Quanvolutional Neural Networks

Classical Quantum Networks

Convolution Neutral Network (CNN) is a standard model for processing images in classical computers. CNN applies series of filters called kernel on images to detect different regions of an image and series of such reduced images are processed differently by applying additional layers. This process is called convolution.

Quantum Convolution

The same idea can be applied in Quantum circuits. A quantum computation, associated to a unitary U, is performed on the system. The unitary could be generated by a variational quantum circuit. The quantum convolution can be followed by further quantum layers or by classical layers. The main difference with respect to a classical convolution is that a quantum circuit can generate highly complex kernels whose computation could be, at least in principle, classically intractable.

QNN for classifying MINIST dataset.

1. **Imports**
   Import Pennylane, Tensorflow, matpoltlib libraries.

2. **Setting Hyper parameters**
   Initialize hyperparameters such as epochs, layers, training and testing split.

3. **Load Images and normalise**
   Load minist dataset using keras load_data module. Split the data into training and testing. Normalize each image by dividing the each pixel with 255. Add extra channel for convolving the image.

4. **Quantum Circuit as convolution kernel**
   Create device and initialize the quantum circuit that can run on the device. Size of each image is 2 by 2 hence we need 4 qubits to encode each image. Initialize each qubit by applying Rot gates on each Qubit and create convolve layers.
   Layers : Layers are created and initialized with random weights by using RandomLayers modules. RandomLayers module will create a random circuit.

5. **Quanvolution**
   Define a quanv function that applies 2 * 2 kernel with stride 2 on the 2 by 2 image. The process reduce the image to half of its size. The expectation values are mapped into different channels of a single output pixel.

6. **Quantum pre-processing of the dataset**
   In this step we call quanv method for each train image and test image and saves the image outputted by four channels and later used them for training.

7. **Hybrid quantum-classical model**

   After the application of the quantum convolution layer we feed the resulting features into a classical neural network that will be trained to classify the 10 different digits of the MNIST dataset. We use a very simple model: just a fully connected layer with 10 output nodes with a final softmax activation function. The model is compiled with a stochastic-gradient-descent optimizer, and a cross-entropy loss function.

8. **Training**
   We initialize the model, tarin and validate the model with the dataset which is preprocess using quantum convolution layer.

9. **Results**
   We test the accuracy and test the loss with respect to training epochs