Nonparametric estimation of conditional class probabilities using deep neural networks

Atsutomo Yara[1], Yoshikazu Terada[1, 2]

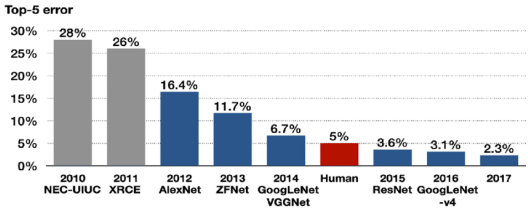
[1]Graduate School of Engineering Science, Osaka University,
[2]Center for Advanced Intelligence Project (AIP), RIKEN

0. Table of Contents

1. Introduction

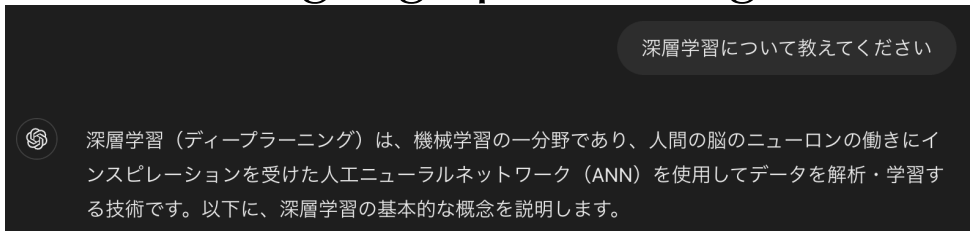
1.1 Deep learning

- **Development of deep learning**
  - Deep learning has shown extremely high performance in various real-world problems:
    - Image recognition



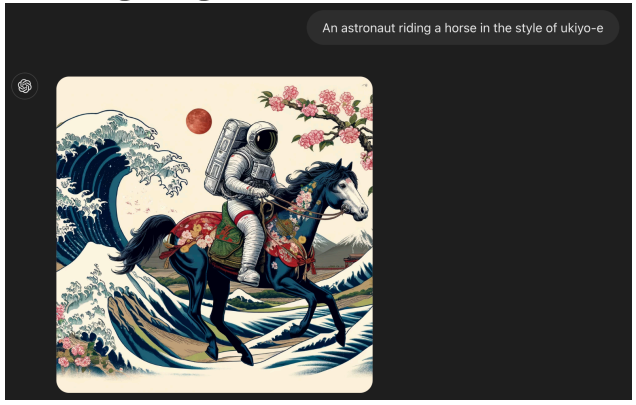
Comparison of performance in ILSVRC.
(Gray) Pre-deep learning methods,
(Blue) Deep learning, (Red) Human, from Kang et al. (2020)

    - Natural language processing



ChatGPT, OpenAI

    - Image generation



1.1 Deep learning

- **Importance of theoretical study**
  - A simple question arises

    Q. Can deep learning solve any problem?

    ⇒ In fact, theoretical studies have provided **negative answers** to this question! (**No-free-lunch theorem**, **Slow rates of convergence**)
  - **Claim of the No-free-lunch theorem (informal)：**

    There are problems for which any learning algorithm performs poorly, meaning there is no universal learning method that works well for all the problems

    ⇒ It is important to develop various methods and investigate when each method works well. This highlights the significance of theoretical research

1.1 Deep learning

- **Explanation of deep learning performance**
  - Theoretical studies to explain the performance of deep learning are also actively being pursued
  - Major perspectives in theoretical research on deep learning:
    - **<u>Generalization ability</u>**
      - ✓**<u>How well it performs on new data</u>**
    - Approximation ability
      - ✓ How efficiently it can be approximate functions
    - Optimization
      - ✓ Whether the parameters that fit the data can be obtained through optimization

1.2 Generalization error analysis in deep learning

- **Statistical learning theory**
  - The goal of statistical learning theory is to provide a theoretical justification for machine learning methods.
  - Evaluation of justification = Evaluation of generalization error
    - Small generalization error ⇒ ″Good″ estimator
  - Points of view for evaluating generalization error
    - Does the generalization error converge to zero as the amount of training data increases?
    - If it does converge, how fast does it converge? (Convergence rate)
    - Optimality of convergence rate

1.2 Generalization error analysis in deep learning

- **Empirical Risk Minimization, ERM**
  - Generalization error (Excess risk):

$$\mathbb{E}_{(X,Y)}[\ell(Y, \hat{f}(X))] - \inf_{f:\text{measurable}} \mathbb{E}_{(X,Y)}[\ell(Y, f(X))]$$

  - **The expected loss of new data is unknown** ⇒ **ERM**
  - ERM: Minimizes the loss over observed data $\mathcal{D}_n$
    - The estimator $\hat{f}$ is searched within a hypothesis space (model) $\mathcal{F}$
    - In this study, the model $\mathcal{F}$ is Deep Neural Networks (DNNs)

$$\hat{f} \in \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(X_i))$$

1.2 Generalization error analysis in deep learning

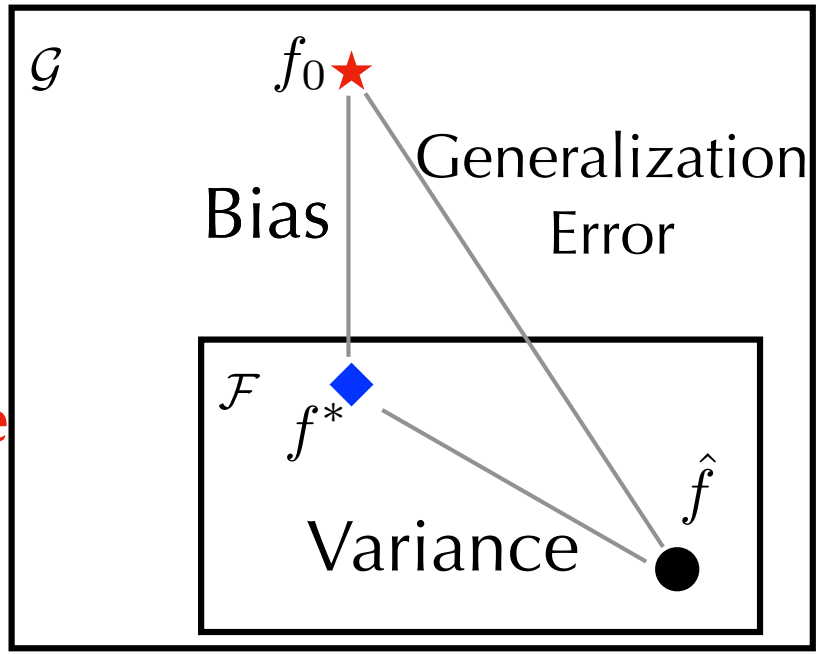
- **Bias-variance decomposition of generalization error**
  - $L(f) = \mathbb{E}_{(X,Y)}[\ell(Y, f(X))]$: Expected risk

$$L(\hat{f}) - \inf_{f:\text{measurable}} L(f)$$

$$= \underbrace{L(\hat{f}) - \inf_{f \in \mathcal{F}} L(f)}_{\text{Variance}} + \underbrace{\inf_{f \in \mathcal{F}} L(f) - \inf_{f:\text{measurable}} L(f)}_{\text{Bias}}$$

$\mathcal{G}$: Measurable functions
$\mathcal{F}$: Model
★ $f_0 = \min_{f \in \mathcal{G}} L(f)$
◆ $f^* = \min_{f \in \mathcal{F}} L(f)$

There is a trade-off between bias and variance that depends on the complexity of the model



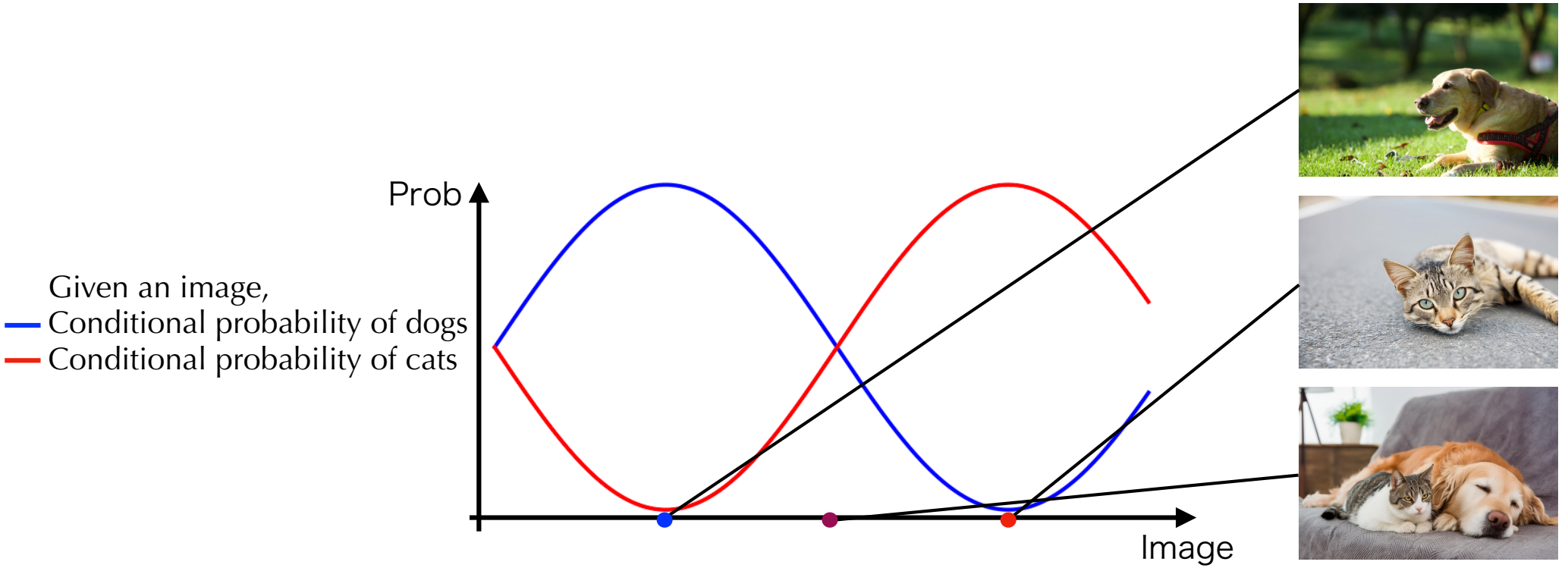
1.3 Classification and conditional probability

- **Set-up of the classification problem**
  - We want to predict its label Y (e.g. whether it is ill or not) from input $\boldsymbol{X}$ (e.g. medical images)
  - True conditional class probability:

$$\eta_k(\boldsymbol{x}) := \mathbb{P}(Y = k \mid \boldsymbol{X} = \boldsymbol{x}), \boldsymbol{\eta}(\boldsymbol{x}) := (\eta_1(\boldsymbol{x}), \ldots, \eta_K(\boldsymbol{x}))^\top$$

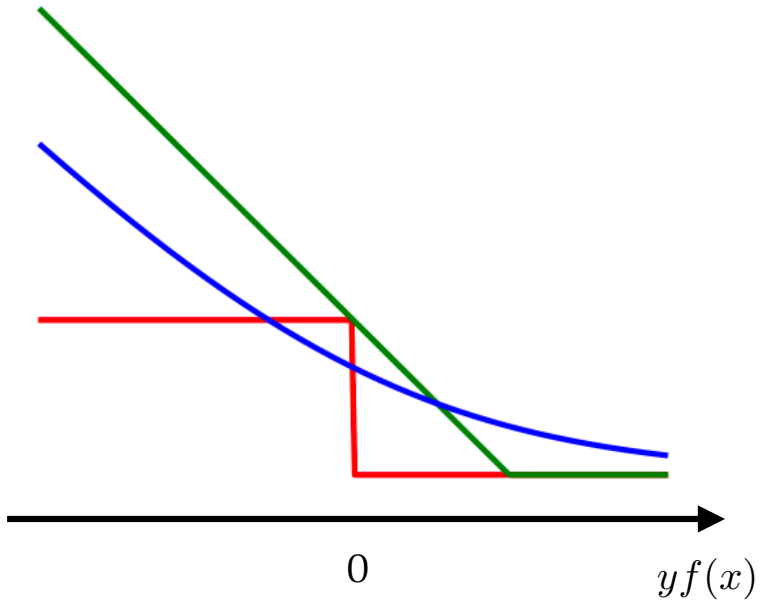
  - Given $\boldsymbol{X}$, Y follows the following multinomial distribution

$$Y|\boldsymbol{X} = \boldsymbol{x} \sim \text{Multi}(\boldsymbol{\eta}(\boldsymbol{x}))$$



1.3 Classification and conditional probability

- **Learning method**
  - Estimators are obtained by ERM
  - There are various loss functions
    - 0-1loss ←Most intuitive loss, but difficult to optimise
      $\ell(y, f(x)) := \mathbb{1}(yf(x) > 0)$
    - Hinge loss
      $\ell(y, f(x)) := 0 \vee (1 - yf(x))$
    - Logistic loss
      $\ell(y, f(x)) := \log(1 + e^{-yf(x)})$



1.3 Classification and conditional probability

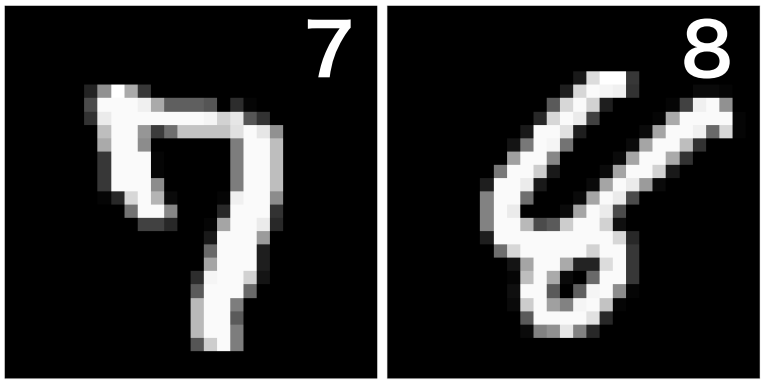
- **Advantage of conditional probability**
  - If hinge loss is used, only the classifier itself can be obtained
    - i.e. only label predictions can be made.
  - It can be more convenient in some situations to estimate conditional probabilities
    - e.g. AI-based disease diagnosis
  - If conditional probabilities can be estimated, they can be used to support human decision-making!
  - Method for estimating conditional probabilities:
    ⇒ **Use the negative log-likelihood of a multinomial distribution as the loss function (Maximum Likelihood Estimation)**

1.4 Classification using deep learning

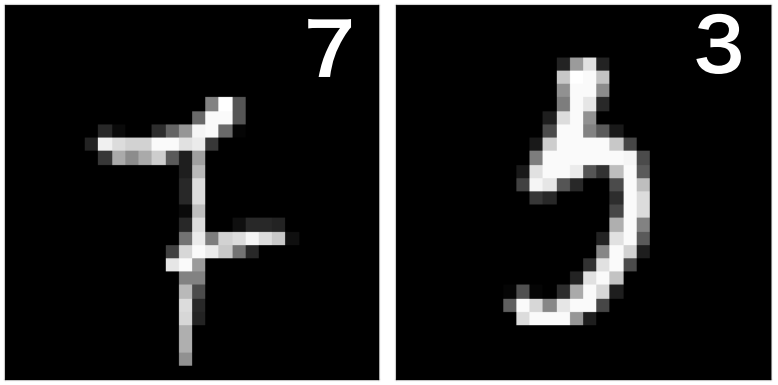
- **Classification using deep learning**
  - In classification problems using deep learning, it is possible to estimate conditional probabilities by using a DNN with a softmax function in the output layer

$$softmax_k(x_1, \ldots, x_K) := \frac{e^{x_k}}{\sum_{j=1}^{K} e^j}$$

  - An example of classification in MNIST (comparison with kNN)
    - **Conditional probabilities for digits that are difficult to distinguish are estimated to be smaller**



DNN **4: 10%, 7: 90%** **4: 10%, 8 : 90%**
kNN 7: 80%, 9: 20% 4: 30%, 8 : 20%, 9: 50%



7: 60%, 8 : 40% **3: 60%, 4: 10%, 5: 30%**
1: 100% 3: 40%, 5: 50%, 9: 10%

1.5 Semi-parametric estimation and conditional probability

- **Relationship with Double/Debiased Machine Learning (DML)**
  - In the context of causal inference, the propensity score (i.e., the **conditional probability** of assignment given covariates) plays an important role as a nuisance parameter in estimating causal effects
  - When the nuisance parameter is complex, machine learning (ML) methods can be used to provide a natural estimator
  - However, the bias and overfitting that can result from using ML may negatively impact the estimator
    - The slow convergence rate (the $n^{-1/2}$-consistency does not hold)

  ⇒ DML (Chernozhukov et al., 2018) can solve this problem!

1.5 Semi-parametric estimation and conditional probability

- **Relationship with Double/Debiased Machine Learning (DML)**
  - In DML, it is necessary to demonstrate a $o_p(n^{-1/4})$ convergence rate for the nonparametric estimation of nuisance parameters
    ⇒ For example, when estimating propensity scores using deep learning, it is essential to analyze the generalization error of the estimator with deep learning
  - In this study, we derive the convergence rate for the estimation of conditional probabilities using deep learning

2. Related works

2.2 Convergence of conditional probability

- **The difficulty of theoretical analysis in maximum likelihood estimation of conditional probabilities**
  - Generalization error:

$$\mathbb{E}_{\boldsymbol{X}}\left[\mathrm{KL}\left(\boldsymbol{\eta}(\boldsymbol{X}) \,\|\, \hat{\boldsymbol{p}}(\boldsymbol{X})\right)\right] \qquad (*)$$

    - KL: Kullback-Leibler divergence
  - Since the KL divergence can easily diverge, directly evaluating (*) is challenging.
    - If $\mathcal{F}_n$ contains all piecewise constant conditional class probabilities with at most two pieces or all piecewise linear conditional class probabilities with at most three pieces

    $\Rightarrow$ (*) diverges. (Lemma 2.1 in Bos & Schmidt-Hieber, 2022)
  - **Various approaches have been proposed to address this problem (Ohn & Kim, 2022; Bos & Schmidt-Hieber, 2022; Bilodeau et al., 2023)**

2.4 Overview of this study

- **In nonparametric maximum likelihood estimation, evaluating the KL divergence is challenging**
  - In studies such as Ohn and Kim (2022), strong assumptions are required...

  ⇒ By using the approach of van de Geer (2000) and directly evaluating the Hellinger distance, the issue can be resolved (Theorem 1)
  - **No strong assumptions are needed, and the proof is more straightforward**
  - **Remark:** In Bos & Schmidt-Hieber (2022), the issue is avoided by considering a truncated KL divergence
  - **Remark:** In Bilodeau et al. (2023), the issue is avoided by using a transformed estimator of the MLE.
- **Application to the theoretical analysis of logistic regression using deep learning (Theorem 2)**
  - **It suggests the possibility of avoiding the curse of dimensionality even in classification problems**
  - **Results have been demonstrated with non-sparse DNNs**
- **It has been proven that deep learning is minimax optimal (Theorem 3)**

2.4 Overview of this study

- **Comparison with related works**

	Metric	Loss functions	Model	Notes
Kim et al.	Misclassification rate	Hinge loss Logistic loss	Sparse DNN	Only classifier Minimax optimal
Ohn & Kim	KL divergence	Logistic loss	Sparse DNN	Strong assumption With reguralization
Bos & Schmidt-Hieber	Truncated KL divergence	Negative log likelihood	Sparse DNN	Weak assumption
Bildeau et al.	KL divergence	Negative log likelihood	Well-specified	Transformed MLE
This work	**Hellinger distance**	**Negative log likelihood**	**Non-sparse DNN**	**Weak assumption** **Minimax optimal**

3. Main results

3.2 Notations and assumptions

- **Notations**
  - $H\left(P,Q\right)^2 := \frac{1}{2}\int\left(\sqrt{dP}-\sqrt{dQ}\right)^2$: The squared Hellinger distance
  - $R(\boldsymbol{\eta}(\boldsymbol{X}),\boldsymbol{p}(\boldsymbol{X})) := \mathbb{E}_{\boldsymbol{X}}\left[H^2(\boldsymbol{\eta}(\boldsymbol{X}),\boldsymbol{p}(\boldsymbol{X}))\right]$
    - In this study, we discuss convergence with respect to $R(\boldsymbol{\eta}(\boldsymbol{X}),\hat{\boldsymbol{p}}(\boldsymbol{X}))$instead of $\mathbb{E}_{\boldsymbol{X}}\left[\mathrm{KL}\left(\boldsymbol{\eta}(\boldsymbol{X})\parallel\hat{\boldsymbol{p}}(\boldsymbol{X})\right)\right]$
    - $R(\boldsymbol{\eta}(\boldsymbol{X}),\hat{\boldsymbol{p}}_n(\boldsymbol{X})) \leq \mathbb{E}_{\boldsymbol{X}}\left[\mathrm{KL}\left(\boldsymbol{\eta}(\boldsymbol{X})\parallel\hat{\boldsymbol{p}}_n(\boldsymbol{X})\right)\right]$
- **Assumptions**
  - $\exists\tilde{\boldsymbol{p}}_n\in\mathcal{F}_n,\ \exists c_0>0;\ \dfrac{\eta_k(\mathbf{x})}{\tilde{p}_{n,k}(\mathbf{x})}\leq c_0^2,\ \forall n\in\mathbb{N}, k=1,\ldots,K$
    - **<u>This is automatically satisfied when estimation is performed using DNNs</u>**

3.3 Maximum likelihood estimation of conditional probability

⭐Oracle inequality

Theorem 1 (Oracle inequality)

Consider a classification problem with K classes. Let $\hat{\boldsymbol{p}}_n$ be the maximum likelihood estimator. Under some conditions, we have

$$\mathbb{E}\left[R\left(\hat{\mathbf{p}}_n, \boldsymbol{\eta}\right)\right] \leq 514(1+c_0^2)\left(\underbrace{\delta_n^2}_{\text{Variance}} + \underbrace{R\left(\tilde{\mathbf{p}}_n, \boldsymbol{\eta}\right)}_{\text{Bias}}\right) + \frac{c}{n},$$

- δ_n corresponds to the complexity of the model $\mathcal{F}_n$

⇒ The more complex the model, the larger the variance becomes

- The more complex the model, the smaller the bias becomes (as a more complex model is more likely to capture the true conditional probabilities)
- If the **complexity of the model** and the **bias** can be determined, the convergence rate of the MLE can be derived!

3.4 Logistic regression using deep learning

- **DNN model**
  - $\sigma(x) := \max(0, x)$: ReLU activation, $\mathbf{v} = (v_1, \ldots, v_r) \in \mathbb{R}$: Bias
  - $\sigma_{\mathbf{v}}(y_1, \ldots, y_r) = (\sigma(y_1 - v_1), \ldots, \sigma(y_r - v_r))$: Activation with bias
  - L: Depth, $\boldsymbol{m} = (m_0, \ldots, m_{L+1})$: Width of each layer,
  - $W_j \in \mathbb{R}^{m_{j+1} \times m_j}$: Weight matrix,
  - $\boldsymbol{\Phi}$: softmax function

$$f : \mathbb{R}^{m_0} \to \mathbb{R}^{m_{L+1}},\ \boldsymbol{x} \mapsto f(\boldsymbol{x}) = \boldsymbol{\Phi} W_L \sigma_{\boldsymbol{v}_L} W_{L-1} \sigma_{\boldsymbol{v}_{L-1}} \cdots W_1 \sigma_{\mathbf{v}_1} W_0 \boldsymbol{x} \quad (\bigstar)$$

$$\mathcal{F}(L, \boldsymbol{m}, B) := \left\{ \text{f of the form } (\bigstar) : \max_{j=0,\ldots,L} \|W_j\|_\infty \vee |\boldsymbol{v}_j|_\infty \leq B \right\}$$

3.4 Logistic regression using deep learning

- **Underlying function space for the true conditional probability**
  - Hölder space with smoothness $\boldsymbol{\beta}$

$$\mathcal{C}^{\beta}(D,Q) = \Bigg\{ f : D \subset \mathbb{R}^m \to \mathbb{R} : \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|<\beta} \|\partial^{\boldsymbol{\alpha}} f\|_{\infty} + \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=\lfloor\beta\rfloor} \sup_{\boldsymbol{x},\boldsymbol{y}\in D,\, \boldsymbol{x}\neq\boldsymbol{y}} \frac{|\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}) - \partial^{\boldsymbol{\alpha}} f(\boldsymbol{y})|}{|\boldsymbol{x}-\mathbf{y}|_{\infty}^{\beta-\lfloor\beta\rfloor}} \leq Q \Bigg\}$$

  - Composition structured functions

$$\mathcal{G}_{\text{comp}}(r,\mathbf{d},\mathbf{t},\boldsymbol{\beta},Q) = \{ f = \boldsymbol{g}_r \circ \cdots \circ \boldsymbol{g}_0 : \boldsymbol{g}_i = (g_{ij})_j : [a_i,b_i]^{d_i} \to [a_{i+1},b_{i+1}]^{d_{i+1}},$$
$$g_{ij} \in \mathcal{C}^{\beta_i}([a_i,b_i]^{t_i},Q), \text{for some} |a_i|,|b_i| \leq Q \}$$

3.4 Logistic regression using deep learning

- **Underlying function space for the true conditional probability**

  - Composition structured functions

$$\mathcal{G}_{\text{comp}}(r, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, Q) = \{f = \boldsymbol{g}_r \circ \cdots \circ \boldsymbol{g}_0 : \boldsymbol{g}_i = (g_{ij})_j : [a_i, b_i]^{d_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}},$$
$$g_{ij} \in \mathcal{C}^{\beta_i}([a_i, b_i]^{t_i}, Q), \text{for some} |a_i|, |b_i| \le Q\}$$

  - $f(x_1, x_2, x_3) = g_1 \circ \boldsymbol{g}_2(x_1, x_2, x_3) = g_{11}(g_{01}(x_1, x_3), g_{02}(x_1, x_2))$

    3 dim　　2 dim

  - **There is an underlying low-dimensional structure**

  - It contains a reasonable example of conditional probability

$$\eta_k(x_1, \ldots, x_d) = \Psi_k \left(\sum_{i=1}^{d} f_{ki}(x_i) \right)$$

    where $\Psi_k(\boldsymbol{x}) = e^{x_k} / \sum_{i=1}^{d} e^{x_i}$

- $\eta_k \in \mathcal{G}_{\text{comp}}(r, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, Q), \quad k = 1, \ldots, K$

3.4 Logistic regression using deep learning

⭐Convergence rate

Theorem 3 (Convergence rate)

Consider the family of neural networks $\mathcal{F}(L, \boldsymbol{m}, B)$ that satisfy the following conditions as the model.

$$\text{(i)}\ \ L \asymp \log(n), \quad \text{(ii)}\ \ \sqrt{n\phi_n} \lesssim \min_{i=1,\ldots,L} m_i, \quad \text{(iii)}\ B \asymp 1/\sqrt{\phi_n}.$$

Then, there exists a constant C depending only on $r, d, \boldsymbol{t}, \boldsymbol{\beta}, Q, K$ s.t.

$$\mathbb{E}\left[R\left(\hat{\boldsymbol{p}}_n, \boldsymbol{\eta}\right)\right] \leq C\phi_n L \log^2(n)$$

For sufficiently large n. Here,

$$\beta_i^* = \beta_i \prod_{l=i+1}^{r} (\beta_l \wedge 1),\ \ \phi_n = \max_{i=0,\ldots,r} n^{-\frac{\beta_i^*}{\beta_i^* + t_i}}\ .$$

- The convergence rate is ϕ_n up to log-factor
- **There is no restriction for the sparseness of DNN**
- **The convergence rate does not depend on the dimension d, thus avoiding the curse of dimensionality**
- Deep learning can effectively capture the underlying low-dimensional structure (the structure of composite functions).

3.4 Logistic regression using deep learning

- **Why sparse DNN?**
  - Many studies assume sparsity in DNN models
    - e.g. Suzuki (2019), Schmidt-Hieber (2020)
  - Typically, generalization error is bounded by

$$\frac{\text{"Complexity"}}{n} + \text{Bias}^2$$

  - The complexity increases as the number of non-zero parameters in the DNN grows.
    ⇒ Imposing sparsity on the DNN leads to a tighter bound
  - In this study, we derived a tighter convergence rate without imposing sparsity by using the tighter bias bound from Kohler & Langer (2021)

3.4 Logistic regression using deep learning

⭐Minimax optimality

- Is the rate derived in Theorem 2 optimal in some sense?

Theorem 3 (Minimax optimality)

Under some conditions, we have

$$\inf_{\hat{\boldsymbol{p}}_n} \sup_{\eta_k \in \mathcal{G}_{\mathrm{comp}}(r,\boldsymbol{d},\boldsymbol{t},\boldsymbol{\beta},Q),\ k=1,\ldots,K} \mathbb{E}\left[R\left(\hat{\boldsymbol{p}}_n, \boldsymbol{\eta}\right)\right] \geq c\phi_n$$

where infimum is taken over all estimators.

- The convergence rate derived in Theorem 3 is minimax optimal up to log factor

3.5 Summary

- **Summary**
  - ☆ **We derived a general inequality for evaluating variance in the maximum likelihood estimation of conditional probabilities (Theorem 1)**
  - ☆ **As an application of Theorem 1, we derived the convergence rate for logistic regression in non-sparse deep learning models (Theorem 2)**
  - ☆ **We demonstrated the minimax optimality of the rate derived in Theorem 2 (Theorem 3)**
  - ☆ **The paper is available in arXiv**
    URL: https://arxiv.org/abs/2401.12482

Appendix

1.2 Generalization error analysis in deep learning

- **Slow rates of convergence**
  - As an example, consider a binary classification problem:
    - $\mathcal{X} \subset \mathbb{R}^d$: Input space, $\mathcal{Y} = \{0, 1\}$: Output space
    - P: Probability distribution on $\mathcal{X} \times \mathcal{Y}$, $P_{\boldsymbol{X}}$: Marginal distribution of X
    - $(\boldsymbol{X}, Y), (\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n) \sim_{i.i.d.} P$
    - $\mathcal{D}_n := \{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n$: Observed data
  - **Goal: Construct a function $\hat{f}_n : \mathcal{X} \to \mathcal{Y}$ from $\mathcal{D}_n$ that predicts Y well when new data $\boldsymbol{X}$ is given**
    - $L_n = \mathbb{P}(\hat{f}_n(\boldsymbol{X}) \neq Y \mid \mathcal{D}_n)$: Misclassification rate (the probability of incorrect classification)
    - $L^* = \inf_{f:\text{measurable}} \mathbb{P}(f(\boldsymbol{X}) \neq Y)$: Optimal misclassification rate

1.2 Generalization error analysis in deep learning

- **Slow rates of convergence**

Theorem 0 (Slow rates of convergence, Devroye, 1982)

Let (a_n) be a sequence of positive numbers converging to zero such that $1/16 \geq a_1 \geq a_2 \geq \ldots$. For any learning method, there exists a distribution of (X, Y) satisfying $L^* = 0$ such that

$$\forall n \in \mathbb{N};\ \mathbb{E}[L_n] \geq a_n$$

- **No learning method is universally optimal; the performance of any method can deteriorate for some distributions.**

3.2 記号と仮定

- **記号**
  - $\bar{\mathcal{F}}_n^{1/2}(\tilde{\boldsymbol{p}}, \delta) = \left\{ \frac{\boldsymbol{p}+\tilde{\boldsymbol{p}}}{2} : R\left(\frac{\boldsymbol{p}+\tilde{\boldsymbol{p}}}{2}, \tilde{\boldsymbol{p}}\right) \leq \delta^2,\ \boldsymbol{p} \in \mathcal{F}_n \right\}$
  - $N_{p,B}(\delta, \mathcal{F}, Q)$:
    関数クラス $\mathcal{F}$ の $L^p(Q)$ ノルムに関する δ-bracketing number
    - 関数クラス $\mathcal{F}$ のある種の"大きさ"や"複雑度"を表す
  - $J_B(\delta, \bar{\mathcal{F}}_n^{1/2}(\tilde{\boldsymbol{p}}_n, \delta), \mu) = \int_{\delta^2/(2^{13}c_0)}^{\delta} \sqrt{\log N_{2,B}\left(u, \bar{\mathcal{F}}_n^{1/2}(\tilde{\boldsymbol{p}}_n, \delta), \mu\right)}\, du \vee \delta$
    - μ は $\{1, \ldots, K\}$ 上の数え上げ測度とP_Xの直積測度
    - $J_B(\delta, \bar{\mathcal{F}}_n^{1/2}(\tilde{\mathbf{p}}_n, \delta), \mu)$ は $\tilde{\boldsymbol{p}}_n$ の近くで見たときの局所的な複雑度

    ⇒ 局所的な複雑度を用いることで tight な不等式を導く

    (van de Geer, 2000; Bartlett et al., 2005)

3.3 条件付き確率のノンパラメトリック最尤推定

☆オラクル不等式

Theorem 1（オラクル不等式 確率ver.）

Kクラスの分類問題を考える. $\hat{\boldsymbol{p}}_n$をNPMLEとする.

$\Psi(\delta) \geq J_B(\delta, \bar{\mathcal{F}}_n^{1/2}(\tilde{\boldsymbol{p}}_n, \delta), \mu)$を$\Psi(\delta)/\delta^2$が$\delta$の非増加関数となるようにとる. δ_n とある普遍定数 c は

$$\sqrt{n}\delta_n^2 \geq c\Psi(\delta_n) \qquad (1)$$

を満たすとする. このとき

$$\mathbb{P}\left(R\left(\hat{\mathbf{p}}_n, \boldsymbol{\eta}\right) > 514(1+c_0^2)(\delta^2 + R\left(\tilde{\mathbf{p}}_n, \boldsymbol{\eta}\right))\right) \leq c\exp\left(-\frac{n\delta^2}{c^2}\right)$$

3.3 条件付き確率のノンパラメトリック最尤推定

☆オラクル不等式（推定量の汎化性能を評価）

Theorem 2（オラクル不等式 期待値ver.）

Kクラスの分類問題を考える. $\hat{\boldsymbol{p}}_n$をNPMLEとする.

$\Psi(\delta) \geq J_B(\delta, \bar{\mathcal{F}}_n^{1/2}(\tilde{\boldsymbol{p}}_n, \delta), \mu)$を$\Psi(\delta)/\delta^2$が$\delta$の非増加関数となるようにとる. このとき, ある普遍定数cと

$$\sqrt{n}\delta_n^2 \geq c\Psi(\delta_n) \qquad (1)$$

を満たすδ_n, 全ての$\delta \geq \delta_n$に対して以下が成り立つ.

$$\mathbb{E}\left[R\left(\hat{\mathbf{p}}_n, \boldsymbol{\eta}\right)\right] \leq 514(1+c_0^2)\left(\underbrace{\delta_n^2}_{\text{Variance}} + \underbrace{R\left(\tilde{\mathbf{p}}_n, \boldsymbol{\eta}\right)}_{\text{Bias}}\right) + \frac{c}{n},$$

- δがVarianceの評価に対応
- J_Bはモデル$\mathcal{F}_n$の"大きさ" = "複雑さ"を表す

⇒ モデルが複雑なほどVarianceは大きくなってしまう

3.3 条件付き確率のノンパラメトリック最尤推定

☆オラクル不等式（推定量の汎化性能を評価）

Theorem 2（オラクル不等式 期待値ver.）

Kクラスの分類問題を考える. $\hat{\boldsymbol{p}}_n$をNPMLEとする.

$\Psi(\delta) \geq J_B(\delta, \bar{\mathcal{F}}_n^{1/2}(\tilde{\boldsymbol{p}}_n, \delta), \mu)$を$\Psi(\delta)/\delta^2$が$\delta$の非増加関数となるようにとる. このとき, ある普遍定数cと

$$\sqrt{n}\delta_n^2 \geq c\Psi(\delta_n) \qquad (1)$$

を満たすδ_n, 全ての$\delta \geq \delta_n$に対して以下が成り立つ.

$$\mathbb{E}\left[R\left(\hat{\mathbf{p}}_n, \boldsymbol{\eta}\right)\right] \leq 514(1+c_0^2)\left(\underbrace{\delta_n^2}_{\text{Variance}} + \underbrace{R\left(\tilde{\mathbf{p}}_n, \boldsymbol{\eta}\right)}_{\text{Bias}}\right) + \frac{c}{n},$$

- モデルが複雑なほどVarianceは大きくなってしまう
- 一方, モデルが複雑なほどBias $R(\tilde{\boldsymbol{p}}_n, \boldsymbol{\eta})$は小さくなる

Bias-Variance トレードオフ

（モデルが複雑なほど真の条件付き確率を含みやすくなるため）

3.3 条件付き確率のノンパラメトリック最尤推定

☆オラクル不等式（推定量の汎化性能を評価）

Theorem 2（オラクル不等式 期待値ver.）

Kクラスの分類問題を考える. $\hat{\boldsymbol{p}}_n$をNPMLEとする.

$\Psi(\delta) \geq J_B(\delta, \bar{\mathcal{F}}_n^{1/2}(\tilde{\boldsymbol{p}}_n, \delta), \mu)$を$\Psi(\delta)/\delta^2$が$\delta$の非増加関数となるようにとる. このとき, ある普遍定数cと

$$\sqrt{n}\delta_n^2 \geq c\Psi(\delta_n) \qquad (1)$$

を満たすδ_n, 全ての$\delta \geq \delta_n$に対して以下が成り立つ.

$$\mathbb{E}\left[R\left(\hat{\mathbf{p}}_n, \boldsymbol{\eta}\right)\right] \leq 514(1+c_0^2)\left(\underbrace{\delta_n^2}_{\text{Variance}} + \underbrace{R\left(\tilde{\mathbf{p}}_n, \boldsymbol{\eta}\right)}_{\text{Bias}}\right) + \frac{c}{n},$$

- モデルの複雑さJ_BとBias $R\left(\tilde{\boldsymbol{p}}_n, \boldsymbol{\eta}\right)$の評価ができれば

  NPMLEの収束レートがわかる！

- Theorem 2は深層学習に限らず一般のNPMLEに対して成立