TÜBİTAK

ODTÜ
METU

INFORMATICS INSTITUTE

EURO[2]

Fundamental Concepts of Generative Machine Learning

Erdem Akagündüz, PhD
Graduate School of Informatics, METU, Türkiye

ncc@ulakbim.gov.tr

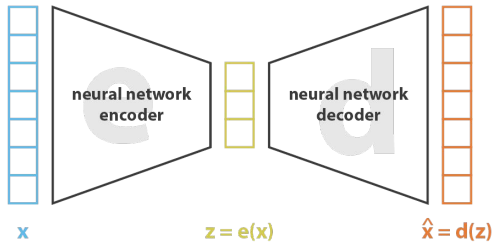
Lesson 3: Auto-Encoding

Welcome to **Part III: “Auto-Encoding”**

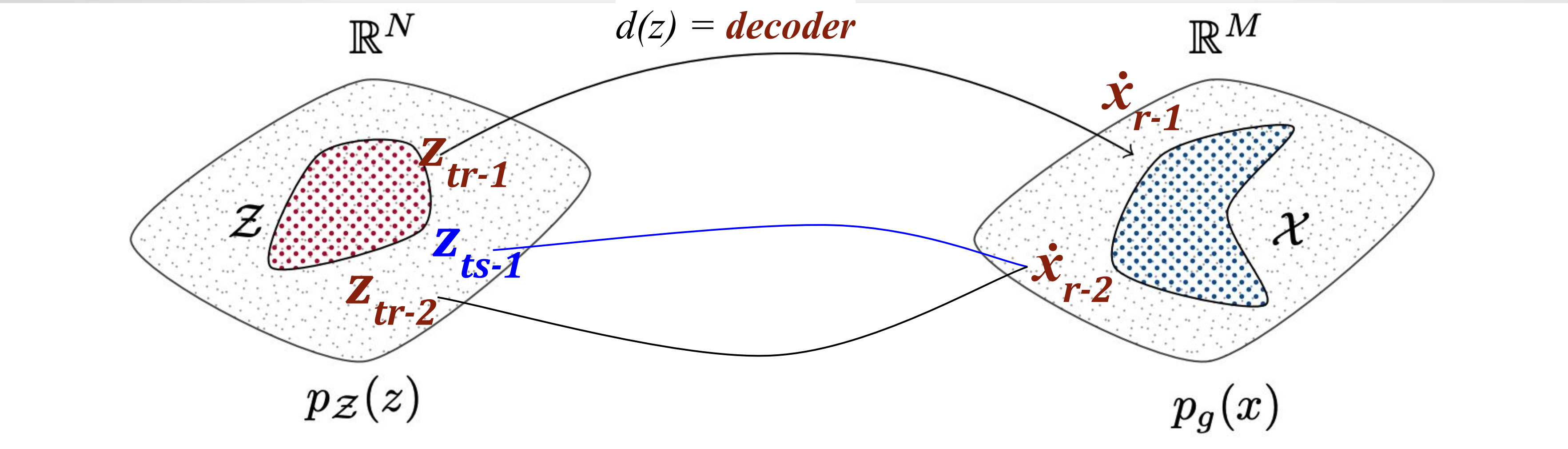
This part includes three subsections:

- Autoencoders and Dimensionality Reduction
- **Variational Inference and VAEs**
- Conclusions

AE problems

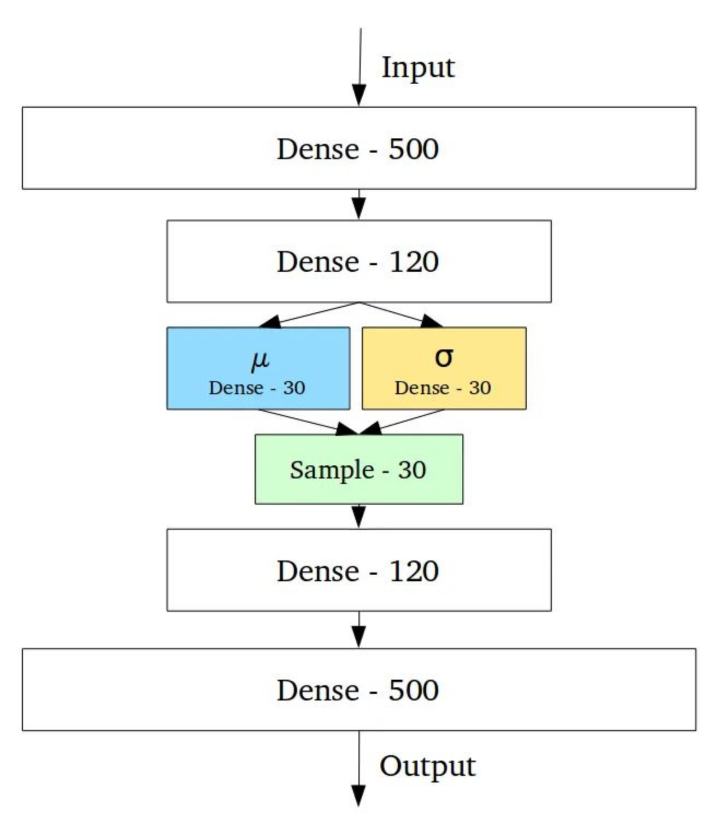


- But when you’re building a generative model, you don’t want to replicate the same input.
- You want to randomly sample from the latent space, or generate variations on an input image, from a continuous latent space.



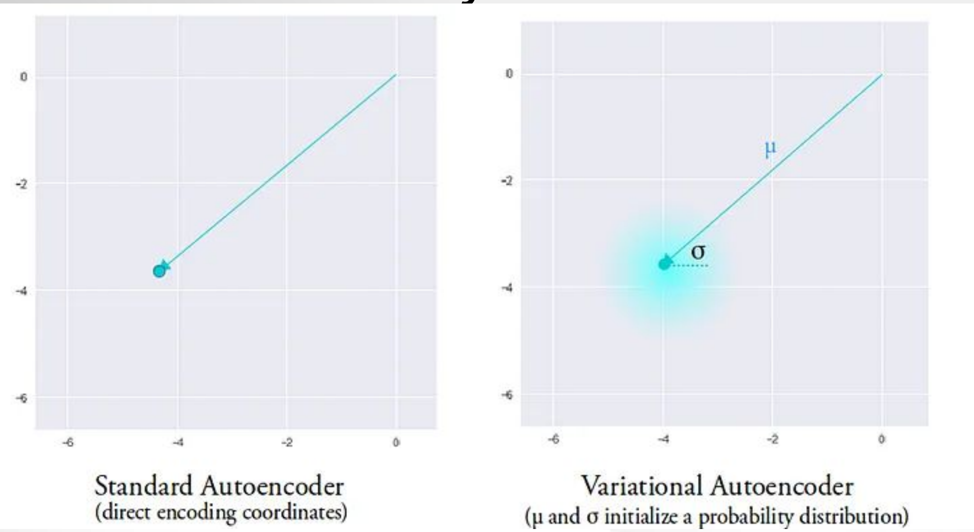
Variational AutoEncoders (VAEs)

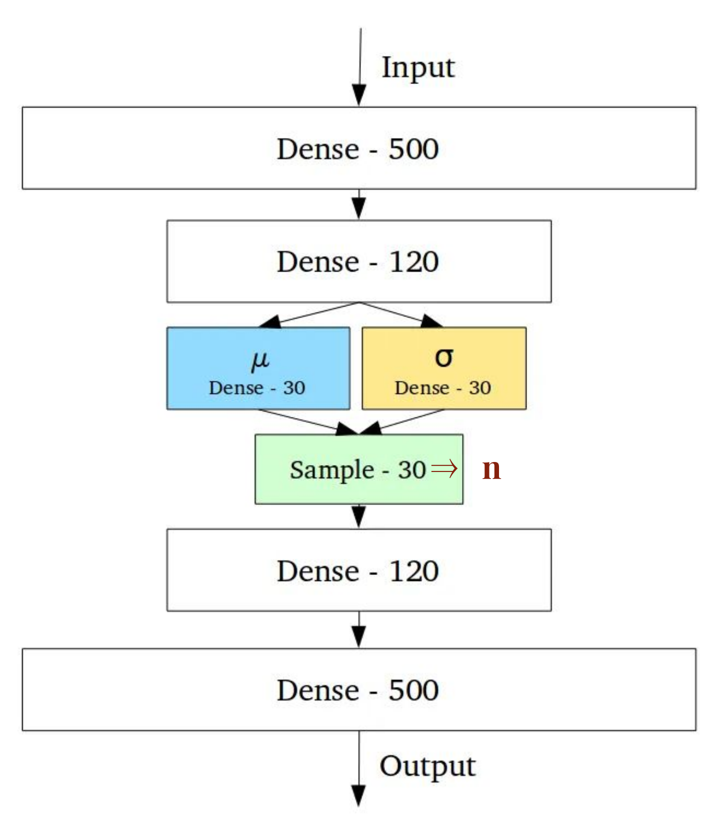
- Variational Autoencoders (VAEs) have one fundamentally unique property that separates them from vanilla autoencoders, and it is this property that makes them so useful for generative modeling:
- their latent spaces are, **by design**, continuous, allowing easy random sampling and interpolation.



Variational AutoEncoders (VAEs)

- Instead of mapping the input to a fixes latent vector, they map it to a distribution!
- What is more, VAEs force the latent variables to be Normal distributed.
- It achieves this by making its encoder :
  *- not output an encoding vector of size **n** (like AEs),*
  *- rather, outputting two vectors of size **n**:*
  - *a vector of means, **μ**,*
  - *and another vector of standard deviations, **σ**.*





Variational AutoEncoders (VAEs)

- Instead of mapping the input to a fixes latent vector, they map it to a distribution!
- What is more, VAEs force the latent variables to be Normal distributed.
- It achieves this by making its encoder :
  *- not output an encoding vector of size **n** (like AEs),*
  *- rather, outputting two vectors of size **n**:*
  - *a vector of means, **μ**,*
  - *and another vector of standard deviations, **σ**.*

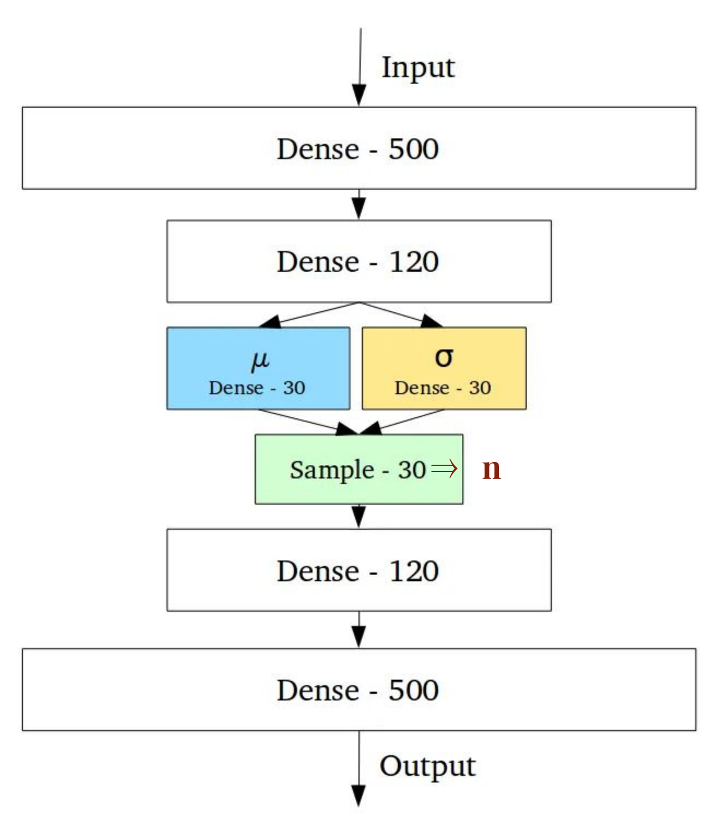
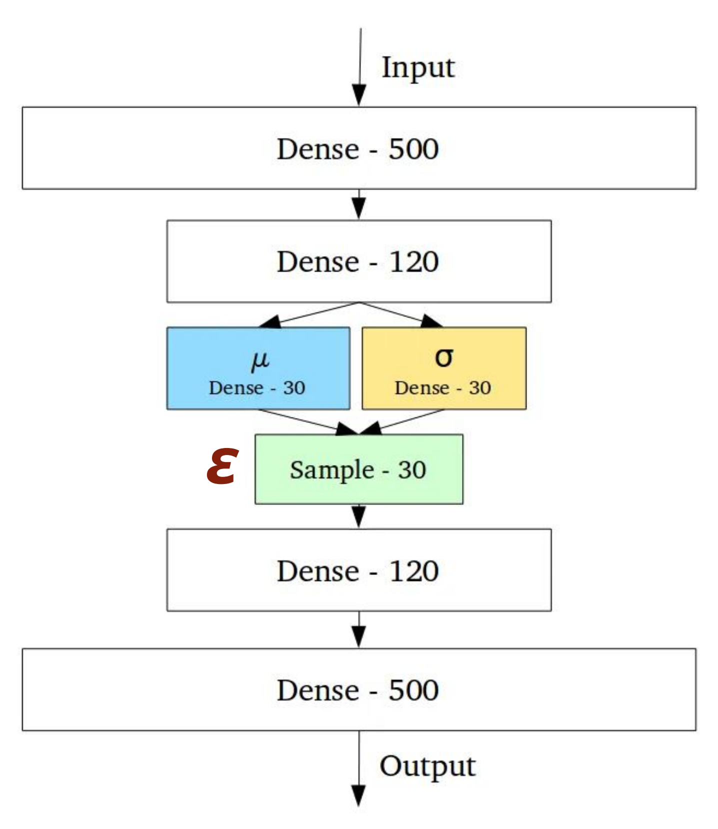


figure from https://towardsdatascien...

Variational AutoEncoders (VAEs)

- To achieve this, VAEs introduce two additional layers in the bottleneck: the mean layer and the standard deviation layer.
- These layers take the output of the previous bottleneck layer and output the mean vector ($\boldsymbol{\mu}$) and the standard deviation vector ($\boldsymbol{\sigma}$) respectively.
- Specifically, we sample a vector $\boldsymbol{\varepsilon}$ from a standard normal distribution (zero-mean, uniti var), and sample **z,** accordingly.

$$\mathbf{z = \mu + \sigma * \varepsilon}$$



Variational AutoEncoders (VAEs)

- Here, $\boldsymbol{\varepsilon}$ serves as a source of randomness and allows us to sample different points from the latent space during training or generation.
- Ok. Perfect. Forward run is stochastic! Great! But:

How is this random operation handled during training? Can you backpropagate a layer that creates a random variable?

$$z = \mu + \sigma * \varepsilon$$

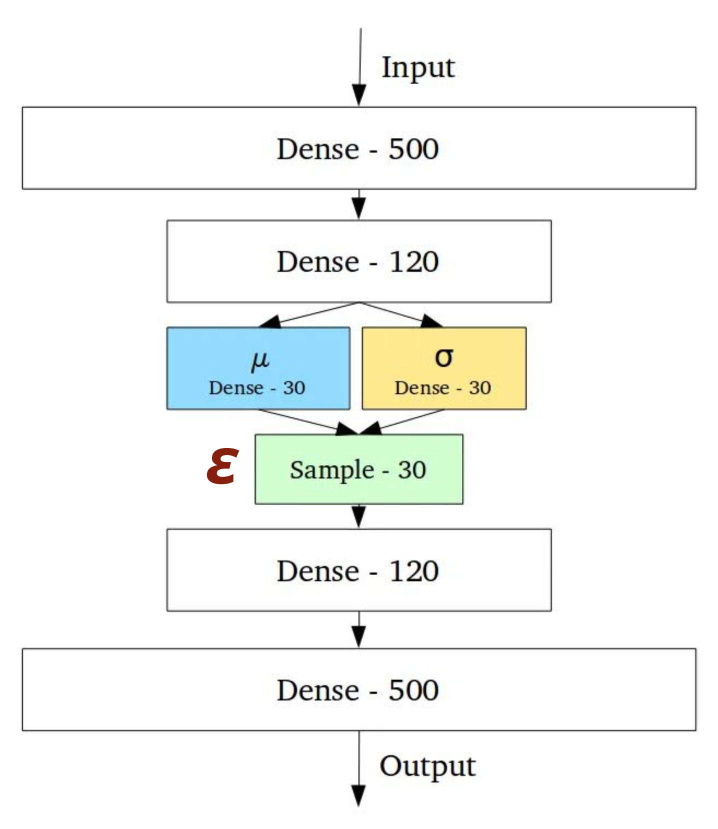
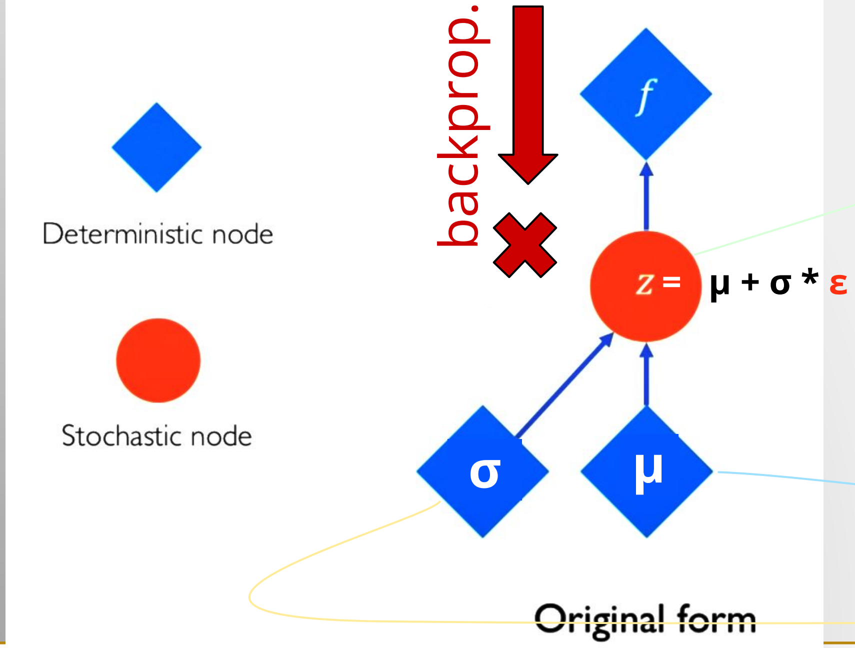


figure from https://towardsdatascien...

Reparameterization Trick

- The random sampling operation in VAEs, where **ε** is drawn from a standard normal distribution, introduces a challenge for backpropagation since it involves a non-differentiable operation.



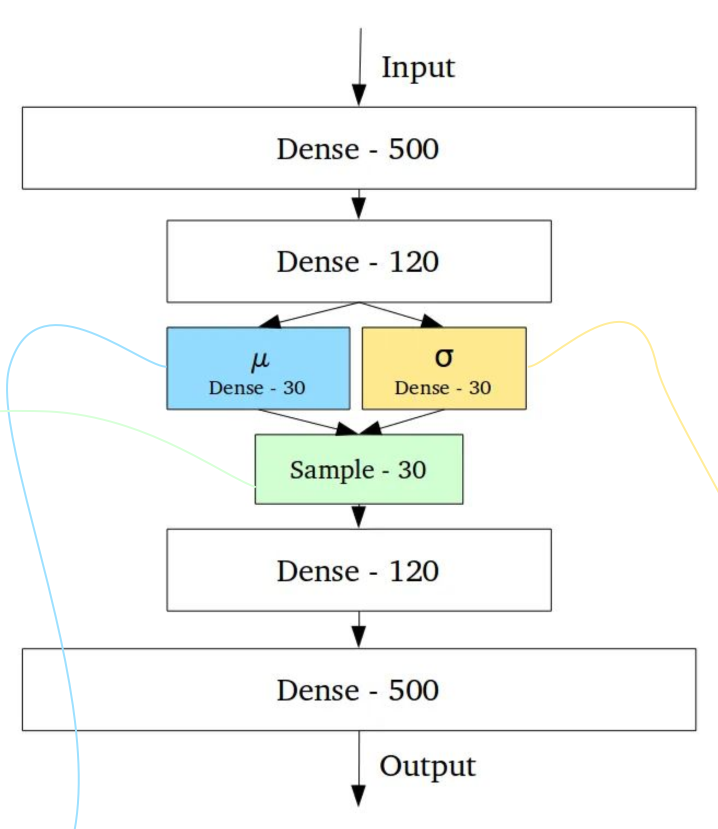
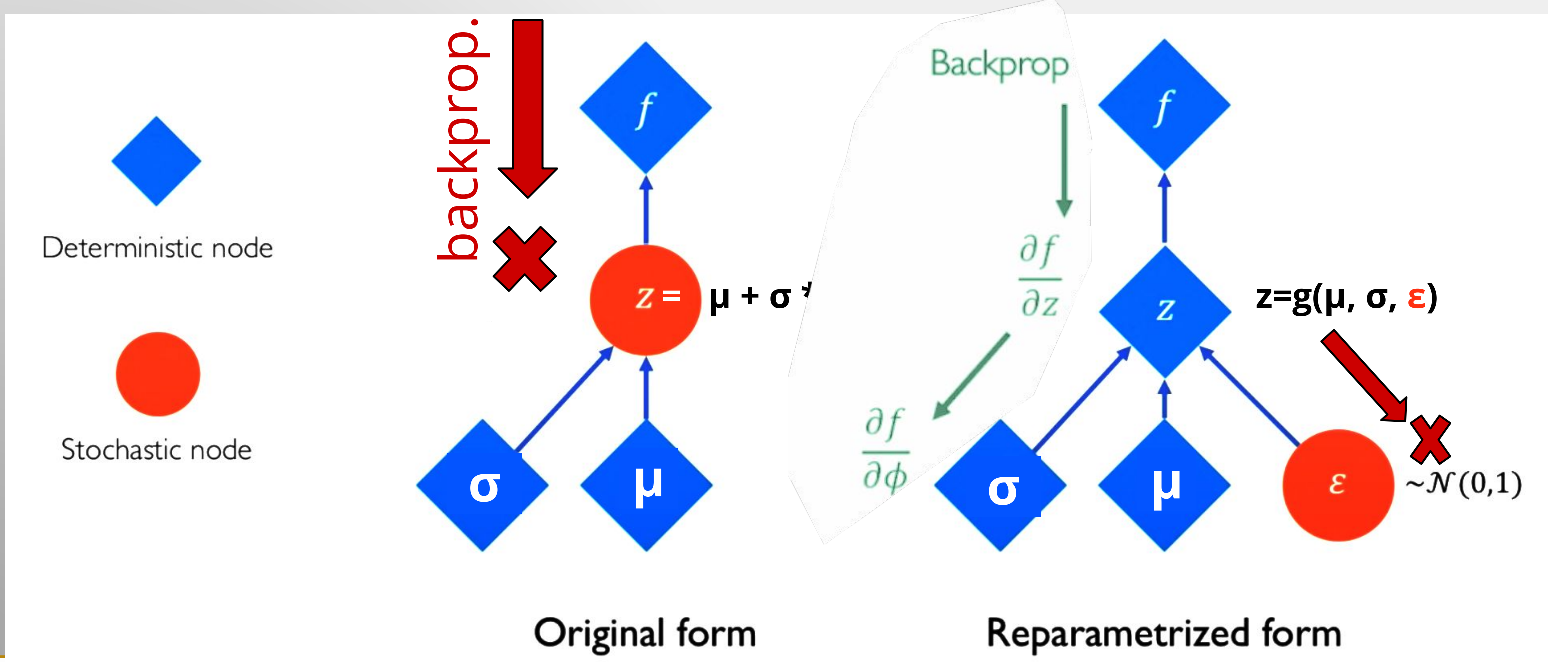


figure from http://introtodeeplearning... . . .

Reparameterization Trick

- The random sampling operation in VAEs, where **ε** is drawn from a standard normal distribution, introduces a challenge for backpropagation since it involves a non-differentiable operation.



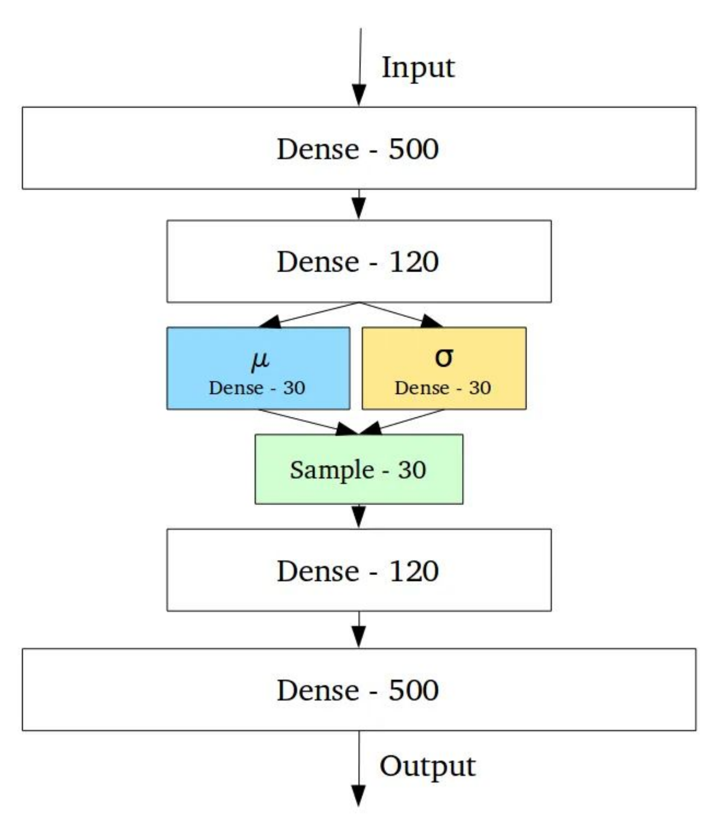


figure from http://introtodeeplearning...

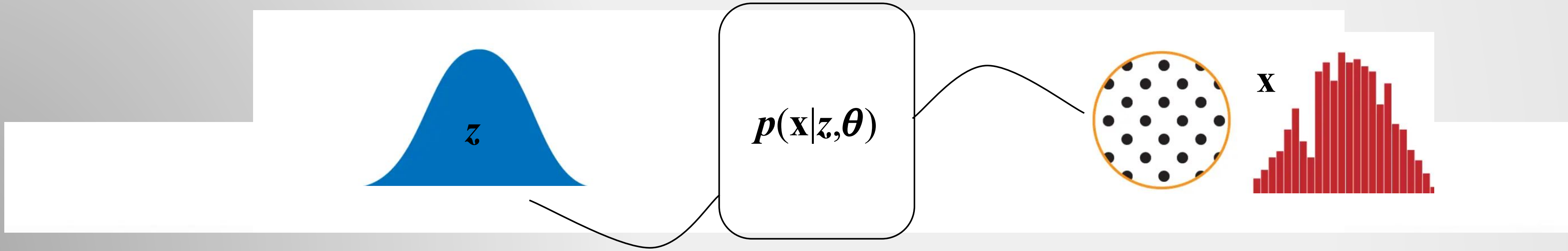
VAE Loss as ELBO

- To train a VAE, we need to maximize the likelihood of the observed data 'x' given the model parameters 'θ'. However, directly computing p(x | z, θ) is not feasible because we do not have access to the “true” latent variables 'z', and the encoding process is not perfectly reversible.
- Instead, VAEs use the Evidence Lower Bound (ELBO) as an approximation to the log-likelihood. The ELBO involves two terms: the expected log-likelihood of the data given the latent space (decoder part) and the negative KL divergence between the approximate posterior (encoded distribution) and the prior distribution over the latent space.

$$\mathcal{L}(\phi, \theta, x) = \boxed{\text{(reconstruction loss)}} + \text{(regularization term)}$$

VAE Loss as ELBO

- By maximizing the ELBO, the VAE effectively encourages the latent space to be structured and follow the prior distribution (usually a standard normal distribution), and it encourages the decoder to generate data that is similar to the observed data.
- The inability to directly calculate p(x|z, θ) due to the non-invertibility of the encoder is the reason why VAEs use ELBO for training instead of maximum likelihood.



VAE Loss

- So we know how to backprop a VAE. Ready for training then!
- During training, VAEs has two optimization goals:
  - minimize the reconstruction error (similar to traditional autoencoders)
  - and to ensure that the generated latent vectors follow the desired probability distribution.
- The first component is the reconstruction loss and it is trivial.

$$\mathcal{L}(\phi, \theta, x) = \boxed{\text{(reconstruction loss)}} + \text{(regularization term)}$$

VAE Reconstruction Loss

- Reconstruction loss is straightforward.

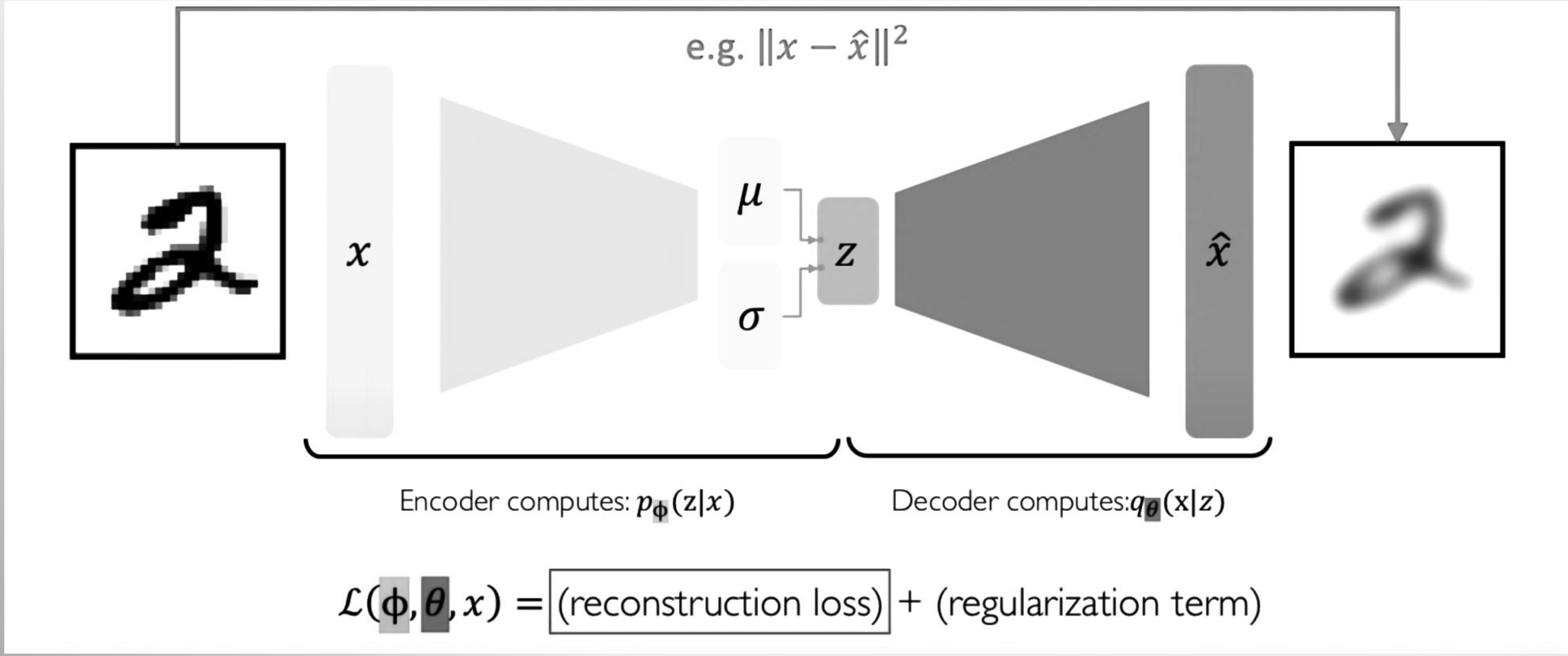


figure from http://introtodeeplearning... . . .

VAE Regularization Loss

- So we know how to backprop a VAE. Ready for training then!
- During training, VAEs has two optimization goals:
  - minimize the reconstruction error (similar to traditional autoencoders)
  - and to ensure that the generated latent vectors follow the desired probability distribution.
- The second is done by including a regularization term in the loss function called the Kullback-Leibler (KL) divergence.
  - The KL divergence measures how different the distribution of the generated latent vectors is from the desired distribution (in this case, a standard normal distribution).

$$\mathcal{L}(\phi, \theta, x) = (\text{reconstruction loss}) + \boxed{(\text{regularization term})}$$

Kullback-Leibler (KL) Divergence

- The KL divergence is a measure of the difference between two probability distributions, P and Q.
- It is defined as the expected value of the logarithmic difference between P and Q, where <u>the expectation is taken with respect to P</u>. The KL divergence is denoted as D(P||Q).

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

VAE Regularization Loss

- Regularization loss shapes the distributions.

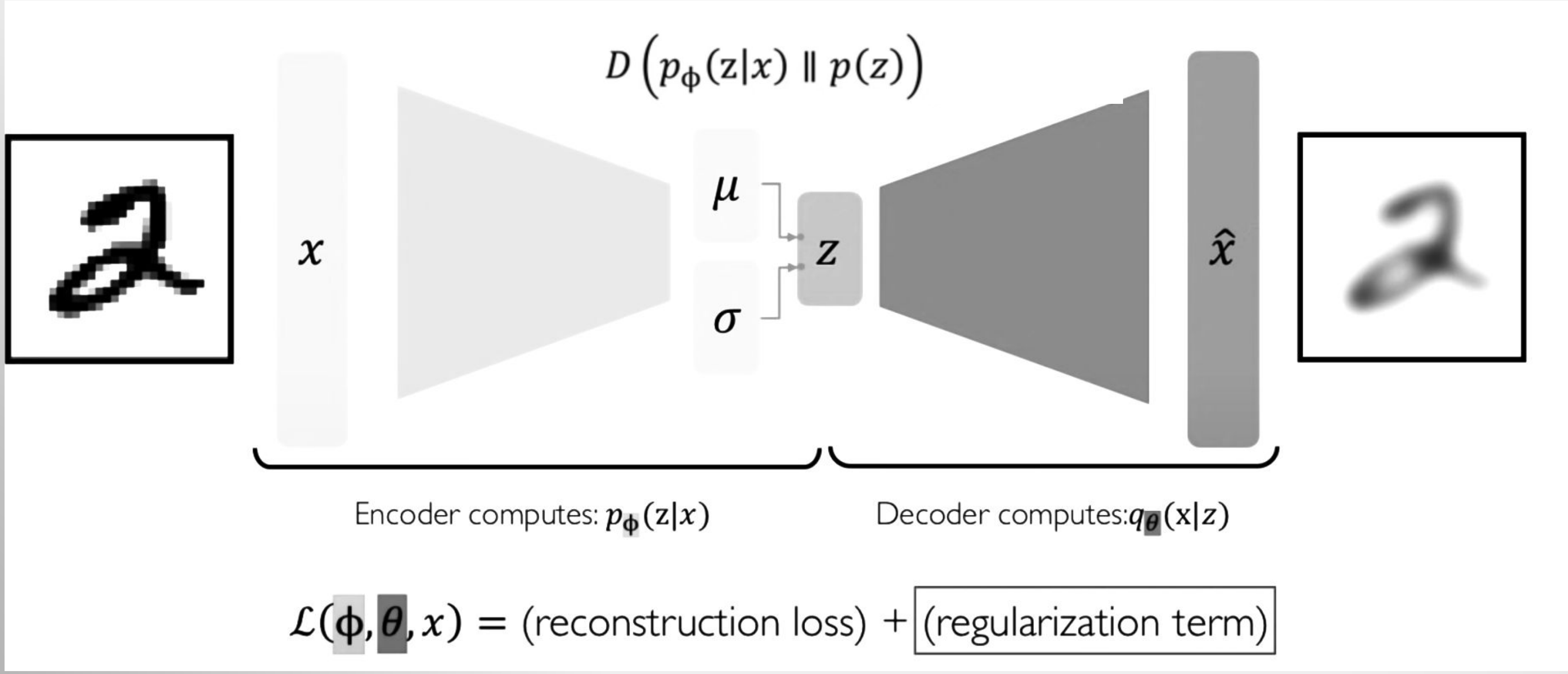
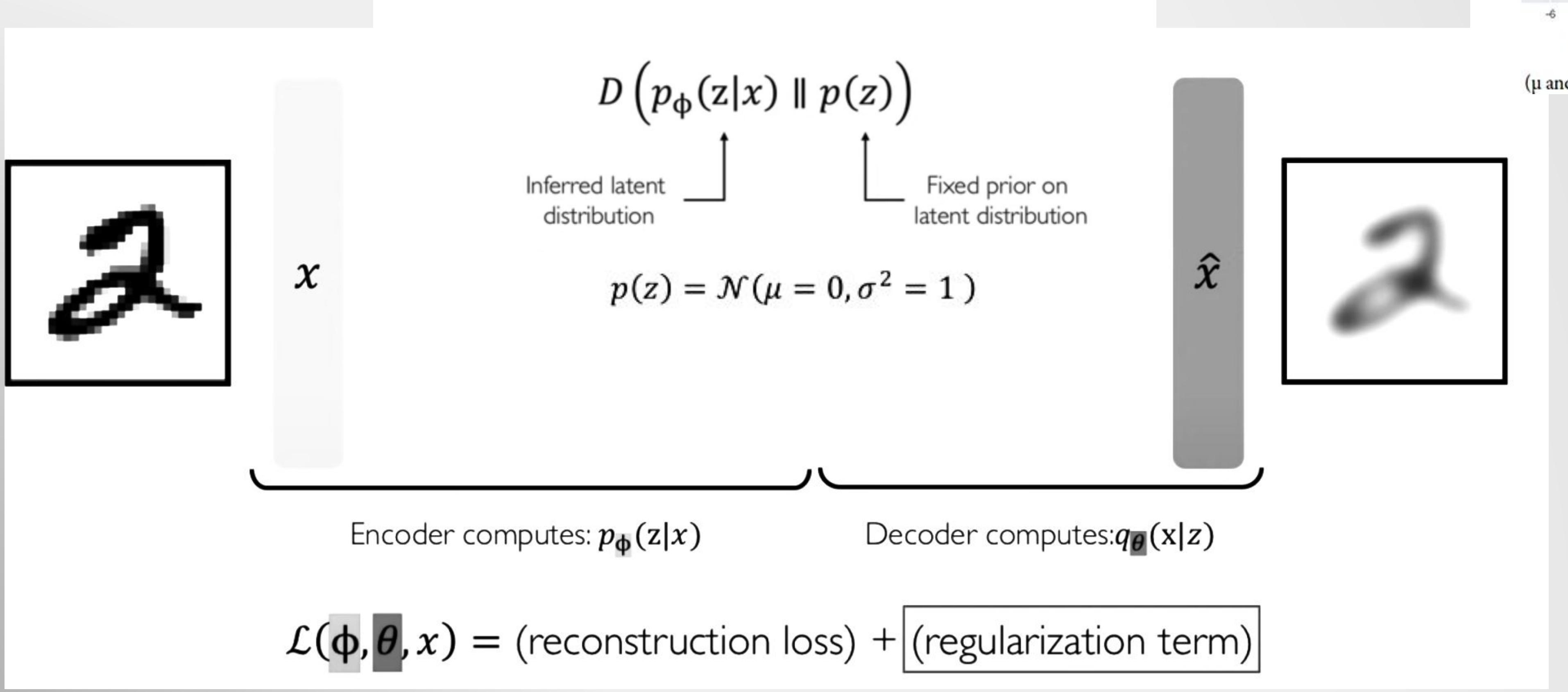
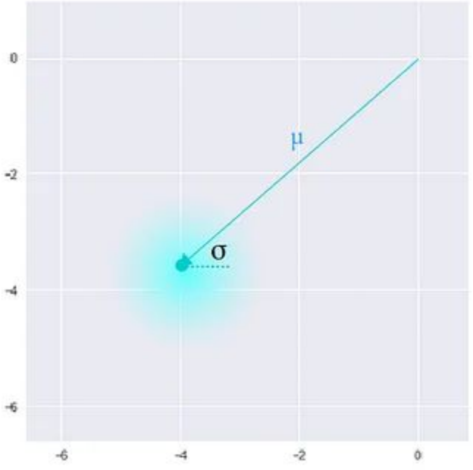


figure from http://introtodeeplearning... . . .

VAE Regularization Loss

- Regularization loss shapes the distributions.





Variational Autoencoder
(μ and σ initialize a probability distribution)

figure from http://introtodeeplearning...

VAE Regularization Loss

- Regularization loss shapes the distributions.

$$D\left(p_{\phi}(z|x) \parallel p(z)\right)$$

Inferred latent distribution

Fixed prior on latent distribution

$$p(z) = \mathcal{N}(\mu = 0, \sigma^2 = 1)$$

- The KL divergence mea ent the distribution of the generated latent vectors is from the desired distribution (in this case, a standard normal distribution).

Next lecture:

- PART III: Auto-Encoding

- Autoencoders and Dimensionality Reduction
- Variational Inference and VAEs
- **Conclusions**

Thanks

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia