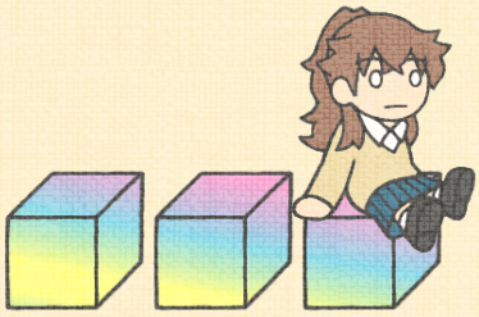
NeurIPS 2021 にみる

最近のニューラル系列モデルへの発見・工夫・理解

未完成ドラフト

この本は未完成ドラフトです。
目次の内容をかくことを目指しましたが
第 2 節以降が完成していません。



目次

まえがき

本書は機械学習の国際会議 NeurIPS 2021 で発表された論文から、(時) 系列データを処理するためのニューラルネットワークモデル——長いのでニューラル系列モデルとよびます——に関連する研究を俯瞰しようとしたものです。が、網羅的でも排他的でもありません。また、論文の内容に関する記述は著者の理解であることに留意ください。著者の誤りは著者に帰属します。

本書の内容についてお気付きの点がありましたら、大変お手数ですがこの原稿があるリポジトリの Issues、または著者ブログのコメント欄までお知らせください。著者ブログへのコメントはただちには公開されません。非公開希望の方はその旨をお知らせください。非公開希望であって返信が必要な場合はご連絡先の明記をお願いいたします。

リポジトリ https://github.com/CookieBox26/notes/
著者ブログ https://cookie-box.hatenablog.com/
本書に関連している記事は以下です。
メモ前編 https://cookie-box.hatenablog.com/entry/2021/11/28/191332
メモ後編 https://cookie-box.hatenablog.com/entry/2021/12/23/124713

登場人物紹介

この人はベイズ統計部の部長です。1 年生です。とある目的のためにベイズ統計部を立ち上げ、統計や機械学習を勉強しています。ベイズ統計部には部長と副部長しかいません。姉が 2 人います。

この人はベイズ統計部の副部長です。2 年生です。海外からの編入生でラクロス部に入部しようとしていましたが、部長に勧誘されてベイズ統計部に入部しました。数学が得意ですがなぜか著者を超える数学力が出せません。

第 1 節　NeurIPS 2021 のニューラル系列モデルを眺める

(時) 系列データを処理するためのニューラルネットワークモデルの動向を知るために、NeurIPS 2021 で発表された研究をみていきましょう。予稿サイトをみると、NeurIPS 2021 で発表された論文の総数は……2334 本[a]!?

[a] 2021 年 11 月 28 日の https://proceedings.neurips.cc/paper/2021 のリンク数に基づく。

あ、あまりに多いので絞り込みましょう。タイトルに time series, sequential, rnn, recurrent, transformer, attention, state space のいずれかを含む論文は……それでも 155 本……。ただこの絞り込みだと画像認識, GAN, 強化学習の研究も多そうですね。それらも興味深いですが、(時) 系列に関する研究を優先するべく断腸の思いでとばしていきましょう……。

とばしても 39 本……多いですね……こう多いと何が何だかわかりません。モデルの切り口でグループ分けしてみましょう。さらにグループ内をアブストラクトからの私の理解の範囲で区分してみましょう。

——作業後。

まず、最大勢力は Transformer ですね、検索語に含めたのでヒットするのは当然ですが、多くを占めます。特に**「セルフアテンションの計算量に対処する」**は昨年以前から引き続いて人気 (?) なテーマであるようです。このサブグループは後で改めてメモしましょう。全体としては、「Transformer の性質を明らかにしようとしたもの」「訓練方法を工夫したもの」「使い方を工夫したもの」「アーキテクチャ自体を工夫したもの」「新しい用途に利用したもの」といった区分をしてみました。無論、この区分も解釈の一例であることに留意ください。

グループ「Transformer」

- Transformer の性質を理解する。
  - 行列分解でパラメータを 10 倍削減しても性能が出ると示す [2]。
  - セルフアテンションを生物学的な記憶モデルと解釈する [16]。
- Transformer の訓練方法を工夫する。
  - ヘッド間で Q, K の分布を一致させる正則化をする [15]。
- Transformer の使い方を工夫する。
  - グリッド分割をさらにグリッド分割する (Vision Transformer) [17]。
  - 機械的にプレ処理 (トレンド-季節性分解) をする [23]。
  - 状態空間モデルと組み合わせて時系列の長期予測等をする [27]。
  - データ間でセルフ (セルフ?) アテンションする [36]。
- Transformer のアーキテクチャを工夫する。
  - **セルフアテンションの計算量に対処する** [3] [4] [7] [11] [19] [20] [24] [25] [37]。
  - その他 Transformer のアーキテクチャを再考する。
    - 言語処理に適した構造を探索する [8]
    - セルフアテンションの代わりにゲート付 MLP にする [10]。
    - 時系列予測のために位相的アテンションを導入する [29]。
    - Softmax しないセルフアテンションで偏微分方程式を解く [30]。
- Transformer を新しい用途に利用する。
  - Transformer でガウス過程モデル適用時のカーネルを同定する [12]。

次の勢力は RNN ですね。**「RNN を理論的に理解する」**という研究が割にみられるように感じられます。理論解析が進めばどのような系列データにどのようなニューラルアーキテクチャを用いるべきかにつながるのでしょうか……?? もちろん、「RNN を工夫する」といったより目先の実用を見据えた動機でありそうな研究もみうけられます。このグループの 3 研究の趣旨はそれぞれ「表現力を上げたい」「訓練時間を短くしたい」「レジームを最大限に利用したい」といったところでしょうか。

グループ「RNN」

- **RNN を理論的に理解する。**
  - RNN がある再生核ヒルベルト空間におけるカーネル法であると示す [5]。
  - 隠れ状態にノイズ添加して訓練すると正則化されると示す [6]。
  - スイッチング線形動的システムで RNN をリバースエンジニアリングする [18]。
  - RNN が学習できると保証される関数の制約を撤廃する [21] [22]。
  - タスクの解空間を構造化して RNN の性質を調べる [31]。
  - 勾配消失/爆発しない RNN のサブセットを突き止める [32]。

  - 成長するメモリ付き固定精度 RNN がチューリング完全であると示す [35]。
- RNN を工夫する。
  - RNN 自体が時間変化できるようにする [9]。
  - ドロップアウトを活用して LSTM の訓練を効率化する [28]。
  - 継続期間も明示的に利用するレジームスイッチングモデルを実現する [39]。

後はその他とでもしましょうか。内容は色々なんですが……例えば Transformer と CNN の比較がありました。微分方程式で記述されるようなシステムを表現するモデルの話題も複数。それ以降は特に時系列といった研究ですね。モデルへの工夫がメインとなる研究は既に Transformer, RNN のグループに分類しましたから、ここに分類されてくるのはプレ処理/ポスト処理といった向きのもののようです。異常検知やクラスタリングについては予測といったものではないですが、たまたま目に付いて興味を引いたので選びました。

その他

- 系列モデルを比較する。
  - Transformer と CNN のロバスト性を比較する [34]。
- 微分方程式で記述されるシステムをニューラルネットで実現する。
  - 線形時不変連続時間システムをニューラルネットで実現する [1]。
  - 連立微分方程式システムをベイズフィルタで解く [13]。
- 時系列モデルの性能を向上させる汎用的なプレ処理/ポスト処理を導入する。
  - 機械的に汎用的なプレ処理 (成分クラスタリング) をする [14]。
  - 誤差の自己相関を調整する [38]。
- 時系列の異常検知を工夫する。
  - 時系列のオンライン異常検知の偽陽性率を制御する [33]。
- 時系列のクラスタリングを工夫する。
  - 時系列を生成する混合分布を推定するためのコアセットを構築する [26]。

最後に**「セルフアテンションの計算量に対処する」**を回収しましょう。そもそも Transformer の計算量が取り沙汰されるのは $\mathrm{Softmax}\left(QK^\top/\sqrt{d}\right) \in \mathbb{R}^{N\times N}$ を求めるのに系列長 N に対して $\mathcal{O}(N^2)$ の計算量がかかるためですが、$\mathcal{O}(N^2)$ を回避するために、以下のようなアプローチが取られているようです。スパース化、低ランク近似自体はこれまでも計算量削減の基本路線であったと思いますが、新たな切り口を導入しているのと、その他の独自路線アプローチもみられるのではないでしょうか。「長距離依存性だけ近似して対処する」という研究が 3 つありますね。

グループ「Transformer」のサブグループ「セルフアテンションの計算量に対処する」

- $QK^\top$ の成分を間引く (スパースにする) 。
  - Transformer 内のすべてのコンポーネントをスパースな亜種にした上でスパースなアテンションも取り入れる [11]。
- $QK^\top$ を低ランク近似する (行列分解する)。
  - カーネル法の計算量削減のアプローチを応用する [3]。
- スパース化と低ランク近似を統合する [19]。
- 長距離依存性の計算量を削減する。
  - 短距離依存性はそのまま計算し、長距離依存性は短い系列に射影する [20]。
  - 長距離依存性については重み付き期待値に対してセルフアテンションする [24]。
  - FMM(高速多重極法) を応用して長距離依存性を低ランク近似する [37]。
- 固定長の系列を利用して計算量を線形に抑える [4]。
- 最初のセルフアテンション層でのみ $QK^\top$ を計算し後はそれを時間発展させる [7]。
- アテンションの計算に高速フーリエ変換を応用する [25]。

大雑把に全体が整理できた気がします。とはいえ、アブストラクトだけではよくわからないので本文も確認したいですが、これだけ選んでしまったので一人で作業するのは骨が折れますね……。

――お疲れさま。随分とホワイトボードがぎっしりだね。

副部長! よいところにいらっしゃいました!!

第 2 節　それぞれのお話 (※ 一言ずつのみ)

えっこれらの論文を全部読みたいの? まあいいけど……何を抑えたいか絞った方がいいんじゃないかな。理論系の論文には当てはまらないけど、「提案手法」「想定データ (実際に検証したデータ)」「ベースライン手法」あたりかな。とりあえず予稿で出てきた順に確認していこうか (2022-01-27 確認できていません) 。

[1] RNN、CNN、連続時間モデルの性質をあわせもつ「線形状態空間層」

著者・原題（予稿リンク）

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, Christopher Ré. Combining Recurrent, Convolutional, and Continuous-time Models with Linear State Space Layers

「『線形状態空間層 (LSSL)』で RNN、CNN、連続時間モデルを結合する」といったタイトルだよね。であれば想定データはこれらのモデルの性質を全て要するデータなんだろうか?

この論文はゴールを「RNN、CNN、連続時間モデルの結合」に置いていると思います。なので RNN と CNN と連続時間モデルをコンポーネントとして組み合わせたようなモデルをイメージしてしまうのですが、どうもそうではないんです。

[2] Transformer を行列分解してパラメータが 10 倍削減できることを示す

著者・原題（予稿リンク）

Aliakbar Panahi, Seyran Saeedi, Tom Arodz. Shapeshifter: a Parameter-efficient Transformer using Factorized Reshaped Matrices.

この論文についてはアブストラクトで何となく研究内容がわかるような気がします。つまり、「Transformer はそんなにパラメータが必要なのか」という問題意識から、Transformer の低ランク近似表現を打ち出し、その表現でも性能を損なわないといったことを検証しているはずです。具体的にどの層をどのように低ランク近似したかまではアブストラクトのみからはわかりませんが……なお、この研究を「セルフアテンションの計算量に対処する」ではなく「Transformer の性質を理解する」に分類した理由は、この研究が専ら「パラメータ数」に主眼を置いているようにみえるからです。無論パラメータ数の削減は計算量の削減と無関係ではありませんが、この研究は指標もパラメータ数そのものになっているようにみえます。実務的に「パラメータ数を小さくすること」は最終目的にはならないと思うんです。なのでこれはむしろ Transformer の性質を解明した研究であると解釈しました。

[3] Skyformer——セルフアテンションの Nyström 近似

著者・原題（予稿リンク）

Yifan Chen, Qi Zeng, Heng Ji, Yun Yang. Skyformer: Remodel Self-Attention with Gaussian Kernel and Nyström method.

これまでのセルフアテンションの計算量削減には往々にして近似誤差の理論保証がないというようにいっていますね。だから手法間の比較もできなくなっているし、ハイパーパラメータによる計算量削減度合いの調整もできなくなっていると。他方、カーネルマシンもまた内積計算がボトルネックになっていると。そうですね、カーネル法のグラム行列のサイズはデータサイズに応じて n×n になるわけですから。だから対処法として Nyström 近似などが…って、タイトル中の文字化けしているのこれですね。Nyström 近似を適用できるようにして適用した Transformer が Skyformer であると。なぜ Skyformer なのかというと Symmetrization of Kernelized attention for NYström method ですか……Y に無理みが強い……。

[4] Luna——固定長変数を利用してセルフアテンションの計算量を線形に抑える

著者・原題（予稿リンク）

Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, Luke Zettlemoyer. Luna: Linear Unified Nested Attention.

先の論文に続いて「セルフアテンションの計算量に対処する」一味です。論文の Fiugre 2 にある黄色いベクトルがミソなんでしょうか。

[5] RNN がある再生核ヒルベルト空間におけるカーネル法であると示す

著者・原題（予稿リンク）

Adeline Fermanian, Pierre Marion, Jean-Philippe Vert, Gérard Biau. Framing RNN as a kernel method: A neural ODE approach.

RNN がある再生核ヒルベルト空間におけるカーネル法であると示しているようですね。まず RNN が常微分方程式の離散化だよねというところから着手していますね。

[6] RNN の隠れ状態にノイズを添加して訓練してロバスト性を向上させる

著者・原題（予稿リンク）

Soon Hoe Lim, N. Benjamin Erichson, Liam Hodgkinson, Michael W. Mahoney. Noisy Recurrent Neural Networks.

RNN の隠れ状態にノイズを添加して訓練することで正則化でき、ロバスト性が向上すると。

[7] セルフアテンション行列を時間発展させる

著者・原題（予稿リンク）

Subhabrata Dutta, Tanya Gautam, Soumen Chakrabarti, Tanmoy Chakraborty. Redesigning the Transformer Architecture with Insights from Multi-particle Dynamical Systems.

これも「セルフアテンションの計算量に対処する」一味なんですが、一味の中でも異彩を放っているのではと思うんです。

[8] 言語処理に適した Transformer の構造を探索する

著者・原題（予稿リンク）

David So, Wojciech Mańke, Hanxiao Liu, Zihang Dai, Noam Shazeer, Quoc Le. Searching for Efficient Transformers for Language Modeling.

こちらの論文は、Transformer の計算コスト削減にもはやアーキテクチャの探索というアプローチをしていますね。そのように特定されたアーキテクチャを Primer とよんでいるようです。学習コストが 3 分の 1 にまで削減されたということですが……。

[9] Self-IRU——繰り返し自身のインスタンスを生み出す RNN

著者・原題（予稿リンク）

Aston Zhang, Yi Tay, Yikang Shen, Alvin Chan Guo Wei, SHUAI ZHANG. Self-Instantiated Recurrent Units with Dynamic Soft Recursion.

自由度が高そうです。

[10] gMLP——ゲーティングを付けた MLP で Transformer と同程度の性能を得る

著者・原題（予稿リンク）

Hanxiao Liu, Zihang Dai, David So, Quoc Le. Pay Attention to MLPs.

gMLP——ゲーティングを付けた MLP で Transformer と同程度の性能が得られると。これによって画像認識ではセルフアテンションが重要でないこともわかったとありますね。

[11] Terraformer——疎な代替コンポーネントからなる Transformer

著者・原題（予稿リンク）

Sebastian Jaszczur, Aakanksha Chowdhery, Afroz Mohiuddin, Łukasz Kaiser, Wojciech Gajewski, Henryk Michalewski, Jonni Kanerva. Sparse is Enough in Scaling Transformers.

gMLP——ゲーティングを付けた MLP で Transformer と同程度の性能が得られると。これによって画像認識ではセルフアテンションが重要でないこともわかったとありますね。

Transformer 内の全てのコンポーネントをスパースな代替品にしたという感じなのかな。3 節 Sparse is Enough のサブセクションが以下のようになっているね。

- Sparse Feedforward Layer
- Sparse QKV Layer
- Sparse loss layer

[12] Transformer によるガウス過程モデルのカーネル同定

著者・原題（予稿リンク）

Fergus Simpson, Ian Davies, Vidhi Lalchand, Alessandro Vullo, Nicolas Durrande, Carl Edward Rasmussen. Kernel Identification Through Transformers.

Figure 1 を覗いてみると、正解付きの訓練データをすべて投入してエンコードしてデコードすると Matern 1/2 + Matern 3/2 + RBF × Matern 1/2 といったカーネルが出力されているように確かにみえますね。これが画像にキャプションを付けるモデルと似ているのでしょうか。

[13] 微分方程式群システムの逐次ベイズ推定

著者・原題（予稿リンク）

Jonathan Schmidt, Nicholas Krämer, Philipp Hennig. A Probabilistic State Space Model for Joint Inference from Differential Equations and Data.

複数の微分方程式で記述されるモデル——例えばシステムモデルや観測モデルが微分方程式で記述されているようなイメージでしょうか?——における推論は計算コストが高く、数値ソルバーとの相性が悪いと。しかし最近では常微分方程式をベイズフィルタで解く方法が打ち出されてきているので、隠れ変数がある状態空間モデ

ルに適用できる…ということでしょうか。ODE を解くのと同様のコストで拡張カルマンフィルタできると。COVID-19 データに SIRD モデルを適用して検証しているのですね。

[14] 時系列データのプレ処理としての成分クラスタリング

著者・原題（予稿リンク）

Zhibo Zhu, Ziqi Liu, Ge Jin, Zhiqiang Zhang, Lei Chen, Jun Zhou, Jianyong Zhou. MixSeq: Connecting Macroscopic Time Series Forecasting with Microscopic Time Series Data.

個別の確率分布が混合されて予測対象の時系列をつくり上げていると仮定して成分クラスタリングするのでしょうか。Table 4 にあるように既存の時系列モデルに好きに組み合わせられるようですね。

[15] セルフアテンションのヘッド間で Q, K の分布を一致させるように訓練する

著者・原題（予稿リンク）

Shujian Zhang, Xinjie Fan, Huangjie Zheng, Korawat Tanwisuth, Mingyuan Zhou. Alignment Attention by Matching Key and Query Distributions.

ヘッドごとにセルフアテンションの強さのようなものが異なると困るということでしょうか。英文を和文に翻訳するのに、ビジネス英語に強い A さんと日常会話に強い B さんを連れてきて、「あなたたちの専門からみて、1 単語目に影響を及ぼす単語に色鉛筆で色を塗ってください」と指示したとして、コントラストの付け方がからっきし違うと困るとか……いや、こんな話ではないかもしれませんが。

[16] セルフアテンションを生物学的な記憶モデルと解釈する

著者・原題（予稿リンク）

Trenton Bricken, Cengiz Pehlevan. Attention Approximates Sparse Distributed Memory.

[17] Vision Transformer でグリッド分割のグリッド分割も Transformer する

著者・原題（予稿リンク）

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing XU, Yunhe Wang. Transformer in Transformer.

[18] スイッチング線形動的システムで RNN をリバースエンジニアリングする

著者・原題（予稿リンク）

Jimmy Smith, Scott Linderman, David Sussillo. Reverse engineering recurrent neural networks with Jacobian switching linear dynamical systems.

[19] スパース化と低ランク近似を併用してセルフアテンションの計算量を削減する

著者・原題（予稿リンク）

Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, Christopher Ré. Scatterbrain: Unifying Sparse and Low-rank Attention.

[20] 長距離依存性は短い系列に射影してセルフアテンションする

著者・原題（予稿リンク）

Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, Bryan Catanzaro. Long-Short Transformer: Efficient Transformers for Language and Vision.

これは「長距離依存性だけ近似してセルフアテンションの計算量に対処する」3 兄弟の長男ですね。なるほど長男らしいストレートなアプローチです。

3 兄弟だったの!?

[21] RNN が学習できると保証される関数の制約を撤廃する

著者・原題（予稿リンク）

Lifu Wang, Bo Shen, Bo Hu, Xing Cao. On the Provable Generalization of Recurrent Neural Networks.

RNN が学習できると保証される関数について、これまでの制約を一部撤廃して誤差の上限を示したり、制約がなくても多項式時間で学習できることを示したというアブストラクトにみえます……。

[22] RNN が学習できると保証される関数の制約を撤廃する

著者・原題（予稿リンク）

Abhishek Panigrahi, Navin Goyal. Learning and Generalization in RNNs.

RNN が学習できると保証される関数について、制約を撤廃したというアブストラクトにみえます……。

[23] Autoformer——トレンド-季節性分解付きの Transformer

著者・原題（予稿リンク）

Haixu Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting.

この論文自体はタイトルで長期予測とはいってもセルフアテンションの計算量削減の話ではないようですね……季節成分とトレンドの分解をするブロックを導入することで時系列の長期予測の精度を上げたという話にみえます。

[24] Combiner——長距離依存性については重み付き期待値に対してセルフアテンションする

著者・原題（予稿リンク）

Hongyu Ren, Hanjun Dai, Zihang Dai, Mengjiao Yang, Jure Leskovec, Dale Schuurmans, Bo Dai. Combiner: Full Attention Transformer with Sparse Computation Cost.

これは「長距離依存性だけ近似してセルフアテンションの計算量に対処する」3 兄弟の次男ですね。お兄さんとは一味違うアプローチにみえます。タイトルの Full Attention は「間引くタイプの計算量削減ではない」というニュアンスがあるのかもしれません。間引いても性能が出れば間引いてもいいとは思いますが……。こちらのモデルには Combiner という名前が付けられていて、論文の 4 節にあるようにいくつかの亜種があるのですね……。

[25] FFT を応用した相対位置エンコーディングにも適用できるセルフアテンション計算量削減

著者・原題（予稿リンク）

Shengjie Luo, Shanda Li, Tianle Cai, Di He, Dinglan Peng, Shuxin Zheng, Guolin Ke, Liwei Wang, Tie-Yan Liu. Stable, Fast and Accurate: Kernelized Attention with Relative Positional Encoding.

アブストラクトは、「セルフアテンションの計算量を削減する既存研究の多くは『内積をとってからソフトマックスする』方式にしか対応できない」と主張しているようにみえます。そして、それだと「相対位置エンコーディング (RPE) に対応できな

い」と……これまでに提案されているセルフアテンション計算量削減ってそんなに制約があったんですか??

[26] 時系列の混合分布を推定するためのコアセットを構築する

著者・原題（予稿リンク）

Lingxiao Huang, K Sudhir, Nisheeth Vishnoi. Coresets for Time Series Clustering.

データが混合分布からなると仮定して、その最尤推定のために必要なコアセットを構築するアルゴリズムを打ち出したと。コアセットというのは訓練データの部分集合の意味ですよね? 訓練データをすべてつかって混合分布の最尤推定をするのは、確かに大変そうですから、効果的なコアセットに絞ることは重要そうに思えます。

[27] Transformer に状態空間モデルを組み合わせて時系列を予測する

著者・原題（予稿リンク）

Binh Tang, David Matteson. Probabilistic Transformer For Time Series Analysis.

状態空間モデルの状態の時間発展を Transformer にしたのでしょうか? ProTran: Probabilistic Transformer というモデル名が付いていますね。Table 1 をみると確かに明確に比較手法よりよさそうにみえますが……何ですかこの評価指標は? CRPS?

Table 1 の下の方に定義があるね。分布予測をしたときの評価指標みたいだ。ああ、この式の $F(z)$ は累積分布関数で、$1_{x \leqq z}$ は真の累積分布関数といったところなのかな? だからぴったり正解の値で階段になる累積分布関数を予測すれば CRPS は 0 だね。

[28] ドロップアウトを活用して LSTM の訓練を効率化する

著者・原題（予稿リンク）

Anup Sarma, Sonali Singh, Huaipan Jiang, Rui Zhang, Mahmut Kandemir, Chita Das. Structured in Space, Randomized in Time: Leveraging Dropout in RNNs for Efficient Training.

LSTM の計算量に課題感を抱いている研究はこれが唯一にみえます……それで、ドロップアウトを活用して計算量を削減する? ドロップアウトで計算量を削減できる

んですか?

論文 4 ページの Figure 1 に Case 1〜4 のドロップアウトの図示があるね。この中の Case 3 を使うのかな? そうするとこの Case 3 で灰色になっている列はそもそも計算しなくていいことになるね。

[29] 時系列予測のための位相的アテンション

著者・原題（予稿リンク）

Sebastian Zeng, Florian Graf, Christoph Hofer, Roland Kwitt. Topological Attention for Time Series Forecasting.

タイトルから想像でいうと、おそらくある時点のデータがある時点のデータにどれだけアテンションすべきか＝とれだけ近いのかを何か位相的データ解析のように考えたのではないかと思いますが、そうであればこれは RNN や CNN ではない Transformer にだから投げ付けられる発想だと思います (RNN や CNN にも位相を活かした亜種があるか存じ上げませんが)。

[30] Softmax しないセルフアテンションで偏微分方程式を解く

著者・原題（予稿リンク）

Shuhao Cao. Choose a Transformer: Fourier or Galerkin.

関係ないですが今回選んだ論文の中で単著なのこれだけなんですよね……偏微分方程式を解くってどういうことですか? 解演算子 (solution operator) とは?

ゴールは偏微分方程式の中に出てくる関数 f を特定することだよね。解演算子は、偏微分方程式の空間から関数の空間への写像だと思う。

[31] タスクの解空間を構造化して RNN の性質を調べる

著者・原題（予稿リンク）

Elia Turner, Kabir Dabholkar, Omri Barak. Charting and Navigating the Space of Solutions for Recurrent Neural Networks.

Figure 1B. は、式 (1) で表されるモデルにしたがって移動する点が最初緑の点にいて、時刻 $T = 10$ に赤い点に到達するようにしたいという図ですね。こうなるようにモデルのパラメータを調整したいと。それでそのようなパラメータは Figure 1C の紫いろの濃いところに分布していて、どの点を選ぶかで Figure 1D, 1E, 1F のように緑の点から赤い点までの軌跡が異なってくるようです。……これって RNN なんですか?

式 (1) の右辺の $-x$ を左辺に移項したら「次の状態 = 今の状態を変換したもの」って感じにはみえなくもないかも……。

[32] SBO-RNN——勾配消失/爆発しない RNN のサブセット

著者・原題（予稿リンク）

Ziming Zhang, Yun Yue, Guojun Wu, Yanhua Li, Haichong Zhang. SBO-RNN: Reformulating Recurrent Neural Networks via Stochastic Bilevel Optimization.

こちらの SBO-RNN は、RNN の中でも勾配消失/爆発せず安定的に学習できる構造を突き止め、そのサブセットに SBO-RNN と名付けたということなのでしょうか?

[33] 時系列のオンライン異常検知の偽陽性率を制御する

著者・原題（予稿リンク）

Quentin Rebjock, Baris Kurt, Tim Januschowski, Laurent Callot. Online false discovery rate control for anomaly detection in time series.

時系列データのオンライン異常検知の話ですが、「FDRC ルール」とは読んで字のごとく偽陽性率を抑えるためのルール、なのでしょうか……?

[34] Transformer と CNN のロバスト性を比較する

著者・原題（予稿リンク）

Yutong Bai, Jieru Mei, Alan L. Yuille, Cihang Xie. Are Transformers more robust than CNNs? .

「Transformer が CNN よりロバストとされているがそんなことはない」といったアブストラクトですが、そもそも Transformer が CNN よりロバストとされているんですか?

[35] 成長するメモリ付き固定精度 RNN がチューリング完全であると示す

著者・原題（予稿リンク）

Stephen Chung, Hava Siegelmann. Turing Completeness of Bounded-Precision Recurrent Neural Networks.

チューリング完全って何ですか?

大雑把にいうと、C 言語がチューリング完全だから、C 言語で実装できる関数を実装できるってことかな。

[36] データ間でセルフ (セルフ?) アテンションする Non-Parametric Transformer

著者・原題（予稿リンク）

Jannik Kossen, Neil Band, Clare Lyle, Aidan N. Gomez, Thomas Rainforth, Yarin Gal. Self-Attention Between Datapoints: Going Beyond Individual Input-Output Pairs in Deep Learning.

検証したのがテーブルデータや CIFAR-10 であって系列データといった向きのデータではなさそうですが、タイトルが気になりました。Self-Attention Between Datapoints というのは、言語データに喩えるなら、単語から文章内の他の単語へアテンションするのではなく、文章から他の文章へアテンションするということなのでしょうか。それって Self なんでしょうか……? それはさておき、本当に「データセット全体を入力とする」のであれば訓練や推論のコストが膨大になりそうですが……?

[37] FMMformer——FMM(高速多重極法) を応用して長距離依存性を低ランク近似する

著者・原題（予稿リンク）

Tan Nguyen, Vai Suliafu, Stanley Osher, Long Chen, Bao Wang. FMMformer: Efficient and Flexible Transformer via Decomposed Near-field and Far-field Attention.

これは「長距離依存性だけ近似してセルフアテンションの計算量に対処する」3 兄弟の 3 男にして、1 番目のお兄さんとも 2 番目のお兄さんとも似ていないアプローチです。

兄弟じゃないんだよなあ。

FMM(高速多重極法) というのは粒子間の相互作用を近距離成分と遠距離成分に分けて計算量を削減する電磁気学分野の手法なのでしょうか。こういわれると、Transformer の相互作用にも応用できる気配がしますが、具体的にどのような手法なのでしょうか?

[38] ニューラル時系列モデルにおける誤差の自己相関の調整

著者・原題（予稿リンク）

Fan-Keng Sun, Chris Lang, Duane Boning. Adjusting for Autocorrelated Errors in Neural Networks for Time Series.

通常ニューラルネットで時系列データを学習するときに誤差系列に自己相関はないとしていますが、現実には自己相関するので誤差系列の自己相関係数も学習するといっていますね? しかし、誤差系列の自己相関係数を学習してどうするんです?

論文の 4〜5 ページをみると、学習の手順は以下かな? 誤差系列の 1 次の自己相関のみ考えているね。

- まずは通常通り 2 乗誤差を最小化するように時系列予測モデルを学習する。
- 学習したモデルの誤差系列の 1 次の自己相関係数を出す。
- 次の学習では、予測値を「モデルの予測値に前ステップの誤差に 1 次の自己相関係数を乗じたもの」として、この 2 乗誤差を最小化するように時系列予測モデルを学習する。
- また誤差系列の 1 次の自己相関を出す。
- 収束するまで繰り返す。

つまり、実績誤差をみて修正していくモデルなんだね。

なるほど? 「実績誤差をみて修正するマン」がいてくれる前提でモデルを最良にするということでしょうか。

[39] RED-SDS——継続期間も明示的に利用するレジームスイッチングモデル

著者・原題（予稿リンク）

Abdul Fatir Ansari, Konstantinos Benidis, Richard Kurle, Ali Caner Turkmen, Harold Soh, Alexander J. Smola, Bernie Wang, Tim Januschowski. Deep Explicit Duration Switching Models for Time Series.

時系列のレジームの切り替わりを捉えたいといっていますね……レジームというのはこの時点を境に好景気から不景気になったというような環境の変化のようなものですよね、適当な訳語がわかりませんが……。それで、RED-SDS: Recurrent Explicit Duration Switching Dynamical System なる提案モデルでは状態にも時間にも依存してレジームをスイッチングできるようにしたんですね？ うーん、いまいちどう価値があることをしたのかわからないのですが……。

おそらくレジームスイッチングモデルは元々は何らかの変数 (観測不可能なら状態といった方がいいかな) に依存してスイッチングするモデルとして考案されたんだよね。ある変数がこうなってきたらここから不景気レジームだな、みたいに。でも、レジームの継続期間にもパターンがあるならそれを積極的に利用した方がいいよね。わからないけど、この病気の流行は 1 ヶ月で落ち着く、みたいな知識があったりしたらさ。それが Explicit Duration Switching の意味かなと思うんだけど、この発想自体は前からあって、この論文の新規性はそれを状態スイッチングモデルと組み合わせてディープで実現したところにあるのかな?

第 3 節　結び——NeurIPS 2021 にみる最近のニューラル系列モデルへの発見・工夫・理解　(※ 未執筆)

Appendix

公開コード

以下の論文はアクセスできる公開コードの存在が確認できました。見落としはあると思います……。[2] [12] については 2022-01-27 時点でリポジトリにコードが確認できなかったので含めていません。

[1] (LSSL) https://github.com/HazyResearch/state-spaces
[3] (Skyformer) https://github.com/pkuzengqi/Skyformer
[5] (RNN as Kernel Method) https://github.com/afermanian/rnn-kernel
[6] (Noisy RNN) https://github.com/erichson/NoisyRNN

[7] (TransEvolve) https://github.com/LCS2-IIITD/TransEvolve
[8] (Primer) https://github.com/google-research/google-research/tree/master/primer
[11] (Terraformer) https://github.com/google/trax/blob/v1.4.1/trax/examples/Terraformer_from_scratch.ipynb
[19] (Scatterbrain) https://github.com/HazyResearch/pixelfly
[23] (Autoformer) https://github.com/thuml/Autoformer
[24] (Combiner) https://github.com/google-research/google-research/tree/master/combiner

[3] Skyformer の公開コード

Skyformer [3] は以下にコードが公開されていますね。以下のリビジョンは 2022-01-27 時点の最新です。

- https://github.com/pkuzengqi/Skyformer/tree/cfe8c8cb48a151fd150ff4a87fdb24b288356869

リポジトリを clone した./src/ 下で適当に以下を実行するとインスタンス化できます（src/requirements.txt のパッケージが充足していれば）。

```
from models.model_LRA import ModelForSC, ModelForSCDual
from config import Config

model_config = Config["lra-text"]["model"]
model_config["mixed_precision"] = True
model_config["attn_type"] = "softmax"
model = ModelForSC(model_config)
print(model)
```

```
ModelForSC(
  (model): Model(
    (embeddings): Embeddings(
      (word_embeddings): Embedding(512, 64)
      (position_embeddings): Embedding(4000, 64)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (transformer_0): TransformerLayer(
      (norm1): LayerNorm((64,), eps=1e-05, elementwise_affine=True)
      (mha): Attention(
        (W_q): Linear(in_features=64, out_features=64, bias=True)
        (W_k): Linear(in_features=64, out_features=64, bias=True)
        (W_v): Linear(in_features=64, out_features=64, bias=True)
        (attn): SoftmaxAttention(
          (drop_attn): Dropout(p=0.1, inplace=False)
        )
# 以下省略
```

これに Long Range Arena のデータを渡せばよいです。データの取得及びプレ処理方法は README に記述されていますが、README には親切にもプレ処理済みのデータへのリンクもあるのでそれを拝借することにします。適当に lra-text.dev.pickle をダウンロードします (ご自身の責任で)。中身を確認すると 25000 の文章が入っており、1 つの文章は 4096 文字 (おそらく ASCII コード) からなっており、何か正解ラベルが付いていることがわかりますね。

```python
import pickle
with open('./Downloads/lra-text.dev.pickle', 'rb') as ifile:
    x = pickle.load(ifile)

print(len(x))
print(x[0]['input_ids_0'])
print(len(x[0]['input_ids_0']))
print(x[0]['label'])
```

```
25000
[ 85 105 102 ...   0   0   0]
4096
1
```

ここまでやっておいてなんですがモデルには文章とみせかけて適当な整数列を渡せば動きます。以下でモデルが値を返却してくれます。

```
import torch
x = torch.tensor([[0, 1, 2, 3, 4]])
label = torch.tensor([0])
y = model(x, None, label)
print(y)
```

```
{'loss': tensor([0.8154], grad_fn=), 'accu': tensor([0.])}
```

なお AttributeError: module 'torch.cuda.amp' has no attribute 'autocast' と怒られたので怒られなくなるまで with torch.cuda.amp.autocast(enabled = False): をコメントアウトしています。後日 GPU 機で試します。

[11] Terraformer の公開コード

Terraformer [11] のソースコードは trax なるライブラリの一部として公開されているそうなのですが……この trax は PyPI に登録されているのですね。pip でインストールできました (CentOS 上の Python 3.9.4 に)。何でも trax パッケージの 1.4.0 にコードを含めたとかいてあるので、GitHub 上の 1.4.0 のリリースノートに何かかいてあるでしょうか……ここには何もかいていませんね。ドキュメントにもリリースノートのようなものはないようです……ライブラリの機能のうち Terraformer に該当するものはどれなのでしょうか……。

もう v1.3.9 と v1.4.0 の差分からそれらしいコミットを探せばいいんじゃない？……以下がそれっぽい。

- https://github.com/google/trax/commit/22384907983e697ec20fe3230cc0988cfc7ac140

つまり以下のノートブックだね。

- https://github.com/google/trax/blob/v1.4.1/trax/examples/Terraformer_from_scratch.ipynb

参考文献

[1] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, Christopher Ré. Combining Recurrent, Convolutional, and Continuous-time Models with Linear State Space Layers. In NeurIPS 2021.

[2] Aliakbar Panahi, Seyran Saeedi, Tom Arodz. Shapeshifter: a Parameter-efficient Transformer using Factorized Reshaped Matrices. In NeurIPS 2021.

[3] Yifan Chen, Qi Zeng, Heng Ji, Yun Yang. Skyformer: Remodel Self-Attention with Gaussian Kernel and Nyström method. In NeurIPS 2021.

[4] Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, Luke Zettlemoyer. Luna: Linear Unified Nested Attention. In NeurIPS 2021.

[5] Adeline Fermanian, Pierre Marion, Jean-Philippe Vert, Gérard Biau. Framing RNN as a kernel method: A neural ODE approach. In NeurIPS 2021.

[6] Soon Hoe Lim, N. Benjamin Erichson, Liam Hodgkinson, Michael W. Mahoney. Noisy Recurrent Neural Networks. In NeurIPS 2021.

[7] Subhabrata Dutta, Tanya Gautam, Soumen Chakrabarti, Tanmoy Chakraborty. Redesigning the Transformer Architecture with Insights from Multi-particle Dynamical Systems. In NeurIPS 2021.

[8] David So, Wojciech Mańke, Hanxiao Liu, Zihang Dai, Noam Shazeer, Quoc Le. Searching for Efficient Transformers for Language Modeling. In NeurIPS 2021.

[9] Aston Zhang, Yi Tay, Yikang Shen, Alvin Chan Guo Wei, SHUAI ZHANG. Self-Instantiated Recurrent Units with Dynamic Soft Recursion. In NeurIPS 2021.

[10] Hanxiao Liu, Zihang Dai, David So, Quoc Le. Pay Attention to MLPs. In NeurIPS 2021.

[11] Sebastian Jaszczur, Aakanksha Chowdhery, Afroz Mohiuddin, Łukasz Kaiser, Wojciech Gajewski, Henryk Michalewski, Jonni Kanerva. Sparse is Enough in Scaling Transformers. In NeurIPS 2021.

[12] Fergus Simpson, Ian Davies, Vidhi Lalchand, Alessandro Vullo, Nicolas Durrande, Carl Edward Rasmussen. Kernel Identification Through Transformers. In NeurIPS 2021.

[13] Jonathan Schmidt, Nicholas Krämer, Philipp Hennig. A Probabilistic State Space Model for Joint Inference from Differential Equations and Data. In NeurIPS 2021.

[14] Zhibo Zhu, Ziqi Liu, Ge Jin, Zhiqiang Zhang, Lei Chen, Jun Zhou, Jianyong Zhou. MixSeq: Connecting Macroscopic Time Series Forecasting with Microscopic Time Series Data. In NeurIPS 2021.

[15] Shujian Zhang, Xinjie Fan, Huangjie Zheng, Korawat Tanwisuth, Mingyuan Zhou. Alignment Attention by Matching Key and Query Distributions. In NeurIPS 2021.

[16] Trenton Bricken, Cengiz Pehlevan. Attention Approximates Sparse Distributed Memory. In NeurIPS 2021.

[17] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing XU, Yunhe Wang. Transformer in Transformer. In NeurIPS 2021.

[18] Jimmy Smith, Scott Linderman, David Sussillo. Reverse engineering recurrent neu-

ral networks with Jacobian switching linear dynamical systems. In NeurIPS 2021.
[19] Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, Christopher Ré. Scatterbrain: Unifying Sparse and Low-rank Attention. In NeurIPS 2021.
[20] Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, Bryan Catanzaro. Long-Short Transformer: Efficient Transformers for Language and Vision. In NeurIPS 2021.
[21] Lifu Wang, Bo Shen, Bo Hu, Xing Cao. On the Provable Generalization of Recurrent Neural Networks. In NeurIPS 2021.
[22] Abhishek Panigrahi, Navin Goyal. Learning and Generalization in RNNs. In NeurIPS 2021.
[23] Haixu Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In NeurIPS 2021.
[24] Hongyu Ren, Hanjun Dai, Zihang Dai, Mengjiao Yang, Jure Leskovec, Dale Schuurmans, Bo Dai. Combiner: Full Attention Transformer with Sparse Computation Cost. In NeurIPS 2021.
[25] Shengjie Luo, Shanda Li, Tianle Cai, Di He, Dinglan Peng, Shuxin Zheng, Guolin Ke, Liwei Wang, Tie-Yan Liu. Stable, Fast and Accurate: Kernelized Attention with Relative Positional Encoding. In NeurIPS 2021.
[26] Lingxiao Huang, K Sudhir, Nisheeth Vishnoi. Coresets for Time Series Clustering. In NeurIPS 2021.
[27] Binh Tang, David Matteson. Probabilistic Transformer For Time Series Analysis. In NeurIPS 2021.
[28] Anup Sarma, Sonali Singh, Huaipan Jiang, Rui Zhang, Mahmut Kandemir, Chita Das. Structured in Space, Randomized in Time: Leveraging Dropout in RNNs for Efficient Training. In NeurIPS 2021.
[29] Sebastian Zeng, Florian Graf, Christoph Hofer, Roland Kwitt. Topological Attention for Time Series Forecasting. In NeurIPS 2021.
[30] Shuhao Cao. Choose a Transformer: Fourier or Galerkin. In NeurIPS 2021.
[31] Elia Turner, Kabir Dabholkar, Omri Barak. Charting and Navigating the Space of Solutions for Recurrent Neural Networks. In NeurIPS 2021.
[32] Ziming Zhang, Yun Yue, Guojun Wu, Yanhua Li, Haichong Zhang. SBO-RNN: Reformulating Recurrent Neural Networks via Stochastic Bilevel Optimization. In NeurIPS 2021.
[33] Quentin Rebjock, Baris Kurt, Tim Januschowski, Laurent Callot. Online false discovery rate control for anomaly detection in time series. In NeurIPS 2021.
[34] Yutong Bai, Jieru Mei, Alan L. Yuille, Cihang Xie. Are Transformers more robust than CNNs? . In NeurIPS 2021.
[35] Stephen Chung, Hava Siegelmann. Turing Completeness of Bounded-Precision Recurrent Neural Networks. In NeurIPS 2021.
[36] Jannik Kossen, Neil Band, Clare Lyle, Aidan N. Gomez, Thomas Rainforth, Yarin Gal. Self-Attention Between Datapoints: Going Beyond Individual Input-Output Pairs

in Deep Learning. In NeurIPS 2021.

[37] Tan Nguyen, Vai Suliafu, Stanley Osher, Long Chen, Bao Wang. FMMformer: Efficient and Flexible Transformer via Decomposed Near-field and Far-field Attention. In NeurIPS 2021.

[38] Fan-Keng Sun, Chris Lang, Duane Boning. Adjusting for Autocorrelated Errors in Neural Networks for Time Series. In NeurIPS 2021.

[39] Abdul Fatir Ansari, Konstantinos Benidis, Richard Kurle, Ali Caner Turkmen, Harold Soh, Alexander J. Smola, Bernie Wang, Tim Januschowski. Deep Explicit Duration Switching Models for Time Series. In NeurIPS 2021.

NeurIPS 2021 にみる **最近のニューラル系列モデルへの発見・工夫・理解**
未完成ドラフト

2021 年 1 月 27 日　初版発行

著　者　クッキー
発行者　クッキーの日記
https://cookie-box.hatenablog.com/