3H3-OS-24a-3in

Deep Sparse Autoencoderによる車両運転状態の可視化

Visualization of driving behavior using Deep Sparse Autoencoder

劉 海龍[*1] 谷口 忠大[*2] 高野 敏明[*2] 竹中 一仁[*3] 坂東 誉司[*3] 田中 雄介[*4]
HaiLong LIU Tadahiro Taniguchi Tosiaki Takano Kazuhito Takenaka Takashi Bando Yusuke Tanaka

[*1]立命館大学 情報理工学研究科
The Graduate School of Information Science and Engineering, Ritsumeikan University

[*2]立命館大学 情報理工学部
The College of Information Science and Engineering, Ritsumeikan University

[*3]株式会社デンソー 研究開発 3 部
Corporate R&D Div.3, DENSO CORPORATION

[*4]株式会社トヨタ I T開発センター 研究部
Technical Research Division, Toyota InfoTechnology Center Co.,Ltd.

With the increase of dimensions in driving behavioral data, the human to intuitively understand the time-series data has become very difficult. We employed a deep sparse autoencoder to extract the three-dimensional representation from raw driving behavioral data with one hundred dimensions. And we proposed a method to visualize driving behavioral on the map by mapping three-dimensional data into the RGB color space. We compared deep sparse autoencoder with other conventional methods such as principal component analysis. As a result, our methods outperformed other conventional methods for visualization of driving behavioral data.

1. はじめに

車両を流れる信号は年々増加しており，車載カメラを通して得られる情報も膨大に蓄積できるようになってきた．現在，運転情報は記録すること自体よりも，その記録し続けた膨大な情報を如何に振り返るか，また，どのようにコンパクトな情報表現にして可視化するかが重要な課題になっている．これに対して，運転時系列データの分類やパターン認識が有用と考えられ，盛んに研究されている．Takenaka らは運転文脈情報を考慮しながら運転意図の変化点を抽出するため，二重分節解析器を運転時系列データに適用した [1]．また，これを用いた運転動画の要約手法を提案し，その有効性を示している [2]. Taniguchi らは二重分節解析器がドライバの運転意図の変化を予測する上でも有効であることを示している [3]．しかし，二重分節解析器は，このようなドライバの運転意図の変化の検出や予測という点では優れていても，運転状況を上手く可視化しドライバの運転行動の振り返りに利用するという意味においては，その有効性が示されていなかった．

そこで本研究では，運転状況の可視化に重点を置き，運転行動を容易に振り返ることができ，走行状態の違いに気づくことのできる低次元情報表現を得ることを目指す．人が理解しやすい様に多次元時系列情報を可視化する上では 与えられた高次元データを 2 次元，もしくは，3 次元へと低次元化することが本質的に重要である．低次元表現を得るためには古典的には主成分分析（PCA）が用いられることが多かったが， PCA は高次元空間上においてデータの分散が大きな軸をとるため，必ずしも，運転時系列データを 3 次元空間で効率的に識別する低次元表現を与えてくれるわけではない．本研究では，近年注目される Deep Learning に着目し，その中でも，特に Deep Sparse Autoencoder を用いて運転時系列データの低次元表現を抽出し，運転状況を地図上にカラー表示する可視化手法を提案する．

連絡先: 劉 海龍，立命館大学，情報理工学研究科，〒 525-8577
滋賀県草津市野路東 1 丁目 1-1，liu@em.ci.ritsumei.ac.jp

2. 先行研究

1984 年，Hinton らは統計力学に基づき，ホップフィールドネットワークを改造し，Boltzmann machines（BM）という二層の相互結合型ネットワークを提案した [4]．しかし，BM の訓練時間は長すぎ，有限時間内に BM でのニューロンの同時確率分布を表すことが難しく，また，この同時確率分布からのサンプリングも困難であるという問題があった．

1986 年，Smolensky は Restricted Boltzmann machine（RBM）を提案した [5]．RBM は BM の同じ深さの層のニューロン間に結合がなく，同じ層内の各ニューロンの事前確率が独立になる特徴がある．これにより，RBM では Gibbs sampling を用いて，この同時確率分布からのサンプリングが可能になった．

同 1986 年，Rumelhart らは並列分散処理の理論に基づいて，ニューラルネットワークの分野で最も有名なバックプロパゲーションニューラルネットワークモデル Back propagation neural network(BPNN）を提案した [6].

2009 年，Bengio は Deep Learning の理論を提案し，RBM と同じ，二層完全無向グラフ構造を持つ Autoencoder(AE) は，モデルの訓練が RBM より容易であることを示した [7].

3. 研究目的

我々が車両を運転する際に存在する潜在的な低次元情報に注目し，観測された運転挙動データに Deep Learning を用いることで，有用な低次元特徴を抽出することを目指す．特に，本研究では Deep Sparse AE (DSAE) を用いて，運転挙動に対する特徴量を抽出する．低次元特徴量は可視化のために三次元特徴量に集約し，地図上にカラー表示する．本研究では，これを Driving Color Map と呼ぶ．

4. 提案手法

4.1 DSAE による高次特徴量の抽出

AE の構造は二層完全無向グラフであり，展開すると隠れ層を中心とする対称構造を持つ三階層有向グラフが得られる．AE は可視層の情報をエンコードして隠れ層の情報を生成し，隠れ層の情報をデコードして可視層のデータを復元する復元層の情報を生成する．復元層のデータと可視層のデータ間の誤差を最小化する場合，隠れ層のデータは可視層のデータに対する特徴ベクトルに相当する．

本稿では AE の隠れ層にスパース性を強める拡張モデルである Sparse AE (SAE) を積み上げた深い構造を持つ教師なし学習モデル DSAE を使用する．

本研究において，運転挙動データの学習セットは $\mathbf{X} \in \mathbb{R}^{D_X \times N_X}$ と定義する．ある時間ステップ t における運転挙動データは

$$\mathbf{x}_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,D_X})^T \in \mathbb{R}^{D_X} \quad (1)$$

であり，D_X はベクトル $\mathbf{x}_t$ の次元数を示し，N_X は学習セット $\mathbf{X}$ の時間ステップ数を表す．

我々は DSAE の活性化関数（Activation function）として，ハイパボリックタンジェント関数 $\mathbf{tanh}(\cdot)$ を使用する．その関数の出力値の値域は $(-1,1)$ であるため，AE の学習のためには学習セット $\mathbf{X}$ を規格化することが必要である．本稿では運転挙動データの各次元の単位が違うことを考慮し，各次元の最大値と最小値を用い，各次元独立に $(-1,1)$ の区間に規格化することが妥当であると考えた．時間ステップ t に対する規格化されたデータ $\mathbf{n}_t \in \mathbb{R}^{D_X}$ は

$$\mathbf{n}_t = (n_{t,1}, n_{t,2}, \ldots, n_{t,D_X})^T \in \mathbb{R}^{D_X}, \quad (2)$$

$$n_{t,d} = 2\left(\frac{x_{t,d} - x_{d\,min}}{x_{d\,max} - x_{d\,min}}\right) - 1 \quad (3)$$

になる．$x_{d\,max}$ と $x_{d\,min}$ は学習セット $\mathbf{X}$ による $\mathbf{x}$ の第 d 次元の最大値と最小値を表す．

また，我々は時系列属性を持つ特徴量を抽出するため，w 時間ステップのデータを 1 つのウィンドウとして設定する．1 つのウィンドウのデータは

$$\mathbf{v}_t = (\mathbf{n}_{t-w+1}, \ldots, \mathbf{n}_t)^T \in \mathbb{R}^{D_V} \quad (t \geq w) \quad (4)$$

と定義する．ここで $\mathbf{v}_t$ は $D_V = w \times D_X$ 次元のデータである．そして，$\mathbf{v}_t$ を時間軸について 1 ステップずつ移動し，$N_V = N_X - w + 1$ 時間ステップのデータセット $\mathbf{V} \in \mathbb{R}^{D_V \times N_V}$ を得る．

第 l 番目の SAE では，まず，入力層のベクトル $\mathbf{v}_t^{(l)}$ に対して，隠れ層の出力値を生成するため，エンコーダの関数を

$$\mathbf{h}(\mathbf{v}_t^{(l)}) = \mathbf{tanh}(\mathbf{W}_e^{(l)} \mathbf{v}_t^{(l)} + \mathbf{b}_e^{(l)}) \in \mathbb{R}^{D_H^{(l)}} \quad (5)$$

と定義する．$D_H^{(l)}$ は第 l 番目の SAE の隠れ層の次元数であり，$\mathbf{W}_e^{(l)} \in \mathbb{R}^{D_H^{(l)} \times D_V^{(l)}}$ は，エンコーダの重み行列であり，$\mathbf{b}_e^{(l)} \in \mathbb{R}^{D_H^{(l)}}$ はエンコーダのバイアスである．

一方，隠れ層から可視層のデータを復元するためのデコーダの関数は

$$\mathbf{r}(\mathbf{v}_t^{(l)}) = \mathbf{tanh}(\mathbf{W}_d^{(l)} \mathbf{h}(\mathbf{v}_t^{(l)}) + \mathbf{b}_d^{(l)}) \in \mathbb{R}^{D_V^{(l)}} \quad (6)$$

と定義する．ただし，$\mathbf{W}_d^{(l)} \in \mathbb{R}^{D_V^{(l)} \times D_H^{(l)}}$ はデコードの重み行列であり，$\mathbf{b}_d^{(l)} \in \mathbb{R}^{D_V^{(l)}}$ はエンコードのバイアスである．また，デコードにより，データを復元可能であるためには，$\mathbf{r}(\mathbf{v}_t^{(l)}) = \mathbf{v}_t^{(l)}$ という仮定が必要である．データセット $\mathbf{V}$ に対して，$\mathbf{r}(\mathbf{v}_t^{(l)})$ と $\mathbf{v}_t^{(l)}$ の間の誤差関数は

$$E(\mathbf{V}^{(l)}) = \frac{1}{2N_V} \sum_{t=1}^{N_V} ||\mathbf{r}(\mathbf{v}_t^{(l)}) - \mathbf{v}_t^{(l)}||_2^2 + \frac{\alpha}{2}(||\mathbf{W}_e^{(l)}||_2^2 + ||\mathbf{W}_d^{(l)}||_2^2) + \beta \sum_{i=1}^{D_H^{(l)}} KL(\theta || \overline{h}_i^{(l)}) \quad (7)$$

とおかれる．エンコードとデコードの重み行列 $\mathbf{W}_e^{(l)}$ と $\mathbf{W}_d^{(l)}$ の要素が大きすぎたりする場合には過学習が起きる可能性が高いため，$\mathbf{W}_e^{(l)}$ と $\mathbf{W}_d^{(l)}$ の L2 ノルムを付加することによって，誤差関数 (式 7) のペナルティ項とする．α はペナルティ項の強さをコントロールするパラメータである．また，隠れ層をスパースにするために，通常は L1 ノルムを使用しるが，L1 ノルムは微分不可能ため，本手法では代わりに式 7 の $\sum_{i=1}^{D_H^{(l)}} KL(\theta || \overline{h}_i^{(l)})$ を用いて，隠れ層に対するスパース項とする．β はスパース項の強さをコントロールするパラメータである．このスパース項は θ と $\overline{h}_i^{(l)}$ をパラメータとする 2 つベルヌーイ分布の KL 情報量である [8].

$$KL(\theta || \overline{h}_i^{(l)}) = \theta \log \frac{\theta}{\overline{h}_i^{(l)}} + (1-\theta) \log \frac{1-\theta}{1-\overline{h}_i^{(l)}} \quad (8)$$

式 8 の θ はスパーシティを制御するパラメータであり，スパース項を最小化すると $\overline{h}_i^{(l)}$ は θ に近づく．$\overline{h}_i^{(l)}$ は隠れ層の第 i 番目次元の平均値を表す．$\overline{h}_i^{(l)}$ を並べた隠れ層の平均値ベクトルは $\overline{\mathbf{h}}^{(l)} \in \mathbb{R}^{D_H^{(l)}}$ と定義し，$\overline{h}_i^{(l)}$ は

$$\overline{h}_i^{(l)} = \frac{1}{2}\left(1 + \frac{1}{N_V} \sum_{t=1}^{N_V} h(\mathbf{v}_t^{(l)})_i\right) \quad (9)$$

で求められる．$h(\mathbf{v}_t^{(l)})_i$ はベクトル $\mathbf{h}(\mathbf{v}_t^{(l)})$ の第 i 番目の要素である．式 8 の中に，log 関数があるため，$\theta / \overline{h}_i^{(l)}$ の範囲は $(0, +\infty)$ であるべきだが，活性化関数 $\mathbf{tanh}(\cdot)$ により，$\overline{h}_i^{(l)}$ の値域は $(-1,1)$ であるため，$\log(\theta / \overline{h}_i^{(l)})$ が計算できない．この問題に対応するため，式 9 は隠れ層の平均値ベクトル $\overline{\mathbf{h}}^{(l)}$ の各要素の区域を $(-1,1)$ から $(0,1)$ に変換している (式 9).

最後に，Back Propagation 法 (BP 法) [6] を用いて，誤差 $E(\mathbf{V}^{(l)})$ を最小化し，バイアス $\mathbf{b}_e^{(l)}, \mathbf{b}_d^{(l)}$ と重み行列 $\mathbf{W}_e^{(l)}, \mathbf{W}_d^{(l)}$ を最適化する．収束を加速するため，勾配を計算した後に，勾配方向への $E(\mathbf{V}^{(l)})$ の最小値まで一次元探索し，学習率を自動的に計算する．一次元探索について，先ず，学習率 λ と探索移動量 z を微小な正の数で初期化する．そして，$\lambda + z$ を用いて各パラメータを更新する．探索移動量 z は更新前の誤差 E と更新後の誤差 E^+ の大小関係により下記で z^+ に更新する．

$$z^+ = \begin{cases} -0.5z & \left(E^+(\mathbf{V}^{(l)}) > E(\mathbf{V}^{(l)})\right) \\ z & \left(E^+(\mathbf{V}^{(l)}) < E(\mathbf{V}^{(l)})\right) \end{cases}, \quad (10)$$

z を更新する際に，誤差の変化量 $|E^+(\mathbf{V}^{(l)}) - E(\mathbf{V}^{(l)})|$ が微小になる場合は，最適な学習率 $\lambda^* = \lambda + z$ を得る．λ^* によって，各重み行列とバイアスを更新する．

以上より，第 l 番目の SAE の最適された隠れ層の特徴量行列 $\mathbf{H}^{(l)} = \left(\mathbf{h}(\mathbf{v}_1^{(l)}), \ldots, \mathbf{h}(\mathbf{v}_t^{(l)})\right) \in \mathbb{R}^{D_H^{(l)} \times N_V}$ はデータセット $\mathbf{V}^{(l)}$

に対する特徴量として得られる．そして，第 l 番目の SAE の特徴行列 $\mathbf{H}^{(l)}$ は次の SAE の可視層の入力行列 $\mathbf{V}^{(l+1)} \in \mathbb{R}^{D_H^{(l)} \times N_V}$ として $(\mathbf{V}^{(l+1)} = \mathbf{H}^{(l)})$，第 $l+1$ 番目の SAE を学習する．以上より，複数の SAE を用いて，運転挙動データに対する高次特徴量を抽出する．

4.2 高次特徴量の可視化

前節で抽出された 3 次元の高次特徴量 $\mathbf{h}(\mathbf{V}^{(FINAL)})$ を 3 次元の RGB 色空間に線形写像する*1．

$$RGB_{t,i} = \frac{1}{2}\left(\frac{h(\mathbf{v}_t^{(FINAL)})_i - h(\mathbf{V}^{(FINAL)})_{min}}{h(\mathbf{V}^{(FINAL)})_{max} - h(\mathbf{V}^{(FINAL)})_{min}} + 1\right), \qquad (11)$$

式 11 の $h(\mathbf{v}_t^{(FINAL)})_i$ は最後の SAE の第 t 時間ステップに対する特徴ベクトル $h(\mathbf{v}_t^{(FINAL)})$ の第 i 番目の要素である．また，$h(\mathbf{V}^{(FINAL)})_{max}$ と $h(\mathbf{V}^{(FINAL)})_{min}$ は抽出された低次元特徴量のデータ全体に対する最大値と最小値である．最後に，得る色を運転地図上で着色して，Driving Color Map を作成する．

5. 実験

我々は DSAE の特徴抽出性能を比較するため，PCA，Kernel PCA（KPCA）と一個の SAE を用いて，3 次元の特徴量を抽出した．

5.1 実験条件

本稿では実際の車両で収集した下記の 10 次元のデータを運転挙動データとして採用する．各次元の情報は以下の式を表す．

$$\begin{aligned}\mathbf{x} = \ & (\text{アクセル開度率 [\%]}, \text{ブレーキ MC 圧 [MPa]},\\ & \text{ステアリング操舵角 [deg]}, \text{車輪速 [km/h]},\\ & \text{メータ車速 [km/h]}, \text{エンジン回転数 [rpm]},\\ & \text{縦加速度 [m/s}^2\text{]}, \text{横加速度 [m/s}^2\text{]},\\ & \text{ヨーレート [rad/s]}, \text{ウィンカ [右:-1, 左:1,OFF:0]})^T \end{aligned} \qquad (12)$$

我々は 2 種類のコースで計 10 周分の運転挙動データを取得した．第 1 周回から第 5 周回はコース 1 を，第 6 周回から第 10 周回はコース 2 を走行した．データを収集するフレームレートは 10 であり，合わせて 12958 フレーム分のデータを得た．

抽出された高次特徴量の不変性を考察するため，第 1〜4 周回，第 6〜9 周回を学習セットとして訓練した．汎化性の確認のために第 5 周回と第 10 周回をテストセットとして低次元特徴量を予測した．

DSAE について，我々は経験的に，パラメータを $\alpha = 0.15$, $\beta = 0.7$, $\theta = 0.5$ に設定した．更新毎の誤差の差が 0.0001 になると，収束したと判定した．そして，ウィンドウサイズ $w = 10$（1 秒間）を設定し，100 次元のウィンドウ付きのデータを得た．この 100 次元のデータを DSAE を用いて，80 次元，50 次元，30 次元，10 次元，3 次元と段階的に，低次元の高次特徴量を抽出した．一方，PCA と KPCA においては 100 次元のデータから 3 次元の特徴量を直接算出した．SAE については，正規化後の 10 次元のデータを入力として 3 次元の特徴量を算出した．

5.2 可視化実験の結果

学習セットに対して DSAE により可視化された Driving Color Map を図 1 に示す．各周回に対して，同じの色は地図上のほぼ同じ場所に分布することが見て取れた．そして，テストセットである第 5 周回，第 10 周回のデータを訓練された DSAE に入力した際の可視化結果を図 2 に示す．予測結果と学習セットによる可視化結果とを対比すると，地図上の色分けの場所がほぼ一致することが見て取れ，新しいデータに対しても適切な可視化が行われる様子が観察された．

次に，我々は DSAE による Driving Color Map(図 1) とカメラ画像を対比させながら観察し，抽出された各色がどんな意味を持っているかを調査した．例えば，白色は高速運転，水色は加速，赤色は減速，緑はウィンカをオンにする右折，オレンジ色は右折する前の減速，濃い青は左折，紫は左折したい場合の減速などである．

最後に，PCA，KPCA と SAE による Driving Color Map を図 3 に示す．PCA と KPCA の結果を見ると，色の種類が DSAE の場合より少ない．運転挙動データに対して，PCA と KPCA は DSAE と比べて，特徴の抽出性能が低く，抽出された特徴が少ないことが分かった．また，SAE の場合は色の種類は DSAE の場合とほぼ同じであったが，DASE の方が色の変化が滑らかであり，細かい運転挙動に対する特徴の可視化が可能であった．

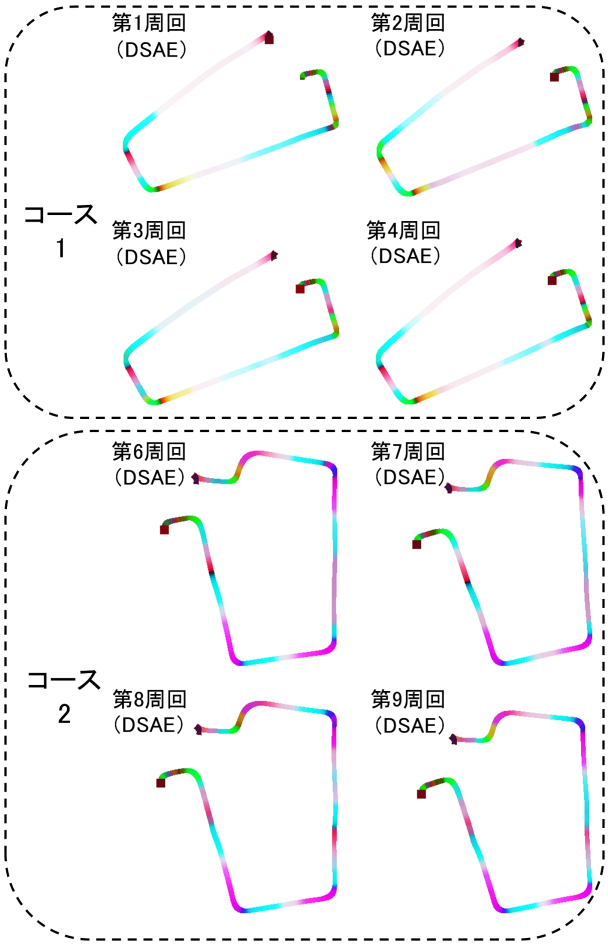

図 1: DSAE による学習セットの Driving Color Map

5.3 評価実験の結果

次に PCA，KPCA，SAE と DSAE により，Driving Color Map 上で可視化された特徴量を比較するため，評価実験を行った．本稿では 8 人の被験者に「どの手法による Driving Color Map

*1 OpenGL の RGB 色空間の区間は [0,1] である

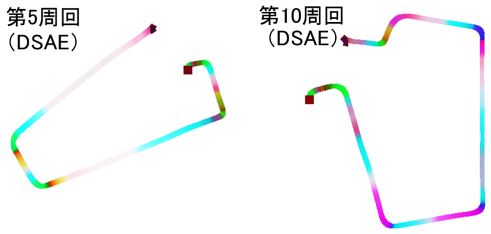

図 2: DSAE によるテストセットの Driving Color Map

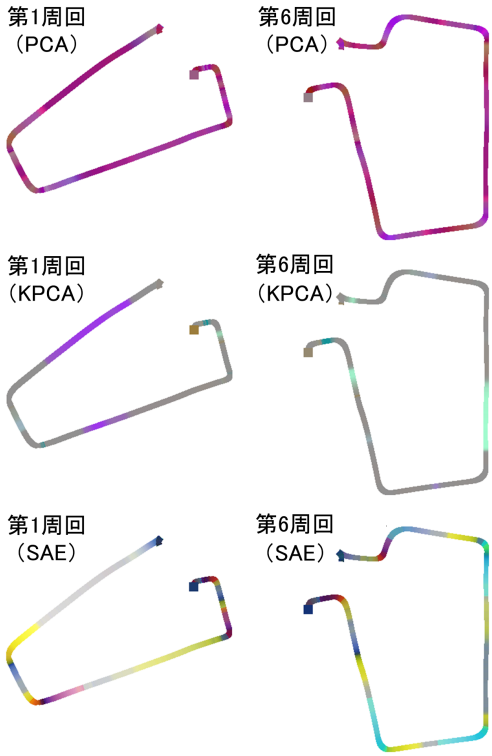

図 3: PCA，KPCA と SAE による Driving Color Map

が車の運転パターンを弁別しやすいか」と質問し，それぞれの手法をランキングさせた．分散分析と多重比較分析により，評価実験の結果の検定を行ったところ，DSAE が最も良いと判定された．評価実験による手法の順位を表 1 に示す．また，評価実験の結果に対して，分散分析と Holm 法による多重比較を用いて有意水準を検定した．多重比較による P 値を表 2 に示す．PCA-DSAE と KPCA-DSAE の場合は 0.001 以下，PCA-SAE の場合は 0.01 以下，KPCA-SAE の場合は 0.05 以下であった．また，分散分析による P 値は 1.72×10^{-5} であった．本評価実験の結果は十分有意であった．

6. まとめ

本研究では DSAE を用いて，運転挙動データに対する低次元特徴量を抽出した．そして， DSAE による可視化手法として Driving Color Map を提案した．実験では DSAE と PCA，KPCA，SAE による可視化の結果を対比した．DSAE はデータの非線形性と時系列性に対応する能力が高く，新しいデータに対して安定的に低次元特徴量を可視化できた．評価実験により，DSAE を用いて作成した Driving Color Map は PCA，KPCA，SAE を用いて作成した場合よりも，人間にとって理解しやすいことが分かった．

表 1: 評価実験による手法の順位

手法	PCA	KPCA	SAE	DSAE
平均順位	3.5	3.1	2.0	1.4

表 2: 多重比較による P 値

	DSAE	KPCA	PCA
KPCA	0.0005	-	-
PCA	0.00004	0.3	-
SAE	0.2	0.02	0.002

参考文献

[1] K. Takenaka, T. Bando, S. Nagasaka, T. Taniguchi, and K. Hitomi, “Contextual scene segmentation of driving behavior based on double articulation analyzer,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4847–4852.

[2] K. Takenaka, T. Bando, S. Nagasaka, and T. Taniguchi, “Drive video summarization based on double articulation structure of driving behavior,” in *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012, pp. 1169–1172.

[3] T. Taniguchi, S. Nagasaka, K. Hitomi, N. P. Chandrasiri, and T. Bando, “Semiotic prediction of driving behavior using unsupervised double articulation analyzer,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012, pp. 849–854.

[4] G. E. Hinton, T. J. Sejnowski, and D. H. Ackley, *Boltzmann machines: Constraint satisfaction networks that learn*. Carnegie-Mellon University, Department of Computer Science Pittsburgh, PA, 1984.

[5] P. Smolensky, “Information processing in dynamical systems: Foundations of harmony theory,” 1986.

[6] D. E. Rumelhart, G. E. Hintont, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[7] Y. Bengio, “Learning deep architectures for ai,” *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[8] A. Ng, “Sparse autoencoder,” *CS294A Lecture notes*, p. 72, 2011.