

В файлах nn.* определены функции, связанные с расчетом рекуррентных нейросетей и классических перцептронов, а также описан алгоритм Левенберга-Марквардта для их обучения.

Расчет выходов перцептрона **NN_calc()** реализован последовательным определением активации нейронов следующего слоя при помощи матричных операций:

$$a_n = f(w \cdot a_{n-1} + b),$$

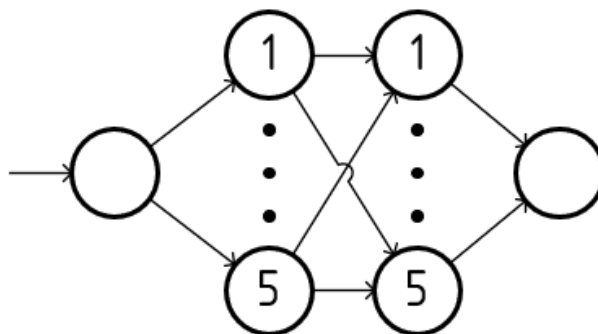
где: a_n и a_{n-1} – вектор активаций текущего слоя нейронов и предыдущего соответственно;

f – функция активации текущего слоя;

w – массив весов, связывающих слои;

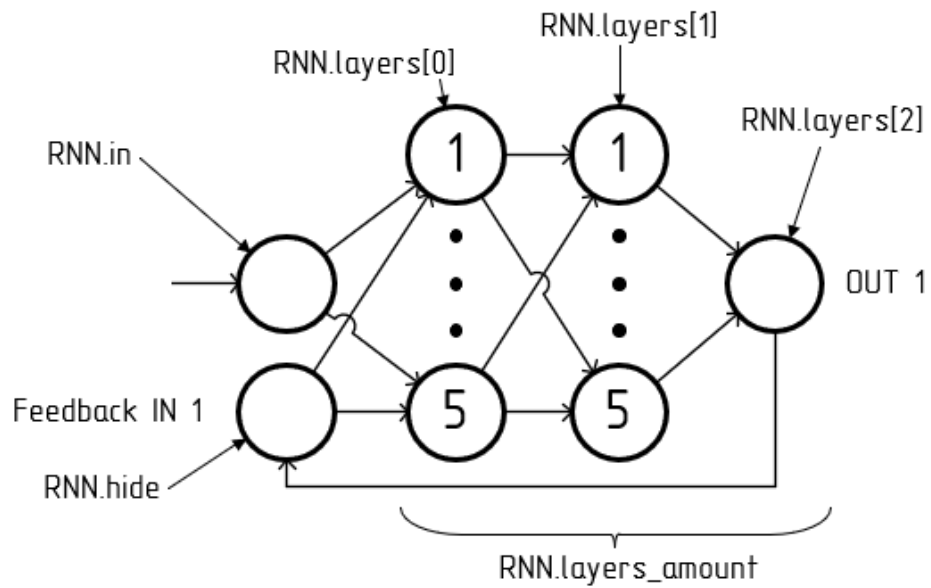
b – массив смещений текущего слоя;

Структура перцептрона задается матрицей, содержащей количество нейронов в каждом из слоев (включая входной и выходной слои):



$$layers = [1, 5, 5, 1]$$

Основываясь на функции **NN_calc**, реализована функция **RNN_calc**, выполняющая расчет выхода рекуррентной нейронной сети (РНС). Выход РНС является выходом перцептрона, однако каждый такт расчета на вход перцептрона кроме входных значений также подаются значений выходов перцептрона на предыдущем такте, что является реализацией механизма обратной связи. Для описания РНС есть специальная структура **RNN**, поля которой заполняются следующим образом:



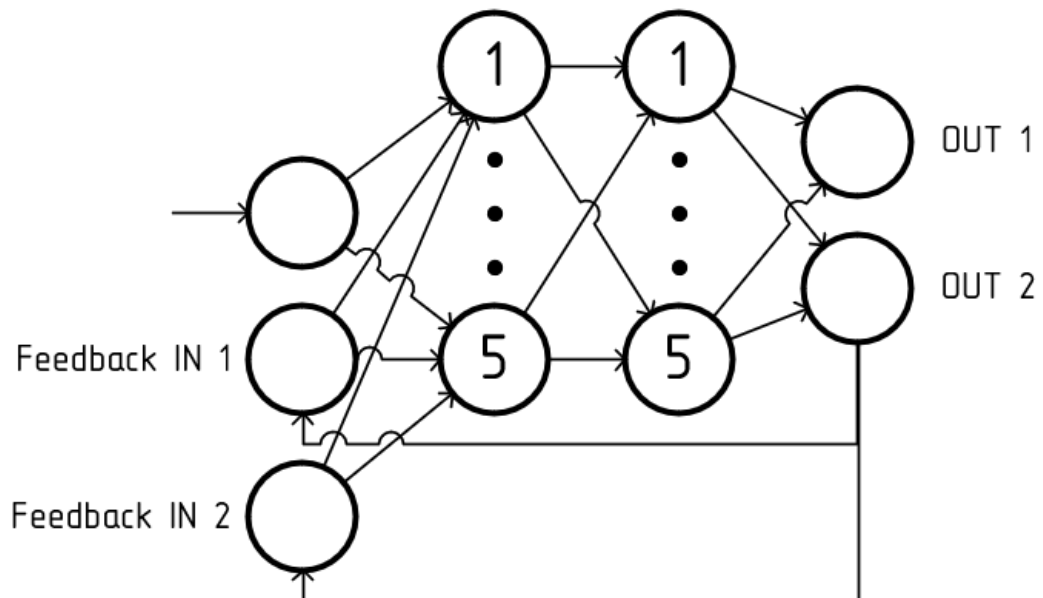
$RNN.in = 1;$

$RNN.hide = 1;$

$RNN.layers_amount = 3;$

$RNN.layers = [5, 5, 1];$

Также в структуре определена матрица $RNN.link$, которая содержит информацию о том, какой выход нейронной сети с каким входом обратной связи соединен. Например:



$$RNN.link = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix},$$

где номер строки – номер выхода, а номер столбца – номер входа обратной связи.

Перед обучением нейронной сети необходимо провести инициализацию начальными значениями ее параметров, для чего используются функции **RNN_par_amount** – определяет количество параметров РНС, **RNN_par_init** – инициализирует параметры РНС, **norm_rand** – генератор нормально распределённых чисел. Функция **RNN_par_init** на основе работы функции **RNN_par_amount** и при помощи **norm_rand** инициализирует веса нейронной сети методом инициализации Ксавье:

$$w \in N\left(0, \frac{2}{n_{in} + n_{out}}\right)$$

где: w – значение каждого отдельного веса;

N – функция нормального распределения, аргументами являются математическое ожидание и дисперсия;

n_{in} и n_{out} – количество нейронов в предыдущем и текущем слое соответственно.

Смещения инициализируются нулями, так как не влияют на прохождение сигнала через нейронную сеть, в отличие от весов.

При помощи вышеописанных функций и функций в файлах **matrix.*** можно описать модель объекта в виде функции, согласно следующим пунктам:

- заполнение структуры, описывающей нейронную сеть;
- реализация задержек входных данных;
- заполнение массива функций активации;
- запуск функции расчета выхода нейронной сети.

При этом необходимо соблюдать передаваемые в функцию модели параметры и их количество согласно примеру функции в **main.cpp**.

В файлах **nn.*** реализован алгоритм Левенберга-Марквардта обучения нейронных сетей согласно его классической формы для многопараметрической оптимизации. В ходе его работы используются:

– функция **mse()**, выполняющая расчет среднеквадратичного отклонения между двумя матрицами данных согласно формуле

$$mse = \sqrt{\frac{\sum_{i=1}^n (x_i^A - x_i^B)^2}{n-1}},$$

где: x^A и x^B – наборы данных;

n – количество элементов в наборах.

– функция **jacob()**, выполняющая расчет производных от ошибок между двумя наборами данных. Вычисление производных выполняется по следующей формуле:

$$\frac{df}{dp_i} = \frac{f(p_i + h) - f(p_i - h)}{2h}$$

где: f – функция ошибок;

p_i – i -тый параметр модели;

h – шаг дифференцирования.