12) T1

11) Summ coeffs

9) Correct coeffs

8) T2

5) Out Layer

3) Work Layer

1) Enter values

(Схема 1)

teachedIndicator – программная реализация нейронной сети, использующей 2 слоя нейронов.

Логика программы представлена на [схеме 1](#_top):

**Инициализация:**

1. Производится вызов основного класса программы – NeyroNet – с параметрами:
   * + 1. Количество входных данных “X - ов”.
       2. Количество выходных данных “Y – ов”, количество нейронов выходного слоя.
       3. Параметр Size – определяет количество нейронов в рабочем (скрытом) слое.
2. inputs – массив – хранилище входных данных.
3. Производится вызов класса, обеспечивающего работу нейронных слоев – Layer – с параметрами:
4. Количество нейронов в слое.
5. Количество входов для каждого нейрона.
6. Производится инициализация класса Neyron параметрами:
7. Количество входов для каждого нейрона.
8. Номер этого нейрона.
9. Экземпляр класса Random для генерации случайных весов для каждого из входов.

**Режим обучения:**

1. Вызов метода Study экземпляра класса NeyroNet с параметрами:
2. Файл, содержащий матрицу “X - ов”.
3. Файл, содержащий матрицу “Y - ов”.
4. Шаг.
5. Количество итераций.

Работа метода Study демонстрируется на [схеме 1](#_top).

Вызов метода Exe для сети с параметром:

* Строка из матрицы обучающих “X - ов”.

Выдает результат работы всех слоев.

Метод вызывает для каждого слоя метод Exe.

Вызов метода Exe для двух слоев:

Первый с параметром:

* Строка из матрицы обучающих “X - ов”.

Второй с параметром:

* Результат работы первого.

Оба результата сохраняются в middle & output соответственно.

Метод вызывает для каждого нейрона метод Execute

Вызов метода Execute для каждого нейрона с параметром:

* Строка из матрицы “X - ов”.

Этот метод использует сумматор, хранящий произведение входящих значений на соответствующие веса; вычитает из сумматора порог активации (поправка, отвечающая за сдвиг сигмоиды); применяет к полученному значению экспоненциальную сигмоиду для получения выходного импульса.

Заводится массив поправок для выходного слоя outGradient, после чего для каждого нейрона рассчитывается градиент, умноженный на шаг

Вызов метода CorrectCoeffs для выходного слоя с параметрами:

* Градиент
* Выходной параметр oldOutLoyerWeights.

Вызывает соответствующий метод CorrectCoeffs для каждого нейрона.

Вызов метода CorrectCoeffs для нейрона с параметрами:

* Градиент
* Выходной параметр oldOutWeights.

С помощью этого метода нейрон выводит свои старые веса, а затем каждый из весов увеличивается на произведение градиента (поправки) на входное значение.

Теперь работаем с рабочим слоем:

Заводится массив поправок для рабочего слоя workGradient, далее высчитываем суммарную ошибку, после чего для каждого нейрона рассчитывается градиент, умноженный на шаг.

Вызов метода CorrectCoeffs для рабочего слоя с параметром:

* Градиент

Вызывает соответствующий метод CorrectCoeffs для каждого нейрона.

Вызов метода CorrectCoeffs для нейрона с параметром:

* Градиент

С помощью этого метода нейрон каждый из своих весов увеличивает на произведение градиента (поправки) на входное значение.

**Это все выполняется для каждой строки из входной матрицы.**

**Режим выполнения:**

В методе Work класса NeyroNet считывается введенный пользователем вектор (с клавиатуры или из файла).

Для него вызывается метод Exe для всей сети (описание работы Exe для сети см. выше),   
после чего результат работы сети выводится на экран.