Handwritten Mathematical Symbol Recognition using Machine Learning

Mak Fazlic, Alexandre Marcel Mourot, Noé Macé

EPFL CS233

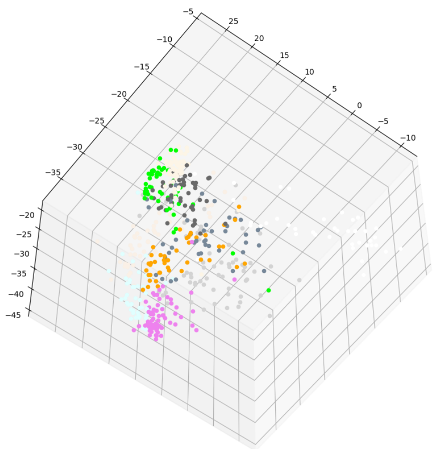
1 Introduction

In this paper, we present our approach to classify handwritten mathematical symbols using various machine learning techniques. Our work is based on a course project for the CS-233(b) Introduction to Machine Learning at the EPFL. We implemented and evaluated various machine learning methods on a subset of the HASYv2 dataset, which consists of small 32x32 images of hand-drawn mathematical symbols. The three methods used in our study are MLP, CNNs and PCA. We evaluated the performance of these models using accuracy and F1-score as evaluation metrics.

2 Method

2.1 PCA

Principal component decomposition or PCA is a methodology where we change the basis of the data such that the variance on each component is maximised. Once we get the principal components we can rank them by importance and reduce the dimension of our data by removing the components with low impact. We implemented a function that reduces the data to dimension d while maximising the importance of the kept data. We compare the performance reduced data classified by linear regression to the while data-set classified with the same method. We also reduced the data down to $3D$ and saw that PCA can be used to show the $1024D$ data in a lower dimension (following figure, only 10 characters for the image due to space).



2.2 MLP

Multilayer Perceptron is a classic neuron network. We give a vector as input, then each neuron, receives input from multiple sources and applies a weighted sum to the inputs. Each input is multiplied by a corresponding weight, and the results are summed together. The weighted sum is then passed through an activation function. We used a trainer to train our model. After each input forwarded we compare the predicted output to the actual one. We want to minimize this loss function. To do that we use a principle called back propagation. We use the difference between the predicted and actual output to adjust the weights and biases in the network using gradient descent optimization through all layers. We train our model with cross-validation to find the best hyper-parameters both in term of performance and efficiency. These hyper parameters include : the number of layers, the activation function as well as the dimensions of the hidden layers.

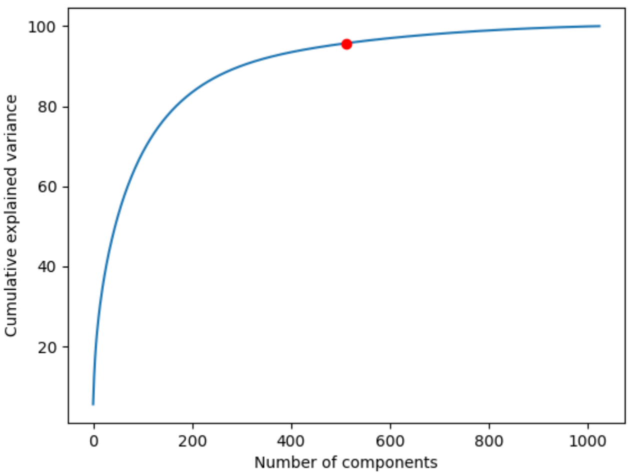
2.3 CNN

Our Convolutional Neural Network (CNN) model dinstiguishable trait lies in it's modularity and scalability. It consists of convolutional layers, each followed by a ReLU activation function, to introduce non-linearity, and max pooling layers for spacial down-sampling. Key parameters such as the number of input channels, the number of filters, kernel size for the convolutional and max pooling layers are all configurable to customize the model based on the dataset characteristics. By using this architecture, our model learns to extract the important features of the input images to classify them with high accuracy. The model follows a 'same' padding that ensures that the feature maps do not shrink in size during the convolutions, thereby retaining spatial information across layers.

3 Results

3.1 PCA

As seen from the figure of dimensionality reduction to $3D$ the method works well and it correctly clusters characters based on their label. The method also works well as the results of linear regression without PCA on $1024D$ was: Accuracy: 73.71% - F1-score: 0.7, while with PCA on $512D$: Accuracy: 68.05% - F1-score: 0.632. This means that with the reduction of half of the dimensions in the data-set we reach results close to the full data-set. With $512D$ we reach the expected cumulative expected covariance of approximately 95% as seen in the image presented below.



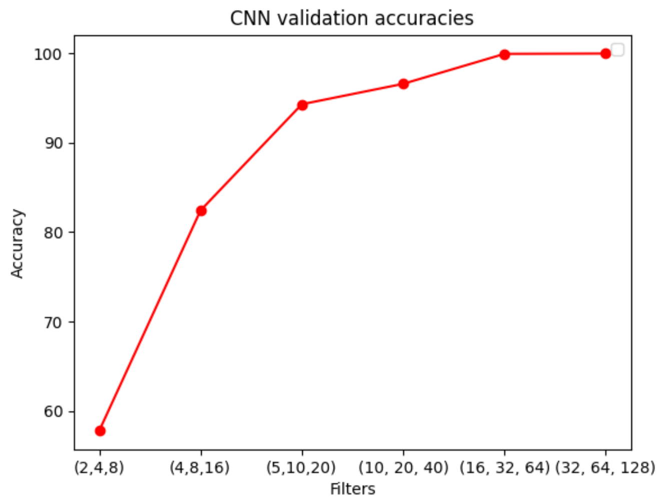
This shows that PCA was highly effective in reducing the dimensions with minimum information loss, plus we gain a lot in time when training the model. For this task the score was lower then the benchmark, as discussed previously, but, on larger datasets when we can not compute linear regression on the full dimensions of the data, this methodology could prove useful. Finally it helped us get a grasp of the relationship between different characters in the $3D$ map, which further demonstrates the utility of the reduction.

3.2 MLP

After optimizing our hyper-parameters we found that the more layers we added the better, for the hidden layer the larger the dimensions there were the better too and the best activation function was ReLu. Obviously increasing too much these parameters lead to overfitting and long runtime while being caped. After training our model with optimal hyper-parameters to balance performance and efficiency we obtain good results around 92% of accuracy. Interesting to note that for an MLP with only 1 hidden layer we are already at 90% of accuracy. We observed a very similar accuracy for our F1 measurement. We observe that model like CNN add features which allow for more efficient improvements.

3.3 CNN

Investigating the optimal architecture and hyperparameters for CNNs often involves a significant amount of trial and error, as various configurations can yield drastically different accuracies on the same task. The process inherently requires iterative experimentation and refinement to identify the model configuration that best generalizes the underlying patterns in the data.



The number of convolutional layers and the learning rate are parameters that have the most impact on the final predictions, and by iterating on them we could obtain a 95% accuracy on the validation set, moreover we have an F1 score of approximately 94%

4 Conclusion

This paper presented the application of PCA, MLP, and CNN techniques in classifying 20 different handwritten Greek letters, using the HASYv2 dataset. The PCA demonstrated a modest accuracy of around 68% with significant reduction in data dimensionality and runtime, MLP yielded about 92% accuracy and F1-score after hyperparameter optimization, while the CNN architecture achieved a remarkable accuracy of 95% after iterative experimentation on its configurations. Thus, the CNN proved to be the most effective machine learning method in predicting handwritten Greek letters in this study. We moreover conclude that PCA is very effective in reducing dimension while keeping uselfull informations. This is a good model to go through while dealing with large dataset.