2020年度卒業論文
2020 Bachelor's Thesis

ネットワーク情報量基準を用いた深層ニューラルネットにおける汎化ギャップ推定の実験的考察

An Experimental Investigation of Estimating the Generalization Gap for Deep Neural Networks Using the Network Information Criterion

早稲田大学 先進理工学部
電気・情報生命工学科
情報学習システム研究室
Information Learning Systems Lab
Department of Electrical Engineering and Bioscience
School of Advanced Science and Engineering
Waseda University

1Y16F102　朝永主竜珠 (Sutashu Tomonaga)

April 16, 2025

Contents

1 Introduction **3**
- 1.1 Background . . . 3
- 1.2 Overview . . . 5

2 Mathematical Context **6**
- 2.1 Parametric Models and Learning Objectives . . . 6
- 2.2 Generalization and the NIC . . . 7
  - 2.2.1 The Generalization Gap . . . 7
  - 2.2.2 Curvature, Noise, and Information Matrices . . . 8
  - 2.2.3 The Network Information Criterion . . . 9

3 The Application of the NIC on Deep Neural Networks **11**
- 3.1 Past Measures . . . 11
- 3.2 An Alternative Measure . . . 11

4 Experiments **14**
- 4.1 Model Design . . . 14
- 4.2 Estimating the Generalization Gap Using the Gradient Covariance Matrix 14
  - 4.2.1 Estimation with a Single Threshold . . . 14
  - 4.2.2 Estimation with Varying Thresholds . . . 16

5 Conclusions and Future Work **18**

Chapter 1

Introduction

1.1 Background

Machine learning is the process by which a self-correcting computer algorithm modifies its parameters to approximate an unknown stochastic system based on a sample of "training" data observed from the system. The field of Machine Learning aims to find the best mathematical models and their applications that would best approximate the unknown stochastic system.

Artificial neural networks are a family of mathematical models that were designed to mimic the information processing mechanisms observed in animal brains. In recent years, these models have undergone tremendous development, especially in the form of Deep Neural Networks, an architecture of multi-layered Artificial Neural Networks. For example, the rise of Deep Neural Networks have had significant impacts on the fields of image classification[1], computer vision and speech recognition[2], natural language processing[3], and machine translation[4].

Deep Neural Networks have astronomical numbers –some ranging to the billions– which give them exceptional degrees of freedom and therefore the potential to approximate a broader range of possible systems. . However, from a statistical point of view, it has been long-established that when a model's number of parameters increases in proportion to the size of available training data, the system would have too many degrees of freedom and thus becomes overly biased to the training data, resulting in a decrease in accuracy to novel data. This phenomenon is known as "overfitting"[1]. Many empirical countermeasures were proposed, some of which have effectively mitigated this problem, such as augmenting new data from the accessible observed data to increase the training data or adding regularization restrictions to weight parameters during the learning of a Deep Neural Network, which effectively reduces the degrees of freedom[5]. It remains, however, as the complexity of these architectures increases, the more difficult it becomes to understand their behaviors from a mathematical and engineering perspective.

Besides the learning mechanisms of a model, another important subject in Machine Learning is model selection. It is desirable that a trained model exhibits sufficient accuracy for novel data as well as training data. When an artificial system performs well on training data and poorly on novel data, a model is said to have a low generalization

[1] Also known as overtraining

capability and has overfitted to the training data. A model's generalization capability is usually evaluated by comparing a model's loss [2] to training data (the "training loss") and its loss to novel data (the "generalization error"). However, since data obtained from an unknown stochastic system is finite, the limited available data must be divided into training data and "test" data to evaluate a model's generalization capability. Classically, techniques to find the most efficient ways to split data and iteratively resampling them as train/test data, such as numerous cross-validation and bootstrap techniques, have empirically proved to be a sufficient measure. However, in order utilize all available information obtained from the unknown stochastic system, it is ideal to use all available data as training data and evaluate its generalization capability without the need to allocate test data.

The problem of estimating the generalization capability without splitting available data for testing has been a hot topic in the field of Deep Learning. Among others, the Information Criteria are such measures capable of estimating the generalization gap from training data alone. Although they have not been successfully applied to models such as Deep Neural Networks, as Information Criteria degenerates under the conditions for these classes of models, Information Criteria are powerful tools in that they can be derived theoretically for many classes of models and maintain computational simplicity[6]. Furthermore, the Network Information Criterion has recently been tested, under certain computational improvisations, to have promising connections to the estimation of the generalization capability of a small-scale Deep Neural Network[7].

This paper attempts to explore other ways of computing the NIC for Deep Neural Networks, and understand its underlying mathematical properties. More specifically, while past research have focused on the spectrum of only the Hessian Matrix of the trained model to improvise computing the NIC, this paper attempts to explore other possibilities, namely utilizing the spectrum of the gradient covariance Matrix of the gradient vector of a trained model. Although a more superior way of improvisation has not been found, the results provide some insight into the interplays between the eigensapces of the two matrices and how both spectra are important in the calculation of the NIC.

[2] A loss is a measure of error of a model, or a measure of the discrepancy between a set of sample data and a model's output

1.2 Overview

The contents of this paper is as follows. First, as a preparation to the rational of the experiments, the mathematical background behind the objective of machine learning, the generalization of a model and the definition of the Network Information Criterion are discussed in Chapter 2. Then, Chapter 3 will describe the implications of applying the Network Information Criterion on Deep Neural Networks and counter measures to overcome them, and then introduces an alternative method for applying the Network Information Criterion on Neural Networks based on a hypothesis. The alternative measure and the hypothesis are tested in Chapter 4. Finally, it is concluded that the hypothesis may prove to be false, and future work should quantitatively investigate this falsification of the hypothesis.

Chapter 2

Mathematical Context

This chapter will focus on discussing the mathematical background behind the learning process in machine learning, the generalization of a model, and the definition and properties of the Network Information Criterion.

2.1 Parametric Models and Learning Objectives

In the context of optimization, the aim of Machine Learning is to find a set of parameters for a parametric model that minimizes the discrepancy between the parametric model and the unknown stochastic system (the "target system"). This is achieved by minimizing an energy function (or the loss function) representing the discrepancy between the two probabilistic distributions representing the parametric model and the unknown stochastic system.

Consider an unknown stochastic system that takes $\boldsymbol{x} \in \boldsymbol{R}^m$ as an input and outputs $\boldsymbol{y} \in \boldsymbol{R}^n$, and let $\boldsymbol{x}$ be a probabilistic variable generated by a distribution $q(\boldsymbol{x})$. If so, the output $\boldsymbol{y}$ can be expressed as a probabilistic variable conditioned by $\boldsymbol{x}$, subject to the distribution $q(\boldsymbol{y} \mid \boldsymbol{x})$, in which case the simultaneous distribution can be denoted as $q(\boldsymbol{y}, \boldsymbol{x}) = q(\boldsymbol{x})q(\boldsymbol{y} \mid \boldsymbol{x})$. In order to approximate the "target" unknown stochastic system, let us define a parametric model following the conditional distribution $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \boldsymbol{R}^m$. An example of such parameters $\boldsymbol{\theta}$ to a model, is the mean and variance to the Gaussian Distribution, which is a parametric model. Hereafter, the conditional distribution representing the target system $q(\boldsymbol{y} \mid \boldsymbol{x})$ will be referred to as the "target distribution", and the family of parametric models $\{p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}); \boldsymbol{\theta} \in \boldsymbol{R}^m\}$ will be referred to as the "model distribution". The objective is, given that a model is specified, to find the optimal set of parameters $\boldsymbol{\theta}$ to a model distribution that best approximates the target distribution, or in other words, minimizes the discrepancy between the two distributions.

Let us now define a discrepancy function between the model distribution and the

target distribution

$$\begin{aligned} D(q, p(\boldsymbol{\theta})) &\equiv \mathbb{E}_q \left[\ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})\right] \\ &= \int \ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) q(\boldsymbol{x}) q(\boldsymbol{y} \mid \boldsymbol{x}) d\boldsymbol{x} d\boldsymbol{y} \\ &= \int \ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) q(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \end{aligned} \tag{2.1}$$

where $\ell(\boldsymbol{x}, \boldsymbol{y})$ is the loss function, which is a measure of error for when a specified model $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})$ that takes an input $\boldsymbol{x}$ and emits an output $\boldsymbol{y}'$, measuring how different the output is to the true output $\boldsymbol{y}'$ emitted by the target system. There can be many definitions of a loss function, such as the mean square error:

$$\ell(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2} \int \|\boldsymbol{y} - \boldsymbol{y}'\|^2 p\left(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{\theta}\right) d\boldsymbol{y}' d\boldsymbol{y} d\boldsymbol{x} \tag{2.2}$$

or the negative log likelihood error:

$$\ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = -\log p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) \tag{2.3}$$

For simplicity, we will only be considering the negative log likelihood error in this paper.

The Discrepancy function $D(q, p(\boldsymbol{\theta}))$, therefore, computes the expectation of a loss for a specified model. Subsequently, minimizing the discrepancy between the model distribution and the target distribution is equivalent to minimizing the expectation of a loss function for a specified model. Hence the optimal set of parameters $\boldsymbol{\theta}_{opt} \in \boldsymbol{R}^m$ can be expressed as

$$\boldsymbol{\theta}_{opt} \in \arg\min_{\boldsymbol{\theta}} D(q, p(\boldsymbol{\theta})) \tag{2.4}$$

In reality, the target system is unknown, which makes it impossible to directly compute $D(q, p(\boldsymbol{\theta}))$. We instead only have access to the "empirical data", a finite set of data observed from the target system. The "empirical distribution" can be denoted as the average of N delta functions:

$$q^*(\boldsymbol{x}, \boldsymbol{y}) \equiv \frac{1}{N} \sum_{i=1}^{N} \delta\left(\boldsymbol{x} - \boldsymbol{x}_i, \boldsymbol{y} - \boldsymbol{y}_i\right) \tag{2.5}$$

where $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\,; i = 1, \ldots, N\}$ are the N example data observed from the target system.

Accordingly, a feasible objective is to find a set of quasi-optimal parameters $\boldsymbol{\theta}^* \in \boldsymbol{R}^m$ for a specified model that best approximates the target system, or minimizes the discrepancy function between the model distribution and the empirical distribution.

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}} D(q^*, p(\boldsymbol{\theta})) \tag{2.6}$$

2.2 Generalization and the NIC

2.2.1 The Generalization Gap

The generalization capability of a model is how well a model performs on novel data as opposed to training data. This may be evaluated by comparing the expected training

loss to the expected generalization error (the expected loss to novel data). When a model exhibits having a low expected training loss while having a large expected generalization error, it is said to be overfitted to the training data. The difference of the expected training loss and the expected generalization error is called the "generalization gap", defined as

$$\mathcal{G} \equiv D(q, p(\boldsymbol{\theta})) - D(q^*, p(\boldsymbol{\theta})) \tag{2.7}$$

In reality, since the number of data samples that may be obtained from a target distribution is finite, in addition to the fact that the target distribution is unknown, the true expected generalization error is unobtainable. Thus as a general practice, to compute the generalization error, a portion of the available data (the "empirical data") is allocated for testing, while the rest is used for training. After a model has been trained, the model is then "tested" by computing the expected loss to the "test data" q^*_{test}, which is used as an estimator for the generalization error, in which case an estimator for the generalization gap is obtained by

$$\hat{\mathcal{G}}^* \equiv D(q^*_{test},\ p(\boldsymbol{\theta})) - D(q^*_{train},\ p(\boldsymbol{\theta})) \tag{2.8}$$

Methods such as cross-validation and bootstrap fit this category of identifying a model's generalization capability.

2.2.2 Curvature, Noise, and Information Matrices

It is ideal for all available data to be used for training, especially when models have significantly larger degrees of freedom than available data for training, such as Deep Neural Networks.

This motivation has lead research to look into the geometric landscape of the loss function in the context of an optimization problem.

Past research have proposed estimators for the generalization gap using the information regarding the geometric landscape of the loss function. For instance, one estimator links the flatness of the loss function landscape at a local optimum, which is characterized by the eigenvalues of the Hessian, to the generalization gap[8] [9]. Another estimator links to the generalization gap the sensitivity of the gradient function to noise, by evaluating the Jacobian formed by taking the derivative of the output vector with respect to the input[10]. However, it has been shown that while these estimators are reliable under certain conditions, their realizability do not hold in general, because they do not take into account all information obtainable from the geometric landscape of the loss function.

One way that these information may be obtained is by calculating the "information matrices"; the Hessian H, the Fisher F, and the gradient covariance matrix C of a loss function, defined as

$$\begin{aligned} C(\boldsymbol{\theta}) &= \mathbb{E}_q \left[\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})^{\mathrm{T}}\right] \\ H(\boldsymbol{\theta}) &= \mathbb{E}_q \left[\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})\right] \\ \mathrm{F}(\boldsymbol{\theta}) &= \mathbb{E}_p \left[\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})\right] \\ &= \mathbb{E}_p \left[\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})^{\mathrm{T}}\right] \end{aligned} \tag{2.9}$$

where $\mathbb{E}_q[\cdot], \mathbb{E}_p[\cdot]$ denotes the expectation subject to the distributions q and p, respectively.[1]

If there exists a model which contains the target system,

$$q(\boldsymbol{y} \mid \boldsymbol{x}) \in \{p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}); \boldsymbol{\theta} \in \boldsymbol{R}^m\} \tag{2.10}$$

or in other words, if there exists $\boldsymbol{\theta}_{\text{opt}}$ and $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}_{\text{opt}})$ such that

$$q(\boldsymbol{y} \mid \boldsymbol{x}) = p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}_{\text{opt}}) \tag{2.11}$$

then the target system is said to be "realizable", and the model is said to be "well-specified" or "faithful". If so, by definition, $H(\boldsymbol{\theta}_{\text{opt}}) = C(\boldsymbol{\theta}_{\text{opt}}) = \mathrm{F}(\boldsymbol{\theta}_{\text{opt}})$. Moreover, when $N \to \infty$,

$$\begin{aligned} H(\boldsymbol{\theta})_{\text{opt}} &= H^*(\boldsymbol{\theta}) \\ C(\boldsymbol{\theta})_{\text{opt}} &= C^*(\boldsymbol{\theta}) \end{aligned} \tag{2.12}$$

where

$$\begin{aligned} C^*(\boldsymbol{\theta}^*) &= \mathbb{E}_{q^*}\left[\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})^{\mathrm{T}}\right] \\ H^*(\boldsymbol{\theta}^*) &= \mathbb{E}_{q^*}\left[\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})\right] \end{aligned} \tag{2.13}$$

However, it is not often the case that a target system is realizable, so there is usually a discrepancy between $q(\boldsymbol{y} \mid \boldsymbol{x})$ and $q^*(\boldsymbol{y} \mid \boldsymbol{x})$. Also, the number of data samples may be limited, so the empirical Hessian and the empirical gradient covariance matrix diverges from the true Hessian and the true gradient covariance matrix in most cases.

2.2.3 The Network Information Criterion

Information Criteria are unbiased estimators of the generalization gap. The simplest of its kind is the AIC[11], which is the number of parameters divided by the number of samples.

$$\hat{\mathcal{G}}_{AIC} = \frac{1}{N} d \tag{2.14}$$

However, this estimator only functions when the unknown stochastic system is realizable. The theory was later expanded as the TIC [12], and then as the NIC[6].

$$\hat{\mathcal{G}}_{NIC} = \frac{1}{N} \operatorname{tr}\left(H(\boldsymbol{\theta})^{-1} C(\boldsymbol{\theta})\right) \tag{2.15}$$

In practice, we only have access to the true distribution, so the NIC may be denoted as

$$\hat{\mathcal{G}}_{NIC} = \frac{1}{N} \operatorname{tr}\left(H^*(\boldsymbol{\theta})^{-1} C(\boldsymbol{\theta}^*)\right) \tag{2.16}$$

[1] Note that the calculation of $\mathbb{E}_q[\cdot]$ can stand for both q and q^*

NIC and the Information Matrices

As mentioned earlier, when a model is well specified, the true Hessian matrix and the true gradient covariance matrix converge to the true Fisher information matrix, and when the data size is large enough, the empirical Hessian matrix and the gradient covariance matrix converge to the true Hessian matrix and the true gradient covariance matrix, respectively. So when both conditions are met, the empirical Hessian matrix and the empirical gradient covariance matrix converfe to the true Fisher information matrix. Therefore it can be said that the true and empirical Hessian matrix and the true and empirical gradient covariance matrix are approximations to the true Fisher information matrix. In this case, the NIC converges to the AIC.

$$\begin{aligned}\hat{\mathcal{G}}_{NIC} &= \frac{1}{N}\mathrm{tr}(H^{*-1}C^{*}) \\ &= \frac{1}{N}\mathrm{tr}(H_{\mathrm{opt}}^{-1}C_{\mathrm{opt}}) \\ &= \frac{1}{N}\mathrm{tr}(F_{\mathrm{opt}}^{-1}F_{\mathrm{opt}}) \\ &= \frac{1}{N}d \\ &= \hat{\mathcal{G}}_{AIC}\end{aligned} \tag{2.17}$$

This means all necessary information for estimating the generalization gap may be contained in the diagonal elements of the two matrices, and the non diagonal elements work to rotate the matrices to match the corresponding diagonal elements representing the eigenspaces of the matrices. This can be shown by eigendecomposing the gradient covariance matrix and conducting simple matrix operations as shown below.

$$\begin{aligned}\mathrm{tr}\left(H^{-1}C\right) &= \mathrm{tr}\left(H^{-1}\left(R_C D_C R_C^{-1}\right)\right) \\ &= \mathrm{tr}\left(H^{-1}R_{\mathrm{C}}D_{\mathrm{C}}R_{\mathrm{C}}^{-1}\right) \\ &= \mathrm{tr}\left(R_{\mathrm{C}}^{-1}H^{-1}R_{\mathrm{C}}D_{\mathrm{C}}\right) \\ &= \mathrm{tr}\left(\left(R_{\mathrm{C}}HR_{\mathrm{C}}^{-1}\right)^{-1}D_{\mathrm{C}}\right) \\ &= \mathrm{tr}\left(H_{\mathrm{C}}^{-1}D_C\right)\end{aligned} \tag{2.18}$$

Note that this operation can be done both ways, from C to H as well as H to C.

Chapter 3

The Application of the NIC on Deep Neural Networks

This chapter discusses the implications of applying the NIC on Deep Neural Networks, proposed measures to overcome them, and a possible alternative to this measure based on a reasonable hypothesis, which may turn out to be false.

3.1 Past Measures

The Problem with Applying the NIC on Neural Networks

Since the NIC requires the computation of the inverse of the Hessian matrix, a problem arises as the Hessian matrix of Neural Networks quickly degenerate. Normally, this problem is mitigated by which reduces the number of parameters in a model, reducing the degrees of freedom, which nullifies the degenerate eigenspaces.. But for Deep Neural Networks this is not an option because the number of parameters are intentionally set to large scales.

NIC and the spectrum of H^*

In order to overcome this problem in Deep Neural Networks, past research focused on the spectrum of the Hessian matrix[7], and found that by selecting a threshold and negating eigenspaces lower than the threshold in the computation of the inverse Hessian matrix, the generalization gap can be effectively estimated. In further research it was discussed that it matters where the threshold should be placed in the spectrum of the Hessian matrix for estimating the generalization gap.[13]

3.2 An Alternative Measure

NIC and the spectrum of C^*

In 2.2.3 it was mentioned that the non diagonal elements work to rotate the matrices in order to match the corresponding matrices. This is crucial for the NIC to function

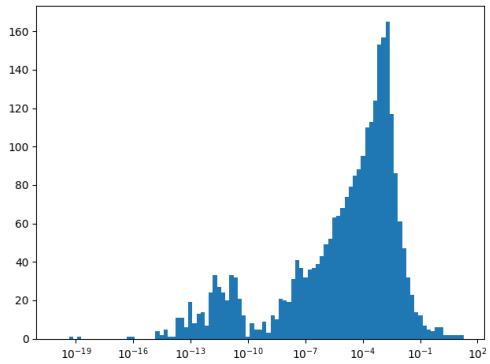


Figure 3.1: Histogram of the eigenvalues of the Hessian matrix

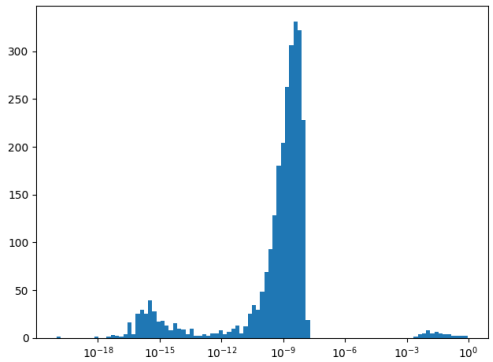


Figure 3.2: Histogram of the eigenvalues of the gradient covariance matrix

as an estimator, because if the eigenspaces do not correspond properly, then the degenerate eigenspaces may not correspond, which may cause an extreme overestimation or underestimation of the generalization gap. Therefore, it should not matter whether the effective eigenspaces to calculate the NIC is selected based on the spectrum of the Hessian matrix or the gradient covariance matrix. Furthermore, as can be seen from Figure 3.2, that the spectrum of C has a clear division between the degenerated eigenspaces and the remaining eigenspaces, so the spectrum of C may be a better basis for selection as it naturally provides a clearer threshold whereas a threshold had to be carefully set with the spectrum of the Hessian matrix.

Selecting a Subspace of H with the spectrum of C

The eigenspaces of the Hessian matrix corresponding to the selected eigenspaces of the gradient covariance matrx can easily be found by eigendecomposing the gradient covariance matrix, nullifying the eigenspaces with eigenvalues smaller than the threshold, and rotating the Hessian matrix with the remaining subspaces. The operations can be

denoted as

$$\begin{aligned}
\operatorname{tr}\left(H^{-1}C\right) &= \operatorname{tr}\left(H^{-1}\left(R_C D_C R_C^{-1}\right)\right) \\
&= \operatorname{tr}\left(H^{-1}\left(R\tilde{I_C} D_C \tilde{I_C}^{-1} R^{-1}\right)\right) \\
&= \operatorname{tr}\left(H^{-1}\left(R\tilde{I_C}\right) D_C \left(\tilde{I_C}^{-1} R^{-1}\right)\right) \\
&= \operatorname{tr}\left(H^{-1}\tilde{R}_C D_C \tilde{R}_C^{-1}\right) \\
&= \operatorname{tr}\left(\tilde{R}_C^{-1} H^{-1} \tilde{R}_C D_C\right) \\
&= \operatorname{tr}\left(\left(\tilde{R}_C H \tilde{R}_C^{-1}\right)^{-1} D_C\right) \\
&= \operatorname{tr}\left(\tilde{H}_C^{-1} D_C\right)
\end{aligned} \tag{3.1}$$

where

$$\tilde{I_C} \equiv \left(\begin{array}{ccc|c} 1 & & 0 & \\ & \ddots & & 0 \\ 0 & & 1 & \\ \hline & 0 & & 0 \end{array}\right) \begin{array}{l} \text{rank } C \\ \\ \dim C - \text{rank } C \end{array} \tag{3.2}$$

Chapter 4

Experiments

4.1 Model Design

For the following experiments, a total of 108 models trained based on the architectures and conditions described below were used. To cut down computational cost, all input images were resized to 7×7 pixels and converted to greyscale, as was done in preceding research.

- 3 architectures: a 1-hidden layer fully connected network, a 2-hidden layer fully connected network, and a convolutional neural network.
- 3 datasets: MNIST, CIFAR-10, and SVHN
- 3 learning rates of SGD with momentum $\mu = 0.9$: $10^{-2}, 5 \cdot 10^{-3}$, and 10^{-3}
- 2 batch sizes: 64,512
- 2 dataset sizes: 5k, 10k

All models were trained for 750k steps.

To calculate the empirical Hessian H^* and the empirical gradient covariance matrix C^* of a model, derivatives were taken per mini batches of losses, and averaged over them, in order to cut down on computational cost.

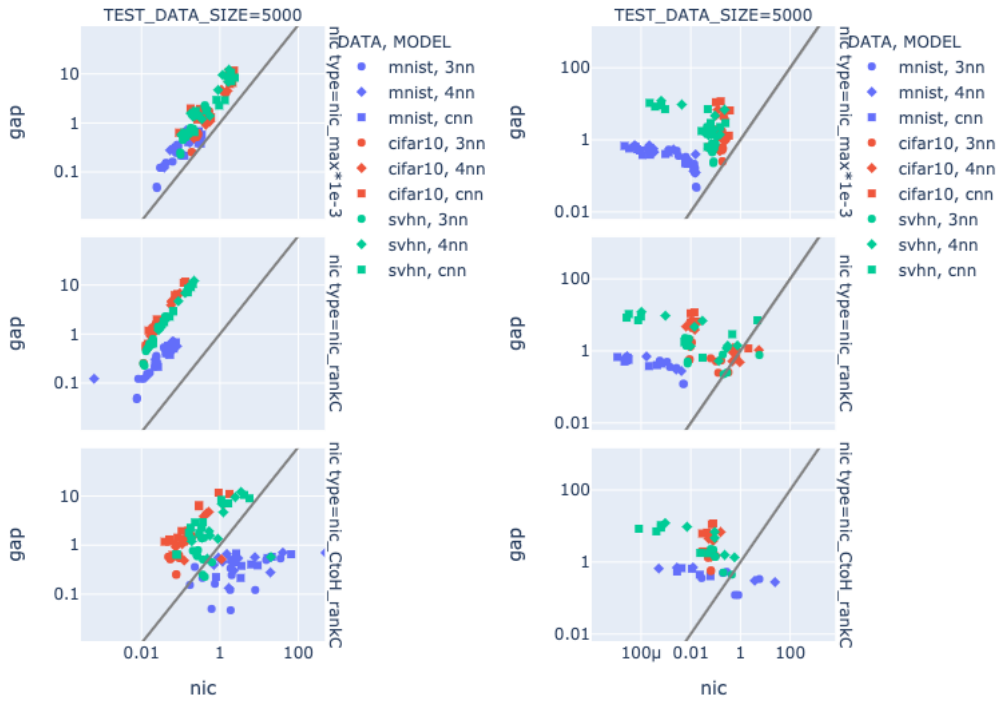
4.2 Estimating the Generalization Gap Using the Gradient Covariance Matrix

4.2.1 Estimation with a Single Threshold

The following experiment aimed to test the hypothesis that the generalization error may be estimated by setting the effective eigenspsaces of the Hessian matrix by a threshold set in the spectrum of the gradient coefficient matrix, and that this rule may be a more superior method for estimating the generalization gap.

The empirical Hessian and the empirical gradient covariance matrix were calculated over mini batches of size 100, for train data and test data. The selection of effective eigenspaces were done accordingly as shown in (3.1).

Results are shown in figure 4.1. The first row represents the NIC calculated following the rule previously proposed which was mentioned in 3.1, and the third row represents the NIC calculated by the rule proposed in this paper, mentioned in 3.2. And for comparison, the second row represents the NIC calculated following the same rule as the first rule, but with the same dimensions of eigenspaces as the third rule.



(a) test data

(b) train data

Figure 4.1: Comparing the accuracy of the NIC calculated according to previous research with effective eigenspace transferring from C to H

As can be seen from figure figure 4.1, the NIC calculated using the spectrum of the gradient covariance matrix adds variance to the estimation in the horizontal direction. Although this shows that the gradient covariance basis of eigenspace selection does not have superior estimation capability, the fact that the variance is added horizontally gives insight into the correspondence of the eigenspaces of the Hessian matrix and the gradient covariance matrix. If the NIC overestimates, small eigenvalues of the Hessian which were not meant to have effect are being selected by the effective eigenspaces of the gradient covariance matrix, and if the NIC underestimates, then the large eigenvalues of the Hessian matrix which were supposed to have effect in estimating the generalization gap have been ignored, because it was not selected by the effective eigenspaces of the gradient covariance matrix. This would imply that the degenerate eigenspaces of the two matrices do not correspond. This is investigated further in the following experiment.

4.2.2 Estimation with Varying Thresholds

Here we estimate the generalization gap with the same conditions as above, but with multiple thresholds. By comparing the estimation with different thresholds in the spectrum of the empirical gradient covariance matrix, we can observe the correspondence of the eigenspaces of the two matrices.

To this end, the large side of the spectrum of the gradient covariance matrix (the small lump on the right side of the spectrum in Figure 3.2) was divided in to 5 sections equi-proportionately in log scale, and the bottom of each section was designated to be the threshold for selecting the effective eigenspaces to compute the inverse Hessian matrix, calculated according to (3.1). Results are shown in Figure 4.2 and Figure 4.3.

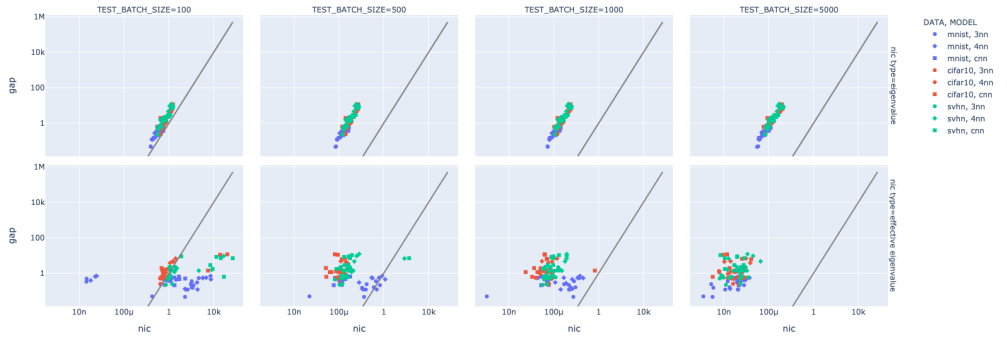


Figure 4.2: Estimating the generalization gap with test data using the spectrum of C and varying thresholds

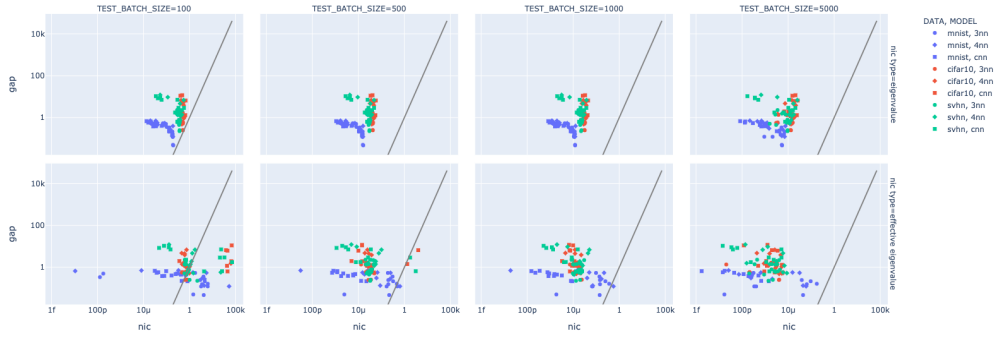


Figure 4.3: Estimating the generalization gap with test data using the spectrum of C and varying thresholds

As can be seen from Figure 4.2 and Figure 4.3, the NIC tends to overestimate the generalization gap when the threshold is set at the smaller end of the spectrum of the gradient coavariance matrix, and underestimate when the threshold is set at the larger end.

As mentioned above, it can be said that, when the NIC overestimates by a large margin, the eigenspace of the Hessian matrix with an extremely small eigenvalue is inverted and multiplied with a corresponding eigenvalue of the gradient covariance matrix that is not small enough to cancel out. In this case the product of the two elements becomes significantly large and causes the estimation to overshoot. Conversely, when the NIC underestimates the generalization gap, it can be said that the eigenspace of the Hessian matrix with a large eigenvalue is multiplied with a corresponding eigenvalue of the gradient covariance matrix that is small, which nullifies the impact of that eigenspace to the estimation of the generalization gap, and causes underestimation.

The hypothesis was that the eigenvalues of both the Hessian matrix and the gradient covariance matrix should correspond in their relative magnitudes, so that the outliers of the eigenvalues (the degenerate eigenspaces) may cancel out each others effect. However this seems to not have been the case based on the results.

Chapter 5

Conclusions and Future Work

The NIC as an estimator for the generalization gap is a powerful tool in that it makes possible to estimate a model's generalization capability only with training data, and has a strong mathematical background. However, it was known to have implications when applied to large scale Deep Neural Networks, as the Hessian matrix quickly degenerates in this context. Past research have empirically shown that the NIC can be applied to Neural Network by setting a threshold and cutting the degenerated eigenspaces of the Hessian matrix and calculating the inverse Hessian matrix with the remaining eigenspaces.

In this paper, it was discussed that since the degenerated space may be shared between the Hessian matrix and the gradient coefficient matrix, it should not matter with which spectrum to use as the selection bases for the eigenspaces to be effective in the calculation of the NIC. Furthermore, since the spectrum gradient coefficient matrix has a clear distinction between the degenerate and non degenerate eigenspaces, it may function as a better way to select the eigenspaces to be effective in the calculation of the NIC. Conversely, it was shown that the spectrum of the gradient coefficient matrix does not function as a good selection tool for the eigenspaces to be effective in the calculation of the NIC. This result negates the presupposition that the Hessian matrix and the gradient coefficient matrix share the same degenerate spaces. Additional experiments seem to support this suggestion. Given that we know that the Hessian matrix and the gradient coefficient matrix are both an approximation to the true Fisher information matrix, if what the results imply are true, it may give a new understanding on how the Hessian matrix and the gradient coefficient matrix diverge from the true Fisher matrix.

Acknowledgements

I would like to express my most profound appreciation to my supervisor, Prof. Noboru Murata, who has guided me through my first years of my academic life. In this past year I have grown immensely as a newly born researcher, all thanks to his kind support. He will continue to serve as a role model for me in the future as I continue to pursue my academic career and beyond.

I am additionally inclined to extend deep gratitude to my senior mentors. They have selflessly devoted their time to assist me this year as my mentors, and I am in much debt. Furthermore, I give thanks to all my seniors who have dedicated their time to help me in my research, as well as my colleagues for their overwhelming positivity and cheerfulness.

Finally, I want to give thanks to my family, who continue to provide loving support and care for my physical and mental well-being.

Thank you.

Sincerely
Sutashu Tomonaga

References

[1] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep Learning for Generic Object Detection: A Survey. *International Journal of Computer Vision*, Vol. 128, No. 2, pp. 261–318, February 2020.

[2] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, Vol. 234, pp. 11–26, April 2017.

[3] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. *IEEE Computational Intelligence Magazine*, Vol. 13, No. 3, pp. 55–75, August 2018.

[4] Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. Transforming machine translation: A deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, Vol. 11, No. 1, p. 4381, September 2020.

[5] Atsushi SATO. Deep Learning Technology for Small Data. *NEC Technical Journal*.

[6] Noboru Murata, Shuji Yoshizawa, and Shun-ichi Amari. Network Information Criterion — Determining the Number of Hidden Units for an Arti cial Neural Network Model 3. p. 19.

[7] Valentin Thomas, Fabian Pedregosa, Bart van Merriënboer, Pierre-Antoine Mangazol, Yoshua Bengio, and Nicolas Le Roux. Information matrices and generalization. *arXiv:1906.07774 [cs, stat]*, June 2019.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Flat Minima. *Neural Computation*, Vol. 9, No. 1, pp. 1–42, January 1997.

[9] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *arXiv:1609.04836 [cs, math]*, February 2017.

[10] Roman Novak, Yasaman Bahri, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and Generalization in Neural Networks: An Empirical Study. *arXiv:1802.08760 [cs, stat]*, June 2018.

[11] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, Vol. 19, No. 6, pp. 716–723, December 1974.

[12] Kei Takeuchi. Distribution of informational statistics and a criterion of model fitting. *Mathematical Sciences*, No. 153, pp. 12–18, 1976.

[13] H. Okado. 2019 年度修士論文ニューラルネットにおけるネットワーク情報量基準の実験的解析, 2019.