Министерство образования и науки Российской Федерации Федеральное агентство по образованию

Амурский гуманитарно-педагогический государственный университет

Физико-математический факультет

Кафедра информатики

**ЛАБОРАТОРНАЯ РАБОТА №2**

**по дисциплине «Искусственные нейронные сети»**

**на тему «Нейронные сети с радиальными базисными функциями»**

2007

**Введение**

**Цель лабораторной работы:** освоить основные принципы решения задачи нейронных сетей с радиальными базисными функциями.

**Задание:** Используя встроенные функции пакета нейронных сетей математической среды Matlab, построить нейронную сеть с радиальными базисными функциями.

**1 Теоретические сведения**

**Сети РБФ имеют ряд преимуществ** перед рассмотренными многослойными сетями прямого распространения. Во-первых, они моделируют произвольную нелинейную функцию с помощью всего одного промежуточного слоя, тем самым, избавляя разработчика от необходимости решать вопрос о числе слоев. Во-вторых, параметры линейной комбинации в выходном слое можно полностью оптимизировать с помощью хорошо известных методов линейной оптимизации, которые работают быстро и не испытывают трудностей с локальными минимумами, так мешающими при обучении с использованием алгоритма обратного распространения ошибки. Поэтому сеть РБФ обучается очень быстро - на порядок быстрее, чем с использованием алгоритма ОР (обратного распространения).

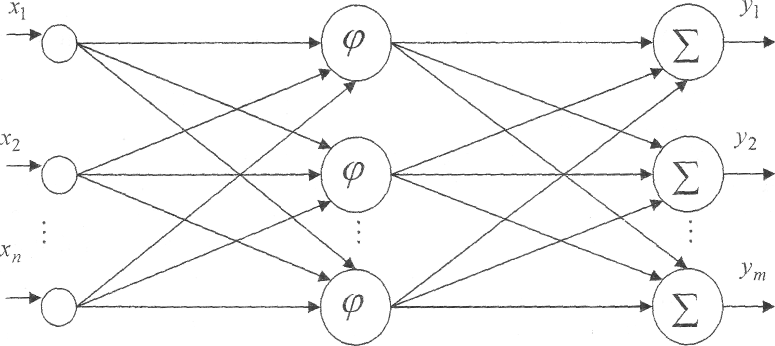
**Недостатки сетей РБФ:** данные сети обладают плохими экстраполирующими свойствами и получаются весьма громоздкими при большой размерности вектора входов.

На рис. 1 представлена структурная схема нейронной сети с радиальными базисными функциями.

Нейронная сеть радиальных базисных функций содержит в наиболее простой форме три слоя: обычный входной слой, выполняющий распределение данных образца для первого слоя весов; слой скрытых нейронов с радиально симметричной активационной функцией, каждый j -й из которых предназначен для хранения отдельного эталонного вектора в виде вектора весов wj(h); выходной слой

Для построения сети РБФ необходимо выполнение следующих условий.

Во-первых, наличие эталонов, представленных в виде весовых векторов нейронов скрытого слоя. Во-вторых, наличие способа измерения расстояния входного вектора от эталона. Обычно это стандартное евклидово расстояние. В-третьих, специальная функция активации нейронов скрытого слоя, задающая выбранный способ измерения расстояния. Обычно используется функция Гаусса, существенно усиливающая малую разницу между входным и эталонным векторами. Выходной сигнал эталонного нейрона скрытого слоя aj- это функция (гауссиан) только от расстояния pj между входным и эталонным векторами.



**Рис. 1. Сеть с радиальными базисными функциями**

Таким образом, выходной сигнал шаблонного нейрона - это функция только от расстояния между входным вектором х и сохраненным центром w v

Обучение слоя образцов-нейронов сети подразумевает предварительное проведение кластеризации для нахождения эталонных векторов и определенных эвристик для определения значений -.

Нейроны скрытого слоя соединены по полносвязной схеме с нейронами выходного слоя, которые осуществляют взвешенное суммирование

Для нахождения значения весов w от нейронов скрытого к выходному слою используется линейная регрессия.

В общем случае активационные функции нейронов скрытого слоя могут отражать законы распределения случайных величин (вероятностные нейронные сети) либо характеризовать различные аналитические зависимости между переменными (регрессионные нейронные сети).

К недостаткам сетей РБФ можно отнести то, что заранее должно быть известно число эталонов, а также эвристики для построения активационных функций нейронов скрытого слоя.

В моделях РБФ могут быть использованы различные способы измерения расстояния между векторами, а также функции активации нейронов скрытого слоя.

Радиальная, базисная сеть общего вида – это двухслойная нейронная сеть с **R** входами, каждый из которых может состоять из нескольких элементов. Передаточной функцией нейронов входного слоя является колоколообразная симметричная функция следующего вида:

Эта функция имеет максимум, равный **1**, при **n = 0** и плавно убывает при увеличении **n**, достигая значения **0.5** при **n = ±0.833**. Передаточной функцией нейронов выходного слоя является линейная функция **perelin**.

Функция взвешивания для входного слоя вычисляет евклидово расстояние между каждой строкой матрицы весов и каждым столбцом матрицы входов:

Затем эта величина умножается на смещение нейрона и поступает на вход передаточной функции, так что

**a{i} = radbas(net.prod(dist(net.IW{1, 1}, p).net.b{i})).**

Для нейронов выходного слоя функцией взвешивания является скалярное произведение **dotprod**, а функцией накопления – функция суммирования взвешенных входов и взвешенного смещения **netsum.**

Для того чтобы понять поведение радиальной базисной сети общего вида, необходимо проследить прохождение вектора входа **p**. При задании значений элементам вектора входа каждый нейрон входного слоя выдает значение в соответствии с тем, как близок вектор входа к вектору весов каждого нейрона. Таким образом, нейроны с векторами весов, значительно отличающимися с вектором входа **p**, будут иметь выходы, близкие к **0**, и их влияние на выходы линейных нейронов выходного слоя будет незначительное. Напротив, входной нейрон, веса которого близки к вектору **p**, выдаст значение, близкое к единице.

Для построения радиальных базисных сетей общего вида и автоматической настройки весов и смещений используются две функции **newrbe** и **newrb**. Первая позволяет построить радиальную базисную сеть с нулевой ошибкой, вторая позволяет управлять количеством нейронов входного слоя. Эти функции имеют следующие параметры:

**net = newrbe(P, T, SPREAD),**

**net = newrb(P, T, GOAL, SPREAD),**

где **P** – массив размера **RxQ** входных векторов, причем **R** – число элементов вектора входа, а **Q** – число векторов в последовательности;

**T** – массив размера **SxQ** из **Q** векторов цепи и **S** классов;

**SPREAD** – параметр влияния, определяющий крутизну функции **radbas**, значение по умолчания которого равно единице;

**GOAL** – средняя квадратичная ошибка, при этом значение по умолчанию равно **0.0**.

Параметр влияния **SPREAD** существенно влияет на качество аппроксимации функции: чем больше его значение, тем более гладкой будет аппроксимация. Слишком большое его значение приведет к тому, что для получения гладкой аппроксимации быстро изменяющейся функции потребуется большое количество нейронов: слишком малое значение параметра **SPREAD** потребует большего количества нейронов для аппроксимации гладкой функции. Обычно параметр влияния **SPREAD** выбирается большим, чем шаг разбиения интервала задания обучающей последовательности, но меньшим размера самого интервала.

Функция **newrbe** устанавливает веса первого слоя равным **P.**, а смещения – равными **0.8326/ SPREAD**, в результате радиальная базисная функция пересекает значение **0.5** при значениях евклидового расстояния **±SPREAD**. Веса второго слоя **LW{2,1}** и смещения **b{2}** определяются путем моделирования выходов первого слоя **A{1}** и последующего решения системы линейных уравнений:

**[LW{2,1} b{2}]\*[A{1}; ones] = T.**

Функция **newrb** формирует сеть следующим образом. Изначально первый слой не имеет нейронов. Сеть моделируется и определяется вектор входа с самой большой погрешностью, добавляется нейрон с функцией активации **radbas** и весами, равными вектору входа, затем вычисляются весовые коэффициенты линейного слоя, чтобы не превысить средней допустимой квадратичной ошибки.

**2 Методика выполнения лабораторной работы**

**Задача.** Используя встроенные функции пакета нейронных сетей математической среды Matlab, построить нейронную сеть с радиальными базисными функциями.

P = zeros(1,20);

for i = 1:20

P(i) = i\*0.1;

end

T=[-2.09 -1.66 -1.06 -0.65 -0.25 0.10 0.56 0.85 1.07 1.16 1.52 1.63 1.78 2.07 2.09 2.10 2.12 2.17 2.21 2.31]

[net,tr] = newrb(P,T);

y = sim(net,P);

figure (1);

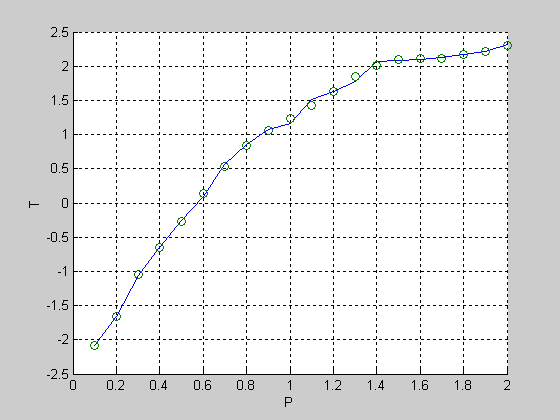
hold on;

xlabel ('P');

ylabel ('T');

plot(P,T,P,y,'o'),grid;

Работа сети представлена на рис.1



**Формы обучения НС.**

Существует три основные парадигмы (формы) обучения нейроных сетей:

- обучение с учителем

- обучение с критиком - усиленное, подкрепленное обучение;

- обучение без учителя) — самоорганизующееся обучение, самообучение.

В первом случае обучение осуществляется под наблюдением внешнего «учителя». Нейронной сети предъявляются значения как входных, так и желательных выходных сигналов, и она по некоторому внутреннему алгоритму подстраивает веса своих синаптических связей.

Во втором случае обучение включает использование «критика», с помощью которого производится обучение на основе метода проб и ошибок.

В третьем случае выходы нейронной сети формируются самостоятельно, а веса и смещения изменяются по алгоритму, учитывающему только входные и производные от них сигналы. Здесь за основу взяты принципы самоорганизации нервных клеток. Для обучения без учителя не нужно знания требуемых ответов на каждый пример обучающей выборки. В этом случае происходит распределение образцов по категориям (кластерам) в соответствии с внутренней структурой данных или степенью корреляции между образцами.

Рассматривают также и смешанное обучение, при котором весовые коэффициенты одной группы нейронов настраиваются посредством обучения с учителем, а другой группы - на основе самообучения.

Основные правила обучения нейронных сетей

Известны четыре основных правила обучения, обусловленные связанными с ними архитектурами сетей: коррекция ошибки, правило Больц-мана, правило Хебба и метод соревнования.

1) Коррекция ошибки

Для каждого входного примера задан требуемый выход и, который может не совпадать с реальным у. Правило обучения при коррекции по ошибке состоит в использовании разницы (с? - у) для изменения весов, с целью уменьшения ошибки рассогласования. Обучение производится только в случае ошибочного результата. Известны многочисленные модификации этого правила обучения.

2) Правило Больцмана

Правило Больцмана является стохастическим правилом обучения, обусловленным аналогией с термодинамическими принципами. В результате его выполнения осуществляется настройка весовых коэффициентов нейронов в соответствии с требуемым распределением вероятностей. Обучение правилу Больцмана может рассматриваться как отдельный случай коррекции по ошибке, в котором под ошибкой понимается расхождение корреляций состояний в двух режимах.

3) Правило Хебба

Правило Хебба является самым известным алгоритмом обучения нейронных сетей, суть которого заключается в следующем: если нейроны с обеих сторон синапса возбуждаются одновременно и регулярно, то сила синаптической связи возрастает. Важной особенностью является то, что изменение синаптического веса зависит только от активности связанных этим синапсом нейронов. Предложено большое количество разновидностей этого правила, различающихся особенностями модификации синап-тических весов.

4) Метод соревнования

В отличие от правила Хебба, в котором множество выходных нейронов могут возбуждаться одновременно, здесь выходные нейроны соревнуются между собой. И выходной нейрон с максимальным значением взвешенной суммы является «победителем» («победитель забирает все»). Выходы же остальных выходных нейронов устанавливаются в неактивное состояние. При обучении модифицируются только веса нейрона - «победителя» в сторону увеличения близости к данному входному примеру.

В состав пакета ППП Neural Network Toolbox входит М-функция **hardlim**, реализующая функцию активации с жесткими ограничениями.

Линейная функция активации **purelin**. Эта функция описывается соотношением, **а = purelin(n) = n**

Логистическая функция активации **logsig**. Эта функция описывается соотношением, **а = logsig(n) = 1/(1 + ехр(-n)).** Она принадлежит к классу сигмоидальных функций, и ее аргумент может принимать любое значение в диапазоне от - до + , а выход изменяется в диапазоне от 0 до 1. В пакете

ППП Neural Network Toolbox она представлена М-функцией **logsig.**

Благодаря свойству дифференцируемости эта функция часто используется в сетях с обучением на основе метода обратного распространения ошибки.