Лабораторная работа №1
Линейная искусственная нейронная сеть.
Правило обучения Видроу-Хоффа.

Цель работы: Изучить обучение и функционирование линейной ИНС при решении задач прогнозирования.

1.Правило обучения Видроу-Хоффа

Используется для обучения нейронной сети, состоящей из распределительных нейронов и одного выходного нейрона, который имеет линейную функцию активации (рис.1):

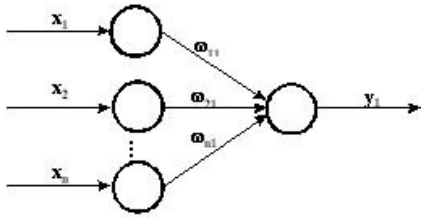

Рис. 1.Линейная сеть.

Такая сеть называется адаптивным нейронным элементом или "ADALINE" (Adaptive Linear Element). Его предложили в 1960 г. Видроу (Widrow) и Хофф (Hoff). Выходное значение такой сети определяется, как

$$y_1 = \sum_{j=1}^{n} \omega_{j1} x_j - T. \qquad (1.2)$$

Правило обучения Видроу Хоффа известно под названием ***дельта правила*** (delta rule). Оно предполагает минимизацию среднеквадратичной ошибки нейронной сети, которая для *L* входных образов определяется следующим образом:

$$E = \sum_{k=1}^{L} E(k) = \frac{1}{2} \sum_{k=1}^{L} (y_1^k - t^k)^2, \qquad (1.3)$$

где $E(k)$ — среднеквадратичная ошибка сети для *k*-го образа; y_1^k и t^k — соответственно выходное и эталонное значение нейронной сети для *k*-го образа.

Критерий (1.3) характеризуется тем, что при малых ошибках ущерб является также малой величиной, т. к. E меньше чем величина отклонения $(y - t)$. При больших ошибках ущерб возрастает, так как E возрастает с ростом величины ошибки.

Среднеквадратичная ошибка нейронной сети для одного входного образа определяется, как

$$E(k) = \frac{1}{2} (y_1^k - t^k)^2. \qquad (1.4)$$

Правило обучения Видроу-Хоффа базируется на методе градиентного спуска в пространстве весовых коэффициентов и порогов нейронной сети. Согласно этому правилу, весовые коэффициенты и пороги нейронной сети необходимо изменять с течением времени по следующим выражениям:

$$\omega_{j1}(t+1) = \omega_{j1}(t) - \alpha \frac{\partial E(k)}{\partial \omega_{j1}(t)}, \quad (1.5)$$

$$T(t+1) = T(t) - \alpha \frac{\partial E(k)}{\partial T(t)}, \quad (1.6)$$

где $j = \overline{1,n}$; α — скорость или шаг обучения. Найдем производные

среднеквадратичной ошибки E по настраиваемым параметрам сети ω_{j1} и T. Тогда

$$\frac{\partial E}{\partial \omega_{j1}(t)} = \frac{\partial E}{\partial y_1^k} \cdot \frac{\partial y_1^k}{\partial \omega_{j1}} = (y_1^k - t^k) x_i^k,$$

$$\frac{\partial E}{\partial T(t)} = \frac{\partial E}{\partial y_1^k} \cdot \frac{\partial y_1^k}{\partial T} = -(y_1^k - t^k),$$

где x_j^k — j-ая компонента k-го образа.

Отсюда получаем следующие выражения для обучения нейронной сети по дельта правилу:

$$\omega_{j1}(t+1) = \omega_{j1}(t) - \alpha(y_1^k - t^k) x_i^k, \quad (1.7)$$

$$T(t+1) = T(t) + \alpha(y_1^k - t^k), \quad (1.8)$$

где $j = \overline{1,n}$.

Видроу и Хофф доказали [18], что данный закон обучения всегда позволяет находить весовые коэффициенты нейронного элемента таким образом, чтобы минимизировать среднеквадратичную ошибку сети независимо от начальных значений весовых коэффициентов.

Алгоритм обучения, в основе которого лежит дельта правило состоит из следующих шагов:

1. Задается скорость обучения $\alpha\ (0 < \alpha < 1)$ и минимальная среднеквадратичная ошибка сети E_m, которой необходимо достичь в процессе обучения.

2. Случайным образом инициализируются весовые коэффициенты и порог нейронной сети.

3. Подаются входные образы на нейронную сеть и вычисляются векторы выходной активности сети.

4. Производится изменение весовых коэффициентов и порога нейронной сети согласно выражениям (1.7) и (1.8).

5. Алгоритм продолжается до тех пор, пока суммарная среднеквадратичная ошибка сети не станет меньше заданной, т. е. $E \le E_m$.

В алгоритме Видроу-Хоффа существует проблема выбора значения шага обучения α. Если коэффициент α слишком мал, то процесс обучения является очень длительным. В случае, когда шаг обучения большой, процесс обучения может оказаться расходящимся, то есть не привести к решению задачи. Таким образом сходимость алгоритма обучения не избавляет от разумного выбора значения шага обучения.

1.13. Использование линейной нейронной сети для прогнозирования

Способность нейронных сетей после обучения к обобщению и пролонгации результатов создает потенциальные предпосылки для построения на базе их различного рода прогнозирующих систем. В данном разделе рассмотрим прогнозирование временных рядов при помощи линейных нейронных сетей. Пусть дан временной ряд $x(t)$ на промежутке $t = \overline{1,m}$. Тогда задача прогнозирования состоит в том, чтобы найти продолжение временного ряда на неизвестном промежутке, т. е. необходимо определить $x(m+1)$, $x(m+2)$ и так далее (рис. 1.2):

Совокупность известных значений временного ряда образуют обучающую выборку, размерность которой равняется m. Для прогнозирования временных рядов используется метод

"скользящего окна". Он характеризуется длиной окна n, которая равняется количеству элементов ряда, одновременно подаваемых на нейронную сеть. Это определяет структуру нейронной сети, которая состоит из n распределительных нейронов и одного выходного нейрона.

Такая модель соответствует линейной авторегрессии и описывается следующим выражением:

$$\overline{x(n)} = \sum_{k=1}^{n} \omega_k x(p - n + k - 1),$$

где $\omega_k, k = \overline{1,n}$ — весовые коэффициенты нейронной сети; $\overline{x(p)}$ — оценка значения ряда $x(p)$ в момент времени p.

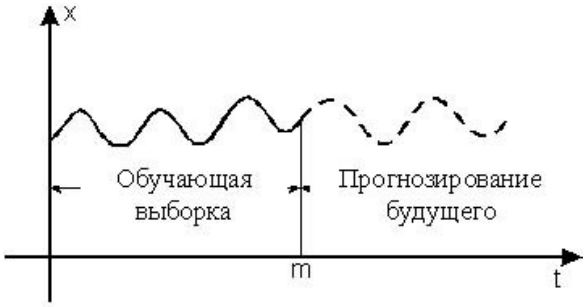

Рис. 1.2. Прогнозирование временного ряда.

Ошибка прогнозирования определяется, как

$$e(p) = x(p) - \overline{x(p)}.$$

Модель линейной авторегрессии формирует значение ряда $x(p)$, как взвешенную сумму предыдущих значений ряда. Обучающую выборку нейронной сети можно представить в виде матрицы, строки которой характеризуют векторы, подаваемые на вход сети:

$$X = \begin{bmatrix} x(1) & x(2) & \dots & x(n) \\ x(2) & x(3) & \dots & x(n-p) \\ \dots & \dots & \dots & \dots \\ x(m-n) & x(m-n+1) & \dots & x(m-1) \end{bmatrix}.$$

Это эквивалентно перемещению окна по ряду $x(t)$ с единичным шагом.

Таким образом для обучения нейронной сети прогнозированию используется выборка известных членов ряда. После обучения сеть должна прогнозировать временной ряд на упреждающий промежуток времени.

Задание.

1. Написать на любом ЯВУ программу моделирования прогнозирующей линейной ИНС. Для тестирования использовать функцию

$$y = a\sin(bx) + d \ .$$

Варианты заданий приведены в следующей таблице:

№ варианта	a	b	d	Кол-во входов ИНС
1	1	5	0.1	3
2	2	6	0.2	4
3	3	7	0.3	5

4	4	8	0.4	3
5	1	9	0.5	4
6	2	5	0.6	5
7	3	6	0.1	3
8	4	7	0.2	4
9	1	8	0.3	5
10	2	9	0.4	3
11	3	5	0.5	4

Обучение и прогнозирование производить на 30 и 15 значениях соответственно табулируя функцию с шагом 0.1. Скорость обучения выбирается студентом самостоятельно, для чего моделирование проводится несколько раз для разных α. Результаты оцениваются по двум критериям - скорости обучения и минимальной достигнутой ошибке. Необходимо заметить, что эти критерии в общем случае являются взаимоисключающими, и оптимальные значения для каждого критерия достигаются при разных α.

2. Результаты представить в виде отчета содержащего:

1. Титульный лист,
2. Цель работы,
3. Задание,
4. Результаты обучения: таблицу со столбцами: эталонное значение, полученное значение, отклонение; график изменения ошибки в зависимости от итерации.
5. Результаты прогнозирования: таблицу со столбцами: эталонное значение, полученное значение, отклонение.
6. Выводы по лабораторной работе.

Результаты для пунктов 3 и 4 приводятся для значения α, при котором достигается минимальная ошибка. В выводах анализируются все полученные результаты.

Контрольные вопросы.

1. ИНС какой архитектуры Вы использовали в данной работе? Опишите принцип построения этой ИНС.
2. Как функционирует используема Вами ИНС?
3. Опишите (в общих чертах) алгоритм обучения Вашей ИНС.
4. Как формируется обучающая выборка для решения задачи прогнозирования?
5. Как выполняется многошаговое прогнозирование временного ряда?
6. Предложите критерий оценки качества результатов прогноза.