

令和5年度 修士論文

格子ボルツマン法を組み込んだ
深層学習モデルによる日本近辺の風速予測

指導教諭 澤田 秀之 教授

2024年1月26日提出

早稲田大学 先進理工学研究科
物理学及応用物理学専攻
澤田秀之研究室 修士2年

5322A083-2 山倉 拓也

目次

第 1 章	提案モデルによる日本近辺の風速予測	2
1.1	実験の概要	2
1.2	利用したデータ及び学習条件	2
1.3	物理的な構造を含まない深層学習モデル	4
1.4	評価指標	5
1.5	実験結果	8
1.6	考察	8
参考文献		12

第 1 章

提案モデルによる日本近辺の風速予測

本論文では、提案モデルの性能検証のために日本近辺の風速予測を実施した。また、性能の比較のために CNN によるエンコーダ・デコーダと LSTM を用いたエンコーダ・デコーダモデルを実装し、それぞれのモデルの性能を比較した。

1.1 実験の概要

本実験では、提案モデルと 1.3 節で説明するエンコーダ・デコーダモデルの 2 つのモデルに 3 時刻分の時系列データを入力し、それぞれの 3 時間後の風速を予測するように訓練した。その学習と評価には、1.2 節で述べるように気象庁が提供している日本近辺の風速と気圧データを用いた。評価指標については 1.4 節にて詳述している。

1.2 利用したデータ及び学習条件

1.2.1 利用したデータ

気象庁が提供している日本近辺の風速と気圧データを京都大学の生存圏データベース [1] から入手した。このデータセットは図 1.1 に示す通り、日本近辺の北緯 22.4° から 47.6° まで、東経 120° から 150° までの領域をそれぞれ $0.05^{\circ} \times 0.0625^{\circ}$ の細かさで等緯度等経度に区切り、各格子点上における風速と気圧を記録したものである [2]。すなわち、ある一時刻のデータは緯度経度について 505×480 の行列となっており、ある一点には風速の南北方向成分 [m/s]、風速の東西方向成分 [m/s]、気圧 [hPa] の 3 つの値が記録されている。

提供されているデータは 3 時間間隔であり、(前日の)21:00, 0:00, 3:00, 6:00 の 4 時刻分をひとまとめにして時系列データとした。これを 2011 年 1 月 1 日から 2020 年 12 月 31 日までの 3650 日分用意し (統計を取る際の簡便化のために閏日は除いてある)、1.2.2 項で述べる処理をする前の全体のデータセットとした。

1.2.2 データの前処理

1.2.1 項で述べたデータセットをそのまま入出力に用いるのではなく、前処理を実施した。その処理の詳細を以下に示す。

図 1.2 に示すように、 505×480 の行列を更に 10×10 の格子によって区切り、この格子内で風速と気圧を平均化することで 50×48 のサイズまで落とした。なお、端数分は切り捨てている。これは、提案モデルの格子の大きさを適切に設定することで、3 時間で変化する風速の空間的な変化を捉えることができると考えたためである。このデータセットを改めて全体のデータセットとした。

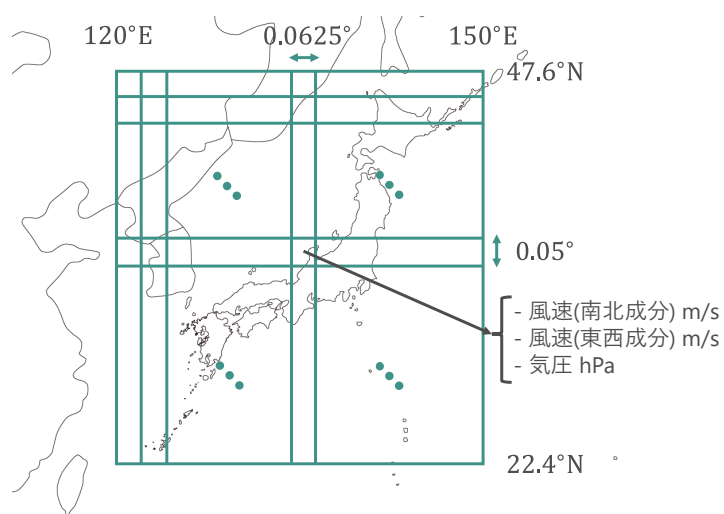


図 1.1 気象庁が提供している日本近辺の風速と気圧データの領域

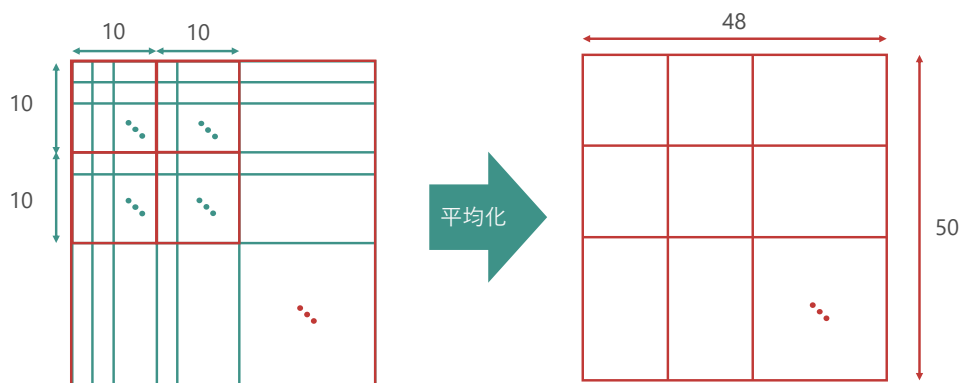


図 1.2 風速と気圧の平均化

1.2.3 学習条件

続いて、モデルの詳細な学習条件について述べる。提案モデルのアーキテクチャについての概要を??項で述べたがここではより具体的に数値を提示し、モデルの仕様を詳細化する。図 1.3 に示したように、このモデルでは並進と衝突をそれぞれ 5 回行った (緑の実線矢印が並進を表し、緑の破線矢印が衝突を表している)。すなわち式 (??) において $\Delta T = 5$ である。??項で述べた通り衝突の回数分だけ外枠が削られるため、図中の赤破線によって強調されているように出力層の大きさは入力層に比べて東西南北の端 5 マス分減少している。(前日の)21:00, 0:00, 3:00 の 3 時刻分のデータを入力として 0:00, 3:00, 6:00 の風速を予測させた。すなわち、式 (??) において $n = 4$ 、式 (??) において無次元時刻 ΔT は 3[h] 相当である。

提案モデルの学習には Adam[3] を用いた。学習率は 10^{-3} とし、ミニバッチサイズは 16 とした。全体のデータセットを 2920 日分と 730 日分に分けそれぞれを学習用データと検証用データとした。学習は 500 エポック行い、かかった時間は約 8 時間であった。実行環境には Google Colaboratory[4] を用い、GPU は Tesla T4 を用いた。また提案モデルの構築には PyTorch[5] を用い、自動微分による学習を行った。

1.3 物理的な構造を含まない深層学習モデル

提案モデルと比較するために用いた物理的な構造をもたないエンコーダ・デコーダモデルを図 1.4 に示す。このモデルは??項で述べた Chen らのモデルと U-Net[6] を参考にしたものである。図中の各層の左に大きさを、上部にチャンネル数を記載してある。入出力部分では、風速の東西方向成分と南北方向成分、そして気圧を正規化して 3 チャンネルにまとめた。エンコーダ・デコーダの各層には畳み込み層とバッチ正規化 [7]、活性化関数である tanh を用いた。また、エンコーダの各層の出力をコピーしてデコーダの対応する層の入力に結合するスキップコネクションを用いた。エンコーダから出力される信号を平滑化してから LSTM ユニットに入力し、その出力をデコーダ

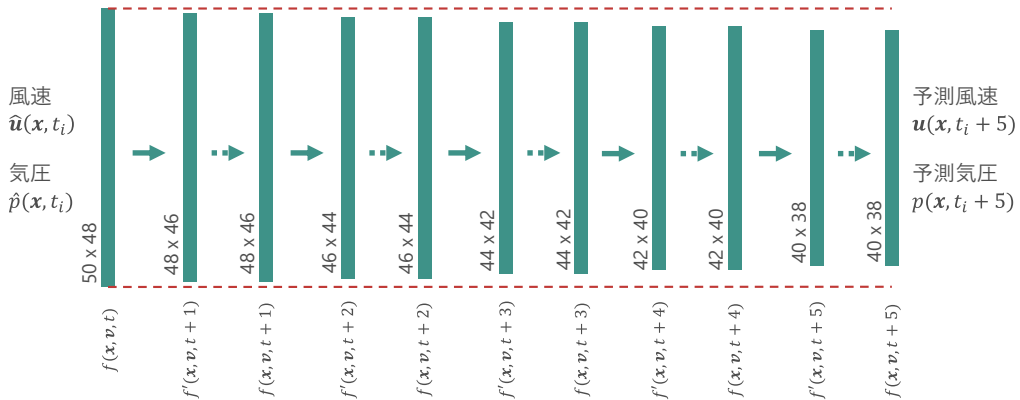


図 1.3 モデルのアーキテクチャ

に入力した。デコードの際にはサイズを復元するために逆畳み込み層を用いた。出力の際に、提案モデルと性能比較をできるように周囲 5 マス分を切り捨てた。

その他の利用したデータ及び学習条件は 1.2 節と同様である。

1.4 評価指標

モデルが出力した最終時刻の予測値と実測値を用いて、以下の評価指標を算出した。ここで、指標に用いられる「風速」はベクトル量ではなく、スカラー量であることに注意されたい。これらの評価指標は一般的に気象予報モデルの統計的検証に用いられるものを参考にした [8]。また、RMSE 改善率とは、提案モデルの RMSE がエンコーダ・デコーダモデルに比べてどれだけ小さいかを表す指標であり以下のように定義される。

$$(\text{RMSE 改善率 } [\%]) = \frac{\text{RMSE}_{\text{prev}} - \text{RMSE}_{\text{prop}}}{\text{RMSE}_{\text{prev}}} \times 100 \quad (1.1)$$

ただし、 $\text{RMSE}_{\text{prev}}$ はエンコーダ・デコーダモデルの RMSE, $\text{RMSE}_{\text{prop}}$ は提案モデルの RMSE を表す。

1.4.1 全体的な誤差指標

全体的な誤差指標として、以下の 3 つを用いた。

1. 全体の風速の RMSE[m/s](Root Mean Squared Error, 二乗平均平方根誤差) と提案モデルの RMSE 改善率 [%]
2. 全体の風速の ME[m/s](Mean Error, 平均誤差)
3. 全体の風向の RMSE[°] と提案モデルの RMSE 改善率 [%]

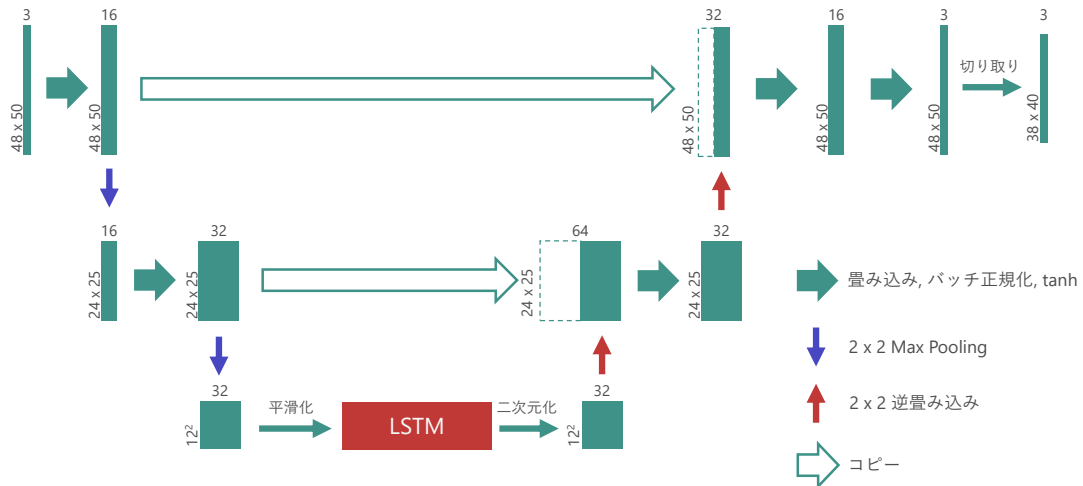


図 1.4 物理的な構造を含まない深層学習モデルのアーキテクチャ

以下にそれぞれの評価指標の定義を示す.

$$(\text{全体の風速の RMSE[m/s]}) = \sqrt{\frac{1}{NM} \sum_{\mathbf{x}, i} (|\mathbf{w}(\mathbf{x}, t_i)| - |\hat{\mathbf{w}}(\mathbf{x}, t_i)|)^2} \quad (1.2)$$

$$(\text{全体の風速の ME[m/s]}) = \frac{1}{NM} \sum_{\mathbf{x}, i} |\mathbf{w}(\mathbf{x}, t_i)| - |\hat{\mathbf{w}}(\mathbf{x}, t_i)| \quad (1.3)$$

$$(\text{全体の風向の RMSE[°]}) = \sqrt{\frac{1}{NM} \sum_{\mathbf{x}, i} \arg(\mathbf{w}(\mathbf{x}, t_i), \hat{\mathbf{w}}(\mathbf{x}, t_i))^2} \quad (1.4)$$

ただし, M は格子点 \mathbf{x} の総数, N はデータセットの総数とし, そのうちの i 番目のデータセットについて $\mathbf{w}(\mathbf{x}, t_i)$ をモデルによる予測風速ベクトル, $\hat{\mathbf{w}}(\mathbf{x}, t_i)$ を実測の風速ベクトルとした. ここで, $\mathbf{w}(\mathbf{x}, t_i)$ と $\hat{\mathbf{w}}(\mathbf{x}, t_i)$ は共に速度の次元を持つ. また, \arg は二つのベクトルが成す角度を返す関数である. 以降の評価指標においても断りがない限り同様である.

1.4.2 座標ごとの誤差指標

座標ごとの誤差指標として, 以下の 3 つを用いた.

1. 座標ごとの風速の RMSE[m/s] と提案モデルの RMSE 改善率 [%]
2. 座標ごとの風速の ME[m/s]
3. 座標ごとの風向の RMSE[°] と提案モデルの RMSE 改善率 [%]

以下にそれぞれの評価指標の定義を示す.

$$(\text{座標ごとの風速の RMSE[m/s]}) = \sqrt{\frac{1}{N} \sum_i (|\mathbf{w}(\mathbf{x}, t_i)| - |\hat{\mathbf{w}}(\mathbf{x}, t_i)|)^2} \quad (1.5)$$

$$(\text{座標ごとの風速の ME[m/s]}) = \frac{1}{N} \sum_i |\mathbf{w}(\mathbf{x}, t_i)| - |\hat{\mathbf{w}}(\mathbf{x}, t_i)| \quad (1.6)$$

$$(\text{座標ごとの風向の RMSE[°]}) = \sqrt{\frac{1}{N} \sum_i \arg(\mathbf{w}(\mathbf{x}, t_i), \hat{\mathbf{w}}(\mathbf{x}, t_i))^2} \quad (1.7)$$

1.4.3 ある時刻における座標ごとの誤差指標

ある時刻における座標ごとの誤差指標として, 以下の 3 つを用いた.

1. ある時刻における座標ごとの風速の誤差 [m/s]
2. ある時刻における座標ごとの風向の誤差 [°]

以下にそれぞれの評価指標の定義を示す.

$$(\text{ある時刻 } t_0 \text{ における座標ごとの風速の誤差 [m/s]}) = |\mathbf{w}(\mathbf{x}, t_0)| - |\hat{\mathbf{w}}(\mathbf{x}, t_0)| \quad (1.8)$$

$$(\text{ある時刻 } t_0 \text{ における座標ごとの風向の誤差 } [^\circ]) = \arg(\mathbf{w}(\mathbf{x}, t_0), \hat{\mathbf{w}}(\mathbf{x}, t_0)) \quad (1.9)$$

1.4.4 日本陸上で最も風速変化が大きい点における誤差指標

日本陸上において風速が大きく変化する際にそれを予測できるかを評価するため、日本陸上で最も風速変化が大きい点における誤差指標として、以下の4つを用いた。

1. 日本陸上で最も風速変化が大きい点における風速の RMSE[m/s]
2. 日本陸上で最も風速変化が大きい点における風向の RMSE[$^\circ$]
3. 日本陸上で最も風速変化が大きい点における実測風速の変化量 [m/s]
4. 日本陸上で最も風速変化が大きい点における実測風向の変化量 [$^\circ$]

以下にそれぞれの評価指標の定義を示す。

$$\begin{aligned} & (\text{日本陸上で最も風速変化が大きい点における風速の RMSE[m/s]}) \\ &= \sqrt{\frac{1}{N} \sum_i (|\mathbf{w}(\mathbf{x}_{\max}, t_i)| - |\hat{\mathbf{w}}(\mathbf{x}_{\max}, t_i)|)^2} \end{aligned} \quad (1.10)$$

$$\begin{aligned} & (\text{日本陸上で最も風速変化が大きい点における風向の RMSE[}^\circ\text{]}) \\ &= \sqrt{\frac{1}{N} \sum_i \arg(\mathbf{w}(\mathbf{x}_{\max}, t_i), \hat{\mathbf{w}}(\mathbf{x}_{\max}, t_i))^2} \end{aligned} \quad (1.11)$$

$$\begin{aligned} & (\text{日本陸上で最も風速変化が大きい点における実測風速の変化量 [m/s]}) \\ &= \sqrt{\frac{1}{N} \sum_i |\hat{\mathbf{w}}(\mathbf{x}_{\max}, t_i - 3[\text{h}])| - |\hat{\mathbf{w}}(\mathbf{x}_{\max}, t_i)|} \end{aligned} \quad (1.12)$$

$$\begin{aligned} & (\text{日本陸上で最も風速変化が大きい点における実測風向の変化量 } [^\circ]\text{)}) \\ &= \sqrt{\frac{1}{N} \sum_i \arg(\hat{\mathbf{w}}(\mathbf{x}_{\max}, t_i - 3[\text{h}]), \hat{\mathbf{w}}(\mathbf{x}_{\max}, t_i))} \end{aligned} \quad (1.13)$$

ただし、 \mathbf{x}_{\max} は日本陸上で最も風速変化が大きい点であり、これは

$$\mathbf{x}_{\max} = \underset{\mathbf{x}}{\operatorname{argmax}} (|\hat{\mathbf{w}}(\mathbf{x}, t_i - 3[\text{h}])| - |\hat{\mathbf{w}}(\mathbf{x}, t_i)|) \quad (1.14)$$

と表され、 t_i に依存することに注意されたい。

1.5 実験結果

1.1 節で述べた評価指標を用いて、モデルの性能を評価した。その結果を順に述べる。

1.5.1 全体的な誤差

提案モデルとエンコーダ・デコーダモデルについて全体の風速の RMSE[m/s] と ME[m/s]、そして風向の RMSE[°] の結果を、それぞれ表 1.1, 1.2, 1.3 に示す。

1.5.2 座標ごとの誤差

提案モデルとエンコーダ・デコーダモデルについて座標ごとの風速の RMSE[m/s] と ME[m/s]、そして風向の RMSE[°] の結果を、それぞれ図 1.5, 1.6, 1.7 に示す。ただし、ME については考察のために重み学習をしないただの格子ボルツマン法をもちいた場合の結果も併せて示す。

1.5.3 ある時刻における座標ごとの誤差

特徴的な気象条件の下での風速と風向の予測精度を評価するために、台風の日 (2020/09/06 6:00) における座標ごとの風速の誤差と風向の誤差の結果を、提案モデルとエンコーダ・デコーダモデルについてそれぞれ図 1.8, 1.9 に示す。

1.5.4 日本陸上で最も風速変化が大きい点における誤差

提案モデルとエンコーダ・デコーダモデルについて日本陸上で最も風速変化が大きい点における風速の RMSE[m/s] と実測風速の変化量 [m/s] を表 1.4 に示す。また、風向については表 1.5 に示す。

1.6 考察

表 1.1 と表 1.3 より、RMSE 改善率 [%] が 0 より真に大きいことから、提案モデルの方がエンコーダ・デコーダモデルよりも風速と風向の予測精度が高いことがわかった。これにより、提案モデルの大局的な優位性を示すことができた。以下、提案モデルとエンコーダ・デコーダモデルの予測精度の共通点と相違点から、日本近辺の大気の性質及び二つのモデルの性質について考察する。

図 1.5(a),(b) をみると、2 つのモデルは共に陸上よりも海上の風速誤差の方が大きいことがわかった。これに対して、図 1.7(a),(b) をみると、2 つのモデルは共に海上よりも陸上の風向誤差の方が大きいことがわかった。これは、海上は遮蔽物が存在しないため風速が陸上よりも大きい傾向にありその分変化量も大きいこと、そして陸上の風向は複雑な地形によって海上よりも変化量が大きくなることが原因として考えられる。

表 1.2 及び図 1.6(a) から、提案モデルは日本の沿岸部において風速を小さく見積もる傾向があることがわかった。これは、図 1.6(c) から、重み学習をしない格子ボルツマン法を用いた場合も同様であることから、提案モデルが時間発展とともに風速を強めるような機構が不足していることが原因として考えられる。これに対して、図 1.6(a) と図 1.6(c) を見比べると日本の陸上部においては重み学習をしない格子ボルツマン法を用いた場合と比べて風速を正しく、小さく見積もることができていることから、提案モデルが時間発展とともに風速を弱める機構を持っていることがわかる。

図 1.5(c) と図 1.7(c) をみると、提案モデルはエンコーダ・デコーダモデルに比べて陸上では精度が改善し陸と海の境界付近で精度が悪化することがわかった。これは上記に述べた通り提案モデルが陸上で風速を小さく見積もる傾向があることが原因として考えられる。

図 1.8, 1.9 をみると、台風の日 (2020/09/06 6:00) において、提案モデルはエンコーダ・デコーダモデルに比べて特に台風の中心部において風速の誤差が小さくなっていることがわかった。これは、データセットに台風という特異な気象条件が多く含まれておらずエンコーダ・デコーダモデルが学習しきれなかったのに対して提案モデルは物理的な構造がその情報を補ったからだと推察される。

最後に、表 1.4 及び表 1.5 をみると、提案モデルはエンコーダ・デコーダモデルに比べて日本陸上で最も風速変化が大きい点において風速と風向の誤差が小さくなっていることがわかった。実際の短期風速予報の運用においても、風速変化が大きい地点で風速と風向の予測精度が高いことは重要であるため、提案モデルの有用性が示されたと言える。

表 1.1 全体の風速の RMSE

提案モデル [m/s]	エンコーダ・デコーダモデル [m/s]	RMSE 改善率 [%]
1.0432	1.0930	4.57

表 1.2 全体の風速の ME

提案モデル [m/s]	エンコーダ・デコーダモデル [m/s]
-0.1909	0.2001

表 1.3 全体の風向の RMSE

提案モデル [°]	エンコーダ・デコーダモデル [°]	RMSE 改善率 [%]
21.444	22.521	4.78

表 1.4 日本陸上で最も風速変化が大きい点における風速の RMSE

提案モデル [m/s]	エンコーダ・デコーダモデル [m/s]	実測風速の変化量 [m/s]
1.7926	1.8399	3.2559

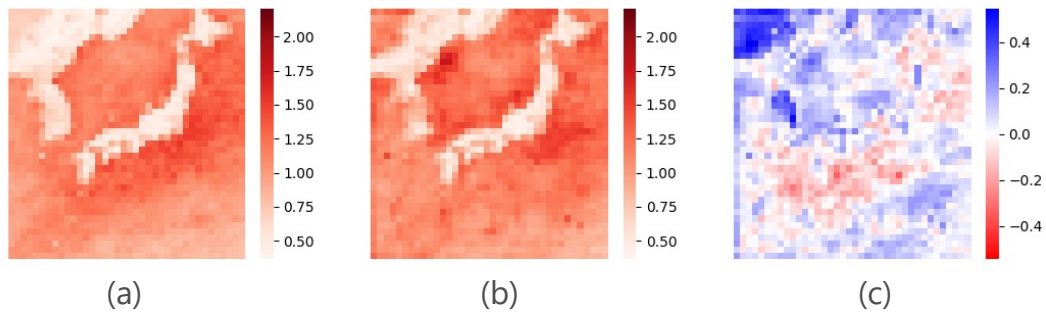


図 1.5 座標ごとの風速の RMSE (a) 提案モデルの RMSE[m/s] (b) エンコーダ・デコーダモデルの RMSE[m/s] (c) 提案モデルによる RMSE 改善率 [%]

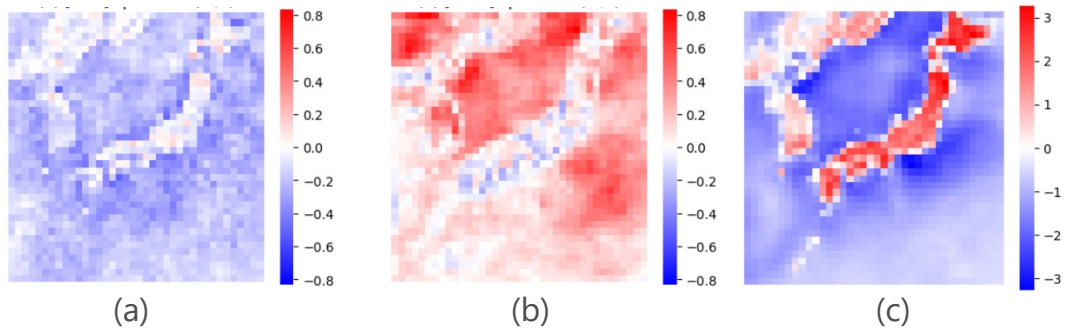


図 1.6 座標ごとの風速の ME (a) 提案モデルの ME[m/s] (b) エンコーダ・デコーダモデルの ME[m/s] (c) 重み学習をしない格子ボルツマン法を用いた場合の ME[m/s]

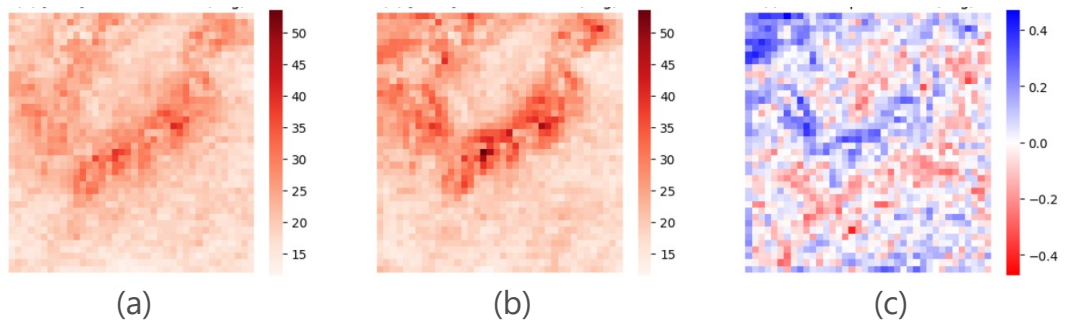


図 1.7 座標ごとの風向の RMSE (a) 提案モデルの RMSE[°] (b) エンコーダ・デコーダモデルの RMSE[°] (c) 提案モデルによる RMSE 改善率 [%]

表 1.5 日本陸上で最も風速変化が大きい点における風向の RMSE

提案モデル [°]	エンコーダ・デコーダモデル [°]	実測風速の変化量 [°]
25.805	30.240	42.192

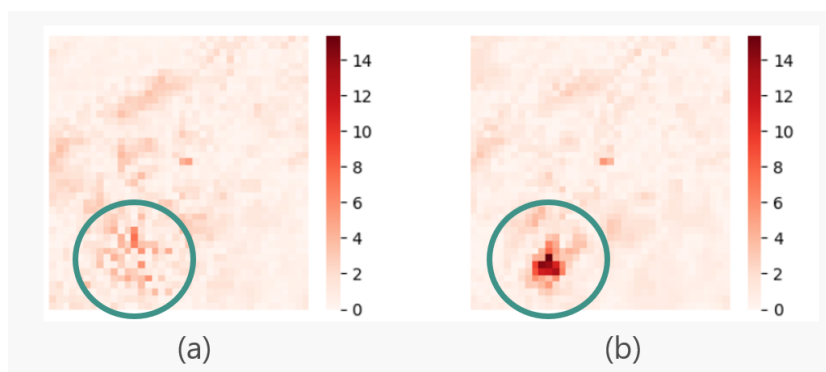


図 1.8 台風の日 (2020/09/06 6:00) における座標ごとの風速の誤差 (緑の実線で囲まれた位置に台風が存在する) (a) 提案モデルの誤差 [m/s] (b) エンコーダ・デコーダモデルの誤差 [m/s]

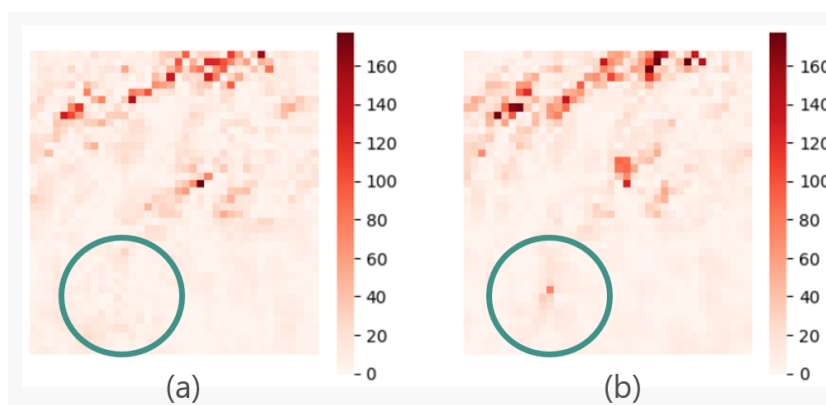


図 1.9 台風の日 (2020/09/06 6:00) における座標ごとの風向の誤差 (緑の実線で囲まれた位置に台風が存在する) (a) 提案モデルの誤差 [°] (b) エンコーダ・デコーダモデルの誤差 [°]

参考文献

- [1] 京大大学生存圏研究所. 生存圏データベース., 2004. <http://database.rish.kyoto-u.ac.jp/>, (参照 2024-01-16).
- [2] 一般財団法人気象業務支援センター. メソ数値予報モデル gpv (msm) , 2022. <http://www.jmbse.or.jp/jp/online/file/f-online10200.html>, (参照 2024-01-16).
- [3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.
- [4] Google Research. Colaboratory, 2023. <https://research.google.com/colaboratory/faq.html>, (参照 2022-03-24).
- [5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.
- [6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, 2015.
- [7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, pp. 448–456. JMLR.org, 2015.
- [8] 気象庁. 付録 d 数値予報研修テキストで用いた表記と統計的検証に用いる代表的な指標, 2018. <https://www.jma.go.jp/jma/kishou/books/nwptext/52/Appendix.D.pdf>, (参照 2024-01-16).