Variational Auto-Encoders

Ravi Kothari, Ph.D.
ravi.kothari@ashoka.edu.in

"You use a glass mirror to see your face; you use works of art to see your soul" – George Bernard Shaw

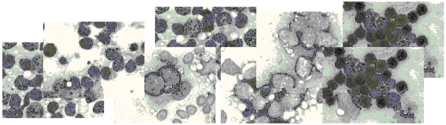
Unsupervised Learning

Unsupervised Learning

- We now turn our attention to unsupervised learning i.e. no explicit class label is present. For example, we might be given the following unlabeled images,

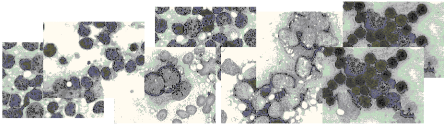
Unsupervised Learning

- We now turn our attention to unsupervised learning i.e. no explicit class label is present. For example, we might be given the following unlabeled images,



Unsupervised Learning

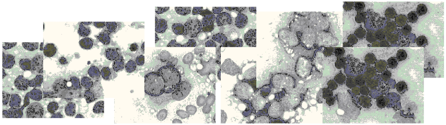
- We now turn our attention to unsupervised learning i.e. no explicit class label is present. For example, we might be given the following unlabeled images,



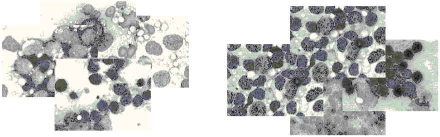
- As we have seen, we can cluster the data such that members of a cluster share more in common than with members of another cluster

Unsupervised Learning

- We now turn our attention to unsupervised learning i.e. no explicit class label is present. For example, we might be given the following unlabeled images,



- As we have seen, we can cluster the data such that members of a cluster share more in common than with members of another cluster



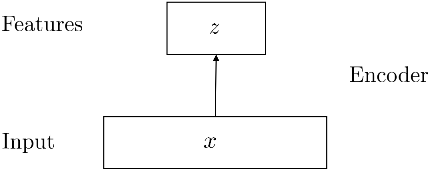
Lower Dimensional Feature Representations

Lower Dimensional Feature Representations

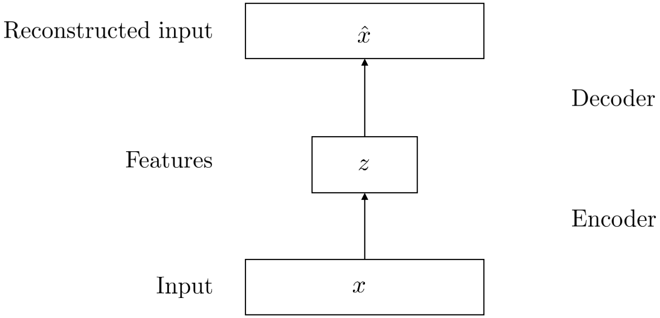
- An interesting application of unsupervised learning is discovering lower dimensional feature representations

Lower Dimensional Feature Representations

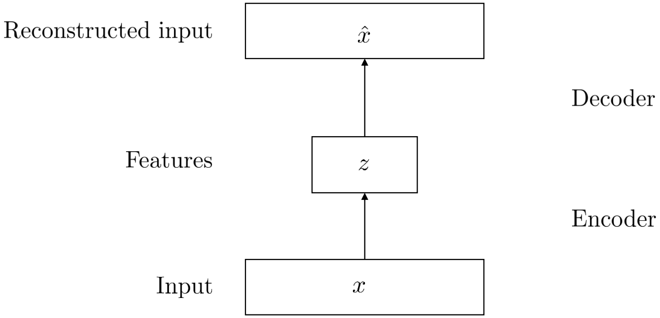
- An interesting application of unsupervised learning is discovering lower dimensional feature representations



Learning Lower Dimensional Feature Representations



Learning Lower Dimensional Feature Representations



- Sometimes, when very little labeled data is available, one can use features obtained using this auto-encoder approach and a following layer (trained using the limited data that is available) for pattern classification

Latent Variables

Latent Variables

- Typically, latent variables might encode specific characteristics. For example, the nose in a face or the lips or the eyes or ...

Latent Variables

- Typically, latent variables might encode specific characteristics. For example, the nose in a face or the lips or the eyes or ...
- Each latent variable is represented by a specific value
- In a variational auto-encoder we think of each latent variable being represented by a probability distribution

Latent Variables

- Typically, latent variables might encode specific characteristics. For example, the nose in a face or the lips or the eyes or ...
- Each latent variable is represented by a specific value
- In a variational auto-encoder we think of each latent variable being represented by a probability distribution
  - This forces the space of latent variables to be continuous

Latent Variables

- Typically, latent variables might encode specific characteristics. For example, the nose in a face or the lips or the eyes or ...
- Each latent variable is represented by a specific value
- In a variational auto-encoder we think of each latent variable being represented by a probability distribution
  - This forces the space of latent variables to be continuous
- Decoding will require sampling from each latent variable distribution to generate a vector which can be used by the decoder

Probabilistic Model Perspective

Probabilistic Model Perspective

- $$p(z|x) = \frac{p(x|z)\,p(z)}{p(x)} \tag{1}$$

Probabilistic Model Perspective

- $$p(z|x) = \frac{p(x|z)\,p(z)}{p(x)} \tag{1}$$

- This requires evaluating $p(x)$ which is,

$$p(x) = \int p(x|z)\,p(z)\,dz \tag{2}$$

which is almost always intractable

Probabilistic Model Perspective

- $$p(z|x) = \frac{p(x|z)\,p(z)}{p(x)} \quad (1)$$

- This requires evaluating $p(x)$ which is,

  $$p(x) = \int p(x|z)\,p(z)\,dz \quad (2)$$

  which is almost always intractable
- Two possibilities – use Monte Carlo or use variational inference. Here, we will use variational inference

Probabilistic Model Perspective

- $$p(z|x) = \frac{p(x|z)\,p(z)}{p(x)} \tag{1}$$
- This requires evaluating $p(x)$ which is,
$$p(x) = \int p(x|z)\,p(z)\,dz \tag{2}$$
which is almost always intractable
- Two possibilities – use Monte Carlo or use variational inference. Here, we will use variational inference
- So, we approximate $p(z|x)$ with a tractable distribution, $q(z|x)$

Probabilistic Model Perspective

- $$p(z|x) = \frac{p(x|z)\,p(z)}{p(x)} \tag{1}$$
- This requires evaluating $p(x)$ which is,
$$p(x) = \int p(x|z)\,p(z)\,dz \tag{2}$$
which is almost always intractable
- Two possibilities – use Monte Carlo or use variational inference. Here, we will use variational inference
- So, we approximate $p(z|x)$ with a tractable distribution, $q(z|x)$
- To be able to make inferences, $q(z|x)$ must be close to $p(z|x)$. So, we can choose $q(z|x)$ to minimize,
$$\min \mathrm{KL}(q(z|x) \parallel p(z|x)) \tag{3}$$

$$
\begin{aligned}
\mathrm{KL}(q(z|x) \parallel p(z|x)) &= -\sum q(z|x) \log \frac{p(z|x)}{q(z|x)} \\
&= -\sum q(z|x) \log \frac{\frac{p(x,z)}{p(x)}}{q(z|x)} \\
&= -\sum q(z|x) \left[\log \frac{p(x,z)}{q(z|x)} - \log p(x)\right] \\
&= -\sum q(z|x) \log \frac{p(x,z)}{q(z|x)} + \sum q(z|x) \log p(x) \\
&= -\sum q(z|x) \log \frac{p(x,z)}{q(z|x)} + \log p(x) \qquad (4)
\end{aligned}
$$

So,

$$
\log p(x) = \mathrm{KL}(q(z|x) \parallel p(z|x)) + \underbrace{\sum q(z|x) \log \frac{p(x,z)}{q(z|x)}}_{\text{Variational Lower Bound (VLB)}} \qquad (5)
$$

$p(x)$ is a constant and minimizing $\mathrm{KL}(\cdot)$ is the same as maximizing VLB. Since $\mathrm{KL}(\cdot)$ is non-negative, VLB cannot be larger than $\log p(x)$

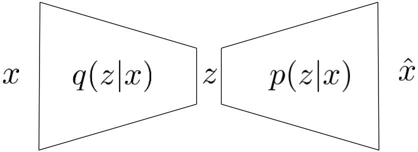
$$
\begin{aligned}
\sum q(z|x) \log \frac{p(x,z)}{q(z|x)} &= \sum q(z|x) \log \frac{p(x|z)\, p(z)}{q(z|x)} \\
&= \sum q(z|x) \left[\log p(x|z) + \log \frac{p(z)}{q(z|x)}\right] \\
&= \underbrace{\sum q(z|x) \log p(x|z)}_{E_{q(z|x)} \log p(x|z)} + \underbrace{\sum q(z|x) \log \frac{p(z)}{q(z|x)}}_{\mathrm{KL}(q(z|x) \| p(z))} \\
&= E_{q(z|x)} \log p(x|z) - \mathrm{KL}(q(z|x) \parallel p(z)) \qquad (6)
\end{aligned}
$$

So, the second term minimizes the distance of the distribution with my assumed distribution and the first term minimizes the loss between x and $\hat{x}$

$$
\begin{aligned}
\sum q(z|x) \log \frac{p(x,z)}{q(z|x)} &= \sum q(z|x) \log \frac{p(x|z)\, p(z)}{q(z|x)} \\
&= \sum q(z|x) \left[\log p(x|z) + \log \frac{p(z)}{q(z|x)} \right] \\
&= \underbrace{\sum q(z|x) \log p(x|z)}_{E_{q(z|x)} \log p(x|z)} + \underbrace{\sum q(z|x) \log \frac{p(z)}{q(z|x)}}_{\mathrm{KL}(q(z|x) \| p(z))} \\
&= E_{q(z|x)} \log p(x|z) - \mathrm{KL}(q(z|x) \parallel p(z)) \qquad (6)
\end{aligned}
$$

So, the second term minimizes the distance of the distribution with my assumed distribution and the first term minimizes the loss between x and $\hat{x}$

$$
\begin{aligned}
\sum q(z|x) \log \frac{p(x,z)}{q(z|x)} &= \sum q(z|x) \log \frac{p(x|z)\, p(z)}{q(z|x)} \\
&= \sum q(z|x) \left[\log p(x|z) + \log \frac{p(z)}{q(z|x)} \right] \\
&= \underbrace{\sum q(z|x) \log p(x|z)}_{E_{q(z|x)} \log p(x|z)} + \underbrace{\sum q(z|x) \log \frac{p(z)}{q(z|x)}}_{\mathrm{KL}(q(z|x) \| p(z))} \\
&= E_{q(z|x)} \log p(x|z) - \mathrm{KL}(q(z|x) \parallel p(z)) \qquad (6)
\end{aligned}
$$



So, the second term minimizes the distance of the distribution with my assumed distribution and the first term minimizes the loss between x and $\hat{x}$

VAE

VAE

- The encoder part of the auto-encoder outputs the parameters of the distribution as opposed to a specific latent state vector

VAE

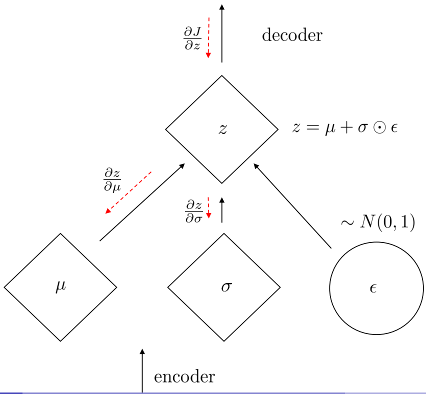
- The encoder part of the auto-encoder outputs the parameters of the distribution as opposed to a specific latent state vector
- The decoder part of the auto-encoder samples from this distribution to generate $\hat{x}$

VAE

- The encoder part of the auto-encoder outputs the parameters of the distribution as opposed to a specific latent state vector
- The decoder part of the auto-encoder samples from this distribution to generate $\hat{x}$
- Since the decoder uses a random sampling from that distribution, we use a reparameterization trick,

VAE

- The encoder part of the auto-encoder outputs the parameters of the distribution as opposed to a specific latent state vector
- The decoder part of the auto-encoder samples from this distribution to generate $\hat{x}$
- Since the decoder uses a random sampling from that distribution, we use a reparameterization trick,



Demo

https://www.siarez.com/projects/variational-autoencoder

http://vdumoulin.github.io/morphing_faces/