MLNS - 1 Report: Digit Addition with ResNet Models

Kyrylo Shyvam Kumar

January 16, 2025

Abstract

This report details the development and evaluation of models for predicting the sum of four digits presented in an image. We explore the use of ResNet architectures, synthetic data generation, and Exponential Moving Average (EMA) to improve model performance. Our results demonstrate that a ResNet50 model trained with EMA achieves approximately 90% accuracy on this task, significantly outperforming a baseline model and a smaller ResNet18 model. We also find that training exclusively on synthetic data can achieve around 50% accuracy on the real test set, highlighting the potential of synthetic data for this problem.

1 Introduction

The task at hand is to develop a model that can accurately predict the sum of four digits displayed in an image. The dataset consists of images similar to the MNIST dataset, but each image contains four digits arranged horizontally. This problem, while seemingly simple, presents challenges related to learning the non-trivial operation of addition and dealing with a limited amount of training data (only 10,000 examples).

This report outlines our approach to solving this problem, focusing on the use of ResNet models, the generation of synthetic training data, and the application of Exponential Moving Average (EMA) to stabilize training. We analyze the performance of different model configurations and discuss the impact of these techniques on the final results.

2 Model

We experimented with two primary model configurations:

1. **ResNet for Direct Sum Prediction:** A ResNet50/ResNet101 model was trained to directly predict the sum of the four digits (ranging from 0 to 36). In this setup, the ResNet model processes the entire image, and

its output layer consists of 37 units (one for each possible sum) with a softmax activation function to produce a probability distribution over the possible sums.

2. **ResNet with Digit Prediction Heads:** A ResNet model (either ResNet50 or ResNet101) was used as a feature extractor, followed by four separate "prediction heads" to predict each digit individually. Each prediction head is a fully connected layer with 10 output units (representing digits 0-9) and a softmax activation. The final sum is then calculated by summing the individual digit predictions. The outputs of the Resnet are connected to the prediction heads. The activation function used is Softmax.

3 Extra Additions

3.1 Synthetic Data

Due to the limited size of the real dataset (10,000 examples), we generated a synthetic dataset to augment the training data and improve model performance.
Synthetic Data Generation Process:

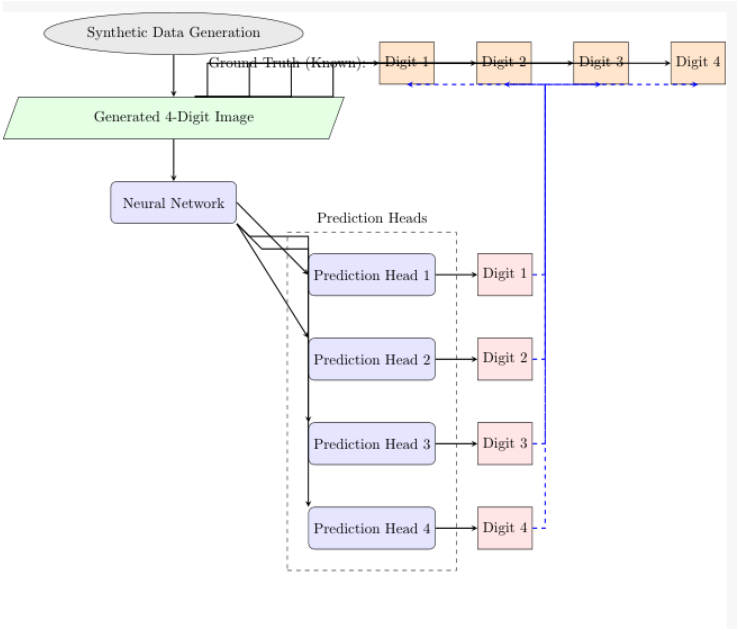

Figure 1: Synthetic data usage and model architecture with prediction heads. The input image is processed by a ResNet feature extractor. The extracted features are then fed into four separate prediction heads, each responsible for predicting one of the four digits. The Ground Truth for each digit is known during the training on synthetic dataset.

1. Four random digits (0-9) were selected.
2. Corresponding digit images were retrieved from the MNIST dataset (or a similar source). If needed, I will provide details on how it was done.
3. These digit images were horizontally concatenated to create a new 4-digit image.
4. Transformations such as random rotations, slight scaling, and noise addition were applied to increase variability and potentially improve robustness.
5. 100,000 synthetic samples were created using this method.

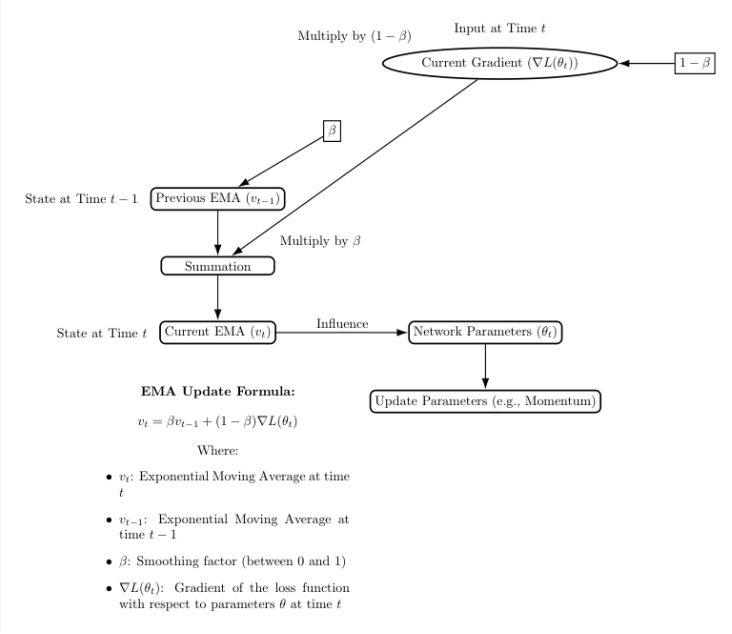

Figure 2: Effect of EMA on training stability. EMA helps to smooth out the fluctuations in the model's weights during training, leading to more stable convergence.

3.2 Exponential Moving Average (EMA)

Initial experiments revealed that training was highly unstable. To address this, we incorporated Exponential Moving Average (EMA) to stabilize the training process. EMA maintains a moving average of the model's weights during training, which helps to smooth out fluctuations and improve convergence.

The EMA update formula is:

$$v_t = \beta v_{t-1} + (1 - \beta)\theta_t \tag{1}$$

where:

* v_t is the EMA of the weights at time step t. * v_{t-1} is the EMA of the weights at the previous time step. * θ_t is the current model weight at time step t. * β is the smoothing factor (a hyperparameter between 0 and 1, set to 0.99 in our experiments).

EMA helps to stabilize training, likely because it reduces the impact of noisy gradients, especially in the presence of a relatively small dataset or when using a larger learning rate.

4 Results

Table 1 summarizes the performance of different model configurations on the test set:

Table 1: Model Performance Comparison

Model	**Test Accuracy**
Baseline (4-layer ConvNet)	∼10%
ResNet50 (trained from scratch)	∼30%
ResNet50 with EMA	∼90%
ResNet50/101 (synthetic data only)	∼50%

Key Observations:

- The baseline model, a 4-layer convolutional network (Conv2D, MaxPooling2D, Conv2D, MaxPooling2D, Flatten, Dense, Dense) with ReLU activations and a final softmax layer, achieved only around 10% accuracy, highlighting the difficulty of the task with a simple model. The learning rate was 0.0001. Optimizer - Adam.
- ResNet50 trained from scratch on the real dataset reached approximately 30% accuracy.
- Incorporating EMA and pre-training on ImageNet during the training of ResNet50 significantly improved performance, boosting accuracy to around 90%. The model was trained for 50 epochs.
- Training ResNet50/101 exclusively on the synthetic dataset achieved approximately 50% accuracy on the real test set, demonstrating the effectiveness of our synthetic data generation process. The model was trained for 150 epochs. The synthetic dataset had 100,000 samples.

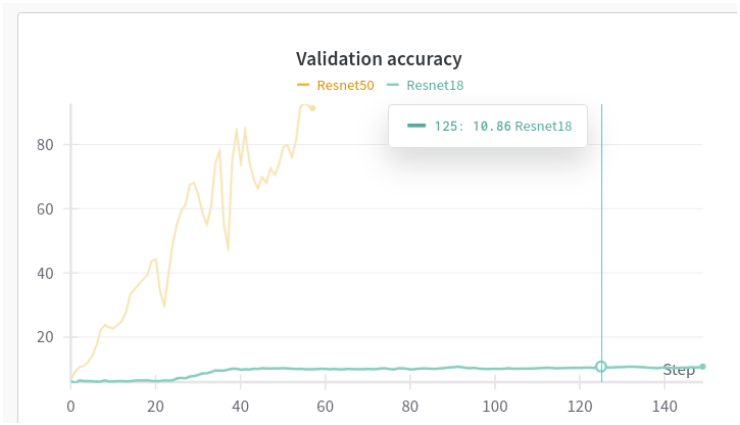

Figure 3: Comparison of different model sizes (ResNet18, ResNet50). Validation accuracy is plotted against the training step. The y-axis represents the validation accuracy.

5 Observations

5.1 Model Complexity

Experiments with a smaller ResNet18 model (using the same hyperparameters as ResNet50) showed that it was unable to learn effectively, achieving very low accuracy. This suggests that the task requires a model with sufficient capacity to capture the complexities of digit addition. The 4-layer ConvNet, despite being adequate for simpler tasks like MNIST digit classification, proved insufficient for this more challenging problem. The non-trivial nature of addition, which involves carrying over values between digits, likely necessitates a deeper and more complex model architecture.

5.2 Training Instability

Training, even with small learning rates (e.g., 0.001, 0.0001), was found to be highly unstable. The use of EMA significantly mitigated this instability, as illustrated in Figure 4. The validation accuracy before using EMA fluctuated wildly, while after applying EMA, the validation accuracy increased smoothly and reached a much higher value.

5.3 Synthetic Data Effectiveness

The ability of the model trained solely on synthetic data to achieve 50% accuracy on the real test set indicates that the synthetic data distribution, while potentially not perfectly aligned with the real data distribution, captures essential features relevant to the task. By matching key statistics and incorporating

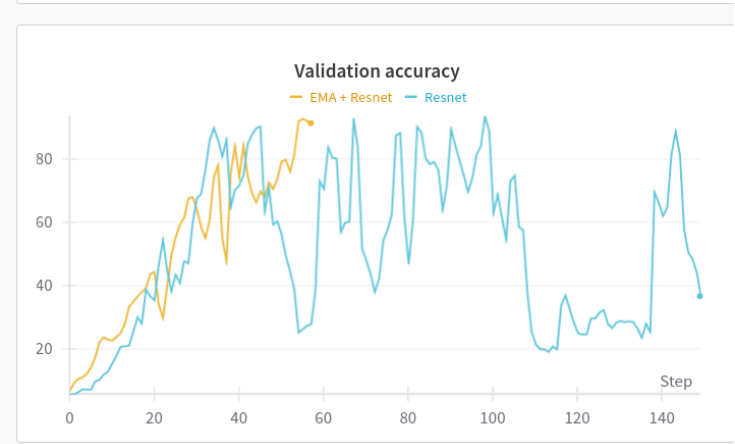

Figure 4: Unstable training behavior. Validation accuracy is plotted against the training step, illustrating the impact of EMA on stabilizing training. The y-axis represents the validation accuracy.

transformations, we aimed to make the synthetic data as representative as possible.

5.4 Predicting Individual Digits

We hypothesize that predicting each digit individually (using prediction heads) and then summing the predictions is an easier task than directly predicting the sum. This is because the constraints imposed by predicting individual digits (each output must be a digit between 0 and 9) can simplify the learning process. However, this approach relies heavily on high quality of synthetic data.

5.5 Pretraining on ImageNet

Surprisingly, using a ResNet model pretrained on ImageNetV2 did not improve performance compared to ImageNetV1 despite having better accuracy on ImageNet. In fact, the pretrained model performed worse, possibly due to overfitting to the ImageNet dataset, which is significantly different from our digit dataset. Pre-trained models were significantly better than those trained from scratch.

5.6 Absence of Overfitting in Large Models

Even when trained on the synthetic dataset to 100% training accuracy and a very low loss (e.g., 1e-5), the ResNet50/101 model continued to improve on the validation set. This suggests that large models, in this particular case, are not prone to overfitting, even with a limited amount of real training data. However, there is a significant discrepancy between the validation accuracy and

the test accuracy for the model trained on synthetic data. This could be due to differences in the distributions of the validation and test sets or potential issues in the evaluation process. Also, as training logs only show up to epoch 22, the overfitting might occur later.

6 Hyperparameters

The following hyperparameters were used in our experiments:

- **Learning Rate (Lr):** 0.001 (for the main experiments with ResNet50 and ResNet101)
- **Weight Decay:** 0.0001 (L2 regularization was used to reduce overfitting)
- **EMA Beta Coefficient:** 0.99
- **Optimizer:** Adam

7 Future Work

Several avenues for future research can be explored:

- **Combining Synthetic and Real Data:** One promising direction is to combine the losses from the synthetic and real datasets during training. This could involve a weighted sum of the two losses, where the weights are adjusted to balance the influence of each dataset.
- **Experiment with Prediction Head Architectures:** Different architectures for the prediction heads could be investigated. For example, adding more layers or using different activation functions might improve performance.
- **Advanced Data Augmentation:** Exploring more sophisticated data augmentation techniques, such as affine transformations, elastic deformations, or generative adversarial networks (GANs) to generate synthetic data, could further enhance model robustness and generalization.
- **Alternative Optimization and Learning Rate Schedules:** Trying different optimization algorithms (e.g., SGD with momentum) or learning rate schedules (e.g., cyclical learning rates) might lead to faster convergence or better performance.
- **Error Analysis:** A thorough analysis of the errors made by the model could provide insights into its weaknesses and guide further improvements. For example, examining images where the model makes incorrect predictions might reveal patterns or biases that can be addressed.

8 Conclusion

This report presented our investigation into using ResNet models for the task of predicting the sum of four digits in an image. Our key findings include the effectiveness of synthetic data in augmenting a small real dataset, the importance of using EMA to stabilize training, and the ability of a sufficiently large ResNet model to achieve high accuracy on this task. While further research is needed to optimize the approach, our results demonstrate the potential of deep learning models for solving this type of problem. The significant difference between the performance on synthetic and real data indicates a need for further investigation into the synthetic data generation process and the evaluation methodology.