

ОБУЧЕНИЕ НА НЕВРОННА МРЕЖА ЗА ОКАЧЕСТВЯВАНЕ НА ХРАНИТЕЛНИ ПРОДУКТИ

гл.ас. Нанко Христов Бозуков, доц. д-р Чавдар Иванов Дамянов-ВИХВП-Пловдив

E-mail: nankob@hiffl-plovdiv.acad.bg

Training of a Neural Network for Quality Evaluation of Foods. In this paper presents algorithm and a Neural Network(NN) training program by the method of Backpropagation. The NN algorithm is able to adapt easily in operating conditions. In the paper are discussed the problems concerning onions quality evaluation using (NN). With data received from the expert evaluation concerning the appurtenance of onions to three qualitative fractions and containing information for the quality of the products, a training course on NN has been carried out. After a sufficient testing of NN with the known combinations of the sample input, NN is able to qualify with an accuracy admitted some new products, as well, beyond the excerpt used for training. In that way, the NN could make a quality evaluation of various food products.

Въведение

Оценката на качеството на хранителните продукти е свързано с редица проблеми, независимо дали те са за консумация в прясно състояние или за преработка. Съществуват различни начини за оценка на качеството на плодовете и зеленчуците за показатели като размер, форма и цвят [3]. Особено важно е окачествяването на лука (*Allium сера L*), една от основните зеленчукови култури в света [6]. Досега проблемът за автоматичното окачествяване и сортиране на лук не е решен задоволително [5]. Лукът, както и различни зеленчукови култури, като обекти за автоматично сортиране по качество са нехомогени и нестационарни структури, за чиято идентификация е необходима информация за цялостното им вътрешно и външно състояние. Тази информация може да бъде получена на базата на спектралната пропускливоност при сканирането на всяка луковица. Редица проблеми при окачествяването и сортирането могат да се решат като се разработят управляващи алгоритми за сортиращата машина, на основата на изкуствена невронна мрежа(НМ). Алгоритъмът за обратно разпространяване на грешката Back-propagation (BP) е в основата на много разработки с използването на НМ.

Материал и методи

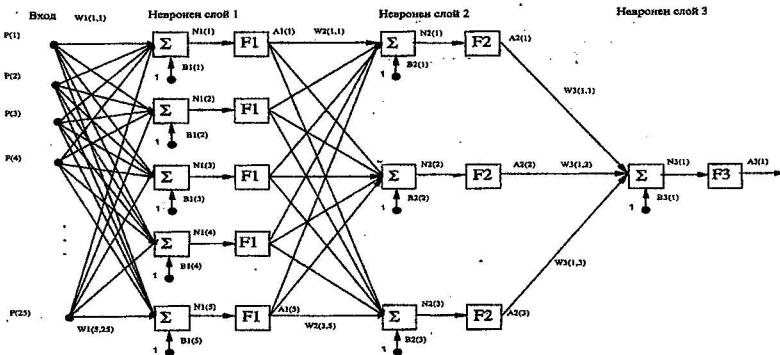
Лукът (*Allium сера L*) е една от основните зеленчукови култури. Зрелите луковици съдържат следните ценни компоненти, необходими за храненето на човека: 7-20% сухо вещество, а листата 7,2-11,8%; витамин С - средно 10 mg % към свежа маса; витамин В - средно 20 mg % към свежа маса. В

пепелта на лука се съдържат значителни количества SiO (16,72%), CaO (21,97%), Cl (2,77%), Na₂O (3,18%), etc. [5].

С помощта на група експерти при използването на фотометричен метод е формирана обучаваща извадка като матрица с 25 реда и 90 стълба за луковици в три качествени фракции.

Невронна мрежа

Изкуствените НМ се строят на принципа на техните биологични аналогии. Типичната НМ се състои от слоеве с неврони, както е показано на фиг.1. Входният и изходният слоеве са с толкова неврони, колкото са респективно входовете и изходите на НМ [6]. Броят на невроните във вътрешните (скритите) слоеве се определя според конкретната задача. Входният векторен сигнал U_o постъпва на първия слой, а изходите на последния m -ти слой образуват вектора на изходите U_m . Изходите на невроните на скрития k -слой постъпват на входовете на невроните само на следващия $k+1$ слой. В общия случай всеки j -ти неврон на i -тия слой (неврон # i,j) преобразува



Фиг.1 Трислойна невронна мрежа

входния вектор в скалар. Това преобразуване се извършва на две стъпки.

Алгоритъм

Алгоритъмът BP дава предписание за промяна на теглата на всяка прива мрежа, която трябва да научи обучаващо множество от двойки образци $\{U_o^{\mu}, D^{\mu}\}$. Основата на алгоритъма е спускане по градиента[2]. На фиг.1 е показана трислойна НМ със следните означения: входните неврони са означени с U_o^{μ} , скритите с U_j и изходните с Y_i . Алгоритъмът BP е следния:

1. Инициализират се теглата с малки случаини стойности w_{ij} , w_{ij} е теглото на връзката между неврон i и j .

2. Представят се обучаващите двойки(μ) на мрежата, където

$$U_o^{\mu} = [u_1^{\mu}, u_2^{\mu}, \dots, u_n^{\mu}] \text{ и } D^{\mu} = [D_1, D_2, \dots, D_r]$$

с n =броя на входовете, r =броя на изходите.

3. Изчислява се изхода за всеки слой.

$$U_i(j) = g(h_i^{\mu}) = g[\sum W_{ij} U_{i-1}^{\mu}]$$

(1)

Изходът на слой k е вход на слой $k+1$. Изходите на последния слой са:

$$U_m^{\mu} = Y_p^{\mu}, (p=1,2,\dots,r).$$

4. Изчисляват се делтите за изходния слой:

$$\delta_i^{\mu} = g'(h_i^{\mu})(D_p^{\mu} - Y_p^{\mu})$$

(2)

където g' е отклонението на активиращата функция g . Изчисляват се действителните изходи Y с желаните изходи на всеки изходен неврон.

Изчисляват се делтите на предишните(скрити) слоеве като се разпространява грешката обратно:

(3)

$$\delta_j^{\mu} = g'(h_j^{\mu}) \sum_k w_{jk}^{k+1} \delta_{k+1}^{k+1}$$

Уравнения (2) и (3) показват механизма за обновяване на теглата с намаляване на грешката.

5. Обновяват се теглата с

$$\Delta w_{ij} = \eta \delta_j^{\mu} Y_i^{\mu-1} \quad (\eta \text{ е стъпка на обучение}), \text{ започва се от изходния}$$

слой и се връща назад с

$$w_{ij}^{\mu+1} = w_{ij}^{\mu-1} + \Delta w_{ij} \quad (4)$$

δ_j^{μ} е периода дефиниран с (2) за изходните слоеве и (3) за останалите слоеве.

6. Определя се грешката за образец(μ):

(5)

$$e = 0,5 \sum_p (D_p^{\mu} - Y_p^{\mu})^2$$

и спира изпълнението на алгоритъма при достигане на определен толеранс, в противен случай се преминава отново към точка 2 и се повтарят действията за следващия образец.

Лесно е да се обобщи ВР за друг вид мрежи, където връзки прескачат един или повече слоеве. В такъв случай се запазва схемата за разпространяване на грешката, докато мрежата е права и без обратни връзки или връзки в рамките на един слой.

От направения преглед на различни НМ се установи, че най-подходящи са еднопосочните НМ, обучени с метода ВР и неговите модификации. Програмните реализации са извършени с продукта MatLab[10] и с тулбокса за НМ [8], който дава възможност за създаване на НМ максимум с три слоя и тяхното обучение.

Инициализиране и обучение на НМ

Невронните мрежи с размер до три слоя могат да се създадат и симулират с

Невронните мрежи с размер до три слоя могат да се създадат и симулират с

initff и *simuff* [8]. Такива мрежи могат да бъдат обучени с *back-propagation* обучаващи функции *trainbp* или с по-бързата версия на *back-propagation-trainbpx*. Друга функция е *trainlm*. Тя е дори по-бърза за обучаването на такива мрежи, но изисква големи размери на паметта. Първоначалните тегла и отклонения за *back-propagation* мрежи са създадени с *initff*. Тази функция взима матрица от входни вектори *p*, размера *S* и трансферните функции на всеки слой и връща теглата *W* и отклоненията *b* за всеки слой. Ето как е създадена двуслойна мрежа с осем(8) *tansig* скрити неврони и четири(4) *purelin* изходни неврона:

$$[W1,b1,W2,b2]=\text{initff}(p,8,\text{'tansig'},4,\text{'purelin'}); \quad (6)$$

Back-propagation обучаващо правило се използва за обучаване на нелинейни и многослойни мрежи, за предпочитане функционални апроксимации, примерна асоциация(връзка) и примерна класификация.

Три различни функции могат да се използват за обучение на поддържащи напреднали мрежи: *trainbp*, *trainbpm* и *trainlm*. Ето как *trainbp* се извиква за еднослойна мрежа с *tansig* неврони на матрица от входни вектори *p* и съответните цели(target)-*T*.

$$tp=[\text{disp_freq} \max_{-}epoch \text{ err_goal} \ lr]; \quad (7)$$

$$[W,b,\text{epochs},tr]=\text{trainbp}(W,b,\text{'F'},p,T,tp) \quad (8)$$

Давайки *P* и *T*, теглата *W*, отклоненията *b*, трансферната функция и обучаващите параметри *tp*, *trainbp* се връщат нови тегла и отклонения, броят на обучените епохи и записваното обучение на грешки *tr*.

Обучаващите параметри *tp*, определят броя на епохите между показания прогрес, максималния брой на епохи за обучение, сумарната средно-квадратична грешка и обучаващото ниво. Обучението продължава докато или грешката е срещната, или максималния брой епохи са намерени.

Обучаващото ниво определя размера на промените, които са направени в теглата и отклоненията на всяка епоха.

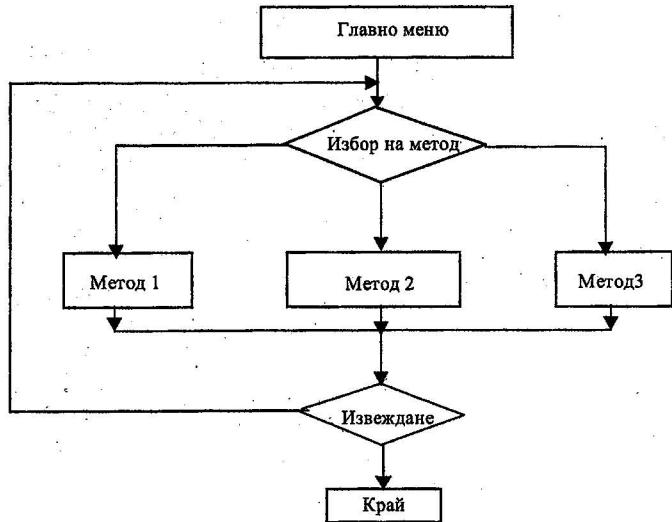
Обучаващата функция *trainbp* може да се използва за обучение на невронни мрежи с един, два или три слоя. За двуслойна мрежа *trainbp* се извиква по следния начин:

$$[W1,b1,W2,b2,\text{epochs},tr]=\text{trainbp}(W1,b1,\text{'F1'},W2,b2,\text{'F2'},p,T,tp) \quad (9)$$

Тук *W1*, *b1* и *F1* определят теглата, отклоненията и трансферната функция на първия слой. Променливите *W2*, *b2* и *F2*, дефинират втория слой. Ето кодът за количествено обучение на трислойна мрежа с *trainbp*:

$$\begin{aligned} & [W1,b1,W2,b2,W3,b3,\text{te},tr]= \\ & =\text{trainbp}(W1,b1,\text{'F1'},W2,b2,\text{'F2'},W3,b3,\text{'F3'},p,T,tp) \end{aligned} \quad (10)$$

Функцията *trainbp* използва три други функции *tbp1*, *tbp2* и *tbp3* за да обучи мрежата, в зависимост от това колко слоеве са в мрежата. Тази мрежа може да бъде използвана за генерален функционален апроксиматор [9].



Фиг.2 Блок-схема на програмата за обучение на НМ

Програма за обучение на НМ

Главната програма и трите програми, съответствуващи на трите метода за обучение са разработени в програмната среда на MatLab и с помощта на Toolbox-а [8] за невронни мрежи. Блок-схемата е показана на фиг.2.

От менюто на главната програма се избира един от методите за обучение, изпълнява се и програмата автоматично се връща в главното меню. От него може да се стартира същият или някой от другите два метода. В програмите при обучението и сортирането се определя бързодействието, а също така и точността на стартирания метод, а след завършване на програмата резултатите с всички променливи се съхраняват във файл. Оценяването на ефективността на алгоритмите при изпълнение на програмите се оценява по показателите: -брой на операциите с плаваща запетая; -общо време за изпълнение. По този начин може да се определи ефективността, бързодействието и точността на метода и да се извърши сравнение между методите.

Заключение

НМ са съвременно средство за окачествяване на хранителни продукти. Техните възможности са използвани при окачествяването на лук. Получените резултати дават основание да се твърди, че така създадения

алгоритъм за обучение на НМ изпълнява успешно своето предназначение. Окачествяването на лука чрез НМ е предпоставка за по-нататъшни изследвания и усъвършенстване на управляващия алгоритъм.

Най-добри резултати при тестването на обучената НМ по трите метода: ВР, ВР с моментум и ВР с моментум и адаптивна стъпка на обучение се получават с последния метод.

Литература

- [1] Атанасова Т. Подходи за интегриране на невронни мрежи в системите за управление. Автоматика и информатика, 1998, № 2, стр. 7-15.
- [2] Кирова Т.С. Невронни мрежи. Основни архитектури и обучаващи алгоритми, Софтех, С., 1995.
- [3] Георгиев А.С., Л.Ф. Костадинова, Ч.И.Дамянов, Възможности за практическо използване на вероятностните методи за разпознаване на образи в класификаторите на земеделски продукти по качество, Н. трудове на ТУ, Габрово том 1, 1999.
- [4] Дамянов Ч. И., Неразрушаващо невронно сортиране на картофи според съдържанието на сухи вещества. Сборник научни трудове на първа научна конференция "Електротехника и електроника", Габрово, 214-219, 3-5 декември, 1998.
- [5] Дамянов Ч. И., Георгиев А. С., Костадинова Л. Ф., Неразрушаващо автоматично сортиране по качество на лук и картофи, Научна конференция с международно участие "Храна и качество на живота-2000", Пловдив, 9-11 ноември, 2000.
- [6]] Дамянов Ч. И. Съвременни методи и средства за окачествяване и сортиране на плодове и зеленчуци., в книгата "Следберитбени технологии на плодове и зеленчуци, под ред. на проф. Танчев, ВИХВП, Пловдив, 2002.
- [7] Дъяконов В.П., Абраменкова И.В. MatLab 5.0/5.3. Система символьной математики.-М.:Нолидж, 1999.
- [8] Demuth H., M.Beale. Neural Network toolbox. For use with Matlab. User's Guide. The Mathworks, Inc., 1994
- [9] Hornik K., M. Stinchcombe, H.White. Multilayer Feedforward Networks are Universke Apptoximators. Neural Networks, 1989, 2, 359-366.
- [10] Matlab, High-Performahce Numeric Computation and Visualization Software, Classroom Kit Guide, The Mathworks, Inc., 1994.