УДК 004.048

Ю.В. УЛЬЯНОВСЬКА
Академія митної служби України, м. Дніпропетровськ

АНАЛІЗ АЛГОРИТМІВ НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ ТА ЇХ РЕАЛІЗАЦІЯ ПРИ АВТОМАТИЗАЦІЇ ПРОЦЕСУ ВИЗНАЧЕННЯ МИТНОЇ ВАРТОСТІ ТОВАРІВ

У роботі проаналізовані основні алгоритми, які використовуються для навчання нейронних мереж. Було промодельована робота нейронної мережі на сформованій множині навчальних векторів. На основі розглянутих методів проведено навчання мережі з різними показниками швидкості навчання. Отримані значення вагових коефіцієнтів вихідного шару та вектори зсувів.

Ключові слова: нейронні мережі, митна вартість

In article the basic algorithms which are used for training of neural networks are analyzed. Work of a neural network has been simulated on the generated set of training vectors. Training of a neural network with various indicators of speed of training are spent on the basis of the considered methods. Values of weight factors are received.

Keywords: neural networks, the customs value.

Вступ. Одним з ключових державних органів управління є Державна митна служба України (ДМСУ), призначення якої полягає в захисті економічних інтересів держави в міжнародних економічних відносинах. Важливим аспектом в діяльності митної служби є здійснення митного контролю, однією з складових частин якого є задача класифікації та визначення митної вартості товарів та об'єктів митного контролю. Визначення митної вартості товарів має безпосередній і визначальний вплив як на створення конкурентного середовища на національному ринку, так і на доходи державного бюджету від міжнародної торгівлі [1].

Визначення митної вартості зажди розглядалось як пріоритетне завдання в галузі реалізації митної справи ДМСУ. Така увага обумовлена великою кількістю ризиків, які пов'язані із можливістю ухилення від сплати юридичними та фізичними особами необхідних податків шляхом неправильного визначення та заниження митної вартості. Зазначені ризики обумовлені зокрема такими факторами як [2]: великий об'єм вхідної інформації, неточність (не достовірність) первинних даних, стислі строки прийняття рішення, складність організації УКТЗЕД, не достатній рівень знань в специфічних сферах таких як хімічна промисловість, легка промисловість, машинобудування, електроніка, сільське господарство тощо. Для зменшення ризиків на сьогоднішній день в ДМСУ створена та впроваджена автоматизована системи аналізу та керування ризиками. Враховуючи зміни у законодавстві та динамічність процедур митного оформлення алгоритми визначення митної вартості потребують автоматизації і мають бути адаптованими до можливих змін нормативної бази. Для розв'язання цієї проблеми у роботі [3] розроблено нейронну мережу для розв`язання сформульованих вище задач на прикладі визначення митної вартості взуття, яка після формування навчальної множини векторів входу та вектору бажаних відгуків (цілей), цілком готова до навчання. Схема нейромережі наведена на рис. 1.

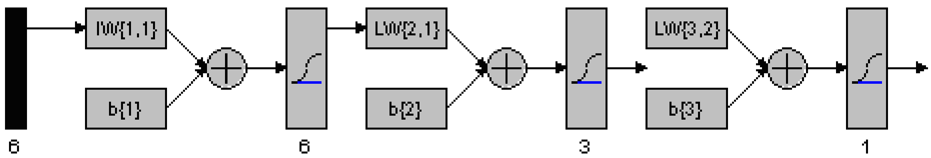


Рис. 1. Схема нейронної мережі

Метою даної роботи є аналіз алгоритмів навчання нейронних мереж, апробація їх на побудованій нейронній мережі, вибір оптимального алгоритму та навчання нейронної мережі.

Результати дослідження. Методи, що використовуються при навчанні нейронних мереж, багато в чому аналогічні методам визначення екстремуму функції декілька змінних. У свою чергу, останні діляться на 3 категорії – методи нульового, першого і другого порядку [4].

У методах *нульового порядку* для знаходження екстремуму використовується тільки інформація про значення функції в заданих точках.

У методах *першого порядку* використовується *градієнт функціонала помилки* по параметрах, що настроюються:

$$x_{k+1} = x_k - a_k g_k \tag{1}$$

де x_k – вектор параметрів;

a_k – параметр швидкості навчання;

g_k – градієнт функціонала, які відповідають ітерації з номером k.

Певну проблему тут викликає вибір параметра швидкості навчання a_k. При великому значенні параметра a_k збіжність буде швидкою, але існує небезпека пропустити рішення або піти в неправильному напрямі. Навпаки, при малому кроці, ймовірно, буде вибрано вірний напрям, але при цьому буде потрібно дуже багато ітерацій. Правильний вибір цього параметра залежить від конкретного завдання і зазвичай здійснюється шляхом проб; у разі змінного параметра його значення зменшується поволі приближення до мінімуму функціонала.

Методи другого порядку вимагають знання других похідних функціонала помилки. До методів другого порядку відноситься метод Ньютона. Основний крок методу Ньютона визначається по формулі:

$$x_{k+1} = x_k - H_k^{-1} g_k , \tag{2}$$

де x_k – вектор значень параметрів на *k*-ї ітерації;

H – матриця других частних похідних цільової функції, або матриця Гессе;

g_k - вектор градієнта на k -ї ітерації.

Зазвичай метод Ньютона сходиться швидше, ніж методи найшвидшого спуску, але вимагає великих затрат часу та пам’яті із-за обчислення гессіана. Для того, щоб уникнути обчислення матриці Гессе, пропонуються різні способи її заміни наближеними виразами, що породжує так звані *квазіньютонові алгоритми* (алгоритм методу січних площин OSS, алгоритм LM Левенберга – Марквардта [5]).

Процес навчання нейронних мереж пов’язаний з такою настройкою її вагових коефіцієнтів і зсувів, щоб мінімізувати деякий функціонал, що залежить від помилок мережі, тобто різниці між бажаним і реальним сигналами на її виході.

Під час навчання нейронної мережі в якості такого функціоналу буде використовуватись середня квадратична помилка:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} e_i^2 , \tag{3}$$

де N – розмір вектору помилок нейронної мережі;

e_i – *i*-та помилка вектору помилок нейронної мережі.

Вектор помилок:

$$E = Y - T , \tag{4}$$

де Y – вектор відгуків;

T – вектор бажаних відгуків.

Значення вагових коефіцієнтів вхідного шару:

$$IW_{S^1 R} = \begin{bmatrix} -1,5489 & 1,9866 & -2,1466 & -0,2392 & -1,5143 & -2,0971 \\ -1,4632 & -1,8304 & -1,5146 & 2,0761 & 0,8186 & 1,7892 \\ -1,9837 & -1,4504 & -0,9407 & 1,9132 & 2,0570 & -0,3077 \\ -1,9507 & 2,2870 & 0,8469 & -1,0898 & 0,7631 & 1,8059 \\ 0,7705 & -0,3295 & -4,1614 & -1,8588 & 2,0267 & 1,2854 \\ -1,3483 & -0,6977 & -4,0293 & 1,6264 & -2,1377 & 0,8171 \end{bmatrix},$$

де S^1 – кількість нейронів вхідного шару; R – розмір вхідного вектору.

Значення вагових коефіцієнтів прихованого шару:

$$LW_{S^2 a^1} = \begin{bmatrix} -1,4852 & -2,9812 & -0,0941 & -0,4116 & 3,8761 & -4,3512 \\ -3,1714 & -0,7364 & 3,0002 & -0,4683 & -4,6950 & 1,8345 \\ -4,1062 & 4,2444 & -0,4678 & -1,0466 & -2,4156 & 1,7898 \end{bmatrix},$$

де S^2 – кількість нейронів прихованого шару; a^1 – вектор входів для прихованого шару. Значення ваг вихідного шару: $OW_{S^3 a^2} = \begin{bmatrix} 5,4529 & 0,5947 & -1,1282 \end{bmatrix}$

Вектори зсувів мають наступні значення.

Для першого шару: $B_1 = \begin{bmatrix} 13,8202 \\ 1,7563 \\ 1,7092 \\ -5,6562 \\ 4,7181 \\ 5,7501 \end{bmatrix}$, для другого шару $B_2 = \begin{bmatrix} 6,0862 \\ 2,1182 \\ -2,3616 \end{bmatrix}$, для вихідного шару: $B_3 = \begin{bmatrix} -2,4597 \end{bmatrix}$

Промоделюємо роботу нейронної мережі на сформованій множині навчальних векторів. В результаті отримаємо наступний вектор відповідей мережі:

$$Y = [0{,}9449 \quad 0{,}8855 \quad 0{,}9464 \quad 0{,}9164 \quad 0{,}9292 \quad 0{,}8336 \quad 0{,}9516 \quad 0{,}9522 \quad 0{,}9467 \quad 0{,}9520 \ldots$$
$$0{,}9480 \quad 0{,}9523 \quad 0{,}9512 \quad 0{,}9505 \quad 0{,}9523 \quad 0{,}9516 \quad 0{,}9522 \quad 0{,}9523 \quad 0{,}9504 \quad 0{,}9504 \ldots$$
$$0{,}9513 \quad 0{,}9387 \quad 0{,}9552]$$

Відповідно до (4) отримаємо вектор помилок:

$$E = [0{,}8949 \quad 0{,}7655 \quad 0{,}8264 \quad 0{,}7164 \quad 0{,}6292 \quad 0{,}4336 \quad 0{,}8516 \quad 0{,}7722 \quad 0{,}6967 \quad 0{,}6720 \ldots$$
$$0{,}5980 \quad 0{,}7023 \quad 0{,}6512 \quad 0{,}9505 \quad 0{,}1505 \quad 0{,}7523 \quad 0{,}6716 \quad 0{,}6022 \quad 0{,}5523 \quad 0{,}4004 \ldots$$
$$0{,}0013 \quad 0{,}3387 \quad 0{,}5052]$$

середня квадратична помилка відповідно до (3): *MSE*=0.3890.

Для навчання *за алгоритмом найшвидшого спуску* використаємо алгоритм *GDM*. Алгоритм *GDM*, або алгоритм градієнтного спуску, з обуренням дозволяє долати локальні нерівності поверхні помилки і не зупинятися в локальних мінімумах. З урахуванням обурення метод зворотного розповсюдження помилки реалізує наступне співвідношення для приросту вектора параметрів, які настроюються:

$$\Delta w_k = mc\Delta w_{k-1} + (1-mc)lr * g_k\,, \tag{5}$$

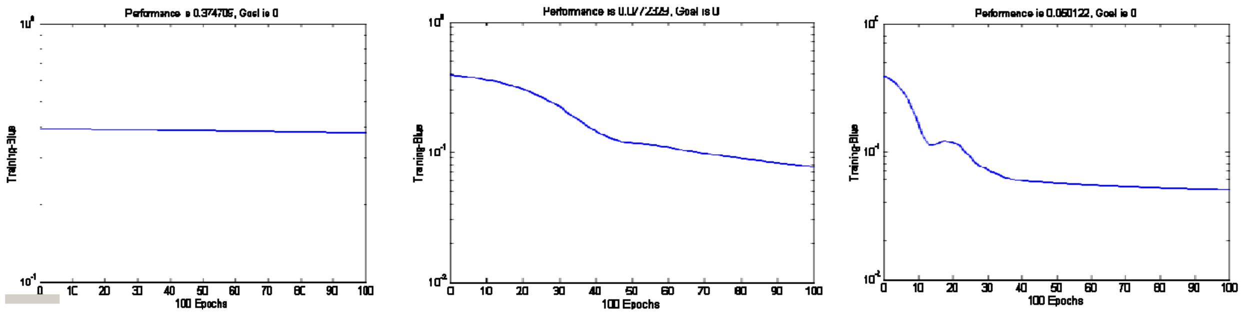
де Δw_k – приріст вектора вагів;

mc – параметр обурення;

lr – параметр швидкості навчання;

g_k – вектор градієнта функціонала помилки на *k*-ій ітерації.

Проведемо навчання мережі. Аналізуючи графік якості навчання мережі (рис. 2 (а)), можна сказати, що за 100 епох мережа натренувалась із показником якості навчання *MSE=0,3747*, це дуже суттєва помилка навчання. Змоделюємо навчання з показником швидкості навчання lr *=0,2* (рис. 2, (б)). Із графіка видно, що зі збільшенням показника швидкості навчання lr за ту ж саму кількість циклів навчання, нейронна мережа вчиться швидше та якісніше. Але при надто високому показнику lr процес навчання може стати нестабільний (рис. 2 (в)). Бачимо, що на 11-му циклі відбувся скачок функції якості навчання.



а) б) в)

Рис. 2. Графік якості навчання мережі за алгоритмом GDM з показником швидкості навчання lr=0,01 (а), lr=0,2 (б), lr=0,9 (в)

Для навчання за *квазіньютоновим алгоритмом* використаємо алгоритм *LM* (Левенберга – Марквардта). Алгоритм LM реалізує наступну стратегію для оцінки матриці Гессе. У припущенні, що функціонал визначається як сума квадратів помилок, що характерно для навчання нейронних мереж з прямою передачею, гессіан може бути приблизно обчислений як:

$$H \cong J^T J\,, \tag{6}$$

а градієнт розрахований по формулі:

$$g = J^T * E\,, \tag{7}$$

де $J = \dfrac{\partial J}{\partial W}$ – матриця Якобі похідних функціонала помилки по параметрам, що настроюються;

E – вектор помилок мережі.

Матриця Якобі може бути обчислена на основі стандартного методу зворотного розповсюдження помилки, що істотно простіше за обчислення матриці Гессен.

Алгоритм LM використовує апроксимацію гессиана наступного вигляду:

$$x_{k+1} = x_k - (J^T * J + \mu * J)^{-1} * J^T E_k\,, \tag{8}$$

де x_k – вектор параметрів, що налаштовуються.

Коли коефіцієнт μ рівний 0, ми отримуємо метод Ньютона з наближенням гессіана у формі (6); коли значення μ велике, отримуємо метод градієнтного спуску з маленьким кроком. Оскільки метод Ньютона має велику точність і швидкість збіжності поблизу мінімуму, завдання полягає в тому, щоб в процесі мінімізації щонайшвидше перейти до методу Ньютона. З цією метою параметр μ зменшують після кожної успішної ітерації і збільшують тільки тоді, коли пробний крок показує, що функціонал помилки зростає. Така стратегія забезпечує зменшення помилки після кожної ітерації алгоритму.

Продемонструємо навчання нейронної мережі (рис. 3).

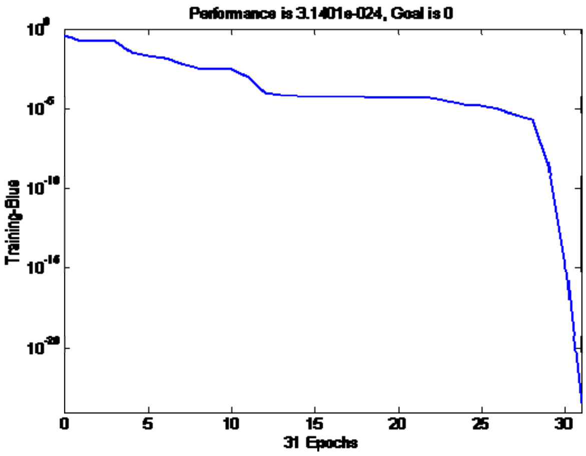


Рис. 3. Навчання нейронної мережі з використанням алгоритму LM.

Як видно, для навчання мережі з використанням алгоритму *LM*, достатньо було всього лише 31 циклів навчання, при цьому точність навчання: $MSE=3,1401*10^{-24}$.

З малюнку 4 видно, що на 3 – 7 циклах навчання фактично реалізується метод градієнтного спуску, а потім метод Ньютона з апроксимацією гессіана.

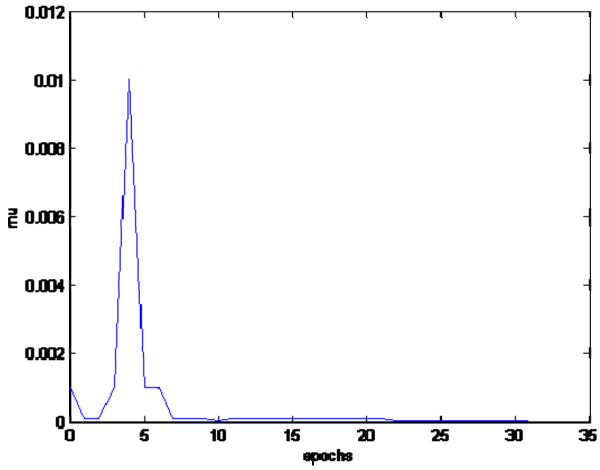


Рис. 4 Графік залежності μ від циклів навчання

Значення вагових коефіцієнтів вхідного шару:

$$IW_{S^1R} = \begin{bmatrix} -1,2101 & -1,5655 & -1,7275 & -4,4855 & -0,3243 & -0,4411 \\ -1,0291 & -1,3742 & -1,2747 & 3,5570 & 0,2269 & -0,4468 \\ -0,1774 & -0,6301 & 0,3041 & 3,1541 & 3,4002 & 1,9478 \\ -0,7038 & 0,8446 & 1,6952 & 1,0466 & 2,1903 & -0,1431 \\ 2,6719 & 0,8185 & -4,0864 & -2,4034 & 0,6024 & 1,8790 \\ -0,4799 & -1,7896 & -1,1248 & 3,5571 & 1,2210 & -2,2509 \end{bmatrix}$$

Значення вагових коефіцієнтів прихованого шару:

$$LW_{S^2a^1} = \begin{bmatrix} -0,9631 & -2,7541 & -1,6483 & 3,5706 & 4,1056 & -4,8642 \\ -7,2255 & -4,6638 & 2,1692 & 1,2008 & -4,8139 & 1,4590 \\ -4,3806 & 2,1099 & -1,1772 & -1,6189 & -4,4028 & 1,8326 \end{bmatrix}$$

Значення вагових коефіцієнтів вихідного шару: $OW_{S^3a^2} = \begin{bmatrix} 4,6183 & 6,8166 & 1,4402 \end{bmatrix}$. Вектори зсувів мають наступні значення

Для першого шару $B_1 = \begin{bmatrix} 14,6190 \\ 0,9189 \\ 2,2050 \\ -7,0356 \\ 3,3496 \\ 8,3043 \end{bmatrix}$, для другого шару $B_2 = \begin{bmatrix} 5,3129 \\ 1,5548 \\ -4,5202 \end{bmatrix}$, для вихідного шару $B_3 = [-5,6520]$.

Висновок. У роботі проаналізовані алгоритми навчання нейронних мереж та проведено навчання раніше розробленої нейронної мережі. Проаналізувавши динаміку змін значень вагових коефіцієнтів для матриці вхідного шару, можна зробити висновок, що деякі параметри змінюються лише на початку, а деякі – протягом всього часу навчання. В першу чергу така поведінка обумовлена множиною навчальних векторів. Якщо навчальні вектори змішати і подати в іншому порядку це буде виглядати по іншому. В цьому і полягає унікальність окремої нейронної мережі, оскільки при кожній новій задачі вона щоразу буде навчатись по іншому, не кажучи про зміну архітектури, функції навчання, функції активації і т.д.

Література

1. Альміз В. Теоретичні аспекти проблем визначення та коригування митної вартості як фактор забезпечення економічної безпеки держави [Текст] / В. Альміз // Вісник Академії митної служби. – 2006. – № 1. – С.3-5.
2. Ульяновська Ю.В. Дослідження структури об’єктів митного контролю з метою їх класифікації [Текст] / Ю.В. Ульяновська // Вісник Академії митної служби України. – 2007. – № 2 (13). – С. 98-102.
3. Ульяновська Ю. В. Застосування нейронних мереж до визначення митної вартості товарів. [Текст] / Ю.В. Ульяновська // Восточно-европейский журнал передовых технологий. – 2012. – № 3 (3). С 46-49.
4. Медведев В.С. НЕЙРОННЫЕ СЕТИ Matlab 6. / В.С. Медведев, В.Г. Потемкин. – М.: Диалог-Мифи, 2002. – 496 с.
5. Hagan M.T. Training feedforward networks with the Marquardt algorithm IEEE Transactions on Neural Networks. 1994. Vol. 5, N 6. P. 989-993.

Надійшла 27.1.2013 р.
Рецензент: д.т.н. Яковенко В.О.