSH, имеющие один шаблон, удаляются, поскольку эти FSH замаскированы FSH их собственного класса. Таким образом оптимизируются FSH FNN, за счет чего сокращается время вычислений для поиска.

Результаты предлагаемого классификатора FNN по сравнению лучше, чем другие классификаторы и лучше сопоставимы по отношению точности тестирования и создания FSH в скрытом слое. Хотя также существует перекрытие кластеров разных классов, как видно в приведенном выше эксперименте. Эффективность для набора обучающих данных будет 100%.

Таким образом, предлагаемый классификатор FNN является быстрым в отношении обучения и поиска и должен использоваться для классификации шаблонов.

Алгоритм может быть изменен с точки зрения выбора центроида, разработки новой нечеткой функции принадлежности и, наконец, алгоритма отсека. Параллельный процесс кластеризации может использоваться вместе с предварительной обработкой для удаления выбросов. Компромисс между количеством кластеров и обобщением можно свести к минимуму.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Carl G. Looney. (2009) "Fuzzy connectivity clustering with radial basis kernel functions" Fuzzy Sets and Systems 160:1868-1885;
2. Jacek M. Zurada (1994) "Introduction to Artificial Neural Systems", Jaico publication;
3. D. Lowe. (1988) "Multi variable functional interpolation and adaptive networks." Complex System 2:321-355;
4. S. N. Sivanandam, S. N. Deepa (2016) "Principles of soft computing", Wiley publication;
5. S. Rajasekaran, G. A. Vijayalakshmi pai (2012) "Neural networks, Fuzzy Logic and Genetic Algorithms synthesis and Applications", PHI Publication;
6. Nandedkar, Abhijeet V and Biswas, Prabir K. (2004) "A fuzzy min-max neural network classifier with compensatory neuron architecture."