Variational Auto-Encoders for Downstream Classification

Xixi Chen xc2444
Zhengyuan Dong zd2216

December 15, 2019

1 Introduction

Variational auto-encoder (VAE) is an unsupervised dimension reduction technique built on top of neural networks. A VAE consists two complementary neural networks, an encoder and a decoder. The encoder takes in a high-dimensional input and maps it to a lower latent dimension, which will be fed into the decoder to generate a reconstructed value. In this way, the latent variables are supposed to preserve most of the variations while having a smaller dimension.

In this project, we mainly focus on implementing our own VAE through convolutional neural networks and applying it to downstream classification. We are also interested in the usefulness of our VAE for both small and large training samples. We believe that for small samples VAE should help the simple classifiers like logistic regressions and simple CNN to achieve better results than multi-layer CNN.

The data we used is Fashion-MNIST, which contains 28×28 gray-scale images of 10 fashion products. There are in total 60,000 training samples and 10,000 test samples.

2 Methods

2.1 Problems

We are interested in three problems:

1. How much variation of original images can latent variables capture?
2. How do simple classifiers combined with VAE compare to multilayer CNN?
3. As a dimension reduction method, does VAE mitigate the problem of overfitting for small samples?

2.2 Overall Structure

Overall, the project includes two parts. First, we implement VAE by modeling the "encoder" $q_\phi(z|x)$ using a convolutional neural network, which allows us to sample from the distribution similar to training data. Then We define the decoder $p_\theta(x|z)$ to generate images like the training ones to check if our encoder captures the variations of original images.

Second, we fix VAE to conduct classification by putting latent variables z generated by encoder into simple classifiers including logistic regression and simple CNN, and compare the results with directly training a multi-layer CNN classifier on the original data.

2.3 Encoder & decoder

Encoder: the probabilistic encoder $q_\phi(z|x)$ is a neural network which takes in $\mathbf{x}$ and outputs hidden representations $\mathbf{z}$. Assume the true posterior follows a Gaussian distribution with diagonal covariance, the encoder produces a Gaussian distribution with μ and σ from the given data $\mathbf{x}$. We mainly use 3 convolutional layers and 2 fully connected layers as our encoder.

Decoder: the probabilistic decoder $p_\theta(x|z)$ tries to reconstruct the original data through the given latent variables. In our example, the original image contains 28×28 pixels, if they are all black and white (0 and 1), the probability distribution of each pixel is a Bernoulli distribution. The decoder gets latent variables $\mathbf{z}$, and outputs 784 Bernoulli parameters $\mathbf{p}$. We use 3 deconvolutional layers and 3 fully connected layers as our decoder.

2.4 Loss

Variational Lower bound: the loss to optimize our VAE can be viewed as the negative log-likelihood with a regularizer. L_{rec} is the reconstruction loss (expected negative log-likelihood), which is estimated by sampling, because it cannot be integrated directly. The term encourages the decoder to reconstruct the data. L_{KL} is KL divergence which measures how much variation is lost.

$\mathrm{L}_{VAE} = L_{rec} + L_{KL}$
$L_{rec} = -E_q[logp_\theta(x|z)]$
$L_{KL} = D_{KL}(q_\phi(z|x) \,||\, p_\theta(z|x)) = -\frac{1}{2}\sum_{j=1}^{J}(1 + log((\sigma_j^2) - (\mu_j)^2 - (\sigma_j^2)$,
where J = *no. of dimensions of* $\vec{z}$

Sparse categorical cross entropy: it is used to optimize the classifier for multi-categorical classification.

2.5 How to train

VAE is unsupervised. We tried several possible combinations of encoders and decoders by optimizing the loss, and the reconstructed images all look like in similar quality, blurry but distinguishable. Since our goal is classification, we combine supervised and unsupervised learning, that is, we optimize loss of VAE first for each chosen model, then input latent variables to classifiers to get the accuracy, and we pick the model with the highest accuracy and our VAE. Then we tune the latent dimensions on small training samples through supervised learning, since we expect VAE with simple classifier is superior than CNN on small samples. Now we get our VAE with tuned latent dimensions.

After fixing VAE, we put derived latent variables into simple classifiers. By optimizing the sparse categorical cross entropy loss via stochastic gradient descent optimizer for logistic regression and Adam optimizer for simple CNN, we get trained classifiers and optimized accuracy.

3 Results

3.1 Visualizing in 2-D Latent Space

Below is a scatter plot for 5000 training samples using VAE with latent dimension 2. Note that with only 2 dimensions, the clusters have severe overlaps between "pullover" and "coat", "t-shirt/top" and "shirt".

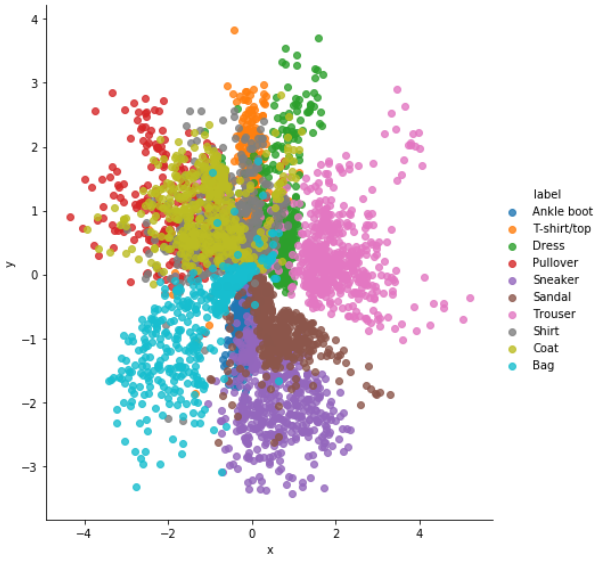

Figure 1: Visualizing 5000 samples in 2-D Latent Space

3.2 Reconstructed Images

The first 10 reconstructed images are shown below. We can see that with only 2 latent dimensions, the sneaker in (row 1, col 1) looks like an ankle boot, and the dress in (row 2, col 2) looks like T-shirt. Hence, dimension 2 is not sufficient. Reconstructed images with higher latent dimensions like 4 and 25 become more distinguishable. However, they are sill not able to capture the details for clothing, such as logos and stripes, which indicates VAE may be good for classification but it is not a powerful generative model.

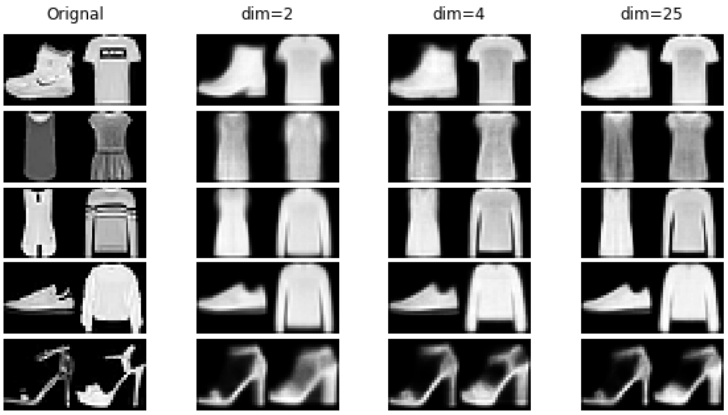

Figure 2: Reconstructed Images with Different Latent Dimensions

3.3 Model Accuracy for Different Latent Dimensions

The reconstructed images above show that 4 seems to be a good choice for latent dimension of VAE. We verify this by tuning the latent dimension and comparing the classification test accuracy. Note here we only used 100 samples to train our classifiers because we expect VAE may be superior in small dimensions. Table 1 shows that as the latent dimension increases, the test accuracy for both logistics and CNN decreases. Thus, dimension 4 is already sufficient for both image reconstruction and classification, which is a big compression of data.

Latent Dimensions of VAE	**VAE + Logistic**	**VAE + CNN**
4	57.4%	63.9%
9	54.3%	53.4%
16	50.1%	52.6%
25	38.5%	20.0%

Table 1: Accuracy vs Latent Dimensions (Only 100 Samples)

3.4 Test Accuracy vs Training Samples

After we fix VAE, We train 3 classifiers, i.e. VAE + logistic regression, VAE + simple CNN, and multi-layer CNN on different number of training samples ranging from 100 to 50000 (maximum due to Colab RAM limitation). From the table below, we can see simple CNN performs better than logistic regression by combing with VAE. But multi-layer CNN always performs the best.

However, we find, for small samples, training accuracy and test accuracy for VAE+logistic is quite close for small number of epochs, which means overfitting problem is mitigated by VAE. For VAE+CNN and multi-layer CNN, training and test accuracy have huge differences, sometimes as high as 20%. It is reasonable overfitting phenomena because for small training samples like 100 or 1000, number of parameters of CNN model is far more than the number of samples.

No. of Training Sample	**VAE+Logistic**	**VAE+CNN**	**Multi-layer CNN**
100	53.4%	66.2%	70.4%
1000	69.7%	77.3%	83.3%
2000	71.7%	78.0%	85.4%
5000	72.2%	79.3%	87.4%
10000	73.3%	81.0%	88.9%
20000	73.1%	80.8%	90.3%
50000	73.7%	81.4%	91.9%

Table 2: Test Accuracy vs Number of Training Samples

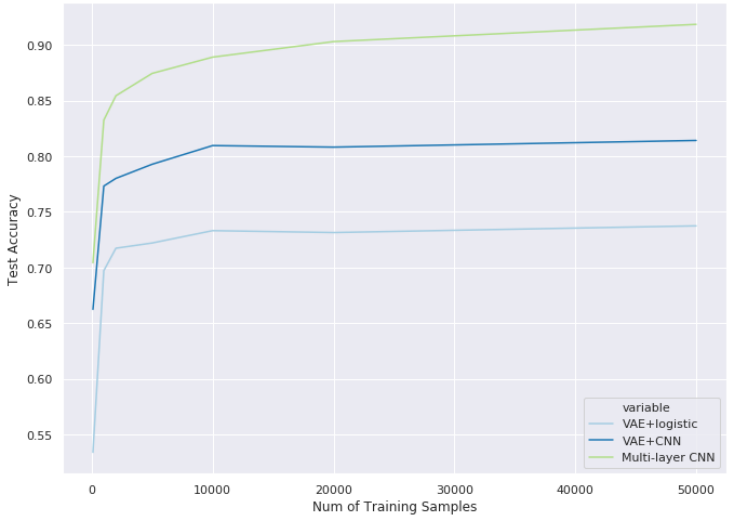

Figure 3: Test Accuracy vs Number of Samples

4 Conclusion

Variational Auto-Encoder is a cool idea based on Bayesian. It uses latent variables to represent image data by neural networks. We can easily identify what kind of fashion product is in our reconstructed images, which means VAE does explain the key variations of original images. For our encoder and decoder, 4 latent variables are sufficient to get satisfying accuracy for afterward classification which is a huge dimension reduction (from $28 \times 28 = 784$ to 4).

Random guess accuracy for fashion-mnist data is 10%. With only 100 samples, the test accuracy of VAE + logisitc regression can reach 53.4%, and VAE + simple CNN can reach 66.2%. With 50,000 training samples, VAE + simple CNN can reach 81.4% accuracy.

Unfortunately, VAEs do have limitations. Obviously they are not quite the state-of-the-art in generative models (blurry reconstructed images). Besides, the accuracy for downstream classification still cannot compare with multi-layer CNN no matter for small samples or large samples. Though logistic regression benefits from the reduction of input dimensions with consistent training and test accuracy, its accuracy is still lower than those using CNN.

References

[1] Diederik, P.K. & Max, W. (2013) Auto-Encoding Variational Bayes.