Exact Loss Convergence in Convolutional Networks

Stochastic Gradient Descent on Separable Data: Exact Convergence with a Fixed Learning Rate (Nacson, et. al. 2018) provides theoretical guarantees for loss convergence using SGD with constant step size and no iterate averaging on linearly separable data. The authors prove that if linear separability and smooth-monotone loss are guaranteed, the norm of the weight vector w diverges to infinity, loss and gradients converge to zero and the direction of the weight vector converges to the maximum margin separator. The original interest in the study of convergence for SGD with constant step size was sparked by earlier experiments (repeated in Soudry 2018b). These experiments demonstrated that training loss can converge to zero in Convolutional Networks using SGD with constant step size and no iterate averaging. This contradicted theoretical predictions for SGD convergence in Neural Networks since in cases when partial strong convexity or the PL condition do not hold, constant step size SGD is only guaranteed to converge to loss below some constant value, proportional to the learning rate. The authors of Nacson et al. saw the gap between theoretical and empirical results and proposed new guarantees for SGD convergence.

However, the original experiments made several assumptions which were not discussed in Nacson et al. The network used in the experiments was a ResNet-18 Convolutional Neural Network. This network is a medium-sized CNN featuring 18 layers composing 8 convolutional blocks and a fully connected layer. Since the main assumption for absolute convergence of last-iterate SGD is linear separability, the large number of parameters may have allowed the network to construct a complex decision boundary that would not be possible for a smaller CNN. ResNets also use skip connections for identity mappings between the input of a residual block and its output. These mappings make it unclear whether the results can be generalized to other Convolutional Networks which do not feature skip connections. Since CNNs are known to suffer from the the degradation problem and vanishing gradient, it is unclear whether convolutional models outside of the ResNet class are capable of the same convergence shown in the experiments. Therefore, the groundbreaking results could apply to a much more narrow class of networks than would appear initially.

This paper seeks to bridge the uncertainty left by the original experiments

by training three Convolutional Networks that follow earlier architectures: LeNet-5 (approximately), AlexNet and VGG-16. These models are in order of increasing size and depth – the base model implementing LeNet-5 has 121,000 parameters while VGG-16 has 37 million. The models are composed of convolutional blocks followed by dense layers. The base model features only 2 convolutional and 2 dense layers while VGG-16 is has 13 convolutional and 3 dense layers. The three models considered do not implement skip connections and are therefore vulnerable to the issues mentioned earlier. We are able to recreate the convergence dynamics from the earlier experiments with all three networks and demonstrate that SGD with no iterate averaging or a decreasing learning rate schedule can be used to achieve exact training loss convergence in CNNs of varying sizes and with no residual mappings.

Methods

The models are trained on the CIFAR-10 dataset - same as used by the original experiments. The training dataset consists of 50,000 images labelled with 10 classes - types of animals and machinery. The validation dataset features 10,000 images and is delivered independently, eliminating the possibility of artificially inflating validation performance by favorable splitting of the data. All images are of size (32, 32, 3) - 32 x 32 pixels and 3 color channels. The images are resized using bilinear interpolation to fit into the input space of AlexNet and VGG-16 which work on 224 x 224 data. All models were trained locally using an NVIDIA RTX 3070 Laptop GPU, Ubuntu Linux, Tensorflow and Keras. Softmax outputs and cross-entropy loss was used in all models. SGD with constant learning rate of 0.01 and no iterate averaging was used in all experiments. The choice of step size comes from the most pessimistic bound on the learning rate given in Nacston et al. as $1/B$. For AlexNet and VGG-16, dense layers of size 10 were added to resize the typical output space of 1000 classes to only 10. For each model, a plot is produced where the left-most plot shows the training loss and error on the logarithmic scale and the middle and right-most plots show training and validation loss and training, validation error respectively. These visualizations allow us to compare the performance of the models and the original experiments.

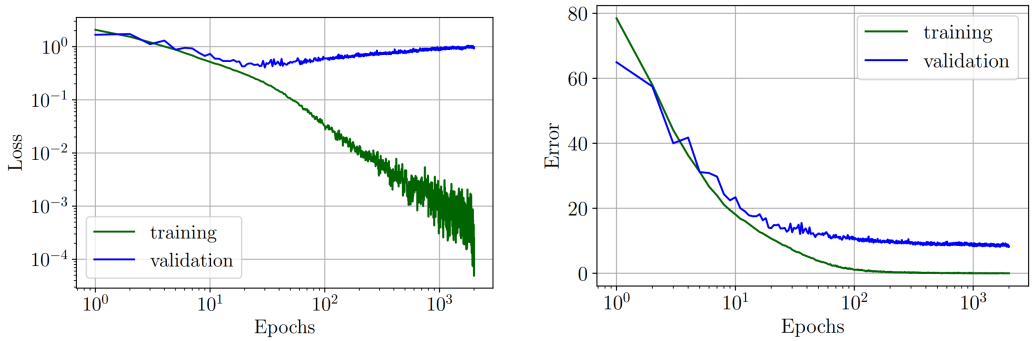


Figure 1: Original Experiments

Experiments

Figure 1 shows the experiments cited in Nacson et al. We can observe that (1) The training loss and (classification) error both decays to zero; (2) after a while, the validation loss starts to increase; and (3) in contrast, the validation (classification) error slowly improves. We achieve similar results for the three considered networks. While the validation loss and error are significantly higher in our experiments, the importance to theoretical results is in the absolute convergence of the training loss and error (all training samples are clssified correctly after sufficient iterations). We are able to recreate these results.

Base Model

The base model provides a reference for convergence of further models. It consists of two convolutional blocks – a convolution with 32 4x4 filters followed by Max. pooling – and two dense layers. The architecture is shown in figure 2. In total, the model has 121,802 parameters and only uses 475.79 KB of space.

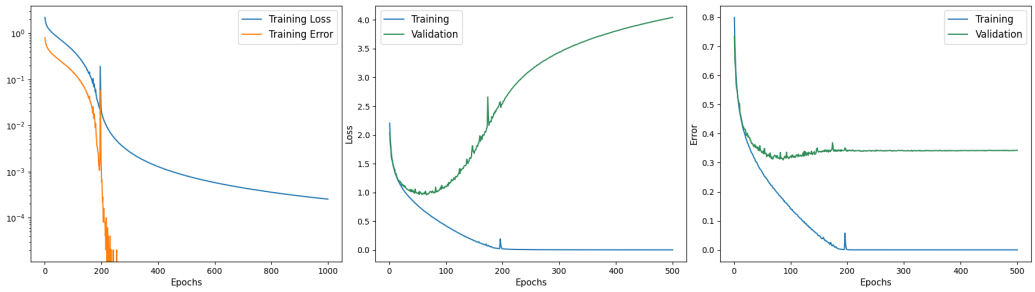


Figure 3: Base Model Training History

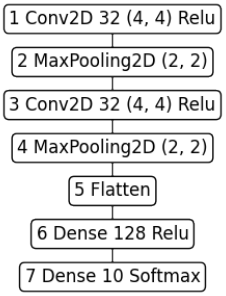


Figure 2: Base Model Architecture

The model is trained for 1000 epochs. Figure 3 shows the training/validation loss and error over the course of training. We can observe that training loss converges slowly with value 2.50×10^{-4} by the 1000th epoch. The training error converges much faster with all training examples classified correctly by the 253rd epoch. From the shape of the training loss plot on the log-scale, we can reason that the loss will likely continue to decrease with more iterations (the slope is non-zero, negative and decreasing slowly). This hints at exact convergence like shown in the original experiments. The validation loss decreases until 80th epoch and then begins to grow due to extreme overfitting. Validation error improves slowly until stabilizing at 34%. These results mirror the convergence dynamics shown in the original experiments. Training loss continues to improve after a large number of iterations, training error reaches 0, validation loss increases while validation error stabilizes.These trends indicate possible exact training loss convergence as epochs tend to infinity.

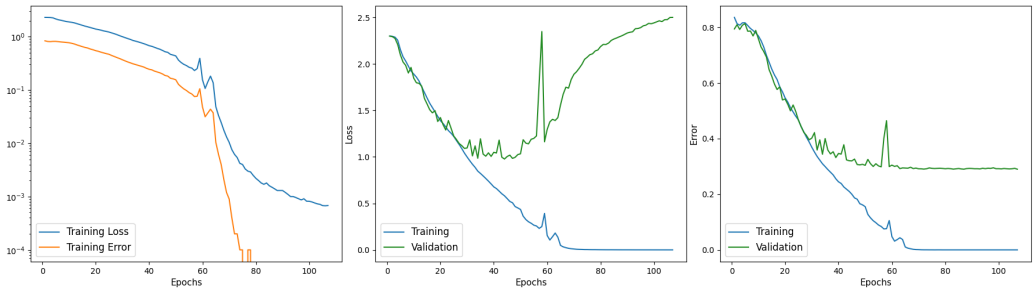


Figure 4: AlexNet Training History

AlexNet

AlexNet is a much larger model than LeNet-5. Our implementation totals 21,622,154 parameters and uses 82.48 MB. The model consists of 5 convolutional blocks followed by 3 dense layers (2 of size 4096 and output layer of 10 neurons). This model was one of the first to introduce ReLU activation which is used in both convolutional and hidden dense layers. A much higher number of filters is used compared to the base model; both the size and depth of this network allow for more sophisticated parameterization. A diagram of the full architecture of this model is given in the appendix.

The training history for AlexNet is given in figure 4. We can see similar convergence shape to the base model and the original experiments. Training loss converges much faster than the base model, achieving 6.7910×10^{-4} by the 100th epoch. Similarly, the loss training is low and has a strong and slowly-decreasing negative slope, indicating likely continued convergence with further epochs. The training error reaches 0 at epoch 78 meaning that all training examples are correctly classified. Validation loss reaches a minimum at 50 epochs and increases; validation error stabilizes at 29%. We can see the same shape for the metrics as in the earlier examples. Small and decreasing training loss and 0 training error show likely exact convergence in the asymptotic behavior.

VGG-16

The last model considered is VGG-16. This is the largest model trained at 37,704,258 parameters and 143.83 MB. The model consists of 5 convolutional

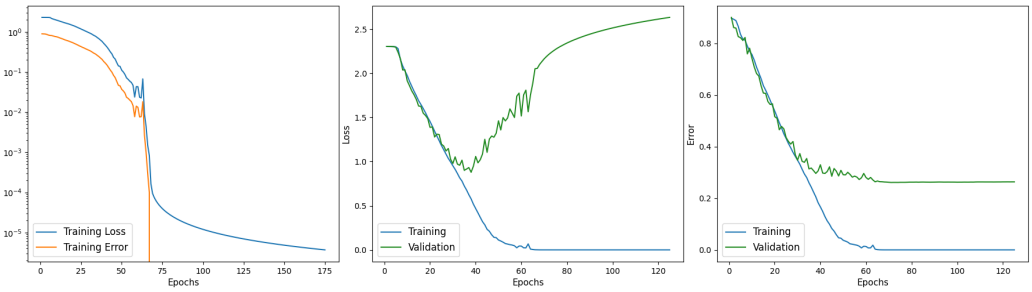
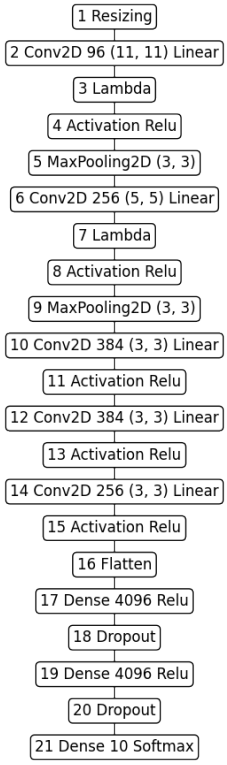


Figure 5: VGG-16 Training History

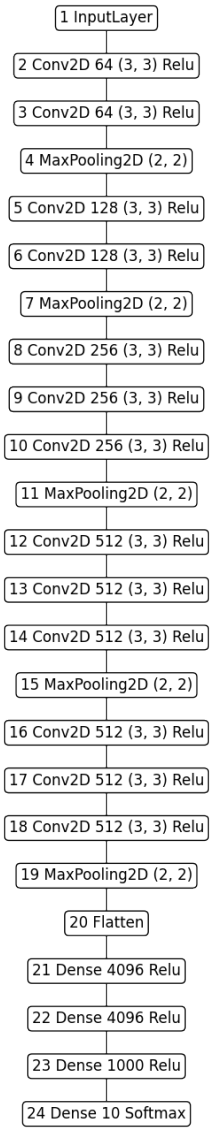
blocks, each with 2 or 3 convolutional layers and a larger number of filters, and 4 dense layers (4th added for output size reduction). Full architecture is shown in the appendix. This model is the largest and deepest considered.

Figure 5 demonstrates the training history of VGG-16. Training loss has the same shape as the earlier models but converges to a much lower value of 3.6917×10^{-6}. Training error becomes 0 after 67 epochs, indicating that the model was able to classify all the training samples. Validation loss and error have the same shapes as the other models, with the error stabilizing at 26%. We can once again observe a training loss that is very small and still decreasing. This likely indicates exact convergence with sufficient iterations.

All three models considered were able to classify all training examples after sufficient iterations. All models achieved low values of training loss with shape that indicates continued convergence (large negative first derivative). All models mirrored the validation loss and error shapes of the original experiments although with higher values for both. Since we were able to observe convergence to training loss of the order 10^{-4} for the base model and AlexNet and 10^{-6} for VGG-16, we can cautiously conclude that the original results for exact loss convergence in CNNs can be extended to smaller models and models with no residual connections. This strengthens the Nacson et al. and Soudry 2018b. papers by confirming the relevance for the study of SGD with constant step size as an optimization method for Convolutional Networks. In our experiments, we did not observe signs of model degradation or vanishing gradient, indicating that the optimizer was able to propagate through the model without skip connections. These results indicate that decreasing stepsize or iterate averaging may not be necessary for training NNs.



(a)



(b)

Figure 6: a) AlexNet, b) VGG-16

Bibliography

Mor Shpigel Nacson and Nathan Srebro and Daniel Soudry. Stochastic *Gradient Descent on Separable Data: Exact Convergence with a Fixed Learning Rate*, 2018

Daniel Soudry and Elad Hoffer and Mor Shpigel Nacson and Suriya Gunasekar and Nathan Srebro. *The Implicit Bias of Gradient Descent on Separable Data*, 2018