**Slovenská technická univerzita v Bratislave**

**Fakulta informatiky a informačných technológii**

Backpropagation algoritmus

**Zadanie č. 3a**

Contents

[Popis zadania 3](#_Toc183986627)

[Reprezentácia údajov a konfigurácie 3](#_Toc183986628)

[Postup riešenia 3](#_Toc183986629)

[Porovnávanie parametrov 6](#_Toc183986630)

[Testovanie 6](#_Toc183986631)

[Najlepšie nájdene parametre 10](#_Toc183986632)

[**Zhodnotenie** 11](#_Toc183986633)

# Popis zadania

Úlohou bolo vytvoríte neurónovú sieť na riešenie XOR problému s použitím **NumPy** knižnice. Implementácie obsahuje lineárnu vrstvu, aktivačnú funkciu sigmoid, tanh, relu a chybovú funkciu MSE (mean squared error). Základný model obsahuje skrytu vrstvu s 4 neurónmi a výstupnú vrstvu s jedným neurónom.

# Reprezentácia údajov a konfigurácie

Na konfiguráciu bol použitý konfiguračný súbor, ktorý vyzerá nasledovne:

[Settings]

epoch\_count = 200

learning\_rate = 0.01

momentum = 0.9

use\_momentum = False

first\_activation\_function\_name(Sigmoid, Tanh, ReLU) = ReLU

second\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

# Postup riešenia

Na zatáčku programu bolo importovane knižnica pre náhodne generovanie, matematická Tato cast

* import numpy as np
* import matplotlib.pyplot as plt
* import time
* import os
* import configparser
* from colorama import Fore, init

Knižnice pre načrtnutie štatistických grafov, knižnicu na sledovanie času, cesty k súborom, farebné výstupy a konfiguračný súbor.

Všetky potrebné knižnice je potrebne nainštalovať použitím príkazu “pip install <názov knižnice>“.

Po implementácii knižníc bolo zadefinovaných niekoľko najdôležitejších funkcií, takých ako:

"""Parameters for the configuration file"""  
epoch\_count = 0  
learning\_rate = 0  
momentum = 0  
first\_activation\_function\_name = 'Sigmoid'  
second\_activation\_function\_name = 'Sigmoid'  
use\_momentum = False

Ďalej sú určené začiatočné parametre, ak konfiguračný súbor neexistuje, s následným načítaním konfiguračných parametrov zo súboru.

Ako funguje jednotlivé časti modelu je popísane v samotnom kóde v tvare komentárov.

Pre reprezentáciu MLP modelu bola vytvorená trieda, ktorá zahŕňa metódy, ako napríklad:

**Зображення, що містить текст, знімок екрана

Автоматично згенерований опис**

Na aktivačne funkcie bola tiež použitá trieda, s nasledujúcimi metódami:

**Зображення, що містить текст, знімок екрана, програмне забезпечення

Автоматично згенерований опис**

Na zatáčku boli nastavene vstupne a výstupne dáta pre XOR problém, inicializovaná model a vytvorené sloji modelu.

start\_time = time.time()  
  
"""Initialize initial input and output for XOR problem"""  
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])  
Y = np.array([[0], [1], [1], [0]])  
  
"""Initialize model"""  
model = MLP(X, Y)  
  
hidden\_layer\_size = 4  
  
"""Create matrix 2x4 for Weights and fill it with random values"""  
W1 = np.random.randn(2, hidden\_layer\_size) \* np.sqrt(1 / 2)  
W2 = np.random.randn(hidden\_layer\_size, 1) \* np.sqrt(1 / 4)  
  
B1 = np.zeros((1, hidden\_layer\_size))  
B2 = np.zeros((1, 1))

Načítane testovacie dáta a zatáčok trénovania modelu:

L1 = model.create\_layer(2, hidden\_layer\_size, W1, B1, activation=first\_activation\_function\_name)  
L2 = model.create\_layer(hidden\_layer\_size, 1, W2, B2, activation=second\_activation\_function\_name)  
  
"""Training data"""  
training\_data = [  
 (np.array([0, 0]).reshape(1, -1), np.array([0]).reshape(1, -1)),  
 (np.array([0, 1]).reshape(1, -1), np.array([1]).reshape(1, -1)),  
 (np.array([1, 0]).reshape(1, -1), np.array([1]).reshape(1, -1)),  
 (np.array([1, 1]).reshape(1, -1), np.array([0]).reshape(1, -1))  
]  
training\_data\_size = len(training\_data)  
losses = []  
  
for epoch in range(epoch\_count):  
 """Reshuffle the training data"""  
 np.random.shuffle(training\_data)  
  
 total\_error = 0  
  
 for i in range(training\_data\_size):  
 for input\_set, label in training\_data:  
 model.set\_X(input\_set)  
 model.set\_Y(label)  
  
 model.forward(input\_set)  
  
 error = model.get\_last\_layer\_error()  
 model.backward(error)  
  
 total\_error += model.MSE\_Loss\_evaluating()  
  
 average = total\_error / len(training\_data)  
 losses.append(total\_error / len(training\_data))  
  
 """Check stopping condition"""  
 if average < 0.00001 or epoch == 20000:  
 print("We reached the optimal model")  
 break  
  
 if (epoch + 1) % 10 == 0:  
 print(f"Epoch {epoch + 1}/{epoch\_count}, Loss: {average:.5f}")  
  
plt.plot(losses)  
plt.xlabel('Epoch')  
plt.ylabel('Average Loss')  
plt.title('Training Progress')  
plt.show()  
  
"""Testing model"""  
model.test\_model(training\_data)

Počas trénovania dopredaný a spetý smer(na propagáciu chyby a aktualizáciu váh a posun, medzi neurónmi). Taktiež bola použitá MSE chybová funkcia na ohodnotenie modelu.

# Testovanie

Počas testovania používal a uviedol rôzne metódy na trénovanie s momentom a bez, počet vrstiev a ich veľkosti, rôzne aktivačné funkcie(ReLU, Sigmoid, Tanh), rýchlosť učenia(learning rate) a rôzne začiatočne hodnoty. Počas testovania v niektorých prípadoch mal zle nastavenú chybovú funkciu.

Testovanie so skrytou vrstvou s 4 neurónmi a východnej s 1.

epoch\_count = 200

learning\_rate = 0.01

use\_momentum = False

first\_activation\_function\_name(Sigmoid, Tanh, ReLU) = ReLU

second\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

Зображення, що містить текст, знімок екрана, Графік, ряд

Автоматично згенерований опис sЗображення, що містить текст, знімок екрана, Шрифт, монітор

Автоматично згенерований опис

epoch\_count = 200

learning\_rate = 0.01

use\_momentum = False

first\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Sigmoid

second\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Sigmoid

Зображення, що містить текст, ряд, схема, знімок екрана

Автоматично згенерований опис

Pre Sigmoid bolo potrebne viac epoch pre nejdenia optimálnych parametrov   
Зображення, що містить текст, знімок екрана, Шрифт, монітор

Автоматично згенерований опис

epoch\_count = 200

learning\_rate = 0.01

use\_momentum = False

first\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

second\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

Зображення, що містить текст, ряд, схема, Графік

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

epoch\_count = 200

learning\_rate = 0.01

use\_momentum = False

first\_activation\_function\_name(Sigmoid, Tanh, ReLU) = ReLU

second\_activation\_function\_name(Sigmoid, Tanh, ReLU) = ReLU

Зображення, що містить текст, знімок екрана, схема, ряд

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Nevzdi nájde riešenie   
Зображення, що містить текст, знімок екрана, ряд, Графік

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, монітор

Автоматично згенерований опис

epoch\_count = 20

learning\_rate = 0.1

momentum = 0.9

use\_momentum = True

first\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

second\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

Зображення, що містить текст, знімок екрана, схема, Графік

Автоматично згенерований описЗображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

epoch\_count = 20

learning\_rate = 0.1

momentum = 0.95

use\_momentum = True

first\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

second\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

Зображення, що містить текст, знімок екрана, схема, ряд

Автоматично згенерований опис

epoch\_count = 500

learning\_rate = 0.01

momentum = 0.90

use\_momentum = True

first\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

second\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

Зображення, що містить текст, знімок екрана, схема, ряд

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, монітор

Автоматично згенерований опис

# Najlepšie nájdene parametre

epoch\_count = 20

learning\_rate = 0.1

momentum = 0.95

use\_momentum = True

first\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

second\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

Pre AND a OR problém dosiahnuť výsledkov je oveľa ľahšie:

epoch\_count = 5

learning\_rate = 0.01

momentum = 0.90

use\_momentum = True

first\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

second\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

Зображення, що містить текст, схема, ряд, Графік

Автоматично згенерований описЗображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

epoch\_count = 5

learning\_rate = 0.01

momentum = 0.90

use\_momentum = True

first\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

second\_activation\_function\_name(Sigmoid, Tanh, ReLU) = Tanh

Зображення, що містить текст, ряд, схема, знімок екрана

Автоматично згенерований описЗображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

# **Zhodnotenie**

Vytvorená neurónová sieť na riešenie AND, OR a XOR problémov s použitím Backpropagation algoritmu určená na trénovanie a testovanie vytvoreného systému s rôznymi parametrami, ako napríklad: použitie momentuma, vhodné aktivačné funkcie, rýchlosť učenia. Boli otestovane a porovnane výsledky testovania a vyhodnotená úspešnosť nerehneného modelu. Boli opísaná model, porovnanie pre rôzne rozmery a počet vrstiev pre vytvorený model.

Výtvory, natrénovaný model je schopný riešiť XOR problém s 100% úspešnosťou. Závisí od poctu epoch a nastavení modelu na trénovaní a inými nastaveniami. Najlepší výsledok bol opísaný v testovacej časti a dokázal riešiť problém za minimálny počet epoch.

Taktiež program umožňuje konfigurovať konfiguráciu pred behom programu. Čo výrazne zrýchľuje prístup k jednotným špecifikáciám a prispôsobeným nástrojom pre určite potreby klienta. Ako napríklad používateľ môže si zvoliť spôsob učenia site, aká najviac mu vyhovuje.

Dosť dôležitou úlohou bolo správne vybrať a nastrojiť parametre pre vytvorený algoritmus. Boli porovnané rôzne spôsoby konfigurácie, a navrhnutá najlepšia konfigurácia, ktorú sa dala vytvoriť.