551 MiniProject3

Fynn Schmitt-Ulms, Joon Hwan Hong, and Daniel Korsunsky

April 1, 2021

Abstract

In this project, we implemented a multilayer perceptron (MLP) from scratch and investigated its performance on classifying image data, in particular the MNIST dataset of handwritten digits [1]. We found that a linear MLP outperformed both one and two hidden layer MLPs with 128 units and ReLU activations. Moreover, the two-hidden-layer MLPs with sigmoid and tanh activations vastly outperformed their ReLU counterpart. Adding L2 regularization to the ReLU model also improved accuracy, while training the model with unnormalized images drastically lowered performance. A three hidden layer sigmoid (logistic) MLP was our highest performing model, with an accuracy of 0.9520. Finally, we implemented a two-layer Leaky ReLU model, which outperformed the regular ReLU model.

1 Introduction

The MNIST dataset used in this analysis has often been used in the past to develop machine learning methodologies: for example, Meier et al. (2011) used the dataset to train the members of a committee of one-hidden-layer neural nets, achieving an recognition error rate of 0.39% [2]. Our work uses the MNIST dataset to test the accuracy of our MLP with no hidden layers as well as with one or two hidden layers each having 128 units and ReLU activations (Exp. 1), of the two-layer model with sigmoid and tanh activations (Exp. 2), and of the two-layer ReLU model with L2 regularization (Exp. 3) and trained with unnormalized images (Exp. 4).

The multilayer perceptron was implemented as a class with `fit`, `predict`, and `gradient` functions. In addition, an `evaluate_acc` function was defined to evaluate model accuracy. Six activation functions were also implemented: `linear`, `ReLU`, `sigmoid (logistic)`, `tanh`, `Leaky ReLU`, and `softmax`. We also incorporated automatic learning rate adjustment and an Adam optimizer, which smooths out gradient updates and does adaptive parameters for each weight.

After the models and activation functions were implemented, we used the MNIST dataset to run 5 experiments on them. In the first, the linear model outperformed the one and two-layer ReLU models (0.8726 vs. 0.8319 vs. 0.7250). In the second, we tested the two-layer model with sigmoid (logistic) and tanh activations, both of which outperformed the ReLU model (0.9503 vs. 0.9331 vs. 0.7250, respectively). In Experiment 3, we added L2 regularization to the two-layer ReLU model, improving accuracy from 0.7250 to 0.8198. In Experiment 4, we trained the two-layer ReLU model using unnormalized image data, which lowered accuracy from 0.7250 to 0.2225. Finally, in Experiment 5, we substituted the ReLU activations in the two-layer MLP with Leaky ReLU, raising accuracy to 0.8011 (by 0.0761). We also maximized the accuracy of our best performing model, the two-layer sigmoid, by adding a third hidden layer, slightly improving accuracy from 0.9503 to 0.9520.

2 Dataset

The MNIST dataset consists of handwritten digits (0 - 9) sampled from high school students and employees of the United States Census Bureau [1]. The images are 28 x 28 pixels. We used the dataset's default partition of 60,000 training images and 10,000 testing images. Once loaded from TensorFlow, we randomized, vectorized, and normalized the training and testing sets. Finally, rather than manually designing the features ourselves, like we have done in previous projects, our neural net models were trained to learn the feature extractor.

3 Results

Table 1: Experiment Accuracy Results

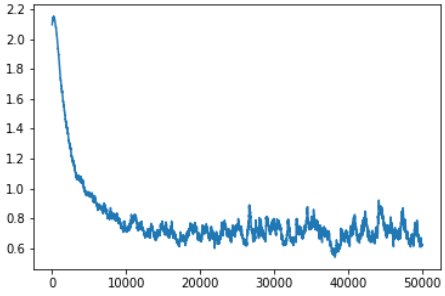
Activation Function and Other Elements (Exp. #)	No Hidden Layers	One Hidden Layer (128)	Two Hidden Layers (128, 128)	Three Hidden Layers (256, 128, 128)
Linear (1)	0.8726	—	—	—
ReLU (1)	—	0.8319	0.7250	—
Sigmoid [Logistic] (2, 5)	—	—	0.9503	0.9520
Tanh (2)	—	—	0.9331	—
L2 Regularization (3)	—	—	0.8198 (ReLU)	—
Unnormalized Images (4)	—	—	0.2225 (ReLU)	—
Leaky ReLU (5)	—	—	0.8011	—

3.1 (Non)linearity and Network Depth

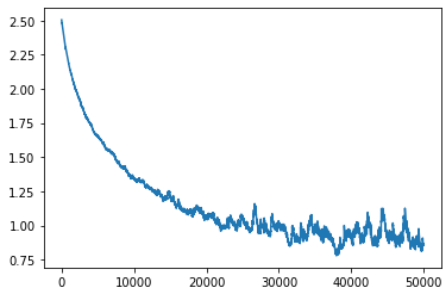
Our linear MLP model, which had no hidden layers and directly mapped inputs to outputs, was given an initial learning rate of `3e-4`, while the models with one or two hidden layers each having 128 units and ReLU activations were given an initial learning rate of `3e-5`. The linear model accuracy outperformed both that of the one and two hidden layer models (0.8726 compared to 0.8319 and 0.7250, respectively). This is unsurprising since when using ReLUs, neurons can die more easily and are difficult to repair with regular gradient descent.

Figure 1: Experiment 1 models' loss value over iterations

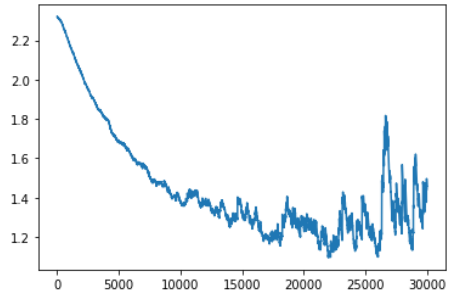
(a) no hidden layers



(b) a single hidden layer having 128 units with ReLU activations



(c) 2 hidden layers each having 128 units with ReLU activations



3.2 Sigmoid and Tanh Activations

In this experiment, we copied the two-hidden-layer model from Experiment 1 and changed its activations to sigmoid and tanh, each with an initial learning rate of `1e-3`. The sigmoid and tanh model accuracies vastly outperformed that of the ReLU model (0.9503 and 0.9331 compared to 0.7250). These activations were better because unlike the ReLU activations, they don't die since their gradients are never 0. With this said, they still have their drawbacks, since the gradients for both sigmoid and tanh activations can get very small for both of them in certain cases.

3.3 L2 Regularization

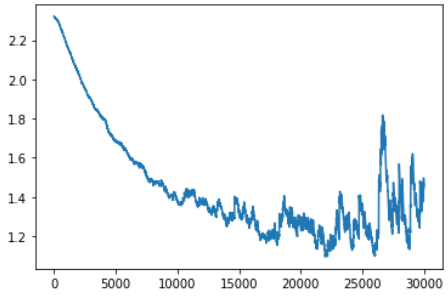
In this experiment, we again copied the two-hidden-layer models from Experiment 1 and added L2 regularization (weight decay) to the cost (λ = `1e-4`). The L2 model accuracy outperformed the regular ReLU model's (0.8198 compared to 0.7250). The L2 model was more accurate because the ridge regression kept the weights from getting too large (unless they really needed to be), which prevents the model from overfitting.

3.4 Unnormalized Images

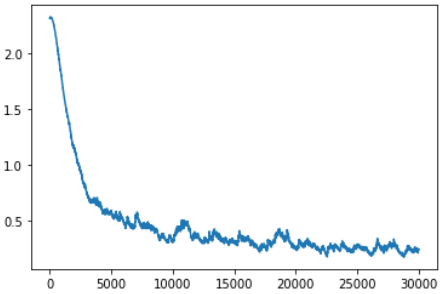
In this experiment, we copied the the two-hidden-layer model from Experiment 1 (with a learning rate of `3e-6`) but trained it with unnormalized images. Unsurprisingly, this significantly reduced the model's accuracy, bringing it down

Figure 2: Experiment 2 models' loss value over iterations

(a) Reference from *Figure 1 (c)*



(b) Sigmoid activation function



(c) tanh activation function

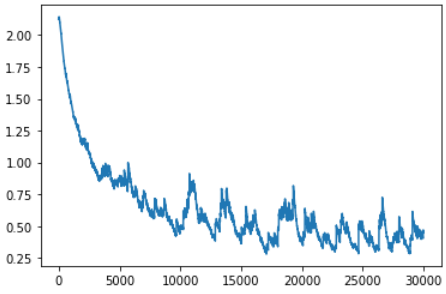
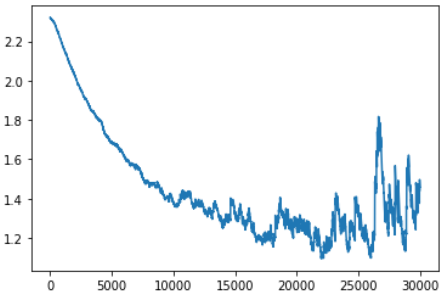
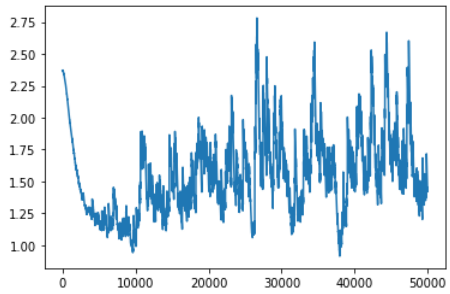


Figure 3: Experiment 3 models' loss value over iterations

(a) Reference from *Figure 1 (c)*



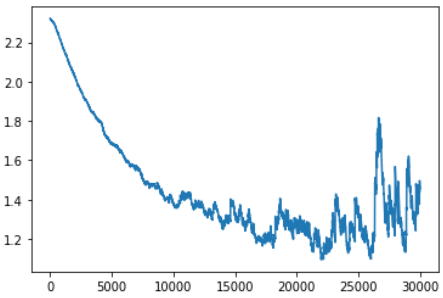
(b) With L2 regularization (weight decay)



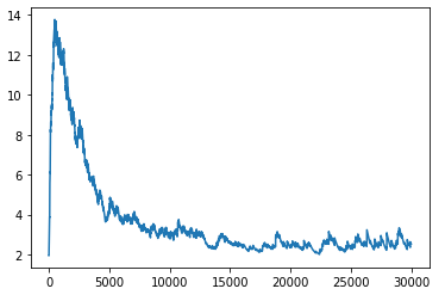
from 0.7250 to 0.2225.

Figure 4: Experiment 4 models' loss value over iterations

(a) Reference from *Figure 1 (c)*



(b) Trained with unnormalized images



3.5 Additional Explorations

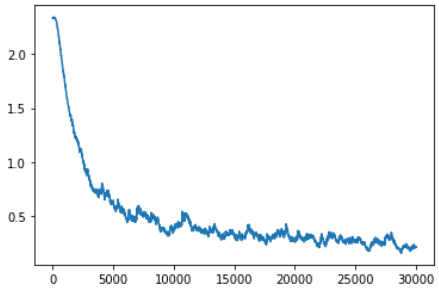
For each MLP model, we employed the Adam optimizer because of its usefulness for noisy gradients: the optimizer uses the recent "average" gradient, which smooths out its updates. It also does adaptive parameters for every weight: the amount each parameter is updated depends on how much that parameter has been updated in the past. We also implemented automatic adjustment of the models' learning rates which is triggered by drastic increases in the loss values.

We also created two models to evaluate the performance of other activation functions. As you can see in the results table, substituting the ReLU activations in the two-layer model with a Leaky ReLU improved accuracy from 0.7250 to 0.8011. Because Leaky ReLU allows small negative values when input is less than zero, we can avoid neuron death

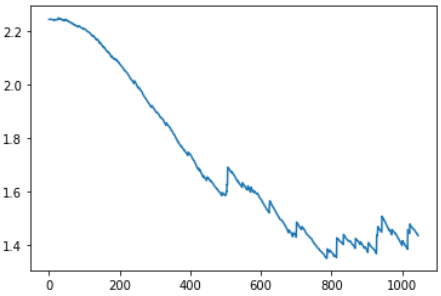
from inactivity, thus improving performance. Additionally, we implemented a three-layer (256, 128, 128) MLP with sigmoid (logistic) activations to evaluate it against the two-layer sigmoid MLP, which was our best-performing model. Indeed, the three-layer model had slightly higher accuracy (0.9520 vs. 0.9503).

Figure 5: Experiment 5 models' loss value over iterations

(a) 3 hidden layers, 256, 128, 128 units, Logistic



(b) 2 hidden layers, 128 units, Leaky ReLU



4 Discussion and Conclusion

Our best performing model was the three-layer MLP with sigmoid activations, achieving an accuracy of 0.9520. Among all models with normalized image data, the two-layer ReLU model, which was used as a base comparison against our other models, performed the most poorly, with an accuracy of 0.7250. When using ReLUs, neurons don't fire with negative inputs and therefore die more easily, which leads to poorer performance than the one-layer ReLU and even the linear model. Because tanh and sigmoid gradients are never zero, they do not die, which would explain their higher accuracies. Although not as much as the other activation functions, Leaky ReLU also improved performance over the standard ReLU because the neurons could still fire with small negative values.

While our top-performing models achieved good results in correctly identifying numbers in the MNIST dataset, handwritten digits more often appear on paper alongside other handwritten text. Thus, possible directions for future investigation could entail expanding our model's functionality by training it to recognize multiple digits (or even letters) appearing in one image. In a similar vein, since scanned handwriting can be much less easily discernible or pre-processed than the relatively clear MNIST images, it would be useful to examine how the models would perform on sample data derived from image augmentation. Finally, we could employ techniques like boosting to reduce variance and make our models more robust.

5 Statement of Contributions

Contributions by Schmitt-Ulms: Task 1, Task 2, report editing.
Contributions by Hong: Task 3, LaTeX figures, report editing.
Contributions by Korsunsky: Report writing.

References

[1] LeCun, Y. & Cortes, C. (2010). MNIST handwritten digit database.

[2] Meier, Ueli & Ciresan, Dan & Gambardella, Luca Maria & Schmidhuber, Juergen. (2011). Better Digit Recognition with a Committee of Simple Neural Nets. 1250 - 1254.