Improved Neural Network-based Multi-label Classification with Better Initialization Leveraging Label Co-occurrence

Gakuto Kurata
IBM Research
gakuto@jp.ibm.com

Bing Xiang
IBM Watson
bingxia@us.ibm.com

Bowen Zhou
IBM Watson
zhou@us.ibm.com

Abstract

In a multi-label text classification task, in which multiple labels can be assigned to one text, label co-occurrence itself is informative. We propose a novel neural network initialization method to treat some of the neurons in the final hidden layer as dedicated neurons for each pattern of label co-occurrence. These dedicated neurons are initialized to connect to the corresponding co-occurring labels with stronger weights than to others. In experiments with a natural language query classification task, which requires multi-label classification, our initialization method improved classification accuracy without any computational overhead in training and evaluation.

1 Introduction

In multi-label text classification, one text can be associated with multiple labels (*label co-occurrence*) (Zhang and Zhou, 2014). Since label co-occurrence itself contains information, we would like to leverage the label co-occurrence to improve multi-label classification using a neural network (NN). We propose a novel NN initialization method that treats some of the neurons in the final hidden layer as *dedicated neurons* for each pattern of label co-occurrence. These dedicated neurons are initialized to connect to the corresponding co-occurring labels with stronger weights than to others. While initialization of an NN is an important research topic (Glorot and Bengio, 2010; Sutskever et al., 2013; Le et al., 2015), to the best of our knowledge, there has been no attempt to leverage label co-occurrence for NN initialization.

To validate our proposed method, we focus on multi-label Natural Language Query (NLQ) classification in a document retrieval system in which users input queries in natural language and the system returns documents that contain answers to the queries. For NLQ classification, we first train a model from training data that contains pairs of queries and corresponding one or more than one document labels, and then predict the appropriate document labels for new queries with the trained model.

Through experiments with a real-world document retrieval system and publicly available multi-label data set, simply and directly embedding label co-occurrence information into an NN with our proposed method improved accuracy of NLQ classification.

2 Related Work

Along with the recent success in NNs (Collobert et al., 2011; Kim, 2014), NN-based multi-label classification has been proposed. An NN for NLQ classification needs to accept queries with variable length and output their labels. Figure 1 shows a typical NN architecture (Collobert et al., 2011). This NN first transforms words in the input query into word embeddings (Mikolov et al., 2013), then applies Convolutional Neural Network (CNN) and Max-pooling over time to extract fixed-length feature vectors, and feed them into the output layer to predict the label for the query (Collobert and Weston, 2008; Collobert et al., 2011; Yih et al., 2014). To take care of multi-labels, label co-occurrence has been incorporated into loss functions such as *pairwise ranking loss* (Zhang and Zhou, 2006). More recently, Nam et

Proceedings of NAACL-HLT 2016, pages 521–526,
San Diego, California, June 12-17, 2016.

al. (2014) reported that *binary cross entropy* can outperform the pairwise ranking loss by leveraging rectified linear units (ReLUs) for nonlinearity (Nair and Hinton, 2010), *AdaGrad* for optimization (Duchi et al., 2011), and *dropout* for generalization (Srivastava et al., 2014). Considering the training efficiency and superior performance, we used the binary cross entropy as one of the baselines in our experiments in Section 4 in addition to *negative log-likelihood* and *cross entropy*.

Let $\boldsymbol{x}$ denote the feature vector of a query, $\boldsymbol{y}$ be the vector representation of the label, $\boldsymbol{o}$ be the output value of the NN, and Θ be the parameters of the NN. Note that the representation of $\boldsymbol{y}$ differs depending on the loss function. For simplicity in the following explanation, assume that we have a finite set of labels $\Lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\}$ and that a query $\boldsymbol{x}$ has multiple labels $\{\lambda_1, \lambda_4\}$:

Negative Log Probability With minimization of negative log probability, a single label is assumed. To circumvent this limitation, we used *copy transformation* (Tsoumakas et al., 2010) and obtained two training examples $((\boldsymbol{x}, \boldsymbol{y}^{(1)}), (\boldsymbol{x}, \boldsymbol{y}^{(2)}))$, where $\boldsymbol{y}^{(1)} = (1, 0, 0, 0, 0)$ and $\boldsymbol{y}^{(2)} = (0, 0, 0, 1, 0)$. The loss for each example becomes $l(\Theta, (\boldsymbol{x}, \boldsymbol{y}^{(1)})) = -\log(o_1)$ and $l(\Theta, (\boldsymbol{x}, \boldsymbol{y}^{(2)})) = -\log(o_4)$, where softmax activation is used to calculate $\boldsymbol{o}$ in the output layer.

Cross Entropy We assumed multi-labels as probabilistic distribution as $\boldsymbol{y} = (0.5, 0, 0, 0.5, 0)$. The cross entropy loss for the training example $(\boldsymbol{x}, \boldsymbol{y})$ becomes $l(\Theta, (\boldsymbol{x}, \boldsymbol{y})) = -\boldsymbol{y}\log(\boldsymbol{o})$, where softmax activation is used in the output layer.

Binary Cross Entropy As Nam et al. (2014) indicated, minimizing binary cross entropy is superior for handling multi-labels. By representing the target labels as $\boldsymbol{y} = (1, 0, 0, 1, 0)$, the binary cross entropy loss for the training example $(\boldsymbol{x}, \boldsymbol{y})$ becomes $l(\Theta, (\boldsymbol{x}, \boldsymbol{y})) = -\sum_{k=1}^{5}(y_k \log(o_k) + (1 - y_k)\log(1 - o_k))$, where sigmoid activation is used in the output layer.

3 Proposed Method

In this section, we explain our proposed method in detail.

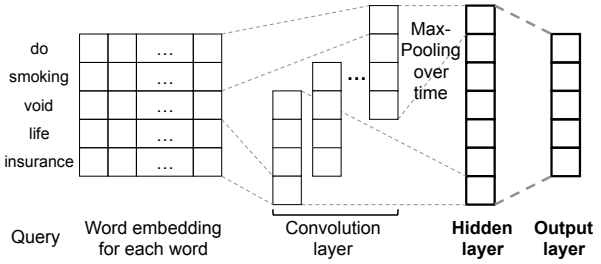

Figure 1: Neural network for NLQ classification. Proposed method is applied to the weight matrix between hidden and output layers as detailed in Figure 2.

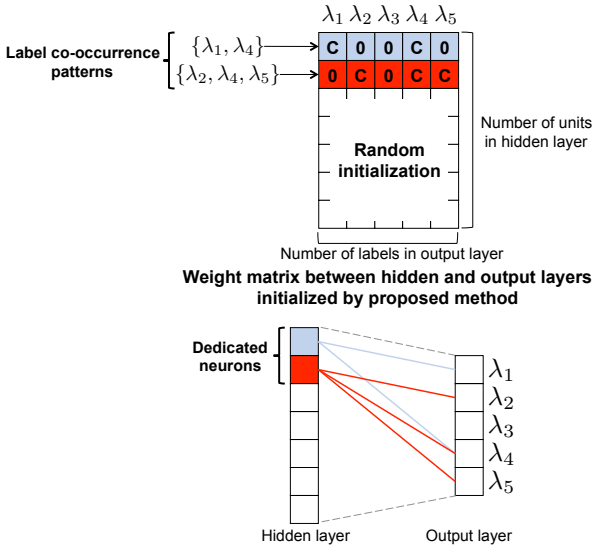

Figure 2: Overview of the proposed method. Label co-occurrence patterns of $\{\lambda_1, \lambda_4\}$ and $\{\lambda_2, \lambda_4, \lambda_5\}$ are used in weight initialization, as shown in the above. This initialization corresponds to preparing dedicated neurons for each label co-occurrence pattern, as shown in the below.

3.1 Weight Initialization Leveraging Label Co-occurrence

We propose an NN initialization method to treat some of the neurons in the final hidden layer as dedicated neurons for each pattern of label co-occurrence. These dedicated neurons simultaneously activate the co-occurring labels. Figure 2 shows the key idea of the proposed method. We first investigate the training data and list up patterns of label co-occurrence. Then, for each pattern of label co-occurrence, we initialize a matrix row so that the columns corresponding to the co-occurring labels have a constant weight C and the other columns

Loss Function	1-best Accuracy			Recall@5			Full Accuracy		
Negative Log Likelihood	49.75	→	51.27	69.80	→	71.07	47.03	→	48.65
Cross Entropy	50.51	→	**52.54**	71.32	→	**72.08**	46.96	→	**48.71**
Binary Cross Entropy	49.75	→	50.51	70.81	→	71.32	48.09	→	48.34

Table 1: 1-best accuracy, recall@5, and full accuracy for evaluation data using different loss functions (Random initialization → Proposed initialization). [%]

have a weight of 0, as shown in Figure 2 (above). Note that the remaining rows that are not associated with the pattern of label co-occurrence are randomly initialized. This initialization is equivalent to treating some of the neurons in the final hidden layer as dedicated neurons for each pattern of label co-occurrence, where the dedicated neurons have connections to the corresponding co-occurring labels with an initialized weight C and to others with an initialized weight of 0, as shown in Figure 2 (below). Finally, we conduct normal back-propagation using one of the loss functions, as discussed in the previous section. Note that all the connection weights in the NN including the connection weights between the dedicated neurons and all labels are updated through back-propagation.

Since (1) computation of proposed initialization itself is negligible and (2) computation of back-propagation and the architecture of NN does not change with or without the proposed initialization, our proposed method does not increase computation in training and evaluation.

3.2 Weight Setting for Dedicated Neurons

For the weight value C for initialization, we used the upper bound UB of the normalized initialization (Glorot and Bengio, 2010), which is determined by the number of units in the final hidden layer n_h and output layer n_c as $UB = \frac{\sqrt{6}}{\sqrt{n_h+n_o}}$. Additionally, we changed this value in accordance with the frequency of the label co-occurrence patterns in the training data. The background idea is that the patterns of label co-occurrence that appear frequently (i.e., the number of queries with this pattern of label co-occurrence is large) are more important than less frequent patterns. Assuming that a specific pattern of label co-occurrence appears in the training data f times, we try $f \times UB$ and $\sqrt{f} \times UB$ for initialization to emphasize this pattern.

C	1-best	Recall@5	Full
—	50.51	71.32	46.96
UB	52.54	**72.08**	48.71
$f \times UB$	51.52	70.81	48.39
$\sqrt{f} \times UB$	**53.55**	**72.08**	**50.04**

Table 2: 1-best accuracy, recall@5, and full accuracy for evaluation data with changing initialization value C. [%]

4 Experiments

We conducted experiments with the real-world NLQ classification data and the publicly available data to confirm the advantage of the proposed method.

4.1 Real-world NLQ classification Data

Experimental Setup We used NLQs for a document retrieval system in the insurance domain for the experiments. Users of the system input queries in natural language, and the system returns the labels of the documents that contain answers. We used $3,133$ queries for training and 394 queries for evaluation, $1,695$ and 158 of which had multiple labels, respectively. The number of unique document labels assigned to the training data was 526.

We used the NN shown in Figure 1. The dimension of word embedding was 100, number of kernels for the CNN was $1,000$, which means $1,000$ units exist in the final hidden layer on top of Max-pooling over time, and number of output units was 526. We used this NN configuration in common for all the experiments. The word embedding was pre-trained with the *skip-gram* model of *word2vec* using the dumped English *Wikipedia* data and the documents of the target insurance domain (Mikolov et al., 2013). The NN except the word embedding layer was randomly initialized in accordance with the normalized initialization (Glorot and Bengio, 2010). We used the ReLU for nonlinearity, AdaGrad for optimization, and dropout for generalization. We fixed

Loss Function	# Survived Neurons (# dedicated neurons:252)	Weights-Dedicated [Mean / Variance]	Weights-All [Mean / Variance]
Negative Log Likelihood	194	0.251 / 0.004	-0.024 / 0.023
Cross Entropy	197	0.267 / 0.005	-0.017 / 0.021
Binary Cross Entropy	168	0.279 / 0.015	-0.007 / 0.011

Table 3: Investigation of neural network after back-propagation training.

the number of training epochs to $1,000$[1].

For the proposed method, we investigated the $1,695$ queries with multiple labels in the training data and found 252 patterns of label co-occurrence. We then embedded this information in a $1,000 \times 526$ weight matrix between the final hidden and output layers. In other words, we treated 252 neurons in the final hidden layer as dedicated neurons in weight initialization.

For the hyper-parameter settings, we first tuned the hyper-parameters including L2-regularization and learning rate so that the accuracy of the baseline system with random initialization was maximized. For the proposed initialization, we used the same hyper-parameters obtained in the former tuning.

We used three evaluation metrics that are closely related to the usability of the document retrieval system: (1) *1-best accuracy* judges if the 1-best result of a system is included in the correct labels[2]. (2) *Recall@5* judges if the 5-best results of a system contain at least one of the correct labels. (3) *Full accuracy* investigates the j-best results of a system and judges if they match the correct labels when j labels are assigned to the query [3].

Different Loss Functions Table 1 shows the experimental results using three different loss functions. Comparing the values to the left of the arrows, which did not use the proposed initialization, superiority of binary cross entropy (Nam et al., 2014) was confirmed in full accuracy, while cross entropy was the best in 1-best accuracy in this experiment. As shown to the right of the arrows, we obtained improvement for all loss functions with every evaluation metric with the proposed method. Overall, cross entropy training with the proposed initialization achieved the best in all three metrics, where 1-best accuracy improvement from 50.51% to 52.54% was statistically significant ($p < 0.05$).

Different Weight Initialization Table 2 shows the results of emphasizing the frequent patterns of label co-occurrence. We used the cross entropy loss function, which was the best in the previous experiments. Using $\sqrt{f} \times UB$ yielded further improvement in 1-best accuracy and full accuracy, though using $f \times UB$ deteriorated in all metrics compared with UB. This suggests that there is room for improvement if we can appropriately emphasize frequent patterns of label co-occurrence.

Analysis on Trained Neural Network We investigated if the dedicated neurons for patterns of label co-occurrences still simultaneously activate the corresponding labels after back-propagation. Table 3 shows the analysis on the NNs trained in the experiments for Table 1. In the *# Survived Neurons* columns, we investigated if the dedicated neurons initialized for the pattern of k-label co-occurrence still had the k largest weights to the corresponding k labels after back-propagation. Large portions of dedicated neurons "survived" after back-propagation. In the *Weights* columns, we calculated the mean of the connection weights between the dedicated neurons and corresponding co-occurring labels and compared them with the mean of all connections in this weight matrix. The trained weights for the connections between the dedicated neurons and corresponding co-occurring labels (*Weights-Dedicated*) were much stronger than the average weights (*Weights-All*). This analysis suggests that the proposed initialization yields dedicated neurons

[1]We confirmed that NN training sufficiently saturated after $1,000$ epochs in preliminary experiments. We also compared the best accuracy achieved in $1,000$ epochs for all experimental conditions and confirmed that the same improvement was achieved with the proposed method.

[2]This metric is comparable with *One-error* (Tsoumakas et al., 2010) by 1-best Accuracy = 100 - One-error.

[3]If a query has three labels, the system needs to return 3-best results that contain the three correct labels of the query to obtain 100% full accuracy.

that simultaneously activate the co-occurring labels even after back-propagation.

There can be an overlap in label co-occurrence patterns. One typical case is "A, B" and "A, B, C", and another case is "D, E", "F, G", and "D, E, F, G". While we prepared the dedicated neurons for each co-occurrence pattern before back-propagation, some overlapped co-occurrences might be explained by the superset or combination of subsets after back-propagation. Table 3 suggests that some of the dedicated neurons did not survive after back-propagation. We confirmed that about half of the label co-occurrence patterns whose dedicated neurons did not survive were covered by the patterns whose neurons survived. "Cover" means that if a neuron for "A, B" did not survive, a neuron for "A, B, C" survived, or if a neuron for "D, E, F, G" did not survive, neurons for "D, E" and "F, G" survived. If we change the network structure by connecting the dedicated neurons only to the corresponding units or preparing the special output units for co-occurring labels (label powerset (Read, 2008)), this flexibility might be lost.

4.2 Publicly Available Data

We used multi-label topic categorization data (*RCV1-v2*) (Lewis et al., 2004) to validate our method. We used the same label assignment and the same training and evaluation data partition with the *LYRL2004* split (Lewis et al., 2004) where $23,149$ training texts and $781,265$ evaluation texts with 103 topic labels are available. We used the bag-of-word (BoW) feature for each text prepared by Chang and Lin (2011) whose dimension was $47,236$ and constructed a feed-forward NN that has an input layer that accepts the BoW feature, hidden layer of $2,000$ units, and output layer of 103 output units with the cross entropy loss function. By embedding the label co-occurrence information between the hidden and output layers with the initial weights set to UB, which corresponded to treating 758 neurons out of $2,000$ hidden units as the dedicated neurons, we improved 1-best accuracy of topic label classification from 93.95% to 94.60%, which was statistically significant ($p < 0.001$).

To the best of our knowledge, 1-best accuracy of 94.18% (5.82% one-error)[4] (Rubin et al., 2012) was the best published result with using the standard LYRL2004 split of RCV1-v2. Our proposed method has advantages in a sufficiently competitive setup.

5 Conclusion

We proposed an NN initialization method to leverage label co-occurrence information. Through experiments using the data of a real-world document retrieval system and publicly available data, we confirmed that our proposed method improved NLQ classification accuracy. The advantage of the proposed method also includes no computational overhead during training and evaluation.

When we have large training data, the number of label co-occurrence patterns can be larger than that of hidden units. In such a case, one option is to select an appropriate set of label co-occurrence patterns with certain criteria such as the frequency in the training data. Another option is to make a larger weight matrix using all patterns and then to reduce its dimension with such as Principal Component Analysis (PCA) in advance of NN training. Our future work also includes setting the initialization weight in a more sophisticated way and combining the proposed method with other NN-based methods (Kim, 2014; Johnson and Zhang, 2015).

Acknowledgments

We would like to show our gratitude to Dr. Ramesh M. Nallapati of IBM Watson for supporting the experiments. We are grateful to Dr. Yuta Tsuboi, Dr. Ryuki Tachibana, and Mr. Nobuyasu Itoh of IBM Research - Tokyo for the fruitful discussion and their comments on this and earlier versions of the paper. We thank the anonymous reviewers for their valuable comments.

[4] Randomly selected $75,000$ texts were used for evaluation, but the results on this subset were confirmed to be almost identical to those on the full evaluation texts (Rubin et al., 2012).

References

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. ICML*, pages 160–167. ACM.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proc. AISTATS*, pages 249–256.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *Proc. NAACL-HLT*, pages 103–112.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proc. EMNLP*, pages 1746–1751.

Quoc V Le, Navdeep Jaitly, and Geoffrey E Hinton. 2015. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proc. ICLR*.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proc. ICML*, pages 807–814.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. Large-scale multi-label text classification —– revisiting neural networks. In *Proc. ECML-PKDD*, pages 437–452.

Jesse Read. 2008. A pruned problem transformation method for multi-label classification. In *Proc. NZCSRS*, pages 143–150.

Timothy N Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. 2012. Statistical topic models for multi-label document classification. *Machine learning*, 88(1-2):157–208.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proc. ICML*, pages 1139–1147.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2010. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *Proc. ACL*, pages 643–648.

Min-Ling Zhang and Zhi-Hua Zhou. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351.

Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837.