CIS 680 Homework #4

Nikolaos Kolotouros
nkolot@seas.upenn.edu

Karl Pertsch
pertsch@seas.upenn.edu

Oleh Rybkin
oleh@seas.upenn.edu

1. Autoencoders

Autoencoders [2] are a powerful class of neural networks that is used for various tasks in unsupervised setup. This assignment is a compliment to our main project, described in a separate report. We implement 3 different autoencoder networks and analyze their performance. Despite very limited data (87 images), the autoencoders learn to capture the input very well.

1.1. Task 1

Along with the guidelines, we implemented a convolutional autoencoder with the parameters given in Tab. 1. We show the training and test results of the implemnented AE in the Fig. 2.

We found overfitting to be a severe problem while training on CUFS dataset, which has 87 training and 100 test images. This explaining the architectural choices of the network. Specifically, we do not use any fully connected layers in between the encoder and decoder. The minimum spatial size our network uses is thus 2 × 2 pixels.

1.2. Task 2

We further train a variational autoencoder [3] on the same data. The results, in the Fig. 3, are more blurry and the reconstruction error for both train and test set is higher. We hypothesize that longer training would be needed to compensate for the random effect of sampling in the training procedure. We found that to produce good results without

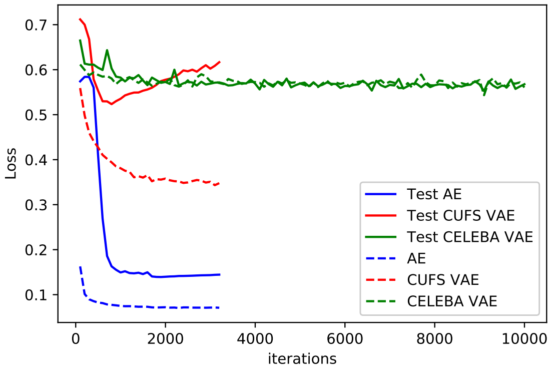
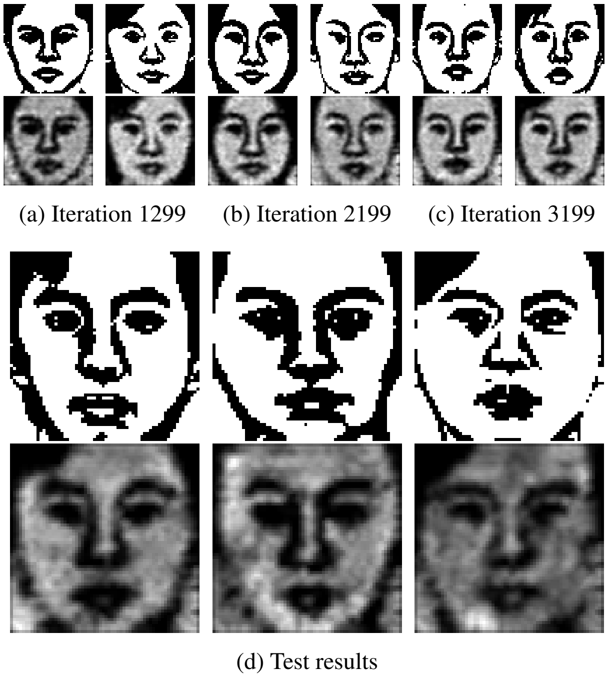


Figure 1: Losses.



(a) Iteration 1299 (b) Iteration 2199 (c) Iteration 3199

(d) Test results

Figure 2: Vanilla Autoencoder. Top: input, Bottom: output.

Input: 64 × 64 ×1 image
3 × 3 ×4 conv. stride 2
3 × 3 ×8 conv. stride 2
3 × 3 ×16 conv. stride 2
3 × 3 ×32 conv. stride 2
3 × 3 ×64 conv. stride 2
4 × 4 ×32 deconv. stride 2
4 × 4 ×16 deconv. stride 2
4 × 4 ×8 deconv. stride 2
4 × 4 ×4 deconv. stride 2
4 × 4 ×1 deconv. stride 2
L2 loss

Table 1: Vanilla Autoencoder.

overfitting the dimensionality of the latent space has to be small, so we use the dimensionality of 16. We report the results at the iteration 1299, as the network still starts to overfit after that point.

We note that the network quickly learns to satisfy the KL-divergence loss, and the reconstruction loss dominates the training process most of the time. The sampling procedure, however, has drastic influence on the training process, acting by itself as a regularizer.

Input: $64 \times 64 \times 1$ image
$3 \times 3 \times 4$ conv. stride 2
$3 \times 3 \times 8$ conv. stride 2
$3 \times 3 \times 16$ conv. stride 2
$3 \times 3 \times 32$ conv. stride 2
$3 \times 3 \times 64$ conv. stride 2
16 f.c. - mean
16 f.c. - std
sampling
$4 \times 4 \times 16$ deconv. stride 2
$4 \times 4 \times 8$ deconv. stride 2
$4 \times 4 \times 4$ deconv. stride 2
$4 \times 4 \times 2$ deconv. stride 2
$4 \times 4 \times 1$ deconv. stride 2
$4 \times 4 \times 1$ deconv. stride 2
CE and KL loss

Table 2: Variational Autoencoder.

1.3. Task 3

We build a bigger VAE and experiment on the CELEBA dataset. We only plot the test results as overfitting is no longer a problem on this data. The network is able to successfully capture most of the high-level information from the input image.

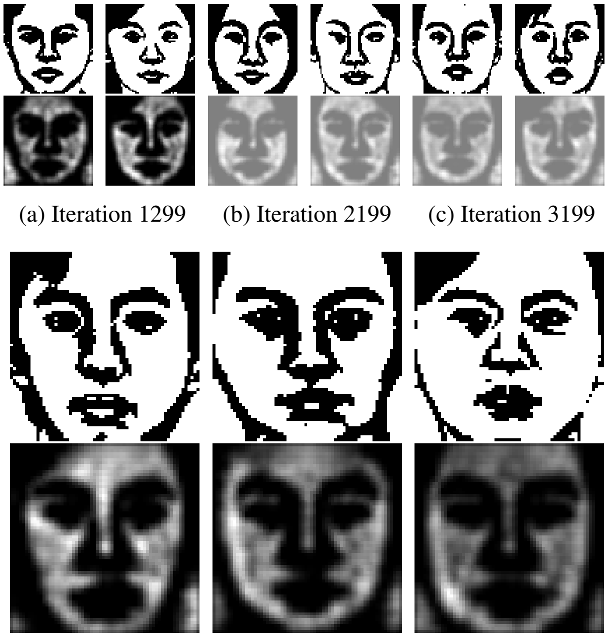
Note that the network has only minimal overfitting, which means that the performance could be improved further by longer training and increasing the depth and width of the networks. We found that increasing the depth hurt the performance at this point, as the depth is already prohibitively big. Further experiments would require switching to residual [1] architecture to combat this.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[2] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

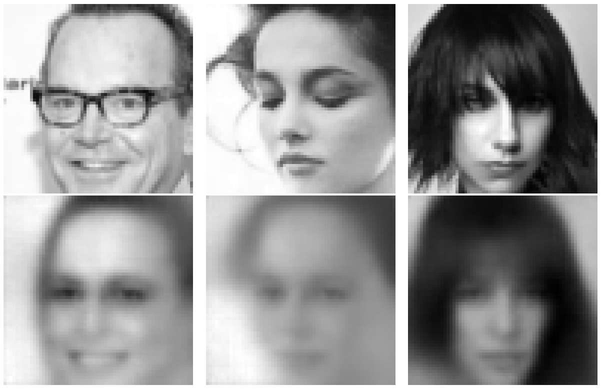
[3] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *ArXiv e-prints*, Dec. 2013.



(a) Iteration 1299 (b) Iteration 2199 (c) Iteration 3199

(d) Test results at iteration 1299

Figure 3: Variational Autoencoder. Top: input, Bottom: output.



(a) Test results

Figure 4: Variational Autoencoder. Top: input, Bottom: output.

Input: 64 × 64 ×1 image
3 × 3 ×32 conv. stride 2 3 × 3 ×32 conv. 3 × 3 ×64 conv. stride 2 3 × 3 ×64 conv. 3 × 3 ×128 conv. stride 2 3 × 3 ×128 conv. 3 × 3 ×256 conv. stride 2 3 × 3 ×256 conv. 3 × 3 ×512 conv. stride 2
512 f.c. - mean 512 f.c. - std sampling
4 × 4 ×256 deconv. stride 2 3 × 3 ×256 conv. 4 × 4 ×128 deconv. stride 2 3 × 3 ×128 conv. 4 × 4 ×64 deconv. stride 2 3 × 3 ×64 conv. 4 × 4 ×32 deconv. stride 2 3 × 3 ×32 conv. 4 × 4 ×16 deconv. stride 2 3 × 3 ×16 conv. 4 × 4 ×1 deconv. stride 2
CE and KL loss

Table 3: A bigger Variational Autoencoder.