УДК 004.8
DOI https://doi.org/10.32838/2663-5941/2019.4-1/13

***Ковальчук А.М.***
Державний університет «Житомирська політехніка»

***Марчук Г.В.***
Державний університет «Житомирська політехніка»

***Марчук Д.К.***
Державний університет «Житомирська політехніка»

# ЗАСТОСУВАННЯ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ РУКОПИСНИХ СИМВОЛІВ

*У роботі досліджено можливості згорткових нейронних мереж для розпізнавання рукописних символів. Архітектура ЗНМ призначена і використовується для ефективного розпізнавання зображень, де чергуються шари згортки з шарами субдискретизації.*

*Для навчання нейронної мережі використовується метод зворотного поширення помилки. Це ітеративний алгоритм, який використовується з метою мінімізації помилки роботи багатошарового перцептрона та отримання бажаного результату.*

*Навчання мережі здійснюється з вчителем. Вхідними даними нейронної мережі є зображення розміром 250x250 пікселів. З цього випливає, що на вхідному шарі мережі розташовуються 62 500 нейронів. Навчання проводиться на прикладах з вже відомою відповіддю. Необхідно зазначати, що під час навчанні з вчителем потрібна велика вибірка, щоб в достатній мірі сформувати робочу нейронну мережу.*

*У роботі представлена модель згорткової нейронної мережі у вигляді діаграми класів.*

*Для реалізації згорткової нейронної мережі створено клас для реалізації системи згортання, клас для реалізації системи підвибірки, клас для реалізації системи навчання.*

*Для розробки програмного додатка обрано рушій Unity і мову програмування C#. Завдяки великому інструментарію з роботи з 2D-об'єктами, Unity відмінно підходить і для роботи з не ігровим контентом. Крім того рушій ідеально підходить під створення кросплатформенних додатків, збірка проектів здійснюється буквально за пару кликів. А це розв'язує проблему багатьох замовників, які хочуть бачити свій додаток і на iOS, і на Android.*

*Розроблено програмний додаток, що розпізнає написаний користувачем символ, збирає у текст та зберігає його у файл. Крім того програмний додаток надає можливість створити власну нейронну мережу під особисті особливості накреслення букв та символів.*

*Для налаштування мережі необхідно ввести кількість шарів нейронів і кількість нейронів в кожному шарі та визначити кількість навчальних прикладів для кожного символу.*

*Навчену нейронну мережу можна зберегти у файл з розширенням *.cnn.*

*Програмний додаток також надає можливості використовувати вже наявну навчену мережу. Для цього потрібно тільки вибрати відповідний файл.*

*Програмний продукт забезпечує розпізнавання букв, цифр, арифметичних і пунктуаційних символів. Серед інших особливостей значиться робота з будь-якою мовою.*

***Ключові слова:*** *нейрон, мережа, символ, текст, згортка, субдіскретізація, навчання, похибка.*

**Постановка проблеми.** Згорткова нейронна мережа (англ. convolutional neural network, CNN) – спеціальна архітектура штучних нейронних мереж, запропонована Яном Лекуном у 1988 році та націлена на ефективне розпізнавання образів, входить до складу технологій глибокого навчання (англ. deep learning) і успішно зарекомендувала себе в завданнях розпізнавання та класифікації зображень.

Назву архітектура мережі отримала через наявність операції згортки, суть якої в тому, що кожен фрагмент зображення множиться на матрицю (ядро) згортки елемент за елементом, а результат підсумовується і записується в аналогічну позицію вихідного зображення.

Актуальність розробок в області нейронних мереж обумовлюється їхніми широкими можливостями застосування в самих різних областях.

**Постановка завдання.** Метою роботи є розробка програмного додатка, що розпізнає написаний користувачем символ, збирає у текст та зберігає його у файл у форматі txt. Крім того програмний додаток надає можливість створити

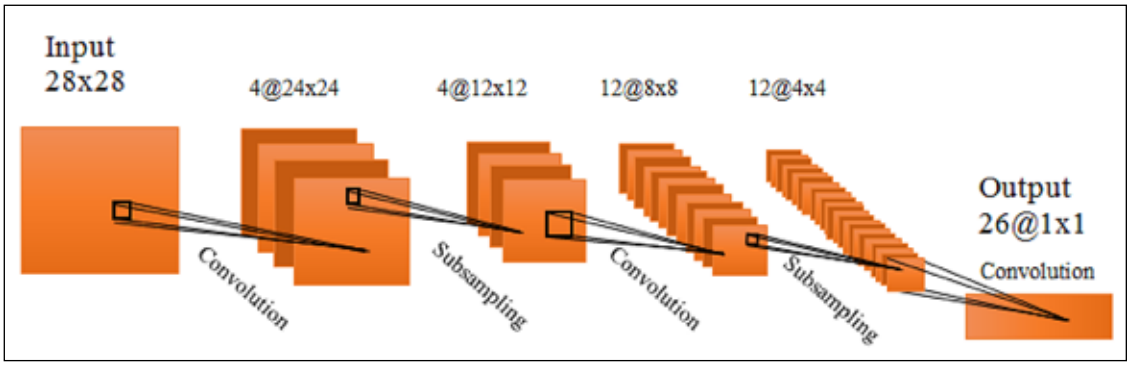


**Рис. 1. Архітектура згорткової нейронної мережі**

власну нейронну мережу або використовувати вже наявну навчену мережу.

**Виклад основного матеріалу.**

***1. Згорткові нейронні мережі***

Згорткова нейронна мережа – спеціальна архітектура нейронних мереж, запропонована Яном Лекуном, від самого початку націлена на ефективне розпізнавання зображень.

Згорткова нейронна мережа зазвичай являє собою чергування шарів згортки (convolution layers, C-layers), шарів субдіскретізації (subsampling layers, S-layers) і за наявності повнозв'язних шарів (fully-connected layer, F-layers) на виході [1, с. 7; 2, с. 4]. Всі три види шарів можуть чергуватися в довільному порядку (рис. 1).

Згортка (convolution) – операція над парою матриць A (розміру $n_x \times n_y$) і B (розміру $m_x \times m_y$), результатом якої є матриця $C = A*B$ розміру $(n_x-m_x+1)\times(n_y-m_y+1)$. Кожен елемент обчислюється як скалярний добуток матриці B і деякої підматриці A такого ж розміру.

Операція згортки (рис.2) визначається таким виразом:

$$C_{i,j} = \sum_{u=0}^{m_x-1}\sum_{v=0}^{m_y-1} A_{i+u,j+v} B_{u,v}$$

де $B_{u,v}$ – значення елемента ядра згортки (u, v), $C_{i,j}$ – значення пікселя зображення, що отримаємо, $A_{i+u,j+v}$ – значення пікселя вхідного зображення, $m_x - 1, m_y - 1$ – розмір ядра згортки.


**Рис. 2. Алгоритм згортки**

Логічний сенс згортки такий – чим більше величина елемента згортки, тим більше ця частина матриці A була схожа на матрицю B (схожа в сенсі скалярного добутку). Тому матрицю A називають зображенням, а матрицю B – фільтром.

За реалізацію згортки відповідає клас Convolution. В ньому зберігається ядро згортки нейрону. Отримуючи на вхід зображення у вигляді матриці раціональних чисел, клас множить кожну частину вхідного зображення з ядром. Результатом цих операцій є нове, менше зображення.

Лістинг класу Convolution:

```
class Convolution{
public double[,] Kernel;
public double[,] LastOutput;
public double[,] GetNewImage(double[,] enterImage) {
 var n1 = enterImage.GetLength(0);
 var n2 = Kernel.GetLength(0);
 var m1 = enterImage.GetLength(1);
 var m2 = Kernel.GetLength(1);
 var n = n1 + 1 - n2;
 var m = m1 + 1 - m2;
 var newImage = new double[n, m];
 for (int i = 0; i < n; i++)
 {
 for (int j = 0; j < m; j++)
 {
 newImage[i, j] = GetNewPixel(enterImage, i, j);
 }
 }
 LastOutput = newImage;
 return LastOutput;
 }
private double GetNewPixel(double[,] enterImage, int posX, int posY) {
 double sum = 0;
 for (int i = 0; i < Kernel.GetLength(0); i++) {
 for (int j = 0; j < Kernel.GetLength(1); j++){
```

```
sum += enterImage[i + posX, j + posY] * Kernel[i, j];
}
}
return sum;
}
}
```

Шари субдискретизації виконують зменшення розмірності (зазвичай в кілька разів). Це можна робити різними способами, але найчастіше використовується метод вибору максимального елемента (max-pooling) вся карта ознак поділяється на осередки, з яких вибираються максимальні за значенням (рис. 3).

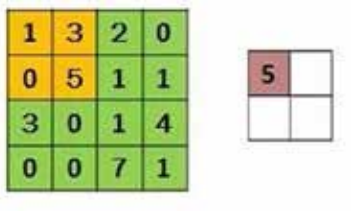


**Рис. 3. Алгоритм субдіскретизації (max poling)**

За реалізацію системи підвибірки відповідає статичний клас Subsampling. Даний клас зменшує вхідне зображення у 2 рази за вертикаллю та горизонталлю. Отримуючи на вхід зображення у вигляді матриці раціональних чисел, клас розділяє вхідне зображення на квадрати розміром 2*2. В кожному з цих квадратів метод знаходить найбільший за значенням елемент. Цей елемент стає значенням відповідного пікселя нового зображення, розмірами якого являється кількість квадратів 2*2 що поміщаються на вхідному зображенні за вертикаллю та горизонталлю відповідно.

Лістинг класу для реалізації системи підвибірки:

```
static class Subsampling{
public static double[,] GetNewImage(double[,] enterImage) {
var n = enterImage.GetLength(0) / 2;
var m = enterImage.GetLength(1) / 2;
var newImage = new double[n, m];
for (int i = 0; i < n; i++) {
for (int j = 0; j < m; j++) {
newImage[i, j] = GetNewPixel(enterImage, i, j);
}
}
return newImage;
}
private static double GetNewPixel(double[,] enterImage, int posX, int posY){
double max = double.NegativeInfinity;
for (int i = 0; i < 2; i++) {
for (int j = 0; j < 2; j++) {
max = Math.Max(enterImage[i + posX * 2, j + posY * 2], max);
}
}
return max;
}
}
```

***2. Навчання***

Найпростішим і популярним способом навчання є метод навчання з учителем – метод зворотного поширення помилки (backpropagation).

Зворотне поширення помилки являє собою ітераційний процес, який починається з останнього шару і рухається у зворотному напрямі через всі


**Рис. 4. Алгоритм зворотного поширення похибки**

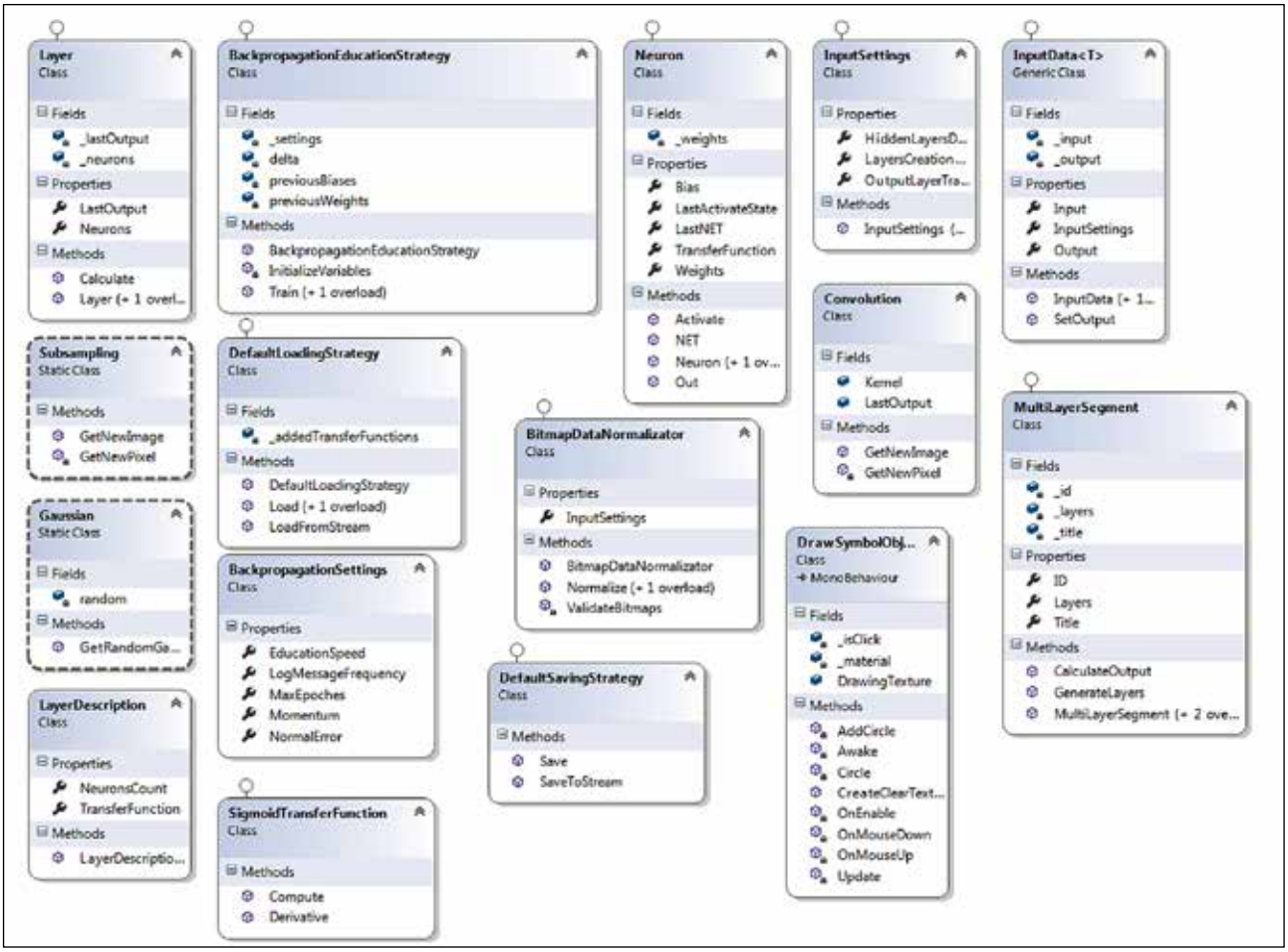


**Рис. 5. Діаграма класів, що реалізує модель ЗНМ**

шари, поки не буде досягнуто перший шар. Припустимо, що для кожного шару відома помилка на виході з шару. Якщо відома помилка на виході, то не важко розрахувати зміни для ваги, тим самим зменшивши помилку. Проблема полягає в тому, що помилку видно тільки на виході самого останнього шару. Тобто алгоритм зворотного поширення помилки дає можливість визначити помилку на виході попереднього шару, з огляду на помилку на виході в поточному шарі (рис. 4).

За реалізацію системи навчання відповідає клас EducationStrategy. Навчання проводиться на прикладах з вже відомою відповіддю.

***3. Реалізація***

В роботі була реалізована модель згорткової нейронної мережі.

Архітектура створеної нейронної мережі реалізує три ідеї:

– кожен нейрон отримує вхідний сигнал від локального рецептивного поля в попередньому шарі, що забезпечує локальну двовимірну зв'язність нейронів;

– кожен прихований шар мережі складається з безлічі карт ознак, на яких всі нейрони мають загальні ваги, що забезпечує інваріантність до зміщення і скорочення загального числа вагових коефіцієнтів мережі;

– за кожним шаром згортки іде обчислювальний шар, який здійснює локальне усереднення і підвибірку, що забезпечує зменшення дозволу для карт ознак.

Слід зазначити, що навчання мережі здійснюється з вчителем. Вхідними даними нейронної мережі є зображення розміром 250x250 пікселів. З цього випливає, що на вхідному шарі мережі розташовується 62 500 нейронів.

На рисунку 5 представлена Діаграма класів, що реалізує модель згорткової нейронної мережі.

Після запуску програмного додатка, у головному вікні з'являється дві кнопки «Load network», «Create network» (рис. 6).

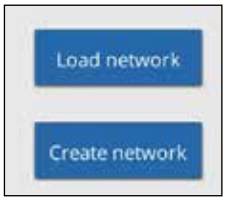


**Рис. 6. Головне вікно програмного додатку**

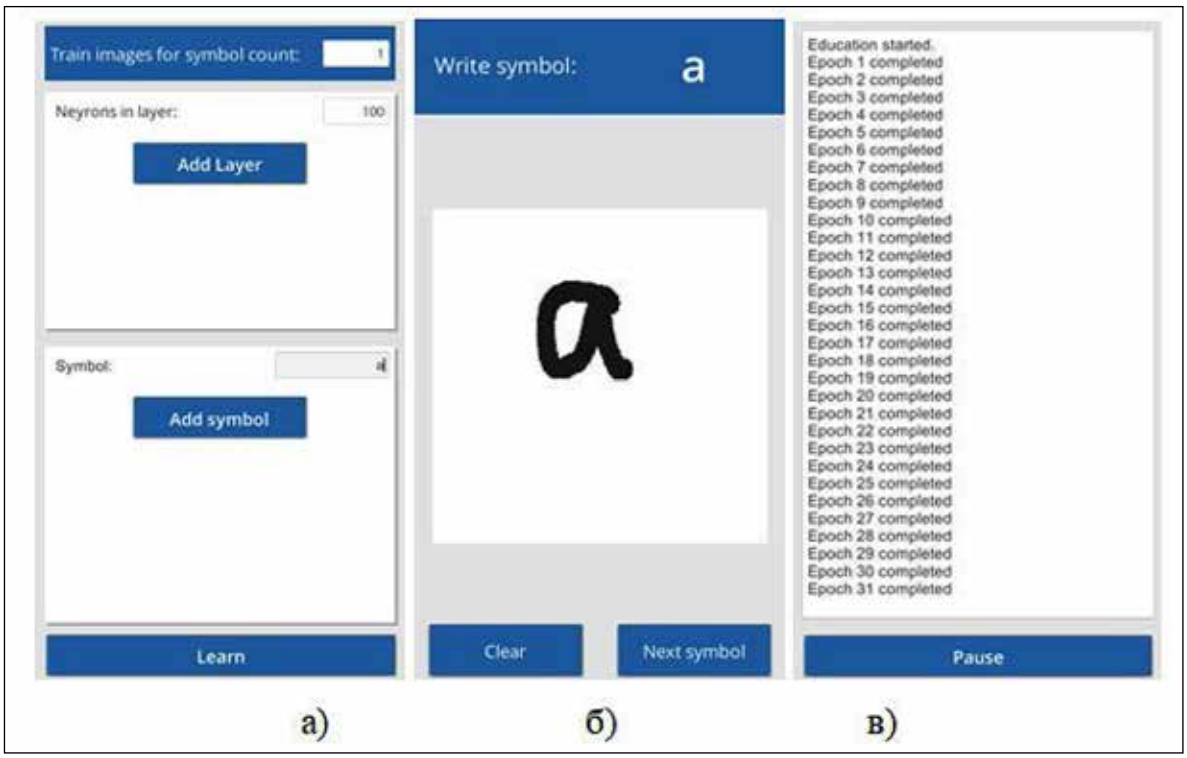


**Рис. 7. Створення власної мережі**


**Рис. 8. Введення і розпізнавання символів**

Для роботи з навченою нейронною мережею необхідно увімкнути файл з розширенням *.cnn.

Якщо користувач із головного меню перейшов по кнопці «Create network», він попадає в меню налаштування нейронної мережі (рис. 7(а)).

Для створення власної мережі потрібно ввести параметри для її налаштування:

– кількість шарів нейронів;

– кількість нейронів в кожному шарі;

– символи, що повинна розпізнати нейронна мережа.

– кількість навчальних прикладів для кожного символу.

На рисунку 7(б) показано приклад введення символу для навчання, на рисунку 7(в) показано навчання мережі.

Якщо нейронна мережа пройшла курс навчання, можна приступити до введення символів.

В центрі вікна введення символів (рис.8) розташовується поле для написання символів. Після того як користувач ввів символ і натиснув кнопку «ОК» в верхній частині екрану з'являється символ, що пройшов через нейронну мережу, і є розпізнаним. Якщо користувач зробив помилку, то він може видалити символ з допомогою кнопки «Backspace».

Користувач може ввести слово буква за буквою, після чого поставити пробіл кнопка «Space». Коли користувач ввів текст, він може скопіювати його у буфер обміну (кнопка «Copy») або зберегти у файл(кнопка «Save to file»).

**Висновки.** Отже, розроблений алгоритм, який використовує згорткову нейронну мережу, може бути застосований для розв'язання завдань

онлайн-розпізнавання рукописного тексту. Точність розпізнавання достатньо висока. Ефективність реалізованої системи залежить від розміру навчальної та тестової вибірок, кількості шарів та нейронів у кожному шарі. Точність вирішення поставленого завдання з використанням моделі згорткової нейронної може бути поліпшена практично до 99%.

Програмний продукт забезпечує розпізнавання букв, цифр, арифметичних і пунктуаційних символів. Серед інших особливостей значиться робота з будь-якими мовами, два режими роботи розпізнавання символів (робота з уже навченою нейронною мережею або спочатку навчити мережу, а потім робота з нею) та система навчання нейронної мережі.

**Список літератури:**

1. LeCun Y., Y. Bengio Convolutional Networks for Images, Speech, and Time-Series, in Arbib, M. A. (Eds), The Handbook of Brain Theory and Neural Networks, 1995.

2. Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E. Imagenet classification with deep convolutional neural networks, NIPS. 2012.

**Marchuk G.V., Marchuk D.K., Kovalchuk A.M. APPLICATION OF A CONVOLUTIONAL NEURAL NETWORK FOR THE RECOGNITION OF HANDWRITTEN CHARACTERS**

*In this work the possibilities of convolutional neural networks for the recognition of handwritten characters are investigated. CTT architecture is designed and used to efficiently recognize images, where alternating layers of convolutions with sub-sampling layers.*

*To train the neural network, we use the method of reverse error propagation. This is an iterative algorithm that is used to minimize the mistake of working a multilayer perceptron and obtain the desired result.*

*Teaching the network is carried out with a teacher. The input data of the neural network is an image of 250x250 pixels. It follows from this that 62500 neurons are present on the input layer of the network. The training is conducted on examples with already known answer. It should be noted that when teaching with a teacher, a large test sample is required to adequately form the working neural network.*

*In the work, the model of the neural system is represented in the diagram of classes.*

*For realization a convolutional neural network, a class has been created for the implementation of a curtailment system, a class for the implementation of a sub-system, a class for the implementation of a training system.*

*Unity engine and C # programming language are used to develop software application. Thanks to the great toolkit for working with 2D-objects, Unity is also great for non-gaming content. In addition, the engine is ideally suited for creating cross-platform applications, a collection of projects carried out literally a couple of clicks. And this solves the problem of many customers who want to see their app on both iOS and Android.*

*A software application is developed that recognizes a character written by the user, creating a text and storing it in a file. In addition, the software application provides the ability to create its own neural network under the personal characteristics of the letters and characters.*

*To configure the network, you must enter the number of neuron layers and the number of neurons in each layer and determine the number of tutorials for each character.*

*A trained neural network can be saved to a file with extension * .cnn.*

*The software application also provides the ability to use an existing trained network. To do this, just select the appropriate file.*

*The software provides recognition of letters, numbers, arithmetic and punctuation symbols. Among other features is the work with your favorite languages.*

***Key words:*** *neuron, network, symbol, text, convolution, sub-sampling, training, mistake.*