Task 6 : Autoencoders

Sanjeev Narayanan

21st March 2018

Introduction

Autoencoders are a specific type of feed-forward neural networks where the input is the same as the output. They compress the input into a lower-dimensional code and then reconstruct the output from this representation. The code is a compact "summary" or "compression" of the input, also called the latent-space representation .

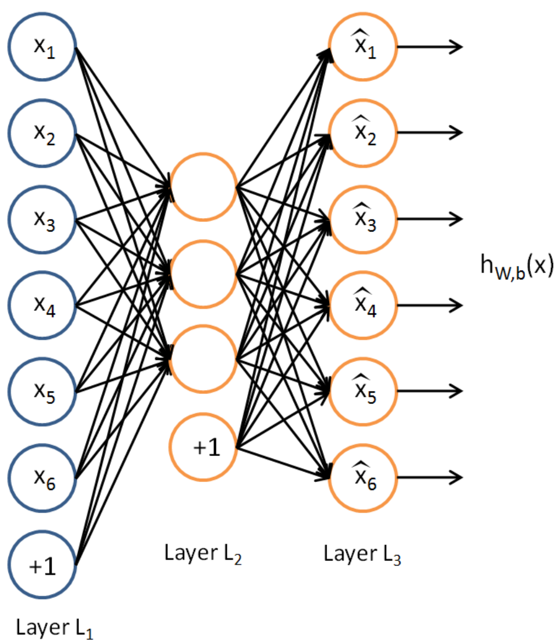

Figure 1: Autoencoder

It learns a function $h_{w,b}(x)$, which in other words is an approximation to the identity function, so as to output $\hat{x}$ that is similar to x. The autoencoder function is mainly a dimensionality reduction technique that has the following properties [1][2]:

- They compress data based on what they are trained on - they are customized for a particular data set

- Lots of loss involved
- Require only raw data as input - unsupervised learning

There are 4 hyperparameters that are tuned in an autoencoder

- Nodes in middle layer
- Number of filter layers
- Number of nodes in a layer
- Loss function

Variational Autoencoders

Variational autoecoders (VAEs) are generative models where we have X samples that are distributed according to some distribution $P_{gen}(X)$ and the goal is to learn a model P such that P approximates to P_{gen}.
The main idea here is that the observed data are controlled by a set of latent variables $\mathcal{Z}$. VAEs deal with bayesian inferencing. One of the first steps involved here is computing the marginal likelihood, from which th posterior distribution is obtained. The inference problem is to compute thie conditional distribution of latent variables, given observed data. In variational inference, $\mathcal{L}$ is a family of distribution in $\mathcal{Z}$ where each member in $\mathcal{Z}$ is a candidate approximation to $\mathcal{L}$. The goal is to find the best approximation, that satisfies the optimization problem

$$q^*(x) = \underset{q(z) \in \mathcal{L}}{\text{argmax}} KL[q(z) \| p(z|x)] \tag{1}$$

Unfortunately, this is hard since

$$\mathcal{D}_{KL}[q(z) \| p(z|x)] = \mathbb{E}[log(q(z)] - \mathbb{E}[log(p(z|x)] \tag{2}$$

$$= \mathbb{E}[log(q(z)] - \mathbb{E}[log(p(z,x)] + logp(x) \tag{3}$$

Since log p(x) can't be computed easily, we use Evidence of Lower Bound (ELBO)

$$ELBO(q) = \mathbb{E}[log(p(z,x)] - \mathbb{E}[log(q(z)] \tag{4}$$

ELBO is maximized so as to minimize KL-divergence.
Assume a set of latent variables z with pdf $p(z)$ which can be sampled easily and a family $f(z;\theta)$, where θ is a parameter that needs to be optimized. Variational autoencoders attempt to approximately optimize

$$p(x) = \int_z p(x|z;\theta)p(z)dz = \int_z p_\theta(x|z)p(z)dz \tag{5}$$

In order to achieve this, VAEs must describe how o define latent variables and deal with integration over z. VAEs often use a Gaussian distribution for $p_\theta(x|z).Since$f(z;θ) defines a multilayered neural network, we can approximate the integral as $p(x) = \frac{1}{n}\sum_{i=i}^{n} p(x|z_i)$.

VAEs attempt to sample the values of z that are likely to have produced x and then uses them to compute $p_\theta(x)$. Since the posterior $p_\theta(z|x)$ is intractable, VAEs use $q_\phi(z|x)$ that learns to approximate $p_\theta(z|x)$. In order to achieve this, a key operation of VAE is explored - the relationship between $E_{z\sim q_\phi(z)}$ $[p_\theta(x|z)]$ and $p(x)$

$$\mathcal{D}_{KL}[q(z)\|p_\theta(z|x)] = E_{z\sim q_\phi(z)}[\log q_\phi(z) - \log p_\theta(x|z)] \tag{6}$$

Applying bayes rule to the above equation,

$$\mathcal{D}_{KL}[q(z)\|p_\theta(z|x)] = E_{z\sim q_\phi(z)}[[\log q_\phi(z) - \log p_\theta(x|z) - \log p_\theta(x)] + \log p(x) \tag{7}$$

This can be simplified to

$$\log p(x) - \mathcal{D}_{KL}[q(z)\|p_\theta(z|x)] = E_{z\sim q_\phi(z)}[\log p_\theta(x|z)] - \mathcal{D}_{KL}[q(z)\|p_\theta(z)] \tag{8}$$

This is the basis of VAE. We can go a step further and define the lower bound $\mathcal{L}$ on the likelihood of the data [3]

$$\mathcal{L}(\theta,\phi,x) = -\mathcal{D}_{KL}[q(z)\|p_\theta(z)] + E_{z\sim q_\phi(z)}[logp_\theta(x|z)] \tag{9}$$

Here, the first term on RHS is a regularization term, second being a reconstruction term, x fixed and q_ϕ beig any distribution. We need to find $q_\phi(z|x)$ such that the regularization term tends to 0. An interesting thing to understand from this is that once the VAE is trained, there is no need for an encoder pathway, since the VAE can generate its own data from $\mathcal{N}(0,I)$. This is why the VAE is often shown by

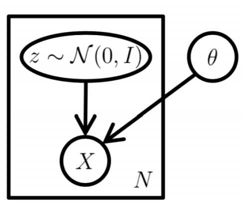

Figure 2: Variational Autoencoder

The reparametrization trick is often employed in VAEs where, instead of computing $z \sim q_{\mu,\sigma}(\epsilon) = \mathcal{N}(\mu,\sigma)$, we can represent the same as $z = g_{\mu,\sigma}(\epsilon) = \mu + \epsilon\cdot\sigma$. The main advantage of this method is that it helps in backpropagation, moving the samples out of the graph and the input at training time. Also it helps to train encoder and decoder pathways simultaneously with the objective function in Equation 9.[3][4]

One of the most popular algorithms employed is the AVEB algorithm [5]. It consists of autoecoding the ELBO, black-box variational inference approach and the reparametrization-based low-variance gradient estimator. It optimizes the auto-encoding ELBO using black-box variational inference with the reparametrized gradient estimator. This algorithm is applicable to any deep generative model p_θ with latent varables differentiable in θ

References

[1] QV Le, G Brain, and G Inc. A tutorial on deep learning part 2: Autoencoders. *Convolutional Neural Networks and Recurrent Neural Networks*, 119, 2015.

[2] Arden Mertat. Applied deep learning - part 3: Autoencoders.

[3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[4] David Meyer. Notes on variational autoencoder.

[5] Shengjia Zhao, Jiaming Song, and Stefano Ermon. The information-autoencoding family: A lagrangian perspective on latent variable generative modeling.