夏季レポート3

情報科学類 3 年 江畑 拓哉 (201611350)

1 Sequence to Better Sequence: Continuous Revision of Combinatorial Structures　の翻訳　続き

1.1 Variational Autoencoder

X, Z の関係における近似推論のために、Variational Autoencoder モデルを利用する。我々の用いる VAE において、シーケンスの生成モデルは、尤度関数 $p_{\boldsymbol{D}}(x|z)$ と組み合わされた潜在値 z に対する我々の事前確率に対して指定される。この尤度関数は、z における任意のシーケンス x の尤度を評価するための decoder network $\mathcal{D}$ が出力するものである。任意のシーケンス x が与えられたとき、encoder network $\mathcal{E}$ は、潜在値 $p(z|x) \propto p_{\boldsymbol{D}}(x|z)p_{\boldsymbol{Z}}(z)$ の真の事後確率の変分近似 (Variational approximation) $q_{\boldsymbol{E}}(z|x)$ を出力する。なおこの変分近似には、Kingma & Welling よ Bowman らによって提唱されているように、対角共分散 (diagonal coviarance) を持つ変分族 (variational family) $q_{\boldsymbol{E}} = N(\mu_{z|x}, \Sigma_{z|x})$ を利用する。

我々の変形の手法では、シーケンスを潜在値 z の最大事後 (MAP) 構成 (the maximum a posteriori configuration) にマッピングする符号化手順を採用している。(これは encoder network $\mathcal{E}$ によって推定される)

$\mathcal{E}, \mathcal{D}$ のパラメータは、訓練データにおけるそれぞれの観測に対する周辺尤度の下限を最大化する確率変分推論 (stochastic variational inference) を用いて学習される。

$$\begin{aligned} \log p_{\boldsymbol{X}} &\geqslant -[\mathcal{L}_{rec}(x) + \mathcal{L}_{pri}(x)] \\ \mathcal{L}_{rec}(x) &= -\mathbb{E}_{q_{\boldsymbol{E}}(z|x)}[\log p_{\boldsymbol{D}}(x|z)] \\ \mathcal{L}_{pri}(x) &= KL(q_{\boldsymbol{E}}(z|x)||p_{\boldsymbol{Z}}) \end{aligned} \tag{1}$$

$\sigma_{z|x} = diag(\Sigma_{z|x})$ と定義すると、$q_{\boldsymbol{E}}, q_{\boldsymbol{Z}}$ が対角ガウス分布であるとき、事前強制 (prior-enforcing)KL ダイバージェンスは異なる閉形表現 (closed form expression)(see. https://minus9d.hatenablog.com/entry/20130624/1372084229) を持つ。 $\mathcal{L}_{rec}$ 項 (すなわち decoder モデルの元での対数尤度) を再構築したものは、ただ一つの取り出されたモンテカルロ標本 $z \sim q_{\boldsymbol{E}}(z|x)$ より効率的に近似することができる。ニューラルネットワーク $\mathcal{E}, \mathcal{D}$ のパラメータに関して、我々のデータ $\mathcal{D}_n$ に対して変分加減を最適化するために、我々は誤差逆伝搬法と Kingma & Welling による再パラメータ化のトリック (see. https://arxiv.org/pdf/1312.6114.pdf) を用いて得られた、(2) の確率的勾配を用いる。

全体を通して、我々の encoder/decoder モデル $\mathcal{E}, \mathcal{D}$ は RNN である。RNN は各時間のステップ $t \in \{1, \dot{;} T\}$ において固定サイズの隠れ層の状態を示すベクトル $h_t \in \mathbb{R}^d$ が入力シーケンスの次の要素に基づいて更新されていくという、シーケンシャルなデータ $x = (s_1, \dot{;} s_T)$ に対するニューラルネットワークである。与えられた x に対して近似的な事後確率を生成するために、我々の encoder network $\mathcal{E}$ には RNN の最終的な隠れ層の状態を表すベクトルに対して以下の層を追加している。(パラメータとして、$\boldsymbol{W}, b$ を取っている)

$$\mu_{z|x} = \boldsymbol{W}_\mu h_T + b_\mu \in \mathbb{R}^d \sigma z|x = exp(-|\boldsymbol{W}_\sigma v + b_\sigma) v = ReLU(\boldsymbol{W}_v h_T + b_v) \tag{2}$$

$\sigma_{z|x} \in \mathbb{R}^d$ の (二乗された) 要素は、我々の近似された事後共分散 (approximate-posterior coviarance) $\Sigma_{z|x}$ の対角要素を形成する。

$\mathcal{L}_{pri}$ は $\sigma_{z|x} = \vec{1}$ で最小化され、encoding の分散が更に増えるとこれが悪化する可能性がある (我々の事後近似は Unimodal(単峰) である)) ため、我々の変分族 (variational family) の 1 を超える $\sigma_{z|x}$ の値を単純に考えることが出来ない。この制限を加えることは、より安定的な学習を促し、また、真の事後確率が分散 $\leqslant 1$ で単峰性 (see. 正規分布) に近づくように encoder, decoder が共進化することを助ける (encourage)。

シーケンスの尤度を評価するため、RNN である $\mathcal{D}$ を隠れ層の状態を表すベクトル h_t のみならず、以下の追加された出力も考慮する。

$$\pi_t = softmax(\boldsymbol{W}_\pi h_t + b_\pi) \tag{3}$$

それぞれのポジション t において、h_t を要約することで、$p(s_t|s_1, \ldots, s_{t-1})$ を予測する。 $p(s_1, \ldots s_T) = \Pi_{t=1}^T p(s_t|s_{t-1}, \ldots, s_1)$ という因数分解を用いることで、 $p_D(x|z) = \Pi_{t=1}^T \pi_t[s_t]$ を得ることができる。これは最初の隠れ層の状態を表すベクトル $h_0 = z$ と、$x = (s_1, \ldots, s_T)$ を $\mathcal{D}$ に与えることで計算される。与えられた潜在設定 z より、我々の変形は以下に示されるよりもっともらしい観測を通してでシーケンスを復号することで得られる。

$$D(z) = \underset{x \in \boldsymbol{X}}{\operatorname{argmax}}\, p_{\boldsymbol{D}}(x|z) \tag{4}$$

(5) を用いたよりもっともらしい復号は、組み合わせ問題それ自身であるが、 $p(x|z)$ の逐次因数分解を利用するビームサーチを用いることでより効率的に見つけることができる。$x^\star = \boldsymbol{D}(z) \in \boldsymbol{X}$ において、この $p_{\boldsymbol{X}}(x^\star)$ でも $p(z|x^\star)$ でもない探索を用いた復号戦略はとても小さいものである。

1.2 出力の構成的な予測

VAE のコンポーネントに加えて、我々は構造について出力予測を行うモデルを標準的な feed forward neural network $\mathcal{F}$ $(F : \mathbb{R}^d \to \mathbb{R})$ で作成し、これに fit させなければならない。我々の考える生成モデルは $F(x) = \mathbb{E}[Y|Z = z]$ である。 Z を経由して

$\mathbb{E}[Y|X=x] = \int \mathbb{F}(z) q_{\mathbb{E}}(z|x) dz$ を計算するよりも、一次のテイラー展開 $F(E(x))$ (この近似誤差は $\mathbb{F}$ がアフィン変換に近いほど縮小します) を用いるほうが良いです。この近似推論の条件付き期待値を正確に推定するために、我々は 損失関数に $\mathcal{E}$ と $\mathcal{F}$ を一緒に取って training します。(joint training)

$$\mathcal{L}_{mse}(x, y) = [y - \boldsymbol{F}(\boldsymbol{E}(x))]^2 \tag{5}$$

基礎となる条件付き関係を補足するのに十分な容量でネットワーク $\mathcal{E}$ $\mathcal{F}$ が指定されている場合は、 十分に大きなデータセット からネットワークパラメータを設定した後に、$\boldsymbol{F}(\boldsymbol{E}(x)) \approx \mathbb{E}[Y|X=x]$ を得る必要がある。($\mathbb{F}$ が非線形なマップ (non lenear map) であったとしても))

1.3 不変性を強制する

理論的には、z のいくつかの次元は結果 y にのみ関係し、複合されたシーケンス $\boldsymbol{D}(z)$ には何も影響を与えてない可能性がある。この種の潜在的な表現を学ぶ事は厄介であり、z に関する推論された y のその後の最適化は、実際には優れた修正されたシーケンスに繋がらない事があるからである。この問題を軽減するために、潜在値 $\boldsymbol{Z}$ の次元数 d が、正確な結果予測と VAE の再構成を生成するために必要な最低限の容量を超えないように注意する。つまりこの望ましくないシナリオを明示的に抑制するには、次のような損失を加えて neural network の training を補助する。

$$\mathcal{L}_{inv} = \mathbb{E}_{z \sin p_{\boldsymbol{Z}}}[\boldsymbol{F}(z) - \boldsymbol{F}(\boldsymbol{E}(\boldsymbol{D}(z)))]^2 \tag{6}$$

neural network のパラメータを損失に基づいて最適化する場合、 $\mathcal{E}, \mathcal{F}$ で単独にモンテカルロ法による勾配 (Monte-Carlo estimated gradients) のバックプロパゲーションをされた $\mathcal{D}$ のパラメータと右辺の修正された $\boldsymbol{F}(z)$ 項を扱う必要がある。 $\mathcal{L}_{inv}$ を 0 へ近づけることは、我々の出力予測が encoding-decoding のプロセスにおいて導入された変化に対して不変であることを保証する。

1.4 Joint training

このモデルに含まれるすべてのコンポーネントのパラメータ $(q_{\boldsymbol{E}}, p_{\boldsymbol{D}}, \boldsymbol{F})$ は 一貫したプロセスで学習される。 Training は $\boldsymbol{D}_n$ の例よりも以下の目的関数を最小化する SGD を適用することになる。

$$\mathcal{L}(x, y) = \mathcal{L}_{rec} + \lambda_{pri}\mathcal{L}_{pri} + \frac{\lambda_{mse}}{\sigma^2_{\boldsymbol{Y}}}\mathcal{L}_{mse} + \frac{\lambda_{inv}}{\sigma^2_{\boldsymbol{Y}}}\mathcal{L}_{inv} \tag{7}$$

ただし、 $\sigma^2_{\boldsymbol{Y}}$ は (経験的な) 出力の分散であり、 $\lambda \geqslant 0$ は全体的なフレームワークの効果を最大限に引き出すための、それぞれの目的に対する重み付のために選ばれた定数であ

る。最初に $\lambda_{mse} = \lambda_{inv} = 0$ とすることで、オプションとして別々のコーパスの入力をラベルなしのものとして同一の入力として教師なしな VAE の学習として使うことができる。これは教師なしの事前学習がうまくいくことが Kiros ら や Erhan らによって示されているためである。

実際に、以下の上手く行く学習戦略を見つけることが出来た。それは次の各ステップの中で多数のミニバッチの SGD の更新 (通常は 10 ~ 30 epoch)) を適用する戦略である。

1.4.1 Step 1

$\lambda_{inv} = \lambda_{pri} = 0$ 、つまり λ_{rec} と λ_{mse} について training を始める。 λ_{mse} について適切な値を指定したにもかかわらず、この joint trainig による最適化を通して $\mathcal{L}_{rec}$ と $\mathcal{L}_{mse}$ の両方が極小な正の値に向かうことがわかった。(それぞれの目的に対して個別に training することで検証された)

1.4.2 Step 2

Bowman らによって提案されたシグモイドアニーリングスケジューリング (sigmoid annealing schedule) に従って、λ_{pri} を 0 から 1 へを大きくする。これは 変分 seq2seq モデルが単に z の encoding を無視しないことを保証するためである。(公式の変分下限は $\lambda_{pri} = 1$ で達成されることに注意しなければならない)

1.4.3 Step 3

$z \sim p_{\boldsymbol{Z}}$ のモンテカルロサンプルを通して平均して $\mathcal{L}_{inv}$ が小さくなるまで λ_{inv} を増加させていく。ここで、 $p_{\boldsymbol{D}}$ は $\mathcal{L}_{inv}$ に関して定数として扱われ、SGD で使用されるミニバッチは、(シーケンス、出力) を対として $\mathcal{L}_{inv}$ を推定するために同数のモンテカルロサンプルを含むように選択される。

1.5 変形の提案

前述の訓練手順は計算集約的 (computationally intensive) であるが、一度学習を行うことで、 neural network を効率的な推論のために活用する事ができる。ユーザ指定な定数 $\alpha > 0$ と修正対象のシーケンス x_0 が与えられた際に、次の手順を経て変形されたシーケンス x^* を提案する。

Revise Algorithm

Input: シーケンス $x_0 \in \boldsymbol{X}$, 定数 $\alpha \in (0, |2\pi\Sigma_{z|x_0}|^{-1/2})$
Output: 変形されたシーケンス $x^* \in \boldsymbol{X}$
1) $\mathcal{E}$ を用いて $q_{\boldsymbol{E}}(z|x_0)$
2) $C_{x_0} = \{z \in \mathbb{R}^d : q_{\boldsymbol{E}}(z|x_0) \geqslant \alpha\}$ とする
3) $z^* = \text{argmax}_{z \in C_{x_0}} \boldsymbol{F}(z)$ を見つける　(勾配降下法)
4) $x^* = \boldsymbol{D}(z^*)$ を返す (ビームサーチ)

直感的には、レベルを決定する制約 $C_{x_0} \subseteq \mathbb{R}^d$ は x^* を decode する潜在的な構成である z^* が、 x_0 の生成に関与する潜在的な特性に近い特性を持っているということを保証する。 x_0 と x^* が潜在因子を共有していると仮定すると、これらのシーケンスは生成モデルに従って根本的に近いと言える。
(α のどのような値に対しても) $z^* = \boldsymbol{E}(x_0)$ は常に $z \in \boldsymbol{C}_{x_0}$ に対する潜在因子の最適化に関する実現可能な解であることに注意しなければならない。(つまり自然なシーケンスを生み出す空間上に位置しているということ) ただ、この制約付き最適化は事後ガウス近似 (Gausian approximate-posterior) の仮定の元では、 C_{x_0} の概形が単純な $\boldsymbol{E}(x_0)$ を中心とする楕円体であるため、簡単に行うことができる。

変形を行う手続きである Step 3 において z^* を見つけるために、初期値を $z = \boldsymbol{E}(x_0)$ として勾配降下法を用いる。これはもし $\boldsymbol{F}$ が単純な feed forward network によってパラメーター化されていると 、局所最適をすぐに調べることができる。

$E(x_0)$ で検索を始めると、ガウス分布を持つ $q_{\boldsymbol{E}}$ のような単峰性の事後近似のためには最も良いと考えられます。実行可能な空間 C_{x_0} においてすべての反復が残っていることを確かめるために、代わりにペナルティ化された目的関数 (penalized objective) $\boldsymbol{F}(z) + \mu \cdot \boldsymbol{J}(z)$ に関して勾配ステップを取る。ただし、

$$\begin{aligned} \boldsymbol{J}(z) &= log[\boldsymbol{K} - (z - \boldsymbol{E}(x_0))^T \Sigma_{z|x_0}^{-1} (z - \boldsymbol{E}(x_0))] \\ K &= -2log[(2\pi)^{d/2} |\Sigma_{z|x}|^{1/2} \alpha] \end{aligned} \tag{8}$$

そして、 $0 < \mu \ll 1$ は 0 に向かって徐々に減少し、最適化が $\boldsymbol{C}_{x_0}$ の境界に近づくことを保証する。結果として得られる変形の品質の観点から、この log バリア関数 (log barrier function) は、投影勾配 (projected gradient) や Franke-Wolfe アルゴリズムなどの制約付き最適化問題のための他の標準的な一次技法 (first-order techniques) よりも優れていることがわかった。

原則として、我々の変形方法は、 Sutskever らや Cho らの seq2seq モデルのようなシーケンスのための従来の決定論的な autoencoder の潜在的表現に作用することができる。しかしながら、 VAE は数多くの実用上の利点があり、そのうちのいくつかには、より一貫性のある文章を生成できると言う点が Bowman らによって強調されている。VAE の事後における不確実性は、ネットワークが潜在的な分布のサポートを、training の例を全体的にスムーズに広げることができることを示している。大賞的に、伝統的な autoencoder の元での潜在値の領域の中央部には、(example にはマッピングされていない) 穴が含まれている可能性があり、 $z*$ の最適化でこれらを避けることは簡単ではない。さらに、さあ遺書のシーケンスがすでに適切でないように構成されていた場合 ($\boldsymbol{D}(\boldsymbol{E}(x_0)) \geq x_0$) の貧弱な変形を避けるように設計された、後述する S1 の decoder の適応型変形について紹介する。

2 日本語を用いた実験

2.1 問題点

本手法を日本語において実験した。但しいくつかの問題点があるため、それをここに列挙する。

- データ数が極端に少ない (元のスタイル 300 文 + 目的のスタイル 300 文) のため、英語での実験と単純に比較することは難しいと考えられる。

- データは極端に調整されたものであるため、一般的に日本語の自然言語処理で用いられているようなノイズの多いデータを用いた実験と比較することは出来ない。

- このモデルでは非平行なデータを対象としているが、先述の調節のため非平行ではなく平行なデータで実験している。そのためこの手法で提案されている自然な短文の空間が小さくなってしまっている (この空間を作る部分に関しては、元の実験に比べデータ数が半分になっていると考えられる。) 可能性がある。

2.2 実験

データは以下のレポジトリのものを用いた。(https://github.com/MokkeMeguru/st-data)

base.csv が です・ます調 な書き言葉に近い短文集であり、 styled.csv が ね・だ調 な話し言葉に近い短文集である。今回は base -¿ styled への変換を “短文を単語ごとに分割して読みに変換して” 実験した。

epoch 数は 100 + 2 * 20 * 100 + more = 4100+ であり、一般的なそれよりもかなり多い。これは本手法の複数の損失関数を別々に訓練を行う必要があるため、仕方がないものである。また、日本語であること (英語のそれに比べて活用や漢字・かなの問題があるため、語彙数が増えやすい) やデータ数が少ないことなどが順調な訓練の妨げになっているようである。

実際、epoch 数を半分にした実験では学習が上手く行かず結果を得ることが出来なかった。

学習時間に関しては容易に考えることが出来ない。なぜなら学習過程においてハイパーパラメータを常に監視し続け、更に直感でパラメータ調節を行う必要があるため、同じデータを用いて学習を行うとしても、必ず学習が最後まで成功するとはわからず、学習時間もまちまちである。今回は短く見積もっても合計で 24 時間かかった。GPU には Quadro K2200 を用いた。

2.3 実験結果

最低限の優秀な実験結果を得ることが出来たと考えられる。
以下に上手く変換できた例と上手く変換できなかった例、判断が難しい例を挙げる。
尚実験のレポジトリは https://github.com/MokkeMeguru/seq2bseq にある。
また未知語彙を含む短文に関しては学習を行っていないため、結果を得ることが出来なかった。

2.3.1 成功例

殆ど学習に用いたデータに近いが、かなり理想的な結果が得られたと考えられる。このような結果が普通に得られるようにデータセットを用意したいと考えている。

```
test_sentence('早く寝たい。', model)
iter=0 obj=[[0.]]
iter=100 obj=[[0.7482172]]
iter=200 obj=[[0.71754885]]
iter=300 obj=[[0.85257566]]
iter=400 obj=[[1.1158444]]
iter=500 obj=[[1.121194]]
iter=600 obj=[[1.3258313]]
iter=700 obj=[[0.90995586]]
Elliptical constraint value:[[0.99999994]]
['ハヤク', 'ネ', 'タイ', '。'] -> ['ハヤク', 'ネ', 'タ', 'ホウ', 'ガ', 'ヨイ
', 'ネ', '。']
test_sentence('それは良い。', model)
iter=0 obj=[[0.]]
iter=100 obj=[[0.31412524]]
Elliptical constraint value:[[0.9980601]]
['ソレ', 'ハ', 'ヨイ', '。'] -> ['ソレ', 'ハ', 'ヨイ', 'ネ', '。']
test_sentence('応援する。', model)
iter=0 obj=[[0.]]
iter=100 obj=[[0.65189254]]
Elliptical constraint value:[[0.9958178]]
['オウエン', 'スル', '。'] -> ['オウエン', 'シ', 'テル', '。']
test_sentence('鳥肌がたった。', model)
iter=0 obj=[[0.]]
iter=100 obj=[[0.37873054]]
Elliptical constraint value:[[0.99807376]]
['トリハダ', 'ガ', 'タッ', 'タ', '。'] -> ['トリハダ', 'ガ', 'タッ', 'タ', '
ネ', '。']
```

2.3.2 失敗例

間違った方向へ文章がねじ曲がってしまったパラメータ調整は最善を尽くしているため、データセットの量を増やすなどして上手く行くかを実験したい。

```
test_sentence('何か不安だなぁ。', model)
iter=0 obj=[[0.]]
iter=100 obj=[[0.33364055]]
Elliptical constraint value:[[0.99713546]]
['ナニ', 'カ', 'フアン', 'ダ', 'ナァ', '。'] -> ['ナニ', 'カ', 'ノム', '?']
```

2.3.3 判断が難しい例

変換こそ出来ていないものの意味そのものは保持できているパターン、変換ではなく補完を行っている可能性があるパターンなどがある。

```
test_sentence('今日は寒かった。', model)
iter=0 obj=[[0.]]
Elliptical constraint value:[[0.9977083]]
['キョウ', 'ハ', 'サムカッ', 'タ', '。'] -> ['キョウ', 'ハ', 'サムカッ', 'タ
', '。']
test_sentence('今日は寒かった', model)
iter=0 obj=[[0.]]
iter=100 obj=[[0.7380588]]
Elliptical constraint value:[[0.9816849]]
['キョウ', 'ハ', 'サムカッ', 'タ'] -> ['キョウ', 'ハ', 'サムカッ', 'タ', 'ネ
', '。']
test_sentence('夕飯は？', model)
iter=0 obj=[[0.]]
iter=100 obj=[[0.36785245]]
Elliptical constraint value:[[0.9974266]]
['ユウハン', 'ハ', '?'] -> ['ユウハン', 'ハ', 'ドウ', 'シヨ', 'ウ', 'カ', '?
']
```

2.4 今後の展望

データセットをより増やす必要がある。しかし精錬されたデータですら中々うまく行かないかなりデリケートな学習手法のため、ノイズの多いデータを用いることは出来ないだろう。

未知語彙を含ませるため　Unsupervised Machine Translation Using Monolingual Corpora Only (https://arxiv.org/abs/1711.00043) にあるように短文の一部を未知語彙として実験を行いたい。またこれを行う副作用として、データセットを増やすことが出来ると考えられるため (ランダムに単語を未知語彙に置換するため、一つの短文に対して複数のデータを得ることが出来る)、それも含めて比較実験を行いたいと考えている。(参考: 未知語

彙を用いた変換 (copy mechanism) https://www.slideshare.net/ToshiakiNakazawa/ibis2017)