


Informatyka Stosowana		
Laboratorium 6	PyTorch - Pakiet Autograd	 <b>POLITECHNIKA BYDGOSKA</b> Wydział Telekomunikacji, Informatyki i Elektrotechniki
Przedmiot	Matematyczne Podstawy Sztucznej Inteligencji	
Prowadzący	mgr inż. Gracjan Kątek	

## 1. Wprowadzenie

Automatyczne różniczkowanie to technika obliczania pochodnych funkcji w sposób programowy, co jest kluczowe w algorytmach uczenia maszynowego. PyTorch umożliwia łatwe obliczanie gradientów za pomocą pakietu `autograd`.

Każdy tensor z flagą `requires_grad=True` śledzi operacje, które zostały na nim wykonane. Dzięki temu można automatycznie obliczyć pochodne tych operacji względem tensorów.

**Przykład:** Obliczenie gradientu funkcji  $y = 3x^2 + 2x + 1$  w punkcie  $x=2$ :

```
import torch

# Tworzenie tensora z requires_grad=True
x = torch.tensor(2.0, requires_grad=True)

# Definicja funkcji
y = 3 * x**2 + 2 * x + 1

# Obliczenie gradientu
y.backward()

# Wyświetlenie gradientu
print("Wartość y:", y.item())
print("Gradient (dy/dx):", x.grad.item())
```

**Spodziewany wynik:**

- Wartość  $y = 17$  (dla  $x=2$ ).
- Gradient  $\frac{dy}{dx} = 6x + 2 = 14$

## 2. Zadania do samodzielnego wykonania

### Zadanie 1: Podstawowe różniczkowanie

1. Utwórz tensor  $x=5.0$  z `requires_grad=True`.
2. Oblicz funkcję  $y = x^3 - 4x^2 + 2x - 7$
3. Oblicz gradient funkcji  $y$  względem  $x$ .

### Zadanie 2: Gradienty dla wielowymiarowych tensorów

1. Utwórz tensor  $x = [2.0 \ 3.0]^T$  z `requires_grad=True`.
2. Oblicz funkcję  $y = x_1^2 + 3x_2^2 + 2x_1x_2$ .
3. Oblicz gradienty  $\frac{\partial y}{\partial x_1}$  i  $\frac{\partial y}{\partial x_2}$ .

W PyTorch zapisy takie jak `T` są bardziej symboliczne – nie musisz implementować ich wprost, ponieważ wektory jednowymiarowe są obsługiwane naturalnie. Jeśli jednak zachowanie wymiarów jest ważne (np. w algebrze macierzowej), możesz użyć `reshape`.

### Zadanie 3: Minimalizacja funkcji kwadratowej

1. Minimalizuj funkcję  $y = x^2$ , zaczynając od  $x = 10.0$ .
2. Wykorzystaj iteracyjne aktualizacje gradientowe:  $x = x - \eta \cdot \frac{\partial y}{\partial x}$ , gdzie  $\eta = 0.1$ .
3. Powtórz proces przez 10 iteracji.

### Zadanie 4: Zarządzanie gradientami

1. Utwórz tensor  $x=3.0$  z `requires_grad=True`.
2. Oblicz funkcję  $y = x^3$ , ale wstrzymaj śledzenie gradientów.
3. Wyświetl wynik i sprawdź, czy gradienty są dostępne.

### Zadanie 5: Obliczanie gradientów dla sieci neuronowej

1. Utwórz losowy tensor wejściowy  $x$  o wymiarach  $1 \times 3$ .
2. Utwórz prostą funkcję liniową  $y = wx + b$ , gdzie  $w = [2.0, -3.0, 1.0]$  i  $b = 5.0$ .
3. Oblicz gradienty  $\frac{\partial y}{\partial w}$  i  $\frac{\partial y}{\partial b}$ .