Deep Learning and Information Theory

Bhumesh Kumar (13D070060) || Alankar Kotwal (12D070010)

November 21, 2016

Abstract

THE machine learning revolution has recently led to the development of a new flurry of techniques called deep learning. These have been extremely successful in a variety of applications, often surpassing its competition and winning top machine learning challenges. These algorithms, however, are oft not as well-founded as 'classical' machine learning techniques: most research involves finding the 'right' network architecture for a particular application. We attempt to find a framework to make concrete this design of deep neural networks; the successive 'refinement' of features provided by the Information Bottleneck principle seems to fit this criterion well. We try to connect the information flow in a neural network to sufficient statistics as introduced by the information bottleneck and justify why this could potentially be used to design optimal architectures for deep learning.

1 Introduction

Deep learning has, in recent years, received tremendous attention due to the abilities of its models to consistently beat the state-of-the-art on various problems of interest in computer vision, linguistics and artificial intelligence. Recent successes include DeepMind's AlphaGo beating Lee Sedol on a full-sized 19 × 19 board, and automatic machine translation and content-based image retrieval. Deep learning has been used in applications as varied as automatic driving, handwriting generation, bioinformatics and genome sequencing due to its ability to produce highly expressive models that efficiently capture high-level information with excellent generalization properties.

However, the theoretical underpinnings of deep neural networks are not very well-understood. Deep learning has, indeed, been described as the triumph of empiricism in [2]. Though the Universal Approximation Theorem for multilayer neural nets [1] provides a bound on point-wise error between the target function and the network approximation for the optimal network, it doesn't specify anything about its architecture. If there exist design principles for deep networks, we do not understand them. Most research on deep nets, therefore, involves tweaking the network architecture for performance on a particular application.

The information bottleneck [4], introduced in 1999, cast the classical statistical concept of sufficient statistics in an information-theoretic framework. It tries to find the best, smallest representation $\hat{X}$ of the 'relevant' information in a random variable X, about a random variable Y. Such a representation is found by formulating a variational principle for efficient representation. This leads to a set of Blahut-Arimoto-like equations which when iterated over yield the probabilistic mapping between X and $\hat{X}$.

We attempt to connect the information bottleneck and deep learning because of their fundamental similarity: both 'refine' representations of their input random variables to learn high-level non-redundant representations of the (variable for the bottleneck and 'truth' for a neural net) Y.

2 Preliminaries

We now introduce the two concepts separately, and devote the next section to the integration of the two.

2.1 Deep Neural Networks

A neural network is an interconnection of multi-input single-output artificial 'neurons', which return a function of the signals input to it. A typical neuron linearly combines the input features and 'squashes' the linear combination using an activation function. This enables a typical neural network to capture rich, nonlinear features within the data and mimic nonlinear functions. A neural network is a feed-forward neural network if it can be organized into layers as shown in Fig. 1. We deal with feed-forward networks only because of their ease of training with backpropogation. A multi-layered neural network is usually called a deep network.

The input to a neural network, therefore, is a set of features and the output is the approximate target function we want to estimate. The data usually comes from a truth $Y \sim p(y)$, which generates $X \sim p(x|y)$. In a supervised learning setting, the random variable X is revealed to us, and we need to model the joint $p(x, y)$. The model extracted from the data, $\hat{p}(x, y)$ is then used to predict the truth $\hat{Y}$.

2.2 The Information Bottleneck

Consider the problem of finding the best, smallest representation $\hat{X}$ of the information about Y in a random variable X, with the Markov relation $Y \to X \to \hat{X}$. For the moment, we assume that the mutual information is a good measure of the relevance between two random variables. We then have a trade-off between two quantities: $I(\hat{X}; X)$, which represents how much of the information in X is retained in $\hat{X}$, and $I(\hat{X}; Y)$, which represents the information about Y retained by $\hat{X}$. For the optimal representation, we need to minimize the former: we want as less of the information from X to make the most concise summary of Y,

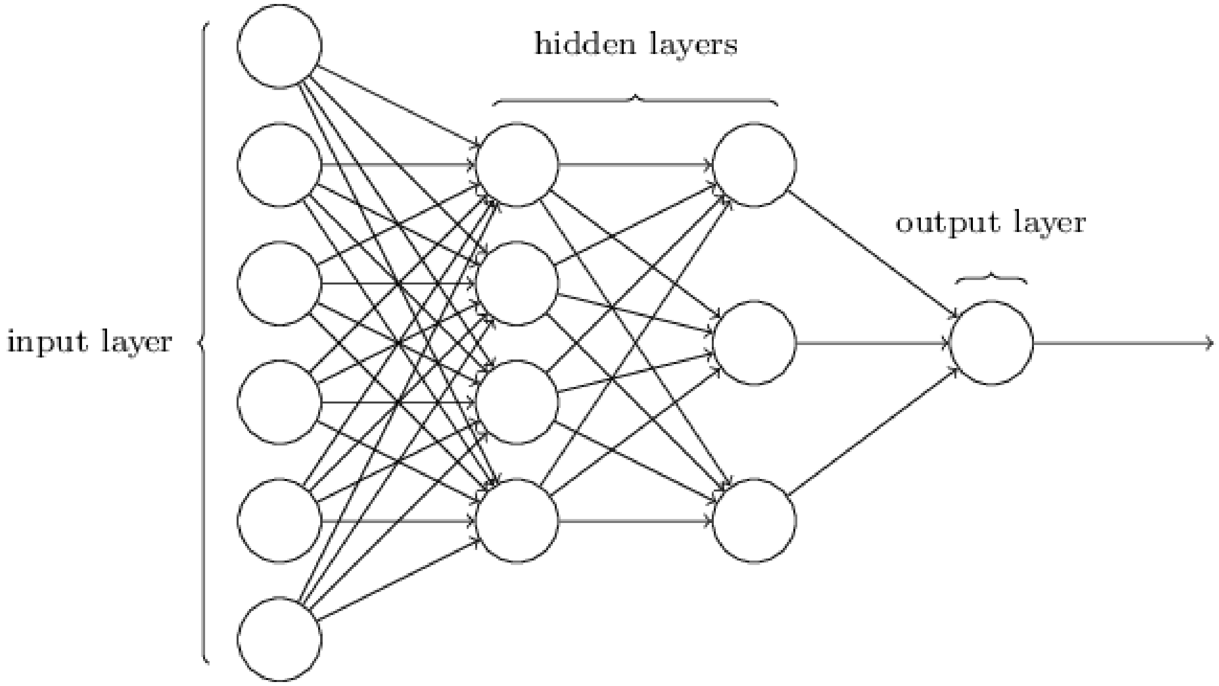

Figure 1: A neural network with two hidden layers. Taken from Neural Networks and Deep Learning

while maximizing the latter to keep most of the relevant information about Y in $\hat{X}$.

The optimization we then propose is the following:

$$min_{p(\hat{x}|x)} I(\hat{X}; X) - \beta I(\hat{X}; Y) \tag{1}$$

As [4] shows, the optimal $p(\hat{x}|x)$ for this Lagrangian has the form

$$p(\hat{x}|x) = \frac{p(\hat{x})}{Z(x,\beta)} \exp\left[-\beta \sum_y p(y|x) \log \frac{p(y|x)}{p(y|\hat{x})}\right] \tag{2}$$

where $Z(x, \beta)$ is a partition function. [4] goes further to derive a set of self-consistent equations to find this optimal conditional distribution.

We now state two important properties [3] that relates to why the proposed optimization function is ideal.

Theorem 2.1. *If T is a random mapping of X, T is a sufficient statistic for Y iff*

$$I(Y; T) = max_{T' \in (F)(X)} I(Y; T') \tag{3}$$

where $(F)(X)$ is the set of random mappings of X.

Theorem 2.2. *If T is a sufficient statistic for Y, T is a minimal sufficient statistic iff*

$$I(X; T) = max_{T' \in (S)(Y)} I(X; T') \tag{4}$$

where $(S)(Y)$ is the set of sufficient statistics for Y.

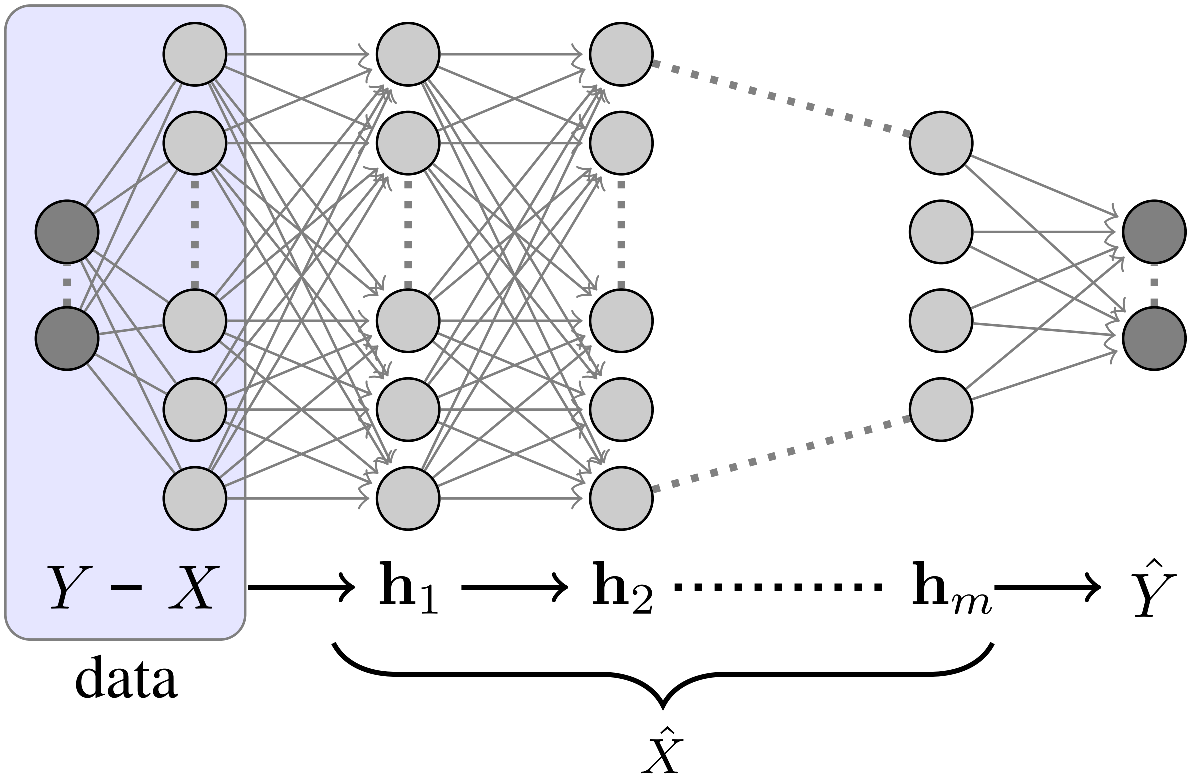

Figure 2: Information flow in a neural network. Figure from [5].

A sufficient statistic T for Y derived from X retains all the information needed to estimate Y, and a minimal sufficient statistic refines X at least as well as any other statistic. Our goal is to build a minimal sufficient statistic: so the implications of these theorems are clear. We need to keep T a sufficient statistic and therefore to maximize $I(Y;T)$, while maintaining the minimality and therefore minimizing $I(X;T)$.

3 The Information Bottleneck in Deep Neural Nets

We now connect the two concepts mentioned in the Preliminaries section. To start off, we notice the fact that the layered structure of the network generates a Markov chain of intermediate refinements. For instance, in Fig. 2, the input feature vector maps (possibly deterministically) to the random variable h_1 on the second layer. The second layer, again, maps to the third layer and so on, till the m^{th} layer maps to the refined random variable $\hat{Y}$. For $i \leq j$, h_i is a sufficient statistic for h_j, and hence as information flows through the network, it gets successively refined to the minimal sufficient statistic if the network parameters and configuration are just right. [5], indeed, postulates that training a neural network should accomplish this goal.

The problem of designing a network architecture, therefore, boils down to finding a 'path' to the optimal mapping between Y and $\hat{Y}$ through the intermediate variables h_i.

4 Conclusion

We showed a possible path to make formal the design neural networks with the information bottleneck. The concept of refinement of a variable to represent information about a target variable from the information bottleneck framework was used for this purpose. We showed the relation between sufficient statistics and the information bottleneck and argued why the optimization function is a good function to optimize on.

References

[1] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989.

[2] Zachary Chase Lipton. Deep learning and the triumph of empiricism. Link, 2015. [Online; accessed 18-November-2015].

[3] Ohad Shamir, Sivan Sabato, and Naftali Tishby. *Learning and Generalization with the Information Bottleneck*, pages 92–107. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[4] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *ArXiv Physics e-prints*, April 2000.

[5] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, April 2015.