1 The Radial Basis Function Kolmogorov-Arnold Network (RBF-KAN)

The RBF-KAN is a novel approach to function approximation, combining the theoretical foundations of the Kolmogorov-Arnold Theorem with the powerful approximation capabilities of radial basis functions (RBFs).

1.1 Kolmogorov-Arnold Theorem

The Kolmogorov-Arnold Theorem, also known as the Superposition Theorem or the Kolmogorov-Arnold Representation Theorem, states that any continuous multivariate function $f : [0, 1]^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}$ on the d_{in}-dimensional unit hypercube $[0, 1]^{d_{\text{in}}}$ can be represented as a superposition (composition) of a limited number of one-variable (univariate) functions g_q and $\psi_{p,q}$ such that:

$$f(\mathbf{x}) = \sum_{q=0}^{2d_{\text{in}}} g_q \left(\sum_{p=1}^{d_{\text{in}}} \psi_{p,q}(x_p) \right) \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_{d_{\text{in}}})$.

1.2 Radial Basis Functions (RBFs)

Radial basis functions (RBFs) are a class of functions that depend only on the distance (or radius) from a center point. A common choice for the RBF is the Gaussian function:

$$\phi(r) = e^{-(\beta r)^2} \tag{2}$$

where r is the Euclidean distance between the input sample and the RBF center, and β is a scaling factor.

1.3 The RBF-KAN Layer

In the RBF-KAN, the target multivariate function $f(\mathbf{x})$ is approximated using a single layer of RBF interpolation:

$$\tilde{f}(\mathbf{x}) = \sum_{j=1}^{d_{\text{in}}} \sum_{k=1}^{M} \Theta_{j,k} \phi(r_{j,k}) \tag{3}$$

where $\mathbf{x}$ is the normalized input tensor, M is the number of RBF centers, $\phi(r_{j,k})$ is the RBF value computed using the distance between the j-th input dimension and the k-th RBF center, and $\mathbf{\Theta} \in \mathbb{R}^{d_{\text{in}} \times M}$ are the learnable coefficients for the RBF interpolation.

1.3.1 Input Normalization

Let $\mathbf{X} \in \mathbb{R}^{N \times d_{\text{in}}}$ be the input tensor, where N is the batch size and d_{in} is the input dimension. The input tensor is normalized to the range $[0, 1]$ using min-max normalization:

$$\mathbf{X}_{\text{norm}} = \frac{\mathbf{X} - \mathbf{X}_{\text{min}}}{\mathbf{X}_{\text{max}} - \mathbf{X}_{\text{min}}} \tag{4}$$

where $\mathbf{X}_{\text{min}}$ and $\mathbf{X}_{\text{max}}$ are the minimum and maximum values of the input tensor, respectively.

1.3.2 RBF Computation

Define a set of M RBF centers $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_M\}$, where $\mathbf{c}_j \in \mathbb{R}^{d_{\text{in}}}$ for $j = 1, 2, \ldots, M$. These centers can be initialized randomly or using techniques like k-means clustering. Compute the RBF values for each input sample and RBF center using the Gaussian RBF:

$$\mathbf{R}_{i,j} = \phi(|\mathbf{X}_{\text{norm},i} - \mathbf{c}_j|) = e^{-\beta^2 |\mathbf{X}_{\text{norm},i} - \mathbf{c}_j|^2} \tag{5}$$

where $\mathbf{R} \in \mathbb{R}^{N \times M}$ is the tensor containing the RBF values, $\mathbf{X}_{\text{norm},i}$ is the i-th row of the normalized input tensor $\mathbf{X}_{\text{norm}}$, and $|\cdot|$ denotes the Euclidean norm.

1.3.3 RBF Interpolation

Define a set of learnable weights $\mathbf{W} \in \mathbb{R}^{M \times d_{\text{out}}}$, where d_{out} is the output dimension. Compute the RBF interpolation as a weighted sum of the RBF values:

$$\mathbf{Y} = \mathbf{R}\mathbf{W} \tag{6}$$

where $\mathbf{Y} \in \mathbb{R}^{N \times d_{\text{out}}}$ is the output tensor representing the approximation of the target function.

1.4 Training and Optimization

Train the RBF-KAN by optimizing the learnable weights $\mathbf{W}$ and the RBF centers $\mathcal{C}$ to minimize a loss function $\mathcal{L}$ between the predicted output $\mathbf{Y}$ and the true output $\mathbf{Y}_{\text{true}}$:

$$\mathcal{L}(\mathbf{W}, \mathcal{C}) = \frac{1}{N} \sum_{i=1}^{N} \ell(\mathbf{Y}_i, \mathbf{Y}_{\text{true},i}) \tag{7}$$

where ℓ is a suitable loss function (e.g., mean squared error for regression tasks, cross-entropy for classification tasks). Compute the gradients of the loss with respect to the learnable weights $\mathbf{W}$ and the RBF centers $\mathcal{C}$ using backpropagation:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \ell(\mathbf{Y}_i, \mathbf{Y}_{\text{true},i})}{\partial \mathbf{Y}_i} \cdot \mathbf{R}_i^{\top} \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial \mathcal{C}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \ell(\mathbf{Y}_i, \mathbf{Y}_{\text{true},i})}{\partial \mathbf{Y}_i} \cdot \mathbf{W}^{\top} \cdot \frac{\partial \mathbf{R}_i^{\top}}{\partial \mathcal{C}} \tag{9}$$

where $\top$ denotes the transpose operation, and $\frac{\partial \mathbf{R}_i^{\top}}{\partial \mathcal{C}}$ is the gradient of the RBF values with respect to the RBF centers, which can be computed using the chain rule and the derivative of the Gaussian RBF.