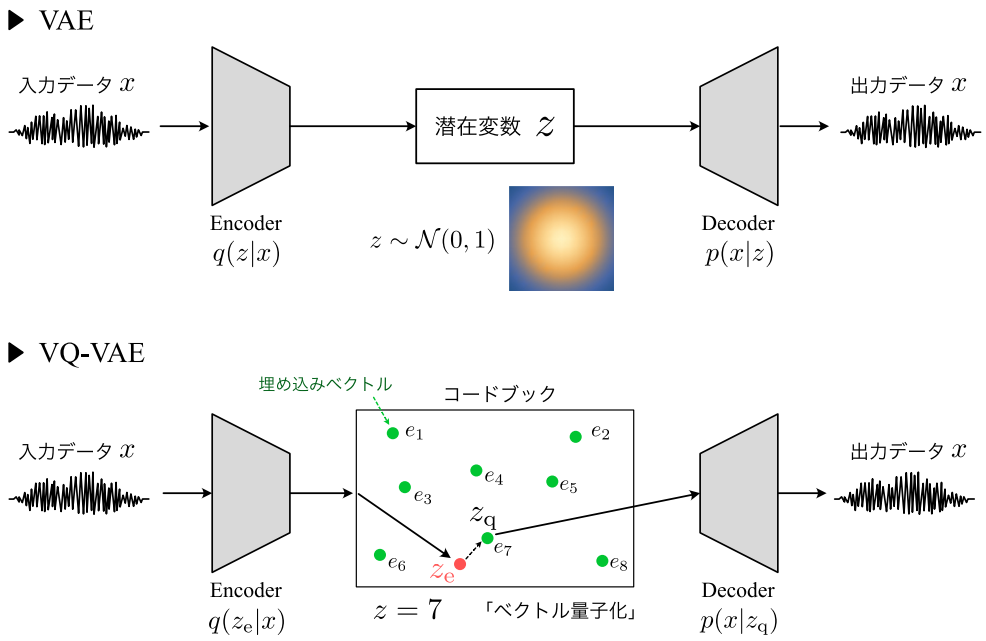
VQ-VAE (Vector Quantised-Variational AutoEncoder)

VQ-VAE は、VAE (Variational AutoEncoder) の派生技術にあたる生成モデルです。「自然界の様々な事物の特徴を捉えるには離散変数の方が適している」という発想から、潜在変数が離散値となるように学習が行われます。これにより、従来の VAE で起こりやすいとされる “posterior collapse” の問題を回避し、高品質のデータを生成することが可能となります。この解説プリントでは、この VQ-VAE の基本的なアイディアを解説しています。内容は VQ-VAE の提案論文 (Van der Oord+, 2017) に基づいていますので、必要に応じて論文の方も参照して下さい[1]。

♣ VAE と VQ-VAE の比較

大雑把に VAE と VQ-VAE の構造を表すと、次のようになります。



両者ともベースの構造はオートエンコーダとなっており、「入力を潜在表現にエンコード → 潜在表現から入力を再構成」という構造は同じです。両者の最大の違いは、

- VAE → 潜在変数 z が Gauss 分布に従うベクトルになるように学習を行う
- VQ-VAE → 潜在変数 z が離散的な数値となるように学習を行う

という点です。VQ-VAE では、Encoder と Decoder の間に Encoder の出力を離散的な潜在変数に対応させる「ベクトル量子化処理 (VQ: Vector Quantization)」が行われます[2]。詳細は後ほど説明しますが、これは Encoder の出力 z_e を「コードブック」にマッピングすることで実装さ

[1]「arXiv:1711.00937」で検索すれば 1 番上に出てきて誰でも無料で読めます。
[2]量子化 (Quantization) とは、連続的な値を取る量を、整数などの離散的な値で近似的に表現することです。

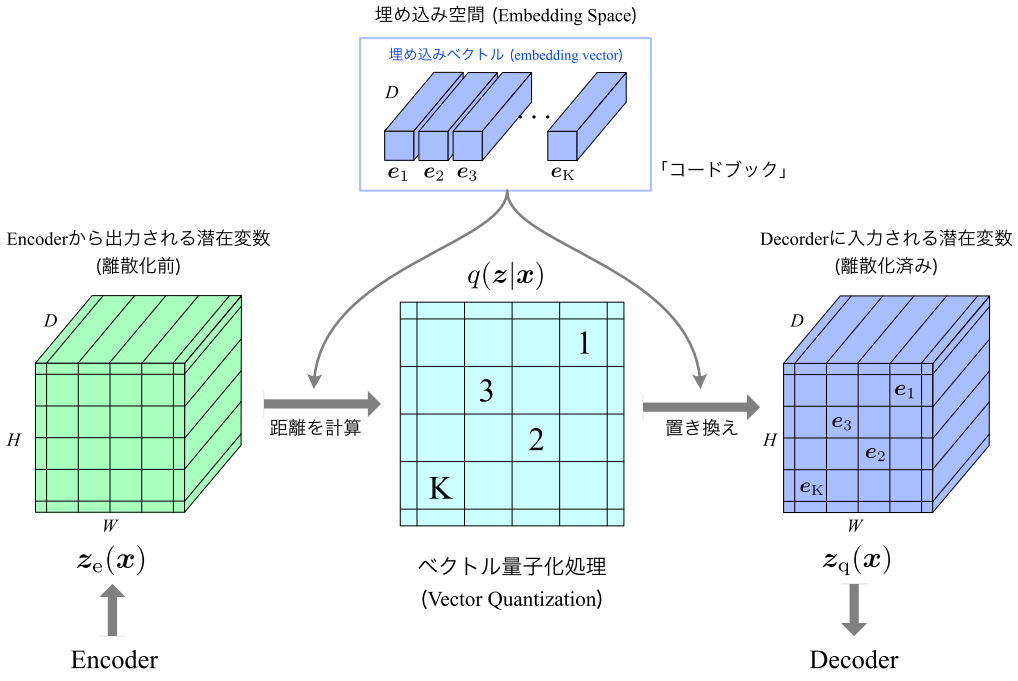
れます。ではなぜ、VQ-VAE ではわざわざベクトル量子化処理を行い離散的な潜在変数を用いているのでしょうか？ 提案論文では、離散的な潜在変数を用いるメリットとして、次の 2 点をあげています。

1. **離散変数の方がデータの特徴を捉えるのに適している**
   例えば、我々が画像を見た時に、それがどのような画像であるかは「猫」や「車」といった言語で簡潔に表現することができます。言語が離散的であるように、離散的な表現を用いた方がより簡潔にデータの特徴を捉えることができると期待されます。すなわち、潜在表現 (Encoder の出力) を量子化することで、意味のある潜在表現を獲得できると期待されます。

2. **“posterior collapse”を防ぐ**
   オリジナルの VAE では、Pixel CNN などの強力なデコーダーと組み合わせ時に、潜在変数がデータの特徴をうまく捉えることが出来なくなる “posterior collapse” と呼ばれる現象がしばしば問題となることが知られています。画像生成の場合には、この問題のために VAE で生成される画像は輪郭がぼやける傾向にあります。一方、離散的な潜在変数を用いた VQ-VAE ではこの posterior collapse の問題を回避し、高品質のデータを生成することができると提案論文では主張されています。

♣ VQ-VAE のアーキテクチャ

以下では、VQ-VAE のアーキテクチャ (主にベクトル量子化処理の部分) を説明します。ここでは、入力データ $\boldsymbol{x}$ として画像データを想定しています。



Encoder から出力される各ピクセルに対応する潜在変数を $\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x})$ とします。チャンネル数を D とすると、$\boldsymbol{z}_{\mathrm{e}}(x)$ は D 次元ベクトルとなります。上図は、この各ピクセルの D 次元ベクトル $\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x})$ のベクトル量子化の手順の概要を表しています。先ず、予め用意しておいた K 個 D 次元の埋め込みベクトル (embedding vector) $\boldsymbol{e}_j$ $(j = 1, 2, \cdots, K)$ と $\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x})$ の離れ具合を確認します[3]。2 つのデータの離れ具合を表す指標は色々ありますが、ここでは次の「ユークリッド距離 (Euclidean distance)」を用いて離れ具合を測ることにします。

$$\|\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x}) - \boldsymbol{e}_j\|_2 = \sqrt{\big(z_{\mathrm{e}}(\boldsymbol{x})_1 - e_{j,1}\big)^2 + \big(z_{\mathrm{e}}(\boldsymbol{x})_2 - e_{j,2}\big)^2 + \cdots + \big(z_{\mathrm{e}}(\boldsymbol{x})_D - e_{j,D}\big)^2}. \tag{1}$$

ただし、それぞれのベクトルの成分は

$$\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x}) = \big(z_{\mathrm{e}}(\boldsymbol{x})_1,\ z_{\mathrm{e}}(\boldsymbol{x})_2,\ \cdots,\ z_{\mathrm{e}}(\boldsymbol{x})_D\big), \quad \boldsymbol{e}_j = \big(e_{j,1},\ e_{j,2},\ \cdots,\ e_{j,D}\big) \tag{2}$$

としています。(1) 式に出てくる $\|\cdot\|_2$ は、L2 ノルムと呼ばれる記号です[4]。一見すると分かりにくいかもしれませんが、$\boldsymbol{a} = (x_1, y_1)$、$\boldsymbol{b} = (x_2, y_2)$ のような 2 変数のベクトルの場合には、

$$\|\boldsymbol{a} - \boldsymbol{b}\|_2 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3}$$

となり、いわゆる x-y 座標上の 2 点 (x_1, y_1)、(x_2, y_2) の距離の公式に対応していることが分かります。つまり、三平方の定理を使って計算されるような距離の概念を、一般の D 次元のベクトルの場合に拡張したものが (1) 式となっています。

(1) 式で計算される距離を K 個の埋め込みベクトルそれぞれに対して計算します[5]。そして、$\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x})$ を最も距離が近い (すなわち最も似ている) 埋め込みベクトルに置き換えるという操作を行います。例えば、このセクションの最初の図では、右上のピクセルの $\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x})$ が 1 番目の埋め込みベクトル $\boldsymbol{e}_1$ と最も距離が近い状況となっています。そのため、このピクセルの $\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x})$ は、$\boldsymbol{e}_1$ に置き換えられています。この操作を数式で表現すると、次のようになります。

$$\boldsymbol{z}_{\mathrm{q}}(\boldsymbol{x}) = \boldsymbol{e}_k \quad \left(\text{ただし、} k = \arg\min_j \|\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x}) - \boldsymbol{e}_j\|_2\right) \tag{4}$$

$\arg\min_j \|\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x}) - \boldsymbol{e}_j\|_2$ という記号は、「$\|\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x}) - \boldsymbol{e}_j\|_2$ が最も小さくなる時の j」という意味で、今説明した置き換えの操作を単に数式で表現しただけです。$\boldsymbol{z}_{\mathrm{q}}(\boldsymbol{x})$ が、埋め込みベクトルへの置き換え (ベクトル量子化) を行なった後の潜在変数です[6]。この量子化された潜在変数 $\boldsymbol{z}_{\mathrm{q}}(\boldsymbol{x})$ を Pixel CNN などの強力な Decorder に入力することで、高品質な画像を再構成することが可能となります。

---

[3] K 個の埋め込みベクトルのセットのことを「コードブック」と呼びます。

[4] 機械学習では、L1 ノルム $\|\cdots\|_1$ で表現される「マンハッタン距離」もよく用いられます。

[5] 実際に実装する際には、$\|\cdots\|_2$ の定義に従って計算するのではなく、より効率的な手法で計算されます。

[6] $\boldsymbol{z}_{\mathrm{q}}(\boldsymbol{x})$ の添字部分の q は、“quantized (量子化された)”から来ています。

♣ VQ-VAE における学習

このセクションでは、VQ-VAE における Encoder、Decorder のネットワーク、そして埋め込みベクトル (コードブック) の学習について見ていきます。VQ-VAE では Encoder の出力 $\boldsymbol{z}_{\mathrm{e}}$ を、最近傍の埋め込みベクトル $\boldsymbol{e}_k$ に置き換えるというベクトル量子化処理が行われます。この操作に対応する (近似) 事後分布は、次のような one-hot な分布で表現されます。

$$q(z=k|\boldsymbol{x}) = \begin{cases} 1 & (k = \arg\min_j \|\boldsymbol{z}_{\mathrm{e}} - \boldsymbol{e}_j\|) \\ 0 & (\text{otherwise}) \end{cases} \tag{5}$$

つまり、K 個の埋め込みベクトル中から最近傍の埋め込みベクトルのみを選ぶため、他の埋め込みベクトルが選ばれる確率は 0 となり、上式のような決定的な one-hot な分布となります。ここで z は、最近傍の埋め込みベクトルの添字 (index) に対応する離散的な潜在変数です。

この時、通常の VAE と同様に、次式の ELBO (evidence lower bound) の最大化で Encoder と Decoder のネットワークの学習を行うことを考えます。

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbf{KL}\big(q(z|\boldsymbol{x})\|p(z)\big) + \log p(\boldsymbol{x}|\boldsymbol{z}_{\mathrm{q}}(\boldsymbol{x})) \tag{6}$$

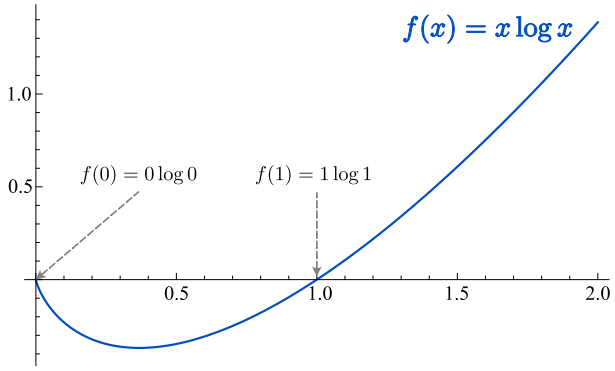
1 項目は事後分布 $q(z|\boldsymbol{x})$ と z の事前分布 $p(z)$ との KL ダイバージェンスを、2 項目は再構成誤差 (reconstraction error) を表しています。VQ-VAE の学習時には、事前分布 $p(z)$ は一様分布 ($p(z) = 1/K$) で固定します[7]。このことを用いると、1 項目の KL ダイバージェンスは次のように計算することができます。

$$\begin{aligned} \mathbf{KL}\big(q(z|\boldsymbol{x})\|p(z)\big) &= \mathbb{E}_{z\sim q(z|\boldsymbol{x})}\left[\log \frac{q(z|\boldsymbol{x})}{p(z)}\right] \\ &= \sum_{k=1}^{K} q(z=k|\boldsymbol{x}) \log q(z=k|\boldsymbol{x}) - \sum_{k=1}^{K} q(z=k|\boldsymbol{x}) \log p(z=k) \\ &= -\log \frac{1}{K} \sum_{k=1}^{K} q(z=k|\boldsymbol{x}) \\ &= -\log \frac{1}{K} = \log K \end{aligned} \tag{7}$$

2 行目の 1 項目は 0 となります。(5) 式にあるように $q(z=k|\boldsymbol{x})$ は one-hot な分布であるため、総和記号の中身 $\big[q(z=k|\boldsymbol{x}) \log q(z=k|\boldsymbol{x})\big]$ は、$0 \log 0$ か $1 \log 1$ となります。下に $f(x) = x \log x$ のグラフを描きましたが、$f(0) = f(1) = 0$ となっています[8]。つまり、総和記号の中身は必ず 0 となるため、2 行目の 1 項目は消えることが分かります。後は、2 行目の 2 項目の部分に関して、事前分布が一様分布で $\log p(z)$ が定数となるため総和記号の外に出し、対数 log の性質を用

[7] 学習が終わったら、事前分布 $p(z)$ は Pixel CNN で学習します。

[8] $\log x$ は $x = 0$ で発散しますが、$\lim_{x\to+0} x \log x = 0$ となっています。



いて式変形しているだけです。最後の行に移る所では確率分布の性質

$$\sum_{i=1}^{K} q(z = k|\boldsymbol{x}) = 1 \tag{8}$$

を用いています[9]。以上の計算から、KL ダイバージェンスは結局のところ定数 $(\log K)$ となるため学習の際には無視し、(6) 式の 2 項目 (再構成誤差) のみを考えればよいことが分かります。ただし、実際の VQ-VAE の学習では、この再構成誤差項に加え、ベクトル量子化処理に伴う誤差も目的関数に反映させる必要があります[10]。そのため、ベクトル量子化処理に伴う誤差を表す 2 つの項 (codebook loss と commitment loss) を加えた、次の量が VQ-VAE の目的関数として用いられます。

$$\mathcal{L}_{\mathrm{VQ-VAE}} = \log p\big(\boldsymbol{x}|\boldsymbol{z}_{\mathrm{q}}(\boldsymbol{x})\big) \ + \ \|\mathrm{sg}\big[\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x})\big] - \boldsymbol{e}\|_2^2 \ + \ \beta\|\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x}) - \mathrm{sg}\big[\boldsymbol{e}\big]\|_2^2. \tag{9}$$

ここで、$\boldsymbol{e}$ は最近傍の埋め込みベクトル、$\mathrm{sg}[\cdot]$ は誤差逆伝播 (backpropagation) 時に勾配を計算しないようにする処理 (stop gradient) を表します。先ほど説明したように 1 項目は再構成誤差を表しており、Encoder と Decoder のネットワークのパラメータ更新に用いられます。誤差逆伝播時には、ベクトル量子化処理の部分が微分不可能となっていますが、勾配を Decorder の入力 $\boldsymbol{z}_{\mathrm{q}}(\boldsymbol{x})$ から Encoder の出力 $\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x})$ に単純にコピーすることで Encoder まで逆伝播させます[11]。$\mathcal{L}_{\mathrm{VQ-VAE}}$ の 2 項目

$$\|\mathrm{sg}[\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x})] - \boldsymbol{e}\|_2^2$$

は codebook loss と呼ばれ、埋め込みベクトル $\boldsymbol{e}$ を Encoder の出力ベクトル $\boldsymbol{z}_{\mathrm{e}}$ に近づける効果があるため、埋め込みベクトル (コードブック) の更新に用いられます。$\mathcal{L}_{\mathrm{VQ-VAE}}$ の 3 項目

$$\beta\|\boldsymbol{z}_{\mathrm{e}}(\boldsymbol{x}) - \mathrm{sg}[\boldsymbol{e}]\|_2^2$$

は commitment loss と呼ばれ、Encoder の出力ベクトル $\boldsymbol{z}_{\mathrm{e}}$ を埋め込みベクトル $\boldsymbol{e}$ に近づける効

[9] (8) 式が成り立っていることは、(5) 式から確認できます。
[10] これにより、Encoder/Decoder のネットワークのみならず、コードブックも学習することができます。
[11] このような処理は一般に “straight-through estimator” と呼ばれます (参考文献 arXiv:1308.3432)。

果があるため、Encoder のネットワークのパラメータ更新に用いられます。この項は、潜在変数 $\boldsymbol{z}_{\mathrm{e}}$ が埋め込みベクトルから離れすぎないようにする (潜在空間の大きさが大きくなり過ぎないようにする) ために導入されています。この項がないと、コードブックの学習が Encoder のネットワークの学習に追いつかない場合に潜在空間がどんどん大きくなってしまいます。β はハイパーパラメータですが、提案論文では β は非常にロバストで、実験的には β の値を $0.1 < \beta < 2.0$ の範囲で変化させても結果は殆ど変わらないと報告されています。

♣ VQ-VAE の展望

VAE の技術は急速に進展しており、最近では VQ-VAE を階層構造にすることでさらに高解像度画像を生成できるようにした「VQ-VAE2」と呼ばれる技術が開発されています (Ali Razavi+, 2019)[12]。VQ-VAE2 では、潜在表現を異なるスケールごとに階層的に学習します。この潜在表現は元のデータより大幅に小さくなりますが、これをデコーダに入力することで非常に鮮明でリアルなデータを再構築することが可能となっています。このように、VAE に関する技術は急速に発展しており、AI の新たな可能性を示し続けています。

[12] arXiv:1906.00446