Proposition 1. Assume that the original network \( f_0 \) is linear and \( F_{\delta} = f_{\delta+1}^{j+1} \) for \( j = 1, \ldots, N - 1 \). Then we have \( f_{\delta+1}^{j+1} = f_0(x) \) for all \( x \in X \).

4 Application to ResNet-1001

We present an application of the proposed methodology to ResNet-1001, which is a typical convolutional neural network for classification problems. ResNet-1001 description:

- \( C_1 \) is a single 3 × 3 convolutional layer.
- \( C_j^1 \) and \( C_j^2 \) is a 1 × 1 convolution with stride for \( j > 1 \).
- \( \eta_{\delta, \epsilon} \) consists of 33 residual units (RUs) which are decomposed into \( \phi_j \) having \([331 \times N]\) RUs.
- \( \Theta_j \) is constructed using \( \eta_j = \phi_j \) RUs, which can have similar coverage to the parallel subnetwork \( g_{\delta, \epsilon} \).
- \( h_\tau \) consists of global average pooling and fully connected layer.

Figure 2: Bottleneck structure of RU used in ResNet-1001.

5 Numerical results

Environment: Python, PyTorch
Machine: Intel Xeon Gold 5155, NVIDIA Titan RTX
Dataset: CIFAR-10, CIFAR-100
Training Strategy: Batchsize 128, Epoch 200, SGD with weight decay 0.0005, momentum 0.9, learning rate 0.1 which is reduced by a factor of 10 in the 80th and 120th epochs.

5.1 Performance comparison

Table 1: Error rates(%)/CIFAR-10. Preprocessing Parallel Total

<table>
<thead>
<tr>
<th>Network</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-1001</td>
<td>5.03</td>
<td>21.13</td>
</tr>
<tr>
<td>Parareal ResNet-5</td>
<td>4.51</td>
<td>21.02</td>
</tr>
<tr>
<td>Parareal ResNet-6</td>
<td>4.35</td>
<td>20.94</td>
</tr>
<tr>
<td>Parareal ResNet-12</td>
<td>4.37</td>
<td>20.77</td>
</tr>
<tr>
<td>Parareal ResNet-18</td>
<td>4.11</td>
<td>20.46</td>
</tr>
</tbody>
</table>

Table 2: Gradient calculating time for ResNet-1001 (\( N = 1 \)) and Parareal ResNet with (\( N = 3, 6, 12, 18, 24 \)). It is a measure of the time taken in one iteration for CIFAR-10 dataset input \( x \in [128 \times 128 \times 3] \) with batch size 128.

6 Conclusion

- We proposed a novel methodology to construct a parallel neural network called the parareal neural network which is suitable for parallel computation using multiple GPUs from a given feed-forward neural network.
- The coarse network that corrects differences at the interfaces among subnetworks was introduced, and it was proven both theoretically and numerically that the performance of the resulting parareal network agrees with the original network.
- To the best of our knowledge, the proposed methodology is a new kind of multi-GPU parallelism in the field of deep learning.

References