Оптимальное проектирование и обучение нейронных сетей

Теорема существования

Из предыдущего изложения следуют два важных вывода.

1. В жизни встречается множество практически важных задач, решить которые можно методом математического моделирования, т. е. путем построения некоторой сложной функции, осуществляющей преобразование вектора входных параметров ***X*** в вектор выходных параметров ***D***.

2. Универсальным инструментом построения такой функции являются нейросетевые технологии.

Естественно, возникают вопросы: всегда ли можно построить нейронную сеть, выполняющую преобразование, заданное любым множеством обучающих примеров, и каким требованиям должна удовлетворять эта нейронная сеть?

С одной из таких трудностей, названной «Проблемой исключающего ИЛИ», мы уже столкнулись при изучении и теперь знаем, что персептрон должен иметь скрытый слой нейронов, но осталось два вопроса.

1. Всегда ли можно спроектировать и обучить многослойный персептрон, обеспечивающий решение любой задачи?

2. Каким образом лучше задавать количество внутренних нейронных слоев и количество нейронов в них? Может быть, как в мозге,—10^{11} нейронов? Может, чем их будет больше, тем лучше?

Ответы на эти вопросы мы выясним, познакомившись с теоретической базой нейронных сетей. Важнейшее место в теории нейронных сетей занимает теорема ***Арнольда—Колмогорова—Хехт-Нильсена***, доказательство которой достаточно сложно и поэтому в нашем курсе не рассматривается.

С физической точки зрения персептрон—это устройство, моделирующее человеческий мозг на структурном уровне. Однако, анализируя формулы, по которым он преобразует сигналы, можно заметить, что с математической точки зрения персептрон—это всего лишь аппроксиматор, заменяющий функцию многих аргументов суммой функций, каждая из которых зависит только от одного аргумента.

Аппроксимация – научный метод, состоящий в замене одних объектов другими, в каком-то смысле близкими к исходным, но более простыми. Аппроксимация позволяет исследовать числовые характеристики и качественные свойства объекта, сводя задачу к изучению более простых или более удобных объектов (например, таких, характеристики которых легко вычисляются или свойства которых уже известны)

Вопрос о том, всегда ли можно любую функцию многих аргументов представить в виде суммы функций меньшего количества аргументов, интересовал математиков на протяжении нескольких столетий.

В 1900 г. на Всемирном математическом конгрессе в Париже знаменитый немецкий математик Давид Гильберт сформулировал 23 проблемы, которые он предложил решать математикам начинающегося XX в. Одна из этих проблем (под номером 13) как раз и декларировала невозможность такого представления.

Таким образом, приговор новой области искусственного интеллекта был вынесен за полвека до ее появления. Получалось, что персептрон, сколько бы нейронов он ни имел, не всегда мог построить нужную математическую функцию.

Сомнения относительно возможностей персептронов развеяли советские математики—академики В. И. Арнольд и А. Н. Колмогоров. Им удалось доказать, что любая непрерывная функция n аргументов $f(x_1, x_2, \ldots, x_n)$ всегда может быть представлена в виде суммы непрерывных функций одного аргумента:

$$f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n).$$

Тем самым гипотеза Гильберта была опровергнута, а нейроинформатике был открыт «зеленый свет».

В 1987–1991 гг. профессор Калифорнийского университета (США) Р. Хехт-Нильсен переработал теорему Арнольда—Комогорова применительно к нейронным сетям. Он доказал, что *для любого множества различающихся между собой пар векторов* $\boldsymbol{X}q$ и $\boldsymbol{D}q$ *произвольной размерности существует двухслойный персептрон с сигмоидными активационными функциями и с конечным числом*

нейронов, который для каждого входного вектора ***X****q формирует соответствующий ему выходной вектор* ***D****q*.

Таким образом, была доказана принципиальная возможность построения нейронной сети, выполняющей преобразование, заданное *любым* множеством различающихся между собой обучающих примеров, и установлено, что такой универсальной нейронной сетью является двухслойный персептрон – персептрон с одним скрытым слоем, причем активационные функции его нейронов должны быть сигмоидными.

Теорема Арнольда—Колмогорова—Хехт-Нильсена имеет очень важное для практики следствие в виде формулы, с помощью которой можно определять необходимое количество синаптических весов нейронной сети:

$$\frac{N_y Q}{1+\log_2(Q)} \leqslant N_w \leqslant N_y\left(\frac{Q}{N_x}+1\right)\left(N_x+N_y+1\right)+N_y,$$

где *Nx* —количество нейронов входного слоя; *Ny* —количество нейронов выходного слоя; *Q*—количество элементов множества обучающих примеров, т. е. количество пар входных и выходных векторов ***X****q* и ***D****q*; *Nw* —необходимое число синаптических связей.

Оценив с помощью этой формулы необходимое число синаптических связей *Nw*, можно рассчитать и необходимое количество нейронов в скрытых слоях. Например, количество нейронов скрытого слоя двухслойного персептрона будет равно:

$$N=\frac{N_w}{N_x+N_y}$$

Последняя формула становится очевидной, если ее левую и правую части умножить на (*Nx* + *Ny*) и нарисовать схему двухслойного персептрона (т. е. персептрона с одним скрытым слоем).

Практические рекомендации по проектированию персептронов

Как следует из теорем Арнольда—Колмогорова—Хехт-Нильсена, для построения нейросетевой модели любого сколь угодно сложного объекта

достаточно использовать персептрон с одним скрытым слоем сигмоидных нейронов. Однако в практических реализациях персептронов оптимальное количество как слоев, так и нейронов в каждом из них нередко отличается от теоретических. К тому же, иногда бывает целесообразно использовать персептроны с большим количеством скрытых слоев.

Строгой теории выбора оптимального количества скрытых слоев и нейронов в скрытых слоях пока не существует. На практике чаще всего используются персептроны, имеющие один или два скрытых слоя, причем количество нейронов в скрытых слоях обычно колеблется от $Nx/2$ до $3Nx$.

Рассмотрим факторы, от которых зависит успешность обучения нейронной сети правильному решению задачи. В первую очередь, сеть должна быть достаточно гибкой, чтобы научиться правильно решать все примеры обучающей выборки. Поэтому в нейронной сети должно быть достаточное количество нейронов и связей.

На основании обучающей выборки достаточно сложно определить, сколько слоев и нейронов необходимо. Поэтому поступают обычно так. Обучают сеть со структурой, предлагаемой программой-нейроимитатором по умолчанию, а в дальнейшем, если сеть не может обучиться, пробуют обучить сеть большего или меньшего размера.

Свойство нейросети терять способность к обобщению при чрезмерном увеличении количества скрытых нейронов (степеней свободы) называют переобучением, или гиперразмерностью. Переобучение – явление, когда построенная модель хорошо объясняет примеры из обучающей выборки, но относительно плохо работает на примерах, не участвовавших в обучении (на примерах из тестовой выборки).

Однако, даже несмотря на все усилия по обучения НС, сеть все же может не обучаться давать много ошибок на тестовой выборке. Причины этого могут заключаться в следующем:

1) *противоречивость ОВ*, т.е. в обучающей выборке присутствуют задачи с одинаковыми условиями, но разными ответами (одинаковыми входными векторами данных, но разными выходными $A_i=A_j$, но $D_i\neq D_j$);

2) *нерепрезентативность ОВ*, т.е. обучающая выборка не охватывает всего множества ситуаций (выборка мала или просто узкоспециализирована);

3) *неравномерность ОВ*, т.е. в обучающей выборке может быть неодинаковое число примеров для разных классов. При этом при тестировании НС будет достаточно хорошо распознавать примеры класса, для которого в обучающей выборке было большинство примеров, и относить к этому же классу много примеров другого класса. Поэтому желательно, чтобы в обучающей выборке было примерно одинаковое число примеров для каждого класса, или, по крайней мере, не было отличия на порядок и более.

После обучения нейронной сети необходимо провести ее тестирование на *тестовой выборке* для определения точности решения не входивших в обучающую выборку задач. Точность правильного решения очень сильно зависит от репрезентативности обучающей выборки. Обычно при решении различных неформализованных задач в разных проблемных областях точность в 70-90% правильных ответов на тестовой выборке соответствует проценту правильных ответов при решении этих же задач специалистом-экспертом.

Нет строго определенной процедуры для выбора количества нейронов и количества слоев в сети. Чем больше количество нейронов и слоев, тем шире возможности сети, тем медленнее она обучается и работает и тем более нелинейной может быть зависимость вход-выход.

Количество нейронов и слоев связано:

1) со сложностью задачи;

2) с размерностью обучающей выборки;

3) с количеством данных для обучения;

4) с требуемым количеством входов и выходов сети;

5) с имеющимися ресурсами: памятью и быстродействием машины, на которой моделируется сеть;

Были попытки записать эмпирические формулы для числа слоев и нейронов, но применимость формул оказалась очень ограниченной.

Если в сети слишком мало нейронов или слоев:

1) сеть не обучится и ошибка при работе сети останется большой (ошибка обобщения);

2) на выход сети не будут передаваться резкие колебания аппроксимируемой функции ***y(x)***.

Превышение требуемого количества нейронов тоже мешает работе сети.

Если нейронов или слоев слишком много:

1) быстродействие будет низким, а памяти потребуется много (на фон-неймановских ЭВМ);

2) сеть переобучится: выходной вектор будет передавать незначительные и несущественные детали в изучаемой зависимости *y(x)*, например, шум или ошибочные данные;

3) зависимость выхода от входа окажется резко нелинейной: выходной вектор будет существенно и непредсказуемо меняться при малом изменении входного вектора *x*;

4) сеть будет неспособна к обобщению: в области, где нет или мало известных точек функции *y(x)* выходной вектор будет случаен и непредсказуем, не будет адекватен решаемой задаче.

Сеть радиально-базисных функций

Радиальные базисные нейронные сети (Radial Basis Function – RBF) или RBF-сети – это двухслойные сети прямого распространения. Базовая архитектура сетей на основе RBF-сетей, предполагает наличие трех слоев, выполняющих совершенно различные функции (рис.1).

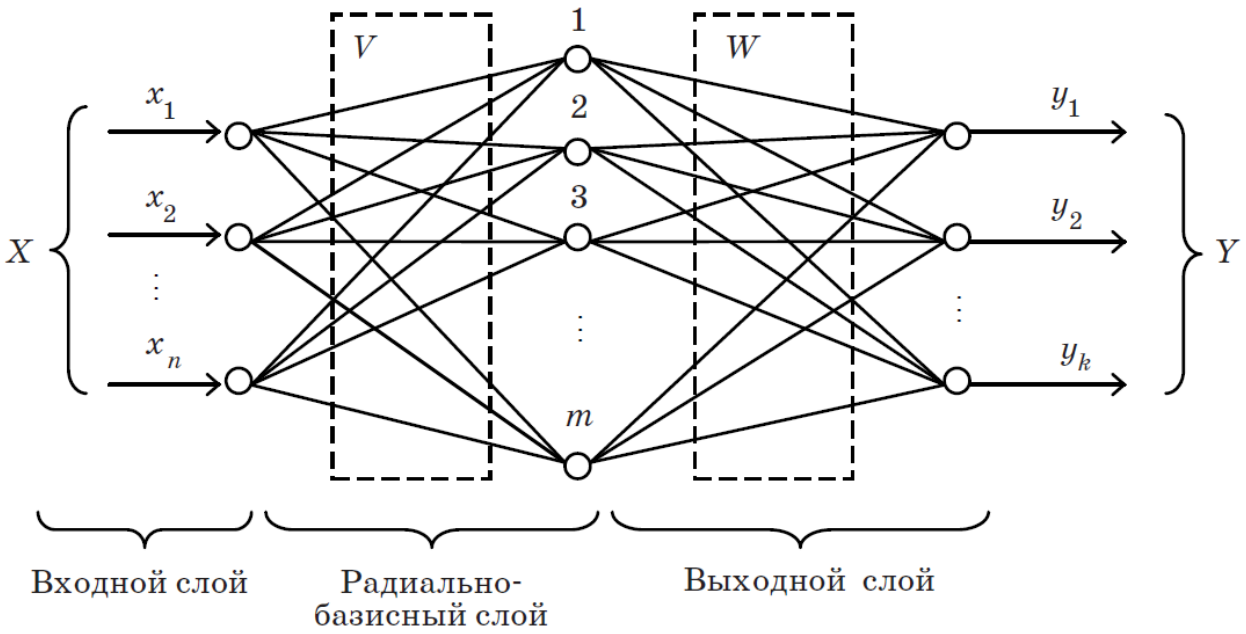


Рис. 1. Структура RBF-сети

Входной слой состоит из сенсорных элементов, которые связывают сеть с внешней средой. Второй слой является единственным скрытым (hidden) слоем сети. Он выполняет нелинейное преобразование входного пространства в скрытое. В большинстве реализаций скрытое пространство имеет более высокую размерность, чем входное.

В задачах классификации данных в пространстве более высокой размерности с большей вероятностью удовлетворяет требованию линейной разделимости. Поэтому в RBF-сетях размерность скрытого слоя, как правило, существенно превышает размерность входного слоя. Также важно отметить тот факт, что размерность скрытого пространства непосредственно связана со способностью сети аппроксимировать гладкое отображение «ВХОД-ВЫХОД». Чем выше размерность скрытого слоя, тем более высокой будет точность аппроксимации.

Математическую основу функционирования радиальных сетей составляет теорема Т. Ковера о разделимости образов, которая утверждает следующее:

Нелинейное преобразование сложной задачи классификации образов в пространство более высокой размерности повышает вероятность линейной разделимости образов.

Теорема Ковера о разделимости образов базируется на двух моментах :

1. Определение нелинейной скрытой функции $\varphi_i(x)$, где x– входной вектор, а $i = 1,2, \dots, K$ – размерность скрытого пространства.

2. Высокая размерность скрытого пространства по сравнению с размерностью входного. Эта размерность определяется значением, присваиваемым K (то есть количеством скрытых нейронов)/

Если вектор радиальных функций $\varphi(x) = [\varphi_1(x), \varphi_2(x), \dots, \varphi_K(x)]^T$ в N-мерном входном пространстве обозначить $\varphi(x)$, то это пространство является нелинейно φ-разделяемым на два пространственных класса X^+и X^- тогда, когда существует такой вектор весов w, что

$$w^T\varphi(x) > 0, x \in X^+$$

$$w^T\varphi(x) < 0, x \in X^-$$

Граница между этими классами определяется уравнением $w^T\varphi(x) = 0$

Ковер доказал, что каждое множество образов, случайным образом размещенных в многомерном пространстве, является φ-разделяемым с вероятностью 1 при условии большой размерности K этого пространства.

На практике это означает, что применение достаточно большого количества скрытых нейронов, реализующих радиальные функции $\varphi_i(x)$, гарантирует решение задачи классификации при построении всего лишь двухслойной сети. При этом скрытый слой должен реализовать вектор $\varphi(x)$, а выходной слой может состоять из единственного линейного нейрона, выполняющего суммирование выходных сигналов от скрытых нейронов с весовыми коэффициентами, заданными вектором w.

Приведем три примера (рис.2) φ-разделимых дихотомий (*деление на два взаимоисключающих понятия*) для различных множеств из пяти точек в двумерном пространстве: линейно-разделимая дихотомия (а); сферически разделимая дихотомия (6); квадратично-разделимая дихотомия (в)

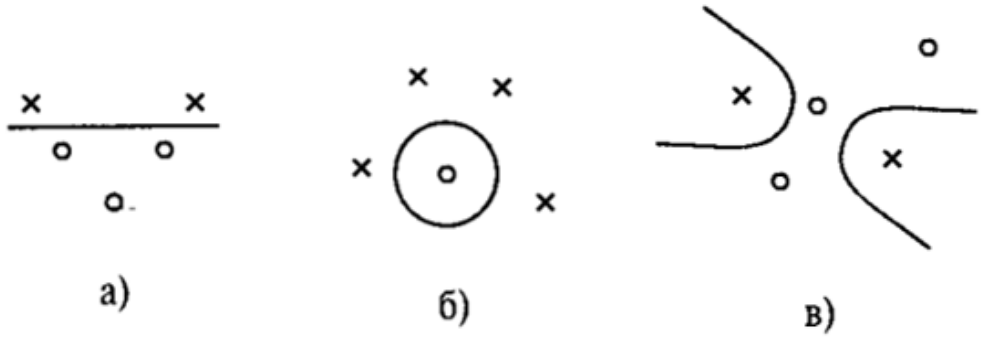


Рис. 2. Примеры φ-разделимых дихотомий.

Главное отличие RBF-сетей от обычных многослойных сетей прямого распространения состоит в функции нейронов скрытого слоя. В обычной многослойной сети каждый нейрон рабочего слоя реализует в многомерном пространстве гиперплоскость (рис. 2а), а RBF-нейрон – гиперсферу (рис. 2б, 2в).

Скрытый слой выполняет нелинейное отображение, реализуемое нейронами с базисными радиальными функциями, параметры которых уточняются в процессе обучения. Таким образом, все веса радиально-базисного слоя (скрытого слоя) полагаются равными единице, и работу i-го нейрона RBF-слоя можно описать формулой

$$f_i(X) = \varphi(\|X - C_i\|),$$

где C_i – вектор центра активационной RBF-функции нейрона: $X, C \in R^n$. Таким образом, входной вектор и вектор центра имеют одинаковую размерность.

В качестве радиальной базисной функции φ обычно используется гауссова функция (рис.3)

$$\varphi(\|X - C_i\|) = exp\left(\frac{\|X - C_i\|^2}{2{\sigma_i}^2}\right)$$

где σ_i – ширина «окна» активационной функции.

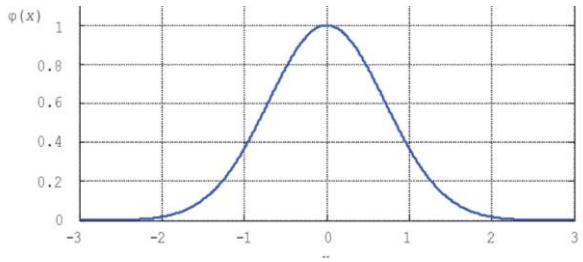


Рис. 2. Функция Гаусса

Проблему подбора параметров радиальных функций и значений весов w_i сети можно свести к минимизации целевой функции, которая при использовании метрики Эвклида записывается в форме

$$\|X - C_i\| = \sqrt{(x_1 - c_{i1})^2 + (x_2 - c_{i2})^2 + \ldots + (x_n - c_{in})^2}.$$

Таким образом, *i*-й нейрон скрытого слоя определяет сходство между входным вектором *X* и эталонным вектором *Ci*. На рис. 2 приведена гауссова функция одной переменной при $c = 0$ и $\sigma = 1$.

Как следует из рис. 2, функция активации RBF-нейрона принимает большие значения лишь в тех случаях, когда входной образ находится вблизи центра нейрона. Вне области, «покрытой» образами обучающей последовательности, сеть формирует лишь малые значения на своих выходах

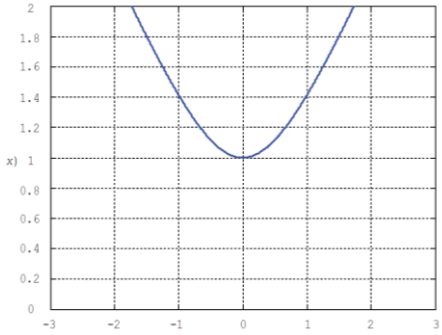
Возможны и другие варианты активационной функции. Например,

$\varphi(\|X - C_i\|) = \|X - C_i\|$ – линейная функция,

$\varphi(\|X - C_i\|) = \|X - C_i\|^3$ – кубическая функция,

$\varphi(\|X - C_i\|) = \left(\|X - C_i\|^2 + \sigma^2\right)^{1/2}$ – мультиквадратическая функция.

График мультиквадратической функции одной переменной на рис. 3.



Нейроны выходного слоя имеют линейную активационную функцию. Их роль сводится исключительно к взвешенному суммированию сигналов, генерируемых нейронами рабочего слоя:

$$y_j = \sum_{i=1}^{m} w_{ij} f_i(X), \quad j = \overline{1,k}.$$

Число нейронов выходного слоя определяется характером представления выходных данных.

Расчет параметров радиальной нейронной сети

Рассмотрим простой вариант определения весов RBF-сети.

Пусть RBF-сеть имеет k входов и один выход (рис. 4).

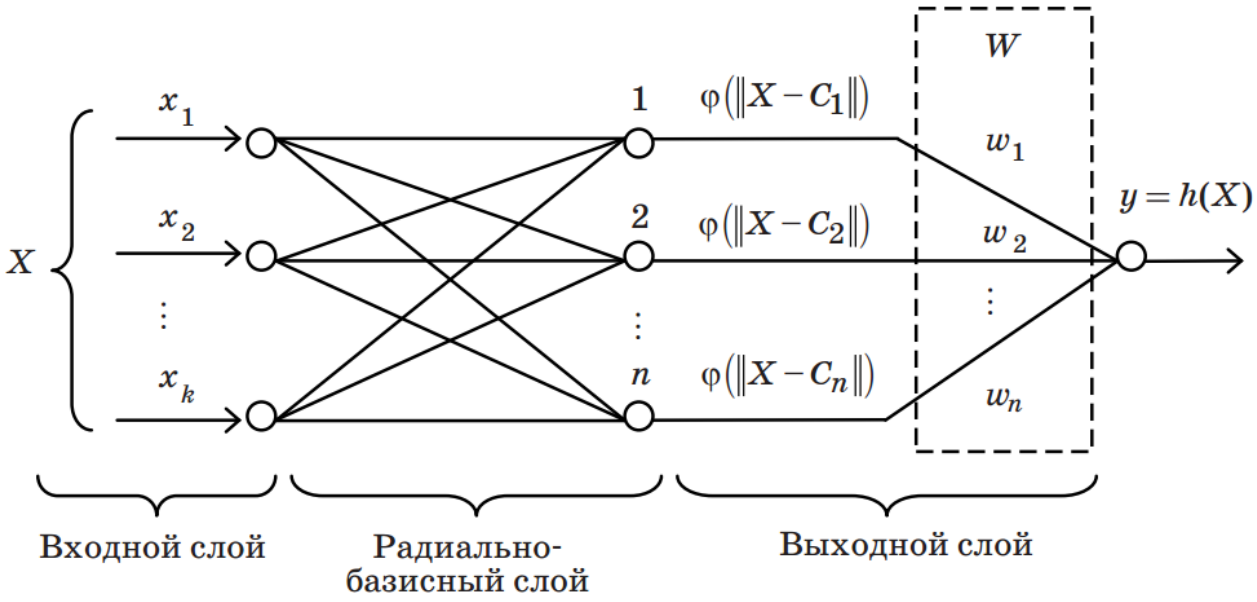


Рис. 4. Структура RBF-сети со скалярным выходом

Выберем число рабочих нейронов m = n, где n – число обучающих пар, заданных набором $\{(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)\}$, где $Xi = [x_{i1}, x_{i2}, \ldots, x_{ik}]^T$.

Для того чтобы каждый нейрон реагировал на «свой» вектор из обучающего набора, полагаем

$$C_i = X_i.$$

Окна активационной функции σ выбирают достаточно большими, но так, чтобы они не перекрывались в пространстве входных сигналов. Требуется найти такие весовые коэффициенты W, чтобы для каждого входного вектора из обучающего набора выполнялось

$$h(X_i) = y_i$$

Для первого входного вектора из обучающего набора можно записать

$$h(X_1)=\sum_{i=1}^{n} w_i f_i(X_1)=w_1 f_1(X_1)+w_2 f_2(X_1)+\ldots+w_n f_n(X_1)$$

Для всех n входных векторов при правильном выборе W должно выполняться

$$\begin{bmatrix} f_1(X_1) & f_2(X_1) & \ldots & f_n(X_1) \\ f_1(X_2) & f_2(X_2) & \ldots & f_n(X_2) \\ \vdots & \vdots & \vdots & \vdots \\ f_1(X_n) & f_2(X_n) & \ldots & f_n(X_n) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Вводим обозначения для матриц:

$$FW = Y.$$

Тогда

$$W = F^{-1}Y$$

Последняя формула позволяет рассчитать веса RBF-сети с одним выходом при числе нейронов скрытого слоя, равном числу обучающих пар.

Далее будет показано, что аналогичный результат может быть получен при произвольном числе нейронов выходного слоя RBF-сети, если число обучающих пар равно числу нейронов скрытого слоя.

Пусть выходной слой содержит p нейронов, так что вектор выхода имеет вид

$$Y_i = [y_{i1}, y_{i2}, \ldots, y_{ip}]^{\mathrm{T}}.$$

Определим веса нейронов выходного слоя w_{ij} $i = \overline{1,n}$; $j = \overline{1,p}$ Для этого сети предъявляется весь набор шаблонов, так что для всех n входных векторов можно записать

$$\begin{bmatrix} f_1(X_1) & f_2(X_1) & \dots & f_n(X_1) \\ f_1(X_2) & f_2(X_2) & \dots & f_n(X_2) \\ \vdots & \vdots & \vdots & \vdots \\ f_1(X_n) & f_2(X_n) & \dots & f_n(X_n) \end{bmatrix} \begin{bmatrix} w_{11} & w_{21} & \dots & w_{1p} \\ w_{21} & w_{22} & \dots & w_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{np} \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1p} \\ y_{21} & y_{22} & \dots & y_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ y_{n1} & y_{n2} & \dots & y_{np} \end{bmatrix}$$

Строки матрицы F соответствуют выходам нейронов скрытого слоя для каждого входного шаблона. Столбцы матрицы W соответствуют весовым коэффициентам нейронов выходного слоя. Строки матрицы Y описывают выходы нейронов второго (выходного) слоя для каждого входного вектора.

Таким образом, веса RBF-сети могут быть рассчитаны по тренировочным шаблонам. Если обучающие пары выбраны удачно, то сеть будет успешно выполнять интерполяцию и порождать близкие выходные сигналы для близких входных сигналов.

Однако в практических задачах условие $m = n$ обычно неприемлемо, поскольку требует использования очень большого числа нейронов. Кроме того, сеть становится чрезмерно чувствительной к шумам в обучающей выборке. Таким образом, обычно $m << n$ (число нейронов скрытого слоя m меньше числа обучающих пар n), и требуется найти приближенное решение задачи аппроксимации.

Процесс подбора приближенного значения весов может рассматриваться как задача минимизации целевой функции, описывающей ошибку выхода сети. Для оптимального выбора коэффициентов RBF-сети может быть использован метод наименьших квадратов. Рассмотрим RBF-сеть с одним выходным и m скрытыми нейронами:

$$y = h(X) = \sum_{i=1}^{m} w_i f_i(X) \tag{1}$$

Пусть необходимо аппроксимировать зависимость, заданную множеством вход-выходных данных (обучающая выборка):

$$\{(X_1, y_1), (X_2, y_2), \dots, (X_n, y_n)\}.$$

Для качественной аппроксимации требуется минимизировать ошибку выхода сети, заданную формулой

$$E = \sum_{i=1}^{n} \left(h(X_i) - y_i \right)^2. \tag{2}$$

Рассмотрим производную (2)

$$\frac{\partial E}{\partial w_j} = 2 \sum_{i=1}^{n} \left(h(X_i) - y_i \right) \frac{\partial h(X_i)}{\partial w_j}$$

В соответствии с (1)

$$\frac{\partial h(X_i)}{\partial w_j} = f_j(X_i)$$

следовательно,

$$\frac{\partial E}{\partial w_j} = 2 \sum_{i=1}^{n} \left(h(X_i) - y_i \right) f_j(X_i).$$

В точке оптимума

$$\frac{\partial E}{\partial w_j} = \sum_{i=1}^{n} \left(h(X_i) - y_i \right) f_j(X_i) = 0,$$

или

$$\sum_{i=1}^{n} f_j(X_i) h(X_i) = \sum_{i=1}^{n} f_j(X_i) y_i.$$

Обозначим

$$F_j = \begin{bmatrix} f_j(X_1) \\ f_j(X_2) \\ \vdots \\ f_j(X_n) \end{bmatrix}, \quad H = \begin{bmatrix} h(X_1) \\ h(X_2) \\ \vdots \\ h(X_n) \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

Тогда

$$F_j^{\mathrm{T}} H = F_j^{\mathrm{T}} Y,\ j = \overline{1,m},$$

$$\begin{bmatrix} F_1^{\mathrm{T}} H \\ F_2^{\mathrm{T}} H \\ \vdots \\ F_m^{\mathrm{T}} H \end{bmatrix} = \begin{bmatrix} F_1^{\mathrm{T}} Y \\ F_2^{\mathrm{T}} Y \\ \vdots \\ F_m^{\mathrm{T}} Y \end{bmatrix},$$

$$F^{\mathrm{T}} H = F^{\mathrm{T}} Y,$$

где

$$F = \begin{bmatrix} F_1 & F_2 & \cdots & F_m \end{bmatrix}.$$

Поскольку

$$H = \begin{bmatrix} h(X_1) \\ h(X_2) \\ \vdots \\ h(X_n) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{m} w_j f_j(X_1) \\ \sum_{j=1}^{m} w_j f_j(X_2) \\ \vdots \\ \sum_{j=1}^{m} w_j f_j(X_n) \end{bmatrix} = \begin{bmatrix} f_1(X_1) & f_2(X_1) & \cdots & f_m(X_1) \\ f_1(X_2) & f_2(X_2) & \cdots & f_m(X_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(X_n) & f_2(X_n) & \cdots & f_m(X_n) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} = FW,$$

можно записать

$$F^{\mathrm{T}} F W = F^{\mathrm{T}} Y$$

и окончательно

$$W = (F^{\mathrm{T}} F)^{-1} F^{\mathrm{T}} H = F^{+} H,$$

где F^{+} – псевдообратная матрица *(псевдоинверсия* прямоугольной матрицы F).

H – вектор ожидаемых значений выходного сигнала сети.

Пример расчета.

Пусть задан набор из трех пар точек (p = 3): {(0,9; 1), (2,1; 1,9), (3,1; 3)}

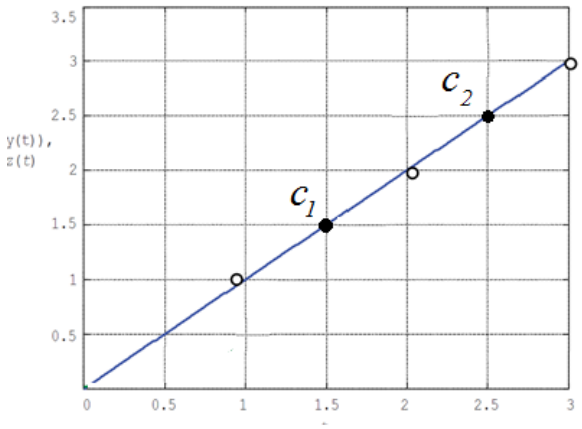
Требуется аппроксимировать эту зависимость функцией

$$y(x) = w_1 h_1(x) + w_2 h_2(x),$$

где $h_1(x)$ и $h_2(x)$ – выходы радиально-базисных нейронов, заданных в виде

$$h_1(x) = \exp\left(-(x-1{,}5)^2\right),\ \ h_2(x) = \exp\left(-(x-2{,}5)^2\right).$$

Так как центры функции активации $c_1 = {(0{,}9 + 2{,}1)}/{2} = 1{,}5$, соответственно $c_2 = 2{,}5$



Требуется найти неизвестные коэффициенты w_1 и w_2:

$h_1(x_1) = \exp(-(x-1{,}5)^2) = \exp(-(0{,}9-1{,}5)^2) = e^{-0{,}36} = 0.6977$

$h_1(x_2) = \exp(-(x-1{,}5)^2) = \exp(-(2{,}1-1{,}5)^2) = e^{-0{,}36} = 0.6977$

$h_1(x_3) = \exp(-(x-1{,}5)^2) = \exp(-(3{,}1-1{,}5)^2) = e^{-2{,}56} = 0.0773$

$h_2(x_1) = \exp(-(x-2{,}5)^2) = \exp(-(0{,}9-2{,}5)^2) = e^{-2{,}56} = 0.0773$

$h_2(x_2) = \exp(-(x-2{,}5)^2) = \exp(-(2{,}1-2{,}5)^2) = e^{-0{,}16} = 0.8521$

$h_2(x_3) = \exp(-(x-2{,}5)^2) = \exp(-(3{,}1-2{,}5)^2) = e^{-0{,}36} = 0.6977$

Тогда

$$F = \begin{bmatrix} h_1(x_1) & h_2(x_1) \\ h_1(x_2) & h_2(x_2) \\ h_1(x_3) & h_2(x_3) \end{bmatrix} = \begin{bmatrix} 0{,}6977 & 0{,}0773 \\ 0{,}6977 & 0{,}8521 \\ 0{,}0773 & 0{,}6977 \end{bmatrix},\ Y = \begin{bmatrix} 1 \\ 1{,}9 \\ 3 \end{bmatrix}$$

Далее

$$F^T = \begin{bmatrix} 0{,}6977 & 0{,}6977 & 0{,}0773 \\ 0{,}0773 & 0{,}8521 & 0{,}6977 \end{bmatrix}$$

Умножаем строку на столбец

$$a_{11} = 0{,}6977^2 + 0{,}6977^2 + 0{,}0773^2 = 0.9795$$

$$a_{12} = 0{,}6977 \times 0{,}0773 + 0{,}6977 \times 0{,}8521 + 0{,}0773 \times 0{,}6977 = 0.7024$$

и т.д.

Таким образом

$$F^{\mathrm{T}}F = \begin{bmatrix} 0{,}9795 & 0{,}7024 \\ 0{,}7024 & 1{,}2188 \end{bmatrix},$$

Найдем обратную матрицу используя формулу

$$\mathbf{A}^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{\det \mathbf{A}} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Обращение матрицы 2x2 возможно только при условии, что $ad - bc = \det A \neq 0$.

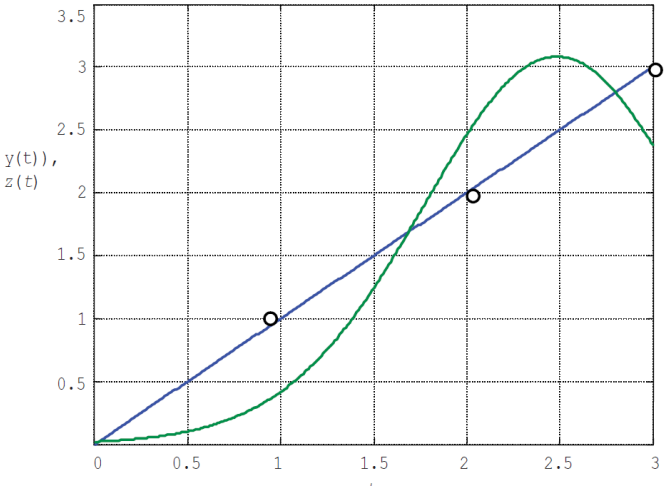
$$(F^{\mathrm{T}}F)^{-1} = \begin{bmatrix} 1{,}7398 & -1{,}0026 \\ -1{,}0026 & 1{,}3982 \end{bmatrix}$$

Далее

$$W = (F^T F)^{-1} F^T Y = \begin{bmatrix} 1{,}7398 & -1{,}0026 \\ -1{,}0026 & 1{,}3982 \end{bmatrix} \times \begin{bmatrix} 0{,}6977 & 0{,}6977 & 0{,}0773 \\ 0{,}0773 & 0{,}8521 & 0{,}6977 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1{,}9 \\ 3 \end{bmatrix}$$

$$= \begin{bmatrix} 0{,}1244 \\ 3{,}0373 \end{bmatrix}$$

На рис. приведен результат аппроксимации. Полученное качество очевидно невысоко по сравнению с линейной функцией z(x), которая обеспечивает в данном случае наилучшую аппроксимацию исходных данных.



Таким образом, если параметры гауссовой функции (центр и радиус) заданы, то задача нахождения весов выходного слоя RBF-сети может быть решена методами линейной алгебры – методом псевдообратных матриц.

Обучение радиальной сети

Таким, образом, процесс обучения сети RBF с учетом выбранного типа радиальной базисной функции сводится к двум этапам:

- к подбору центров c_i и параметров σ_i формы базисных функций (обычно используются алгоритмы обучения без учителя);
- к подбору весов нейронов выходного слоя (обычно используются алгоритмы обучения с учителем).

При этом второй этап значительно проще первого, поскольку сводится к вычислению выражения $W = F^{+} * H$, где основные вычислительные затраты – расчет псевдоинверсии матрицы F.

Для первого этапа как правило обучение на основе самоорганизации. Его целью является оценка подходящих положений центров радиальных базисных функций. Процесс самоорганизации обучающих данных автоматически разделяет пространство на так называемые области Вороного, определяющие различающиеся группы данных. Данные, сгруппированные внутри кластера, представляются центральной точкой, определяющей среднее значение всех его элементов. Центр кластера отождествляется с центром соответствующей радиальной функции.

Разделение данных на кластеры можно выполнить с использованием алгоритма k-means (к-средних, к-усреднений).

Согласно этому алгоритму центры радиальных базисных функций размещаются только в тех областях входного пространства, в которых имеются информативные данные. Если обучающие данные представляют непрерывную функцию, начальные значения центров в первую очередь размещают в точках, соответствующих всем максимальным и минимальным значениям функции.

Пусть N - число нейронов скрытого слоя, t – номер итерации алгоритма.

Тогда алгоритм К-усреднений можно описать следующим образом:

1. *Инициализация.* Случайным образом выбираем начальные значения центров $c_i(0)$, которые должны быть различны. При этом значения эвклидовой нормы по возможности должны быть небольшими.

2. *Выборка.* Выбираем вектор x_t из входного пространства.

3. *Определение центра-победителя.* Выбираем центр c_w, ближайший к x_t , для которого выполняется соотношение:

$$w = \arg\min_i \| x_t - c_i(t) \|, i = 1,2,\ldots,N.$$

4. *Уточнение.* Центр-победитель подвергается уточнению в соответствии с формулой

$$c_i(t+1) = c_i(t) + \eta(x_t - c_i(t)), \quad (1)$$

где η - коэффициент обучения, имеющий малое значение (обычно η<<1), причем уменьшающееся во времени. Остальные центры не изменяются.

5. *Продолжение.* Увеличиваем на единицу значение t и возвращаемся к шагу 2, пока положение центров не стабилизируется.

Также применяется разновидность алгоритма, в соответствии с которой значение центра-победителя уточняется в соответствии с формулой (1), а один или несколько ближайших к нему центров отодвигаются в противоположном направлении, и этот процесс реализуется согласно выражению

$$c_i(t+1) = c_i(t) - \eta_1(x_t - c_i(t)).$$

Такая модификация алгоритма позволяет отдалить центры, расположенные близко друг к другу, что обеспечивает лучшее обследование всего пространства данных ($\eta_1 < \eta$).

После фиксации местоположения центров проводится подбор значений параметров σ_i , соответствующих конкретным базисным функциям. Параметр σ_i радиальной функции влияет на форму функции и величину области ее охвата, в которой значение этой функции не равно нулю. Подбор σ_i должен проводится таким образом, чтобы области охвата всех радиальных функций накрывали все пространство входных данных, причем любые две зоны могут перекрываться

только в незначительной степени. При такой организации подбора значения σ_i, реализуемое радиальной сетью отображение функции будет относительно монотонным.

Для расчета σ_i может быть применен алгоритм, при котором на значение σ_i влияет на расстояние между i-м центром c_i и его R ближайшими соседями. В этом случае значение σ_i определяется по формуле

$$\sigma_i = \sqrt{\frac{1}{R}\sum_{k=1}^{R} \| c_i - c_k \|^2}.$$

На практике значение R обычно лежит в интервале [3; 5].

Данный алгоритм обеспечивает только локальную оптимизацию, зависящую от начальных условий и параметров процесса обучения. При неудачно выбранных начальных условиях, некоторые центры могут застрять в области, где количество обучающих данных ничтожно мало, либо они вообще отсутствуют. Следовательно, процесс модификации центров затормозится или остановится.

Для решения данной проблемы могут быть применены два различных подхода:

1. Задать фиксированные значения η для каждого центра. Центр, наиболее близкий к текущему вектору x, модифицируется сильнее, остальные - обратно пропорционально их расстоянию до этого текущего вектора x.

2. Использовать взвешенную меру расстояния от каждого центра до вектора *x*. Весовая норма делает «фаворитами» те центры, которые реже всего побеждают.

Оба подхода не гарантируют 100% оптимальность решения.

Подбор коэффициента η тоже является проблемой. Если η имеет постоянное значение, то оно должно быть мало, чтобы обеспечить сходимость алгоритма, следовательно, увеличивается время обучения.

Адаптивные методы позволяют уменьшать значение η по мере роста времени t. Наиболее известным адаптивным методом является алгоритм Даркена-Муди:

$$\eta(t) = \frac{\eta_0}{1 + \frac{t}{T}},$$

где T – постоянная времени, подбираемая для каждой задачи. При $t < T$ η не изменяется, при $t > T$ – уменьшается до нуля.

Применение метода обратного распространения ошибки для радиально-базисных сетей

В рамках следующего подхода центры радиальных базисных функций и все остальные свободные параметры сети настраиваются в процессе обучения с учителем. Другими словами, сеть RBF принимает самый общий вид.

Естественным выбором для такой ситуации является обучение, которое составляют градиентные алгоритмы обучения с учителем, где используется алгоритм обратного распространения ошибки. Их основу составляет целевая функция, которая для одного обучающего примера имеет вид:

$$E = \frac{1}{2}\left[\sum_{i=1}^{K} w_i \varphi_i(x) - d\right]^2 \quad (2)$$

Предположим, что применяется гауссовская радиальная функция вида:

$$\varphi_i(x(t)) = \exp\left(-\frac{1}{2}u_i(t)\right) \quad (3)$$

где i– индекс нейрона скрытого слоя, j–индекс компонента входного вектора, t – индекс обучающего примера в выборке.

Обучение сети с использованием алгоритма обратного распространения ошибки проводится в два этапа. На первом этапе предъявляется обучающий пример и рассчитываются значения сигналов выходных нейронов сети и значение целевой функции, заданной выражением (2). На втором этапе минимизируется значение этой функции. Подбор значений параметров можно осуществлять, используя градиентные методы оптимизации независимо от объекта обучения – будь то вес или центр. Независимо от выбираемого метода градиентной оптимизации, необходимо, прежде всего, получить вектор градиента целевой

функции относительно всех параметров сети. В результате дифференцирования этой функции получим:

$$\frac{\partial E(t)}{\partial w_0(t)} = y(t) - d(t) \tag{4}$$

$$\frac{\partial E(t)}{\partial w_i(t)} = \exp\left(-\frac{1}{2}u_i(t)\right)(y(t) - d(t)) \tag{5}$$

$$\frac{\partial E(t)}{\partial c_{ij}(t)} = (y(t) - d(t))w_i(t)\exp\left(-\frac{1}{2}u_i(t)\right)\frac{(x_j(t) - c_{ij}(t))}{\sigma_{ij}^2(t)} \tag{6}$$

$$\frac{\partial E(t)}{\partial \sigma_{ij}(t)} = (y(t) - d(t))w_i(t)\exp\left(-\frac{1}{2}u_i(t)\right)\frac{(x_j(t) - c_{ij}(t))}{\sigma_{ij}^3(t)} \tag{7}$$

Если в выходном слое содержится несколько нейронов, то формулы (4), (5), (6), (7) соответственно примут следующий вид:

$$\frac{\partial E(t)}{\partial w_{0s}(t)} = y_s(t) - d_s(t) \tag{4a}$$

$$\frac{\partial E(t)}{\partial w_{is}(t)} = \exp\left(-\frac{1}{2}u_i(t)\right)(y_s(t) - d_s(t)) \tag{5a}$$

$$\frac{\partial E(t)}{\partial c_{ij}(t)} = (y(t) - d(t))w_{is}(t)\exp\left(-\frac{1}{2}u_i(t)\right)\frac{(x_j(t) - c_{ij}(t))}{\sigma_{ij}^2(t)} \tag{6a}$$

$$\frac{\partial E(t)}{\partial \sigma_{ij}(t)} = (y(t) - d(t))w_{is}(t)\exp\left(-\frac{1}{2}u_i(t)\right)\frac{(x_j(t) - c_{ij}(t))}{\sigma_{ij}^3(t)} \tag{7a}$$

где s – номер нейрона выходного слоя.

При использовании метода наискорейшего спуска формулы для корректировки параметров радиально-базисной сети примут следующий вид:

$$w_i(t+1) = w_i(t) - \eta\frac{\partial E(t)}{\partial w_i(t)}, \tag{8}$$

$$c_{ij}(t+1) = c_{ij}(t) - \eta \frac{\partial E(t)}{\partial c_{ij}(t)}, \tag{9}$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \eta \frac{\partial E(t)}{\partial \sigma_{ij}(t)}. \tag{10}$$

Если в выходном слое содержится несколько нейронов, то формула (8) примет следующий вид:

$$w_{is}(t+1) = w_{is}(t) - \eta_1 \frac{\partial E(t)}{\partial w_{is}(t)} \tag{8a}$$

Сравнение сетей RBF и многослойных персептронов

Сети на основе радиальных базисных функций (RBF) и многослойный персептрон (MLP) являются примерами нелинейных многослойных сетей прямого распространения. И те и другие являются универсальными аппроксиматорами. Таким образом, неудивительно, что всегда существует сеть RBF, способная имитировать многослойный персептрон (и наоборот). Однако эти два типа сетей отличаются по некоторым важным аспектам.

1. Сети RBF (в своей основной форме) имеют один скрытый слой, в то время как многослойный персептрон может иметь большее количество скрытых слоев.

2. Обычно вычислительные (computationa1) узлы многослойного персептрона, расположенные в скрытых и выходном слоях, используют одну и ту же модель нейрона. С другой стороны, вычислительные узлы скрытого слоя сети RBF могут в корне отличаться от узлов выходного слоя и служить разным целям.

3. Скрытый слой в сетях RBF является нелинейным, в то время как выходной линейным. В то же время скрытые и выходной слои многослойного персептрона, используемого в качестве классификатора, являются нелинейными. Если многослойный персептрон используется для решения задач нелинейной регрессии, в качестве узлов выходного слоя обычно выбираются линейные нейроны.

4. Аргумент функции активации каждого скрытого узла сети RBF представляет собой *Евклидову норму* (расстояние) между входным вектором и центром радиальной функции. В то же время аргумент функции активации каждого скрытого узла многослойного персептрона – это скалярное произведение входного вектора и вектора синоптических весов данного нейрона.

5. Многослойный персептрон обеспечивает глобальную аппроксимацию нелинейного отображения. С другой стороны, сеть RBF с помощью экспоненциально уменьшающихся локализованных нелинейностей (т.е. функций raycca) создает локальную аппроксимацию нелинейного отображения.

Это, в свою очередь, означает, что для аппроксимации нелинейного отображения с помощью многослойного персептрона может потребоваться меньшее число параметров, чем для сети RBF при одинаковой точности вычислений.

Линейные характеристики выходного слоя сети RBF означают, что такая сеть более тесно связана с персептроном Розенблатта, чем с многослойным персептроном. Тем не менее сети RBF отличаются от этого персептрона тем, что способны выполнять нелинейные преобразования входного пространства. Это было хорошо продемонстрировано на примере решения задачи XOR, которая не может быть решена ни одним линейным персептроном, но с легкостью решается сетью RBF.

https://basegroup.ru/community/articles/rbf