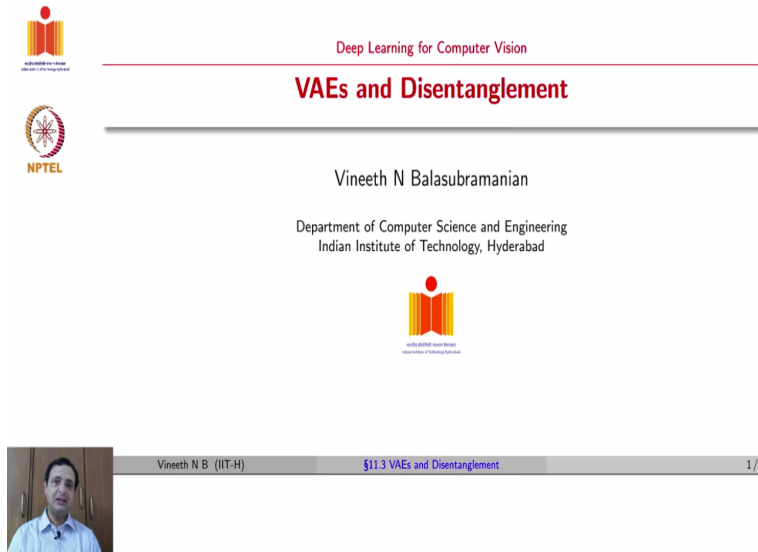


**Deep Learning for Computer Vision**  
**Professor Vineeth N Balasubramanian**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Hyderabad**  
**Lecture 70**  
**VAEs and Disentanglement**



(Refer Slide Time: 00:14)



The slide features a header with the text "Deep Learning for Computer Vision" and "VAEs and Disentanglement". It includes logos for the Department of Computer Science and Engineering at IIT Hyderabad and NPTEL. The presenter's name, "Vineeth N Balasubramanian", and his affiliation are listed. A small video inset shows the presenter, and a footer bar contains the text "Vineeth N B (IIT-H)", "§11.3 VAEs and Disentanglement", and "1 / 14".


We have seen different variants of GANs over the last couple of lectures. We will now move on to another important notion in Generative Models, which is called Disentanglement. This notion is more closely associated with Variational Auto Encoders VAEs, and we will also discuss why this is so as part of this lecture.

(Refer Slide Time: 00:45)



### What is Disentanglement?

- Isolating sources of variation in observational data
  - E.g. separating underlying concepts of "**Big Red Apple**": size (*big*), color (*red*) and shape (*apple*)
- Can we isolate these factors using some representation learning method?
- Why do we need this? Useful to generate new images that are not in observed dataset
  - E.g. Generate an image corresponding to "**Small Black Apple**" using a model that was trained on "*Small Black Grapes*" and "*Big Red Apples*"



Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 2 / 14


To start with, what is Disentanglement? Disentanglement is about isolating sources of variation in observational data. If you had an image of a Big Red Apple, can you separate the generative factors for such images as corresponding to size, big, color, red and shape or object apple? Can we enforce Deep Learning Models, Deep Generative Models, in particular, to isolate these factors while learning such a model? Why do we need such an approach?

If we could disentangle the generative factors, it allows us to generate new images that may not be in an observed dataset. Suppose your training dataset had images of small black grapes and big red apples. Can we generate an image corresponding to a small black apple? You may not find such an image in a real-world dataset.

But using a deep generative model can hypothesize how this would look by setting the color to a particular value, setting the size to a particular value and the object to a particular value. You would be able to do this reliably only if the latent variables in your generative model isolate these components of images.

(Refer Slide Time: 02:32)

**Disentanglement: Example**



Images generated when latents (dimensions encoding generative factors) corresponding to gender are changed; more control when latents are disentangled

Credit: Chen et al, *Isolating Sources of Disentanglement in Variational Autoencoders*, NeurIPS 2018

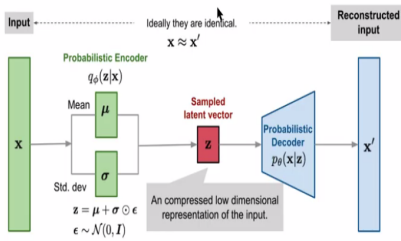
Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 3 / 14

Here is an example of face images. In this case, the latent variables could correspond to gender, age, hair, and perhaps race so on and so forth. So if we knew which latent variable corresponded to gender, one could manipulate that latent variable alone to generate different images of different variations going from, say, female to the male gender, as you can see in the example here.

(Refer Slide Time: 03:12)

**Disentanglement: Why VAEs?**

Recall VAEs:



Input  $x$  is processed by a Probabilistic Encoder  $q_{\phi}(z|x)$  to produce a latent vector  $z$ . The encoder outputs a Mean  $\mu$  and Std. dev  $\sigma$ . The latent vector  $z$  is sampled from a distribution  $z = \mu + \sigma \odot \epsilon$ , where  $\epsilon \sim \mathcal{N}(0, I)$ . The latent vector  $z$  is then processed by a Probabilistic Decoder  $p_{\theta}(x'|z)$  to produce a Reconstructed input  $x'$ . Ideally they are identical:  $x \approx x'$ .

An compressed low dimensional representation of the input.

VAEs learn latent variables which can be used to generate data; if these latent variables are disentangled, allows controlled generation of images

Credit: Lilian Weng


Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 4 / 14

Why Variational Auto Encoders? You perhaps know the reason already. We probably already used the word latent multiple times. In GANs, generative adversarial networks, the latents are not learned per se. The latent vector is a noise from a Gaussian. In a Variational Auto Encoder, the latent variables are learned. If one could now ensure that those latent variables are disentangled, you may have a lot of control over what kind of images you can generate out of the VAE. So recall the VAE overall architecture and formulation.

So you have your input data  $x$ , the encoder provides the mean and variance of an approximate posterior, which over learning tries to become close to a pre-assumed prior. Then a vector is sampled from the prior. The decoder reconstructs the data from that sample vector. These latent variables could be a vector of multiple dimensions. If they are disentangled, you can generate more control data.

VAE-GAN frameworks, such as Adversarial Auto Encoders, can benefit from disentangling this latent variable in a VAE.

(Refer Slide Time: 04:56)



### β-VAE<sup>1</sup>

- A variant of VAE which allows disentanglement
- Recall **VAE loss**:  $L_{VAE} = -\log p_{\theta}(x) + D_{KL}(q_{\phi}(z|x) || p_{\theta}(z))$
- Another way of writing the VAE objective:

$$\left[ \begin{array}{l} \max_{\phi, \theta} \mathbb{E}_{x \sim D} [\mathbb{E}_{z \sim q_{\phi}(z|x)} \log p_{\theta}(x|z)] \\ \text{subject to } D_{KL}(q_{\phi}(z|x) || p_{\theta}(z)) < \delta \end{array} \right.$$

Maximize probability of generating real data, while keeping distance between real and approximate posterior distributions small (under a small constant  $\delta$ )

$D_{KL}(q_{\phi}(z|x) || p_{\theta}(z))$

$|| p_{\theta}(z)$

<sup>1</sup>Higgins et al, beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, ICLR 2017

Vineeth N B (IIT-H)
§11.3 VAEs and Disentanglement
5 / 14

The first work that brought this notion to the community's attention and developed a method to allow disentanglement was  $\beta$ -VAE. Their work was published in ICLR 2017. It is primarily a variant of VAE itself. Let us see what kind of a variant. If you recall the variational autoencoder

loss, there are two terms in your evidence lower bound, one term which minimizes the negative log-likelihood.


In other words, it maximizes the log-likelihood of generating that kind of data that is in the training set. The second part minimizes the KL-divergence between the approximate posterior and the true posterior. It breaks down into two terms which we finally use while training the VAE.

We finally use only the KL divergence between  $q_{\phi}(z | x)$  and the prior  $p_{\theta}(z)$  after applying the evidence lower bound. This is the correct KL to start. But this gets simplified to the KL that is written on the right side.

Now, this entire objective can be written in a slightly different manner. We can say that we would like to maximize the log-likelihood of generating  $x$  from  $z$ . Subject to the constraint that the approximate posterior  $q_{\phi}$  and  $p_{\theta}(z)$ , the prior on  $z$ . The KL divergence between these two quantities is as small as possible. We say that the KL divergence should be less than some positive constant  $\delta$ .

This is another way of writing out the same objective. You can say now that we are maximizing the probability of generating the real data while keeping the distance between real and approximate posterior distributions small, which boils down using the evidence lower bound to keeping the distance between the approximate posterior and the prior small. How does this help?


(Refer Slide Time: 07:52)




### β-VAE

- VAE maximization objective can then be rewritten as a Lagrangian with a Lagrangian multiplier  $\beta$  under KKT conditions (similar to SVM):
 
$$\begin{aligned} & \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \beta (D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})) - \delta) \\ &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \beta D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})) + \beta \delta \\ &\geq \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \beta D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})) \end{aligned}$$

since  $\beta, \delta \geq 0$



Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 6 / 14




### β-VAE<sup>1</sup>

- A variant of VAE which allows disentanglement
- Recall VAE loss:  $L_{\text{VAE}} = -\log p_\theta(\mathbf{x}) + D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}))$
- Another way of writing the VAE objective:

$$\left[ \begin{array}{l} \max_{\phi, \theta} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} [\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z})] \\ \text{subject to } D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})) < \delta \end{array} \right. \rightarrow \beta(D_{\text{KL}} - \delta)$$

Maximize probability of generating real data, while keeping distance between real and approximate posterior distributions small (under a small constant  $\delta$ )



<sup>1</sup>Higgins et al, beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, ICLR 2017  
Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 5 / 14

Now, keeping this optimization problem in mind, we can write it as a Lagrangian. With the Lagrangian multiplier using the KKT conditions. This is very similar to how one would write out the support vector machine objective. So this would turn out to be maximizing the log-likelihood and minimizing the constraint term that we had. This constraint term, when we have a Lagrangian, would turn out to be  $D_{KL} - \delta$ .

And that would then go to the numerator, and you would have a Lagrangian multiplier  $\beta$ , using the standard Lagrangian approach to optimization. So here, we write the first term as it is the

objective function minus  $\beta$ , which is the Lagrangian multiplier into the constraint, which is KL-divergence between approximate posterior and prior minus  $\delta$ . If you expand this, the first term stays as it is, the second term becomes minus  $\beta$  into the KL. When we say KL, we mean KL-divergence plus  $\beta * \delta$ .

Since both  $\beta$  and  $\delta$  are quantities that are greater than or equal to 0, that is how we define them. So you are left with saying that this quantity will be greater than or equal to the log-likelihood minus the KL-divergence. We are writing this as a maximization problem. When we do minimization, the sign will change.

(Refer Slide Time: 09:37)

**$\beta$ -VAE**

- VAE maximization objective can then be rewritten as a Lagrangian with a Lagrangian multiplier  $\beta$  under KKT conditions (similar to SVM):
 
$$\begin{aligned} & \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \beta (D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})) - \delta) \\ &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \beta D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})) + \beta \delta \\ &\geq \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \beta D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})) \quad \text{since } \beta, \delta \geq 0 \end{aligned}$$
- $\beta$ -VAE loss hence given by:
 
$$L_{\text{BETA}}(\phi, \beta) = -\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + \beta D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}))$$
- When  $\beta = 1 \rightarrow$  standard VAE
- When  $\beta > 1 \rightarrow$  stronger constraint on latent bottleneck, follow generative process and thus encourage **disentanglement**
- Could limit representation capacity of  $\mathbf{z}$ , creating a trade-off between reconstruction quality and extent of disentanglement

Credit: Lilian Weng

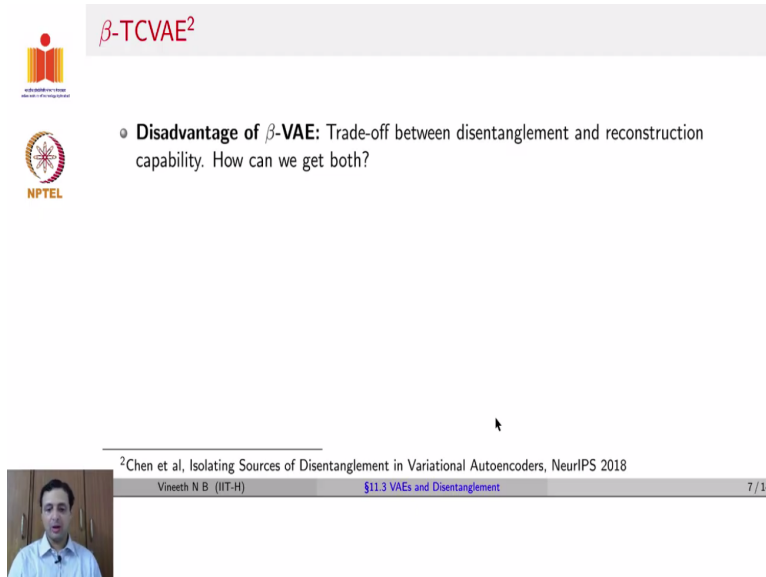
Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 6 / 14

One can now write the  $\beta$ -VAE loss to minimize, minus log-likelihood, or negative log-likelihood plus beta times the KL-divergence between approximate posterior and prior. It almost seems like nothing changed from a standard VAE which is partly true. In this case, when  $\beta$  is equal to 1, you would have the standard VAE. However, when  $\beta$  is made greater than 1, it introduces stronger disentanglement in the generative model. Why is the so?

Between these two terms used to train a VAE, the first term recall, the goal is to improve the reconstruction capability of the decoder. It is the second term that tries to learn the latents of the variational autoencoder. So by giving it a stronger weight, we are trying to make the latents be

learned better in a more disentangled way. The only problem now is this could limit the representation capacity of  $z$ , thus causing reconstruction problems in the entire VAE.

(Refer Slide Time: 11:05)



**$\beta$ -TCVAE<sup>2</sup>**

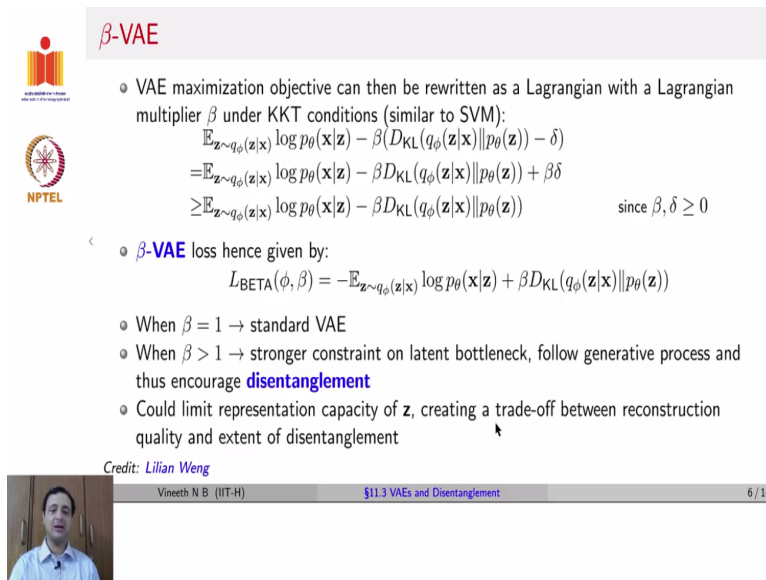
- Disadvantage of  $\beta$ -VAE: Trade-off between disentanglement and reconstruction capability. How can we get both?

<sup>2</sup>Chen et al, Isolating Sources of Disentanglement in Variational Autoencoders, NeurIPS 2018

Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 7 / 14

That brings us to another question which almost looks like a tradeoff between disentanglement and reconstruction capability. By increasing  $\beta$  in a  $\beta$ -VAE, we get better disentanglement, but the training procedure now thinks that the second term is more important.

(Refer Slide Time: 11:28)



**$\beta$ -VAE**

- VAE maximization objective can then be rewritten as a Lagrangian with a Lagrangian multiplier  $\beta$  under KKT conditions (similar to SVM):
 
$$\begin{aligned} & \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta (D_{\text{KL}}(q_\phi(z|x) \| p_\theta(z)) - \delta) \\ &= \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta D_{\text{KL}}(q_\phi(z|x) \| p_\theta(z)) + \beta \delta \\ &\geq \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta D_{\text{KL}}(q_\phi(z|x) \| p_\theta(z)) \quad \text{since } \beta, \delta \geq 0 \end{aligned}$$
- $\beta$ -VAE loss hence given by:
 
$$L_{\text{BETA}}(\phi, \beta) = -\mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) + \beta D_{\text{KL}}(q_\phi(z|x) \| p_\theta(z))$$
- When  $\beta = 1 \rightarrow$  standard VAE
- When  $\beta > 1 \rightarrow$  stronger constraint on latent bottleneck, follow generative process and thus encourage **disentanglement**
- Could limit representation capacity of  $z$ , creating a trade-off between reconstruction quality and extent of disentanglement

Credit: Lilian Weng

Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 6 / 14



In this case, the second term is more important and the first term is slightly less important. So, if the first term is slightly less important, this leads to lesser reconstruction performance.

(Refer Slide Time: 11:49)

**$\beta$ -TCVAE<sup>2</sup>**

$$q_{\phi}(z, x_n) \parallel p_{\theta}(z) = D_{KL}(q_{\phi}(z, x_n) \parallel q_{\phi}(z) p(x_n))$$

- Disadvantage of  $\beta$ -VAE: Trade-off between disentanglement and reconstruction capability. How can we get both?  $\beta$ -TCVAE the solution
- KL-divergence term can be decomposed as:

$$D_{KL}(q_{\phi}(z|x) \parallel p_{\theta}(z)) = \underbrace{I_q(z, \mathbf{n})}_{\text{index-code mutual information (MI)}} + \underbrace{D_{KL}(q_{\phi}(z) \parallel p_{\theta}(z))}_{\text{marginal KL to prior}}$$

Handwritten notes on the slide:

- $q_{\phi}(z|x) - q_{\phi}(z|x_n) = q_{\phi}(z, x_n)$
- $\frac{1}{N}$
- $x_i, i \in \{1, \dots, N\}$

<sup>2</sup>Chen et al, Isolating Sources of Disentanglement in Variational Autoencoders, NeurIPS 2018  
 Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 7 / 14

So to address this issue, to be able to get good reconstruction and good disentanglement, there was another method introduced in NeurIPS 2018 by Chen et al.,  $\beta$ -TCVAE. Let us try to understand this time. So the  $\beta$ -TCVAE looks at the KL-divergence term between the approximate posterior and prior and then decomposes it into two parts. How is this decomposition done? This decomposition is done by looking at the term, the approximate posterior  $q_{\phi}(z | x)$ , which could also be written as  $q_{\phi}(z | x_n)$ .


Now assume that you have a set of data points going from, say 1 to N, and each  $x_i$  is one data point where  $i$  comes from 1 to N. So that is the  $x_i$  that we are talking about is each of the data points. This is the same just expansion of writing the approximate posterior. So by standard probability, we can now write this as the joint probability,  $q_{\phi}(z, x_n)$  by the probability on  $x_n$ ,  $p(x_n)$ . Assuming all data points are equally likely, the denominator here would be  $1/N$ , which is a constant.

So, you could now say that we could replace the approximate posterior with the joint probability between the latent and each data point  $x_n$ . This means that the KL-divergence between the

approximate posterior and the prior can be broken down into two parts. It can also be written as KL between  $q_{\phi}(z, x_n)$ , the joint, with respect to the prior on  $z$ . The first term here,  $q_{\phi}$  can be broken down into two parts.

The first term would be the KL-divergence between the marginal on  $z$  with respect to the approximate prior, and the second term would be a KL-divergence between the joint distribution  $q_{\phi}(z, x_n)$  and the product of the marginals  $q(z) * p(x_n)$ . This is given by the mutual information between  $z$  and  $n$ ;  $n$  denotes the indices of the data points on  $x$ . Note that mutual information is defined as a constant factor of a KL-divergence between the joint distribution between two random variables and the product of its marginals. Now how does this decomposition help?


(Refer Slide Time: 15:13)



## β-TCVAE<sup>2</sup>

- **Disadvantage of β-VAE:** Trade-off between disentanglement and reconstruction capability. How can we get both? **β-TCVAE** the solution
- KL-divergence term can be decomposed as:
 

$$D_{\text{KL}}(q_{\phi}(z|x)||p_{\theta}(z)) = \underbrace{I_q(z, \mathbf{n})}_{\text{index-code mutual information (MI)}} + \underbrace{D_{\text{KL}}(q_{\phi}(z)||p_{\theta}(z))}_{\text{marginal KL to prior}}$$
- **Marginal KL to prior** more important to learn disentangled representations; reducing **MI** might be causing poor reconstruction. What to do?



<sup>2</sup>Chen et al, Isolating Sources of Disentanglement in Variational Autoencoders, NeurIPS 2018

Vineeth N B. (IIT-H) [§11.3 VAEs and Disentanglement](#) 7 / 14

Once we have this decomposition, one notice is that the second term is the marginal KL; we will call that marginal KL because the approximate posterior has now been marginalized. Earlier, we had  $q_{\phi}(z | x)$ , but that got marginalized. The other term now came into the mutual information.

This marginal KL is the component responsible for disentanglement.

Hence, trying to penalize the mutual information may lead to poorer reconstruction. Keeping this in mind, we now want to ensure that we focus on the marginal KL while learning a VAE; that is the term that we want to weigh with a beta.

(Refer Slide Time: 16:08)


$\beta$ -TCVAE<sup>3</sup>

Further decompose marginal KL:

$$D_{\text{KL}}(q_{\phi}(\mathbf{z}) \| p_{\theta}(\mathbf{z})) = \underbrace{D_{\text{KL}}(q_{\phi}(\mathbf{z}) \| \prod_j q_{\phi}(\mathbf{z}_j))}_{\text{Total Correlation}} + \sum_j \underbrace{D_{\text{KL}}(q_{\phi}(\mathbf{z}_j) \| p_{\theta}(\mathbf{z}_j))}_{\text{Dimension-wise KL}}$$

<sup>3</sup>Chen et al, Isolating Sources of Disentanglement in Variational Autoencoders, NeurIPS 2018

Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 8 / 14





Before we do that, we will do one more thing: decompose the marginal KL divergence even further. The marginal KL divergence can be decomposed into a term that looks at the KL divergence between  $\mathbf{z}$ , the random variable, and the product of the marginals of each dimension of  $\mathbf{z}$ . This term is known as Total Correlation. Although the name is a misnomer, Total Correlation is a concept from Information Theory which is a generalization of mutual information to multiple random variables.

If you add two random variables  $z$  and  $n$  that we saw on the previous slide, you look at the joint and the product of the marginals of the two random variables and take the KL divergence. In total correlation, we do this for all the random variables involved in  $\mathbf{z}$  in this particular context. Those random variables for us are the different dimensions of the  $\mathbf{z}$ . The second term here is the dimension-wise KL-divergence and the sum of all of them.

So we have broken the overall KL divergence of  $\mathbf{z}$  into dimension-wise quantities. Why is that important? Because in disentanglement, we would like each dimension to have a unique existence.

(Refer Slide Time: 17:45)

### β-TCVAE<sup>3</sup>

- Further decompose marginal KL:


$$D_{\text{KL}}(q_{\phi}(\mathbf{z}) \| p_{\theta}(\mathbf{z})) = \underbrace{D_{\text{KL}}(q_{\phi}(\mathbf{z}) \| \prod_j q_{\phi}(\mathbf{z}_j))}_{\text{Total Correlation}} + \underbrace{\sum_j D_{\text{KL}}(q_{\phi}(\mathbf{z}_j) \| p_{\theta}(\mathbf{z}_j))}_{\text{Dimension-wise KL}}$$

- Total Correlation** important for learning disentangled representation
- Hence, final **β-TCVAE** loss:

$$\underbrace{-\mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} \log p_{\theta}(\mathbf{x}|\mathbf{z}) + I_q(\mathbf{z}, \mathbf{n})}_{\text{Mutual Information}} + \beta \underbrace{D_{\text{KL}}(q_{\phi}(\mathbf{z}) \| \prod_j q_{\phi}(\mathbf{z}_j))}_{\text{Total Correlation}} + \underbrace{\sum_j D_{\text{KL}}(q_{\phi}(\mathbf{z}_j) \| p_{\theta}(\mathbf{z}_j))}_{\text{Dimension-wise KL}}$$

<sup>3</sup>Chen et al, Isolating Sources of Disentanglement in Variational Autoencoders, NeurIPS 2018


Vineeth N B (IIT-H)
§11.3 VAEs and Disentanglement
8 / 14



Now, suppose you look at this decomposition. In that case, one notices that total correlation is perhaps most important for disentangled representations. That term is responsible for looking at each dimension of  $\mathbf{z}$  to the overall  $\mathbf{z}$ . This leads us to the final loss for the  $\beta$ -TCVAE, which simply puts together all the components that we have seen so far, the negative log-likelihood, the mutual information, the total correlation, and the dimension-wise KL, which are the different components we have seen.

What is different? Notice,  $\beta$  is only on total correlation and not on any other terms in the overall objective.

(Refer Slide Time: 18:36)




### β-TCVAE<sup>3</sup>

- Further decompose marginal KL:
 
$$D_{\text{KL}}(q_{\phi}(\mathbf{z}) \| p_{\theta}(\mathbf{z})) = \underbrace{D_{\text{KL}}(q_{\phi}(\mathbf{z}) \| \prod_j q_{\phi}(\mathbf{z}_j))}_{\text{Total Correlation}} + \underbrace{\sum_j D_{\text{KL}}(q_{\phi}(\mathbf{z}_j) \| p_{\theta}(\mathbf{z}_j))}_{\text{Dimension-wise KL}}$$
- Total Correlation** important for learning disentangled representation
- Hence, final **β-TCVAE** loss:
 
$$-\mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} \log p_{\theta}(\mathbf{x}|\mathbf{z}) + I_q(\mathbf{z}, \mathbf{n}) + \beta D_{\text{KL}}(q_{\phi}(\mathbf{z}) \| \prod_j q_{\phi}(\mathbf{z}_j) + \sum_j D_{\text{KL}}(q_{\phi}(\mathbf{z}_j) \| p_{\theta}(\mathbf{z}_j))$$
- Weight  $\beta > 1$  to disentangle without affecting reconstruction


<sup>3</sup>Chen et al, Isolating Sources of Disentanglement in Variational Autoencoders, NeurIPS 2018

Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 8 / 14



This allows us to focus  $\beta$  on disentanglement only on that term and not affect the reconstruction capabilities of the VAE.

(Refer Slide Time: 18:47)




### Disentangled Representation Learning: How to evaluate?

#### Mutual Information Gap (MIG)

- Use mutual information between generative factors ( $\mathbf{g}$ ) and latent dimensions ( $\mathbf{z}$ ) in some way; how?
- Compute mutual information between each generative factors ( $\mathbf{g}_i$ ) and each latent dimension ( $\mathbf{z}_i$ )
- For each  $\mathbf{g}_i$ , take  $\mathbf{z}_j, \mathbf{z}_l$  that have highest and second highest mutual information with  $\mathbf{g}_i$
- $\text{MIG} = \frac{1}{K} \sum_{i=1}^K \frac{1}{H(\mathbf{g}_i)} (I(\mathbf{g}_i, \mathbf{z}_j) - I(\mathbf{g}_i, \mathbf{z}_l))$  where  $H(\mathbf{g}_i)$  is entropy of  $\mathbf{g}_i$  and  $0 \leq I(\mathbf{g}_i, \mathbf{z}_j) \leq H(\mathbf{g}_i)$
- Averaging by  $K$  and normalizing by  $H(\mathbf{g}_i)$  provides values between 0 and 1
- $\text{MIG} \rightarrow 0$  : bad disentanglement,  $\text{MIG} \rightarrow 1$  : good disentanglement
- Why not simply use MI? Why MI gap? **Homework!** (Hint: Read metric section in Chen et al, Isolating Sources of Disentanglement in Variational Autoencoders, NeurIPS 2018)

Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 9 / 14



Having seen  $\beta$ -VAE and  $\beta$ -TCVAE, one question that arises now is how do you evaluate whether your generative model has learned to disentangle effectively? While one way is to generate different images and check qualitatively whether those images represent different generative factors, that can become a tedious exercise for many generative factors. One such metric that has

been proposed in recent years is known as the Mutual information Gap (MIG). The idea is to use the mutual information between generative factors,  $g$  and latent dimensions,  $z$  in some way.

What do we mean by generative factors? These are factors that we know exist in the dataset. If we say, big red apple, size, color, and object represent the generative factors. The idea is to see if the latent dimensions  $z$  that are learned capture these generative factors somehow. We would like to use the mutual information between the random variables  $g$  and the latent dimensions  $z$  to capture this.

How do we do this? We compute the mutual information between each generator factor,  $g_i$  and each latent dimension,  $z_i$ . You would then have an entire matrix of mutual information between every pair,  $g_1$  and  $z_1$ ,  $g_1$  and  $z_2$ ,  $g_2$  and  $z_1$ , so on and so forth. What do we do with all of these mutual information values? For each generator factor  $g_i$  consider the latent factors that have the top two mutual information values. Let us call them  $z_j$  and  $z_l$ .

Once we have this, we define the mutual information gap as the difference in the mutual information values between these top two latent factors. So, the mutual information of  $g_i$  with  $z_j$  and  $g_i$  with  $z_l$ , will be the mutual information gap with some normalization factor on the outside.



What is the normalization factor?  $1/H(g_i)$ , the entropy of  $g_i$  intrinsically, i.e., entropy is  $-\sum p \log p$  of that generative factor.

And this normalization takes care of averaging this across all of the generative factors. So averaging by  $K$  and normalizing by  $H$ , entropy, provides us values between 0 and 1. If the mutual information gap is zero, both these latent factors have high mutual information with the same generative factor. It would be considered a bad disentanglement because both those latents are learning the same thing. They are not disentangled.

On the other hand, when MIG is 1, it is good disentanglement. One question here is why do we use the mutual information gap and not just mutual information itself? Think about it. It is

homework for you. If you need to understand this better, read the paper “*Isolating Sources of Disentanglement*”, NeurIPS 2018 paper, which defined this metric.

(Refer Slide Time: 23:06)




### Disentangled Representation Learning: How to evaluate?

#### DCI Metric<sup>a</sup>

<sup>a</sup>Eastwood and Williams, A Framework for the Quantitative Evaluation of Disentangled Representations, ICLR 2018

- Considers three properties of representations: D - Disentanglement, C - Completeness, I - Informativeness
- Train a model (e.g.  $\beta$ -VAE) to get latent representations
- Get latent representation of each image in a dataset
- Train  $k$  linear regressors (one for each  $\mathbf{g}_i$ ),  $f_1 \dots f_k$ , to predict  $\mathbf{g}_i$  given  $\mathbf{z}$
- From the regressors, we get  $W_{ij}$  (how much  $\mathbf{z}_i$  is important to predict  $\mathbf{g}_j$ )
- Create a **relative importance matrix**  $R$  such that  $R_{ij} = |W_{ij}|$



Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 10 / 14



Another metric for checking disentanglement that has been recently proposed in ICLR 2018 is known as the DCI Metric. DCI stands for Disentanglement, Completeness and Informativeness. There is a quantity defined for each of these. To compute them, let us first train any model, say Beta-VAE, to learn latent representations. Get the latent representation of each image in a training dataset or the test dataset, for that matter, if that is where you would like to study for disentanglement.

Then we train a linear regressor. So you learn  $k$  different linear regressors,  $f_1, \dots, f_k$ , that predicts each generative factor  $g_i$  given the entire latent vector  $z$ . So you have  $k$  different generative factors, and hence  $k$  different linear regressors. How do you learn them? For each input image, you would get a latent factor  $z$ , and you want to use that now to predict the gender here? Or what was the color of this Apple?

That would be the value of each generative factor. Once we train these linear regressors, it will give you a weighted combination of each latent factor. That is what linear regression does. So you would now have an entire matrix,  $W_{ij}$ , which tells us how much a latent factor  $z_i$  is important to predict a generative factor  $g_j$ . We will call this the relative importance matrix, which is the absolute value of  $W_{ij}$ 's, obtained through regressors.



(Refer Slide Time: 25:10)




### Disentangled Representation Learning: How to evaluate?

**DCI Metric: Disentanglement**

- Degree to which a representation disentangles underlying factors of variation
- Disentanglement score of  $i^{\text{th}}$  latent:  $D_i = (1 - H(P_i))$  where  $H$  is entropy and  $P_{ij} = \frac{R_{ij}}{\sum_k R_{ik}}$ , importance of  $z_i$  to predict  $g_j$
- Total disentanglement score:  $D = \sum_i \rho_i D_i$  where  $\rho_i = \frac{\sum_j R_{ij}}{\sum_{ij} R_{ij}}$ , relative latent importance used to normalize the score

**DCI Metric: Completeness**

- Degree to which each underlying generative factor is captured by a single latent variable
- For each generative factor  $g_j$ ,  $C_j = (1 - H(P_j))$  where the distribution  $P_j$  is as above
- If a single latent variable contributes to  $g_j$ 's prediction, score is 1 (**complete**); if all latent variables equally contribute to  $g_j$ 's prediction, score is 0 (**maximally overcomplete**)



Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 11 / 14

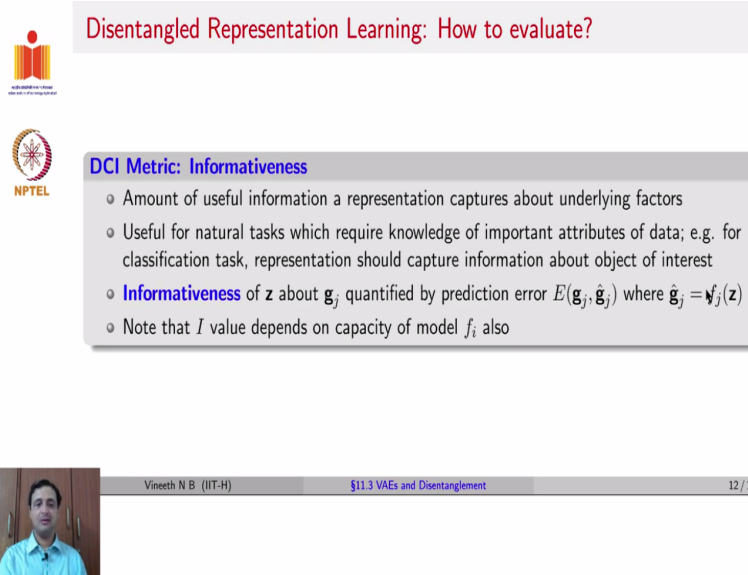
Once we get the relative importance matrix, there are quantities defined for disentanglement: completeness and informativeness. Let us look at disentanglement first. This metric tries to capture the degree to which representation disentangles underlying factors of variation. This is obtained by defining  $D_i$  as  $1 - H(P_i)$ , where  $H$  is the entropy of  $P_i$ . What is the probability distribution  $P_i$ ?  $P_i$  is defined as a vector of  $P_{ij}$ 's for each generator factor  $g_j$ , where each  $P_{ij}$  is given by  $R_{ij}$  in that matrix divided by all the entries in that particular row  $i$  corresponding to that latent factor.

So, that is the disentanglement score of the  $i^{\text{th}}$  latent. So, we ideally want the latent to predict only one generative factor and not predict all. So, the Total Disentanglement score is then given by summation  $\rho_i D_i$ , where  $D_i$  was the disentanglement score of only the  $i^{\text{th}}$  latent. So, the overall disentanglement score is given by summation  $\rho_i D_i$ , where the coefficient  $\rho_i$  is given by summation over  $j$ ,  $R_{ij}$  divided by summation over  $ij$ ,  $R_{ij}$ , which is the normalization over the column of that importance matrix.

The second metric is completeness, which is the degree to which a single latent variable captures each underlying generative factor. For each generative factor  $g_j$ , the completeness is defined as

$1 - H(P_j)$ , where  $P_j$  is defined as above. If a single latent variable contributes to  $g_j$ 's prediction, the score would be 1. We only want one latent variable to correspond to a generative factor. If all latent variables contribute equally to  $g_j$ 's prediction, the score is 0 because that represents the opposite of disentanglement. We call that situation maximally over complete, where all latent factors correspond to just one generative factor in the data.

(Refer Slide Time: 28:00)



**Disentangled Representation Learning: How to evaluate?**

**DCI Metric: Informativeness**


- Amount of useful information a representation captures about underlying factors
- Useful for natural tasks which require knowledge of important attributes of data; e.g. for classification task, representation should capture information about object of interest
- Informativeness** of  $z$  about  $g_j$  quantified by prediction error  $E(g_j, \hat{g}_j)$  where  $\hat{g}_j = f_j(z)$
- Note that  $I$  value depends on capacity of model  $f_j$ ; also

Vineeth N B (IIT-H) §11.3 VAEs and Disentanglement 12 / 14


The third metric is Informativeness. How informative are the disentangled latent representations? This measures how useful, a representation is in capturing the underlying factors. This is considered with respect to a specific task. For example, a classification task in which you would like the latent representation to capture information about the object of interest. How do we measure this?

The prediction error gives the Informativeness of  $z$  about a particular generator factor  $g_j$  between the original generator factor and the predicted generator factor. It is obtained using one of those regressors that we had defined earlier. These metrics, including informativeness, depend on the goodness of those regressors that we use in the first step.

(Refer Slide Time: 29:11)



### Homework




#### Readings

- [Lilian Weng, From Autoencoders to Beta-VAE](#)
- [Prashna Gyawali, Disentanglement with VAEs: A Review](#)
- (Optional) Papers on respective slides

#### Questions

- Why is MI Gap and not  $M_{\text{L}}$  used as a metric for disentanglement?



Vineeth N B. (IIT-H) §11.3 VAEs and Disentanglement 13 / 14

That completes our discussion of disentanglement. I hope it provided you with an introduction to the topic, a couple of methods that enforce disentanglement, as well as how to measure whether a generative model is disentangled. As homework, please read this excellent blog, “*From Autoencoders to Beta-VAE*”. Another blog on “*Review of Disentanglement with VAEs*” and optionally, the papers that we referred to in the slides.

We left behind the question in the first mutual information gap metric: *Why the gap and not just mutual information itself?* Think about it and we will discuss it next time.

(Refer Slide Time: 30:00)



## References

- 1. Ricky TQ Chen et al. "Isolating sources of disentanglement in variational autoencoders". In: *Advances in Neural Information Processing Systems*. 2018, pp. 2610–2620.
- 2. Cian Eastwood and Christopher KI Williams. "A framework for the quantitative evaluation of disentangled representations". In: *International Conference on Learning Representations*. 2018.
- 3. Irina Higgins et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *ICLR*. 2017.
- 4. Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. arXiv: [1312.6114](https://arxiv.org/abs/1312.6114) [stat.ML].



Vineth N B (IIT-H)

§11.3 VAEs and Disentanglement

14 / 14

References.