

```

#Matematyka Konkretna

#Laboratorium 9

#Paweł Wawrzuta https://github.com/PawelWawrzuta/MK-Lab9

#Wariant 9

import tensorflow as tf
import numpy as np

# Tworzymy dane treningowe
def generate_data(num_samples=1000):
    X = np.random.randint(0, 2, size=(num_samples, 16, 2)) #
    Generujemy dwie liczby binarne o długości 16 bitów
    Y = np.abs(X[:, :, 0] - X[:, :, 1]) # Obliczamy różnicę dwóch
    liczb binarnych

    # Zmiana sumy na różnicę dla każdej liczby 12-bitowej
    X = X[:, :12, :] # Wybieramy tylko pierwsze 12 bitów
    Y = np.abs(X[:, :, 0] - X[:, :, 1]) # Obliczamy różnicę dwóch
    liczb binarnych dla pierwszych 12 bitów

    return X, Y

# Tworzymy model RNN
model = tf.keras.Sequential([
    tf.keras.layers.SimpleRNN(8, input_shape=(12, 2),
    activation='relu', return_sequences=True),
    tf.keras.layers.SimpleRNN(8, activation='relu'),
    tf.keras.layers.Dense(12, activation='sigmoid')
])

# Kompilujemy model
model.compile(optimizer='adam', loss='mean_squared_error',
metrics=['mae'])

# Generujemy dane treningowe
X_train, Y_train = generate_data()

# Trenujemy model
model.fit(X_train, Y_train, epochs=10, batch_size=32)

# Testujemy model na nowych danych
X_test, Y_test = generate_data(10)
predictions = model.predict(X_test)

# Wyświetlamy wyniki
for i in range(10):
    input_data = X_test[i]
    true_output = Y_test[i]

```

```
predicted_output = predictions[i].round()

print(f"Wejscie: {input_data}")
print(f"Prawdziwa roznica: {true_output}")
print(f"Przewidziana roznica: {predicted_output}")
print()
```