Program sieci neuronowej
Death Timer

Szymon Sandura
Patryk Reszczyński

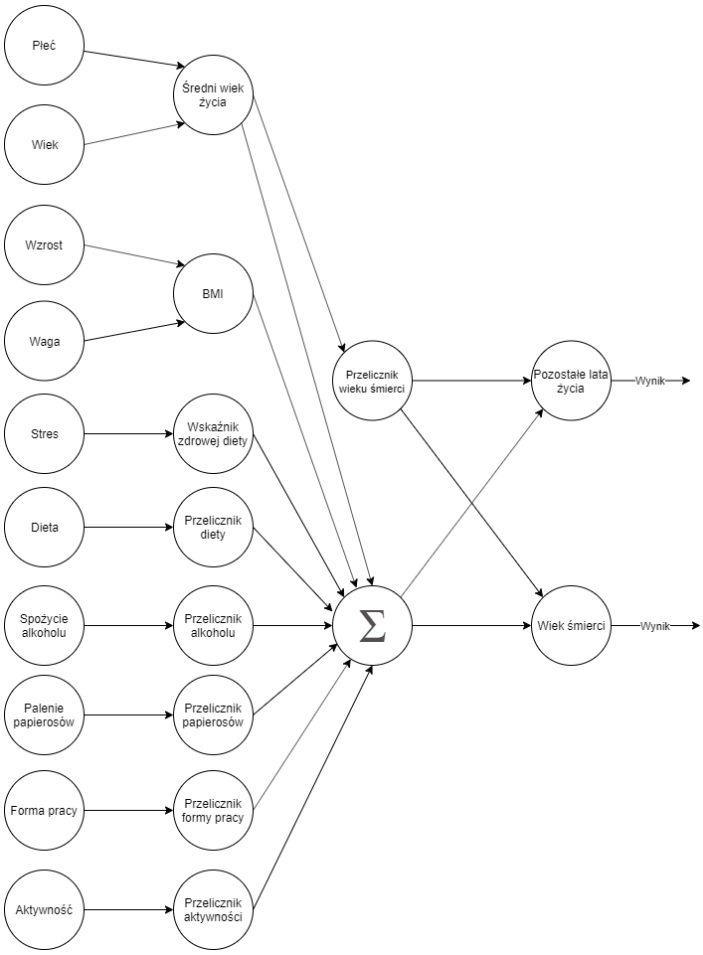
Spis treści

Wstęp

Program z wbudowanym mechanizmem sieci neuronowej ma na celu przeliczyć sugerowany czas śmierci i długość życia użytkownika. Zadaniem aplikacji jest uzmysłowienie odbiorcy, że na nasze życie wpływ mają takie aspekty jak spożywanie alkoholu i palenie papierosów, dieta, forma pracy i aktywność fizyczna czy utrzymywana waga. Użytkownik po skorzystaniu z aplikacji może się zreflektować i podjąć działania w kierunku zdrowszego i dłuższego trybu życia.

Algorytm działania

Użytkownik wprowadza dane o sobie za pośrednictwem interfejsu aplikacji. Algorytmy przeliczają zadeklarowane wartości na odpowiednie wartości istotne dla uzyskania wyniku. Wybrane opcje i wpisane dane oddziałują między sobą modyfikując otrzymaną ilość lat. Na przykład: wprowadzony wzrost i waga przeliczane są na BMI (Body Mass Index ang. Wskaźnik Masy Ciała), zależnie od uzyskanej wartości zmniejsza się lub zwiększa uzyskany sugerowany wiek, do którego użytkownik dożyje. W algorytmie zastosowano metodę uczenia pod nadzorem, dzięki której otrzymujemy pożądane dane wyjściowe.



Rysunek 1: Algorytm działania

Fragment kodu programu

```
class WeightedInput
{
    2 references
    public int Gender { get; set; }
    0 references
    public int Age { get; set; }
    0 references
    public int Height { get; set; }
    0 references
    public int Weight { get; set; }
    2 references
    public int Stress { get; set; }
    1 reference
    public int Calories { get; set; }
    3 references
    public int Alcohol { get; set; }
    2 references
    public int Cigarettes { get; set; }
    2 references
    public int Job { get; set; }
    2 references
    public int Activity { get; set; }
}
```

Rysunek 2: Wprowadzane dane

```
class Converter
{
    public WeightedInput Input = new WeightedInput();
    private double BMI;
    2 references
    public double Output { get; set; }

    2 references
    public double Calculate()
    {
        Output = Input.Activity
            + Input.Alcohol
            + Input.Calories
            + Input.Cigarettes
            + Input.Gender
            + Input.Job
            + Input.Stress
            + this.BMI;
        return Output;
    }
    1 reference
    public void ConvertActivity(string bindedInput)
    {
       Input.Activity = (bindedInput=="Niska") ? -5 : (bindedInput=="Przeciętna") ? 0 : 10 ;
    }
    1 reference
    public void ConvertAlcohol(string bindedInput)
    {
        Input.Alcohol = (bindedInput == "Niskie/brak") ?  0 : (bindedInput == "Średnie") ?  -5 :  -15;
    }
    1 reference
    public void ConvertCalories(string bindedInput)
    {
        Input.Alcohol = (bindedInput == "Średnia") ? 0 : -5;
    }
    1 reference
    public void ConvertCigarettes(string bindedInput)
    {
        Input.Cigarettes = (bindedInput == "Brak") ?  0 : (bindedInput == "5-10+") ? -5 : -15;
    }
    1 reference
    public void ConvertGender(string bindedInput)
    {
        Input.Gender = (bindedInput == "M") ? 74 : 82;
    }
    1 reference
    public void ConvertJob(string bindedInput)
    {
        Input.Job = (bindedInput == "Biurowa") ? -5 : (bindedInput == "Mieszana") ? +5 : 0;
    }
    1 reference
    public void ConvertStress(string bindedInput)
    {
        Input.Stress = (bindedInput == "Niski") ? 5 : (bindedInput == "Przeciętny") ? 0 : -5;
    }
    2 references
    public void ConvertBMI(string Weight, string Height)
    {
        this.BMI = Double.Parse(Weight) / Math.Pow(Double.Parse(Height) / 100, 2);
        this.BMI = (BMI > 40) ? -30 : (BMI > 30) ? -15 : (BMI >= 25) ? -5 : (BMI > 19) ? 5 : -5;
    }
```

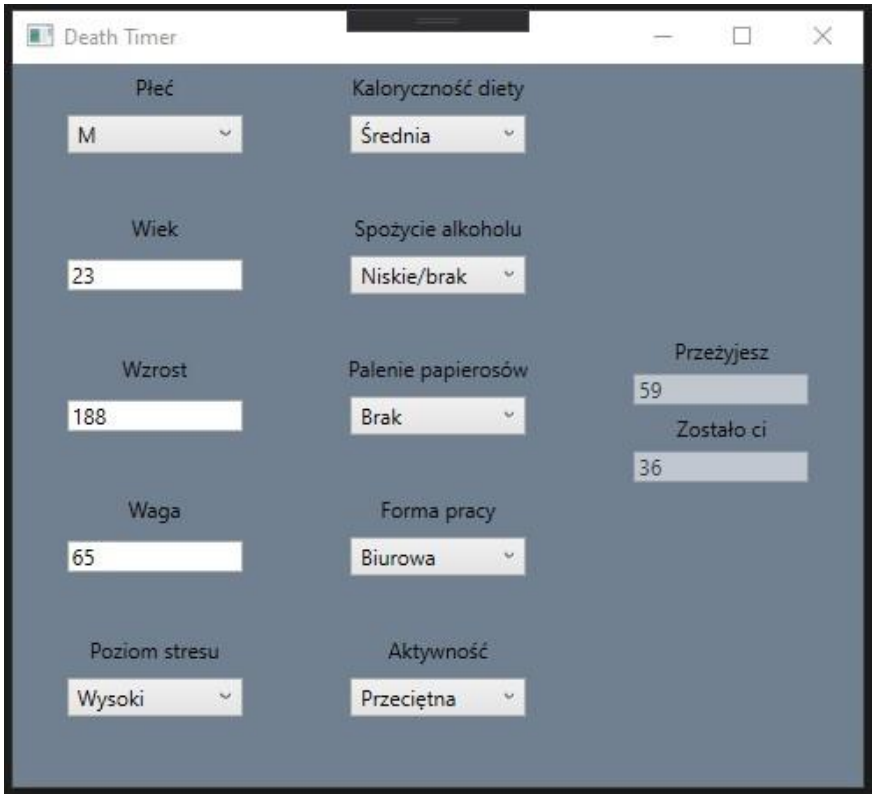
Rysunek 3: Przeliczniki danych

```
private void Refresh()
{
    Output = MyConverter.Calculate();
    if ((int)Output - Int32.Parse(Age) <= 0)
    {
        OutputLeft = 0;
    }
    else
    {
        OutputLeft = (int)Output - Int32.Parse(Age);
    }

}
```

Rysunek 4: Wyniki

Interfejs użytkownika



Rysunek 5: Interfejs użytkownika

Wnioski

Sztuczna inteligencja w oparciu o sieci neuronowe jest dziedziną o szerokim spektrum wykorzystania oraz wysokim poziomie skomplikowania implementacji. Wykorzystywane są do analizy, prognozowania i planowania. Powyższy projekt ma na celu zwizualizować mechanizm działania sieci neuronowych. Algorytm, po wprowadzeniu danych wejściowych przez użytkownika, prognozuje wiek, w którym można spodziewać się nadejścia śmierci. Dzięki metodzie uczenia pod nadzorem mogliśmy skonstruować aplikację prostą w konstrukcji i zrozumieniu.