Variational Auto Encoder

nzw

2016 年 12 月 1 日

1 はじめに

深層学習における生成モデルとして Generative Adversarial Nets (GAN) と Variational Auto Encoder (VAE) [1] が主な手法として知られている．本資料では，VAE を紹介する．本資料は，提案論文 [1] とチュートリアル資料 [2] をもとに作成した．おまけとして潜在表現が離散値を仮定したパターンと Keras による実験結果をつけている．間違いなどがあれば指摘してほしい．

2 Variational Auto Encoder

2.1 導入

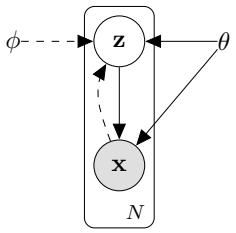

図 1　今回考えるグラフィカルモデル

VAE では図 1 にあるような潜在変数を含んだグラフィカルモデルで表現される生成過程を考える．$\mathbf{x}$ は 1 つのデータで，i.i.d を仮定する．各 $\mathbf{x}$ に対して潜在変数 $\mathbf{z}$ が 1 つ定義され，$\mathbf{z}$ は実数値を要素にもつベクトルとする．また θ はモデル全体に共通した潜在変数，N はデータ数である．これらの依存関係は図 1 において実線で表される．データの生成過程は以下のとおりである．

1. Sample $\mathbf{z}^{(i)}$ from prior $p(\mathbf{z};\theta)$
2. Sample $\mathbf{x}^{(i)}$ from $p(\mathbf{x}^{(i)}|\mathbf{z}^{(i)};\theta)$

MNIST [*1] の例で考えると，$\mathbf{x}$ は 1 つのデータなので MNIST の 1 画像と対応する．潜在変数 $\mathbf{z}$ を仮定することで，潜在変数で表した抽象的な表現をもとに具体的な画像が生成されることをモデリングしている．

図 1 のような潜在変数を仮定した場合，学習データ $\mathbf{x}$ から以下の周辺尤度を最大化することで潜在変数を

[*1] 0~9 までのグレースケール手書き文字画像．1 つの画像は x とすると $0 \leq x_{ij} \leq 1$ かつ $0 \leq i, j \leq 27$ をみたす．

推定する．

$$p(\mathbf{x};\theta) = \int p(\mathbf{x}|\mathbf{z};\theta)p(\mathbf{z};\theta)d\mathbf{z} \tag{1}$$

このとき，事前分布 $p(\mathbf{z};\theta)$ と事後分布 $p(\mathbf{x}|\mathbf{z};\theta)$ はパラメトリックな分布を仮定するため，それらのパラメータ θ について PDF は微分可能だが，潜在変数 θ と $\mathbf{z}$ は未知であるため微分できない．このような場合，EM アルゴリズム，平均場近似を導入した変分ベイズ学習，MCMC 法による推定を行うことが多い．今回の VAE では別の変分推論を与えている．まず VAE では，推定が困難な $p(\mathbf{z}|\mathbf{x};\theta)$ の近似として

$$q(\mathbf{z}|\mathbf{x};\phi) \tag{2}$$

という分布を置く．$q(\mathbf{z}|\mathbf{x};\phi)$ の依存関係は図 1 の点線で与えられる．次節では，この $q(\mathbf{z}|\mathbf{x};\phi)$ と式 1 の関係からニューラルネットワークで最小化する損失関数を定義する．

2.2 Variational Lower Bound

変分ベイズ学習で変分下限を求めたように，KL ダイバージェンス（以下 KL ）との関係式から最適化の対象である *evidence lower bound* (ELBO) を求める．VAE ではこの値を最大化するようにパラメータの推定を行う．推定困難な潜在変数 $\mathbf{z}$ の事後確率分布 $p(\mathbf{z}|\mathbf{x};\theta)$ の推定をするために，比較的容易に推定可能な確率分布 $q(\mathbf{z}|\mathbf{x};\phi)$ を導入した．確率分布間がどれほど違っているかを表す KL を使って $p(\mathbf{z}|\mathbf{x};\theta)$ との関係式から ELBO を求める．

$$KL\big(q(\mathbf{z}|\mathbf{x};\phi)||p(\mathbf{z}|\mathbf{x};\theta)\big) \tag{3}$$

この式には推定困難な事後分布 $p(\mathbf{z}|\mathbf{x};\theta)$ が含まれているため，直接最適化できない．そこで，尤度と式 3 の関係性を利用した式変形を行う．以下の式変形では，見た目を優先してパラメータ ϕ と θ は省略している．

$$\begin{aligned}
KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})) &= \int q(\mathbf{z}|\mathbf{x}) \log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})} d\mathbf{z} && (4)\\
&= \int q(\mathbf{z}|\mathbf{x}) \times \big(\log q(\mathbf{z}|\mathbf{x}) - \log p(\mathbf{z}|\mathbf{x})\big) d\mathbf{z} && (5)\\
&= \int q(\mathbf{z}|\mathbf{x}) \times \big(\log q(\mathbf{z}|\mathbf{x}) - \log \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}\big) d\mathbf{z} && (6)\\
&= \int q(\mathbf{z}|\mathbf{x}) \times \big(\log q(\mathbf{z}|\mathbf{x}) - \log p(\mathbf{x}|\mathbf{z}) - \log p(\mathbf{z}) + \log p(\mathbf{x})\big) d\mathbf{z} && (7)\\
&= \log p(\mathbf{x}) + \int q(\mathbf{z}|\mathbf{x}) \times \big(\log q(\mathbf{z}|\mathbf{x}) - \log p(\mathbf{x}|\mathbf{z}) - \log p(\mathbf{z})\big) d\mathbf{z} && (8)\\
&= \log p(\mathbf{x}) + \int q(\mathbf{z}|\mathbf{x}) \log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} d\mathbf{z} - \int q(\mathbf{z}|\mathbf{x}) \log p(\mathbf{x}|\mathbf{z}) d\mathbf{z} && (9)\\
&= \log p(\mathbf{x}) + KL\big(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\big) - \int q(\mathbf{z}|\mathbf{x}) \log p(\mathbf{x}|\mathbf{z}) d\mathbf{z} && (10)\\
\log p(\mathbf{x}) - KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})) &= \int q(\mathbf{z}|\mathbf{x}) \log p(\mathbf{x}|\mathbf{z}) d\mathbf{z} - KL\big(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\big) && (11)
\end{aligned}$$

となる．KL は負の値は取らないことから，左辺の 2 項目を取り除くことで

$$\log p(\mathbf{x}) \geq \int q(\mathbf{z}|\mathbf{x}) \log p(\mathbf{x}|\mathbf{z}) d\mathbf{z} - KL\big(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\big) = L(\theta, \phi; \mathbf{x}) \tag{12}$$

が常に成立する．このときの右辺が ELBO で

$$L(\theta, \phi; \mathbf{x}) = \int q(\mathbf{z}|\mathbf{x}) \log p(\mathbf{x}|\mathbf{z}) d\mathbf{z} - KL\big(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\big) \tag{13}$$

である．左辺を最大化する代わりに右辺の最大化を行う．これで事後分布 $p(\mathbf{z}|\mathbf{x})$ を含まない ELBO の最適化問題に帰着される．

式 13 の各項についてみると．2 項目の KL は解析的に求まる形で導出され，また正規化項の役割を果たす．1 項目は期待値計算なため，VAE ではサンプリング近似によって求める．例えばサンプル数 L のサンプリング近似は次式となる．

$$L(\theta, \phi; \mathbf{x}) \simeq -KL\big(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\big) + \frac{1}{L}\sum_{l=1}^{L} \log p(x^i|z^{(i,l)}) \tag{14}$$

確率的勾配法で使用するミニバッチ数 M が十分に確保できれば (論文では 100)，$L = 1$ でよいとしている [*2]．また $L > 1$ とする手法もあり，L が大きいほど推定性能が上がるが，その分計算量が増えるのでトレードオフの関係にある．

2.3 VAE における Neural Networks

VAE では

- $q(\mathbf{z}|\mathbf{x}; \phi)$
- $p(\mathbf{x}|\mathbf{z}; \theta)$

の 2 つを NN で近似する．前者が encoder で，後者が decoder に対応する．図 2 に VAE のアーキテクチャを示す．青い部分が損失関数である． 以下では，それぞれの NN について説明する．

2.3.1 encoder

ELBO で $\mathbf{z}$ のサンプル近似が必要であった．ニューラルネットワークの中でサンプルを行うことは難しいので，encoder では入力 $\mathbf{x}$ から $\mathbf{z}$ をサンプルする分布のパラメータ (ガウス分布なら平均と分散) を出力する．

encoder 側の損失関数である KL の項には，事前分布 $p(\mathbf{z}; \theta)$ が含まれる．VAE では，事前分布 $p(\mathbf{z}; \theta)$ として平均ベクトル $\mathbf{0}$，共分散行列 I の多変量正規分布 $\mathcal{N}(\mathbf{z}; \mathbf{0}, I)$ を仮定する．事前分布が多変量正規分布とすれば，事後分布 $p(\mathbf{z}|\mathbf{x})$ も同様に多変量正規分布となるので，$q(\mathbf{z}|\mathbf{x}; \phi)$ も多変量正規分布とする．encoder の損失関数は事前分布 $p(\mathbf{z})$ との KL で定義されるため，以下ではその損失関数を導出する．実は，多変量正規分布同士の KL は，closed form で求まり，さらに $\mathcal{N}_1(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, I)$ なので代入して単純化できる．

$$KL(\mathcal{N}_0||\mathcal{N}_1) = \frac{1}{2}\Big(\mathrm{tr}\big(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0\big) + \big(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\big)^T\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - k + \log_e \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|}\Big) \tag{15}$$

$$= \frac{1}{2}\big(\mathrm{tr}(\boldsymbol{\Sigma}_0) + \boldsymbol{\mu}_0^T\boldsymbol{\mu}_0 - k - \log_e|\boldsymbol{\Sigma}_0|\big) \tag{16}$$

このとき， K は潜在変数の次元数 (=多変数正規分布の次元数) であり，$\boldsymbol{\mu}_0$ は $q(\mathbf{z}|\mathbf{x})$ の平均ベクトル，$\boldsymbol{\Sigma}_0$ は， $q(\mathbf{z}|\mathbf{x})$ の対角の共分散行列である．対角の共分散行列なので，行列ではなく対角成分を要素にもつベクトルとして encoder で推定する．

encoder のネットワーク設計はタスクに応じて異なるが，VAE の論文では，隠れ層 1 層，出力層が $\boldsymbol{\mu}$ と $\sqrt{\boldsymbol{\Sigma}}$ の 2 つからなる NN とする．VAE の元論文では 活性化関数は $tanh$ としているが，非線形な関数でいいので $relu$ とする文献もある．encoder の計算式を以下に示す．

[*2] これが [3] の evidence lower bound と対応する

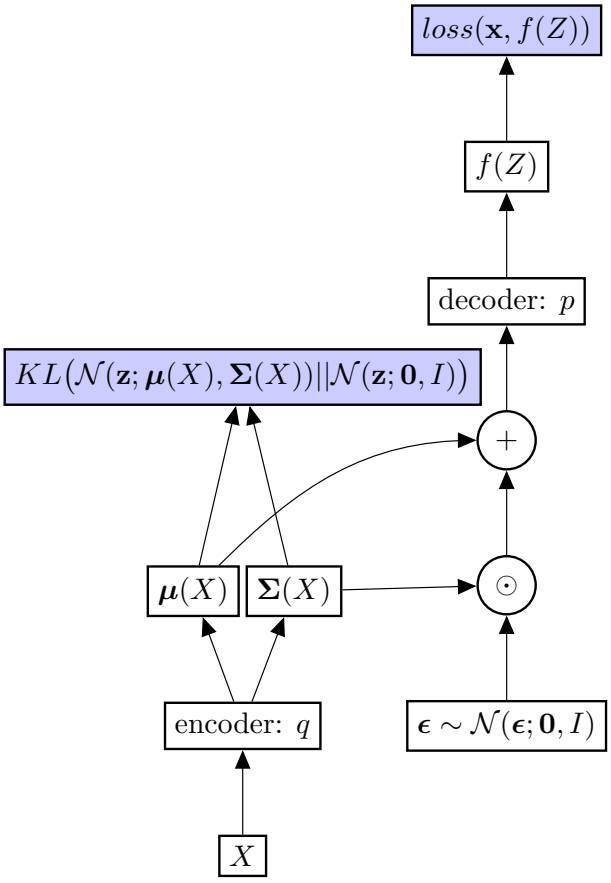

図 2　VAE の外観図

$$\sqrt{\boldsymbol{\Sigma}} = \mathbf{W_s}\mathbf{h} + \mathbf{b}_s \tag{17}$$
$$\boldsymbol{\mu} = \mathbf{W_m}\mathbf{h} + \mathbf{b}_m \tag{18}$$
$$\mathbf{h} = f_*(\mathbf{W}_x\mathbf{x} + \mathbf{b}_x) \tag{19}$$

2.3.2　decoder

decoder は encoder の出力したパラメータ $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ をパラメータにもつ確率分布からサンプルした，$\mathbf{z}$ を入力とし，$\mathbf{x}$ を復元する NN である．auto-encoder と系譜としてみると encoder をひっくり返したような NN となる．

$$\mathbf{y} = f_\sigma(W_o f_*(W_h\mathbf{z} + b_h) + b_o) \tag{20}$$

MNIST やカラー画像では値はスケーリングされているため，は 0~1 の間に収まるロジスティクスシグモイド関数を出力層の活性化関数とする．loss 関数はデータに依存するが，ロジスティックシグモイド関数であれば binary cross entropy となる．

2.4 Reparameterization Trick

VAE のキモがここで紹介する reparametarization trick にある．

式 14 の第 2 項は encoder で推定したパラメータから $\mathbf{z}$ をサンプルする必要があった:

$$\mathbf{z} \sim q(\mathbf{z}|\mathbf{x};\phi) \tag{21}$$

しかし $\mathbf{z}$ のサンプリングを多変量正規分布から行うと decoder と encoder の計算グラフが途切れてしまうため誤差逆伝播法が使えない．そこで VAE では encoder で推定した $q(\mathbf{z}|\mathbf{x};\phi)$ のパラメータ用いた確率分布に従って $\mathbf{z}$ をサンプルせずに，関数から決定的に $\mathbf{z}$ を生成する．つまり

$$\begin{aligned} \mathbf{z}^i &= g(\epsilon^i, \mathbf{x}^i; \phi) \\ where &\quad \epsilon^i \sim p(\epsilon) \end{aligned} \tag{22}$$

となるような関数 g を考える．ニューラルネットワークの最適化とは無関係な項 ϵ と encoder で推定した $q(\mathbf{z}|\mathbf{x})$ のパラメータとで $\mathbf{z}$ を表現できれば，誤差逆伝播法が可能になる．これを reparametarization trick という．

共分散行列が対角行列の多変量正規分布の場合，

$$\begin{aligned} \mathbf{z} = g(\epsilon^i, \mathbf{x}^i; \phi) &= \boldsymbol{\mu} + \boldsymbol{\Sigma}^{\frac{1}{2}} \odot \boldsymbol{\epsilon} \\ where &\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, I) \end{aligned} \tag{23}$$

が成り立つことが知られている [*3] ．このとき $\odot$ は要素積である．

3 潜在変数が離散値のときの reparametarization trick

VAE の論文では潜在変数 $\mathbf{z}$ を連続値で構成した．潜在変数を離散値で表現するための自然な reparametarization trick の論文は最近になって arXiv に投稿されている [*5]．個人的な印象として，[4] のほうが順を追って説明している．reparametarization trick で使う変数 (これまで説明してきた， $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ に対応) を実装に落としにくいが，証明や分布の形については詳しく述べられているため理論的な参考になる．同様のアイディアである [5] は実験で半教師あり学習と教師なしの 2 つを行っており，実装上参考にしやすい．今回の説明は [5] を元に行う．

潜在変数 $\mathbf{z}$ を離散値で表すことを考える．以下では，離散値は 1 から K までの値をとるものとする．深層学習では，基本的に離散値は one-hot ベクトル (あるいは one-of-K 表現) とするため，例えば $K = 6$ において 2 は one-hot ベクトル $\mathbf{z} = [0, 1, 0, 0, 0, 0]$ で表される．このような離散値を出力する分布のパラメータを encoder で推定し，reparametarization trick でパラメータから one-hot ベクトルを構成する．離散値を 1 つ生成するような分布といえばカテゴリカル分布

$$p(k = i) = \pi_i.\ where\ \sum \pi_i = 1.\ and\ \forall i\ \ \pi_i \geq 0. \tag{24}$$

[*3] 余談だが，nzw は確率統計に疎いので，ガウス分布を一様分布から構成する方法を少し調べて見たところ．この証明は PRML の演習問題 11.5 の回答と対応しているようで，wikipedia にも記述があるくらいの事実らしい [*4]．念のため 1 変数については証明を appendix につけた．

[*4] https://en.wikipedia.org/wiki/Multivariate_normal_distribution#Geometric_interpretation

[*5] 今回紹介した手法以外にも離散分布に対する reparametarization trick の別のアプローチを PFN の得居さんと東大の佐藤さんが arXiv に同時期に投稿している

がある．encoder では，入力 $\mathbf{x}$ を K 次元のベクトル $\boldsymbol{\pi}$ に変換する．カテゴリカル分布からサンプルを得るには，多変量正規分布と同様に関数としてサンプル値を決定する．ここでは Gumbel 分布を用いた Gumbel-Max trick [6] を使用する．Gumbel 分布も正規分布と同様に一様乱数からサンプルが生成可能であることが明らかになっているため，一様分布からのサンプルを使って関数的に決定できる．まず乱数 u を K 個生成してから，one-hot ベクトルを構成ために $argmax$ の演算を行う:

$$G_*(0,1) = -\log\big(-\log(u)\big) \ \ where \ \ u \sim Uniform(0,1) \tag{25}$$

$$argmax_k[G_k(0,1) + \log \pi_k] \sim \frac{\pi_k}{\sum_i^K \pi_i} \tag{26}$$

$argmax$ を使うと対応する次元の π_i 以外の勾配は 0 になってしまう．そこで $argmax$ を使わずに one-hot ベクトルを Gumbel-Softmax [*6] で直接近似する:

$$o_k = \frac{\exp\Big(\big(\log \pi_k + G_k(0,1)\big)/\tau\Big)}{\sum_i^K \exp\Big(\big(\log \pi_i + G_i(0,1)\big)/\tau\Big)} \tag{27}$$

τ は *temperature* と呼ばれるパラメータで，$\tau \to 0$ で one-hot 表現に近づき，大きいほど複数の次元が非零をとり，$\tau \to \infty$ で離散一様分布になる．τ を変化させた時の Gumbel-softmax 関数の出力は付録を参照されたい．

実装上，τ の値は重要で，提案論文では τ を annealing している．もう一方の [4] では，潜在変数が $K \in \{2, 4, 8\}$ のときは $\tau = \frac{2}{3}$ がよいとしている．

さらに細かい話だが，one-hot ベクトルを構築する際に，Gumbel 分布と足し合わせる π は正の値であればよい [6] ．今回の場合，encoder で KL を計算する都合上，確率値としたほうが扱いやすいので $\boldsymbol{\pi}$ を確率ベクトルとしている．

encoder の出力はカテゴリカル分布のパラメータ $\boldsymbol{\pi}$ であった．事前分布は K 次元の離散一様分布になるため，簡単な形で求めることができる．

$$KL\big(p(k;\boldsymbol{\pi})||p(k;\frac{1}{K}\times\mathbf{1})\big) = \sum_k^K \pi_k \log K\pi_k \tag{28}$$

実際には $\boldsymbol{\pi}$ は複数あること (潜在変数は複数存在) に注意したい．

参考文献

[1] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2014.

[2] Carl Doersch. Tutorial on Variational Autoencoders. *arXiv*, pages 1–23, 2016.

[3] M D Hoffman, D M Blei, C Wang, and J Paisley. Stochastic Variational Inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[4] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *arXiv*, 2016.

[5] Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. *arXiv*, 2016.

[6] Chris J Maddison, Daniel Tarlow, and Tom Minka. A* Sampling. In *NIPS*, 2014.

[*6] もう一方の論文では *Concrete distribution* と呼んでいる

付録 A　1 次元正規分布における関数化

1 変数の正規分布 $x \sim \mathcal{N}(x; \mu, \sigma^2)$ に対して $x = \mu + \sigma\epsilon \ \ where \ \ \epsilon \sim \mathcal{N}(\epsilon; 0, 1)$ であることを示す．

A.1　事前計算

- $\mathbb{E}[\epsilon] = 0$
- $\mathbb{V}[\epsilon] = \mathbb{E}[\epsilon^2] + \mathbb{E}[\epsilon]^2 = \mathbb{E}[\epsilon^2] = 1$

A.2　期待値

$$\begin{aligned} \mathbb{E}[\mu + \epsilon\sigma] &= \mathbb{E}[\mu] + \sigma\mathbb{E}[\epsilon] && (29)\\ &= \mu && (30)\\ &= \mathbb{E}[x] && (31) \end{aligned}$$

A.3　分散

$$\begin{aligned} \mathbb{V}[\mu + \epsilon\sigma] &= \mathbb{E}[\epsilon^2] - \mathbb{E}[\epsilon]^2 && (32)\\ &= \mathbb{E}[\mu^2 + 2\mu\sigma\epsilon + \epsilon^2\sigma^2] - \mu^2 && (33)\\ &= \mathbb{E}[\mu^2] + \mathbb{E}[2\mu\sigma\epsilon] + \mathbb{E}[\epsilon^2\sigma^2] - \mu^2 && (34)\\ &= \mu^2 + 2\sigma\mu\mathbb{E}[\epsilon] + \sigma^2\mathbb{E}[\epsilon^2] - \mu^2 && (35)\\ &= \sigma^2 && (36)\\ &= \mathbb{V}[x] && (37) \end{aligned}$$

付録 B　Gumbel-softmax 関数

図 4 のような 6 つの離散値におけるカテゴリカル分布を考える．この分布に従ってサンプルした one-hot ベクトルを近似する Gumbel-softmax 関数は，τ によって異なるため，τ を変化させた時の Gumbel-softmax 関数を示す．

付録 C　実験

C.1　MNIST

提案論文と同様に VAE で MNIST の生成モデルを学習する．Keras の公式サンプルにも含まれているためこれを使うのが最も早いと思われる．潜在変数 $\mathbf{z}$ を 2 次元ベクトルとした．図 6 では test データを encoder の入力とし，得られた平均値ベクトルを 2 次元座標にプロットしている．各点の色は MNIST の数字 (クラス)

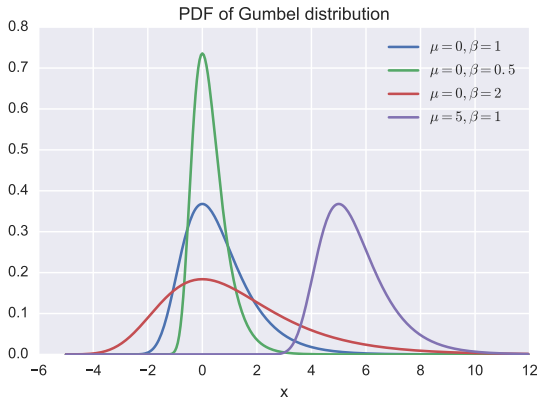

図 3　Gumbel 分布の確率密度関数

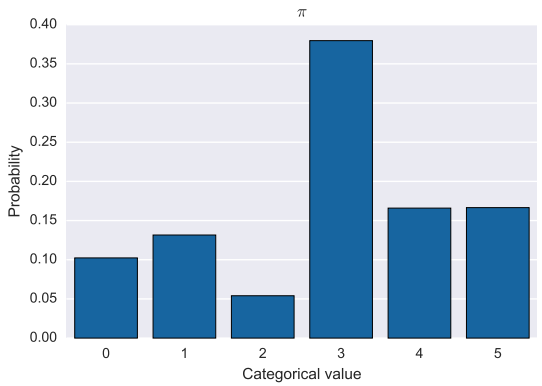

図 4　6 変数のカテゴリカル分布例

に対応する．同じクラスが特定の箇所に偏っていることが確認できる． 図 C.1 では 2 変数標準正規分布からサンプルした平均ベクトルから decoder で構成した数値である．中央部分がおよそ **0** に対応し，左上に近いほど両方の次元の値が負のベクトルで，右下に近いほど両方の次元の値が正のベクトルである．

C.2　CIFAR10

MNIST と同様に潜在変数を 2 次元の VAE で CIFAR10 の学習を行った．図 C.1 と同様に構成した画像を図 C.2 に示す．潜在変数の次元数が少ないためか，背景と中央部分のみ構成できている．使用したコード は github 上で公開している．

C.3　Categorical MNIST

Gumbel-softmax を使って MNIST を VAE で学習した．潜在変数は 20 個の離散値をとる変数を 10 個から構成される．図 C.3 は，test データを encoder で変換した π の分布である．色が濃いほどその次元に対応する離散値を取りやすくなる．図 C.3 は，ランダムに生成した one-hot ベクトルから decoder で復元結果である．使用したコード は github 上で公開している．

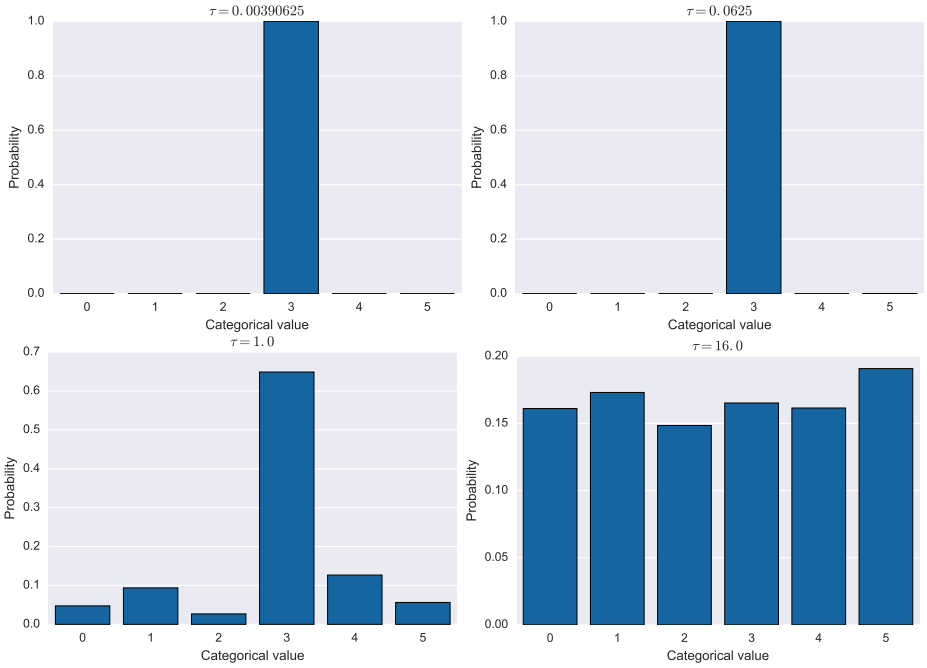

図 5　図 4 のカテゴリカル分布において τ を変化させたときの Gumbel-softmax の出力値．τ が高いほど，離散一様分布に近づく．

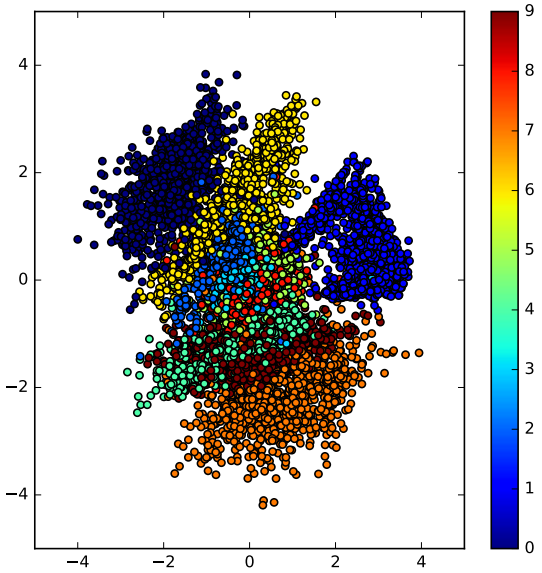

図 6　test データから求めた平均ベクトル．色はクラスに対応している．