Auto-Encoding Variational Bayes

Diederik P Kingma, Max Welling

June 18, 2018

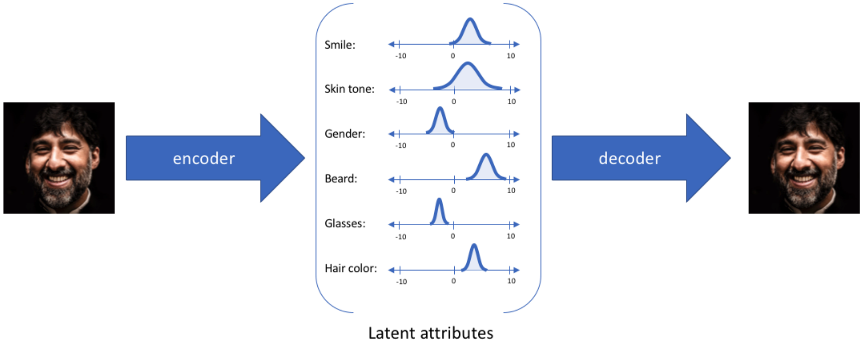
Outline

Motivation

- Deep learning using Autoencoder shows great success on feature extraction.
- Scaling variational inference to large data set.
- Approximating the intrtactable posterior can be used for multiply tasks:
  - Recognition
  - Denoising
  - Representation
  - Visualiation
  - Generative Model

Motivation - Intuition

How to move from our sample x_i to latent space z_i, and reconstruct $\tilde{x}_i$.



Problem

- Intractability: the case where the integral of the marginal likelihood $p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$ is intractable (we cannot evaluate or differentiate the marginal likelihood), where the true posterior density $p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}$ is intractable (so the EM algorithm cannot be used), and where the required integrals for any reasonable mean-field VB algorithm are also intractable.
- A large dataset: we have so much data that batch optimization is too costly; we would like to make parameter updates using small minibatches or even single datapoints. Sampling based solutions, e.g. Monte Carlo EM, would in general be too slow, since it involves a typically expensive sampling loop per datapoint.

Bayesian inference

- θ: Distribution parameters
- α: Hyper-parameters of the parameters distribution, e.g. $\theta \sim p(\theta|\alpha)$, our prior.
- $\mathbf{X}$: Samples
- $p(\mathbf{X}|a) \sim \int p(\mathbf{X}|\theta)p(\theta|\alpha)d\theta$: Marginal likelihood
- $p(\theta|\mathbf{X}, \alpha) \propto p(\mathbf{X}|\theta)p(\theta|\alpha)$: Posterior distribution.
- MAP -*Maximum a posteriori*: $\hat{\theta}_{MAP}(\mathbf{X}) = \underbrace{argmax}_{\theta} p(\mathbf{X}|\theta)p(\theta|\alpha)$
- Sample x is sampled by initially sampling θ from $\theta \sim p(\theta|\alpha)$, and then sampling x from $x \sim p(x|\theta)$

Bayesian inference - Intuition

Let $z \in \theta$ be an animal generator. with

$$z \sim p(\alpha) = \begin{cases} 0.3 & CatGenerator \\ 0.2 & DogGenerator \\ 0.5 & ParrotGenerator \end{cases}$$

Given our sample $x =$ , what is the chances that z is a cat generator?

$$p(z = CG|x, \alpha) = \frac{p(x|z = CG)p(z = CG|\alpha)}{\sum_{Gen \in \theta} p(x|z = Gen)p(z = Gen|\alpha)}$$

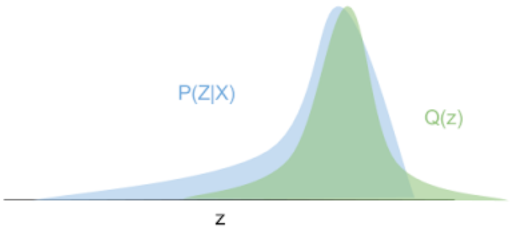
Bayesian inference - Problems

Very often directly 'solving' bayesian inference problem will require evaluating intractable integrals.
In order to overcome this there are two approaches:

- Sampling, Mostly methods of MCMC, such as Gibs Sampling, are used in order to find the optimal parameters.
- Variational methods, such as Mean Field Approximation.

Variational Lower Bound

Instead of evaluating intractable distibution P, we will find a simpler distribution Q, and use it instead of P where needed:



Variational Lower Bound 1

How can we choose our Q?

- Pick some tractable Q which can explain our data well.
- Optimize its parameters, using P and our given data.

For example, we could pick $Q \sim \mathcal{N}(\mu, \sigma^2)$ and optimize its parameters μ, σ.

Variational Lower Bound 2

Looking at the log probability of the observations **X**:

$$log_p(X) = log \int_Z p(X,Z) = log \int_Z p(X,Z)\frac{q(Z)}{q(Z)} = log(\mathbb{E}_q[\frac{p(X,Z)}{q(Z)}]) \geq$$

*Jensen's inequality

$$\mathbb{E}_q[log\frac{p(X,Z)}{q(Z)}] = \mathbb{E}_q[log(p(X,Z)) - log(q(Z))] = \mathbb{E}_q[log(p(X,Z))] + H(Z)$$

$$\mathcal{L} = \mathbb{E}_q[log(p(X,Z))] + H(Z))$$

$\mathcal{L}$ is the **Variational Lower Bound**.

Kullback–Leibler divergence

KL Divergence is a measure of how one probability distribution diverges from another.

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

$$D_{KL}(P||Q) = \int P(x) log \frac{P(x)}{Q(x)} dx$$

when $P(x) = Q(x)$ for all $x \in X \rightarrow D_{KL}(P||Q) = 0$
$D_{KL}(P||Q)$ is always non-negative.

Variational Lower Bound 3

$$D_{KL}(q(Z)||p(Z|X)) = \int q(Z) log \frac{q(Z)}{P(Z|X))} dZ = -\int q(Z) log \frac{P(Z|X)}{Q(Z))} dZ =$$

$$= -(\int q(Z) log \frac{p(X,Z)}{q(Z)} dZ - \int q(Z) log(p(X) dZ) =$$

$$= -\int q(Z) log \frac{p(X,Z)}{q(Z)} dZ + log(p(X)) \underbrace{\int q(Z)}_{=1} = -\mathcal{L} + log(p(X) \Rightarrow$$

$$log(p(X)) = \mathcal{L} + D_{KL}(q(Z)||p(Z|X))$$

As KL Divergence is always non negative, we get that $log(p(X)) \geq \mathcal{L}$, and either maximizing $\mathcal{L}$ or minimizing $D_{KL}(q(Z)||p(Z|X))$ will optimize q.

Stochastic search variational Bayes

Let ψ be distibution q variational parameters. We will want to optimize $\mathcal{L}$ w.r.t both $\mathbb{Z}$ (generative parameters) and ψ.
Seperate $\mathcal{L}$ into $\mathbb{E}f$ and h, where $h(X, \psi)$ contains everything in $\mathcal{L}$ except for $\mathbb{E}f$.

$$\nabla_{\psi}\mathcal{L} = \nabla_{\psi}\mathbb{E}_q[f(z)] + \nabla_{\psi}h(X, \psi)$$

$\nabla_{\psi}h(X, \psi)$ is tractable, while for intractable $\nabla_{\psi}\mathbb{E}_q[f(z)]$ we will approximate it using Monte Carlo integration:

$$\nabla_{\psi}\mathbb{E}_q[f(z)] \approx \frac{1}{S}\sum_{s=1}^{S} f(z^{(s)})\nabla_{\psi} ln\ q(z^{(s)}|\psi)$$

where $z^{(s)} \sim_{iid} q(z|\psi)$ for $s = 1..., S$. denote the above approximation as ζ. and we recieve the gardient step:

$$\psi^{t+1} = \psi^{t} + \rho_t \nabla_{\psi} h(X, \psi^{(t)}) + \rho_t \zeta_t$$

SGVB Estimator

The above method exibits very high variance, may converge very slow, and is impractical for our purpose.
We will reparametrize the random variable $\tilde{z} \sim q_\psi(z|x)$ using a diffrentiable transformation $g_\psi(\epsilon, x)$ of an auxilary noise variable ϵ:

$$\tilde{z} = g_\psi(\epsilon, x) \;\; \textit{with} \;\; \epsilon \sim p(\epsilon)$$

We can now form Monte Carlo estimation of expection of function $f(z)$ w.r.t $q_\psi(z|x)$:

$$\mathbb{E}_{q_\psi(z|x^{(i)})}[f(z)] = \mathbb{E}_{p(\epsilon)}[f(g_\psi(\epsilon, x^{(i)})] \simeq \frac{1}{S}\sum_{s=1}^{S} f(g_\psi(\epsilon^{(s)}, x^{(i)}))$$

$$\textit{where} \;\; \epsilon^{(s)} \sim p(\epsilon)$$

Estimator A

Applying the above technique to the Variational lower bound $\mathcal{L}$, we yield our generic SGVB estimator $\tilde{\mathcal{L}}^A(\theta, \psi; x^{(i)}) \simeq \mathcal{L}(\theta, \psi; x^{(i)})$:

$$\tilde{\mathcal{L}}^A(\theta, \psi; x^{(i)}) = \frac{1}{S}\sum_{s=1}^{S} log \;\; p_\theta(x^{(i)}, z^{(i,s)}) - log \;\; q_\psi(z^{(i,s)}|x^{(i)})$$

$$where \;\; z^{(i,l)} = g_\psi(\epsilon^{(i,l)}, x^{(i)}) \quad and \quad \epsilon^{(l)} \sim p(\epsilon)$$

Estimator B

Alternitivly, we can use the above technique on the *KL* divergence, the recieve another version of our SGVB estimator:

$$\tilde{\mathcal{L}}^B(\theta, \psi; x^{(i)}) = -D_{KL}(q_\psi(z|x^{(i)})||p_\theta(z)) + \frac{1}{S}\sum_{s=1}^{S} log \;\; p_\theta(x^{(i)}|z^{(i,s)}))$$

$$where \;\; z^{(i,l)} = g_\psi(\epsilon^{(i,l)}, x^{(i)}) \;\;\; and \;\;\; \epsilon^{(l)} \sim p(\epsilon)$$

Reparameterization trick - intuition

By adding the noise, we reduce the variance, and pay for it with accuracy:

Auto Encoding Variational Bayes

Given multiple data points from data set $\mathbf{X}$ with N data points, we can construct an estimator of the marginal likelihood of the data set, based on mini-batches:

$$\mathcal{L}(\theta, \psi; x^{(i)}) \simeq \tilde{\mathcal{L}}^{M}(\theta, \psi; x^{M}) = \frac{N}{M} \sum_{i=1}^{M} \tilde{\mathcal{L}}(\theta, \psi; x^{(i)})$$

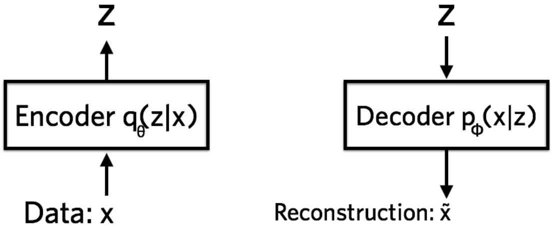
Both estimators A and B can be used.

Minibatch AEVB algorithm

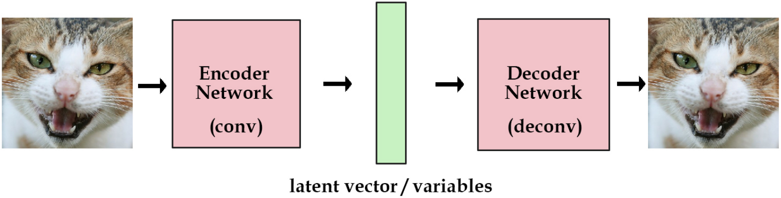
1. $\theta, \psi \leftarrow$ Initialize parameters
2. repeat
    1. $\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints (drawn from the full dataset)
    2. $\epsilon \leftarrow$ Random samples from noise distibution $p(\epsilon)$
    3. $g \leftarrow \nabla_{\psi,\theta}\tilde{\mathcal{L}}^M(\theta, \psi; \mathbf{X}^M, \epsilon)$ (Gardients of minibatch estimator)
    4. $\psi, \theta \leftarrow$ Update parameters using gardients g (SGD or Adagrad)
3. until convergence of parameters θ, ψ
4. Return θ, ψ

Variational Auto-Encoder

- Variational autoencoders (VAEs) were defined in 2013 by Kingma et al. (This article) and Rezende et al. (Google, simulationsly).
- A variational autoencoder consists of an encoder, a decoder, and a loss function:

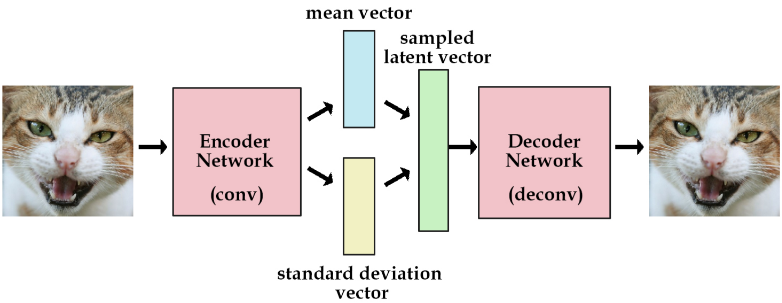


Variational Auto-Encoder



Auto-Encoder

Variational Auto-Encoder



Variational Auto-Encoder

Variational Auto-Encoder

- The encoder is a neural network. Its input is a datapoint x, its output is a latent (hidden) representation z, and it has weights and biases ψ:
    - The encoder 'encodes' the data which is N-dimensional into a latent (hidden) representation space z which is much less than N dimensions.
    - The lower-dimensional space is stochastic: the encoder outputs parameters to $q_\psi(z|x)$.

Variational Auto-Encoder

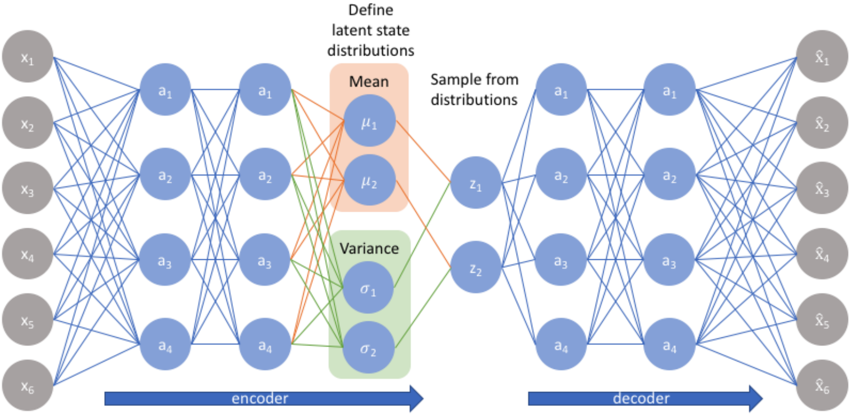
- The decoder is another neural network. Its input is the representation z , it outputs the parameters to the probability distribution of the data, and has weights and biases θ:
  - The decoder outputs parameters to $p_\theta(x|z)$
  - The decoder 'decodes' the real-valued numbers in z into N real-valued numbers.
- The decoder loosing information. Information is lost because it goes from a smaller to a larger dimensionality:
  - How much information is lost?
  - It measured using the reconstruction log-likelihood $log(p_\theta(x|z))$.

Experiments

- Using a neural network for our encoder $q_\psi(z|s)$ (the approximation of $p\theta(x,z)$).
- Optimizing ψ, θ by the AEVB algorithm.
- Assume $p_\theta(x|z)$ is a MV Gaussian/Bernouli, computed from z with a MLP $\Rightarrow p_\theta(z|x)$ is intractable.
- Let $q_\psi(z|x)$ be a Gaussian with diagonal covariance: $\log q_\psi(z|x^{(I)}) = \log \mathcal{N}(z; \mu^{(i)}, \sigma^{2^{(i)}} I)$
- $\mu^{(i)}, \sigma^{2^{(i)}} I$ are outputs of the encoding MLP, i.e. nonlinear functions of $x^{(i)}$
- Maximizing the objective function:

$$\mathcal{L}(\theta, \psi; x^{(i)}) \simeq \frac{1}{2} \sum_{j=1}^{J} (1 + log((\sigma_j^{(i)})^2) - {\mu_j^{(i)}}^2 - {\sigma_j^{(i)}}^2) + \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(x^{(i)}|z^{(i,l)})$$

Experiments



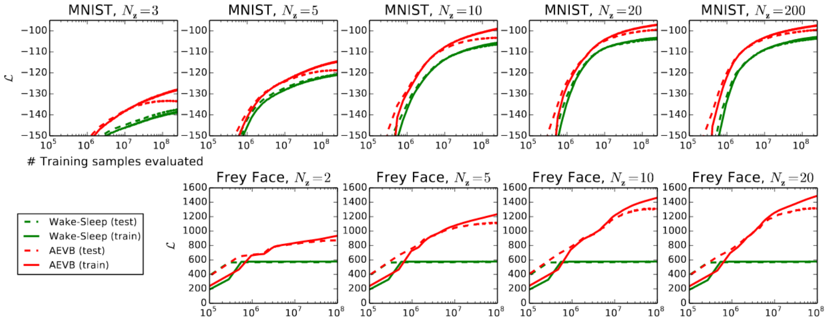
Experiments

- Trained generative models of images from the MNIST and Frey Face datasets and compared learning algorithms in terms of the variational lower bound, and the estimated marginal likelihood.
- The encoder and decoder have an equal number of hidden units.
- For the Frey Face data used decoder with Gaussian outputs, identical to the encoder, except that the means were constrained to the interval (0,1) using a sigmoidal activation function at the decoder output.
- The hidden units are the hidden layer of the neural networks of the encoder and decoder.
- Compared performance of AEVB to the wake-sleep algorithm

Experiments

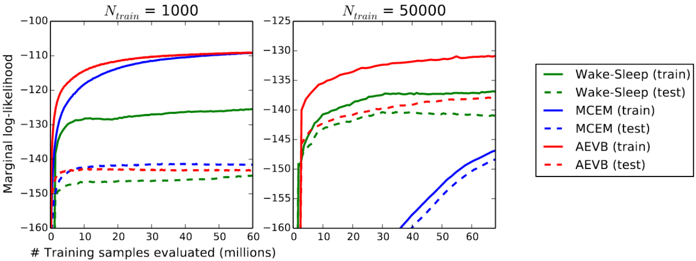
- Likelihood lower bound:
  - Trained generative models (decoders) and corresponding encoders having 500 hidden units in case of MNIST , and 200 hidden units in case of the Frey Face data.
- Marginal likelihood:
  - For very low-dimensional latent space it is possible to estimate the marginal likelihood of the learned generative models using an MCMC estimator. For the encoder and decoder used neural networks, this time with 100 hidden units, and 3 latent variables.

Experiments



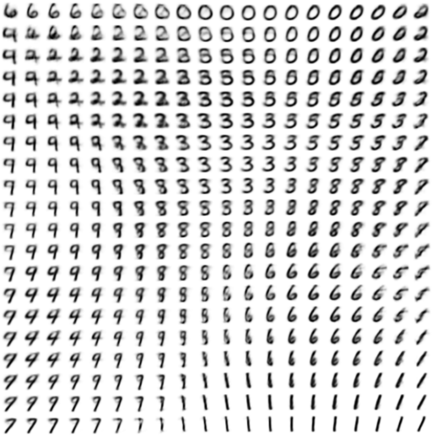
Comprasion of AEVB method to the wake-sleep algorithm, in terms of optimizing the lower bound, for different dimensionality of latent space. Vertical axis is the estimated avg Variational Lower Bound per data point, Horizontal axis is the amount of training points evaluated

Experiments



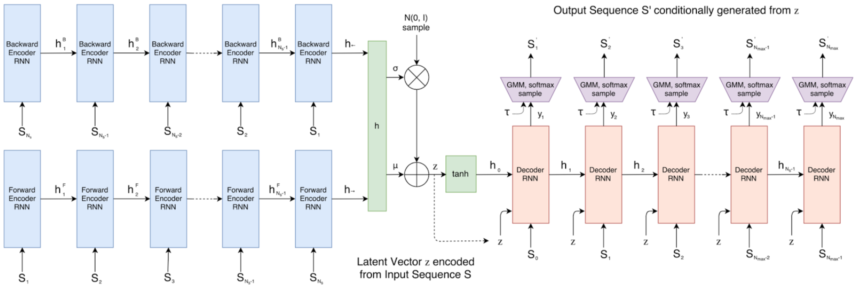
Comparison of AEVB to the wake-sleep algorithm and Monte Carlo EM, in terms of the estimated marginal likelihood, for a different number of training points. Monte Carlo EM is not an on-line algorithm, and (unlike AEVB and the wake-sleep method) can't be applied efficiently for the full MNIST dataset.

Experiments



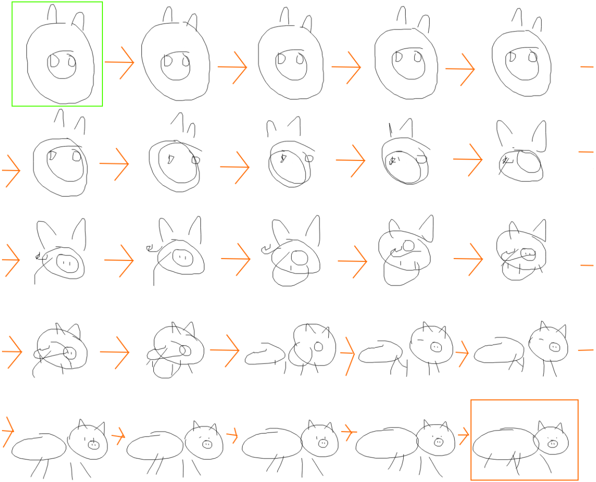
Learned MNIST manifold, on a 2d latent space.

SketchRNN



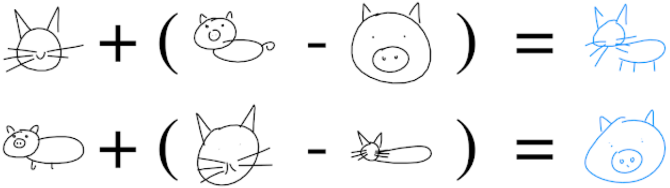
Schematic of sketch-rnn.

SketchRNN



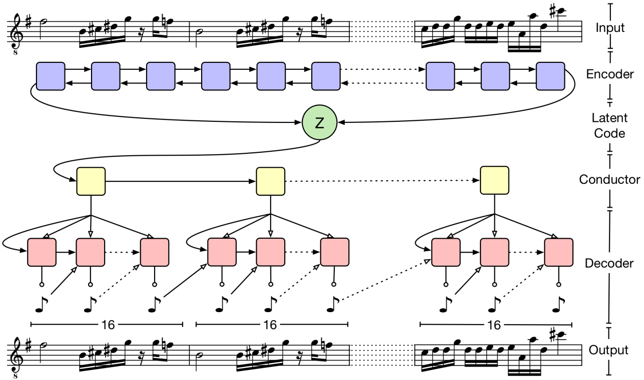
Latent space interpolations generated from a model trained on pig sketches.

SketchRNN



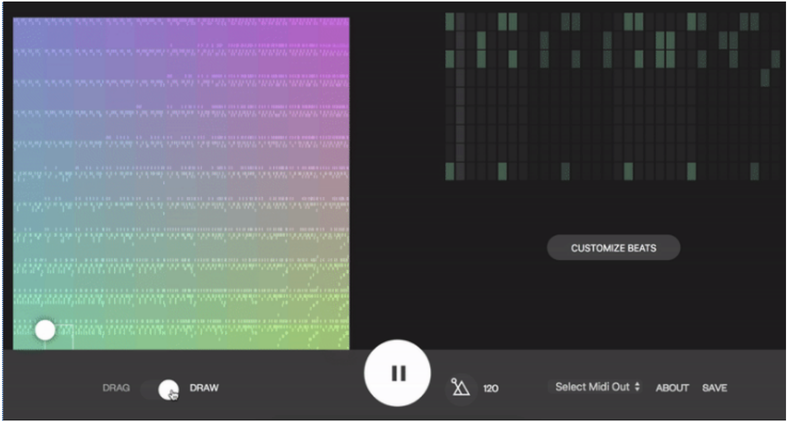
Learned relationships between abstract concepts, explored using latent vector arithmetic.

MusicVAE



Schematic of MusicVAE

MusicVAE



Beat blender

References I

- Musicvae.
  https://magenta.tensorflow.org/music-vae.
- Teaching machines to draw.
  https://ai.googleblog.com/2017/04/teaching-machines-to-draw.html.
- Variational autoencoders.
  https://www.jeremyjordan.me/variational-autoencoders/.
- What is variational auto encoder tutorial.
  https://jaan.io/what-is-variational-autoencoder-vae-tutorial/.
- B. J. F. Geoffrey E Hinton, Peter Dayan and R. M. Neal.
  The wakesleep algorithm for unsupervised neural networks.
  SCIENCE, page 1158–1158, 1995.

References II

M. I. J. John Paisley, David M. Blei.
Variational bayesian inference with stochastic search.
ICML, 2012.

C. W. Matthew D Hoffman, David M Blei and J. Paisley.
Stochastic variational inference.
Journal of Machine Learning Research, 14(1):1303–1347.

D. P.Kingma and M. Welling.
Auto-encoding variational bayes.

B. J. P. Simon Duane, Anthony D Kennedy and D. Roweth.
Hybrid monte carlo.
Physics letters B, 195(2):216–222, 1987.