Deep Learning

Vazgen Mikayelyan

YSU, Krisp

December 10, 2020

Contractive Autoencoders

- A contractive autoencoder makes this encoding less sensitive to small variations in its training dataset.

Contractive Autoencoders

- A contractive autoencoder makes this encoding less sensitive to small variations in its training dataset.
- This is accomplished by adding a regularizer, or penalty term, to whatever cost or objective function the algorithm is trying to minimize.

Contractive Autoencoders

- A contractive autoencoder makes this encoding less sensitive to small variations in its training dataset.
- This is accomplished by adding a regularizer, or penalty term, to whatever cost or objective function the algorithm is trying to minimize.
- The end result is to reduce the learned representation's sensitivity towards the training input.

Contractive Autoencoders

Let f is our encoder, g is the decoder and D is our training dataset. In the previous cases we minimize this kind of loss function:

$$\sum_{x \in D} L\left(x, g\left(f\left(x\right)\right)\right).$$

Contractive Autoencoders

Let f is our encoder, g is the decoder and D is our training dataset. In the previous cases we minimize this kind of loss function:

$$\sum_{x \in D} L\left(x, g\left(f\left(x\right)\right)\right).$$

In the case of contractive autoencoders we will minimize this one

$$\sum_{x \in D} \left(L\left(x, g\left(f\left(x\right)\right)\right) + \lambda \left\|J_f\left(x\right)\right\|_F^2\right),$$

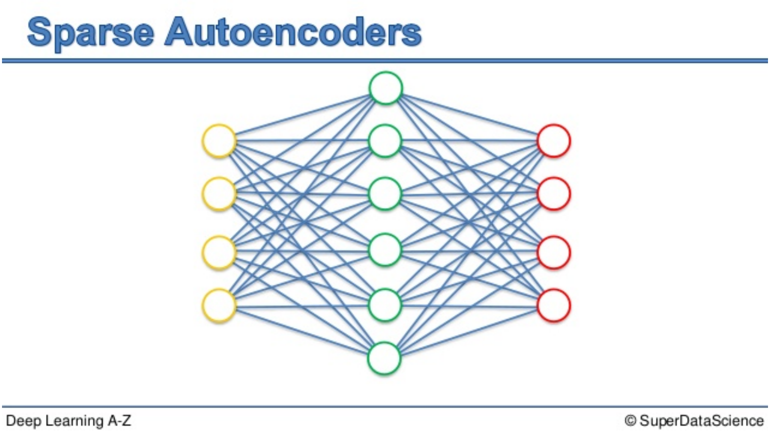
where the added summand is the square of Frobenius norm of the following Jacobian matrix:

$$\left[J_f\left(x\right)\right]_{i,j} = \frac{\partial f_j\left(x\right)}{\partial x_i}$$

i.e.

$$\left\|J_f\left(x\right)\right\|_F^2 = \sum_{i,j} \left(\frac{\partial f_j\left(x\right)}{\partial x_i}\right)^2.$$

Sparse Autoencoders



Sparse Autoencoders

- Sparse autoencoders have hidden nodes greater than input nodes. They can still discover important features from the data.

Sparse Autoencoders

- Sparse autoencoders have hidden nodes greater than input nodes. They can still discover important features from the data.
- Sparsity penalty is introduced on the hidden layer. This is to prevent output layer copy input data. This prevents overfitting.

Sparse Autoencoders

Let f is our encoder, g is the decoder and D is our training dataset, which has n samples. Denote

$$\rho_j = \frac{1}{n} \sum_{x \in D} f_j\left(x\right).$$

Let f is our encoder, g is the decoder and D is our training dataset, which has n samples. Denote

$$\rho_j = \frac{1}{n} \sum_{x \in D} f_j(x).$$

We would like to (approximately) enforce the constraint $\rho_j = \rho$, where ρ is a sparsity parameter, typically a small value close to zero (say $\rho = 0.05$).

Sparse Autoencoders

Let f is our encoder, g is the decoder and D is our training dataset, which has n samples. Denote

$$\rho_j = \frac{1}{n} \sum_{x \in D} f_j(x).$$

We would like to (approximately) enforce the constraint $\rho_j = \rho$, where ρ is a sparsity parameter, typically a small value close to zero (say $\rho = 0.05$). To achieve this we will minimize the following loss function

$$\sum_{x \in D} L(x, g(f(x))) + \lambda \sum_j KL(\rho || \rho_j),$$

Sparse Autoencoders

Let f is our encoder, g is the decoder and D is our training dataset, which has n samples. Denote

$$\rho_j = \frac{1}{n} \sum_{x \in D} f_j(x).$$

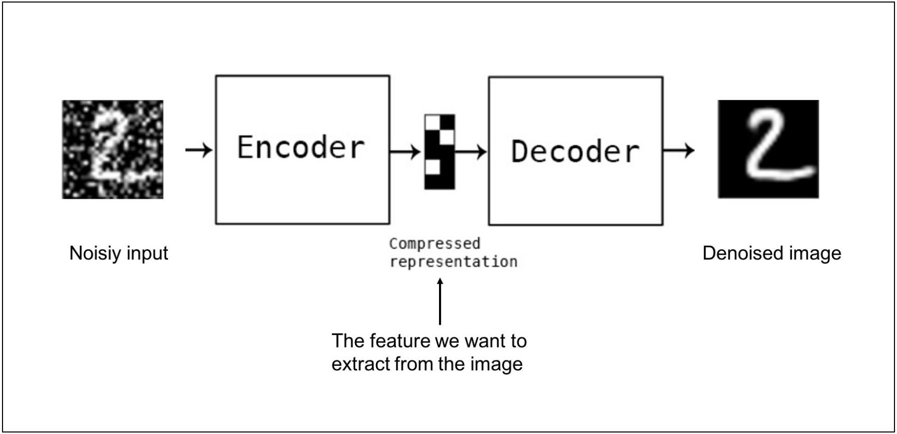
We would like to (approximately) enforce the constraint $\rho_j = \rho$, where ρ is a sparsity parameter, typically a small value close to zero (say $\rho = 0.05$). To achieve this we will minimize the following loss function

$$\sum_{x \in D} L(x, g(f(x))) + \lambda \sum_j KL(\rho || \rho_j),$$

where

$$KL(\rho || \rho_j) = -\rho \log \frac{\rho_j}{\rho} - (1 - \rho) \log \frac{1 - \rho_j}{1 - \rho}.$$

Denoising Autoencoders



Denoising Autoencoders

- Denoising autoencoders create a corrupted copy of the input by introducing some noise.

Denoising Autoencoders

- Denoising autoencoders create a corrupted copy of the input by introducing some noise.
- This helps to avoid the autoencoders to copy the input to the output without learning features about the data.

Denoising Autoencoders

- Denoising autoencoders create a corrupted copy of the input by introducing some noise.
- This helps to avoid the autoencoders to copy the input to the output without learning features about the data.
- The model learns a vector field for mapping the input data towards a lower dimensional manifold which describes the natural data to cancel out the added noise.

Variational Autoencoders

- Can we generate content with autoencoders?

Variational Autoencoders

- Can we generate content with autoencoders?
- Variational autoencoder can be defined as being an autoencoder whose training is regularised to avoid overfitting and ensure that the latent space has good properties that enable generative process.

Variational Autoencoders

- Can we generate content with autoencoders?
- Variational autoencoder can be defined as being an autoencoder whose training is regularised to avoid overfitting and ensure that the latent space has good properties that enable generative process.
- Instead of encoding an input as a single point, we encode it as a distribution over the latent space.

Variational Autoencoders

- Denote distribution of latent representations by $p(z)$.

- Denote distribution of latent representations by $p(z)$.
- Denote distribution of ideal encoder by $p(z|x)$.

- Denote distribution of latent representations by $p(z)$.
- Denote distribution of ideal encoder by $p(z|x)$.
- We want to find a neural network $q_w(z|x)$ such that

$$q_w(z|x) \approx p(z|x).$$

Variational Autoencoders

- Denote distribution of latent representations by $p(z)$.
- Denote distribution of ideal encoder by $p(z|x)$.
- We want to find a neural network $q_w(z|x)$ such that

$$q_w(z|x) \approx p(z|x).$$

- Let minimize the function

$$L(w) = KL(q_w(z|x) || p(z|x)).$$

Note that

$$KL\left(q_w\left(z|x\right)||p\left(z|x\right)\right) = \mathbb{E}_{q_w(z|x)}\left[\log \frac{q_w\left(z|x\right)}{p\left(z|x\right)}\right]$$

Variational Autoencoders

Note that

$$KL\left(q_w\left(z|x\right)||p\left(z|x\right)\right) = \mathbb{E}_{q_w(z|x)}\left[\log\frac{q_w\left(z|x\right)}{p\left(z|x\right)}\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log q_w\left(z|x\right)\right] - \mathbb{E}_{q_w(z|x)}\left[\log p\left(z|x\right)\right]$$

Variational Autoencoders

Note that

$$KL\left(q_w\left(z|x\right)||p\left(z|x\right)\right) = \mathbb{E}_{q_w(z|x)}\left[\log \frac{q_w\left(z|x\right)}{p\left(z|x\right)}\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log q_w\left(z|x\right)\right] - \mathbb{E}_{q_w(z|x)}\left[\log p\left(z|x\right)\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log q_w\left(z|x\right)\right] - \mathbb{E}_{q_w(z|x)}\left[\log \frac{p\left(x|z\right)p\left(z\right)}{p\left(x\right)}\right]$$

Note that

$$KL\left(q_w\left(z|x\right)||p\left(z|x\right)\right) = \mathbb{E}_{q_w(z|x)}\left[\log\frac{q_w\left(z|x\right)}{p\left(z|x\right)}\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log q_w\left(z|x\right)\right] - \mathbb{E}_{q_w(z|x)}\left[\log p\left(z|x\right)\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log q_w\left(z|x\right)\right] - \mathbb{E}_{q_w(z|x)}\left[\log\frac{p\left(x|z\right)p\left(z\right)}{p\left(x\right)}\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log\frac{q_w\left(z|x\right)}{p\left(z\right)}\right] - \mathbb{E}_{q_w(z|x)}\left[\log p\left(x|z\right)\right] + \mathbb{E}_{q_w(z|x)}\left[\log p\left(x\right)\right]$$

Note that

$$KL\left(q_w\left(z|x\right)||p\left(z|x\right)\right) = \mathbb{E}_{q_w(z|x)}\left[\log\frac{q_w\left(z|x\right)}{p\left(z|x\right)}\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log q_w\left(z|x\right)\right] - \mathbb{E}_{q_w(z|x)}\left[\log p\left(z|x\right)\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log q_w\left(z|x\right)\right] - \mathbb{E}_{q_w(z|x)}\left[\log\frac{p\left(x|z\right)p\left(z\right)}{p\left(x\right)}\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log\frac{q_w\left(z|x\right)}{p\left(z\right)}\right] - \mathbb{E}_{q_w(z|x)}\left[\log p\left(x|z\right)\right] + \mathbb{E}_{q_w(z|x)}\left[\log p\left(x\right)\right]$$

$$= KL\left(q_w\left(z|x\right)||p\left(z\right)\right) - \mathbb{E}_{q_w(z|x)}\left[\log p\left(x|z\right)\right] + \log p\left(x\right).$$

Note that

$$KL\left(q_w\left(z|x\right)||p\left(z|x\right)\right) = \mathbb{E}_{q_w(z|x)}\left[\log\frac{q_w\left(z|x\right)}{p\left(z|x\right)}\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log q_w\left(z|x\right)\right] - \mathbb{E}_{q_w(z|x)}\left[\log p\left(z|x\right)\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log q_w\left(z|x\right)\right] - \mathbb{E}_{q_w(z|x)}\left[\log\frac{p\left(x|z\right)p\left(z\right)}{p\left(x\right)}\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log\frac{q_w\left(z|x\right)}{p\left(z\right)}\right] - \mathbb{E}_{q_w(z|x)}\left[\log p\left(x|z\right)\right] + \mathbb{E}_{q_w(z|x)}\left[\log p\left(x\right)\right]$$

$$= KL\left(q_w\left(z|x\right)||p\left(z\right)\right) - \mathbb{E}_{q_w(z|x)}\left[\log p\left(x|z\right)\right] + \log p\left(x\right).$$

So we have to model the distribution $p\left(x|z\right)$ too, which will be our decoder:

Note that

$$KL\left(q_w\left(z|x\right)||p\left(z|x\right)\right) = \mathbb{E}_{q_w(z|x)}\left[\log\frac{q_w\left(z|x\right)}{p\left(z|x\right)}\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log q_w\left(z|x\right)\right] - \mathbb{E}_{q_w(z|x)}\left[\log p\left(z|x\right)\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log q_w\left(z|x\right)\right] - \mathbb{E}_{q_w(z|x)}\left[\log\frac{p\left(x|z\right)p\left(z\right)}{p\left(x\right)}\right]$$

$$= \mathbb{E}_{q_w(z|x)}\left[\log\frac{q_w\left(z|x\right)}{p\left(z\right)}\right] - \mathbb{E}_{q_w(z|x)}\left[\log p\left(x|z\right)\right] + \mathbb{E}_{q_w(z|x)}\left[\log p\left(x\right)\right]$$

$$= KL\left(q_w\left(z|x\right)||p\left(z\right)\right) - \mathbb{E}_{q_w(z|x)}\left[\log p\left(x|z\right)\right] + \log p\left(x\right).$$

So we have to model the distribution $p\left(x|z\right)$ too, which will be our decoder:

$$\underset{w,w'}{\operatorname{argmin}}\left(KL\left(q_w\left(z|x\right)||p\left(z\right)\right) + \mathbb{E}_{q_w(z|x)}\left[\|x - f_{w'}\left(z\right)\|^2\right]\right)$$

Problem: how to evaluate the term $KL\left(q_w\left(z|x\right)||p\left(z\right)\right)$?

Variational Autoencoders

Problem: how to evaluate the term $KL\left(q_w\left(z|x\right)||p\left(z\right)\right)$?
Solution: we will assume that $q_w = \mathcal{N}\left(\mu, \Sigma\right)$ and $p = \mathcal{N}\left(0, I\right)$.

Variational Autoencoders

The model is then trained as follows:

- first, the input is encoded as distribution over the latent space,

Variational Autoencoders

The model is then trained as follows:

- first, the input is encoded as distribution over the latent space,
- second, a point from the latent space is sampled from that distribution,

Variational Autoencoders

The model is then trained as follows:

- first, the input is encoded as distribution over the latent space,
- second, a point from the latent space is sampled from that distribution,
- third, the sampled point is decoded and the reconstruction error can be computed,

The model is then trained as follows:

- first, the input is encoded as distribution over the latent space,
- second, a point from the latent space is sampled from that distribution,
- third, the sampled point is decoded and the reconstruction error can be computed,
- finally, the reconstruction error is backpropagated through the network.

Variational Autoencoders

The model is then trained as follows:

- first, the input is encoded as distribution over the latent space,
- second, a point from the latent space is sampled from that distribution,
- third, the sampled point is decoded and the reconstruction error can be computed,
- finally, the reconstruction error is backpropagated through the network.

Assumptions

- In practice, the encoded distributions are chosen to be normal so that the encoder can be trained to return the mean and the covariance matrix that describe these Gaussians.

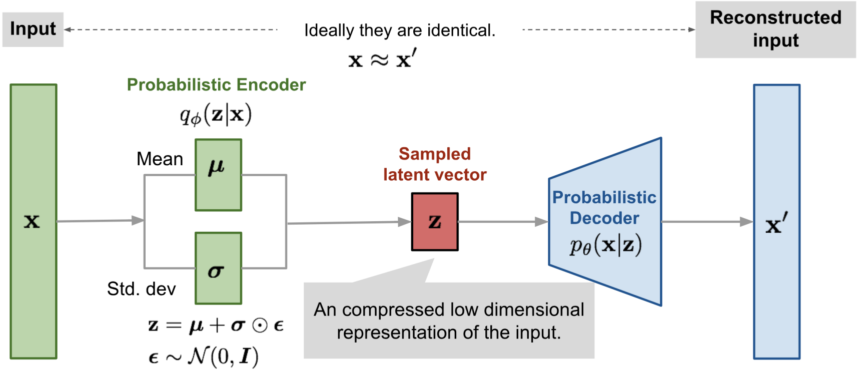
The model is then trained as follows:

- first, the input is encoded as distribution over the latent space,
- second, a point from the latent space is sampled from that distribution,
- third, the sampled point is decoded and the reconstruction error can be computed,
- finally, the reconstruction error is backpropagated through the network.

Assumptions

- In practice, the encoded distributions are chosen to be normal so that the encoder can be trained to return the mean and the covariance matrix that describe these Gaussians.
- The distributions returned by the encoder are enforced to be close to a standard normal distribution.

Variational Autoencoders



Let f is our encoder, g is the decoder and D is our training dataset. In this case we will minimize the following loss function

$$\sum_{x \in D} L\left(x, g\left(f\left(x\right)\right)\right) + \lambda KL\left(N\left(\mu, \Sigma\right) || N\left(0, I\right)\right).$$