Лекция 10.

Тема: “Радиально-базисная сеть”

Радиально-базисные сети (Radial Basis Function Nets. RBF) были предложены для аппроксимации функций многих переменных. Как показано в этих работах, с помощью радиально-базисных функций можно сколь угодно точно аппроксимировать заданную функцию. Как и многослойный персептрон, радиально-базисная сеть (РБС) является *универсальным аппроксиматором.*

Математическую основу РБ сети составляет метод потенциальных функций, разработанный *М.А. Айзерманом, Э.М. Бравермaном и Л.И. Розоноэром*, позволяющий представить некоторую функцию y(x) в виде суперпозиции *потенциальных* или *базисных функций* $f_i(x)$

$$y(x) = \sum_{i=1}^{N} a_i f_i(x) = a^T f(x), \qquad (12.1)$$

где $a_i = (a_1, a_2, ...a_N)^T$ - вектор подлежащих определению параметров; $f(x) = (f_1(x), f_2(x), ..., f_N(x))^T$ - вектор базисных функций.

В РБС в качестве базисных выбираются некоторые функции расстояния между векторами

$$f_i(x) = f(\| x - c_i \|). \qquad (12.2)$$

Векторы c_i называют *центрами базисных функций*. Функции $f_i(x)$ выбираются неотрицательными и возрастающими при увеличении $\| x - c_i \|$. В качестве меры близости векторов x и c_i выбираются обычно либо *евклидова метрика* $\| x - c_i \| = (\sum_{j=1}^{N} (x_j - c_{ij})^2)^{\frac{1}{2}}$, либо *манхэттенская*

$$\| x - c_i \| = \sum_{j=1}^{N} | x_j - c_{ij} |, \text{ где } | x_j - c_{ij} | = (x_j - c_{ij}) \operatorname{sign} (x_j - c_{ij}) ,$$

$$\operatorname{sign} (x_j - c_{ij}) = \begin{cases} 1, \text{если } (x_j - c_{ij}) > 0; \\ 0, \text{если } (x_j - c_{ij}) = 0; \\ -1, \text{если } (x_j - c_{ij}) < 0. \end{cases} \qquad (12.3)$$

Радиально-базисные сети имеют много общего со стохастическими сетями (см. раздел 13). Как и стохастические сети, РБС обладают большой скоростью обучения. Следует также отметить, что при их обучении не возникает проблем с “застреванием” в локальных минимумах. Однако в связи с тем, что при выполнении непосредственно классификации проводятся довольно сложные вычисления, возрастает время получения результата.

12.1 Архитектура РБС

Особенностью этих сетей является наличие *радиально-симметричного шаблонного слоя*.

12.1.1 Структура сети

Структура РБС соответствует сети прямого распространения первого порядка (рис.12.1).

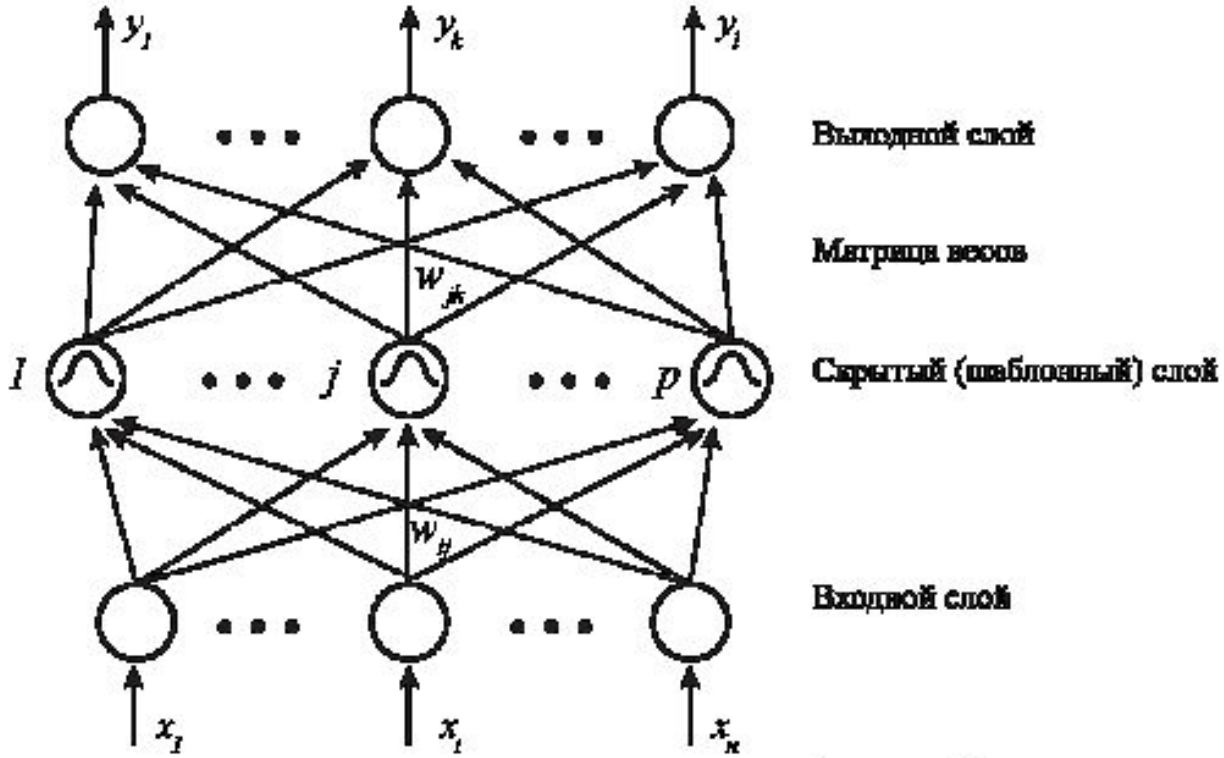


Рис.12.1 – Структура радиально-базисной сети

Информация об образах передается с входного слоя на скрытый, являющийся шаблонным и содержащий p нейронов. Каждый нейрон шаблонного слоя, получая полную информацию о входных сигналах **x**, вычисляет функцию

$$f_i(x) = f((x - c_i)^T R^{-1}(x - c_i)), i = \overline{1,p}, \qquad (12.4)$$

где x – вектор входных сигналов (Nx1); c_i - вектор центров (Nx1); R – весовая матрица.

Как уже отмечалось, особенностью данных сетей является наличие радиально-симметричного шаблонного слоя, в котором анализируется расстояние $(x - c_i)^T R^{-1}(x - c_i)$ между входным вектором и центром, представленным в виде вектора во входном пространстве. Вектор центров определяется по обучающей выборке и сохраняется в пространстве весов от входного слоя к слою шаблонов.

12.1.2 Нейрон шаблонного слоя сети

На рис. 12.2 представлен i-й нейрон шаблонного слоя РБ сети. Обработку поступающей на него информации условно можно разделить на два этапа: на первом – вычисляется расстояние между предъявленным образом x и вектором центров c_i с учетом выбранной метрики и нормы матрицы R, на втором это расстояние преобразуется нелинейной активационной функцией f(x). Двойные стрелки на рисунке обозначают векторные сигналы, а тройные – матричный сигнал.

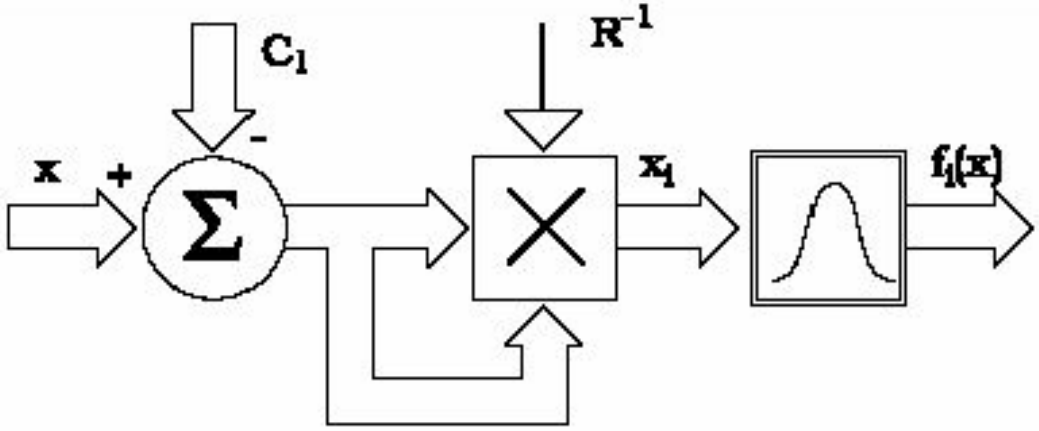


Рис. 12.2 Нейрон шаблонного слоя РБС

В качестве функции преобразования f(•) наиболее часто выбираются следующие:

- гауссова функция

$$f(x) = \exp\left\{-\frac{(x-c)^2}{2\sigma^2}\right\}; \qquad (12.5)$$

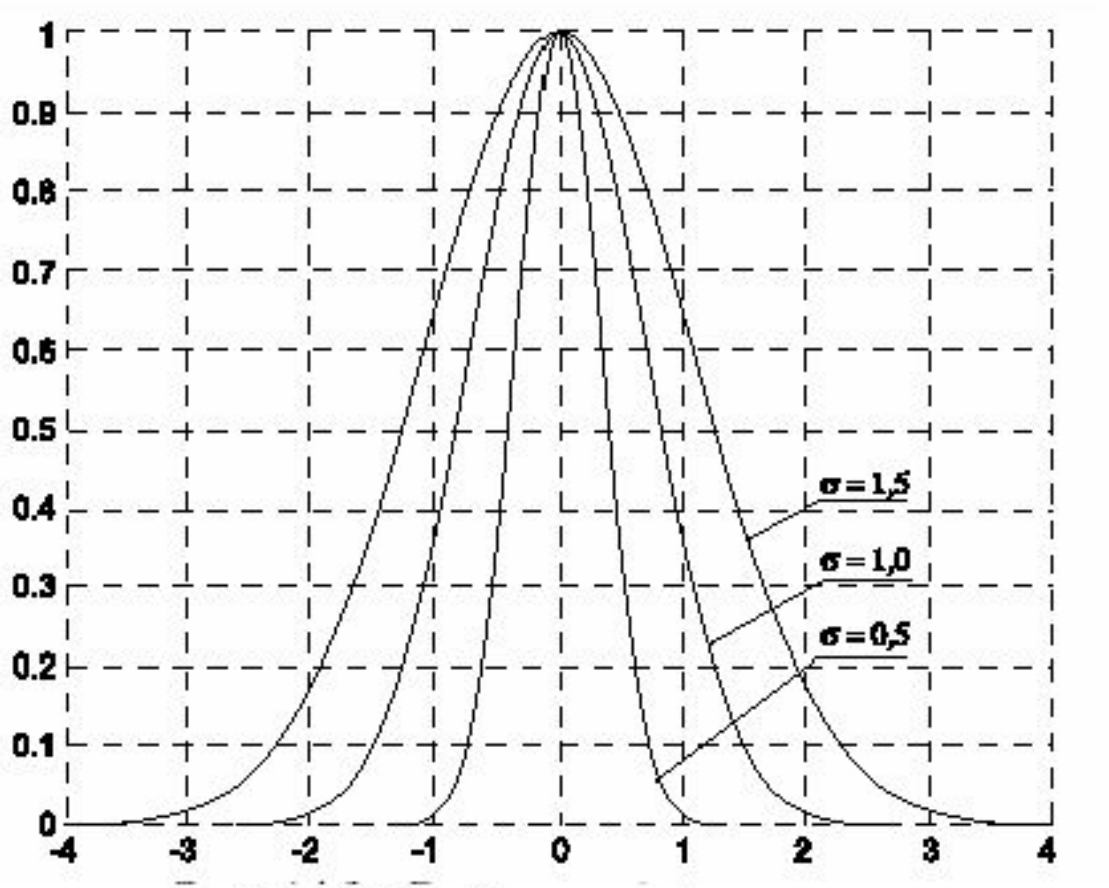


Рис. 12.3 – Гауссова функция

- мультиквадратичная функция

$$f(x) = \left[\frac{(x - c)^2}{\sigma^2} + a^2 \right]^{\frac{1}{2}}; \qquad (12.6)$$

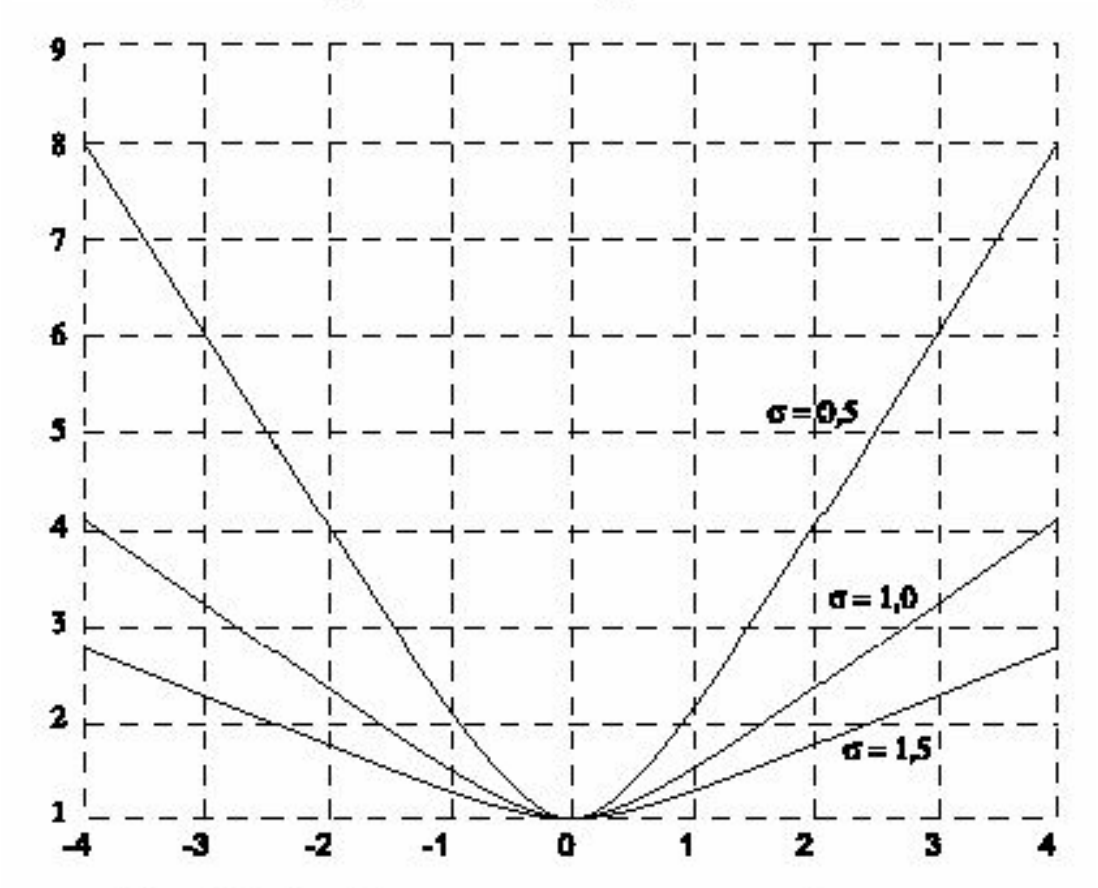


Рис.12.4 – Мультиквадратичная функция

- обратная мультиквадратичная функция

$$f(x) = \left[\frac{(x-c)^2}{\sigma^2} + a^2 \right]^{-\frac{1}{2}}; \qquad (12.7)$$

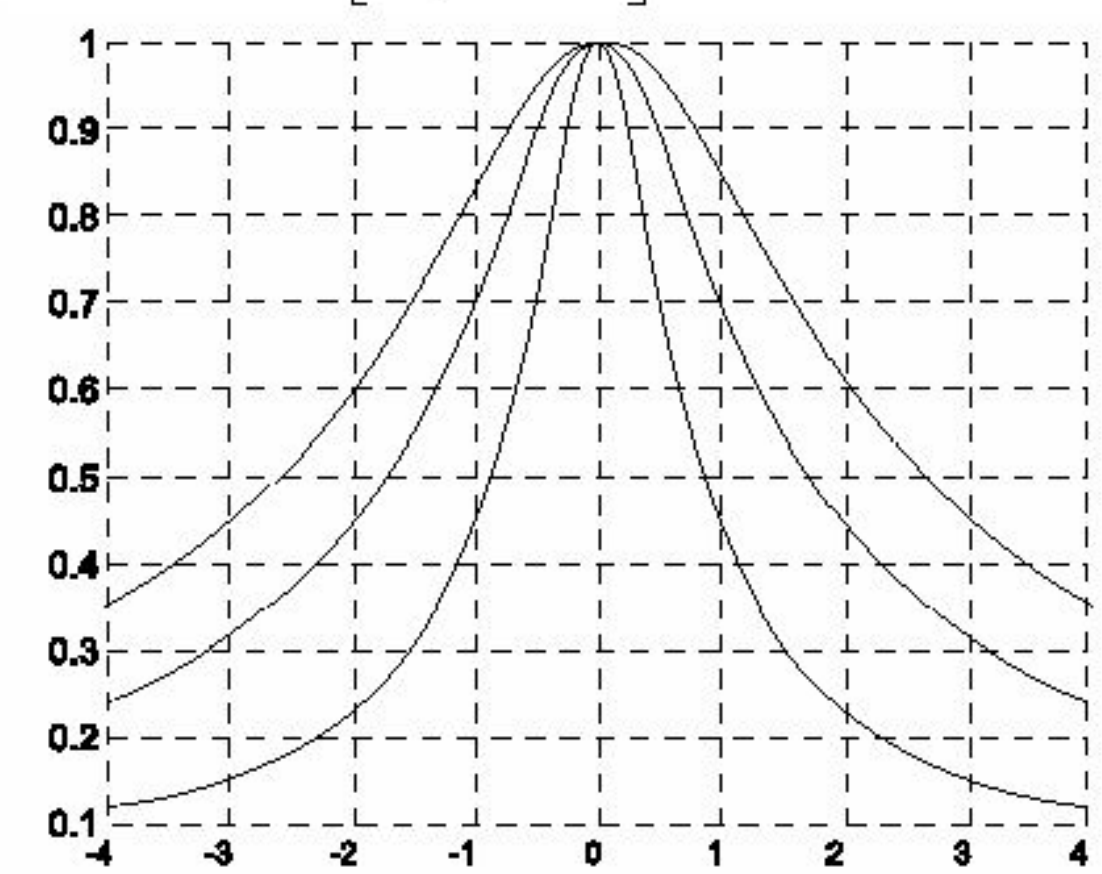


Рис.12.5 – Обратная мультиквадратичная функция (a=1, c=0)

- сплайн-функция

$$f(x) = x^2 \log(x); \qquad (12.8)$$

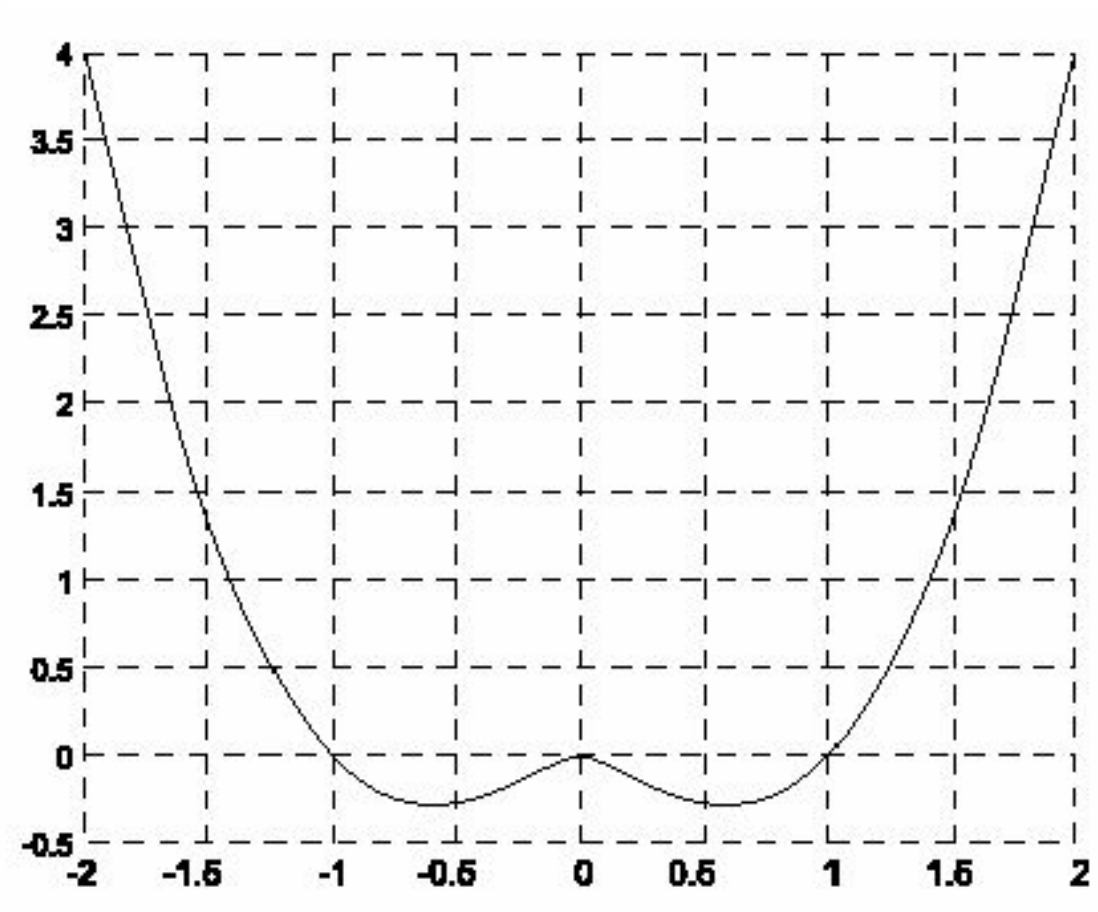


Рис.12.6 – Сплайн-функция

- функция Коши

$$f(x) = (1 + |x|)^{-1}. \qquad (12.9)$$

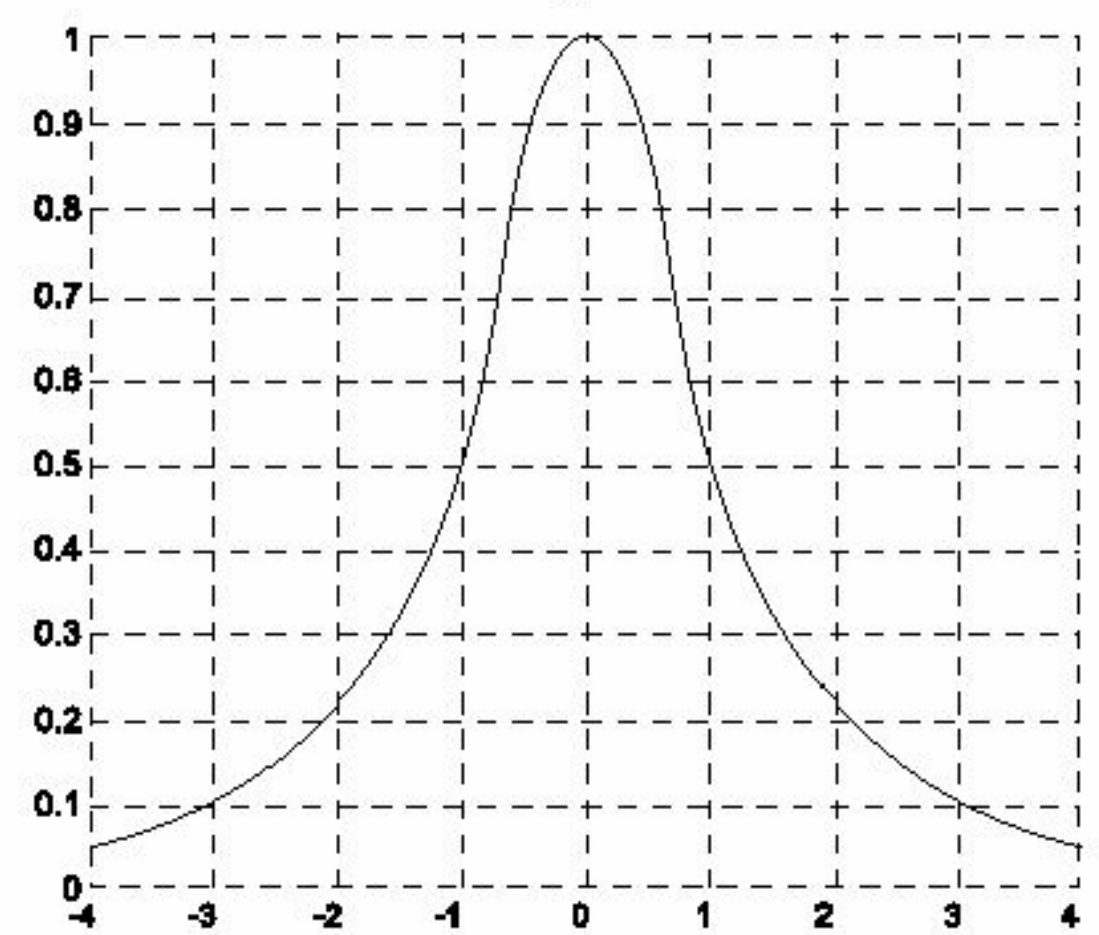


Рис.12.7 – Функция Коши

Норма матрицы R^{-1} определяет положение осей в пространстве. В общем виде матрица R^{-1} может быть представлена следующим образом:

$$R^{-1} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1p} \\ r_{21} & r_{22} & \dots & r_{2p} \\ \dots & \dots & \dots & \dots \\ r_{p1} & r_{p2} & \dots & r_{pp} \end{bmatrix}. \qquad (12.10)$$

Весовую матрицу R_1 также называют *обратной ковариационной матрицей*. Элементы этой матрицы равны

$$r_{ij} = \sigma_{ij}^{-2}, \qquad i, j = \overline{1, p}. \qquad (12.11)$$

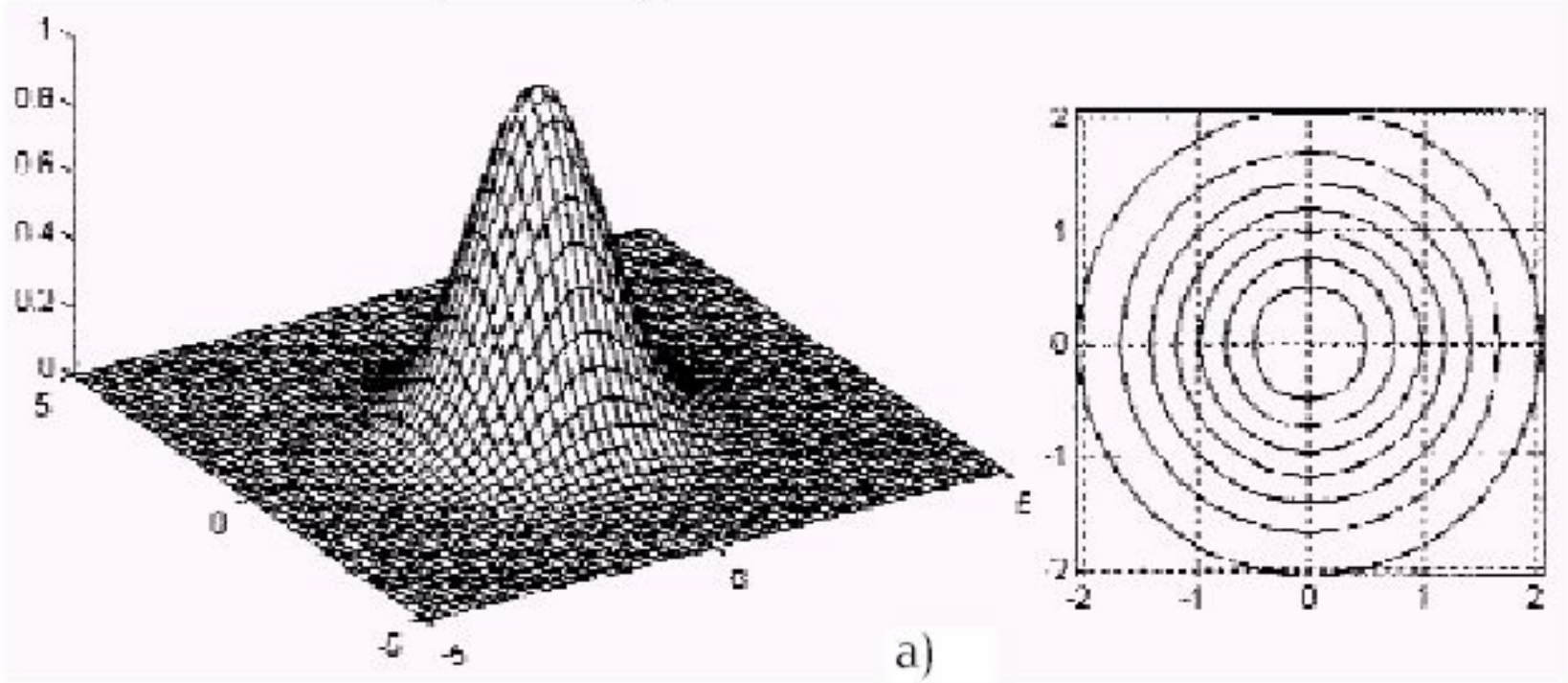
Здесь $\sigma_{ij}^{-2} = \sigma_{ji}^{-2}$ - некоторые управляемые параметры.

Часто матрица R^{-1} выбирается диагональной, т.е. $r_{ij} = 0$ для $i \neq j$, и более того, принимают $r_{ii} = \sigma_{ii}^{-2} = \sigma^{-2} = \text{const}$. В этом случае для гауссовой функции f(x) (12.5) σ представляет собой стандартное отклонение.

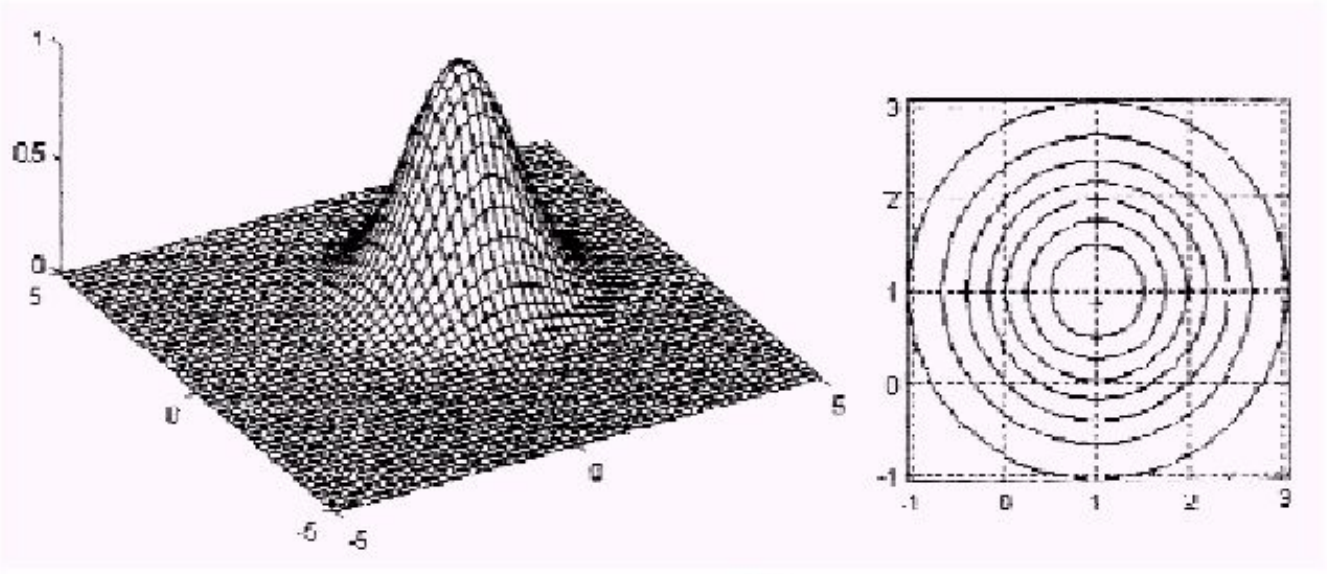
Параметр σ_i определяется экспериментально, хотя сеть не является столь критичной к его выбору. Зависимость базисных функций от σ показана на рис.12.3-12.5.

На рис.12.8 показано влияние выбора величин c и σ_{ij} на вид базисной функции i-го нейрона скрытого слоя для двухвходовой РБ сети.

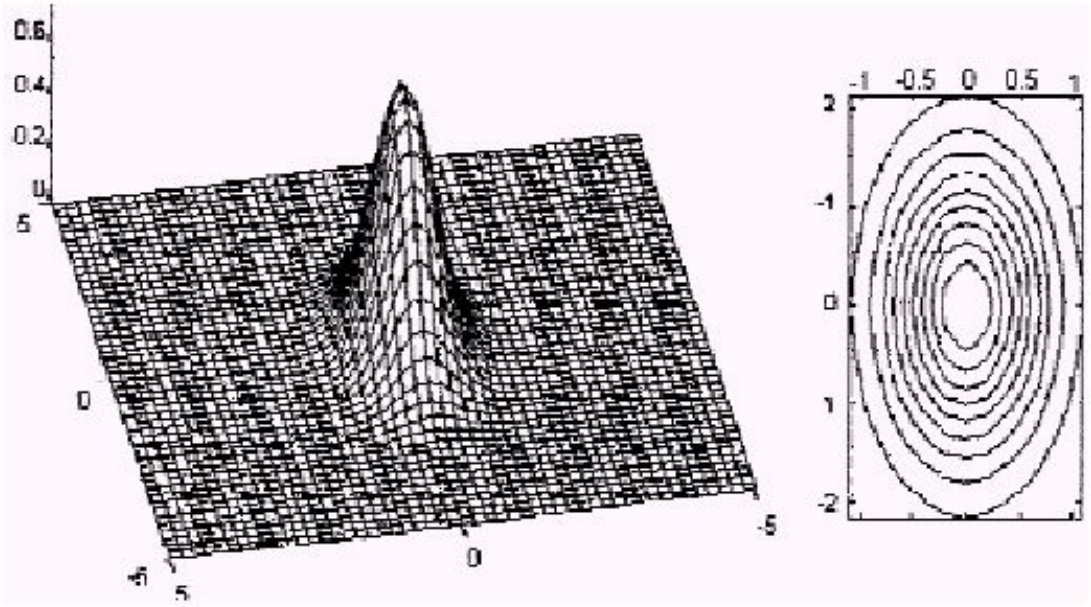
Рис. 12.8-а,б отражают случай $R^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ с $c=(0,0)^T$ и $c=(1,1)^T$ соответственно; рис. 12.8-в, г – случай $R^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$ и $c=(0,0)^T$, $c=(1,1)^T$; рис. 12.8-д – случай $R^{-1} = \begin{pmatrix} 2.0 & -1.5 \\ 1.5 & 2.0 \end{pmatrix}$ и $c=(0,0)^T$.



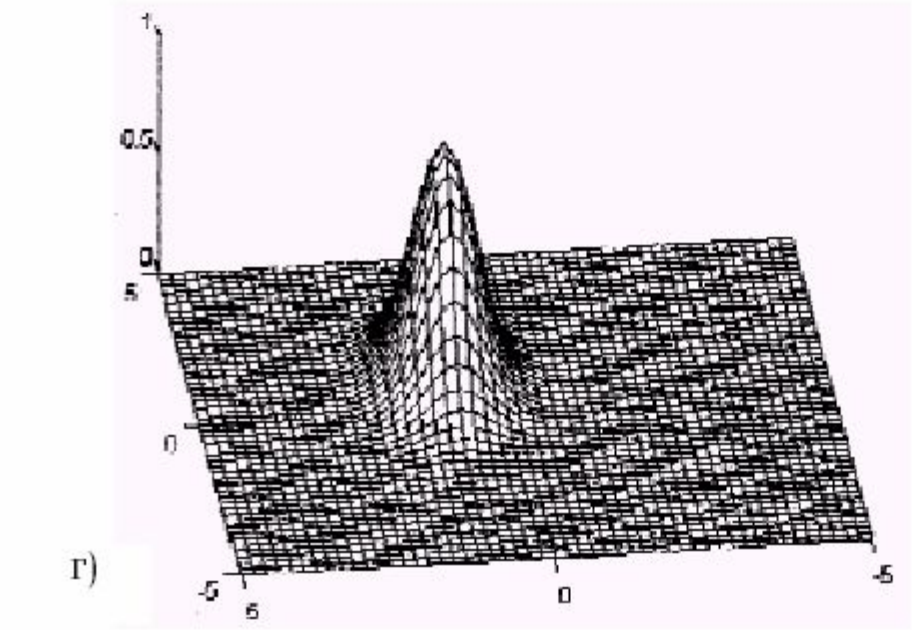
а)



б)



в)



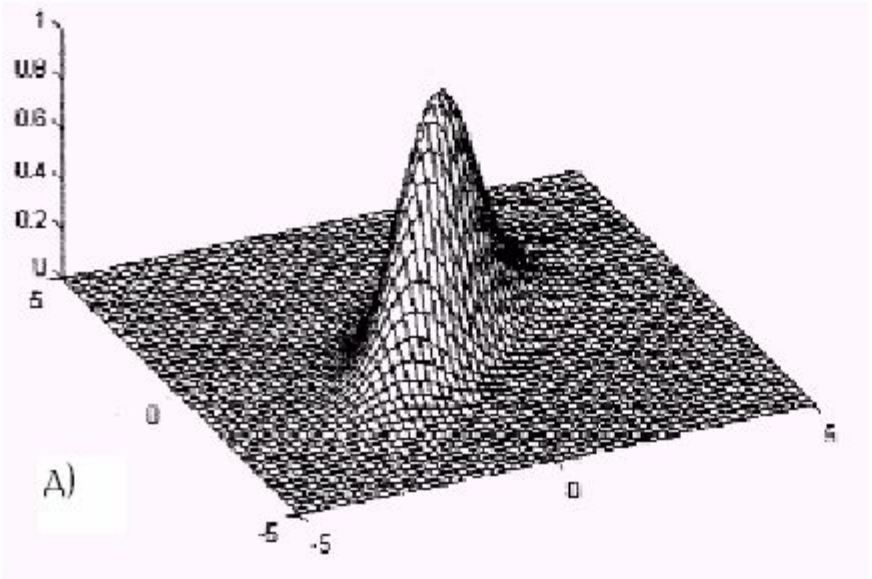


Рис.12.8 – Влияние выбора величин c и σ_{ij} на вид базисной функции

Величина сигнала j-ого нейрона выходного слоя y_j зависит от того, насколько близок предъявляемый входной сигнал x запомненному этим нейроном центру c_j. Значение y_j определяется как взвешенная сумма функций (12.1), т.е.

$$y_j = \sum_{i=1}^{P} f_i(x) w_{ij}. \qquad (12.12)$$

Обычно выходными сигналами сети являются нормализованные значения $\hat{y}_i$, вычисленные по формуле

$$\hat{y}_j = \frac{y_j}{\sum_{i=1}^{P} f_i(x)}. \qquad (12.13)$$

Пример 12.1. На рис.12.9 приведен пример аппроксимации функции $\varphi(x)$ (показана пунктирной линией) тремя базисными экспоненциальными функциями (показаны сплошными линиями)

$$\varphi(x) = \sum_{i=1}^{3} w_i e^{-(x-c_i)^2} = w^T f(x),$$

где $w = (0.5, -0.3, 0.4)^T$; $c_1 = 1; c_2 = 3; c_3 = 4$.

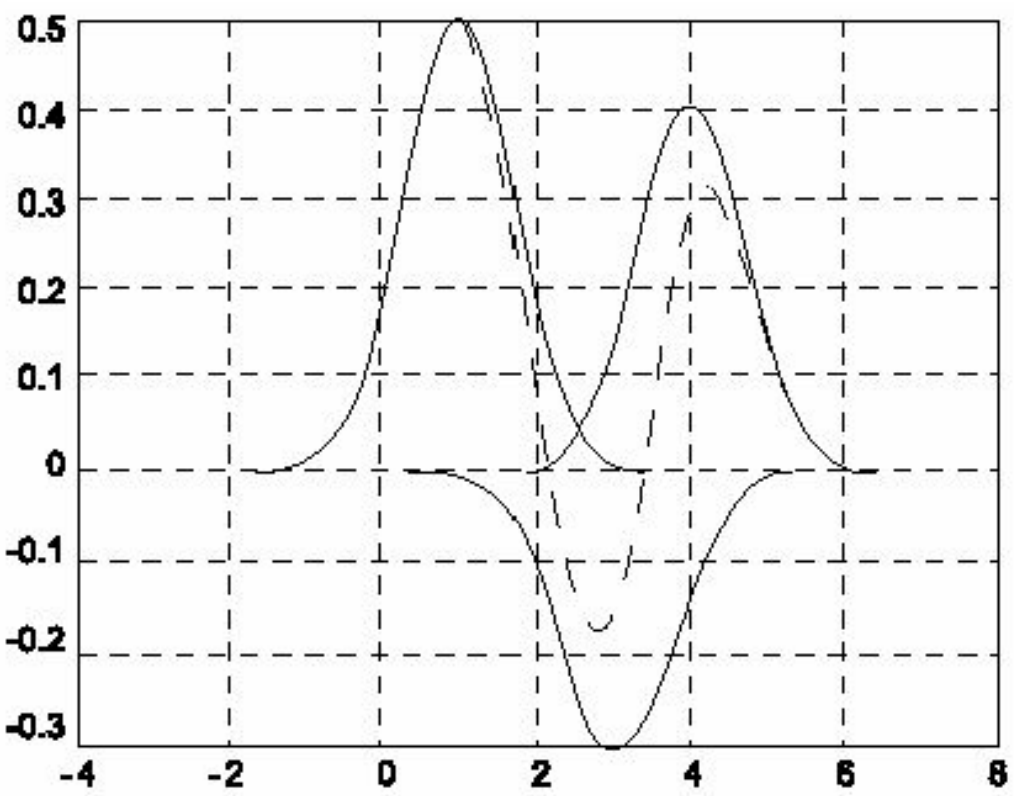


Рис.12.9 – Аппроксимация функции $\varphi(x)$

Таким образом, РБ сеть является нейросетевой реализацией конкретной аппроксимируемой функции, при которой каждому предъявляемому образу соответствует свой нейрон шаблонного слоя. Определение весовых параметров нейронов выходного слоя осуществляется путем решения системы линейных алгебраических уравнений, аналогичных (4.25)

$$y^* = Fw, \tag{12.14}$$

где $y^* = (y_1^*, y_2^*, ..., y_N^*)^T$ – вектор заданных значений аппроксимируемой функции в опорных точках $c_1, c_2, ..., c_N$; $f_{ij} = f\left(\left\|x_i - c_j\right\|\right)$ – базисные функции; $w = (w_1, w_2, ..., w_N)^T$ – вектор весовых параметров сети.

Как уже отмечалось, решение уравнения (12.14) имеет вид

$$w = F^+ y^*, \tag{12.15}$$

которое в случае квадратичной (NxN) матрицы F принимает известную форму оценки МНК.

$$w = \left[F^T F\right]^{-1} F^T y^*. \tag{12.16}$$

12.2 Обучение радиально-базисной сети

Как следует из вышеизложенного, РБ сеть характеризуют три типа параметров:

- *линейные весовые параметры* выходного слоя w_{ij} (входят в описание сети линейно);
- *центры* c_i - нелинейные (входят в описание нелинейно) параметры скрытого слоя;
- *отклонения (радиусы базисных функций)* σ_{ij} - нелинейные параметры скрытого слоя.

Обучение сети, состоящее в определении этих параметров, может сводится к одному из следующих вариантов:

1. Задаются центры и отклонения, а вычисляются только веса выходного слоя.
2. Определяются путем самообучения центры и отклонения, а для коррекции весов выходного слоя используется обучение с учителем.
3. Определяются все параметры сети с помощью обучения с учителем.

Первые два варианта применяются в сетях, использующих базисные функции с жестко заданным радиусом (отклонением). Третий же вариант, являясь наиболее сложным и трудоемким в реализации, предполагает использование любых базисных функций.

Таким образом, обучение сети заключается в том, что

- определяются центры c_i;
- выбираются параметры σ_i;
- вычисляются элементы матрицы весов W.

12.2.1 Выбор параметров центров и отклонений σ

Центры c_i определяют точки, через которые должна проходить аппроксимируемая функция. Так как большая обучающая выборка приводит к затягиванию процесса обучения, в РБ сетях широко используется кластеризация образов, при которой схожие векторы объединяются в *кластеры*, представляемые затем в процессе обучения только одним вектором. В настоящее время существует достаточно большое число эффективных алгоритмов кластеризации.

Использование кластеризации отражается на формулах (12.12), (12.13) следующим образом:

$$\hat{y}_i = \frac{\sum_{i=1}^{P} m_i f_i(x) w_{ij}}{\sum_{i=1}^{P} m_i f_i(x)}. \qquad (12.17)$$

Здесь m_i - число входных векторов в i-ом кластере.

В наиболее простом варианте алгоритм кластеризации, алгоритм k-среднего, направляет каждый образ в кластер, имеющий ближайший к данному образу центр. Если количество центров заранее задано или определено, алгоритм, обрабатывая на каждом такте входной вектор сети, формирует в пространстве входов сети центры кластеров. С ростом числа тактов эти центры сходятся к центрам данных. Кандидатами в центры являются все выходы скрытого слоя, однако в результате работы алгоритма будет сформировано подмножество наиболее существенных выходов.

Как уже указывалось, параметр σ_i, входящий в формулы для функций преобразования, определяет разброс относительно центра c_i. Варьируя параметры c_i и σ_i, пытаются перекрыть всё пространство образов, не оставляя пустот. Используя метод k-ближайших соседей, определяют k соседей центра c_i и, усредняя, вычисляют среднее значение $\hat{c}_i$. Величина отклонения $\hat{c}_i$ от c_i служит основанием для выбора параметра σ_i. На практике часто оправдывает себя выбор

$$\sigma = \frac{d}{\sqrt{2p}},$$

где $d = \max(c_i - c_k)$ – максимальное расстояние между выбранными центрами;
p – количество нейронов шаблонного слоя (образов).

Сами же весовые коэффициенты нейронов выходного слоя определяется из соотношений (12.15) либо (12.16).

Если качество аппроксимации является неудовлетворительным, выбор параметров c_i и σ, а также определение весов w повторяют до тех пор, пока полученное решение не окажется удовлетворительным.

12.2.2 Самообучение параметров центров

Настройка параметров центров может происходить методами, аналогичными методам *LVQ*, или с помощью *карт Кохонена*. При этом чаще всего все обучающие образы предварительно нормализуют, что позволяет использовать скалярное произведение входного вектора (образа) и вектора весов в качестве их меры соответствия. Компонентам весового вектора присваиваются случайные начальные значения. Коррекция компонентов вектора центров $c_i(k)$, $i = \overline{1,p}$, расположенного наиболее близко к предъявляемому обучающему образу x(k), происходит по алгоритму

$$c_i(k+1) = c_i(k) + \alpha(k)\delta_{ij}[x(k) - c_i(k)], i = \overline{1,p}, \qquad (12.18)$$

где $i = \arg\min_j \{d_j(k)\}$;

$d_j(k) = \| x(k) - c_j(k) \|$.

Изменяющийся во времени коэффициент усиления $\alpha(k) \in (0,1)$ задаётся соотношением

$$\alpha(k) = \frac{\alpha(k-1)}{\sqrt{1 + \mathrm{int}\left(\frac{k}{N+p+1}\right)}},$$

где int(x) - целая часть x.

Как показывают эксперименты, алгоритм (12.18) некритичен к выбору начального значения $\alpha(0)$, и хорошие результаты обеспечиваются выбором $\alpha(0) \approx 1$.

Более простым правилом задания изменения $\alpha(k)$ является следующее:

$$\alpha(k) = k^{-1}.$$

Вычисленные значения центров $c_i(k), i = \overline{1,p}$ используются затем для определения функции преобразования f_i и вычисления выходов сети (12.13).

При использовании самоорганизующихся карт Кохонена изменяются параметры центра не только нейрона-победителя шаблонного слоя, но и его соседей, находящихся в этом же слое. Использование как методов LVQ, так и карт Кохонена преследует цель такого размещения в пространстве векторов весов, чтобы их плотность распределения соответствовала плотности распределения обучающих образов.

12.2.3 Обучение сети с учителем

Как и в случае любого обучения с учителем, сети последовательно предъявляются обучающие пары $(x_i, y_i^*) i = \overline{1,k}$, где x_i и y_i^* - входной и желаемый выходной векторы i-ой обучающей пары соответственно. На выходе сети появляется вектор $y_i = (y_{1i}, y_{2i}, ..., y_{Li})^T$. Этот вектор сравнивается с требуемым y_i^* и вычисляется рассогласование $(y_i - y_i^*)$, используемое в заранее выбранном выпуклом функционале качества, минимизация которого дает требуемые значения элементов весовой матрицы.

Обычно в качестве минимизируемого функционала принимается квадратичный вида

$$I = \sum_{i=1}^{k} (y_i^* - y_i)^T (y_i^* - y_i) = \sum_{i=k}^{k} (y_i^* - Wf(x_i))^T (y_i^* - Wf(x_i)). \qquad (12.19)$$

Так все пары (x_i, y_i^*) $i = \overline{1,k}$ образуют матрицы входных сигналов X(k) размерности $N \times k$ и матрицы требуемых выходных сигналов $Y^*(k)$ размерности $L \times k$, РБ сеть описывается следующим матричным уравнением:

$$Y(k) = WF(k), \qquad (12.20)$$

где $Y(k) - (L \times k)$ матрица выходных сигналов; $W - (L \times p)$ - матрица весов; $F(k) - (p \times k)$ - матрица используемых в сети функций преобразования, а критерий (12.19) может быть записан так

$$I = tr[(Y(k) - WF(k))^T (Y(k) - WF(k))]. \qquad (12.21)$$

Здесь tr[·] - след матрицы [•].

Заметим, что (12.21) является обобщением (12.14) на случай k образов. Минимизация (12.21), осуществляемая путём решения системы линейных алгебраических уравнений

$$\frac{\partial I}{\partial W_{ij}} = 0, \qquad (12.22)$$

приводит после предъявления k обучающих пар к получению оценки МНК, аналогичной (12.15)

$$W = F^{+}(k)Y(k) \qquad (12.23)$$

или

$$W(k) = Y(k)F^{T}(k)[F(k)F^{T}(k)]^{-1}. \qquad (12.24)$$

При предъявлении новой обучающей пары $\left\{x_{k+1}, y^{*}_{k+1}\right\}$ оценка матрицы весовых коэффициентов по аналогии (12.24) записывается в виде

$$W(k+1) = Y(k+1)F^{T}(k+1)[F(k+1)F^{T}(k+1)]^{-1} \qquad (12.25)$$

Так как размерности матриц $Y(k+1)$ и $F(k+1)$ увеличиваются, усложняется процедура вычисления обратной матрицы и возрастает общее количество арифметических операций, необходимых для вычисления оценки (12.25).

Существенного упрощения можно достичь, если учесть, что используемые в (12.24) матрицы формируются следующим образом:

$$Y(k+1) = \left[Y(k) : y_{k+1}\right], \quad F(k+1) = \left[F(k) : f_{k+1}\right], \tag{12.26}$$

т.е. являются блочными.

Предположим, что матрица $\left[F(k+1)F^T(k+1)\right]$ невырождена. Обозначим

$$P^{-1}(k) = F(k)F^T(k) = \sum_{i=1}^{k} f_i f_i^T . \tag{12.27}$$

Тогда с учетом (12.26), (12.27)

$$P^{-1}(k+1) = F(k+1)F^T(k+1) = \sum_{i=1}^{k+1} f_i f_i^T = \sum_{i=1}^{k} f_i f_i^T + f_{k+1} f_{k+1}^T = P^{-1}(k) + f_{k+1} f_{k+1}^T . \tag{12.28}$$

Применение к (12.28) леммы об обращении матриц [68] дает

$$P(k+1) = P(k) - P(k) f_{k+1} \left(1 + f_{k+1}^T P(k) f_{k+1}\right)^{-1} f_{k+1}^T P(k). \tag{12.29}$$

Подстановка в (12.25) соотношений (12.26) и (12.28) и несложные преобразования приводят к следующей формуле:

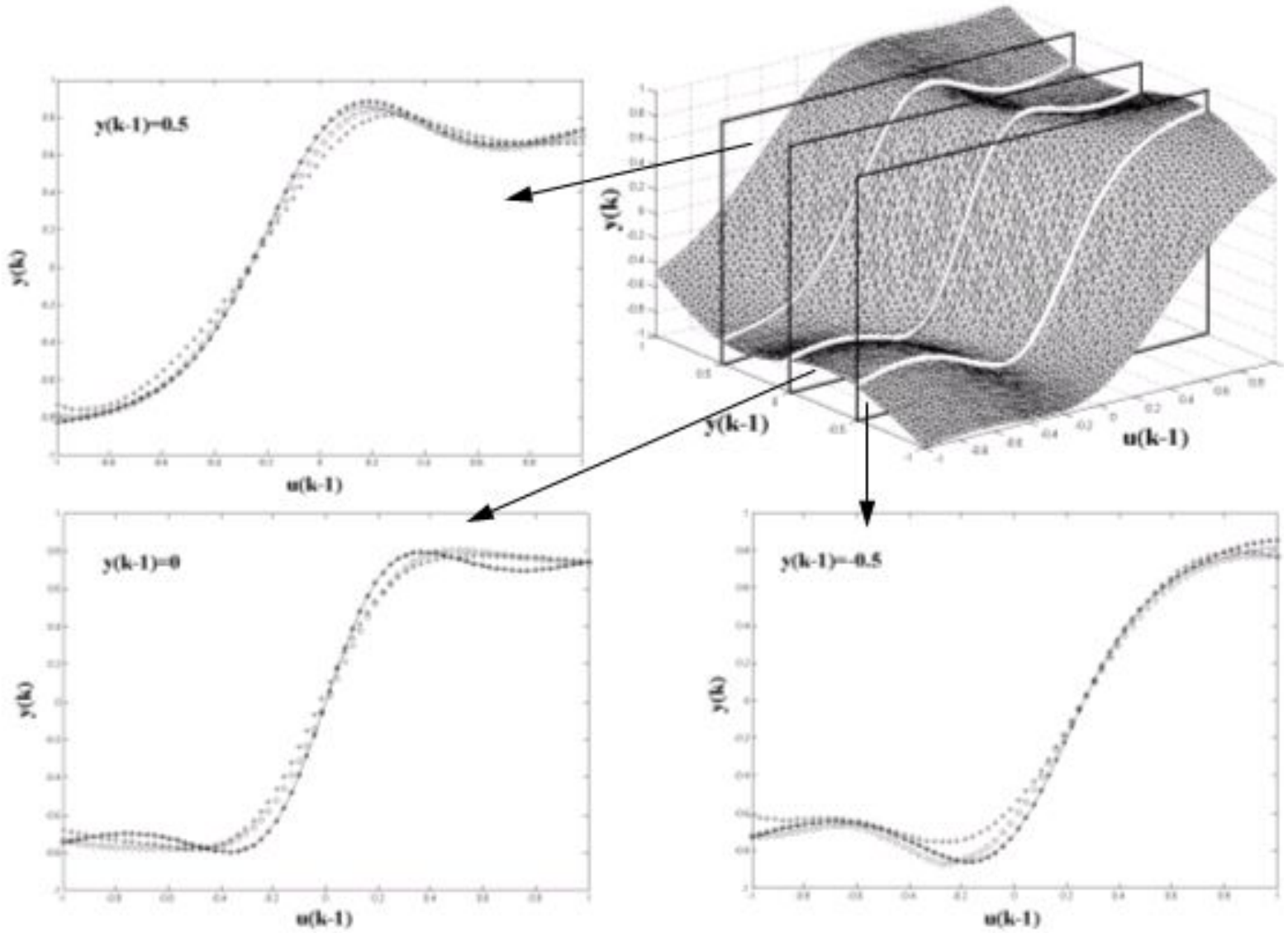
$$W(k+1) = W(k) + \left[y(k+1) - W(k) f_{k+1}\right] f_{k+1}^T P(k) \tag{12.30}$$

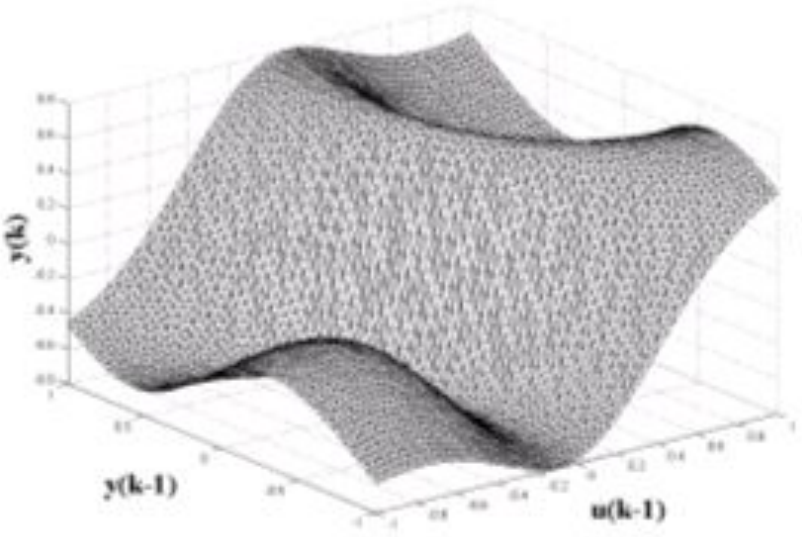
Моделирование. Рассматривалась задача идентификации нелинейного динамического объекта, описываемого уравнением [12]

$$y(k)=0.725\beta(k)\sin\left(\frac{16u(k-1)+8y(k-1)}{\beta(k)\left(3+4u^2(k-1)+4y^2(k-1)\right)}\right)+ \quad (19)$$
$$+0.2u(k-1)+0.2y(k-1),$$

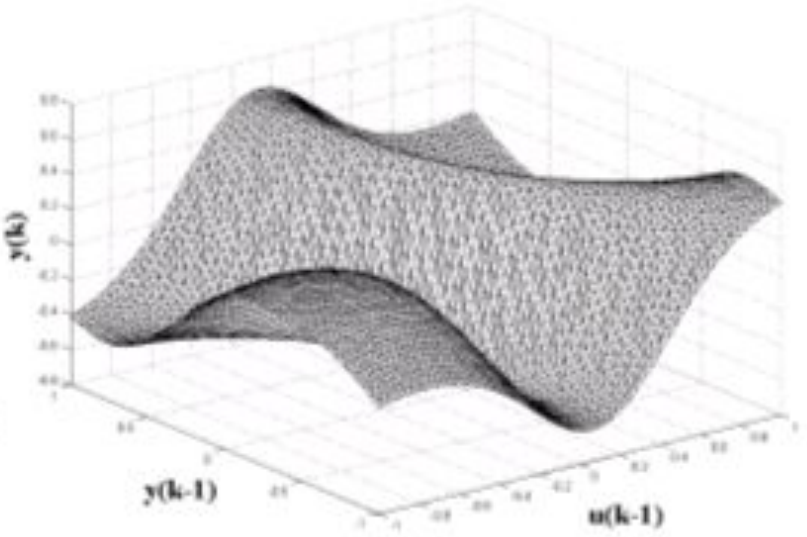
где $\beta(k)$ – изменяемый во времени параметр, задающий степень нестационарности объекта.

Входной сигнал $u(k)$ представлял собой стационарную случайную последовательность с равномерным законом распределения в интервале [-1, 1], генерируемую датчиком случайных чисел. При исследовании стационарного случая использовалось 5000 обучающих пар и задавались значения $\varepsilon=0.01$, $\rho = 0.4$. Кроме того, начальное значение параметра смещения c_0 (см. рис. 1) принималось равным $c_0 = 0.001$.

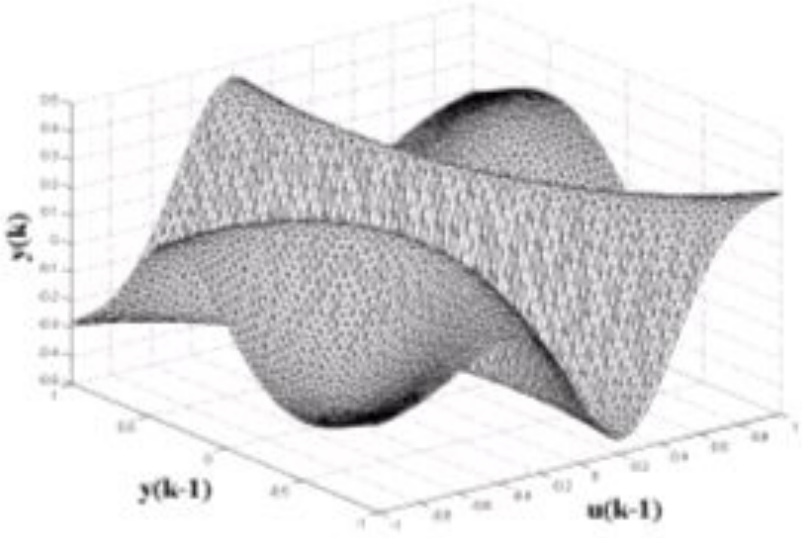

y(k-1)=0.5
y(k)
u(k-1)
y(k)
y(k-1)
u(k-1)
y(k-1)=0
y(k)
u(k-1)
y(k-1)=-0.5
y(k)
u(k-1)



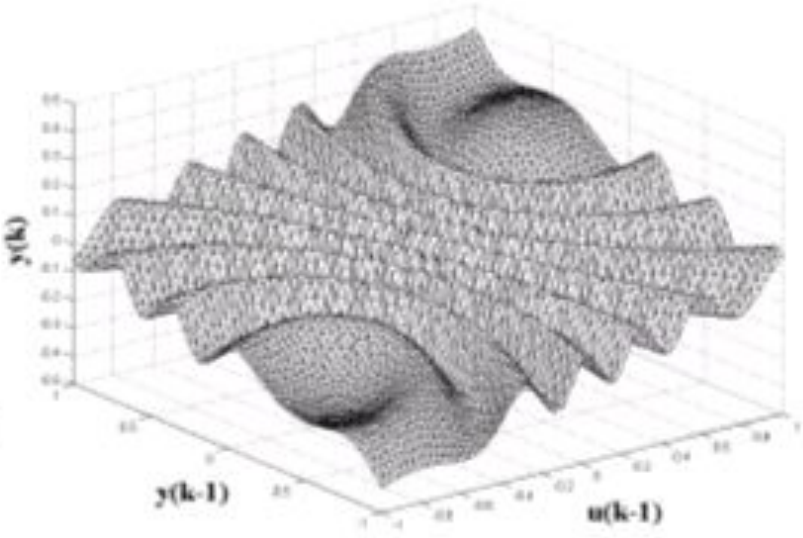
$\beta(k)=0.8$



$\beta(k)=0.6$



$\beta(k)=0.4$



$\beta(k)=0.1$

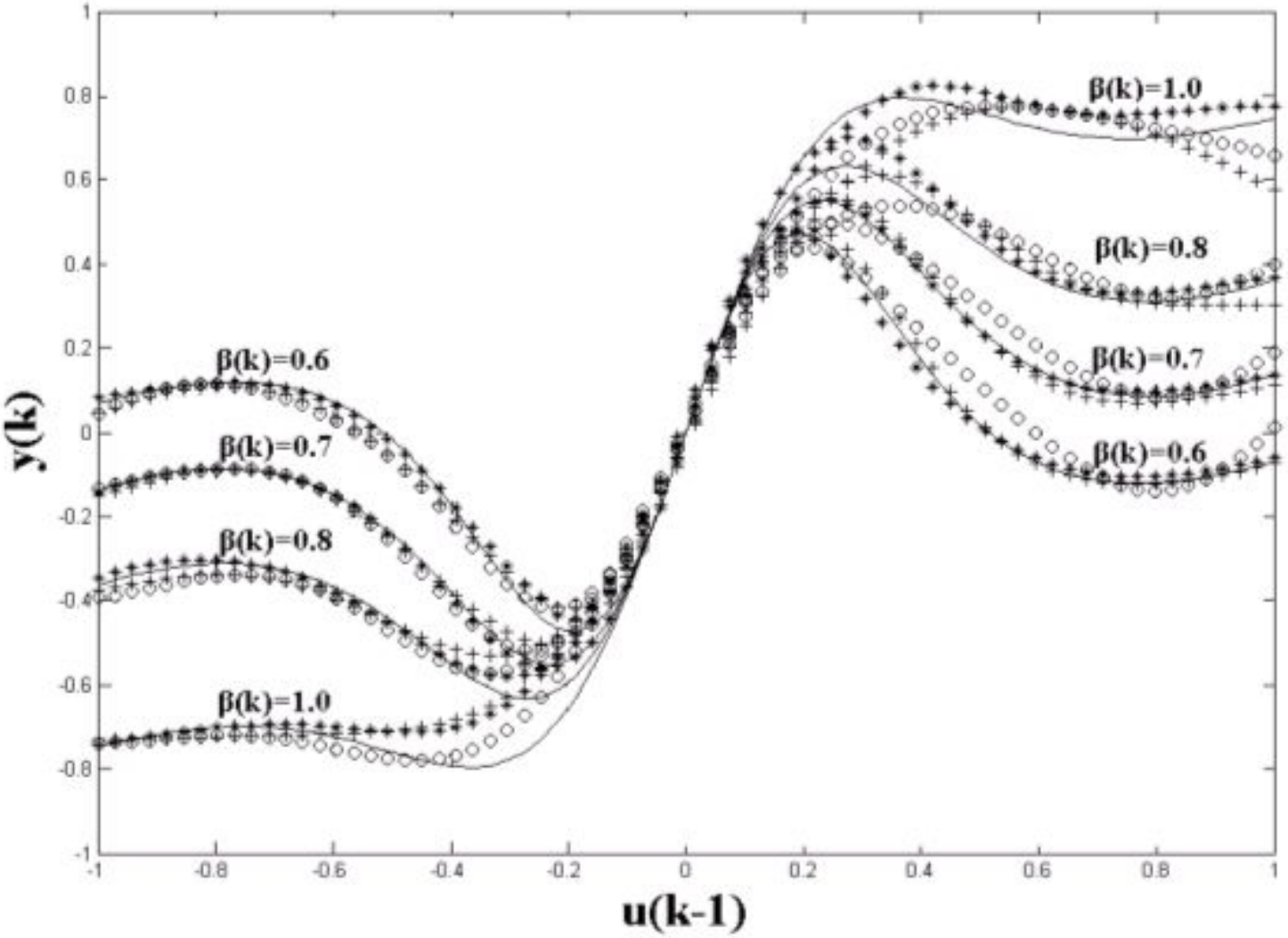


Abb. 5: Identifikationsergebnisse für ein instationäres System

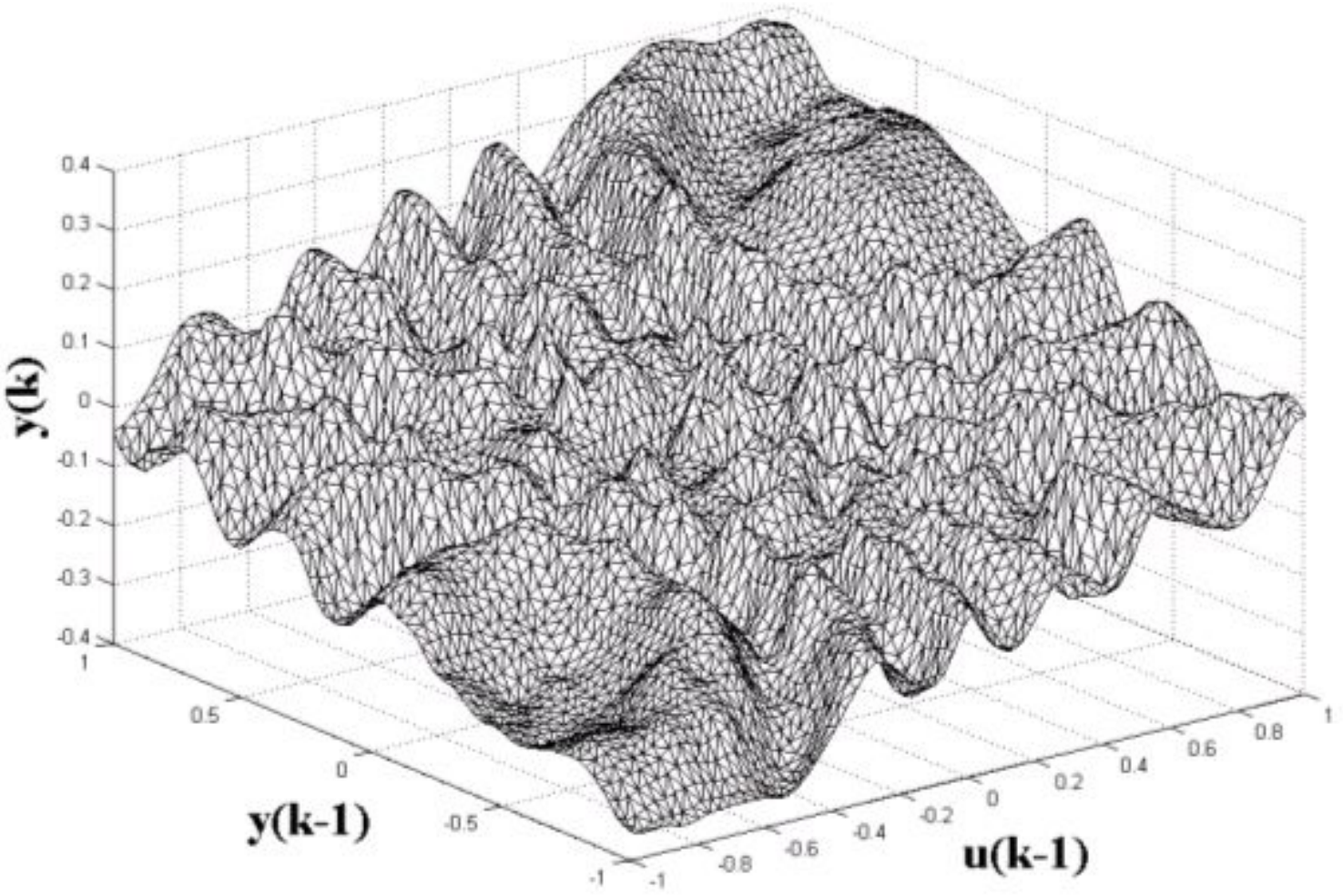


Abb. 6: Geschätzte Wirkungsfläche des Fünfschrittalgorithmus

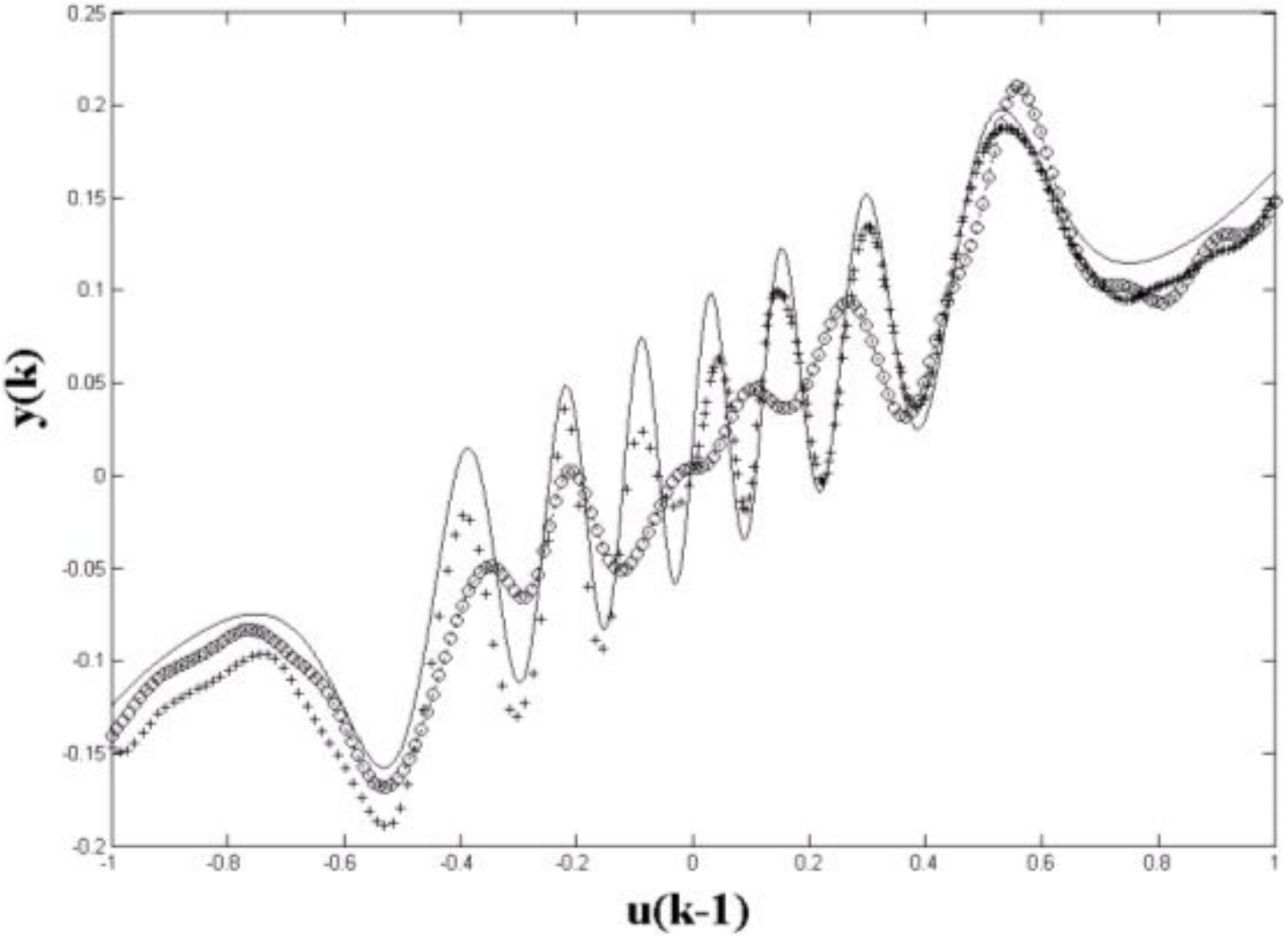


Abb. 7: Ergebnisse des Trainings des Netzes

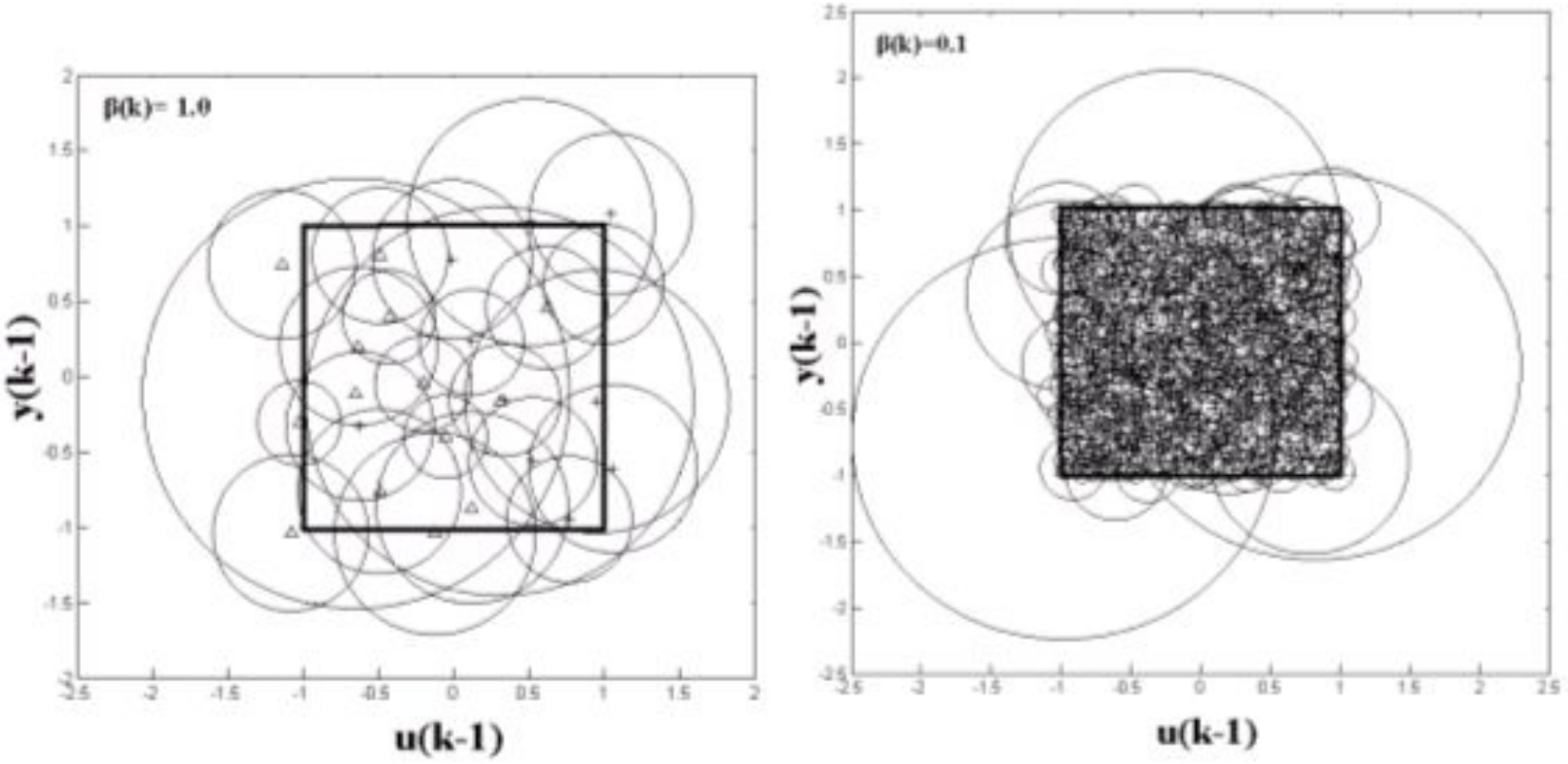
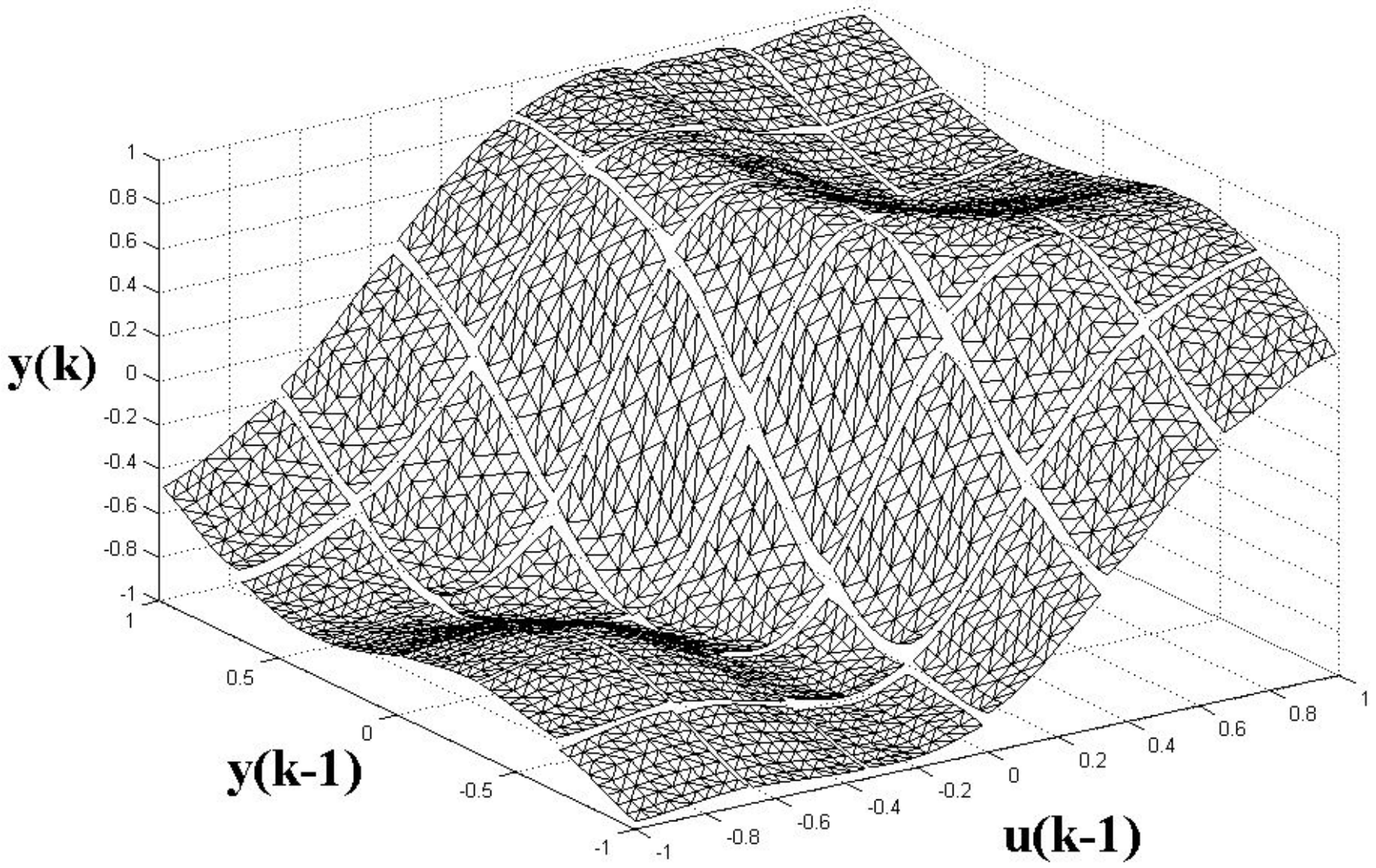
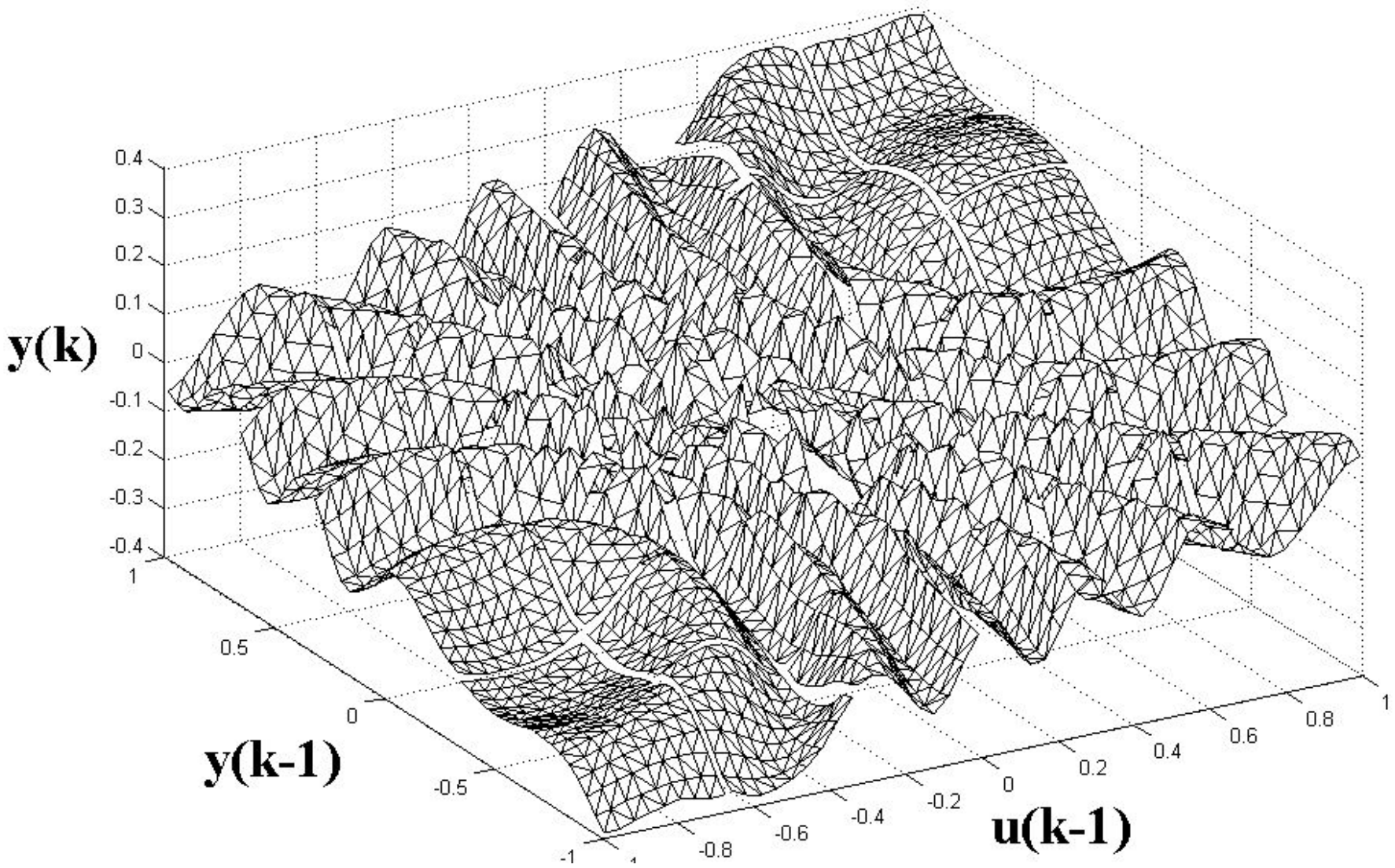
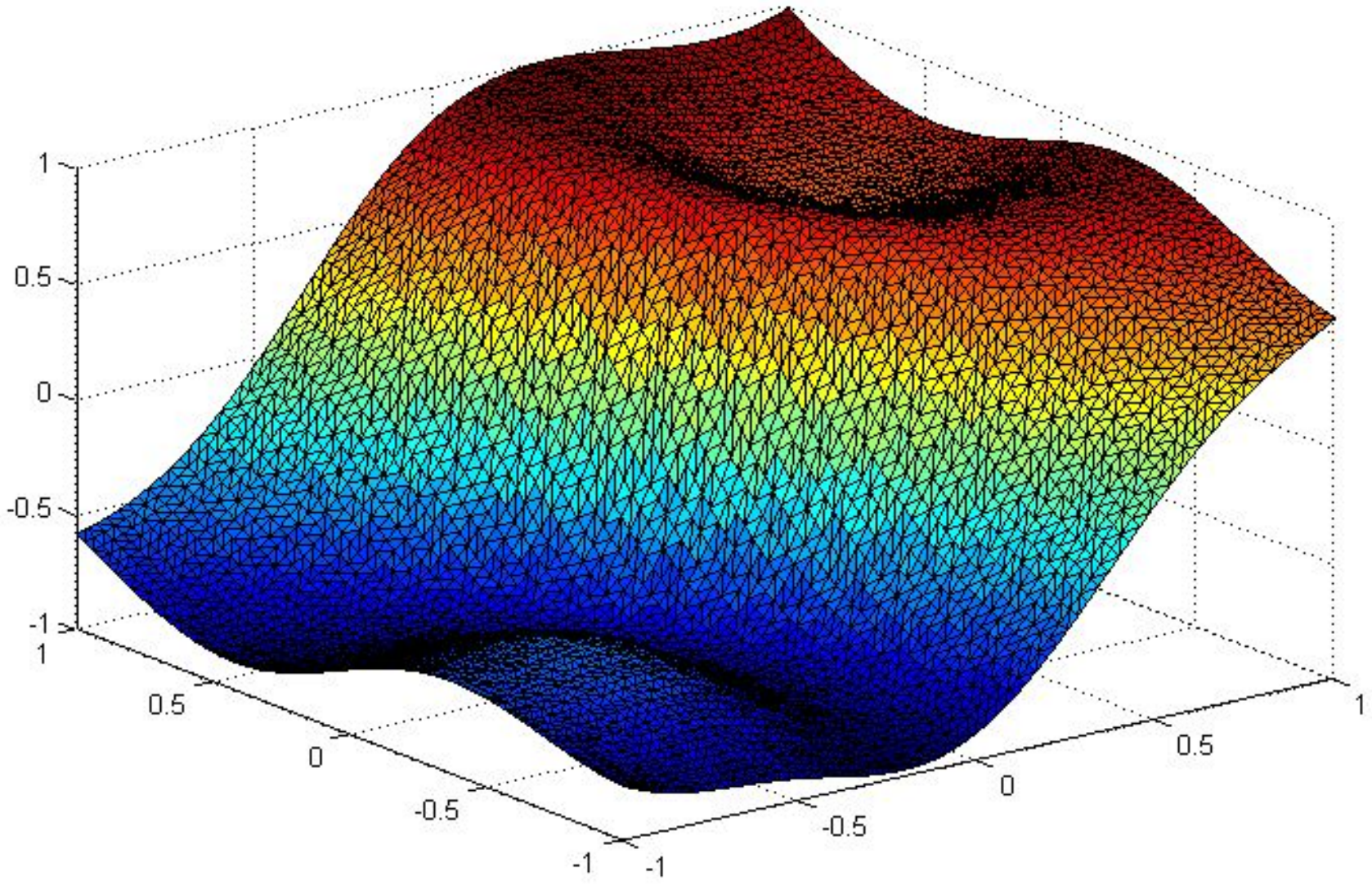
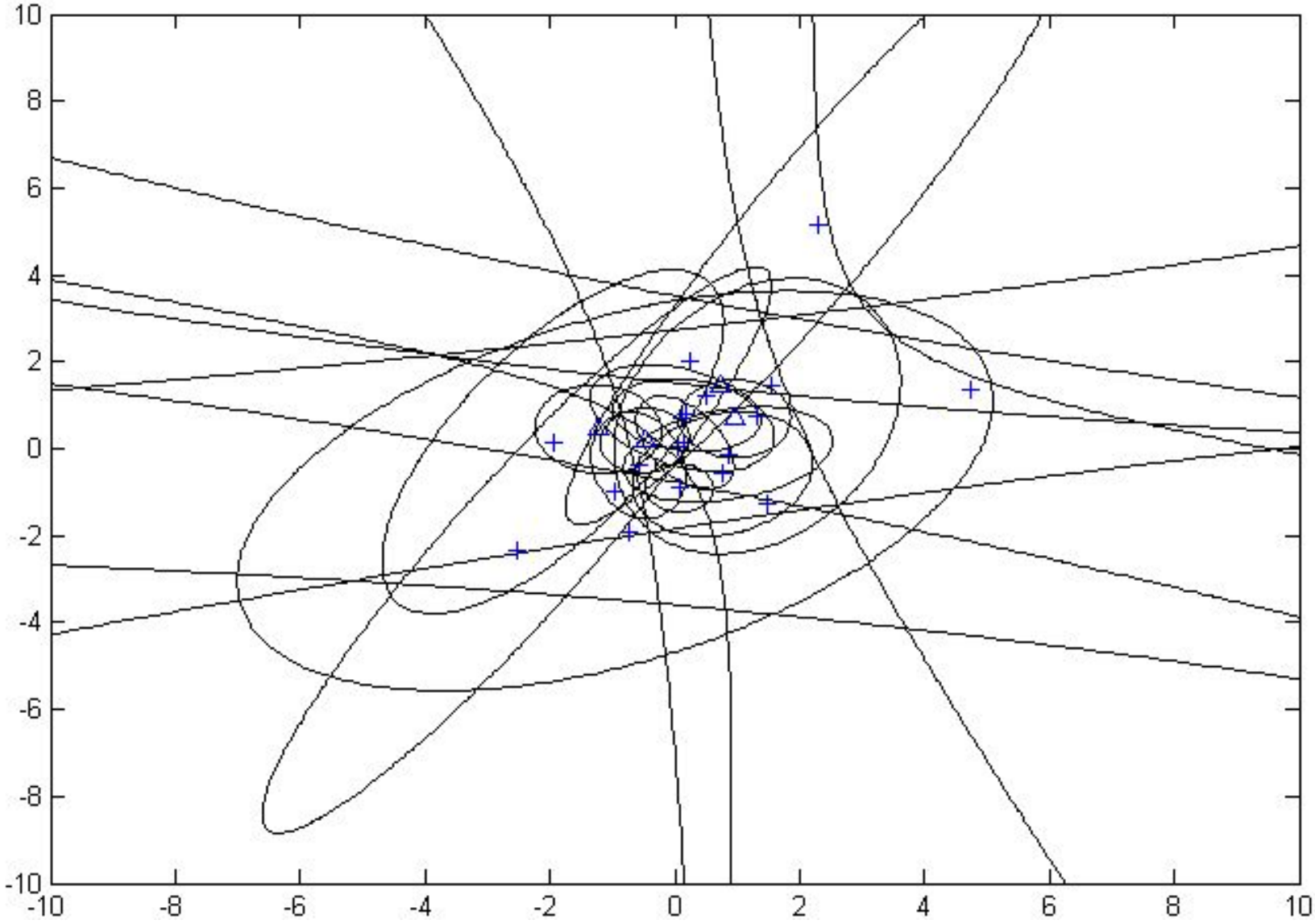


Abb. 8: Endgültige Lage der Neuronen


y(k)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
y(k-1)
1
0.5
0
-0.5
-1
u(k-1)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


y(k)
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
y(k-1)
1
0.5
0
-0.5
-1
u(k-1)
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1


10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10