Section 6

From Supervised Learning to Generative Modeling

Subsection 1

Logistic Regression

Logistic Regression Model

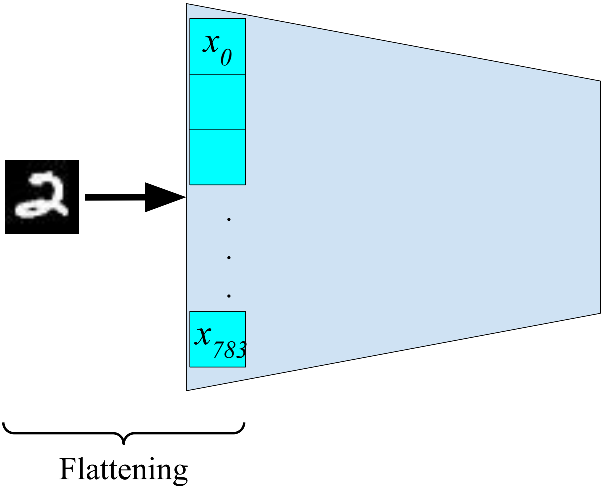


Figure: Logistic regression steps

Logistic Regression Model

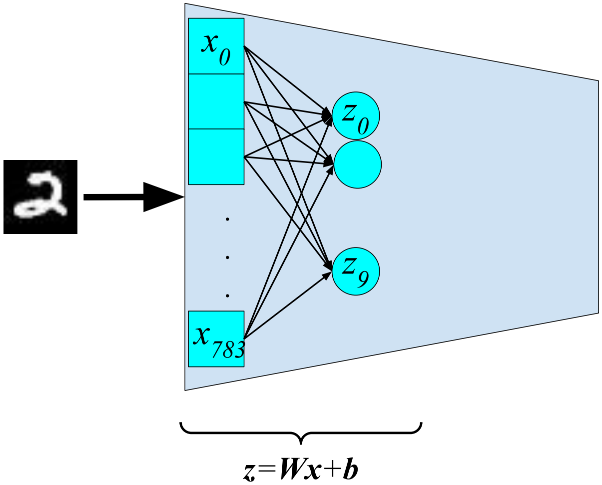


Figure: Logistic regression steps

Logistic Regression Model

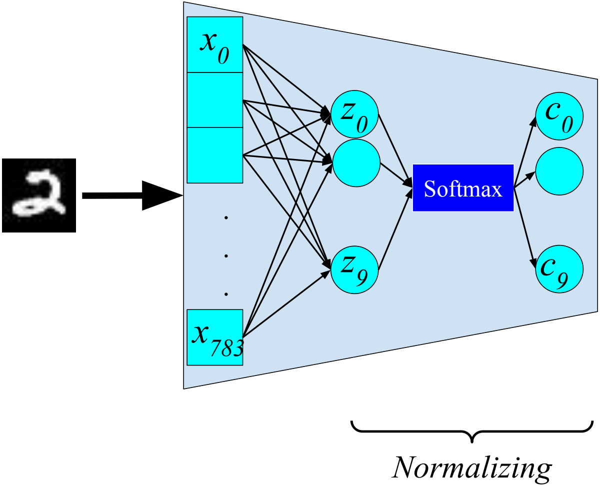


Figure: Logistic regression steps

Logistic Regression Model

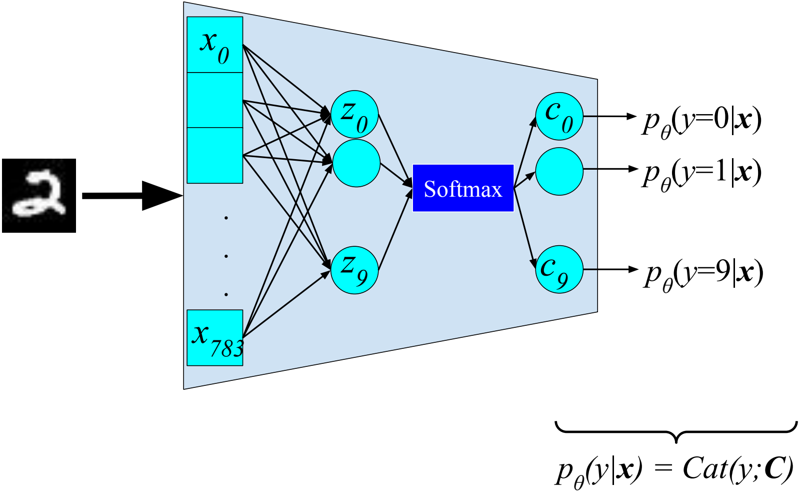


Figure: Logistic regression steps

Logistic Regression Model

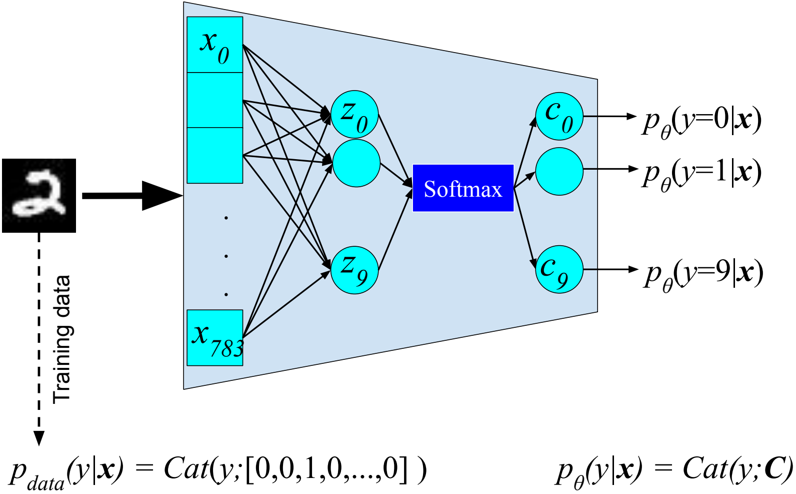


Figure: Logistic regression steps

Logistic Regression Model

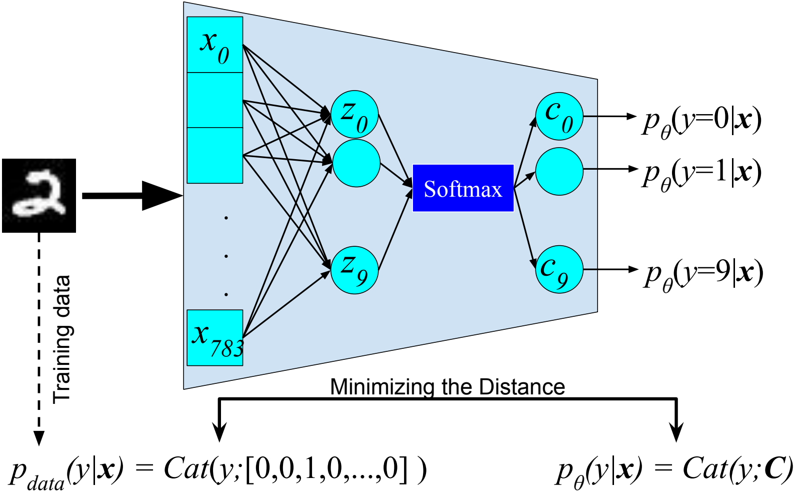


Figure: Logistic regression steps

Distance Metric

One option for distance metric is:

Learning

Distance Metric

One option for distance metric is:

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

Learning

Distance Metric

One option for distance metric is:

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

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

Distance Metric

One option for distance metric is:

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

Distance Metric

One option for distance metric is:

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

Learning

Distance Metric

One option for distance metric is:

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

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:

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

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:

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

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

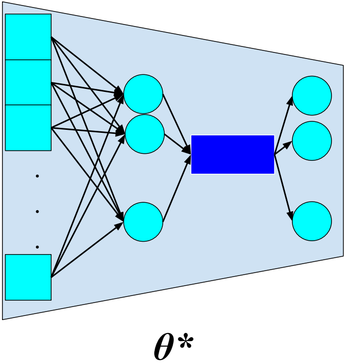


Figure: Sampling a trained model

Sampling

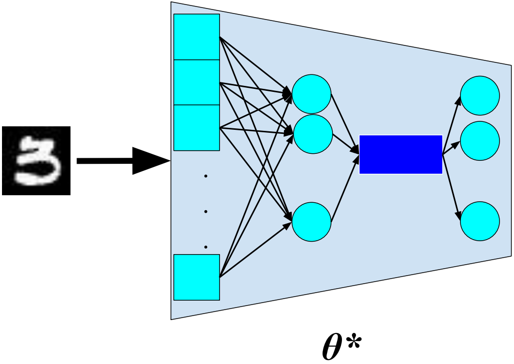


Figure: Sampling a trained model

Sampling

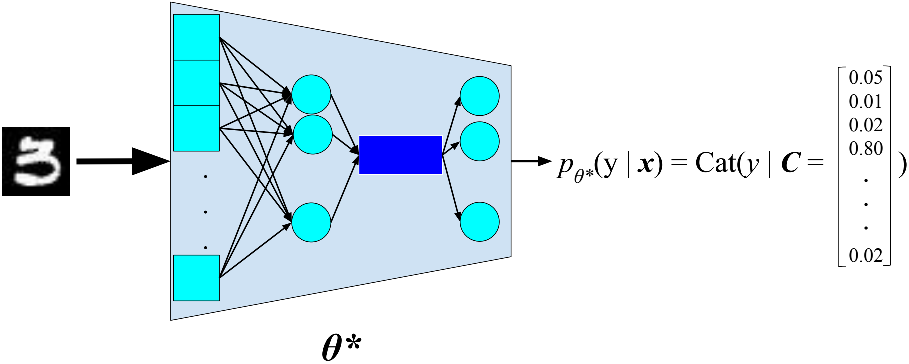


Figure: Sampling a trained model

Sampling

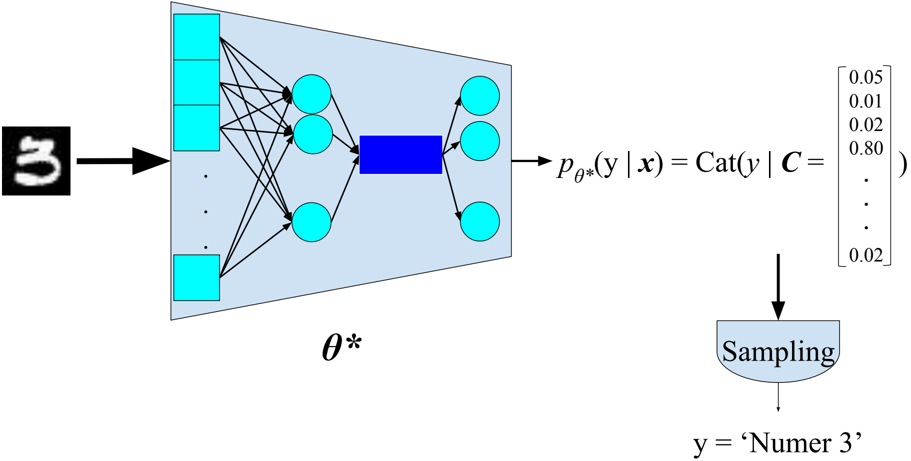


Figure: Sampling a trained model

Sampling a Categorical Distribution

$$\mathrm{Cat}\left(y \mid \boldsymbol{C} = \begin{bmatrix} c_0 = 0.1 \\ c_1 = 0.7 \\ c_2 = 0.2 \end{bmatrix}\right)$$

Figure: Sampling a categorical distribution using a Uniform sampler

Sampling a Categorical Distribution

$$\text{Cat}(y \mid \boldsymbol{C} = \begin{bmatrix} c_0 = 0.1 \\ c_1 = 0.7 \\ c_2 = 0.2 \end{bmatrix})$$

PMF

$$\boldsymbol{C} = \begin{bmatrix} c_0 = 0.1 \\ c_1 = 0.7 \\ c_2 = 0.2 \end{bmatrix}$$

Figure: Sampling a categorical distribution using a Uniform sampler

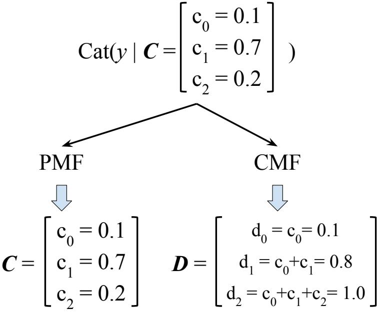


Figure: Sampling a categorical distribution using a Uniform sampler

Sampling a Categorical Distribution

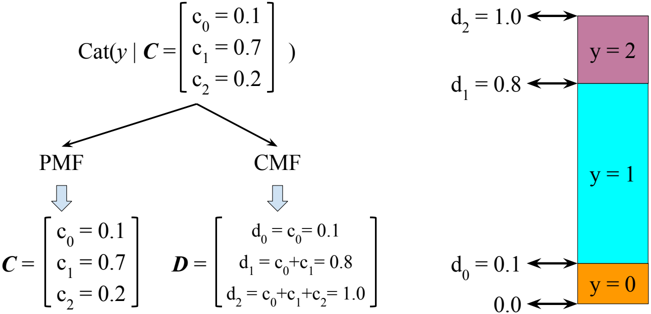


Figure: Sampling a categorical distribution using a Uniform sampler

Sampling a Categorical Distribution

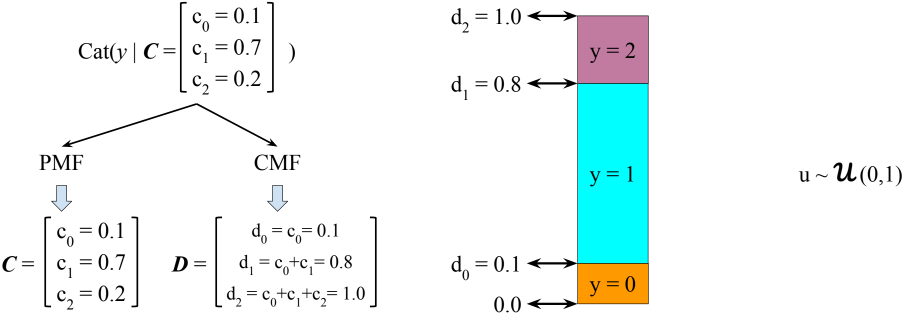


Figure: Sampling a categorical distribution using a Uniform sampler

Sampling a Categorical Distribution

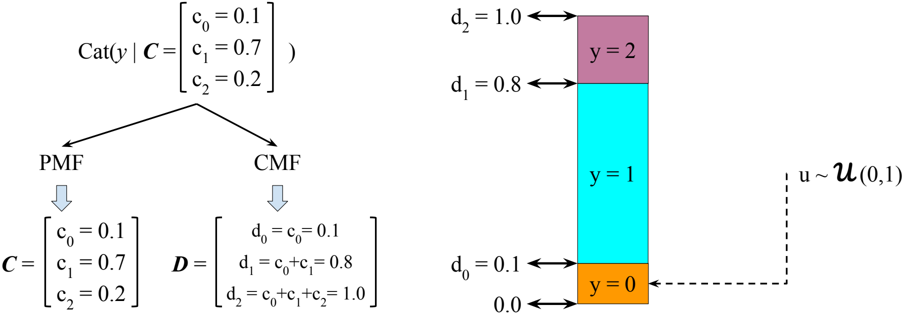


Figure: Sampling a categorical distribution using a Uniform sampler

Sampling a Categorical Distribution

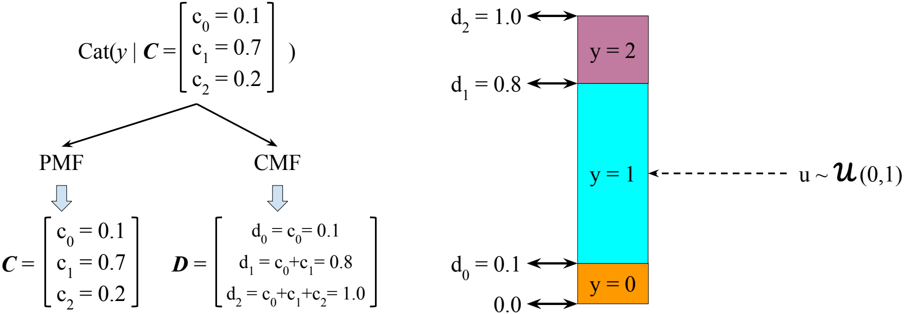


Figure: Sampling a categorical distribution using a Uniform sampler

Sampling a Categorical Distribution

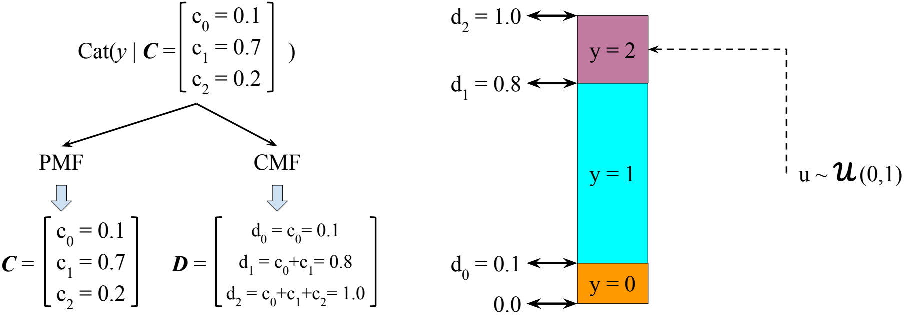


Figure: Sampling a categorical distribution using a Uniform sampler

Subsection 2

Deep Autoregressive Models

Modeling

Model Specification

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 $\boldsymbol{x} \in \mathbb{R}^{784}$.

Modeling

Model Specification

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.

Modeling

Model Specification

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.

Modeling

Model Specification

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 $x_i \in \{0, 1, 2, \ldots, 255\}$

Modeling

Model Specification

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

$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \mid \boldsymbol{x}_{<1}) \times \ldots\ldots \times p(x_d \mid \boldsymbol{x}_{<d}) \times \ldots\ldots \times p(x_{D\text{-}1} \mid \boldsymbol{x}_{<D\text{-}1})$$

Figure: Using logistic regression for generative modeling

Modeling

$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \mid \boldsymbol{x}_{<1}) \times \ldots\ldots \times \boxed{p(x_d \mid \boldsymbol{x}_{<d})} \times \ldots\ldots \times p(x_{D-1} \mid \boldsymbol{x}_{<D-1})$$

Figure: Using logistic regression for generative modeling

Modeling

$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \mid \boldsymbol{x}_{<1}) \times \ldots\ldots \times p(x_d \mid \boldsymbol{x}_{<d}) \times \ldots\ldots \times p(x_{D-1} \mid \boldsymbol{x}_{<D-1})$$

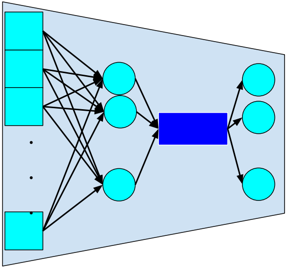
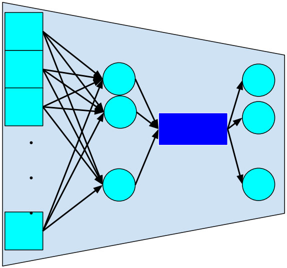


Figure: Using logistic regression for generative modeling

$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \mid \boldsymbol{x}_{<1}) \times \ldots\ldots \times p(x_d \mid \boldsymbol{x}_{<d}) \times \ldots\ldots \times p(x_{D-1} \mid \boldsymbol{x}_{<D-1})$$



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

Figure: Using logistic regression for generative modeling

Modeling

$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \mid \boldsymbol{x}_{<1}) \times \ldots\ldots \times p(x_d \mid \boldsymbol{x}_{<d}) \times \ldots\ldots \times p(x_{D-1} \mid \boldsymbol{x}_{<D-1})$$

Figure: Using logistic regression for generative modeling

Modeling

$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \mid \boldsymbol{x}_{<1}) \times \ldots\ldots \times p(x_d \mid \boldsymbol{x}_{<d}) \times \ldots\ldots \times p(x_{D-1} \mid \boldsymbol{x}_{<D-1})$$

$p(x_0) \rightarrow \boldsymbol{b}_0$

Figure: Using logistic regression for generative modeling

Modeling

$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \mid \boldsymbol{x}_{<1}) \times \ldots\ldots \times p(x_d \mid \boldsymbol{x}_{<d}) \times \ldots\ldots \times p(x_{D-1} \mid \boldsymbol{x}_{<D-1})$$

$\boldsymbol{b}_0$

$\boldsymbol{b}_1, \boldsymbol{W}_1$

Figure: Using logistic regression for generative modeling

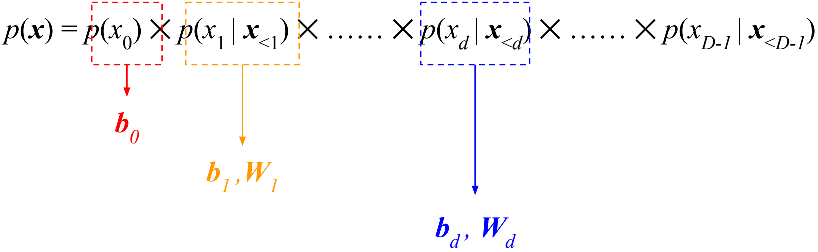


Figure: Using logistic regression for generative modeling

Modeling

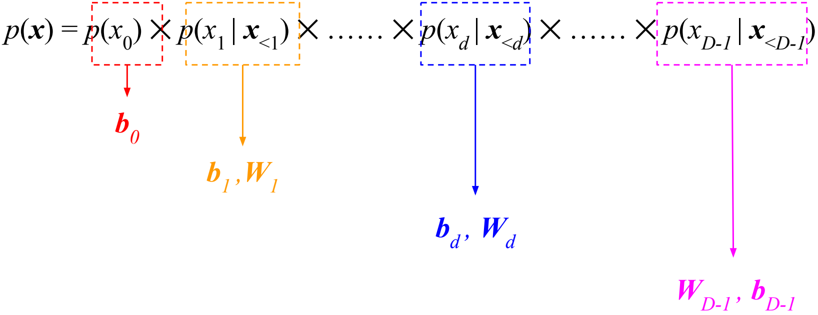


Figure: Using logistic regression for generative modeling

$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \mid \boldsymbol{x}_{<1}) \times \ldots\ldots \times p(x_d \mid \boldsymbol{x}_{<d}) \times \ldots\ldots \times p(x_{D-1} \mid \boldsymbol{x}_{<D-1})$$

$\boldsymbol{b}_0$

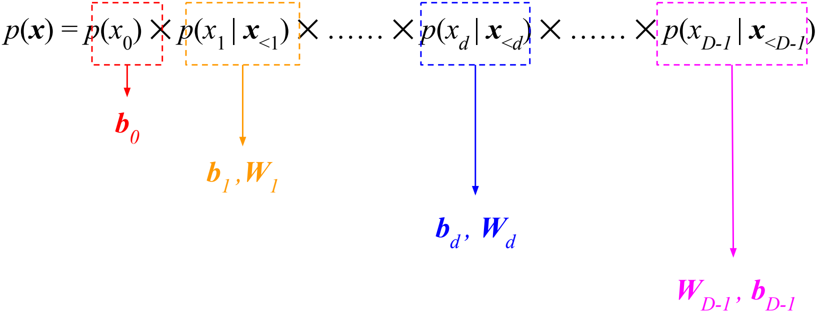
$\boldsymbol{b}_1, \boldsymbol{W}_1$

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

$\boldsymbol{W}_{D-1}, \boldsymbol{b}_{D-1}$

$$x_d \in \{0,1,\ldots,255\} \Longrightarrow \begin{cases} \boldsymbol{b}_d \in R^{256} \\ \boldsymbol{W}_d \in R^{256\times d} \end{cases} \forall \quad 0 \leq d \leq \text{D-1}$$

Figure: Using logistic regression for generative modeling



$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$

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

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:

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

We can rewrite $L(\boldsymbol{\theta})$ as:

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

Distance Metric

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

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

We can rewrite $L(\boldsymbol{\theta})$ as:

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

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

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

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}

Training

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

Monte Carlo Estimation

Consider the following expectation:

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

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

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:

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

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:

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

Parametric Density Calculation

$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\theta = \{ b_0, b_1, W_1, \ldots, b_d, W_d, \ldots, b_{D-1}, W_{D-1} \}$$

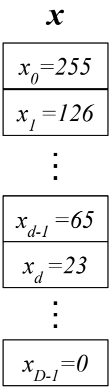
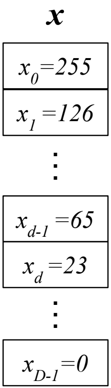


Figure: Calculating the likelihood as a function of model parameters

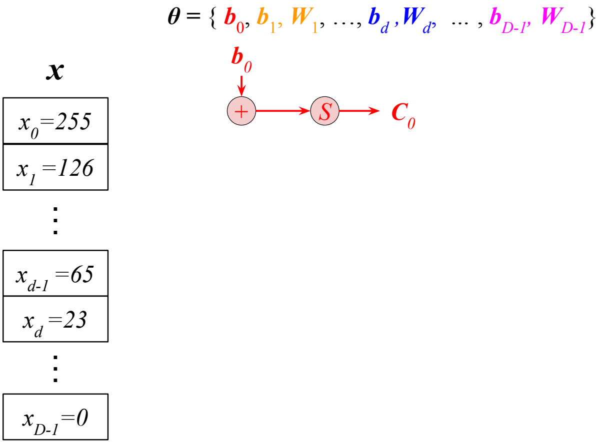
$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$



$$p(\boldsymbol{x}) = p(x_0)p(x_1 | \boldsymbol{x}_{<1}) \ldots p(x_d | \boldsymbol{x}_{<d}) \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

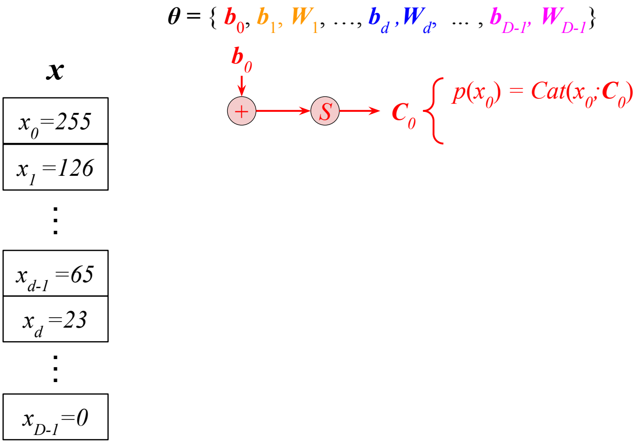
Parametric Density Calculation



$$p(\boldsymbol{x}) = p(x_0)p(x_1 | \boldsymbol{x}_{<1}) \ldots\ldots p(x_d | \boldsymbol{x}_{<d}) \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

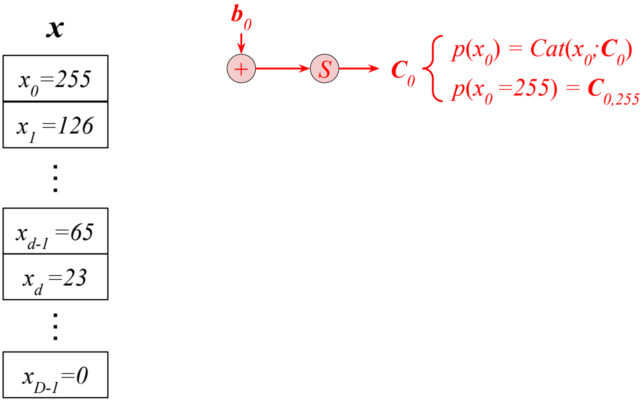


$$p(\boldsymbol{x}) = p(x_0)p(x_1|\, \boldsymbol{x}_{<1}) \;....\; p(x_d|\, \boldsymbol{x}_{<d}) \;...\; p(x_{D-1}|\, \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$

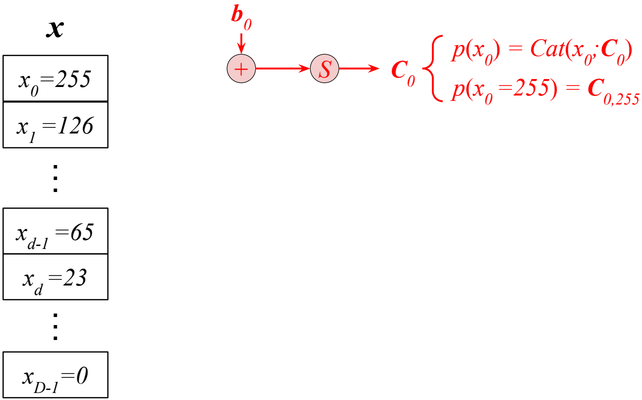


$$p(\boldsymbol{x}) = p(x_0)p(x_1 | \boldsymbol{x}_{<1}) \ldots p(x_d | \boldsymbol{x}_{<d}) \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$

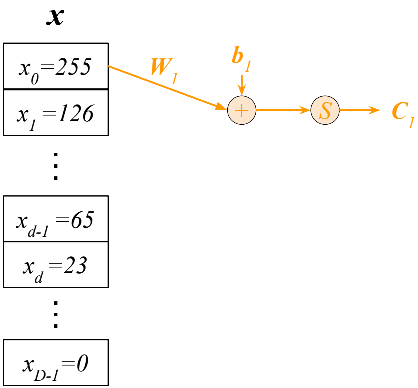


$$p(\boldsymbol{x}) = \boldsymbol{C}_{0,255}\, p(x_1 | \boldsymbol{x}_{<1}) \ldots\, p(x_d | \boldsymbol{x}_{<d}) \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$

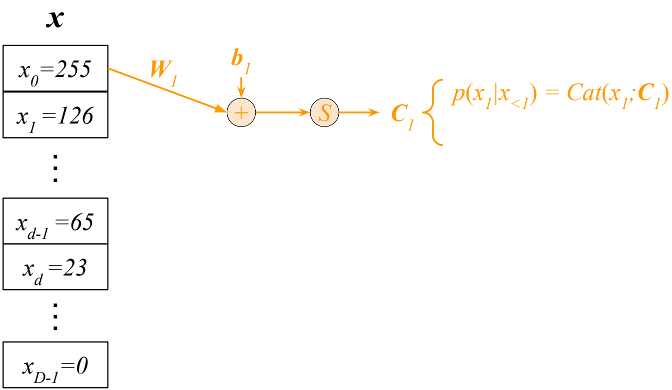


$$p(\boldsymbol{x}) = \boldsymbol{C}_{0,255}\, p(x_1 | \boldsymbol{x}_{<1}) \ldots.\, p(x_d | \boldsymbol{x}_{<d}) \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$

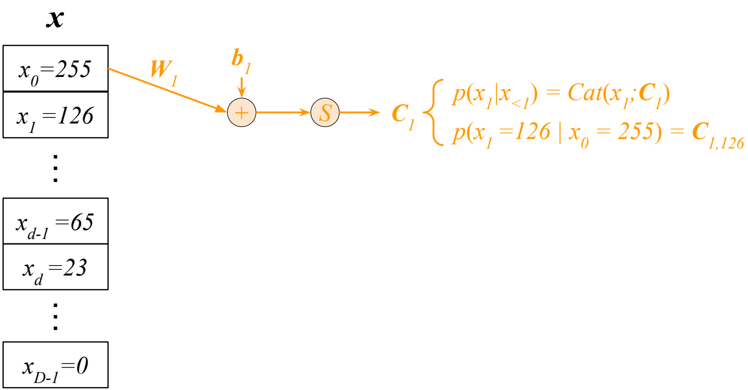


$$p(\boldsymbol{x}) = \boldsymbol{C}_{0,255}\, p(x_1 | \boldsymbol{x}_{<1}) \ldots\, p(x_d | \boldsymbol{x}_{<d}) \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$

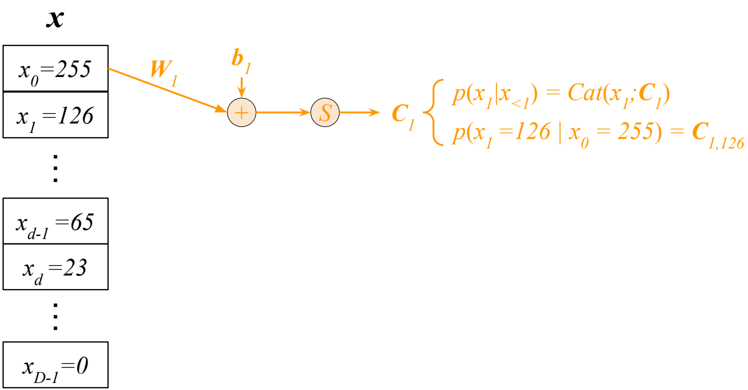


$$p(\boldsymbol{x}) = \boldsymbol{C}_{0,255}\, p(x_1 | \boldsymbol{x}_{<1}) \ldots\ldots p(x_d | \boldsymbol{x}_{<d}) \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\theta = \{ b_0, b_1, W_1, \ldots, b_d, W_d, \ldots, b_{D-1}, W_{D-1} \}$$

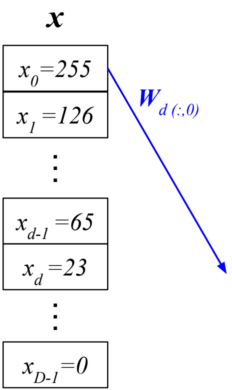


$$p(\boldsymbol{x}) = C_{0,255}\, C_{1,126} \ldots\, p(x_d \mid \boldsymbol{x}_{<d}) \ldots p(x_{D-1} \mid \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$

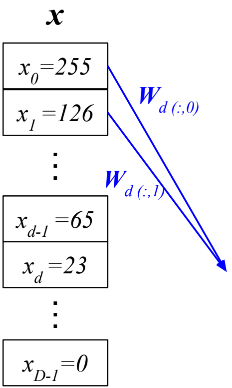


$$p(\boldsymbol{x}) = \boldsymbol{C}_{0,255}\ \boldsymbol{C}_{1,126} \ldots.\ p(x_d | \boldsymbol{x}_{<d}) \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$

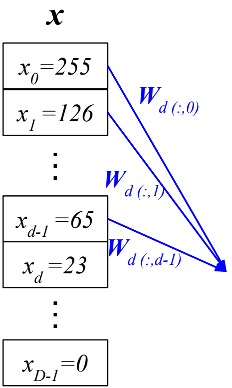


$$p(\boldsymbol{x}) = C_{0,255}\ C_{1,126} \ldots.\ p(x_d \mid \boldsymbol{x}_{<d}) \ldots p(x_{D-1} \mid \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$

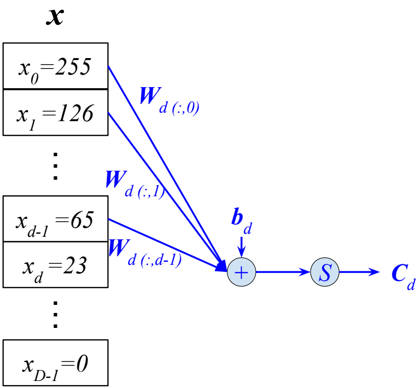


$$p(\boldsymbol{x}) = C_{0,255}\, C_{1,126} \ldots p(x_d \mid \boldsymbol{x}_{<d}) \ldots p(x_{D-1} \mid \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

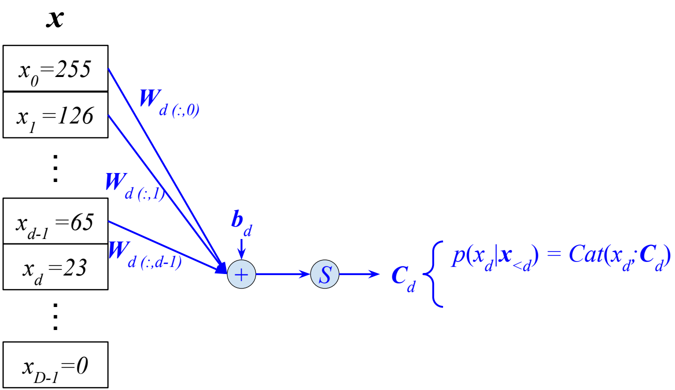
$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$



$$p(\boldsymbol{x}) = \boldsymbol{C}_{0,255}\ \boldsymbol{C}_{1,126} \ldots.\ p(x_d | \boldsymbol{x}_{<d}) \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

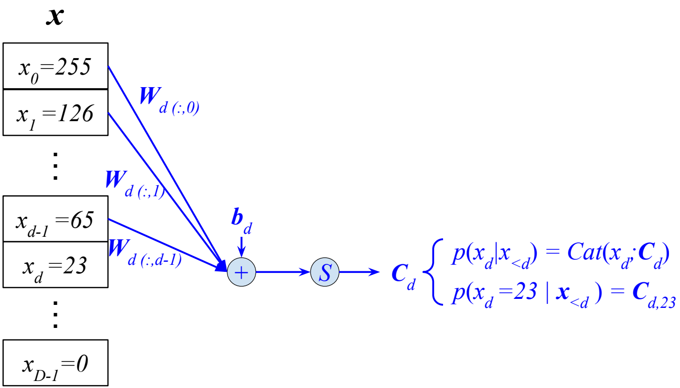
$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$



$$p(\boldsymbol{x}) = \boldsymbol{C}_{0,255}\ \boldsymbol{C}_{1,126} \ldots\ p(x_d | \boldsymbol{x}_{<d}) \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$

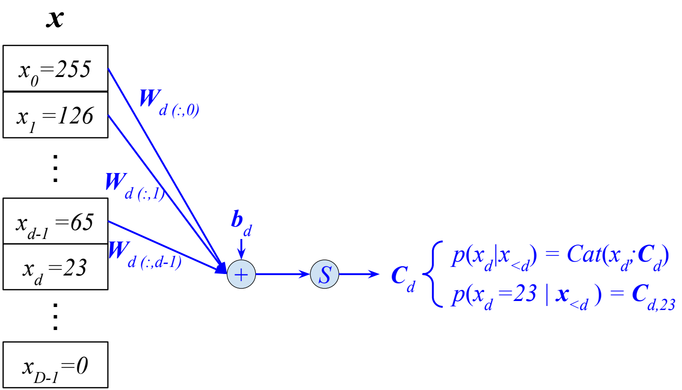


$$p(\boldsymbol{x}) = \boldsymbol{C}_{0,255}\ \boldsymbol{C}_{1,126} \ldots\ p(x_d | \boldsymbol{x}_{<d}) \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

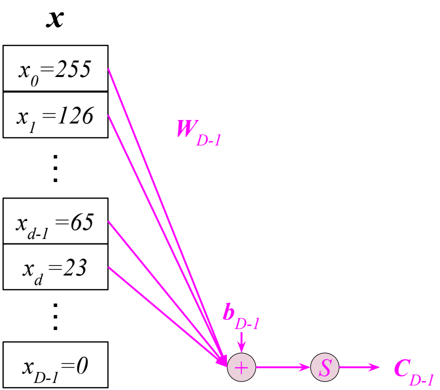
$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$



$$p(\boldsymbol{x}) = \boldsymbol{C}_{0,255}\ \boldsymbol{C}_{1,126} \ldots \boldsymbol{C}_{d,23} \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

$$\boldsymbol{\theta} = \{\, \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$

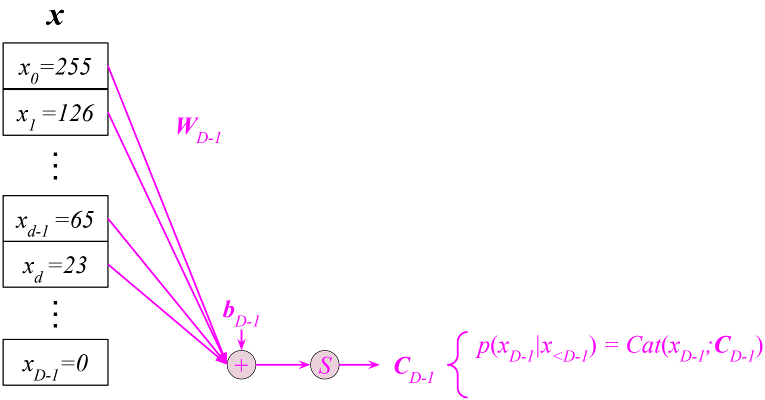


$$p(\boldsymbol{x}) = \boldsymbol{C}_{0,255}\ \boldsymbol{C}_{1,126} \ldots\ \boldsymbol{C}_{d,23} \ldots p(x_{D-1} \mid \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\theta = \{ b_0, b_1, W_1, \ldots, b_d, W_d, \ldots, b_{D-1}, W_{D-1} \}$$

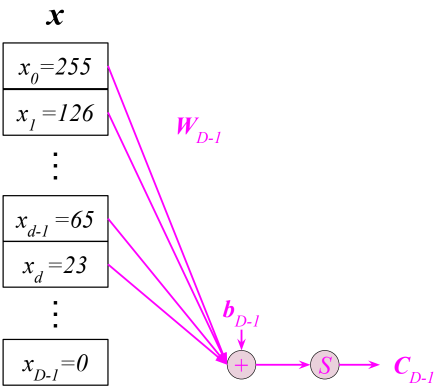


$$p(\boldsymbol{x}) = C_{0,255}\ C_{1,126}\ \ldots\ C_{d,23}\ \ldots\ p(x_{D-1} \mid \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$



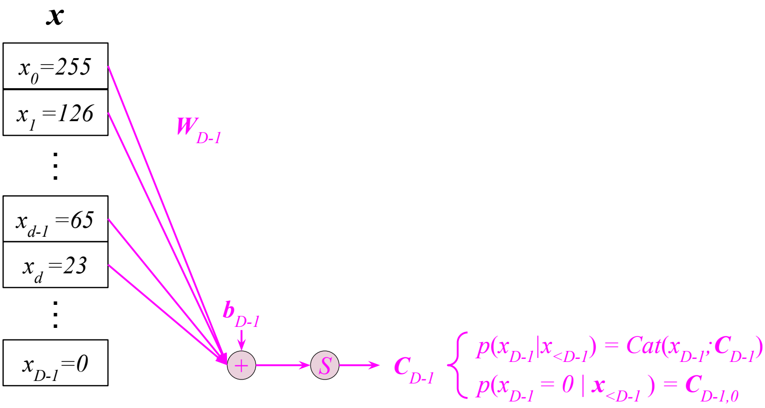
$$\begin{cases} p(x_{D-1} | x_{<D-1}) = Cat(x_{D-1}; \boldsymbol{C}_{D-1}) \\ p(x_{D-1} = 0 \mid \boldsymbol{x}_{<D-1}) = \boldsymbol{C}_{D-1,0} \end{cases}$$

$$p(\boldsymbol{x}) = \boldsymbol{C}_{0,255}\ \boldsymbol{C}_{1,126} \ldots \boldsymbol{C}_{d,23} \ldots p(x_{D-1} | \boldsymbol{x}_{<D-1})$$

Figure: Calculating the likelihood as a function of model parameters

Parametric Density Calculation

$$\boldsymbol{\theta} = \{ \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{W}_1, \ldots, \boldsymbol{b}_d, \boldsymbol{W}_d, \ldots, \boldsymbol{b}_{D-1}, \boldsymbol{W}_{D-1} \}$$



$$p(\boldsymbol{x}) = \boldsymbol{C}_{0,255}\ \boldsymbol{C}_{1,126} \ldots\ \boldsymbol{C}_{d,23} \ldots \boldsymbol{C}_{D-1,0}$$

Figure: Calculating the likelihood as a function of model parameters

Sampling from a Generative Model

$$\boldsymbol{\theta}^{\star} = \{\, \boldsymbol{b}_0^{\star}, \boldsymbol{b}_1^{\star}, \boldsymbol{W}_1^{\star}, \ldots, \boldsymbol{b}_d^{\star}, \boldsymbol{W}_d^{\star}, \ldots, \boldsymbol{b}_{D-1}^{\star}, \boldsymbol{W}_{D-1}^{\star} \}$$

Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

$$\boldsymbol{\theta}^{\star} = \{\, \boldsymbol{b}_0^{\star}, \boldsymbol{b}_1^{\star}, \boldsymbol{W}_1^{\star}, \ldots, \boldsymbol{b}_d^{\star}, \boldsymbol{W}_d^{\star}, \ \ldots\ , \boldsymbol{b}_{D-1}^{\star}, \boldsymbol{W}_{D-1}^{\star} \}$$

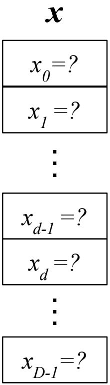


Figure: Sampling a trained Autoregressive Model

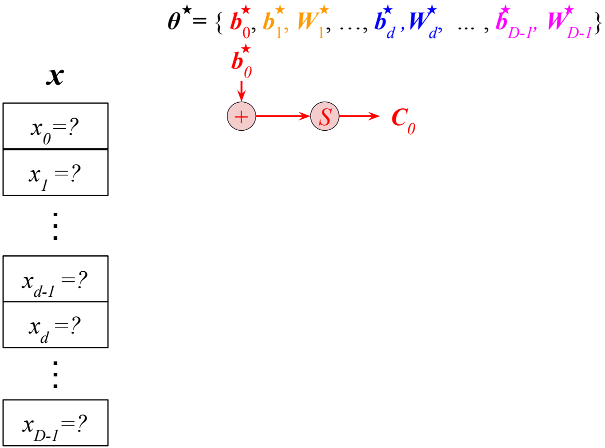


Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

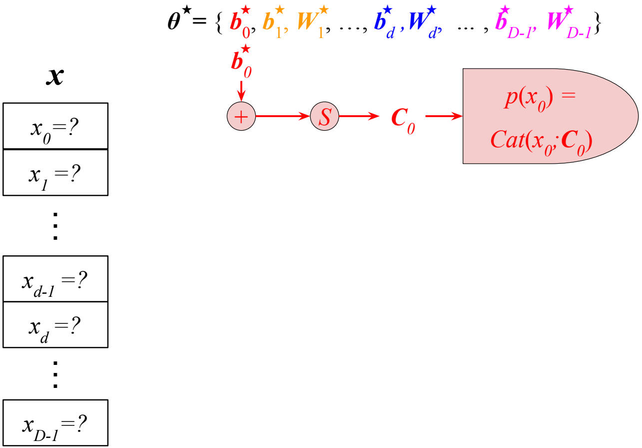


Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

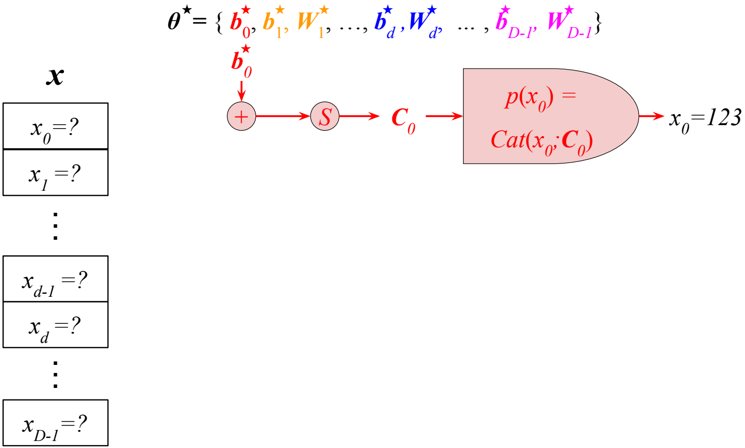


Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

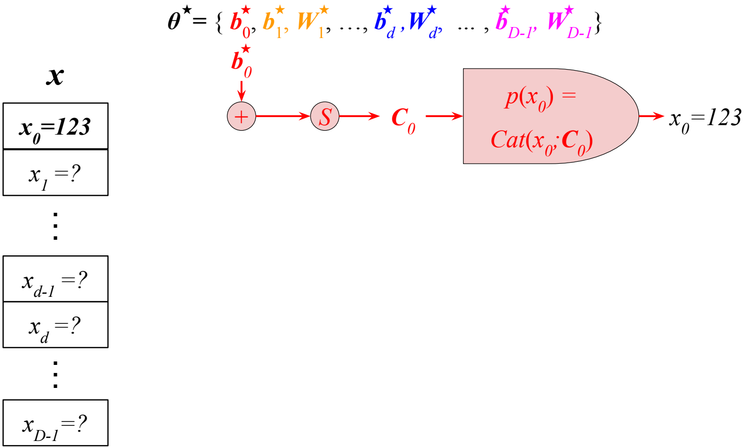


Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

$$\theta^\star = \{ b_0^\star, b_1^\star, W_1^\star, \ldots, b_d^\star, W_d^\star, \ldots, b_{D-1}^\star, W_{D-1}^\star \}$$

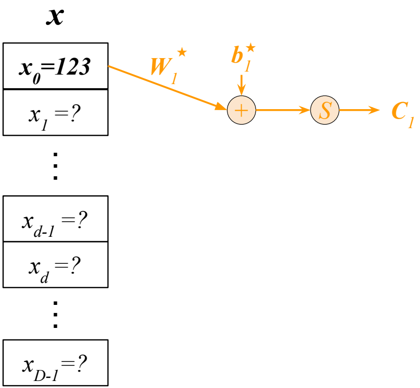


Figure: Sampling a trained Autoregressive Model

$$\theta^\star = \{ b_0^\star, b_1^\star, W_1^\star, \ldots, b_d^\star, W_d^\star, \ldots, b_{D-1}^\star, W_{D-1}^\star \}$$

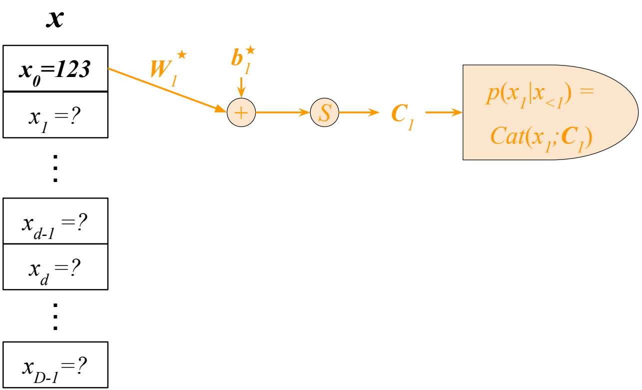


Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

$$\theta^\star = \{ b_0^\star, b_1^\star, W_1^\star, \ldots, b_d^\star, W_d^\star, \ldots, b_{D-1}^\star, W_{D-1}^\star \}$$

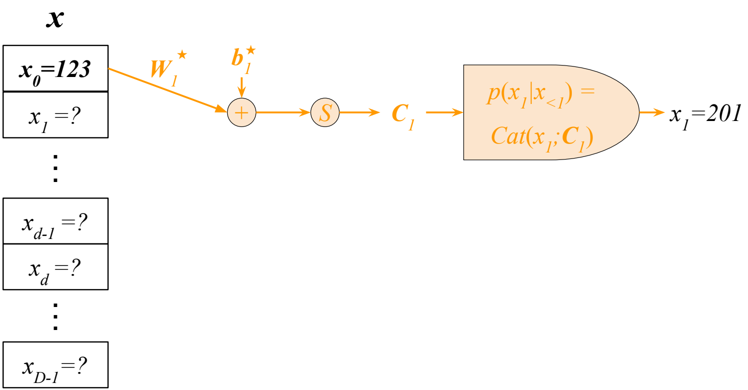


Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

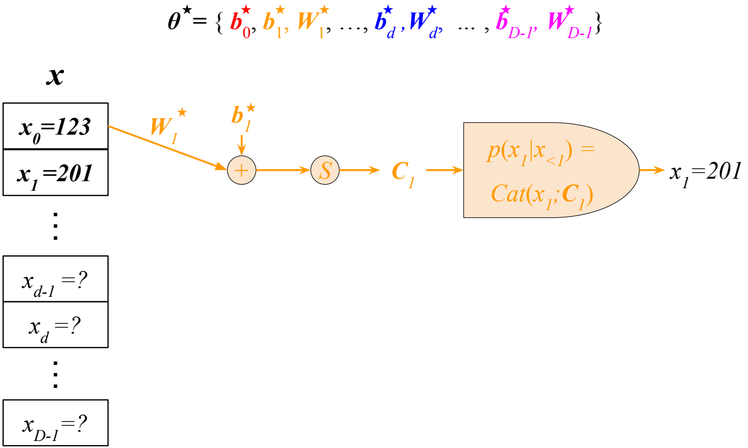


Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

$\theta^\star = \{ b_0^\star, b_1^\star, W_1^\star, \ldots, b_d^\star, W_d^\star, \ldots, b_{D-1}^\star, W_{D-1}^\star \}$

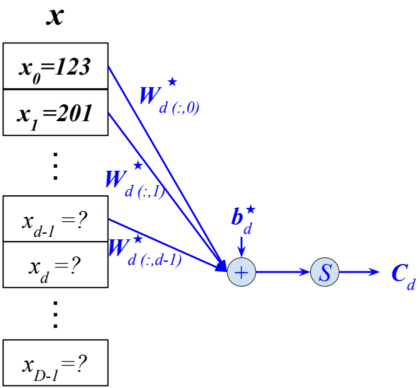


Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

$$\theta^{\star} = \{ b_0^{\star}, b_1^{\star}, W_1^{\star}, \ldots, b_d^{\star}, W_d^{\star}, \ldots, b_{D-1}^{\star}, W_{D-1}^{\star} \}$$

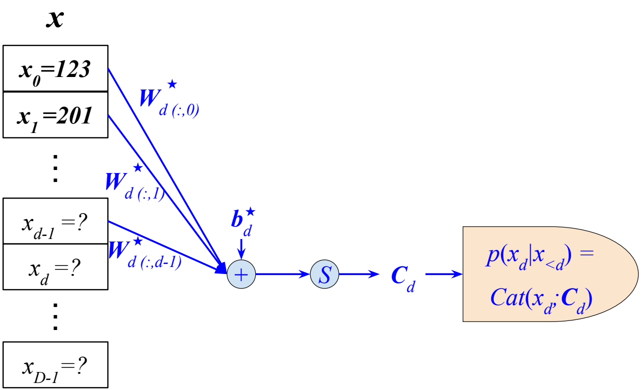


Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

$$\theta^\star = \{ b_0^\star, b_1^\star, W_1^\star, \ldots, b_d^\star, W_d^\star, \ldots, b_{D-1}^\star, W_{D-1}^\star \}$$

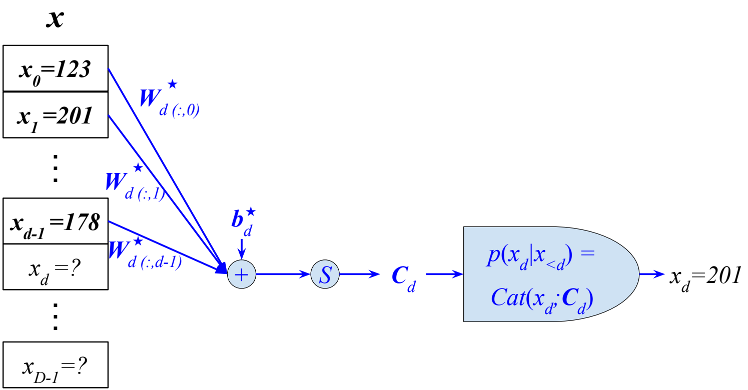


Figure: Sampling a trained Autoregressive Model

$$\theta^{\star} = \{ b_0^{\star}, b_1^{\star}, W_1^{\star}, \ldots, b_d^{\star}, W_d^{\star}, \ldots, b_{D-1}^{\star}, W_{D-1}^{\star} \}$$

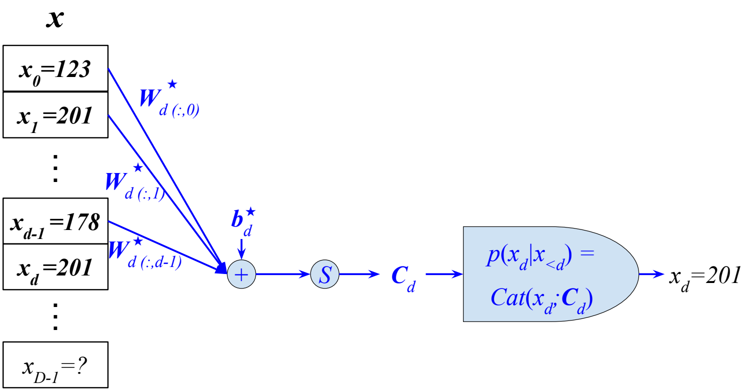


Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

$$\theta^\star = \{ b_0^\star, b_1^\star, W_1^\star, \ldots, b_d^\star, W_d^\star, \ldots, b_{D-1}^\star, W_{D-1}^\star \}$$

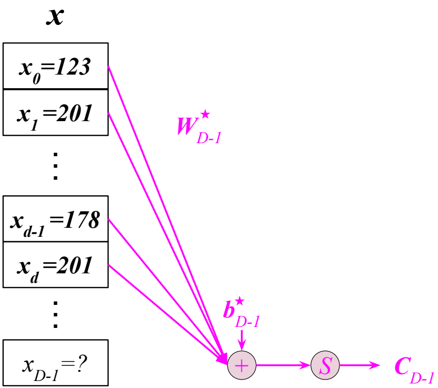


Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

$$\theta^\star = \{ b_0^\star, b_1^\star, W_1^\star, \ldots, b_d^\star, W_d^\star, \ldots, b_{D-1}^\star, W_{D-1}^\star \}$$

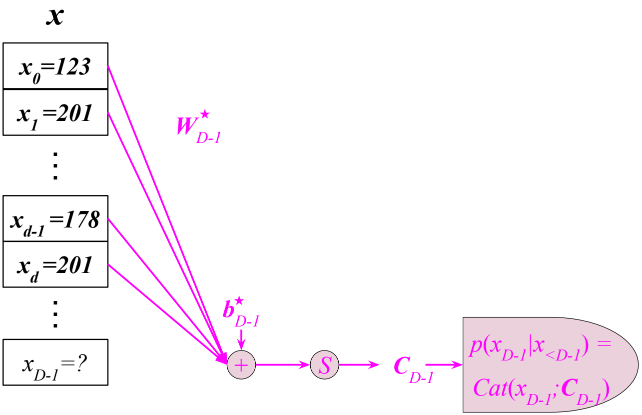


Figure: Sampling a trained Autoregressive Model

Sampling from a Generative Model

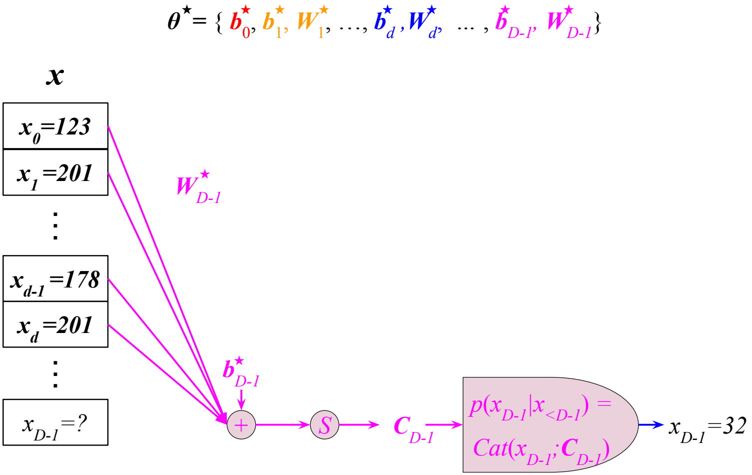


Figure: Sampling a trained Autoregressive Model

$$\theta^\star = \{ b_0^\star, b_1^\star, W_1^\star, \ldots, b_d^\star, W_d^\star, \ldots, b_{D-1}^\star, W_{D-1}^\star \}$$

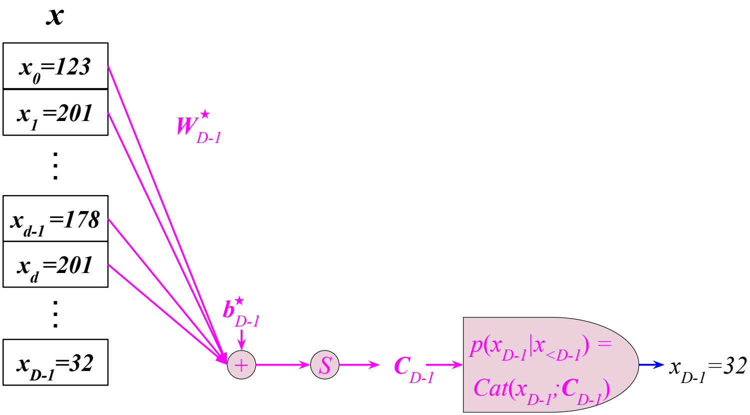


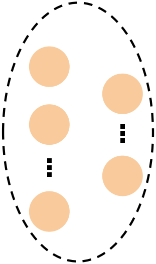
Figure: Sampling a trained Autoregressive Model

Section 7

Extensions

Some of Autoregressive Modeling Extensions

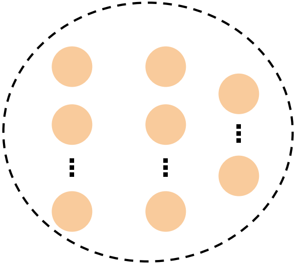
$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \mid \boldsymbol{x}_{<1}) \times \ldots\ldots \times p(x_d \mid \boldsymbol{x}_{<d}) \times \ldots\ldots \times p(x_{D\text{-}1} \mid \boldsymbol{x}_{<D\text{-}1})$$



Fully Visible Sigmoid Belief Networks
(FVSBN)

Some of Autoregressive Modeling Extensions

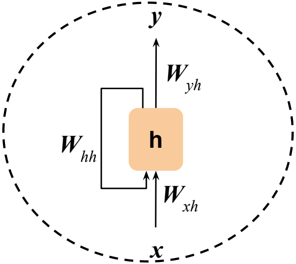
$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \mid \boldsymbol{x}_{<1}) \times \ldots\ldots \times p(x_d \mid \boldsymbol{x}_{<d}) \times \ldots\ldots \times p(x_{D\text{-}1} \mid \boldsymbol{x}_{<D\text{-}1})$$



Neural Autoregressive Density Estimation
(NADE)

Some of Autoregressive Modeling Extensions

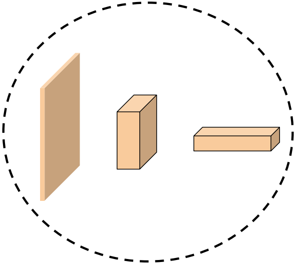
$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \mid \boldsymbol{x}_{<1}) \times \ldots\ldots \times p(x_d \mid \boldsymbol{x}_{<d}) \times \ldots\ldots \times p(x_{D\text{-}1} \mid \boldsymbol{x}_{<D\text{-}1})$$



Pixel Recurrent Neural Networks
(PixelRNN)

Some of Autoregressive Modeling Extensions

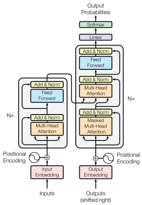
$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \,|\, \boldsymbol{x}_{<1}) \times \ldots\ldots \times p(x_d \,|\, \boldsymbol{x}_{<d}) \times \ldots\ldots \times p(x_{D\text{-}1} \,|\, \boldsymbol{x}_{<D\text{-}1})$$



Pixel Convolutional Neural Networks
(PixelCNN)

Some of Autoregressive Modeling Extensions

$$p(\boldsymbol{x}) = p(x_0) \times p(x_1 \mid \boldsymbol{x}_{<1}) \times \ldots\ldots \times p(x_d \mid \boldsymbol{x}_{<d}) \times \ldots\ldots \times p(x_{D\text{-}1} \mid \boldsymbol{x}_{<D\text{-}1})$$

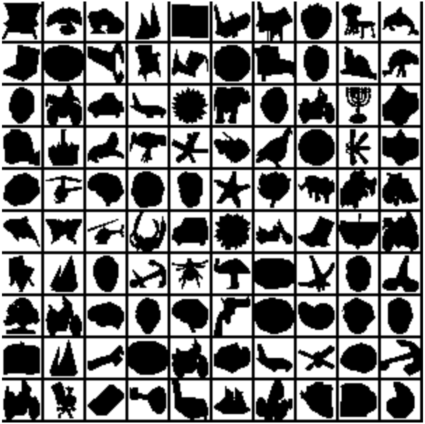


Transformer
(ChatGPT)

Section 8

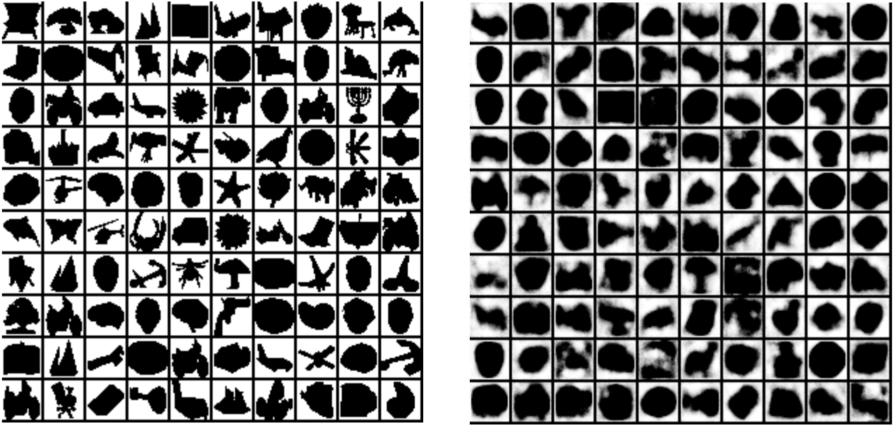
Results

FVSBN



(a) Dataset samples

FVSBN



(a) Dataset samples

(b) Generated samples

Figure: FVSBN performance over Caltech 101 dataset (source: [5])

NADE

Figure: NADE performance over BMNIST dataset (source: [6])

PixelRNN

Figure: Pixel RNN results in image completion (source: [7])

PixelCNN++

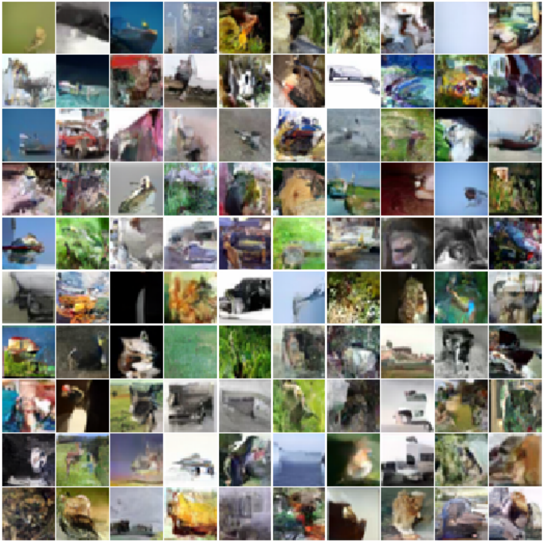


Figure: Samples from our PixelCNN model trained on CIFAR-10 (source: [8])

Section 9

Applications

Adversarial Robustness

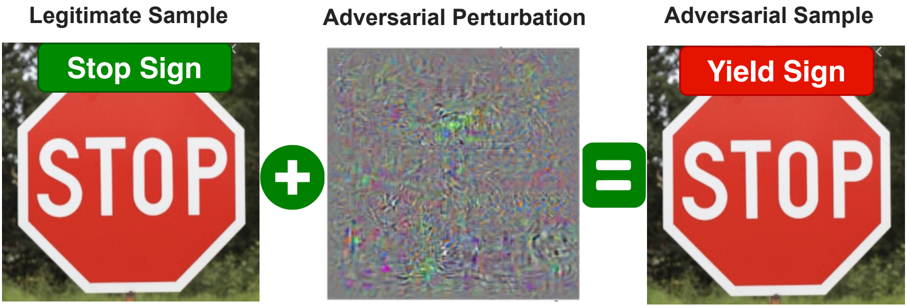


Figure: Different adversarial attacks to Frog image from Cifar10 dataset (source: [9])

Adversarial Robustness

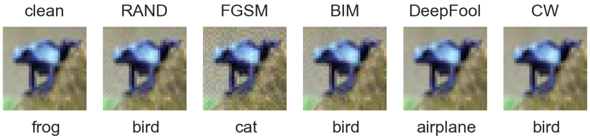
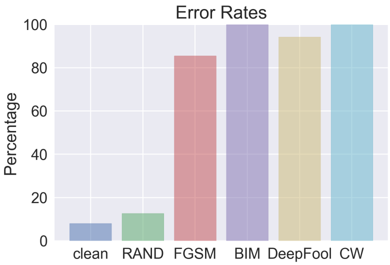
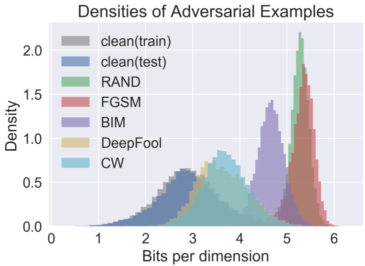


Figure: Sample adversarial attack to deep learning architectures (source: [9])



(a) Error rate



(b) Density change

Figure: Using autoregressive models to detect adversarial samples (source: [9])

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, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al.,
"Photorealistic text-to-image diffusion models with deep language understanding,"
Advances in Neural Information Processing Systems, vol. 35, pp. 36479–36494, 2022.

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.

Zhe Gan, Ricardo Henao, David Carlson, and Lawrence Carin,
"Learning deep sigmoid belief networks with data augmentation,"
in *Artificial Intelligence and Statistics*. PMLR, 2015, pp. 268–276.

References II

Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle,
"Neural autoregressive distribution estimation,"
The Journal of Machine Learning Research, vol. 17, no. 1, pp. 7184–7220, 2016.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu,
"Pixel recurrent neural networks,"
in *International conference on machine learning*. PMLR, 2016, pp. 1747–1756.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma,
"Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications,"
arXiv preprint arXiv:1701.05517, 2017.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman,
"Pixeldefend: Leveraging generative models to understand and defend against adversarial examples,"
arXiv preprint arXiv:1710.10766, 2017.