Laboratory exercise on convolutional neural networks

Dániel MÁCSAI
Mauro VÁZQUEZ CHAS

Master in Artificial Intelligence

Computational Intelligence
CNN project

20th December 2024

Contents

TODO: Data augmentation: flipping, rotation, zooming, shifting, etc.

We are using the Adam optimizer with the categorical$_{c}rossentropyloss$.

For the activation function in the non-linear output layer OL we are using the

ADAM:

Beta1 controls the moving average of the first moment (i.e., momentum), while Beta2controls the moving average of the second moment (i.e., variance). Common settings: Beta1 is often set to 0.9 to capture most of the momentum of the gradients. Beta2 is typically set to 0.999 to provide better stabilization by giving more weight to past gradients.

We used this settings.

For weight decay, we used 1e-5, which is common and widely used for Adam.

1 Introduction

The objective of this study is to analyze the performance of different configurations of Convolutional Neural Networks (CNNs) for image classification tasks using the CalTech 101 Silhouettes dataset [**dataset**]. The dataset contains 8671 images of size 28×28, classified into 101 silhouette categories. We evaluate the CNN configurations based on mean accuracy, using various hyperparameter settings and dataset splits.

2 Methodology

2.1 Dataset and Preprocessing

The dataset [**dataset**] is provided as a MATLAB file containing the image data and corresponding labels. Each image is reshaped into a 28×28 grayscale format. Labels are one-hot encoded for compatibility with the CNN architecture.

2.2 CNN Architecture

The base CNN structure comprises:

1. An input layer.
2. Convolutional blocks (NB), each consisting of:
   - A convolutional layer (kernel size = 3, filter size = FS).
   - A non-linear hidden layer with activation functions (Sigmoid or ReLU).
   - A max-pooling layer (size = 2, stride = 2).
3. A fully connected layer (128 units, ReLU activation).
4. An output layer (softmax activation).
5. A categorical cross-entropy loss function.

We compared two configurations:

- NB = 1 with FS = 128.
- NB = 3 with FS = [32, 64, 128].

2.3 Parameter Selection

Hyperparameters were tuned as follows:

- Optimizer: Adam with exponential decay learning rate.
- Learning rate: 0.001 with a decay rate of 0.96 every 100,000 steps.
- Epochs: 20 (60 for extended evaluation).
- Batch size: 32 (128 for extended evaluation).

Dataset splits of 80/10/10, 40/20/40, and 10/10/80 were used to evaluate the model's performance under varying data availability scenarios.

3 Results

3.1 Mean Accuracy

Table 1 summarizes the mean accuracies for different configurations and dataset splits.

NB	FS	Activation	Data Split	Mean Accuracy
1	[128]	Sigmoid	80/10/10	0.0791
1	[128]	ReLU	80/10/10	0.6206
3	[32, 64, 128]	Sigmoid	80/10/10	0.6448
3	[32, 64, 128]	ReLU	80/10/10	0.7097

Table 1: Mean accuracies for different configurations.

TODO training and validation times

4 Discussion and Conclusions

The results indicate that increasing the number of convolutional blocks and using the ReLU activation function significantly improve performance. The best configuration achieved a mean accuracy of 70.97%, demonstrating the efficacy of deeper architectures and non-linear activation functions. Future work could explore additional architectures, such as ResNets, and experiment with data augmentation techniques.

5 Usage of generative AI models

We used ChatGPT for spelling checks, grammar corrections, and sentence structure improvements.

References

- CalTech 101 Silhouettes Dataset: `https://people.cs.umass.edu/~marlin/data.shtml`
- ChatGPT by OpenAI as mentioned above.