

## SYSTEMY AI – LAB 8 (2023)

Zanim rozpoczniesz zadania zapoznaj się z notatnikiem: [SystemyAI\\_lab\\_8\\_1\\_MNIST.ipynb](#)

### Zadanie 1

Zastosuj **regresję logistyczną** do problemu z dwoma gangami **obliczając wprost gradient funkcji błędu**. Wykorzystaj następującą definicję (tzn. entropia krzyżowa):

```
def loss_fn(y, y_model):  
    return tf.reduce_mean(-y*tf.math.log(y_model)-  
                           (1-y)*tf.math.log(1-y_model))
```

oraz funkcję `tf.sigmoid`.

- A. Sprawdź działanie modelu dla 3 wybranych punktów.
- B. Przetestuj działanie modelu dla kilku różnych wartości hiperparametrów: **ilość epok** i **współczynnik uczenia**.
- C. Wykorzystując funkcję `subset_dataset` (zmodyfikuj ją o ile to konieczne) przetestuj działanie modelu w wersji **mini batch** dla różnych wielkości **batch**.
- D. Dla jakich wartości hiperparametrów uzyskałeś najlepszy/najgorszy wynik?

### Zadanie 2

Zastosuj **regresję softmax** do problemu z dwoma gangami. **Wykorzystaj bibliotekę Keras**.

- A. Przygotuj dane. Pamiętaj o odpowiedniej postaci wartości oczekiwanych.  
(Podpowiedź: operacja `np.array([[1,2]])*2+np.array([[3,4]])*3` daje wynik: `array([[11, 16]])`)
- B. Przetestuj działanie modelu dla **różnych wartości hiperparametrów**. Dla jakich wartości uzyskałeś najlepszy/najgorszy wynik?
- C. Przyjmij podział zbioru danych: **80%** - dane **treningowe**, **20%** - dane **walidacyjne**.
- D. Przetestuj działanie modelu w wersji **mini batch** dla różnych wielkości **batch**.
- E. Wygeneruj wykres zmiany błędu dla **danych treningowych i walidacyjnych**.

### Zadanie 3

To samo co w **zadaniu 2**, ale dla problemu **trzech gangów** (UWAGA: musisz wygenerować odpowiedni zbiór danych).

### Zadanie 4

Zastosuj **regresję softmax** do problemu z dwoma gangami. **Gradient funkcji błędu oblicz wprost**.

- A. Parametry zapisz w macierzach:

```
w = tf.Variable(np.random.random((2,2)))  
b = tf.Variable(np.random.random((2)))
```

B. Wykorzystaj funkcję błędu:

```
loss_fn = tf.keras.losses.BinaryCrossentropy()
```

C. Do wyliczenia predykcji modelu wykorzystaj operację mnożenia macierzy:

```
tf.matmul(A,B)
```

D. Wykorzystaj funkcję aktywacji:

```
tf.nn.softmax()
```

E. Przyjmij podział zbioru danych: **80%** - dane **treningowe**, **20%** - dane **walidacyjne**.

F. Przetestuj działanie programu dla kilku różnych wartości parametrów: **ilość epok**, **współczynnik uczenia**, **wielkość batch**. Dla jakich wartości wynik jest najlepszy/najgorszy.

### **Zadanie 5**

To samo co w **zadaniu 4**, ale dla problemu **trzech gangów**.