# ПРЕДОБРАБОТКА ВХОДНОЙ ИНФОРМАЦИИ ДЛЯ ПОСТРОЕНИЯ И ОБУЧЕНИЯ ЭКСПЕРТНОЙ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ СИНДРОМА ВНЕЗАПНОЙ СМЕРТИ ГРУДНЫХ ДЕТЕЙ

**Васяева Т.А.** ✓
Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления
E-mail: vasyaeva_tanya@tr.dn.ua

***Abstract***
***Vasyaeva T. A. Prepare input data for building and learning the expert system to forecasting sudden infant death syndrome.*** *It is required to get useful information from set parameter (risk factors) and prepare data for learning the expert system. The problem is solved by means of neural network realizing nonlinear variant of the PCA. The methods of the coding and scaling data were considered too.*

**Актуальность**

Недавний прорыв в области технологий записи и хранения информации, основанный на использовании баз данных, привел к тому, что стали накапливаться громадные объемы данных, содержащие информацию о миллионах объектов, описанных с помощью сотен параметров. Из-за большого объема данных ручной подход для извлечения полезной информации из этих данных оказался практически неприменимым. В связи с этим возникла необходимость автоматизировать процесс извлечения полезной информации из больших объемов данных. Сложность проблемы состоит в обработке больших массивов, из которых требуется получить информацию, которая была бы практически полезной и по возможности более глубокой и существенной. Задача сводится не только к применению методов анализа закономерностей и извлечения полезной информации из наборов данных, но и к предварительной подготовке этих данных для анализа [1].

Предобработка предполагает представление данных в нужном виде. Следует отметить, что методы анализа данных, в основе которых лежит как статистический анализ данных, так и методы искусственного интеллекта, умеют обрабатывать только числовые значения, более того иногда значения только из определенного диапазона. Однако данные могут включать номинальные переменные, указывающие на принадлежность к одному из нескольких классов, даты, целочисленные значения и текстовые строки, а также числовые переменные, меняющиеся в самых различных диапазонах. Итак, основной этап предобработки данных включает в себя кодирование информации в виде, необходимом для конкретного метода анализа.

**Постановка задачи и ее реализация**

Уже несколько десятилетий внимание ученых всего мира продолжает привлекать одна из самых драматичных и до сих пор не выясненных проблем медицины – синдром внезапной смерти грудных детей (СВСГД) [2,3]. Интерес к этой проблеме не ослабевает, прежде всего, потому, что число жертв СВСГД не имеет тенденции к снижению. СВСГД является одной из ведущих причин смерти младенцев в развитых странах, ежегодно унося жизни нескольких тысяч детей грудного возраста. Украина не является исключением, в частности, Донецкий регион, где факторы окружающей среды, могут дополнительно негативно влиять на состояние имунноэндокринной системы беременных и новорожденных. Причины внезапной смерти младенцев в большинстве случаев остаются невыясненными. При этом имеется большое число факторов риска, которые могут в той или иной степени относится непосредственно к СВСГД. Необходимо так же отметить, что имеется немало случаев

СВСГД, не имеющих ни одного из факторов риска, в то время как младенцы с наличием даже нескольких из них не погибают внезапно.

Целью данной работы является получить полезную информацию из набора параметров (факторов риска) и подготовить данные для обучения экспертной системы, предназначенной для прогнозирования степени риска СВСГД.

В данной задаче, как и во многих других, нет никакой дополнительной информации о том, какие входные переменные действительно нужны для решения поставленной задачи, а именно, прогнозирования степени риска СВСГД. Анализ всех факторов риска вызывает существенные затруднения при построении и обучении экспертной системы.

Самый распространенный метод понижения размерности – это метод главных компонент (АГК). Метод состоит в следующем: к данным применяется линейное преобразование, при котором направлениям новых координатных осей соответствуют направления наибольшего разброса исходных данных. Как правило, уже первая компонента отражает большую часть информации, содержащейся в данных. Очень часто метод АГК выделяет из многомерных исходных данных совсем небольшое число компонент, сохраняя при этом структуру информации. Однако один из недостатков метода заключается в его линейности и, следовательно, в невозможности учесть некоторые важные характеристики структуры данных. Используется и нелинейный вариант АГК, который основан на применении автоассоциативных нейронных сетей [4].

Автоассоциативная сеть – это сеть, предназначенная для воспроизведения на выходе своих же входных данных. У такой сети число выходов совпадает с числом входов. Число скрытых элементов делается меньше числа входов/выходов, и это заставляет сеть «сжимать» информацию, представляя ее в меньшей размерности.

Трехслойная автоассоциативная сеть сначала линейно преобразует входные данные в меньшую размерность промежуточного слоя, а затем снова линейно разворачивает их в выходном слое. Такая сеть на самом деле реализует стандартный алгоритм АГК. Для того чтобы выполнить нелинейное понижение размерности, нужно использовать пятислойную сеть. Ее средний слой служит для уменьшения размерности, а соседние с ним слои, отделяющие его от входного и выходного слоев, выполняют нелинейные преобразования.

Для осуществления нелинейного понижения размерности с помощью автоассоциативной сети необходимо:

1. Сформировать обучающий набор данных для автоассоциативной сети.
2. Построить автоассоциативную сеть с пятью слоями. В среднем скрытом слое должно быть меньше элементов, чем во входном и выходном слоях. В двух оставшихся промежуточных слоях должно быть достаточно большое (и одинаковое) число элементов.
3. Обучить автоассоциативную сеть на подготовленном обучающем множестве.
4. Удалить два последних слоя автоассоциативной сети и в результате получается сеть понижающая размерность.

Для решаемой задачи в качестве обучающего набора использовались данные, полученные при обследовании 120 детей, которые умерли за период 1990-1999г. (71 мальчик и 49 девочек) в Донецкой области от СВСГД, и контрольная группа из 120 живых детей на первом году жизни, подобранных по принципу копий-пар в соответствии с возрастом, полом, годом и месяцем рождения, а также географическим распределением в рамках города. Полученный массив содержит различные типы данных. Это и числовые переменные, которые изменяются в различных диапазонах (Например: возраст на момент родов, рост и вес новорожденного); так и не числовые, которые несут информацию о перенесенных заболевания, вредных привычках и др.

Нейронные сети могут работать только с числовыми данными, лежащими в определенном ограниченном диапазоне. Это создает проблемы в тех случаях, когда данные

имеют нестандартный масштаб или являются нечисловыми. Соответственно, все такие переменные следует закодировать - перевести в численную форму, прежде чем начать собственно нейросетевую обработку. Сначала рассмотрим задачу – предварительная обработка данных нечислового характера [5].

Можно выделить два основных типа нечисловых переменных: упорядоченные (называемые также ординальными - от англ. order - порядок) и категориальные. В обоих случаях переменная относится к одному из дискретного набора классов $\{x_1, x_2,\ldots, x_n\}$. Но в первом случае эти классы упорядочены - их можно ранжировать: $\{x_1>x_2>\ldots>x_n\}$, тогда как во втором такая упорядоченность отсутствует. В качестве примера упорядоченных переменных можно привести сравнительные категории: плохо - хорошо - отлично, или медленно - быстро. Категориальные переменные просто обозначают один из классов, являются именами категорий. Например, это могут быть названия цветов: белый, синий, красный.

Ординальные переменные более близки к числовой форме, т.к. числовой ряд также упорядочен. Соответственно, для кодирования таких переменных остается лишь поставить в соответствие номерам категорий такие числовые значения, которые сохраняли бы существующую упорядоченность. Естественно, при этом имеется большая свобода выбора - любая монотонная функция от номера класса порождает свой способ кодирования. При этом мы должны стремиться к тому, чтобы максимизировать энтропию закодированных данных. При использовании сигмоидных функций активации все выходные значения лежат в конечном интервале - обычно [0, 1] или [-1, 1]. Из всех статистических функций распределения, определенных на конечном интервале, максимальной энтропией обладает равномерное распределение. Применительно к данному случаю это подразумевает, что кодирование переменных числовыми значениями должно приводить, по возможности, к равномерному заполнению единичного интервала закодированными примерами, захватывая при этом и этап нормировки. При таком способе все примеры будут нести примерно одинаковую информационную нагрузку. Учитывая выше сказанное для нашей задачи, ординальные переменные кодировались следующим образом: единичный отрезок разбивается на n отрезков (n равно числу классов), с длинами пропорциональными числу примеров каждого класса в обучающей выборке. Например, переменная, которая содержит информацию о курении во время данной беременности, может принимать значения: мало, средне, много. Пусть $x_1$ - мало, $x_2$ – средне и $x_3$ – много. Тогда $\Delta x_n=P_n/P$, где $P_n$ - число примеров данного класса n, а P - общее число примеров. Центр каждого такого отрезка будет являться численным значением для соответствующего ординального класса (Рис. 1).

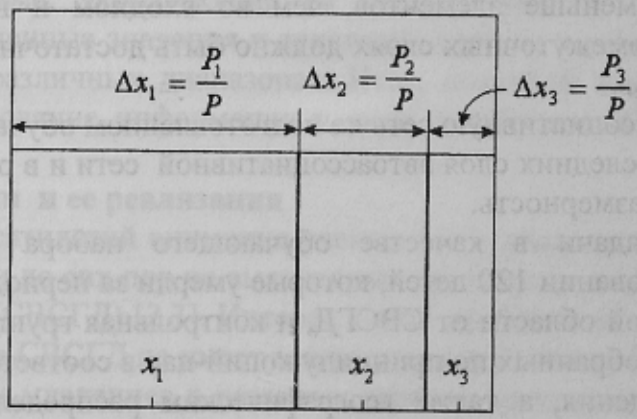


Рисунок 1 - Иллюстрация способа кодирования одной переменной ординального типа.

В принципе, категориальные переменные также можно закодировать описанным выше способом, пронумеровав их произвольным образом. Однако, такое навязывание несуществующей упорядоченности только затруднит решение задачи. Оптимальное

кодирование не должно искажать структуры соотношений между классами. Если классы не упорядочены, схема кодирования должна быть такой же.

Наиболее естественной выглядит и чаще всего используется на практике двоичное кодирование типа «n → n» когда имена n категорий кодируются значениями n бинарных нейронов, причем первая категория кодируется как (1,0,0,…,0), вторая, соответственно – (0,1,0,…,0) и т.д. вплоть до n-ной: (0,0,0,…,1). Легко убедиться, что в такой симметричной кодировке расстояния между всеми векторами-категориями равны.

Такое кодирование, однако, не будет оптимальным в случае, когда классы представлены существенно различающимся числом примеров. В этом случае, функция распределения значений переменной крайне неоднородна, что существенно снижает информативность этой переменной. Тогда имеет смысл использовать более компактный, но симметричный код n → m, когда имена n классов кодируются m-битным двоичным кодом. Причем, в новой кодировке активность кодирующих нейронов должна быть равномерна: иметь приблизительно одинаковое среднее по примерам значение активации. Это гарантирует одинаковую значимость весов, соответствующих различным нейронам.

В качестве примера рассмотрим фактор TORCH инфекция, значения которого могут быть такими: хламидии, микоплазма, бак. вагиноз, герпес, краснуха, токсоплазмоз. При этом значение краснуха представлено гораздо большим числом примеров, чем остальные. Простое кодирование n → n привело бы к тому, что первый нейрон активировался бы гораздо чаще остальных. Соответственно, веса оставшихся нейронов имели бы меньше возможностей для обучения. Это можно избежать, закодировав шесть классов тремя бинарными нейронами следующим образом: хламидии –{0,0,0}, микоплазма – {0,0,1}, бак. вагиноз – {0,1,0}, герпес – {0,1,1}, краснуха – {1,0,0}, токсоплазмоз – {1,0,1}, что обеспечивает равномерную "загрузку" кодирующих нейронов.

Теперь рассмотрим работу с числовыми значениями, они могут быть совершенно разнородными величинами. Очевидно, что результаты нейросетевого моделирования не должны зависеть от единиц измерения этих величин. А именно, чтобы сеть трактовала эти значения единообразно, все входные и выходные величины должны быть приведены к единому - единичному - масштабу. Кроме того, для повышения скорости и качества обучения полезно провести дополнительную предобработку данных, выравнивающую распределение значений еще до этапа обучения.

Приведение к единому масштабу обеспечивается нормировкой каждой переменной на диапазон разброса ее значений. В простейшем варианте это – линейное преобразование:

$$\tilde{x}_i = \frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}} \qquad (1)$$

в единичный отрезок: $\tilde{x}_i \in [0,1]$.

Линейная нормировка оптимальна, когда значения переменной $x_i$ плотно заполняют определенный интервал. Но подобный «прямолинейный» подход применим далеко не всегда. Так, если в данных имеются относительно редкие выбросы, намного превышающие типичный разброс, а именно эти выбросы определят согласно предыдущей формуле масштаб нормировки, это приведет к тому, что основная масса значений нормированной переменной $\tilde{x}_i$ сосредоточится вблизи нуля $|\tilde{x}_i| \ll 1$. Гораздо надежнее, поэтому, ориентироваться при нормировке не на экстремальные значения, а на типичные, т.е. статистические характеристики данных, такие как среднее и дисперсия.

$$\tilde{x}_i = \frac{x_i - \bar{x}_i}{\sigma_i}, \qquad (2)$$

где $\bar{x}_i \equiv \frac{1}{P}\sum_{\alpha=1}^{P} x_i^{\alpha}$, $\sigma_i^2 \equiv \frac{1}{P-1}\sum_{\alpha=1}^{P}(x_i^{\alpha} - \bar{x}_i)^2$ (3), (4)

В этом случае основная масса данных будет иметь единичный масштаб, т.е. типичные значения всех переменных будут сравнимы (рис. 2).

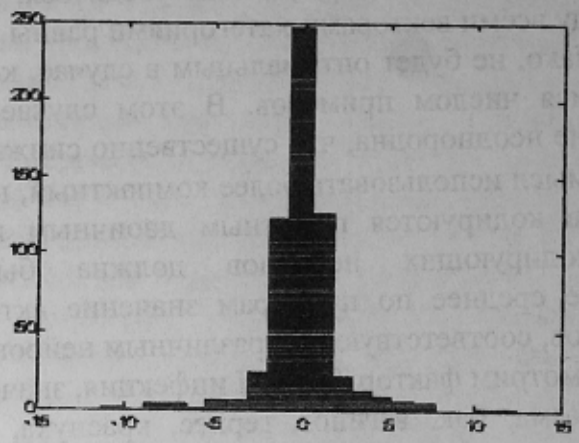

Рисунок 2 - Гистограмма значений переменной при наличии редких, но больших по амплитуде отклонений от среднего.

Теперь нормированные величины не принадлежат гарантированно сдиничному интервалу, более того, максимальный разброс значений $\tilde{x}_i$ заранее не известен. Для входных данных это может быть и не важно, но выходные переменные будут использоваться в качестве эталонов для выходных нейронов. В случае, если выходные нейроны – сигмоидные, они могут принимать значения лишь в единичном диапазоне. Чтобы установить соответствие между обучающей выборкой и нейросетью в этом случае необходимо ограничить диапазон изменения переменных.

Линейное преобразование, представленное выше, не способно отнормировать основную массу данных и одновременно ограничить диапазон возможных значений этих данных. Естественный выход из этой ситуации – использовать для предобработки данных функцию активации тех же нейронов [5]. Например, нелинейное преобразование

$$\tilde{x}_i = f\left(\frac{x_i - \bar{x}_i}{\sigma_i}\right), \quad f(x) = \frac{1}{1 + e^{-x}} \qquad (5), (6)$$

нормирует основную массу данных одновременно гарантируя что $\tilde{x}_i \in [0,1]$ (рис. 3).

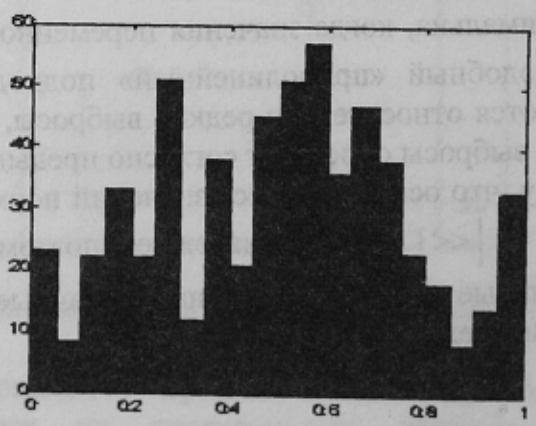

Рисунок 3 - Нелинейная нормировка, использующая логистическую функцию активации $f(a) = (1 + e^{-a})^{-1}$

Как видно из приведенного выше рисунка, распределение значений после такого нелинейного преобразования гораздо ближе к равномерному.

Для нашего обучающего массива линейной нормировки было достаточно, так как данные не содержат редких выбросов, которые бы существенно превышали типичный разброс.

Теперь непосредственно о понижении размерности уже закодированных данных. Как говорилось ранее для осуществления нелинейного понижения размерности используется автоассоциативная сеть с пятью слоями.

Для построения и обучения автоассоциативной сети написана программа в среде визуального объектно-ориентированного программирования C++ Builder 6. Данная программа позволяет задавать архитектуру сети, а именно определять количество нейронов на входном/выходном слоях, в среднем скрытом и двух промежуточных слоях многослойного персептрона. Выбирать активационную функцию для каждого слоя, здесь предусмотрены три наиболее распространенных варианта: линейная, гладкая ступенчатая, гиперболический тангенс. Предусмотрена возможность использования пре- и пост-процессинга входных данных. Учитывая то, что для наших данных использовалась нормировка, нет необходимости в выполнении пре- и пост-процессинга. Вид окна для задания выше указанных параметров представлен на рисунке 4.

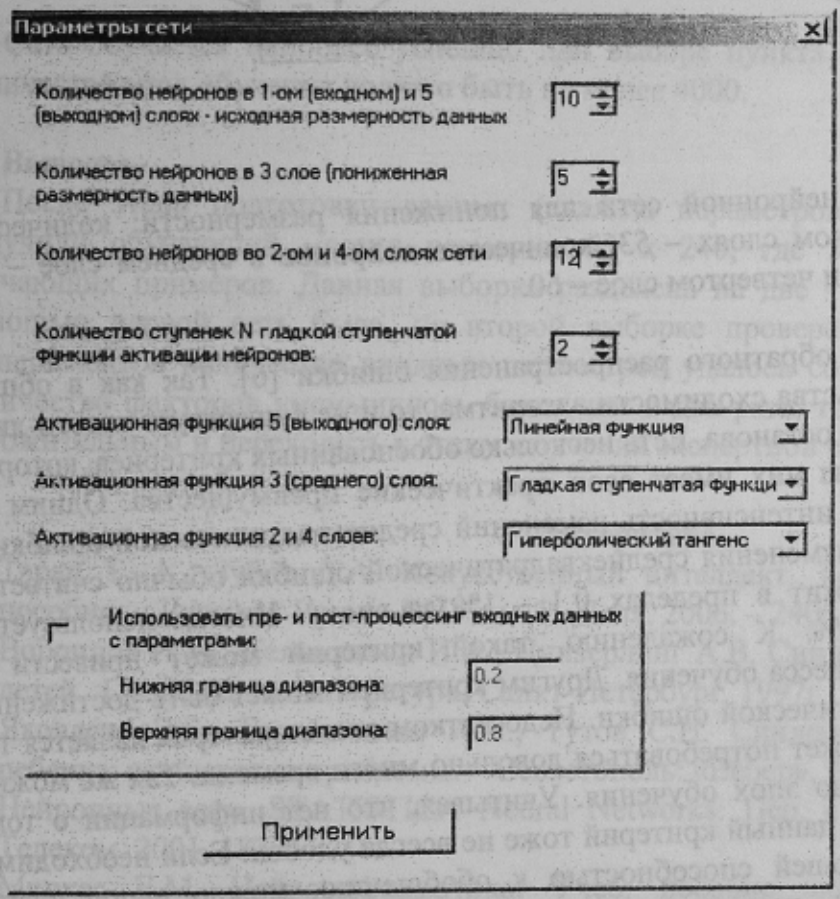


Рисунок 4 - Создание новой автоассоциативной сети.

В результате проведенных экспериментов, оказалось, что наилучшие результаты достигаются при следующих параметрах сети: количество нейронов во входном и выходном слоях – 53 (обусловлено входными факторами, которых 53); количество нейронов в среднем слое – 25 (пониженная размерность входных факторов); количество нейронов во втором и четвертом слое – 60. Функции активации для 5,4 и 2 слоев – гиперболический тангенс, для 3 слоя – гладкая ступенчатая. Архитектура сети представлена на рисунке 5.

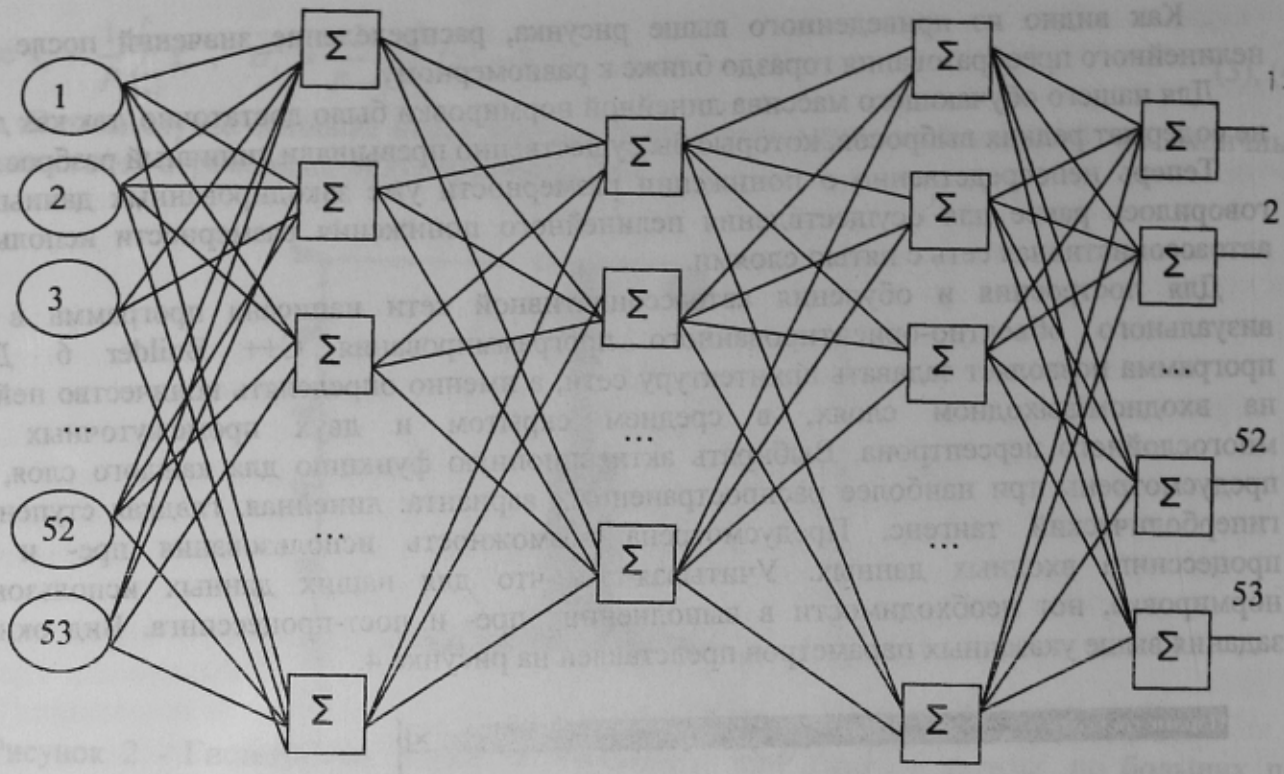


Рисунок 5 - Архитектура нейронной сети для понижения размерности: количество нейронов во входном и выходном слоях – 53; количество нейронов в среднем слое – 25; количество нейронов во втором и четвертом слое – 60.

Сеть обучается алгоритмом обратного распространения ошибки [6]. Так как в общем случае не существует доказательства сходимости алгоритма, то и не существует какого-либо четкого определенного критерия останова. Есть несколько обоснованных критериев, которые можно использовать. Каждый из них имеет свои практические преимущества. Одним из таких критериев является малая интенсивность изменений среднеквадратической ошибки в течение эпохи. Интенсивность изменения среднеквадратической ошибки обычно считается достаточно малой, если она лежит в пределах 0,1 – 1% за эпоху. Иногда используется уменьшенное значение – 0,01%. К сожалению, такой критерий может привести к преждевременной остановке процесса обучения. Другим критерием может быть достижение целевого значения среднеквадратической ошибки. Недостатком этого критерия является то, что для сходимости обучения может потребоваться довольно много времени. Так же можно задавать определенное количество эпох обучения. Учитывая, что нет информации о том, сколько их может потребоваться, данный критерий тоже не всегда удобен. Если необходимо создать сеть, обладающую хорошей способностью к обобщению, можно использовать перекрестную проверку [6]. В данном случае обучающее множество делится на подмножество для обучения и проверочное подмножество, которое используется для проверки эффективности различных моделей сети, из которых необходимо выбрать лучшую. В разработанной программе предусмотрены все перечисленные выше критерии останова (рисунок 6).

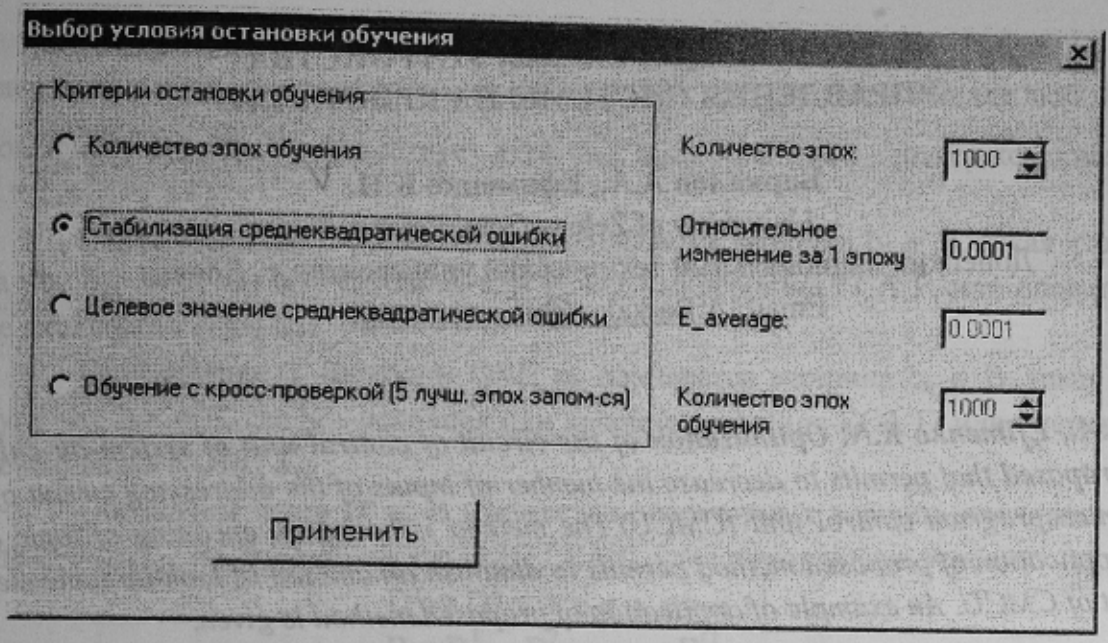


Рисунок 6 -Выбор условия остановки обучения.

Сеть обучается наиболее успешно при выборе пункта «Обучение с кросс-проверкой», количество эпох обучения должно быть не менее 4000.

**Выводы**

После этапа подготовки данных (анализа параметров, кодирования и нормировки) получили обучающий массив размером 53 х 240, где 53 входных параметров и 240 обучающих примеров. Данная выборка разделена на две - по 120 примеров в каждой. С помощью первой сеть была, на второй выборке проверена. В результате проведенных экспериментов количество входных параметров удалось сократить до 25. Таким образом, количество факторов уменьшилось более чем в два раза, то есть результат можно считать положительным и переходить к этапу обучения экспертной системы.

**Литература**

1. Таран Т. А., Зубов Д.А. Искусственный интеллект. Теория и приложения. Учебное пособие. – Луганск: Изд-во ВНУ им. В. Даля, 2006. – 240с
2. Воронцов И.М., Кельмансон И.А., Цинзерлинг А.В. Синдром внезапной смерти грудных детей. Специальная литература, Санкт-Петербург, 1997г.
3. Яковлева Э.Б., Герасименко А.И., Тутов С.Н. Синдром внезапной смерти грудного ребенка: акушерские проблемы. – Севастополь: «Вебер», 2006. – 128 с.
4. Нейронные сети. STATISTICA Neural Networks: Пер. с англ. – М.: Горячая линия – Телеком. 2001. 182 с.
5. Миркес Е.М. Нейроинформатика: Учеб. пособие для студентов / Е.М. Миркес. Красноярск: ИПЦ КГТУ, 2002, 347 с. Рис. 58, табл. 59, библиогр. 379 наименований. http://softcraft.ru/neuro/ni/p07.shtml
6. Саймон Хайкин Нейронные сети: полный курс, 2-е издание. : Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1104 с.

.