



Auto-Encoding Variational Bayes

Development of Intellectual Property Creation Method
Using Language Generation Model on Patent Information

Shigeaki Onoda

Graduate School of Information Engineering, Toyama Prefectural University
t855005@st.pu-toyama.ac.jp

Friday., November 22, 2019,
Toyama Prefectural Univ.



1.1. 本研究の背景

背景

画像生成のタスクに置いて、ある少ないサンプルからそれに似てるが異なるサンプルを複数作るような処理が3D デザインやイラスト作成において重要な技術となっている。

現状の手法の問題

現状の手法では以下 3 つの欠点がある

- 1 データにおける構造についての強い仮説が必要
- 2 近似解に到達するために厳密な近似値を作る必要性
- 3 マルコフ連鎖モンテカルロ法のように高コストな計算





本発表の流れ

目的

本研究は上述の欠点を改善したモデルを紹介する. そのモデルは自己符号化器 (Autoencoder) を拡張して変分自己符号化器 **Variational Autoencoder** と呼ばれる

このモデルを利用することでデータの潜在的な特徴を得ることができる.

これを利用してデータの増強, 圧縮, 合成など様々な応用研究に活用できる.

Variational Autoencoder(以下 VAE) は Autoencoder(以下 AE) と地続きの手法であるためまず Autoencoder から説明する.



2.1. Autoencoder とは

Autoencoder の特徴

- 教師なし学習
- データの潜在的な特徴を獲得するためのニューラルネット (以下 NN)

エンコーダー・デコーダーからなるネットワークで入力サンプルをエンコーダーで圧縮、デコーダーで復元することによりデータを表現する特徴である潜在変数 z を獲得できる。

言い換えると入力データを再構成させることを学習させる

Autoencoder の応用例

近似サンプルの生成以外にも工学的に有用な利用例がある。

- ・ 潜在変数 z を他の機械学習における入力として利用
- ・ 重みを教師ありニューラルネットの初期重みで利用



2.1. Autoencoder とは

具体例

図 1 のように手書き数字の画像を入力した場合, 文字の特徴を z にマッピングしてそれをデコーダーを用いて復元することで元の画像に近い画像を生成する

学習時の振る舞い

入力データに近づくように NN の重みを更新することで, より入力データの特徴を反映した潜在変数を得ることができる。

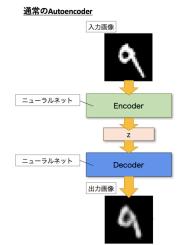


Figure: 1: AE の概念図



2.2. Variational Autoencoders

AE から VAE へ

AE では入力の特徴を謎のベクトル z に圧縮していた。 z は特徴を表現できるものの入力同士の繋がりを表現できていなかった
そこで z を $z \sim \mathcal{N}(0,1)$ の確率分布として定義したものが VAE となる

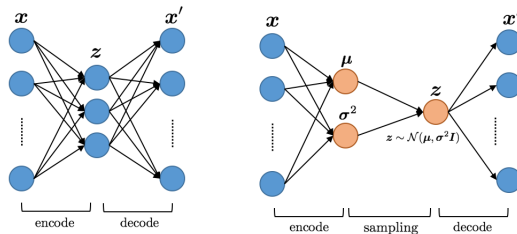


Figure: 2: 左 AE, 右 VAE のネットワーク



2.3. VAE の定式化

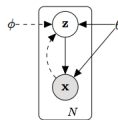


Figure: 3: VAE の変数間の関係

実線は生成モデル $p_{\theta}(z)p_{\theta}(x|z)$
破線は事後確率 $p_{\theta}(z|x)$ への変分近似モデル $q_{\phi}(z|x)$ を示す. 変分
パラメータ ϕ は重みパラメータ θ と同時に学習する
推定方法: 最尤推定及び MAP 推定

概要

θ^* が完全な圧縮・復元ができるときの真のパラメータとして目的としてはそれに近似する θ および潜在変数 z を推定することである. しかし, これをモンテカルロ法等で推定するのは困難である. そこで真の事後分布を近似する確率分布関数 $q_{\phi}(z|x)$ を導入する



2.3. VAE の定式化

$$\log p_{\theta}(\mathbf{x}^{(i)}) = D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\theta}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})$$

$$\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) = -D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\theta}(\mathbf{z})) + \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})} \left[\log p_{\theta}(\mathbf{x}^{(i)}|\mathbf{z}) \right]$$

定式化

通常は $p(\mathbf{x})$ を最大にするパラメータを最尤法や MAP 推定にて求めるが計算を簡単にするため本研究では変分下限を用いる
ただし変分下限は実際の確率モデルとギャップがあるため確率分布同士の差を出す KL ダイバージェンスを用いて式 1 のようになるまた、式 3 は変分下限を表す。



3.1. 実験設定

設定

学習対象のデータ・セットとして手書き数字の MNIST を用いる
評価には周辺対数尤度の大きさを比較する

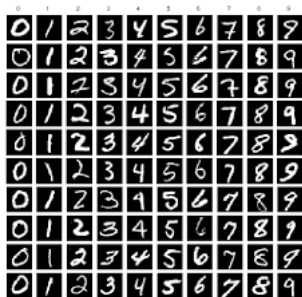


Figure: 4: MNIST の一部サンプル



3.2. 実験結果

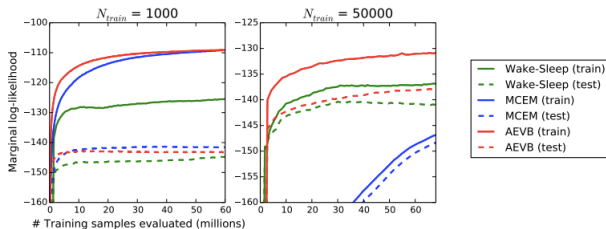


Figure 3: Comparison of AEVB to the wake-sleep algorithm and Monte Carlo EM, in terms of the estimated marginal likelihood, for a different number of training points. Monte Carlo EM is not an on-line algorithm, and (unlike AEVB and the wake-sleep method) can't be applied efficiently for the full MNIST dataset.

Figure 4: 実験結果

結果

関連手法である **wake-sleep algorithm**, **Monte Carlo EM** と比較しても提案手法がもっとも高い数値を示している.



4.1. モデルの特徴

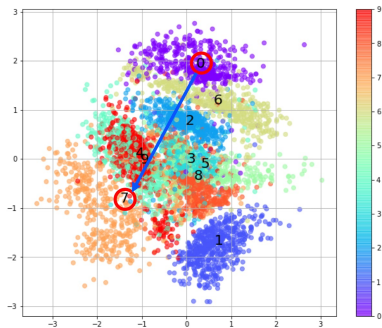


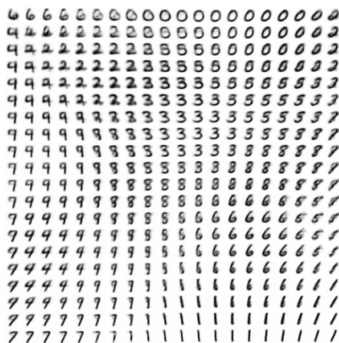
Figure: 5: MNIST における特徴空間のマッピング

発見

図 5 は z を二次元に設定して学習させた特徴空間を可視化した結果である。本モデルの z は正規分布に従うように学習させているため SOM と同じように似た要素を近くに寄せる効果がある



4.2. モデルの応用例



(b) Learned MNIST manifold

Figure: 6: MNIST における近傍要素の表示

数字データでの応用

z の空間ごとに数字を生成した結果が 6 である興味深いことに学習してない数字と数字の間の数字? も生成できている.



4.2. モデルの応用例



顔写真データの応用

同じように喜怒哀楽など表情の一部を学習させた結果喜と怒の間の顔など学習してない微妙な表情も表すことができる。



6. おわりに



ご清聴ありがとうございました.