Лабораторная работа №2
Линейная искусственная нейронная сеть.
Правило обучения Видроу-Хоффа

Цель работы: изучить процессы обучения и функционирования линейной искусственной нейронной сети (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 (рисунок 1):

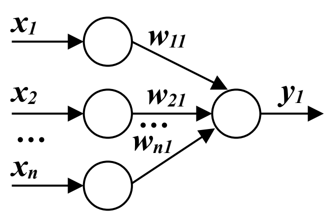

Рисунок 1 – Линейная нейронная сеть

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

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

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:

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

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 (1.2) 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

$$E(k) = \frac{1}{2}(y_1^k - t^k)^2. \tag{3}$$

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:

$$\omega_{j1}(t+1) = \omega_{j1}(t) - \alpha \frac{\partial E(k)}{\partial \omega_{j1}(t)}, \tag{4}$$

$$T(t+1) = T(t) - \alpha \frac{\partial E(k)}{\partial T(t)}, \tag{5}$$

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:

$$\omega_{j1}(t+1) = \omega_{j1}(t) - \alpha(y_1^k - t^k)x_i^k, \tag{6}$$

$$T(t+1) = T(t) + \alpha(y_1^k - t^k), \tag{7}$$

где $j = \overline{1,n}$.

Видроу и Хофф доказали, 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. Для каждого образа из обучающей выборки выполняются следующие действия:

3.1. Подаётся входной образ на нейронную сеть и вычисляется вектор выходной активности сети.

3.2. Производится изменение весовых коэффициентов и порога нейронной сети согласно выражениям (6) и (7).

4. Шаг 3 алгоритма 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.

2. Использование линейной нейронной сети для прогнозирования временных рядов

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 (рисунок 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.

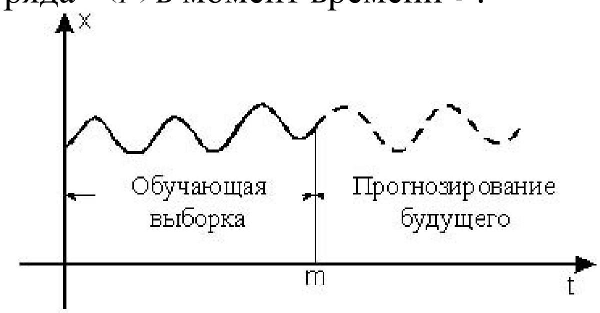

Рисунок 2 – Процесс прогнозирования временного ряда

Ошибка прогнозирования определяется следующим образом:

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

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.

Пример. Пусть исходный временной ряд (см. рисунок 2) состоит из пяти наблюдений: 1, 2, 3, 5, 8, 13, а размер окна (количество входных нейронов) n=3. Тогда обучающие эталоны будут иметь вид:

- эталон 1: входной вектор (1, 2, 3), желаемый выход 5;
- эталон 2: входной вектор (2, 3, 5), желаемый выход 8;
- эталон 3: входной вектор (3, 5, 8), желаемый выход 13.

Таким образом, обучающая выборка состоит из трех эталонов. После обучения, для осуществления первого прогноза на вход сети подается образ (5,8,13); на выходе будет рассчитан прогноз *x'(1)*. Для осуществления второго прогноза на вход сети подается образ (8,13,*x'(1)*); на выходе будет рассчитан второй прогноз *x'(2)*. Процесс повторяется для осуществления нужного количества шагов прогноза.

3. Задание

3.1. Изучить теоретические сведения

3.2. Написать на любом языке высокого уровня программу моделирования прогнозирующей линейной ИНС. Для тестирования использовать функцию $y = a \cdot sin(bx) + d$. Варианты заданий приведены в таблице 1.

Таблица 1 – варианты заданий для построения нейронной сети и генерации временного ряда

№ 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
12	1	6	0.1	5
13	2	7	0.2	3
14	3	8	0.3	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 значений α. Результаты оцениваются по двум критериям – 1) скорости обучения и 2) минимальной достигнутой среднеквадратичной ошибке. Необходимо заметить, что эти критерии в общем случае являются взаимоисключающими, и оптимальные значения для каждого критерия достигаются при разных значениях α.

3.3. Результаты представить в виде отчета содержащего:
1) Титульный лист
2) Цель работы
3) Задание
4) Текст программы
5) Результаты обучения:
- таблицу следующего содержания (можно в виде скриншотов)

эталонное значение временного ряда	полученное значение временного ряда	среднеквадратичное отклонение

- график изменения среднеквадратичной ошибки в зависимости от итерации.

6) Результаты прогнозирования:
- таблицу следующего содержания (можно в виде скриншотов)

номер шага прогнозирования	эталонное значение временного ряда	спрогнозированное значение временного ряда	среднеквадратичное отклонение

- график, содержащий кривую прогнозирования и кривую эталонного временного ряда.

7) Выводы по лабораторной работе.

Примечание: результаты для пунктов 5 и 6 приводятся для значения α, при котором достигается минимальная ошибка. В выводах анализируются все полученные результаты.

3.4. Подготовиться к защите работы по полученным результатам и контрольным вопросам

4. Контрольные вопросы

4.1. ИНС какой архитектуры использована в данной работе? Опишите принцип построения этой ИНС.

4.2. Как функционирует используемая в работе ИНС?

4.3. Сформулируйте правила нахождения градиентов ошибок $\frac{\partial E}{\partial \omega_{j1}(t)}$, $\frac{\partial E}{\partial T(t)}$ в алгоритме обучения используемой ИНС.

4.4. Сформулируйте пошагово алгоритм обучения используемой ИНС.

4.5. Как формируется обучающая выборка для решения задачи прогнозирования?

4.6. Как выполняется многошаговое прогнозирование временного ряда?

4.7. Предложите критерий оценки качества результатов прогноза.

Литературные источники

1. Головко, В. А. Нейросетевые технологии обработки данных : учеб. пособие / В. А. Головко, В. В. Краснопрошин. − Минск : БГУ, 2017. − 263 с.