Order-Free RNN with Visual Attention for Multi-Label Classification

Shang-Fu Chen,[1*] **Yi-Chen Chen,**[2*] **Chih-Kuan Yeh,**[3] **Yu-Chiang Frank Wang**[1,2]

[1]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
[2]Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan
[3]Machine Learning Department, Carnegie Mellon University, Pittsburgh, USA
Email: {b02901030, r06942069, ycwang}@ntu.edu.tw, cjyeh@cs.cmu.edu

Abstract

We propose a recurrent neural network (RNN) based model for image multi-label classification. Our model uniquely integrates and learning of visual attention and Long Short Term Memory (LSTM) layers, which jointly learns the labels of interest and their co-occurrences, while the associated image regions are visually attended. Different from existing approaches utilize either model in their network architectures, training of our model does not require pre-defined label orders. Moreover, a robust inference process is introduced so that prediction errors would not propagate and thus affect the performance. Our experiments on NUS-WISE and MS-COCO datasets confirm the design of our network and its effectiveness in solving multi-label classification problems.

Introduction

Multi-label classification has been an important and practical research topic, since it needs to assign more than one label to each observed instance. From machine learning, data mining, and computer vision, a variety of applications benefit from the development and success of multi-label classification algorithms (Zhang and Zhou 2014; Boutell et al. 2004; Schapire and Singer 2000; Godbole and Sarawagi 2004; Lin et al. 2014; Kang et al. 2016b; 2016a; Boutell et al. 2004; Shao et al. 2016). A fundamental and challenging issue for multi-label classification is to identify and recover the co-occurrence of multiple labels, so that satisfactory prediction accuracy can be expected.

Recently, development of deep convolutional neural networks (CNN) (Krizhevsky, Sutskever, and Hinton 2012; Szegedy et al. 2015; Simonyan and Zisserman 2014; He et al. 2016) have made a remarkable progress in several research fields. Due to its ability of representation learning with prediction guarantees, CNNs contribute to the recent success in image classification tasks and beyond (Deng et al. 2009; Fei-Fei, Fergus, and Perona 2007; Griffin, Holub, and Perona 2007). Despite its effectiveness, how to extend CNNs for solving multi-label classification problems is still a research direction to explore.

---

*- indicates equal contribution.

While a number of research works (Zhang and Zhou 2006; Nam et al. 2014; Gong et al. 2013; Wei et al. 2014; Wang et al. 2016) start to advance the CNN architecture for multi-label classification, CNN-RNN (Wang et al. 2016) embeds image and semantic structures by projecting both features into a joint embedding space. By further utilizing the component of Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997), a recurrent neural network (RNN) structure is introduced to memorize long-term label dependency. As a result, CNN-RNN exhibits promising multi-label classification performance with cross-label correlation implicitly preserved.

Unfortunately, the above frameworks suffer from the following three different problems. First, due to the use of LSTM, a pre-defined label order is required during training. Take (Wang et al. 2016) for example, its label order is determined by the frequencies of labels observed from the training data. In practice, such pre-defined orders of label prediction might not reflect proper label dependency. For example, based on the number of label occurrences, one might obtain the label sequence as {*sea, sun, fish*}. However, it is obvious that *fish* is less semantically relevant to *sun* than *sea*. For better learning and prediction of such labels, the order of {*sea, fish, sun*} should be considered. On the other hand, (Jin and Nakayama 2016) consider four experimental settings with different label orders: alphabetical order, random order, frequency-first order and rare-first order (note that rare-first is exactly the reverse of frequency-first). It is concluded in (Jin and Nakayama 2016) that the rare-first order results in the best performance. Later we will conduct thorough experiments for verification, and show that orders automatically learned by our model would be desirable.

The second concern of the above methods is that, labels of objects which are in smaller scales/sizes in images would often be more difficult to be recovered. As a possible solution, attention map (Xu et al. 2015) has been widely considered in image captioning (Xu et al. 2015), image question answering (Yang et al. 2016b), and segmentation (Hong et al. 2016). Extracted by different kernels from a certain convolutional layer in CNN, the corresponding feature maps contain rich information of different patterns from the input image. By further attending on such feature maps, the resulting attention map is able to identify important components or objects in an image. By exploiting the label co-occurrence be-
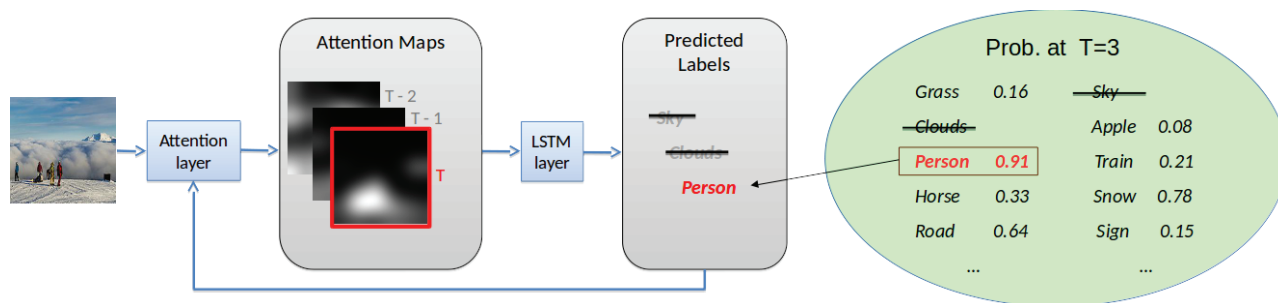
Figure 1: Illustration of our proposed model for multi-label classification. Note that the joint learning of attention and LSTM layers allows us to identify the label dependency without using any predetermined label order, while the corresponding image regions of interest can be attended to accordingly.

tween the associated objects in different scales or patterns, the above problem can be properly alleviated. However, this technique could not be easily applied to RNN-based methods for multi-label problems. As noted above, such methods determine the label order based on the occurrence frequency. For example, the class *person* may appear more often than *horse* in an image collection, and thus the label sequence would be derived as {*person, horse*}. Even if the image region of *horse* is typically larger than that of *person*, it might not assist in identifying the rider on its back (i.e., requiring the prediction order as {*horse, person*}).

Thirdly, inconsistency between training and testing procedures would often be undesirable for solving multi-label classification tasks. To be more precise, during the training phase, the labels to be produced at each recurrent layer is selected from the ground truth list during the training phase; however, the labels to be predicted during testing are selected from the entire label set. In other words, if a label is incorrectly predicted during a time step during prediction, such an error would propagate during the recurrent process and thus affect the results.

To resolve the above problems, we present a novel deep learning framework of visually attended RNN, which consists of visual attention and LSTM models as shown in Fig. 1. In particular, we propose a confidence-ranked LSTM which reflects the label dependency with the introduced visual attention model. Our joint learning framework with the introduced attention model allows us to identify the regions of interest associated with each label. As a result, the order of labels can be automatically learned without any prior knowledge or assumption. As verified later in the experiments, even the objects are presented in small scales in the input image, the corresponding image regions would still be visually attended. More importantly, as detailed later in Sect. III, our network architecture can be applied to both training and testing, and thus the aforementioned inconsistency issue is addressed.

The main contributions of this paper are listed below:

- Without pre-determining the label order for prediction, our method is able to sequentially learn the label dependency using the introduced LSTM model.

- The introduced attention model in our architecture allows

us to focus on image regions of interests associated with each label, so that improved prediction can be expected even if the objects are in smaller sizes.

- By jointly learning attention and LSTM models in a unified network architecture, our model performs favorably against state-of-the-art deep learning approaches on multi-label classification, even if the ground truth label might not be correctly presented during training.

Related Work

We first review the development of multi-label classification approaches. Intuitively, the simplest way to deal with multi-label classification problems is to decompose them into multiple binary classification tasks (Tsoumakas and Katakis 2006). Despite its simplicity, such techniques cannot identify the relationship between labels.

To learn the interdependency between labels for multi-label classification, approaches based on classifier chains (Read et al. 2011) were proposed, which capture label dependency by conditional product of probabilities. However, in addition to high computation cost when dealing with a larger number of labels, classifier chains have limited ability to capture the high order correlations between labels. On the other hand, probabilistic graphical model based methods (Li, Zhao, and Guo 2014; Li et al. 2016) learn label dependencies with graphical structure, and latent space methods (Yeh et al. 2017; Bhatia et al. 2015) choose to project features and labels into a common latent space. Approaches like (Yang et al. 2016a) further utilize additional information like bounding box annotations for learning their models.

With the recent progress of neural networks and deep learning, BP-MLL (Zhang and Zhou 2006) is among the first to utilize neural network architectures to solve multi-label classification. It views each output node as a binary classification task, and relies on the architecture and loss function to exploit the dependency across labels. It was later extended by (Nam et al. 2014) with state-of-the-art learning techniques such as dropout.

Furthermore, state-of-the-art DNN based multi-label algorithms have proposed different loss functions or architectures (Gong et al. 2013; Wei et al. 2014; Hu et al. 2016). For example, Gong et al. (Gong et al. 2013) design a rank-based
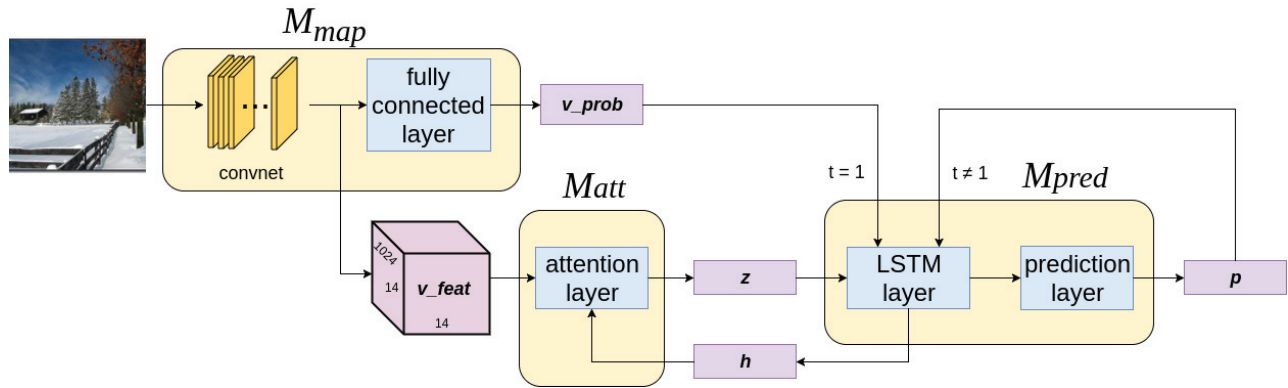
Figure 2: Architecture of our proposed network architecture for multi-label classification. Note that $\mathbf{M}_{map}$, $\mathbf{M}_{att}$, and $\mathbf{M}_{pred}$ indicate the layers for feature mapping, attention, and label prediction, respectively. v_{feat} is a set of feature maps extracted from $\mathbf{M}_{map}$, and the vector output v_{prob} represents the preliminary label prediction of $\mathbf{M}_{map}$ to initiate LSTM prediction. z and h are the attention context vector and the LSTM hidden state, respectively. Finally, p denotes the vector output indicating the label probability, updated at every time step.

loss and compensate those with lowest ranks ones, Wei et al. (Wei et al. 2014) generate multi-label candidates on several grids and combine results with max-pooling, and Hu et al. propose structured inference NN (Hu et al. 2016), which uses concept layers modeled with label graphs.

Recurrent neural networks (RNN) is a type of NN structure, which is able to learn the sequential connections and internal states. When RNN has been successfully applied to sequentially learn and predict multiple labels of the data, it typically requires a large number of parameters to observe the above association. Nevertheless RNN with LSTM (Hochreiter and Schmidhuber 1997) is an effective method to exploit label correlation. Researches in different fields also apply RNNs to deal with sequential prediction tasks which utilize the long-term dependency in a sequence, such as image captioning (Mao et al. 2014), speech recognition (Graves, Mohamed, and Hinton 2013), language modeling (Sundermeyer, Schlüter, and Ney 2012), and word embedding learning (Le and Zuidema 2015). Among multi-label classification, CNN-RNN (Wang et al. 2016) is a representative work with promising performance. However, CNN-RNN requires a pre-defined label order for learning, and its limitation to recognize labels corresponding to objects in smaller sizes would be the major concern.

Our Proposed Method

We first define the goal of the task in this paper. Let $\mathbf{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N} = \{\mathbf{X}, \mathbf{Y}\}$ denote the training data, where $\mathbf{X} \in \mathbb{R}^{d \times N}$ indicates a set of N training instances in a d dimensional space. The matrix $\mathbf{Y} \in \mathbb{R}^{\mathcal{C} \times N}$ indicates the associated multi-label matrix, where $\mathcal{C}$ is the number of labels of interest. In other words, each dimension in $\mathbf{y}_c$ is a binary value indicating whether $\mathbf{x}_i$ belongs to the corresponded label c. For multi-label classification, the goal is to predict the multi-label vector $\hat{\mathbf{y}}$ for a test input $\hat{\mathbf{x}}$.

A Brief Review of CNN-RNN

CNN-RNN (Wang et al. 2016) is a recent deep learning based model for multi-label classification. Since our method can be viewed as an extension, it is necessary to briefly review this model and explain the potential limitations.

As noted earlier, exploiting label dependency would be the key to multi-label classification. Among the first CNN works for tackling this issue, CNN-RNN is composed of a CNN feature mapping layer and a Long Short-Term Memory (LSTM) inference layer. While such an architecture jointly projects the input image and its label vector into a common latent space, the LSTM particularly recovers the correlation between labels. As a result, outputs of multiple labels can be produced at the prediction layer via nearest neighbor search.

Despite its promising performance, CNN-RNN requires a predefined label order for training their models. In addition to the lack of robustness in learning optimal label orders, as confirmed in (Wang et al. 2016), labels of objects in smaller sizes would be difficult to predict if their visual attention information is not properly utilized. Therefore, how to introduce the flexibility in learning optimal label order while jointly exploiting the associated visual information would be the focuses of our proposed work.

Order-Free RNN with Visual Attention

As illustrated in Fig. 2, our proposed model for multi-label classification has three major components: feature mapping layer $\mathbf{M}_{map}$, attention layer $\mathbf{M}_{att}$, and LSTM inference layer $\mathbf{M}_{pred}$. The feature mapping layer $\mathbf{M}_{map}$ extracts the visual features from the input image $\mathbf{x}_i$ using a pre-trained CNN model. With the attention layer $\mathbf{M}_{att}$, we would observe a set of feature maps v_{feat}, in which each map is learned to describe the corresponding layer of image semantic information. The output of $\mathbf{M}_{att}$ then goes through the LSTM inference process via $\mathbf{M}_{pred}$, followed by a final prediction layer for producing the label outputs.

During the LSTM inference process, the hidden state vector $\mathbf{h}$ would update the attention layer $\mathbf{M}_{att}$ with the label inference from the previous time step, guiding the network to visually attend the next region of interest in the input image. Thus, such network designs allow one to exploit *label correlation* using the associated visual information. As a result, the optimal order of label sequences can be automatically observed. In the following subsections, we will detail each component of our proposed model.

Feature Mapping Layer $\mathbf{M}_{map}$ The feature mapping layer $\mathbf{M}_{map}$ first extracts visual features v_{feat} by pre-trained CNN models. Following the design in (Liu et al. 2016), we add a fully-connected layer with the output dimension of c after the convolutional layers, which produces the predicted probability $\mathbf{v}_{prob}$ for each label as an additional feature vector. Therefore, the CNN probability outputs can be viewed as preliminary label prediction.

With the ground truth labels given during training (note that positive labels as 1 and negative ones as 0), the learning of $\mathbf{M}_{map}$ would update the parameters of the fully-connected layer via observing log-likelihood cross-entropy, while the parameters of the pre-trained CNN remain fixed. By concatenating m feature maps of dimension k in v_{feat}, we convert v_{feat} into a single input vector learning visual attention. As a result, the output probability vector of $\mathbf{M}_{map}$ can be expressed as follows:

$$\mathbf{V} = [\mathbf{V}_{feat}, \mathbf{v}_{prob}] \tag{1}$$

$$\mathbf{V}_{feat} = [\mathbf{v}_1, ..., \mathbf{v}_m], \mathbf{v}_i \in \mathbb{R}^k \tag{2}$$

$$\mathbf{v}_{prob} \in [0, 1]^c. \tag{3}$$

Attention Layer $\mathbf{M}_{att}$ When predicting multiple labels from an input image, one might suffer from the fact that labels of objects in smaller sizes are not properly identified. For example, *person* typically occupies a significant portion of an input image, while *birds* might appear in smaller sizes and around the corners.

In order to alleviate this problem, we introduce an attention layer $\mathbf{M}_{att}$ to our proposed architecture, with the goal of focusing on proper image regions when predicting the associated labels. Inspired by Xu et al. (Xu et al. 2015), who advocated a soft attention-based image caption generator, we advance the same network component in our framework. For multi-label classification, this allows us to focus and describe the image regions of interest during prediction, while implicitly exploiting the label co-occurrence information. In our proposed framework, this attention layer would generate a context vector consisting of weights for each feature map, so that the attended image region can be obtained during each iteration. Later we will also explain that, with such network designs, we can observe optimal label order when learning our RNN-based multi-label classification model.

Following the structure of multi-layer perceptron (Xu et al. 2015), our attention layer $\mathbf{M}_{att}$ is conditioned on the previous hidden state $\mathbf{h}_{t-1}$. For each $\mathbf{v}_i$ in 2, the attention layer

---

Algorithm 1: Training of Our Proposed Model

Input: Feature maps $\mathbf{V}_{feat} = [\mathbf{v}_1,...,\mathbf{v}_m]$ and label vector $\mathbf{y}$ of an image

Parameter: Resnet fully-connected layer θ_R, attention layer θ_a, LSTM layer θ_L and prediction layer θ_p, iteration number $iter$

Output: Soft confidence vector $\tilde{\mathbf{y}}$

Randomly initialize parameters
Train θ_R by log-likelihood cross-entropy loss and obtain $\mathbf{v}_{prob}$
repeat
    for $t = 1; t \leq iter; t{+}{+}$ **do**
        Obtain the context vector $\mathbf{z}_t$ by (4) and (5)
        Obtain the hidden state $\mathbf{h}_t$ by (6)
        Obtain the soft confidence vector $\mathbf{p}_t$ by (7)
        Obtain the hard predicted label vector $\tilde{\mathbf{y}}_\mathbf{t}$ by (9)
        Update the candidate pool by (10)
        Compute the log-likelihood cross-entropy between $\mathbf{v}_{prob}$ and $\mathbf{y}$
        Perform gradient descent on θ_a, θ_L and θ_p
        $\mathbf{v}_{pred} = \mathbf{p}_t$
until θ_a, θ_L and θ_p *converge*

---

generates a weight α_i, $\alpha_i \in [0, 1]$, which represents the importance weight of feature i in the input image, and predicts the label at this time step. To be more specific, we have:

$$\epsilon_{i,t} = f_{att}(\mathbf{v}_i, \mathbf{h}_{t-1})$$
$$\alpha_{i,t} = \frac{e^{\epsilon_{i,t}}}{\sum_{j=1}^{m} e^{\epsilon_{j,t}}}, \tag{4}$$

where f_{att} is the same as the model in (Xu et al. 2015), and $\mathbf{h}_{t-1}$ would be detailed later in Sec. .

With $\alpha_{i,t}$, we derive the context vector $\mathbf{z}_t$ with the soft attention mechanism:

$$\mathbf{z}_t = \sum_{i=1}^{m} \alpha_{i,t}\mathbf{v}_i. \tag{5}$$

Later, our experiments will visualize and evaluate the contribution of our attention model for multi-label classification.

Confidence-Ranked LSTM $\mathbf{M}_{pred}$ As an extension of recurrent neural network (RNN), LSTM additionally consists of three gate neurons: forget, input, and output gates. The forget gate is to learn proper weights for erasing the memory cell, the input gate is learned to describe the input data, while the output gate aims to control how the memory should be omitted.

In order to exploit and capture the dependency between labels, the LSTM model $\mathbf{M}_{pred}$ in our network architecture needs to identify which label would exhibit a high confidence at each time step. Thus, we concatenate the soft confidence vector $\mathbf{v}_{pred}$ from the previous time step (note that

$\mathbf{v}_{pred} = \mathbf{v}_{prob}$ when t=1, and $\mathbf{v}_{pred} = \mathbf{p}_{t-1}$ otherwise), the context vector $\mathbf{z}_t$ and the previous predicted hard label vector $\tilde{\mathbf{y}}_{t-1}$ for deriving the current hidden state vector $\mathbf{h}_t$. This state vector is thus controlled by the aforementioned three gate components. By observing the long-term dependency between labels via the above structure, we can exploit and utilize the resulting label correlation for improved multi-label classification.

We note that, to predict multi-label outputs using LSTM, we pass $\mathbf{h}_t$ through an additional prediction layer consisting of two fully-connected layers, and result in a soft confidence vector $\mathbf{p}_t \in \mathbb{R}^c$ at time t. The hard predicted label $l_t = argmax \ \mathbf{p}_t$ indicates the most confident class at the time step t, which is then appended to the hard predicted label vector $\tilde{\mathbf{y}}_t$. In the testing phase, by collecting l till the ultimate condition, which will be described later in the next section, the final predicted multi-label vector $\tilde{\mathbf{y}}$ can be obtained.

More specifically, we calculate:

$$\mathbf{h}_t = f_{LSTM}(\mathbf{v}_{pred}, \mathbf{z}_t, \tilde{\mathbf{y}}_{t-1}, \mathbf{h}_{t-1}), \qquad (6)$$

where f_{LSTM} denotes the LSTM model. In order to predict $\mathbf{p}_t$, we have:

$$\mathbf{p}_t = f_{pred}(\mathbf{v}_{pred}, \mathbf{z}_t, \tilde{\mathbf{y}}_{t-1}, \mathbf{h}_t), \qquad (7)$$

On the other hand, the cross-entropy loss function to minimize at the output layer at time t is:

$$\mathbf{L}_t = -\sum_{i=1}^{c} y_i log(\sigma(p_{t,i})) + (1 - y_i)log(1 - \sigma(p_{t,i})), \quad (8)$$

where $y_i \in \mathbf{y}$, $p_{t,i} \in \mathbf{p}_t$, and σ is the sigmoid function.

It is worth noting that, the main difficulty of applying LSTM for multi-label classification is its requirement of the ground truth label order during the training process. By simply calculating the cross-entropy between the confidence vector $\mathbf{p}_t$ and the ground truth multi-label vector $\mathbf{y}$, one would not be able to define the order of label prediction for learning LSTM. Moreover, it would be desirable if the label order would reflect semantic dependency between labels presented in training image data.

With the above observation, we view our $\mathbf{M}_{pred}$ in the proposed network architecture as *confidence-ranked LSTM*. Once the previous soft confidence vector $\mathbf{p}_{t-1}$ and hard predicted label vector $\tilde{\mathbf{y}}_{t-1}$ are produced, our model would update $\mathbf{h}_t$ and the attention layer $\mathbf{M}_{att}$. As a result, we will be able to produce $\mathbf{p}_t$ accordingly. In other words, our model achieves the visually attention of objects of semantic interest in the input image, which does not require one to pre-define any specific label order. Therefore, unlike previous works like (Wang et al. 2016), the training of our model does not require the selection of ground truth labels in a predetermined order. Instead, we calculate the loss by comparing the soft confidence vector with the ground truth label vector directly. With our visual attention plus LSTM components, the training process would be consistent with the testing stage. Since the above process relies on visual semantic information for multi-label prediction, one of the major advantages of our model that possible error propagation problems when applying RNN-based approaches can be alleviated.

Order-Free Training and Testing

Training We now explain how our model achieves order-free learning and prediction. As shown in Fig. 2, our network design produces outputs of labels $[l_1, ...l_T]$ at time T, where each l_i denotes the label with the highest confidence at the i-th time step. To avoid duplicate label outputs at different time steps, we apply the concept of candidate label pool as follows.

To initialize the inference process for multi-label learning using our model, the candidate label pool would simply be $\mathcal{C}$ containing all labels. At each time step, the most confident label l_t would be selected from the candidate pool, and thus this candidate pool will be updated by removing l_t from it. More specifically, for l_t, we denote it as:

$$l_t = arg \max_{\mathcal{C}'_t} \ \mathbf{p}_t,$$
$$\tilde{\mathbf{y}}_t = \tilde{\mathbf{y}}_{t-1} + l_t \qquad (9)$$

$$\mathcal{C}'_t = \mathcal{C}'_{t-1} - \{l_{t-1}\}. \qquad (10)$$

where $\mathcal{C}'_0$ denotes the full set of labels $\mathcal{C}$, and $\mathcal{C}'_t$ is the set of candidate labels to be predicted at time t. From the above label update process, the cardinal of the candidate label set would be subtracted by one at each time step.

Testing We note that, the labels to be predicted during the testing stage can be obtained sequentially using the learned model. However, even with the introduction of the attention layers, prediction error at a time step would be propagated and significantly degrade the prediction performance.

Inspired by (Wang et al. 2016), we apply the technique of *beam search* to alleviate the above problem, and thus the predicting process would be more robust to intermediate prediction errors. More precisely, beam search would keep the best-K prediction paths at each time step. At time step $t+1$, it then searches from all $K \times c$ successor paths generated from the K previous paths, updates the path probability for all $K \times c$ successor paths, and maintains the best-K candidates for the following time steps.

In our work, a prediction path represents a sequence of predicted labels with a corresponding path probability, which can be calculated by multiplying the probabilities of all the nodes along the prediction path. At each time step t given a prediction path $[l_1, l_2, \cdots, l_{t-1}]$ and image I, its path probability before predicting l_t is calculated as:

$$P_{path} = P(l_1|I) \times P(l_2|I, l_1) \times \cdots$$
$$\times P(l_{t-1}|l_1, l_2, \cdots, l_{t-2}). \qquad (11)$$

Finally, the prediction process via beam search would terminate under the following two conditions:
1. The probability output of a particular prediction path is below a threshold (which is determined by cross-validation).
2. The length of the prediction path reaches a pre-defined maximum length (which is the largest number of labels in the training set).

Table 1: Evaluation of NUS-WIDE. Note that Macro/Micro P/R/F1 scores are abbreviated as O/C-P/R/F1, respectively. Ours (w/o attention) and Frequency/Rare-first (w/ atten) denote our method with the attention layer removed and using associated pre-defined label orders, respectively.

Method	C-P	C-R	C-F1	O-P	O-R	O-F1
KNN	32.6	19.3	24.3	43.9	53.4	47.6
Softmax	31.7	31.2	31.4	47.8	59.5	53.0
WARP	31.7	35.6	33.5	48.6	60.5	53.9
CNN-RNN	40.5	30.4	34.7	49.9	61.7	55.2
Resnet-baseline	46.5	47.6	47.1	61.6	68.1	64.7
Frequency-first (w/ atten)	48.9	48.7	48.8	62.1	69.4	65.5
Rare-first (w/ atten)	53.9	51.8	52.8	55.1	65.2	59.8
Ours (w/o atten)	60.8	49.5	54.5	68.3	72.4	70.2
Ours	59.4	50.7	**54.7**	69.0	71.4	**70.2**

Table 2: Performance comparisons on MS-COCO. Ours (w/o attention) and Ours Frequency/Rare-first (w/ atten) denote our method with the attention layer removed and using associated pre-defined label orders, respectively.

Method	C-P	C-R	C-F1	O-P	O-R	O-F1
Softmax	59.0	57.0	58.0	60.2	62.1	61.1
WARP	59.3	52.5	55.7	59.8	61.4	60.7
CNN-RNN	66.0	55.6	60.4	69.2	66.4	**67.8**
Resnet-baseline	58.3	49.3	53.4	63.9	58.4	61.0
Frequency-first (w/ atten)	55.8	54.7	55.2	61.4	62.6	62.0
Rare-first (w/ atten)	59.5	56.5	58.0	57.3	56.7	57.0
Ours (w/o atten)	69.9	52.6	60.0	73.4	60.3	66.2
Ours	71.6	54.8	**62.1**	74.2	62.2	**67.7**

Experiments

Implementation

To implement our proposed architecture, we apply a ResNet-152 (He et al. 2016) network trained on Imagenet without fine-tuning, and use the bottom fourth convolution layer for visual feature extraction. We also add a fully-connected layer with dimension of c after the convolutional layer. We employ the Adam optimizer with the learning rate at 0.0003, and the dropout rate at 0.8 for updating f_{pred}. We perform validation on the stopping threshold for beam search. As for the parameters for attention and LSTM models, we follow the settings of (Xu et al. 2015) for implementation.

To evaluate the performance of our method and to perform comparisons with state-of-the-art methods, we report results on the benchmark datasets of NUS-WIDE and MS-COCO as discussed in the following subsections.

NUS-WIDE

NUS-WIDE is a web image dataset which includes 269,648 images with a total of 5,018 tags collected from Flickr. The collected images are further manually labeled into 81 concepts, including objects and scenes. We follow the setting of WARP (Gong et al. 2013) for experiments by removing images without any label, i.e., 150,000 images are considered for training, and the rest for testing.

We compare our result with state-of-the-art NN-based models: *WARP* (Gong et al. 2013) and *CNN-RNN* (Wang et al. 2016). We also also perform several controlled experiments: (1) removing the attention layer, and (2) fixing orders by different methods as suggested by (Jin and Nakayama 2016) during training. Frequency-first indicates the labels are sorted by frequency, from high to low, and rare-first is exactly the reverse of frequency-first. The results are listed in Table 1. From this table, we see that our model performed favorably against baseline and state-of-the art multi-label classification algorithms. This demonstrates the effectiveness of our method in learning proper label ordering for sequential label prediction. Finally, our full model achieved the best

performance, which further supports the exploitation of visually attended regions for improved multi-label classification.

In Fig. 3(a), we present example images with correct label prediction. We see that our model was able to predict labels depending on what it was actually attended to. For example, since 'person' is a frequent label in the dataset, CNN-RNN framework tended to predict it first, because their label order was defined by label occurrence frequency observed during the training stage. In contrast, our model was able to predict animal and horses first, which were actually easier to be predicted based on their visual appearance in the input image. On the other hand, examples of *incorrect* predictions are shown in Fig 3(b). It is worth pointing out that, as can be seen from these results, the prediction results were actually intuitive and reasonable, and the incorrect prediction was due to the noisy ground truth label. From the above observations, it can be successfully verified that our method is able to identify semantic ordering and visually adapt to objects with different sizes, even given noisy or incorrect label data during the training stage.

MS-COCO

MS-COCO is the dataset typically considered for image recognition, segmentation and captioning. The training set consists of 82,783 images with up to 80 annotated object labels. The test set of this experiment utilizes the validation set of MS-COCO (40,504 images), since the ground truth labels of the original test set in MS-COCO are not provided. , In the experiments, we compare our model with the *WARP* (Gong et al. 2013) and *CNN-RNN* (Wang et al. 2016) models in Table 2. It can be seen that the full version of our model achieved performance improvements over the Resnet-based baseline by 4.1% in C-F1 and by 5.6% in O-F1.

In Figures 3(c) and 3(d), we also present example images with correct and incorrect prediction. It is worth noting that, in the upper left example in Fig. 3(c), although the third attention map corresponded to the label prediction of surfboard, it did not properly focus on the object itself. Instead, it took the surrounding image regions into consideration. Combining the information provided by the hidden state, it still successfully predicted the correct label. This illustrates

Figure 3: Examples images with correct label prediction in NUS-WISE (a) and MS-COCO (c), those with incorrect prediction are shown in (b) and (d), respectively. For each image (with ground truth labels noted below), the associated attention maps are presented at the right hand side, showing the regions of interest visually attended to. Note that some incorrect predicted labels (in red) were expected and reasonable due to noisy ground truth labels, while the resulting visual attention maps successfully highlight the attended regions.

the ability of our model to utilize both *local* and *global* information in an image during multi-label prediction.

Conclusion

We proposed a deep learning model for multi-label classification, which consists of a visual attention model and a confidence-ranked LSTM. Unlike existing RNN-based methods requiring predetermined label orders for training, the joint learning of the above components in our proposed architecture allows us to observe proper label sequences with visually attended regions for performance guarantees. In our experiments, we provided quantitative results to support the effectiveness of our method. In addition, we also verified its robustness in label prediction, even if the training data are noisy and incorrectly annotated.

References

Bhatia, K.; Jain, H.; Kar, P.; Varma, M.; and Jain, P. 2015. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*, 730–738.

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern recognition* 37(9):1757–1771.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 248–255. IEEE.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2007. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding* 106(1):59–70.

Godbole, S., and Sarawagi, S. 2004. Discriminative methods for multi-labeled classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 22–30. Springer.

Gong, Y.; Jia, Y.; Leung, T.; Toshev, A.; and Ioffe, S. 2013. Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, 6645–6649. IEEE.

Griffin, G.; Holub, A.; and Perona, P. 2007. Caltech-256 object category dataset.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual

learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hong, S.; Oh, J.; Lee, H.; and Han, B. 2016. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3204–3212.

Hu, H.; Zhou, G.-T.; Deng, Z.; Liao, Z.; and Mori, G. 2016. Learning structured inference neural networks with label relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2960–2968.

Jin, J., and Nakayama, H. 2016. Annotation order matters: Recurrent image annotator for arbitrary length image tagging. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, 2452–2457. IEEE.

Kang, K.; Li, H.; Yan, J.; Zeng, X.; Yang, B.; Xiao, T.; Zhang, C.; Wang, Z.; Wang, R.; Wang, X.; et al. 2016a. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *arXiv preprint arXiv:1604.02532*.

Kang, K.; Ouyang, W.; Li, H.; and Wang, X. 2016b. Object detection from video tubelets with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 817–825.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Le, P., and Zuidema, W. 2015. Compositional distributional semantics with long short term memory. *arXiv preprint arXiv:1503.02510*.

Li, Q.; Qiao, M.; Bian, W.; and Tao, D. 2016. Conditional graphical lasso for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2977–2986.

Li, X.; Zhao, F.; and Guo, Y. 2014. Multi-label image classification with a probabilistic label enhancement model. In *UAI*, volume 1, 3.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 740–755. Springer.

Liu, F.; Xiang, T.; Hospedales, T. M.; Yang, W.; and Sun, C. 2016. Semantic regularisation for recurrent image annotation. *arXiv preprint arXiv:1611.05490*.

Mao, J.; Xu, W.; Yang, Y.; Wang, J.; Huang, Z.; and Yuille, A. 2014. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*.

Nam, J.; Kim, J.; Mencía, E. L.; Gurevych, I.; and Fürnkranz, J. 2014. Large-scale multi-label text classificationrevisiting neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 437–452. Springer.

Read, J.; Pfahringer, B.; Holmes, G.; and Frank, E. 2011. Classifier chains for multi-label classification. *Machine learning* 85(3):333.

Schapire, R. E., and Singer, Y. 2000. Boostexter: A boosting-based system for text categorization. *Machine learning* 39(2-3):135–168.

Shao, J.; Loy, C.-C.; Kang, K.; and Wang, X. 2016. Slicing convolutional neural network for crowd video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5620–5628.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sundermeyer, M.; Schlüter, R.; and Ney, H. 2012. Lstm neural networks for language modeling. In *Interspeech*, 194–197.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

Tsoumakas, G., and Katakis, I. 2006. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining* 3(3).

Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; and Xu, W. 2016. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2285–2294.

Wei, Y.; Xia, W.; Huang, J.; Ni, B.; Dong, J.; Zhao, Y.; and Yan, S. 2014. Cnn: Single-label to multi-label. *arXiv preprint arXiv:1406.5726*.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A. C.; Salakhutdinov, R.; Zemel, R. S.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 14, 77–81.

Yang, H.; Tianyi Zhou, J.; Zhang, Y.; Gao, B.-B.; Wu, J.; and Cai, J. 2016a. Exploit bounding box annotations for multi-label object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 280–288.

Yang, Z.; He, X.; Gao, J.; Deng, L.; and Smola, A. 2016b. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 21–29.

Yeh, C.-K.; Wu, W.-C.; Ko, W.-J.; and Wang, Y.-C. F. 2017. Learning deep latent space for multi-label classification. In *AAAI*, 2838–2844.

Zhang, M.-L., and Zhou, Z.-H. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering* 18(10):1338–1351.

Zhang, M.-L., and Zhou, Z.-H. 2014. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering* 26(8):1819–1837.