EECS 598 Homework 3

Chenchen Ma

March 27, 2019

1 Text Classification using CNNs

1.1

$Y_{n,f} = \sum_c X_{n,c} *_{\text{filt}} W^{\text{conv}}_{f,c} + b_f$.

1.2

$Y_{n,f}$ is of size $(1, 1, H - H' + 1)$.

1.3

The size of the output of the pooling layer is $(N, F, 1)$.

1.4

I used GloVe embedding in my implementation. The accuracies under different conditions are listed below.

	Global average-pooling	Global max-pooling
Kernel size: 5	94.15%	95.33%
Kernel size: 7	93.40%	95.03%

Table 1: Test accuracy under different conditions

2 Siamese Networks for Learning Embeddings

2.1 Training loss

2.2

Figure 1: Train Loss History

2.3 Results on the Train Set


train, dissimilarity:1.79


train, dissimilarity:5.81


train, dissimilarity:4.10


train, dissimilarity:4.59


train, dissimilarity:4.14


train, dissimilarity:3.32


train, dissimilarity:1.99


train, dissimilarity:1.78


train, dissimilarity:2.53


train, dissimilarity:2.75


train, dissimilarity:3.87


train, dissimilarity:3.51

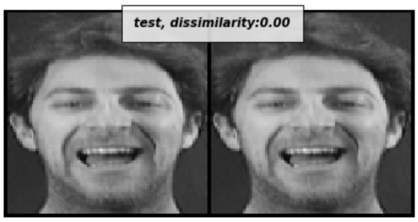

train, dissimilarity:2.42


train, dissimilarity:2.68

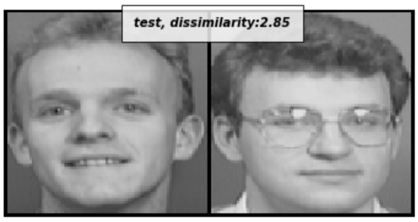

train, dissimilarity:4.62


train, dissimilarity:2.23

2.4 Results on the Test Set

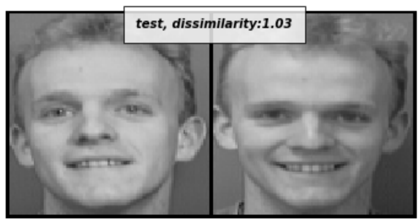

test, dissimilarity:0.00

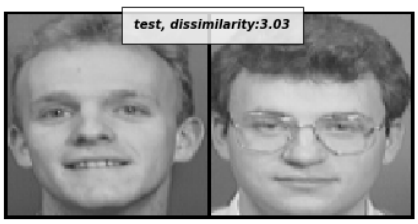

test, dissimilarity:1.06

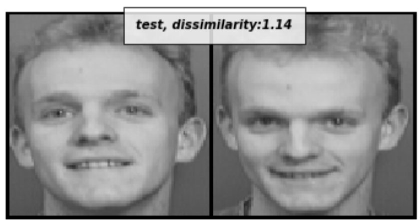

test, dissimilarity:2.85


test, dissimilarity:2.09


test, dissimilarity:3.12


test, dissimilarity:1.03


test, dissimilarity:3.03


test, dissimilarity:1.14


test, dissimilarity:1.02


test, dissimilarity:2.07


test, dissimilarity:3.15


test, dissimilarity:2.40

3 Conditional Variational Autoencoders

3.1

Derive the variational lower bound of a conditional variational autoencoder.

First note that by Bayes rule, we have

$$p_\theta(z|x,y) = \frac{p_\theta(z,x|y)}{p_\theta(x|y)}.$$

Thus,

$$\begin{aligned}
\log p_\theta(x|y) &= \mathbb{E}_{q_\phi(z|x,y)} \log p_\theta(x|y) \\
&= \mathbb{E}_{q_\phi(z|x,y)} \log \Big(\frac{p_\theta(z,x|y)}{p_\theta(z|x,y)} \frac{q_\phi(z|x,y)}{q_\phi(z|x,y)} \Big) \\
&= \mathbb{E}_{q_\phi(z|x,y)} \log \Big(\frac{p_\theta(x|y,z) p_\theta(z|y)}{p_\theta(z|x,y)} \frac{q_\phi(z|x,y)}{q_\phi(z|x,y)} \Big) \\
&= \mathbb{E}_{q_\phi(z|x,y)} \Big[\log \frac{q_\phi(z|x,y)}{p_\theta(z|x,y)} \Big] - \mathbb{E}_{q_\phi(z|x,y)} \Big[\log \frac{q_\phi(z|x,y)}{p_\theta(z|y)} \Big] + \mathbb{E}_{q_\phi(z|x,y)} \big[\log p_\theta(x|y,z) \big] \\
&= D_{KL}\big(q_\phi(z|x,y) || p_\theta(z|x,y)\big) - D_{KL}\big(q_\phi(z|x,y) || p_\theta(z|y)\big) + \mathbb{E}_{q_\phi(z|x,y)} \big[\log p_\theta(x|y,z) \big] \\
&\geq -D_{KL}\big(q_\phi(z|x,y) || p_\theta(z|y)\big) + \mathbb{E}_{q_\phi(z|x,y)} \big[\log p_\theta(x|y,z) \big],
\end{aligned}$$

since KL-divergence is non-negative.

3.2

Derive the analytical solution to the KL-divergence between two Gaussian distributions $D_{KL}\big(q_\phi(z|x,y)||p_\theta(z|y)\big)$. Assume that $p_\theta(z|y) \sim N(0, I)$ and $q_\phi(z|x,y) \sim N(\mu, \Sigma)$, where $\mu = (\mu_1, ..., \mu_J)^T$ and $\Sigma = \text{diag}(\sigma_1^2, ..., \sigma_J^2)$ are the outputs of the neural network that estimate the parameters of the posterior distribution $q_\phi(z|x,y)$, then

$$\begin{aligned}
D_{KL}\big(q_\phi(z|x,y)||p_\theta(z|y)\big) &= \mathbb{E}_{q_\phi(z|x,y)} \log \frac{q_\phi(z|x,y)}{p_\theta(z|y)} \\
&= \mathbb{E}_{q_\phi(z|x,y)} \log \Big(\frac{\exp\{-\frac{1}{2}(z-\mu)^T \Sigma^{-1} (z-\mu)\}}{\sqrt{|\Sigma|}} \times \frac{1}{\exp\{-\frac{1}{2} z^T z\}} \Big) \\
&= \mathbb{E}_{q_\phi(z|x,y)} \Big(-\frac{1}{2}(z-\mu)^T \Sigma^{-1} (z-\mu) + \frac{1}{2} z^T z - \frac{1}{2} \log |\Sigma| \Big) \\
&= \mathbb{E}_{q_\phi(z|x,y)} \Big(-\frac{1}{2} \sum_{j=1}^{J} \big((z_j - \mu_j)^2 / \sigma_j^2 - z_j^2 \big) - \frac{1}{2} \sum_{j=1}^{J} \log \sigma_j^2 \Big) \\
&= -\frac{1}{2} \sum_{j=1}^{J} \big(1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2 \big),
\end{aligned}$$

since $\mathbb{E}_{q_\phi(z|x,y)}(z_j - \mu_j)^2 = \sigma_j^2$ and $\mathbb{E}_{q_\phi(z|x,y)}(z_j^2) = \sigma_j^2 + (\mathbb{E}_{q_\phi(z|x,y)} z_j)^2 = \sigma_j^2 + \mu_j^2$.

3.3

Figure 2: Generated images by CVAE

4 Generative Adversarial Networks

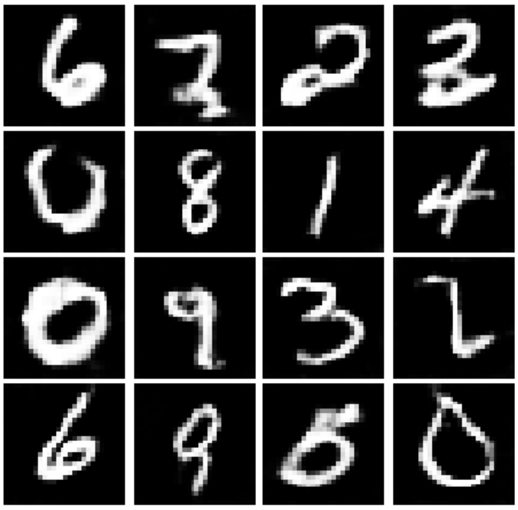


Figure 3: Generated images by DCGAN