1　データと手法

1.1　データ

1.1.1　P-POTEKA データセット

日本と開発途上国との国際科学技術協力の強化や地球規模課題の解決、科学技術水準の向上につながる新たな知見や技術の獲得等を目的とした地球規模課題対応国際科学技術協力プログラム（SATREPS, Science and Technology Research Partnership for Sustainable Development ）のプロジェクトの 1 つに、ULAT（Understanding Lightning and Thunderstorms Project）がある。ULAT は東南アジアを中心に大規模な災害を引き起こしている雷雨や台風の高精度な活動把握や予測を目的としたプロジェクトである。このプロジェクトにて、我々の研究グループは P-POTEKA と呼ばれる自動気象観測装置（図 1）を用いてマニラ首都圏を中心にフィリピン全土をカバーする 35 カ所の雷・気象観測網（図 2）を構築した。世界的にも類を見ないほどの高密度観測網であり、大規模な災害を引き起こす雷雨や台風による降雨やそれに伴って変化する気温や湿度などの気象データを 1 分の時間解像度で観測が可能である。P-POTEKA から得られるデータを表 1 に示す。本研究では時間雨量・気温・湿度、風向と風速から計算される東西/南北風成分のデータを用いた。表 1 の降水量は 1 分間の降水量を示す。過去 60 分間の降水量を積算した時間雨量を降雨データとして用いた。最終的に 2019 年 10 月から 2020 年 10 月の 1 年間の P-POTEKA の観測データを用いて、10 分間隔の時系列気象データを作成した。

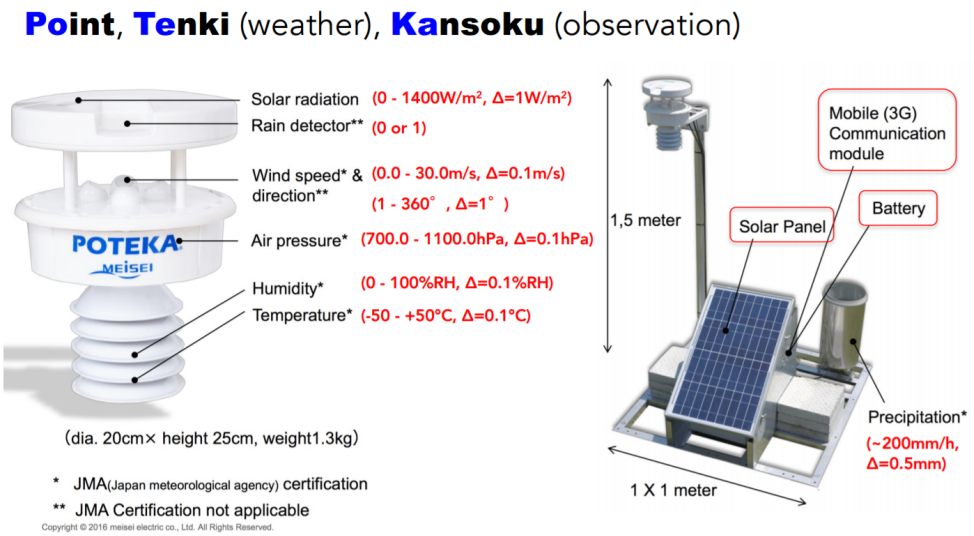

図 1: P-POTEKA 観測装置。左図は気象センサを搭載した部分で、各部分で観測している気象データを示す。右図は観測装置全体。

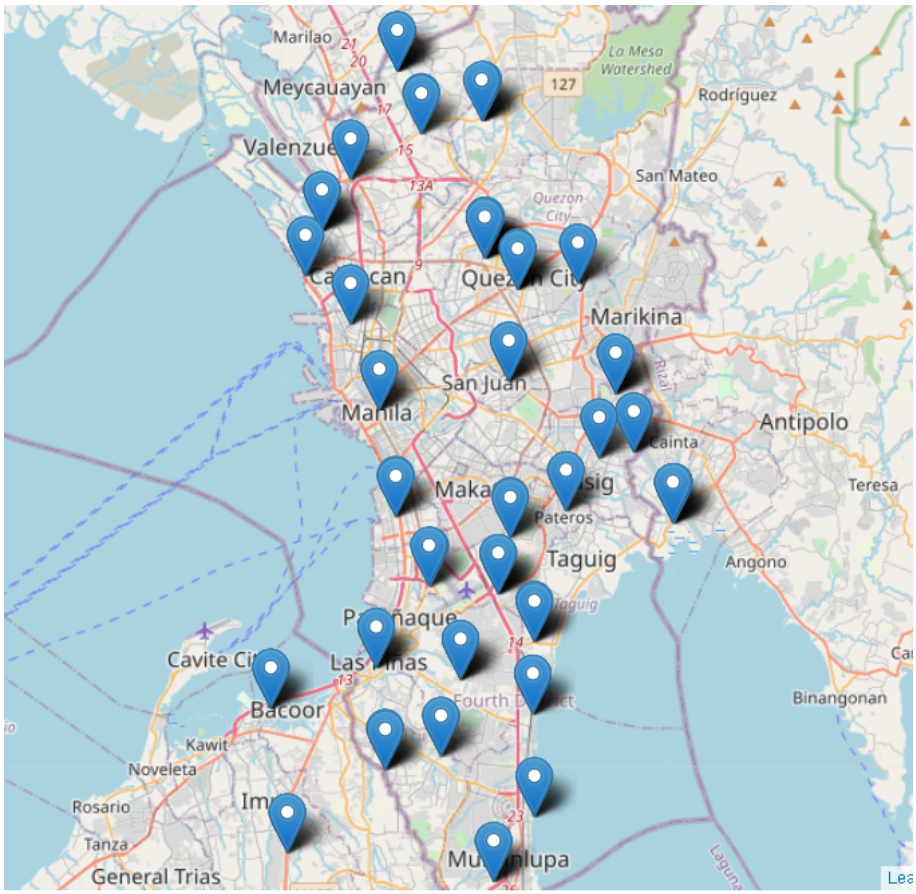

図 2: マニラ首都圏の P-POTEKA 分布図。

データ名 [単位]	最小値	最大値	解像度
日射 [W/m^2]	0	1400	1
感雨	0	1	1
風速 [m/s]	0	30	0.1
風向 [°]	1	360	1
気圧 [hPa]	700	1100	0.1
湿度 [%RH]	0	100	0.1
気温 [°C]	-50	50	0.1
降水量 [mm]	0	200	0.5

表 1: P-POTEKA が観測する気象データ一覧。

さらに観測データのうち、時間雨量の大きかった降雨イベントの日・最大時間雨量・熱帯低気圧などの情報を表 2 に示す。

最大時間雨量の観測日時	最大時間雨量 [mm/h]	熱帯低気圧等の情報
2020/08/01 5:00 (UTC)	88.0	台風 Bagyong Dindo が付近を通過。
2020/10/12 8:10 (UTC)	82.0	熱帯低気圧 Nika が付近を通過。
2020/07/27 12:20 (UTC)	80.0	特になし。
2020/07/12 22:00 (UTC)	76.5	特になし。
2020/09/14 6:00 (UTC)	76.0	熱帯低気圧 Leon が付近を通過。
2020/08/07 6:00 (UTC)	72.5	熱帯低気圧 Enteng が付近を通過。
2020/07/04 8:30 (UTC)	60.0	特になし。
2019/10/14 6:10 (UTC)	63.5	特になし。

表 2: P-POTEKA が観測した非常に強い降雨イベントの例。PAGASA（フィリピン大気地球天文局）の台風・熱帯低気圧名を使用。

1.1.2 内挿処理

P-POTEKA のデータは 35 個の観測点におけるそれぞれの位置での気象データである。本研究では機械学習モデルが学習可能なデータ形式とするために、生の P-POTEKA に対してガウス過程回帰と呼ばれる手法を用いてグリッドデータに変換した。

まずガウス過程とは関数 $f(x)$ を確率変数と見立てた確率分布のことである。そしてガウス過程回帰とは、データから関数 $f(x)$ の確率分布をガウス過程の形で求める方法のことである。関数 $f(x)$ の任意個数の入力点 $x_1, ..., x_N$

に対する出力 $f(x_1), ..., f(x_N)$ が N 次元の多変量ガウス分布でモデル化される。したがってその予測値 $f(x_*)$ もガウス分布に従い、期待値と分散で表現される。ガウス過程回帰では予測値の不確実性を分散で評価できるという特徴を持つ。さらに線形回帰などとは異なり、非線形な関数に対してもフィッティングできる柔軟性を持つ。

多変量ガウス分布とは D 次元のベクトル $\boldsymbol{x} = (x_1, ..., x_D)$ の変数がそれぞれ独立にガウス分布に従う場合における同時分布のことである。ベクトル $\boldsymbol{x}$ が平均 $\boldsymbol{\mu}$、共分散行列 $\boldsymbol{\Sigma}$ のガウス分布 $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ に従っているとき、確立密度関数は以下のように表される。

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\} \tag{1}$$

$\boldsymbol{\mu} = (\mu_1, ..., \mu_N)$ は $\boldsymbol{x}$ の期待値を表す平均ベクトル、Σ は $D \times D$ の共分散行列でその要素 (i, j) が x_i と x_j の共分散を表している。

一般的な線形回帰モデルでは基底関数 $\phi_0(\boldsymbol{x}), \phi_1(\boldsymbol{x}), ..., \phi_H(\boldsymbol{x})$ によって $\boldsymbol{x}$ の写像を並べた特徴ベクトル $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_0(\boldsymbol{x}), \phi_1(\boldsymbol{x}), ..., \phi_H(\boldsymbol{x}))^T$ とパラメータ $\boldsymbol{w}$ を用いて以下のように表される。

$$\hat{y} = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) \tag{2}$$

$\hat{y}$ は観測データ y に近づけるあてはめ値である。複数のデータ点 $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$ の場合に拡張すると以下のような行列形式で表すことができる。

$$\begin{bmatrix} \hat{y_1} \\ \hat{y_2} \\ \cdots \\ \hat{y_N} \end{bmatrix} = \begin{pmatrix} \phi_0(\boldsymbol{x}_1) & \phi_1(\boldsymbol{x}_1) & \ldots & \phi_H(\boldsymbol{x}_1) \\ \phi_0(\boldsymbol{x}_2) & \phi_1(\boldsymbol{x}_2) & \ldots & \phi_H(\boldsymbol{x}_2) \\ \ldots & \ldots & \ddots & \vdots \\ \phi_0(\boldsymbol{x}_N) & \phi_1(\boldsymbol{x}_N) & \ldots & \phi_H(\boldsymbol{x}_N) \end{pmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ \vdots \\ w_H \end{bmatrix} \tag{3}$$

この方法では入力 $\boldsymbol{x}$ の次元数が小さい場合でしか使えないという問題がある。なぜなら学習すべきパラメータ $\boldsymbol{w}$ の次元が入力次元の増大に対して指数的に増加してしまうためである。この問題は一般的に次元の呪いと呼ばれている。ガウス過程回帰ではパラメータ $\boldsymbol{w}$ に対して期待値をとり、積分消去することでこの問題を回避している。$\hat{\boldsymbol{y}} = (\hat{y_1}, ..., \hat{y_N})$、$\boldsymbol{w} = (w_0, ..., w_H)$、および $\Phi_{nh} = \phi_h(\boldsymbol{x}_n)$ を要素とする計画行列 $\boldsymbol{\Phi}$ を用いると、先ほどの行列形式は以下のように書ける。

$$\hat{\boldsymbol{y}} = \boldsymbol{\Phi}\boldsymbol{w} \tag{4}$$

ガウス過程では重み $\boldsymbol{w}$ が平均 $\boldsymbol{0}$、分散 $\lambda^2\boldsymbol{I}$ のガウス分布 (計算式 5) から生成されたものだと仮定する。

$$\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \lambda^2\boldsymbol{I}) \tag{5}$$

この場合、式 (4) の $\hat{\boldsymbol{y}}$ はガウス分布に従うベクトル $\boldsymbol{w}$ を定数行列 $\boldsymbol{\Phi}$ で線形変換したものとなっている。したがって $\hat{\boldsymbol{y}}$ もガウス分布に従う。$\hat{\boldsymbol{y}}$ の期待値、共分散行列は以下のように求められる。

$$\mathbb{E}[\hat{\boldsymbol{y}}] = \mathbb{E}[\boldsymbol{\Phi}\boldsymbol{w}] = \boldsymbol{\Phi}\mathbb{E}[\boldsymbol{w}] = \boldsymbol{0} \tag{6}$$

$$\begin{aligned} \Sigma &= \mathbb{E}[\hat{\boldsymbol{y}}\hat{\boldsymbol{y}}^T] - \mathbb{E}[\hat{\boldsymbol{y}}]\mathbb{E}[\hat{\boldsymbol{y}}]^T \\ &= \mathbb{E}[(\boldsymbol{\Phi}\boldsymbol{w})(\boldsymbol{\Phi}\boldsymbol{w})^T] \\ &= \boldsymbol{\Phi}\mathbb{E}[\boldsymbol{w}\boldsymbol{w}^T]\boldsymbol{\Phi}^T \\ &= \lambda^2\boldsymbol{\Phi}\boldsymbol{\Phi}^T \end{aligned} \tag{7}$$

結果として $\hat{\boldsymbol{y}}$ の分布は、多変量ガウス分布となることがわかる (計算式 8)。

$$\hat{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{0}, \lambda^2\boldsymbol{\Phi}\boldsymbol{\Phi}^T) \tag{8}$$

この式からわかるように重み $\boldsymbol{w}$ は期待値がとられることによって消去されている。したがって入力 $\boldsymbol{x}$ や $\phi(\boldsymbol{x})$ の次元がどれだけ高くとも、対応する高次元の重み $\boldsymbol{w}$ を求める必要はなく、$\boldsymbol{y}$ の分布はデータ数 N に依存する共分散行列 $\lambda^2\boldsymbol{\Phi}\boldsymbol{\Phi}^2$ で決まる。

式 (8) の共分散行列を以下のようにおく。

$$\boldsymbol{K} = \lambda^2\boldsymbol{\Phi}\boldsymbol{\Phi}^T \tag{9}$$

$\boldsymbol{K}$ の (n, n') 要素は以下のように与えられる。

$$K_{nn'} = \lambda^2\phi(\boldsymbol{x}_n)^T\phi(\boldsymbol{x}_{n'}) \tag{10}$$

すなわち $\boldsymbol{x}_n$ と $\boldsymbol{x'}_{n'}$ の特徴ベクトル $\boldsymbol{\phi}(\boldsymbol{x}_n)$ と $\boldsymbol{\phi}(\boldsymbol{x}_{n'})$ の内積の定数倍が共分散行列 $\boldsymbol{K}$ の (n, n') 要素 $K_{nn'}$ になっている。多変量ガウス分布において 2 つの変数間の共分散が大きいということは似た値を取りやすいことを意味する。したがって、$\boldsymbol{x}_n$ と $\boldsymbol{x}_{n'}$ が似ているなら、対応する $\hat{y_n}$ と $\hat{y_{n'}}$ も似た値を持つことになる。さらにガウス過程回帰では共分散行列 $\boldsymbol{K}$ の要素 $\phi(\boldsymbol{x}_n)^T\phi(\boldsymbol{x}_{n'}$ を明示的に求めることはせず、カーネル関数 $k(\boldsymbol{x}_n, \boldsymbol{x}_{n'})$ で以下のように置き換える。

$$k(\boldsymbol{x}_n, \boldsymbol{x}_{n'}) = \phi(\boldsymbol{x}_n)^T\phi(\boldsymbol{x}_{n'}) \tag{11}$$

こうすることで特徴ベクトル $\phi(\boldsymbol{x})$ を明示的に表現することなしに、カーネル関数のみに基づく簡単な計算で計画行列を求めることができる。カーネル関数を適切に設計することで、$\boldsymbol{K}$ の逆行列が計算できない場合を避けることができる。代表的なカーネル関数として線形カーネル (式 12) や指数カーネル (式 13, θ はパラメータ) がある。カーネル関数はデータ間の距離に応じた出力の類似度を表現していることから、出力の特徴に応じて設計可能である。データに関する事前情報を反映できることを意味し、これによってより柔軟なモデルの構築を可能とする。

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \boldsymbol{x}^T\boldsymbol{x'} \tag{12}$$

$$k(\boldsymbol{x}, \boldsymbol{x'}) = exp(-|\boldsymbol{x} - \boldsymbol{x'}|/\theta) \tag{13}$$

ここで以下の N 個のデータ点がすでに観測されているとする。

$$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_N, y_N)\} \tag{14}$$

今までの議論により、ガウス過程回帰において $\boldsymbol{y}$ はガウス分布に従い、データ $\mathcal{D}$ とカーネル関数から求められる共分散行列 $\boldsymbol{K}$ を用いて以下のように表される。

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}) \tag{15}$$

この分布を用いて新しい観測点 $\boldsymbol{X}^* = (\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, ..., \boldsymbol{x}_M^*)$ での予測値 $\boldsymbol{y}^* = (y_1^*, y_2^*, ..., y_M^*)$ を求めたい場合を考える。$\boldsymbol{y}^*$ も多変量ガウス分布に従うことからすでに得られている分布 $\boldsymbol{y}$ との同時分布もガウス分布に従うことが分かる。この同時分布を以下に示す。

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \\ y_1^* \\ \vdots \\ y_M^* \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{pmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_1, \boldsymbol{x}_N) & k(\boldsymbol{x}_1, \boldsymbol{x}_1^*) & \dots & k(\boldsymbol{x}_1, \boldsymbol{x}_M^*) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) & k(\boldsymbol{x}_N, \boldsymbol{x}_1^*) & \dots & k(\boldsymbol{x}_N, \boldsymbol{x}_M^*) \\ k(\boldsymbol{x}_1^*, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_1^*, \boldsymbol{x}_N) & k(\boldsymbol{x}_1^*, \boldsymbol{x}_1^*) & \dots & k(\boldsymbol{x}_1^*, \boldsymbol{x}_M^*) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_M^*, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_M^*, \boldsymbol{x}_N) & k(\boldsymbol{x}_M^*, \boldsymbol{x}_1^*) & \dots & k(\boldsymbol{x}_M^*, \boldsymbol{x}_M^*) \end{pmatrix} \right) \tag{16}$$

ガウス過程回帰における予測とは、新しい観測点 $\boldsymbol{X}^*$ における出力 $\boldsymbol{y}^*$ の周辺分布 $p(\boldsymbol{y}^*|\boldsymbol{X}^*, \mathcal{D})$ を求めることである。同時分布がガウス分布に従っていることから周辺分布もガウス分布となることがわかる。予測値を求めていることから、この周辺分布は予測分布とも呼ばれる。以下のように計算される。

$$p(\boldsymbol{y}^*|\boldsymbol{X}^*, \mathcal{D}) = \mathcal{N}(\boldsymbol{k}_*^T\boldsymbol{K}^{-1}\boldsymbol{y}, \boldsymbol{k}_{**}\boldsymbol{K}^{-1}\boldsymbol{k}_*) \tag{17}$$

ただし $\boldsymbol{k}_*(n,m) = k(\boldsymbol{x}_n, \boldsymbol{x}^*_m)$、$\boldsymbol{k}_{**}(m,m') = k(\boldsymbol{x}_m, \boldsymbol{x}^*_{m'})$ である。

本研究においては、$\boldsymbol{x}$ は緯度・経度の 2 次元で y は地点 $\boldsymbol{x}$ における気象パラメータである。また、気象パラメータの特徴ごとにカーネル関数を設定した。具体的には、雨や風はデータの距離が近いほど似たデータになるが距離が大きくなると途端に類似度は下がると仮定し指数カーネルを用いた。気温や湿度など雨や風程には地域間で差が生じないようなデータには線形カーネルを用いた。最終的に P-POTEKA の設置個所を含めた東経 120°9′ 121°15′、北緯 14°35′ 14°76′ の範囲における $H \times W = 50 \times 50$ のグリッドデータを作成した。

1.2 Self-Attention ConvLSTM

1.2.1 Self-Attention 機構

空間データにおける Self-Attention 機構とは、ある点の値をその周りすべての点の加重和で表現するための機構となっている。従来の畳み込み処理でも似たような処理が行われるが、すべての点ではなく一定領域の点のみが畳み込みに用いたられる。したがって空間全体の情報を取り込むことができなかった。しかし Self-Attention 機構では対象データ点を除く空間内すべてのデータ点の情報を取り込むことでより空間全体の特徴を捉えることが可能になった。

Self-Attention 機構は空間全体との関連度合いを計算して、入力データを重みづけして出力する処理を行っている。したがって入力と出力の形式は同じ変換処理となる。計算の概要図を 3 左側に示す。

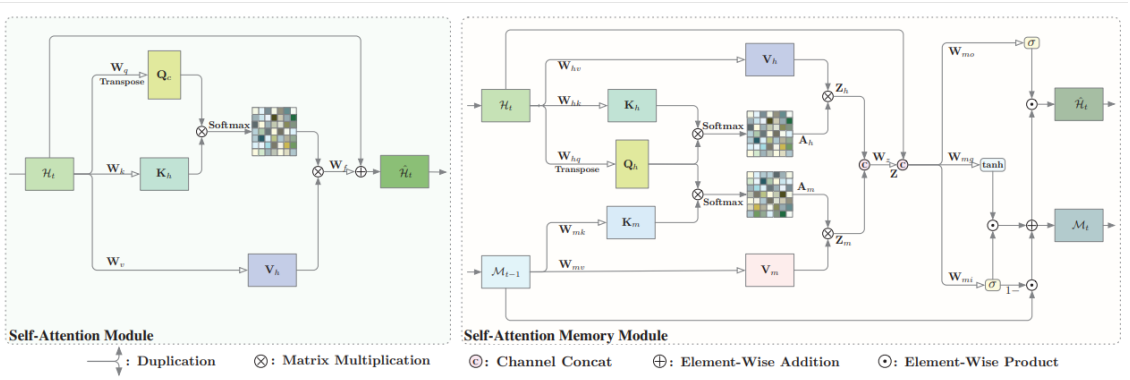

図 3: Lin *et al.*[2020] Figure1 より引用。Self-Attention 機構の計算概要図（左側）と Self-Attention メモリー機構の計算概要図（右側）。

Self-Attention 機構では入力 $\mathcal{H}_t$ に対して以下の 3 つの特徴空間が計算される。

$$\boldsymbol{query}: \boldsymbol{Q}_h = \boldsymbol{W}_q \mathcal{H}_t \in \mathbb{R}^{\hat{C} \times N} \tag{18}$$

$$\boldsymbol{key}: \boldsymbol{K}_h = \boldsymbol{W}_k \mathcal{H}_t \in \mathbb{R}^{\hat{C} \times N} \tag{19}$$

$$\boldsymbol{value}: \boldsymbol{V}_v = \boldsymbol{W}_v \mathcal{H}_t \in \mathbb{R}^{C \times N} \tag{20}$$

$\boldsymbol{W}_q$、$\boldsymbol{W}_q$、$\boldsymbol{W}_q$ は入力 $\mathcal{H}_\sqcup$ の全データ点における $\boldsymbol{query}$、$\boldsymbol{key}$、$\boldsymbol{value}$ の重みである。また C と $\hat{C}$ は入力データのチャンネル数である。N は入力データの総データ点数（$H \times W$）である。Self-Attention 機構ではあるデータ点同士の類似度 $\boldsymbol{e}$ が以下のように計算される。

$$\boldsymbol{e} = \boldsymbol{Q}_h^T \boldsymbol{K}_h \in \mathbb{R}^{N \times N} \tag{21}$$

したがって i 番目のデータ点と j 番目のデータ点との関連度は $e_{i,j} = (\mathcal{H}_{t,i}^T \boldsymbol{W}_q^T)(\boldsymbol{W}_k \mathcal{H}_{t,j})$ となる。$\mathcal{H}_{t,j}$ と $\mathcal{H}_{t,j}$ はそれぞれ i 番目と j 番目のチャンネルに対応したベクトルである。さらに以下のように正規化することで最終的な関連度が得られる。

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{N} \exp(e_{i,k})}, i, j \in 1, 2, ..., N. \tag{22}$$

i 番目のデータ点における Self-Attention 機構の最終的な出力は以下のように表される。

$$\boldsymbol{Z}_i = \sum_{j=1}^{N} \alpha_{i,j} (\boldsymbol{W}_v \mathcal{H}_{t;j}) \tag{23}$$

$\boldsymbol{W}_v \mathcal{H}_{t;j} \in \mathbb{R}^{C \times 1}$ は $\boldsymbol{value:V_h}$ の j 番目のチャンネルに対応したベクトルである。最終的に i 番目のデータが空間全体のデータ点に対する関連度合いで重みづけされて変換されていることがわかる。

1.2.2 Self-Attention メモリーモジュール

未来の状態を予測する際に過去の情報を用いることは非常に重要である。Lin *et al.*[2020] らは Self-Attention 機構を時空間モデルである ConvLSTM と組み合わせるために Self-Attention 機構の情報を時系列で扱うための Self-Attention メモリー機構（SAM）を提案した。計算の概要は図 3 右側に示されている。

SAM は時刻 t における入力 $\mathcal{H}_t$ とひとつ前の時刻 $t-1$ における SAM 自身の状態 $\mathcal{M}_{t-1}$ の 2 つを入力として受け取る。上記の計算過程は 3 つのステップ（特徴量抽出、メモリー状態更新、出力）に分かれている。

特徴量抽出ステップ: Self-Attention メモリー機構（図 3）の $\boldsymbol{Z}$ は $\boldsymbol{Z}_h$ と $\boldsymbol{Z}_m$ を合成することで得られる。$\boldsymbol{Z}_h$ は入力 $\mathcal{H}_t$ に対する Self-Attention 機構の計算処理を経て計算される。$\boldsymbol{Z}_m$ は $\mathcal{M}_{t-1}$ に対する特徴量抽出によって計算される。具体的には入力された $\mathcal{M}_{t-1}$ と重み $\boldsymbol{K}_{mk}$・$\boldsymbol{K}_{mv}$ から $\boldsymbol{key:K_m}$ と $\boldsymbol{value:V_m}$ が計算される。そして入力 $\mathcal{H}_t$ と過去のメモリー $\mathcal{M}_{t-1}$ に対する関連度が以下のように計算される。

$$\boldsymbol{e}_m = \boldsymbol{Q}_h^T \boldsymbol{K_m} \in \mathbb{R}6N \times N. \tag{24}$$

この関連度は Self-Attention 機構の場合と同様に正規化され以下のようになる。

$$\alpha_{i,j} = \frac{\exp(e_{m;i,j})}{\sum_{k=1}^{N} \exp(e_{m;i,k})}, i, j \in 1, 2, ..., N \tag{25}$$

そして i 番目のデータ点における特徴量 $\boldsymbol{Z_m}$ は過去のメモリー状態 $\mathcal{M}_{t-1}$ を先ほど計算した関連度を用いて変換することで得られる。

$$\boldsymbol{Z}_{m;i} = \sum_{j=1}^{N} \alpha_{m;i,j} \boldsymbol{V}_{m;j} = \sum_{j=1}^{N} \alpha_{m;i,j} \boldsymbol{W}_{mv} \mathcal{M}_{t-1;j} \tag{26}$$

$\mathcal{M}_{t-1;j}$ は j 番目のデータ点に対応する過去のメモリー状態である。最終的に特徴量 $\boldsymbol{Z}$ は以下のように得られる。

$$\boldsymbol{Z} = \boldsymbol{W}_z[\boldsymbol{Z}_h; \boldsymbol{Z}_m] \tag{27}$$

メモリー状態更新ステップ: 過去の情報を効率よく保持しておくために LSTM モデルでも用いられているゲートを応用した。具体的な計算式を以下に示す。

$$i'_t = \sigma(W_{m;zi} * \boldsymbol{Z} + W_{m;hi} * \mathcal{H}_t + b_{m;i}) \tag{28}$$

$$g'_t = tanh(W_{m;zg} * \boldsymbol{Z} + W_{m;hg} * \mathcal{H}_t + b_{m;g}) \tag{29}$$

$$\mathcal{M}_t = (1 - i'_t) \circ \mathcal{M}_{t-1} + i'_t \circ g'_t \tag{30}$$

入力ゲート i'_t と新たな合成された特徴量 g'_t の計算には抽出された特徴量 $\boldsymbol{Z}$ と入力 $\mathcal{H}_t$ が用いられる。$b_{m;i}$ と $b_{m;g}$ はバイアス項である。加えて計算量を減らすために忘却ゲートを $(1 - i'_t)$ で代用した。忘却ゲートで過去のメモリー状態を調整し、さらにそこに入力ゲートで調査された特徴量 g'_t を足すことで新たなメモリー状態に更新している。

出力ステップ: 最終的な Self-Attention メモリー機構の出力 $\hat{\mathcal{H}}_t$ は出力ゲート o'_t と更新されたメモリ $\mathcal{M}_t$ のドット積から得られる。

$$o'_t = \sigma(W_{m;zo} * \boldsymbol{Z} + W_{m;ho} * \mathcal{H}_t + b_{m;o}) \tag{31}$$

$$\hat{\mathcal{H}}_t = o'_t \circ \mathcal{M}_t \tag{32}$$

1.2.3 Self-Attention ConvLSTM

最終的に Self-Attention メモリー機構は図 4 のように ConvLSTM に組み込まれる。

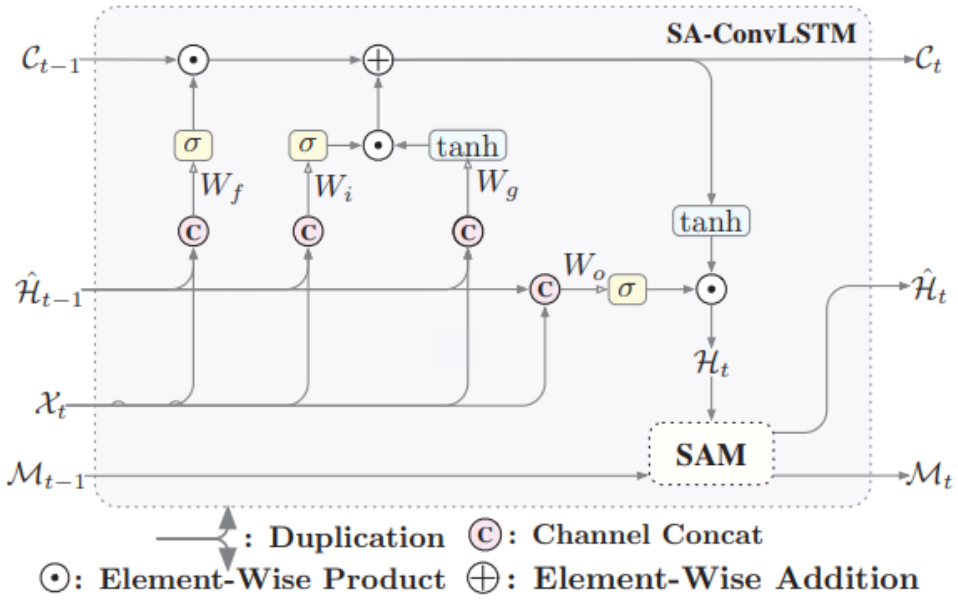

図 4: Lin *et al.*[2020] Figure2 より引用。Self-Attention ConvLSTM の計算概要図。

具体的には以下のように計算される。

$$\left.\begin{aligned} i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot \mathcal{C}_{t-1} + b_i) \\ f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot \mathcal{C}_{t-1} + b_f) \\ \mathcal{C}_t &= f_t \odot \mathcal{C}_{t-1} + i_t \odot \tanh(W_{xc} * X_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \\ o_t &= \sigma(W_{xo} * X_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \odot \mathcal{C}_t + b_o) \\ \mathcal{H}_t &= o_t \odot \tanh(\mathcal{C}_{\sqcup}) \end{aligned}\right\} \text{ConvLSTM の処理} \quad (33)$$

$$\left.\begin{aligned} i'_t &= \sigma(W_{m;zi} * \boldsymbol{Z} + W_{m;hi} * \mathcal{H}_t + b_{m;i}) \\ g'_t &= tanh(W_{m;zg} * \boldsymbol{Z} + W_{m;hg} * \mathcal{H}_t + b_{m;g}) \\ \mathcal{M}_t &= (1 - i'_t) \circ \mathcal{M}_{t-1} + i'_t \circ g'_t \\ o'_t &= \sigma(W_{m;zo} * \boldsymbol{Z} + W_{m;ho} * \mathcal{H}_t + b_{m;o}) \\ \hat{\mathcal{H}}_t &= o'_t \circ \mathcal{M}_t \end{aligned}\right\} \text{Self-Attention メモリー機構の処理} \quad (34)$$

1.3 学習と評価

1.3.1 学習

学習データとして内挿処理を施した 10 分間隔の P-POTEKA のグリッドデータを用いた。気象データ各々は最大値 1、最小値 0 に正規化した。入力と出力のペア（例；入力 1 時間 6 ステップ、出力 1 時間 6 ステップ）をサブセットとして時間をずらしながらこのサブセットを大量に作成した。さらにこのサブセットの集合を学習用・評価用・検証用データセットに分割した。比率はそれぞれ学習

用データセットが 80%、評価用データセットが 15%、検証用データセットが 5%とした。モデルの学習に学習用データセットを用いて、同時並行でモデルの汎化性能を評価するために評価用データセットを用いた。最後に 2 から強い降雨イベントを取り出た検証用データセットを用いて SelfAttention ConvLSTM の性能や内部状態を検証した。

学習では Self-Attention ConvLSTM を 4 層に重ねたモデルを用いた。さらに各々の Self-Attention ConvLSTM の隠れ状態の数は 64 に設定した。損失関数にはバイナリークロスエントロピー誤差関数を用いた。最適化関数には学習率を 0.0001 に設定した Adam 関数を用いた。バイナリークロスエントロピー誤差関数は正規化された値に対する誤差を評価するのに適した関数で以下のように定義される。

$$l_i = -ylog\hat{y} - (1-y)log(1-\hat{y}), i \in 1, 2, ..., N \tag{35}$$

N は対象データ点の数、y はラベル（正解値）、$\hat{y}$ は予測値である。実際の損失にはすべての与えられたデータ点におけるラベルと予測値から上記の値を計算しその平均値をとる。最終的な損失 L は以下のようになる。

$$L = \frac{\sum_{i=1}^{N} l_i}{N} \tag{36}$$

また、Adam 関数は確率的勾配降下法における機械学習モデルの重み $\boldsymbol{W}$ 更新方法の 1 つである。確率的というのは、重みの更新にすべてのデータを用いず、ランダムに抽出されたデータを用いるということである。Adam 関数は損失 L と学習率 η を用いて以下のように定義される。

$$\boldsymbol{m}_{t+1} = \beta_1 \boldsymbol{m}_t + (1-\beta_1)\frac{\partial L}{\partial \boldsymbol{W}_t} \tag{37}$$

$$\boldsymbol{v}_{t+1} = \beta_2 \boldsymbol{m}_t + (1-\beta_2)\frac{\partial L}{\partial \boldsymbol{W}_t} \odot \frac{\partial L}{\partial \boldsymbol{W}_t} \tag{38}$$

$$\hat{\boldsymbol{m}}_{t+1} = \frac{\boldsymbol{m}_{t+1}}{1-\beta_1^t} \tag{39}$$

$$\hat{\boldsymbol{v}}_{t+1} = \frac{\boldsymbol{v}_{t+1}}{1-\beta_2^t} \tag{40}$$

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \frac{1}{\sqrt{\hat{\boldsymbol{v}}_{t+1}} + \epsilon} \odot \hat{\boldsymbol{m}}_{t+1} \tag{41}$$

t は更新ステップを示す。実際の学習ではランダムに抽出されたデータが複数作成され、逐次的に重みが更新されていく。$t+1$ は t に対して次の更新ステップであることを示す。β_1 と β_2 は減衰率パラメータであり 0 から 1 の範囲をとる。

1.3.2 評価

このモデルの予測値は入力と同じ $H \times W = 50 \times 50$ のグリッドデータである。このグリッドデータからある観測点での予測値を得るために、その観測点を含むグリッドを中心として周囲 3×3 のグリッドの予測値の平均値を計算した。検証ステップにおいて予測値はすべてこの変換後の値を用いた。モデルの元の予測値（グリッドデータ）を Y とする。(i, j) のグリッドがある観測点 k を含んでいるとすると、この観測点における予測値は以下のように計算される。

$$\hat{y}_k = \frac{\sum_{s=i-1}^{i+1} \sum_{t=j-1}^{j+1} Y_{i,j}}{9}, \quad i, j \in \{1, ..., 50\} \tag{42}$$

検証では予測誤差を評価するために平均平方二乗誤差（RMSE）を用いた。k 番目の観測点における実測値を y_k、観測点の総数を N とおく。RMSE は以下のように計算される。

$$RMSE = \sqrt{\frac{\sum_{k=1}^{N}(y_k - \hat{y}_k)}{N}} \tag{43}$$

参考文献

[1] Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. NIPS 2015, 802–810.

[2] Lin Z.; Li M.; Zheng Z.;, Cheng Y.; and Yuan C. 2020. Self-Attention ConvLSTM for Spatiotemporal Prediction. Association for the Advancement of Artificial Intelligence

[3] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need., NIPS, 2017.

[4] Su A.; Li H.; Cui L.; and Chen Y. 2020. A Convection Nowcasting Method Based on Machine Learning. Hindawi.

[5] Hubel D.H. and Wiesel T.N. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology.

[6] Hochreiter S. and Schmidhuber J. 1997. Long Short-Term Memory. Neural Computation.

[7] Ko C.M.; Jeong Y.Y.; Lee Y.M.; and Kim B.S. 2020. The Development of a Quantitative Precipitation Forecast Correction Technique Based on Machine Learning for Hydrological Applications. MDPI.

[8] 気象庁. 2018. 気候変動監視レポート 2018 世界と日本の気候変動および温室効果ガスとオゾン層等の状況. 気象庁.

[9] 気象庁. 2020. 気象庁業務評価レポート（令和 2(2020) 年度板）. 気象庁.

[10] 佐藤正樹;. 2020. 近年における降雨状況の実態：極端豪雨は増えているか 水環境学会誌 第 43 巻（A）第 5 号 pp.142 147. 公益社団法人 日本水環境学会.

[11] Fumikai F.; Nobuo Y.; and Kenji K. 2006. Long-Term Changes of Heavy Precipitation and Dry Weather in Japan(1901-2004). Meteorological Society of Japan.