DEPARTMENT OF COMPUTER SCIENCE

Understanding the learning dynamics of Deep Neural Networkssubtitle

Seunghyeon Choi

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Bachelor of Science in the Faculty of Engineering.

Monday 9th May, 2022

Abstract

This work provides an overview of long history of information theoretic view of Deep learning which is now available to apply the real world data sets. In the first section, I will explain the background of both Deep Learning and Information Theory, mainly Information Bottleneck method. Then describe what I have implemented and in Chapter 4, evaluate what I did and what could have been done. In the end, conclude it with looking at this project a broader perspective.

- I spent 160 hours collecting material on and learning about the Deep Neural Network and Information Theory.
- I implemented a version of the Deep Neural Network.

Dedication and Acknowledgements

A compulsory section

Thanks for all my friends caring for me from the nearest location and family keeps supporting me until the end of this journey, My supervisor holds me through taking my back and gives adequate advice if I am in need.

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Taught Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, this work is my own work. Work done in collaboration with, or with the assistance of others, is indicated as such. I have identified all material in this dissertation which is not my own work through appropriate referencing and acknowledgement. Where I have quoted or otherwise incorporated material which is the work of others, I have included the source in the references. Any views expressed in the dissertation, other than referenced material, are those of the author.

Seunghyeon Choi, Monday 9th May, 2022

Contents

List of Figures

List of Tables

Ethics Statement

This project did not require ethical review.

Supporting Technologies

- The implementation has been made in Google Colab environment.
- I used the Python "pytorch" library to implement the deep learning model. PyTorch [54], Python's Machine Learning library. It was used for the implementation of the deep learning model. The implemented code for loading, sampling, training and evaluating data was dependent on this library.
- I used the Python library matplotlib to visualise results.

Notation and Acronyms

An optional section

CN	:	Computational Neuroscience
ML	:	Machine Learning
DL	:	Deep Learning
ANN	:	Artificial Neural Network
DNN	:	Deep Neural Network
IB	:	Information Bottleneck
SGD	:	Stochastic Gradient Descent
DPI	:	Data Processing Inequality
ERM	:	Empirical Error Minimization

Chapter 1

Introduction

This paper have investigated the deep learning dynamics of Deep Neural Network(DNN)s applied by the Information theory.

Understanding how the brain works is one of the most mysterious research fields. One approach I am interested in is a computational view of its function, namely Computational Neuroscience(CN), and it assumes the brain is an information processing machine and leans more mathematical modelling of the brain. Machine Learning(ML), which is trending these days, emerges from CN focusing more on the brain's learning process from the surrounding environments. Deep Learning (DL) emerges from ML. Artificial Neural Network (ANN) Deep Neural Network (DNN)

Information theory originally formulated by Shannon by "A Mathematical Theory of Communication"[9] and in early stage of diverging to outside of its original scope, there were several works that provide the foundation of its application in CN called "Neural Information Flow" [6]. There is a series of papers on about explaining the DL in a view of the information theory by Tishby[11] and related researchs on them[8]. It is considered that the framework is somewhat defined however need more generalize it in order to apply this framework to the real world data-sets. In need of that, this project is hoping to produce a principled and transferable description of the DL dynamics of DNNs.

> The high-level objective of this project is to reduce the gap between theoretical and computational implementations of how information relationships develop in the network. More specifically, the concrete aims are:
>
> 1. Research on Information bottleneck method and Deep Neural Networks.
> 2. Produce a principled and transferable description of the deep learning dynamics of DNNs.
> 3. Implement a framework for information theoretic point of view of deep linear networks.
> 4. Use the framework to visualise how information relationships develop in the network.

Chapter 2

Literature Review

I first review literature relevant to this project. There are 3 papers that became a starting point of this work[10] [12] [4]. See to what extent each of paper presents about the relationship between the information theory and deep learning. Those are narrowing down the depth of focus from the DL to the layer level of the DNNs.

2.1 Background

2.1.1 Deep Learning

Deep learning is inspired by the biological modeling of brain, and it has been generalized as a learning principle of multiple levels of composition that can be applied in machine learning frameworks, which are not necessarily neurally inspired. It has been increasing the amount of available training data and solved increasingly complicated applications with sufficient amount of accuracy.[3]

It discovers the sophisticated structure in large data sets to go through multiple processing layers of computational model to learn representations of data with multiple levels of abstraction. The methodology for doing that is the backpropagation algorithm indicating how a machine should update its parameters that are used to compute the representation in each layer from the representation in the previous layer.

It has beaten other machine-learning techniques at predicting the activity of potential drug molecules, analysing particle accelerator data, reconstructing brain circuits, and predicting the effects of mutations in non-coding DNA on gene expression and disease. Perhaps more surprisingly, deep learning has produced extremely promising results for various tasks in natural language understanding, particularly topic classification, sentiment analysis[13] and language translation.[5]

Structure Multiple Hidden Layers

2.1.2 Information Theory

The Information theory was originally formulated from solving the fundamental problem of formalizing the intuitive ideas about "meaningful" or "relevant" information.

Relevant Information

$$I(X;Y) = D_{KL}[p(x,y)||p(x)p(y)] = \sum_{x\in X, y\in Y} p(x,y)\log(\frac{p(x,y)}{p(x)p(y)}) \tag{2.1}$$

where D_{KL} is asymmetric.

2.1.3 Information Theory of DL

It is formalized that find a short code for A that preserves the maximum information about B. The process is described as squeezing the information that X provides about Y through a 'bottleneck' forming a limited set of codewords T. The degree of one signal $A \in \mathcal{A}$ is being the information to another signal

$b \in \mathcal{B}$ is defined as the mutual information $I(A;B)$. The relevant part of A which respect to B, denoted by $\hat{A}$, is a minimal sufficient statistics of x with respect to y. [11]

In supervised learning we are interested in good representations, $T(X)$, of the input patterns $x \in X$, that enable good predictions of the label $y \in Y$. Given any two random variables, X and Y , with a joint distribution $p(x, y)$, their Mutual Information is defined as: $I(X;Y)$ The mutual information quantifies the number of relevant bits that the input variable X contains about the label Y , on average. y implicitly determines the relevant and irrelevant features in x.[10]

Markov Chain

$$Y \to X \to \hat{X} \to \hat{Y}. \tag{2.2}$$

The DNN layers form a Markov chain of successive internal representations of the input layer X. Any representation of the input, T, is defined through an encoder, $P(T|X)$, and a decoder $P(\hat{Y}|T)$, and can be quantified by its *information plane* coordinates: $I_X = I(X;T)$ and $I_Y = I(T;Y)$. Any DNN can be quantified and this representation can be used for calculating the optimal information theoretic limits of the DNN and obtain finite sample generalization bounds. The idea of studying the Markov chain of layered neural networks in the Information Plane is suggested[12].

Data Processing Inequality (DPI)

$$I(X;Y) \geq I(T_1;Y) \geq I(T_2;Y) \geq \cdots \geq I(T_k;Y) \geq I(\hat{Y};Y) \tag{2.3}$$

SGD mean parameter - phase epoch matches [10]

The Information Bottleneck bound characterizes the optimal representations, which maximally compress the input X, for a given mutual information on the desired output Y . After training, the network receives an input X, and successively processes it through the layers, which form a Markov chain, to the predicted output $\hat{Y}$. $I(Y;\hat{Y})/I(X;Y)$ quantifies how much of the relevant information is captured by the network. At the beginning of the optimization the deeper layers of the randomly-initialize network fail to preserve the relevant information, and there is a sharp decrease in $I_Y = I(T_i;Y)$ along the path.
= focus on X (fast ERM phase)

Training

Optimization Process (minimal sufficient statistics) (Lagrangian) In a way of solving a variational problem, a Lagrange multiplier β has introduced to find the rate distortion function.[11] The variational principle implies that

$$\frac{\delta I(X,\hat{Y})}{\delta I(X,\hat{X})} = \beta^{-1} > 0 \tag{2.4}$$

$$\mathcal{F} \tag{2.5}$$

$$I(X;\hat{X}) - \beta I(\hat{X};Y) \tag{2.6}$$

During the SGD optimization, the layers first increase I_Y along the path, then include Y and lose irrelevant info about X, reducing the empirical error. later significantly decrease I_X thus compressing the representation.
= + until convergence (compression phase) - resulted as a general property of SGD training of DNNs related to overfitting - what determines the convergence points of the layers = 'decrease I_X = adding random noise to the weights = Fokker-Planck equation, whose stationary distribution maximizes the entropy of the weights distribution, under the training error constraint(= I_Y). The layers of the different randomized networks seem to follow very similar paths(network-path) during the optimization and eventually converge to nearby points in the information plane.
= This isn't for specific network. not only The dynamics and but also resulting points are similar * decrease I_X = adding random noise to the weights = increase $H(X|T_i)$ = minimize mutual information $I(X;T_i) = H(X) - H(X|T_i)$ ($H(X)$ doesn't change) [10] DualIB adds the predicted label, $\hat{Y}$ from the trained representation[8].

2.2 Deep learning and the information bottleneck principle

2.3 Opening the black box of Deep Neural Networks via Information

[10] Following research for Deep learning and the IB principle(2015), Demonstrate the effectiveness of the Information Plane visualization of DNNs and interprete the representation in terms of the learning phase of DNNs. the ability of DNNs to generalize can be seen as a type of representation compression.

2.4 Dual Information Bottleneck 2020

1. The IB is completely non-parametric and it operates only on the probability space
2. The IB formulation does not relate to the task of prediction over unseen patterns and assumes full access to the joint probability of the patterns and labels.

Focus more on the prediction over unseen patterns[8].

The distortion function:

$$d_{IB}(x, \hat{x}) = D[p(y|x)||p_\beta(y|\hat{x})] \tag{2.7}$$

As the decoder defines the prediction stage ($p_\beta(y = \hat{y}|\hat{x})$)

$$_{p_\beta(x,\hat{x})}[d_{dualIB}(x, \hat{x})] = \underbrace{I(\hat{X}; \hat{Y}) - I(X; \hat{Y})}_{\text{(a)}} + \underbrace{_{p_\beta(x)}[D[p_\beta(\hat{y}|x)||p(y = \hat{y}|x)]]}_{\text{(b)}}. \tag{2.8}$$

The *VdualIB* objective is given by:

$$\min_{q(\hat{x}|y),p(\hat{x}|x)} {}_{\tilde{p}(y|x)p(\hat{x}|x)p(x)} \left[\log \frac{p(\hat{x}|x)}{q(\hat{x}|y)}\right] + \beta_{p(y|\hat{x})p(\hat{x}|x)} \left[\log \frac{p(y|\hat{x})}{\tilde{p}(y|x)}\right], \tag{2.9}$$

2.5 Deep variational information bottleneck (2016)

It parameterises the information bottleneck model using a neural network and opens up the possibility of its application in unsupervised learning. L2 optimisation Tested on MNIST dataset and shows different result depending on the Possibly putting the VIB objective at multiple or every layer of a network [2]

2.6 Critical slowing Down near Topological Transitions in Rate-Distortion problems(2021)

RD problems are looking for reduced representations of a source that meets a target distortion constraint. The convergence time is slowed down near those critical points It take a deeper look at the background knowledge of rate distortion problems [1]

2.7 Adaptive estimators show information compression in deep neural netowrks

It focuses on the fact the generalisation of DNN can be seen as a type of representation compression and the quantified relevance of information, considering the hidden layers of the network as an intermediate representation T. The optimal value for T is the relevant part of X,which is minimized by the information bottleneck, represented as the Lagrangian : [4]

Chapter 3

Project Execution

3.1 Implementation of Deep Neural Network

Build a network based on pytorch library pass MNIST data set, given by the library, through the network. I compare loss and mutual information between the label Y and the output predicted value $\hat{Y}$. It was based on pytorch library.

3.1.1 Fully connected Feed Forward Artificial Neural Network

A neural network with single hidden layer. The version I implemented has linear connections from the input layer to the hidden layer and from the hidden layer to the output layer and the activation function is chosen by ReLu function. The parameters of the network is updated by the backpropagation algorithm.

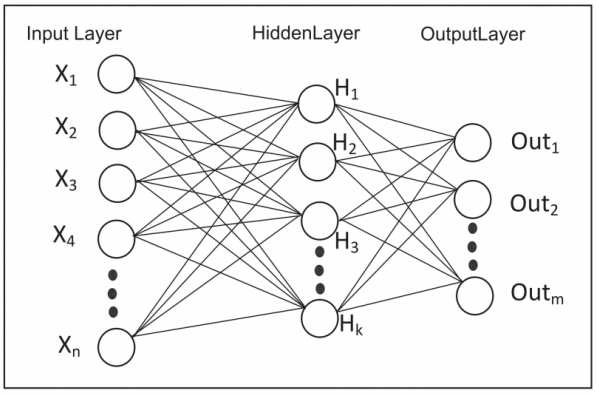

Figure 3.1: [7] The number of neurons in the network is denoted as n, k, m in (fig.3.1), The input layer. n, is 784 which is the size of input pixels $28 * 28$, the number of neurons in the hidden layer, k, is initially set to 100 and m is 10 which ranges from 0 to 9.

```
class NeuralNet(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self).__init__()
        self.input_size = input_size
        self.l1 = nn.Linear(input_size, hidden_size)
        self.relu = nn.ReLU()
        self.l2 = nn.Linear(hidden_size, num_classes)

    def forward(self, x):
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        return out
```

Listing 3.1: The first feed forward model.

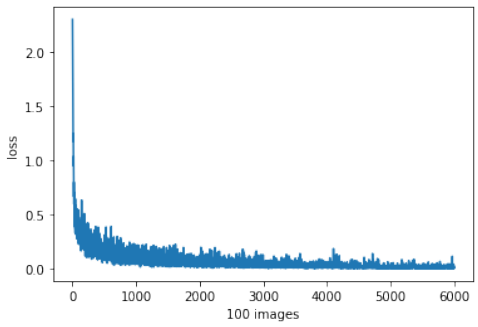

Figure 3.2: CrossEntropyLoss per 100 images.

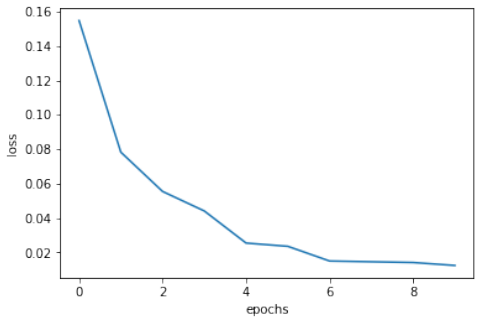

Figure 3.3: CrossEntropyLoss per epoch.

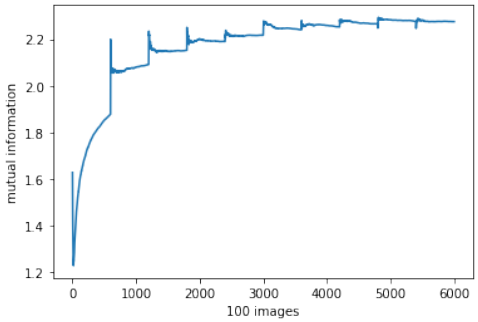

Figure 3.4: The peaks appear because of the joint probability of the label and the predicted label is too low, very close to 0, the mutual information returns infinity. It looks like converge to a value in the end of epoch.

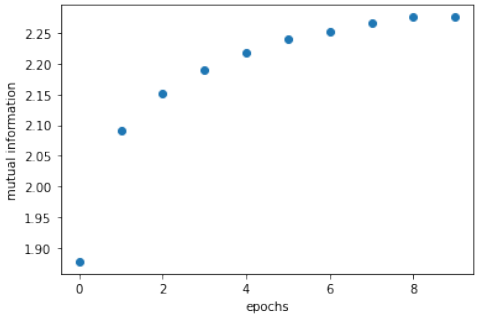

Figure 3.5: The mutual information calculated at the end of every epoch. It seems to grow logarithmically.

Chapter 4

Critical Evaluation

A topic-specific chapter

This chapter is intended to evaluate what you did. The content is highly topic-specific, but for many projects will have flavours of the following:

1. functional testing, including analysis and explanation of failure cases,
2. behavioural testing, often including analysis of any results that draw some form of conclusion wrt. the aims and objectives, and
3. evaluation of options and decisions within the project, and/or a comparison with alternatives.

This chapter often acts to differentiate project quality: even if the work completed is of a high technical quality, critical yet objective evaluation and comparison of the outcomes is crucial. In essence, the reader wants to learn something, so the worst examples amount to simple statements of fact (e.g., "graph X shows the result is Y"); the best examples are analytical and exploratory (e.g., "graph X shows the result is Y, which means Z; this contradicts [1], which may be because I use a different assumption"). As such, both positive *and* negative outcomes are valid *if* presented in a suitable manner.

Chapter 5

Conclusion

The concluding chapter of a dissertation is often underutilised because it is too often left too close to the deadline: it is important to allocate enough attention to it. Ideally, the chapter will consist of three parts:

1. (Re)summarise the main contributions and achievements, in essence summing up the content.
2. Clearly state the current project status (e.g., "X is working, Y is not") and evaluate what has been achieved with respect to the initial aims and objectives (e.g., "I completed aim X outlined previously, the evidence for this is within Chapter Y"). There is no problem including aims which were not completed, but it is important to evaluate and/or justify why this is the case.
3. Outline any open problems or future plans. Rather than treat this only as an exercise in what you *could* have done given more time, try to focus on any unexplored options or interesting outcomes (e.g., "my experiment for X gave counter-intuitive results, this could be because Y and would form an interesting area for further study" or "users found feature Z of my software difficult to use, which is obvious in hindsight but not during at design stage; to resolve this, I could clearly apply the technique of Smith [7]").

Bibliography

[1] Shlomi Agrnon, Etam Benger, Or Ordentlich, and Naftali Tishby. Critical slowing down near topological transitions in rate-distortion problems. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 2625–2630. IEEE, 2021.

[2] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[3] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*, volume 1. MIT press Cambridge, MA, USA, 2017.

[4] Ivan Chelombiev, Conor Houghton, and Cian O'Donnell. Adaptive estimators show information compression in deep neural networks. *arXiv preprint arXiv:1902.09037*, 2019.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[6] Warren S. McCulloch. An upper bound on the informational capacity of a synapse. In *Proceedings of the 1952 ACM National Meeting (Pittsburgh)*, ACM '52, page 113–117, New York, NY, USA, 1952. Association for Computing Machinery.

[7] Ramesh Kumar Mohapatra, Banshidhar Majhi, and Sanjay Kumar Jena. Classification performance analysis of mnist dataset utilizing a multi-resolution technique. In *2015 International Conference on Computing, Communication and Security (ICCCS)*, pages 1–5, 2015.

[8] Zoe Piran, Ravid Shwartz-Ziv, and Naftali Tishby. The dual information bottleneck. *arXiv preprint arXiv:2006.04641*, 2020.

[9] Claude E Shannon. A mathematical theory of communications. *Bell Syst. Tech. J.*, 27:623–656, 1948.

[10] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[11] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[12] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015.

[13] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018.

Appendix A

An Example Appendix

Content which is not central to, but may enhance the dissertation can be included in one or more appendices; examples include, but are not limited to

- lengthy mathematical proofs, numerical or graphical results which are summarised in the main body,
- sample or example calculations, and
- results of user studies or questionnaires.

Note that in line with most research conferences, the marking panel is not obliged to read such appendices. The point of including them is to serve as an additional reference if and only if the marker needs it in order to check something in the main text. For example, the marker might check a program listing in an appendix if they think the description in the main dissertation is ambiguous.