An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification: Reproducibility study

Prateek Jeet Singh Sohi

1 Introduction

One of the fast growing machine learning applications nowadays is image classification. In order to perform this application while achieving high accuracies neural networks, especially MLP and CNN, are common employed. MLP is a feedforward artificial neural network that generates a set of outputs from a set of inputs. An advantage of MLP is that it is extremely flexible when dealing with different data types besides being good at learning important features. However, it has many hyper parameters that should be tuned [1]. In this regard, CNN-MLP hybrid method was being used for image classification [2]. Despite having very different behaviors, these two algorithms were integrated in an effective way using a rule-based decision fusion approach. The duty of this method was to classify very fine spatial resolution (VFSR) remotely sensed imagery. Additionally, in [3], five different architectures with varying convolutional layers, filter size and fully connected layers is used to present application of CNN for image classification. Authors in [4] used an object-based method for summer crops classification. Various machine learning algorithms such as decision tree, Logistic regression (LR), support vector machine (SVM) and MLP were used; while, MLP being the most accurate. Moreover, in [5] artificial meta-plasticity is applied to a MLP for image classification. Artificial meta-plasticity is a novel Artificial Neural Network (ANN) training algorithm that gives more relevance to less frequent training patterns and subtracts relevance to the frequent ones during training phase, achieving a much more efficient training, while at least maintaining the MLP performance. On the other hand, the combination between SVMs, including linear SVM, and convolutional nets were previously proposed as a way of performance improvement [6]. Additionally, unlike the common practice of using softmax function as the last layer classifier of CNN's structure, some other studies used linear SVM and tested its accuracy compared to the softmax function [7, 8]. In this report, we are going to perform a reproducibility study on the technique of combining SVM in the structure of a CNN for image classification developed in [8]. The datasets used in this study are MNIST and Fashion MNIST. Besides reproducing the paper's results, additional experiments including changing the number of hidden layers, learning rate and adding dropout at different levels have been performed on the model and the results have been compared.

2 Scope of reproducibility

3 Methodology

3.1 Model descriptions

the CNN+SVM architecture proposed in [8] is a convolutional neural network with the following architecture.

- Input: Image of size $32 \times 32 \times 1$
- Layer-1: Type (CONV2D) kernel size = 5×5 size, 32 filters, 1 stride
- Activation: ReLu()

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

- Layer-2 : Type (Pooling) kernel size = 2 $\times$2 , 1 stride
- Layer-3 : Type (CONV2D) kernel size = 5 $\times$ 5 size, 64 filters, 1 stride
- Activation: ReLu()
- Layer-4 : Type (Pooling) kernel size = 2 $\times$2 , 1 stride
- Layer-5 : Type (Fully Connected/Flatten) 1024 Hidden Neurons
- Layer-6 : Type (Dropout) p=0.5
- layer-7: Type (Fully Connected) 10 output classes

3.2 Datasets

The datasets used throughout our reproducibility study were MNIST and Fashion MNIST datasets. For both data sets, TensorFlow was used to load them with their standard training and test portions in order to acquire the data. Fashion MNIST dataset consists of several images (28x28 gray scale) of clothes, bags and shoes; whereas, MNIST dataset consists of several images (28x28 gray scale) of handwritten numbers from 0 to 9.

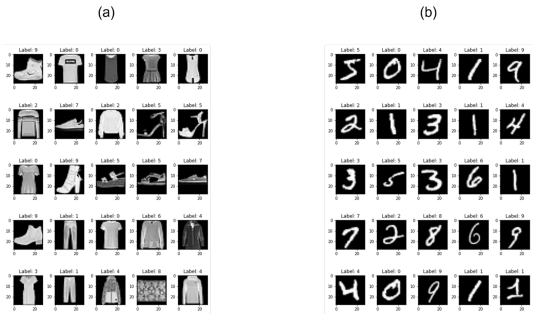

Figure 1: Results of reproducibility using MNIST dataset

In case of Fashion MNIST dataset, classes are evenly distributed in both training and test datasets as illustrated in Figure 2. For instance, each class appeared 6000 and 1000 times in the training and test datasets, respectively. Where, the number of classes are 10; hence, training and test datasets contain 60000 and 10000 data points/instances, respectively. Each class takes a numeric label from 0 to 9. In particular, the classes are T-shirt/Top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag and Ankle boot; whereas, their respective labels are 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. While, in the case of MNIST dataset, the labels are not evenly distributed; where, the label 1 is the most one that appeared in both training and test datasets. On the other hand, 5 is the least label that appeared in both MNIST standard training and test datasets. It is worth noting that similar to Fashion MNIST dataset, MNIST training and test datasets contain 60000 and 10000 data points/instances, respectively.

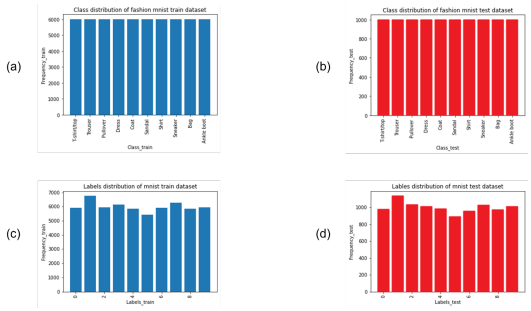

Figure 2: Results of reproducibility using MNIST dataset

It is worth noting that the images present in both datasets have a range of pixels from 0 to 255. Thus, after loading the datasets, their data was simply normalized by being divided on 255 in order to make the data in the range of 0 to 1.

3.3 Hyperparameters

For, reproducing the claims of the paper we used the following hyper-paraments as mentioned in the original text.

Hyper-parameter	Values
Batch Size	128
Dropout	0.5
Learning Rate	1e-3
epoch	10,000

Table 1: hyper-parameters for the base line model

To understand the effect of various hyper-parameters on the models performance we performed experiments by changing just one hyper-parameter at a time while keeping all the other values same as the base line. The results of the experiments conducted are summarized in the results section below.

3.4 Experimental setup

The same methodology developed and presented in [8] was also followed in the first part of the current study. After building and running the codes according to the parameters' values stated by the author of the original study, additional experiments were done as a way of investigating the model generalization as well as improving the prediction/classification accuracy. Table 2 summarizes the different values of hyperparameters(HP's) changed across the new experiments during this study in while keeping the rest of the HP's the same as mentioned as in Table 1. It is worth mentioning that training accuracy, training loss, validation accuracy and validation loss were considered and compared in the current study. Besides, these experiments were only done on the MNIST dataset employing the CNN-SVM model.

Hyper-parameter	Values
Learning rate	0.1, 0.05, 0.025, 0.01
Dropout	0.5, 0.3, 0.1
Activation function for SVM layer	LeakyRelu
Number of Conv. Hidden layers	3, 4, 5
Batch size	16, 32, 64

Table 2: Additional experiments on CNN-SVM model

3.5 Computational requirements

For the purpose of reproducing the authors claims we re-implemented the authors CNN architecture based on the information porvided in [8]. since we had to run about 10,000 training epochs we used a NVIDIA Tesla P6 GPUs with 16 GB of memory to perform all our experiments.

4 Results

In [8] it is claimed that CNN-SVM hybrid model performs almost similarly to the CNN-softmax model architecture with CNN-SVM model being slightly in-accurate in comparison. The paper also predicts that using a slightly more complex base model could help improve the performance of the CNN-SVM hybrid. By means of reproducing the papers experiments we were able to verify that the performance of both the models are quiet similar when considering the best accuracies. However, the claim that a deeper model would perform better is not entirely true as we will prove in the coming sub-sections.

4.1 Results reproducing original paper

4.1.1 Result 1

Figures 3 and 4 illustrate the results of the reproducibility test done on both MNIST and Fashion MNIST datasets. Training and validation Losses as well as the corresponding accuracies are compared in figure-3 and figure-4.

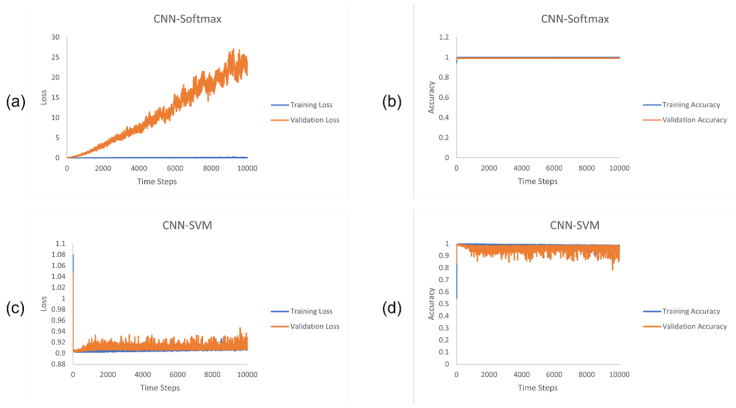

Figure 3: Results of reproducibility using MNIST dataset

In this regard, as shown in the figures, the average training accuracies obtained 98.5% and 94.1% for the MNIST and Fashion-MNIST datasets, respectively. Whereas, the test set accuracies reached 96.9% and 89.9%.

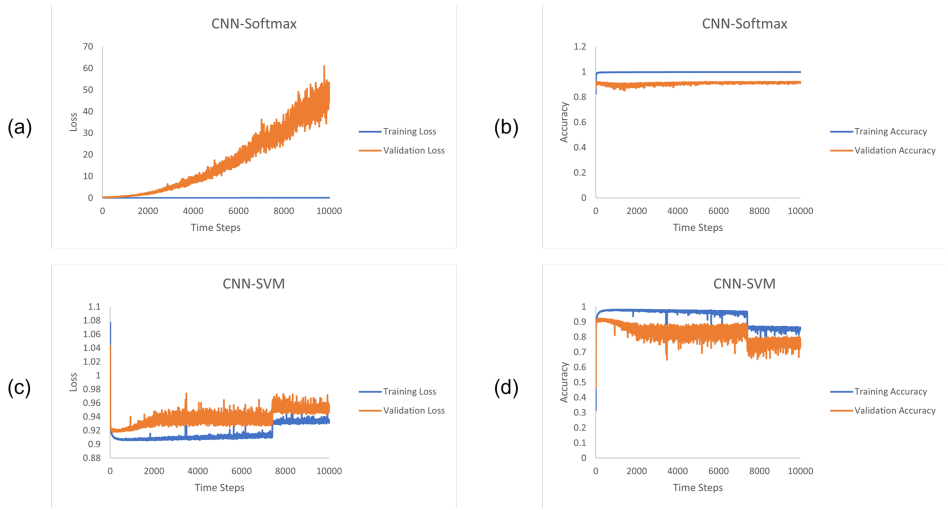

Figure 4: Results of reproducibility using Fashion-MNIST dataset

4.1.2 Result 2

Dataset	training accuracy	validation accuracy
MNIST	99.71	99.25
Fashion-MNIST	98.29	92.34

Table 3: Best training and validation set accuracy of CNN-SVM for MNIST and Fashion-MNIST dataset

Table-3 shows that CNN-SVM over-fits on the Fashion-MNIST data-set because the difference between the training and validation set is 5.95%. This supports the claims of [8] that it is difficult to generalize on Fashion-MNIST dataset as compared to MNIST.

Dataset	CNN-Softmax	CNN-SVM
MNIST	98.5	96.9
Fashion-MNIST	91.2	89.9

Table 4: Test-set accuracy of CNN-softmax and CNN-SVM for MNIST and Fashion-MNIST dataset

Table-4 show that the test set accuracy of both CNN-SVM and CNN-softmax differ by 1.6% and 1.3% for MNIST and Fashion-MNIST respectively. These figures are similar to the claims made in table-3 of [8].

4.2 Results beyond original paper

Exp. Value	Max. Training Accuracy	Max. Validation Accuracy	Min. Training Loss	Min. Validation Loss
Learning Rate				
0.01	11.5	15.9	1.077	1.075
0.025	11.6	16.0	1.077	1.068
0.05	11.5	15.3	1.077	1.070
0.1	11.4	17.3	1.078	1.066
Dropout				
0.1	99.9	99.2	0.900	0.902
0.3	99.9	99.2	0.901	0.902
Activation function for SVM layer				
Leaky ReLU	99.3	98.8	0.906	0.906
Number of Conv. Hidden layers				
3 Layers	99.4	99.2	0.903	0.904
4 Layers	98.5	98.9	0.906	0.905
5 Layers	46.7	48.2	1.013	1.009
Batch Size				
16	96.8	98.3	0.910	0.907
32	98.6	98.9	0.907	0.905
64	99.3	99.2	0.903	0.904

Table 5: Summary of additional experiments results

4.2.1 Additional Result-1 (Learning Rate)

Based on our experiments related to learning rate in Table-5 it is observed that a learning rate grater than the order of e-3 is detrimental to the training process. At these higher learning rates the CNN-SVM model can not even approximate to a local minima.

4.2.2 Additional Result 2 (Dropout)

The effect of decreasing the dropout ratio(p) is as expected and follows the general concept that lower values of the dropout ratio will result in over fitting this is supported by Table-5 Dropout section where for p=0.3 the difference between the best validation loss and the best training loss is 0.001 while the difference between the best validation loss and the best training loss is 0.002 for p=0.1.

4.2.3 Additional Result 3 (activation function)

By comparing results of Table-3 and Table-5 we come to a conclusion that the authors choice of using ReLu activation is optimal because we see that when using Leaky Relu the model under fits.

4.2.4 Additional Result 4 (Number of Convolutional Layer)

From values of table-5, Number of Conv. Hidden Layer section we come to a conclusion that unlike other CNN models where increasing the model depth results in higher model variance we observe that CNN-SVM actually shows a lower variance when the number of convolutional layers are increased.

4.2.5 Additional Result 5 (Batch Size)

From values of table-5, Batch Size section we come to a conclusion that performance increases with larger batch sizes.

5 Statement of Contribution

All the team members have contributed equally towards the completion of mini project 4. We had regular meetings via zoom, and have developed each part of this mini project equally.

References

References

P. Naraei, A. Abhari, and A. Sadeghian, "Application of multilayer perceptron neural networks and support vector machines in classification of healthcare data," FTC 2016 - Proc. Futur. Technol. Conf., no. December, pp. 848–852, 2017.

C. Zhang et al., "A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification," ISPRS J. Photogramm. Remote Sens., vol. 140, pp. 133–144, 2018.

S. S. Kadam, A. C. Adamuthe, and A. B. Patil, "CNN Model for Image Classification on MNIST and Fashion-MNIST Dataset," J. Sci. Res., vol. 64, no. 02, pp. 374–384, 2020.

J. M. Pe~na, P. A. Guti´errez, C. Herv´as-Mart´ınez, J. Six, R. E. Plant, and F. L´opez-Granados, "Object-based image classification of summer crops with machine learning methods," Remote Sens., vol. 6, no. 6, pp. 5019–5041, 2014.

A. Marcano-Cede~no, A. Alvarez-Vellisco, and D. Andina, "Artificial metaplasticity MLP applied to image classification," IEEE Int. Conf. Ind. Informatics, pp. 650–653, 2009.

Huang, F. J. and LeCun, Y. Large-scale learning with SVM and convolutional for generic object categorization. In CVPR, pp. I: 284–291, 2006. URL http://dx.doi.org/ 10.1109/CVPR.2006.164

Yichuan Tang. 2013. Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239 (2013).

Agarap A. F., An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification, 2019, https://doi.org/10.48550/arXiv.1712.03541

Appendix

Reproducibility Summary

Scope of Reproducibility

Most of the studies previously done in the literature for image classification using famous datasets such as MNIST and Fashion MNIST utilized convolutional and fully-connected neural networks to perform this duty. Where, softmax activation function was utilized to minimize cross-entropy loss and perform the prediction task. In this project, we reproduce a study entitled "An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification" where the softmax layer was replaced by a linear support vector machine (SVM). Hence, we are trying to test the author's hypothesis that this method is comparable with the common softmax-based one.

Methodology

The same methodology developed in the above mentioned paper was also followed in the first part of the current study where the author's codes were used. After building and running the codes according to the parameters' values stated by the author of the original study, additional experiments were done as a way of investigating the model generalization as well as improving the prediction/classification accuracy.

Results

The methodology was followed and reproduced on the MNIST and Fashion-MNIST datasets with average training accuracies of 98.5% and 94.1%. Whereas, the average validation accuracies reached 96.9% and 82.9%. Additional experiments were performed to further test the model's accuracy for image classification of MNIST dataset. In these runs, different hyper-parameters, e.g. dropout value, activation function for SVM layer and batch size, were changed to investigate their effect on the training and validation accuracies. For instance, the best result that we obtained, shows the average training and validation accuracies reaches 99.3% and 98% respectively at a dropout of 0.03.

What was easy

Generally, CNN-Softmax code and the majority of the CNN-SVM code (except certain part to be mentioned in the next section) was easy to be implemented based on the information provided in [8].

What was difficult

At the beginning it was difficult to run the part of the code representing the SVM layer in the CNN-SVM model. However, fortunately, this problem was mitigated after several debugging trials. In addition, the model was not found stable as claimed by the paper's authors. This issue appeared upon performing the additional experiments.

Communication with original authors

No communications were done with original authors.