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 (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 $\boldsymbol{x}_i$、写した後の k 次元ベクトルを $\boldsymbol{z}_i$ と書くことにする
- AE では、$\boldsymbol{z}_i$ の良し悪しを問題にする
- どういう $\boldsymbol{z}_i$ なら良いと考えられているのか？

復習: 主成分分析 (PCA)

1. まず、データ集合を中心化（＝重心を原点へ移動）する
2. 原点を通るベクトルのうち、データ集合が最も大きく散らばっている方向を向いているものを選ぶ
   - これが第 1 主成分。
3. 次に、その方向に垂直な超平面へ、データ集合を押し潰す
4. すると空間の次元が一つ下がるので、次元が下がった空間の中で、2. と 3. を繰り返す
   - 第 2 主成分、第 3 主成分、・・・と続けて、第 k 主成分まで見つける

- 全体のばらつきを最もよく表す軸を k 本選ぶ、ということ
  - これらを座標軸として設定し直し、各データ点 $\boldsymbol{x}_i$ を k 次元空間の点 z_i として表現し直すのが PCA

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 $\boldsymbol{x}_i$ 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 $\boldsymbol{z}_i$ 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 $\boldsymbol{x}_i$ を入力とし、コード $\boldsymbol{z}_i$ を出力する NN
2. デコーダ: コード $\boldsymbol{z}_i$ を入力とし、$\boldsymbol{x}_i$ と同じ次元のベクトル $\hat{\boldsymbol{x}}_i$ を出力する NN

- 2 つの NN は、$\hat{\boldsymbol{x}}_i$ ができるだけ $\boldsymbol{x}_i$ に近くなるように訓練する

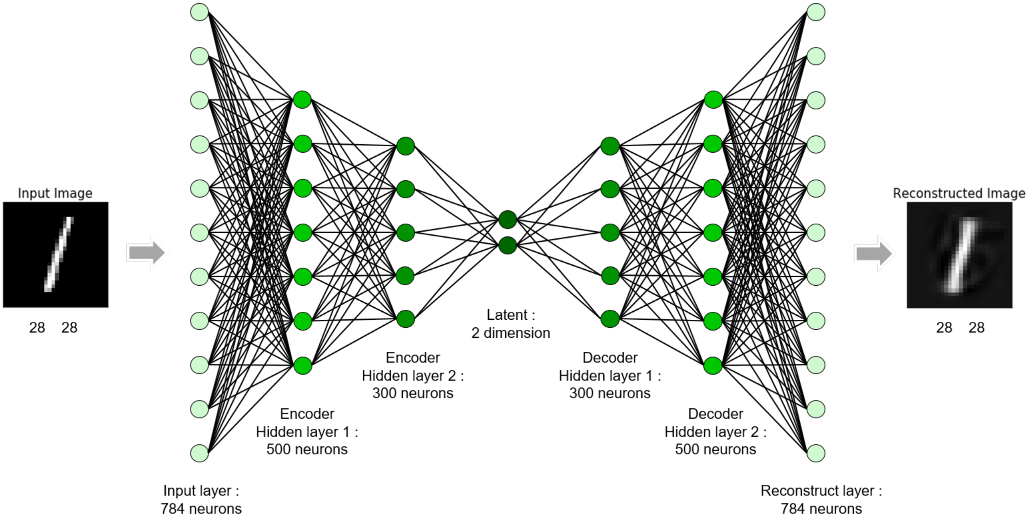

Figure: https://encodebox.medium.com/auto-encoder-in-biology-9264da118b83

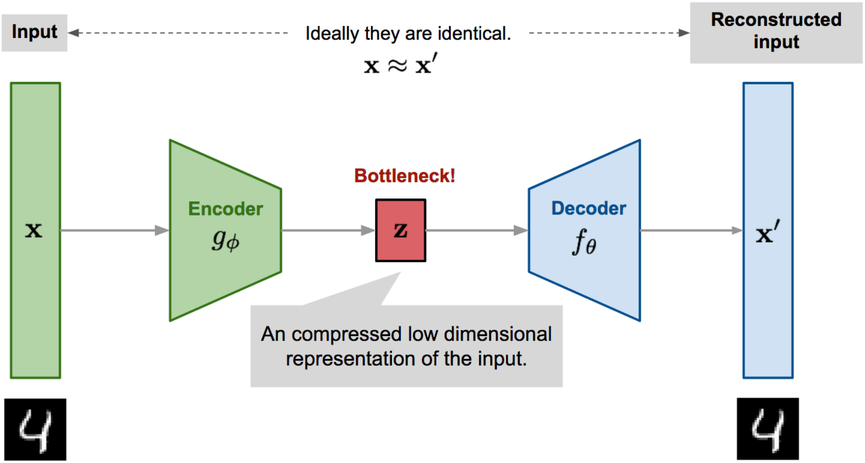

Figure: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

Contents

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
- $\boldsymbol{x}_i$ 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 $\boldsymbol{z}_i$ となる
- さらに、デコーダの出力は、元のデータ点 $\boldsymbol{x}_i$ 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 $\boldsymbol{x}_i$ を生成する分布のパラメータを出力する

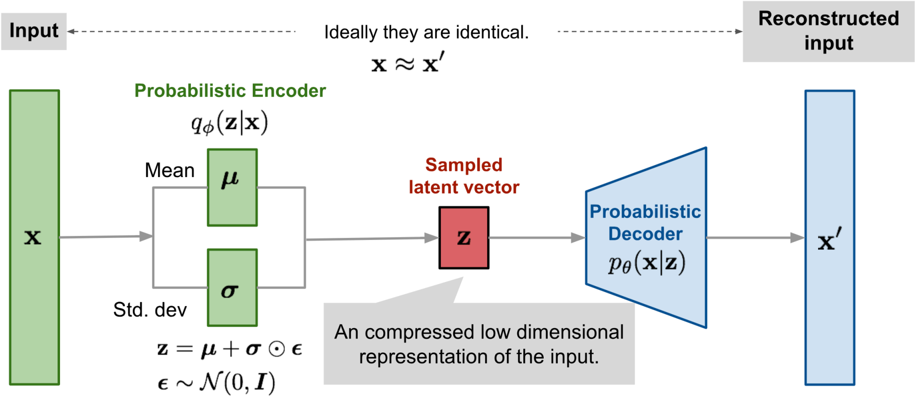

Figure: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

変分ベイズ法の一種としての VAE

- VAE を、AE の一種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

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

- KL情報量の項は $q(\boldsymbol{\Theta})$ を $p(\boldsymbol{\Theta})$ 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
     $\boldsymbol{\theta}_i = (\theta_{i,1}, \dots, \theta_{i,K})$ 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
     $\boldsymbol{\phi}_k = (\phi_{k,1}, \dots, \phi_{k,W})$ 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タ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、$\boldsymbol{z}$ 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 $\boldsymbol{x}_i$ の生成に関与する確率分布のパラメータは $\boldsymbol{\theta}_i$ と書きたいが、なぜか $\boldsymbol{z}_i$ と書く
    - たぶんAEの世界で潜在表現を $\boldsymbol{z}$ 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 $\mathbf{z}$." [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 $\boldsymbol{x}_i$ に対して別々に用意されたパラメータ $\boldsymbol{z}_i$ だけを考慮するなら

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

  - 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 $\mathbf{\Phi}$ と書くことにする。
- $\mathbf{\Phi}$ について事前分布を導入しないなら、ELBO を $\mathbf{\Phi}$ の関数だと思って最大化することで $\mathbf{\Phi}$ の値を推定すればよい。
- そうでなければ、ELBO は以下のようになる。（ただし $\mathcal{Z}$ は $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ を意味する。）

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

Contents

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 $q(\boldsymbol{z}_i)$

- $q(\boldsymbol{z}_i)$ は、観測データ $\boldsymbol{x}_i$ 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 $p(\boldsymbol{x}_i|\boldsymbol{z}_i)$ が表される
- VAEの変分事後分布 $q(\boldsymbol{z}_i)$ 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、$q(\boldsymbol{z}_i)$ のパラメータは、平均 $\boldsymbol{\mu}_i$ と標準偏差 $\boldsymbol{\sigma}_i$
  - 平均パラメータ $\boldsymbol{\mu}_i = (\mu_{i,1}, \ldots, \mu_{i,K})$
  - 標準偏差パラメータ $\boldsymbol{\sigma}_i = (\sigma_{i,1}, \ldots, \sigma_{i,K})$
- 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

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

- 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…
  - $q(\boldsymbol{z}_i)$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

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

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

- 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 $q(\boldsymbol{z}_i)$ 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 $p(\boldsymbol{z}_i)$ 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 $-D_{\mathsf{KL}}(q(\boldsymbol{z}_i) \parallel p(\boldsymbol{z}_i))$ は、以下のように解析的に計算できてしまう

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

  - 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 $\boldsymbol{z}_i^{(1)}$ の (1) は、サンプルを1個だけ取っている、という意味。

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

- 変分事後分布 $q(\boldsymbol{z}_i)$ のパラメータ $\boldsymbol{\mu}_i$ と $\boldsymbol{\sigma}_i$ はどう準備する？
  - エンコーダによって準備する
- 尤度 $p(\boldsymbol{x}_i|\boldsymbol{z}_i^{(1)})$ 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 $\boldsymbol{\mu}_i$ と $\boldsymbol{\sigma}_i$ 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 $\boldsymbol{x}_i$ を入力とする
- VAEのエンコーダは $\boldsymbol{\mu}_i$ と $\boldsymbol{\sigma}_i$ とを出力する
  - 観測データのモデルのパラメータ $\boldsymbol{z}_i$ を出力するのではない！
  - 通常は $\boldsymbol{\sigma}_i$ ではなく $\ln \boldsymbol{\sigma}_i$ を出力するように実装する
- $\boldsymbol{x}_i$ に対応する潜在表現 $\boldsymbol{z}_i$ は、正規分布 $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$ 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 $\boldsymbol{x}_i$ に対応するコード $\boldsymbol{z}_i$ 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 $\boldsymbol{x}_i$ の尤度 $p(\boldsymbol{x}_i|\boldsymbol{z}_i)$ 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、$\boldsymbol{x}_i$ 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 $\hat{\boldsymbol{x}}_i$ を、データ点 $\boldsymbol{x}_i$ に、ユークリッド距離 $\|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|$ 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 $\mathtt{Enc}(\boldsymbol{x}_i)$、デコーダが表す関数を $\mathtt{Dec}(\boldsymbol{z}_i)$ 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. $\mathtt{Enc}(\boldsymbol{x}_i)$ 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 $\boldsymbol{z}_i$ をランダムに生成
3. $\mathtt{Dec}(\boldsymbol{z}_i)$ を計算して尤度 $p(\boldsymbol{x}_i|\boldsymbol{z}_i)$ のパラメータを得る
4. 尤度 $p(\boldsymbol{x}_i|\boldsymbol{z}_i)$ とKL項を計算

- しかし、$\boldsymbol{z}_i$ を単なる数値として生成すると、BPが $\boldsymbol{z}_i$ 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 $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$ 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 $\boldsymbol{z}_i$ を単なる数値として生成してしまうと、BP がデコーダの入り口で止まり、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 $\boldsymbol{\epsilon}_i$ を生成し⋯
2. その $\boldsymbol{\epsilon}_i$ を $\boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}_i$ という計算によって、本当に欲しかった $q(\boldsymbol{z}_i)$ からのサンプルへ変換する
    - $\boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}_i$ 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 $\mathtt{Enc}(\boldsymbol{x}_i)$、デコーダが表す関数を $\mathtt{Dec}(\boldsymbol{z}_i)$ 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. $\mathtt{Enc}(\boldsymbol{x}_i)$ 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 $\boldsymbol{z}_i$ を得る
4. $\mathtt{Dec}(\boldsymbol{z}_i)$ を計算して尤度 $p(\boldsymbol{x}_i|\boldsymbol{z}_i)$ のパラメータを得る
5. 尤度 $p(\boldsymbol{x}_i|\boldsymbol{z}_i)$ とKL項を計算

VAEにおけるamortized inference

- $\boldsymbol{\mu}_i$ と $\boldsymbol{\sigma}_i^2$ は、$\boldsymbol{x}_i$ を入力とするNNの出力として得た
- だが $\boldsymbol{\mu}_i$ と $\boldsymbol{\sigma}_i^2$ をデータ点 $\boldsymbol{x}_i$ 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 $\boldsymbol{\mu}_i$ と $\boldsymbol{\sigma}_i$ 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 $\boldsymbol{x}_i$ 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）