Kolmogorov-Arnold Representation Theorem with Haskell code examples

Tobi Lehman

May 28, 2024

Contents

1 Introduction **1**

2 Theorem Statement **1**

3 Proof **1**

4 Haskell Implementation **1**

1 Introduction

The Kolmogorov-Arnold representation theorem states that any continuous function $f : [0,1]^n \to \mathbb{R}$ of several variables can be represented as a finite sum of continuous functions ϕ_{ij} of one variable.

2 Theorem Statement

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^{n} \varphi_{ij}(x_j) \right)$$

3 Proof

4 Haskell Implementation

```
module KolmogorovArnold where
```

```
-- Example function
f :: Double -> Double -> Double
f x y = x^2 + y^2

-- Functions g_i and phi_ij for the representation
g1, g2 :: Double -> Double
g1 z = z
g2 z = z

phi1, phi2 :: Double -> Double
phi1 x = x
phi2 y = y

-- Kolmogorov-Arnold representation
karRepresentation :: Double -> Double -> Double
karRepresentation x y = g1 (phi1 x + phi2 y) + g2 (phi1 x + phi2 y)
```