

Домашня робота №3 з предмету МТІАД

Шевченко Юлія, ФІ-92

Завдання 1

Розрахувати результат мережі (нормалізація - сигмоїд) та обрахувати помилку (MSE) для мережі з параметрами.

```
In [1]: # Імпорт необхідних бібліотек
from scipy.special import expit
import random
import math
```

```
In [2]: x1 = 1
x2 = 0
w1 = 0.3
w2 = 0.8
w3 = 0.2
w4 = 0.1
w5 = 1.5
w6 = 1.8
```

```
In [3]: h1_input = (x1 * w1) + (x2 * w3)
h2_input = (x1 * w2) + (x2 * w4)
h1_output = round(expit(h1_input), 2)
h2_output = round(expit(h2_input), 2)
o1_input = (h1_output * w5) + (h2_output * w6)
o1_output = round(expit(o1_input), 2)
MSE = round((pow((1 - o1_output), 2) / 1), 3)

print(f"Input h1: {h1_input}")
print(f"Input h2: {h2_input}")
print(f"Output h1: {h1_output}")
print(f"Output h2: {h2_output}")
print(f"Input o1: {o1_input}")
print(f"Output o1: {o1_output}")
print(f"Network MSE: {MSE}")
```

```
Input h1: 0.3
Input h2: 0.8
Output h1: 0.57
Output h2: 0.69
Input o1: 2.097
Output o1: 0.89
Network MSE: 0.012
```

Змінити параметри вагів довільним чином 2 рази, розрахувати результат і помилку для змінених даних.

```
In [4]: x1 = 1
x2 = 0
w1_1 = round(random.uniform(0, 2), 2)
w2_1 = round(random.uniform(0, 2), 2)
w3_1 = round(random.uniform(0, 2), 2)
w4_1 = round(random.uniform(0, 2), 2)
w5_1 = round(random.uniform(0, 2), 2)
```

```

w6_1 = round(random.uniform(0, 2), 2)

h1_input_1 = (x1 * w1_1) + (x2 * w3_1)
h2_input_1 = (x1 * w2_1) + (x2 * w4_1)
h1_output_1 = round(expit(h1_input_1), 2)
h2_output_1 = round(expit(h2_input_1), 2)
o1_input_1 = (h1_output_1 * w5_1) + (h2_output_1 * w6_1)
o1_output_1 = round(expit(o1_input_1), 2)
MSE_1 = round((pow((1 - o1_output_1), 2) / 1), 3)

print(f"Input h1: {h1_input_1}")
print(f"Input h2: {h2_input_1}")
print(f"Output h1: {h1_output_1}")
print(f"Output h2: {h2_output_1}")
print(f"Input o1: {o1_input_1}")
print(f"Output o1: {o1_output_1}")
print(f"Network MSE: {MSE_1}")

```

```

Input h1: 1.56
Input h2: 0.97
Output h1: 0.83
Output h2: 0.73
Input o1: 0.17359999999999998
Output o1: 0.54
Network MSE: 0.212

```

In [5]:

```

x1 = 1
x2 = 0
w1_2 = round(random.uniform(0, 3), 2)
w2_2 = round(random.uniform(0, 3), 2)
w3_2 = round(random.uniform(0, 3), 2)
w4_2 = round(random.uniform(0, 3), 2)
w5_2 = round(random.uniform(0, 3), 2)
w6_2 = round(random.uniform(0, 3), 2)

h1_input_2 = (x1 * w1_2) + (x2 * w3_2)
h2_input_2 = (x1 * w2_2) + (x2 * w4_2)
h1_output_2 = round(expit(h1_input_2), 2)
h2_output_2 = round(expit(h2_input_2), 2)
o1_input_2 = (h1_output_2 * w5_2) + (h2_output_2 * w6_2)
o1_output_2 = round(expit(o1_input_2), 2)
MSE_2 = round((pow((1 - o1_output_2), 2) / 1), 3)

print(f"Input h1: {h1_input_2}")
print(f"Input h2: {h2_input_2}")
print(f"Output h1: {h1_output_2}")
print(f"Output h2: {h2_output_2}")
print(f"Input o1: {o1_input_2}")
print(f"Output o1: {o1_output_2}")
print(f"Network MSE: {MSE_2}")

```

```

Input h1: 0.33
Input h2: 0.72
Output h1: 0.58
Output h2: 0.67
Input o1: 1.7118999999999998
Output o1: 0.85
Network MSE: 0.023

```

Розрахувати Root MSE для загального результату з трьох проходів.

In [6]:

```

RMSE = round(math.sqrt((pow((1 - o1_output), 2) + pow((1 - o1_output_1), 2) + pow((1 - o1_
print(f"Network RMSE: {RMSE}")

```

Network RMSE: 0.286

Завдання 2

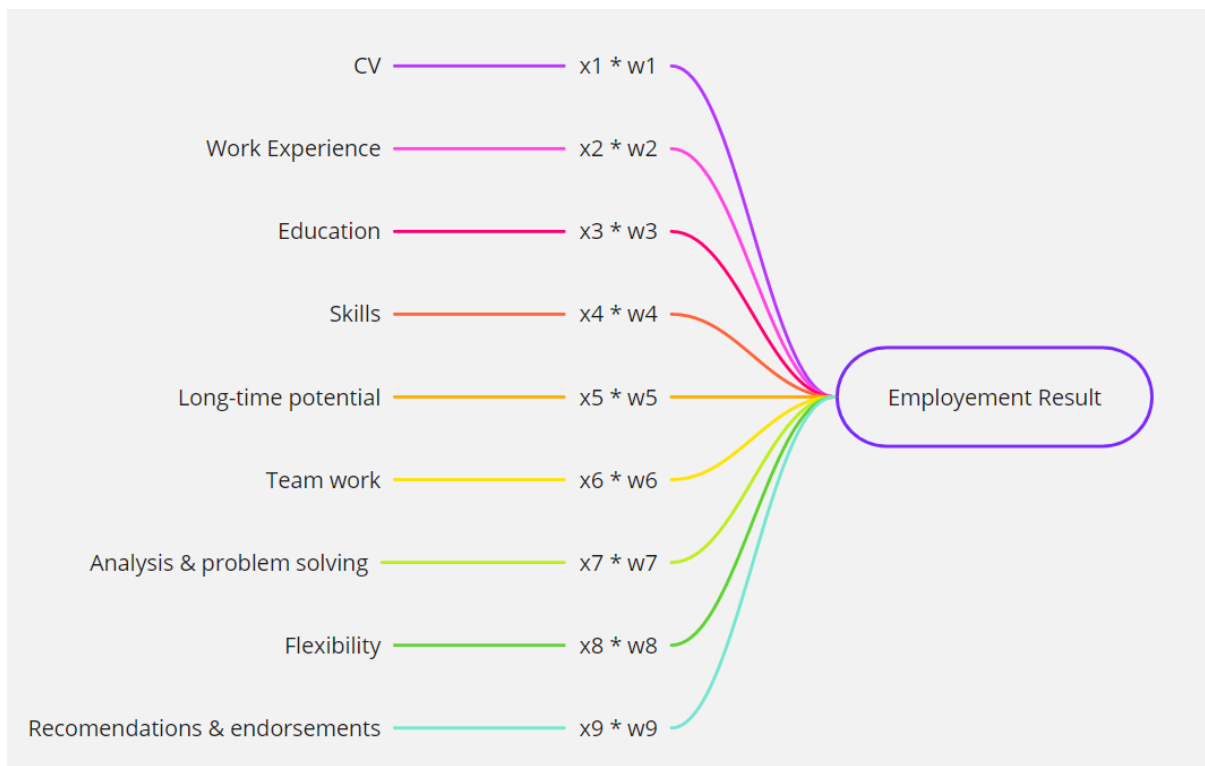
Для прикладу задачі для розв'язання, я оберу задачу HR-відділу з пошуку кандидата на посаду.

Отже для того, аби проаналізувати, чи візьме HR-відділ ту чи іншу особу на дану посаду нам потрібно визначити ключові фактори – вхідні дані. Створимо перелік основних факторів.

- Резюме (при поганому резюме кандидатура особи не буде розглядатися)
- Досвід роботи (при дуже малому досвіді менше бажання розглядати кандидатуру особи)
- Отримана освіта (це також є не останнім по важливості фактором)
- Skills (навички людини – як soft так і hard)
- Довготривалий потенціал (більшу перевагу мають кандидати, які планують працювати в одній компанії на довготривалій перспективі)
- Досвід та спроможність роботи в команді (це є немало важливим фактором, особливо якщо робота вимагає командної роботи над одним великим проектом)
- Спроможність критичного мислення, аналізу та пошуку рішення при будь-якій ситуації
- Вміння адаптуватися та вчитися
- Хороші відгуки про кандидата

Таких даних нам повністю вистачить для того, аби після аналізу визначити, чи візьме HR-відділ певну особу на посаду.

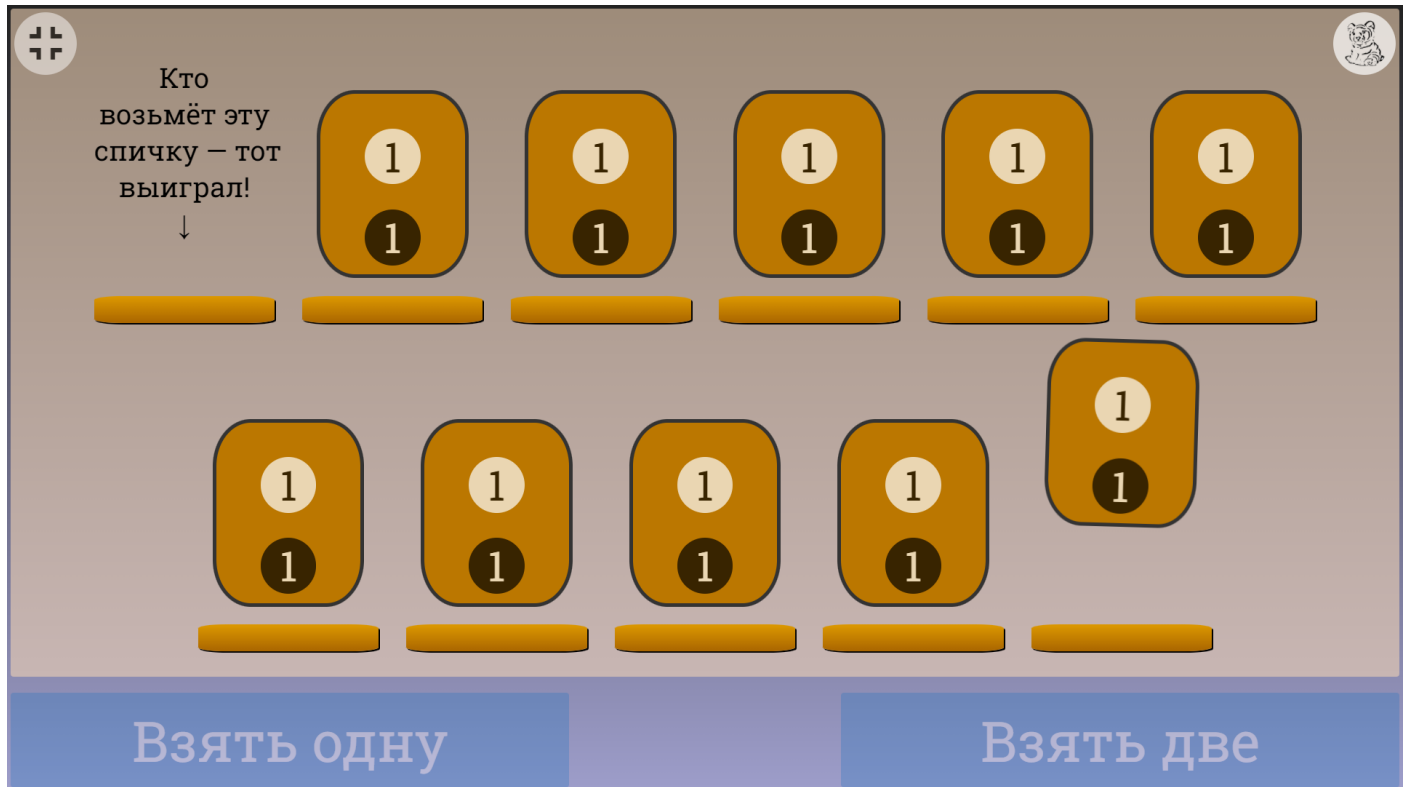
Можемо розробити приблизний малюнок нейромережі для її розв'язання.



Завдання 3

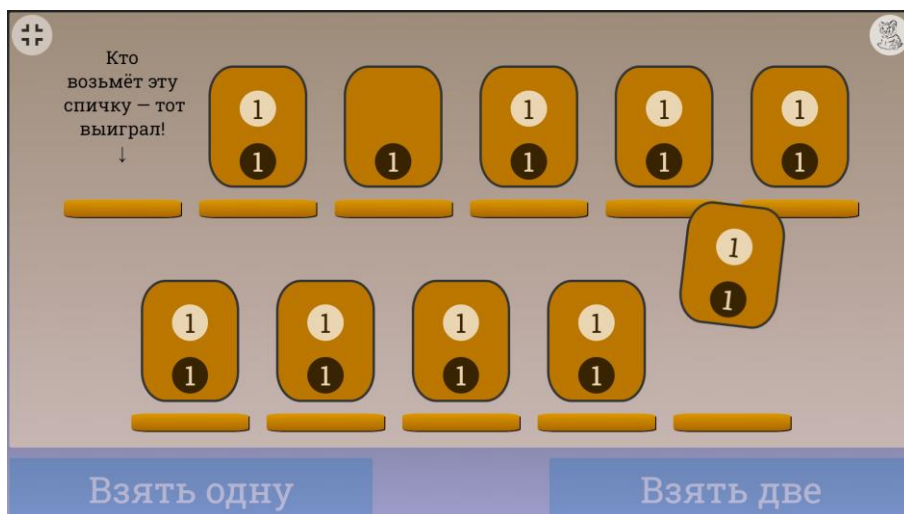
Наступним завданням є зіграти в гру з елементарною [нейромережею](#).

В цій грі можна брати одну паличку або дві. Хто візьме останню – той виграв. Проти нас грає нейромережа з коробочок – вона робить випадкові ходи, але після кожної партії буде отримувати нагороду і розумнішати. Чорний камінець – мережа візьме одну паличку. Білий – дві.

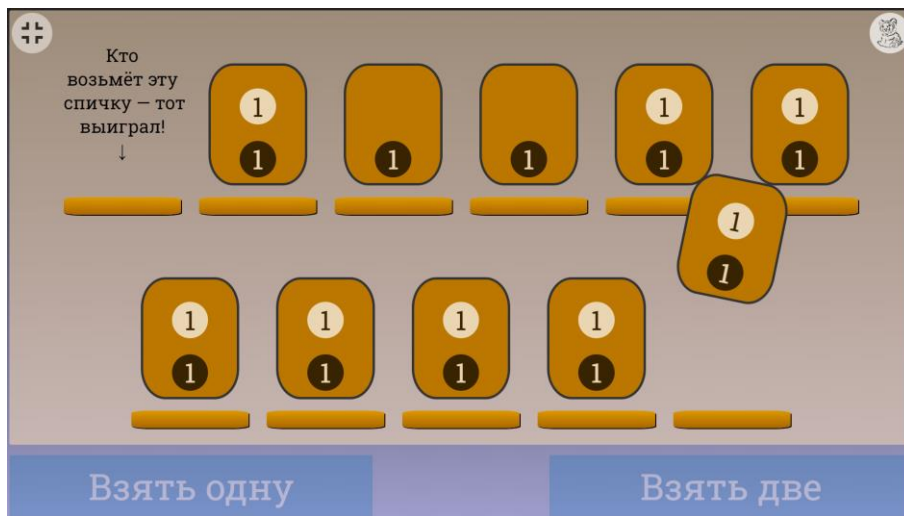


Ось таким чином виглядає поле для гри на початку.

Перший раунд закінчився програшем нейромережі. І тому поле вже виглядає інакше.



Другий раунд також закінчився програшем мережі. Нейромережа не знає правил гри, але за рахунок штрафів і нагород навчається після кожного раунду. Поле вже виглядає наступним чином.



Коли нейромережа програє – за це у неї відбирають останній камінець, який зіграв. Отже, в наступний раз у неї такого вибору не буде і цей останній провальний хід не випаде.

Якщо у нас трапляється порожній стаканчик – це мертвий нейрон. Комп'ютер розуміє, що далі грати нема сенсу і одразу здається.

Коли нейромережа перемагає – за це їй в стаканчики, за які було зроблено правильні кроки, підсипають ще камінців того ж кольору. В наступний раз у них буде більше шансів випасти.

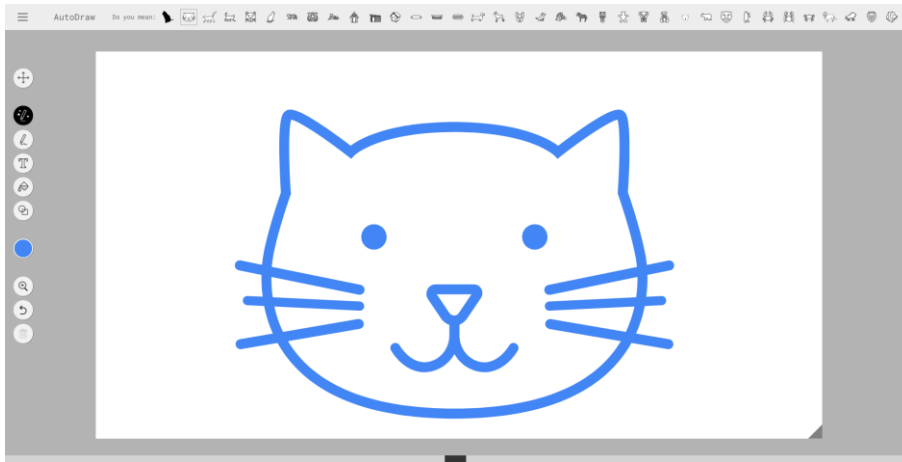
У моєму випадку, нейромережа змогла вперше виграти після 7-го раунду.

І після 19-го раунду нейромережа змогла виграти 3 гри підряд.

Завдання 4

Мною було протестовано декілька нейромереж з запропонованих. Ось, які мені найбільше сподобалися і їх результати.

- [AutoDraw](#)



- [ThisCatDoesNotExist](#)



- [Image-to-Image](#)

edges2cats

