QUESTIONS

LOGITS

$f(\mathbf{x}) = \text{softmax}(\ \mathbf{W}\quad \mathbf{x}\ )$

Are the values of the model before the soft-max.

CROSS ENTROPY FORMULA

The cross-entropy loss is defined for two vectors $\mathbf{y}, \hat{\mathbf{y}} \in \Delta_c$ as:

$$\text{CE}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_i y_i \log(\hat{y}_i)\ .$$

$$\text{CE}(\mathbf{y}, \hat{\mathbf{y}}) = -\log(\hat{y}_t)$$

One-hot encoding -> just one yi is 1

RNN FORMULA

$$\mathbf{H}_t = \phi(\mathbf{X}_t \mathbf{W}_{xh} + \mathbf{H}_{t-1} \mathbf{W}_{hh} + \mathbf{b}_h)$$
$$\mathbf{O} = \mathbf{H}\mathbf{W}_{hq} + \mathbf{b}_q$$

GRU

$$\mathbf{R}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xr} + \mathbf{H}_{t-1} \mathbf{W}_{hr} + \mathbf{b}_r)$$
$$\mathbf{Z}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xz} + \mathbf{H}_{t-1} \mathbf{W}_{hz} + \mathbf{b}_z)$$
$$\tilde{\mathbf{H}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{xh} + (\mathbf{R}_t \odot \mathbf{H}_{t-1}) \mathbf{W}_{hh} + \mathbf{b}_h)$$

$$\mathbf{H}_t = \mathbf{Z}_t \odot \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \odot \tilde{\mathbf{H}}_t$$

LSTM

$$\mathbf{I}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xi} + \mathbf{H}_{t-1} \mathbf{W}_{hi} + \mathbf{b}_i),$$
$$\mathbf{F}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xf} + \mathbf{H}_{t-1} \mathbf{W}_{hf} + \mathbf{b}_f)$$
$$\mathbf{O}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xo} + \mathbf{H}_{t-1} \mathbf{W}_{ho} + \mathbf{b}_o),$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{xc} + \mathbf{H}_{t-1} \mathbf{W}_{hc} + \mathbf{b}_c)$$
$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tilde{\mathbf{C}}_t$$
$$\mathbf{H}_t = \mathbf{O}_t \odot \tanh(\mathbf{C}_t)$$

LOSS GAN:

$$J_D = -\frac{1}{2}\,\mathrm{E}_{\mathbf{x}\sim p_{\text{data}}} \log D\left(\mathbf{x}\right) - \frac{1}{2}\,\mathrm{E}_{\mathbf{z}\sim p_G(\mathbf{z})} \log\left(1 - D\left(G\left(\mathbf{z}\right)\right)\right)$$
$$J_G = -J_D$$

MINMAX The generator minimizes the log-probability of the discriminator being correct.

$$J_D = -\frac{1}{2}\,\mathrm{E}_{\mathbf{x}\sim p_{\text{data}}} \log D\left(\mathbf{x}\right) - \frac{1}{2}\,\mathrm{E}_{\mathbf{z}\sim p_G(\mathbf{z})} \log\left(1 - D\left(G\left(\mathbf{z}\right)\right)\right)$$
$$J_G = -\frac{1}{2}\,\mathrm{E}_{\mathbf{z}\sim p_G(\mathbf{z})} \log D\left(G\left(\mathbf{z}\right)\right)$$

NON-SATURATING GAME the generator maximizes the log-probability of the discriminator being mistaken.
The non-saturating game is heuristically motivated. Generator can still learn even when discriminator successfully rejects all generator samples.

BATCH NORMALIZATION

$$\mathbf{X}' = \frac{\mathbf{X} - \mu}{\sqrt{\sigma^2}}$$

$$\tilde{\mu}_j = \frac{1}{b}\sum_i [\mathbf{H}]_{i,j}\ ,\quad \tilde{\sigma}_j^2 = \frac{1}{b}\sum_i ([\mathbf{H}]_{i,j} - \tilde{\mu}_j)^2$$

Empirical mean and variance

$$\mathbf{H}' = \frac{\mathbf{H} - \tilde{\mu}}{\sqrt{\tilde{\sigma}^2 + \varepsilon}}$$

BN in convolution layers

$$\tilde{\mu}_z = \frac{1}{bhw}\sum_{i,j,k} [\mathbf{H}]_{i,j,k,z}\ ,\quad \tilde{\sigma}_z^2 = \frac{1}{bhw}\sum_{i,j,k} ([\mathbf{H}]_{i,j,k,z} - \tilde{\mu}_z)^2$$

Where h and w are the height and width respectively of the feature map

LAYER NORMALIZATION

$$\tilde{\mu}_i = \frac{1}{f}\sum_j [\mathbf{H}]_{i,j}\ ,\quad \tilde{\sigma}_i^2 = \frac{1}{f}\sum_j ([\mathbf{H}]_{i,j} - \tilde{\mu}_i)^2$$

CONVOLUTION

$$h[i,j] = u[i,j] + \sum_{k_1,k_2} W[i,j,k_1,k_2] \cdot x[i+k_1, j+k_2]$$

Translation Invariance-> a shift in the inputs x should lead to a shift in h -> W and u docent depend on i and j

$$h[i,j] = u + \sum_{k_1,k_2} W[k_1,k_2] \cdot x[i+k_1, j+k_2]$$

Locality principle This means that outside some range |k1|, |k2| > Δ, we set W[k1, k2] = 0:

$$h[i,j] = u + \sum_{k_1=-\Delta}^{\Delta} \sum_{k_2=-\Delta}^{\Delta} W[k_1,k_2] \cdot x[i+k_1, j+k_2]$$

QUESTIONS

DILATED CONVOLUTION

$$y[i,j] = \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} h[k_1,k_2]\, x[i - d\cdot k_1, j - d\cdot k_2]$$

which expands the receptive field without loss of resolution. The expansion of the filter allows to increase its dimensions by filling the empty positions with zeros.

CAUSAL CONVOLUTION

$$\underset{(n,c')}{[\mathbf{H}]_{i,d}} = \phi\left(\sum_{i'=-k}^{+k}\sum_{z=1}^{c} \underset{(s,c,c')}{[W]_{i'+k+1,z,d}}[\mathbf{X}]_{i+i',z}\right)$$

n temporal seq, c' output features, n temporal seq, c input features.

i temporal indices through n, d indices trough c', i' temporal kernel indices, k dim of window convolution , z features, s filter dimensions

$$\underset{(n,c')}{[\mathbf{H}]_{i,d}} = \phi\left(\sum_{i'=0}^{k}\sum_{z=1}^{c} \underset{(k+1,c,c')}{[W]_{i'+1,z,d}}[\mathbf{X}]_{i-i',z}\right)$$

i-i' stands for only the left values of the x

SELF ATTENTION LAYER

$$\mathbf{h}_i = \sum_{j=-k}^{k} \mathcal{W}_j \mathbf{x}_{i+j}$$ 1D Convolution-> Remove Locality

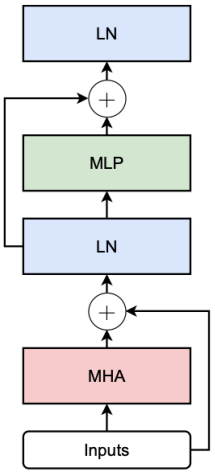
$$\mathbf{h}_i = \sum_{j=-i+1}^{n-i} g(i+j)\mathbf{x}_j$$ Continuous Convolution

$$\mathbf{h}_i = \sum_{j=1}^{n} \alpha(\mathbf{x}_i, \mathbf{x}_j)\mathbf{x}_j$$ Remove regular Dependencies Non local nn

$$\mathbf{h}_i = \sum_{j=1}^{n} \text{softmax}_j\left(\frac{1}{\sqrt{d}}\mathbf{x}_i^\top \mathbf{x}_j\right)\mathbf{x}_j$$ Normdot prod Fast efficient parallelize

$$\underset{(n,q)}{\mathbf{Q}} = \mathbf{X}\mathbf{W}_q, \quad \underset{(n,q)}{\mathbf{K}} = \mathbf{X}\mathbf{W}_k, \quad \underset{(n,v)}{\mathbf{V}} = \mathbf{X}\mathbf{W}_v$$

$$\underset{(n,v)}{\mathbf{H}} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{q}}\right)\mathbf{V}.$$



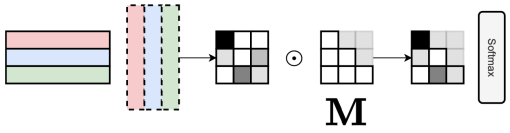
(a) Post-normalized block

$$\mathbf{X}' = [\mathbf{X} \,\|\, \mathbf{E}] \quad \textit{or} \quad \mathbf{X}' = \mathbf{X} + \mathbf{E}$$

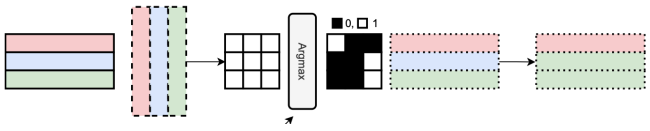
CAUSAL TRANSFORMER

$$\mathbf{H} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top \odot \mathbf{M}}{\sqrt{q}}\right)\mathbf{V}.$$

$$M_{ij} = \begin{cases} 1 & \text{if } j \le i \\ -\infty & \text{otherwise} \end{cases}$$



HARD ATTENTION



EMBEDDINGS

Permutations

$$\text{MHA}(\mathbf{P}\mathbf{X} \,\|\, \mathbf{E}) \neq \mathbf{P}\cdot\text{MHA}(\mathbf{X} \,\|\, \mathbf{E})$$

QUESTIONS

LINEAR TRASFORMERS

$$\mathbf{h}_i = \frac{\sum_j \phi(\mathbf{q}_i)^\top \phi(\mathbf{k}_j)\mathbf{v}_j}{\sum_j \phi(\mathbf{q}_i)^\top \phi(\mathbf{k}_j)} = \frac{\phi(\mathbf{q}_i)^\top \overbrace{\sum_j \phi(\mathbf{k}_j)\mathbf{v}_j^\top}^{\mathbf{S}}}{\phi(\mathbf{q}_i)^\top \underbrace{\sum_j \phi(\mathbf{k}_j)}_{\mathbf{Z}}}.$$

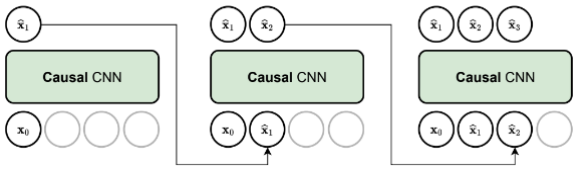
o(n^2) -> o(n)

RELATIVE POSITION EMBEDDING ATTENTION

$$\alpha(\mathbf{x}_i, \mathbf{x}_j, i-j) = \mathbf{x}_i^\top \mathbf{x}_j + b_{ij}$$

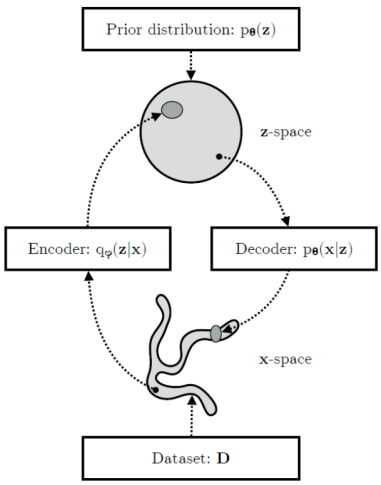
Depend on the relative distance

AUTOREGRESSIVE MODEL FOR TIME SERIES



train the model to predict the next time step by shifting the target sequence and teaching the model to predict future values based only on past data. This enables autoregressive generation, where the model predicts one step at a time, using previous predictions to generate the next step.

VARIATIONAL AUTOENCODER



$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \approx p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) \quad p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\, d\mathbf{z}$$

$$p_{\boldsymbol{\theta}}(\mathbf{x}_1, \ldots, \mathbf{x}_M) = \prod_{j=1}^{M} p_{\boldsymbol{\theta}}(\mathbf{x}_j | A(\mathbf{x}_j))$$

$$\boldsymbol{\eta} = f(A(\mathbf{x}))$$
$$p_{\boldsymbol{\theta}}(\mathbf{x}|A(\mathbf{x})) = p_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{\eta})$$

Parametric inference model- join distribution

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\mathbf{x}) &= \mathrm{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\{\log p_{\boldsymbol{\theta}}(\mathbf{x})\} \\
&= \mathrm{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left\{\log\left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right)\right\} \\
&= \mathrm{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left\{\log\left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right)\right\} \\
&= \underbrace{\mathrm{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left\{\log\left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\right)\right\}}_{\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})} + \underbrace{\mathrm{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left\{\log\left(\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right)\right\}}_{\mathcal{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))}
\end{aligned}$$

ELBO - Evidence Lower Bound

A lower bound on the log-likelihood of the data, that if it is maximized Maximize the marginal likelihood, $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ which improves the generation.
Minimize the KL Divergence improve the approximation of

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \text{ to } p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}).$$

Kullback - Leibler distance distance from 2 probabilities distribution

Calculating the gradient

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) &= \nabla_{\boldsymbol{\theta}} \mathrm{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\{\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\} \\
&= \mathrm{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\{\nabla_{\boldsymbol{\theta}}(\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}))\} \\
&\simeq \nabla_{\boldsymbol{\theta}}(\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})) \\
&= \nabla_{\boldsymbol{\theta}}(\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}))
\end{aligned}$$

QUESTIONS

$$\nabla_{\phi}\mathcal{L}_{\boldsymbol{\theta},\phi}\left(\mathbf{x}\right)=\nabla_{\phi}\,\mathrm{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}\left\{\log p_{\boldsymbol{\theta}}\left(\mathbf{x},\mathbf{z}\right)-\log q_{\phi}\left(\mathbf{z}|\mathbf{x}\right)\right\}$$
$$\neq\mathrm{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}\left\{\nabla_{\boldsymbol{\theta}}\left(\log p_{\boldsymbol{\theta}}\left(\mathbf{x},\mathbf{z}\right)-\log q_{\phi}\left(\mathbf{z}|\mathbf{x}\right)\right)\right\}$$

$$\log d_{\phi}\left(\mathbf{x},\boldsymbol{\epsilon}\right)=\log\left|\det\left(\frac{\partial\mathbf{z}}{\partial\boldsymbol{\epsilon}}\right)\right|=\sum_{i}\log\sigma_{i}$$

REPARAMETRIZATION TRICK

Can't be approximated, difficult to obtain -> Changes of variables of invertible and differentiable

$$\mathbf{z}=g\left(\boldsymbol{\epsilon},\boldsymbol{\phi},\mathbf{x}\right)$$

$$\log q_{\phi}\left(\mathbf{z}|\mathbf{x}\right)=\log p\left(\boldsymbol{\epsilon}\right)-\log d_{\phi}\left(\mathbf{x},\boldsymbol{\epsilon}\right)=\sum_{i}\log\mathcal{N}\left(\epsilon_{i};0,1\right)-\log\sigma_{i}.$$

$$\nabla_{\phi}\,\mathrm{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}\left\{f\left(\mathbf{z}\right)\right\}=\nabla_{\phi}\,\mathrm{E}_{p(\boldsymbol{\epsilon})}\left\{f\left(\mathbf{z}\right)\right\}$$
$$=\mathrm{E}_{p(\boldsymbol{\epsilon})}\left\{\nabla_{\phi}f\left(\mathbf{z}\right)\right\}$$
$$\simeq\nabla_{\phi}f\left(\mathbf{z}\right).$$

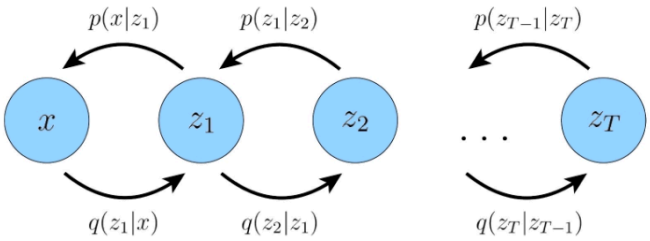
HIERARCHICAL VARIATIONAL AUTOENCODER

A generalization of VAE that extends to multiple hierarchies over latent variables, "more abstract latents"

MARKOV HVAE

Each Transition down hierarchy is Markovian, where decoding each latent zt only condition on previous latent zt+1

$$\mathcal{L}_{\boldsymbol{\theta},\phi}\left(\mathbf{x}\right)=\mathrm{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}\left\{\log p_{\boldsymbol{\theta}}\left(\mathbf{x},\mathbf{z}\right)-\log q_{\phi}\left(\mathbf{z}|\mathbf{x}\right)\right\}$$
$$=\mathrm{E}_{p(\boldsymbol{\epsilon})}\left\{\log p_{\boldsymbol{\theta}}\left(\mathbf{x},\mathbf{z}\right)-\log q_{\phi}\left(\mathbf{z}|\mathbf{x}\right)\right\}$$



Monte Carlo estimator

$$\boldsymbol{\epsilon}\sim p\left(\boldsymbol{\epsilon}\right)$$
$$\mathbf{z}=g\left(\boldsymbol{\phi},\mathbf{x},\boldsymbol{\epsilon}\right)$$
$$\widetilde{\mathcal{L}}_{\boldsymbol{\theta},\phi}\left(\mathbf{x}\right)=\log p_{\boldsymbol{\theta}}\left(\mathbf{x},\mathbf{z}\right)-\log q_{\phi}\left(\mathbf{z}|\mathbf{x}\right)$$

$$p\left(\mathbf{x},\mathbf{z}_{1:T}\right)=p\left(\mathbf{z}_{T}\right)p_{\boldsymbol{\theta}}\left(\mathbf{x}|\mathbf{z}_{1}\right)\prod_{t=2}^{T}p_{\boldsymbol{\theta}}\left(\mathbf{z}_{t-1}|\mathbf{z}_{t}\right)$$

$$q_{\phi}\left(\mathbf{z}_{1:T}|\mathbf{x}\right)=q_{\phi}\left(\mathbf{z}_{1}|\mathbf{x}\right)\prod_{t=2}^{T}q_{\phi}\left(\mathbf{z}_{t}|\mathbf{z}_{t-1}\right)$$

$$\log q_{\phi}\left(\mathbf{z}|\mathbf{x}\right)=\log p\left(\boldsymbol{\epsilon}\right)-\log d_{\phi}\left(\mathbf{x},\boldsymbol{\epsilon}\right)$$

$$\log d_{\phi}\left(\mathbf{x},\boldsymbol{\epsilon}\right)=\log\left|\det\left(\frac{\partial\mathbf{z}}{\partial\boldsymbol{\epsilon}}\right)\right|$$

$$\log p\left(\mathbf{x}\right)\geq\mathrm{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_{1:T}|\mathbf{x})}\left[\log\frac{p\left(\mathbf{x},\mathbf{z}_{1:T}\right)}{q_{\boldsymbol{\phi}}\left(\mathbf{z}_{1:T}|\mathbf{x}\right)}\right]$$

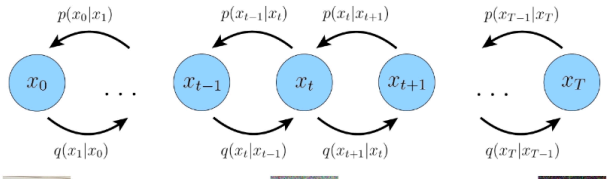
$$q_{\phi}\left(\mathbf{z}|\mathbf{x}\right)=\mathcal{N}\left(\mathbf{z};\boldsymbol{\mu},\mathrm{diag}\left(\boldsymbol{\sigma}^{2}\right)\right)$$

Factorize Gaussian Encoder

$$\boldsymbol{\epsilon}\sim\mathcal{N}\left(0,\mathbf{I}\right)$$
$$\left(\boldsymbol{\mu},\log\boldsymbol{\sigma}\right)=f_{\phi}\left(\mathbf{x}\right)$$
$$\mathbf{z}=\boldsymbol{\mu}+\boldsymbol{\sigma}\odot\boldsymbol{\epsilon}$$

QUESTIONS

VARIATIONAL DIFFUSION MODELS



Latent dimension is exactly equal to the data dimensions.

The structure of the latent encoder at each tilmestep is not learned but pre-defined as a Liner Gaussian model

Parameters of latent encoder vary over time such as way that the distribution of the latent at final time step T is a standard gaussian

$$q\left(\mathbf{x}_{1:T}|\mathbf{x}_0\right)=\prod_{t=1}^{T}q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)$$

$$\boldsymbol{\mu}_t\left(\mathbf{x}_t\right)=\sqrt{\alpha_t}\mathbf{x}_{t-1}\quad\text{and}\quad\boldsymbol{\Sigma}_t\left(\mathbf{x}_t\right)=\left(1-\alpha_t\right)\mathbf{I},$$

$$q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)=\mathcal{N}\left(\mathbf{x}_t;\sqrt{\alpha_t}\mathbf{x}_{t-1},\left(1-\alpha_t\right)\mathbf{I}\right)$$

$$p\left(\mathbf{x}_{0:T}\right)=p\left(\mathbf{x}_T\right)\prod_{t=1}^{T}p_{\boldsymbol{\theta}}\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)$$

where $p\left(\mathbf{x}_T\right)=\mathcal{N}\left(\mathbf{x}_T;\mathbf{0},\mathbf{I}\right).$

$$\begin{aligned}\log p\left(\mathbf{x}\right)&=\log\int p\left(\mathbf{x}_{0:T}\right)d\mathbf{x}_{1:T}\\&\geq\mathrm{E}_{q\left(\mathbf{x}_{1:T}|\mathbf{x}_0\right)}\left[\log\frac{p\left(\mathbf{x}_{0:T}\right)}{q\left(\mathbf{x}_{1:T}|\mathbf{x}_0\right)}\right]\\&=\underbrace{\mathrm{E}_{q\left(\mathbf{x}_1|\mathbf{x}_0\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}_0|\mathbf{x}_1\right)\right]}_{\text{reconstruction term}}-\underbrace{\mathrm{E}_{q\left(\mathbf{x}_{T-1}|\mathbf{x}_0\right)}\left[\mathcal{D}_{\mathrm{KL}}\left(q\left(\mathbf{x}_T|\mathbf{x}_{T-1}\right)||p\left(\mathbf{x}_T\right)\right)\right]}_{\text{prior matching term}}\\&-\sum_{t=1}^{T-1}\underbrace{\mathrm{E}_{q\left(\mathbf{x}_{t-1},\mathbf{x}_{t+1}|\mathbf{x}_0\right)}\left[\mathcal{D}_{\mathrm{KL}}\left(q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)||p_{\boldsymbol{\theta}}\left(\mathbf{x}_t|\mathbf{x}_{t+1}\right)\right)\right]}_{\text{consistency term}}\end{aligned}$$

DROPOUT

M is a binary matrix drawn from a Bernoulli matrix with probability (1-p). Dropout in the training .

$$\widetilde{\mathbf{H}}=\mathbf{H}\odot\mathbf{M},$$

MONTECARLO DROPOUT

Present Inside the inference.
The output is computed averaging the result with different dropout masks

$$\hat{y}=\mathbb{E}_{p(\mathbf{M})}\left[f(\mathbf{x};\mathbf{M})\right]\approx\frac{1}{N}\sum_{i=1}^{N}f(\mathbf{x};\mathbf{M}_i)$$

WASSERSTEIN GAN

We want a mass transportation of the distribution to the preferred one.
We want to calculate “Earth Mover Distance” minim amount of “Work” to this transportation

$$\max_{w\in\mathcal{W}}\mathrm{E}_{x\sim p(x)}\left\{D_w\left(x\right)\right\}-\mathrm{E}_{\tilde{x}\sim p(\tilde{x})}\left\{D_w\left(\tilde{x}\right)\right\}$$

$$\leq\sup_{\|f\|_L\leq1}\mathrm{E}_{x\sim p(x)}\left\{D\left(x\right)\right\}-\mathrm{E}_{\tilde{x}\sim p(\tilde{x})}\left\{D\left(\tilde{x}\right)\right\}=K\cdot\mathsf{W}\left(\cdot\right)$$

(4)

We replace the discriminator with the critic. Its goal is to output scores for real and fake data that allow the generator to minimize the real data distribution and the generated distribution

$$\mathsf{W}\left(p\left(x\right),p\left(\tilde{x}\right)\right)=\sup_{\|f\|_L\leq1}\mathrm{E}_{x\sim p(x)}\left\{f\left(x\right)\right\}-\mathrm{E}_{\tilde{x}\sim p(\tilde{x})}\left\{f\left(\tilde{x}\right)\right\}$$