

変分オートエンコーダ

variational autoencoder

正田 備也

masada@rikkyo.ac.jp

Contents

オートエンコーダ

変分オートエンコーダ

変分オートエンコーダの実装

オートエンコーダ (AE; autoencoder)

- ▶ dimensionality reduction (次元圧縮、次元削減) の手法の一つ
 - ▶ 高次元ベクトルを低次元の空間へと写す手法
- ▶ 元の空間の次元を d 、写す先の空間の次元を k とする
- ▶ 元の d 次元ベクトルを x_i 、写した後の k 次元ベクトルを z_i と書くことにする
- ▶ AE では、 z_i の良し悪しを問題にする
- ▶ どういう z_i なら良いと考えられているのか？

オートエンコーダによる次元圧縮

- ▶ AE は、データ集合に属する個々のデータ点 x_i について、それ自身をより良く再現できるような低次元表現 z_i を求める
 - ▶ AE での低次元表現をコード (code) と呼ぶ
- ▶ オートエンコーダは2つのニューラルネット (NN) から成る
 1. エンコーダ: 個々のデータ点 x_i を入力とし、コード z_i を出力する NN
 2. デコーダ: コード z_i を入力とし、 x_i と同じ次元のベクトル \hat{x}_i を出力する NN
- ▶ 2つの NN は、 \hat{x}_i ができるだけ x_i に近くなるように訓練する

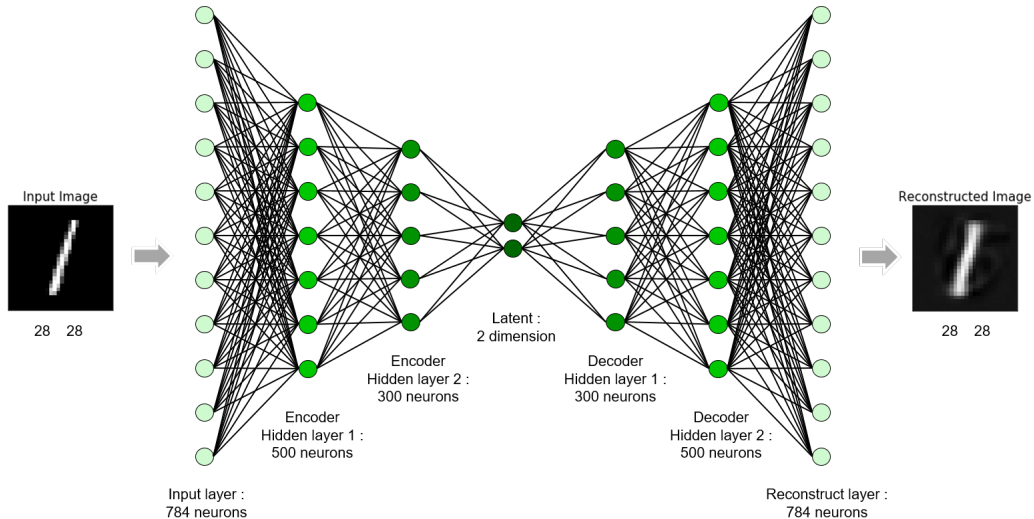


Figure: <https://encodebox.medium.com/auto-encoder-in-biology-9264da118b83>

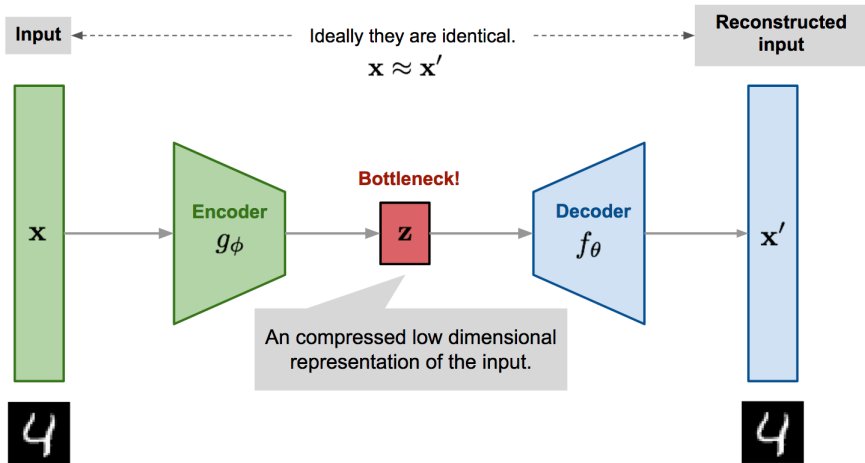


Figure: <https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html>

Contents

オートエンコーダ

変分オートエンコーダ

変分オートエンコーダの実装

変分オートエンコーダ (VAE; variational autoencoder)

- ▶ 一見、AEと似ている・・・が、かなり違う
- ▶ x_i 自身を再現するための低次元表現として、AEで言えばエンコーダにあたるNNの出力を、そのまま使うことはない
- ▶ なぜなら、NNの出力は、それがそのままコードであるわけではなく、変分事後分布のパラメータだから
- ▶ この変分事後分布から得たサンプルが、コード z_i となる
- ▶ さらに、デコーダの出力は、元のデータ点 x_i と直接比較できるようなものであるとは限らない
 - ▶ 元のデータ点 x_i を生成する分布のパラメータを出力する

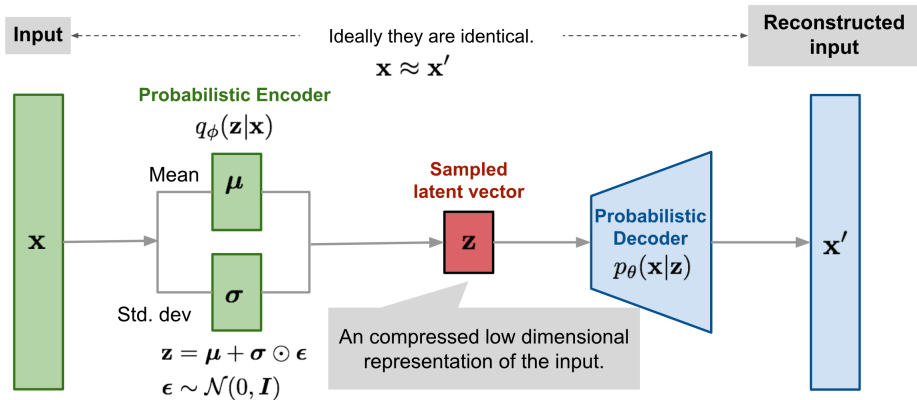


Figure: <https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html>

変分ベイズ法作为一种としてのVAE

- ▶ VAE を、AE 的一种として理解するには、無理がある
- ▶ 変分ベイズ法から理解するのが、吉
- ▶ そうでないと、VAE が、なぜあれでいいのか、分からない
 - ▶ エンコーダからデコーダへ移るときに、なぜ、サンプルを得るというステップが挟まっているのか？
 - ▶ エンコーダについて、なぜ、KL 情報量による正則化が考えられているのか？
 - ▶ これらが、AE からの延長で VAE を考えていると、分からない

変分ベイズ法における ELBO の書き換え

$$\begin{aligned}\ln p(\mathcal{X}) &\geq \int q(\Theta) \ln \frac{p(\Theta)p(\mathcal{X}|\Theta)}{q(\Theta)} d\Theta \\ &= \int q(\Theta) \ln \prod_{i=1}^N p(\mathbf{x}_i|\Theta) d\Theta - \int q(\Theta) \ln \frac{q(\Theta)}{p(\Theta)} d\Theta \\ &= \sum_{i=1}^N \int q(\Theta) \ln p(\mathbf{x}_i|\Theta) d\Theta - D_{\text{KL}}(q(\Theta) \parallel p(\Theta)) \quad (1)\end{aligned}$$

- ▶ KL 情報量の項は $q(\Theta)$ を $p(\Theta)$ に近づけるはたらきをする
 - ▶ これは一種の正則化 (regularization)

変分オートエンコーダ提案の背景

- ▶ PyTorch や TensorFlow などの深層学習フレームワークが普及
- ▶ 複雑な関数でも、簡単に勾配を計算できるようになった
 - ▶ 自動微分が身近なものになったため。
- ▶ Adam などの優れた最適化アルゴリズムも提案された
- ▶ ならば・・・
- ▶ ELBO の最大化も勾配法で解いてしまえばいいのでは？
 - ▶ 手計算でパラメータ更新の式を求めたりするのではなく。

観測データのモデルのパラメータには2種類ある

1. 個々のデータ点に対して別々に用意されているパラメータ

- ▶ 例：潜在的ディリクレ配分法での各文書のトピック確率

$$\theta_i = (\theta_{i,1}, \dots, \theta_{i,K}) \text{ for } i = 1, \dots, N$$

2. データ集合全体に対して用意されているパラメータ

- ▶ 例：潜在的ディリクレ配分法での各トピックの単語確率

$$\phi_k = (\phi_{k,1}, \dots, \phi_{k,W}) \text{ for } k = 1, \dots, K$$

- ▶ どちらの種類のパラメータにも事前分布を導入できる
- ▶ VAE では、個々のデータ点に対して別々に用意されているパラメータのほうに、事前分布を導入する

VAEにおける notation

- ▶ 確率モデリングの世界では、 z は、離散値をとる確率変数を表すために使うことが多い
 - ▶ 例：各データ点がどのクラスに属するかを表す潜在確率変数 z_i
- ▶ ところが、VAE の話をするときには、 z を、観測データを生成するモデルのパラメータを表すために使う
 - ▶ 気分としては、 i 番目の観測データ x_i の生成に関する確率分布のパラメータは θ_i と書きたいが、なぜか z_i と書く
 - ▶ たぶん AE の世界で潜在表現を z と書いていたからかも。
 - ▶ “We assume that the data are generated by some random process, involving an unobserved continuous random variable z .” [Kingma+arXiv:1312.6114v10]

VAEにおけるELBO

- ▶ 観測データのモデルのパラメータのうち、各データ点 x_i に対して別々に用意されたパラメータ z_i だけを考慮するなら

$$\begin{aligned}\ln p(\mathcal{X}) &\geq \sum_{i=1}^N \int q(\boldsymbol{\Theta}) \ln p(\mathbf{x}_i | \boldsymbol{\Theta}) d\boldsymbol{\Theta} - D_{\text{KL}}(q(\boldsymbol{\Theta}) \parallel p(\boldsymbol{\Theta})) \\ &= \sum_{i=1}^N \int q(\mathbf{z}_i) \ln p(\mathbf{x}_i | \mathbf{z}_i) d\mathbf{z}_i - \sum_{i=1}^N D_{\text{KL}}(q(\mathbf{z}_i) \parallel p(\mathbf{z}_i))\end{aligned}\tag{2}$$

- ▶ 他のパラメータは自由パラメータのままでもいいし、事前分布を使ってベイズ化されていてもいいが、VAEには関係しない。

全データに関わるパラメータがある場合

- ▶ データ集合全体に関わるモデル・パラメータを、まとめて Φ と書くことにする。
- ▶ Φ について事前分布を導入しないなら、ELBO を Φ の関数だと思って最大化することで Φ の値を推定すればよい。
- ▶ そうでなければ、ELBO は以下ようになる。(ただし \mathcal{Z} は $\{z_1, \dots, z_N\}$ を意味する。)

$$\begin{aligned}\ln p(\mathcal{X}) &= \ln \int p(\mathcal{Z})p(\Phi)p(\mathcal{X}|\mathcal{Z}, \Phi)d\mathcal{Z}d\Phi \\ &\geq \int q(\mathcal{Z})q(\Phi) \ln \frac{p(\mathcal{Z})p(\Phi)p(\mathcal{X}|\mathcal{Z}, \Phi)}{q(\mathcal{Z})q(\Phi)}d\mathcal{Z}d\Phi \\ &= \int q(\mathcal{Z})q(\Phi) \ln \prod_{i=1}^N p(\mathbf{x}_i|z_i, \Phi)d\mathcal{Z}d\Phi - \int q(\mathcal{Z}) \ln \frac{q(\mathcal{Z})}{p(\mathcal{Z})}d\mathcal{Z} - \int q(\Phi) \ln \frac{q(\Phi)}{p(\Phi)}d\Phi \\ &= \sum_{i=1}^N \int q(z_i)q(\Phi) \ln p(\mathbf{x}_i|z_i, \Phi)d\Phi dz_i - \sum_{i=1}^N D_{\text{KL}}(q(z_i) \parallel p(z_i)) - D_{\text{KL}}(q(\Phi) \parallel p(\Phi)) \quad (3)\end{aligned}$$

Contents

オートエンコーダ

変分オートエンコーダ

変分オートエンコーダの実装

VAEの実装の目標

- ▶ 目標は ELBO の最大化の計算を実装すること
 - ▶ これによって、事後分布の近似としての変分事後分布が得られる
- ▶ そこで、ELBO の計算全体の計算グラフを作り・・・
- ▶ ELBO 最大化の問題を、通常のニューラルネットの学習と同じように行う（ELBO にマイナスを付けて最小化する）
- ▶ 以下、このようなことを効率的に実現するために、実際には多くの場合どのように VAE を実装するか、説明する

VAEにおける変分事後分布 $q(z_i)$

- ▶ $q(z_i)$ は、観測データ x_i をモデリングする確率分布のパラメータが従う分布である
 - ▶ この確率分布の密度関数を使って、尤度 $p(x_i|z_i)$ が表される
- ▶ VAE の変分事後分布 $q(z_i)$ としては、正規分布を使う
- ▶ しかも、共分散行列が対角行列であることを仮定する
- ▶ よって、 $q(z_i)$ のパラメータは、平均 μ_i と標準偏差 σ_i
 - ▶ 平均パラメータ $\mu_i = (\mu_{i,1}, \dots, \mu_{i,K})$
 - ▶ 標準偏差パラメータ $\sigma_i = (\sigma_{i,1}, \dots, \sigma_{i,K})$
- ▶ パラメータの個数は $2K$ になる

VAEにおけるELBOの前向き計算

- ▶ VAEにおけるELBOは

$$\mathcal{L} = \sum_{i=1}^N \int q(\mathbf{z}_i) \ln p(\mathbf{x}_i | \mathbf{z}_i) d\mathbf{z}_i - \sum_{i=1}^N D_{\text{KL}}(q(\mathbf{z}_i) \parallel p(\mathbf{z}_i))$$

- ▶ このELBOの計算グラフを作り、backpropagationすれば、ELBOに現われる様々なパラメータを更新していけるが…
 - ▶ $q(\mathbf{z}_i)$ についての積分はどう計算すればいい？
 - ▶ KL 情報量の項はどう計算すればいい？

積分のモンテカルロ近似

- ▶ $\int q(\mathbf{z}_i) \ln p(\mathbf{x}_i | \mathbf{z}_i) d\mathbf{z}_i$ はモンテカルロ近似する
- ▶ つまり、 $q(\mathbf{z}_i)$ からサンプル $\{\mathbf{z}_i^{(1)}, \dots, \mathbf{z}_i^{(S)}\}$ を生成し、以下のように近似する

$$\int q(\mathbf{z}_i) \ln p(\mathbf{x}_i | \mathbf{z}_i) d\mathbf{z}_i \approx \frac{1}{S} \sum_{s=1}^S \ln p(\mathbf{x}_i | \mathbf{z}_i^{(s)}) \quad (4)$$

- ▶ 通常、 $S = 1$ と設定する
 - ▶ この設定でも構わないのは、結構、驚くべきこと。

VAEのKL情報量

- ▶ いま、変分事後分布 $q(z_i)$ として、共分散行列が対角行列である正規分布を使っている
- ▶ さらに、事前分布 $p(z_i)$ として、成分ごとに標準正規分布を使うとすると…
- ▶ ELBO に現れている KL 情報量の項 $-D_{\text{KL}}(q(z_i) \parallel p(z_i))$ は、以下のように解析的に計算できてしまう

$$-D_{\text{KL}}(q(z_i) \parallel p(z_i)) = \frac{1}{2} \sum_{k=1}^K (1 + \ln((\sigma_{i,k})^2) - (\mu_{i,k})^2 - (\sigma_{i,k})^2)$$

- ▶ 計算グラフの一部として上の式の計算が入ってくる

VAEにおけるELBOの前向き計算（続）

- ▶ ELBO は、いまや以下のように計算される
 - ▶ 下式の $z_i^{(1)}$ の (1) は、サンプルを 1 個だけ取っている、という意味。

$$\mathcal{L} = \sum_{i=1}^N \ln p(\mathbf{x}_i | \mathbf{z}_i^{(1)}) + \frac{1}{2} \sum_{k=1}^K (1 + \ln((\sigma_{i,k})^2) - (\mu_{i,k})^2 - (\sigma_{i,k})^2)$$

- ▶ 変分事後分布 $q(\mathbf{z}_i)$ のパラメータ μ_i と σ_i はどう準備する？
 - ▶ エンコーダによって準備する
- ▶ 尤度 $p(\mathbf{x}_i | \mathbf{z}_i^{(1)})$ をどう表現する？
 - ▶ デコーダによって表現する
- ▶ 話がAEっぽくなってきた

VAEのエンコーダ

- ▶ VAEでは μ_i と σ_i をNNの出力として得る
- ▶ このNNを、VAEではエンコーダと呼ぶ
- ▶ VAEのエンコーダは x_i を入力とする
- ▶ VAEのエンコーダは μ_i と σ_i とを出力する
 - ▶ 観測データのモデルのパラメータ z_i を出力するのではない！
 - ▶ 通常は σ_i ではなく $\ln \sigma_i$ を出力するように実装する
- ▶ x_i に対応する潜在表現 z_i は、正規分布 $\mathcal{N}(\mu_i, \sigma_i^2)$ からランダムにサンプルを生成することで得られる
 - ▶ このサンプリングが、モンテカルロ近似のためのサンプリング

VAEのデコーダ

- ▶ VAE のデコーダは x_i に対応するコード z_i を入力とする
 - ▶ エンコーダの出力をそのまま入力とするのではない
- ▶ VAE のデコーダは、確率分布のパラメータを出力する
- ▶ そして、デコーダの出力をパラメータとする確率分布を使って x_i の尤度 $p(x_i|z_i)$ を求める

例 デコーダの出力を平均パラメータ、単位行列（の定数倍）を共分散行列とする正規分布を使って、 x_i の尤度を求め、それを ELBO 最大化を通して、最大化する

- ▶ これは、デコーダの出力 \hat{x}_i を、データ点 x_i に、ユークリッド距離 $\|\hat{x}_i - x_i\|$ の意味で近づけることと、全く同じことになる

VAEにおけるELBOの前向き計算（続々）

- ▶ エンコーダが表す関数を $\text{Enc}(\mathbf{x}_i)$ 、デコーダが表す関数を $\text{Dec}(\mathbf{z}_i)$ と書くことにする
- ▶ VAE における ELBO の前向き計算は、以下の通り
 1. $\text{Enc}(\mathbf{x}_i)$ を計算して $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$ を得る
 2. $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$ を使って \mathbf{z}_i をランダムに生成
 3. $\text{Dec}(\mathbf{z}_i)$ を計算して尤度 $p(\mathbf{x}_i|\mathbf{z}_i)$ のパラメータを得る
 4. 尤度 $p(\mathbf{x}_i|\mathbf{z}_i)$ と KL 項を計算
- ▶ しかし、 \mathbf{z}_i を単なる数値として生成すると、BP が \mathbf{z}_i を越えてエンコーダへと遡れない！

reparametrization trick

- ▶ VAE のエンコーダが出力した (μ_i, σ_i) によって、そこからサンプルを生成すべき正規分布は確定する
- ▶ しかし、サンプル z_i を単なる数値として生成してしまうと、BP がデコーダの入りで止まり、エンコーダへ遡れない！
- ▶ そこで・・・
 1. $\mathcal{N}(0, \mathbf{I}_k)$ から単なる数値としてサンプル ϵ_i を生成し…
 2. その ϵ_i を $\mu_i + \sigma_i \odot \epsilon_i$ という計算によって、本当に欲しかった $q(z_i)$ からのサンプルへ変換する
 - ▶ $\mu_i + \sigma_i \odot \epsilon_i$ という計算は、BP を行う計算グラフの一部になる

VAEにおけるELBOの前向き計算（完）

- ▶ エンコーダが表す関数を $\text{Enc}(\mathbf{x}_i)$ 、デコーダが表す関数を $\text{Dec}(\mathbf{z}_i)$ と書くことにする
- ▶ VAE における ELBO の前向き計算は、以下の通り
 1. $\text{Enc}(\mathbf{x}_i)$ を計算して $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$ を得る
 2. $\boldsymbol{\epsilon}_i$ を $\mathcal{N}(0, \mathbf{I}_k)$ から生成
 3. $\boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}_i$ によって \mathbf{z}_i を得る
 4. $\text{Dec}(\mathbf{z}_i)$ を計算して尤度 $p(\mathbf{x}_i|\mathbf{z}_i)$ のパラメータを得る
 5. 尤度 $p(\mathbf{x}_i|\mathbf{z}_i)$ と KL 項を計算

VAEにおける amortized inference

- ▶ μ_i と σ_i^2 は、 x_i を入力とする NN の出力として得た
- ▶ だが μ_i と σ_i^2 をデータ点 x_i ごとに単なる未知数として準備し、ELBO 最大化によって更新するので良かったのでは？
 - ▶ 変分ベイズ法では普通こうする
- ▶ 全てのデータ点に同じ一つの NN を共有させて、その出力としてデータ点ごとのパラメータ μ_i と σ_i を得る、という考え方は、変分ベイズ法には元々はなかった考え方！
- ▶ 一般に、変分事後分布のパラメータをデータ点 x_i の関数として表現してベイズ推論することを、amortized inference と呼ぶ（エンコーダがこの関数を表現している）