Lecture 9

Generative Models

Haiping Lu

YouTube Playlist:
https://www.youtube.com/c/HaipingLu/

COM4059/6059: MLAI20

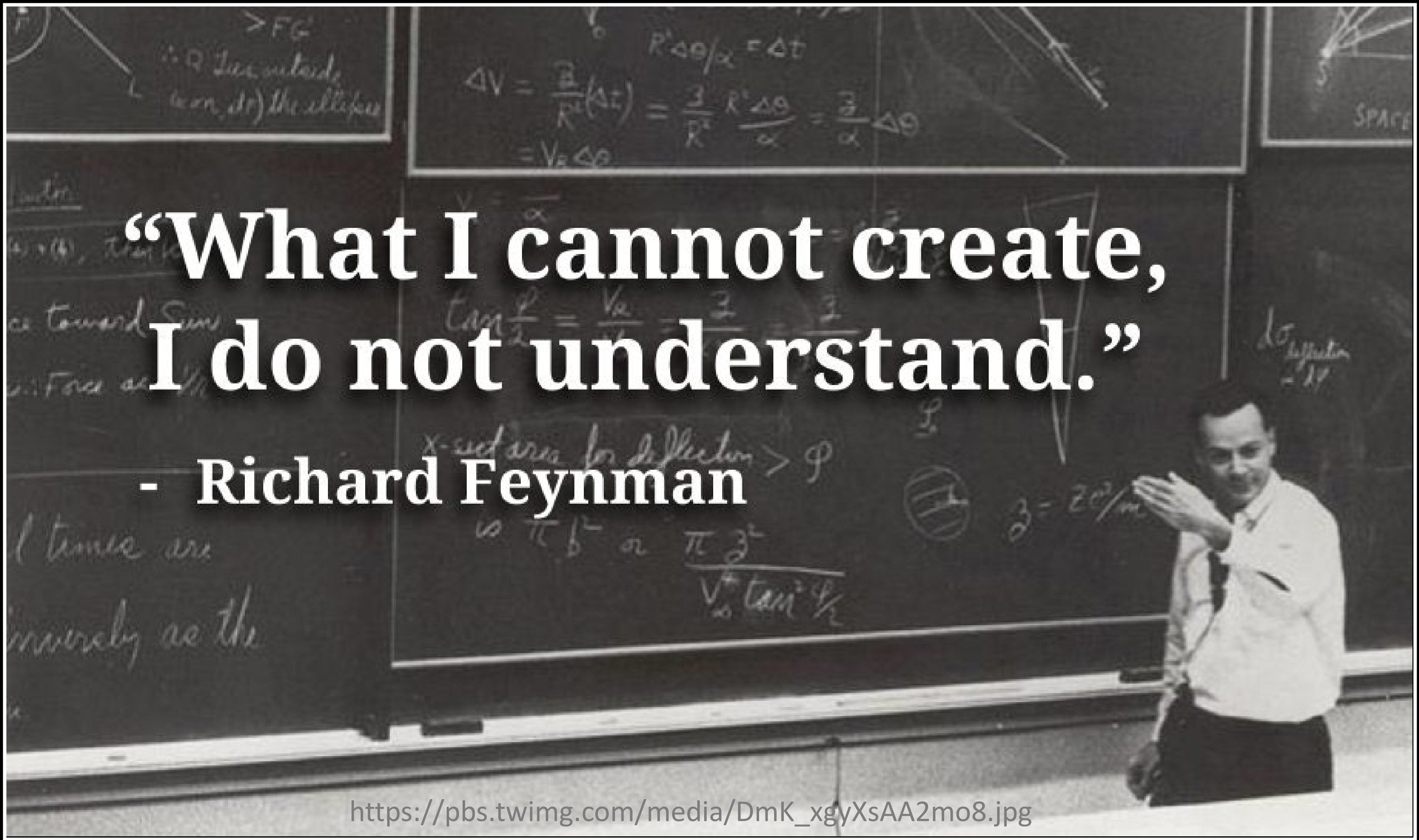
@The University of Sheffield

Week 9 Contents / Objectives

- Why Generative Models?
- Bayesian Inference
- Bayesian Linear Regression
- Variational Autoencoder (VAE)
- VAE Unboxing

Week 9 Contents / Objectives

- **Why Generative Models?**
- Bayesian Inference
- Bayesian Linear Regression
- Variational Autoencoder (VAE)
- VAE Unboxing



The holy grail in ML: understand data → create data

Generating Faces (VAE)

https://www.youtube.com/watch?v=XNZIN7Jh3Sg

Digital Generative Art (VAE)

https://blog.otoro.net/2016/04/01/generating-large-images-from-latent-vectors/

Generating Images (GAN)

https://www.youtube.com/watch?v=XOxxPcy5Gr4

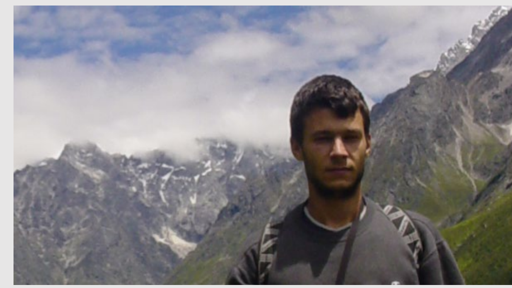
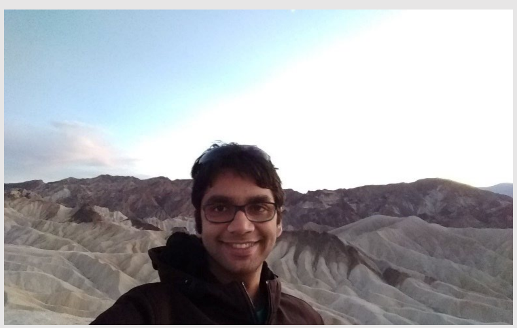
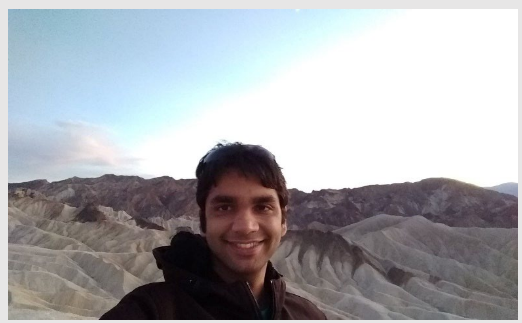
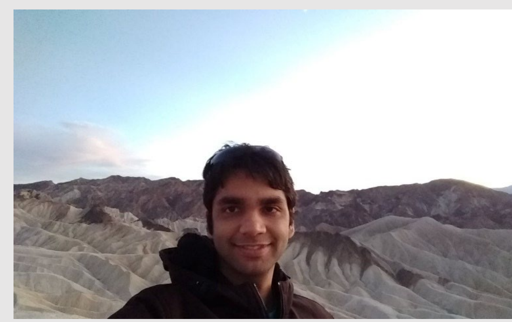
DeepFakes

Which image is real?

DeepFakes

Neither!



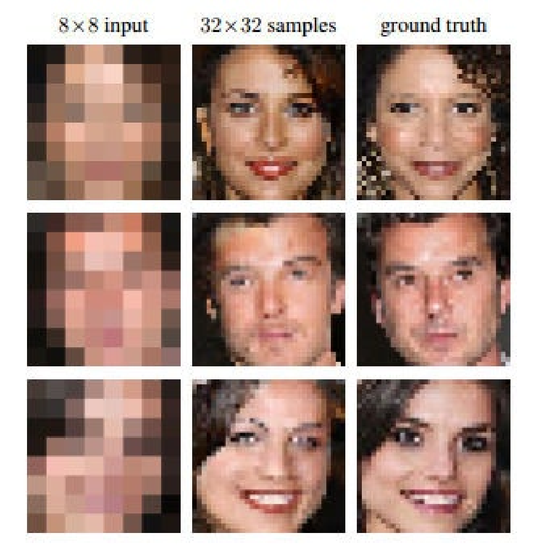
No glasses!

No smile!

Image Super Resolution

- Conditional generative model

P(high res image | low res image)

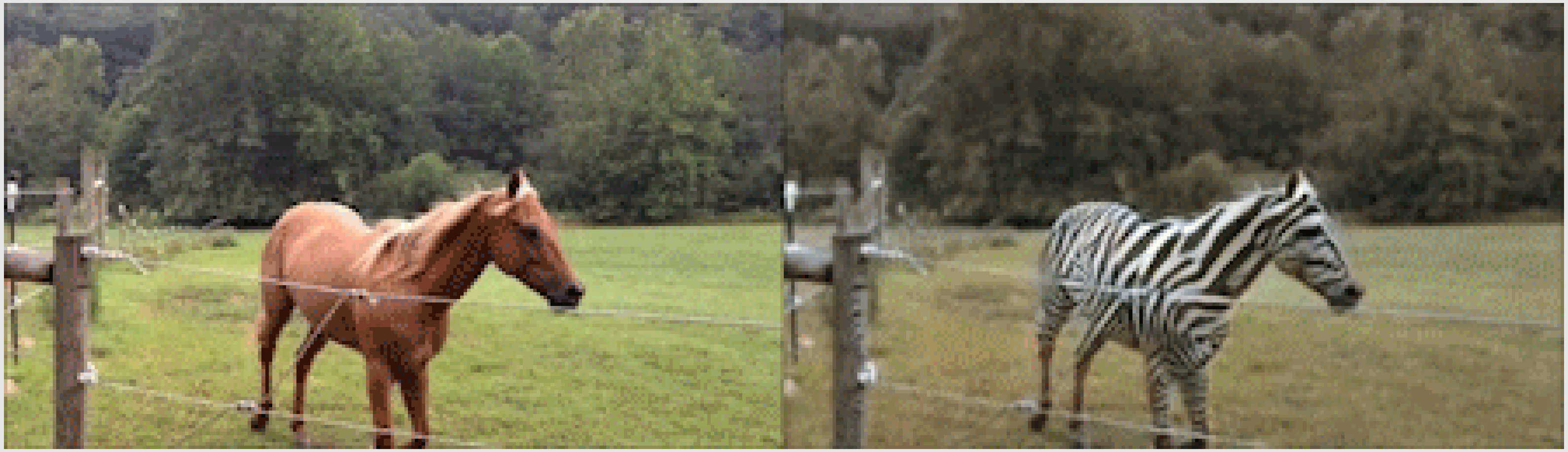


Ledig et al., 2017

Image Translation / Colorization

- Conditional generative model

P(zebra images| horse images)



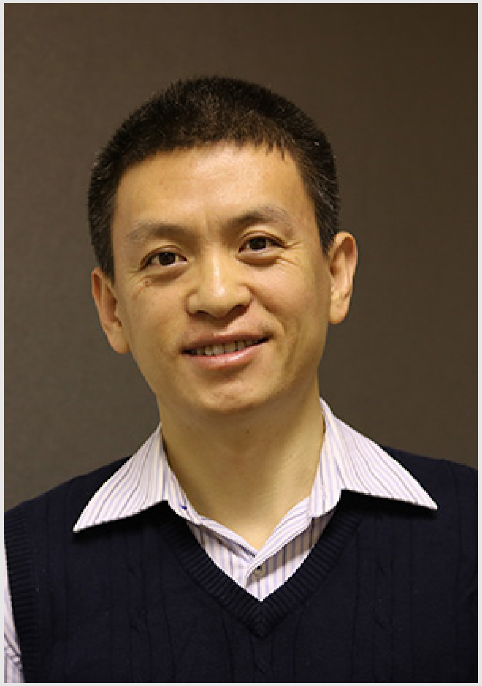
Zhu et al., 2017

Week 9 Contents / Objectives

- Why Generative Models?
- **Bayesian Inference**
- Bayesian Linear Regression
- Variational Autoencoder (VAE)
- VAE Unboxing

Question

- Which year was this photo taken?
  A. 1996
  B. 2006
  C. 2016
  D. 2026



Bayes' Rule

Given data x and parameters θ, their joint probability can be written as

$$p(\theta|x)p(x) = p(x,\theta) \qquad\qquad p(x,\theta) = p(x|\theta)p(\theta)$$

Eliminating $p(x,\theta)$ gives Bayes' rule:

$$\boxed{\underbrace{p(\theta|x)}_{\text{Posterior}} = \frac{\overbrace{p(x|\theta)}^{\text{Likelihood}}\,\overbrace{p(\theta)}^{\text{Prior}}}{\underbrace{p(x)}_{\text{Evidence}}}}$$

Key Concepts

- **Prior** probability: the estimate of the probability of the model **before** the data (evidence) is observed
- **Posterior** probability: the probability of the model **after** observing the data (evidence)
- **Likelihood**: the probability of observing a (random) data point given a model (*fixed*) →the **compatibility** of the data (evidence) with the given model
- **Marginal likelihood**: "model **evidence**", the probability of observing a (random) data point under all possible model variations

Principles of Bayesian Inference

⇨ Formulation of a generative model

likelihood $p(x|\theta)$

prior distribution $p(\theta)$

⇨ Observation of data

x

⇨ Update of beliefs based upon observations, given a prior state of knowledge

$$p(\theta|x) \propto p(x|\theta)p(\theta)$$

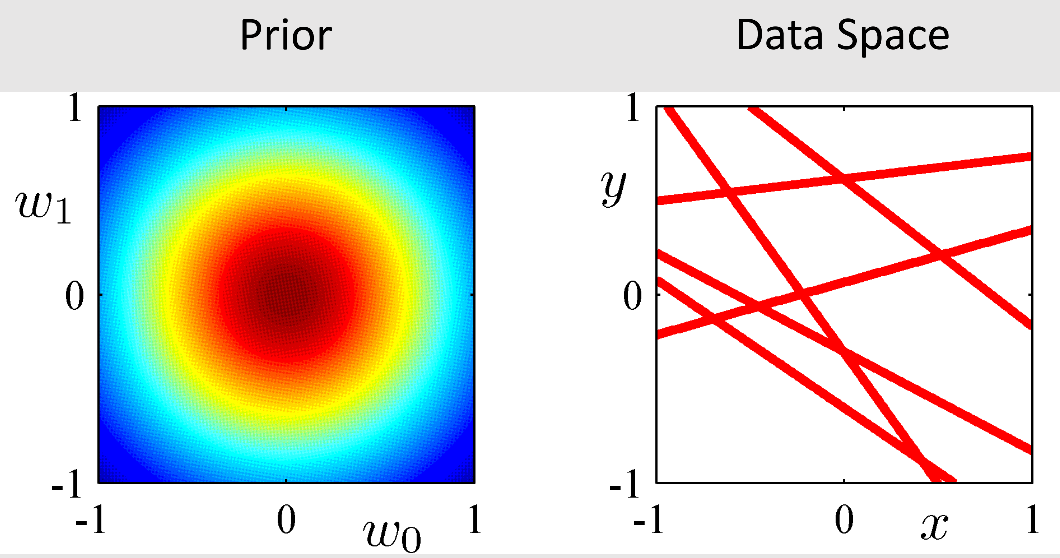
Week 9 Contents / Objectives

- Why Generative Models?
- Bayesian Inference
- **Bayesian Linear Regression**
- Variational Autoencoder (VAE)
- VAE Unboxing

Bayesian Linear Regression (1)

Aim: Estimate model parameters w_0 & w_1

Six samples of *y(x, **w**)*



Bayesian inference: placing a probability distribution (prior density) over the model parameters w_0 & w_1

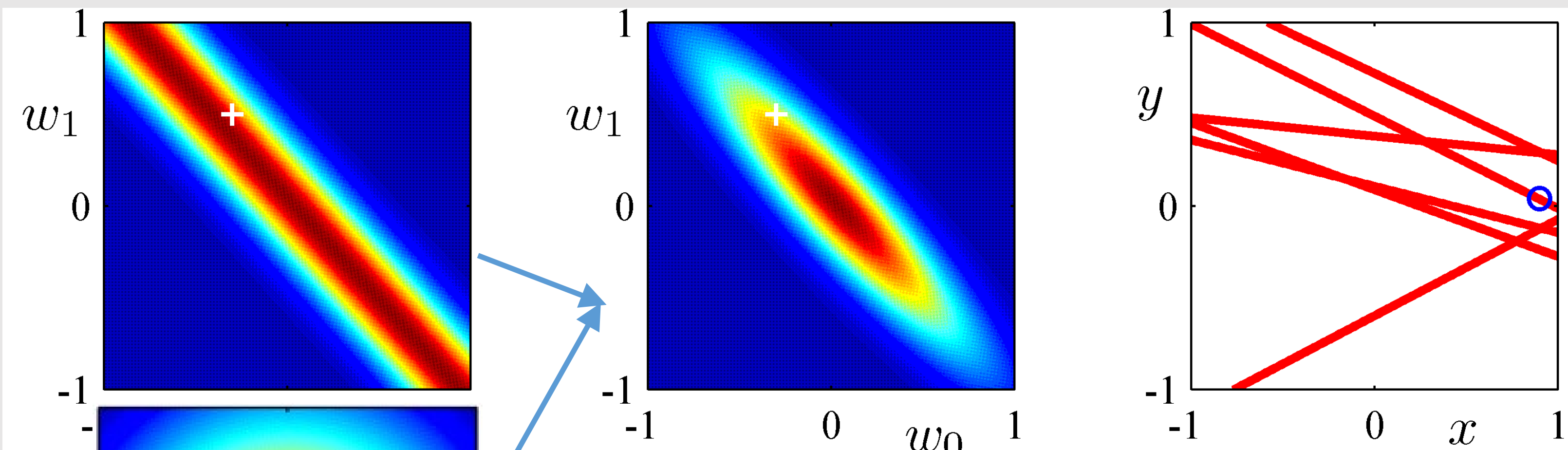
Now: No data points are observed.

Bayesian Linear Regression (2)

1 data point observed → soft constraint.

This posterior → prior for the next data point observed

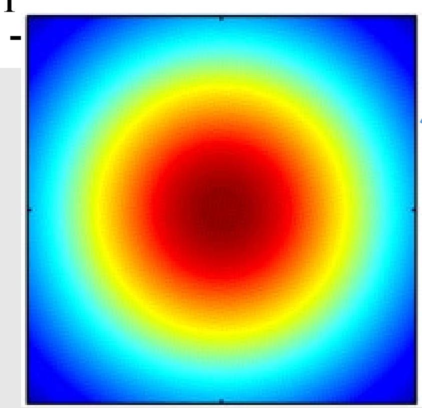
Likelihood Posterior Data Space



$y=w_1x + w_0$

(0.9, 0.1)

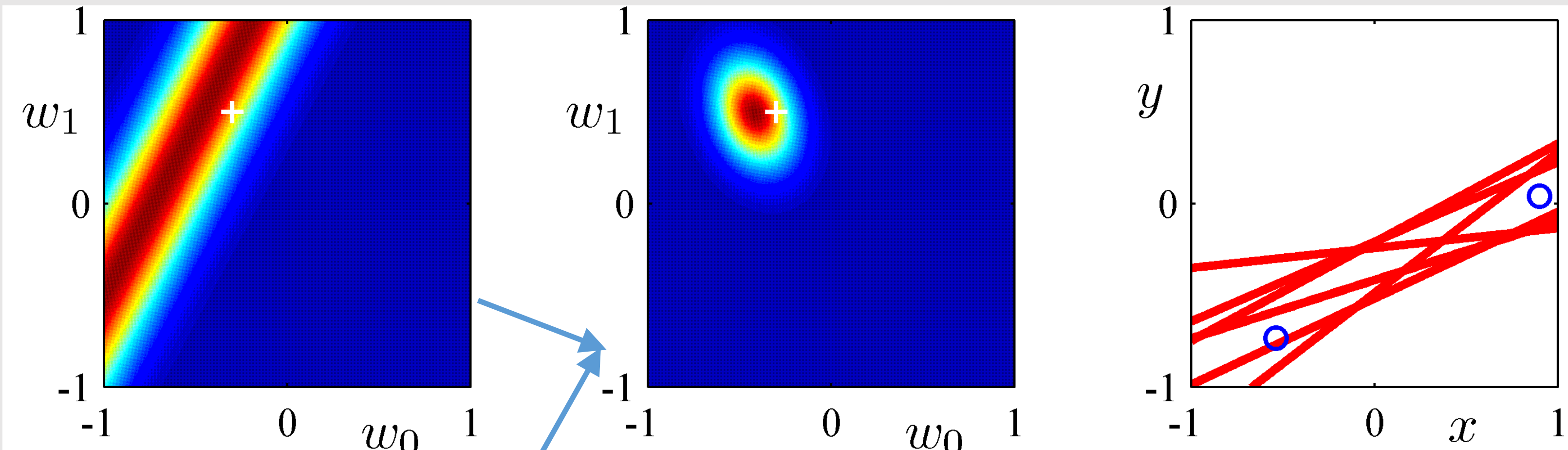
Prior



Bayesian Linear Regression (3)

A second data point observed

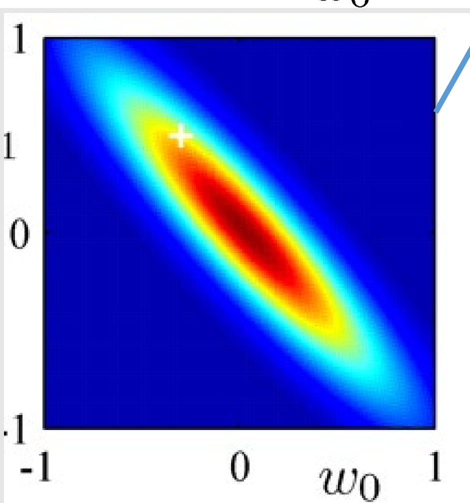
Likelihood of 2nd pt | Posterior | Data Space



$y = w_1 x + w_0$

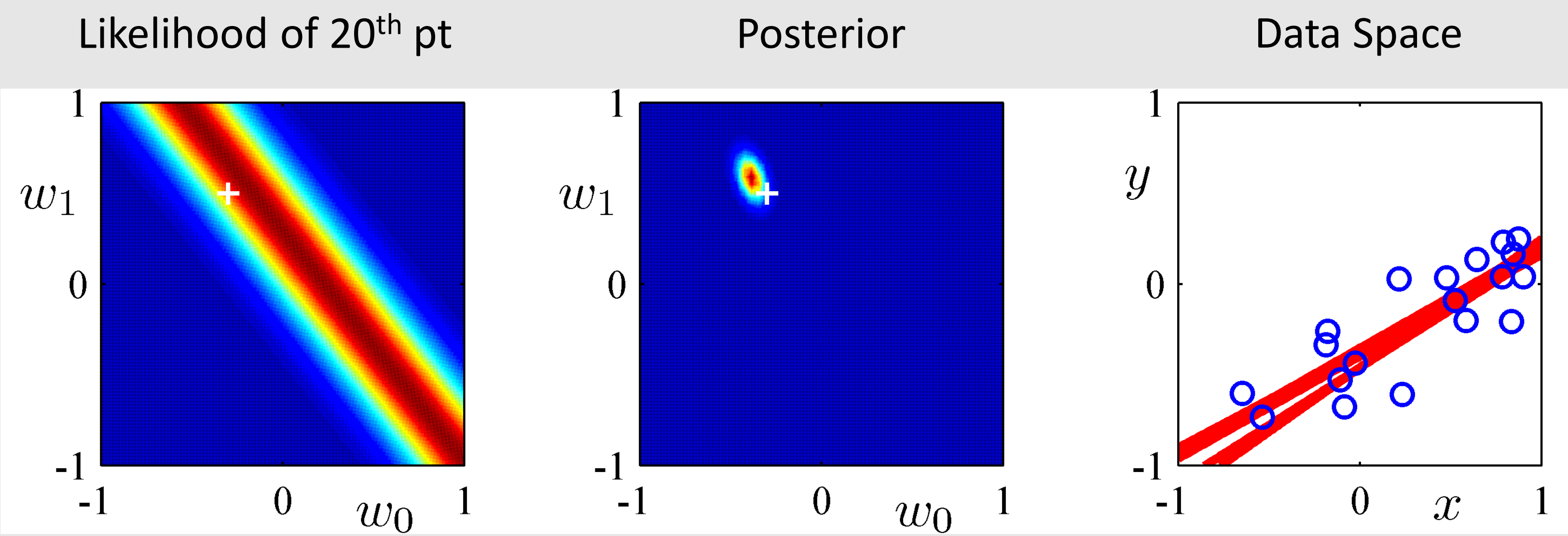
(-0.7, -0.8)

Current Prior = Previous posterior



Bayesian Linear Regression (4)

20 data points → very close to true values of w_0 & w_1

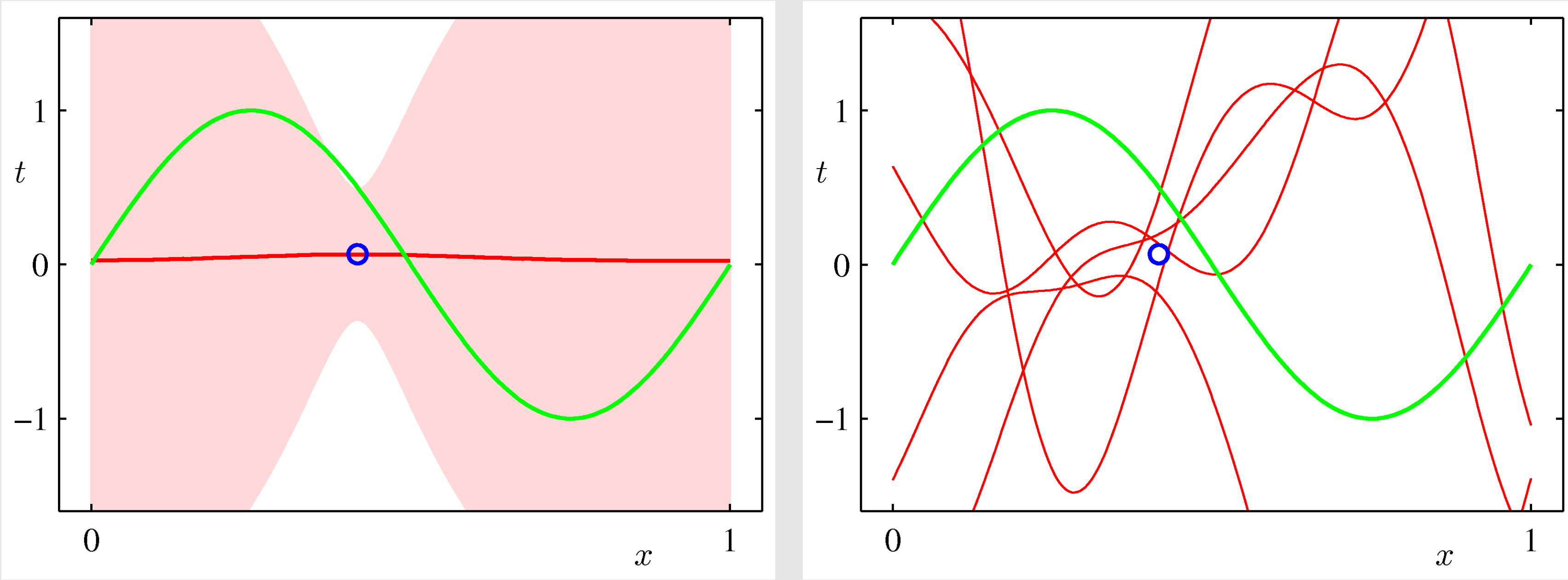


How about making **probabilistic** predictions for any x?

Bayesian inference: Evaluate the predictive **distribution**

Predictive Distribution (1)

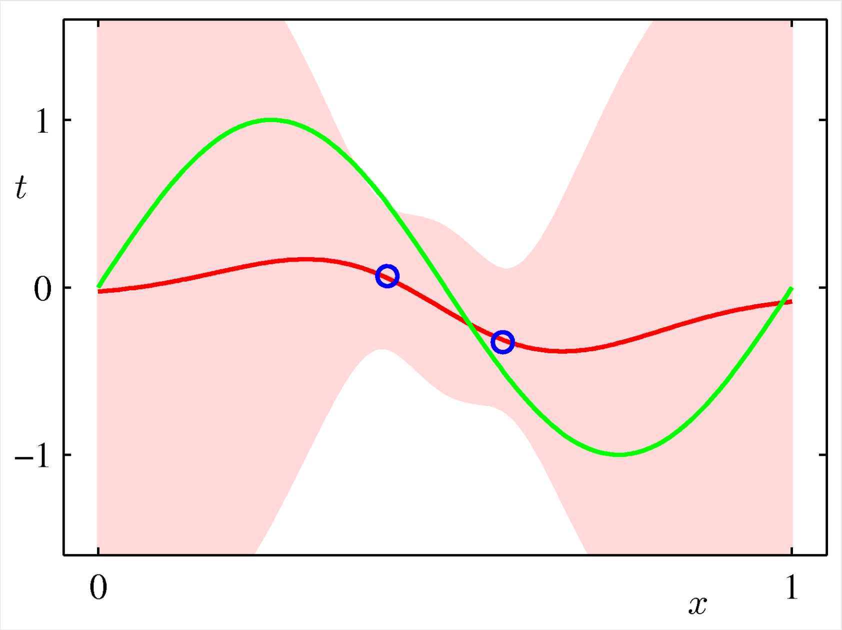
- Data: green curve + noise → sinusoidal data (blue circles)
- Model: 9 Gaussian basis functions



- Aim: Predict the output distribution
- Now: 1 data point. Red: model; shade: model uncertainty

Predictive Distribution (2)

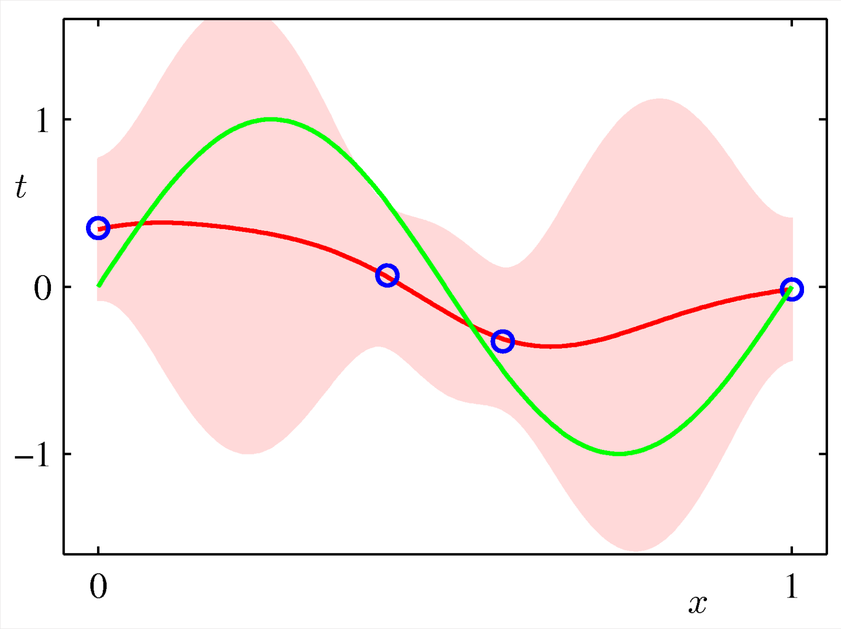
- 2 data points observed → reduced uncertainty near the points

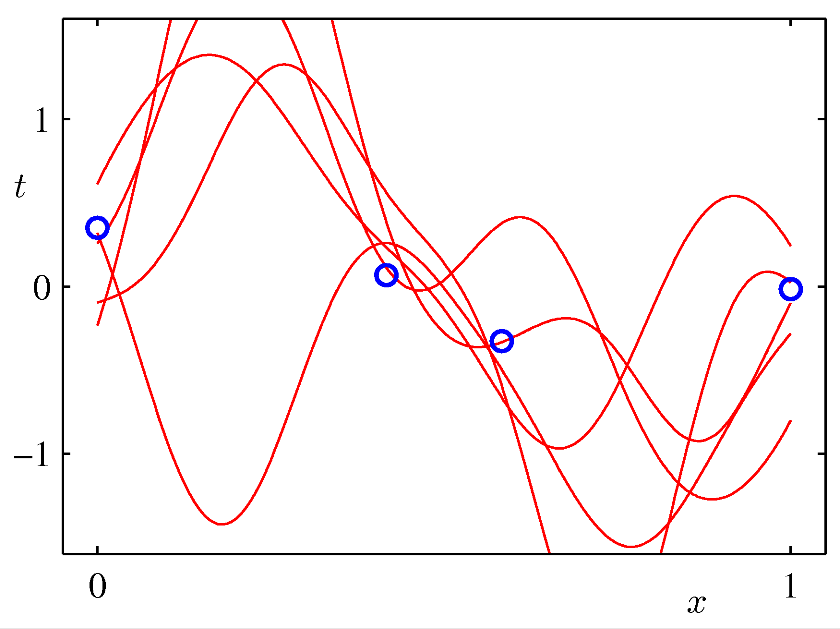


- Left: the predictive distribution
- Right: samples from the predictive distribution

Predictive Distribution (3)

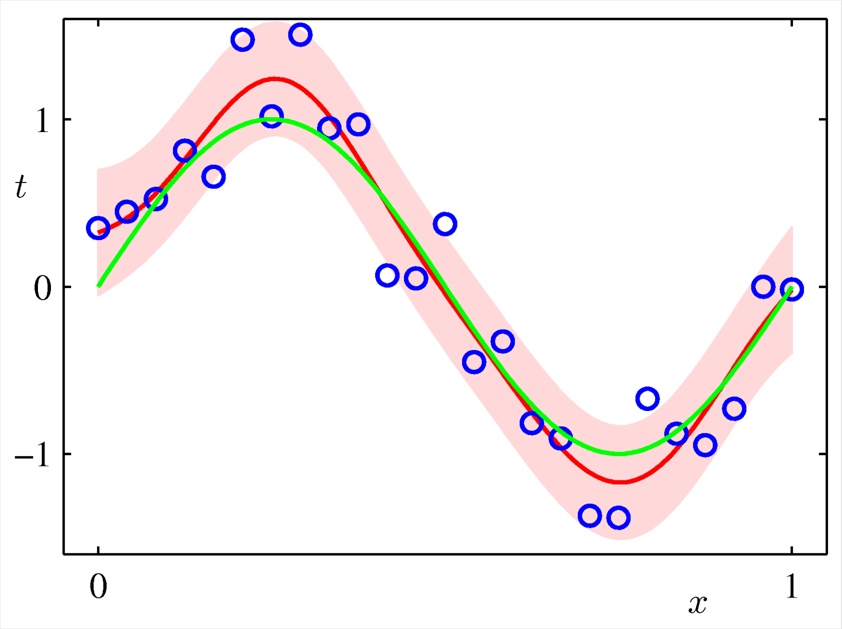
- 4 data points observed → further reduced uncertainty near the points

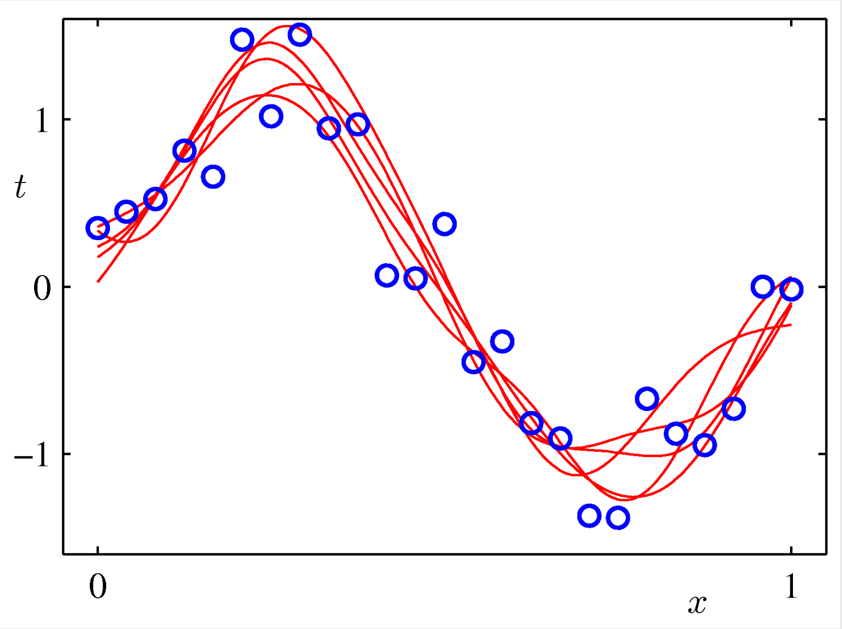




Predictive Distribution (4)

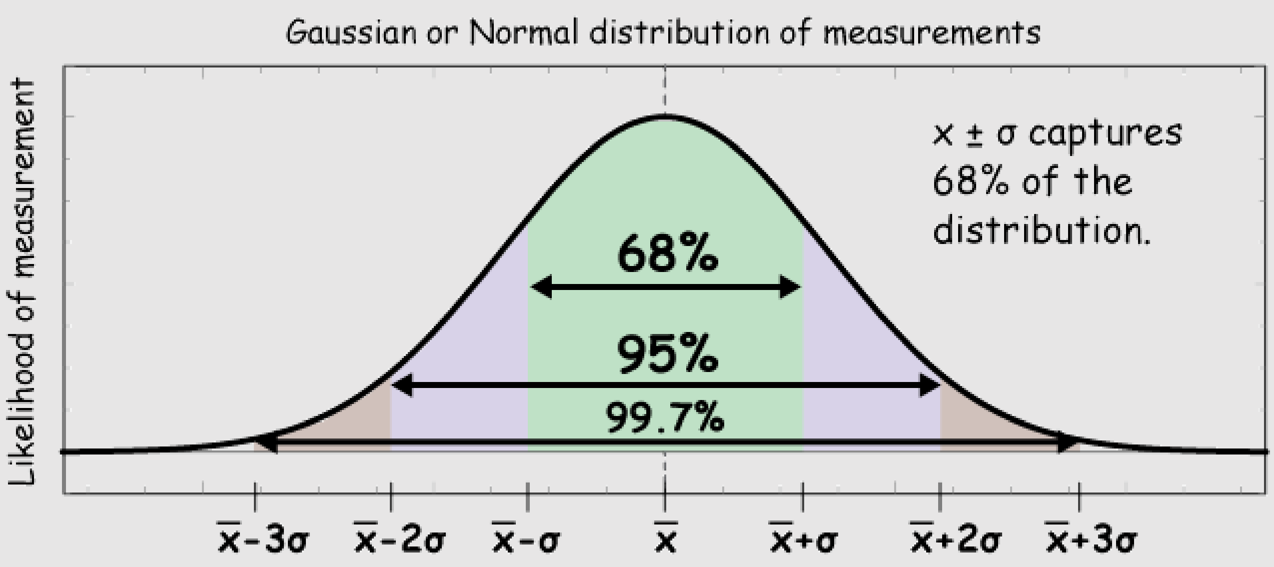
- 25 data points → significantly reduced uncertainty





Gaussian/Normal Distribution

- Knowing the **mean and (co)variance** (std) is sufficient to specify the distribution (*sufficient statistics*)
  - Closed form solution often feasible
- Density estimation: estimate mean and (co)variance



Bayesian Regression Ingredients

- Data: + pre-processing, e.g., $\mathcal{N}(0,1)$
- Model
  - Structure/Architecture: basis function chosen, e.g. poly, Gaussian
  - Hyper-parameter: for basis function (e.g., degree) & prior
  - Parameters (theta): weights and bias
- Evaluation metric: MSE
- Optimisation: closed form for Gaussian distributions, SGD etc. otherwise

Pros and Cons of Bayesian Methods

- Pros
  - Provide **uncertainty estimation**, e.g. predicting an output distribution with mean and (co)**variance**
  - Make use of more information (prior, if available)
  - Less overfitting in general

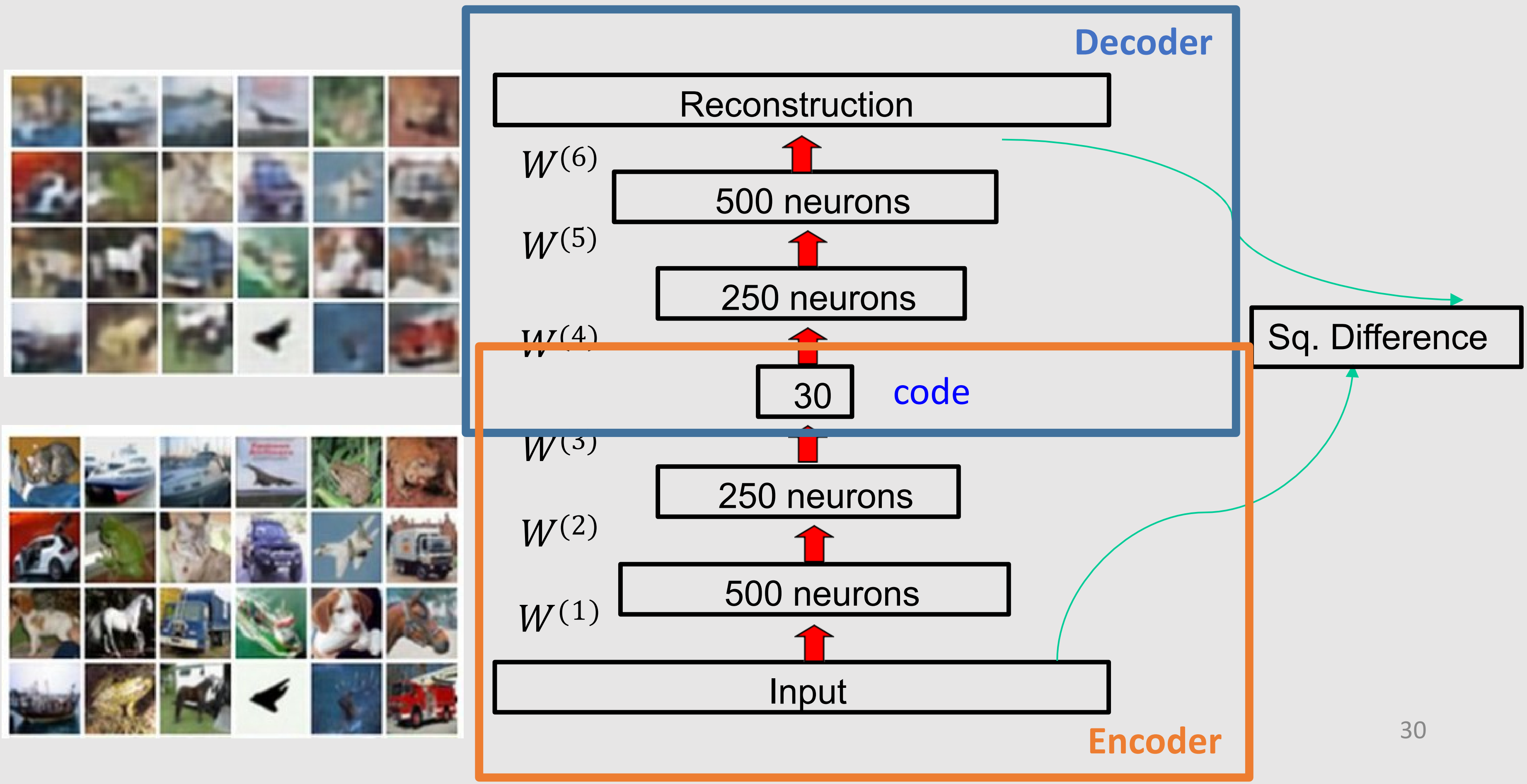
- Cons
  - Complexity
  - Subjectivity: all inferences are based on beliefs. Which prior to choose? If prior is wrong, …

Week 9 Contents / Objectives

- Why Generative Models?
- Bayesian Inference
- Bayesian Linear Regression
- **Variational Autoencoder (VAE)**
- VAE Unboxing

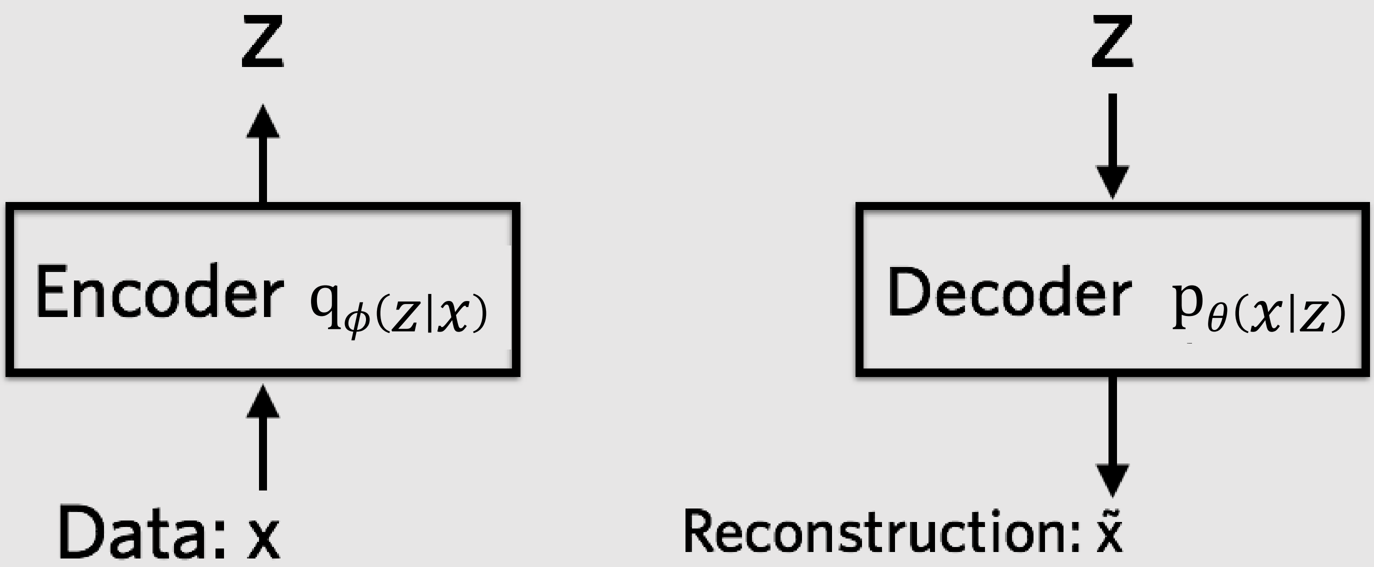
Autoencoders

- The **decoder** reproduces the input from a representation (the code) learned by the **encoder**



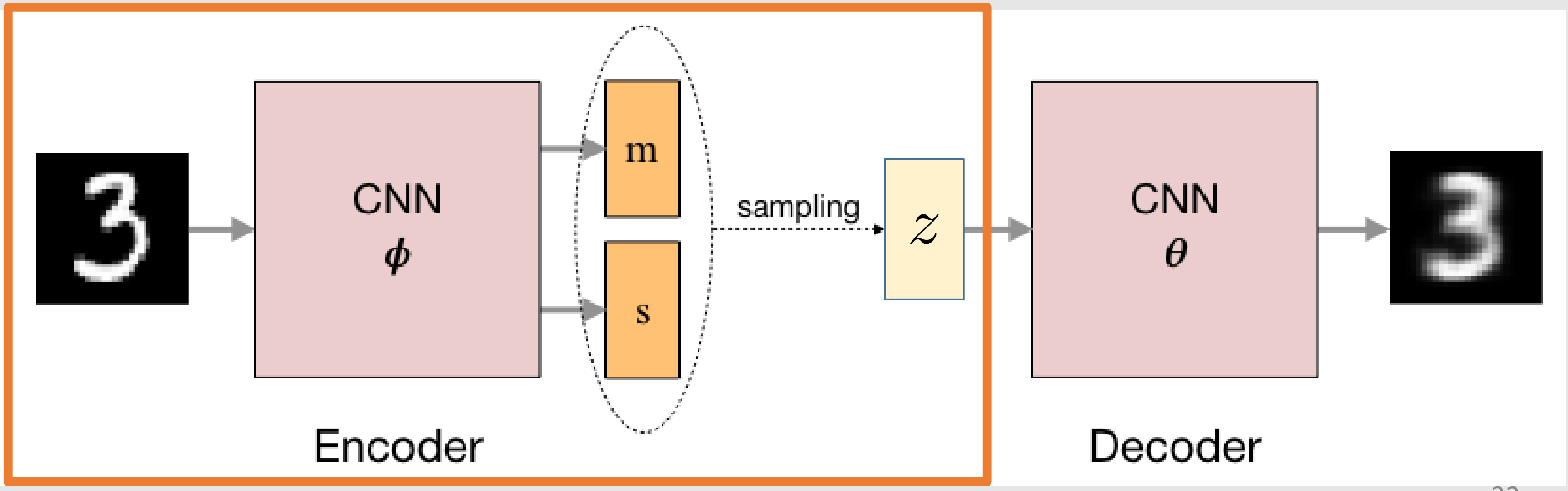
Variational Autoencoder (VAE)

- Make both the encoder and decoder **probabilistic**
- **Encoder**: draw latent variables z (the code) from a probability distribution conditioned on the input x
- **Decoder**: reconstruct x probabilistically conditioned on z



VAE Encoder

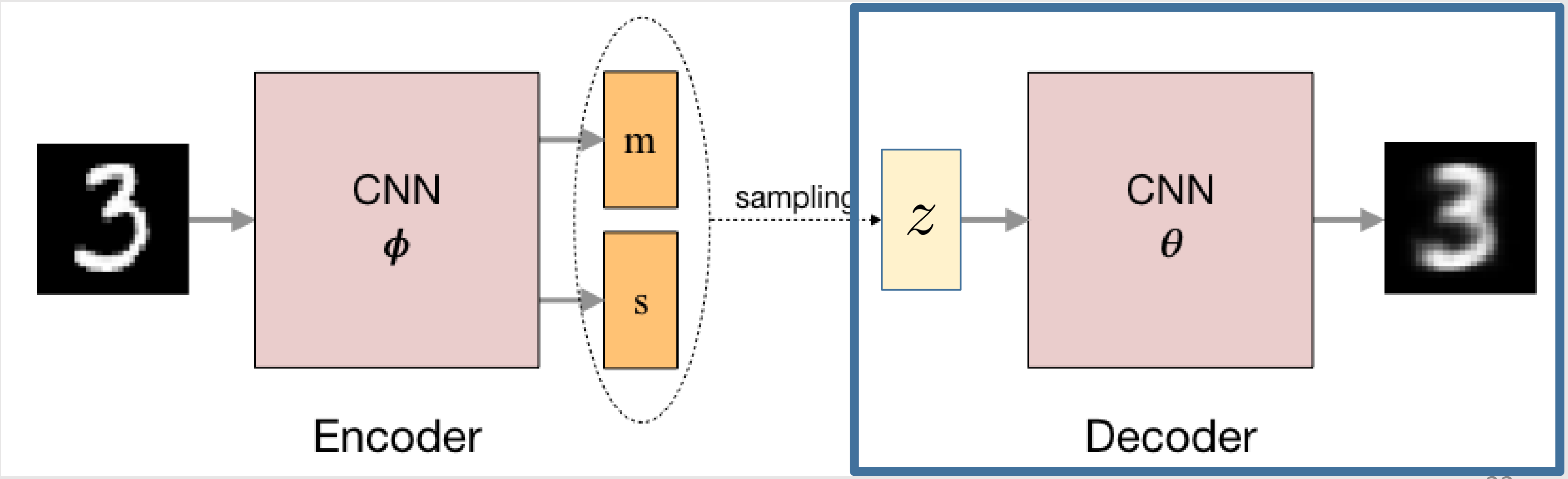
- Take the input x and output parameters for a probability distribution $q_\phi(z \mid x)$. For Gaussian: output the mean and standard deviation
  - Use a neural network with parameter ϕ to do this
- Sample from this distribution to get *random* values of the lower-dimensional representation z



https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

VAE Decoder

- Takes latent variable z and out parameters for a distribution $p_\theta(x \mid z)$, e.g. the mean and standard deviation for each pixel in the output
  - Use a neural network with parameter θ to do this
- Sample $p_\theta(x \mid z)$ to get the reconstruction $\tilde{x}$



https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

VAE Loss Function

- Objective: learn parameters of two probability distributions ϕ and θ
- For a single data point, the loss function is

$$l_i(\phi, \theta) = -\mathbb{E}_{z \sim q_\phi(z \mid x_i)}[\log p_\theta(x_i \mid z)] + \mathbb{KL}(q_\phi(z \mid x_i) \mid\mid p(z))$$

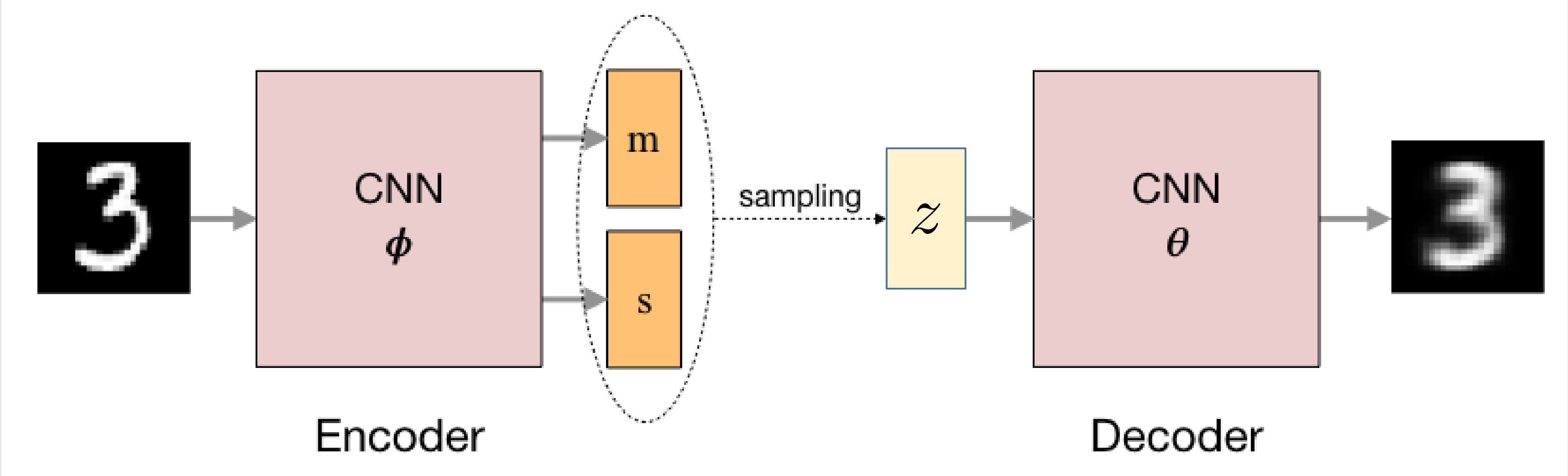
- Term #1: the expected negative log-likelihood → the reconstruction loss
- Term #2: a regularisation, the Kullback-Leibler divergence between the encoder's distribution $q_\phi(z \mid x)$ and the marginal distribution $p(z)$, measuring their mismatch
  - $q_\phi(z \mid x)$ is an approximation to the true posterior $p(z \mid x)$ based on variational inference, hence the name **variational**

https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

Optimization Challenge

- The expectation in the loss function will be approximated by choosing samples and averaging. This is not differentiable w.r.t. ϕ and θ.

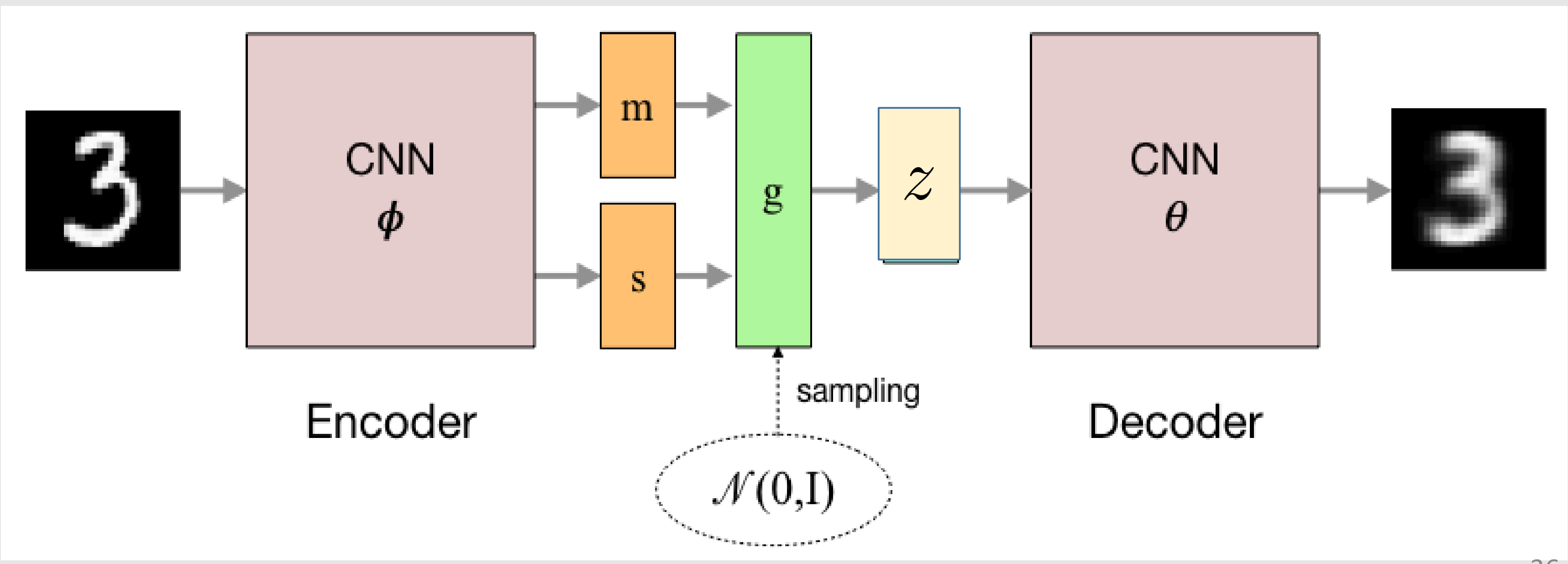
$$l_i(\phi, \theta) = -\mathbb{E}_{z \sim q_\phi(z|x_i)}[\log p_\theta(x_i \mid z)] + \mathbb{KL}(q_\phi(z \mid x_i) \mid\mid p(z))$$



https://nbviewer.jupyter.org/github/krasserm/bayesian-machine-learning/blob/master/variational_autoencoder.ipynb

Reparameterization Trick

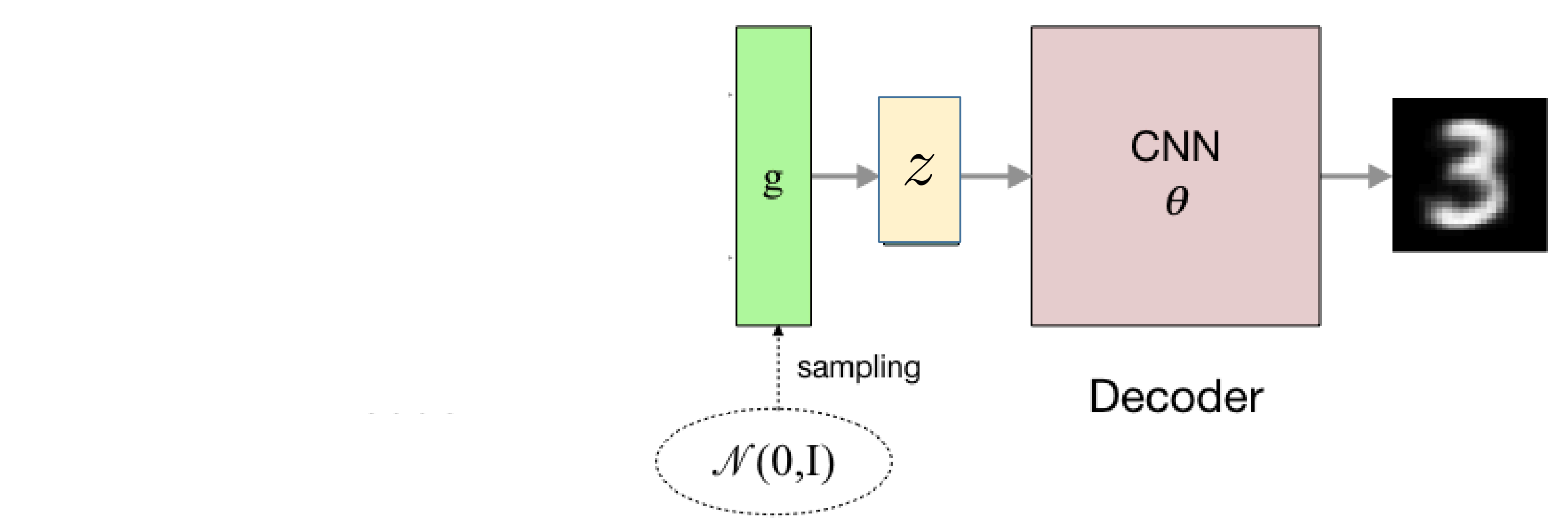
- If z is $N(\mu(x_i), \Sigma(x_i))$, then we can sample z using $z = \mu(x_i) + \sqrt{(\Sigma(x_i))}\, \epsilon$, where ϵ is $N(0,1)$. So we can draw samples from $N(0,1)$, which doesn't depend on the parameters.



https://nbviewer.jupyter.org/github/krasserm/bayesian-machine-learning/blob/master/variational_autoencoder.ipynb

Generative Mode of VAE

- After training, sample any z from $N(0,1)$ and decode it to get a sample of the entire data distribution $p(x)$
  → Generate new samples that look like the input but aren't in the input.

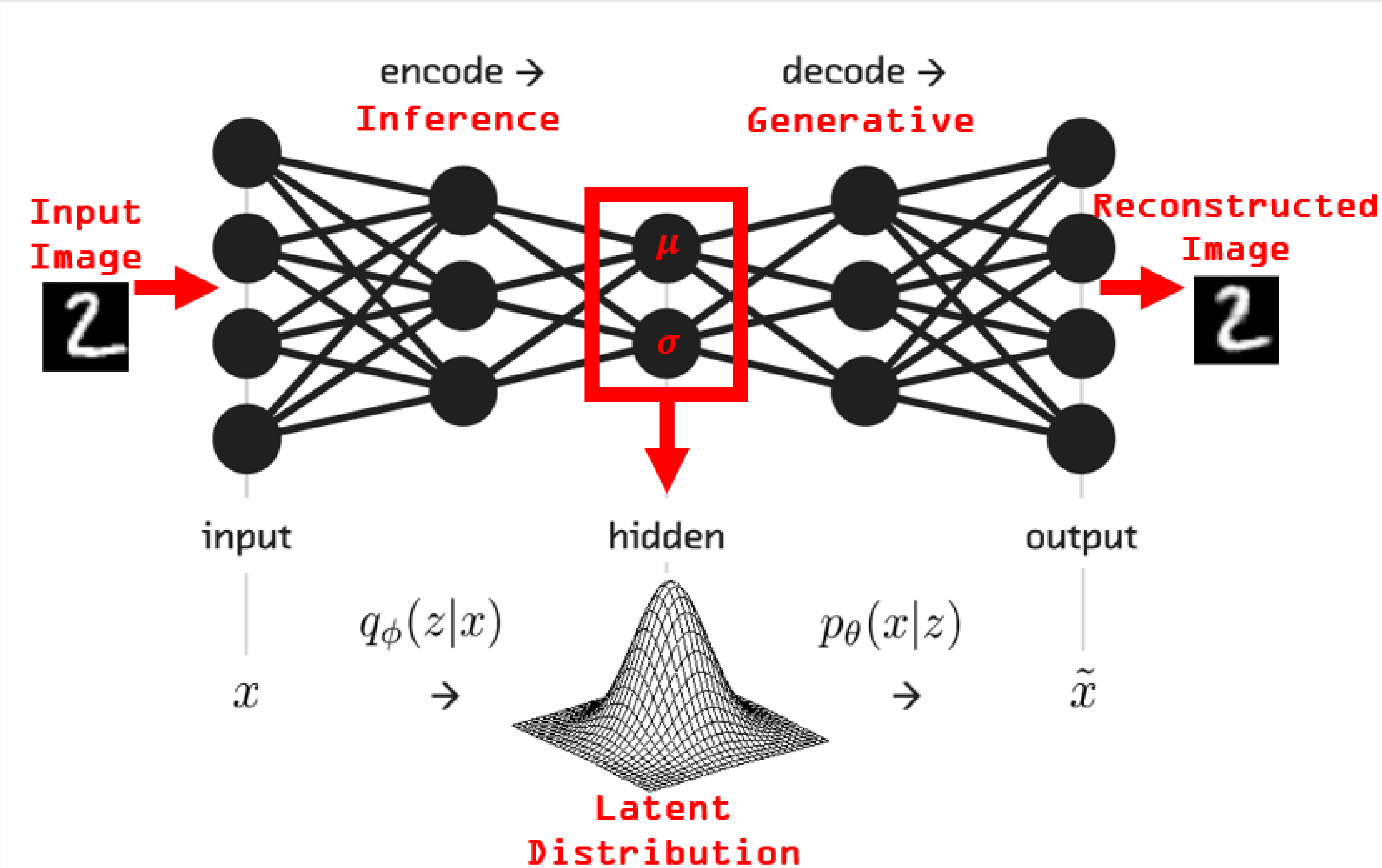


https://nbviewer.jupyter.org/github/krasserm/bayesian-machine-learning/blob/master/variational_autoencoder.ipynb

Week 9 Contents / Objectives

- Why Generative Models?
- Bayesian Inference
- Bayesian Linear Regression
- Variational Autoencoder (VAE)
- **VAE Unboxing**

Probabilistic Modelling in VAE

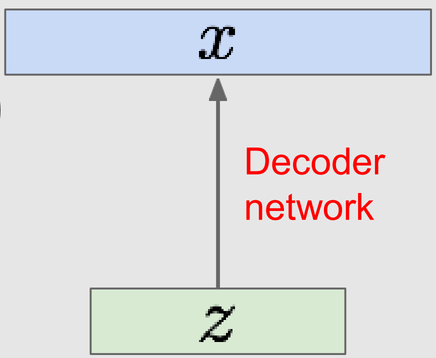


https://miro.medium.com/max/3080/1*82EghOQR2Z5uuwUjFiVV2A.png

Generative Modelling in VAE

Sample from the true likelihood

$$p_{\theta^*}(x \mid z^{(i)})$$



Sample from the true prior

$$p_{\theta^*}(z)$$

We want to estimate the true parameters θ^* of this generative model.

How should we represent this model?

Choose prior $p(z)$ to be simple, e.g. Gaussian.

Likelihood $p(x|z)$ is complex (generates image) → represent with a neural network

Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

Intractability Challenge

Evidence
(Marginal likelihood)

$$p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$$

Intractible to compute *p(x|z)* for every z!

Posterior also intractable: $p_\theta(z|x) = p_\theta(x|z)p_\theta(z)/p_\theta(x)$

Intractable evidence

Solution: Define an additional encoder network $q_\phi(z \mid x)$ that approximates $p_\theta(z \mid x)$ to make the problem tractable → the **variational** inference method

Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

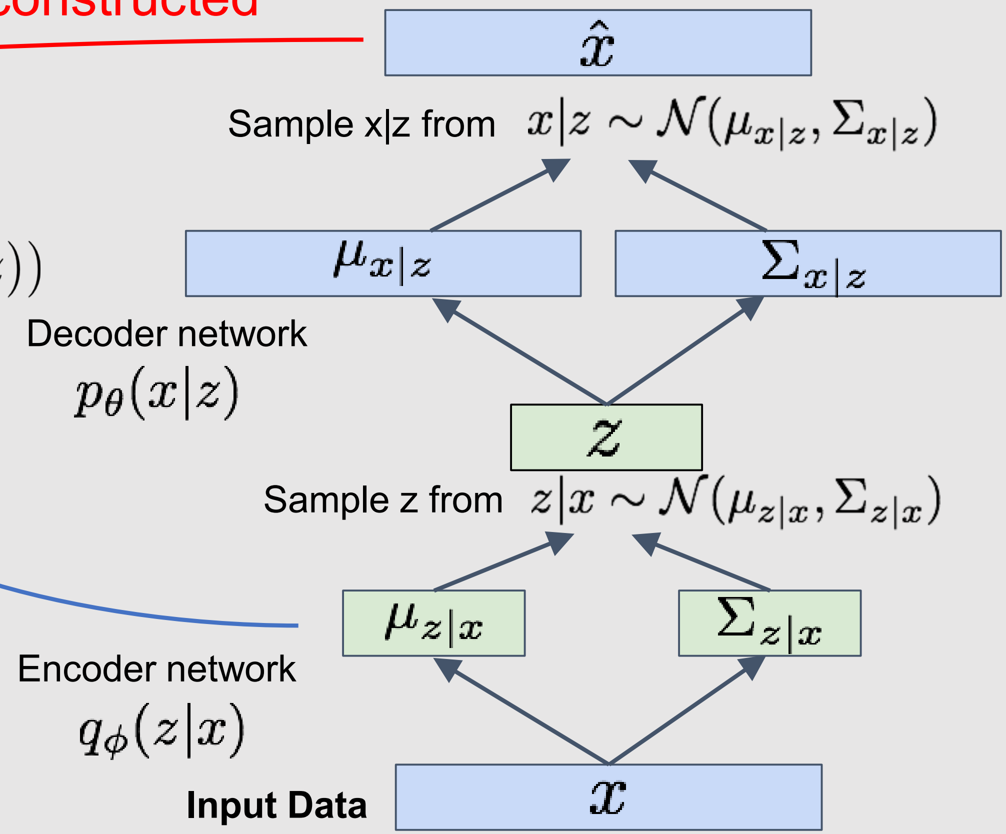
Variational Autoencoder Construction

Maximize likelihood of original input being reconstructed

Objective: maximise the Evidence Lower BOund (ELBO)

$$\mathbb{E}_z[\log p_\theta(x_i \mid z)] - \mathbb{KL}(q_\phi(z \mid x_i) \mid\mid p(z))$$

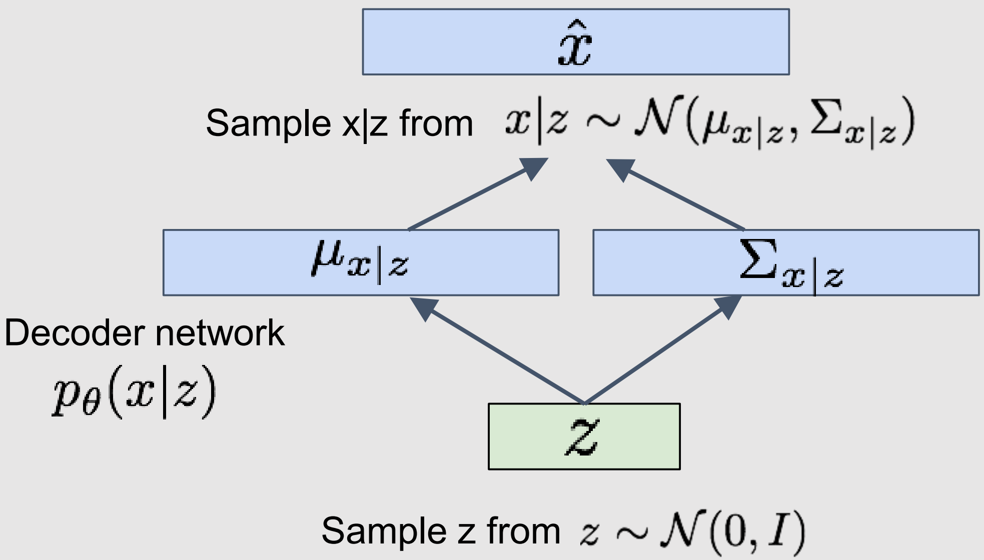
Make approximate posterior distribution close to prior to minimise the KL divergence



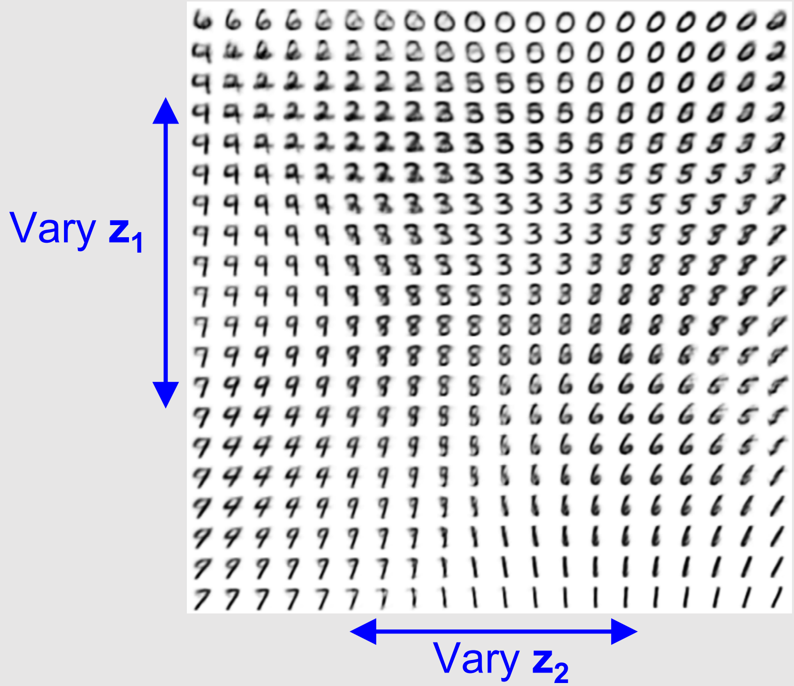
Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Generating Data with VAE

Use decoder network. Sample z from prior.



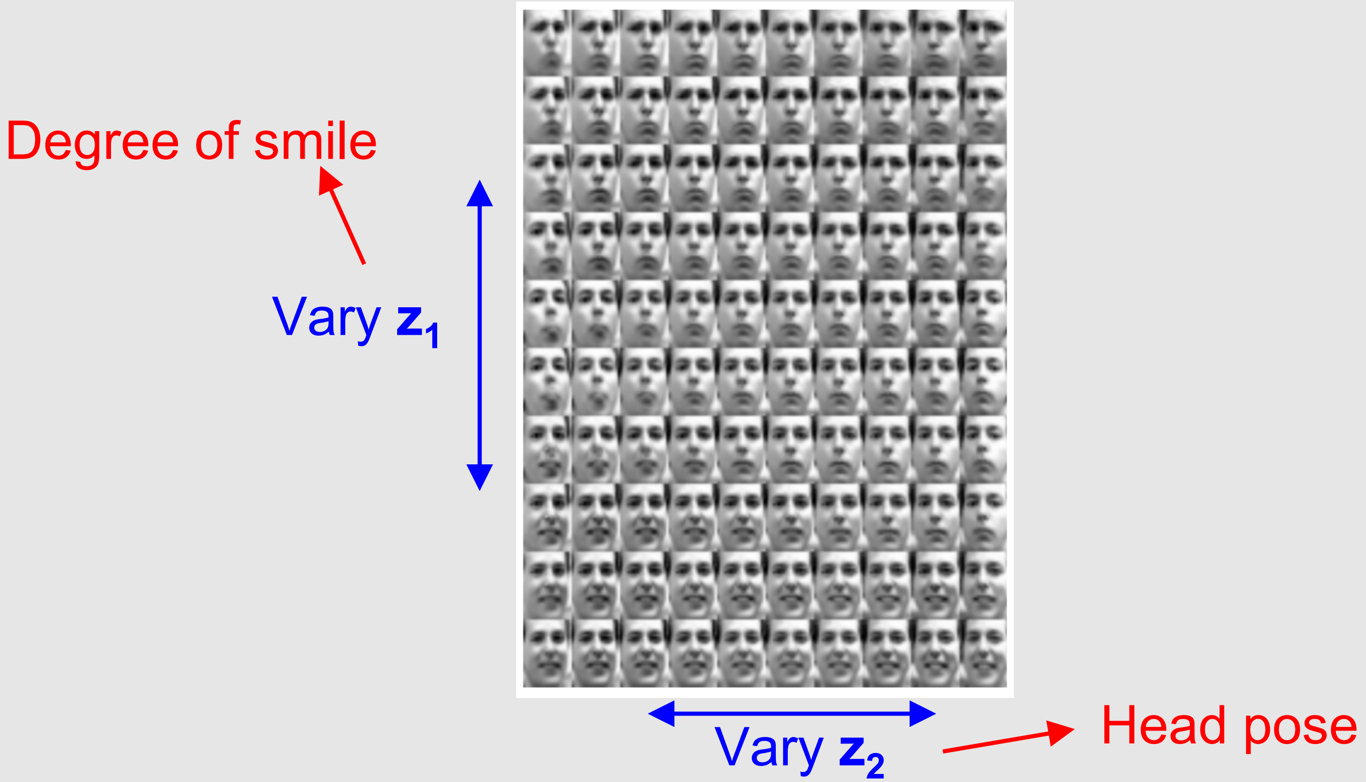
Data manifold for 2-d **z**



Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Face Generation & Interpretation



- Diagonal prior on **z** → independent latent variables
- Different dimensions of **z** encode interpretable factors of variation

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Variational Autoencoder Ingredients

- Data: + pre-processing, e.g., $\mathcal{N}(0,1)$
- Model
  - Structure/Architecture: layered network
  - Hyper-parameter: layer specs, e.g. #layers #units, (convolutional) kernel size
  - Parameters (theta): layer weights and biases
- **Evaluation metric: max evidence lower bound**
- Optimisation: backprop, SGD or the like

Pros and Cons of VAE

- Pros
  - Principled approach to generative models
  - Inference of $q(z|x)$ → useful feature representation for other tasks
- Cons
  - Samples blurrier and lower quality compared to state-of-the-art (GANs)

Acknowledgement

- The slides used materials from: *Christopher Bishop, Neil Lawrence, Lee Harrison, John Gosling, Chuck Huber, Greg Buzzard, Mike Mozer, Stefano Ermon, Aditya Grover, Martin Krasser, Dhruv Batra, Fei-Fei Li, Justin Johnson, Serena Yeung*

Recommended Reading

- PRML book: Section 3.3 on Bayesian Linear Regression
- CS231n: Convolutional Neural Networks for Visual Recognition from Stanford (Lecture 11-2020)
- CS236: Deep Generative Models @Stanford
- Wikipedia entries on related topics
- The lab notebook and references

Next

Lab notebooks

Feedback (if any)