

## Електронні таблиці як засіб навчання нейромережевого моделювання технічних об'єктів бакалаврів електромеханіки

Євгеній Олександрович Модло

Криворізький металургійний інститут Національної металургійної академії України, вул. Революційна, 5, м. Кривий Ріг, 50006, Україна  
modea@mail.ru

Ілля Олександрович Теплицький<sup>#</sup>, Сергій Олексійович Семеріков<sup>#</sup>  
Кафедра інформатики та прикладної математики, Криворізький державний педагогічний університет, пр. Гагаріна, 54, м. Кривий Ріг, 50086, Україна

ilia2306@gmail.com<sup>#</sup>, semerikov@gmail.com<sup>#</sup>

**Анотація.** *Метою* даного дослідження є розробка елементів методики використання мобільних електронних таблиць як засобу навчання нейромережевого моделювання технічних об'єктів бакалаврів електромеханіки.

*Задачі дослідження:*

1. Обґрунтування вибору нейронних мереж як засобу моделювання технічних об'єктів із прихованою або нечіткою структурою.
2. Реалізація алгоритму глибинного машинного навчання у середовищі мобільних електронних таблиць Google Sheets.

*Об'єкт* дослідження – процес професійної підготовки бакалаврів електромеханіки в технічних ЗВО, *предмет* – формування компетентності бакалавра електромеханіки в моделюванні технічних об'єктів.

*Результати:* 1) розроблено та реалізовано у середовищі мобільних електронних таблиць модель роботи інтелектуального датчику на основі нейронної мережі з двома прихованими шарами; 2) наведено приклади використання доповнення Solver для визначення вагових коефіцієнтів нейронної мережі.

*Основні висновки:* у процесі розробки елементів методики використання мобільних електронних таблиць як засобу навчання нейромережевого моделювання технічних об'єктів бакалаврів електромеханіки показано можливість та доцільність формування критичного мислення студентів.

**Ключові слова:** електронні таблиці; нейронні мережі; моделювання технічних об'єктів; критичне мислення; бакалаври електромеханіки; Google Sheets.

**Ye. O. Modlo<sup>\*</sup>, I. O. Teplytskyi<sup>#</sup>, S. O. Semerikov<sup>‡</sup>. Spreadsheets as a learning tools of bachelors in electromechanics for neural network modeling of technical objects**

**Abstract.** The *purpose of this study* is to develop elements of the method of using mobile spreadsheets as learning tools of bachelors in electromechanics for neural network modeling of technical objects.

*Research objectives:*

1. Justification of the choice of neural networks as tools for modeling technical objects with a hidden or fuzzy structure.

2. Implementation of the deep learning algorithm in the Google Sheets mobile spreadsheet environment.

The *object of research* is the process of professional training of bachelor in electromechanics at technical university, the *subject of research* is the formation of bachelor's in electromechanics competence in modeling technical objects.

*Results:* 1) a model of operation of an intelligent sensor based on a neural network with two hidden layers was developed and implemented into mobile spreadsheet; 2) the examples of the Solver add-on use to determine the weights of the neural network are given.

The main *conclusions:* in the process of developing elements of the method of using mobile spreadsheets as a learning tools of bachelors in electromechanics for neural network modeling of technical objects, the possibility and expediency of forming critical thinking of students was shown.

**Keywords:** spreadsheets; neural networks; modeling of technical objects; critical thinking; bachelors in electromechanics; Google Sheets.

**Affiliation:** Kryvyi Rih Metallurgical Institute of the National Metallurgical Academy of Ukraine, 5, Revoliutsiina Str., Kryvyi Rih, 50006, Ukraine<sup>\*</sup>;

Department of Computer Science and Applied Mathematics, Kryvyi Rih State Pedagogical University, 54, Gagarin Ave., Kryvyi Rih, 50086, Ukraine<sup>#</sup>.

E-mail: modea@mail.ru<sup>\*</sup>, ilia2306@gmail.com<sup>#</sup>, semerikov@gmail.com<sup>‡</sup>.

Серед загальнопрофесійних складових компетентності бакалавра електромеханіки в моделюванні технічних об'єктів важливе [1; 2] місце займає *компетенція із критичного мислення*, зміст якої складають знання та уміння постановки завдання з недостатньою кількістю вхідних даних, аналізу наявності способів і засобів виконання завдання, оцінювання власної готовності до розв'язування задачі, самостійного пошуку відсутніх даних та способів розв'язування задачі; уміння здійснювати контроль власної діяльності – як розумової, так і практичної; уміння контролювати логіку розгортання власних думок; уміння визначати

послідовність та ієрархію етапів діяльності, тощо. Так само, як й для компетенції із застосування різних способів подання моделей, для компетенції із критичного мислення не можна виокремити провідні навчальні дисципліни – формування цих складових відбувається протягом усієї професійної підготовки бакалавра електромеханіки [3].

Ефективним засобом моделювання технічних об'єктів з прихованою або нечіткою структурою, опис якої традиційними засобами ускладнений, є нейромережеве моделювання [4]. Як зазначають редактори другої книги серії «Нейрокомп'ютери та їх застосування», актуальність його застосування обумовлюється необхідністю розробки методів управління складними нелінійними системами: «Перші приклади розробки методів управління нелінійними системами ... пов'язані переважно з методами розв'язання нелінійних диференціальних рівнянь, адекватних однопроцесорним фон-Неймановським обчислювальним машинам. ... Розвиток обчислювальних машин з масовим паралелізмом ... привів до створення принципово нових алгоритмів і методів управління нелінійними динамічними системами. Вони пов'язані з нейромережевими алгоритмами розв'язання звичайних нелінійних диференціальних рівнянь та, як наслідок, із включенням нейрокомп'ютера в контур управління нелінійною динамічною системою. ... Достатньо широкий розвиток та поширення подібних алгоритмів призвели ... до створення цілого розділу науки під назвою «нейроуправління» [5, с. 9-10].

В інженерному контексті інтелектуальне управління повинно володіти наступними властивостями: здатність до навчання та адаптивність; живучість; простий алгоритм управління та дружній до користувача людино-машинний інтерфейс; здатність до включення нових компонентів, що забезпечують найкращі рішення в умовах обмежень, що накладаються технічними засобами [5, с. 15].

Під *глибинним машинним навчанням* розуміється клас алгоритмів інтелектуального управління, які використовують багатоваріантні нейронні мережі з нелінійними вузлами. Розглянемо побудову нейромережевої моделі для апроксимації даних, описану в 23 главі [6], та отриманих з інтелектуального датчику – одного або декількох стандартних датчиків, які пов'язані з нейронною мережею для отримання каліброваного вимірювання одного із параметрів.

Інтелектуальний датчик положення використовує значення напруги, що надходять від двох фотоелементів, для оцінки розташування об'єкта. На рис. 1 кругом показано об'єкт, розташований між джерелом світла і двома фотоелементами. Об'єкт, рухаючись вздовж осі  $y$ , відкидає тінь на фотоелементи, що призводить до зміни напруг  $v_1$  та  $v_2$ .

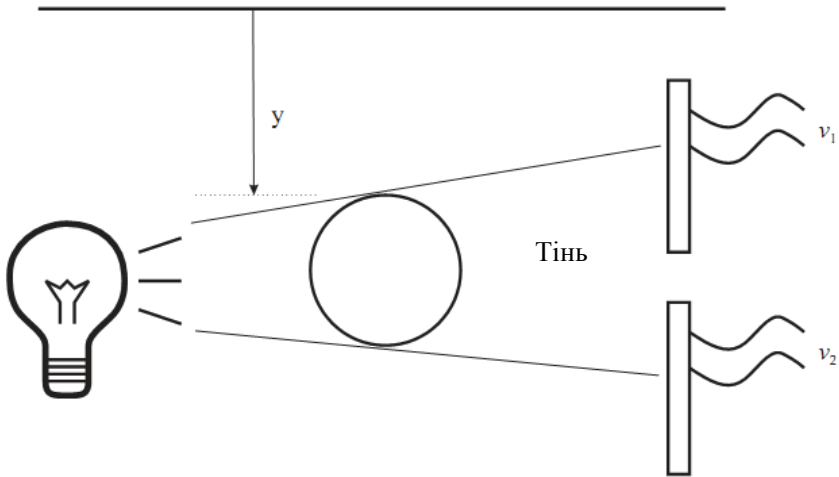


Рис. 1. Інтелектуальний датчик положення

При збільшенні позиції об'єкта  $y$  напруга  $v_1$  спочатку зменшується, потім зменшується напруга  $v_2$ , потім збільшується  $v_1$  і нарешті збільшується  $v_2$  (рис. 2).

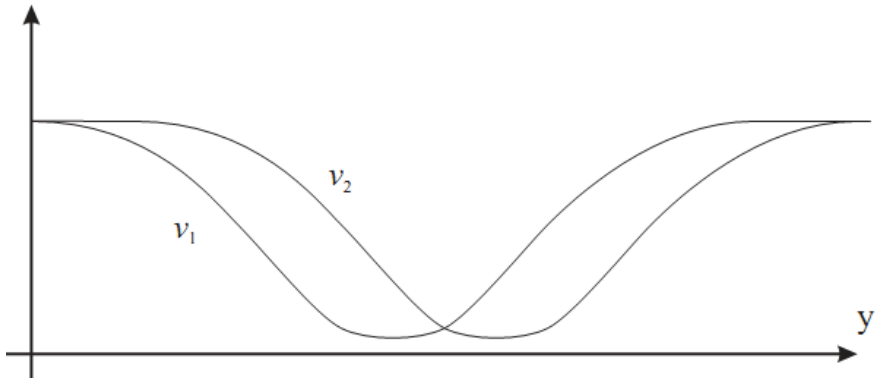


Рис. 2. Ідеальна модель залежності напруг  $v_1$ ,  $v_2$  від координати  $y$  об'єкту

Метою моделювання є визначити положення об'єкта за вимірюваннями двох напруг.

Для того, щоб зібрати дані для апроксимації, проводяться вимірювання двох напруг фотоелементів в ряді еталонних положень об'єкта. Автори [6] використовували для цих експериментів м'яч для настільного тенісу. Усього є 67 наборів вимірювань, представлених у файлах `ball_p.txt` та `ball_t.txt` в архіві за посиланням <http://hagan.okstate.edu/CaseStudyData.zip>. Кожна точка на графіку (рис. 3)

представляє вимір напруг в каліброваному положенні. Координати виміряні в дюймах, а напруга – у вольтах. Плоскі області в 0 вольт для кожної кривої виникають там, де тінь кулі повністю покриває датчик. Якщо тінь була досить великою, щоб покрити обидва датчики одночасно, ми не зможемо відновити координату за напругами.

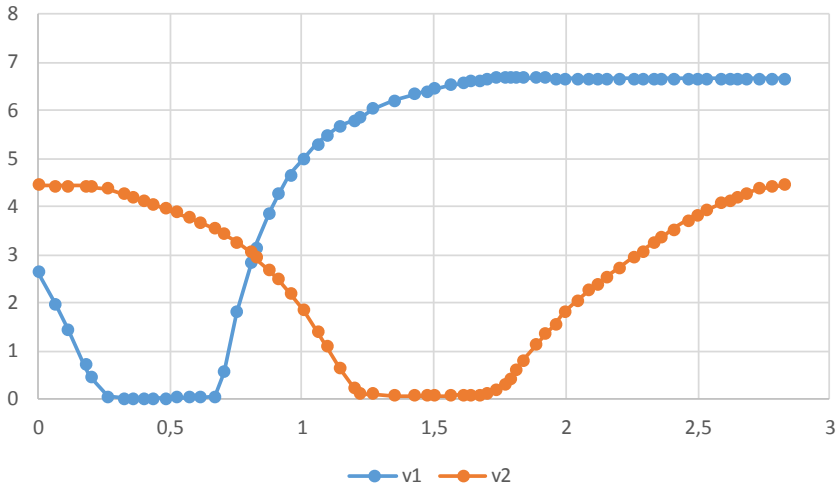


Рис. 3. Графік залежності напруг  $v_1$ ,  $v_2$  від координати у об'єкту

Для реалізації моделі глибинного навчання застосуємо хмаро зорієнтовані електронні таблиці Google Sheets із доповненням Solver за авторською методикою, описаною у [7; 8].

Для того, щоб визначити координату об'єкта, побудуємо чотиришарову нейронну мережу із архітектурою, поданою на рис. 4:

- вхідний шар – двовимірний арифметичний вектор  $(x_1, x_2)$ , компонентами якого є відповідні вимірювані напруги  $v_1, v_2$ , нормовані згідно з функцією активації мережі;

- перший прихований шар матиме розмірність 5 та описуватиметься вектором  $(h_1^{(1)}, h_2^{(1)}, h_3^{(1)}, h_4^{(1)}, h_5^{(1)})$ ;

- другий прихований шар матиме розмірність 3 та описуватиметься вектором  $(h_1^{(2)}, h_2^{(2)}, h_3^{(2)})$ ;

- вихідний шар – значення координати  $u_{\text{норм}}$ , нормована згідно з функцією активації мережі.

До нейронів вхідного та прихованих шарів додаються нейрони зміщення (bias), значення яких завжди дорівнює одиниці (на рис. 4 їх позначено червоним кольором). Особливістю нейронів зміщення є те, що вони не мають вхідних синапсів, а тому не можуть бути розташовані на

вихідному шарі.

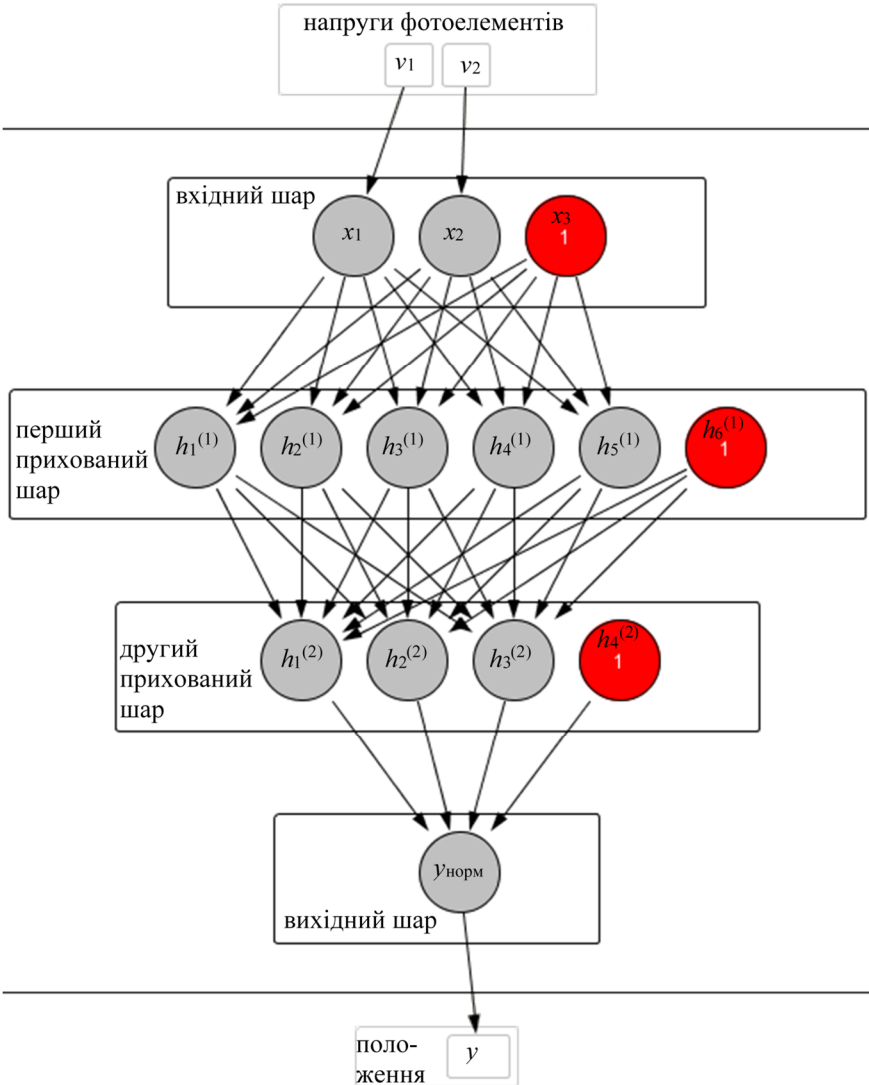


Рис. 4. Архітектура нейронної мережі для розв'язання задачі визначення положення об'єкту

Спочатку внесемо дані вимірюваних напруг до електронних таблиць. Оскільки дані представлені у текстовому файлі, скористуємось функцією імпорту даних до таблиці (рис. 5).

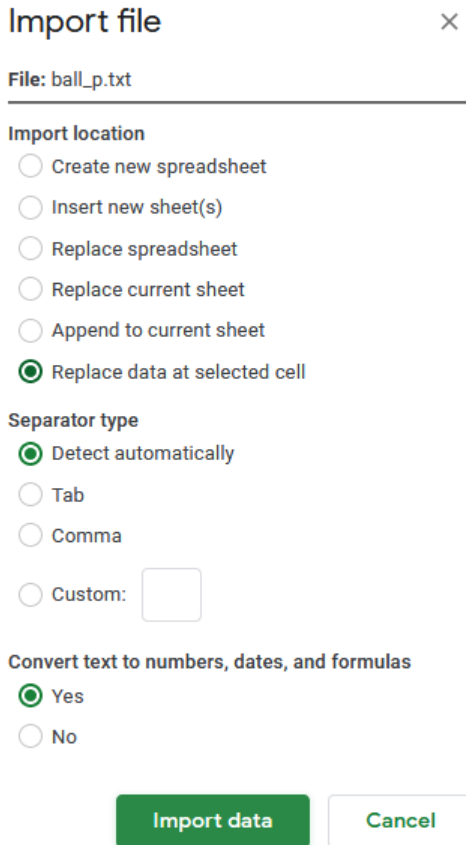


Рис. 5. Імпорт даних до таблиці

У результаті імпорту в комірки таблиці занесені наступні значення:

B1:BP1 вихідні дані напруги  $v_1$

B2:BP2 вихідні дані напруги  $v_2$

B3:BP3 координати у об'єкту

Для зручності опрацювання транспонуємо отримані дані та розмітимо їх у комірках A10:A76 для  $v_1$ , C10:C76 для  $v_2$ , C10:C76 для  $y$ . Для цього внесемо у комірки таблиці наступні значення:

A8 *дані*

A9  $v_1$

B9  $v_2$

C9  $y$

A10 =TRANSPOSE(B1:BP3)

Виходячи з того, що будована нейронна мережа матиме полярну функцію активації, усі значення на вході мережі повинні бути нормовані (приведені до діапазону [0; 1]). На виході мережі необхідно виконати обернену операцію.

Нормування виконуватимемо для кожного стовпця окремо. Для цього знайдемо для них мінімальні та максимальні значення, увівши у комірки такі формули:

|    |                  |
|----|------------------|
| E5 | $v_1$            |
| F5 | $v_2$            |
| G5 | $y$              |
| D6 | <i>max</i>       |
| D7 | <i>min</i>       |
| E6 | $=\max(A10:A76)$ |
| E7 | $=\min(A10:A76)$ |

Далі діапазон E6:E7 копіюється у F6:G7.

Сутність нормування легко зрозуміти із виразу:

$$\text{нормоване значення} = \frac{\text{вхідне значення} - \text{мінімальне значення}}{\text{максимальне значення} - \text{мінімальне значення}}$$

За такого підходу мінімальне значення нормується до 0, а максимальне – до 1. Нормовані значення напруг необхідно подати на вхідний шар нейронної мережі:

|     |                           |
|-----|---------------------------|
| E8  | <i>вхідний шар</i>        |
| E9  | $x_1$                     |
| F9  | $x_2$                     |
| E10 | $=(A10-E\$7)/(E\$6-E\$7)$ |

Комірку E10 поширюємо у діапазоні E10:G76.

Відповідно до обраної архітектури нейронної мережі, додамо до 2 нейронів вхідного шару нейрон зміщення. Для цього у комірку G9 внесемо його назву ( $x_3$ ), а у діапазон G10:G76 – його значення (1). На цьому кроці вхідний шар сформовано у вигляді сигнального вектору ( $x_1, x_2, x_3$ ).

Наступний крок – передавання сигналу із вхідного шару нейронної мережі на перший прихований. Для визначення сили сигналу необхідно мати вагові коефіцієнти нейронної мережі. Позначимо через:

–  $w_{ij}^{xh(1)}$  ваговий коефіцієнт синапсу, що пов'язує нейрон  $x_i$  ( $i = 1, 2, 3$ ) вхідного шару із нейроном  $h_j^{(1)}$  ( $j = 1, 2, \dots, 5$ ) першого прихованого шару;

–  $w_{kp}^{h(1)h(2)}$  ваговий коефіцієнт синапсу, що пов'язує нейрон  $h_k^{(1)}$  ( $k = 1, 2, \dots, 6$ ) першого прихованого шару із нейроном  $h_p^{(2)}$  ( $p = 1, 2, 3$ ) другого прихованого шару;

–  $w_{dq}^{h(2)y}$  ваговий коефіцієнт синапсу, що пов'язує нейрон  $h_d^{(2)}$  ( $d = 1, 2, \dots, 4$ ) другого прихованого шару із нейроном  $y_{\text{норм}q}$  ( $q = 1$ )



вихідного шару.

Тоді сила сигналу, що приходить до нейрону  $h_j^{(1)}$  першого прихованого шару, визначатиметься як скалярний добуток значень сигналу на вхідному шарі та відповідних вагових коефіцієнтів. Для визначення сигналу, який піде далі на другий прихований шар, застосуємо логістичну функцію активації  $f(S) = 1/(1+e^{-S})$ , де  $S$  – відповідний скалярний добуток. Формули для визначення сигналів на першому та другому прихованому і вихідному шарах матимуть вигляд:

$$h_j^{(1)} = f\left(\sum_{i=1}^{2+1} x_i w_{ij}^{xh(1)}\right), h_p^{(2)} = f\left(\sum_{k=1}^{5+1} h_k^{(1)} w_{kp}^{h(1)h(2)}\right),$$

$$y_{\text{норм}q} = f\left(\sum_{d=1}^{3+1} x_d w_{dq}^{h(2)y}\right).$$

Відповідно необхідно створити три матриці:

– матриця  $w^{xh(1)}$  розмірністю  $3 \times 5$  містить вагові коефіцієнти зв'язків 3 нейронів вхідного шару (перші два містять нормовані значення напруг, а третій є нейроном зміщення) з нейронами першого прихованого шару;

– матриця  $w^{h(1)h(2)}$  розмірністю  $6 \times 3$  містить вагові коефіцієнти зв'язків 6 нейронів першого прихованого шару (з них п'ять обчислюються, а шостий є нейроном зміщення) з нейронами другого прихованого шару;

– матриця  $w^{h(2)y}$  розмірністю  $4 \times 1$  містить вагові коефіцієнти зв'язків 4 нейронів другого прихованого шару (з них три обчислюються, а четвертий є нейроном зміщення) з нейронами вихідного шару.

Для «ненавченої» нейронної мережі початкові значення вагових коефіцієнтів можна встановити або у випадковий спосіб, або залишити невизначеними, або рівними нулеві. Для реалізації останнього способу заповнимо комірки такими значеннями:

|     |                              |
|-----|------------------------------|
| I8  | $w^{xh(1)}$                  |
| I9  | <i>вхідний шар</i>           |
| J8  | <i>перший прихований шар</i> |
| J9  | 1                            |
| K9  | =J9+1                        |
| I10 | 1                            |
| I11 | =I10+1                       |
| J10 | 0                            |
| I14 | $w^{h(1)h(2)}$               |
| I15 | <i>перший прихований шар</i> |
| J14 | <i>другий прихований шар</i> |

- J15 1
- K15 =J15+1
- I16 1
- I17 =I16+1
- J16 0
- I23  $w^{h(2)y}$
- I24 *другий прихований шар*
- J23 *вихідний шар*
- J10 1
- I25 1
- I26 =I25+1
- J25 0

Для створення матриць необхідно виконати копіювання комірки K9 у діапазон L9:N9, I11 – у I12, J10 – у J10:N12, K15 – у L15, I17 – у I18:I21, J16 – у J16:L21, I26 – у I27:I28, J25 – у J26:J28 (рис. 6).

$f_x = (A10-E\$7)/(E\$6-E\$7)$

|    | E                   | F         | G         | H | I                            | J                            | K      | L      | M      | N      |
|----|---------------------|-----------|-----------|---|------------------------------|------------------------------|--------|--------|--------|--------|
| 8  | <i>вихідний шар</i> |           |           |   | <b>wxh(1)</b>                | <i>перший прихований шар</i> |        |        |        |        |
| 9  | <b>x1</b>           | <b>x2</b> | <b>x3</b> |   | <i>вихідний шар</i>          | 1                            | 2      | 3      | 4      | 5      |
| 10 | 0.3986              | 0.9982    | 1         |   |                              | 1                            | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 11 | 0.2963              | 0.9959    | 1         |   |                              | 2                            | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 12 | 0.2155              | 0.9957    | 1         |   |                              | 3                            | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 13 | 0.1065              | 0.9938    | 1         |   |                              |                              |        |        |        |        |
| 14 | 0.0686              | 0.9904    | 1         |   | <b>wh(1)h(2)</b>             | <i>другий прихований шар</i> |        |        |        |        |
| 15 | 0.0079              | 0.9811    | 1         |   | <i>перший прихований шар</i> | 1                            | 2      | 3      |        |        |
| 16 | 0.0004              | 0.9578    | 1         |   |                              | 1                            | 0.0000 | 0.0000 | 0.0000 |        |
| 17 | 0.0001              | 0.9430    | 1         |   |                              | 2                            | 0.0000 | 0.0000 | 0.0000 |        |
| 18 | 0.0000              | 0.9231    | 1         |   |                              | 3                            | 0.0000 | 0.0000 | 0.0000 |        |
| 19 | 0.0013              | 0.9092    | 1         |   |                              | 4                            | 0.0000 | 0.0000 | 0.0000 |        |
| 20 | 0.0018              | 0.8871    | 1         |   |                              | 5                            | 0.0000 | 0.0000 | 0.0000 |        |
| 21 | 0.0036              | 0.8679    | 1         |   |                              | 6                            | 0.0000 | 0.0000 | 0.0000 |        |
| 22 | 0.0036              | 0.8442    | 1         |   |                              |                              |        |        |        |        |
| 23 | 0.0040              | 0.8241    | 1         |   | <b>wh(2)y</b>                | <i>вихідний шар</i>          |        |        |        |        |
| 24 | 0.0082              | 0.7903    | 1         |   | <i>другий прихований шар</i> | 1                            |        |        |        |        |
| 25 | 0.0826              | 0.7664    | 1         |   |                              | 1                            | 0.0000 |        |        |        |
| 26 | 0.2688              | 0.7271    | 1         |   |                              | 2                            | 0.0000 |        |        |        |
| 27 | 0.4236              | 0.6788    | 1         |   |                              | 3                            | 0.0000 |        |        |        |
| 28 | 0.4703              | 0.6537    | 1         |   |                              | 4                            | 0.0000 |        |        |        |

Рис. 6. Фрагмент електронної таблиці після нормування вхідних даних та створення матриць вагових коефіцієнтів

Для обчислення скалярного добутку вектор-рядка значень вхідного шару на вектор-стовпець матриці вагових коефіцієнтів  $w^{xh(1)}$  доцільно скористатись функцією множення матриць:

|     |   |
|-----|---|
| P8  | <i>перший прихований шар</i>                          |
| P9  | $h_1^{(1)}$   |
| Q9  | $h_2^{(1)}$   |
| R9  | $h_3^{(1)}$   |
| S9  | $h_4^{(1)}$   |
| T9  | $h_5^{(1)}$   |
| U9  | $h_6^{(1)}$   |
| P10 | $=1/(1+\exp(-\text{mmult}(\$E10:\$G10,\$J10:\$J12)))$ |
| U10 | 1   |

Далі копіюємо комірку P10 у діапазон P10:T76, а U10 – у U11:U76.

Ураховуючи, що всі елементи матриці вагових коефіцієнтів  $w^{xh(1)}$  спочатку дорівнюють нулю, після копіювання формул всі обчислювані елементи прихованого шару дорівнюватимуть 0,5.

Аналогічно виконаємо обчислення елементів другого прихованого та вихідного шарів:

|      |   |
|------|---|
| W8   | <i>другий прихований шар</i>                          |
| W9   | $h_1^{(2)}$   |
| X9   | $h_2^{(2)}$   |
| Y9   | $h_3^{(2)}$   |
| Z9   | $h_4^{(2)}$   |
| W10  | $=1/(1+\exp(-\text{mmult}(\$P10:\$U10,\$J16:\$J21)))$ |
| Z10  | 1   |
| AB8  | <i>вихідний шар</i>                                   |
| AB9  | $u_{\text{норм}}$                                     |
| AB10 | $=1/(1+\exp(-\text{mmult}(\$W10:\$Z10,\$J25:\$J28)))$ |

Далі копіюємо комірку W10 у діапазон W10:Y76, Z10 – у Z11:Z76, AB10 – у AB11:AB76 (рис. 7).

Для отримання результату  $u_{\text{обч}}$  із нормованого значення вихідного шару необхідно обчислити його за формулою, оберненою до початкової:

вихідне значення = мінімальне значення +  
нормоване значення \* (максимальне значення – мінімальне значення)

| fx | =1/(1+exp(-mmult(\$W10:\$Z10,\$J25:\$J28))) |       |       |       |       |       |   |                              |       |       |       |    |                     |
|----|---|-------|-------|-------|-------|-------|---|------------------------------|-------|-------|-------|----|---------------------|
|    | P   | Q     | R     | S     | T     | U     | V | W                            | X     | Y     | Z     | AA | AB                  |
| 8  | <i>перший прихований шар</i>                |       |       |       |       |       |   | <i>другий прихований шар</i> |       |       |       |    | <i>вихідний шар</i> |
| 9  | h(1)1                                       | h(1)2 | h(1)3 | h(1)4 | h(1)5 | h(1)6 |   | h(2)1                        | h(2)2 | h(2)3 | h(2)4 |    | унорм               |
| 10 | 0.500                                       | 0.500 | 0.500 | 0.500 | 0.500 | 1     |   | 0.500                        | 0.500 | 0.500 | 1     |    | 0.500               |
| 11 | 0.500                                       | 0.500 | 0.500 | 0.500 | 0.500 | 1     |   | 0.500                        | 0.500 | 0.500 | 1     |    | 0.500               |
| 12 | 0.500                                       | 0.500 | 0.500 | 0.500 | 0.500 | 1     |   | 0.500                        | 0.500 | 0.500 | 1     |    | 0.500               |

Рис. 7. Фрагмент електронної таблиці після обчислення початкових значень вагових коефіцієнтів прихованих та вихідного шарів

Для цього вводимо такі значення у комірки таблиці:

AD8            результат  
 AD9             $y_{обч}$   
 AD10           $=\$G\$7+AB10*(\$G\$6-\$G\$7)$

Далі копіюємо комірку AD10 у діапазон AD11:AD76.

Навчання нейронної мережі відбувається шляхом варіювання вагових коефіцієнтів у такий спосіб, щоб із кожним кроком навчання різниця між обчисленими значеннями  $y_{обч}$  та бажаними (еталонними) значеннями  $y$  зменшувалась. Для визначення різниці між обчисленим та еталонним вихідними векторами обчислимо квадрати відхилень та їх суму:

AF8            *квадрат відхилення*  
 AH8            *сума*  
 AF9             $(y-y_{обч})^2$   
 AH9             $S$   
 AF10           $=(C10-AD10)^2$   
 AH10           $=sum(AF10:AF76)$

Далі копіюємо комірку AF10 у діапазон AF11:AF76. Комірка AH10 міститиме суму квадратів відхилень.

За такої постановки навчання нейронної мережі може розглядатися як задача оптимізації, у якій цільова функція – сума квадратів відхилень у комірці AH10 – підлягатиме мінімізації шляхом варіювання вагових коефіцієнтів матриць  $w^{xh(1)}$  (діапазон J10:N12),  $w^{h(1)h(2)}$  (діапазон J16:L21) та  $w^{h(2)y}$  (діапазон J25:J28). Для розв’язання цієї задачі стандартних засобів Google Sheets недостатньо, тому необхідним є встановлення доповнення Solver вибором пункту меню Add-ons → Get add-ons... (рис. 8).

Доповнення

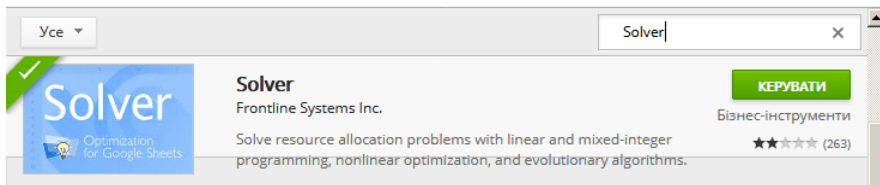


Рис. 8. Встановлення доповнення Solver у електронних таблицях Google Sheets

На рис. 9 показані налаштування доповнення Solver для розв’язання поставленої задачі: цільова функція (Set Objective) мінімізується (To: Min) шляхом зміни значень (By Changing) матриць вагових коефіцієнтів у діапазоні (Subject To) від -20 до +20 одним із методів оптимізації (Solving Method).

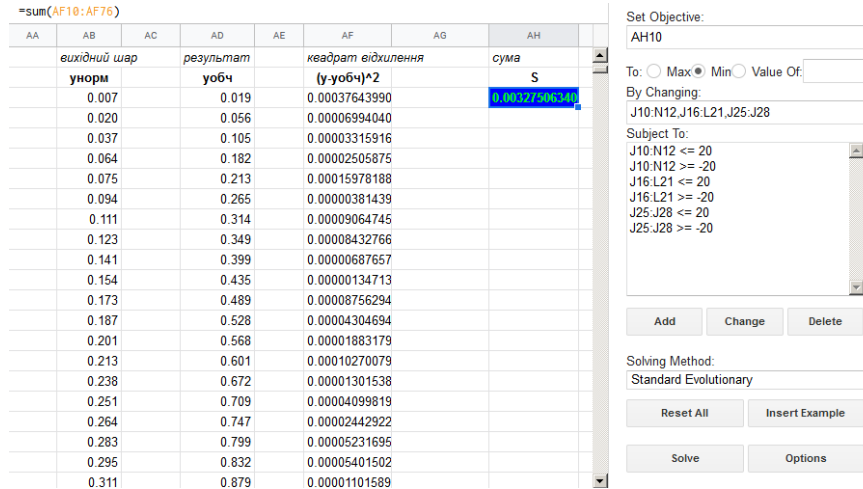


Рис. 9. Результат оптимізації за обраних параметрів Solver

Для зменшення суми квадратів відхилень Solver може викликатися багаторазово: доцільно поекспериментувати із комбінованим використанням різних методів оптимізації, змінюючи межі варіювання вагових коефіцієнтів. При цьому необов'язково намагались довести значення суми відстаней до нуля – це може бути й більше (достатньо невелике) значення (рис. 9).

у / убоч

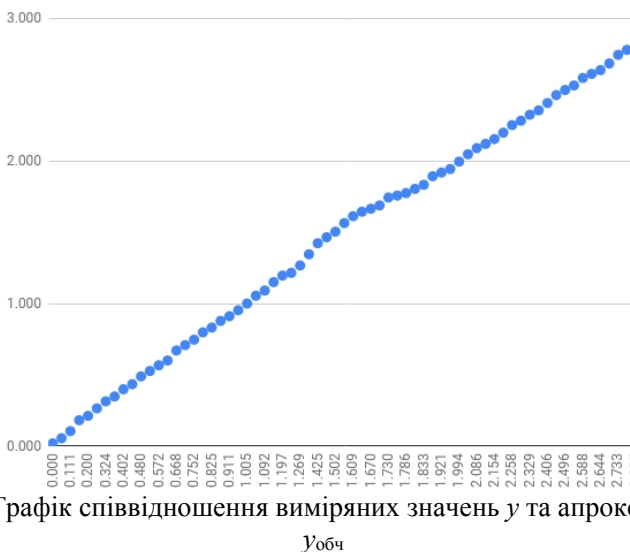


Рис. 10. Графік співвідношення вимірних значень у та апроксимованих

убоч

Для унаочнення результатів моделювання доцільно побудувати графік співвідношення експериментально отриманих (вимірних) значень  $y$  та обчислених (апроксимованих)  $y_{обч}$  (рис. 10). Якість апроксимації визначається ступенем відхилення точок графіку від променя, що ділить перший квадрант навпіл. За рисунком та за отриманою після оптимізації сумою квадратів відхилень можна зробити висновок про достатньо високу адекватність побудованої нейромережевої моделі.

### Список використаних джерел

1. Модло Є. О. Проектування системи компетенцій бакалавра електромеханіки в моделюванні / Модло Є. О. // Інформаційні технології в освіті та науці : збірник наукових праць. – Випуск 7. – Мелітополь : Видавництво МДПУ ім. Б. Хмельницького, 2015. – С. 111-116.

2. Модло Є. О. Компетентність бакалавра електромеханіки в моделюванні / Є. О. Модло // Вісник Дніпропетровського університету імені Альфреда Нобеля. Серія: Педагогіка і психологія. – 2015. – № 1 (9). – С. 17-24, 294.

3. Модло Є. О. Комп'ютерне моделювання в підготовці бакалаврів електромеханіки / Є. О. Модло // Комп'ютерне моделювання в освіті : матеріали VI Всеукраїнського науково-методичного семінару (Кривий Ріг, 12 квітня 2013 р.). – Кривий Ріг : Видавничий відділ КМІ, 2013. – С. 25-26.

4. Пермякова О. С. Застосування нейронних мереж у задачах прогнозування / О. С. Пермякова, С. О. Семеріков // Молодий науковець XXI століття : матеріали Міжнародної науково-практичної конференції (Кривий Ріг, 17–18 листопада 2008 р.). – Кривий Ріг : Видавничий центр КТУ, 2008. – С. 237-239.

5. Омату С. Нейроуправление и его приложения / Сигеру Омату, Марзуки Халид, Рубия Юсоф. – М. : Радиотехника, 2000. – 272 с. – (Нейрокомпьютеры и их применение. Книга 2)

6. Hagan M. T. Neural Network Design (2nd Edition) [Electronic resource] / Martin T. Hagan, Howard B. Demuth, Mark H. Beale, Orlando De Jesús. – [2014?]. – 1012 p. – Access mode : <http://hagan.okstate.edu/NNDesign.pdf>.

7. Соловйов В. М. Інструментальне забезпечення курсу комп'ютерного моделювання / В. М. Соловйов, С. О. Семеріков, І. О. Теплицький // Комп'ютер у школі і сім'ї. – 2000. – № 4. – С. 28-31.

8. Теплицький І. О. Використання електронних таблиць у комп'ютерному моделюванні / І. О. Теплицький // Комп'ютер у школі та сім'ї. – 1999. – № 2. – С. 27-32.

**References (translated and transliterated)**

1. Modlo E. O. Proektuvannia systemy kompetentsii bakalavra elektromekhaniky v modeliuvanni [Competence bachelor of electromechanics in simulation] / Modlo E. O. // Informatsiini tekhnologii v osviti ta nautsi : zbirnyk naukovykh prats. – Vypusk 7. – Melitopol : Vydavnytstvo MDPU im. B. Khmelnytskoho, 2015. – S. 111-116. (In Ukrainian)
2. Modlo E. O. Kompetentnist bakalavra elektromekhaniky v modeliuvanni [Competence of bachelor in electromechanics in simulation] / E. O. Modlo // Visnyk Dnipropetrovskoho universytetu imeni Alfreda Nobelia. Serii: Pedahohika i psykholohiia. – 2015. – # 1 (9). – S. 17-24, 294. (In Ukrainian)
3. Modlo Ye. O. Kompiuterne modeliuvannia v pidhotovtsi bakalavriv elektromekhaniky [Computer simulation in the training of bachelors in electromechanics] / Ye. O. Modlo // Kompiuterne modeliuvannia v osviti : materialy VI Vseukrainskoho naukovo-metodychnoho seminaru (Kryvyi Rih, 12 kvitnia 2013 r.). – Kryvyi Rih : Vydavnychi viddil KMI, 2013. – S. 25-26. (In Ukrainian)
4. Permiakova O. S. Zastosuvannia neuronnykh merezh u zadachakh prohnozuvannia [Application of neural networks in forecasting tasks] / O. S. Permiakova, S. O. Semerikov // Molodyi naukovec KhKhI stolittia : materialy Mizhnarodnoi naukovo-praktychnoi konferentsii (Kryvyi Rih, 17–18 lystopada 2008 r.). – Kryvyi Rih : Vydavnychi tsestr KTU, 2008. – S. 237-239. (In Ukrainian)
5. Omatu S. Neiroupravlenie i ego prilozeniia [Neuro-Control and its Applications] / Sigeru Omatu, Marzuki Khalid, Rubiia Iusof. – M. : Radiotekhnika, 2000. – 272 s. – (Neirokompiutery i ikh primenenie. Kniga 2) (In Russian)
6. Hagan M. T. Neural Network Design (2nd Edition) [Electronic resource] / Martin T. Hagan, Howard B. Demuth, Mark H. Beale, Orlando De Jesús. – [2014?]. – 1012 p. – Access mode : <http://hagan.okstate.edu/NNDesign.pdf>.
7. Soloviov V. M. Instrumentalne zabezpechennia kursu kompiuternoho modeliuvannia [Toolware of computer simulation course] / V. M. Soloviov, S. O. Semerikov, I. O. Teplytskyi // Kompiuter u shkoli i simi. – 2000. – No. 4. – S. 28-31. (In Ukrainian)
8. Teplytskyi I. O. Vykorystannia elektronnykh tablyts u kompiuternomu modeliuvanni [Using spreadsheets in computer simulation] / I. O. Teplytskyi // Kompiuter u shkoli ta simi. – 1999. – No. 2. – S. 27-32. (In Ukrainian)