機械学習の基礎

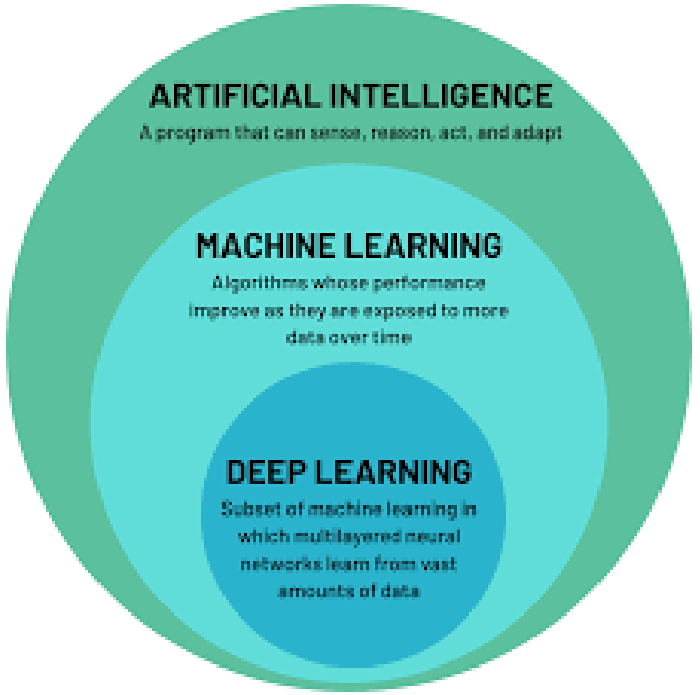
ニューラルネットワーク

M1 李 康民

機械学習とは

→ 観測データから規則性を抽出し、未知の入力に対する予測や意思決定を可能にする方法論

① 観測標本を生み出した確率分布を，**どのようなモデル族によって近似**するか
② その近似を**どのような最適化問題として解く**か



教師あり学習

・訓練データセット→ $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$

$x_i \in X$：第 i 標本の入力, $y_i \in Y$ ：対応する出力

・パラメータθを持つモデル→ $f_\theta(x)$：入力 x に対する予測値

θはモデルの自由度を定める未知量、**学習とはθをデータから推定すること**

損失関数

・どれほど間違っているか　→ **損失関数**(loss function)

・損失関数を $\ell(y_i, f_\theta(x_i))$と書くと、経験リスクは

$$\hat{R}_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i))$$

→ ℓ は予測 $f_\theta(x_i)$ が正解 y_i からどれだけ離れているかを測る関数
→ $\hat{R}_n(\theta)$ は訓練データ上の平均誤差

・**経験リスクの最小化**が目標

損失関数と最適化

回帰

・出力が平均、分散のガウス分布に従うと仮定

・平均二乗誤差 (MSE) :

$$\mathcal{L}_{MSE}(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu_\theta(x_i))^2$$

分類

・softmax により、クラスに属する確率を計算　$\pi_{\theta,k}(x_i)$：クラス k の条件付き確率と解釈

・クロスエントロピー損失 (cross-entropy loss) :

$$\mathcal{L}_{CE}(\theta) = -\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \log \pi_{\theta,k}(x_i)$$

→ **損失関数は問題の種類によって異なり、**誤差を最小化するように最適化を行う

確率的勾配降下法

・深層学習　→ パラメータ数、データ数大きい　→ 計算量、メモリ使用量の限界

→ **確率的勾配降下法**(SGD)：
　　制約のもとで勾配情報を効率よく利用するための基本アルゴリズム

目的関数

平均損失	$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\theta)$
標本 i に対する個別損失	$\ell_i(\theta) = \ell(y_i, f_\theta(x_i))$

→ 各反復 t における更新　$\theta_{t+1} = \theta_t - \eta_t \nabla L(\theta_t) = \theta_t - \eta_t \frac{1}{n}\sum_{i=1}^{n} \nabla \ell_i(\theta_t)$

→ SGDでは、**一部のデータだけを利用**して学習　$g_t = \frac{1}{b}\sum_{i \in B_t} \nabla \ell_i(\theta_t), \theta_{t+1} = \theta_t - \eta_t g_t$
・効率よく計算できる！

g_t：ミニバッチ勾配、b：バッチサイズ

誤差逆伝播法

・多数のパラメータそれぞれについて損失の勾配を求める必要

→ **誤差逆伝播法**(backpropagation)
メリット：複合関数モデルの勾配を、一度の順伝播と一度の逆伝播で求められる
→ 計算量はグラフの辺数にほぼ線形に比例

・ネットワークの各層では、

→ $a^{(l)} = W^{(l)}h^{(l-1)} + b^{(l)}$　　$W^{(l)}$：第 l 層の重み行列　$b^{(l)}$：バイアスベクトル
$h^{(l)} = \phi^{(l)}(a^{(l)})$　　$a^{(l)}$：線形結合　$h^{(l)}$：層出力　$\phi^{(l)}$：活性化関数

のように入力にWとbを適用し、その後に活性化関数を通して次の層へ出力を渡す

誤差逆伝播法

・各層の前活性化$a^{(l)}$に関する損失の勾配　　$\delta^{(l)} = \dfrac{\partial \mathcal{L}}{\partial a^{(l)}}$

　　→ その層が誤差にどれだけ影響したか

・**出力層**では　　$\delta^{(L)} = \dfrac{\partial \mathcal{L}}{\partial h^{(L)}} \odot \phi^{(L)'}(a^{(L)})$　→ **誤差を直接計算**可能

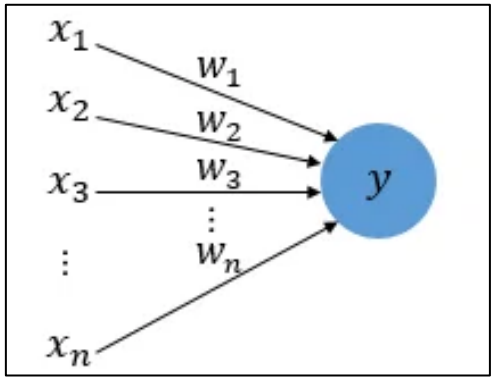
・**中間層**は正解を直接見れない　→ **誤差を後ろから前え伝える**必要：**逆伝播**

$$\delta^{(l)} = ((W^{(l+1)})^{\mathsf{T}}\delta^{(l+1)}) \odot \phi^{(L)'}(a^{(L)}), \qquad l = L-1, \ldots, 1$$

・各パラメータの勾配を求めることができるようになる

$$\frac{\partial \mathcal{L}}{\partial W^{(l)}} = \delta^{(l)}(h^{(l-1)})^{\mathsf{T}}, \qquad \frac{\partial \mathcal{L}}{\partial b^{(l)}} = \delta^{(l)}$$

ニューラルネットワーク

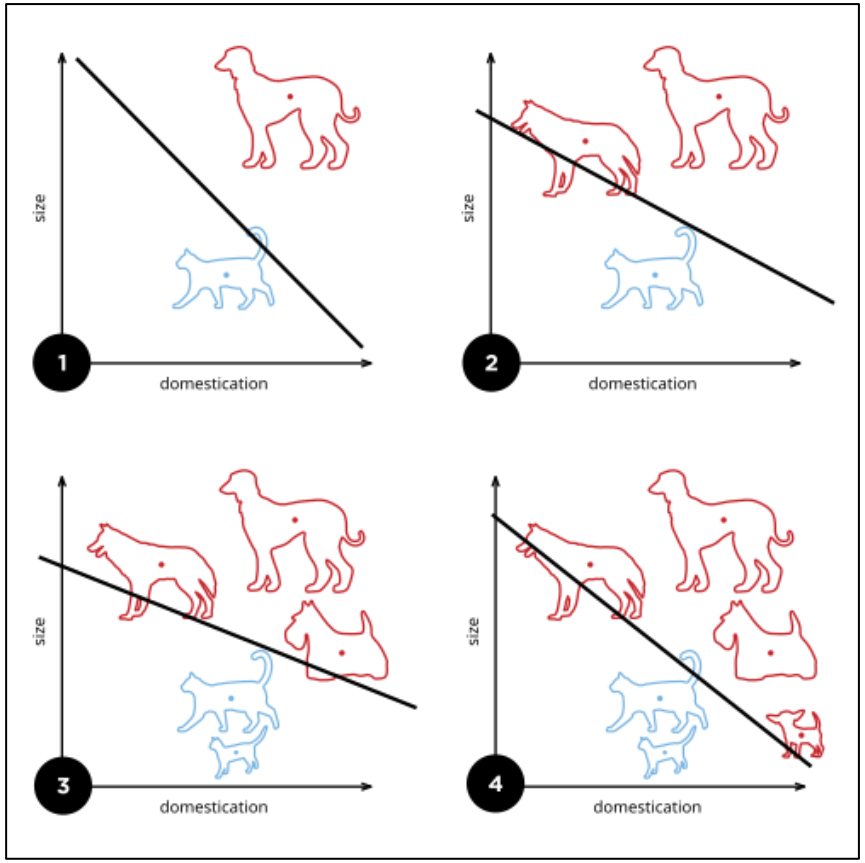


・**パーセプトロン**(perceptron)：最も基本的な**線形分類**モデル

→ 多数のInputから一つのOutputを出力

決定関数　$\hat{y} = sign(w^{\mathsf{T}}x + b)$

→ **超平面による二値分類**を表す
→ 入力空間における複雑な**非線形境界を直接表すことはできない**

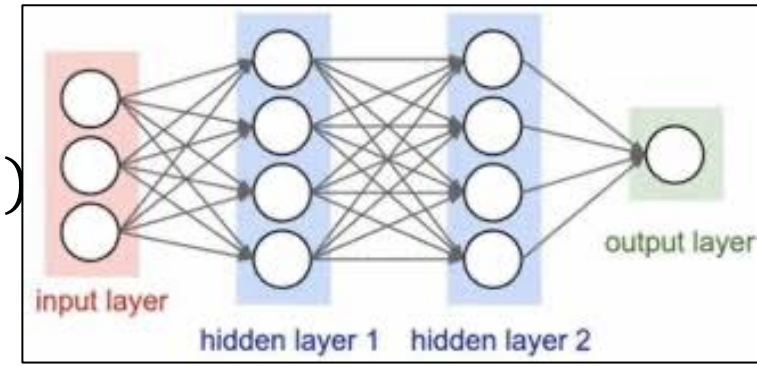


→**多層パーセプトロン**（MLP）：中間表現を導入

ニューラルネットワーク

・ **多層パーセプトロン**(MLP)　$h^{(l)} = \phi^{(l)}(W^{(l)}h^{(l-1)} + b^{(l)})$

→ まず前の層から入力を受け取り、 線形変換を行う



しかし、活性化関数が存在しなければ、　$h^{(L)} = W_{eff}x + b_{eff}$

→ 何層重ねても**一層の線形モデルになってしまう**

・ **活性化関数**

→ ネットワークに非線形性を与える役割

・sigmoid関数　$\sigma(u) = \dfrac{1}{1 + \exp(-u)}$　　出力を 0～1 の範囲に変換　→ 確率

・ReLU関数　　$\mathrm{ReLU}(u) = \max(0, u)$　　正の領域で勾配が保たれる

深いネットワークでも勾配が消失しにくい

CNN, RNN

・ **CNN** (Convolutional Neural Network)

ニューラルネットワーク(MLP)では、入力を単純なベクトルとして扱っていたが、
画像データでは、
“近くの画素同士が強く関係している、同じ特徴が位置を変えて現れる”
という**空間的構造**を持っている

「画像の局所的特徴を効率的に抽出するためのニューラルネットワーク」

・局所結合(local receptive field)
・重み共有(weight sharing)

CNN, RNN

・畳み込み：小さな領域に対して同じ filter を適用すること

$$(X * K)_{i,j} = \sum_{u=0}^{r-1} \sum_{v=0}^{s-1} K_{u,v} X_{i+u,j+v}$$

X:入力画像, K：filter（カーネル）, r, s ：カーネルの高さと幅, u, v ：カーネル内部の添字, i, j ：出力位置

→ ① パラメータ数が大幅に削減 、② **平行移動等変性** が得られる

→ 入力をずらせば出力も同じだけずれる
→ 同じ局所パターンを画像中のどこでも検出OK

時系列，文章，音声のようなデータでは，過去の情報をどのように状態として保持し，将来の予測に反映させるか　→ **RNN**

CNN, RNN

・ **RNN** (Recurrent Neural Network)

現在の観測や予測が過去の文脈に依存するため，入力を単に固定長ベクトルとして扱うだけでは不十分

「現在の出力は過去の入力や過去の出力に依存」→ **有限次元の状態で近似**する

$$h_t = \phi(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

可変長系列を固定次元の状態遷移として扱う動的システム

PINN

・ **PNN** (Physics-Informed Neural Network)

現実の工学問題では、単にデータへ適合するだけでは不十分な場合が多い
→ 「**物理法則を満たす必要**」

→ 物理法則を損失関数に組み込んだニューラルネットワーク

$Loss = Loss_{data} + \lambda Loss_{physics}$ (データ誤差＋重み係数＊物理法則からの誤差)

深層学習による交通流予測モデルの開発

研究目的

・将来の交通状態を予測　→ ① NN（MLP）の性能を最大化
② 移動平均、LWR方程式、NNの性能比較
（RMSE, MAEの評価）

・Q:NNの性能を最大化するためには？NNが得意/不得意なケースは？

・使用したデータ：東名高速での実験結果 (2024 NEXCO中日本)

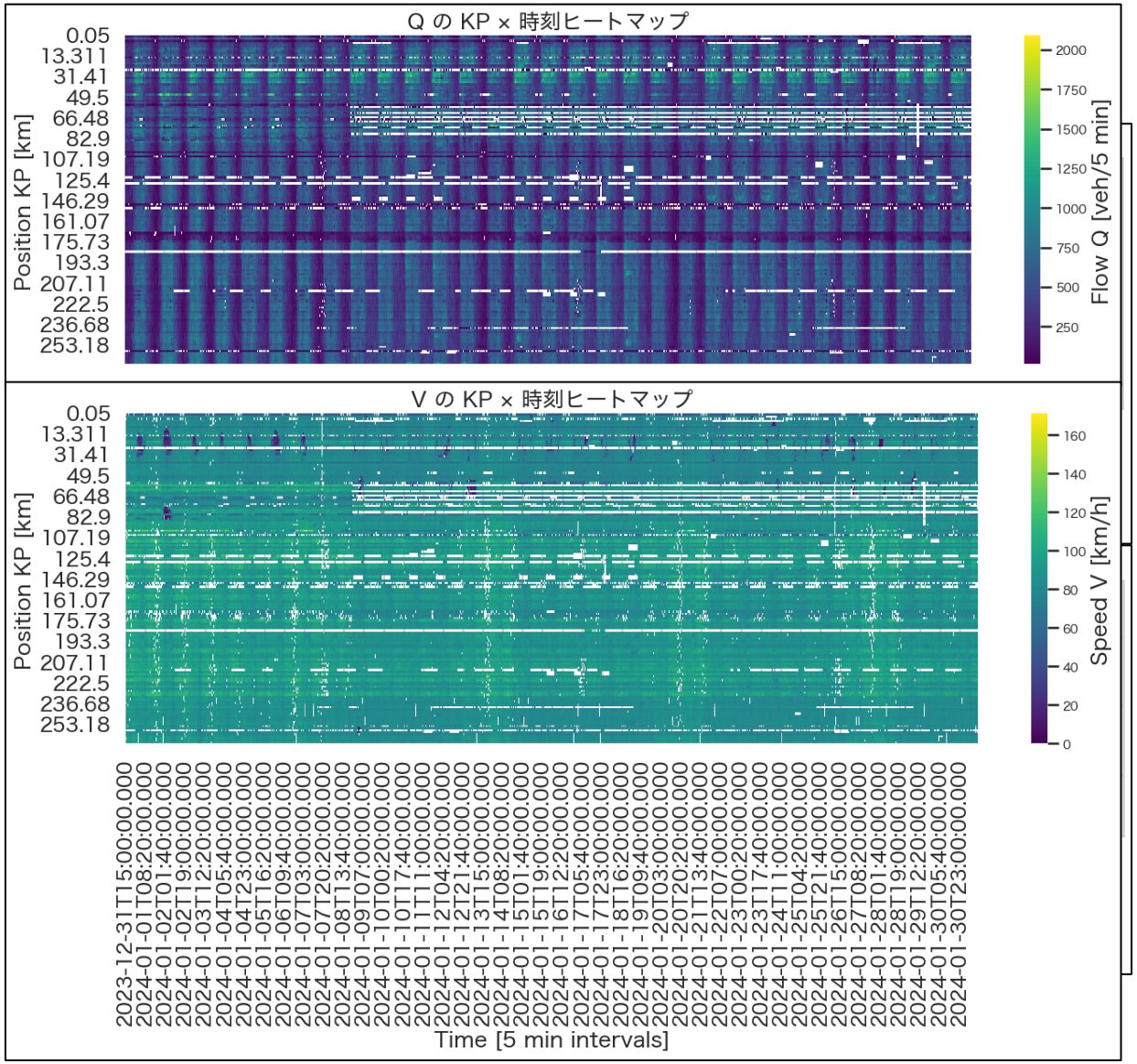
shape = (19855872, 10)
timestamp range = 2024-01-01 00:00:00+09:00 -> 2024-01-31 23:55:00+09:00

データの集計

・交通量Qが一定の周期で繰り返されている

→夜間には交通量が低下,
昼間に再び増加するパターン

・速度Vは多くの時間帯で50～60 km/h程度を維持



データの集計

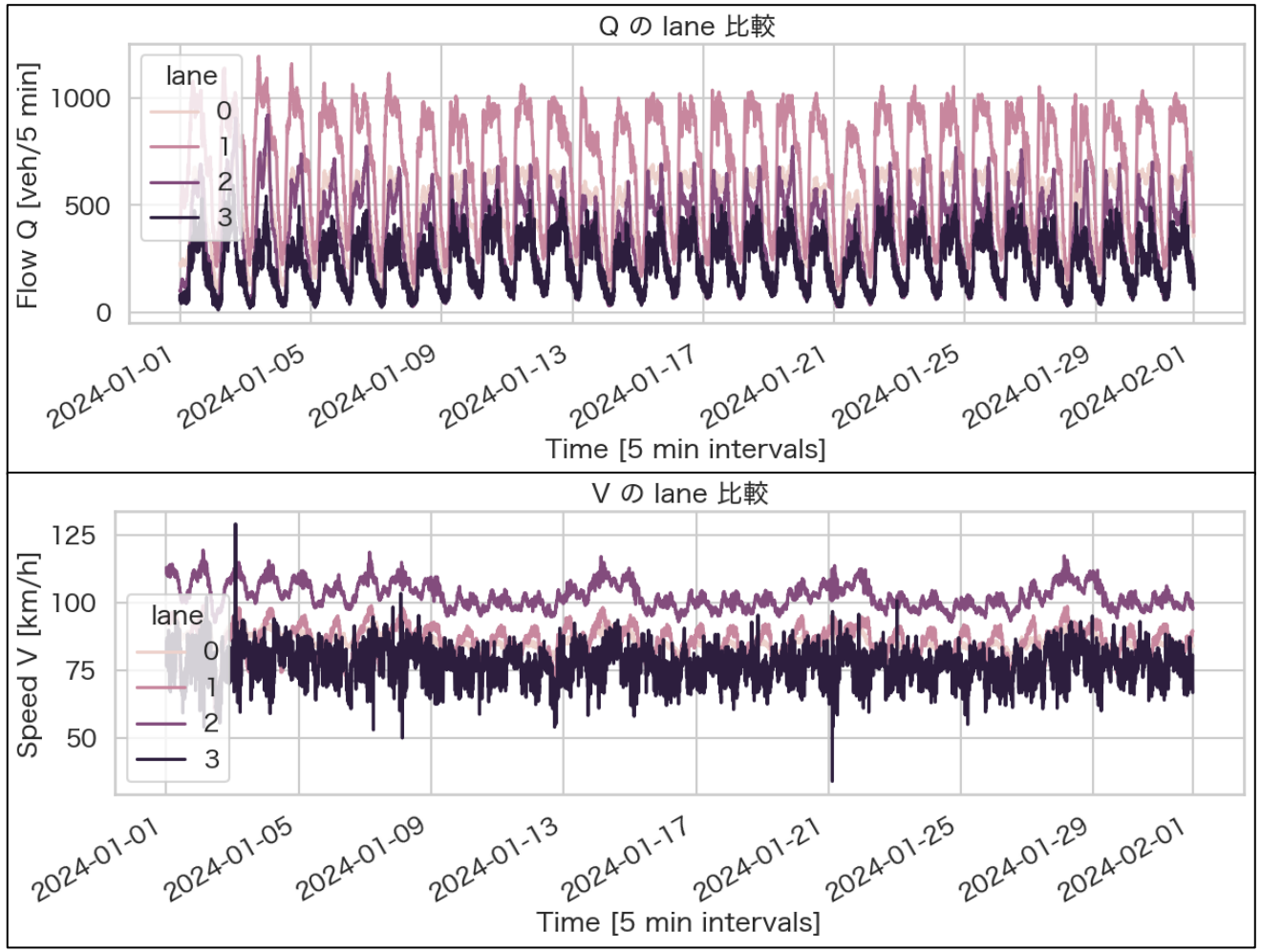
・全laneで日周期的な変動が確認

→ しかし，laneごとに交通量の絶対値が
大きく異なっている

Lane1：非常に大きなピークが発生
Lane3：全体的に交通量が低く，変動幅も小

・laneごとに速度差あり

Lane2：全体的に100 km/h前後の高い速度
Lane3：速度変動が大きく，急激な低下も確認

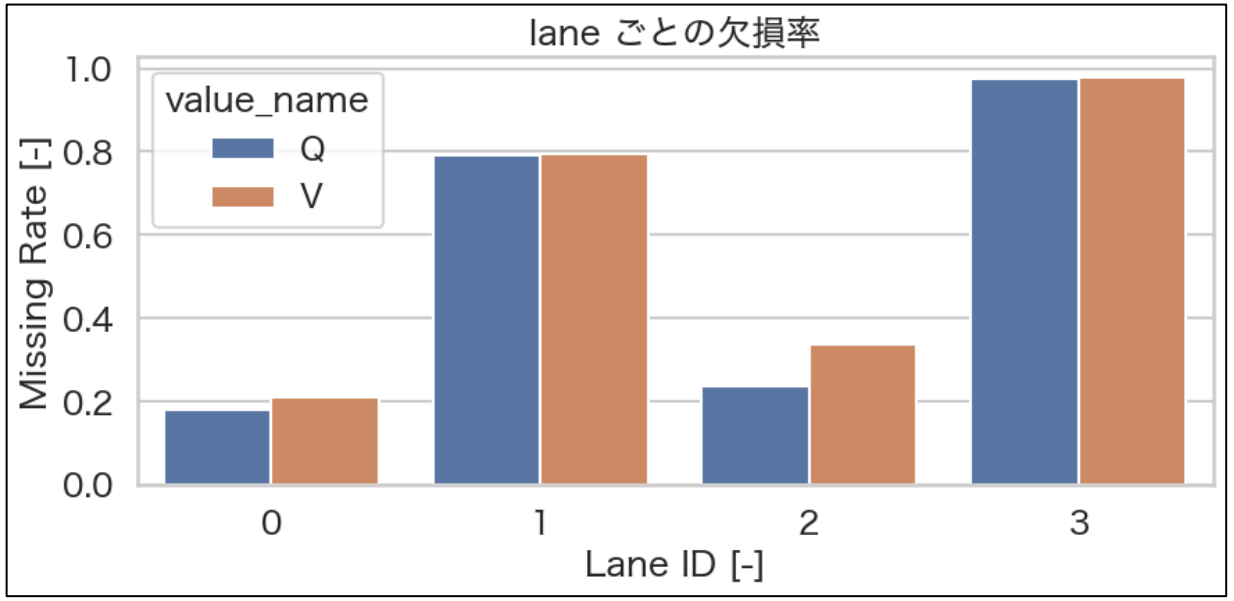


データの集計

・**欠損率の確認**

Lane1：約80%，lane3：ほぼ100%に近い欠損率

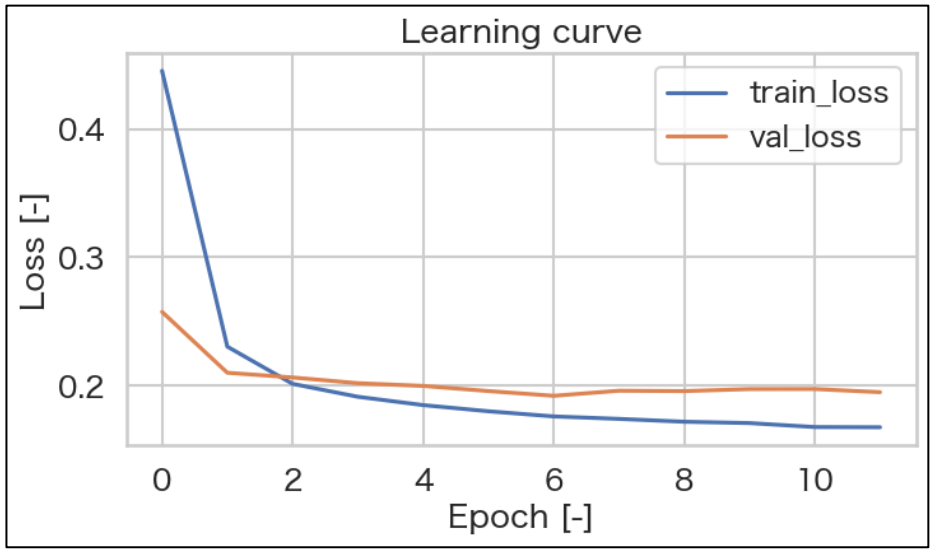
lane3は，ほとんどデータが存在していない状態？
→ そのまま学習に利用するとノイズや
　学習不安定性の原因になる可能性が高い



・**予測タスクの定義**

Lookback= 12, horizon= 12 → 過去60分で未来60分予測
Train = 0.6 / validation = 0.2 / test = 0.2
Epoch = 12回

NN（MLP）による予測



・**Learning Curve**

Epoch 0～2付近で急激にlossが低下，MLPが交通
データの基本的な周期パターンを早く学習できている

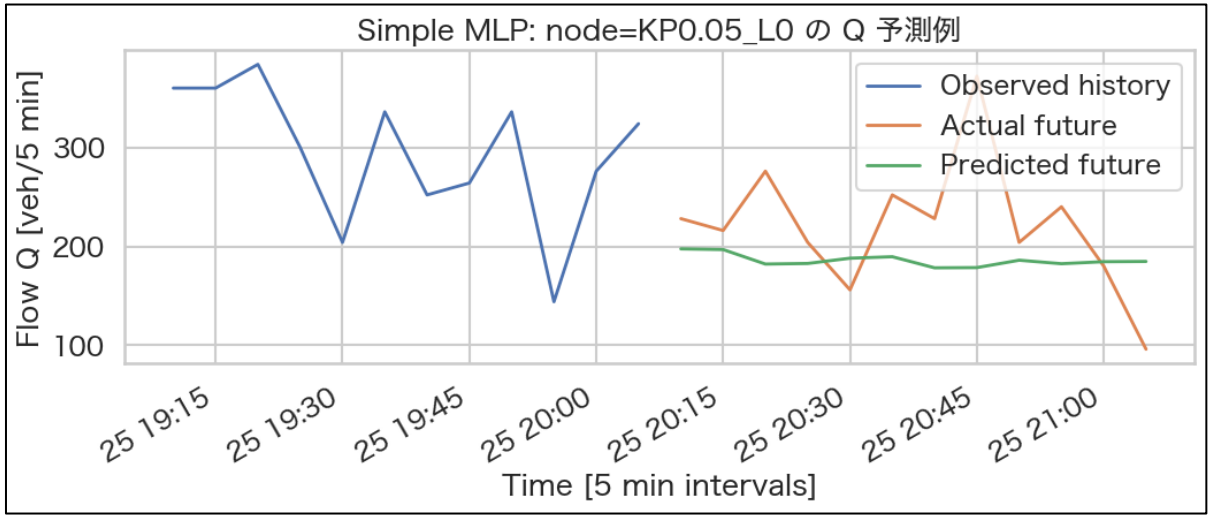
→ 突発的変動や複雑な非線形パターンの学習は難しい？

validation lossが大きく増加しているわけではないため，**安定して学習**できている状態！

・**MLPの予測結果**

予測値（Predicted future）は比較的平坦

突発的変動を積極的に予測するよりも，
平均的な値へ近づく方向に学習



予測結果の比較

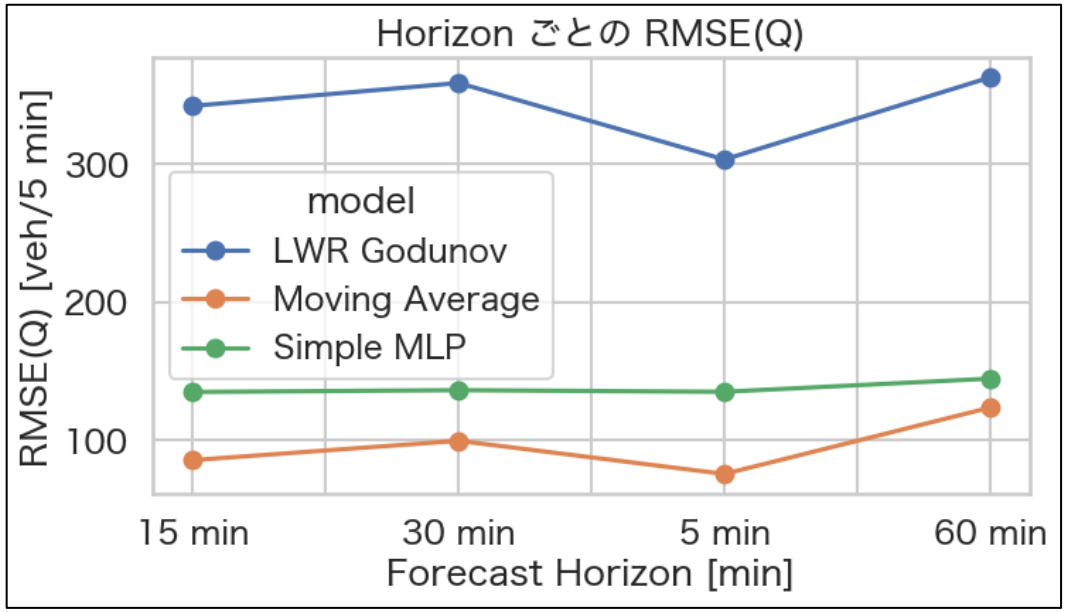
	model	RMSE(Q)	MAE(Q)	RMSE(V)	MAE(V)	RMSE(K)	MAE(K)	Congestion F1
0	Moving Average	102.159004	45.230785	5.600901	2.062885	177.853821	1.061327	0.920158
1	Simple MLP	138.130447	89.582596	8.609674	4.588020	256.156403	2.165269	0.129266
2	LWR Godunov	350.828766	194.017273	60.704292	43.414917	478.873108	19.961706	0.165694

・**移動平均による予測が最も高い性能**

RMSE : $\sqrt{\frac{\sum_{i=1}^{n}(y-\hat{y})^2}{n}}$ → 大きな誤差に強く影響

MAE : $\frac{\sum|y-\hat{y}|}{n}$ → 平均的にどれくらい誤差があるか

→ 両方において**移動平均が最も小さい誤差**



MLP学習の予想問題　→ 欠損のデータ、パターンの周期、学習量

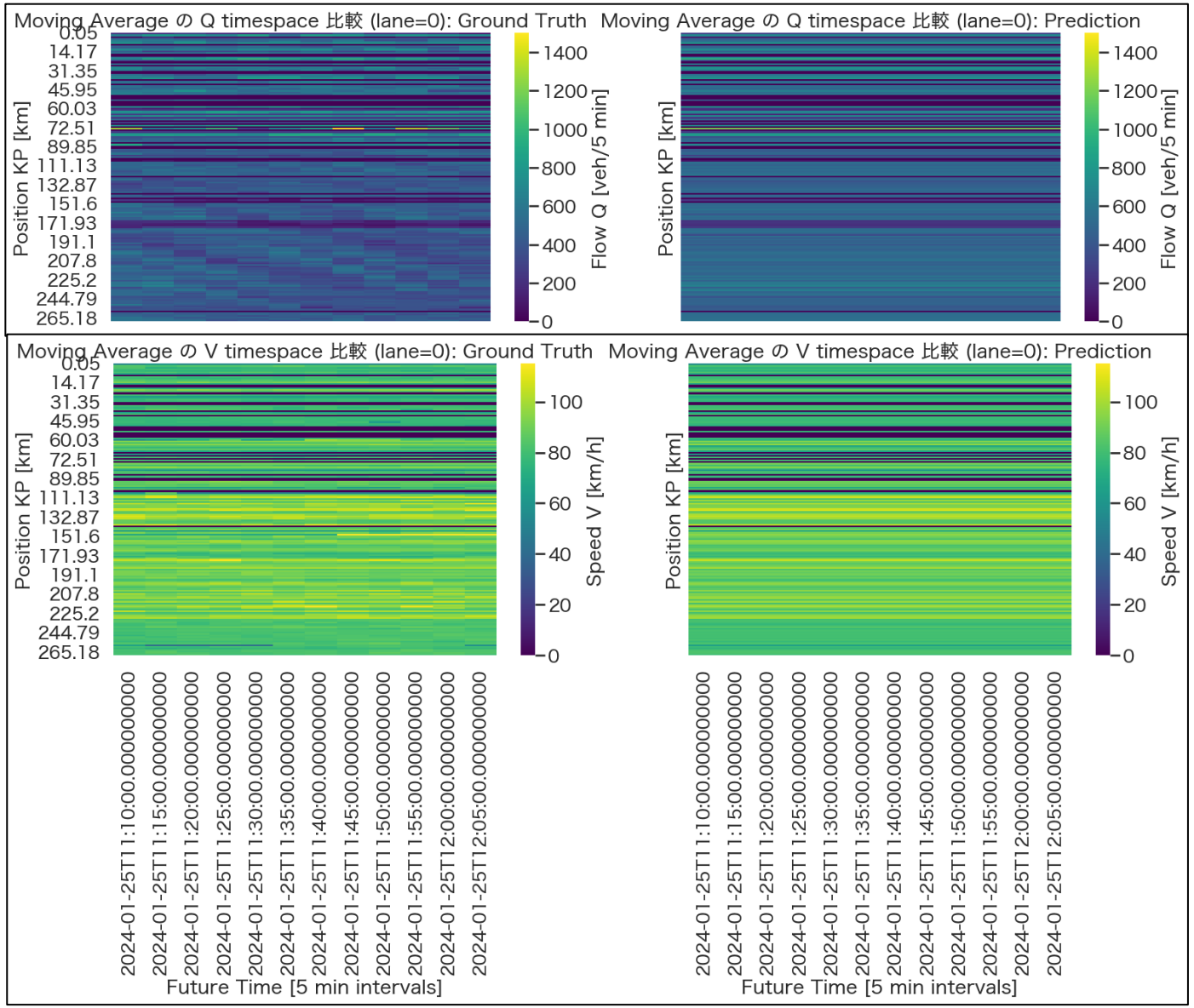
予測結果の比較

・**移動平均による予測結果の検証**

KP方向の空間分布はかなりよく一致

時間方向の細かな変化はほとんど見られない

交通流自体が強い周期性と安定性を持っているため，そのような単純平均ベースの予測でも非常に高い性能を示したと考えられる

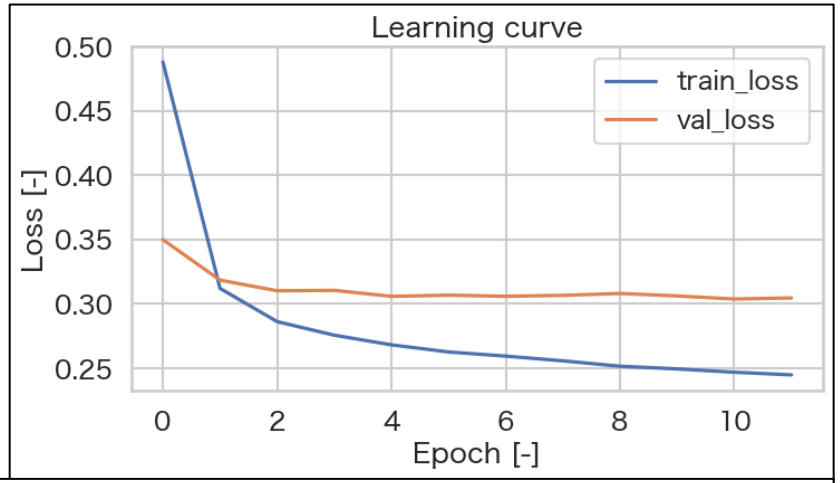


NN（MLP）による予測②

・**変更点**

Lane1, 3 排除 → 欠損率の高いデータを利用しない

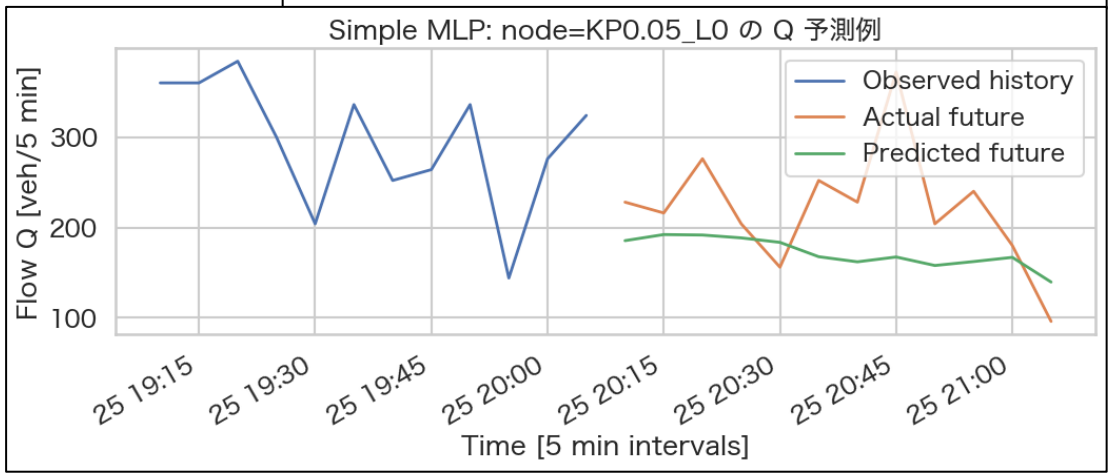
	model	RMSE(Q)	MAE(Q)	RMSE(V)	MAE(V)	RMSE(K)	MAE(K)	Congestion F1
0	Simple MLP	119.271034	75.569946	7.839900	4.314888	228.559494	1.824562	0.370542
1	Moving Average	131.906631	76.688637	7.268131	3.598302	210.239120	1.679780	0.750666
2	LWR Godunov	429.363525	313.796783	65.245750	50.678993	662.990051	38.504036	0.314718



・**MLPの性能は以前より改善**

RMSE(Q) 138 → 119、 MAE(Q) 89 → 75

→ 欠損率の高いlaneがノイズとして働いていた可能性

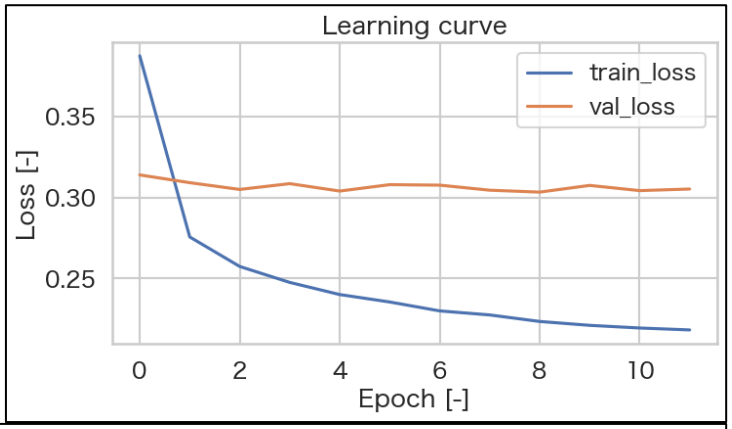


NN（MLP）による予測③

・変更点

Lookback =6 → 過去30分から未来1時間を予測

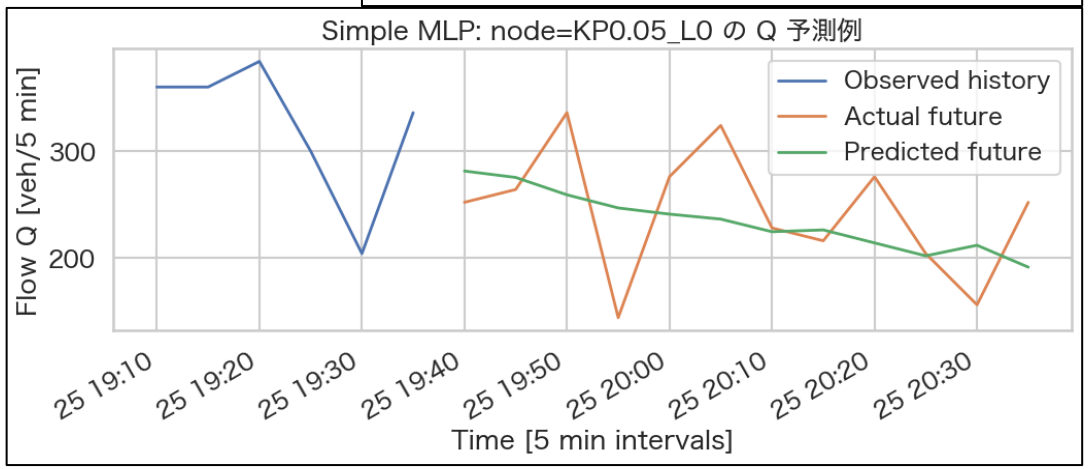
	model	RMSE(Q)	MAE(Q)	RMSE(V)	MAE(V)	RMSE(K)	MAE(K)	Congestion F1
0	Simple MLP	113.330978	73.030251	7.653793	4.242486	799.572998	2.946721	0.407852
1	Moving Average	120.655106	69.658966	7.087114	3.503405	209.857239	1.593376	0.765920
2	LWR Godunov	429.192261	313.711456	65.242790	50.677803	662.876953	38.501007	0.314724



・lookbackを短くしても性能に大きな変化は見られなかった。

→ 直近の交通状態や日周期的な繰り返しパターンの影響が大きいと考えられる

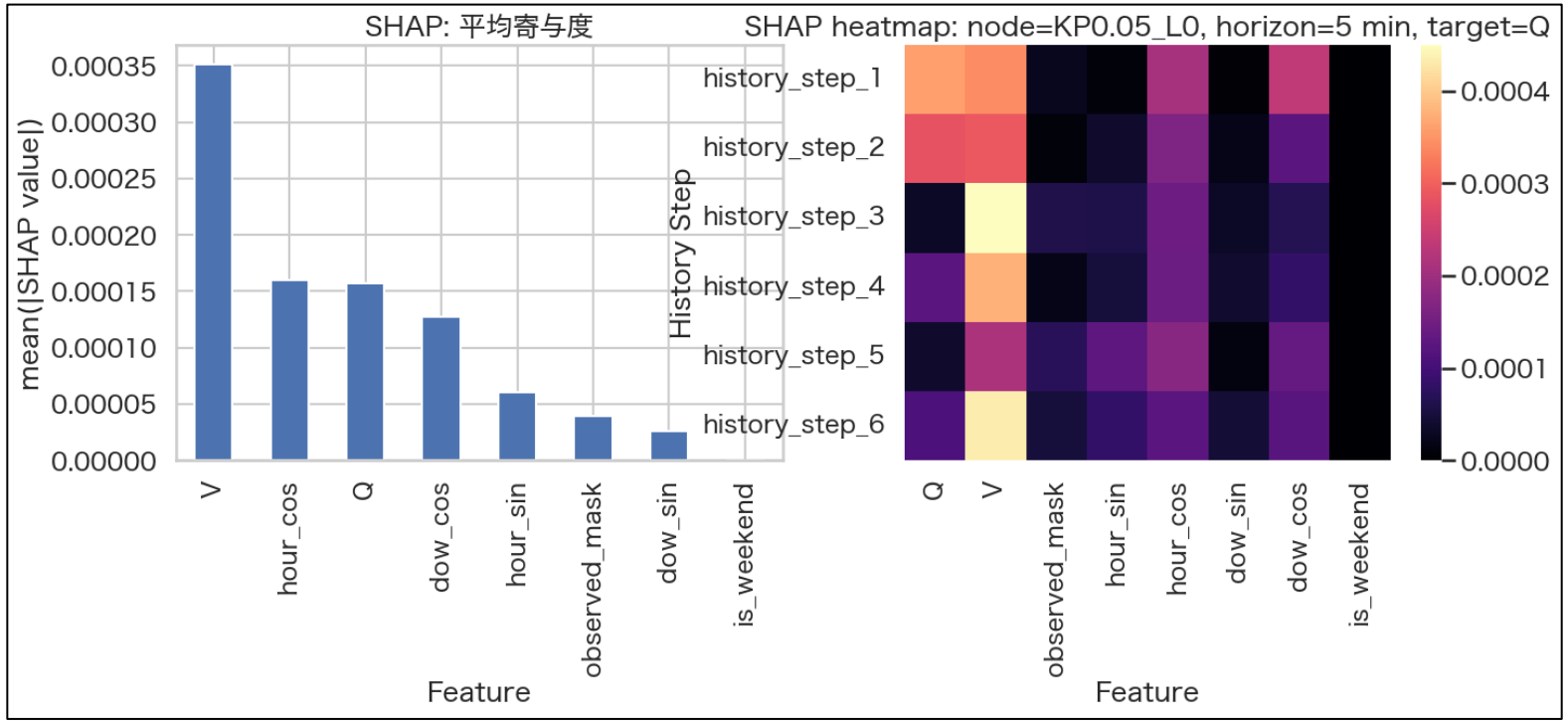
→ MLPが長期的な依存関係を学習するというよりも，平均的な時間帯パターンを学習している可能性



NN（MLP）による予測③

・変更点

Lookback =6 → 過去30分から未来1時間を予測



・に速度Vを最も重要な特徴量として利用していることが確認
→ 交通流において速度変化が混雑状態や流量変化を早期に反映するためであると考えられる

・直近stepの特徴量が強く利用されており，主に短期的な交通状態へ依存して予測している

まとめ

・Moving Averageは非常に単純なモデルであるにもかかわらず，高い性能を示した

　→ 今回の交通データが比較的強い周期性と短期安定性を持っており，「現在状態がしばらく継続する」という仮定が有効に機能したためであると考えられる

・Learning curveからは，基本的な周期パターンは学習できているものの，
　validation lossは途中から改善が小さくなっており，表現力には限界があることが確認

・MLPは全体的に平均的で平坦な予測へ収束する傾向が見られた
　→ ハイパーパラメータを調整しても大きな変化はない