Recurrent Quantum Neural Networks - Progress Report II

Ankit Kumar, Arnab Dey, and Mangesh Singh
Third Year UG, Plaksha University
PL3001: Quantum Computing
Prof. Nitin Upadhyaya
24 March 2024

QRNN

In the innovative domain of Quantum Recurrent Neural Networks (QRNNs), the fusion of quantum mechanics and machine learning architectures opens new avenues for overcoming classical limitations like the vanishing gradient problem. This integration is primarily facilitated through a series of meticulously defined equations that leverage the unique properties of quantum states, enabling enhanced computational capabilities and addressing complex dependencies within data sequences.

1. Quantum Neuron Activation:

The fundamental aspect of QRNNs is the activation of quantum neurons, governed by the equation $\mathbf{R}(\theta) := \exp(\mathrm{i}\mathbf{Y}\theta)$ where **Y** is the Pauli-Y matrix. This equation results in a rotation within the Bloch sphere of the qubit, represented by:

$$\mathbf{R}(\theta) = \exp\left(\mathrm{i}\theta\begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}\right) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

Such rotations are crucial as they apply non-linear transformations to the quantum states—akin to the activation functions in classical neural networks. These transformations modulate the amplitude of the quantum states based on the inputs, enabling the QRNN to model complex patterns and dependencies within the data.

2. Controlled Rotation for Composite Inputs:

Further complexity is introduced through controlled rotations for composite inputs, encapsulated in the equation:

$$\mathbf{R}(\theta_0)\mathbf{cR}(1,\theta_1)\cdots\mathbf{cR}(n,\theta_n)\,|x\rangle\,|0\rangle = |x\rangle\,\big(\cos(\eta)\,|0\rangle + \sin(\eta)\,|1\rangle\big) \quad \text{for } \eta = \theta_0 + \sum_{i=1}^{n}\theta_i x_i.$$

This series of conditional rotations based on the binary string xx illustrates how QRNNs can dynamically adjust their internal states in response to input sequences. Each qubit's state directly influences subsequent transformations, thereby creating a non-linear output that is crucial for capturing and processing intricate data patterns.

3. Amplitude Cosine Transformation:

The amplitude cosine transformation, defined as follows:

$$\cos(f(\eta)) = \frac{1}{\sqrt{1+\tan(\eta)^{2\times 2^{\mathrm{ord}}}}} \quad \text{and} \quad \sin(f(\eta)) = \frac{\tan(\eta)^{2^{\mathrm{ord}}}}{\sqrt{1+\tan(\eta)^{2\times 2^{\mathrm{ord}}}}},$$

It introduces a further layer of non-linearity. Here, "ord" adjusts the steepness of the response curve, enhancing the neuron's ability to deal with nonlinear data relationships. This transformation is pivotal in QRNNs, functioning similarly to traditional neural activation functions but within a quantum computational framework.

4. General QRNN Dynamics

Moreover, the overall dynamics of QRNNs are driven by the unitary evolution $U_{QRNN}(t) = \exp(-iHt)$, where H is the Hamiltonian that governs the system. This evolution is essential for maintaining coherence and integrity of the quantum information over time, ensuring that each sequence's output reflects a coherent evolution from its predecessors.

5. Gradient Approximation via Parameter Shift:

Now coming to the training part, training these networks involves calculating gradients for optimization, a process elegantly handled by the gradient approximation via the parameter shift rule:

$$\frac{dL}{d\theta} \approx L\left(\theta + \frac{\pi}{2}\right) - L\left(\theta - \frac{\pi}{2}\right)$$

This equation facilitates the application of gradient-based learning algorithms, akin to backpropagation in classical settings, allowing for efficient tuning of parameters to minimize loss functions.

This mathematical framework underpinning QRNNs not only enhances their computational capabilities but also addresses some of the intrinsic limitations of classical neural networks. This integration of detailed quantum operations within neural networks paves the way for advancements in both theoretical and applied machine learning fields.