МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
«Брестский государственный технический университет»

Кафедра ЭВМ и систем

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ по курсу
«Интеллектуальные системы принятия решений»
для студентов специальности Т.10.01.00

ЧАСТЬ 1

Брест 2001

УДК 681.3

Методические указания предназначены для выполнения лабораторно-практических работ по дисциплине "Интеллектуальные системы принятия решений" и содержат описания двух лабораторных работ по изучению теоретических основ систем принятия решений. Являются первой частью лабораторно-практического курса по изучению современных методов обработки реальных, неполных и противоречивых данных и способов автоматизированного принятия решений на их основе. Задания, требования к выполнению работ и содержанию отчетов разработаны с учетом имеющейся на кафедре ЭВМ и систем ВЦ аппаратно-технической базы.

Методические указания предназначены для использования студентами специальности Т.10.01.00 «Автоматизированные системы обработки информации».

Ил. 10., табл. 6., список лит. – 1 назв.

ОГЛАВЛЕНИЕ

1. КЛАССИФИКАЦИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Нейронные сети можно классифицировать в зависимости от различных характеристик:

1) *По типу входной информации* различают:

1.1) Аналоговые нейронные сети, которые используют информацию в форме действительных чисел;

1.2) Двоичные нейронные сети. Они оперируют с информацией, представленной в двоичном виде.

2) *По характеру обучения:*

2.1) С учителем, когда известно выходное пространство решений нейронной сети;

2.2) Без учителя. В этом случае нейронная сеть формирует выходное пространство решений только на основе входных воздействий. Такие сети называются самоорганизующимися.

3) *По характеру настройки синапсов:*

3.1) Сети с фиксированными связями. В этом случае весовые коэффициенты нейронной сети выбираются сразу, исходя из условия задачи. При этом

$$\frac{dW}{dt} = 0,$$

где W - характеризует весовые коэффициенты сети.

3.2) Сети с динамическими связями. Для них в процессе обучения происходит настройка синаптических связей, т.е.

$$\frac{dW}{dt} \neq 0.$$

4) *По методу обучения:*

4.1) Нейронные сети с алгоритмом обратного распространения ошибки;

4.2) Нейронные сети с конкурентным обучением;

4.3) Нейронные сети, использующие правило Хебба;

4.4) Нейронные сети с гибридным обучением, в которых используются различные алгоритмы обучения.

5) *По характеру связей:*

5.1) Нейронные сети с прямыми связями (feedforward networks). В таких сетях происходит распространение информации только в одном направлении от слоя к слою (рис. 1.1). На рис. 1.1. W - характеризует синаптические связи, а F - оператор нелинейного преобразования нейронной сети. К такого типа сетям относятся различного рода персептронные сети: однослойные, многослойные, гомогенные и гетерогенные.

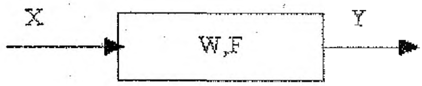

Рис. 1.1. Нейронная сеть с прямыми связями

5.2) Нейронные сети с обратным распространением информации (feedback networks). Они характеризуются как прямым, так и обратным распространением информации между слоями нейронной сети. К сетям с такими связями относятся следующие:

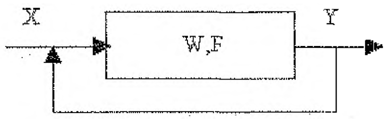

Рис. 1.2. Структура сети Хопфилда

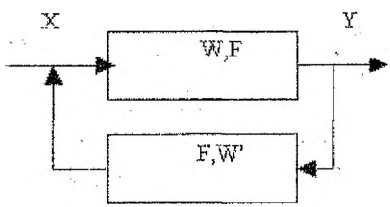

Рис. 1.3. Двунаправленная ассоциативная память

5.2.1) Релаксационные, в которых циркуляция информации происходит до тех пор, пока не перестанут изменятся выходные значения нейронной сети (состояние равновесия). К ним относятся нейронные сети Хопфилда, Хэмминга и двунаправленная ассоциативная память. Общая структура сети Хопфилда изображена на рис. 1.2. Она характеризуется единичной обратной связью. Структура двунаправленной ассоциативной памяти представлена на рис. 1.3. Она характеризуется неединичной обратной связью.

5.2.2) Многослойные сети, в которых отсутствует процесс релаксации. Архитектура их базируется на многослойном персептроне, а в основе обучения лежит алгоритм обратного распространения ошибки. К таким сетям относятся следующие:

5.2.2.1) Рекуррентные сети, в которых присутствует обратная связь между входом и выходом. Выходное значение при этом определяется в зависимости как от входных, так и от предшествующих выходных значений нейронной сети (рис.1.4).

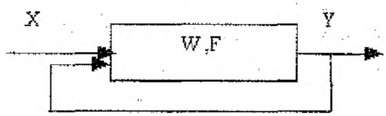

Рис. 1.4. Рекуррентная сеть

5.2.2.2) Рециркуляционные нейронные сети. Такие сети характеризуются как прямым $y = f(x)$, так и обратным $x = f(y)$ преобразованием информации. (рис.1.5). В рециркуляционных сетях обучение производится без учителя, т.е. они являются самоорганизующимися в процессе работы

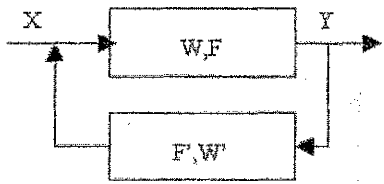

Рис. 1.5. Рециркуляционная нейронная сеть

6) *По архитектуре и обучению* различают следующие известные нейронные сети:

6.1) Персептронные сети с прямыми связями;

6.2) Самоорганизующиеся нейронные сети. К ним относятся следующие сети:

6.2.1) Нейронные сети Кохонена;

6.2.2) Нейронные сети адаптивного резонанса;

6.2.3) Рециркуляционные нейронные сети;

6.3) Нейронные сети с обратными связями:

6.3.1) Нейронные сети Хопфилда;

6.3.2) Нейронные сети Хэмминга;

6.3.3) Двунаправленная ассоциативная память;

6.3.4) Рекуррентные нейронные сети;

6.4) Гибридные нейронные сети:

6.4.1) Нейронные сети встречного распространения;

6.4.2) Нейронные сети с радиально-базисной функцией активации.

2. ЛИНЕЙНЫЕ НЕЙРОННЫЕ СЕТИ

2.1. Правило обучения Видроу-Хоффа

Используется для обучения нейронной сети, состоящей из распределительных нейронов и одного выходного нейрона, который имеет линейную функцию активации (рис. 2.1):

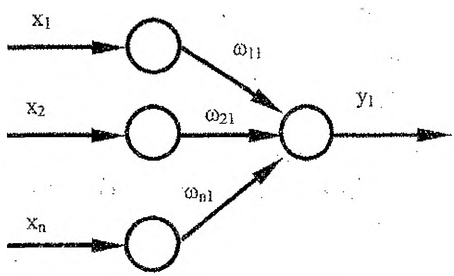

Рис. 2.1. Линейная сеть

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 выражением

$$y_1 = \sum_{j=1}^{n} \omega_{j1} x_j - T. \qquad (2.1)$$

где ω_{j1} – весовые коэффициенты, x_j – входы, T – порог нейронной сети.

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:

$$E = \sum_{k=1}^{L} E(k) = \frac{1}{2}\sum_{k=1}^{L}(y_1^k - t^k)^2, \qquad (2.2)$$

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 (2.2) 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

$$E(k) = \frac{1}{2}(y_1^k - t^k)^2. \qquad (2.3)$$

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:

$$\omega_{j1}(t+1) = \omega_{j1}(t) - \alpha\frac{\partial E(k)}{\partial \omega_{j1}(t)}, \qquad (2.4)$$

$$T(t+1) = T(t) - \alpha\frac{\partial E(k)}{\partial T(t)}, \qquad (2.5)$$

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

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

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

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

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

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

$$\omega_{j1}(t+1)=\omega_{j1}(t)-\alpha(y_1^k-t^k)x_j^k, \quad (2.6)$$

$$T(t+1)=T(t)+\alpha(y_1^k-t^k). \quad (2.7)$$

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 α $(0<\alpha<1)$ 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 (2.6) и (2.7);

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, т. е. не привести к решению задачи. Таким образом, сходимость алгоритма обучения не избавляет от разумного выбора значения шага обучения.

2.2. Выбор адаптивного шага обучения

Для ускорения процедуры обучения градиентного спуска вместо постоянного шага обучения можно использовать ***адаптивный шаг*** обучения $\alpha(t)$. Назовем адаптивным шагом обучения такой шаг, который целенаправленно выбирается на каждом этапе алгоритма таким образом, чтобы минимизировать среднеквадратичную ошибку сети.

Рассмотрим линейную нейронную сеть, которая состоит из распределительного слоя нейронных элементов и выходного слоя (рис. 2.2).

В качестве нейронов выходного слоя используются нейронные элементы с линейной функцией активации.

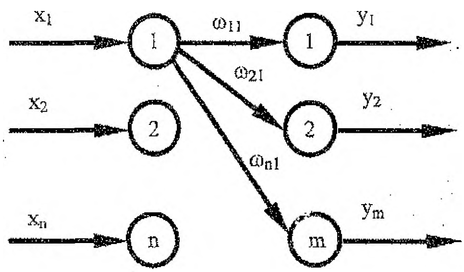

Рис. 2.2. Линейная нейронная сеть

Каждый нейрон распределительного слоя имеет синаптические связи со всеми нейронами обрабатывающего слоя. Выходное значение j-го нейрона сети определяется, как

$$y_j = \sum_i \omega_{ij} x_i - T_j . \qquad (2.8)$$

Среднеквадратичная ошибка сети для всей обучающей выборки будет равна

$$E_s = \frac{1}{2} \sum_k \sum_j (y_j{}^k - t_j{}^k)^2 . \qquad (2.9)$$

Соответственно среднеквадратичную ошибку для одного образа представим в виде:

$$E = \frac{1}{2} \sum_j (y_j - t_j)^2 . \qquad (2.10)$$

Для нахождения адаптивного шага обучения будем использовать метод наискорейшего спуска. Он заключается в том, чтобы на каждом шаге обучения нейронной сети необходимо выбирать такую скорость обучения $\alpha(t)$, чтобы минимизировать среднеквадратичную ошибку:

$$\alpha(t) = \min E\{(\omega_{ij}(t) - \alpha(t)\frac{\partial E}{\partial \omega_{ij}(t)}); (T_j - \alpha(t)\frac{\partial E}{\partial T_j(t)})\} . \qquad (2.11)$$

Подставляя в (2.8) выражения для изменения весовых коэффициентов и порогов нейронных элементов, получим:

$$y_j' = \sum_i (\omega_{ij}(t) - \alpha(t)\frac{\partial E}{\partial \omega_{ij}(t)}) \cdot x_i - T_j(t) + \alpha(t)\frac{\partial E}{\partial T_j(t)}.$$

Представим последнее выражение в следующем виде:

$$y_j' = \sum_i \omega_{ij}(t) \cdot x_i - T_j(t) - \alpha(t)\sum_i (\frac{\partial E}{\partial \omega_{ij}(t)} x_i - \frac{\partial E}{\partial T_j(t)}). \qquad (2.12)$$

Обозначим:

$$a_j = \sum_i (\frac{\partial E}{\partial \omega_{ij}(t)} x_i - \frac{\partial E}{\partial T_j(t)}). \qquad (2.13)$$

Тогда, с учетом выражения (2.13), выражение (2.12) можно записать как

$$y_j' = y_j - \alpha(t) a_j. \qquad (2.14)$$

Для определения адаптивного шага обучения необходимо найти такое значение $\alpha(t)$, чтобы среднеквадратичная ошибка была минимальной:

$$E = \frac{1}{2}\sum_j (y_j' - t_j)^2 = \frac{1}{2}\sum_j (y_j - \alpha a_j - t_j)^2 \to \min.$$

Тогда

$$\frac{\partial E}{\partial \alpha} = \sum_j \alpha a_j^2 - a_j(y_j - t_j) = 0.$$

Отсюда

$$\alpha = \frac{\sum_j a_j (y_j - t_j)}{\sum_j a_j^2}. \qquad (2.15)$$

Так как $\frac{\partial^2 E}{\partial^2 \alpha} > 0$, то при данном α обеспечивается минимум среднеквадратичной ошибки.

Найдем выражение для a_j. Так как

$$\frac{\partial E}{\partial \omega_{ij}(t)} = (y_j - t_j) x_i,$$

$$\frac{\partial E}{\partial T_j} = -(y_j - t_j),$$

то

$$a_j = (y_j - t_j)(1 + \sum_i x_i^2). \qquad (2.16)$$

Подставляя (2.16) в выражение (2.15), получим окончательное выражение для адаптивного шага обучения на каждой итерации алгоритма:

$$\alpha(t) = \frac{1}{1 + \sum_i x_i^2(t)}. \tag{2.17}$$

2.3. 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. 2.3).

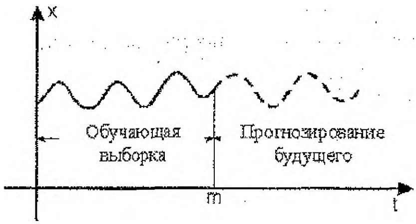

Рис. 2.3. Прогнозирование временного ряда

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:

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

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

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:

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

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.

3. МНОГОСЛОЙНЫЕ НЕЙРОННЫЕ СЕТИ

3.1. Математические основы алгоритма обратного распространения ошибки

Алгоритм обратного распространения ошибки является эффективным средством для обучения многослойных нейронных сетей.

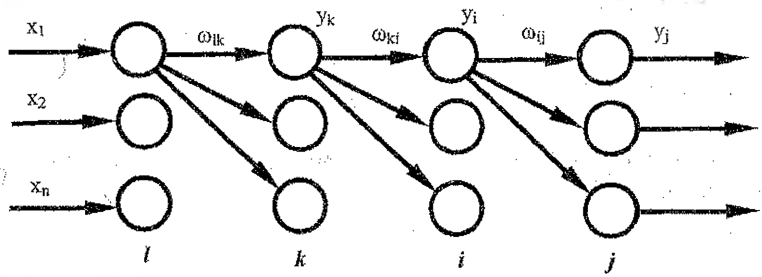

Рис. 3.1. Четырехслойная нейронная сеть

Рассмотрим нейронную сеть, состоящую из четырех слоев (рис. 3.1). Обозначим слои нейронных элементов от входа к выходу соответственно через l, k, i, j. Тогда выходное значение j-го нейрона последнего слоя равняется:

$$y_j = F(S_j), \qquad (3.1)$$

$$S_j = \sum_i \omega_{ij} y_i - T_j, \qquad (3.2)$$

где S_j — взвешенная сумма j-го нейрона выходного слоя; y_i — выходное значение i-го нейрона предпоследнего слоя; ω_{ij} и T_j — соответственно весовой коэффициент и порог j-го нейрона выходного слоя.

Аналогичным образом выходное значение i-го нейрона предпоследнего слоя определяется как:

$$y_i = F(S_i), \qquad (3.3)$$

$$S_i = \sum_k \omega_{ki} y_k - T_i. \qquad (3.4)$$

Соответственно для k-го слоя:

$$y_k = F(S_k), \qquad (3.5)$$

$$S_k = \sum_l \omega_{lk} x_l - T. \qquad (3.6)$$

Алгоритм обратного распространения ошибки минимизирует среднеквадратичную ошибку нейронной сети. Для этого с целью настройки синаптических связей используется метод градиентного спуска в пространстве весовых коэффициентов и порогов нейронной сети. Согласно методу градиентного спуска, изменение весовых коэффициентов и порогов нейронной сети происходит по следующему правилу:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha \frac{\partial E}{\partial \omega_{ij}(t)}, \qquad (3.7)$$

$$T_j(t+1) = T_j(t) - \alpha \frac{\partial E}{\partial T_j(t)}, \qquad (3.8)$$

где E — среднеквадратичная ошибка нейронной сети для одного образа.

Она определяется, как

$$E = \frac{1}{2} \sum_j (y_j - t_j)^2, \qquad (3.9)$$

где t_j — эталонное выходное значение j-го нейрона.

Ошибка j-го нейрона выходного слоя равняется:

$$\gamma_j = y_j - t_j. \qquad (3.10)$$

Теорема 3.1. Для любого скрытого слоя ошибка i-го нейронного элемента определяется рекурсивным образом через ошибки нейронов следующего слоя j:

$$\gamma_i = \sum_{j=1}^{m} \gamma_j F'(S_j)\omega_{ij}, \tag{3.11}$$

где m — количество нейронов следующего слоя по отношению к слою i; ω_{ij} — синаптическая связь между i-м и j-м нейроном различных слоев; S_j — взвешенная сумма j-го нейрона.

Теорема 3.2. Производные среднеквадратичной ошибки по весовым коэффициентам и порогам нейронных элементов для любых двух слоев i и j многослойной сети определяются следующим образом:

$$\frac{\partial E}{\partial \omega_{ij}} = -\gamma_j F'(S_j) y_i, \tag{3.12}$$

$$\frac{\partial E}{\partial T_j} = \gamma_j F'(S_j). \tag{3.13}$$

Следствие 3.3. Для минимизации среднеквадратичной ошибки сети весовые коэффициенты и пороги нейронных элементов должны изменяться с течением времени следующим образом:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha\gamma_j F'(S_j) y_i, \tag{3.14}$$

$$T_j(t+1) = T_j(t) + \alpha\gamma_j F'(S_j), \tag{3.15}$$

где α — скорость обучения.

Данное следствие является очевидным. Оно определяет правило обучения многослойных нейронных сетей в общем виде, которое называется ***обобщенным дельта правилом*** .

3.2. Обобщенное дельта правило для различных функций активации нейронных элементов

Определим выражения (3.14) и (3.15) для различных функций активации нейронных элементов.

3.2.1. Сигмоидная функция

Выходное значение j-го нейронного элемента определяется следующим образом:

$$y_j = \frac{1}{1+e^{-S_j}}, \tag{3.16}$$

$$S_j = \sum_i \omega_{ij} y_i - T_j. \tag{3.17}$$

Тогда

$$F'(S_j) = \frac{\partial y_j}{\partial S_j} = y_j(1-y_j) \tag{3.18}$$

В результате обобщенное дельта правило для сигмоидной функции активации можно представить в следующем виде:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha\gamma_j y_j(1-y_j)y_i, \tag{3.19}$$

$$T_j(t+1) = T_j + \alpha\gamma_j y_j(1-y_j). \tag{3.20}$$

Ошибка для j-го нейрона выходного слоя определяется как

$$\gamma_j = y_j - t_j. \tag{3.21}$$

Для j-го нейронного элемента скрытого слоя:

$$\gamma_j = \sum_{i=1}^{m} \gamma_i y_i(1-y_i)\omega_{ji}, \tag{3.22}$$

где m — количество нейронных элементов следующего слоя по отношению к слою i (рис. 3.2).

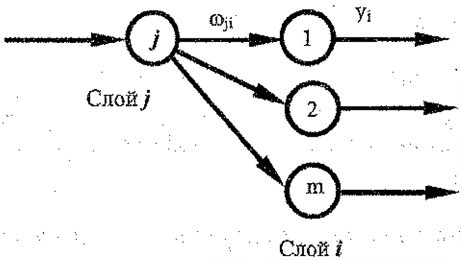

Рис. 3.2. Определение ошибки j-го нейронного элемента

3.2.2. Биполярная сигмоидная функция

Выходное значение j-го нейрона определяется как

$$y_j = \frac{2}{1+e^{-S_j}} - 1. \tag{3.23}$$

Тогда

$$F'(S_j) = \frac{\partial E}{\partial S_j} = \frac{1}{2}(1-y_j^2). \quad (3.24)$$

Отсюда получаем следующие выражения для обучения нейронной сети с биполярной сигмоидной функцией активации:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha\gamma_j \frac{1}{2}(1-y_j^2)y_i, \quad (3.25)$$

$$T_j(t+1) = T_j(t) + \alpha\gamma_j \frac{1}{2}(1-y_j^2). \quad (3.26)$$

Ошибка для j-го нейрона выходного и скрытого слоев определяется соответственно как:

$$\gamma_j = y_j - t_j, \quad (3.27)$$

$$\gamma_j = \sum_i \gamma_i \frac{1}{2}(1-y_i^2)\omega_{ji}. \quad (3.28)$$

3.2.3. Гиперболический тангенс

Для данной функции активации выходное значение j-го нейрона определяется следующим образом:

$$y_j = th(S_j) = \frac{e^{S_j} - e^{-S_j}}{e^{S_j} + e^{-S_j}}. \quad (3.29)$$

Определим производную функции гиперболический тангенс:

$$F'(S_j) = \frac{\partial y_j}{\partial S_j} = \frac{1}{ch^2(S_j)} = (1-y_j^2). \quad (3.30)$$

Тогда правило обучения можно представить в виде следующих выражений:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha\gamma_j(1-y_j^2)y_i, \quad (3.31)$$

$$T_j(t+1) = T_j(t) + \alpha\gamma_j(1-y_j^2). \quad (3.32)$$

Ошибка для j-го нейрона выходного и скрытого слоев соответственно равняется:

$$\gamma_j = y_j - t_j, \quad (3.33)$$

$$\gamma_j = \sum_i \gamma_i(1-y_i^2)\omega_{ji}. \quad (3.34)$$

Используя полученные в данном разделе выражения, можно определить алгоритм обратного распространения ошибки для различных функций активации нейронных элементов.

3.3. Алгоритм обратного распространения ошибки

Как уже отмечалось, алгоритм обратного распространения ошибки был предложен в 1986 г. рядом авторов независимо друг от друга. Он является эффективным средством обучения нейронных сетей и представляет собой следующую последовательность шагов:

1) Задается шаг обучения α $(0 < \alpha < 1)$ и желаемая среднеквадратичная ошибка нейронной сети E_m;

2) Случайным образом инициализируются весовые коэффициенты и пороговые значения нейронной сети;

3) Последовательно подаются образы из обучающей выборки на вход нейронной сети. При этом для каждого входного образа выполняются следующие действия:

а) производится фаза прямого распространения входного образа по нейронной сети. При этом вычисляется выходная активность всех нейронных элементов сети:

$$y_j = F(\sum_i \omega_{ij} y_i - T_j),$$

где индекс j характеризует нейроны следующего слоя по отношению к слою i.

b) производится фаза обратного распространения сигнала, в результате которой определяется ошибка γ_j нейронных элементов для всех слоев сети. При этом соответственно для выходного и скрытого слоев:

$$\gamma_j = y_j - t_j,$$

$$\gamma_j = \sum_i \gamma_i F'(S_i) \omega_{ji}.$$

В последнем выражении индекс i характеризует нейронные элементы следующего слоя по отношению к слою j;

c) для каждого слоя нейронной сети происходит изменение весовых коэффициентов и порогов нейронных элементов:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha \gamma_j F'(S_j) y_i,$$

$$T_j(t+1) = T_j(t) + \alpha \gamma_j F'(S_j);$$

4) Вычисляется суммарная среднеквадратичная ошибка нейронной сети:

$$E = \frac{1}{2} \sum_{k=1}^{L} \sum_j (y_j^k - t_j^k)^2,$$

где L — размерность обучающей выборки.

5) Если $E > E_m$, то происходит переход к шагу 3 алгоритма. В противном случае алгоритм обратного распространения ошибки заканчивается.

Таким образом, данный алгоритм функционирует до тех пор, пока суммарная среднеквадратичная ошибка сети не станет меньше заданной, т. е. $E \le E_m$.

3.4. Адаптивный шаг обучения

В стандартном алгоритме обратного распространения ошибки существует проблема выбора подходящего шага обучения, чтобы увеличить быстродействие и обеспечить сходимость алгоритма. Для выбора адаптивного шага обучения α можно использовать метод наискорейшего спуска. В соответствии с ним, на каждой итерации обучения нейронной сети, необходимо выбирать шаг обучения для каждого слоя таким, чтобы минимизировать среднеквадратичную ошибку сети:

$$\alpha(t) = \min E(y_j(t+1)), \tag{3.35}$$

где $j = \overline{1,m}$, m - количество нейронных элементов последнего слоя.

Выходное значение j-го нейрона зависит от функции активации нейронных элементов и в общем случае определяется следующим образом:

$$y_j(t+1) = F(\omega_{ij}(t+1), T_j(t+1)). \tag{3.36}$$

При этом весовые коэффициенты и пороги нейронной сети модифицируются согласно выражениям:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha(t)\frac{\partial E}{\partial \omega_{ij}(t)}, \tag{3.37}$$

$$T_j(t+1) = T_j(t) - \alpha(t)\frac{\partial E}{\partial T_j(t)}. \tag{3.38}$$

Среднеквадратичная ошибка нейронной сети равняется:

$$E = \frac{1}{2}\sum_j (y_j - t_j)^2. \tag{3.39}$$

Тогда для нахождения $\alpha(t)$ необходимо решить следующее уравнение:

$$\frac{\partial E}{\partial \alpha(t)} = \frac{\partial E}{\partial y_j(t+1)} \cdot \frac{\partial y_j(t+1)}{\partial \alpha(t)}. \tag{3.40}$$

Данное уравнение невозможно решить относительно $\alpha(t)$ аналитическим путем. Поэтому в ряде работ для определения адаптивного шага обучения предлагается использовать методы линейного поиска. Однако это связано со значительными вычислениями. Поэтому можно предложить приближенный метод нахождения скорости обучения $\alpha(t)$. Он базируется на разложении функции активаций нейронных элементов в ряд Тейлора. Рассмотрим это подробно.

Пусть выходное значение j-ого нейрона последнего слоя нейронной сети равняется:

$$y_j(t) = F(s_j(t)),$$
$$s_j(t) = \sum_i y_i(t)\omega_{ij}(t) - T_j(t). \tag{3.41}$$

где $y_i(t)$ - выходное значение i-ого нейрона скрытого слоя.

Для определения взвешенной суммы j-ого нейрона в момент времени $t+1$ подставим в (3.41) выражения (3.37) и (3.38):

$$s_j(t+1) = \sum_i y_i(\omega_{ij} - \alpha\frac{\partial E}{\partial \omega_{ij}}) - T_j + \alpha\frac{\partial E}{\partial T_j} =$$
$$= \sum_i y_i\omega_{ij} - T_j - \alpha\cdot(\sum_i y_i\cdot\frac{\partial E}{\partial \omega_{ij}} - \frac{\partial E}{\partial T_j}). \tag{3.42}$$

Обозначим

$$a_j = \sum_i y_i\frac{\partial E}{\partial \omega_{ij}} - \frac{\partial E}{\partial T_j}. \tag{3.43}$$

Тогда выражение (3.42) можно представить в следующем виде:

$$S_j(t+1) = S_j(t) - \alpha\cdot a_j. \tag{3.44}$$

Выходное значение j-ого нейрона в момент времени $t+1$ равняется:

$$y_j(t+1) = F(S_j(t+1)). \tag{3.45}$$

Разложим данное выражение по формуле Тейлора и ограничимся первыми двумя членами:

$$y_j(t+1) = F(0) + F'(0)\cdot F(S_j(t+1), \tag{3.46}$$

где

$$F'(0) = \frac{\partial F}{\partial S_j},$$

при $S_j=0$.

Подставим в (3.46) выражение (3.44). Тогда

$$y_j(t+1) = F(0) + F'(0)S_j(t) - \alpha F'(0)a_j. \tag{3.47}$$

Так как

$$y_j(t) = F(0) + F'(0)S_j(t), \tag{3.48}$$

то выражение (3.47) можно представить в следующем виде:

$$y_j(t+1) = y_j(t) - \alpha F'(0)a_j. \qquad (3.49)$$

Для определения адаптивного шага обучения необходимо обеспечить:

$$E = \frac{1}{2}\sum_j (y_j(t+1) - t_j)^2 \to \min \qquad (3.50)$$

Тогда

$$\frac{\partial E}{\partial \alpha} = \sum_j (y_j(t) - t_j - \alpha F'(0)a_j)\cdot(-F'(0)a_j) = 0$$

Выражая из последнего уравнения $\alpha(t)$, получим:

$$\alpha(t) = \frac{\sum_j (y_j(t) - t_j)a_j}{F'(0)\sum_j a_j^2} \qquad (3.51)$$

Так как $\frac{\partial^2 E}{\partial \alpha^2} > 0$, то при данном $\alpha(t)$ обеспечивается минимум среднеквадратичной ошибки. Найдем выражение для a_j. Для этого определим

$$\frac{\partial E}{\partial \omega_{ij}} = \frac{\partial E}{\partial y_j}\cdot\frac{\partial y_j}{\partial s_j}\cdot\frac{\partial s_j}{\partial \omega_{ij}} = (y_j - t_j)F'(S_j)y_i, \qquad (3.52)$$

$$\frac{\partial E}{\partial T_j} = \frac{\partial E}{\partial y_j}\cdot\frac{\partial y_j}{\partial s_j}\cdot\frac{\partial s_j}{\partial T_j} = -(y_j - t_j)F'(S_j). \qquad (3.53)$$

Подставляя (3.52) и (3.53) в выражение (3.43), получим:

$$a_j = (1 + \sum_i y_i^2)\cdot(y_j - t_j)\cdot F'(S_j). \qquad (3.54)$$

Исходя из принципа независимости слоев, предполагаем, что

$$\gamma_j = y_j - t_j. \qquad (3.55)$$

Подставляя выражения (3.55) и (3.54) в (3.51), получим приближенное выражение для вычисления адаптивного шага обучения различных слоев нейронной сети:

$$\alpha(t) = \frac{\sum_j \gamma_j^2 F'(S_j)}{F'(0)\cdot(1 + \sum_j \gamma_j^2 (F'(S_j))^2}, \qquad (3.56)$$

где γ_j - ошибка j-ого нейронного элемента, которая для различных слоев сети вычисляется в соответствии с (3.55).

Рассмотрим определение выражения (3.56) для различных функций активации нейронных элементов.

3.4.1. Сигмоидная функция

Так как производные сигмоидной функции:

$$y_j' = F'(S_j) = y_j(1 - y_j),$$

$$y_j'(0) = F'(0) = \frac{1}{4},$$

то выражение для адаптивного шага обучения можно представить в следующем виде:

$$\alpha(t) = \frac{4\sum_j \gamma_j^2 y_j(1-y_j),}{(1+\sum_i y_i^2)\cdot\sum_j \gamma_j^2 y_j^2(1-y_j)^2}, \tag{3.57}$$

где ошибка j-ого нейронного элемента скрытого слоя равняется:

$$\gamma_j = \sum_i \gamma_i y_i (1-y_i)\omega_{ji} \tag{3.58}$$

Здесь i характеризует нейроны следующего слоя по отношению к слою j.

3.4.2. Гиперболический тангенс

Так как

$$y_j' = F'(s_j) = \frac{1}{ch^2(s_j)} = (1 - y_j^2),$$

$$y_j'(0) = F'(0) = 1,$$

то

$$\alpha(t) = \frac{\sum_j \gamma_j^2 (1-y_j^2),}{(1+\sum_i y_i^2)\cdot\sum_j \gamma_j^2 (1-y_j^2)}. \tag{3.59}$$

Ошибка j-ого нейрона скрытого слоя определяется как

$$\gamma_j = \sum_i \gamma_i (1-y_i^2)\omega_{ji}. \tag{3.60}$$

3.4.3. Биполярная сигмоидная функция

Определим производные данной функции:

$$y_j' = F'(s_j) = \frac{1}{2}(1 - y_j^2),$$

$$y_j'(0) = F'(0) = \frac{1}{2}.$$

Тогда

$$\alpha(t) = \frac{4\sum_j \gamma_j^2 y_j (1 - y_j),}{(1 + \sum_i y_i^2) \cdot \sum_j \gamma_j^2 y_j^2 (1 - y_j)^2}, \tag{3.61}$$

где

$$\gamma_j = \sum_i \gamma_i \frac{1}{2} \cdot (1 - y_i^2) \omega_{ji}. \tag{3.62}$$

Заметим, что в приведенных выше выражениях $\alpha(t)$ вычисляется отдельно для каждого слоя нейронной сети. Как показывают эксперименты, при использовании адаптивного шага обучения могут получаться слишком большие значения $\alpha(t)$. Это может привести к десинхронизации процесса обучения, когда весовые коэффициенты резко изменяются в определенном направлении. В результате изменение среднеквадратичной ошибки с течением времени будет иметь колебательный характер. Поэтому рекомендуется ограничивать $\alpha(t)$ по абсолютному значению. Полученные выше выражения для адаптивного шага обучения позволяют значительно повысить скорость обучения нейронной сети и избежать выбора подходящего шага. Это является существенным преимуществом по сравнению со стандартным алгоритмом обратного распространения ошибки. Хотя при удачном выборе постоянного шага обучения данный алгоритм будет сходиться не быстрее, чем метод градиентного спуска.

ЛАБОРАТОРНАЯ РАБОТА №1
ЛИНЕЙНАЯ ИСКУССТВЕННАЯ НЕЙРОННАЯ СЕТЬ

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

1. Порядок проведения работы

1.1. Изучить основные теоретические сведения об особенностях организации и обучения линейных моделей ИНС (см. разделы 1, 2).

1.2. Написать на любом ЯВУ программу моделирования прогнозирующей линейной ИНС. Для организации обучающего и тестового подмножеств использовать следующую функцию:

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

Обучение и прогнозирование производить на 30 и 15 значениях соответственно, табулируя прогнозируемую функцию с шагом $h=0.1$. Скорость обучения выбирается студентом самостоятельно, для чего проводится несколько экспериментов для различных значений 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 значениях α. Варианты заданий приведены в таблице П.1 приложения.

1.3. Модифицировать программу из п.1.2, используя правило адаптивного шага обучения. Исследовать получившуюся модель ИНС в задачах прогнозирования.

2. Содержание отчета

Исходные данные взять из таблицы П.1.

2.1. Титульный лист;

2.2. Цель работы;

2.3. Задание;

2.4. Результаты обучения: таблица в форме П.3, график изменения ошибки в зависимости от итерации для п. 1.2;

2.5. Результаты прогнозирования: для п.1.2;

2.6. Результаты обучения: таблица в форме П.3, график изменения ошибки в зависимости от итерации для п. 1.3;

2.7. Результаты прогнозирования: таблица в форме П.4 для п.1.3;

2.8. Сравнение результатов пунктов 5 и 7;

2.9. Выводы по лабораторной работе.

3. Контрольные вопросы

3.1. ИНС какой архитектуры использовалась в данной работе? Опишите принцип построения этой ИНС.

3.2. Как функционирует использованная ИНС?

3.3. Опишите алгоритм обучения использованной в работе ИНС.

3.4. Согласно какому критерию определяется адаптивный шаг обучения для использованной в работе ИНС?

3.5. Как формируется обучающая выборка для решения задачи прогнозирования?

3.6. Как выполняется многошаговое прогнозирование временного ряда?

3.7. Предложите критерий оценки качества результатов прогноза.

ЛАБОРАТОРНАЯ РАБОТА №2
НЕЛИНЕЙНЫЕ ИНС В ЗАДАЧАХ ПРОГНОЗИРОВАНИЯ

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

1. Порядок проведения работы

1.1. Изучить основные теоретические сведения относительно способов организации, функционирования и обучения многослойных нелинейных нейронных сетей (см. раздел 3).

1.2. Написать на любом ЯВУ программу моделирования прогнозирующей нелинейной ИНС. Для организации обучающего и тестового подмножеств использовать следующую функцию:

$$y = a\cos(bx+e)+c\sin(dx+g).$$

Для прогнозирования использовать многослойную ИНС с одним скрытым слоем. В качестве функций активации использовать сигмоидную функцию. Варианты заданий приведены в таблице П.2.

1.3. Спрогнозировать временной ряд, используя алгоритм обучения многослойной ИНС с адаптивным шагом.

2. Содержание отчета

Исходные данные взять из таблицы П.2.

2.1. Титульный лист;

2.2. Цель работы;

2.3. Задание;

2.4. Результаты обучения: таблица в форме П.3, график изменения ошибки в зависимости от итерации для п. 1.2;

2.5. Результаты прогнозирования: для п.1.2;

2.6. Результаты обучения: таблица в форме П.3, график изменения ошибки в зависимости от итерации для п. 1.3;

2.7. Результаты прогнозирования: таблица в форме П.4 для п.1.3;

2.8. Сравнение результатов пунктов 5 и 7;

2.9. Выводы по лабораторной работе.

3. Контрольные вопросы

3.1. ИНС какой архитектуры использовалась в данной работе? Опишите принцип построения этой ИНС.

3.2. Как функционирует использованная в работе ИНС?

3.3. Докажите, что многослойная нейронная сеть, содержащая *n* входных элементов, любое количество скрытых слоев и линейных нейроэлементов и *m* выходных линейных нейронов, эквивалентна двухслойной линейной нейронной сети, содержащей *n* входных и *m* выходных линейных элементов.

3.4. Опишите общий алгоритм обучения многослойной нелинейной ИНС.

3.5. Как формируется обучающая выборка для решения задачи прогнозирования?

3.6. Как выполняется многошаговое прогнозирование временного ряда?

3.7. Предложите критерий оценки качества результатов прогноза.

ЛАБОРАТОРНАЯ РАБОТА №3
НЕЛИНЕЙНЫЕ ИНС В ЗАДАЧАХ РАСПОЗНАВАНИЯ ОБРАЗОВ

Цель работы: Изучить обучение и функционирование нелинейной ИНС при решении задач распознавания образов.

1. Порядок проведения работы

1.1. Выполнение работы опирается на теоретические сведения, предложенные для изучения в лабораторной работе №2. Нейропостановка задачи распознавания образов формулируется следующим образом:

- для представления каждого образа используется пара эталонов в виде битовых массивов данных; при этом входной массив используется для кодирования исходного образа; соответствующий ему выходной эталон представляет собой битовую последовательность размером L (L - количество пар эталонов в выборке), в которой один из битов выставлен в единицу, а все остальные являются нулевыми;

- в используемой архитектуре многослойной нелинейной ИНС количество входных нейронов определяется количеством элементов входных эталонов, а число выходных нейронов соответствует размерности векторов выходных эталонов; число нейронных элементов в скрытом слое должно быть подобрано эмпирически с целью достижения наилучшего качества распознавания;
- следующим этапом является обучение ИНС на основе сформированного подмножества пар эталонов до достижения требуемой погрешности обучения;
- в процессе распознавания на вход обученной ИНС подаются тестовые образы. В процессе функционирования ИНС формирует на выходном слое битовый вектор, по которому можно идентифицировать входной образ.

Пример: *входные массивы данных могут представлять собой изображения десятичных цифр, закодированные в битовой матрице 4x4.)*

1.2. Написать на любом ЯВУ программу моделирования нелинейной ИНС для распознавания образов. Программа должна моделировать две архитектуры ИНС: со скрытым слоем (вариант б) и без скрытого слоя (вариант а). Рекомендуется использовать сигмоидную функцию, однако это не является обязательным. Количество нейроэлементов в скрытом слое взять согласно варианту работы №2. Количество нейроэлементов можно варьировать, если сеть не обучается либо некорректно функционирует.

1.3. Провести исследование полученных моделей. При этом на вход сети необходимо подавать искаженные образы, в которых инвертированы некоторые биты. Критерий эффективности процесса распознавания - максимальное кодовое расстояние (количество искаженных битов) между исходным и поданным образом. Для формирования обучающей выборки использовать номера эталонов, приведенные в табл. П.5 согласно варианту задания. Соответствующие обучающие эталоны приведены в табл. П.6.

2. Содержание отчета

Исходные данные взять из таблицы П.5.

2.1. Титульный лист;
2.2. Цель работы;
2.3. Задание;
2.4. Результаты обучения: график изменения ошибки в зависимости от итерации для п. 1.2;
2.5. Описание результатов распознавания;
2.6. Сравнение результатов распознавания для разных вариантов архитектур ИНС;
2.7. Выводы по лабораторной работе.

3. Контрольные вопросы

3.1. ИНС какой архитектуры были использовали в данной работе? Опишите принцип построения этих ИНС.
3.2. Как функционируют использованные в работе ИНС?
3.3. Опишите алгоритм обучения использованной в работе ИНС.
3.4. В чем заключается нейропостановка задачи распознавания образов?

СПИСОК ЛИТЕРАТУРЫ

1. Головко В.А. Нейроинтеллект: теория и применение. Книга 1.: Организация и обучение нейронных сетей с прямыми связями. Брест Изд. БПИ, 1999 – 264 с.

ПРИЛОЖЕНИЕ

Варианты заданий к лабораторной работе №1

Таблица П.1.

№ 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

Варианты заданий к лабораторной работе №2

Таблица П.2.

№ варианта	a	b	c	d	Кол-во входов ИНС	Кол-во НЭ в скрытом слое	e	g
1	0.1	1	0.3	11	6	2	5	1
2	0.2	2	0.3	10	8	3	6	2
3	0.3	3	0.1	9	10	4	7	3
4	0.25	4	0.2	8	6	2	8	4
5	0.1	5	0.3	7	8	3	9	5
6	0.2	3	0.3	4	10	4	10	6
7	0.3	7	0.1	5	6	2	5	1
8	0.25	8	0.2	4	8	3	6	2
9	0.1	10	0.3	3	10	4	7	3
10	0.2	9	0.3	2	6	2	8	4
11	0.3	11	0.1	1	8	3	9	5

Примеры оформления результатов к лабораторным работам №1 и №2

Таблица П.3.

№ эталона обучающей выборки	Эталонное значение ряда	Полученное значение ряда	Отклонение
1			
....			

Таблица П.4.

№ шага прогнозирования	Эталонное значение ряда	Спрогнозированное значение ряда	Погрешность прогноза
1			
....			

Варианты заданий к лабораторной работе №3

Таблица П.5.

Вариант	Эталон 1	Эталон 2	Эталон 3
1	1	6	8
2	2	1	8
3	3	2	8
4	4	3	8
5	5	4	8
6	6	5	8
7	7	6	8
8	1	3	8
9	2	4	8
10	3	5	8
11	4	6	8

Таблица П.6.

№	Данные эталона																			
1	0	1	0	0	1	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0
2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
3	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
4	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
5	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
6	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1

Учебное издание

Составители: Головко В. А., к.т.н., профессор
Дунец А. П., старший преподаватель
Савицкий Ю. В. к.т.н., доцент
Шуть В. Н., к.т.н., доцент

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению лабораторных работ по курсу
«Интеллектуальные системы принятия решений»
для студентов специальности Т10.01.00

ЧАСТЬ 1

Ответственный за выпуск Дунец А.П.
Редактор Строкач Т. В.
Технический редактор: Никитчик А.Д.
Корректор: Никитчик Е.В.

Подписано к печати 04.07.01г.
Формат 60х84/16
Усл. п.л. 1.86. Уч. изд. л. 2.0. Тираж 120 экз. Заказ № 465
Отпечатано на ризографе учреждения образования
«Брестский государственный технический университет»
224017, Брест, ул. Московская, 267.