A Generalized Recurrent Neural Architecture for Text Classification with Multi-Task Learning

Abstract

Multi-task learning leverages potential correlations among related tasks to extract common features and yield performance gains. However, most previous works only consider simple or weak interactions, thereby failing to model complex correlations among three or more tasks. In this paper, we propose a multi-task learning architecture with four types of recurrent neural layers to fuse information across multiple related tasks. The architecture is structurally flexible and considers various interactions among tasks, which can be regarded as a generalized case of many previous works. Extensive experiments on five benchmark datasets for text classification show that our model can significantly improve performances of related tasks with additional information from others.

Introduction

Neural network based models have been widely exploited with the prosperities of Deep Learning BIBREF0 and achieved inspiring performances on many NLP tasks, such as text classification BIBREF1 , BIBREF2 , semantic matching BIBREF3 , BIBREF4 and machine translation BIBREF5 . These models are robust at feature engineering and can represent words, sentences and documents as fix-length vectors, which contain rich semantic information and are ideal for subsequent NLP tasks.

One formidable constraint of deep neural networks (DNN) is their strong reliance on large amounts of annotated corpus due to substantial parameters to train. A DNN trained on limited data is prone to overfitting and incapable to generalize well. However, constructions of large-scale high-quality labeled datasets are extremely labor-intensive. To solve the problem, these models usually employ a pre-trained lookup table, also known as Word Embedding BIBREF6 , to map words into vectors with semantic implications. However, this method just introduces extra knowledge and does not directly optimize the targeted task. The problem of insufficient annotated resources is not solved either.

Multi-task learning leverages potential correlations among related tasks to extract common features, increase corpus size implicitly and yield classification improvements. Inspired by BIBREF7 , there are a large literature dedicated for multi-task learning with neural network based models BIBREF8 , BIBREF9 , BIBREF10 , BIBREF11 . These models basically share some lower layers to capture common features and further feed them to subsequent task-specific layers, which can be classified into three types:

In this paper, we propose a generalized multi-task learning architecture with four types of recurrent neural layers for text classification. The architecture focuses on Type-III, which involves more complicated interactions but has not been researched yet. All the related tasks are jointly integrated into a single system and samples from different tasks are trained in parallel. In our model, every two tasks can directly interact with each other and selectively absorb useful information, or communicate indirectly via a shared intermediate layer. We also design a global memory storage to share common features and collect interactions among all tasks.

We conduct extensive experiments on five benchmark datasets for text classification. Compared to learning separately, jointly learning multiple relative tasks in our model demonstrate significant performance gains for each task.

Our contributions are three-folds:

Single-Task Learning

For a single supervised text classification task, the input is a word sequences denoted by INLINEFORM0

, and the output is the corresponding class label INLINEFORM1 or class distribution INLINEFORM2 . A lookup layer is used first to get the vector representation INLINEFORM3 of each word INLINEFORM4 . A classification model INLINEFORM5 is trained to transform each INLINEFORM6 into a predicted distribution INLINEFORM7 . DISPLAYFORM0

and the training objective is to minimize the total cross-entropy of the predicted and true distributions over all samples. DISPLAYFORM0

where INLINEFORM0 denotes the number of training samples and INLINEFORM1 is the class number.

Multi-Task Learning

Given INLINEFORM0 supervised text classification tasks, INLINEFORM1 , a jointly learning model INLINEFORM2 is trained to transform multiple inputs into a combination of predicted distributions in parallel. DISPLAYFORM0

where INLINEFORM0 are sequences from each tasks and INLINEFORM1 are the corresponding predictions.

The overall training objective of INLINEFORM0 is to minimize the weighted linear combination of costs for all tasks. DISPLAYFORM0

where INLINEFORM0 denotes the number of sample collections, INLINEFORM1 and INLINEFORM2 are class numbers and weights for each task INLINEFORM3 respectively.

Three Perspectives of Multi-Task Learning

Different tasks may differ in characteristics of the word sequences INLINEFORM0 or the labels INLINEFORM1 . We compare lots of benchmark tasks for text classification and conclude three different perspectives of multi-task learning.

Multi-Cardinality Tasks are similar except for cardinality parameters, for example, movie review datasets with different average sequence lengths and class numbers.

Multi-Domain Tasks involve contents of different domains, for example, product review datasets on books, DVDs, electronics and kitchen appliances.

Multi-Objective Tasks are designed for different objectives, for example, sentiment analysis, topics classification and question type judgment.

The simplest multi-task learning scenario is that all tasks share the same cardinality, domain and objective, while come from different sources, so it is intuitive that they can obtain useful information from each other. However, in the most complex scenario, tasks may vary in cardinality, domain and even objective, where the interactions among different tasks can be quite complicated and implicit. We will evaluate our model on different scenarios in the Experiment section.

Methodology

Recently neural network based models have obtained substantial interests in many natural language processing tasks for their capabilities to represent variable-length text sequences as fix-length vectors, for example, Neural Bag-of-Words (NBOW), Recurrent Neural Networks (RNN), Recursive Neural Networks (RecNN) and Convolutional Neural Network (CNN). Most of them first map sequences of words, n-grams or other semantic units into embedding representations with a pre-trained lookup table, then fuse these vectors with different architectures of neural networks, and finally utilize a softmax layer to predict categorical distribution for specific classification tasks. For recurrent neural network, input vectors are absorbed one by one in a recurrent way, which makes RNN particularly suitable for natural language processing tasks.

Recurrent Neural Network

A recurrent neural network maintains a internal hidden state vector INLINEFORM0 that is recurrently updated by a transition function INLINEFORM1 . At each time step INLINEFORM2 , the hidden state INLINEFORM3 is updated according to the current input vector INLINEFORM4 and the previous hidden state INLINEFORM5 . DISPLAYFORM0

where INLINEFORM0 is usually a composition of an element-wise nonlinearity with an affine transformation of both INLINEFORM1 and INLINEFORM2 .

In this way, recurrent neural networks can comprehend a sequence of arbitrary length into a fix-length vector and feed it to a softmax layer for text classification or other NLP tasks. However, gradient vector of INLINEFORM0 can grow or decay exponentially over long sequences during training, also known as the gradient exploding or vanishing problems, which makes it difficult to learn long-term dependencies and correlations for RNNs.

BIBREF12 proposed Long Short-Term Memory Network (LSTM) to tackle the above problems. Apart from the internal hidden state INLINEFORM0 , LSTM also maintains a internal hidden memory cell and three gating mechanisms. While there are numerous variants of the standard LSTM, here we follow the implementation of BIBREF13 . At each time step INLINEFORM1 , states of the LSTM can be fully represented by five vectors in INLINEFORM2 , an input gate INLINEFORM3 , a forget gate

INLINEFORM4 , an output gate INLINEFORM5 , the hidden state INLINEFORM6 and the memory cell INLINEFORM7 , which adhere to the following transition functions. DISPLAYFORM0

where INLINEFORM0 is the current input, INLINEFORM1 denotes logistic sigmoid function and INLINEFORM2 denotes element-wise multiplication. By selectively controlling portions of the memory cell INLINEFORM3 to update, erase and forget at each time step, LSTM can better comprehend long-term dependencies with respect to labels of the whole sequences.

A Generalized Architecture

Based on the LSTM implementation of BIBREF13 , we propose a generalized multi-task learning architecture for text classification with four types of recurrent neural layers to convey information inside and among tasks. Figure FIGREF21 illustrates the structure design and information flows of our model, where three tasks are jointly learned in parallel.

As Figure FIGREF21 shows, each task owns a LSTM-based Single Layer for intra-task learning. Pair-wise Coupling Layer and Local Fusion Layer are designed for direct and indirect inter-task interactions. And we further utilize a Global Fusion Layer to maintain a global memory for information shared among all tasks.

Each task owns a LSTM-based Single Layer with a collection of parameters INLINEFORM0 , taking Eqs.() for example. DISPLAYFORM0

Input sequences of each task are transformed into vector representations INLINEFORM0 , which are later recurrently fed into the corresponding Single Layers. The hidden states at the last time step INLINEFORM1 of each Single Layer can be regarded as fix-length representations of the whole sequences, which are followed by a fully connected layer and a softmax non-linear layer to produce class distributions. DISPLAYFORM0

where INLINEFORM0 is the predicted class distribution for INLINEFORM1 .

Besides Single Layers, we design Coupling Layers to model direct pair-wise interactions between tasks. For each pair of tasks, hidden states and memory cells of the Single Layers can obtain extra information directly from each other, as shown in Figure FIGREF21 .

We re-define Eqs.(EQREF26) and utilize a gating mechanism to control the portion of information flows from one task to another. The memory content INLINEFORM0 of each Single Layer is updated on the leverage of pair-wise couplings. DISPLAYFORM0

where INLINEFORM0 controls the portion of information flow from INLINEFORM1 to INLINEFORM2 , based on the correlation strength between INLINEFORM3 and INLINEFORM4 at the current time step.

In this way, the hidden states and memory cells of each Single Layer can obtain extra information from other tasks and stronger relevance results in higher chances of reception.

Different from Coupling Layers, Local Fusion Layers introduce a shared bi-directional LSTM Layer to model indirect pair-wise interactions between tasks. For each pair of tasks, we feed the Local Fusion Layer with the concatenation of both inputs, INLINEFORM0 , as shown in Figure FIGREF21 . We denote the output of the Local Fusion Layer as INLINEFORM1 , a concatenation of hidden states from the forward and backward LSTM at each time step.

Similar to Coupling Layers, hidden states and memory cells of the Single Layers can selectively decide how much information to accept from the pair-wise Local Fusion Layers. We re-define Eqs.(EQREF29) by considering the interactions between the memory content INLINEFORM0 and outputs of the Local Fusion Layers as follows. DISPLAYFORM0

where INLINEFORM0 denotes the coupling term in Eqs.(EQREF29) and INLINEFORM1 represents the local fusion term. Again, we employ a gating mechanism INLINEFORM2 to control the portion of information flow from the Local Coupling Layers to INLINEFORM3 .

Indirect interactions between Single Layers can be pair-wise or global, so we further propose the Global Fusion Layer as a shared memory storage among all tasks. The Global Fusion Layer consists of a bi-directional LSTM Layer with the inputs INLINEFORM0 and the outputs INLINEFORM1 .

We denote the global fusion term as INLINEFORM0 and the memory content INLINEFORM1 is calculated as follows. DISPLAYFORM0

As a result, our architecture covers complicated interactions among different tasks. It is capable of mapping a collection of input sequences from different tasks into a combination of predicted class distributions in parallel, as shown in Eqs.(EQREF11).

Sampling & Training

Most previous multi-task learning models BIBREF8 , BIBREF9 , BIBREF10 , BIBREF11 belongs to Type-I or Type-II. The total number of input samples is INLINEFORM0 , where INLINEFORM1 are the sample numbers of each task.

However, our model focuses on Type-III and requires a 4-D tensor INLINEFORM0 as inputs, where

INLINEFORM1 are total number of input collections, task number, sequence length and embedding size respectively. Samples from different tasks are jointly learned in parallel so the total number of all possible input collections is INLINEFORM2 . We propose a Task Oriented Sampling algorithm to generate sample collections for improvements of a specific task INLINEFORM3 .

[ht] Task Oriented Sampling [1] INLINEFORM0 samples from each task INLINEFORM1 ; INLINEFORM2 , the oriented task index; INLINEFORM3 , upsampling coefficient s.t. INLINEFORM4 sequence collections INLINEFORM5 and label combinations INLINEFORM6

each INLINEFORM0 generate a set INLINEFORM1 with INLINEFORM2 samples for each task: INLINEFORM3 repeat each sample for INLINEFORM4 times INLINEFORM5 randomly select INLINEFORM6 samples without replacements randomly select INLINEFORM7 samples with replacements each INLINEFORM8 randomly select a sample from each INLINEFORM9 without replacements combine their features and labels as INLINEFORM10 and INLINEFORM11 merge all INLINEFORM12 and INLINEFORM13 to produce the sequence collections INLINEFORM14 and label combinations INLINEFORM15

Given the generated sequence collections INLINEFORM0 and label combinations INLINEFORM1 , the overall loss function can be calculated based on Eqs.(EQREF12) and (EQREF27). The training process is conducted in a stochastic manner until convergence. For each loop, we randomly select a collection from the INLINEFORM2 candidates and update the parameters by taking a gradient step.

Experiment

In this section, we design three different scenarios of multi-task learning based on five benchmark datasets for text classification. we investigate the empirical performances of our model and compare it to existing state-of-the-art models.

Datasets

As Table TABREF35 shows, we select five benchmark datasets for text classification and design three experiment scenarios to evaluate the performances of our model.

Multi-Cardinality Movie review datasets with different average lengths and class numbers, including SST-1 BIBREF14 , SST-2 and IMDB BIBREF15 .

Multi-Domain Product review datasets on different domains from Multi-Domain Sentiment Dataset BIBREF16 , including Books, DVDs, Electronics and Kitchen.

Multi-Objective Classification datasets with different objectives, including IMDB, RN BIBREF17 and QC BIBREF18 .

Hyperparameters and Training

The whole network is trained through back propagation with stochastic gradient descent BIBREF19 . We obtain a pre-trained lookup table by applying Word2Vec BIBREF20 on the Google News corpus, which contains more than 100B words with a vocabulary size of about 3M. All involved parameters are randomly initialized from a truncated normal distribution with zero mean and standard deviation.

For each task INLINEFORM0 , we conduct TOS with INLINEFORM1 to improve its performance. After training our model on the generated sample collections, we evaluate the performance of task INLINEFORM2 by comparing INLINEFORM3 and INLINEFORM4 on the test set. We apply 10-fold cross-validation and different combinations of hyperparameters are investigated, of which the best one, as shown in Table TABREF41 , is reserved for comparisons with state-of-the-art models.

Results

We compare performances of our model with the implementation of BIBREF13 and the results are shown in Table TABREF43 . Our model obtains better performances in Multi-Domain scenario with an average improvement of 4.5%, where datasets are product reviews on different domains with similar sequence lengths and the same class number, thus producing stronger correlations. Multi-Cardinality scenario also achieves significant improvements of 2.77% on average, where datasets are movie reviews with different cardinalities.

However, Multi-Objective scenario benefits less from multi-task learning due to lacks of salient correlation among sentiment, topic and question type. The QC dataset aims to classify each question into six categories and its performance even gets worse, which may be caused by potential noises introduced by other tasks. In practice, the structure of our model is flexible, as couplings and fusions between some empirically unrelated tasks can be removed to alleviate computation costs.

We further explore the influence of INLINEFORM0 in TOS on our model, which can be any positive integer. A higher value means larger and more various samples combinations, while requires higher computation costs.

Figure FIGREF45 shows the performances of datasets in Multi-Domain scenario with different INLINEFORM0 . Compared to INLINEFORM1 , our model can achieve considerable improvements when INLINEFORM2 as more samples combinations are available. However, there are no more salient gains as INLINEFORM3 gets larger and potential noises from other tasks may lead to performance degradations. For a trade-off between efficiency and effectiveness, we determine INLINEFORM4 as the optimal value for our experiments.

In order to measure the correlation strength between two task INLINEFORM0 and INLINEFORM1 , we learn them jointly with our model and define Pair-wise Performance Gain as INLINEFORM2 , where INLINEFORM3 are the performances of tasks INLINEFORM4 and INLINEFORM5 when learned individually and jointly.

We calculate PPGs for every two tasks in Table TABREF35 and illustrate the results in Figure FIGREF47 , where darkness of colors indicate strength of correlation. It is intuitive that datasets of Multi-Domain scenario obtain relatively higher PPGs with each other as they share similar cardinalities and abundant low-level linguistic characteristics. Sentences of QC dataset are much shorter and convey unique characteristics from other tasks, thus resulting in quite lower PPGs.

Comparisons with State-of-the-art Models

We apply the optimal hyperparameter settings and compare our model against the following state-of-the-art models:

NBOW Neural Bag-of-Words that simply sums up embedding vectors of all words.

PV Paragraph Vectors followed by logistic regression BIBREF21 .

MT-RNN Multi-Task learning with Recurrent Neural Networks by a shared-layer architecture BIBREF11 .

MT-CNN Multi-Task learning with Convolutional Neural Networks BIBREF8 where lookup tables are partially shared.

MT-DNN Multi-Task learning with Deep Neural Networks BIBREF9 that utilizes bag-of-word representations and a hidden shared layer.

GRNN Gated Recursive Neural Network for sentence modeling BIBREF1 .

As Table TABREF48 shows, our model obtains competitive or better performances on all tasks except for the QC dataset, as it contains poor correlations with other tasks. MT-RNN slightly outperforms our model on SST, as sentences from this dataset are much shorter than those from IMDB and MDSD, and another possible reason may be that our model are more complex and requires larger data for training. Our model proposes the designs of various interactions including coupling, local and global fusion, which can be further implemented by other state-of-the-art models and produce better performances.

Related Work

There are a large body of literatures related to multi-task learning with neural networks in NLP BIBREF8 , BIBREF9 , BIBREF10 , BIBREF11 .

BIBREF8 belongs to Type-I and utilizes shared lookup tables for common features, followed by task-specific neural layers for several traditional NLP tasks such as part-of-speech tagging and semantic parsing. They use a fix-size window to solve the problem of variable-length texts, which can be better handled by recurrent neural networks.

BIBREF9 , BIBREF10 , BIBREF11 all belong to Type-II where samples from different tasks are learned sequentially. BIBREF9 applies bag-of-word representation and information of word orders are lost.

BIBREF10 introduces an external memory for information sharing with a reading/writing mechanism for communicating, and BIBREF11 proposes three different models for multi-task learning with recurrent neural networks. However, models of these two papers only involve pair-wise interactions, which can be regarded as specific implementations of Coupling Layer and Fusion Layer in our model.

Different from the above models, our model focuses on Type-III and utilize recurrent neural networks to comprehensively capture various interactions among tasks, both direct and indirect, local and global. Three or more tasks are learned simultaneously and samples from different tasks are trained in parallel benefitting from each other, thus obtaining better sentence representations.

Conclusion and Future Work

In this paper, we propose a multi-task learning architecture for text classification with four types of recurrent neural layers. The architecture is structurally flexible and can be regarded as a generalized case of many previous works with deliberate designs. We explore three different scenarios of multi-task learning and our model can improve performances of most tasks with additional related information from others in all scenarios.

In future work, we would like to investigate further implementations of couplings and fusions, and conclude more multi-task learning perspectives.