**Лабораторная работа № 12**

**E16.1**. Предположим, что весовая матрица для слоя 2 сети Хэмминга задается формулой

W2 = [1 ¾ ¾ -3/4 1 ¾ ¾ ¾ 1]T

Эта матрица нарушает уравнение (16.6), поскольку

ε = ¾ > 1/(S-1) = ½ .

Приведите пример вывода из уровня 1, для которого уровень 2 не будет работать правильно.

**E16.2** Рассмотрим входные векторы и начальные веса, показанные на рисунке E16.1.

(Смотри рисунок в книге!)

Рисунок E16.1. Векторы данных кластера

1. Нарисуйте диаграмму конкурентной сети, которая может классифицировать данные выше, чтобы каждый из трех кластеров векторов имел свой собственный класс.
2. Нарисуйте сеть графически (используя начальные веса), представив помеченные векторы в следующем порядке:

p1, p2, p3, p4.

Напомним, что конкурентная функция передачи выбирает нейрон с наименьшим индексом, чтобы выиграть, если более одного нейрона имеет тот же чистый ввод. Правило Кохонена вводится графически на рисунке 16.3.

**E16.3**. Обучайте конкурентную сеть, используя следующие шаблоны ввода:

p1 = [1 -1]T, p2 = [1 1]T, p3 = [-1 -1]T,

1. Используйте закон обучения Кохонена и тренируйтесь за один проход через шаблоны ввода. (Представьте каждый вход один раз, в указанном порядке.) Отобразите результаты графически. Предположим, что начальная весовая матрица

W = [√2 00 √2].

1. После прохождения через входные шаблоны, как сгруппированы шаблоны? (Другими словами, какие шаблоны группируются вместе в одном классе?) Будет ли это изменяться, если входные шаблоны были представлены в другом порядке? Объясните.
2. Повторите часть (i) используя. Как это изменение влияет на обучение?

**E16.4**. Ранее в этой главе термин «совесть» использовался для обозначения метода для предотвращения проблемы мертвых нейронов, сталкивающейся с конкурирующими слоями и LVQ сетью.

Нейроны, которые слишком далеки от входных векторов, чтобы выиграть конкуренцию, могут получить шанс, используя адаптивные смещения, которые становятся более негативными каждый раз, когда нейрон выигрывает конкуренцию. В результате нейроны, которые выигрывают очень часто, начинают чувствовать себя «виновными», пока другие нейроны не смогут выиграть.

На рисунке E16.2 показана конкурентная сеть со смещениями. Типичным правилом обучения для смещения b нейрона является

bnewi = {0.9boldi, если i <> i\*

boldi – 0.2, если i = i\*

(Смотри рисунок в книге!)

Рисунок E16.2 Конкурентный уровень с отклонениями

1. Изучите векторы на рисунке E16.3. Есть ли порядок, в котором могут быть представлены векторы, которые приведут к победе 1w в конкурсе и приблизиться к одному из векторов? (Примечание: предположите, что адаптивные предубеждения не используются).

(Смотри рисунок в книге!)

Рисунок E16.3 Входные векторы и мертвый нейрон

1. С учетом входных векторов и начальных весов и смещений, определенных ниже, вычислите веса (используя правило Кохонена) и смещения (используя вышеупомянутое правило смещения). Повторите последовательность, показанную ниже, пока нейрон 1 не выиграет соревнование.

p1 = [-1 0]T, p2 = [0 1]T, p3 = [1/√21/√2]T,

1w = [0 -1]T, 2w = [-2/√5 -1/√5]T, 3w = [-1/√5 -2/√5]T,

b1(0) = b2(0) = b3(0) = 0

Последовательность входных векторов: p1, p2, p3, p1, p2, p3,…

1. Сколько презентаций происходит до победы 1w в конкурсе?

**E16.5**. Чистое входное выражение для сетей LVQ рассчитывает расстояние между входом и каждым весовым вектором напрямую, вместо использования внутреннего продукта. В результате сеть LVQ не требует нормализованных входных векторов. Этот метод также можно использовать, чтобы позволить конкурентному слою классифицировать ненормализованные векторы. Такая сеть показана на рисунке E16.4.

(Смотри рисунок в книге!)

Рисунок E16.4 Конкурентный уровень с альтернативным сетевым выражением

Используйте этот метод для обучения конкурентному слою с двумя нейронами на (ненормализованных) векторах ниже, используя скорость обучения, равную 0,5.

p1 = [1 1]T, p2 = [-1 2]T, p3 = [-2 -2]T

Представьте векторы в следующем порядке:

p1, p2, p3, p2, p3, p1.

Вот начальные веса сети:

1w = [0 1]T, 2w = [1 0]T.

**E16.6** Повторите E16.5 для следующих входов и начальных весов. Покажите движения графиков для каждого шага. Если сеть подготовлена для большого количества итераций, как три вектора будут кластеризованы в конечной конфигурации?

p1 = [2 0]T, p2 = [0 1]T, p3 = [2 2]T

1w = [1 0]T, 2w = [-1 0]T.

**E16.7** У нас есть проблема конкурентного обучения, где входные векторы

p1 = [0 1]T, p2 = [0 2]T, p3 = [1 1]T, p4 = [2 2]T,

и исходная весовая матрица

W = [0 -1-1 1].

1. Используйте закон об обучении Кохонена для обучения конкурентоспособной сети с использованием скорости обучения α = 0.5. (Представьте каждый вектор один раз, в показанном порядке.) Используйте модифицированную конкурентную сеть на рисунке E16.4, которая использует отрицательное расстояние вместо внутреннего продукта.
2. Отобразить результаты части i графически, как показано на рисунке 16.3. (Показать все четыре итерации.)
3. Где весы в конечном итоге сходятся (приблизительно)? Объясните. Нарисуйте приблизительные границы окончательного решения.

**E16.8**. Покажите, что модифицированная конкурентная сеть на рисунке E16.4, которая напрямую вычисляет расстояние, даст те же результаты, что и стандартная конкурентная сеть, которая использует внутренний продукт, когда нормали входных векторов нормализованы.

**E16.9**. Мы хотели бы, чтобы классификатор разделил интервал входного пространства, определенный ниже, на пять классов.

0 <= p1 <= 1

1. Используйте MATLAB для случайного генерирования 100 значений в указанном выше интервале с равномерным распределением.
2. Возведите в квадрат каждое число так, чтобы распределение больше не было однородным.
3. Напишите M-файл MATLAB для реализации конкурентного уровня. Используйте M-файл для обучения пятиуровневого конкурентного слоя квадрату значений до тех пор, пока его веса не будут достаточно стабильными.
4. Как распределяются значения веса конкурентного слоя? Существует ли какая-то взаимосвязь между распределением весов и распределением квадратов входных значений?

**E16.10** Нам нужен классификатор, который делит квадратную область, определенную ниже, на шестнадцать классов примерно равного размера.

0 <= p1 <= 1, 2 <= p2 <= 3

1. Используйте MATLAB для случайной генерации 200 векторов в области, показанной выше.
2. Напишите M-файл MATLAB для реализации конкурентного уровня с обучением Кохонена. Вычислите чистый ввод, непосредственно определяя расстояние между входными и весовыми векторами, как это делает сеть LVQ, поэтому векторы не нуждаются в нормализации. Используйте M-файл для обучения конкурентного слоя для классификации 200 векторов. Попробуйте различные курсы обучения и сравните производительность.
3. Напишите M-файл MATLAB для реализации четырех нейронной карты с четырьмя нейронами (двухмерными). Используйте карту функций, чтобы классифицировать одни и те же векторы. Используйте разные уровни обучения и размеры окрестностей, а затем сравните производительность.

**E16.11** Мы хотим обучить следующую 1-D карту функций (которая использует расстояние вместо внутреннего продукта для вычисления чистого ввода):

(Смотри рисунок в книге!)

Рисунок E16.5 1-D Карта функций для упражнений E16.11

Начальная весовая матрица

W(0) = [2 -1 -1 1 2 1 -2 0]T.

1. Выделите начальные весовые векторы в виде точек и соедините соседние весовые векторы в виде строк (как на рисунке 16.10, за исключением того, что это одномерная карта функций 1 - D).
2. В сеть применяется следующий входной вектор. Выполните одну итерацию правила обучения карты объектов. (Вы можете сделать это графически.) Используйте размер окрестности 1 и скорость обучения α = 0.5.

p1 = [-2 0]T

1. Постройте новые весовые векторы в виде точек и соедините соседние весовые векторы как линии.

**E16.12**. Рассмотрим следующую карту характеристик, где вместо внутреннего произведения используется расстояние для вычисления сетевого ввода.

(Смотри рисунок в книге!)

Рисунок E16.6 2-D Карта функций для упражнений E16.12

Начальная весовая матрица

W = [0 1 1 0 0 0 1 -1]T.

1. Нарисуйте начальные веса и покажите их топологические соединения, как на рисунке 16.10.
2. Примените ввод p = [-1 1]T и выполните одну итерацию функции карты правила обучения, скорость обучения α = 0.5 и радиус соседства 1.
3. Выделите веса после первой итерации и покажите их топологические соединения.

**E16.13** Сеть LVQ имеет следующие веса:

W1 = [0 1 -1 0 0 0 0 0 1 -1]T, W2 = [10 0 0 0 0 1 1 0 0 0 0 0 1 1].

1. Сколько классов имеет эта сеть LVQ? Сколько подклассов?
2. Нарисуйте диаграмму, показывающую весовые векторы первого уровня и границы решений, которые разделяют входное пространство на подклассы.
3. Пометьте каждую область подкласса, чтобы указать, к какому классу принадлежит.

**E16.14** Мы хотели бы, чтобы сеть LVQ классифицировала следующие векторы в соответствии с указанными классами:

 класс 1: {[-1 1 -1]T ,[1 -1 -1]T }, класс 2: {[-1 -1 1]T , [1 -1 1]T , [1 1 -1]T }, класс 3: {[-1 -1 -1]T , [-1 1 1]T}.

1. Сколько нейронов требуется в каждом слое сети LVQ?
2. Определите весы для первого слоя.
3. Определите веса для второго слоя.
4. Проверьте свою сеть как минимум на один вектор из каждого класса.

**E16.15** Нам нужна сеть LVQ, которая классифицирует следующие векторы в соответствии с указанными классами:

класс 1: {p1 = [1 1]T , p2 = [0 2]T } , класс 2: {p3 = [-1 1]T , p4 = [1 2]T }

1. Может ли эта проблема классификации быть решена персептроном? Поясните свой ответ.
2. Сколько нейронов должно быть в каждом слое сети LVQ, которое может классифицировать вышеуказанные данные, учитывая, что каждый класс состоит из двух выпуклых подклассов?
3. Определите весы второго уровня для такой сети.
4. Инициализируйте весы первого уровня сети ко всем нулям и вычислите изменения, сделанные в весах правилом Кохонена (со скоростью обучения α = 0,5) для следующих рядов векторов:

p4, p2, p3, p1, p2.

1. Нарисуйте диаграмму, показывающую входные векторы, конечные весовые векторы и границы решения между двумя классами.

**E16.16** Сеть LVQ имеет следующие веса и данные обучения.

W1 = [1 0 0 0 1 0]T, W2 = [1 1 0 0 0 1],

{p1 = [-2 2]T, t1 = [1 0] T }, {p2 = [2 0]T, t2 = [0 1] T }, {p3 = [2 -2]T, t3 = [1 0] T }, {p4 = [-2 0]T, t4 = [0 1] T },

1. Нарисуйте входные векторы учебных данных и весовые векторы (как на рисунке 16.14).
2. Выполните четыре итерации правила обучения LVQ со скоростью обучения α = 0,5, поскольку вы представляете следующую последовательность входных векторов: p1, p2, p3, p4 (одна итерация для каждого входа). Сделайте это графически, на отдельной диаграмме из части i.
3. После завершения итераций в части ii, на новой диаграмме, нарисуйте области входного пространства, составляющие каждый подкласс и каждый класс. Пометьте каждую область, чтобы указать, к какому классу она принадлежит.

**E16.17**. Сеть LVQ имеет следующие веса:

W1 = [0 1 -1 0 0 -1 -1 0 0 0 1 -1 -1 1]T, W2 = [1 0 1 0 1 1 0  0 1 0 1 0 0 1]T,

1. Сколько классов имеет эта сеть LVQ? Сколько подклассов?
2. Нарисуйте диаграмму, показывающую весовые векторы первого уровня и границы решений, которые разделяют входное пространство на подклассы.
3. Пометьте каждую область подкласса, чтобы указать, к какому классу принадлежит.
4. Предположим, что ввод p = [1 0 5]T класса 1 представлен
5. сеть. Выполните одну итерацию алгоритма LVQ с α = 0,5.

**E16.18** Сеть LVQ имеет следующие веса:

W1 = [0 0 2 1 1 -1 0 2 2 1 -1 -1]T, W2 = [1 1 1 0 0 0 0 0 0 1 1 1] T.

1. Сколько классов имеет эта сеть LVQ? Сколько подклассов?
2. Нарисуйте диаграмму, показывающую весовые векторы первого уровня и границы решений, которые разделяют входное пространство на подклассы.
3. Пометьте каждую область подкласса, чтобы указать, к какому классу принадлежит.
4. Выполните одну итерацию алгоритма LVQ со следующей входной / целевой парой: p = [-1 -2]T, t = [1 0]T,. Использовать скорость обучения α = 0,5.