Binyamin Friedman
Michtom School of Computer Science
Brandeis University

February 15, 2024

ASSIGNMENT 1

1 MNIST Classification

An MNIST classifier was created and trained in PyTorch according to the architecture provided in the template code. The model is very simple and comprises of two convolutional layers, each followed by a 2×2 max pooling layer, a dropout layer, two dense layers, and a logarithmic softmax. ReLU is used at each stage. See Figure 1 for a visualization of the model. The model was trained using PyTorch's provided MNIST dataset, which includes 60,000 images for training and 10,000 images for testing.

Training was done in two groups of 60 epochs. For the first group, a standard learning rate of 0.002 was chosen, along with a batch size of 32 (due to memory considerations). At the end of this group of epochs, the testing set accuracy was 99.02%. For a second group of 60 epochs, the learning rate was decreased to 0.0005 and the batch size was increased to 64, in order to further finetune the model's parameters. The final testing set accuracy was 99.19%. The model was trained with 60 additional epochs at a learning rate of 0.00001 to verify that it had effectively converged. The testing set accuracy remained at ∼99.20%, while the training accuracy hovered around 99.78%. Figure 2 shows the model's test accuracy over epochs.

The code base was made to be modular. The classifier, model, training and testing routines, and program arguments were all separated into different files.

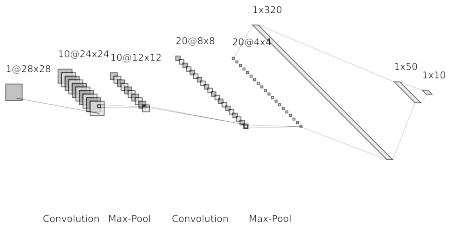

Figure 1: This MNIST classifier model architecture was provided in the assignment template.

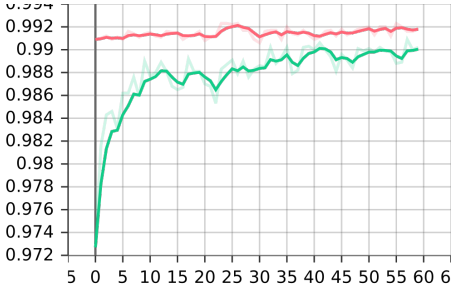

Figure 2: The green points show the first 60 epochs and the red points show the second 60 epochs. As expected, the test accuracy increases very quickly in the first few epochs but slows down afterwards.

2 Paper Reading and Implementation

2.1 Paper Reading

1. How many types of ResNet have been proposed in the paper?

The two key parameters in a ResNet are the number of layers and the type of shortcut operation. In Table 3, the paper lists ResNet-34 A, ResNet-34 B, ResNet-34 C, ResNet-50, ResNet-101, and ResNet-152. A, B, and C refer to the type of shortcut operation used: Zero-padding, projection when necessary, and projection on every layer respectively. Deeper models are also tested, including one with 1202 layers.

2. What is the fundamental difference between ResNet and VGG?

   The fundamental difference is that ResNet uses dramatically more layers, with residual information passing through shortcuts. Despite using more layers, ResNet maintains lower complexity and higher accuracy than VGG.

3. What is the input size of the images for ResNet-50?

   The input images are of size $3 \times 224 \times 224$.

4. What research question does this paper propose?

   The research question that motivates ResNet is whether a deeper model can be constructed in a way that improves performance. With standard models, increasing the number of layers leads to significantly worse accuracy, perhaps because more layers could take an exponentially longer time to converge. However, a deep model should theoretically be at least as expressive as a similarly constructed shallow model. The question is how to use more layers without dramatically increasing the effort required to converge.

5. What is the philosophy of this paper?

   The philosophy of ResNet is using residual connections to maintain good performance. They posit that this is effective because adding shortcuts between layers makes approximating the identity function trivial. If the optimal function is closer to an identify function than it is to a zero mapping, then it should be easier for a solver to find this function through a residual network. In contrast, regular networks may struggle to approximate identity functions as they get deeper. It makes intuitive sense that an optimal solution could be expressed as small changes from an identity function; especially as a network gets deeper and more expressive, the individual layers may express more subtle transformations.

6. What is the contribution of this paper to the field of deep learning and computer vision?

   ResNet is an extremely famous deep learning model, with nearly 200k citations. It has had a tremendous impact on the field of machine learning, enabling deeper and more complex models than were previously possible. Compared to older models like VGG, ResNet can manage higher accuracy with lower complexity.

2.2 Implementation

Another MNIST classifier was created and trained with PyTorch's provided ResNet50 model. This model follows the original paper with a small modification; The required stride in each bottleneck block is in the 3×3 convolution instead of the initial 1×1 convolution. This model was trained the same way as the one in Task 1, using PyTorch's provided MNIST dataset.

Training was done on the entire model, with and without PyTorch's pretrained ResNet50 weights. Each epoch took considerably more computation but less were needed in total. 30 epochs were done on the pretrained weights, 15 at a learning rate of 0.005 and 15 at a learning rate of 0.0001, with a batch size of 32 for all epochs. After the first 15 epochs, the test accuracy was at 99.40%. After 30 epochs, the test accuracy was at 99.53%. See Figure 3

Training was also done on a model without the pretrained weights. 15 epochs were done at a learning rate of 0.001 and 15 were done at a learning rate of 0.0001, with a batch size of 32 for all epochs. After the first 15 epochs, the test accuracy was at 99.09%. After 30 epochs, the test accuracy was at 99.42%. While the model without the pretrained weights took longer to reach the same points, it ultimately reached about the same accuracy. See Figure 4.

For both tests, the training accuracy reached 99.99% by epoch 30. The fact that the test accuracy did not decrease at this point suggests that ResNet50 can easily reach its full potential on the MNIST dataset. It is possible that more regularization techniques could be used to help ResNet better generalize to the test dataset, however the difference here would be very slim. Ultimately, my conclusion is that ResNet50 is NOT the right fit for the MNIST dataset, despite it's appealing high performance. The basic model reaches 99.19% with 21,840 parameters. ResNet50 achieves only slightly better results with 23,528,522 parameters, which is 1000x more complexity. A model with complexity somewhere in between could almost definitely be constructed to reach ResNet50's accuracy, without the unnecessary complexity. If a handwritten digits dataset much larger than MNIST was provided, then ResNet50's complexity would be better justified.

The code from Task 1 was only modified slightly for this task to allow for a different model depending on the program arguments.

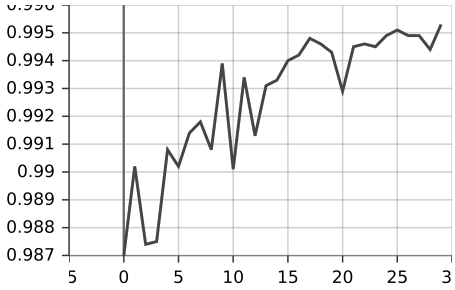

Figure 3: The testing rate accuracy increases steadily (with small fluctuations) for the first 20 epochs, and then appears to mostly level out.

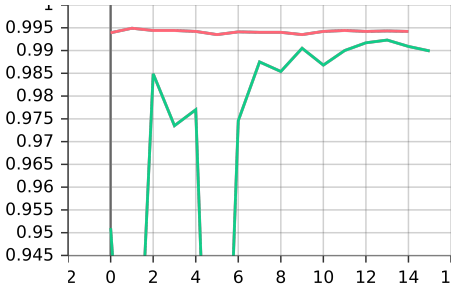

Figure 4: The green points show the first 15 epochs and the red points show the second 15 epochs. The accuracy begins quite unstable but levels out.

Submitted by Binyamin Friedman on February 15, 2024.