ІНТЕЛЕКТУАЛЬНА СИСТЕМА ДЛЯ ЛЕКСИЧНОГО АНАЛІЗУ ТЕКСТУ З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ

дипломна робота

Виконав студент гр. КНзн-14сп Сошніков Д.Ю..
Науковий керівник: к.т.н., доц., Колесницький О.К.

Об’єкт дослідження – процес лексичного аналізу слів природної мови.

Предмет дослідження – інтелектуальні програмні засоби для лексичного аналізу слів природної мови та швидкодія їх функціонування.

Мета роботи – підвищення швидкодії лексичного аналізу слів природної мови інтелектуальною програмною системою за рахунок застосування нейронної мережі.

ПОСТАНОВКА ЗАДАЧІ

Завдання роботи - створення програми розпізнавання частин мови (іменник, прикметник, дієслово та ін.), до якої відносяться слова у заданому тексті з використанням нейромереж.

Для виконання завдання була обрана нейронна мережа, яку реалізовано на мові програмування C #.
Вхідні параметри: навчальна вибірка; тестова вибірка; кількість нейронів прихованого шару; коефіцієнт навчання; кількість епох навчання.

Для вирішення поставленого завдання було створено 4 класи:
- клас нейромережі та засобів її навчання;
- клас переведення тексту в двійковий вигляд;
- клас хеш-таблиці і методів роботи з нею;
- клас розбиття тексту на лексеми і розпізнавання.

ТЕХНІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ДЛЯ ЛЕКСИЧНОГО АНАЛІЗУ ТЕКСТУ

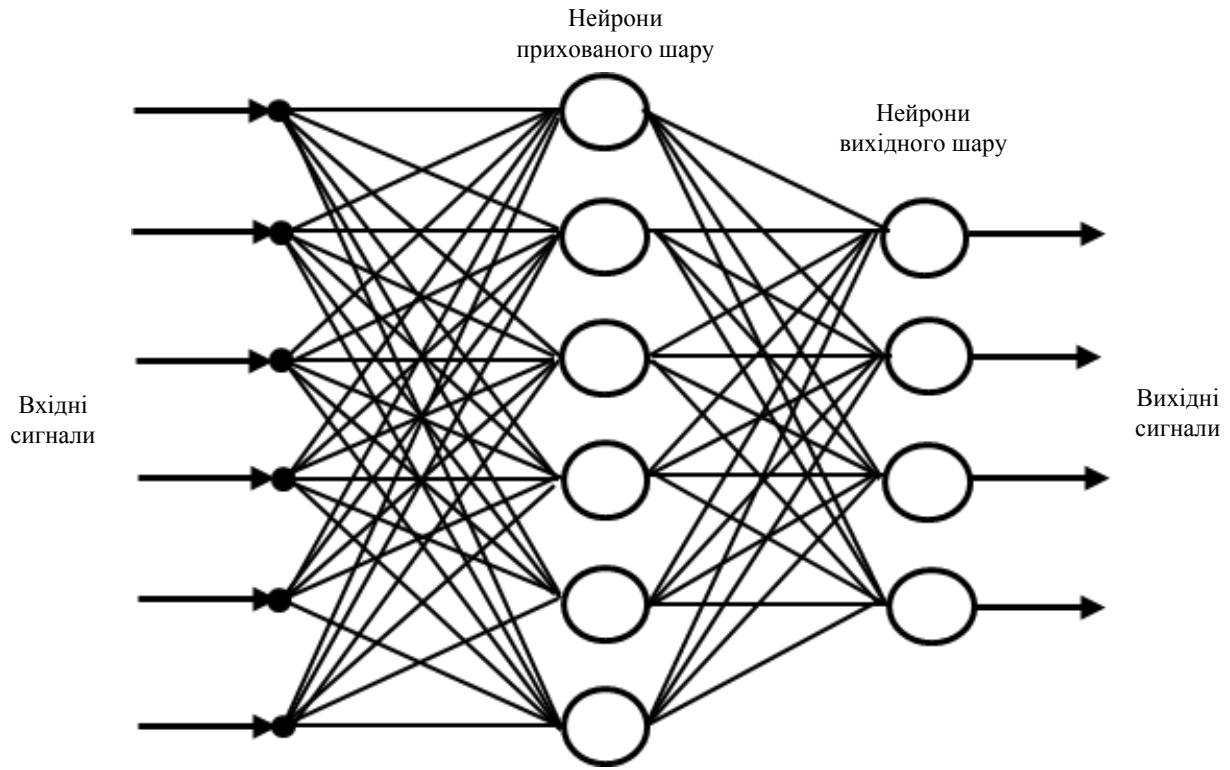
Таблиця 1.3 – Порівняльна характеристика аналогу та нової розробки

№	Показники	Одиниця виміру	Аналог - система StarLing	Система, що розробляється	Співвідношення параметрів нової системи до параметрів аналогу
1	2	3	4	5	6
1	Можливість працювати з текстами українською мовою		нема	є	-
2	Затрати часу на лексичний аналіз тексту у 10000 знаків	сек.	2	1,5	0,75
3	Точність лексичного аналізу тексту	%	80	88	1,1
4	Кількість виконуваних функцій	шт	3	1	0,33

Нова розробка є кращою ніж аналог. Вона є більш зручною у використанні, у неї вищі показники швидкості роботи та точності лексичного аналізу тексту

Аналіз предметної області (Вибір різновиду нейронної мережі)

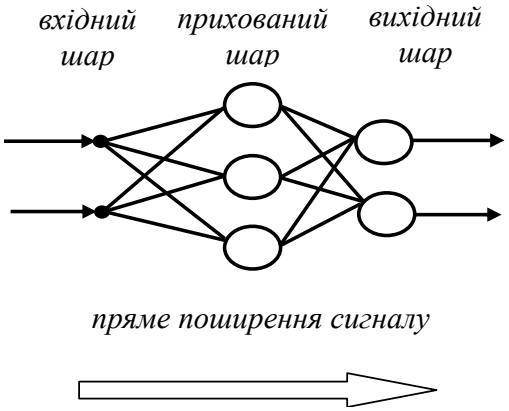
Було проведено порівняння нейронних мереж Кохонена, Хопфілда та багатошарового персептрона та обрано багатошаровий персептрон



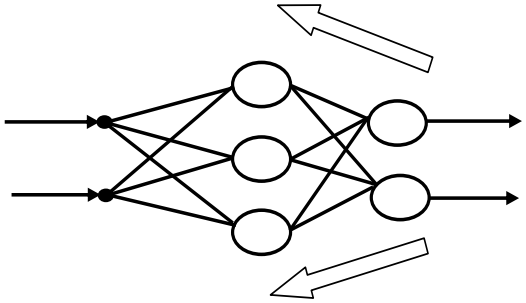
Структура багатошарового персептрона

МАТЕМАТИЧНА МОДЕЛЬ

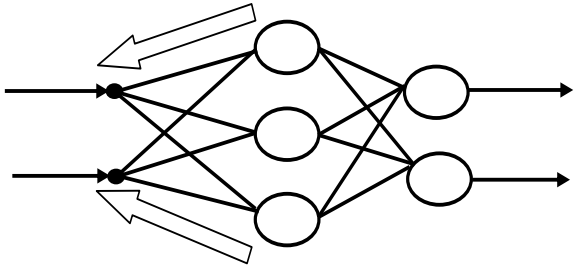
багатошарового персептрона з навчанням за методом зворотного поширення помилки



Пряме поширення сигналу



Зміна вагових коефіцієнтів вихідного шару (зворотне поширення помилки)



Зміна вагових коефіцієнтів прихованого шару (зворотне поширення помилки)

Нейрони вхідного шару обчислюють відповідні лінійні комбінації a_i:

$$a_i^{[n]} = \sum_j w_{ij}^{[n]} x_j^{[n]}$$

і передають їх на наступний шар, пропускаючи через нелінійну функцію активації:

$$x_i^{[n+1]} = f\left(a_i^{[n]}\right).$$

Нелінійна функція f називається активаційною, Однією з найбільш поширених є нелінійна функція з насиченням (сигмо їда):

$$f(x) = \frac{1}{1+e^{-\alpha x}}.$$

Цінна властивість сигмоїди – простий вираз для її похідної.

$$f'(x) = \alpha \cdot f(x) \cdot (1 - f(x))$$

а незв'язки кожного шару обчислюються під час зворотного поширення помилки від останнього шару (де вони визначаються по виходах мережі) до першого:

$$\delta_i^{[n]} = f'\left(a_i^{[n]}\right) \sum_k w_{ki}^{[n+1]} \delta_k^{[n+1]}.$$

Для побудови алгоритму навчання необхідно знати похідну помилки по кожному з ваг мережі:

$$\frac{\partial E}{\partial w_{ij}^{[n]}} = \frac{\partial E}{\partial a_i^{[n]}} \frac{\partial a_i^{[n]}}{\partial w_{ij}^{[n]}} \equiv \delta_i^{[n]} x_j^{[n]}.$$

Таким чином, внесок у загальну помилку кожної ваги обчислюється локально, простим множенням нев'язки нейрона $\delta_i^{[n]}$ на значення відповідного входу. (Через це, у разі, коли ваги змінюють за напрямом найшвидшого спуску

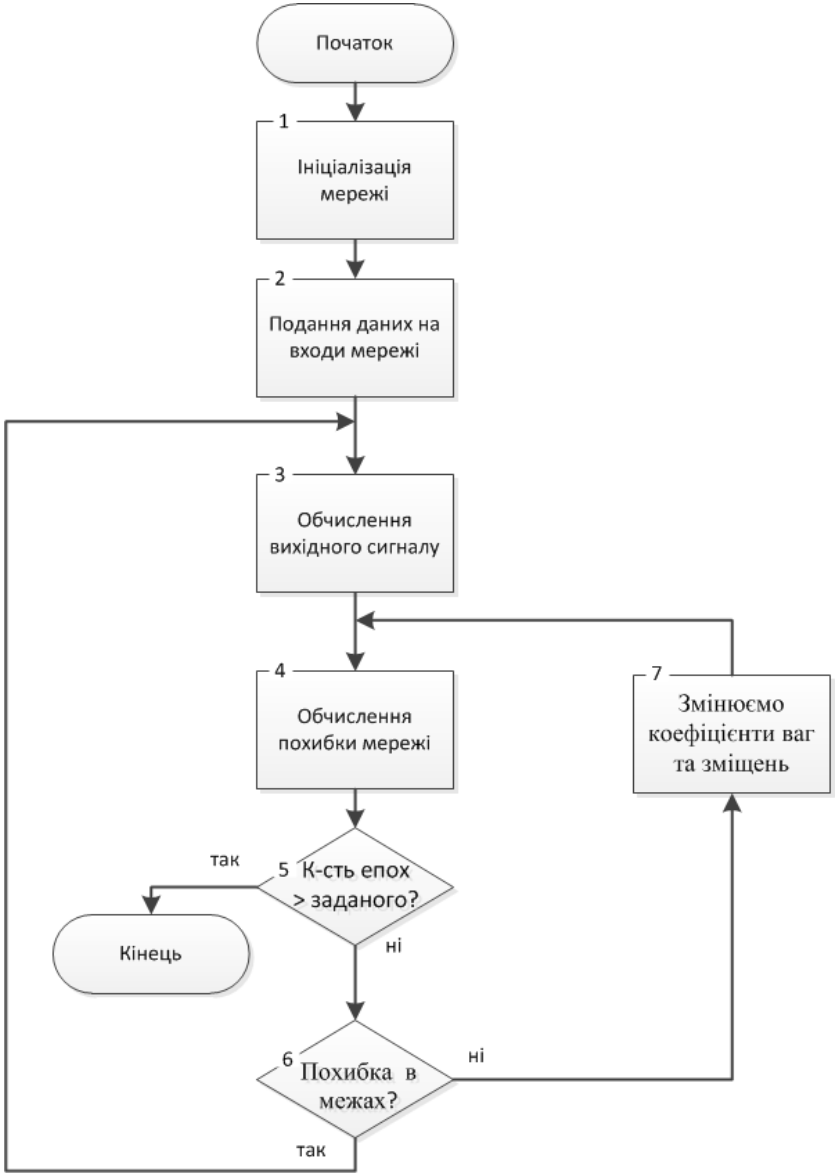
$$\Delta w_{ij} \propto -\partial E / \partial w_{ij} = -\delta_i x_j,$$

таке правило навчання називають дельта-правилом.)

Входи кожного шару обчислюються послідовно від першого шару до останнього під час прямого поширення сигналу:

$$x_i^{[n+1]} = f\left(\sum_j w_{ij}^{[n]} x_j^{[n]}\right),$$

Алгоритм навчання нейронної мережі



UML-діаграма класу NeuroNetwork

Клас NeuroNetwork
Закриті поля
string filename;
double[,] INP_PATTERNS;
double[,] OUT_PATTERNS;
int MAX_INP;
int MAX_HID;
int MAX_OUT;
int MAX_PAT;
double[] test_pat;
double[] desired;
neuron_type[] ipl;
neuron_type[] hl;
neuron_type[] ol;
double BETA;
double M;
int num_cycles;
Закриті методи
private double sigmoid(double x)
private void run_input_layer()
private void run_hidden_layer()
private void run_output_layer()
private void run_the_network()
private void display_the_results(out string[] outp)
private void AddWeightsToFile()
private void blank_changes()
private void calculate_output_layer_errors()
private void calculate_hidden_layer_errors()
private void calculate_input_layer_errors()
private void weight_change()
private void back_propagate()
private void ExtractWeights()
Відкриті методи
public void random_weights()
public void get_test_pattern(double[] tests)
public void train_the_network()
public string[] test_the_network(double[] test)
Public NeuroNetwork(double[,] INP_PATTERNS1, double[,] OUT_PATTERNS1, int Max_inp, int N_HID, int Max_pat, double beta, double m,int Epoch,string name,bool indicate)

ОПИС ІНТЕЛЕКТУАЛЬНОЇ ПРОГРАМНОЇ СИСТЕМИ

Інтелектуальна система виконана у вигляді Windows - додатку мовою C# в середовищі Microsoft Visual Studio і має віконний інтерфейс.

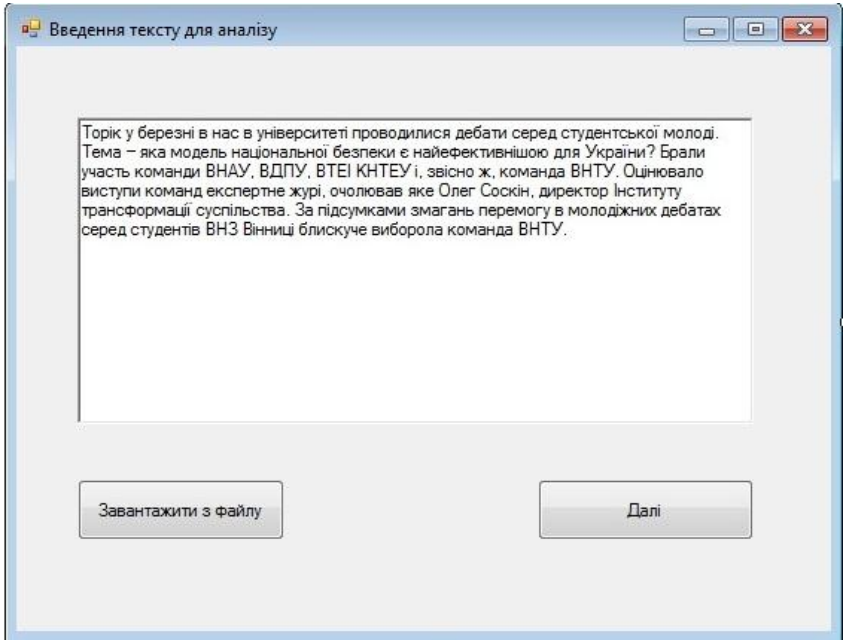
Принцип роботи програми полягає в спільному використанні для розпізнавання слів хеш-таблиці та нейромережі. Залежно від складності, тип слова визначається:

1) або за допомогою порівняння закінчення із використанням простих умовних операторів,
2) або пошуком слова в хеш-таблиці,
3) або за допомогою нейромережі (для випадку з однаковими закінченнями в різних частинах мови).

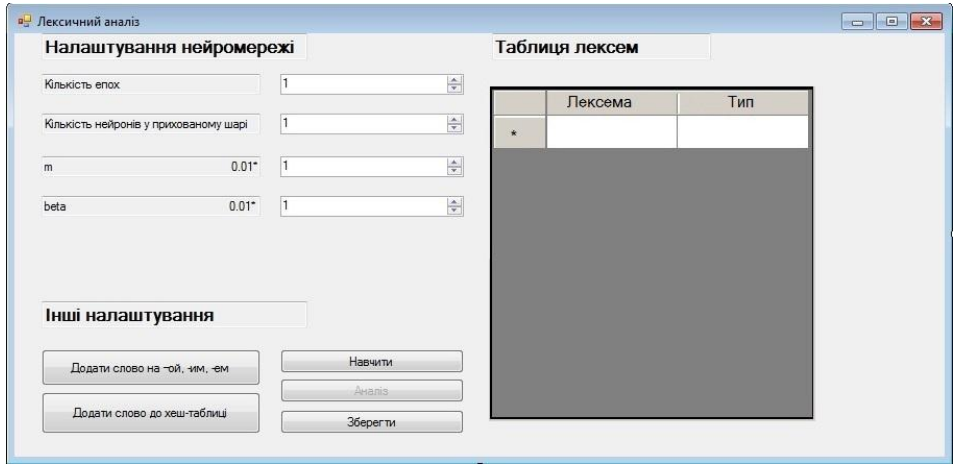
СКЛАД ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

Інтелектуальна система складається з трьох вікон:

вікно введення тексту для аналізу



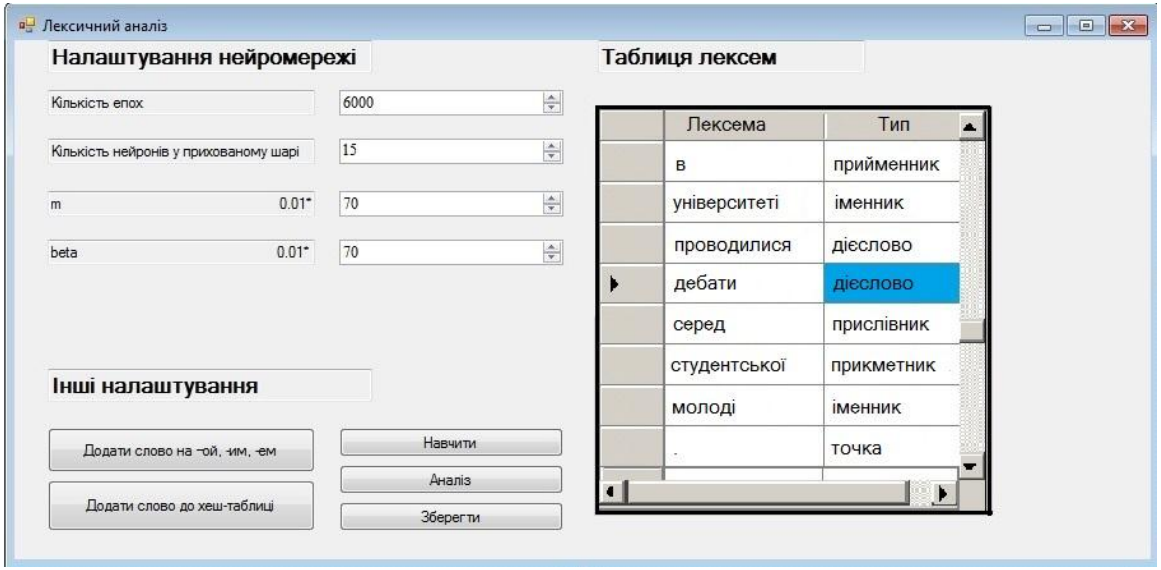
вікно аналізу тексту

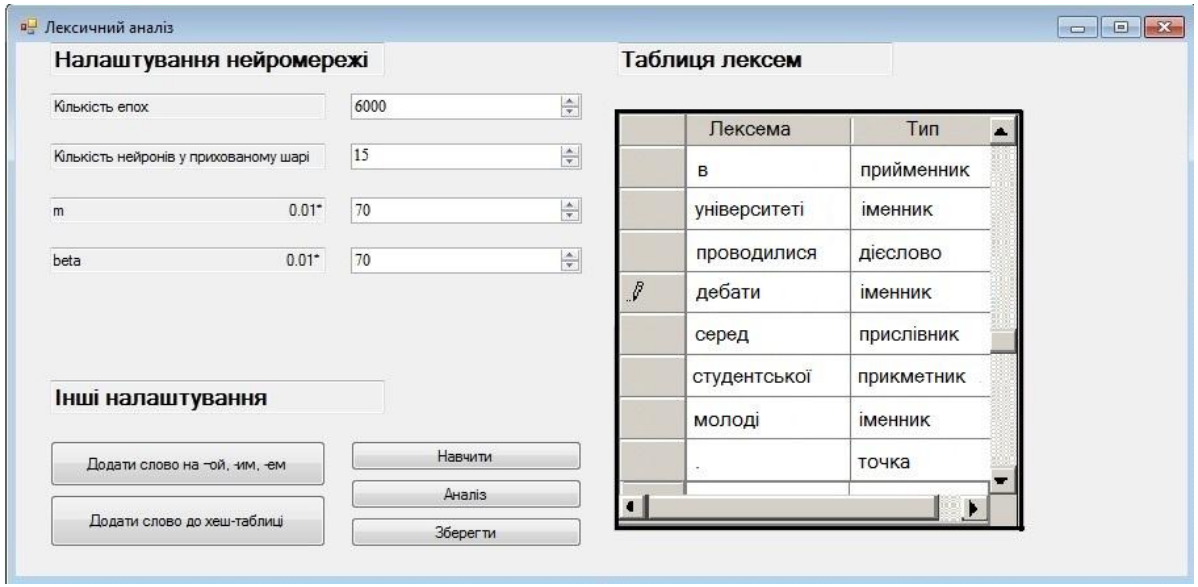


вікно додавання лексеми в хеш-таблицю або в навчальну вибірку

РЕЗУЛЬТАТИ ТЕСТУВАННЯ

При аналізі отриманого результату була виявлена помилка





Вікно програми з результатом аналізу без помилки

ЕКОНОМІЧНА ЧАСТИНА

Проведені відповідні економічні розрахунки підтверджують економічну доцільність розробки програмного забезпечення інтелектуальної системи для лексичного аналізу тексту, оскільки вона є дешевше ніж аналог на 1517,39 грн., термін її окупності складає менше року, а саме 1,5 місяці. Загальні витрати на розробку нового програмного продукту складають 25664,67 грн., прогнозований прибуток склав 169236,28 грн.

Висновок

- При виконання даної дипломної роботи було створено та досліджено інтелектуальну систему для лексичного аналізу тексту мовою програмування C# в середовищі розробки Microsoft Visual Studio. Інтелектуальна система побудована на основі нейромережі зворотного поширення помилки. При тестуванні, після відповідного налаштування, програма не допустила жодної помилки. Швидкодії лексичного аналізу тексту обсягом 10000 слів у розробленій системі склала 1,5 сек, а у програмі StarLing, яку взято за аналог - 2 сек. Це свідчить про те, що мета роботи досягнута, а саме – швидкодія розробленої системи вища, ніж швидкодія аналога.

Дякую за увагу.