УДК 004.415:378.091.212.2

Осадчий В.В.
Мелітопольський державний педагогічний університет імені Богдана Хмельницького

Круглик В.С.
Мелітопольський державний педагогічний університет імені Богдана Хмельницького

Осадча К.П.
Мелітопольський державний педагогічний університет імені Богдана Хмельницького

Сердюк І.М.
Мелітопольський державний педагогічний університет імені Богдана Хмельницького

Букреєв Д.О.
Мелітопольський державний педагогічний університет імені Богдана Хмельницького

ОСОБЛИВОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАСОБУ ДЛЯ ПРОГНОЗУВАННЯ ВСТУПУ АБІТУРІЄНТІВ ДО ЗАКЛАДІВ ВИЩОЇ ОСВІТИ

У статті відображено етапи й особливості розробки програмного засобу для прогнозування вступу абітурієнтів до закладів вищої освіти. Автори розглядають особливості розробки математичного апарату майбутнього програмного засобу та розкривають перелік використаних у програмному засобі функцій.

Ключові слова: *програмний засіб, прогнозування, нейронні мережі, статистика, управління.*

Постановка проблеми. Нині доцільно використовувати методи нейронних мереж у задачах із неповною інформацією або інформацією з великою кількістю шумів, особливо в задачах, де рішення можна знайти інтуїтивно, але традиційні математичні моделі не дають бажаного результату. Властивість мережі навчатися на прикладах робить її більш привабливою порівняно із системами, які працюють за заздалегідь закладеним правилом [1]. Методи нейронних мереж можуть використовуватися незалежно від інших або ж служити одним із найкращих доповнень до традиційних методів статистичного аналізу, більшість із яких пов'язані з побудовою моделей, заснованих на тих чи інших припущеннях і теоретичних висновках. Нейромережевий підхід однаково придатний для лінійних і складних нелінійних залежностей, особливо ж ефективний у розвідувальному аналізі даних, коли ставиться мета з'ясувати, чи є залежності між змінними. Дані можуть бути неповними, суперечливими і навіть свідомо спотвореними. Якщо між вхідними та вихідними даними існує якийсь зв'язок, навіть такий, який неможливо вирахувати за допомогою традиційних кореляційних методів, то нейронна мережа здатна автоматично налаштуватися на неї з заданим ступенем точності. У час постійної модернізації світу ці зміни не проходять без проблем і в освітньому просторі, прогнозування вступу абітурієнтів до закладів вищої освіти стає все більш неточним, тому було зроблено висновок про доцільність використання нейромережевого підходу у прогнозуванні результатів вступної компанії.

Аналіз останніх досліджень і публікацій. До вивчення проблематики аналізу та порівняння методів навчання нейронних мереж у різні часи долучалися провідні вчені світу, серед них: С.А. Федосин, Д.О. Ладяєв, О.А. Мар'їна, Д.О. Васенко, Е.В. Пучков. Проблеми створення і застосування нейронних мереж досліджували П. Вассермен, Р. Ліпман, Х. Мохамад, Б. Перлмуттер, Д. Спешт, Д. Тархов, К. Фунахаши, С. Хайкін, Дж. Хопфілд, І. Чуча, Д. Шофелт, Л.Г. Комарцова, А.В. Максимов [3] та ін. Нейронні мережі Хопфілда і Хеммінга описував у своїх роботах С. Короткий. Основні етапи побудови інтелектуальних систем прийняття рішень розглядали О.В. Нестеренко, О.І. Савенков та О.О. Фаловський [5].

Водночас проблема використання нейромережевого підходу для вирішення завдання з прогнозування вступу до закладів вищої освіти і результатів вступної кампанії є недостатньо дослідженою.

Постановка завдання. Розробка математичного апарату для прогнозування вступу абітурієнтів до закладів вищої освіти.

Виклад основного матеріалу дослідження. Для того, щоб оцінити можливості алгоритму машинного навчання, ми повинні розробити кількісну міру своєї діяльності. Зазвичай ця міра продуктивності *P* є специфічною для завдання *T*, що виконується у системі, для подальшої роботи називатимемо цю міру «точністю моделі».

Ми часто вимірюємо точність моделі. Точність – це тільки один із прикладів оцінки правильності моделі. Ми можемо також отримати еквівалентну інформацію шляхом вимірювання частоти помилок, тобто частки прикладів, для яких модель виробляє неправильний висновок [3].

Імовірнісні моделі часто є ймовірними розподілами, тільки у неявному вигляді. Обчислення фактичного значення ймовірності, призначеного до певної точки у просторі, у таких моделях є нерозв'язним. У цих випадках необхідно розробити альтернативний критерій, який відповідає проектним завданням, або створити наближення до бажаного критерію. Для створення такого наближення виникає потреба у навчанні моделі, тож розглянемо його особливості на прикладі навчань із вчителем і навчанням без вчителя.

Навчання без вчителя і навчання з учителем – неформально визначені терміни. Багато технологій машинного навчання можуть бути використані для виконання обох завдань. Наприклад, правило ланцюга ймовірності говорить, що для вектора xER_n спільний розподіл можна розкласти:

$$p(x) = N_p (X_i \in x_1, x_2, \ldots, x_p) \quad (1)$$

Це розкладання означає, що ми можемо вирішити нібито неконтрольоване завдання моделювання *p(x)*, розділивши його на *p* підконтрольних проблем навчання. З іншого боку, ми можемо вирішити проблему навчання контролем навчання *p(y|x)* з використанням традиційних неконтрольованих технологій для вивчення спільного розподілу *p(x,y)* і виводячи:

$$P(y|x) = E_y\, p(x,y) \quad (2)$$

Для опису набору даних у вигляді матриці плану необхідно описати всі приклади як вектори однакового розміру. Це не завжди можливо. У таких випадках опис виконується не у вигляді набору даних, а у вигляді матриці з *m* рядків. Ми описуватимемо його як набір, що містить *T* елементів: *{x (1), x (2), ..., x (t)}*.

Одним зі способів вимірювання характеристик моделі є обчислення середньої квадратичної помилки моделі на тестовому наборі [5]. Якщо *y(test)* дає передбачення моделі на тестовому наборі, то середній квадрат помилки визначається:

$$MSEtest = -Y(y\ (test) - y\ (test))\ 1/2 \quad (3)$$

Інтуїтивно можна побачити, що ця наявність помилки зменшується до 0, коли *y (test) = y (test)*. Ми також можемо бачити, що

$$MSEtest = -\ ||\ y\ (test) - y\ (test)\ |\ 1/2 \quad (4),$$

тому похибка зростає щоразу, коли евклідова відстань між передбаченнями та ціллю збільшується.

Нам необхідно розробити алгоритм, який дозволить зменшити MSE_{test}, коли алгоритм дозволить отримати досвід спостереження *(x (train), y (train))*. Для цього потрібно мінімізувати середній квадрат помилки на навчальному наборі. Щоб звести до мінімуму MSE_{train}, ми можемо просто вказати, де її градієнт дорівнює 0 (рис. 1):

$$V_w MSE_{train} = 0 \quad (5)$$

$$V_w -\ ||\ y\ (train) - y\ (train)\ ||\ 2 = 0, \quad (6)$$

$$V_w\ ||\ X\ (train)\ sh - y\ (train)\ ||\ 2 = 0. \quad (7)$$

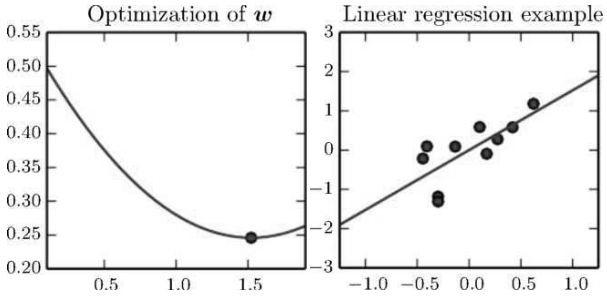

Рис. 1. Лінійна задача регресії, з навчальним набором, що складається з десяти точок даних, кожна з яких містить одну особливість

$$V_w(x(rain)_g - y(rain))\ T(x(rain)_g - y(rain)) = 0 \quad (8)$$

$$V_w(W*TX(rain)TX(rain)_g - 2wTX\ (rain)T_y(rain) + + y\ (rain)\ T_y\ (rain) = 0 \quad (9)$$

$$2X\ (rain)\ T_X\ (rain)_g - 2X\ (rain)\ T_y\ (rain) = 0 \quad (10)$$

$$S = \wedge X\ (rain)\ T_X\ (rain)\ 1x\ (rain)\ T_i(rain). \quad (11)$$

Система рівнянь, рішення якої дається формулою (11), відома як нормальне рівняння. (11) становить простий алгоритм навчання.

Варто зазначити, що термін лінійної регресії часто використовується для позначення дещо складнішої моделі з одним додатковим параметром-терміном *b* (12).

$$y = shTx + b \quad (12)$$

Відображення параметрів передбачення, як і раніше, є лінійною функцією, а відображення з особливими прогнозами – тепер афінна функ-

ція. Це розширення афінних функцій означає, що сюжет передбачень моделі, як і раніше, виглядає як лінія, але вона не повинна проходити через початок координат.

Більшість алгоритмів машинного навчання мають кілька параметрів, котрі можна використовувати для керування поведінкою алгоритму навчання. Вони називаються гіперпараметрами. Їх значення не пристосовані до самого алгоритму навчання.

Статистика пропонує багато інструментів, які можуть бути використані для вирішення задачі не тільки на навчальному наборі, але і для загального випадку. Такі основоположні поняття, як оцінки параметрів, помилка і дисперсії можуть бути використані, щоб формально характеризувати поняття узагальнення і перенавчання.

Оцінка точки є спробою забезпечити єдине «краще» пророкування деякої кількості інтересу. У загальному випадку кількість інтересу може бути одним параметром або вектором параметрів деякої параметричної моделі, як у прикладі лінійної регресії (13), або цілою функцією.

Для того, щоб відрізнити оцінки параметрів від їх справжнього значення, наша конвенція позначатиме точкову оцінку параметра b на 0.

Нехай *{x (1) x (m)}* деяка безліч *m* незалежних і однаково розподілених точок даних. Блок оцінки точки або статистики є будь-якою функцією даних:

$$bt = g\ (x\ (1)\ x\ (m)) \qquad (13)$$

Визначення не вимагає повертати значення g, близькі до дійсних *b*, або навіть ті, для яких діапазон g збігається з набором допустимих значень *b*. Це визначення точки оцінки є дуже загальним і дозволяє розробити блок оцінки з великою гнучкістю, тоді як майже будь-яка функція, таким чином, кваліфікується як оцінювач.

Оцінка точки може також стосуватися оцінки взаємозв'язку між вхідними і цільовими змінними. Згідно з моделлю Хопфельда ми називаємо ці типи точкових оцінок функціями оцінок [5].

Іноді ми зацікавлені у проведенні оцінки функції (або функції наближення). Тут ми намагаємося передбачити змінну у заданий вхідний вектор *x*. Ми припускаємо, що існує функція *F (X)*, яка описує приблизну залежність між *y* і *x*. Наприклад, ми можемо вважати, що $y = f(x) + e$, де *e* позначає частини *y*. В оцінці функції ми зацікавлені у наближенні *p* з моделлю або оцінкою.

Незміщена оцінка визначається як:

$$зміщення\ (m_0) = E\ (m_0) - 0 \qquad (14),$$

де очікування даних (розглядаються як випадкові величини) і 0 є істинним базовим значенням і використовується для визначення розподілу генерування даних.

Оцінка 0 називається незміщеною, якщо зсув $(e) = 0$, звідки випливає, що $E\ (0_i) = 0$. Оцінювач *bm* називається асимптотично незміщеним, якщо $limmbias\ (0_m) = 0$, звідки випливає, що $limmE\ (0_m) = 0$.

Згідно з формулою Бернуллі розподілу із середнім значенням 0:

$$P\ (x\ (z);\ 0) = 0_x\ (I)\ (1 - 0)\ (1\text{-}x\ (z)) \qquad (15)$$

Дисперсія або стандартна похибка блоку оцінки дає міру того, як можна було б очікувати оцінку, що обчислюється з даних, змінювати незалежно один від одного, як ми дискретизуємо набір даних з основного процесу генерування даних. Подібно до того, як ми могли б, як оцінювач, проявляти низьку упередженість, ми також хотіли б мати його відносно низьку дисперсію.

Коли ми обчислимо будь-яку статистику, використовуючи кінцеве число вибірок, наша оцінка істинного основного параметра є невизначеною, в тому сенсі, що ми могли б отримали інші зразки з того ж розподілу, і їх статистика матиме різні прогнози. Очікувана ступінь зміни будь-якої оцінки є джерелом помилок, які ми хочемо виміряти [5].

Стандартна похибка середнього значення є дуже корисною у машинному навчанні експериментів. Ми часто оцінюємо помилки узагальнення шляхом обчислення середнього значення вибірки помилки на тестовому наборі. Кількість прикладів у тестовому наборі визначає точність цієї оцінки. Так, наприклад, 95% довірчий інтервал по центру середнього значення JM є

$$(JM - 1.96SE\ (JM),\ Am + 1.96SE\ (JM)) \qquad (16)$$

У машинному навчанні експериментів прийнято говорити, що алгоритм *A* краще, ніж алгоритм *B*, якщо верхня межа 95% довірчого інтервалу для похибки алгоритму *A* менша, ніж нижня межа 95% довірчого інтервалу похибки алгоритму *B*.

Дисперсія оцінки зменшується як функція *m* у зв'язку з числом прикладів у наборі даних. Це загальна властивість популярних оцінок, які ми будемо повертати до того, як опишемо послідовність.

Алгоритми майже всіх глибоких навчань можна охарактеризувати як досить простий рецепт: об'єднати специфікації набору даних, функції витрат, процедури оптимізації та моделі [5].

Розуміючи, що ми можемо замінити будь-який із цих компонентів, переважно незалежно від інших, ми можемо отримати дуже широкий спектр алгоритмів.

Функція вартості, зазвичай, включає в себе щонайменше один член, який викликає процес

навчання, щоб виконати статистичну оцінку. Найбільш поширена функція витрат – це негативний логарифм правдоподібності, у якому мінімізація функції витрат призводить до оцінки максимальної правдоподібності.

Функція витрат може також включати в себе додаткові умови, такі як точки регуляризації. Так ми додаємо значення розпаду функції лінійної регресії витрат для отримання

$$J\ (sh,\ b) - L \mid\mid sh \mid g - Ex,\ y \sim PDATA\ 10g\ Pmodel\ (y \mid x) \qquad (17)$$

Це дозволяє оптимізувати замкнуту форму. Якщо ми змінимо модель на нелінійну, то більшість функцій витрат не може бути більше неоптимізованими в закритій формі. Це вимагає від нас вибрати ітераційну чисельну процедуру оптимізації, таку як градієнтний спуск.

Неконтрольоване навчання може підтримуватися шляхом визначення набору даних, що містить тільки *X* і забезпечення відповідної неконтрольованої вартості і моделі. Так ми отримаємо перший вектор *PCA*, вказавши, що це наша функція втрати:

$$J\ (sh) – Exp(PDATA) \mid\mid x - g\ (x;\ j) \mid\mid 2 \qquad (18),$$

тоді як наша модель визначається, щоб мати *sh* із нормою однією функцією відновлення *g (x) - sh TXW*.

У деяких випадках функція витрат може бути функцією, яку ми не можемо реально оцінити. Ми можемо звести її до мінімуму за допомогою ітераційної чисельної оптимізації настільки, наскільки у нас є певний спосіб апроксимації її градієнти.

Для реалізації поставлених завдань щодо програмного засобу, згідно з виявленим математичним апаратом, було розроблено ряд функцій.

Формули (8) – (18) дозволяють розрахувати основні параметри, що впливають на кількісний прогноз вступу абітурієнтів до закладів вищої освіти.

Згідно з формулами (8) – (11) розраховуємо стартову суму вхідних коефіцієнтів:

k1_sum += Math.Pow((arr[s, 1] - k1_sr), 2)

Після розрахунку кількісних мір графів виникає потреба у визначенні кожного вектора, відповідно до цього розраховуємо значення коефіцієнтів для подальшого розрахунку остаточного прогнозу:

y_k1_sum += (arr[s, 0] - y_sr) * (arr[s, 1] - k1_sr)

Після розрахунку коефіцієнтів проводимо остаточний прогноз згідно з формулами (15) – (17):

prognoz = b0 + b1 * arr[size - 1, 1] + … + bn *arr[size-1,n];

Після проведення розрахунків прогнозу оцінюємо рівень похибки за формулою (3) та, у разі низької точності, уточнюємо базу знань шляхом редагування результатів прогнозування (рис. 2).

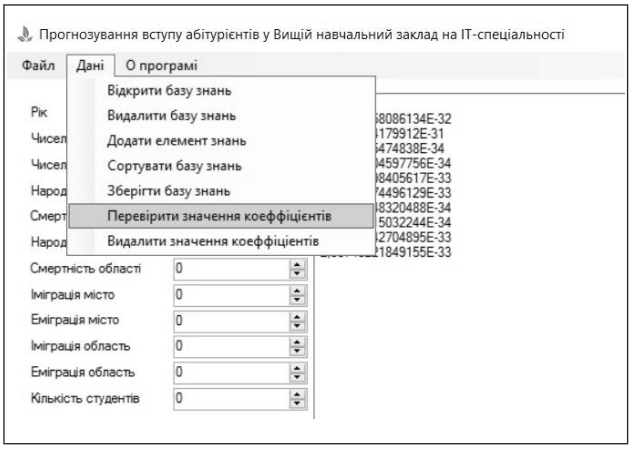

Рис. 2. Відображення перевірки коефіцієнтів

Для створення можливості коректного відображення інформації у зрозумілому для користувачів вигляді (графіки) було розроблено рід функцій [2], наприклад Kol_Vstup, які генерують графічне відображення даних за блоками (рис. 3).

```
privatevoidKol_Vstup(object sender, EventArgs e)
        {   double[] z = newdouble[size + 1];
double[] y = newdouble[size + 1];
for (int i = 0; i <size; i++)
          {  z[i] = i + 1;
             y[i] = arr[i, 11];
         chart1.ChartAreas[0].AxisX.Minimum = 0;
chart1.ChartAreas[0].AxisX.Maximum = size + 1;
chart1.ChartAreas[0].AxisX.MajorGrid.Interval = 1;
chart1.Series[0].Points.DataBindXY(z, y);  }
label2.Text = "Кількість вступників";  }
```

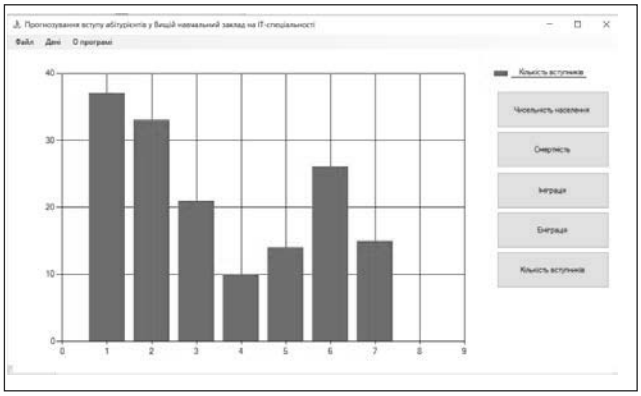

Рис. 3. Графічне відображення отриманих даних

Висновки. У дослідженні було визначено, що нині доцільно використовувати методи нейронних мереж у задачах із неповною інформацією або інформацією з великою кількістю шумів, особливо в задачах, де рішення можна знайти інтуїтивно, але

традиційні математичні моделі не дають бажаного результату. Було проаналізовано сфери використання нейронних мереж та особливості їх побудови, визначено основні проблеми прогнозування. З'ясовано, що ефективне рішення проблем прогнозування досягається лише в тому разі, коли нейронна мережа навчається на великому обсязі даних і використовується якісна навчальна вибірка, тоді алгоритм дасть задовільний результат і значення похибки прогнозування зменшиться до мінімального рівня, у зв'язку з чим було проаналізовано особливості розрахунку точності правильності моделі та визначено подальший вектор досліджень. Надалі планується доробка та розвиток програмного засобу з метою розширення його функціоналу і зменшення рівня похибки для більш точного відображення результатів прогнозування в умовах постійного коливання ринку навчальних послуг.

Список літератури:

1. Букреєв Д.О. Прогнозування фондового ринку за допомогою нейронних мереж. *Інформаційні технології в освіті та науці*. 2018. № 10. С. 36–43.

2. Осадчий В.В., Круглик В.С., Букреєв Д.О. Розробка програмного засобу для прогнозування вступу абітурієнтів до закладів вищої освіти. *Ukrainian Journal of Educational Studies and Information Technology*. 2018. Вип. 6. № 3. С. 55–69.

3. Комарцова Л.Г., Максимов А.В. Нейрокомпьютеры. М.: Изд-во МГТУ им. Баумана, 2004.

4. Карандашев Я.М., Крыжановский Б.В., Литинский Л.Б. Обобщенная модель Хопфилда и статфизический подход: общий случай. *Нейроинформатика-2011. XIII Всероссийская научно-техническая конференция:* сб. науч. трудов. М., НИЯУ МИФИ, 2010. Ч. 3. С. 181–190.

5. Нестеренко О.В., Савенков О.І., Фаловський О.О. Інтелектуальні системи підтримки прийняття рішень: навч. посіб. Київ, 2016.

ОСОБЕННОСТИ РАЗРАБОТКИ ПРОГРАММНОГО СРЕДСТВА ПРОГНОЗИРОВАНИЯ ВСТУПЛЕНИЯ АБИТУРИЕНТОВ В ВЫСШИЕ УЧЕБНЫЕ ЗАВЕДЕНИЯ

В статье отражены этапы и особенности разработки программного средства прогнозирования поступления абитуриентов в высшие учебные заведения. Авторы рассматривают особенности разработки математического аппарата будущего программного средства и раскрывают перечень использованных в программном средстве функций.

Ключевые слова: *программное средство, прогнозирование, нейронные сети, статистика, управление.*

FEATURES OF DEVELOPMENT OF THE SOFTWARE FOR FORECASTS OF APPLICATION OF ABITURIENTS TO HIGHER EDUCATION UNITS

The article presents the stages and peculiarities of developing a software tool for predicting the entrance of entrants to higher education institutions. The authors reflect the peculiarities of the development of the mathematical apparatus of the future software, and break the list of functions used in the software tool.

Key words: *software tool, forecasting, neural networks, statistics, management.*