



# „Sieci neuronowe”

„Uczenie kolorów”

Metody sztucznej inteligencji

Adrian Rudnicki - 39662

Oliwier Jędrzejczak - 41264

**Prowadzący:**

mgr inż. Krystian Sadowski

**Spis treści**

I.	Temat.....	2
II.	Opis.....	2
III.	Funkcjonalność.....	2
IV.	Kod i algorytm.....	3
V.	Podsumowanie.....	3

### I. Temat

Wybieranie kolorów (Colorpicker) - algorytm, który uczy się wyboru użytkownika i na podstawie nauki wybiera koloru tekstu do losowo przedstawionego koloru tła.

### II. Opis

Sieć neuronowa uczy się, na podstawie wyborów użytkownika dobierać kolor czcionki tak, aby tekst był czytelny na losowo dobranym kolorze tła. Użytkownik odpowiada na 20 losowych zdarzeń, a sieć neuronowa zapamiętuje odpowiedzi i uczy się na ich podstawie. Dzięki temu po udzieleniu odpowiedzi na wszystkie pytania, zaczyna odpowiadać sieć neuronowa dobierając odpowiedni kolor tekstu, oraz wyświetla procentową wartość pewności danej decyzji.

### III. Funkcjonalność

- Użytkownik ma dostęp do “wyboru koloru” z poziomu przeglądarki

## Colorpicker dla tekstu

Aplikacja, która dzięki sieci neuronowej dobiera najlepszy kolor tekstu tak aby był widoczny na losowo wybranym kolorze tła.

AI potrzebuje uzyskać od użytkownika 20 odpowiedzi, aby nauczyć się odpowiedniego zachowania.

Pozostało **20** kroków

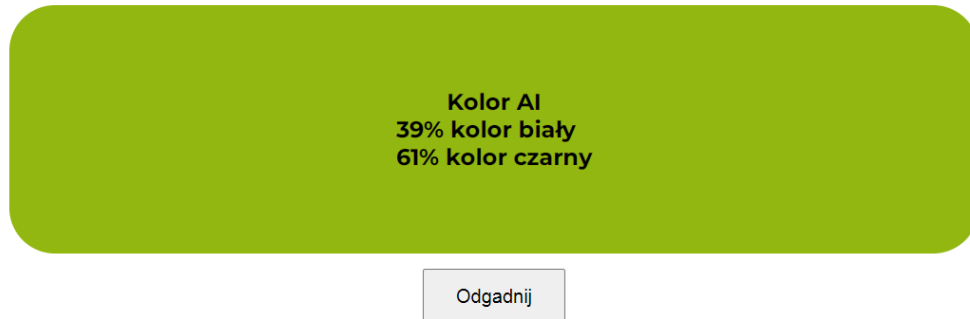
The image shows a user interface for a color picker. It features a large red rounded rectangle. Inside this rectangle, the text "Biały tekst" (White text) is displayed in white, and "Czarny tekst" (Black text) is displayed in black. Below the red rectangle, there are two buttons: "Biały" (White) and "Czarny" (Black).

- Po przebrnięciu przez proces uczenia sieci neuronowej, zostaje wyświetlona opcja “odgadnij”, która daje znak sieci neuronowej, że to ona będzie musiała dobrać odpowiedni kolor tekstu. Pojawienie się tej opcji oznacza również to, że sieć neuronowa zakończyła swoją naukę.

## Colorpicker dla tekstu

Aplikacja, która dzięki sieci neuronowej dobiera najlepszy kolor tekstu tak aby był widoczny na losowo wybranym kolorze tła.

AI potrzebuje uzyskać od użytkownika 20 odpowiedzi, aby nauczyć się odpowiedniego zachowania.



### IV. Kod i algorytm

- System dostaje informację, który kolor wybrał użytkownik i uruchamia funkcję przekazując odpowiednio dopasowany wybór i jeśli system zauważy, że etap uczenia się kończy, uruchamia sieć neuronową, która zaczyna podejmować decyzje.

```
whiteButton.addEventListener("click", () => {
  chooseColorByUser(1);
});

blackButton.addEventListener("click", () => {
  chooseColorByUser(0);
});
```

```
function chooseColorByUser(value) {
  if (learn > 1) {
    data.push({
      input: color,
      output: [value],
    });
    learn--;
    setRandomColor();
    step.innerHTML = `Pozostało <strong>${learn}</strong> kroków`;
    return;
  }

  whiteButton.style.display = "none";
  blackButton.style.display = "none";
  whiteText.style.display = "none";
  blackText.style.display = "none";
  step.style.display = "none";

  guessButton.style.display = "";
  guessText.style.display = "";

  net.train(data);
  guessTextColor();
}
```

- następnie generowany jest losowy kolor tła, który jest przypisywany do obiektu "color", a ten obiekt jest wykorzystywany przez sieć neuronową do podjęcia decyzji o kolorze tekstu.

```
function setRandomColor() {  
  color = {  
    r: Math.random(),  
    g: Math.random(),  
    b: Math.random(),  
  };  
  colorElement.style.backgroundColor = `rgba(${color.r * 255}, ${  
    color.g * 255  
  }, ${color.b * 255})`;  
}
```

- Po naciśnięciu przycisku “odgadnij” uruchamiana jest funkcja, która analizuje wszystkie dane i podejmuje decyzję i wyświetla ją w oknie przeglądarki

```
function guessTextColor() {  
  setRandomColor();  
  const guess = net.run(color)[0];  
  guessText.style.color = guess > 0.5 ? "#fff" : "#000";  
  guessProcent.innerHTML = `${parseInt(guess * 100)}% kolor biały<br />${  
    100 - parseInt(guess * 100)  
  }% kolor czarny`;  
}
```

## V. Podsumowanie

- Biblioteka, z której skorzystaliśmy [BRAIN.JS](#).
- Pliki źródłowe projektu dostępne na [REPOZYTORIUM SYSTEMU](#)
- Aplikacja dostępna pod adresem [COLORPICKER](#)