Stats 231C - Homework 2
Review: Deep Belief Networks are Compact Universal Approximators (by N. Le Roux and Y. Bengio)

Peter Racioppo

May 4, 2020

1 Introduction

The authors improve on a prior upper bound on the number of parameters a deep but narrow Deep Belief Network (DBN) needs to represent distributions on its inputs. These bounds show that deep but narrow generative networks require no more parameters than shallow ones to achieve universal approximation. The authors then use the same proof technique to improve on prior bounds on the number of parameters needed for universal approximation in a deep but narrow neural network, and show that deep but narrow feed-forward neural networks with sigmoidal activations can represent any Boolean expression.

2 Background

Hopfield Networks: A Hopfield network is a type of recurrent neural network consisting of a complete, undirected graph with binary-threshold activations. Typically, we impose the contraints that all neurons have symmetric connectivity weights $(w_{ij} = w_{ji})$ and no connection with themselves $(w_{ii} = 0)$. We may define an energy function (Hamiltonian) for the network, analogous to an Ising model, which can be shown, under the previous constraints, to monotonically decrease under updating of the network weights. With the Hamiltonian as a Lyapunov function, LaSalle's Invariance Principle from dynamical systems theory can be used to show that the state space trajectory of a Hopfield network must converge to a local minimum [1].

Restricted Boltzmann Machines: A Boltzmann machine can be thought of as a stochastic, generative variant of a Hopfield network (it is also a Markov random field). Boltzmann machines with unconstrained connectivity have not been useful in machine learning problems. The restricted Boltzmann machine (RBM), introduced by P. Smolensky in 1968, is the restricted form of a Boltzmann machine in which neurons form a bipartite graph [2], with hidden units $\{h_j\}$ and visible units $\{v_j\}$. Thus, the hidden units are independent of each other, given the visible units, and vice-versa. This network architecture is capable of learning probability distributions over its inputs. As in the Hopfield network, we define a Hamiltonian $E(h, v) = -a^T v - b^T h - v^T W h$, and probability distributions over unit weights as $P(v, h) = (1/Z)e^{-E(v,h)}$, where Z, the partition function, is a normalizing constant. The marginal probability distribution of a vector of Boolean inputs (in the visible layer) is $P(v) = (1/Z)\sum_h e^{-E(v,h)}$. By mutual independence of the visible/hidden units, $P(v|h) = \prod_{i=1}^{m} P(v_i|h)$ and the individual activation probabilities are $P(v_i = 1|h) = \sigma(a_i + \sum_{j=1}^{n} w_{ij}h_j)$, where σ denotes a sigmoidal activation. The contrastive divergence (CD) algorithm, an efficient algorithm for training RBMs, was introduced by

Hinton et al. in 2006 [3]. The algorithm performs Gibbs sampling inside of gradient descent, as an approximation of the maximum likelihood method. (In Hinton's paper, $\sigma(a) = 1/(1 + e^{-a})$.)

Deep Belief Networks: The discovery by Bengio et al. in 2007 that DBNs stacked in a deep neural network can be trained one layer at a time led to one of the first successful deep learning methods [4]. Thus, a deep belief network is trained greedily, by successive applications of the CD algorithm. The model can be parameterized by factorized conditional probability distributions: $P(x, h_1, h_2, ...h_l) = P(x_1|h_1)P(h_1|h_2)...P(h_{l-2}|h_{l-1})P(h_{l-1}, h_l)$. Here, x is the input to the network, and the $\{h_i\}$ represent subsequent hidden network layers. In Hinton et al (2006), and the present paper, all network connections are feedforward except for the final connection between layers $l - 1$ and l. These top two layers have undirected connections and together comprise a restricted Boltzmann machine.

Gray Code: Reflected binary code (RBC), also known as Gray code after Frank Gray, are binary codes, in which the Hamming distance between any two successive sequences is 1. In other words, any two successive Gray code sequences differ by only one bit. Gray codes are useful for error correction, esp. in digital communications. They appear in the work of Sutskever and Hinton (2008) and are employed in the proofs in the present paper.

Universal Approximation: Deep network architectures are useful because they can represent certain functions exponentially more efficiently than shallow networks, or as the authors write, "there are functions that can be represented compactly with a neural network of depth k but that would require exponential size (with respect to input size) networks of depth $k - 1$ [6,7]. The first universal approximation result for deep but narrow neural networks was obtained by Rojas in 2003 for deep feedforward networks [10]. Le Roux and Bengio explored the representational power of DBNs in a 2008 paper, and Sutskever and Hinton demonstrated in the same year that DBNs are universal approximators, which can represent any distribution on binary inputs (distributions over $\{0, 1\}^n$) with no more than 3×2^n layers [8,9]. The present paper improves this bound to $2^n/n + 1$ layers of size n, which is no more parameters than the number required in a single RBM. In their 2008 paper [9], Sutskever and Hinton demonstrated that a universal approximator can be built iteratively, by a series of 3-layer networks. Given a sequence $\{a_i\}$ of binary vectors in $\{0, 1\}^n$, the top-level DBN assigns probability 1 to a_1. The next 3-layer network assigns probability a_2 probability $1 - p(a_1)$. Continuing in this way, (at the ith step leaving vectors a_1 through a_{i-2} unchanged and transferring probability mass from a_{i-1} to a_i) each vector in $\{a_i\}$ is assigned its correct probability. This method is improved in the present paper, by replacing the three-layer sigmoid belief networks of size $n + 1$ with one-layer networks of size n.

3 Overview

In Theorem 1, the authors show how to iteratively perform the transfer of probability mass operation with a single layer per step. They then improve upon this result in Theorems 2 and 3 by showing how to change $O(n)$ bits per layer rather than 1, which reduces the required number of layers from $O(2^n)$ to $O(2^n/n)$. In Theorem 4, they prove that a DBN constructed in this manner is a universal approximator. In Sec. 5, the authors employ the same proof technique to show that deep but narrow feedforward neural networks are universal approximators, and they improve slightly upon the bounds in the 2003 paper by Rojas [10], reducing the necessary number of hidden units from $n + 1$ to n.

4 Quantitative Evaluation

The authors begin their analysis by demonstrating a method to change one bit of a Gray code at a time in a layer. We consider layers h and v, with weight W_{ij} linking unit v_i and h_j and b_i the corresponding bias. Letting $W_{ij} = 0, \forall i \neq j$, $W_{ii} = 2w$, and $b_i = -w$ the input to v_i is $I = 2wh_i - w$. Setting $w = \sigma^{-1}(1 - \epsilon)$, since h_i can be 0 or 1, $I = \pm w = \pm \sigma^{-1}(1-\epsilon)$, so the output of the layer is $1-\epsilon$.

Theorem 1: Let a_t be an arbitrary binary vector in $\{0,1\}^n$. Given a binary vector h and a probability p, weights W_n and b_n can be chosen such that if $h \neq a_t$, the last bit of h remains unchanged with probability $1-\epsilon$, and if $h = a_t$, the last bit of h is flipped with probability p. Following the method in Sutskever and Hinton [9], such a network is a universal approximator.

The previous theorem still requires $2^n + 1$ layers, one for each bit, and each layer requires on the order of n^2 parameters, so that the total number of parameters is $O(n)$ times more the number required in a very (exponentially) fat RBM. Theorem 2 shows how to improve upon Theorem 1 by an order of n.

Theorem 2: Let a_t be an arbitrary binary vector in $\{0,1\}^n$ and define a second vector c_t which has the same value as a_t but with its last bit flipped. Given a binary vector h and probabilities p_0, p_1, weights W_n and b_n can be chosen such that if $h \neq a_t, c_t$, the last bit of h remains unchanged with probability $1-\epsilon$. If $h = a_t$, the last bit of h is flipped with probability p_0 and if $h = c_t$, the last bit of h is flipped with probability p_1.

Central to Theorems 1 and 2 is the realization that all terms except the bias can be forced to zero when h is equal to the given binary vectors and that otherwise we can bound the input I as no closer than a constant w from the desired probability p. It is then straightforward to select a w that guarantees that $\sigma(I) > 1-\epsilon$.

Theorem 3: This theorem shows, by a constructive argument, that there exist sequences of vectors of n bits, such that these sequences partition the set of all such vectors, appropriately indexed pairs of elements of these sets have Hamming distance 1, and the same bit is not repeatedly flipped on the same vector. It is evident in the introduction of Theorem 3 that this result will be necessary for Theorem 4's improvment on Theorem 1. However, for improved clarity, the authors should make it clear from the start why this result will be necessary. The authors' example of the sequences in Theorem 3 should also be better explained.

Theorems 2 and 3, and the following lemma, provide the basis for Theorem 4, which is the main result of the paper.

Lemma 1: The authors show, using the sequences defined in Theorem 3, that if a DBN with $2^n/n + 1$ layers can be constructed, this network can generate an arbitrary distribution over vectors of n bits as its output. The authors show constructively how the hidden layers can be chosen to obtain this result, and they claim that this is the only possible such construction (though evidently without proof). This construction is written very compactly and with sparse explanation.

Theorem 4: The authors now show how to construct a DBN such that the conditions in Lemma 1 are met. Let t be such that $n = 2^t$. At each layer, the first t bits are copied to the next layer (with probability arbitrarily close to 1). Of the remaining $n-t$ bits, $n/2$ are changed with the desired probability if they do not match one of two binary vectors, which is possible by Theorem 2. The remaining bits are copied to the next layer. These choices are made in order to satisfy the constraints on the conditional probabilitiy distributions required in Lemma 1. It would be helpful if the authors had explained how these probability distributions were obtained.

The authors now use the proof technique in the previous section to prove the universal approximation property of feedforward neural nets. By specifying that each layer in the network be connected not only to the previous layer but also directly to the input, the authors improve the number of hidden units required in each layer from $n+1$ to n.

Theorems 5 & 6: Let $f : \{0,1\}^n \mapsto \{0,1\}$. Theorem 5 demonstrates that it is possible, at every layer (each of size n), to choose an arbitrary binary vector h, with $h \neq h_0$ and $f(h) = 0$, and map it to h_0, leaving all other vectors unchanged. It follows (Theorem 6) that a neural network with $2^{n-1} + 1$ such layers can then perform the same mapping as f.

5 Qualitative Evaluation

The paper proves an important result. The authors substantially improve on the upper bound a narrow (DBN) needs to represent distributions on its inputs, and their constructive and novel proof technique will likely be of future use. Indeed, the authors demonstrate its utility by using it to prove a second result on feed-forward networks. This result is significantly less important than the first results, as the authors themselves note, as it provides only a small improvement on the previous bound. Although similar universal approximation results had already been obtained in previous work, the novel proof method of this paper is an interesting contribution. The paper is well-written, its purpose is clearly explained, and the background material is well-introduced. The first several proofs are simple enough that they can be followed without much exposition. However, the authors' terse style of proof is more problematic in Theorems 2-4. Remaining challenges include proving lower bounds for these network architectures and showing how tight the upper bounds are. The authors indicate their belief that the upper bounds they provide can be improved by no more than a factor of 2. The authors also highlight the need to explore which architectures are best suited to modeling distributions of interest.

6 Impact Score

Impact Score: 8/10
The impact of an $O(n)$ improvement on parameter bounds is substantial. The universal approximation results are also interesting, and had similar results not already been obtained in past works, these results would merit a 10/10 impact score.

Confidence Score: 5/10
The importance of the improvement on the parameter bounds is clear, especially given the importance of deep belief networks. It is less clear how impactful the introduction of this method of proof will be. As a nonexpert in this field, I feel confident that this paper represents a substantially above-average contribution, but I lack the expertise to assign it a precise impact score.

7 References

[1] H. K. Khalil, "Nonlinear Systems," 3rd Edition, Prentice Hall, Upper Saddle River, 2002.

[2] Smolensky, Paul (1986). "Chapter 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory" (PDF). In Rumelhart, David E.; McLelland, James L. (eds.). Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations. MIT Press. pp. 194–281. ISBN 0-262-68053-X.

[3] Salakhutdinov, R.; Mnih, A.; Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. Proceedings of the 24th international conference on Machine learning - ICML '07. p. 791. doi:10.1145/1273496.1273596. ISBN 9781595937933.

[4] Hinton GE, Osindero S, Teh YW (July 2006). "A fast learning algorithm for deep belief nets" (PDF). Neural Computation. 18 (7): 1527–54. CiteSeerX 10.1.1.76.1541. doi:10.1162/neco.2006.18.7.1527. PMID 16764513.

[5] Bengio Y, Lamblin P, Popovici D, Larochelle H (2007). Greedy Layer-Wise Training of Deep Networks (PDF). NIPS.

[6] Bengio, Y. (2009). "Learning Deep Architectures for AI" (PDF). Foundations and Trends in Machine Learning. 2: 1–127. CiteSeerX 10.1.1.701.9550. doi:10.1561/2200000006.

[7] Hastad, J. and Goldmann, M. (1991). "On the power of small-depth threshold circuits." Computational Complexity, 1, 113–129.

[8] Le Roux, N. and Bengio, Y. (2008). "Representational power of restricted boltzmann machines and deep belief networks." Neural Computation, 20(6), 1631–1649.

[9] Sutskever, I. and Hinton, G. E. (2008). "Deep, narrow sigmoid belief networks are universal approximators." Neural Computation, 20(11), 2629–2636.

[10] Rojas, R. (2003). Networks of width one are universal classifiers. In International Joint Conference on Neural Networks, volume 4, pages 3124–3127.