Тема 2/Занятие 6/Упражнение

Софтуерни средства за моделиране на невронни мрежи.

Разработване на програмни решения на типични задачи от областта на идентификацията и разпознаване на образи с използване на невронни мрежи.

Да се анализира как форматът на структурите на входните данни влияе върху симулацията на мрежи. Мрежите могат да са статични и динамични.

1. Анализ на статични мрежи.

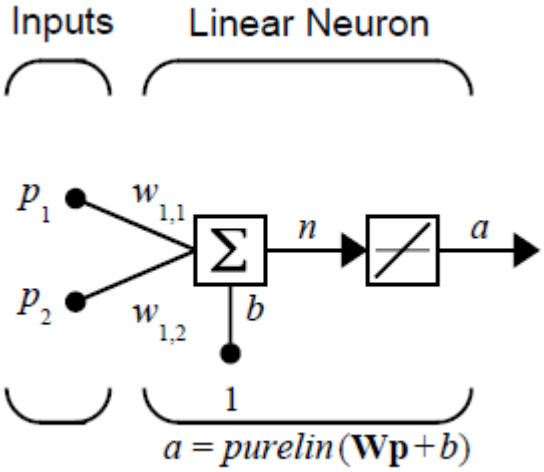
Ще бъдат разгледани два основни типа входни вектори: тези, които възникват едновременно (по едно и също време или в неопределена времева последователност) и тези, които се появяват последователно във времето.

За едновременни вектори редът не е важен и ако имахме няколко мрежи, работещи паралелно, можем да представим един входен вектор към всяка от мрежите. За последователни вектори, редът, в който се появяват векторите, е важен.

1.1. Симулация с едновременни входове в статична мрежа

Най-простата ситуация за симулиране на мрежа възниква, когато мрежата, която ще се симулира, е статична (няма обратна връзка или закъснения). В такъв случай, не е от значение дали входните вектори се случват в определена времева последователност, така че можем да третираме входовете като едновременни. В допълнение, ние правим проблема още по-прост, като приемем че мрежата има само един входен вектор.

Нека за пример да се използва следната мрежа.



За да настроим тази препращаща мрежа, можем да използваме следната команда.

net = newlin([1 3;1 3],1);

За простота задайте матрицата на теглото и отклонението, което трябва да бъде

W = [1 2] *b* =[0]

Командите за тези назначения са

net.IW{1,1} = [1 2];

net.b{1} = 0;

Да предположим, че наборът от данни за мрежова симулация се състои от Q = 4 едновременни вектори:

$$\mathbf{p}_1 = \begin{bmatrix}1\\2\end{bmatrix}, \quad \mathbf{p}_2 = \begin{bmatrix}2\\1\end{bmatrix}, \quad \mathbf{p}_3 = \begin{bmatrix}2\\3\end{bmatrix}, \quad \mathbf{p}_4 = \begin{bmatrix}3\\1\end{bmatrix},$$

Едновременните вектори се представят на мрежата като единична матрица: **P = [1 2 2 3; 2 1 3 1];**

Вече можем да симулираме мрежата:

A = sim(net,P)

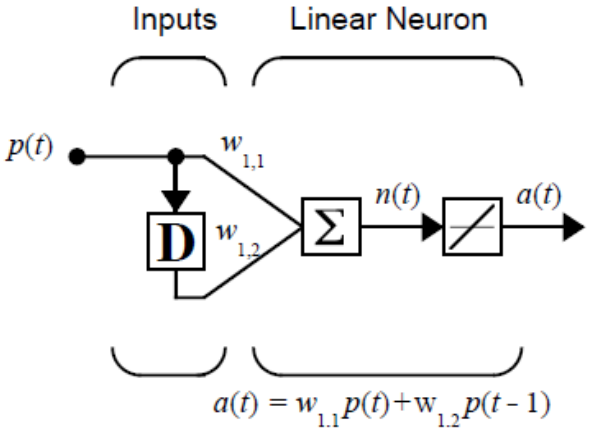
A =

5 4 8 5

Единична матрица от едновременни вектори се представя на мрежата и на мрежата произвежда единична матрица от едновременни вектори като изход. Резултатът би бил същият, ако имаше четири работещи мрежи паралелно и всяка мрежа получава един от входните вектори и произвежда един от изходите. Подреждането на входните вектори не е важно, тъй като те не взаимодействат помежду си.

1.2. Симулация с последователни входове в динамична мрежа

Когато мрежата съдържа закъснения, входът към мрежата е последователност от входни вектори, които се появяват в определен времеви ред. За да илюстрираме този случай, използваме проста мрежа, която съдържа едно забавяне.



Следните команди създават тази мрежа:

net = newlin([-1 1],1,[0 1]);

net.biasConnect = 0;

Задайте тегловната матрица, която да бъде

W = [1 2]

Командата е

net.IW{1,1} = [1 2];

Да предположим, че входната последователност е

$\mathbf{p}1 = \begin{bmatrix}1\end{bmatrix}, \ \mathbf{p}2 = \begin{bmatrix}2\end{bmatrix}, \ \mathbf{p}3 = \begin{bmatrix}3\end{bmatrix}, \ \mathbf{p}4 = \begin{bmatrix}4\end{bmatrix}$

Последователните входове се представят на мрежата като елементи от клетъчен масив:

P = {1 2 3 4};

Вече можем да симулираме мрежата:

A = sim(net,P)

A =

[1] [4] [7] [10]

Въвеждаме клетъчен масив, съдържащ последователност от входове, и мрежата произвежда клетъчен масив, съдържащ последователност от изходи. Имайте предвид, че редът на входовете е важен, когато са представени като последователност.

В този случай текущият изход се получава чрез умножаване на текущия вход по 1 и предходния вход по 2 и сумиране на резултата. Ако променим реда на входовете, това ще промени числата, които ще получим в изхода.

2. Симулация с едновременни входове в динамични мрежи

Ако трябваше да приложим същите входове от предишния пример като набор на едновременни входове вместо последователност от входове, бихме получили а напълно различен отговор.Би било така, сякаш всеки вход бяха приложени едновременно към отделна паралелна мрежа. За предишния пример, ако използваме паралелен набор от входове, които имаме

$$\mathbf{p}1 = \begin{bmatrix}1\end{bmatrix},\ \mathbf{p}2 = \begin{bmatrix}2\end{bmatrix},\ \mathbf{p}3 = \begin{bmatrix}3\end{bmatrix},\ \mathbf{p}4 = \begin{bmatrix}4\end{bmatrix},$$

който може да бъде създаден със следния код:

P = [1 2 3 4];

Когато симулираме с едновременни входове, получаваме

A = sim(net,P)

A =

1 2 3 4

Резултатът е същият, както ако бяхме приложили едновременно всеки един от входове към отделна мрежа и изчислен един изход. Имайте предвид, че оттогава ние не присвоихме никакви начални условия на мрежовите закъснения, те бяха се приема за нула. В този случай резултатът просто ще бъде 1 пъти по-голям вход, тъй като теглото, което умножава текущия вход, е 1. В някои специални случаи може да искаме да симулираме отговора на мрежата към няколко различни последователности едновременно. В този случай бихме искате да представите мрежата с едновременен набор от последователности. За например, да речем, че искаме да представим следните две последователности на мрежата:

$$\mathbf{p}_1(1) = \begin{bmatrix}1\end{bmatrix},\ \mathbf{p}_1(2) = \begin{bmatrix}2\end{bmatrix},\ \mathbf{p}_1(3) = \begin{bmatrix}3\end{bmatrix},\ \mathbf{p}_1(4) = \begin{bmatrix}4\end{bmatrix}$$

$$\mathbf{p}_2(1) = \begin{bmatrix}4\end{bmatrix},\ \mathbf{p}_2(2) = \begin{bmatrix}3\end{bmatrix},\ \mathbf{p}_2(3) = \begin{bmatrix}2\end{bmatrix},\ \mathbf{p}_2(4) = \begin{bmatrix}1\end{bmatrix}$$

Входът P трябва да бъде клетъчен масив, където всеки елемент от масива съдържа двата елемента от двете последователности, които се появяват едновременно:

P = {[1 4] [2 3] [3 2] [4 1]};

Вече можем да симулираме мрежата:

A = sim(net,P);

Полученият мрежов изход ще бъде

A = {[1 4] [4 11] [7 8] [10 5]}

Както можете да видите, първата колона на всяка матрица съставлява изхода последователност, произведена от първата входна последователност, която беше тази, която беше използвана в предходен пример. Втората колона на всяка матрица съставлява изходната последователност, произведена от втората входна последователност. Няма взаимодействие между двете едновременни последователности. Все едно са били приложени към отделни мрежи, работещи паралелно.

Литература:

1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на http://www.dkriesel.com/en/science/neural_networks, посетено на 12.08.2022 г.

2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.

3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5-03-002115-9.

4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0-13-273350-1.

5. Гульнара Яхъяева, Лекция 3. Персептроны. Обучение персептрон, достъпно на https://intuit.ru/studies/courses/88/88/print_lecture/20531

6. David Kriesel, A Brief Introduction to Neural Networks, достъпно на http://www.dkriesel.com/en/science/neural_networks