

[要旨]

LSTMと強化学習を用いてコード進行を学習させ、コードを生成するモデルを作成した。一度高いrewardを見つけると、連続した高得点のコードばかりが生成されてしまうので、強化学習が終わる直前のモデルが生成AIとしては適していると考えられる。

[研究背景・目的]

近年、ChatGPTなどの生成AIは急速に進化し、多くの人々から注目を集めている。一方で、生成AIは著作権侵害などの深刻な問題を抱えている。強化学習による方法で曲の作り方を自分自身でAIに教えることで解決できないだろうか。

[研究方法]

C++言語でLibTorchライブラリを用いて、以下の手法で学習を行うプログラムを実装した。

①Simple RNN (Recurrent Neural Network)とLSTM (Long Short Term Memory) の実装

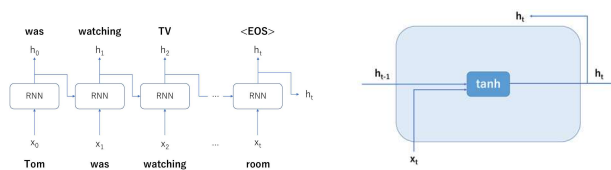


図1 RNNの構造(1つ未来を予測するネットワーク(左))、Simple RNNの構造(右)と式

時刻 $t + 1$ では、1つ前のRNNの出力 h_t と、入力データ x_{t+1} を入力し、 h_{t+1} が出力される。

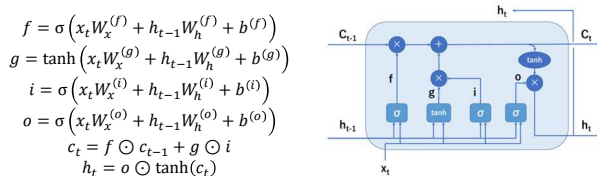


図2 LSTMの式(左)と構造(右)

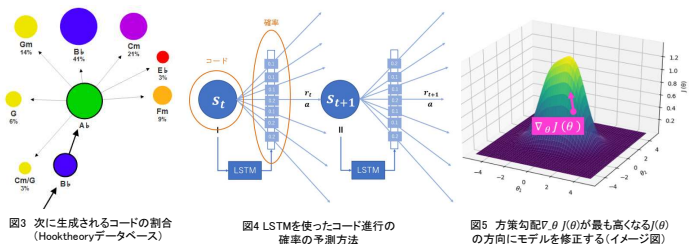
LSTMでは、勾配消失問題を解決するため3つのゲートと記憶セルを追加している。

②Penn Treebankデータセットの予測

The Wall Street Journalなどの原稿から作成された、無料で利用できる英単語912344語のコーパスを用いて、Simple RNNとLSTMのモデルを学習させた。レイヤーの数や隠れ層の大きさ、ドロップアウトの割合を変化させた時の性能を比較した。

③コード進行の予測

LSTMと強化学習を用いて、Hooktheoryデータベースのコード進行の確率を得点としてREINFORCE法で学習させ、コードを生成させた。強化学習では、方策と言う、状態sの時にaという行動をとる確率を用いた。未来から報酬を計算することで、収益 $G(\tau)$ を求めることができる。



$$= -\mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^{\tau} \left(G(\tau) \log \pi_{\theta}(A_t | S_t) + \beta \sum_{a \in \mathcal{A}} \pi_{\theta}(a | S_t) \log \pi_{\theta}(a | S_t) \right) \right]$$

$$L(\theta) = -\mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^{\tau} G(\tau) \log \pi_{\theta}(A_t | S_t) \right] - \beta \mathbb{E}_{\tau \sim \pi_{\theta}} [H(\pi_{\theta})]$$

方策勾配が簡単に高くなってしまふのを防ぐために、エントロピー正則化という、生成されるコードの確率の乱雑さを高めることを行った。

エントロピー正則化の β の値を変化させてコードの割合を比較した。

[結果・考察]

①Penn Treebankデータセットの予測

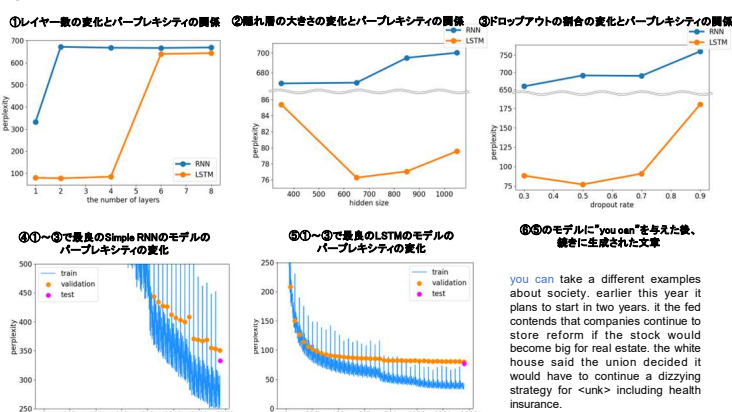


図6 RNNとLSTMの条件を変えた時のパープレキシティの変化と⑤のモデルから生成された文章

パープレキシティとは、モデルが絞り込んだ“次に出現する単語の候補の数”を表していて、値が低い方が性能が良いことを示している。

図6の①～⑤から、LSTMの最良のモデルは、レイヤー数2、隠れ層の大きさ650、ドロップアウトの割合は0.5の時で、パープレキシティの値がSimple RNNの最良のモデルよりも圧倒的に低いため、LSTMの性能の高さが確認できた。

②コード進行の予測

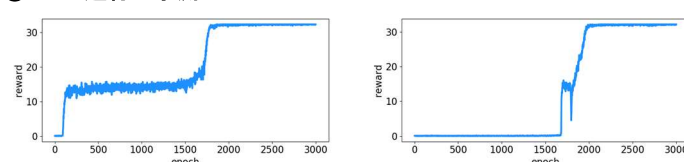


図7 コード進行を学習させた時のrewardの変化(左: $\beta = 0.00$, 右: $\beta = 0.15$)

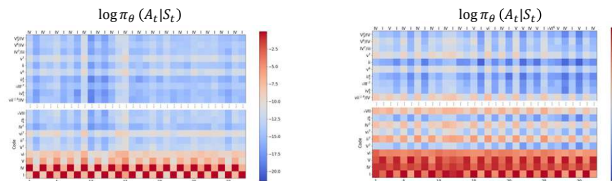


図8 コード進行を1900epoch学習させた時のコードの生成される割合の変化(左: $\beta = 0.00$, 右: $\beta = 0.15$)



図9 生成されたコード進行

学習が進むと、連続した高得点の同じ組み合わせのコードが生成されやすくなった(図7)。そのため、強化学習が終わる直前のepochのモデルが生成AIとしては適していると考えられる。エントロピー正則化の β の値が0.00よりも0.15の方が、多様性が見られ、生成AIとして意味のあるコード進行が得られた。図8から、よく使われる基本的コードの確率が特に高く、珍しいコードがたまに生成されるようになったと考えられる。

[結論・今後の展望]

LSTMと強化学習を組み合わせることで、多大な学習データを必要としない生成モデルを作成できた。一度高いrewardを見つけると、連続した高得点のコードばかりが生成されてしまうので、強化学習が終わる直前のepochのモデルが生成AIとしては適していると考えられる。今後は、同じ仕組みを用いてメロディーを生成するモデルを作成したい。

[引用文献・参考文献]

斎藤 康毅 (2018). 「ゼロから作るDeep Learning ②」. O'Reilly Japan.
 Tomas Mikolov et al. Empirical Evaluation and Combination of Advanced Language Modeling Techniques, Interspeech, 2011.
 David Foster (2020). 「生成Deep Learning」(松田晃一訳). O'Reilly Japan.
 Amit Kohli. (2014). Musical-chord-progressions. GitHub. <https://github.com/DataStrategist/Musical-chord-progressions>.