Semi-Supervised Learning: A Comparative Study

Richmond Azumah
002710785
azumahrichmond6@gmail.com

Abstract

In this paper, we explore semi-supervised learning techniques using the MNIST digits dataset and a custom Two-Moon toy dataset. We train a 2-layer neural network with Rectified Linear Unit (ReLU) activations for MNIST and a 2-layer network with sigmoid activations for the Two-Moon dataset. We compare baseline supervised learning with three semi-supervised learning algorithms: Entropy Minimization, Pseudo Label, and Virtual Adversarial Training. Additionally, we propose a novel Pseudo Label technique. Our experiments show the effectiveness of these methods in utilizing unlabeled data for improved classification accuracy. This paper also describes how a Back-propagation Neural Network is applied to the problem of classifying the MNIST handwritten digit database. The network's performance is evaluated by analyzing the classification accuracy and visualizing the loss during the training phase. This two-pronged investigation demonstrates a comprehensive grasp of semi-supervised learning strategies and their use in particular classification tasks, offering insightful information about the capability and flexibility of neural networks.

1. Introduction

One major challenge in optical character recognition is handwritten digit recognition, which is also a key test case for theories pertaining to pattern recognition and different machine learning algorithms. Many standardized databases have been created to promote developments in the domains of pattern recognition and machine learning. The handwritten numbers are preprocessed, such as segmented and normalized, to lessen the workload and enable researchers to compare the recognition outcomes of their methods on a common basis [1].

The MNIST handwritten digit database [3] is a benchmark for quickly evaluating theories about pattern recognition and machine learning algorithms. The database contains 10,000 images for testing and 60,000 images of handwritten digits from the same distribution that were used to train the classifier. Every single black and white image is centered within a fixed-size frame that is 28 x 28 pixels and is subjected to size normalization. The center of the intensity and the center of the image line up, yielding a sample vector dimensionality of 784 (28 * 28).

Semi-supervised learning leverages both labeled and unlabeled data to improve classification models. In this study, we apply semi-supervised learning techniques to the MNIST digits dataset and a custom Two-Moon toy dataset. We evaluate the performance of the baseline supervised learning model against various semi-supervised approaches.

2. Models

As mentioned earlier, we will use 10,000 samples from the MNIST dataset, with 10 labeled samples from each class and the rest treated as unlabeled to train the model. We then use the test dataset to measure the performance of our model. The neural network architecture for our MNIST data classification is [784, 200, 10], representing input dimensions, hidden layer dimensions, and output dimensions, respectively. Rectified Linear Unit (ReLU) activation defined mathematically as:

$$\text{ReLU}(x) = \max(0, x), \tag{1}$$

was used for the hidden layer and softmax activation defined mathematically as:

$$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2}$$

where z is the input vector, and K the different classes (elements of the vector)

was used for the output layer. We also created a custom Two-Moon dataset made up of 300 data samples with 3 labeled examples from each category and the remaining data as unlabeled. We then trained our model using the dataset. The neural network architecture for this dataset is [2, 10, 2], representing input dimensions, hidden layer dimensions, and output dimensions, respectively. Sigmoid activation defined mathematically as:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3}$$

was used for both the hidden layer and the output layers.

2.1. Baseline Algorithm

In this phase of our research, we implemented a baseline algorithm to establish a benchmark for evaluating our neural network model's performance. The neural network was trained using labeled data. We employ the network loss function as the cross-entropy error function. The cross-entropy error function appears to be a better fit for classification issues than the mean squared error (MSE) function. Regarding the optimizer, in this case, we decided to use Adam, which implements the Adam algorithm. The trained neural network was used to make predictions on unlabeled data. Relevant metrics (such as accuracy, precision, recall, etc.) were calculated.

2.1.1 Entropy Minimization

Entropy Regularization [2] is a technique for utilizing unlabeled data in the context of maximum a posteriori estimation. This scheme minimizes the conditional entropy of class probabilities for unlabeled data, thereby favoring low density separation between classes without any modeling of the density.

$$H(Y|X) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} P(Y=\omega_k|x_i)\log P(Y=\omega_k|x_i) \tag{4}$$

where n is the number of unlabeled data, K is the number of classes, ω_k is the unknown label of the ith unlabeled sample, x_i is the input vector of ith unlabeled sample

The MAP estimate is defined as the maximizer of the posterior distribution:

$$C(\theta,\lambda) = \sum_{j=1}^{n}\log P(y_j|x_j;\theta) - \lambda H(y_i|x_i;\theta) \tag{5}$$

where m is the number of labeled data, x_j is the jth labeled sample, λ is a coefficient balancing the two terms.

In this phase of our project, we calculated entropy for the unlabeled data since it does not require labels and also calculated cross-entropy for the labeled data. We then combined the loss functions and set a hyperparameter λ which controls the entropy minimization rate. The neural network was trained using both labeled and unlabeled data. We employed the network loss function as our customized entropy minimization loss function. We still maintained the Adam as the optimizer for the neural network, which implements the Adam algorithm. The trained neural network was used to make predictions on test data. Relevant metrics (such as accuracy, precision, recall, etc.) were calculated.

2.1.2 Pseudo Label

Neural networks can be trained semi-supervisedly using the pseudo-label technique. Pseudo-Labels are used as real labels for the unlabeled data by simply selecting the class with the highest predicted probability. This is essentially the same as regularizing entropy. It encourages low-density class separation, which is a prerequisite for semi-supervised learning that is frequently assumed [4].

In this phase of our project, we calculated the predicted labels for the unlabeled data and assigned the labels to them. The neural network was trained using both labeled and pseudo-labeled data. We employ the network loss function as our customized entropy minimization loss function. We still maintained the Adam as the optimizer for the neural network, which implements the Adam algorithm. The trained neural network was used to make predictions on test data. Relevant metrics (such as accuracy, precision, recall, etc.) were calculated.

2.1.3 Virtual Adversarial Training (VAT)

In this phase of our project, we employed Virtual Adversarial Training (VAT), a novel neural network regularization technique to train our model. VAT calculates the model's local smoothness by evaluating the model's resilience to minor perturbations surrounding each input [5]. VAT is appropriate for semi-supervised learning because, in contrast to adversarial training, it only uses the output distribution to determine the perturbation direction. Relevant metrics (such as accuracy, precision, recall, etc.) were calculated.

3. Experiments and Results

In this section of the paper, we will discuss the results of each method on both the mnist and two moon dataset. We will compare their performances and discuss any observed patterns. The metrics we used to evaluate our models performance are accuracy, recall, precision, and F1 score.

These metrics are defined mathematically as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall (Sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

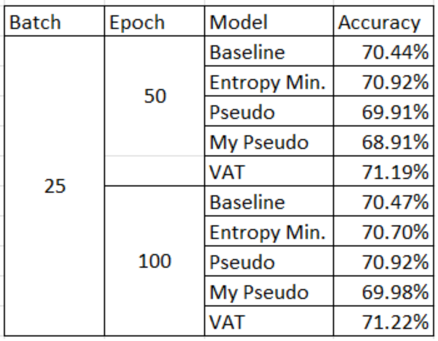
$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

where:

- TP is the number of true positives.
- TN is the number of true negatives.
- FP is the number of false positives.
- FN is the number of false negatives.

3.1. MNIST Dataset

We present the results comparison using tables and figures for analytical purposes. The testing accuracy that we compare is the average across 5 runs because random bias can occur on a single result. The system's parameters are batch size, learning rate was set to 0.1, and number of epochs were varied as well. Table 1 shows the outcome of our experiments. From the table, it can be seen that as the number of epochs increased, all the methods except entropy minimization performed better.



Batch	Epoch	Model	Accuracy
25	50	Baseline	70.44%
		Entropy Min.	70.92%
		Pseudo	69.91%
		My Pseudo	68.91%
		VAT	71.19%
	100	Baseline	70.47%
		Entropy Min.	70.70%
		Pseudo	70.92%
		My Pseudo	69.98%
		VAT	71.22%

Table 1. Accuracy of the model for classifying the mnist dataset using the baseline, entropy minimization and pseudo labeling methods with a fixed batch size of 25, varied epochs of 50 and 100, and learning rate 0.1

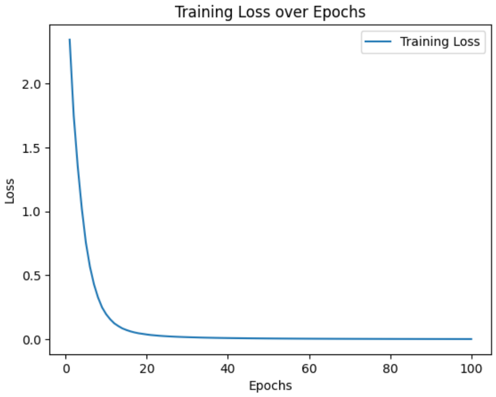


Figure 1. The figure above shows the training error for the dataset using the Baseline method with 100 epochs.

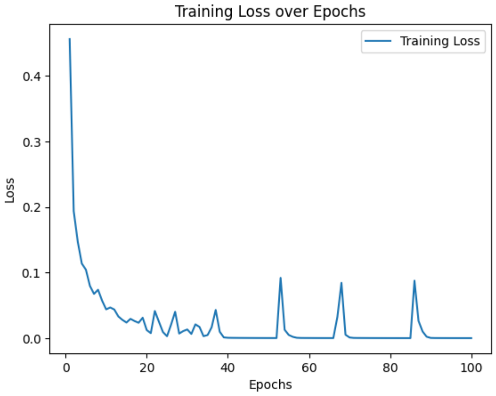


Figure 2. The figure above shows the training error for the dataset using the Entropy Minimization method with 100 epochs.

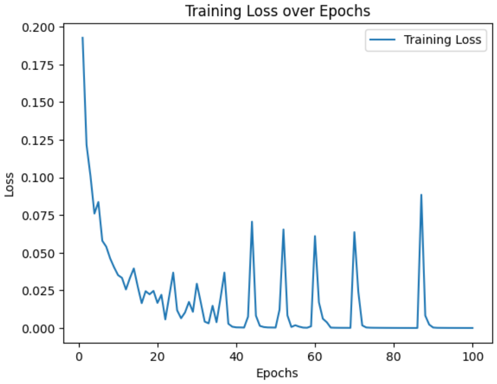


Figure 3. The figure above shows the training error for the dataset using the Pseudo Labeling method with 100 epochs.

Each model's historical loss from "all losses" during network learning is plotted, as seen above. The Baseline model's loss plot is displayed in Figure 1, the entropy minimization model's loss plot is displayed in Figure 2, the

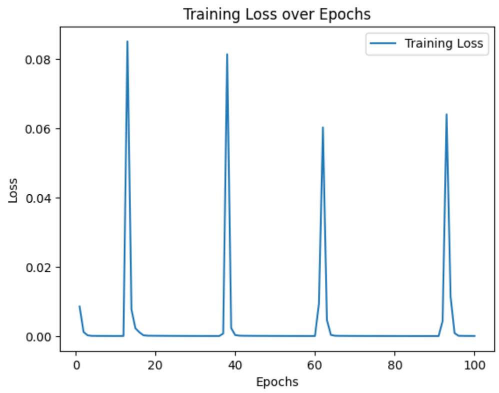


Figure 4. The figure above shows the training error for the dataset using a customized Pseudo Labeling method with 100 epochs.

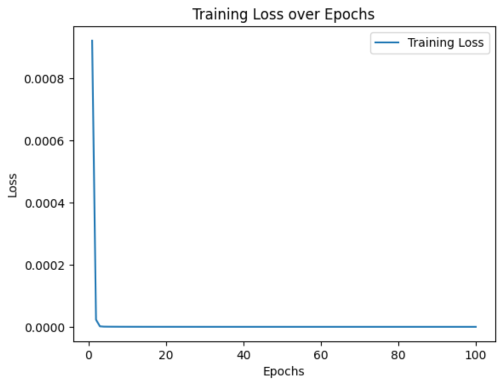


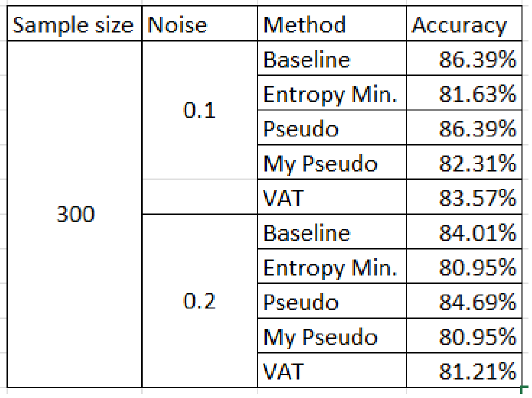
Figure 5. The figure above shows the training error for the dataset using a VAT method with 100 epochs.

pseudo label model's loss plot is displayed in Figure 3, and our customized pseudo label model's loss is indicated in Figure 4. It is easy to see that the VAT model's loss is the smallest.

3.2. Two-Moon Dataset

We present the results comparison using tables and figures for analytical purposes. The testing accuracy that we compare is the average across 5 runs because random bias can occur on a single result. The system's parameters are sample size and sampling noise. Table 2 shows the outcome of our experiments.

As you can see, the neural network trained with only 6 labeled samples does not capture the shape of the two moons very well. It has a high accuracy on the unlabeled data and a linear decision boundary. This is the baseline performance that we want to improve with semi-supervised learning algorithms. However, from our analysis it seems the baseline algorithm outdoes all the other algorithms.



Sample size	Noise	Method	Accuracy
300	0.1	Baseline	86.39%
		Entropy Min.	81.63%
		Pseudo	86.39%
		My Pseudo	82.31%
		VAT	83.57%
	0.2	Baseline	84.01%
		Entropy Min.	80.95%
		Pseudo	84.69%
		My Pseudo	80.95%
		VAT	81.21%

Table 2. Accuracy of the model for classifying the two-moon dataset using the baseline, entropy minimization, pseudo labeling, VAT and our customized pseudo labeling methods with a fixed sample size of 300 and varied sampling noise of 0.1 and 0.2

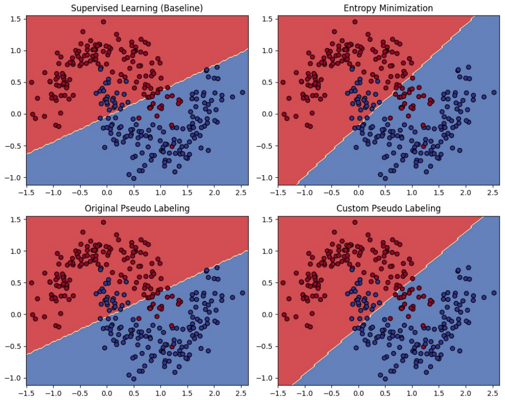


Figure 6. The figure above shows all the decision boundaries for the Two Moon dataset using baseline, entropy minimization, pseudo labeling, and our customized pseudo labeling methods.

4. Conclusion

To sum up, we explore semi-supervised learning methods using the MNIST digits and Two-Moon datasets. Four models are compared, revealing subtle variations in performance. VAT is greatly improved by longer epochs, but Entropy Minimization lags. CNN models surpass others, as seen by the smoothness of the loss plot. The study highlights the usefulness of specific methods and provides directions for further investigation, especially with regard to improving model robustness and semi-supervised methods for a variety of datasets.

References

[1] Liang Deng. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[2] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, pages 529–536, 2005.

[3] Yann LeCun and Corinna Cortes. Mnist handwritten digit database. `http://yann.lecun.com/exdb/mnist/`, 2010.

[4] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, Atlanta, 2013.

[5] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2018.

5. Appendix

5.1. Workload

I performed this research project with Alhassan Faharudeen. I was in charge of creating and executing the neural network architectures for the MNIST digit datasets and making sure they were efficient at identifying underlying patterns. Alhassan was in charge of creating and executing the neural network architectures for the Two-moon dataset. I explored the details of the Baseline, Entropy Minimization, Pseudo-Label, and VAT techniques in the context of semi-supervised learning, taking advantage of its ability to improve the model's performance with unlabeled data. Alhassan on the other hand performed the same methods on the two-moon dataset. However, at times when I was confronted with some coding challenges I could not figure out, Alhassan stepped in to help out and I did likewise when he confronted some issues. I took the lead in creating an extensive project report during the documentation phase, making sure it was clear and comprehensive. I worked well with Alhassan as a team, took an active part in conversations, and was crucial in creating a supportive and productive team environment. The project's course was equally determined by my contribution and Alhassan's, and I'm proud of the variety of roles I played to help us accomplish our goals.