

7. ЛЕКЦИЯ. Сети на основе радиальных базисных функций

Процесс создания нейронных сетей, обучаемых с учителем, можно упростить множеством способов. Рассмотренный алгоритм обратного распространения для многослойного персептрона можно рассматривать как реализацию рекурсивной технологии, которая в статистике называется стохастической аппроксимацией. На этой лекции рассмотрим еще один подход, в рамках которого построение нейронной сети рассматривается как задача аппроксимации кривой по точкам в пространстве высокой размерности. В соответствии с такой точкой зрения обучение эквивалентно нахождению такой поверхности в многомерном пространстве, которая наиболее точно соответствует данным обучения. При этом критерий «наилучшего соответствия» выбирается в не котором статистическом смысле. Таким образом, обобщение эквивалентно использованию этой многомерной поверхности для интерполяции данных тестирования. Такой подход лежит в основе метода радиальных базисных функций, состоящего в традиционной интерполяции в многомерном пространстве. В контексте нейронных сетей скрытые нейроны реализуют набор «функций», являющихся произвольным «базисом» для разложения входных образов (векторов). Соответствующие преобразования называют радиальными базисными функциями. Понятие радиальных базисных функций впервые было введено при решении задачи интерполяции вещественных функций нескольких переменных.

В настоящее время радиальные базисные функции составляют одно из главных направлений исследований в области численного анализа. Базовая архитектура сетей на основе радиальных базисных функций, или RBF-сетей, предполагает наличие трех слоев (рис.1), выполняющих совершенно различные функции.

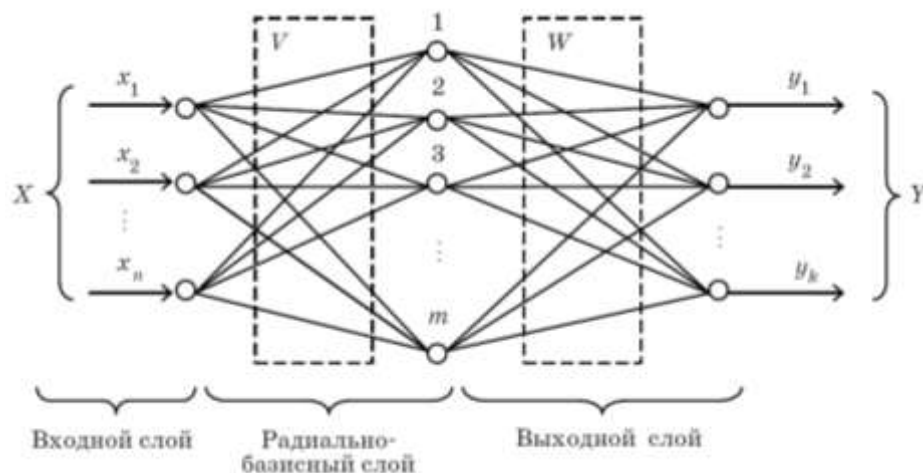


Рис.1. Архитектура RBF-сети

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

Задача 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.

Теорема Ковера о разделимости множеств

Если сеть на основе радиальных базисных функций используется для решения сложных задач классификации образов, то основная идея решения обычно состоит в нелинейном преобразовании входных данных в пространстве более высокой размерности. Теоретическую основу такого подхода составляет теорема 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

$$\begin{aligned}w^T \varphi(x) &> 0, x \in X^+ \\w^T \varphi(x) &< 0, x \in X^-\end{aligned}$$

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 (б); квадратично-разделимая дихотомия (в)

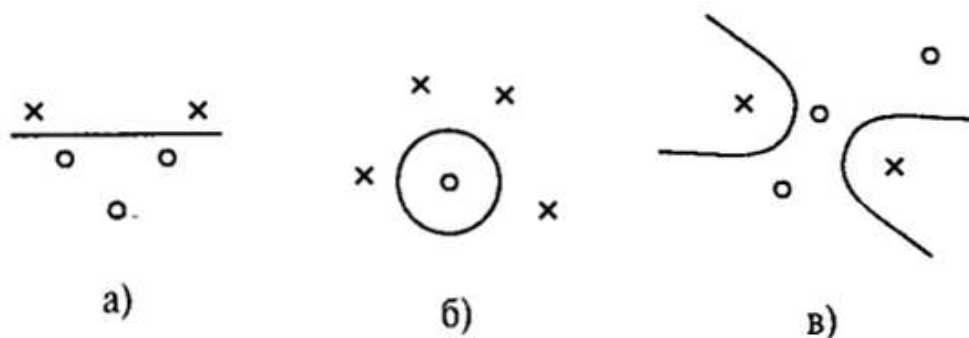


Рис. 2. Примеры φ -разделимых дихотомий.

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).

RBF-функции

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.

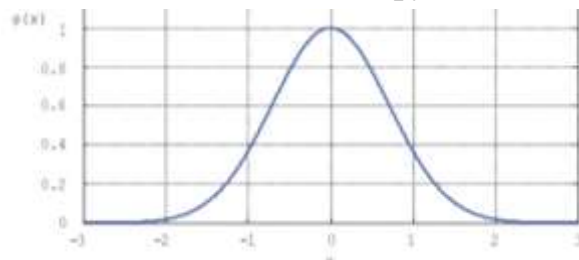


Рис. 3. Функция Гаусса

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

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

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

Как следует из рис. 3, 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,

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

График мультикватратической функции одной переменной на рис. 4.

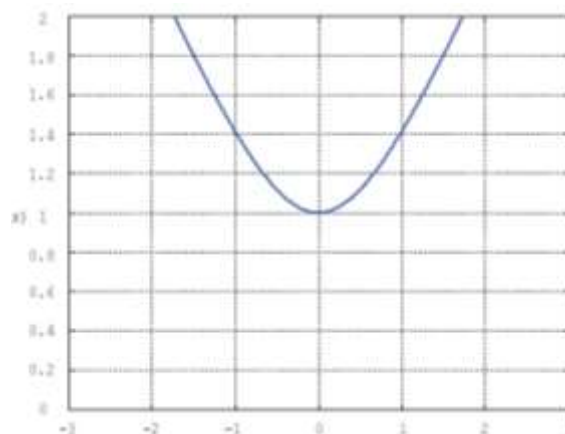


Рис.4. Мультикватратическая функция

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. 5).

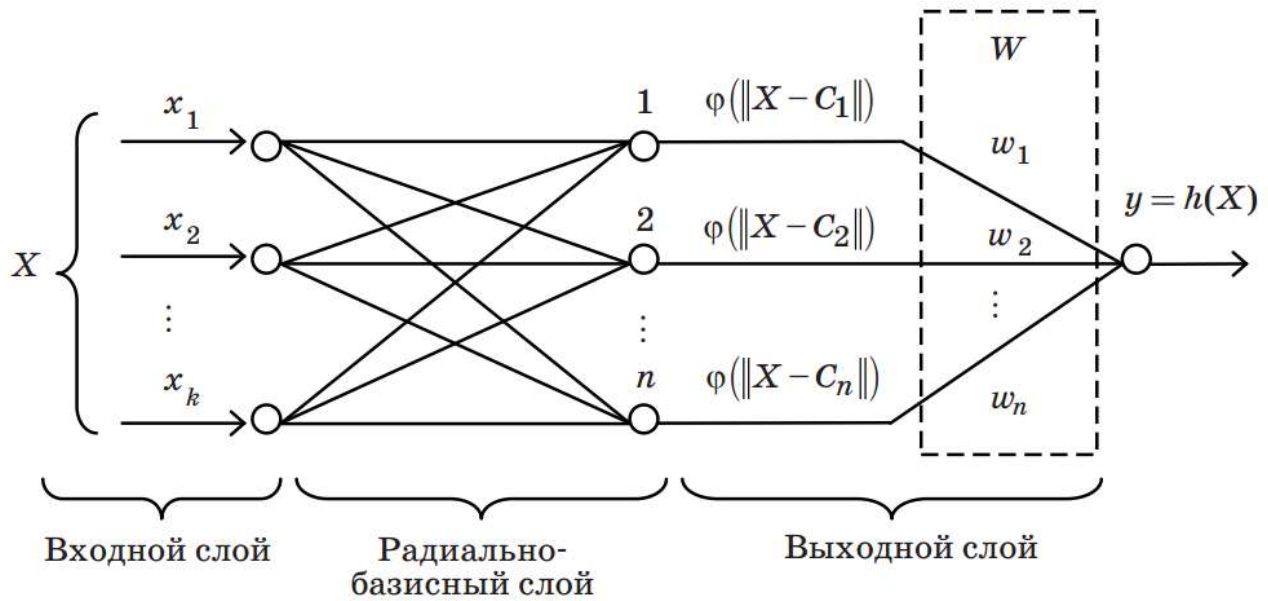


Рис. 5. Структура RBF-сети со скалярным выходом

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

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

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

Для всех n входных векторов при правильном выборе W должно выполняться

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

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

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

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

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

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 $m \ll n$ (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:

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

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

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

Рассмотрим производную (2)

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

В соответствии с (1)

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

следовательно,

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

В точке оптимума

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

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

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

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

$$F^T H = F^T Y,$$

где

$$F = [F_1 \quad F_2 \quad \dots \quad F_m]$$

Поскольку

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

можно записать

$$F^T FW = F^T Y$$

и окончательно

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

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

H – вектор ожидаемых значений выходного сигнала сети.

Пример расчета.

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

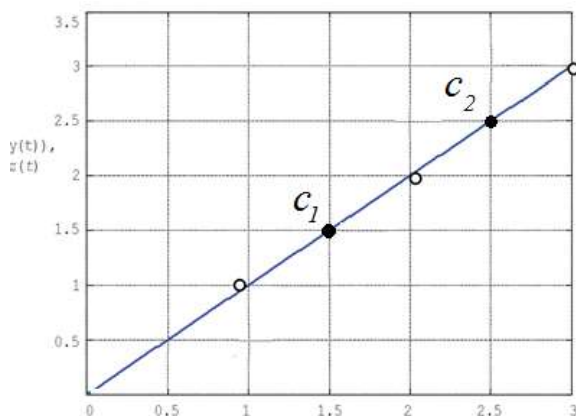
Требуется аппроксимировать эту зависимость функцией

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

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

$$h_1(x) = \exp(-(x - 1,5)^2), \quad h_2(x) = \exp(-(x - 2,5)^2).$$

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

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

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

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

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

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

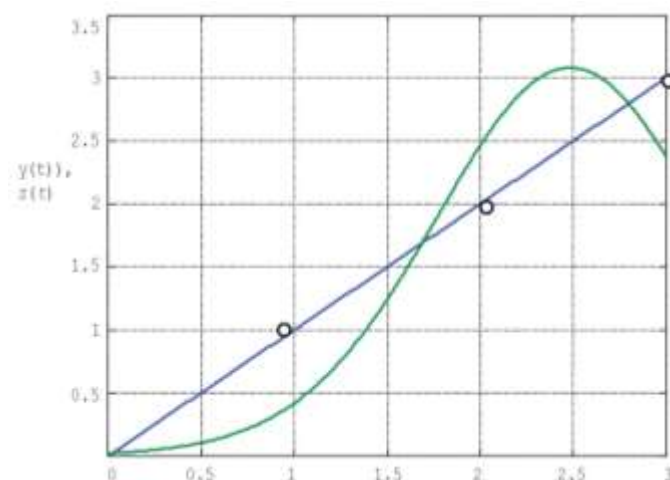
Далее

$$W = (F^T F)^{-1} F^T Y$$

$$= \begin{bmatrix} 1,7398 & -1,0026 \\ -1,0026 & 1,3982 \end{bmatrix} \times \begin{bmatrix} 0,6977 & 0,6977 & 0,0773 \\ 0,0773 & 0,8521 & 0,6977 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1,9 \\ 3 \end{bmatrix}$$

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

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 евклидовой 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:

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

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 $\eta \ll 1$), 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 проводиться 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.

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 (computational) 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 гаусса) 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.