Introduction to Deep Generative Modeling

COMPSCI 589 - Summer 2024

Sajjad Amini

Manning College of Information and Computer Sciences
University of Massachusetts Amherst

Disclaimer

- The financial examples and data presented are for illustrative purposes only.

Image Sources

- All images in this presentation were generated using ChatGPT unless otherwise cited.
- Each image has been created to visually enhance the topics discussed and provide illustrative support.
- For images not generated by ChatGPT, sources are cited directly in the title.

Contents

Section 1

Intuition

Investment Challenge

Figure: Investment Challenge (Budget: *$1M*, Divesting is not allowed)

Investment Challenge

Figure: Investment Challenge (Budget: *$1M*, Divesting is not allowed)

Investment Challenge

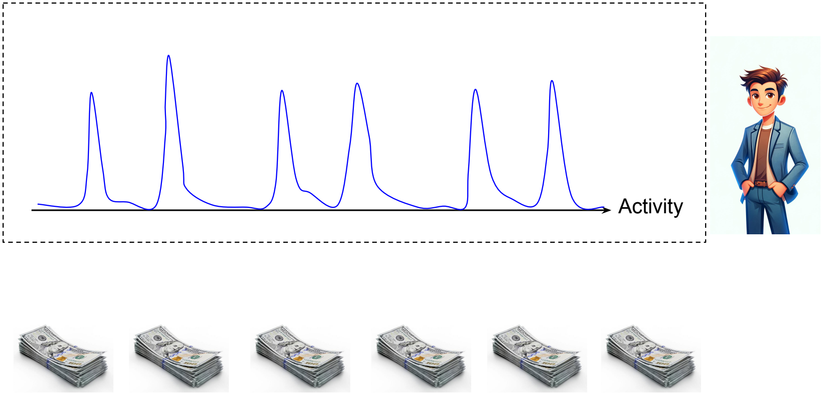


Figure: Investment Challenge (Budget: *$1M*, Divesting is not allowed)

Investment Challenge

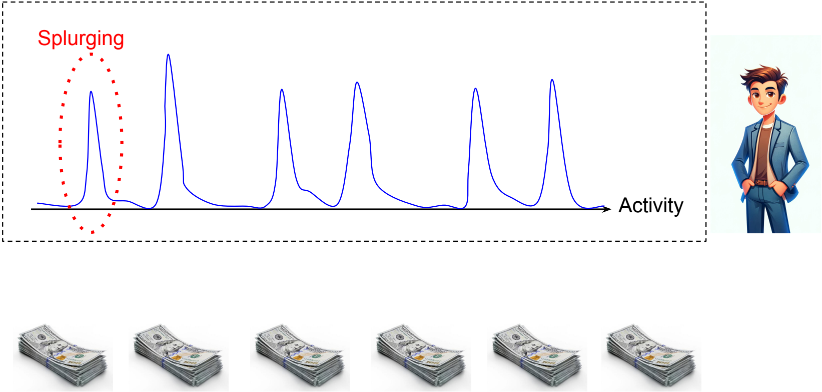


Figure: Investment Challenge (Budget: *$1M*, Divesting is not allowed)

Investment Challenge

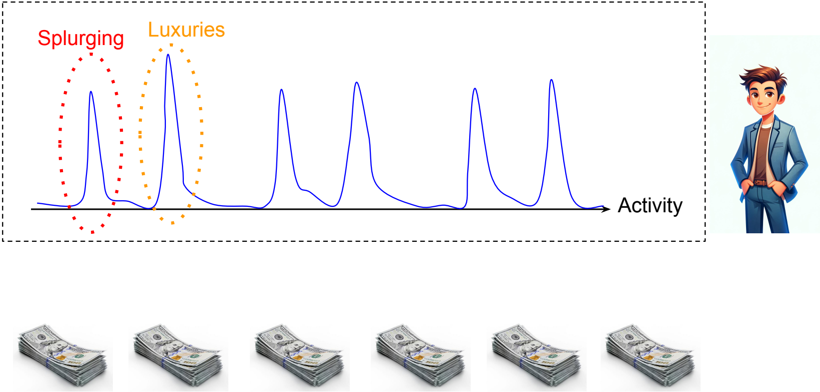


Figure: Investment Challenge (Budget: *$1M*, Divesting is not allowed)

Investment Challenge

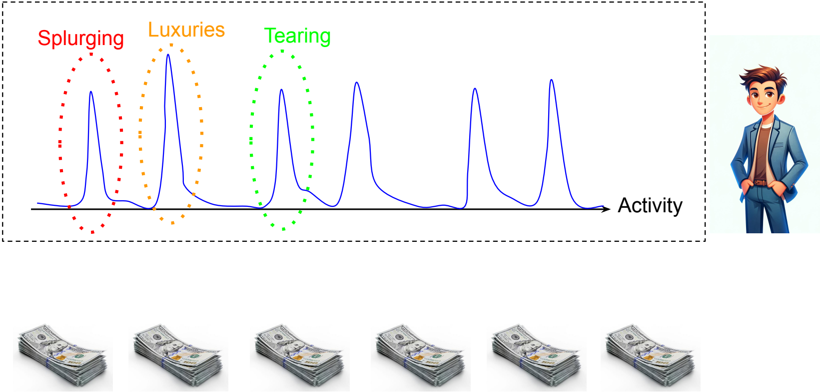


Figure: Investment Challenge (Budget: *$1M*, Divesting is not allowed)

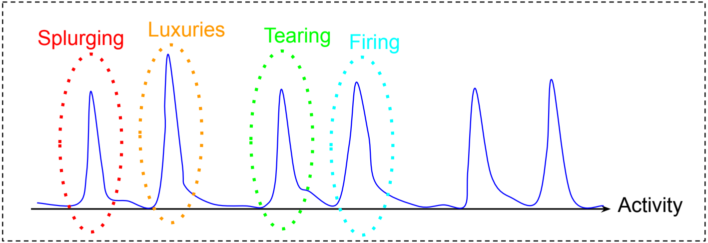


Figure: Investment Challenge (Budget: *$1M*, Divesting is not allowed)

Investment Challenge

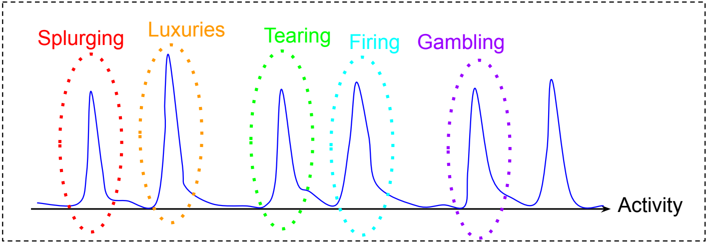


Figure: Investment Challenge (Budget: *$1M*, Divesting is not allowed)

Investment Challenge

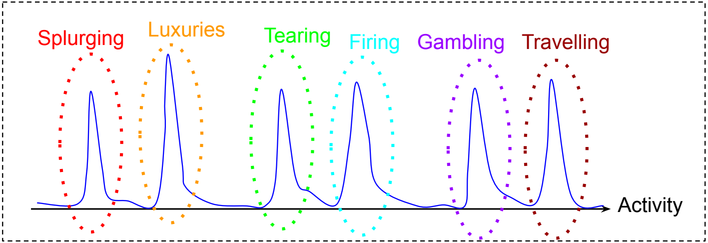


Figure: Investment Challenge (Budget: *$1M*, Divesting is not allowed)

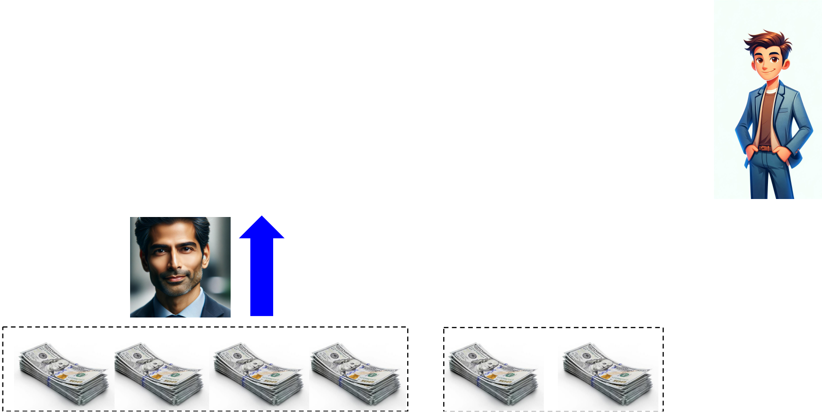


Figure: Investment challenge with help of investor (Budget: *$1M*, Divesting is not allowed)

Investment Challenge

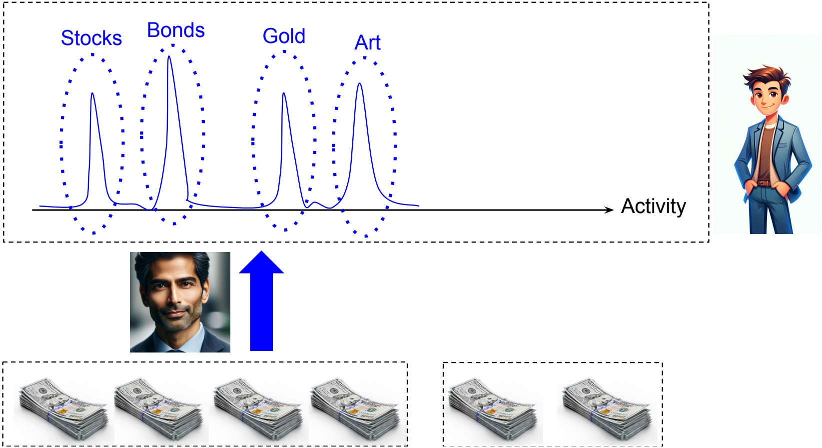


Figure: Investment challenge with help of investor (Budget: *$1M*, Divesting is not allowed)

Investment Challenge

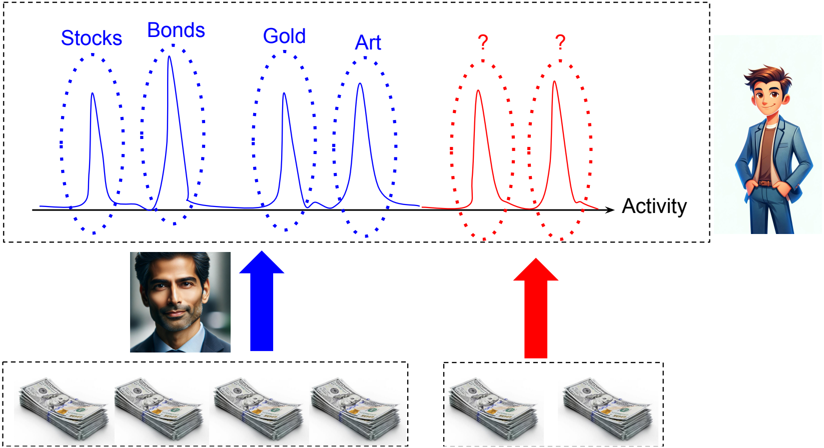


Figure: Investment challenge with help of investor (Budget: *$1M*, Divesting is not allowed)

Investment Challenge

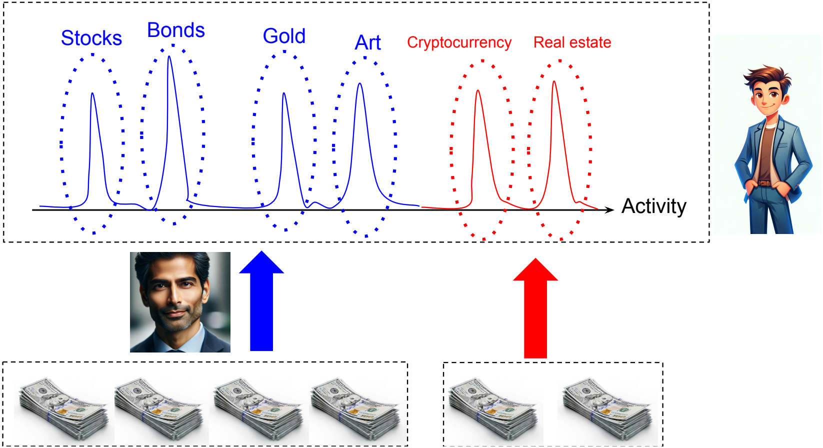


Figure: Investment challenge with help of investor (Budget: *\$1M*, Divesting is not allowed)

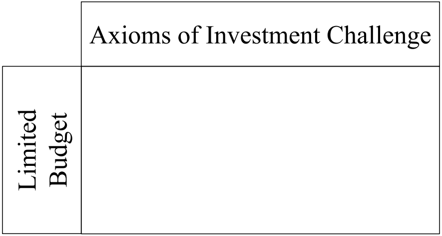


Figure: From Axioms of our challenge to Axioms of probability

Investment Challenge

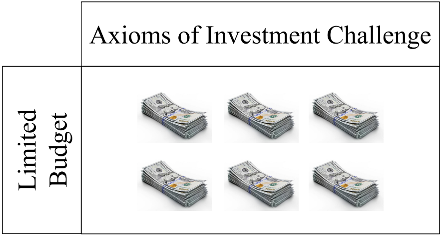


Figure: From Axioms of our challenge to Axioms of probability

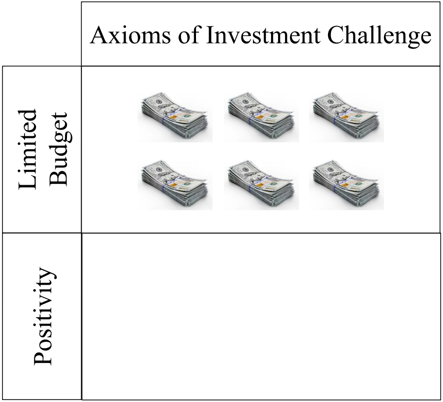


Figure: From Axioms of our challenge to Axioms of probability

Investment Challenge

	Axioms of Investment Challenge
Limited Budget	
Positivity	INVEST DIVEST

Figure: From Axioms of our challenge to Axioms of probability

Investment Challenge

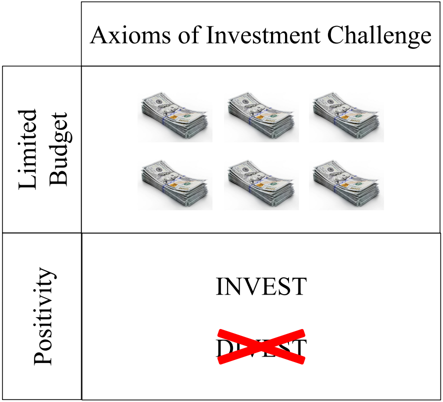


Figure: From Axioms of our challenge to Axioms of probability

	Axioms of Investment Challenge	Axioms of Probability
Limited Budget		
Positivity	INVEST ~~DIVEST~~	

Figure: From Axioms of our challenge to Axioms of probability

	Axioms of Investment Challenge	Axioms of Probability
Limited Budget		$$\int_{\boldsymbol{x}} p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = 1$$
Positivity	INVEST ~~DIVEST~~	

Figure: From Axioms of our challenge to Axioms of probability

Investment Challenge

	Axioms of Investment Challenge	Axioms of Probability
Limited Budget		$\int_{\boldsymbol{x}} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1$
Positivity	INVEST ~~DIVEST~~	$p(\boldsymbol{x}) \geq 0$

Figure: From Axioms of our challenge to Axioms of probability

Section 2

Concept

Parametric Probability Density Function (PDF)

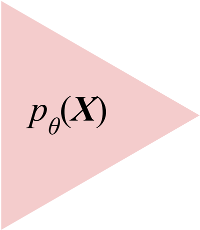


Figure: Your new budget is your parametric PDF

Parametric Probability Density Function (PDF)

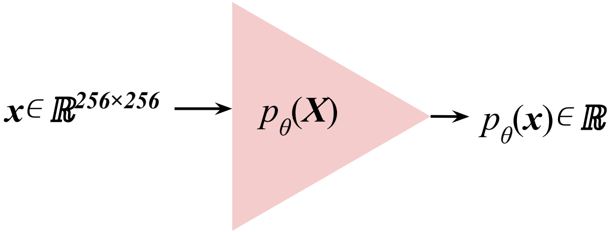


Figure: Your new budget is your parametric PDF

Parametric Probability Density Function (PDF)

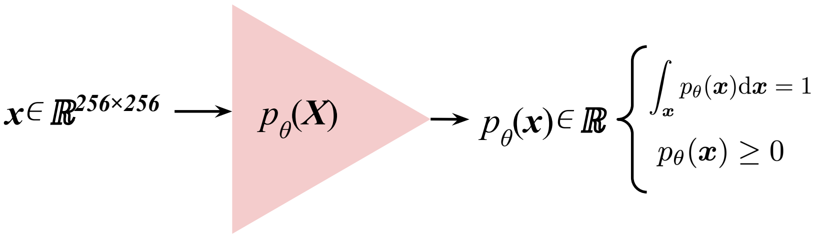


Figure: Your new budget is your parametric PDF

Parametric Probability Density Function (PDF)

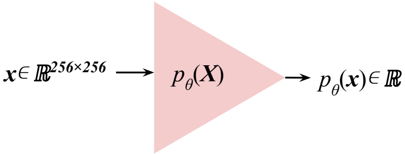


Figure: Your new budget is your parametric PDF

Parametric Probability Density Function (PDF)

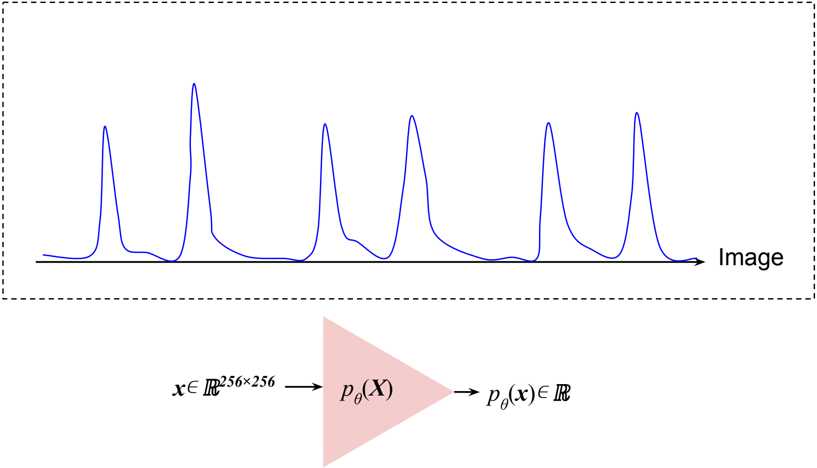


Figure: Your new budget is your parametric PDF

Parametric Probability Density Function (PDF)

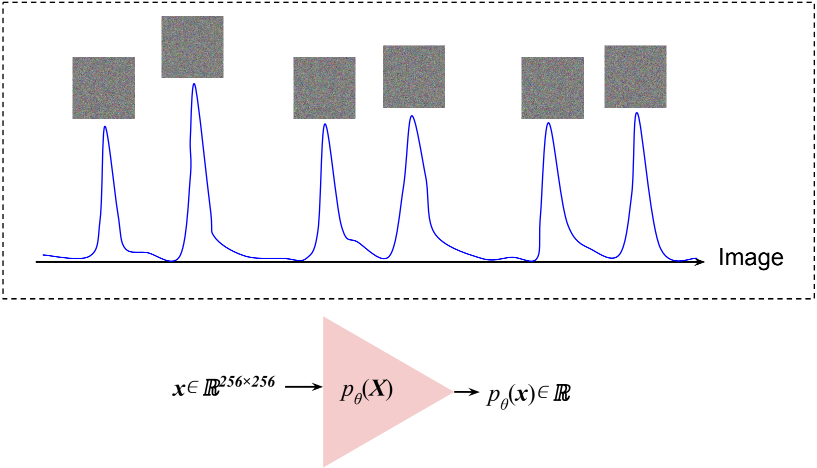
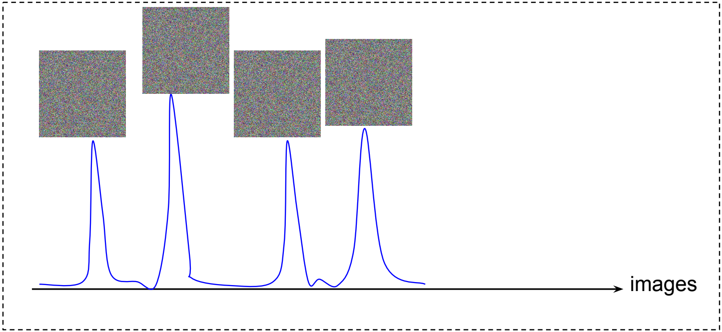


Figure: Your new budget is your parametric PDF

Learning Rooms!



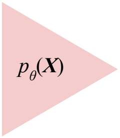


Figure: Learning to represent rooms

Learning Rooms!

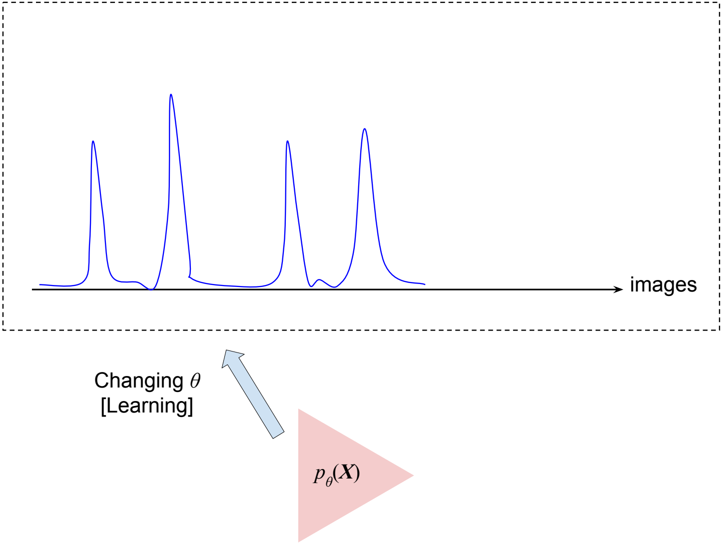


Figure: Learning to represent rooms

Learning Rooms!

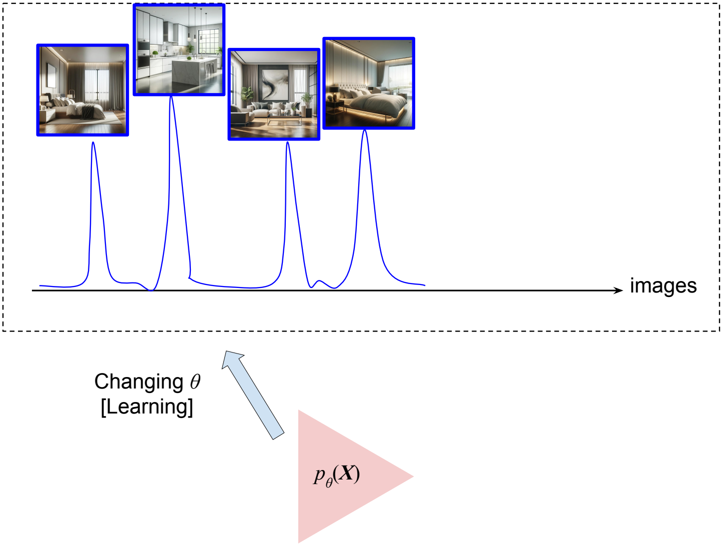
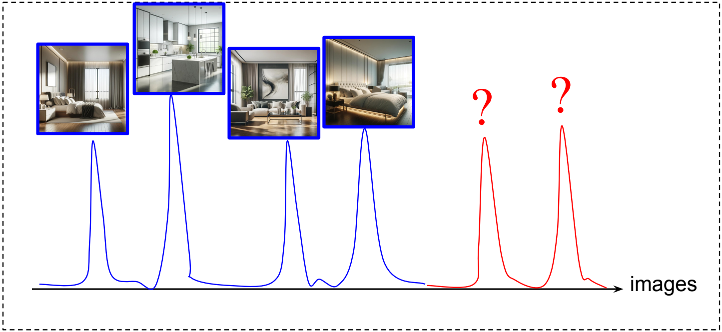


Figure: Learning to represent rooms

Learning Rooms!



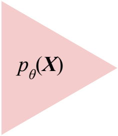


Figure: Learning to represent rooms

Learning Rooms!

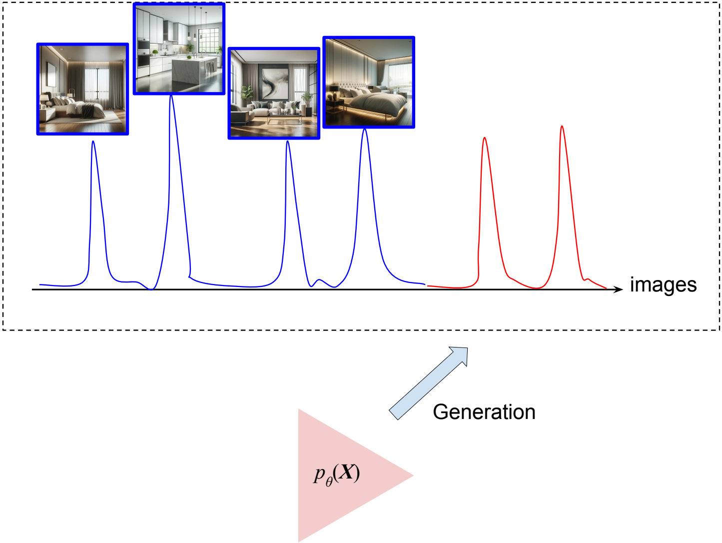


Figure: Learning to represent rooms

Learning Rooms!

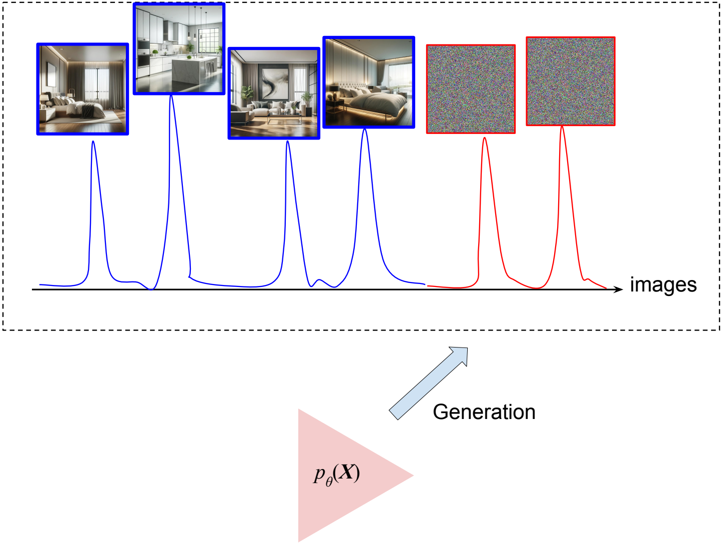


Figure: Learning to represent rooms

Learning Rooms!

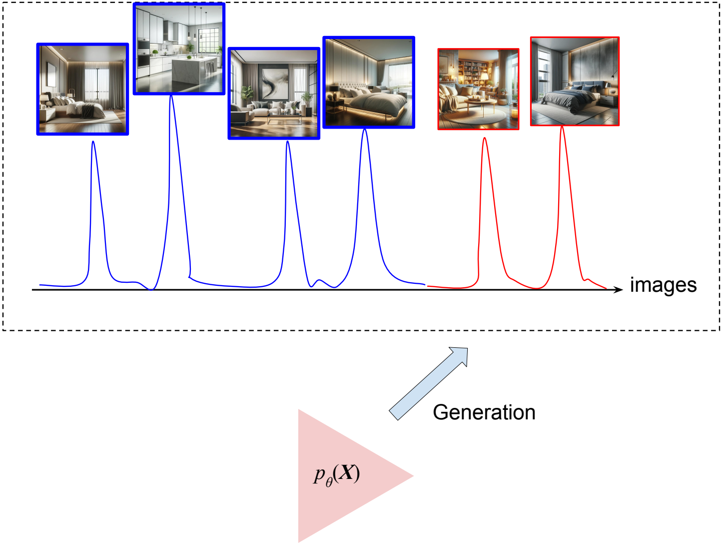


Figure: Learning to represent bedrooms

Section 3

Approaches

Subsection 1

Autoregressive Modeling

Autoregressive Modeling

"You can generate data if you can predict its future given its past!"

Language Modeling Using Autoregressive Models

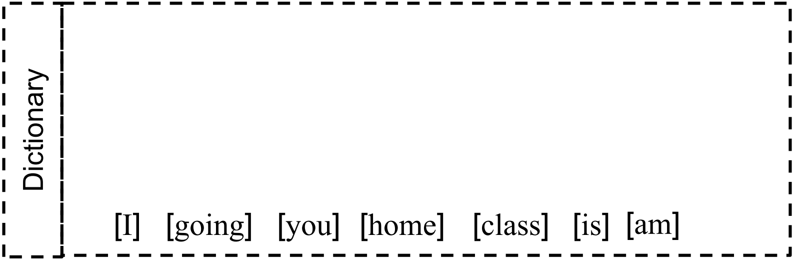


Figure: Generating the remaining part of a sentence

Language Modeling Using Autoregressive Models

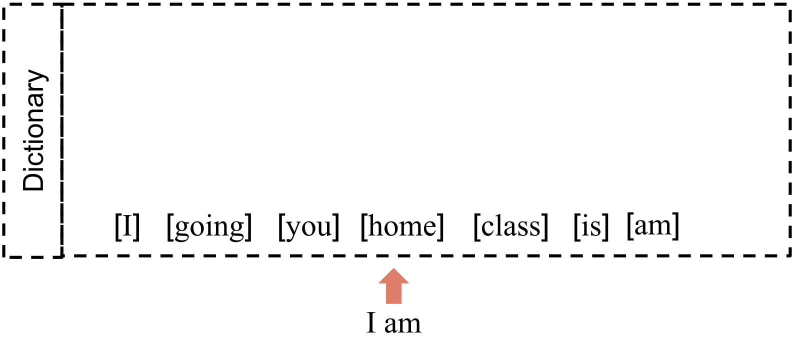


Figure: Generating the remaining part of a sentence

Language Modeling Using Autoregressive Models

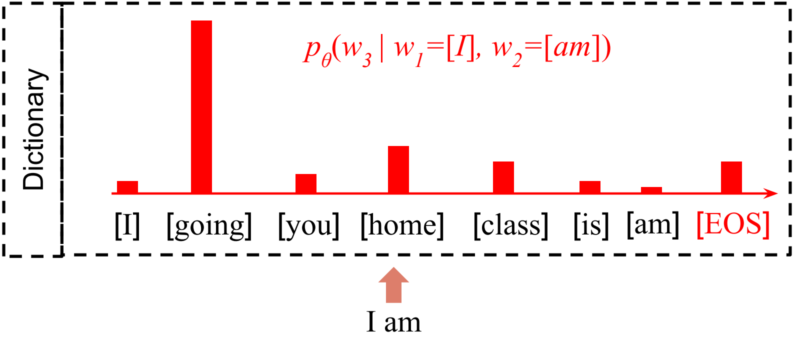


Figure: Generating the remaining part of a sentence

Language Modeling Using Autoregressive Models

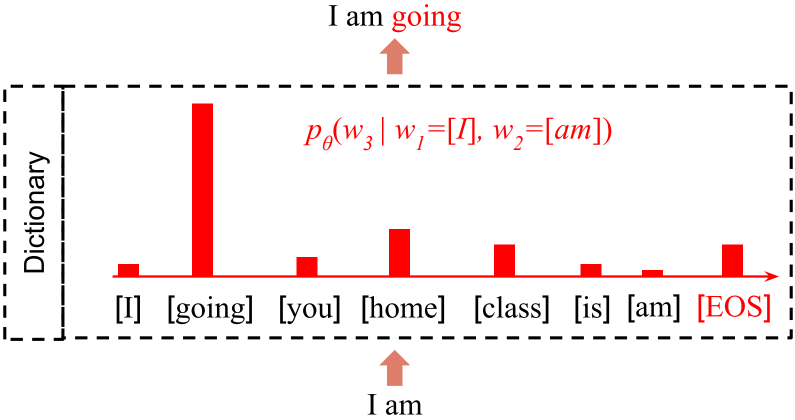


Figure: Generating the remaining part of a sentence

Language Modeling Using Autoregressive Models

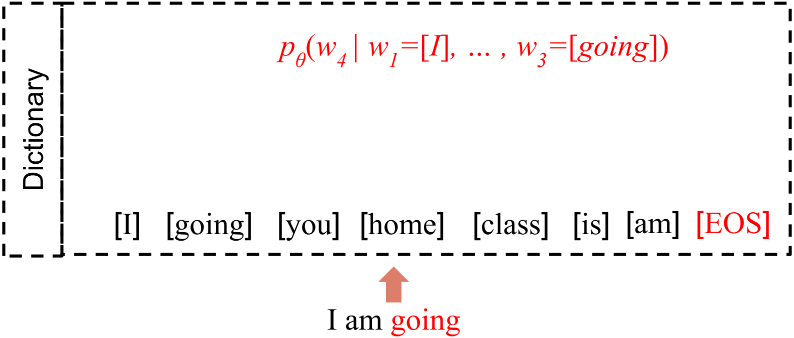


Figure: Generating the remaining part of a sentence

Language Modeling Using Autoregressive Models

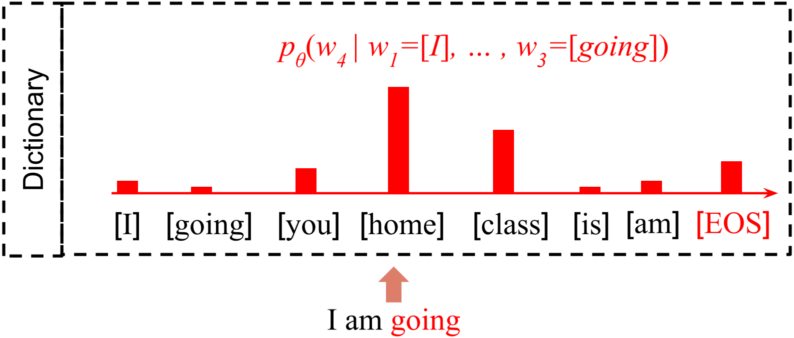


Figure: Generating the remaining part of a sentence

Language Modeling Using Autoregressive Models

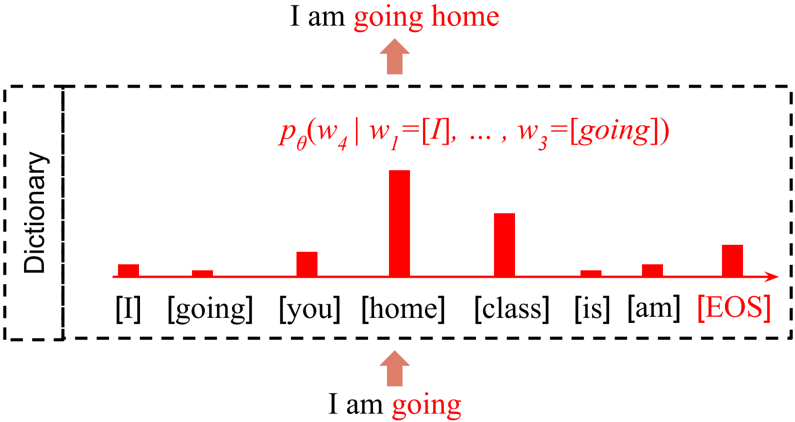


Figure: Generating the remaining part of a sentence

Language Modeling Using Autoregressive Models

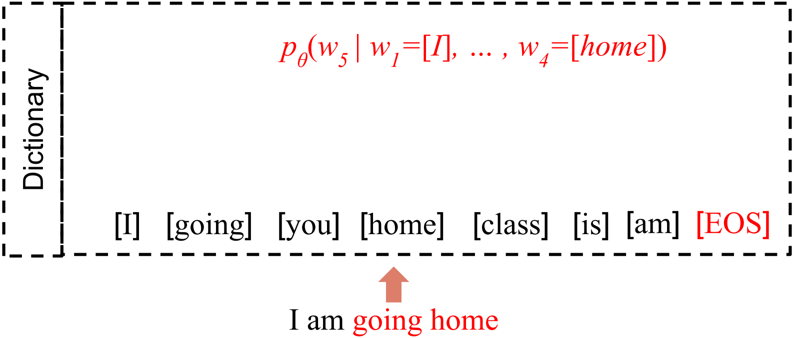


Figure: Generating the remaining part of a sentence

Language Modeling Using Autoregressive Models

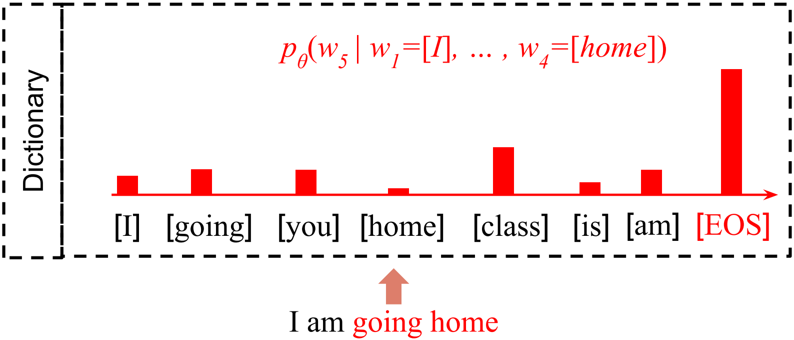


Figure: Generating the remaining part of a sentence

Language Modeling Using Autoregressive Models

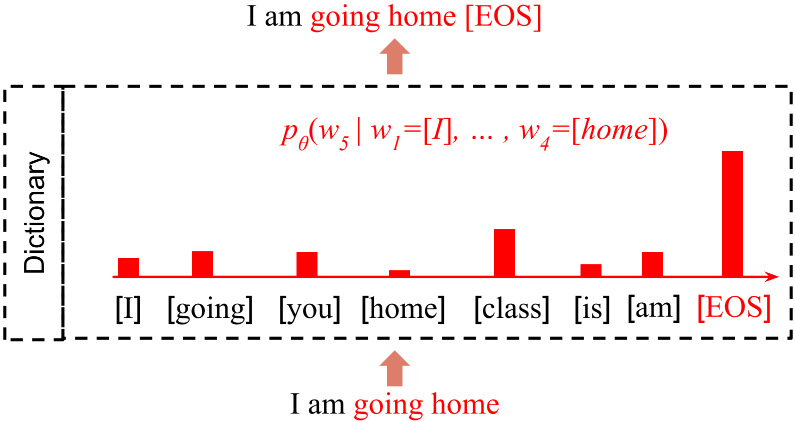


Figure: Generating the remaining part of a sentence

Scaling to ChatGPT

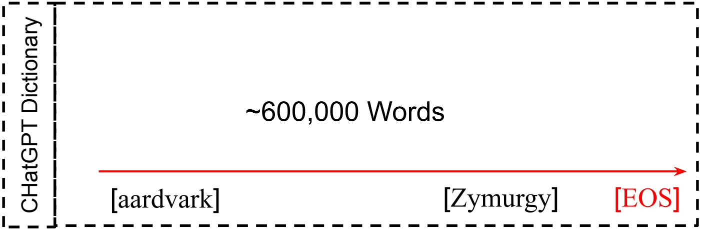


Figure: ChatGPT built on top of an Autoregressive model

Scaling to ChatGPT

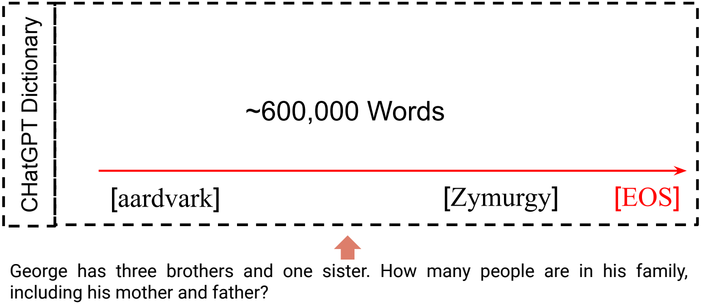
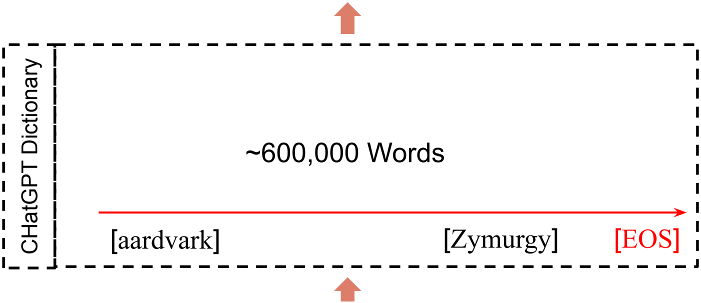


Figure: ChatGPT built on top of an Autoregressive model

Scaling to ChatGPT

George has three brothers and one sister. How many people are in his family, including his mother and father? George has three brothers and one sister, making a total of five children. Including his mother and father, there are seven people in George's family.



George has three brothers and one sister. How many people are in his family, including his mother and father?

Figure: ChatGPT built on top of an Autoregressive model

Subsection 2

Variational Autoencoder

Variational Autoencoder

"You can generate data if you can compress it efficiently!"

Variational Autoencoders

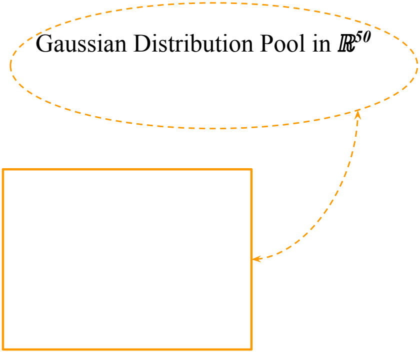


Figure: Compression learning as a method of generative modeling

Variational Autoencoders

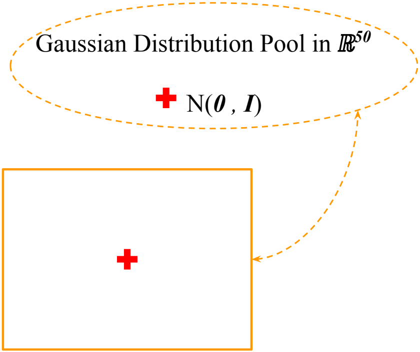


Figure: Compression learning as a method of generative modeling

$\boldsymbol{x} \in \mathbb{R}^{256 \times 256}$

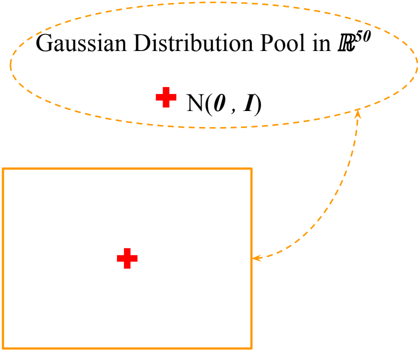


Figure: Compression learning as a method of generative modeling

Variational Autoencoders

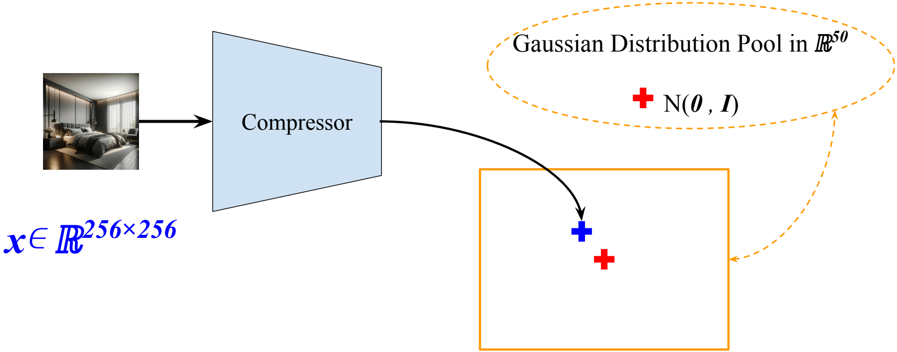


Figure: Compression learning as a method of generative modeling

Variational Autoencoders

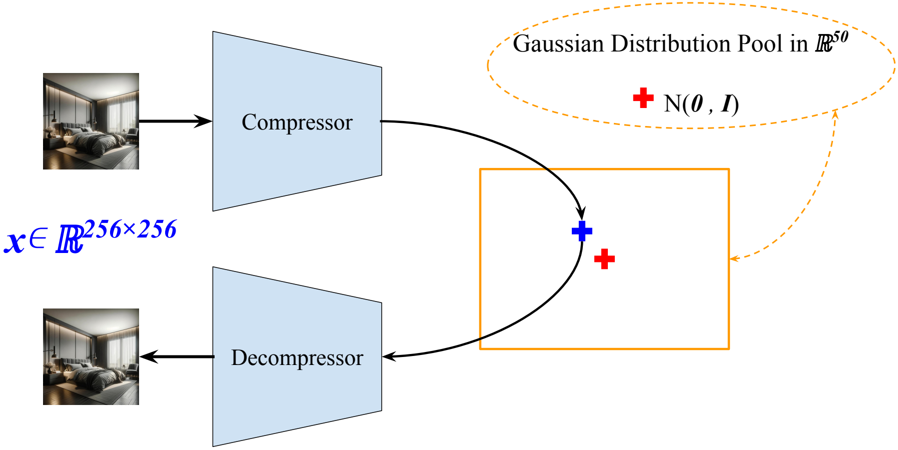


Figure: Compression learning as a method of generative modeling

Variational Autoencoders

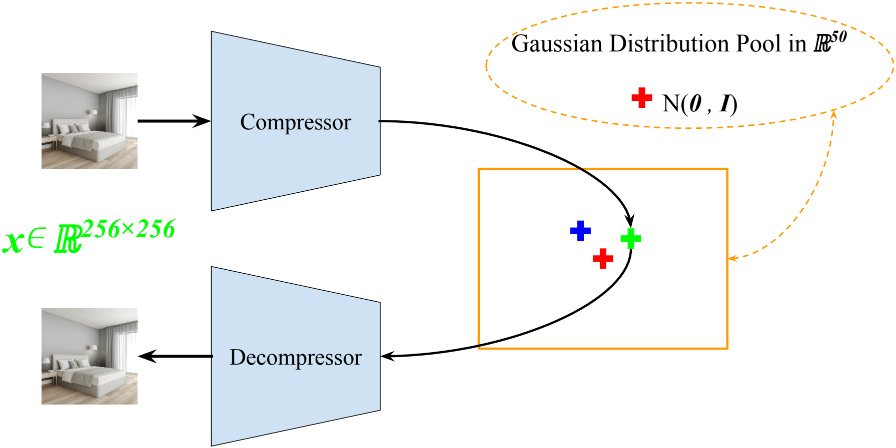


Figure: Compression learning as a method of generative modeling

Variational Autoencoders

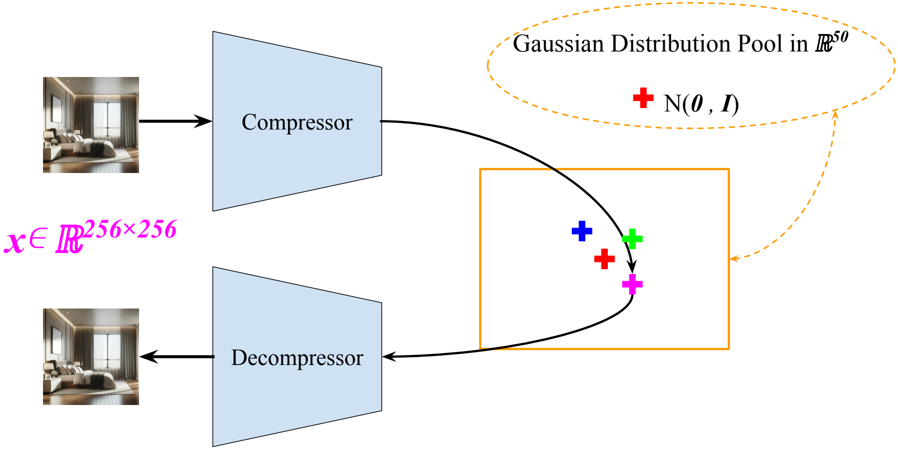


Figure: Compression learning as a method of generative modeling

Variational Autoencoders

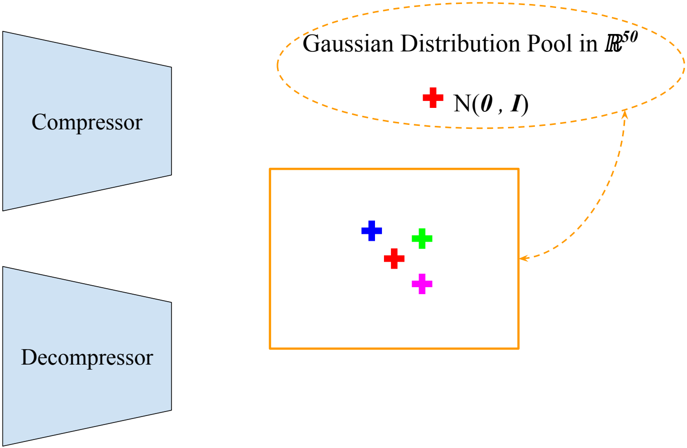


Figure: Compression learning as a method of generative modeling

Variational Autoencoders

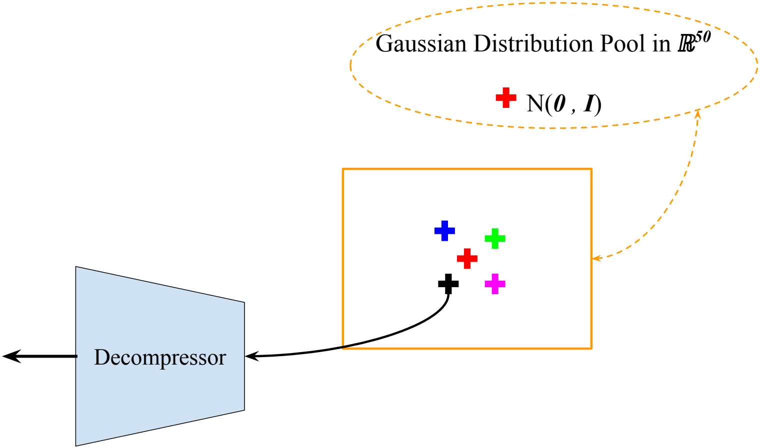


Figure: Compression learning as a method of generative modeling

Variational Autoencoders

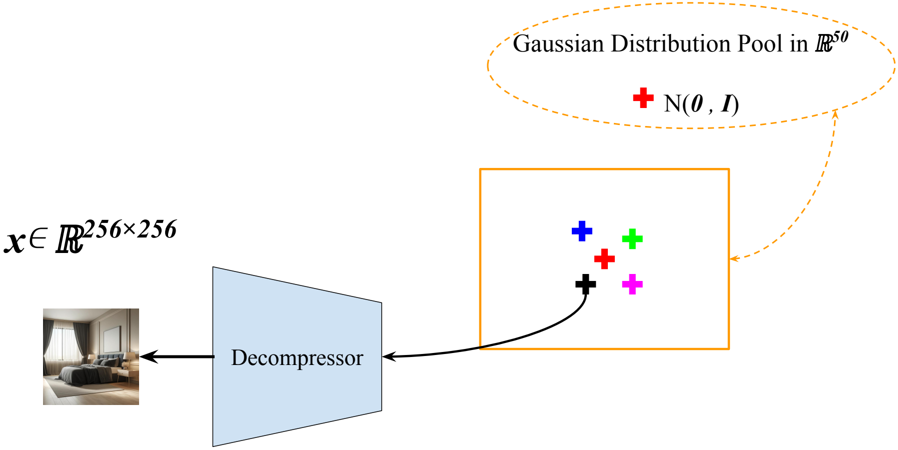


Figure: Compression learning as a method of generative modeling

Subsection 3

Generative Adversarial Nets

Generative Adversarial Nets

"Good generated samples are those that are indistinguishable from the real ones!"

Generative Adversarial Nets

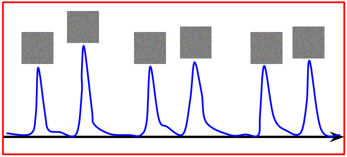


Figure: Using an Inspector [Discriminator] to detect generation

Generative Adversarial Nets

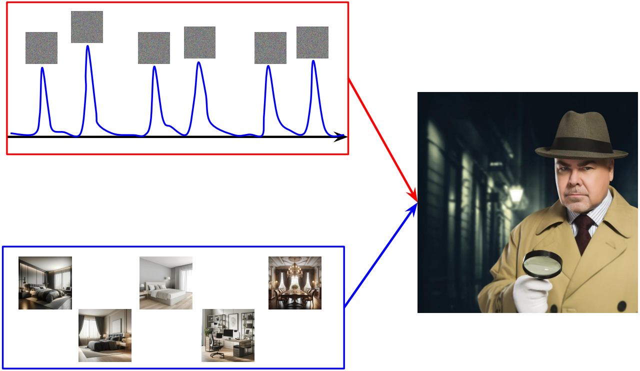


Figure: Using an Inspector [Discriminator] to detect generation

Generative Adversarial Nets

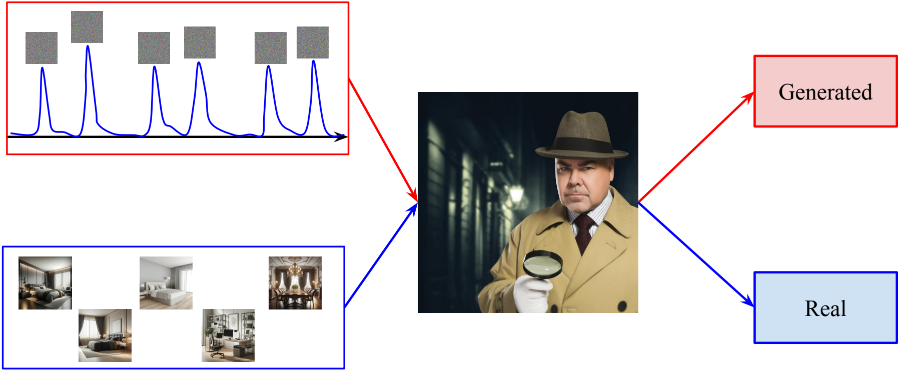


Figure: Using an Inspector [Discriminator] to detect generation

Generative Adversarial Nets

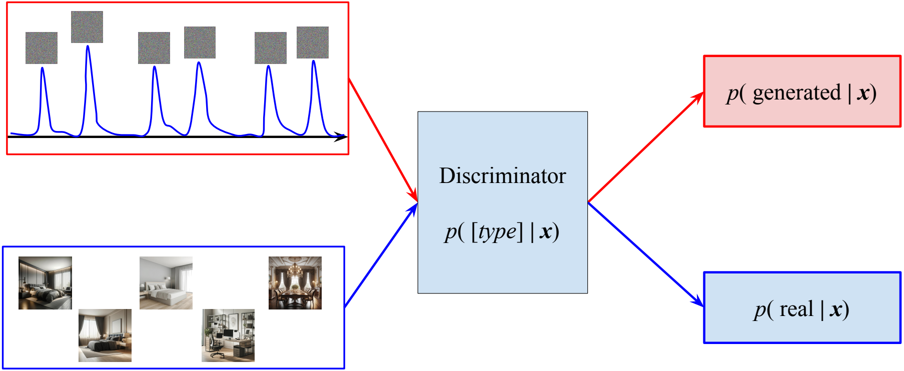


Figure: Using an Inspector [Discriminator] to detect generation

Generative Adversarial Nets

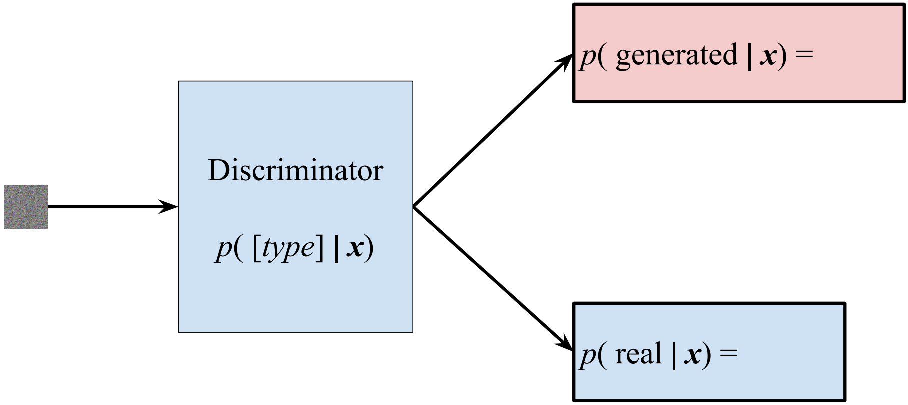


Figure: Examining the Discriminator

Generative Adversarial Nets

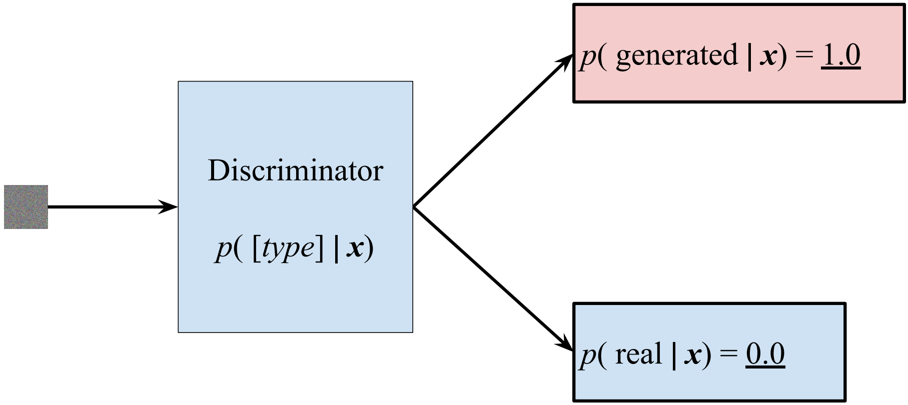


Figure: Examining the Discriminator

Generative Adversarial Nets

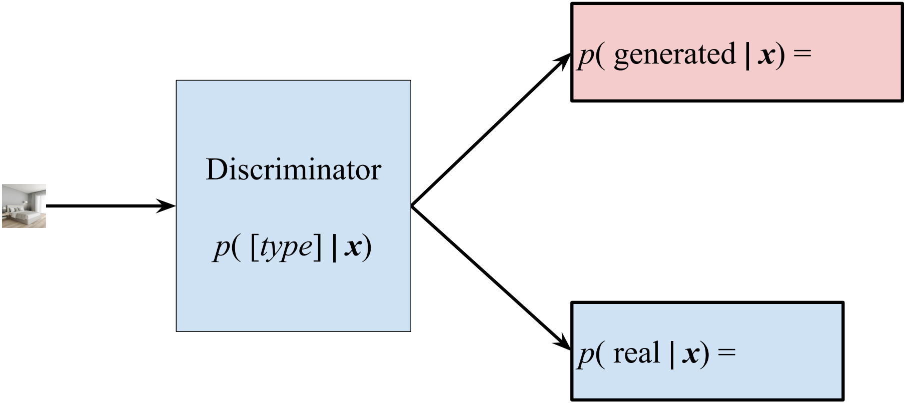


Figure: Examining the Discriminator

Generative Adversarial Nets

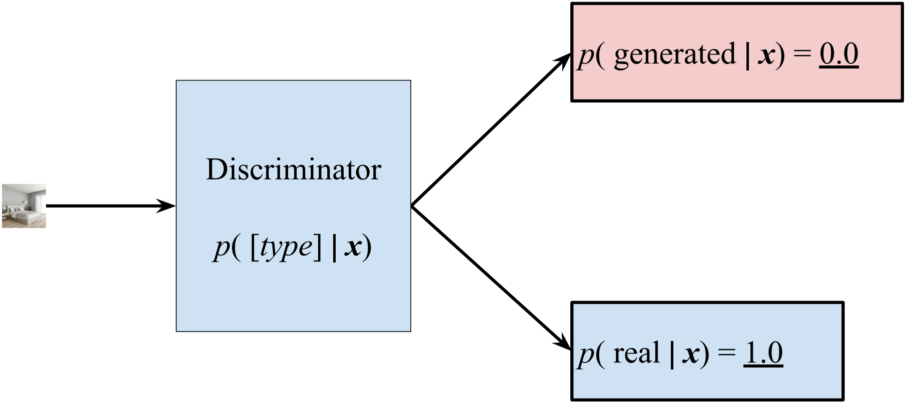


Figure: Examining the Discriminator

Generative Adversarial Nets

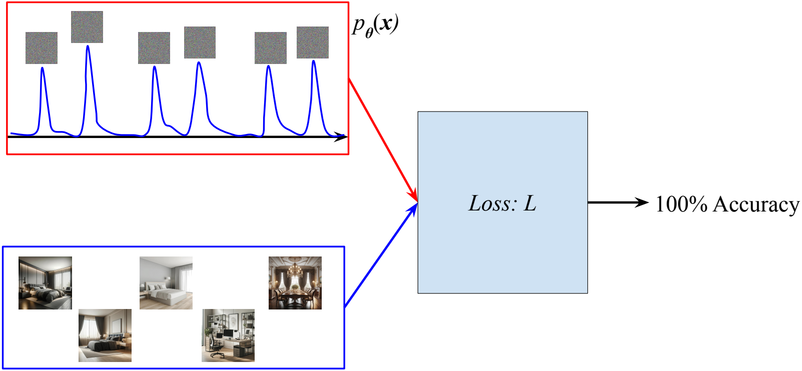


Figure: Updating generation

Generative Adversarial Nets

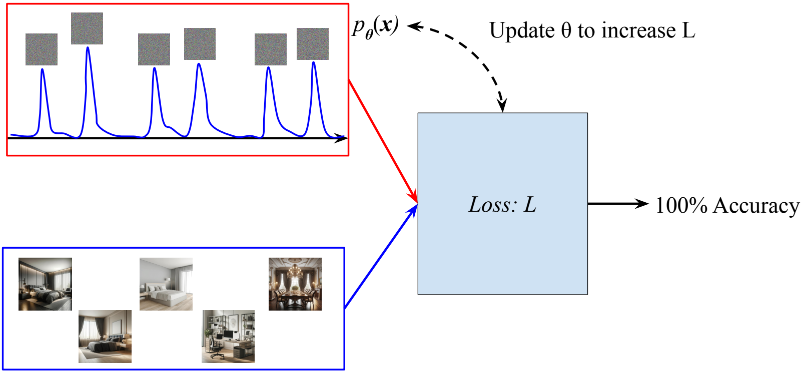


Figure: Updating generation

Generative Adversarial Nets

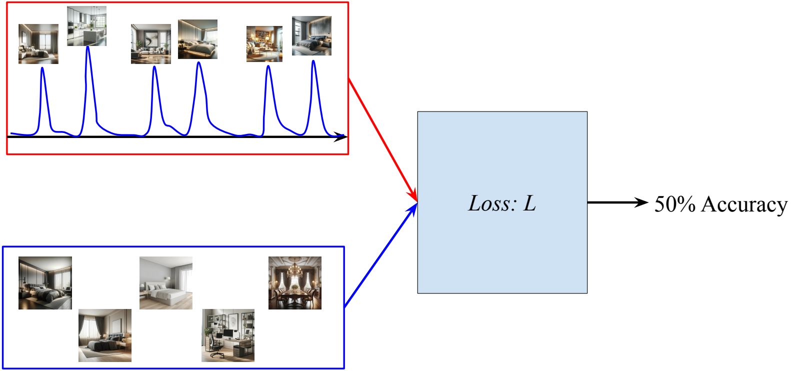


Figure: Updating generation

Subsection 4

Diffusion Models

Diffusion Models

"You can generate data if you can denoise it"

σ

Figure: Denoiser module

Diffusion Models Denoiser

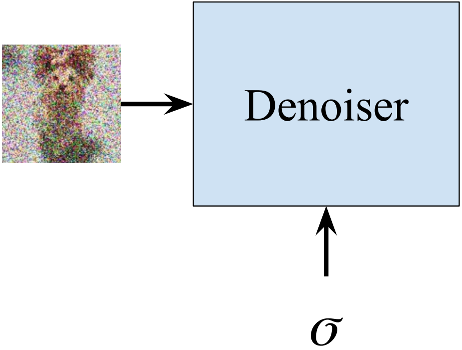


Figure: Denoiser module

Diffusion Models Denoiser

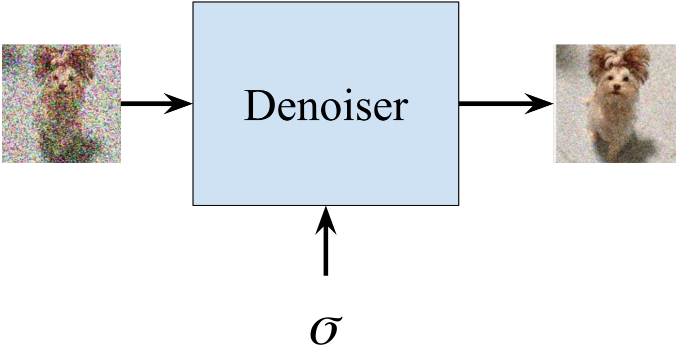


Figure: Denoiser module

Diffusion Models Generation

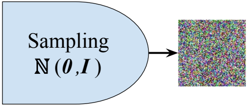


Figure: Generation using diffusion modeld (images source: [1])

Diffusion Models Generation

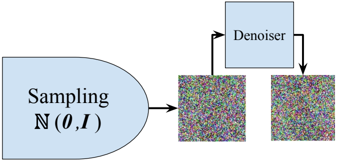


Figure: Generation using diffusion modeld (images source: [1])

Diffusion Models Generation

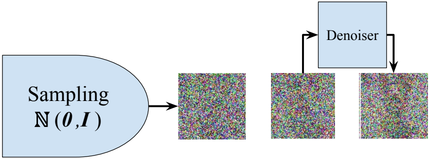


Figure: Generation using diffusion modeld (images source: [1])

Diffusion Models Generation

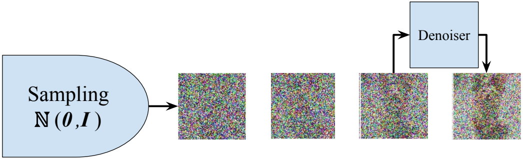


Figure: Generation using diffusion modeld (images source: [1])

Diffusion Models Generation

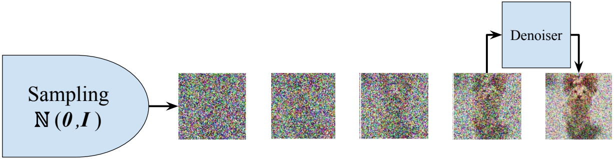


Figure: Generation using diffusion modeld (images source: [1])

Diffusion Models Generation

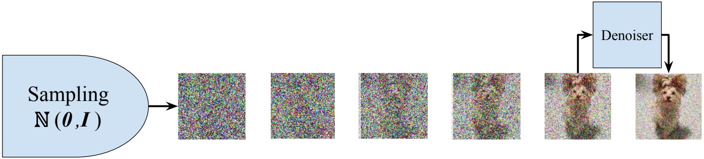


Figure: Generation using diffusion modeld (images source: [1])

Diffusion Models Generation

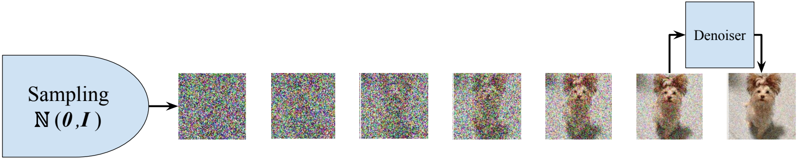


Figure: Generation using diffusion modeld (images source: [1])

Diffusion Models Generation

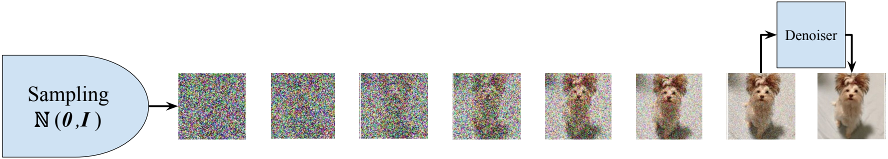


Figure: Generation using diffusion modeld (images source: [1])

Section 4

Extention to Conditional Generation

Learning Conditional Distributions

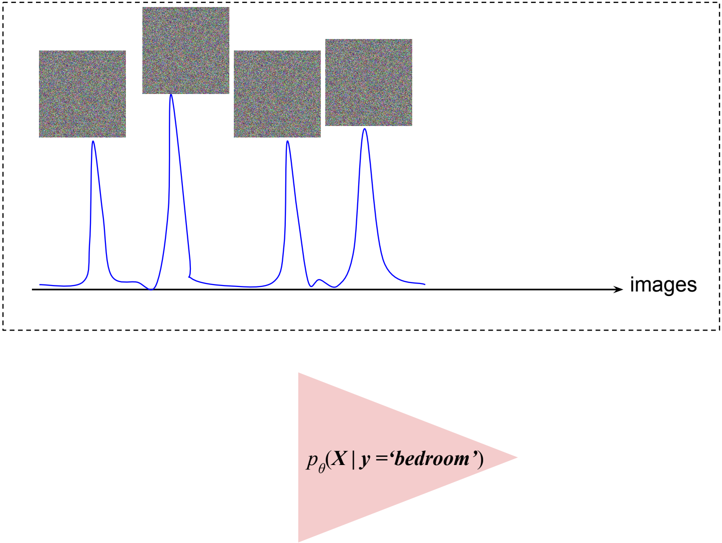


Figure: Learning to represent bedrooms

Learning Conditional Distributions

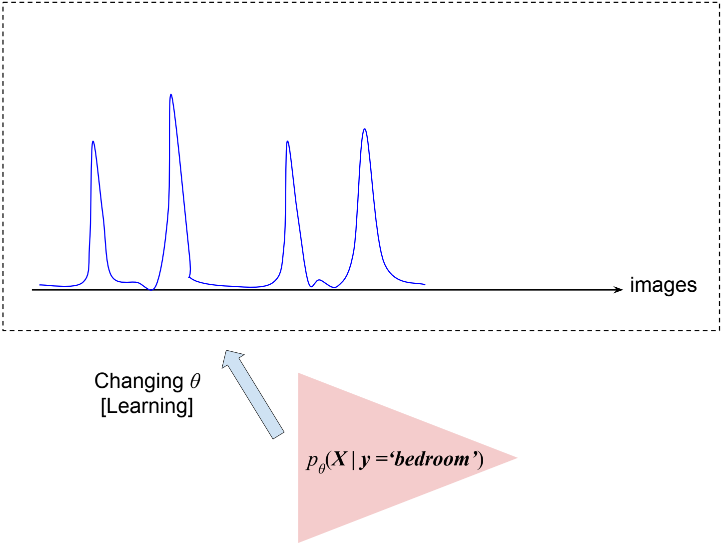


Figure: Learning to represent bedrooms

Learning Conditional Distributions

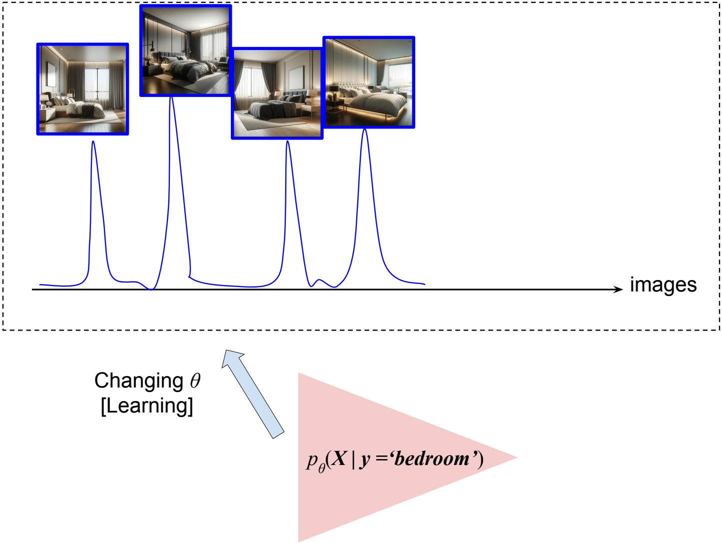
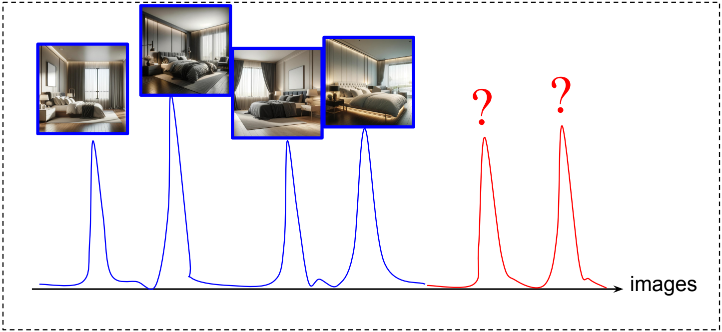


Figure: Learning to represent bedrooms

Learning Conditional Distributions



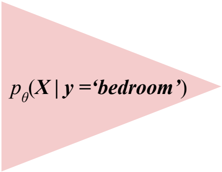


Figure: Learning to represent bedrooms

Learning Conditional Distributions

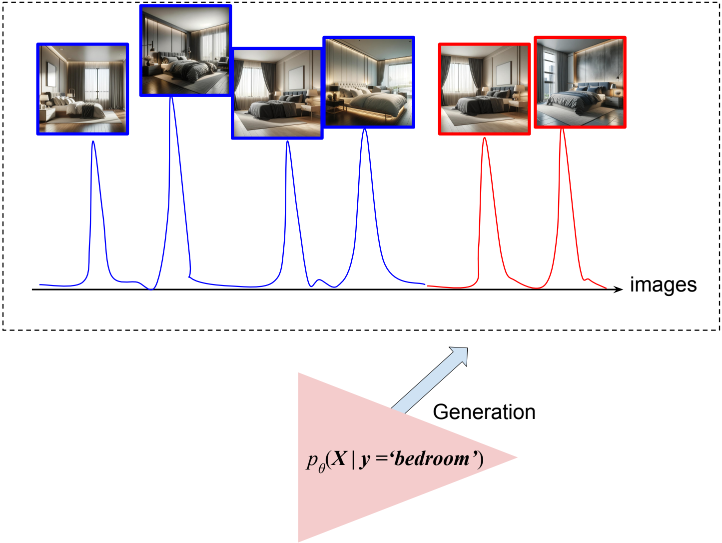


Figure: Learning to represent bedrooms

Learning Conditional Distributions

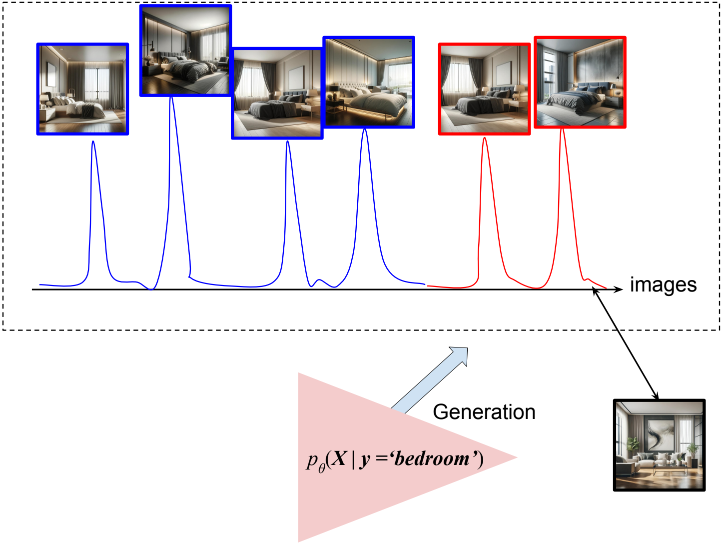


Figure: Learning to represent bedrooms

Section 5

Applications

Text-to-Speech Models

Text-to-Speech Models

$$p(\boldsymbol{x}|\boldsymbol{y}) : \begin{cases} \boldsymbol{x} : \text{An audio file} \\ \boldsymbol{y} : \text{A text} \end{cases}$$

Text-to-Speech Models

Text-to-Speech Models

$$p(\boldsymbol{x}|\boldsymbol{y}) : \begin{cases} \boldsymbol{x} : \text{An audio file} \\ \boldsymbol{y} : \text{A text} \end{cases}$$

Real-World Sample

Listen to the following speech synthesis (source: [2])

$$\boldsymbol{y} = \begin{array}{c}\text{"A single Wavenet can} \\ \text{capture the characteristics of many} \\ \text{different speakers with equal fidelity,} \\ \text{not it's fast."}\end{array} \xRightarrow{\text{Sampling} p(\boldsymbol{x}|\boldsymbol{y})} \boldsymbol{x} = \boxed{\text{Play}}$$

Text-to-Image Models

Text-to-Image Models

$$p(\boldsymbol{x}|\boldsymbol{y}) : \begin{cases} \boldsymbol{x} : \text{An image} \\ \boldsymbol{y} : \text{A text} \end{cases}$$

Text-to-Image Models

Text-to-Image Models

$$p(\boldsymbol{x}|\boldsymbol{y}) : \begin{cases} \boldsymbol{x} : \text{An image} \\ \boldsymbol{y} : \text{A text} \end{cases}$$

Figure: $\boldsymbol{x}$ for $\boldsymbol{y}$ = "Teddy bears swimming at the Olympics 400m Butterfly event." (source: [?])

Image-to-Image Translation

Image Colorization

$$p(\boldsymbol{x}|\boldsymbol{y}) : \begin{cases} \boldsymbol{x} : \text{A Colored image} \\ \boldsymbol{y} : \text{A Gray} - \text{scale image} \end{cases}$$

Image-to-Image Translation

Image Colorization

$$p(\boldsymbol{x}|\boldsymbol{y}) : \begin{cases} \boldsymbol{x} : \text{A Colored image} \\ \boldsymbol{y} : \text{A Gray} - \text{scale image} \end{cases}$$

(a) $\boldsymbol{y}$

(b) $\boldsymbol{x}$

(c) Ground truth

Figure: Image colorization (source: [3])

Image-to-Image Translation

Image Inpainting

$$p(\boldsymbol{x}|\boldsymbol{y}) : \begin{cases} \boldsymbol{x} : \text{A clean image} \\ \boldsymbol{y} : \text{A corrupted image} \end{cases}$$

Image-to-Image Translation

Image Inpainting

$$p(\boldsymbol{x}|\boldsymbol{y}) : \begin{cases} \boldsymbol{x} : \text{A clean image} \\ \boldsymbol{y} : \text{A corrupted image} \end{cases}$$

(a) $\boldsymbol{y}$ (b) $\boldsymbol{x}$ (c) Ground truth

Figure: Image inpainting (source: [3])

Image-to-Image Translation

Image Uncropping

$$p(\boldsymbol{x}|\boldsymbol{y}) : \begin{cases} \boldsymbol{x} : \text{A clean image} \\ \boldsymbol{y} : \text{A cropped image} \end{cases}$$

Image-to-Image Translation

Image Uncropping

$$p(\boldsymbol{x}|\boldsymbol{y}) : \begin{cases} \boldsymbol{x} : \text{A clean image} \\ \boldsymbol{y} : \text{A cropped image} \end{cases}$$

(a) $\boldsymbol{y}$ (b) $\boldsymbol{x}$ (c) Ground truth

Figure: Image uncropping (source: [3])

Image-to-Image Translation

Image Restoration

$$p(\boldsymbol{x}|\boldsymbol{y}) : \begin{cases} \boldsymbol{x} : \text{A clean image} \\ \boldsymbol{y} : \text{A degraded image} \end{cases}$$

Image-to-Image Translation

Image Restoration

$$p(\boldsymbol{x}|\boldsymbol{y}) : \begin{cases} \boldsymbol{x} : \text{A clean image} \\ \boldsymbol{y} : \text{A degraded image} \end{cases}$$

(a) $\boldsymbol{y}$

(b) $\boldsymbol{x}$

(c) Ground truth

Figure: Image restoration (source: [3])

Section 6

Deep Autoregressive Models

Logistic Regression Model

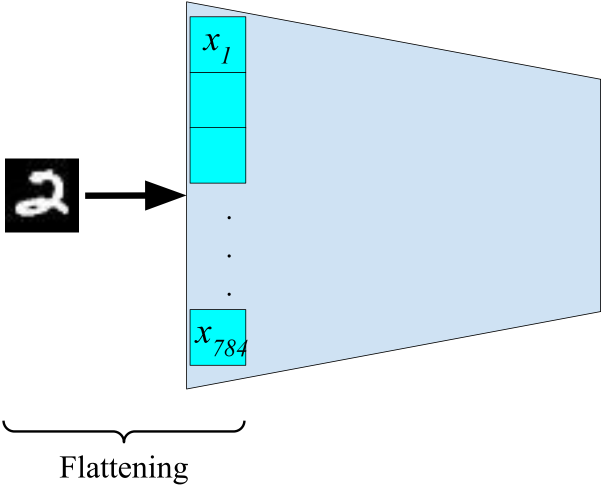


Figure: Logistic regression steps

Logistic Regression Model

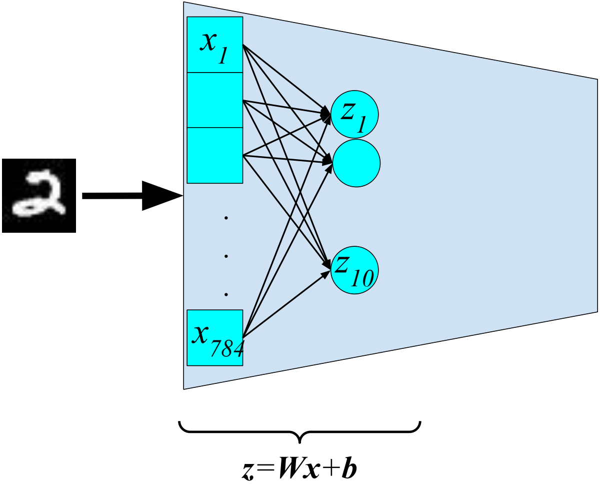


Figure: Logistic regression steps

Logistic Regression Model

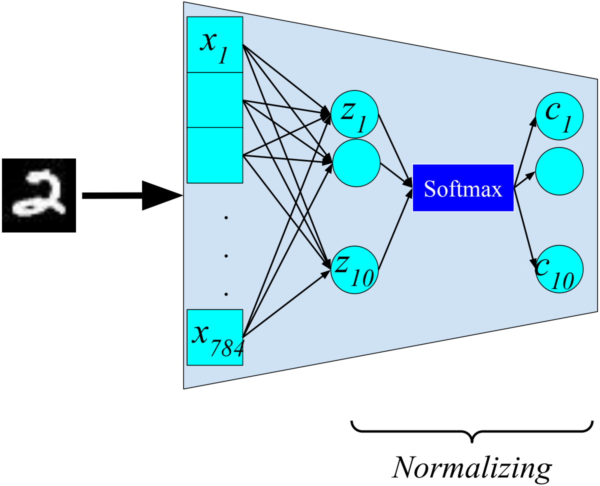


Figure: Logistic regression steps

Logistic Regression Model

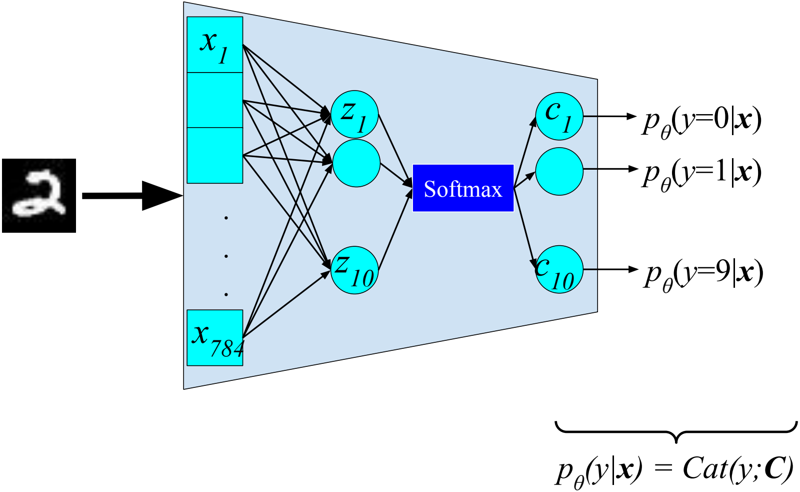


Figure: Logistic regression steps

Logistic Regression Model

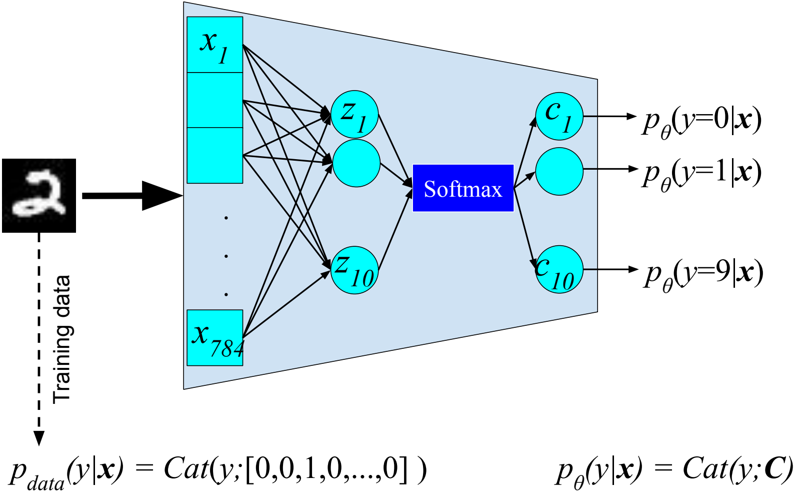


Figure: Logistic regression steps

Logistic Regression Model

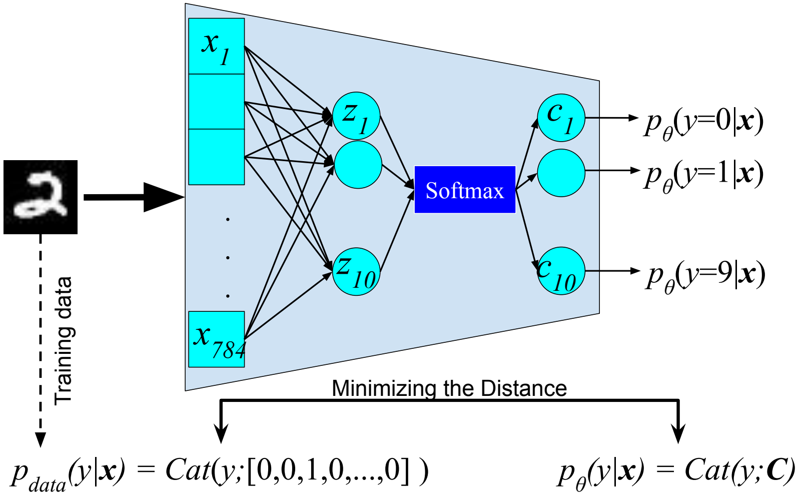


Figure: Logistic regression steps

Learning

Distance Metric

One option for distance metric is:

Distance Metric

One option for distance metric is:

$$L(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\mathbb{X})} \left[\text{KL} \left(p_{\text{data}}\big(y|\boldsymbol{x}\big) \parallel p_{\theta}\big(y|\boldsymbol{x}\big) \right) \right]$$

Learning

Distance Metric

One option for distance metric is:

$$L(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\mathbb{X})} \left[\text{KL} \left(p_{\text{data}}(y|\boldsymbol{x}) \parallel p_\theta(y|\boldsymbol{x}) \right) \right]$$

$$= \sum_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \left[\sum_y p_{\text{data}}(y|\boldsymbol{x}) \log \frac{p_{\text{data}}(y|\boldsymbol{x})}{p_\theta(y|\boldsymbol{x})} \right]$$

Learning

Distance Metric

One option for distance metric is:

$$
\begin{aligned}
L(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\mathbb{X})}\left[\text{KL}\left(p_{\text{data}}(y|\boldsymbol{x}) \parallel p_{\theta}(y|\boldsymbol{x})\right)\right] \\
&= \sum_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x})\left[\sum_{y} p_{\text{data}}(y|\boldsymbol{x}) \log \frac{p_{\text{data}}(y|\boldsymbol{x})}{p_{\theta}(y|\boldsymbol{x})}\right] \\
&= \underbrace{\sum_{y}\sum_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}, y) \log p_{\text{data}}(y|\boldsymbol{x})}_{\mathbb{E}_{(\boldsymbol{x},y)\sim p_{\text{data}}(\mathbb{X},Y)}[\log p_{\text{data}}(y|\boldsymbol{x})]}
\end{aligned}
$$

Learning

Distance Metric

One option for distance metric is:

$$\begin{aligned} L(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\mathbb{X})}\left[\text{KL}\left(p_{\text{data}}(y|\boldsymbol{x}) \parallel p_\theta(y|\boldsymbol{x})\right)\right] \\ &= \sum_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \left[\sum_y p_{\text{data}}(y|\boldsymbol{x}) \log \frac{p_{\text{data}}(y|\boldsymbol{x})}{p_\theta(y|\boldsymbol{x})}\right] \\ &= \underbrace{\sum_y \sum_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}, y) \log p_{\text{data}}(y|\boldsymbol{x})}_{\mathbb{E}_{(\boldsymbol{x},y)\sim p_{\text{data}}(\mathbb{X},Y)}[\log p_{\text{data}}(y|\boldsymbol{x})]} - \underbrace{\sum_y \sum_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}, y) \log p_\theta(\boldsymbol{x}|y)}_{\mathbb{E}_{(\boldsymbol{x},y)\sim p_{\text{data}}(\mathbb{X},Y)}[\log p_\theta(y|\boldsymbol{x})]} \end{aligned}$$

Learning

Distance Metric

One option for distance metric is:

$$L(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\mathbb{X})} \left[\text{KL} \left(p_{\text{data}}(y|\boldsymbol{x}) \parallel p_{\theta}(y|\boldsymbol{x}) \right) \right]$$

$$= \sum_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \left[\sum_{y} p_{\text{data}}(y|\boldsymbol{x}) \log \frac{p_{\text{data}}(y|\boldsymbol{x})}{p_{\theta}(y|\boldsymbol{x})} \right]$$

$$= \underbrace{\sum_{y} \sum_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}, y) \log p_{\text{data}}(y|\boldsymbol{x})}_{\mathbb{E}_{(\boldsymbol{x},y) \sim p_{\text{data}}(\mathbb{X},Y)}[\log p_{\text{data}}(y|\boldsymbol{x})]} - \underbrace{\sum_{y} \sum_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}, y) \log p_{\theta}(\boldsymbol{x}|y)}_{\mathbb{E}_{(\boldsymbol{x},y) \sim p_{\text{data}}(\mathbb{X},Y)}[\log p_{\theta}(y|\boldsymbol{x})]}$$

While the second term is a function of your model parameters, the first one is independent of the selected Autoregressive model and thus can be omitted in optimization.

Training

Distance Metric

So:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, L(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} -\mathbb{E}_{(\boldsymbol{x},y)\sim p_{\text{data}}(\mathbb{X},Y)}\left[\log p_\theta(y|\boldsymbol{x})\right]$$

Monte Carlo Estimation

Consider the following expectation:

$$\mathbb{E}_{\boldsymbol{x}\sim p(\mathbb{X})}\left[f(\boldsymbol{x})\right] = \int p(\boldsymbol{x}) f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

Training

Distance Metric

So:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, L(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} -\mathbb{E}_{(\boldsymbol{x},y)\sim p_{\text{data}}(\mathbb{X},Y)}\left[\log p_\theta(y|\boldsymbol{x})\right]$$

Monte Carlo Estimation

Consider the following expectation:

$$\mathbb{E}_{\boldsymbol{x}\sim p(\mathbb{X})}\left[f(\boldsymbol{x})\right] = \int p(\boldsymbol{x}) f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

Now assume that instead of $p(\mathbb{X})$, we just have access to N independent samples of random variable $\mathbb{X}$ as $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.

Training

Distance Metric

So:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, L(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} -\mathbb{E}_{(\boldsymbol{x},y)\sim p_{\text{data}}(\mathbb{X},Y)}\left[\log p_\theta(y|\boldsymbol{x})\right]$$

Monte Carlo Estimation

Consider the following expectation:

$$\mathbb{E}_{\boldsymbol{x}\sim p(\mathbb{X})}\left[f(\boldsymbol{x})\right] = \int p(\boldsymbol{x})f(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

Now assume that instead of $p(\mathbb{X})$, we just have access to N independent samples of random variable $\mathbb{X}$ as $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.Then expectation can be approximated as:

$$\mathbb{E}_{\boldsymbol{x}\sim p(\mathbb{X})}\left[f(\boldsymbol{x})\right] \simeq \frac{1}{N}\sum_n f(\boldsymbol{x}_n)$$

Training

Optimization

Using Monte-Carlo estimation, we have the following optimization problem:

$$\begin{aligned}\boldsymbol{\theta}^{\star} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} -\mathbb{E}_{(\boldsymbol{x},y)\sim p_{\text{data}}(\mathbb{X},Y)}\left[\log p_{\theta}(y|\boldsymbol{x})\right] \\ &\simeq \underset{\boldsymbol{\theta}}{\operatorname{argmax}} -\frac{1}{N}\sum_{i=1}^{N}\log p_{\theta}(y_i|\boldsymbol{x}_i)\end{aligned}$$

Sampling

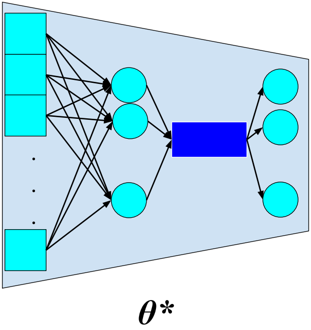


Figure: Sampling a trained model

Sampling

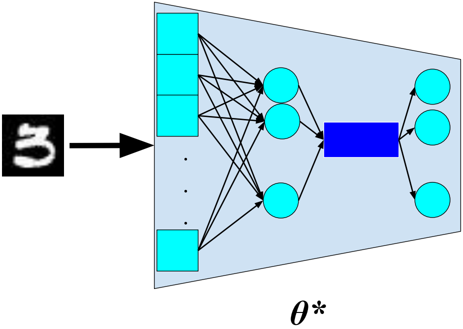


Figure: Sampling a trained model

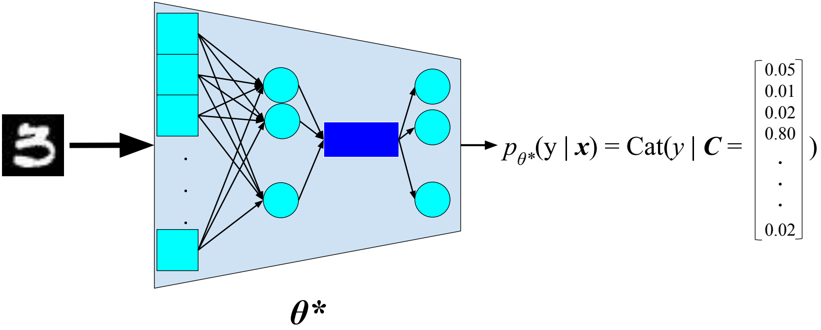


Figure: Sampling a trained model

Sampling

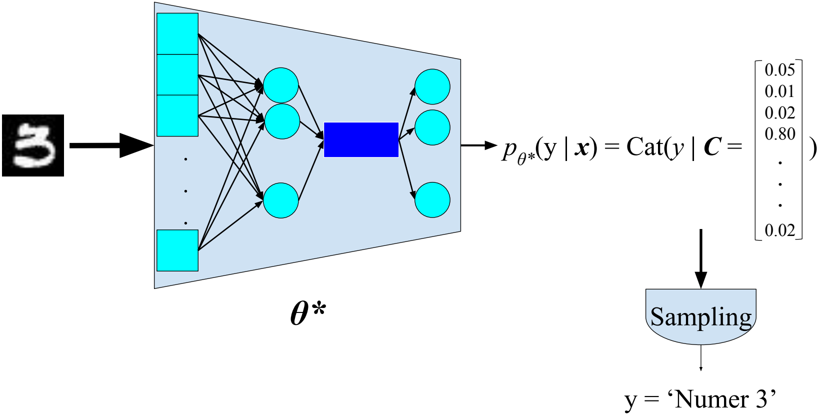


Figure: Sampling a trained model

Modeling

Generative Modeling

Assume we just have MINST image $\{\boldsymbol{x}_i\}_{i=1}^{N}$ without any label and we want to estimate generating distribution $p(\boldsymbol{x})$ where $\boldsymbol{x} \in \mathbb{R}^{784}$.

Challenge: High-dimensional Random Vector

In contrast to logistic regression where we model $p_{\text{data}}(y|\boldsymbol{x})$ and y was a one-dimensional random variable, here $\boldsymbol{x}$ is a high-dimensional random vector.

- ☞ It seems that we can't use logistic regression here.
- ☞ We can model each dimension separately because $x_i \in \{0, 1, 2, \ldots, 255\}$

Chain Rule

Based on the chain rule, we have:

$$p(\boldsymbol{x}) = p(x_1)p(x_2|\boldsymbol{x}_{<2}) \ldots p(x_d|\boldsymbol{x}_{<d}) \ldots p(x_D|\boldsymbol{x}_{<D}),\ \boldsymbol{x}_{<d} \triangleq [x_1, \ldots, x_{d-1}]^T$$

Modeling

$$p(\boldsymbol{x}) = p(x_1) \times p(x_2 \mid \boldsymbol{x}_{<2}) \times \ldots\ldots \times p(x_i \mid \boldsymbol{x}_{<i}) \times \ldots\ldots \times p(x_D \mid \boldsymbol{x}_{<D})$$

Figure: Using logistic regression for generative modeling

Modeling

$$p(\boldsymbol{x}) = p(x_1) \times p(x_2 \mid \boldsymbol{x}_{<2}) \times \ldots\ldots \times \boxed{p(x_i \mid \boldsymbol{x}_{<i})} \times \ldots\ldots \times p(x_D \mid \boldsymbol{x}_{<D})$$

Figure: Using logistic regression for generative modeling

Modeling

$$p(\boldsymbol{x}) = p(x_1) \times p(x_2 \mid \boldsymbol{x}_{<2}) \times \ldots\ldots \times \boxed{p(x_i \mid \boldsymbol{x}_{<i})} \times \ldots\ldots \times p(x_D \mid \boldsymbol{x}_{<D})$$

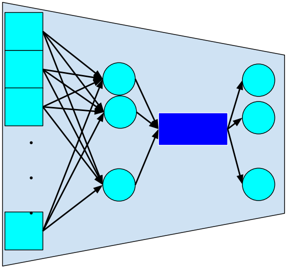
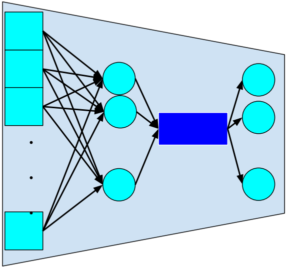


Figure: Using logistic regression for generative modeling

$$p(\boldsymbol{x}) = p(x_1) \times p(x_2 \mid \boldsymbol{x}_{<2}) \times \ldots\ldots \times p(x_i \mid \boldsymbol{x}_{<i}) \times \ldots\ldots \times p(x_D \mid \boldsymbol{x}_{<D})$$



$\boldsymbol{W}_i, \boldsymbol{b}_i$

Figure: Using logistic regression for generative modeling

Modeling

$$p(\boldsymbol{x}) = p(x_1) \times p(x_2 \mid \boldsymbol{x}_{<2}) \times \ldots\ldots \times p(x_i \mid \boldsymbol{x}_{<i}) \times \ldots\ldots \times p(x_D \mid \boldsymbol{x}_{<D})$$

Figure: Using logistic regression for generative modeling

Modeling

$$p(\boldsymbol{x}) = p(x_1) \times p(x_2 \mid \boldsymbol{x}_{<2}) \times \ldots\ldots \times p(x_i \mid \boldsymbol{x}_{<i}) \times \ldots\ldots \times p(x_D \mid \boldsymbol{x}_{<D})$$

$\boldsymbol{b}_1$

Figure: Using logistic regression for generative modeling

$$p(\boldsymbol{x}) = p(x_1) \times p(x_2 \mid \boldsymbol{x}_{<2}) \times \ldots\ldots \times p(x_i \mid \boldsymbol{x}_{<i}) \times \ldots\ldots \times p(x_D \mid \boldsymbol{x}_{<D})$$

$\boldsymbol{b}_1$

$\boldsymbol{b}_2, \boldsymbol{W}_2$

Figure: Using logistic regression for generative modeling

Modeling

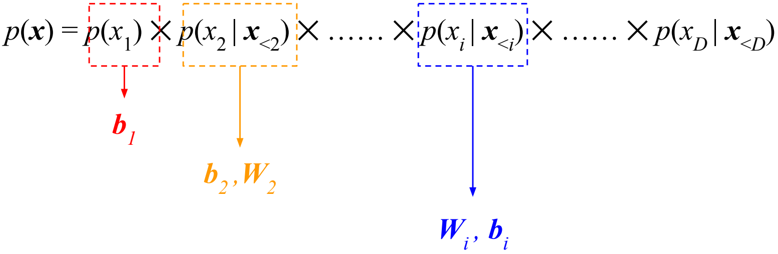


Figure: Using logistic regression for generative modeling

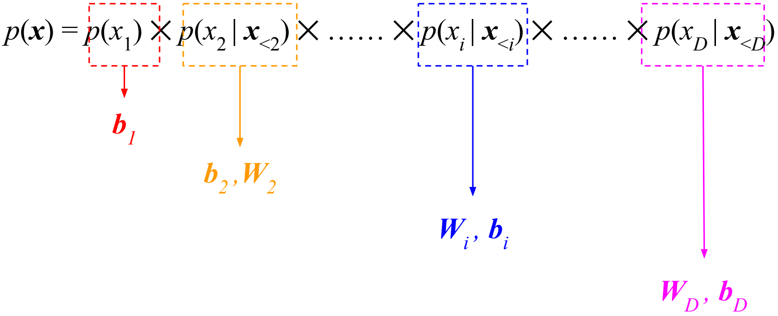


Figure: Using logistic regression for generative modeling

Modeling

$$p(\boldsymbol{x}) = p(x_1) \times p(x_2 \mid \boldsymbol{x}_{<2}) \times \ldots\ldots \times p(x_i \mid \boldsymbol{x}_{<i}) \times \ldots\ldots \times p(x_D \mid \boldsymbol{x}_{<D})$$

$p(x_1) \rightarrow \boldsymbol{b}_1$

$p(x_2 \mid \boldsymbol{x}_{<2}) \rightarrow \boldsymbol{b}_2, \boldsymbol{W}_2$

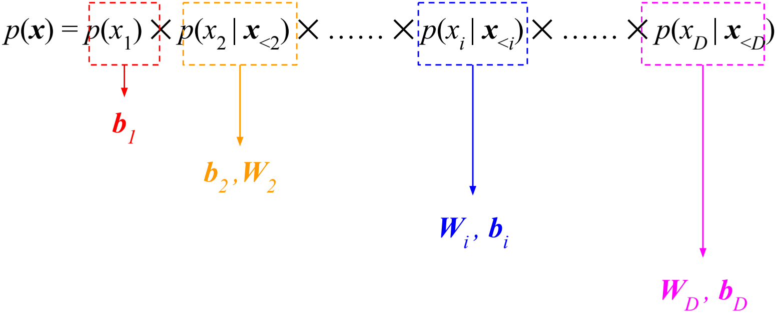
$p(x_i \mid \boldsymbol{x}_{<i}) \rightarrow \boldsymbol{W}_i, \boldsymbol{b}_i$

$p(x_D \mid \boldsymbol{x}_{<D}) \rightarrow \boldsymbol{W}_D, \boldsymbol{b}_D$

$$x_i \in \{0,1,\ldots,255\} \Longrightarrow \begin{cases} \boldsymbol{b}_i \in R^{256} \\ \boldsymbol{W}_i \in R^{256\times i} \end{cases} \forall \quad 1 \leq i \leq \mathrm{D}$$

Figure: Using logistic regression for generative modeling

Modeling



$$\boldsymbol{\theta} = \{ \boldsymbol{b}_1, \boldsymbol{W}_2, \boldsymbol{b}_2, \ldots, \boldsymbol{W}_i, \boldsymbol{b}_i, \ldots, \boldsymbol{W}_D, \boldsymbol{b}_D \}$$

Figure: Using logistic regression for generative modeling

Distance Metric

Distance Metric

We want to comapre two distribution p_{data} and p_θ, thus we can use KL divergence as:

$$L(\boldsymbol{\theta}) = \text{KL}(p_{\text{data}} \| p_\theta) =$$

Distance Metric

Distance Metric

We want to comapre two distribution p_{data} and p_θ, thus we can use KL divergence as:

$$L(\boldsymbol{\theta}) = \text{KL}(p_{\text{data}} \| p_\theta) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\mathbb{X})} \left[\log \left(\frac{p_{\text{data}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})} \right) \right]$$

Distance Metric

Distance Metric

We want to comapre two distribution p_{data} and p_θ, thus we can use KL divergence as:

$$L(\boldsymbol{\theta}) = \text{KL}(p_{\text{data}} \| p_\theta) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\mathbb{X})} \left[\log \left(\frac{p_{\text{data}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})} \right) \right]$$

$$= \sum_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log \frac{p_{\text{data}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}$$

Distance Metric

We want to comapre two distribution p_{data} and p_θ, thus we can use KL divergence as:

$$L(\boldsymbol{\theta}) = \text{KL}(p_{\text{data}} \| p_\theta) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\mathbb{X})} \left[\log \left(\frac{p_{\text{data}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})} \right) \right]$$
$$= \sum_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log \frac{p_{\text{data}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}$$

Using above definition, we know $L(\boldsymbol{\theta}) = 0$ iff $p_\theta(\mathbb{X}) = p_{\text{data}}(\mathbb{X})$. We can rewrite $L(\boldsymbol{\theta})$ as:

$$L(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \left[\log p_{\text{data}}(\boldsymbol{x}) \right] - \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \left[\log p_\theta(\boldsymbol{x}) \right]$$

Distance Metric

Distance Metric

We want to comapre two distribution p_{data} and p_θ, thus we can use KL divergence as:

$$L(\boldsymbol{\theta}) = \text{KL}(p_{\text{data}} \| p_\theta) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\mathbb{X})} \left[\log \left(\frac{p_{\text{data}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})} \right) \right]$$

$$= \sum_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log \frac{p_{\text{data}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}$$

Using above definition, we know $L(\boldsymbol{\theta}) = 0$ iff $p_\theta(\mathbb{X}) = p_{\text{data}}(\mathbb{X})$. We can rewrite $L(\boldsymbol{\theta})$ as:

$$L(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} [\log p_{\text{data}}(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} [\log p_\theta(\boldsymbol{x})]$$

Because the first term on the right-hand side is independent of $\boldsymbol{\theta}$, we have:

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\text{argmin}}\, \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \left[\log \left(\frac{p_{\text{data}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})} \right) \right] \equiv \underset{\boldsymbol{\theta}}{\text{argmax}}\, \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} [\log p_\theta(\boldsymbol{x})]$$

From KL divergence to Model Likelihood

Model Likelihood

We see:

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \left[\log p_\theta(\boldsymbol{x})\right]$$

Thus:

From KL divergence to Model Likelihood

Model Likelihood

We see:

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \, \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \left[\log p_\theta(\boldsymbol{x}) \right]$$

Thus:

- Desirable situation is when $p_\theta(\mathbb{X})$ assign high probability to probable regions in $p_{\text{data}}(\mathbb{X})$

From KL divergence to Model Likelihood

Model Likelihood

We see:

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \left[\log p_\theta(\boldsymbol{x})\right]$$

Thus:

- Desirable situation is when $p_\theta(\mathbb{X})$ assign high probability to probable regions in $p_{\text{data}}(\mathbb{X})$
- We have yet a problem: No access to p_{data}

From KL divergence to Model Likelihood

Model Likelihood

We see:

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \left[\log p_\theta(\boldsymbol{x})\right]$$

Thus:

- Desirable situation is when $p_\theta(\mathbb{X})$ assign high probability to probable regions in $p_{\text{data}}(\mathbb{X})$
- We have yet a problem: No access to p_{data}
- $\mathbb{H}(p_{\text{data}}(\mathbb{X})) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\mathbb{X})} \left[\log p_{\text{data}}(\boldsymbol{x})\right]$ is the maximum accessible objective value where $\mathbb{H}(p_{\text{data}}(\mathbb{X}))$ is the *entropy* defined as:

$$\mathbb{H}(p_{\text{data}}(\mathbb{X})) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \left[\log p_{\text{data}}(\boldsymbol{x})\right]$$

Model Likelihood Estimation

Model Likelihood Estimation

We are interested in solving the following problem:

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\mathbb{X})}\left[\log p_\theta(\boldsymbol{x})\right]$$

but we don't have access to p_{data} and instead, we have access to independent samples from the distribution $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^N$.

Model Likelihood Estimation

Model Likelihood Estimation

We are interested in solving the following problem:

$$\boldsymbol{\theta}^{\star} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\mathbb{X})} \left[\log p_{\theta}(\boldsymbol{x}) \right]$$

but we don't have access to p_{data} and instead, we have access to independent samples from the distribution $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N}$.

Solution via Monte Carlo Estimate

Using the Monte Carlo estimate we have:

$$\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\mathbb{X})} \left[\log p_{\theta}(\boldsymbol{x}) \right] \simeq \frac{1}{N} \sum_{n=1}^{N} \log p_{\theta}(\boldsymbol{x}_n)$$

Thus:

$$\boldsymbol{\theta}^{\star} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \frac{1}{N} \sum_{n=1}^{N} \log p_{\theta}(\boldsymbol{x}_n)$$

Thank You!

Thank you for your attention!

Do you have any questions or comments?

Contact Information

Sajjad Amini
Email: samini@umass.edu

References I

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole,
"Score-based generative modeling through stochastic differential equations,"
arXiv preprint arXiv:2011.13456, 2020.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu,
"Wavenet: A generative model for raw audio,"
arXiv preprint arXiv:1609.03499, 2016.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi,
"Palette: Image-to-image diffusion models,"
in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–10.