



## 2.1 Variational Autoencoder(VAE)

データセット  $x$  のサンプルから  $x$  の確率密度を推定する方法として VAE[Kingma 14] は提案された. 対数尤度を展開すると

$$\log p(x) = E_{q_\phi(z|x)} [\log p_\theta(x|z)] - D_{\text{KL}}(q_\phi(z|x) || p(z)) + D_{\text{KL}}(q_\phi(z|x) || p(z|x)) \quad (1)$$

$$\geq E_{q_\phi(z|x)} [\log p_\theta(x|z)] - D_{\text{KL}}(q_\phi(z|x) || p(z)) \quad (2)$$

となる. ここで  $D_{\text{KL}}$  はカルバックライブラー情報量と呼ばれる確率分布間の距離のようなものである. VAE では  $\phi, \theta$  を用いて尤度関数の下限を最大化を行うことで  $\log p(x)$  を推定する. ここでわざわざ下限を用いたのは  $p(z|x)$  を推定することが困難なので, 下限を最大化することで  $D_{\text{KL}}(q_\phi(z|x) || p(z|x))$  を小さくし  $p(z|x)$  を  $q_\phi(z|x)$  で近似するためである. VAE の学習を行うと  $x \rightarrow z$  を encode する  $q_\phi(z|x)$  と  $z \rightarrow x$  を decode する  $p_\theta(x|z)$  が得られる. 本研究では具体的な  $x$  から抽象的な表現である  $z$  を獲得することで系列のデータの抽象化を行う.

## 2.2 $\beta$ -VAE

VAE の隠れ層において獲得する表現を Disentangle 化できる  $\beta$ -VAE について説明する.  $\beta$ -VAE は  $p(z) = \mathcal{N}(0, I)$  として VAE の正則化項の係数  $\beta \geq 0$  に拡張した学習手法である.

$$\mathcal{L} \equiv E_{q_\phi(z|x)} [\log q_\theta(x|z)] - \beta D_{\text{KL}}(q_\phi(z|x) || p(z)) \quad (3)$$

ここで  $\beta$  はハイパーパラメータである.  $\beta$  が大きくなると潜在変数  $q_\phi(z|x_n)$  を平均 0 分散 1 の Gauss 分布に近づける効果を優先されるので各  $z$  は 0 に近づこうとする. しかし全ての  $z$  が 0 になると入力データを再構成できないので必要最低限の  $z$  を残して  $q_\phi(z|x_n) = \mathcal{N}(0, I)$  になる. 結果として出力変数  $x$  をできるだけ少ない  $z$  の組み合わせで表現しようとし Disentangle な表現を獲得することができる.

次に [Burgess 17] で提案されている  $\beta$ -VAE を安定させる重要な改善について説明する.  $\beta$ -VAE の loss 関数は  $\beta \neq 1$  の場合  $\log p(x)$  の変分下限の最大化として理解することはできないが, Information bottleneck の考え方から理解できる [Achille 17]. 今, 入力データ  $x$  を利用してタスク  $y$  を解くために最適な表現を  $z$  とした場合

$$\min I(x, z), \quad \text{s.t. } H(y|x) = H(y|z) \quad (4)$$

の最適化から  $\beta$ -VAE を導出することができる. まず Eq. 4 の最初の式は  $x$  と  $z$  をできるだけ無関係にする最適化になっている. 次に Eq. 4 の 2 つめの式は  $x \rightarrow y$  を行うのに必要な情報を保存するという制約条件になっている つまり  $x \rightarrow y$  を行うための情報を保ちながら  $z$  をできるだけ  $x$  と無関係に最適化するという問題になっている. Eq. 4 をラグランジュ未定乗数法で解くと

$$H(y|z) \simeq E_{x, y \sim p(x, y)} [E_{z \sim q(z|x)} [-\log p(y|z)]] \quad (5)$$

$$I(x, z) = E_{x \sim p(x)} [D_{\text{KL}}(q(z|x) || p(z))] \quad (6)$$

なので

$$\mathcal{L} = E_{q(z|x)} [\log p(y|z)] - \beta D_{\text{KL}}(q(z|x) || p(z))$$

となり出力の  $y$  を  $x$  とみなせば  $\beta$ -VAE を導出できた. ここで  $\beta$  はラグランジュ未定乗数法から来る定数である. これまでの導出により  $\beta$ -VAE は第一項目の再構成誤差 ( $y$  の場合は cross-entropy) の最大化問題に, 第二項目のカルバックライブラー情報量の正則加工をつけたものと理解することができるので  $\beta$  はバランスを取る項になっていて,  $\beta$  が大きいと再構成より Disentangle が優先され再構成が上手く行かず (例えば画像であるとぼやける), 逆に  $\beta$  が小さいと再構成はうまくいくが disentangle がうまくいかないということが実験的にも知られている [Burgess 17]. そこで大きい  $\beta$  を使い再構成を行う方法として提案されたのが以下のように第二項目に線形シフトを入れるという方法である.

$$\mathcal{L} \equiv E_{q_\phi(z|x)} [\log q_\theta(x|z)] - \beta (D_{\text{KL}}(q_\phi(z|x) || p(z)) - C) \quad (7)$$

シフトの効果により最適化問題は  $I(x, z) = C$  となり  $C$  だけ情報を持つことが許されるため大きい  $\beta$  を用いてもきれいに再構成を行うことができる.

## 2.3 Variational Ladder Autocoder

教師ありの深層学習は階層的に特徴を学習することで成功を収めている. 一方で教師なしの生成モデルにおいても ladder 型の Variational Autoencoder を利用することで実現している研究がある [Zhao 17]. 先行研究のポイントは階層化には 2 パターンの方法があり 1 つは stacked hierarchy と呼ばれるもので layer 方向に VAE を多層化したものである. もう 1 つは Architectural hierarchy と呼ばれるもので ladder 型のネットワーク構造になっており潜在変数  $z$  が直接はつながっていないところがポイントである. 先行研究では ladder 型の接続を用いると layer ごとに階層的な意味の分離ができることを実験的に示している.

## 3. Proposal model

この章では系列の意味を Disentangle できるモデルを提案する. 系列の Disentangle 表現とは, 潜在変数  $z$  から系列データ  $x_1, x_2, \dots, x_T$  の生成を以下のように行った際に

$$p(x_1, x_2, \dots, x_T) = \int p(x_1, x_2, \dots, x_T | z) p(z) dz \quad (8)$$

$z$  ごとに情報が絡まらずに存在する状態のことを指す. 提案手法はいずれも  $p(z|x_1, \dots, x_T)$  で encode した  $z$  間の相互情報量を小さくすることで Disentangle な表現学習を実現している.

### 3.1 TCLVAE (Timeseries Convolutional VLAE)

系列を扱う主流な Neural network のモデルは Long Short-term Memory (LSTM)[Hochreiter 97] や Gated Recurrent Unit (GRU)[Chung 14] などいくつかあるが, VLAE は CNN による抽象化によって階層化をなしていると考えるとまずは系列方向に 1 次元の Convolution を加えるのがシンプルなアイデアである. 提案手法である TCVLAE は Time Convolution を用いた VLAE に  $\beta$ -VAE 型の loss 関数を適応したものであり, ネットワークの詳細は 図 2 に示す. Ladder network の合流部は VLAE ではゲートを使っていたが今回は単純に和をとった. Time Convolution は filter size 3, stride 2, padding 1 を使い channel は常に入力の倍に増やすようにした.  $\beta = 100$  と

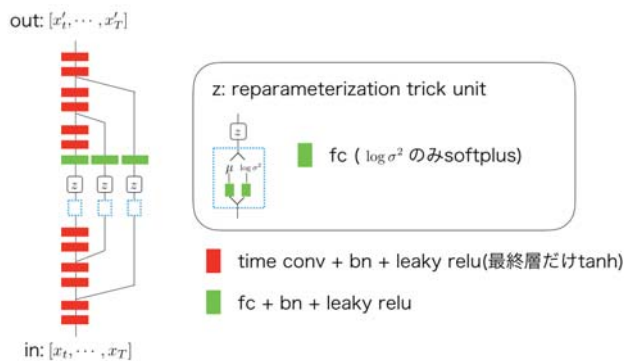


図 2: TCVLAE のネットワーク構造. 3 つある  $z$  それぞれが属するネットワークを右から ladder1, ladder2, ladder3 とした. 図中で各 ladder の  $z$  は 1 次元だが実際は高次元.

してカルバックライブラー情報量に対する  $C$  のシフトはスケジューリングして線形シフトさせるのではなく、直近 50 epoch のカルバックライブラー情報量誤差 (kl loss) の平均で決定した. kl loss は各 ladder ごとに計算し,  $C$  のシフトも各 ladder ごとに計算した.

### 3.2 Seq2seq model

Convolution を用いるモデルにはいくつか問題点がある. 1 つは Convolution window size 以上の系列の関係を結果に反映させることができないという点である. この問題を解決した時間方向の Convolution を用いるモデルとして quasi-recurrent neural networks (QRNN) がある. このモデルは LSTM や GRU などと比較して学習が速く精度も LSTM と同程度であることが実験的に確認されている [Bradbury 17]. TCVLAE のもう一つの問題としては, 固定長の系列入力しか扱うことができない点である. これについては系列データを扱う際によく用いられている seq2seq model [Sutskever 14] を用いることで可変長なデータに対しても扱うことができるようにする. QRNN は seq2seq model を扱うことができる. seq2seq モデルでは潜在変数  $z$  に encode する encoder  $p(z|x_1, \dots, x_T)$  と潜在変数  $z$  から系列データを生成する decoder  $p(x_t|z, x_{t-1})$  から成り立つ. Decoder での入力として  $z$  と  $x_{t-1}$  は和を取った.

Seq2Seq タイプの TCVLAE では学習時の decoder の入力は decoder の出力を再帰的に入力するのではなく, 正解データを入力として与えた (decoder の最初の時刻の入力は 0 を用いた). 学習済みモデルでの生成時 ( $z \rightarrow x$ ) では再帰的に 1 時刻前の decoder の出力を decoder の入力とした (decoder の最初の時刻の入力は 0 を用いた). action の低次元表現,

## 4. Experiment

### 4.1 setup

この章では TCVLAE と TCVLAE+seq2seq(QRNN) 2 つの結果について紹介する. 学習データは 2 次元上で階層的な運動を模擬した系列データを用いた. 2 次元座標 (0, 0) から 2 つのターゲット (-0.1, 1) と (0.1, 1) に点が移動する系列データになっており, 2 つのターゲットへの軌道にはバリエーションがあるデータ 図 3 である. 今回 どちらの target に向かっているのかという抽象的な意味と, 外側に膨らむ軌道か内側に膨らむ軌道かという具体的な意味を系列データから分離することを目的とする.

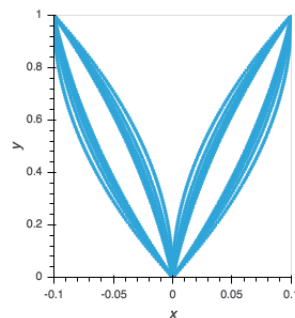


図 3: 2 つのターゲットに複数の軌道を描きながら点が飛んでいく系列データ

### 4.2 TCVLAE における階層性の効果の比較

学習済みの TCVLAE に対して  $z$  を入力して  $x_1, x_2, \dots, x_T$  を生成した結果を図 4 の上段に示す. どちらのターゲットを指すかという抽象的な意味が抽出されており, 下の図の  $z$  には外側に膨らむか内側に膨らむ軌道を選ぶかという具体的な軌道の違いの意味が抽出されている. ladder は 3 つあり, 各 ladder の  $z$  の次元はそれぞれ [8, 4, 2] 次元である. 入出力に近い位置から分岐した ladder から順番に ladder1, ladder2, ladder3 とした. ladder3 内の 1 つの  $z$  にターゲット選択の意味が抽出され, ladder2 内の 1 つの  $z$  には具体的な軌道の膨らみの意味が抽出され, その他の  $z$  には意味は現れなかった ( $z$  を変化しても  $x$  に変化が現れなかった). 次に ladder の効果を確かめるために ladder なしの TCVLAE を図 4 の下段示す. ladder ありと比較すると ladder なしではターゲットの選択の意味は獲得できているが, 軌道の膨らみの意味を明確に獲得することができなかった. よって今回のデータセットに関しては ladder を入れたことで系列データの意味の分離に効果があることが実験的に確かめられた. ladder なしのネットワークは encode と decode 層にそれぞれ Time Convolution を 4 層使い  $z$  の次元は 4 次元を用いた ( $z$  の次元を増やしても結果は良くなかったため).

### 4.3 TCVLAE + seq2seq

学習済みの TCVLAE+seq2seq に対して  $z$  を入力して再帰的に  $x_1, x_2, \dots, x_T$  を生成した結果を図 5 に示す. TCVLAE+seq2seq でも系列の意味の分離が可能であることがわかる. ただし外側への軌道の膨らみを綺麗に獲得することはできなかった. まだ seq2seq モデルでは ladder1 にすべての情報が集まり ladder の恩恵を得ることができなかった.

## 5. Conclusion and Future work

系列データにたいして Disentangle な表現の獲得を目指すモデルとして TCVLAE と TCVLAE+seq2seq を提案した. 単純なデータに対しては TCVLAE を用いることで系列データから Disentangle な表現の獲得が可能であることを実験的に示した. 今後はエキスパートデータが学習済みの強化学習モデルを用いて ( $s, a$ ) をサンプリングして TCVLAE から Disentangle な表現として抽象的な action  $z$  を獲得して,  $p(z_T|s_t, \dots, T)$  を学習する強化学習を作り, 既存手法である  $p(a_T|s_t, \dots, T)$  と学習効率や抽象的な action を学習したことによる転移学習の性能を比較したい.

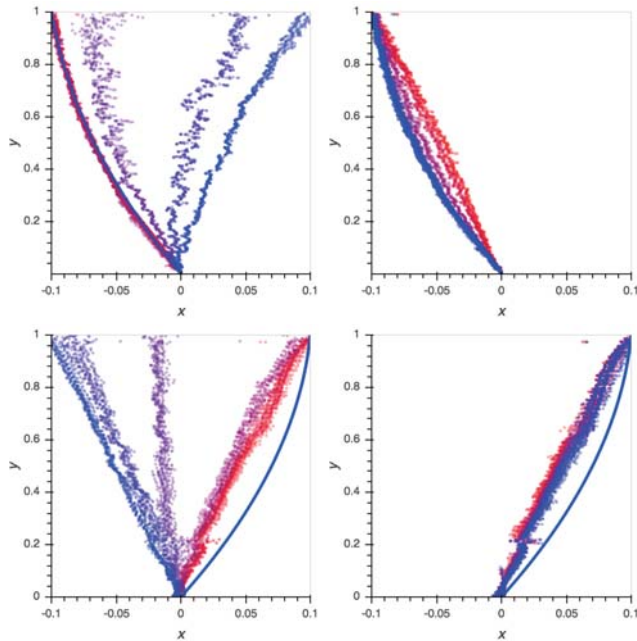


図 4: TCVLAE に対して、青い実線でプロットされている系列データから encode された  $z$  を用いて、各  $z$  を変化させて decode した  $x_t$  をプロットした図。ただし  $z$  の値を変化させる時、他の  $z$  の値は encode した値で固定している。色の違いは  $z$  の値の違い。上段は ladder ありの TCVLAE の結果で、下段は ladder が無い TCVLAE の結果。左上:ladder 3 の  $z_1$  を  $[-2, 2]$  に変化させた時に生成された系列データ。右上: ladder 2 の  $z_2$  を  $[-2, 2]$  に変化させた時に生成された系列データ。左下:  $z_1$  を  $[-2, 2]$  に変化させた時に生成された系列データ。右下: $z_4$  を  $[-2, 2]$  に変化させた時に生成された系列データ。

## 参考文献

- [Achille 17] Achille, A. and Soatto, S.: Information Dropout: learning optimal representations through noise, in *ICLR* (2017)
- [Bradbury 17] Bradbury, J., Merity, S., Xiong, C., and Socher, R.: QUASI-RECURRENT NEURAL NETWORKS, in *ICLR* (2017)
- [Burgess 17] Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A.: Understanding disentangling in  $\beta$ -VAE, in *NIPS* (2017)
- [Chen 16] Chen, X., Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., and Abbeel, P.: InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, in *NIPS* (2016)
- [Chung 14] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling, in *NIPS* (2014)
- [Durugkar 16] Durugkar, I. P., Dernbach, S., Rosenbaum, C., and Mahadevan, S.: Deep Reinforcement Learning with Macro-Actions, [*arXiv:1606.04615 [cs.LG]*] <https://arxiv.org/abs/1606.04615> (2016)

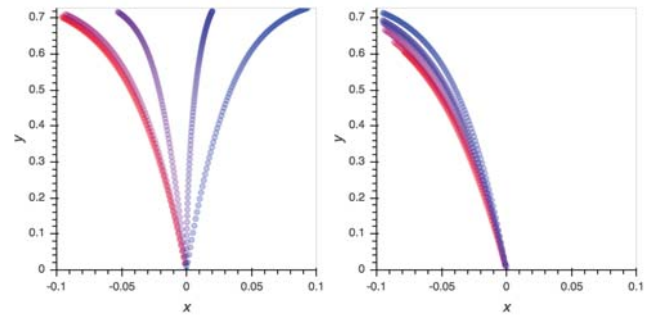


図 5: TCVLAE+seq2seq にたいして各  $z$  を変化させて decode した  $x_t$  をプロットした図。左: ladder1 の  $z_1$  を変化させた結果。右: ladder1 の  $z_2$  を変化させた結果

- [Goodfellow 14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y.: Generative Adversarial Nets, in *NIPS* (2014)
- [Higgins 14] Higgins, I., Matthey, L., Xavier Glorot, A. P., Uria, B., Blundell, C., Mohamed, S., and Lerchner, A.: Early Visual Concept Learning with Unsupervised Deep Learning, [*arXiv:1606.05579 [stat.ML]*] <https://arxiv.org/abs/1606.05579> (2014)
- [Higgins 17] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A.: beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, in *ICLR* (2017)
- [Hochreiter 97] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Comput.* (1997)
- [Kingma 14] Kingma, D. P. and Welling, M.: Auto-Encoding Variational Bayes, in *ICLR* (2014), <https://arxiv.org/abs/1312.6114>
- [Mnih 15] Mnih, V., Kavukcuoglu, K., et al.: Human-level control through deep reinforcement learning, No. 7540, pp. 529–533 (2015)
- [Sharma 17] Sharma, S., Lakshminarayanan, A. S., and Ravindran, B.: Learning to Repeat: Fine Grained Action Repetition for Deep Reinforcement Learning, in *ICLR* (2017)
- [Sutskever 14] Sutskever, I., Vinyals, O., and Le, Q. V.: Sequence to Sequence Learning with Neural Networks, in *NIPS* (2014)
- [Vezhnevets 16] Vezhnevets, A., Mnih, V., Osindero, S., Graves, A., Vinyals, O., Agapiou, J., and kavukcuoglu, k.: Strategic Attentive Writer for Learning Macro-Actions, in *NIPS* (2016)
- [Zhao 17] Zhao, S., Song, J., and Ermon, S.: Learning Hierarchical Features from Deep Generative Models, in *ICML* (2017)