

EE4-62 Selected Topics in Computer Vision (2018-2019) CW2

Vasil Zlatanov (01120518), Nunzio Pucci (01113180)
vz215@ic.ac.uk, np1915@ic.ac.uk

I. DCGAN

A. DCGAN Architecture description

B. Tests on MNIST

Try some **different architectures, hyper-parameters**, and, if necessary, the aspects of **virtual batch normalization**, balancing G and D. Please discuss, with results, what challenge and how they are specifically addressing, including the quality of generated images and, also, the **mode collapse**.

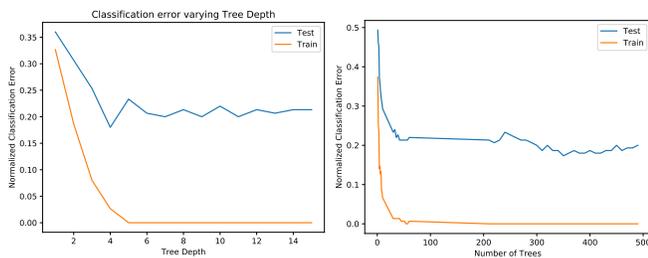


Figure 1. K-means Classification error varying tree depth (left) and forest size (right)

II. CGAN

A. CGAN Architecture description

B. Tests on MNIST

Try **different architectures, hyper-parameters**, and, if necessary, the aspects of **one-sided label smoothing, virtual batch normalization**, balancing G and D. Please perform qualitative analyses on the generated images, and discuss, with results, what challenge and how they are specifically addressing. Is there the **mode collapse issue**?

III. INCEPTION SCORE

A. Classifier Architecture Used

B. Results

Measure the inception scores i.e. we use the class labels to generate images in CGAN and compare them with the predicted labels of the generated images.

Also report the recognition accuracies on the MNIST real testing set (10K), in comparison to the inception scores.

Please measure and discuss the inception scores for the different hyper-parameters/tricks and/or architectures in Q2.

IV. RE-TRAINING THE HANDWRITTEN DIGIT CLASSIFIER

A. Results

Retrain with different portions and test BOTH fake and real queries. Please **vary** the portions of the real training and synthetic images, e.g. 10%, 20%, 50%, and 100%, of each.

B. Adapted Training Strategy

Using even a small number of real samples per class would already give a high recognition rate, which is difficult to improve. Use few real samples per class, and, plenty generated images in a good quality and see if the testing accuracy can be improved or not, over the model trained using the few real samples only. Did you have to change the strategy in training the classification network in order to improve the testing accuracy? For example, use synthetic data to initialise the network parameters followed by fine tuning the parameters with real data set. Or using realistic synthetic data based on the confidence score from the classification network pre-trained on real data. If yes, please then specify your training strategy in details. Analyse and discuss the outcome of the experimental result.

V. BONUS

This is an open question. Do you have any other ideas to improve GANs or have more insightful and comparative evaluations of GANs? Ideas are not limited. For instance,

- How do you compare GAN with PCA? We learnt PCA as another generative model in the Pattern Recognition module (EE468/EE9SO29/EE9CS729). Strengths/weaknesses?
- Take the pre-trained classification network using 100 to extract the penultimate layer's activations (embeddings) of 100 randomly sampled real test examples and 100 randomly sampled synthetic examples from all

the digits i.e. 0-9. Use an embedding method e.g. t-sne [1] or PCA, to project them to a 2D subspace and plot them. Explain what kind of patterns do you observe between the digits on real and synthetic data. Also plot the distribution of confidence scores on these real and synthetic sub-sampled examples by the classification network trained on 100 separate graphs. Explain the trends in the graphs.

- Can we add a classification loss (using the pre-trained classifier) to CGAN, and see if this improve? The classification loss would help the generated images maintain the class labels, i.e. improving the inception score. What would be the respective network architecture and loss function?

VI. REFERENCES

VII. APPENDIX