Feature-Extracting Functions for Neural Logic Rule Learning

Shashank Gupta   Antonio Robles-Kelly
School of IT, Faculty of Eng., Sci. and the Built Env., Deakin University, Waurn Ponds, Victoria 3216, Australia

***Abstract*—In this paper, we present a method aimed at integrating domain knowledge abstracted as logic rules into the predictive behaviour of a neural network using feature extracting functions. We combine the declarative first-order logic rules which represents the human knowledge in a logically-structured format akin to that introduced in [1] with feature-extracting functions which act as the decision rules presented in [2]. These functions are embodied as programming functions which can represent, in a straightforward manner, the applicable domain knowledge as a set of logical instructions and provide a cumulative set of probability distributions of the input data. These distributions can then be used during the training process in a mini-batch strategy. We also illustrate the utility of our method for sentiment analysis and compare our results to those obtained using a number of alternatives elsewhere in the literature.**

***Index Terms*—Neural logic, feature extracting functions, rule learning**

I. Introduction

Deep Neural Networks provide high levels of performance in various the Pattern Recognition tasks but they require large amounts of labelled training data mainly due to the notion that their training is often purely data-driven, with no direct or indirect human intervention involved. As a result, the interpretation of the transformation between input and output is often challenging if not almost intractable, whereby they do not have an inherent representation of causality. Previous work has shown that supervision purely in the form of data can lead them to learn some unwanted patterns and provide wrong predictions [3] [4].

These drawbacks hinder their application in a wide variety of areas such as safety critical systems, medical applications, food security, fault detection, power generation and transmission and critical environmental management which require a level of trust or confidence associated with the output of the network [5]. In [1] the authors encode human or domain knowledge represented as a set of Declarative First Order Logic rules into the parameters of the network via indirect supervision making use of Knowledge Distillation [6] at each iteration of training. Note that the approach in [1] is such that the domain knowledge is "imprinted" into the network permanently through the parameters of the network and, thus, any change in the existing rules or addition of new ones will require a re-training of the whole network. This approach also requires the transformation of the knowledge from natural language to logic rules and the subsequent soft-encoding [7] task, which is application-specific.

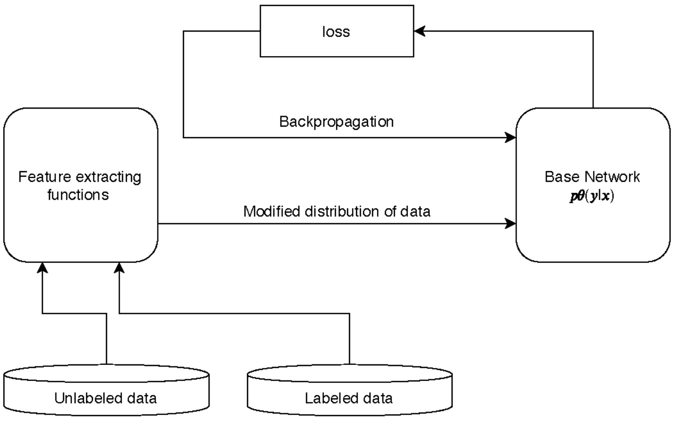

Fig. 1. Proposed system overview. At each iteration, a batch of data is passed to the feature extracting functions whose output is then used to update the distribution $p_\theta(y|x)$. This updated distribution is then used to train the neural network.

Here, we present the use of feature-extracting functions instead which are directly applied on the input data so as to transfer the human knowledge into a distribution and influence the output of the Neural Network [1]. This eliminates the need of assuming, *a priori*, the initial posterior distributions of the features in the data. In our approach, we derive a feature-extracting function from each logic rule by viewing it as a mini-batch processing step during each iteration. Since this function is applied directly to the data, we do not compute probability distributions nor construct a teacher network as in [1]. This effectively reduces the complexity of the method. Also, these feature-extracting functions can be modified at any time during the training process, thus providing a lot of flexibility in adapting to qualitative and quantitative characteristics of data to provide a more direct nature of supervision based upon the input data [8]. In Figure 1, we show the diagrammatic representation of our approach.

II. Experiments

In order to show a comparison with the method in [1], we performed sentence-level sentiment analysis and classified each sentence into the positive or negative categories. We have used the Convolutional Neural Network architecture proposed in [9] employing it's "non-static" version with the exact same configuration as that presented by the authors. Again, we have initialised word vectors using word2vec [10] and used fine

[1] For a detailed description of the method, go to https://arxiv.org/abs/2008.06326

tuning, training the neural network using stochastic gradient descent (SGD) with the AdaDelta updates [11].

Since contrasting senses are hard to capture, we define a linguistically motivated rule called "A-but-B" rule akin to that in [1] which states that if a sentence has an "A-but-B" structure, the sentiment of the whole sentence will be consistent with the sentiment of it's "B" statement. From this rule, we can define a feature-extracting function $F_1 = A - but - B(x, y)$ on set D which takes the input pair of sentence-label (x, y) and outputs $(x*, y)$ where $x*$ is corresponding "B" features of x.

We evaluate our method on three public data-sets. The first of these is the Stanford Sentiment Treebank (SST2) [12] which contains 2 classes (negative and positive), and 6920/872/1821 sentences in the train/dev/test sets, respectively. Following [9] we train the models on both, sentences and phrases. The second data set used here is the movie review one (MR) introduced in [13]. This data set consists of 10,662 one-sentence movie reviews with negative or positive sentiments. Finally, we also employ the customer reviews of various products data set (CR) presented in [14], which contains 2 classes and 3,775 instances. For the MR and CR, we use 10-fold cross validation so as to be consistent with previous works in [1] and [9].

Here, we have compared our results with the non-static version of the network in [9] as published by the authors and the Iterative-distillation method in [1] on the three data sets under consideration. To this end, in Table I, we show the accuracy yielded by our method (CNN-F), the method in [1] (CNN-rule) and that in [9] (CNN). Table II shows the Precision, Recall and F1-scores for the three data sets. In both tables, where applicable, *i.e.* the MR and CR data sets, we also show the corresponding variance over the ten trails corresponding to the 10-fold cross validation.

From the experimental results, we can appreciate that our method performs better on both, the SST2 and MR data sets by all measures. It is quite competitive on the CR dataset too, just barely behind the method in [1]. Thus, we can conclude that our method works as intended and is quite competitive, outperforming the alternatives despite using only one rule for comparison. Since our method represents knowledge purely in terms of a distribution on input data and is able to perform better, we can infer that Iterative-Knowledge Distillation [1] may not be very effective in transferring knowledge as it intends to. Moreover, this difference can be further highlighted if we can construct more rules and perform experiments on a wide variety of datasets from various domains like Computer Vision.

Approach	**SST2**	**MR**	**CR**
CNN	87.2	81.3±0.1	84.3±0.2
CNN-rule	88.8	81.6±0.1	85.0±0.3
CNN-F	**89.1**	**81.8±0.4**	**84.8±0.1**

TABLE I
ACCURACY PERCENTAGES OF SENTIMENT CLASSIFICATION TASK OBTAINED USING OUR METHOD (CNN-F), THE METHOD IN [1] (CNN-RULE) AND THAT IN [9] (CNN).

SST2			
Approach	Precision	Recall	F1-score
CNN	0.89	0.85	0.87
CNN-rule	0.90	0.87	0.88
CNN-F	**0.91**	**0.87**	**0.89**

MR			
Approach	Precision	Recall	F1-score
CNN	0.80±0.004	0.82±0.005	0.81±0.003
CNN-rule	0.81±0.005	0.82±0.007	0.81±0.003
CNN-F	**0.81±0.005**	**0.83±0.004**	**0.82±0.002**

CR			
Approach	Precision	Recall	F1-score
CNN	0.78±0.012	0.78±0.017	0.78±0.009
CNN-rule	0.77±0.014	0.79±0.014	0.78±0.008
CNN-F	**0.76±0.014**	**0.78±0.015**	**0.77±0.008**

TABLE II
PRECISION, RECALL AND F1-SCORES YIELDED BY OUR METHOD (CNN-F), THE METHOD IN [1] (CNN-RULE) AND THAT IN [9] (CNN) AS APPLIED TO THE THREE DATA SETS UNDER STUDY.

REFERENCES

[1] Z. Hu, X. Ma, Z. Liu, E. H. Hovy, and E. P. Xing, "Harnessing deep neural networks with logic rules," *CoRR*, vol. abs/1603.06318, 2016.

[2] A. Ratner, S. H. Bach, H. R. Ehrenberg, J. A. Fries, S. Wu, and C. Ré, "Snorkel: Rapid training data creation with weak supervision," *CoRR*, vol. abs/1711.10160, 2017.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[4] A. M. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," *CoRR*, vol. abs/1412.1897, 2014.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," *CoRR*, vol. abs/1602.04938, 2016.

[6] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv*, vol. abs/1503.02531, 2015.

[7] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor, "Hinge-loss markov random fields and probabilistic soft logic," *CoRR*, vol. abs/1505.04406, 2015.

[8] D. D. Lewis, "Feature selection and feature extraction for text categorization," in *Proceedings of the Workshop on Speech and Natural Language*, 1992, pp. 212–217.

[9] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, vol. abs/1408.5882, 2014.

[10] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014.

[11] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012.

[12] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[13] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005.

[14] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 168–177.