Multiclass handwritten character classification using CNN

Mayank Gupta
Electrical and Computer Engineering
University of Florida

Mohit Patil
Electrical and Computer Engineering
University of Florida

Parth Shah
Electrical and Computer Engineering
University of Florida

Rajandeep Singh
Electrical and Computer Engineering
University of Florida

***Abstract*—The study presents CNN as a classifier for eight-character classifier on the set of English alphabet characters (a, b, c, d, h, i, j, k). The data needed dimensional normalization using padding for the optimal model performance.We found out that the proposed model works satisfactorily in the classification and offers convenient processing time for the limited computational load. The CNN model developed in the study achieved 95.05% overall classification accuracy. The paper conducted experiments on the data preprocessing and found out that the model performance improves with the larger training dataset. The model also experimented with epoch limit to find the optimal epoch number needed to minimize the training time, this limit was found to be 20 epochs. The learning rate showed maximum performance at the value of 0.01 and was fixed as a hyperparameter. The comparison carried out for different numbers of layers and weight decay parameters concluded that while performance improved when the number of layers was doubled, the weight decay regularization did not have a substantial effect on the performance.**

1. Introduction

Natural language processing has become one of the most important topics of research due to the digitization of data. Text recognition and more specifically character-level recognition is at the core of it.[1] Over the years, almost all types of machine learning approaches have tried to tackle this problem to the best of their ability. One of the most popular and provenly efficient machine learning approaches for this application is Neural Networks. Neural networks are a better choice for character recognition as designing machine learning algorithms to directly operate on the binary images can be advantageous than working with manually designed features extraction of handwritten characters. Neural networks, therefore, eliminate the element of designers' skills to extract appropriate features [2]. Furthermore, Neural Networks can analyze and extract relationships that are not linear in nature.[3]. Since character recognition can have very different types of inputs, the best algorithm for tackling unseen data is the neural network according to various studies[3], [4]. Additionally, the size of the letters does affect the accuracy of the model [4]. Within neural networks, Convolutional neural networks (CNNs) present a strong case as they can be trained to adapt to the variability in the images like handwritten characters [2]. CNNs have proven to be more efficient within Neural networks and studies dedicate this efficiency to reduced parameters and in turn reduced computing due to CNN characteristics such as weights sharing, max-pooling [5].
In the presented study, we demonstrate the usefulness and efficacy of CNNs in the classification of 8 characters from the English alphabet, a,b,c,d,h,i,j,k. The data under consideration in the presented study, the classroom data (referred to as 'classroom data' here onwards), is 6400 binary images of handwritten, cursive lower case character a, b, c, d, h, i, j, k. Each class has 800 images in the dataset. We also include a 9 the class i.e. neither of the 8 characters. We do so by conducting several experiments to find the optimal combination of parameters in the proposed CNN but also with the dataset provided. We will discuss the results of our experiments and how they reflect on the methodology chosen for the task and fit of the model for character classification in general. We conclude the study with the comments on the limitations as well as potential improvements in the method.

2. Implementation

We split the data in a 70:30 ratio and preserved the 30% data for the final testing. This test data was kept untouched for the entire length of experiments and validation. Cross-validation was performed on 70% of the data for all the conducted experiments.
We ran experiments by using additional data by the "EMNIST dataset" in the training. For this purpose, 70% of the classroom data was split further into 30:40 ratio and 40% was kept for validating while the 30% was combined with the EMNIST dataset and use for training.

As part of the objective, we wanted to classify a "false class" labeled "-1" which is neither of the 8 classes defined. We came up with a dual strategy for this issue. We set up a low confidence threshold at 2 and a confidence difference threshold at 3. If the confidence of all 8 classes was below the lower threshold, we classified that data as -1. In addition, if the difference between the confidence of the 2 highest confidence classes is lower than the confidence difference threshold, we classified it as -1. This approach followed

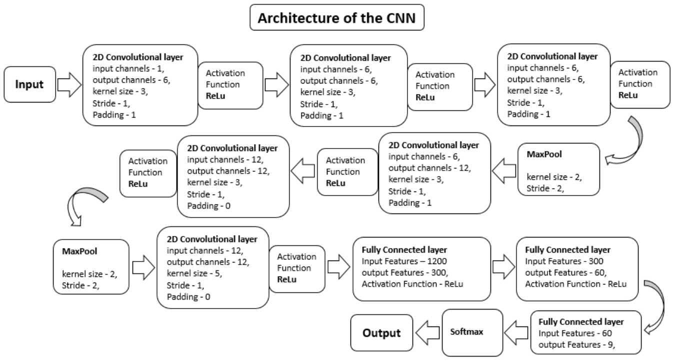

Figure 1. Architecture of CNN.

from the logical reasoning that if the model is not sure about any of the classes, or if the model can not clearly decide which class to assign (that will result in multiple classes getting accuracy above higher threshold), it must be that the data in none of the predefined classes. We had confidence in this approach only after achieving the 8 class classification accuracy of 95.01% on the test subset of the classroom data. ReLu was chosen as activation function due to it's characteristic of fast and generalized training (8). Softmax was used in the outermost layer as an activation function based on its proven effectiveness in 1 of K classification problems (9). Softmax is especially useful in multiclass classification as the output provided is normalized between unit value giving the output as probabilities of classification (10). After the conclusions from experiments carried out to optimize the model, we finalized the following architecture for the CNN described in the study (Figure 1).

3. Experiments

Several experiments were conducted on datasets as well as the algorithm to validate our approach. The experiment results lead us to finalize the parameters and fine-tune the model for the best classification performance. All the experiments and results are described below.

3.1. Algorithm and architecture

Given the data and the objective of the study, we propose multiple competing methods to be judged at the initial stage. We compared the performance of CNNs[Amelia] against 3 Algorithms, Multinomial Naive Bayes, Logistical Regression and a Perceptron. (Figure 2) to confirm the efficacy of our chosen approach for character classification.
We observed that the accuracy remained in the range 72-77% and we were not able to go beyond that. Also, by using the Canny Edge detector, the performance either remains almost the same and can even go down in some cases, as seen in the case of the Multinomial Naive Bayes.the most numbers of misclassifications happened in the case of the letter 'e' on an average as it the accuracy was only in the range of 62-70%. CNN outperformed others

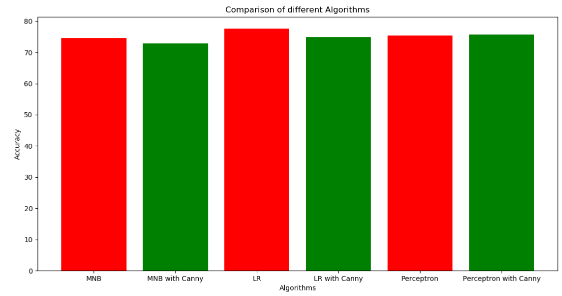

Figure 2. Comparison of algorithm performances to finalize the algorithm.

with 91% average accuracy(Experiment - Is data adequate?) solidifying our speculation based on the literature review [6]

3.2. Analysis of the data

The dimensions of each image in the class are variable which affects the classifier performance. The normalization of images is required in this case and can be done in multiple ways, padding and stretching are some of the commonly used techniques. The principle behind padding the images has an inherent advantage over stretching the images to normalize. In stretching the image, we always risk manipulating the object of classification, i.e. the handwritten character. The stretch is bound to distort the character and hence the classification can not be fully trusted due to changes shape, ration and defining features of the handwritten character. Padding on the other hand just extends the blank canvas and keeps the handwritten character unaffected while providing the standard dimension of the image. We are using padded data for all the subsequent tests in the study.

3.3. Is data adequate?

The classroom dataset provided is limited. Especially in the cases of handwritten character recognition, it is historically known that having large data for training is always beneficial. We conducted an experiment to test the limitations due to the dataset size by using the "EMNIST dataset of handwritten digits" [EMNIST] and comparing the performance against classroom data used independently. The classroom data independently gave us an accuracy of 91%. Using only the EMNIST dataset for training resulted in validation accuracy of 86% which suggests that the classroom dataset and EMNIST dataset have differences and cannot be used independently and to improve performance we need to use their combination.

3.4. A hybrid approach to optimize the combination of two datasets

We used the EMNIST dataset in combination with 40% of the classroom that was separated for training. The EMNIST dataset is high density and has a size almost ten times

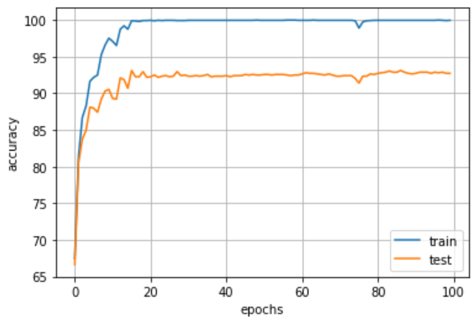

Figure 3. Variation of accuracy with the number of epochs.

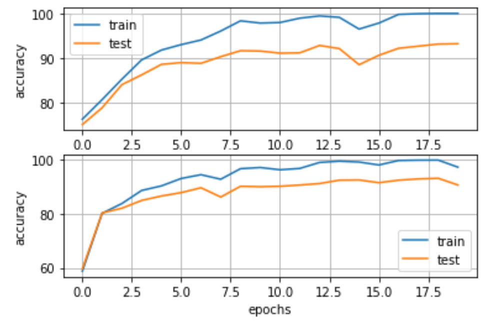

Figure 4. Variation of accuracy for a) 1st fold and b) 2nd fold in 2-fold cross-validation.

than the classroom training data subset. To counter this bias, we gave more weight (3 fold) to the classroom training sub-data in the number of epochs ran while training. The validation accuracy achieved was 94.68% which is more than the individual accuracy of either dataset.

3.5. Epoch limit

We ran CNN for a range of epoch to monitor how performance varies with the number of epochs. We found that training accuracy saturated at 100% after 20 epochs while the validation accuracy settles at 93% after 20 epochs with a marginal decrease occasionally (figure 3). Based on these observations, we kept the epoch limit to 20 in further analysis to save computation time.

3.6. k fold Cross-validation on the classroom data

From the classroom data that was split 70:30, 30% was kept for the final test. K fold validation ran on 70% data with 2 fold cross-validation and 20:50 data split. (Figure 4 a,b)

3.7. Number of layers in the CNN architecture

Adding layers can lead to the finer feature set but above a certain limit, additional layers will start overfitting the data and fitting to irregularities in the dataset. We doubled the number of layers in the CNN to find the optimal architecture to be used in further analysis.(Table 1)

TABLE 1. PERFORMANCE COMPARISON UPON DOUBLING THE NUMBER OF LAYERS

Architecture 1 2 convolution layers, 2 maxpool layers, 3 fully connected layers	Architecture 2 4 convolution layers, 2 maxpool layers, 3 fully connected layers	Architecture 3 6 convolution layers, 2 maxpool layers 3 fully connected layers
Class no. : accuracy	Class no. : accuracy	Class no. : accuracy
1: 95.03	1: 99.37	1: 92.40
2: 91.77	2: 98.10	2: 95.09
3: 90.50	3: 96.20	3: 96.12
4: 91.61	4: 98.20	4: 96.29
5: 93.75	5: 97.5	5: 91.41
6: 96.87	6: 98.12	6: 96.83
7: 79.37	7: 99.37	7: 93.03

3.8. Regularization- weight decay

For the regularization we tried a range of weight decay coefficients and compared the average classification accuracies. However, we didn't observe improvement in the performance with an increased penalty, instead the performance deteriorated at the higher penalties (Table 2)

TABLE 2. EFFECT OF WEIGHT DECAY ON THE CLASSIFIER PERFORMANCE

Weight decay coefficients	[0.0001, 0.001, 0.01, 0.1]
Average accuracy	[93.24, 92.69, 92.18, 12.26]

3.9. Learning rate vs epochs

We experimented with the learning rate by varying it from 0.001 to 0.1 on the logarithmic scale. As we can see in the figure (Figure 5), when the rate is set to 0.001, the model is unable to learn any features effectively and the accuracy remains low throughout. This suggests that this value is too low for the learning rate. On the other hand, when the learning rate is set to 0.1, while the model learns faster initially, it drops suddenly after a couple of epochs. This suggests that the loss is diverging because the learning rate is too high. Finally, when set to 0.01, we see a steady and stable increase in accuracy. Hence, we choose the hyperparameter learning rate as 0.01.

We experimented with the learning rate by varying it from 0.001 to 0.1 on the logarithmic scale. As we can see in the figure (Figure 5), when the rate is set to 0.001, the model is unable to learn any features effectively and the accuracy remains low throughout. This suggests that this value is too low for the learning rate. On the other hand, when the learning rate is set to 0.1, while the model learns faster initially, it drops suddenly after a couple of epochs. This suggests that the loss is diverging because the learning rate is too high. Finally, when set to 0.01, we see a steady and stable

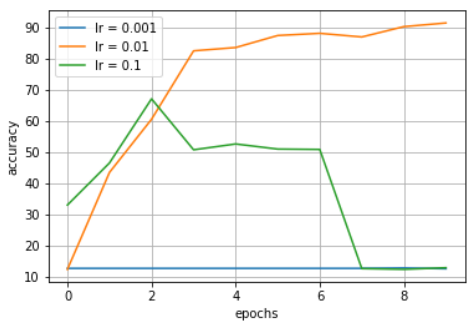

Figure 5. Effect of varying learning rate on performance.

increase in accuracy. Hence, we choose the hyperparameter learning rate as 0.01.

3.10. Final testing on testing subset

All the experiments described in the section above concluded different aspects of the CNN model we developed. The finalized model was then used to classify the test subset that was separated and kept untouched during all experiments. The test accuracy was model was achieved to be 95.05%.

4. Conclusion

The experiments concluded that CNN is well suited for the handwritten character classification by approaching them as raw image objects. It can be optimized to be computation and process time-efficient to provide good performance. The CNN model developed in the study achieved 95.05% maximum overall accuracy after optimization. The experiments on the data showed that classroom data on its own was limited and EMNIST data had differences from classroom data due to which combining them both with a modified training approach proved to yield the best results. Epoch limit experiment showed that the performance converged at 20 epochs and limiting all experiments to this limit provided us with a very time-efficient method. The stability of the model was confirmed by two-fold cross-validation. The number of convolutional layers showed significant improvement in performance and hence architecture with more layers was finalized. This improvement in performance could be due to an increased number of output channels and in turn, an increased number of learnable features for CNN. however, this might have lead to increased computational load. As a countermeasure, max-pooling after the convolutional layer has substantially reduced the computational load. Dropout could be added int he future studied to reduce the computational load even further. The learning rate showed expected trends with an inability to learn at very low learning rates and spiked losses at high rates. The optimal learning rate was then fixed as the hyperparameter. Also, regularization by weight decay did not provide any significant improvement in this case. However, since we still see a large difference in the train and validation accuracies, better regularization techniques must be explored. In conclusion, the developed CNN performs satisfactorily as a character classifier in handwritten character recognition with room for improvement.

References

[1] Zhang, Xiang et al. *Character-level Convolutional Networks for Text Classification.* NIPS (2015).

[2] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, *Gradient-based learning applied to document recognition,* in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[3] Tu, Jack, V., *Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes,* Journal Of Clinical Epidemiology. 49(11): 1225-1231, 1996.

[4] M. Mishra and M. Srivastava, *A view of Artificial Neural Network,* 2014 International Conference on Advances in Engineering & Technology Research, (ICAETR - 2014) Unnao, 2014, pp. 1-3.

[5] Abdel-Hamid, O., Mohamed, A.-r., Jiang, H. & Penn, G., 2012. *Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition* Kyoto, IEEE.

[6] Amelia Amelia, *Convolution Neural Network to Solve Letter Recognition Problem* Research School of Computer Science Australian National University 108 North Rd, Acton, ACT 2601, Australia.

[7] Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017) *EMNIST: an extension of MNIST to handwritten letters* Retrieved from http://arxiv.org/abs/1702.05373

[8] Zeiler, M. D. et al., 2013. *On rectified linear units for speech processing.* Vancouver, IEEE.

[9] Bishop, C., 1995.*Neural networks for pattern recognition.* New York: Oxford University Press, Inc.

[10] Graves, A. Schmidhuber, J., 2005.*Framewise phoneme classification with bidirectional LSTM and other neural network architectures.* Neural Networks, 18(5-6), pp. 602-610