Author: sheng wang
study group: mengdi han, ruoxi yu

Homework 3

EECS 498
March 22, 2019

Problem 1

1. express the output

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

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

3. the size of output pooling layer is N*F

4. cnn.py is implemented

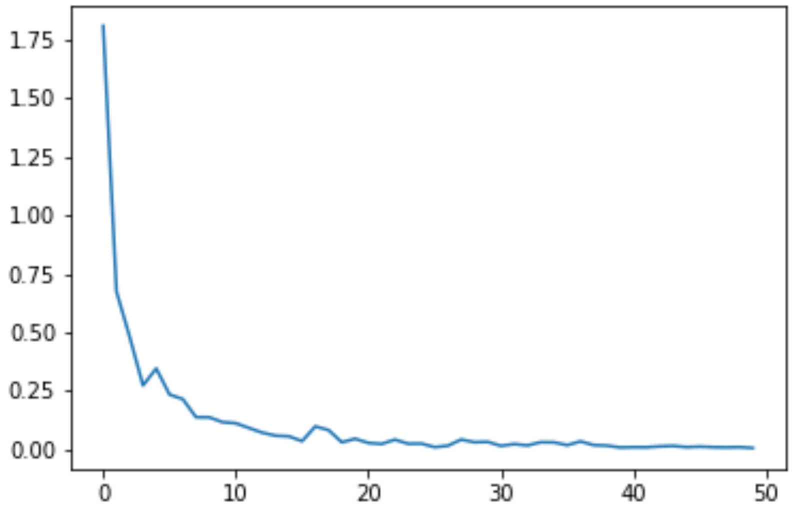
5. accuracy for the following architectural choices.

Pooling	Kernel size	test accuracy
Average	5	93.5
Max	5	94.15
Average	7	93.09
Max	7	94.08

Problem 2

1. implement the contrasive loss class ContrasiveLoss.

2. learning curve of losses on training dataset:



visualization result on training dataset:

visualization result on test dataset:

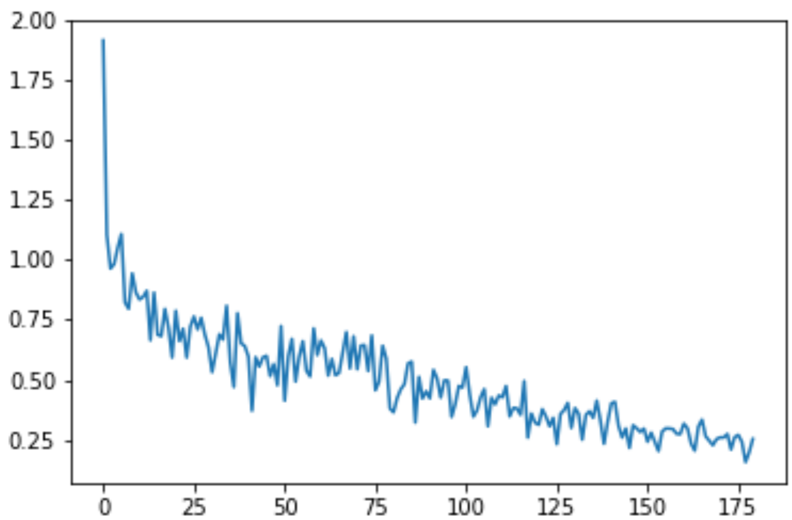
20 pairs of result on training data. The one filled by yellow denotes the same picture.

train_similarity									
4.42588	3.65022	2.23243	0.236141	2.62496	4.28673	1.92209	4.28673	4.00318	4.02741
2.83E-06	2.00242	2.22764	3.16553	3.31644	3.93388	4.46665	2.00376	3.00252	2.17041

20 pairs of result on test data. The one filled by yellow denotes the same picture.

test_similarity									
2.28493	2.83E-06	1.18627	0.517889	2.1472	3.06642	2.4211	2.35412	2.52136	0.923357
0.823287	3.46641	3.6382	3.02094	3.42243	3.39101	3.2378	0.650862	3.66825	3.01226

3. extra credit:
   learning curve of losses on training dataset on lfw-faces:



visualization result on training dataset:

cisualization result on test datset:

20 pairs of result on training data. The one filled by yellow denotes the same picture.

2.5351	2.32097	2.02878	1.95026	1.67795	1.78243	2.99391	1.66393	0.481988	2.6191
4.05632	3.66769	1.81854	1.30862	1.27843	2.73807	1.91488	2.42994	2.28421	1.66276

20 pairs of result on test data. The one filled by yellow denotes the same picture.

1.32866	1.55159	3.077	1.72855	0.930154	2.81809	1.55508	0.719239	2.37018	1.5025
1.87846	1.46348	1.61036	1.76122	2.21531	1.87846	1.5152	1.56832	2.47106	2.57719

Problem 3

1. derive the lower bound of a conditional variational autoencoder.

$$
\begin{aligned}
log_{p_\theta}(x|y) &= E_{q_\phi(z|x,y)}[log_{p_\theta}(x,y)] \qquad (\ p_\theta(x|y) \text{ does not depend on } z) \\
&= E_{q_\phi(z|x,y)}[log\frac{p_\theta(x|z,y)p_\theta(z|y)}{P_\theta(z|x,y)}] \qquad \text{(Bayes' rule)} \\
&= E_{q_\phi(z|x,y)}[log\frac{p_\theta(x|z,y)p_\theta(z|y)}{p_\theta(z|x,y)}\frac{q_\phi(z|x,y)}{q_\phi(z|x,y)}] \qquad \text{(multiply by constant)} \\
&= E_{q_\phi(z|x,y)}[logp_\theta(x|z,y)] - E_{q_\phi(z|x,y)}[log\frac{q_\phi(z|x,y)}{p_\theta(z|y)}] + E_{q_\phi(z|x,y)}[log\frac{q_\phi(z|x,y)}{p_\theta(z|x,y)}] \\
&\qquad \text{(logarithms)} \\
&= E_{q_\phi(z|x,y)}[logp_\theta(x|z,y)] - D_{KL}(q_\phi(z|x,y)||p_\theta(z|y)) + D_{KL}(q_\phi(z|x,y)||p_\theta(z|x,y)) \\
&\geq E_{q_\phi(z|x,y)}[logP_\theta(x|z,y)] - D_{KL}(q_\phi(z|x,y)||p_\theta(z|y)) \\
&\qquad \text{(KL divergence always} \geq 0)
\end{aligned}
$$

2. Derive the analytical solution to the KL divergence between two Gaussian distributions.

$D_{KL}(q_\phi(z|x,y) \,\|\, P_\theta(z|y))$

$= E_{q_\phi(z|x,y)}\left[\log(q_\phi(z|x,y)) - \log(P_\theta(z|y))\right]$

(Assume $q_\phi(z|x,y) \sim N(\mu_1, \Sigma_1)$ $\quad P_\theta(z|y) \sim N(\mu_2, \Sigma_2)$

then $\mu_2 = 0$, $\Sigma_2 = I$)

$\Rightarrow q_\phi(z|x,y) = \dfrac{1}{(2\pi)^{n/2}\det(\Sigma_1)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(z-\mu)^T \Sigma_1^{-1}(z-\mu)\right)$

$\therefore D_{KL}(q_\phi(z|x,y) \,\|\, P_\theta(z|y))$

$= \frac{1}{2} E_{q_\phi(z|x,y)}\left[-\log(\det \Sigma_1) - (z-\mu_1)^T\Sigma_1^{-1}(z-\mu_1) + \log(\overset{0}{\overset{\|}{\det \Sigma_2}}) + \overset{z^T z}{\overset{\|}{(z-\mu_2)^T \Sigma_2^{-1}(z-\mu_2)}}\right]$

(outer product property)

$= -\frac{1}{2}\log(\det \Sigma_1) + \frac{1}{2} E_{q_\phi(z|x,y)}\left[-\mathrm{tr}(\Sigma_1^{-1}(z-\mu_1)(z-\mu_1)^T + z^T z\right]$

(covariance matrix definition)

$= -\frac{1}{2}\log(\det \Sigma_1) + \frac{1}{2} E_{q_\phi(z|x,y)}\left[-\mathrm{tr}(\Sigma_1^{-1}\Sigma_1) + z^T z\right]$

$= -\frac{1}{2}\log(\det \Sigma_1) - \frac{1}{2} J + \frac{1}{2} E_{q_\phi(z|x,y)}(z^T z)$

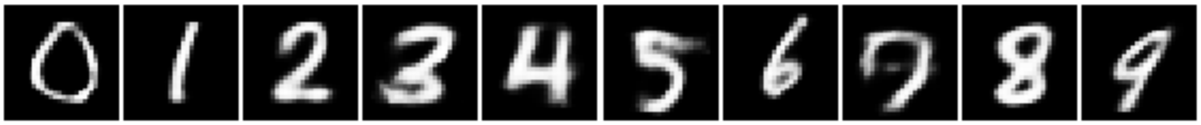
(Because $E(x^T x) = (Ex)^2 + Dx$)

$= -\frac{1}{2}\log(\det \Sigma_1) - \frac{1}{2} J + \frac{1}{2}\mu_1^T \mu + \frac{1}{2}\Sigma_1$

(Assuming Σ_1 is diagnal matrix, which means features are mutual independent)

$= -\frac{1}{2}\sum_{j=1}^{J}\left(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2\right)$

3. implement cave.py.
   Generated images with condition labels:



Problem 4

1. implement sample-noise function.

2. implement the build-discriminator function.

3. implement the build-generator function.

4. implement the get-optimizer function.

5. implement the bce-loss function.

6. implement the discriminator-loss function.

7. implement the generator-loss function.

8. train DCGAN.
   Here is some generated images with the increase of iteration, which are 250, 500, 1000 and finished training separately.