Confidence Values and Compact Rule Extraction From Probabilistic Neural Networks

Simon Odense and Artur d'Avila Garcez

City, University of London, UK
simon.odense@city.ac.uk
a.garcez@city.ac.uk,

One of the key challenges of extracting rules from neural networks is accommodation of the inherent flexibility of knowledge representation in neural networks to more rigid rule based systems. Neural networks are often seen as having 'soft constraints' as opposed to the 'hard constraints' of rule based systems. This distinction has been identified as one of the key differences between the connectionist approach and more traditional symbolic AI [1]. For deterministic networks, the distinction becomes somewhat fuzzy as every input/output relationship of a network can be encoded in a set of propositional rules with arbitrary precision, however, representing a neural network this way will be at the very least incomprehensible and in most cases intractable. Thus the issue of flexibility is tied to the issue of compactness of representation. For probabilistic networks, the issue of flexibility becomes even more of a challenge. Here we go over results showing that a previous rule extraction method applied to Restricted Boltzmann machines(RBMs) [5] can be improved by considering more compact rules called M of N rules. We also consider an example highlighting the advantage that these rules have in terms of minimizing the incidents of 'false negatives' over traditional conjunctive rules. Finally we look at the notion of 'confidence values', numeric values we associate with a rule meant to represent our degree of belief in the rule, and show that a more refined notion of confidence may be helpful when considering extracted rules from RBMs and other probabilistic networks.

Tran and Garcez developed a rule extraction algorithm for RBMs (and DBNs built from RBMs) which associates confidence values with extracted rules by looking at the average weight of the literals in the rule [2]. The extraction algorithm works by starting with the conjunction of every literal in the rule and iteratively updating the confidence value and pruning literals with small enough weights until equilibrium is achieved. The extracted rules are then composed in a deep belief network along with an inference rule in order to calculate confidence values for the output given a (partial) set of confidence values for the input. When looking at RBMs in isolation, the extracted rules can be thought of as biconditionals, however, the following example shows that when looking at RBMs a high confidence value does not necessarily correspond to a high probability of the rule being true. First, when we say that an extracted rule has a certain probability in the network we mean that, given that the visible units are uniformly distributed (if the visible distribution is defined by the network it can be shown that local rule extraction preserving the probabilities is impos-

sible assuming some basic conditions on the network), the rule has a certain probability of being true in the distribution of the network. For example, if a network has a probability distribution P, for two hidden units in a network, h_1 and h_2, $h_1 \vee h_2$ is given a probability $P(h_1) + P(h_2)$. Given a biconditional $h \leftrightarrow x_1, ..., x_k, \neg x_{k+1} ... \neg x_n$, where each x_i represents a visible unit, we will consider the probability of the biconditional being true in an RBM. For brevity we will denote the antecedent of the biconditional as ANT, the probability of this biconditional in an RBM is then $P(h = 1, ANT) + P(h = 0, \neg ANT)$ Where the distribution on the set of literals in the antecedent is uniform (since they represent visible units). We will consider an example of a rule extracted using the algorithm mentioned above to show that the associated confidence doesn't reflect the probability of the biconditional in the network. Define a network with a single hidden neuron with k identical weights W and bias 0, the antecedent of the extracted rule is the conjunction of all the literals and the confidence is W. This means that the antecedent is satisfied only when all k literals are satisfied. Using some algebra, the probability of the biconditional being true in the network can be written as

$$P(h = 1|ANT)P(ANT) + \sum_{i=0}^{k-1} \binom{k}{i} (1 - P(h = 1|ANT^i))P(ANT^i)$$

Where $P(h = 1|ANT)$ is the probability of the hidden neuron being on when the antecedent is satisfied and $P(h = 1|ANT^i)$ is the probability of the hidden neuron being on when exactly i literals of the rule are satisfied, since all the weights are the same this does not depend on which specific literals are not satisfied. Furthermore we are assuming that this visible units are taken from a uniform distribution so we have $P(ANT) = P(ANT^i) = \frac{1}{2^k}$. This gives us

$$\frac{1}{2^k} \left(\sigma(Wk) + \sum_{i=1}^{k-1} \binom{k-1}{i} (1 - \sigma(iW)) \right)$$

Since W and k are arbitrary we can take them to be as large as possible, in which case the limit of the right term goes to $1 - \sigma(0) = 0.5$ and the left hand term goes to 1 so as $k \to \infty$ the whole thing goes to 0. This shows we can extract rules with arbitrarily high confidence but arbitrarily low probability.

The issue with this example is that the extracted rule give many false negatives. There are many cases where the rule should be giving an output of 1 but is failing to since not every literal is satisfied. Rather than requiring every literal in the antecedent be satisfied in order to predict 1 we really only need one of them. It's difficult to extract a single conjunctive rule which can accurately capture the behaviour of a probabilistic network and by extracting many different rules you lose compactness. In order to find a compact way to more faithfully capture the behaviour of an RBM we relax the condition that every literal in the antecedent needs to be satisfied. This give us the so called M of N rules. In an

M of N rule the antecedent is satisfied if only M of the N literals are. In the previous example the correct rule would be 1 of the set of literals. By first applying the rule extraction algorithm and selecting M by looking for the minimum value of M for which $M \cdot c$ (where c is the confidence given to the rule) is greater than a predetermined threshold (in our case the minimum input to the hidden node) we can convert the purely conjunctive rules into M of N ones. If we cannot find an appropriate M we add new literals until there either is an appropriate M or we run out of literals. The rules produced by this algorithm perform much better than the purely conjunctive rules when tested with a variety of small datasets [6].

Assigning values to logical sentences to measure degrees of belief has been done before. The most relevant examples for us are penalty logic [4] and Markov logic networks [3], in both cases 'weights' were given to logical sentences which were then translated into weights of a network (Hopfield networks and Markov random fields respectively). A similar philosophy was used to define confidence values for deep belief networks by using the weights of the RBM in the previous algorithm. The above example shows that the extracted confidence really does not accurately reflect the underlying probability of structure of the constituent RBMs and that the extracted rules are perhaps better considered in the feed forward context rather rather than biconditionals. Extending this algorithm to M of N relieves some of the problems by loosening the requirements for the rule to be satisfied but it remains to be seen whether the confidence values extracted with M of N rules more accurately reflect the probability structure of the RBM. One possible avenue of research is, rather than look simply at the weights attached to the literals to derive confidence, look at both the minimum input to a node when the rule is satisfied and the maximum input to the node when the rule is not satisfied. Ultimately the M of N rule is a promising way of representing knowledge in a neural network with more possibilities to imbue it with more flexibility by exploring various notions of confidence values

References

[1] Smolensky, P..: On the Proper Treatment of Connectionism. Behavioral and Brain Sciences. 11(1), 1–23 (1988)

[2] Son, T., d'Avila Garcez, A.: Deep Logic Network: Inserting and Extracting Knowledge from Deep Belief Networks. IEEE Transactions on Neural Networks and Learning Systems, pp(99), 1–13(2016)

[3] Richardson, M., Domingos, P.: Markov Logic Networks. Machine Learning. 62(1), 107–136(2006)

[4] Pinkas, G.: Reasoning, Connectionist Nonmonoticity and Learning in Networks that Capture Propositional Knowledge. Artificial Intelligence. 77, 203–247

[5] Smolensky, P.: Information Processing in Dynamical Systems: Foundations of Harmony Theory. Parallel Distributed Processing: Volume 1: Foundations. MIT Press, Cambridge, 194–281 (1986) Comm. Pure Appl. Math. 33, 609–633 (1980)

[6] Odense, S., d'Avila Garcez, A.: Extracting M of N Rules From Restricted Boltzmann Machines. The 26th International Conference on Artificial Neural Networks. to appear, (2017)