

Neural Architecture Searchを取り入れた 時系列予測モデル運用に向けた一検討

高橋 佑里子¹ 鈴木 成人² 田原 司睦² 小口 正人¹

概要: 仮想環境を用いたクラウドサービスにおいて, CPU 資源のオーバーコミットに由来する仮想マシン (Virtual Machine: VM) の性能低下を防ぐことを目的として, VM の CPU 使用率を予測し, その結果に基づいて制御を行う技術が知られている. VM とそこで実行されるアプリケーションは時間が経つにつれ変化していくため, 環境の変化に合わせて予測モデルを継続的に学習し, モデルを更新することで予測精度を担保することが望ましいが, 従来は学習させるデータを変えるのみで, 予測モデルのネットワーク構造を変えることはなかった. しかし, 学習させるデータによって最適な予測モデルの構造は異なる. そこで本研究では, Neural Architecture Search(NAS) を取り入れた時系列予測モデルの運用を提案する. NAS 運用実現の準備として, 本論文ではモデルのネットワーク構造と学習時間/予測精度の関係についての調査を行った.

A Study on the Operation of Time Series Prediction Models Incorporating Neural Architecture Search

YURIKO TAKAHASHI¹ SHIGETO SUZUKI² TSUGUCHIKA TABARU² MASATO OGUCHI¹

1. はじめに

近年のクラウドサービスでは, 物理サーバ (Physical Machine: PM) の低 CPU 使用率が課題となっており [1], これを改善すべく, 事業者では, サーバを仮想化することで CPU 使用率を向上させ, PM 数を削減する取り組みが行われている. この取り組みでは, PM が自身の CPU 資源を超えた仮想 CPU を割り当てられるオーバーコミット状態に陥ることで, 仮想マシン (Virtual Machine: VM) の性能が低下する可能性がある [2].

これを防ぐことを目的として, VM の CPU 使用率を予測モデルによって予測し, その結果に基づいて制御を行う技術が知られている [3]. 例えば, 図 1 のように運用エリア (PM1) とテンポラリエリア (PM2) が存在し, PM1 のみに VM1,2,3 が割り当てられている状態で, 10 分後に VM3 の CPU 使用率が大幅に上昇し PM1 の CPU 資源を超える

という予測がされた場合, VM3 をあらかじめ PM2 に移行しておき, 10 分後にオーバーコミット状態が発生しないようにする, といった制御である.

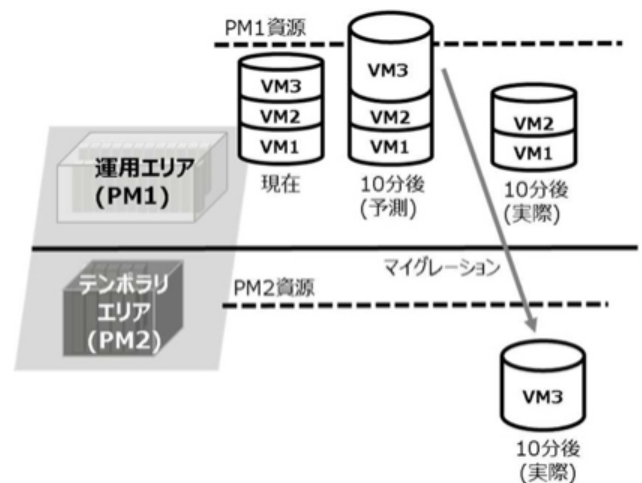


図 1 VM 制御のイメージ

¹ お茶の水女子大学

Ochanomizu University

² 富士通株式会社 富士通研究所

Fujitsu Ltd., Fujitsu Research

VM 上で多く実行されるアプリケーションは時間が経つ

につれ変化していくため、それに伴って実行される VM のワークロードも変化していく。図 2 のように、VM のワークロードが大きく 3 種類に分類できるとしたとき、日によって 3 種類の含まれる割合が変わっていく、といったイメージである。そのため、環境の変化に合わせて予測モデルを継続的に学習し、モデルを更新することで予測精度を担保することが望ましいが、従来は学習させるデータを変えるのみで、予測モデルのネットワーク構造を変えることはなかった。しかし、学習させるデータによって最適な予測モデルのネットワーク構造は異なる。

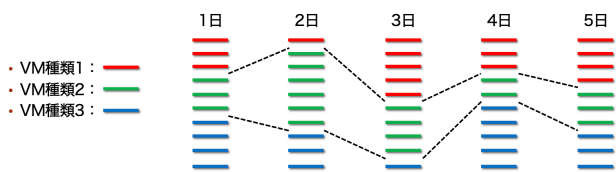


図 2 実行される VM の種類の割合が変化していくイメージ

そこで本研究では、Neural Architecture Search(NAS) を取り入れた時系列予測モデルの運用を提案する。NAS 運用実現の準備として、本論文ではモデルのネットワーク構造と学習時間/予測精度の関係についての調査を行った。本研究の成果は以下の通りである。

- VM の CPU 使用率予測モデルの運用において、NAS を取り入れるアイデアを提案した。
- 予測モデルの精度を向上させるために、NAS が有効であることを確認した。
- モデルのネットワーク構造と学習時間/予測精度の関係を調べた。
- 運用に向けて、予測モデルに適用する NAS 探索範囲の方針を定めた。

2. 関連研究

クラウドサービスにおける VM の CPU 使用率の予測や制御に関する研究は、以前から多く行われている。[4] では、ラックユニットの消費電力を分析し、API 連携を用いて VM の管理ソフトウェアと連携したシステムの提案を行っている。仮想データセンタを想定して計算したところ、実利用エリアの稼働率は常に 90%以内にとまることや、フットプリントを 40%削減できること、コンピュータールームエアコンとサーバを合わせた消費電力も 6.8%削減できることが示されている。また、[5] では、エネルギー消費量に基づいて、リアルタイムに VM を統合するフレームワークの提案を行っている。提案されたフレームワークを使用することで、フレームワークを使用していないデータセンタと比較して最大 80%の改善が示され、PM の高使用率と省エネルギーの実現に成功している。

3. 提案手法

本研究では、VM の CPU 予測モデルの継続的な運用において、NAS を導入する手法を提案する。NAS とは、ニューラルネットワークの構造自体を最適化することである [6]。この NAS を、予測モデルの継続学習時のフローとして適切なタイミングで組み込むことで、より高精度な VM の CPU 使用率予測が可能になると考えた。

3.1 予備実験

予備実験では、NAS の有効性を確認するために、異なる時系列データセット 20 種類 (4.1 で記述) に対して NAS を行い、出力されたモデルのネットワーク構造を比較した。

この実験では、ライブラリとして AutoKeras[7][8] を使用した。AutoKeras とは、AutoML 対応の Keras モジュールであり、決められた探索範囲の中で様々なネットワーク構造やハイパーパラメータで学習を行い、複数回の試行 (トライアル) の後、最も精度が良くなったモデルを出力する機能を持つ。また、これ以上のトライアルは不要と判断した場合に自動で打ち切る機能もある。

NAS は表 1 のような条件で行い、その他の条件は、エポック数 100, バッチサイズ 100, ウィンドウ幅 200, 最大トライアル数 100 とした。

表 1 NAS の範囲

パラメータ	範囲
最適化関数	adam または sgd
ラーニングレート	0.01 または 0.001
GRU の層数	1-3
GRU のユニット数	2-10

実験の結果、出力されたモデル 20 種類において、層数はすべて 3 層となったが、ユニット数には違いが見られた。ユニット数の内訳は、表 2 の通りである。

表 2 出力されたモデルのユニット数の内訳

ユニット数	個数
6	1
7	2
8	2
9	9
10	6

学習データによって最適なモデルのネットワーク構造は異なることから、NAS を行うことは精度を上げるために有効であるということが確認できた。層数が 3, ユニット数が 10 であるモデルは、図 3 の通りである。

また、この実験では各 NAS のトライアル回数が、少ないもので 20 程度、多いもので 60 程度、平均で 40 程度となった。NAS にかかる時間はトライアル回数に比例する

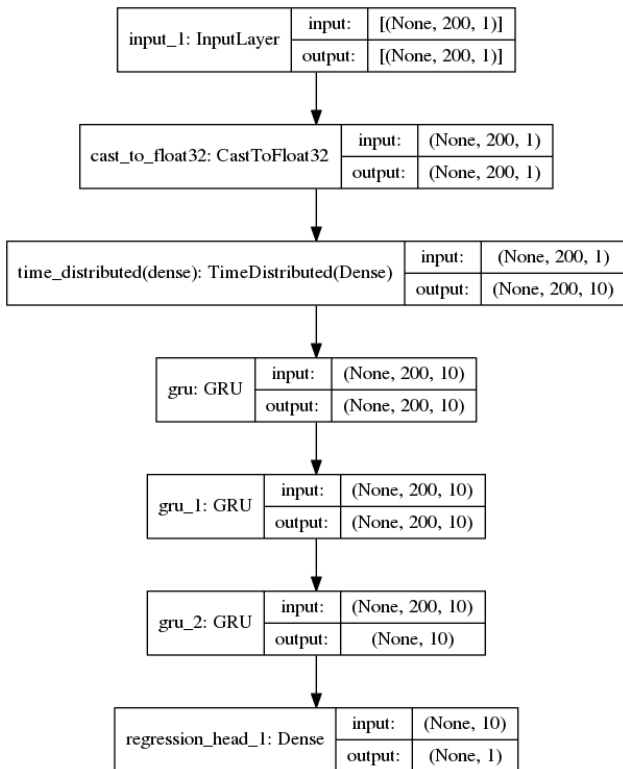


図 3 出力されたモデルの例 (層数 3, ユニット数 10)

ので、トライアル回数が多くなることで NAS にかかる時間が長くなる。そのため、運用で NAS を取り入れるには、事前に探索範囲を絞る必要があるということが分かった。

そこで、最適な NAS 探索範囲の方針を定めるために、以下の 2 つの実験を行った。

4. 実験

4.1 概要

本研究では、Microsoft 社が提供している Azure の VM トレースデータセット [9][10] の一部の、平均 CPU 使用率の項目を使用した。このデータセットは、1 点が 5 分間隔となっている。異なる傾向を持つデータセットを作成するために、以下の手順でデータセットを作成した。

- (1) 元のデータセットを 0-1 に正規化する
- (2) 開始点を 1 点ずつずらしながら 360 点 (30 時間分に相当) ごとに時系列データを切り出す
- (3) なるべく似ていないデータを選択するために、切り出されたデータに対して段階的に k-means 法クラスタリングを行い、100 種類を選択する (選択した 100 種類の一部を図 4 に示す)
- (4) 100 種類データを異なる割合で合計 100 個含むデータセットを 20 種類作成する

機械学習ライブラリには、Keras[11] を使用した。Keras を使用することで、簡潔なコードで予測モデルを記述することができ、迅速な実験が可能になる。なお、実験環境は表 3 の通りである。

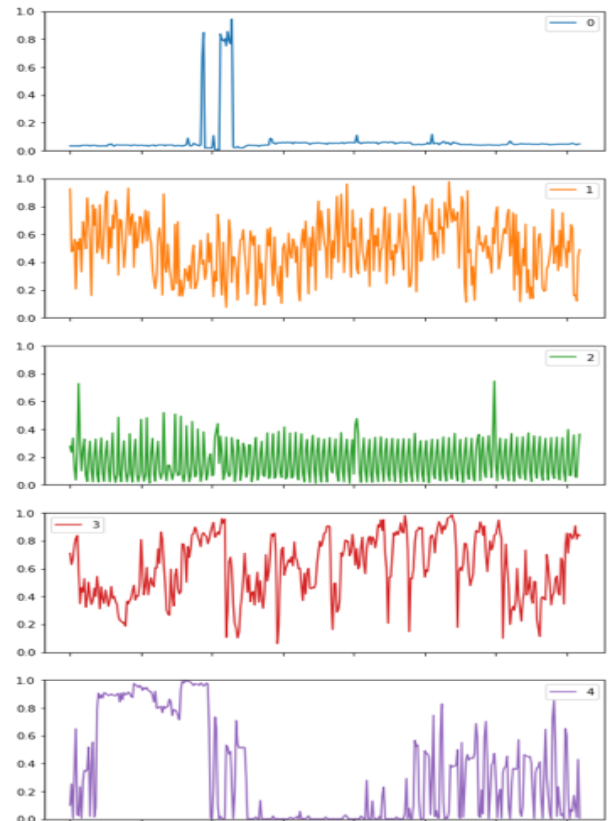


図 4 選択した 100 種類の波形の例 (5 種類)

表 3 実験環境

OS	CentOS Linux release 7.5.1804(Core)
サーバ	FUJITSU Primergy CX400 M1
CPU	Intel Xeon Haswell 2 ソケット 14 コア 2.6GHz E5-2697 145W
GPU	NVIDIA Tesla P100 16GB
Storage HDD	270GB read 0.21GB/s write 1.07GB/s
Memory	256GB DDR4 2133MHz
Python	3.6.8
CUDA	11.0

4.2 実験 1

実験 1 では、モデルのネットワーク構造によって学習時間がどのように変化するかを確認するために、同一の学習データに対し、表 4 から得られるすべての組み合わせ (72 通り) のネットワーク構造でそれぞれ 3 回学習を行い、かかった時間の平均値を取得し、比較した。その他の条件は、エポック数 100, バッチサイズ 100, ウィンドウ幅 200 で統一した。

表 4 実験 1 でのネットワーク構造探索範囲

パラメータ	範囲
レイヤ	LSTM または GRU
層数	1-3
ユニット数	1-12

実験結果は、図 5 のようになった。縦軸が 100 エポック

にかかった学習時間の平均値、横軸がユニット数である。

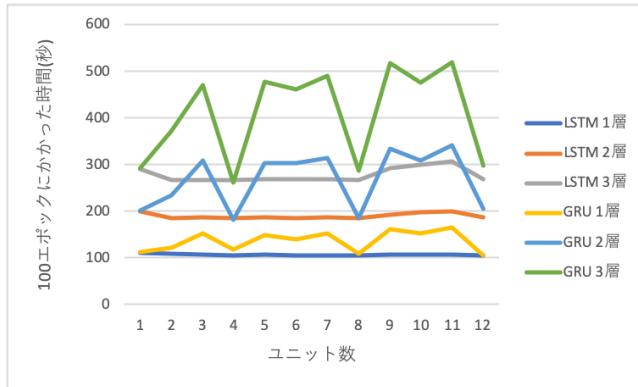


図 5 学習時間 (3 回の平均値)

この結果から、まず、同じユニット数同士の結果を比較すると、学習時間は LSTM, GRU とともに層数に比例していることが読み取れる。また、LSTM の学習時間はユニット数による変化が少ないのに対し、GRU の学習時間はユニット数が 4 の倍数の時に小さくなっており、全体的な学習時間は、LSTM の方が GRU より短いことも読み取れる。さらに、LSTM では、ユニット数が 9 以上になると学習時間が少し大きくなることも読み取れる。

4.3 実験 2

実験 2 では、モデルのネットワーク構造によって予測精度がどのように変化するかを確認するために、5 種類の学習データに対し、表 5 から得られるすべての組み合わせ (72 通り) のネットワーク構造で学習を行い、実際の値と学習を行ったモデルの予測精度比較した。その他の条件は、エポック数 100, バッチサイズ 100, ウィンドウ幅 200 で統一した。

表 5 実験 2 でのネットワーク構造探索範囲

パラメータ	範囲
レイヤ	LSTM または GRU
層数	1-3
ユニット数	1-12

モデルの評価指標として RMSE(Root Mean Squared Error) を使用した。RMSE は、二乗平均平方根誤差と呼ばれる回帰モデルの誤差を評価する指標の 1 つであり、この値が小さく 0 に近いほど精度が良いということになる。長さ n の時系列データの正解値を y_t , 予測値を \hat{y}_t とすると、以下のような式で計算される。

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2}$$

実験結果は、図 6,7,8,9,10 のようになった。縦軸が 100 個の時系列データの予測値と実際の値との RMSE の平均

値、横軸がユニット数である。

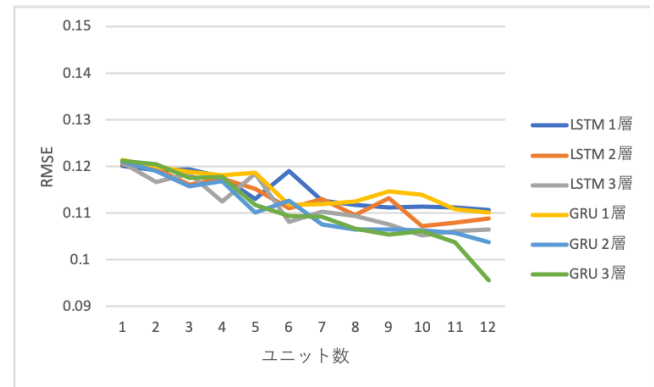


図 6 学習データ 1 での精度

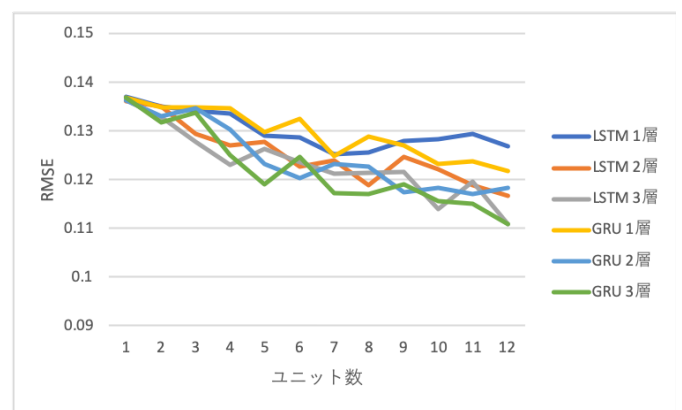


図 7 学習データ 2 での精度

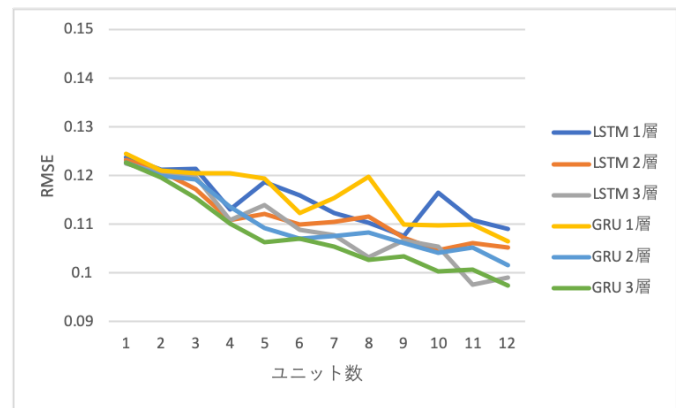


図 8 学習データ 3 での精度

これらの結果から、まず、GRU と LSTM では、同じ層数同士の値を比較すると、精度の差がほとんどないことが読み取れる。そして、同じレイヤ間での層数による値の差は、1 層は 2 層,3 層と比較して悪いが、2 層と 3 層の間では比較的差が少ないことが読み取れる。

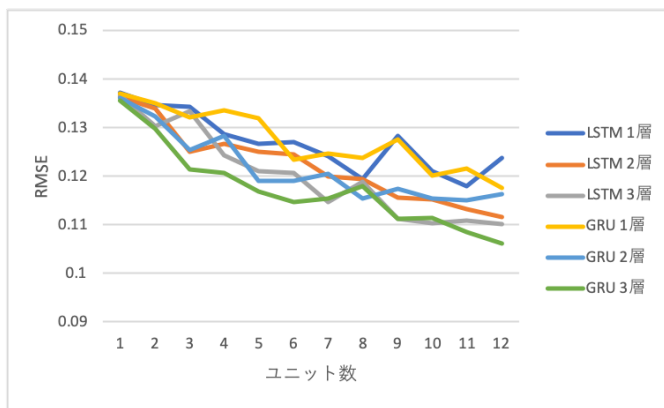


図 9 学習データ 4 での精度

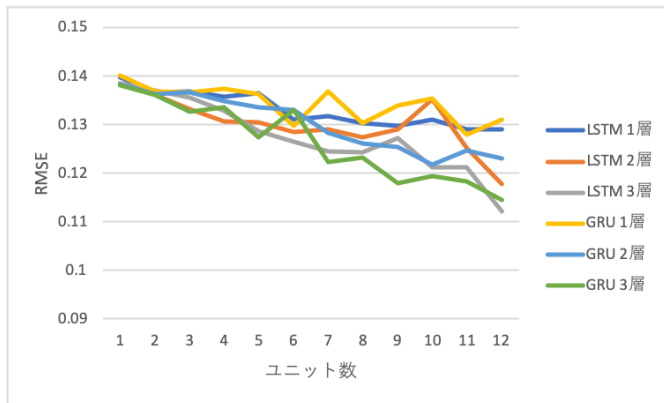


図 10 学習データ 5 での精度

5. NAS 探索範囲の方針

以上 2 つの実験の結果，NAS 探索範囲を以下の表 6 ように設定するのが良いと考えた。

パラメータ	範囲
レイヤ	LSTM(固定)
層数	2(固定)
ユニット数	8-12

レイヤを LSTM に固定する理由としては，学習時間が全体的に GRU より短く，ユニット数によるブレも少ないためである。層数を 2 に固定する理由としては，2 層→3 層では学習時間が 1.5 倍になるのに対し，精度の差が少ないためである。ユニット数は，同じ層数の中でも，大きいほど精度が良くなる傾向があるためである。

表 6 の範囲に事前に絞ることで，30 時間分のデータ 100 個に対して 100 エポックで学習を行う場合，1 回の NAS にかかる時間は合計 20 分程度に抑えられると見積もられる。これは，運用において 30 時間に 1 回行うと想定しても，十分に実現可能であると考えられる。

6. まとめと今後の予定

Neural Architecture Search を取り入れた時系列予測モデルの運用に向けて，モデルのネットワーク構造と学習時

間/予測精度の関係についての調査を行った。実験の結果，学習データによって最適な予測モデルのネットワーク構造が異なることが分かり，NAS の有効性を確認できたほか，使用するレイヤによる学習時間の違いや，ユニット数の違いによる精度の違いも確認できた。そして，実験結果から適切な NAS 探索範囲の方針を定め，その範囲で NAS を行った場合にかかる時間を見積もったところ，運用に取り入れることが十分可能であることが判明した。

今後は，NAS を取り入れた時系列予測モデル運用アルゴリズムの検討を行いたいと考えている。

謝辞

本研究の一部はお茶の水女子大学と富士通株式会社との共同研究契約に基づくものであり，JST CREST JP-MJCR1503 の支援を受けたものである。

参考文献

- [1] Josh Whitney and Pierre Delforge. Data center efficiency assessment. *Issue paper on NRDC (The Natural Resource Defense Council)*, 2014.
- [2] Rahul Ghosh and Vijay K Naik. Biting off safely more than you can chew: Predictive analytics for resource over-commit in iaas cloud. In *2012 IEEE Fifth International Conference on Cloud Computing*, pages 25–32. IEEE, 2012.
- [3] 児玉宏喜, 鈴木成人, 福田裕幸, 吉田英司, et al. マイグレーションを利用したデータセンターの高効率運用手法の提案とオーバコミット時における vm の性能評価. *研究報告システムソフトウェアとオペレーティング・システム (OS)*, 2018(13):1–7, 2018.
- [4] Hiroyoshi Kodama, Hiroshi Endo, Shigeto Suzuki, and Hiroyuki Fukuda. High efficiency cloud data center management system using live migration. In *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, pages 733–738. IEEE, 2017.
- [5] Salam Ismaeel and Ali Miri. Real-time energy-conserving vm-provisioning framework for cloud-data centers. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0765–0771. IEEE, 2019.
- [6] Thomas Elsken, Jan Hendrik Metzen, Frank Hutter, et al. Neural architecture search: A survey. *J. Mach. Learn. Res.*, 20(55):1–21, 2019.
- [7] Haifeng Jin, Qingquan Song, and Xia Hu. Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1946–1956. ACM, 2019.
- [8] Autokeras. <https://autokeras.com/>.
- [9] Mohammad Shahrads, Rodrigo Fonseca, Íñigo Goiri, Gohar Chaudhry, Paul Batum, Jason Cooke, Eduardo Laureano, Colby Tresness, Mark Russinovich, and Riccardo Bianchini. Serverless in the wild: Characterizing and optimizing the serverless workload at a large cloud provider. *arXiv preprint arXiv:2003.03423*, 2020.
- [10] Azure/azurepublicdataset: Microsoft azure traces. <https://github.com/Azure/AzurePublicDataset>.
- [11] Home - keras documentation. <https://keras.io/ja/>.