An Alternative to Restricted-Boltzmann Learning for Binary Latent Variables based on the Criterion of Maximal Mutual Information

David Edelman
University College Dublin, Ireland

Keywords: Machine Learning, Data Compression, Information Theory, Unsupervised Learning.

Abstract: The latent binary variable training problem used in the pre-training process for Deep Neural Networks is approached using the Principle (and related Criterion) of Maximum Mutual Information (MMI). This is presented as an alternative to the most widely-accepted 'Restricted Boltzmann Machine' (RBM) approach of Hinton. The primary contribution of the present article is to present the MMI approach as the arguably more logically 'natural' and logically simple means to the same ends. Additionally, the relative ease and effectiveness of the approach for application will be demonstrated for an example case.

1 INTRODUCTION

As has become evident in recent years, the use of pre-training is crucial to the overall training of Deep Neural Networks. Historically, the training of feed-forward Neural Networks involved weight initialisation had been carried out by mere pseudo-random sampling, which worked satisfactorily for networks of few hidden layers. The inadequacy of this form of initialisation, however, inhibited research into networks of deeper architecture, and it was the key breakthrough of Hinton in 1999 (Hinton, 1999) introducing new methods for unsupervised 'pre-training', which first enabled the widespread use of Networks of Deeper architecture, which in turn marked the beginning of the resurgence in the research area of Neural Networks known as Deep Learning. In essence, the notion of 'pre-training' in a feed-forward network amounts to an iterated succession of unsupervised data compressions continuing forward into a network before supervised learning or training has begun. The method of compression proposed by Hinton is referred to as the 'Restricted Boltzmann Machine' (hereafter, RBM), a construct which owes its heuristics to analogy with problems in thermodynamics, and requires an intricate estimation procedure involving application of advanced Monte Carlo simulation including Gibbs Sampling, in a process referred to as Contrastive Divergence. The RBM approach in pre-training has been proven to be effective, and indeed become one of the most widely-used heuristics for carrying out pre-training. One question worth asking, however, is whether a logically simpler, more direct approach (not involving analogies, heuristics or requiring intricate simulations or calculations) might be found. It is this question to which the present article addresses itself.

In what follows, a method based on a probability-based measure called Mutual Information, which, it is argued, should be maximum between a pair of variables if one is considered to be an optimal compression of the other. Therefore, a Maximimum Mutual Information (hereafter, MMI) Criterion is introduced and applied in training to attempt or approach optimal compression from one network layer to the next. It will be argued that this leads to a practicable algorithm for achieving a similar aim as an RBM, and this algorithm is then exhibited as being effective and simple to implement, where a practical example from the Financial Markets is used to demonstrate.

Before proceeding, it should be mentioned that while the methods proposed here for the 'pre-training' problem would generally be applied in place of the RBM methodology, the latter will not be reviewed here. This is because it is believed that the RBM construct and the advanced techniques involved in its application do not lend themselves well to brief description and explanation, so it is therefore felt that readers unfamiliar with RBMs would not benefit from an attempt to describe it all here, even in general terms. By contrast, it is believed that a wide variety of readers will be able to follow the (arguably much simpler) approach adopted here for addressing the 'pre-training' problem, where many such readers might not readily be able to grasp and apply the RBM approach without considerable further study.

Edelman, D.
An Alternative to Restricted-Boltzmann Learning for Binary Latent Variables based on the Criterion of Maximal Mutual Information.
DOI: 10.5220/0007618608650868
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 865-868
ISBN: 978-989-758-350-6

2 BACKGROUND

As was mentioned earlier, unsupervised pre-training of Feed-forward Neural networks has enabled Deep Network architectures which were not possible to train previously. The RBM notwithstanding, we approach the pre-training afresh and consider how one might carry out such pre-training, based on first principles. In essence, the object is to compress a set of input variables into a set of binary (or, 'sigmoidally-approximated' binary) variables. We propose an 'Information-Theoretic' approach.

Consider the Shannon 'Information' (Shannon, 1949) of a random variable X with density $p_X(x)$, or merely the *Entropy* of X (in units of Information).

$$\mathcal{E}(X) = -E_X\{\log p_X(X)\},$$

where the expectation is understood to be with respect to the distribution of X.

Next, the *Mutual Information* (see (Cover and Thomas, 2006) and elsewhere) between variables X and Y is given by

$$\mathcal{H}(X;Y) = E_{XY}\{\log \frac{p_{XY}(X,Y)}{p_X(X)\cdot p_Y(Y)}\}.$$

[Note that the above is in the form of a cross-entropy and hence nonnegative, and also that if X and Y are independent, the ratio is identically unity and the expectation zero]

The approach proposed here, then, is based on directly maximisation of the (estimated) shared Information between the probability distributions of input and output ('compression') variables.

We proceed to specifics in the next section.

3 FORMULATION

Given observable variables $X = (X_1,\ldots,X_{m_X})$ and latent binary variables $h = (h_1,\ldots,h_{m_h})$, consider the problem usually addressed by a classic RBM in the present context, where

$$p(h|X) = \prod_{j=1}^{m_h} p(h_j|X)$$

where

$$p(h_j = 1|X) = 1/(1+\exp(b_j + X\cdot W_{\cdot j})),$$

and b denotes a vector of biases and $W_{\cdot j}$ denotes the j^{th} column of the weight vector W. For convenience, in the sequel we shall refer to this quantity as $f_j(X;b,W)$.

In what follows we seek to estimate b and W by maximising (as foreshadowed in the previous section) the Mutual Information between h and X.

Before proceeding, it is important to note that the Mutual Information may be written in terms of Entropy ($\mathcal{E}$):

$$E_{XH}(\log[p(X,h)/\{p(X)p(h)\}]) = \mathcal{E}(h) - E\{\mathcal{E}(h|X)\}$$

Since components of h are conditionally independent given X, the quantity $\mathcal{E}(h|X)$ is straightforward to calculate, as the sum of entropies of the components of h:

$$\mathcal{E}(h|X) = -\sum_j f_j(X;b,W)\log(f_j(X;b,W)) + (1-f_j(X;b,W))\log(1-f_j(X;b,W))$$

For a random sample of X,$x_1,\ldots,x_n$, if one conditions on the sample, then under the permutation distribution, the probability of any particular x_i is $\frac{1}{n}$. Hence, the quantity $E\{\mathcal{E}(h|X)\}$ is just the average of the above expression over the sample:

$$E\{\mathcal{E}(h|X)\} = -\frac{1}{n}\sum_{i=1}^{n}\sum_j f_j(x_i;b,W)\log(f_j(x_i;b,W)) + (1-f_j(x_i;b,W))\log(1-f_j(x_i;b,W))$$

Next, in order to compute $\mathcal{E}(h)$ some special attention will prove necessary. This is because the marginal density of h conditional on the sample is given by

$$\frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{m} f_j(x_i;b,W)^{h_j}(1-f_j(x_i;b,W))^{1-h_j}$$

which means that computation of the Entropy requires $n\cdot 2^m$ terms which is arguably intractable for any but cases of very small m. This being the case, if one wishes to avoid the requirement of Monte Carlo sampling (which is certainly one way forward), it is helpful to make further (fairly broad) assumptions about the the distribution of h.

If, for instance, one were to assume that the components of the latent variable vector h were independent (or that this were 'nearly' true, in some sense), then under the hypothesis of 'near'-product densities, the joint density would be well-approximated by

$$\prod_{j=1}^{m}\frac{1}{n}\sum_{i=1}^{n} f_j(x_i;b,W)^{h_j}(1-f_j(x_i;b,W))^{1-h_j},$$

or equivalently,

$$\prod_{j=1}^{m}(\frac{1}{n}\sum_{i=1}^{n} f_j(x_i;b,W))^{h_j}(1-\frac{1}{n}\sum_{i=1}^{n} f_j(x_i;b,W))^{1-h_j},$$

and the Entropy would be simple to compute:

$$-\sum_{j=1}^{m}\overline{f}_j(\cdot;b,W)\log(\overline{f}_j(\cdot;b,W))$$

$$... + (1-\overline{f}_j(\cdot;b,W))\log(1-\overline{f}_j(\cdot;b,W)),$$

where $\overline{f}_j(\cdot;b,W) = \frac{1}{n}\sum_i f_j(x_i;b,W)$ are the average proportions.

However, this assumption is not reasonable, but fortunately a much more tenable one allows a similar approximation, namely that higher-order dependence between components is characterised by the pairwise distributions [this would be true for correlated jointly-distributed Gaussian variables, for example]. In the present case of binary variables, there is a simple representation of a joint density:

$$p(h) = p(h_1,\ldots,h_m) = c\cdot\{\prod_{ij} p_{ij}(h_i,h_j)\}^{\frac{1}{m-1}},\ m>1$$

where p_{ij} denotes the joint density of (h_i,h_j) and c a normalisation constant typically close to unity. [It may be easily seen that in the special case of independence, the above expression reduces to the product density of the components h_i with $c \equiv 1$.].

Assuming the form of the above density, the Entropy may be computed

$$E\log\{p(h)\} = E[\frac{1}{m-1}\sum_{ij}\log\{p_{ij}(h_i,h_j)\}] + \log(c)$$

$$= \frac{1}{m-1}\sum_{ij} E[\log\{p_{ij}(h_i,h_j)\}] + \log(c)$$

$$= \frac{1}{m-1}\sum_{ij}[p_{ij}(1,1)\log\{p_{ij}(1,1)\} + \cdots$$

$$+p_{ij}(1,0)\log\{p_{ij}(1,0)\} + p_{ij}(0,1)\log\{p_{ij}(0,1)\} + \cdots$$

$$+ p_{ij}(0,0)\log\{p_{ij}(0,0)\}] + \log(c).$$

Given the input sample, the $p_{ij}(h_i,h_j)$ for the four cases of the argument may be computed by

$$p_{ij}(1,1) = \overline{f_i f_j}$$

$$p_{ij}(1,0) = \overline{f}_i - \overline{f_i f_j}$$

$$p_{ij}(0,1) = \overline{f}_j - \overline{f_i f_j}$$

$$p_{ij}(0,0) = 1 - \overline{f}_i - \overline{f}_j + \overline{f_i f_j}$$

where the expectations (here denoted by bars over the respective variables), conditional on the sample, are computed by averaging.

4 RESULTS OF PRELIMINARY EXPERIMENT

As part of a larger experiment to forecast the daily return of the S& P 500 Index, a dataset was organised into a vector of output targets consisting of daily returns from 4500 days, each to be forecast by the returns of the respective 5 previous days' returns, organised into a 5-column matrix with the same number of rows as the output target. As a first step to train a Deep Network, it was decided to identify a feature array of 20 binary variables for each input vector, using the MMI method described above.

As a test, the MMI algorithm was implemented in MATLAB@(MATLAB, 2011) and applied to the input data.

It was desired to encode the input information to estimate the 20 quasi-binary latent variables containing the most Information from the Input, as the beginning of a stack of encoders to eventually be used to initialise training for a Deep Learning forecast of the next day's return (the output target). The Matlab@(MATLAB, 2011) implementation required encoding of the Mutual Information between the input data and predictions as a function of connection weights, where a small (.00001) L2 penalty was applied to the weights for stability. The Unconstrained Function Minimisation ('fminunc') routine of MATLAB@(MATLAB, 2011) was applied with gradients supplied, which converged in approximately 3 minutes on a Macbook Pro laptop (2015, OSX 10.11.3, Intel 2.9GHz core-i5, 8Gb RAM).

The resulting output was (to 5 significant figures) an array of 20 columns of uncorrelated binary (0-1) variables with column means each equal to 0.50000.

It is worth noting that the resulting 'Maximum-Entropy' character of the estimated output distribution was not anticipated but is arguably promising for application, and not surprising given the use of Entropy as a fitting criterion. [It is also worth noting that an earlier attempt of the same experiment using the (over?)simplification of assuming independence of the components of h (mentioned in an earlier section of this article) failed, with the resulting output having mostly degenerate (all zero or all one) columns].

5 CONCLUSION

In the previous, it has been seen that data compression to binary (or near-binary) variables may be simply achieved using the principle of Maximum Mutual Information (MMI) using a 'one-size-fits-all' algorithm. The primary claim of the present paper is that the approach presented here is far simpler logically and algorithmically than the most widely-accepted method based on the Restricted Boltzmann Machine (RBM) construct. As it has not been argued here that the resulting compressions achieved by MMI are in anyway superior to those achieved via RBMs given suf-

ficient computation time, it is believed that side-by-side practical comparison would entail careful benchmarked studies involving specific hardware-related performance, which is not included here. It is, however, hoped that the new methods suggested here may help contribute to the logical and algorithmic simplification of pre-training of Deep Neural Networks, and that practitioners will confirm the author's conjecture (and limited experience) that the computations of implementing MMI are comparable in scope and perhaps even more efficient numerically than those of RBMs.

REFERENCES

Cover, T. and Thomas, J. (2006). *Elements of Information Theory (2nd ed.), Ch. 2*. Wiley New York.

Hinton, G. (1999). Products of experts. In *Proceedings of the Ninth International Conference on Artificial Neural Networks, Vol. I*, pages pages 1–6. ICANN.

MATLAB (2011). *version 7.12.0.635 (R2011a)*. The MathWorks Inc., Natick, Massachusetts.

Shannon, C. E. (1949). Communication in the presence of noise. *Proc. Institute of Radio Engineers*, 37(1):10–21.