Tensor Mathematics Lessons

Abdulla Ali Alshamsi
ORCID:0009-0006-4905-5469

Contents

Chapter 1

Introduction

Chapter 2

T00

2.1 Lesson T00

Exercise E00 : Tensor Addition Law

Understanding the Problem Tensor addition is a fundamental operation in tensor algebra. Given two tensors $\mathbf{X}$ and $\mathbf{Y}$ of the same shape, the goal is to compute a new tensor $\mathbf{Z}$ by adding corresponding elements from $\mathbf{X}$ and $\mathbf{Y}$. This operation is performed independently on each component of the tensors.

Mathematical Formulation Let

$$\mathbf{X} = (x_1, x_2) \in \mathbb{R}^2, \quad \mathbf{Y} = (y_1, y_2) \in \mathbb{R}^2$$

The tensor addition operator

$$+ : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$$

is defined component-wise as:

$$\mathbf{Z} = \mathbf{X} + \mathbf{Y} = (x_1 + y_1,\ x_2 + y_2)$$

Mathematical Solution For numerical computation, each component of the output tensor is evaluated independently according to the rule:

$$z_i = x_i + y_i, \quad i = 1, 2$$

This rule generalizes naturally to tensors of arbitrary dimension, since tensor addition is always performed element-wise on corresponding positions.

Code Implementation

```
def E00(x, y):
    # apply tensor addition law
    return z
```

Exercise E01 : Tensor Subtraction Law

Understanding the Problem Tensor subtraction is a fundamental operation derived from tensor addition. Given two tensors $\mathbf{X}$ and $\mathbf{Y}$ of the same shape, the objective is to compute a new tensor $\mathbf{Z}$ by subtracting each element of $\mathbf{Y}$ from the corresponding element of $\mathbf{X}$. This operation is performed independently on each component of the tensors.

Mathematical Formulation Let

$$\mathbf{X} = (x_1, x_2) \in \mathbb{R}^2, \quad \mathbf{Y} = (y_1, y_2) \in \mathbb{R}^2$$

The tensor subtraction operator

$$- : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$$

is defined component-wise as:

$$\mathbf{Z} = \mathbf{X} - \mathbf{Y} = (x_1 - y_1,\ x_2 - y_2)$$

Tensor subtraction can also be interpreted using additive inverses:

$$\mathbf{X} - \mathbf{Y} = \mathbf{X} + (-\mathbf{Y})$$

Mathematical Solution For numerical computation, each component of the resulting tensor is evaluated independently according to the rule:

$$z_i = x_i - y_i, \quad i = 1, 2$$

This rule generalizes naturally to tensors of arbitrary dimension, since tensor subtraction is performed element-wise on corresponding positions.

Code Implementation

```
def E01(x, y):
    # apply tensor subtraction law
    return z
```

Exercise E02 : Tensor Hadamard Multiplication Law

Understanding the Problem Tensor Hadamard multiplication is an element-wise multiplication operation. Given two tensors $\mathbf{X}$ and $\mathbf{Y}$ of the same shape, the goal is to compute a new tensor $\mathbf{Z}$ by multiplying each element of $\mathbf{X}$ with the corresponding element of $\mathbf{Y}$. This operation is performed independently on each component of the tensors.

Mathematical Formulation Let

$$\mathbf{X} = (x_1, x_2) \in \mathbb{R}^2, \quad \mathbf{Y} = (y_1, y_2) \in \mathbb{R}^2$$

The Hadamard multiplication operator

$$\odot : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$$

is defined component-wise as:

$$\mathbf{Z} = \mathbf{X} \odot \mathbf{Y} = (x_1 y_1,\ x_2 y_2)$$

Mathematical Solution For numerical computation, each component of the resulting tensor is evaluated independently according to the rule:

$$z_i = x_i \cdot y_i, \quad i = 1, 2$$

This rule extends naturally to tensors of arbitrary dimension, since Hadamard multiplication is always applied element-wise on corresponding positions.

Code Implementation

```
def E02(x, y):
    # apply Hadamard multiplication law
    return z
```

Chapter 3

Conclusion