Mario Fiorino 1871233

Report

Running experiments related:

Auto-Encoding Variational Bayes

1. Introduction

An autoencoder is a neural network aim to learn parameters to reconstructed the output as close as possible to the input: learning the identity function seems trivial, but with the addition of constraints on the network (e.g., limiting the number of hidden neurons, or, regularization) we can extract information about the structure of the dataset (since the network is forced to represent the same data in a lower dimensional space).
More specifically, a classical autoencoder is composed of two modules:
encoder module, that maps input(e.g., an image) to a latent vector space (of lower dimensional);
decoder module, from latent space reconstructed the input (with the same original dimensions).
Minimizing a loss function, that evaluates the difference between input and output, allows the network to learn "meaningful" latent representations of the data.

Variational autoencoder (VAE) has a similar architecture to autoencoder, but, unlike the latter , in VAE the encoder turns the input in parameters of a statistical distribution, precisely two vectors: a vector of means μ a vector of variances Σ , of gaussian distribution (learned by training). This mean and variance are used to randomly sample a vector *z* from the latent space. Then, the decoder will map *z* back to the original input image.
The VAE loss function combines reconstruction loss (e.g., MSE), that forces the decoded samples t to match the initial inputs, and regularizer loss that forces to generate latent vectors that follow a gaussian distribution (this ensure that the latent space has "required" properties).
More formal: the loss function, decomposed into single terms depending on i-th datapoint, is defined:

$$l_i(\phi, \theta) = -\mathbb{E}_{z\sim q_\theta(z|x_i)}[\log p_\phi(x_i|z)] + \mathbb{KL}(q_\theta(z|x_i)||p(z))$$

ϕ,θ denote the parameters of networks; encoder's distribution is denoted by $q_\theta(z\,|\,x_i)$; the decoder by $p_\phi(x_i\,|\,z)$. The first term of equation is expected negative log-likelihood of the i-th datapoint. The second term is the Kullback-Leibler divergence between the encoder's distribution $q_\theta(z|x_i)$ and p(z); the latter (prior probability over the latent variables) is usually set p(z)=Normal(0,1).

In simple word, we can say: VAE aim to learn a distribution that approximates (maximize the likelihood) the distribution of the dataset training.

2. Experiment - Generative model

Variational autoencoder model can be used for generate new samples (how : sampling from latent space z-distribution, trained appropriately, and feed it to the decoder). In particular, in this example, VAE is used for generate new images of face starting from a dataset of celebrate faces,

Dataset trainng [1]: CelebFaces Attributes (CelebA) Dataset, from Kaggle, is a large-scale face attributes dataset with 202.599 celebrity images. Examples:

Resolution input image: 128x128 pixel, 3 channel (RGB).

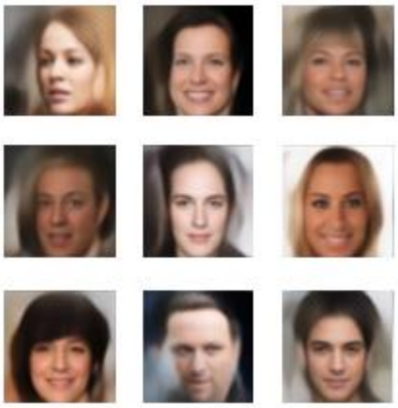
Model Architecure

- Encoder : consists of a stack of 4 convolutional layers, number of filter for each layer respectively [32, 64, 64, 64], LeakyReLU as activation function for each layer. Followed by a flatten layer; ended with with two dense fully connected layer : one for a mean vector μ and another one for variance vector Σ (Note: in code is used logarithm of the variance vector, this makes training easier).
- Sampling function : from the normal distribution obtained by the outputs of the encoder: mean vector μ and variance vector Σ, is sampled latent vector (input of decoder) : $Z=\mu+ \Sigma\, \varepsilon$ where ε is sampled from a multivariate standard normal distribution (`mean=0, stddev =1).`
- Decoder : consists of a stack of 4 Conv2DTranspose layer (this performs the inverse convolution), number of filter for each layer respectively [64, 64, 32, 3]. LeakyReLU as activation function for each layer except for the last one that use Sigmoid (to restrict the outputs between 0 and 1).

Loss function is a sum of Root Mean Square Error (RMSE) and KL Divergence.
A hyperparameter-weight is multiplied to the RMSE loss (for ensure the quality of the reconstructed images).

Training
Adam optimization algorithm (lr = 0.0005).
Number of epochs = 400. Batch size = 512.
Time = 10.5 hours (with GPU)

Evalutation
For generate new samples, some image of the dataset training are passed through the VAE trained. The ouptups are displayed below.
More convincing artificial faces generated:



.

3. Experiment - Anomaly Detection in Dataset

An anomaly is a pattern, in a dataset, that does not conform to an expected behavior (i.e., some data points are significantly different from the remaining dataset). Anomaly detection is the process of finding anomalies in a dataset. VAEs are also used in for this task (i.e., use the probability that a certain data is generated from learned latent distribution to evaluate the anomaly). In this example the problem is applied to recognise credit card fraud transactions: the goal is to learn a model from real data tranactions and then isolate non-compliant samples and consider them as anomalies.

Datasets training [2] contains transactions made by credit cards. This dataset presents transactions that occurred in two days in 2013, where we have 492 frauds out of 284,807 transactions; dataset is highly unbalanced, the frauds 0.17% of all transactions. It contains only numerical input variables.

Model Architecure

- Encoder : consists of a stack of 3 dense fully connected layer, respectively inter_dims = [20, 10, 8] ; ended with with two dense fully connected layer : one for a mean vector μ and another one for variance vector Σ (Note: in code is used logarithm of the variance)
- Sampling function : $Z=\mu+\Sigma\,\varepsilon$ where ε is sampled from a multivariate standard normal distribution (`mean=0, stddev =1`).
- Decoder: consists of a stack of 3 dense fully connected layer, respectively inter_dims = [8, 10,20] (output gets original dimension of input).

Loss function is a sum of Root Mean Square Error (RMSE) and KL Divergence.

Training

NAdam optimization algorithm (lr = 0.001). Number of epochs = 1000. Batch size = 128 Time = 7.5 hours (GPU)

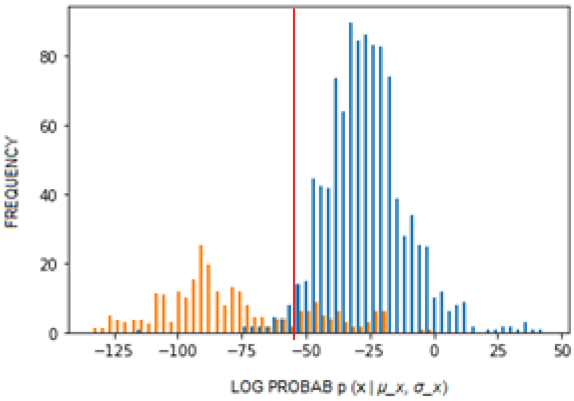
Evalutation

For testing, it is used the reconstruction probability: this is essentially the probability of the data being generated from a given latent variable drawn from the learned posterior distribution. It is computed in this way: given a data point x^i, first, using it as input to trained VAE's encoder. Then, in this experiment, is drawn 100 samples from a normal distribution and reparameterize it with the mean and standard deviation of the posterior distribution. Then, trained decoder is used on the samples. By using the mean and standard deviation of the resulting distribution for every sample we can calculate the reconstruction probability:

$$\frac{1}{100} * \sum_{l=1}^{100} p_\theta\ (x^i \mid \mu_{x(i,l)}, \sigma_{x(i,l)})$$

When it is "high" then the chance of reconstructing it from the encoder is high, so it is not an anomaly. When the reconstruction probability is " low", it is ananomaly.

The histogram of log of the reconstruction probability, on dataset test, displayed below show that: real transactions have higher reconstruction probability than fraud transactions, and how these two class are clearly separeted in most cases (exception made for some overlaps). In red is drawn the boundary value (caculated with statistical procedures on data) to evaluate whether datapoint is an anomaly or not; when the reconstruction probability of a data point is above the line, it is considered not anomaly. Note also: there are few cases of False valuated Positive and very few cases of True_Negative.



4. Conclusion

VAEs can be applied to various real-world problems. In these two experiments we have exploited the VAE's capacity to model complex distributions in a latent space in order to generate optimal artificial samples and detect anomalies in dataset.

Reference

- Diederik P Kingma; Welling, Max (2013). "Auto-Encoding Variational Bayes". arXiv:1312.6114
- Welling, Max; Kingma, Diederik P. (2019). "An Introduction to Variational Autoencoders".
- An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability.
- [1] Dataset CelebFaces: https://www.kaggle.com/jessicali9530/celeba-dataset
- VAE generative model of faces,code : https://github.com/dhanushkamath/VariationalAutoencoder
- [2] Dataset transactions https://www.kaggle.com/mlg-ulb/creditcardfraud
- VAE model detect fraudulent,code: https://github.com/koenvandevelde/fd-autoencoder