Application of ResNet and Other Techniques on Image Classfication Task Modified MNIST

Doreen He, Gengyi Sun, Hehuimin Cheng

Department of Computer Science McGill University

December 26, 2019

Abstract

The goal of this project is to recognize the largest numeric value out of three hand-written numbers for each image in the modified MINIST dataset. A linear model was implemented to provide a good baseline for the classification task. Then the experiments mainly focus on convolutional neural network(CNN), and specifically investigated the performance of the state-of-the-art Residual Network (ResNet). We discussed basic intuitions behind ResNet architecture, and presented some recent works on possible improvement on ResNet in last decade. A task-specific ResNet model was implemented and experimented with several factors such as the depth of network, different choices of optimizers, and the batch size. In the end, a final convolutional neural network pipeline is presented with a 97.6% accuracy on the held-out test set.

Introduction

Computer vision studies how digital images or videos are viewed and interpreted by machines, which can be applied to object recognition, event detection, video tracking, and etc.[1][2] Once, the traditional method of stimulating the computer vision used to focus on good feature engineering. Later, researchers came up with the idea of using a multilayer neural network as a combination of feature descriptors and classifiers. Staring with Yaan LeCun's Neural network in the 1990s[3] to the latest popular models like VggNet[4], Google LeNet[3], ResNet[5]. Instead of continuously digging in feature engineering, researchers shifted to investigate neural network architecture engineering. ResNet, short for Residual Network, is one of the recent year breakthroughs in machine learning community. It has now become a classic neural network structure used as a back bone for many computer vision tasks. This model allows extremely deep neural networks to be trained, by solving the problem of vanishing gradients in training very deep neural network prior to its existence. In this work, we investigated the application of ResNet on the modified MNIST classification task. The main task is to train a model that takes in an image and outputs the largest numerical value within the image, in which the single-digit number ranges from 0 to 9. Our work started with traditional approach. We experimented with two linear models (SVM and LR) with combinations of different feature descriptors (HOG and LBP), which provides us with a decent baseline for the complex ResNet model. Then we performed and compared the ResNet architectures, namely ResNet18 and ResNet34, which produced significantly higher accuracy compared to linear classifiers. To further optimize the model performance, a series of experiments are conducted, including the number of layers, batch sizes and different optimization techniques.

Related work

- Feature engineering and Linear model: Traditional computer vision approach is more focused on feature representation with the usage of linear models. Dala and Triggs proposed HOG descriptors implemented with SVM model for human detection and classification, achieving around 1% miss rate.[6] Decoste and Scholkopt successfully implemented the SVM on MNIST digit recognition task in a fast and effective way by incorporating the prior knowledge about the invariances, achieving 1.4% of error rate, approaching the 1.1% rate by LeNet4.[7] Those results give us the confidence that a linear model can achieve low test error by a good feature engineering.

- Training very deep convolutional neural networks with residual learning: Simonyan and Zisserman introduce the idea that classification accuracy is benefited by the representation depth. [4] However, training extremely deep neural networks

was difficult due to the notorious vanishing gradients problem. That is, as the gradient is back propagated to earlier layers, repeated multiplication may make the gradient extremely small. In 2015, Kaiming He proposed deep residual learning [5] for image recognition and won the 1st place in the ILSVRC and COCO 2015 competition in ImageNet Detection with top-5 error rate of 3.57%. His ResNet architecture based on the plain CNN network with shortcut connections, introducing the idea of residual block.

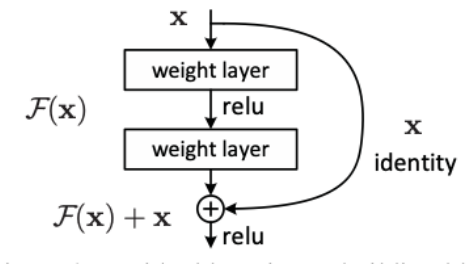

Figure 1: Residual Block: a basic building block for residual learning

The residual block turned the network into its counterpart residual version. The shortcut can handle two situations, one is that input and output have the same dimension, another is that when output's dimension increased, the shortcut can perform identity mapping or projection. Denote the input by **x**. The ideal mapping (activation function) we want to obtain by learning is: **H(x)** = **F(x)** + **x**. In the paper Deep Residual Learning for Image Recognition, the authors argue that, in practice, the residual function **F(x)** = **H(x)** - **x** is much easier to optimize than the original, unreferenced mapping **H(x)**.

- Optimization Algorithms:
Stochastic Gradient Descend (SGD) with momentum has been successfully used for decades. The momentum idea is to accelerate progress along dimensions in which gradient consistently point in the same direction. However, choosing a higher learning rate or momentum can easily lead to oscillation problems. To solve this drawback, many adaptive learning modifications to SGD algorithms has been researched in recent years. Adadelta optimizer [8] is proposed by Zeiler, M.D. in 2012. Adam optimizer [9] was introduced in 2014 by Diederik P. Kingma and Jimmy Ba. Both of the optimizers are gradient-based optimization with adaptive learning rate. The hyper-parameters of these algorithms topically require very little tuning due to their adaptive nature. In particular, Adadelta was experimented on the original MNIST dataset when it is firstly proposed. It shows promising results compared to SGD with momentum. The author achieved the optimal error rate with hyper-parameter setting rho = 0.95 and epsilon = 1e-6 on the original MNIST dataset. Based on this literature, in the experiments that investigated in this task, we adopted the same hyper-parameters for Adadelta.

	$\rho = 0.9$	$\rho = 0.95$	$\rho = 0.99$
$\epsilon = 1e^{-2}$	2.59%	2.58%	2.32%
$\epsilon = 1e^{-4}$	2.05%	1.99%	2.28%
$\epsilon = 1e^{-6}$	1.90%	**1.83%**	2.05%
$\epsilon = 1e^{-8}$	2.29%	2.13%	2.00%

Figure 2: MNIST test error rate after 6 epochs for various hyperparameter settings using Adadelta reported in the paper **Adadelta: An adaptive learning rate method**[8]

Dataset and Setup

Dataset Description

In this project, we used a modified dataset based on a classical image classification data set, MNIST dataset. The training dataset is composed of 50,000 grayscale images with dimensions of 128 and 128. Each image in the dataset contains 3 manuscript numbers, in which range from 0 to 9. In total, the dataset contains 10 classes and we want to find the largest numerical value among 3 numbers for each image. We set the split ratio to 0.8 and split the original dataset into train/validation: 40,000/10,000 for experiments.

Data Preprocessing

To prepare the data, we started with reducing the background noise. In all images, we set all grayscale pixel values to 0 if it is less than 180; that is, blackening the background and emphasising on the manuscript number. Here, we show some examples of the resulting images in Figure 3. Since the ResNet architecture takes image data with dimension of 224x224, we resized the images to fit in ResNet model. Finally, we followed the common image data practice to divide the pixel values by 255 to normalize the values into range [0, 1].

Experiments Design

In all model experiment, we trained the model with 80% of training set and validated the model with 20% of training set. All accuracy mentioned in the

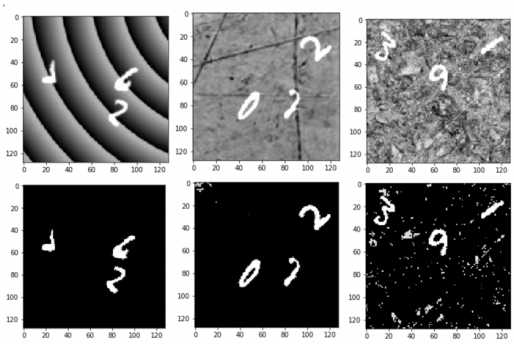

Figure 3: A comparison of some examples of the original images and the processed images

following experiments refers to the number of right predictions over the total number of data points. All loss refers to the cross-entropy loss.

Experiment 1: As discussed in the related work section, simple classification models might achieve low test error by a good feature engineering. We processed the images dataset with HOG(pixel per cell=20x20, cell per block= 7x7, orientations=9), LBP (radius= 8, number of points=8). We also implemented the experiment with no descriptor: all pixel values by 255. Then we reduced the dimension of features with Principal component analysis with components of 1000. We implemented logistic regression and support vector machine for comparison.

Experiment 2: Our model is constructed by modifying ResNet architecture to suit our requirement for the modified MNIST dataset, specifically, single-channel image classification. The first layer of PyTorch ResNet implementation was changed so that it took single channel training data.[10] The rest of the ResNet architecture was kept unchanged. To avoid training for too many iterations thus causing over-fitting, we stopped iterating when the training loss did not show the potential to decrease or validation error started to climb up. ResNet18 and ResNet34 were compared to test if more layers increase the prediction accuracy.

Experiment 3: In the context of neural networks, it can be very difficult to locate the global optima on the non-convex loss surface. Different optimizers can have different optimizing mechanism that finds various local minimum with different prediction error. Here, we proposed three optimizing approaches to train our model: SGD optimizer with momentum, Adam optimizer and Adadelta optimizer.

Experiment 4: It is often reported that a smaller batch size with more iterations tend to produce a better predictive result; but for each problem there exists a threshold after which there is a deterioration in the quality of the model. This behaviour can be observed by experimenting different batch sizes on the same mode setting. We compare the validation accuracies of models with small batch size and large batch size after the learning process saturated.

Results

Experiment 1: Based on the results shown in Table 1, the highest accuracy with tested linear models appears to be only 25.78%, despite all the feature engineering work we have done. We safely conclude that simple linear models have some predictive ability (better than random prediction), but it is very limited. Although we have discussed that there has been successful stories with good feature engineering for linear models, the chance for linear model to provide promising results on image classification task is still very little.

Model	IBP	HOG	RAW
LR ($\lambda = 0.1$)	24.07%	**25.78%**	16.69%
SVM ($\alpha = 0.1$)	20.56%	22.49%	19.50%

Table 1: The table of accuracy on validation set for different linear models with different descriptor or no descriptor.

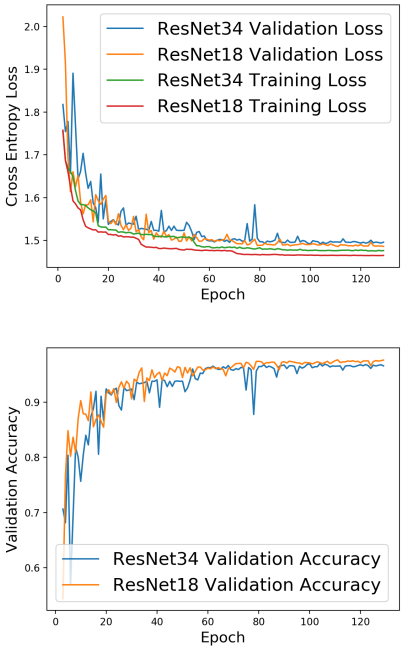

Figure 4: A comparison of ResNet18 model and ResNet34 model: The top graph shows the cross entropy loss for both models on training set and validation set. The bottom graph shows the accuracy of both models over 120 epochs.

Experiment 2: We compared the performance of ResNet18 and ResNet34. We provide both models with Adadelta optimizer and a batch size of 128. Under the completely identical pre-training process, the results for both models are shown in Figure 4. Even though the 18 layer network is just the subspace in 34 layer network, ResNet18 still outperformed the deeper layer version. After 130 epochs, ResNet18 predicts with 97.6% validation accuracy, whereas ResNet34 predicts with only 96.6%. ResNet18's better performance can also be observed in the training loss and validation loss comparison. The increased number of layers does not boost the accuracy as we expected.

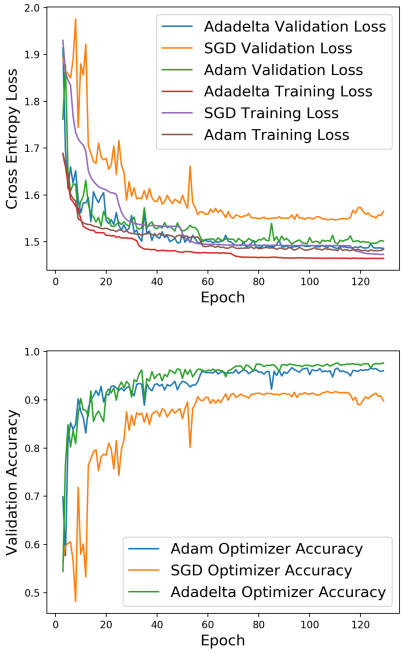

Figure 5: A comparison of different optimizers including SGD, Adam and Adadelta optimizer: the top graph shows the cross entropy loss for models and the bottom graph shows the accuracy of models on training and validation set after 120 epoches.

Experiment 3: The result of different optimizers shown in Figure 5. We observed that the model with the Adadelta optimizer with $\rho = 0.95$ and learning rate = 1e-6 achieved the highest accuracy of 97% on the validation set, on average of 7% accuracy better than the model with the SGD optimizer, and of 1% accuracy better than the Adam Optimizer. Observing the loss graph, we also found that the model with Adadelta advantages on a faster decline in both training loss and validation loss. For the stated reasons, we chose the Adadelta optimizer and continue the experiment.

Experiment 4: The size of the batch also has a great impact on the ResNet18 performance as shown in Figure 6. Observing the graph, we found that the model with a batch size of 50 performed best, achieving a loss of 1.465 on average in the last 5 epochs. We also found the model with a batch size of 128 can achieve the lowest training loss, but poor performance on the validation set. We believed that the model might experience the overfitting problem when a large batch size was set.

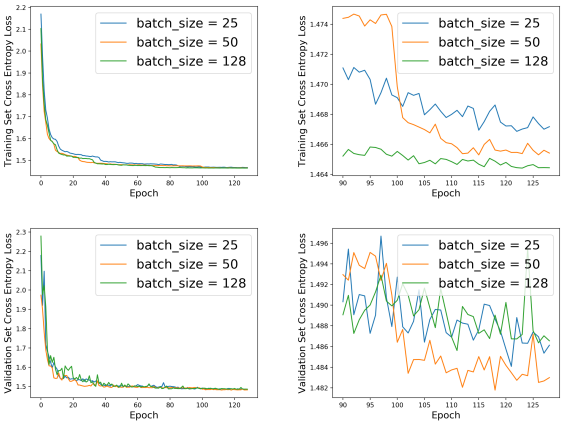

Figure 6: The top two plots correspond to training cross entropy loss. The bottom two plots correspond to validation cross entropy loss plots. The left plots illustrate the overall decreasing validation loss among 130 epochs and the right plots show the zoomed in pictures of the validation loss difference in the last 40 epochs for models with batch size of 25, 50, and 128.

Discussion and Conclusion

In this project, we applied ResNet18 architecture on the modified MNIST classification task. After a series of experiments, we utilized many current machine learning techniques and presented our learning model as a modified ResNet with a batch size of 50 and Adadelta optimizer. This model achieves 97.6% accuracy.

For image classification tasks, linear models such as SVM and LR do not perform relatively well. We conjecture that the noise and spatial correlation is too complex for the linear model to learn. To overcome the defects, we introduced the convolutional neural networks approach. With the idea of residual learning, it is possible to build and train very deep neural networks by stacking the residual blocks. However, simply seeking a deeper model does not necessarily result in a higher classification accuracy. To advantage from good neural network architecture, it is important to choose optimization methods wisely, as the choice of optimization algorithms

significantly influences the learning outcome. It has been observed in practice that using a large batch size degrades the model learning quality. Large batches tend to converge to sharp minimizers[11], therefore, lack the ability of generalization. Smaller batches help to converge to a lower loss, but only to a certain threshold. For future works, it is possible to consider deeper ResNet variants and apply dropout as a regularization technique to reduce overfitting and complex co-adaptation in training data.

References

[1] Dana H. Ballard; Christopher M. Brown. *Computer Vision*. 1982.

[2] C. Stockman Linda, G. Shapiro; George. *Computer Vision*. Prentice Hall, 2001.

[3] Yann LeCun. Lenet-5, convolutional neural networks. 2015.

[4] Zisserman A. Simonyan K. Very deep convolutonal networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015.

[5] Zhang X. Ren S. Sun J. He, K. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Triggs B. Navennet D. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'05)*, 2005.

[7] Scholkopt B. Dennis D. Training invariant support vector machines. *Machine learning*, 2002.

[8] Zeiler M. D. Adadelta: An adaptive learning rate method. *International Conference on Learning Representations (ICLR)*, 2012.

[9] Ba J. Kingma D.P. Adam: A method for stochastic optimization. 2014.

[10] Marcin Zablocki. Using resnet for mnist in pytorch tutorial. Available at `https://zablo.net/blog/post/using-resnet-for-mnist-in-pytorch-tutorial/`.

[11] Nocedal J. Smelyanskiy M. Tang P. T. P. Keskar N. S., Mudigere D. On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations (ICLR)*, 2016.