2019 Spring COM526000 Deep Learning - Homework 5

Variational Autoencoder for Image Reconstruction

105061210 楊雅婷

Problems

1. Image Preprocessing (file: data_prepro.py)

- Read the file `'emnist-balanced-train.csv'`.
- We know from the description of the dataset that there are 47 class according to `'0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabdefghnqrt'` characters in the data. However, we only want characters A~Z, which correspond to class 10~35, so I compare column 0 to 9 and 36, keeping data that have class 10~35.
- Transepose (flip) the data, so that it might look correct when plotting.
- Then adding some random noise to the original data and doing normalize (/255)
- Return the original data (and image), the data added noise (and image added noise), and the labels.
- Additional function '`print_img(vector)`' will show the image reshape from the vector. And the function '`label2char(label)`' will translate the input label to the corresponding character.

2. VAE model with L2 regularization (file: hw5_105061210.py)

- I construct a VAE model with an encoder, latent vector, and a decoder
- model:

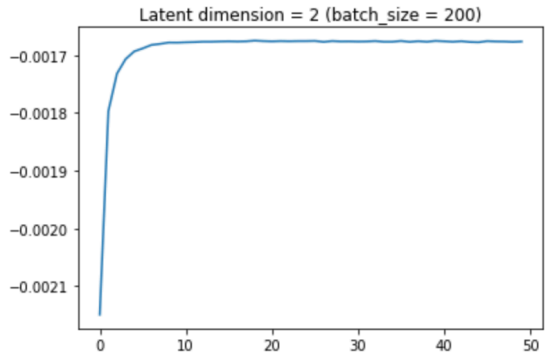
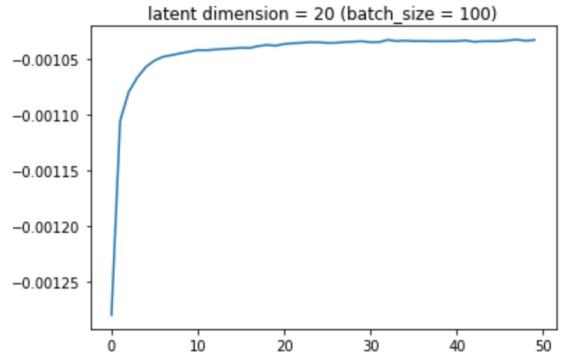
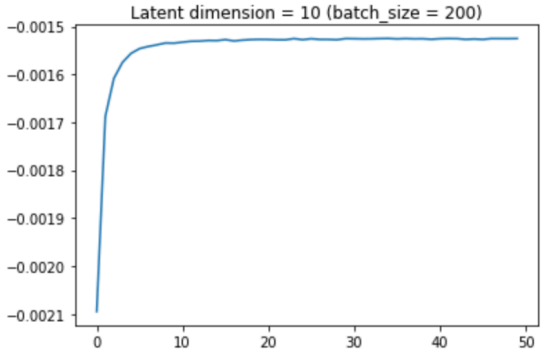
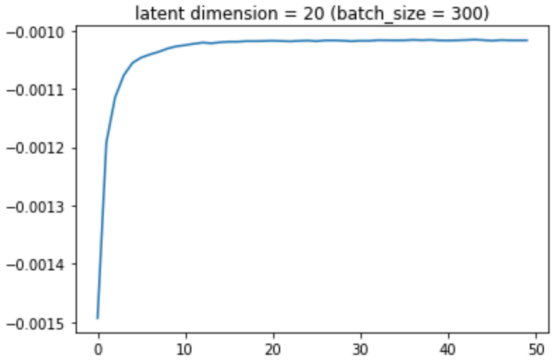
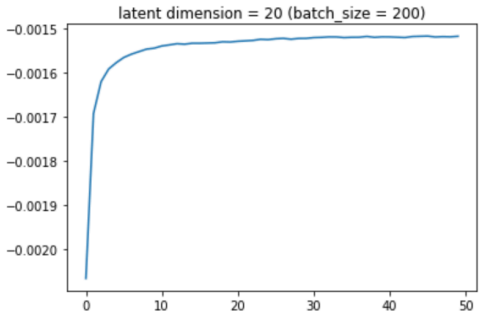
Encoder	Decoder
Input: (batch_size, 784)	Latent vector dimension: 2 or 10 or 20
Fully connected layer 1: 392 units	Fully connected layer 1: 100 units
Fully connected layer 2: 196 units	Fully connected layer 2: 196 units
Fully connected layer 3: 100 units	Fully connected layer 3: 392 units
Latent vector dimension: 2 or 10 or 20	Output: (batch_size, 784)
Activation function: relu	Activation function: relu

Optimizer: Adam
Code vector (z): $mu + \exp(sigma)^{1/2} \times \epsilon,\ \epsilon \sim N(0,1)$
Reconstruction loss: $\sum[x \times \log(y) + (1-x) \times \log(1-y)]$
KL-divergence: $-\frac{1}{2}\sum(1 + sigma - mu^2 - \exp(sigma))$

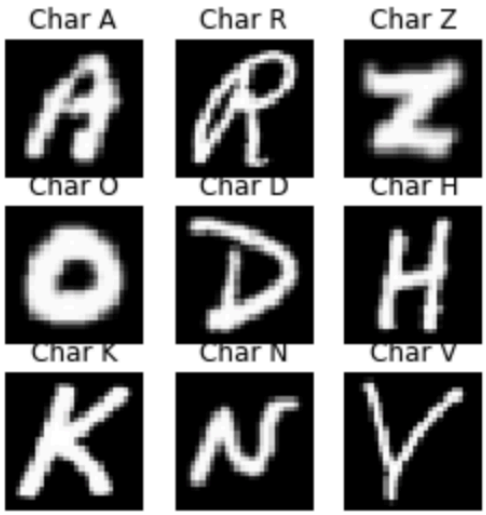
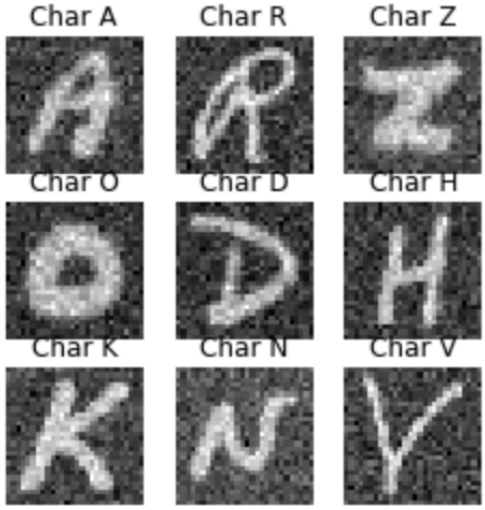
- Parameters:

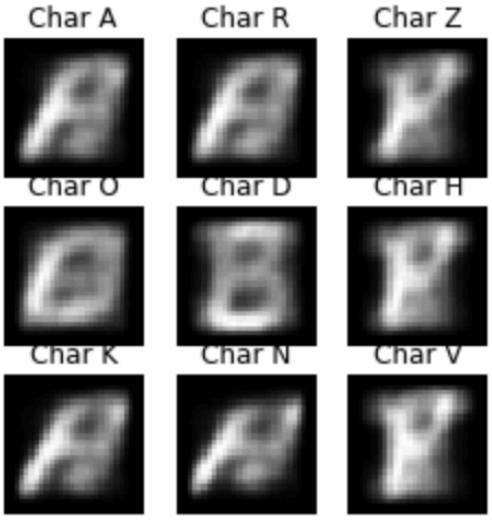
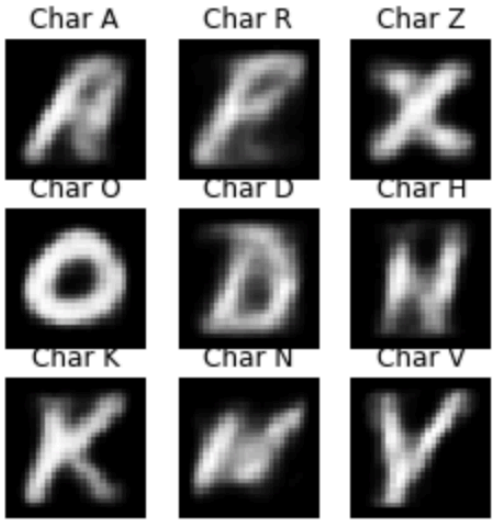
Model: vae_2			
Batch_size	Latent dimension	Learning rate	compare
200	2	0.003	worse
Model: vae_10			
Batch_size	Latent dimension	Learning rate	compare
200	10	0.003	
Model: vae_20			
Batch_size	Latent dimension	Learning rate	compare
200	20	0.003	better
Model: vae_20_100			
Batch_size	Latent dimension	Learning rate	compare
100	20	0.003	
Model: vae_20_300			
Batch_size	Latent dimension	Learning rate	compare
300	20	0.003	

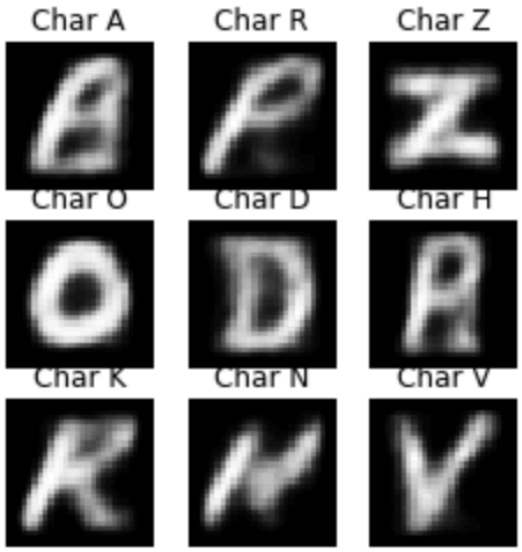
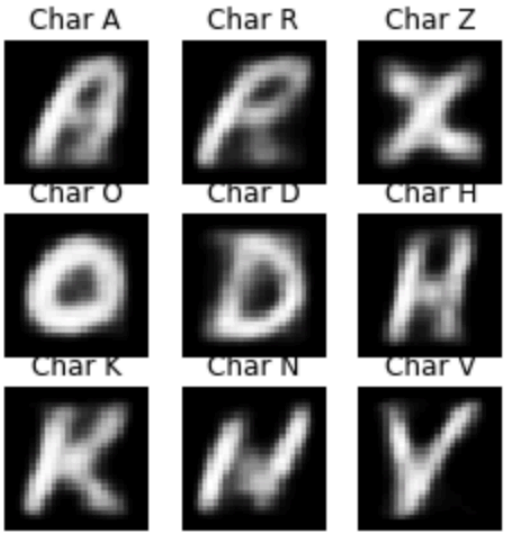
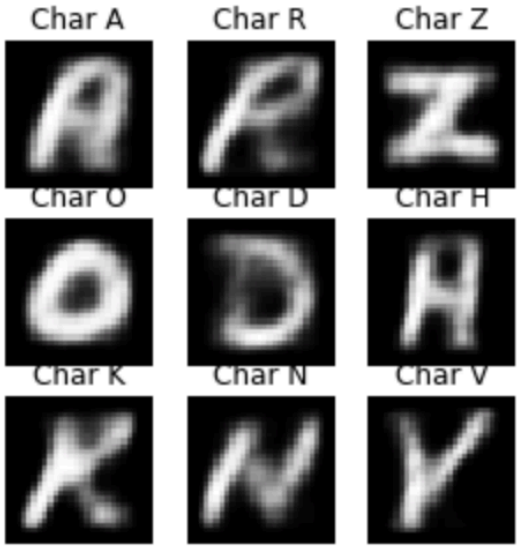
3. Learning curve

Model: vae_2	Model: vae_20_100
	
Model: vae_10	**Model: vae_20_300**
	
Model: vae_20	**note**
	1. Total loss = Reconstruction loss + KL-divergence + Regularization loss 2. Learning curve print: Total loss/784/batch_size

4. show some examples reconstructed from the decoder

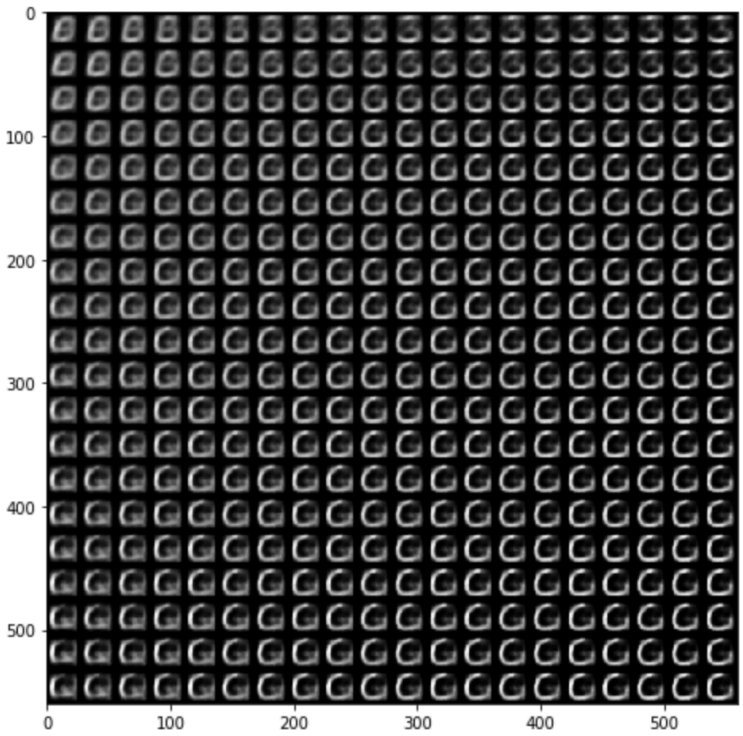
Original input image	Input image with noise
	

Model: vae_2	Model: vae_20_100
	
Model: vae_10	**Model: vae_20_300**

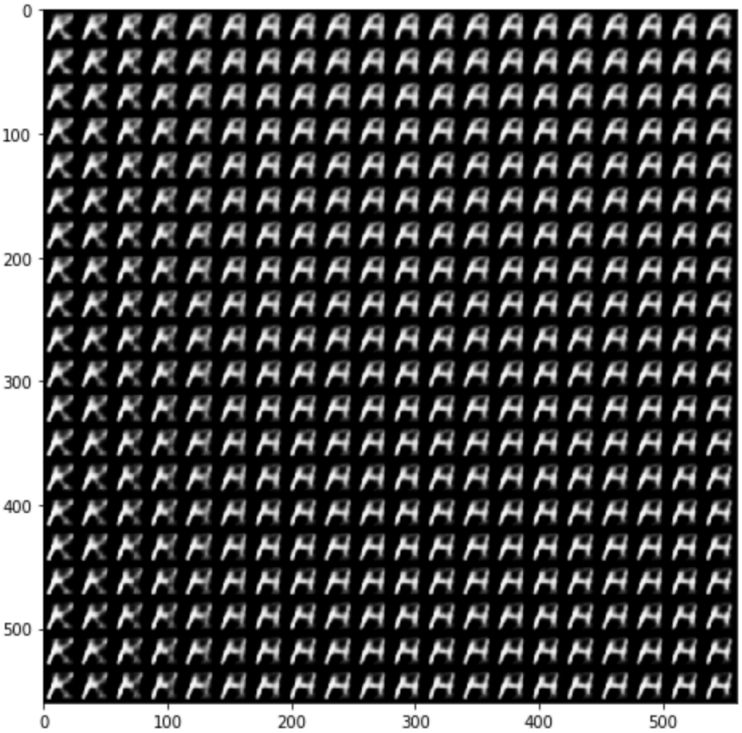
	
Model: vae_20	**note**
	

5. **plot the reconstructed images by varying the value of the latent variable.**

- **Use model vae_2**

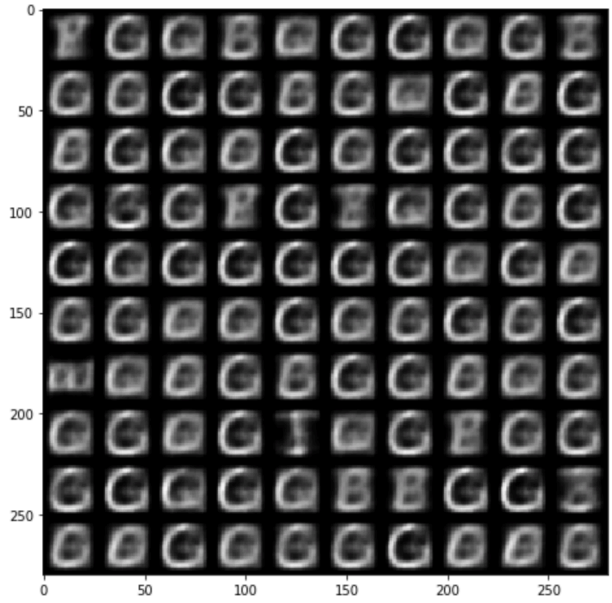


- **Use model vae_20**



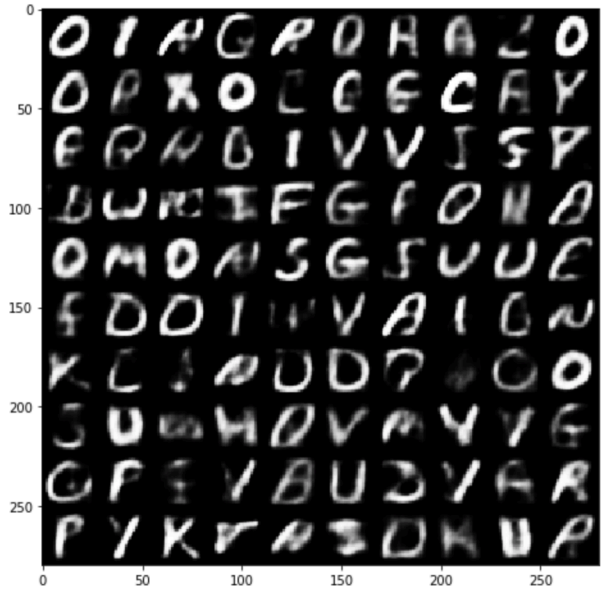
6. **Sample noise vectors from N(0,1) as latent variables and use the decoder to generate some images.**

- **Use model vae_2**



- **Use model vae_10**

- **Use model vae_20**



- Additional: reconstructed results from model vae_20