Not All Attention Is Needed: Gated Attention Network for Sequence Data

Lanqing Xue,[1] **Xiaopeng Li,**[2*] **Nevin L. Zhang**[1,3]
[1]The Hong Kong University of Science and Technology, Hong Kong
[2]Amazon Web Services, WA, USA
[3]HKUST-Xiaoi Joint Lab, Hong Kong
{lxueaa, lzhang}@cse.ust.hk, xiaopel@amazon.com

Abstract

Although deep neural networks generally have fixed network structures, the concept of dynamic mechanism has drawn more and more attention in recent years. Attention mechanisms compute input-dependent dynamic attention weights for aggregating a sequence of hidden states. Dynamic network configuration in convolutional neural networks (CNNs) selectively activates only part of the network at a time for different inputs. In this paper, we combine the two dynamic mechanisms for text classification tasks. Traditional attention mechanisms attend to the whole sequence of hidden states for an input sentence, while in most cases not all attention is needed especially for long sequences. We propose a novel method called Gated Attention Network (GA-Net) to dynamically select a subset of elements to attend to using an auxiliary network, and compute attention weights to aggregate the selected elements. It avoids a significant amount of unnecessary computation on unattended elements, and allows the model to pay attention to important parts of the sequence. Experiments in various datasets show that the proposed method achieves better performance compared with all baseline models with global or local attention while requiring less computation and achieving better interpretability. It is also promising to extend the idea to more complex attention-based models, such as transformers and seq-to-seq models.

Introduction

In recent years, deep learning has achieved great success in many applications, such as computer vision and natural language processing. Various neural network structures have been proposed to solve challenging problems. Although deep neural networks generally have fixed network structures, the concept of dynamic mechanism has drawn more and more attention. Instead of having a fixed computational graph, neural networks with dynamic mechanism adaptively determine how the computation should be conducted based on the inputs.

Attention mechanism is one of such dynamic mechanisms with dynamic weights. Motivated by human visual attention, attention mechanism computes input-dependent dynamic attention weights to select a portion of the input to pay attention to in a soft manner. In image captioning, attention mechanism allows the model to learn alignment between the visual portion of an image and the corresponding word in its text description (Xu et al. 2015). In neural machine translation, an encoder computes a sequence of hidden states from an arbitrary-length sentence, and the decoder needs to extract relevant information from the encoder in order to make predictions of each word. Attention mechanism aggregates the whole sequence of hidden states in the encoder by taking the weighted average of them with attention weights computed according to current decoding context. In such a manner, the decoding of different words in the target sentence pays attention to different words in the source sentence (Bahdanau, Cho, and Bengio 2015; Vaswani et al. 2017). And it has achieved remarkable performance in those applications.

Dynamic network configuration is another dynamic mechanism with dynamic connections for convolutional neural networks (CNNs), and has drawn more and more attention recently. Different from attention mechanism, it selectively activates only part of the network at a time in an input-dependent fashion (Bengio et al. 2016; Chen et al. 2019). For example, if we think we are looking at a car, we only need to compute the activations of the vehicle detecting units, not of all features that a network could possible compute (Bengio et al. 2016). The benefit of including only part of units for each input is that the propagation through the network will be faster both at training and test time since redundant computations are avoided while the cost of deciding which units to turn on and off is not high. While several works on dynamic network configuration have been proposed for CNNs, such as (Bengio et al. 2016; Veit and Belongie 2018; Chen et al. 2019), few such attempts have been made in sequence models such as recurrent neural networks (RNNs) for natural language processing.

In this paper, we seek to combine the two dynamic mechanisms for text classification tasks, and improve attention mechanism by dynamically adjusting attention connections in attention networks. Although attention-based neural networks achieved promising results, common attention mechanism has its limitations. Traditional attention mechanism is

*Work was done prior to joining Amazon.

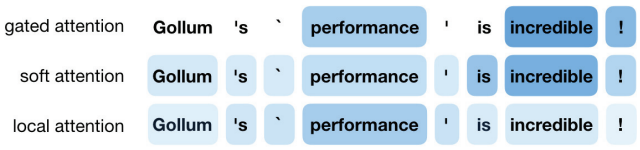

Figure 1: Examples of attention weights indicated by blue color from the proposed GA-Net and attention networks with global and local attentions for sentiment classification. GA-Net has a sparse attention structure and only attends to important words.

generally global, and it attends to all the words in the sentence though some attention weights might be small. However, through investigations into several natural language processing tasks, we observed that only a small part of inputs is related to output targets. It also aligns with our intuition that not all attention is needed especially for long sequences. The computation of attention weights on unrelated elements is redundant.

Not only that, since attention mechanism assigns a weight to each input unit and even an unrelated unit has a small weight, the attention weights on related units become much smaller especially for long sequences, leading to degraded performance.

To resolve the problem, we propose a novel method called Gated Attention Network (GA-Net) to dynamically select a subset of elements to attend to using an auxiliary network, and compute attention weights to aggregate the selected elements. A GA-Net contains an auxiliary network and a backbone attention network. The auxiliary network takes a glimpse of the input sentence, and generates a set of input-dependent binary gates to determine whether each word should be paid attention to. The backbone attention network is a regular attention network that performs the major recurrent computation, but only computes attention weights to aggregate the hidden states for the selected words. The attention units are sparsely connected to the sequence of hidden states, instead of densely connected as that in traditional attention mechanism. As an example in Figure 1, the proposed GA-Net has a sparse attention structure and has learned to only attend to important words. The auxiliary network and backbone attention network are trained jointly in an end-to-end manner. In summary, our contributions are as follows:

- We propose GA-Net, a novel sparse attention network, to dynamically select a subset of elements to attend to using an auxiliary network. It avoids unnecessary attention computation, and allows the model to focus on important elements in the sequence.
- An efficient end-to-end learning method using gumbel-softmax is proposed to relax the binary gates and enable backpropagation.
- We conduct experiments on several text classification tasks, and achieve better performance compared with all baseline models in our experiments with global and local attention networks.

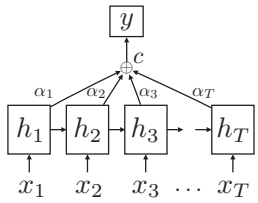

Figure 2: An example of attention mechanism: $\mathbf{c}$ is attention output, and α is attention weight.

Related Works

There have been a lot of works utilizing traditional attention mechanism together with CNNs and RNNs for various applications including computer vision (Xu et al. 2015), speech recognition (Chorowski et al. 2015) and natural language processing (Radford et al. 2018; Devlin et al. 2019; Cho, Courville, and Bengio 2015; Hermann et al. 2015; Rush, Chopra, and Weston 2015; Lu et al. 2016; Yang et al. 2016; Zhou et al. 2016). Attention mechanism has also been extended to act as a sequence model in place of RNNs, such as the Transformer network. Besides the traditional global structure of attention mechanism, recently, several works have been proposed to make adjustments on the attention mechanism, including inducing task-oriented structure biases to attentions (Kim et al. 2017; Liu and Lapata 2018; Zhu et al. 2017; Niculae et al. 2018), combining variational approaches with attentions (Deng et al. 2018), sparsifing attentions by handcrafted attention structures (Guo et al. 2019; Ye et al. 2019; Child et al. 2019; Luong, Pham, and Manning 2015), introducing variations of softmax regularizers (Martins and Astudillo 2016; Niculae and Blondel 2017; Mensch and Blondel 2018; Niculae et al. 2018), and utilizing probabilistic attention with marginalized average method (Yuan et al. 2019). However, none of these uses an auxiliary network to achieve input-dependent dynamic sparse attention structure.

Several works on dynamic network configuration have been seen in recent years. It is also similar to conditional computation (Bengio 2013). Gating modules are generally introduced to generate binary gates and dynamically activate part of the network for processing (Bengio et al. 2016; Veit and Belongie 2018; Chen et al. 2019; Bengio 2014). Policy gradient methods or relaxed gating methods are needed in order to enable backpropagation and end-to-end learning. However, these methods are designed for CNNs, and few attempts have been made for sequence models.

Attention Networks

In natural language processing (NLP), RNNs compute a sequence of hidden states for an arbitrary-length sentence. Instead of requiring the last hidden state to contain all information of the sentence, attention mechanism is widely used to aggregate information from the sequence of hidden states in an input-dependent manner. It computes an attention weight for each position in the input source, and

takes a weighted average of the hidden states as the output. For example, Figure 2 is an LSTM recurrent neural network. Let $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]$ be a sequence of inputs, and $\mathbf{h} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T]$ be a sequence of hidden states generated by the LSTM. The output $\mathbf{y}$ of the model is predicted as follows:

$$\hat{\mathbf{y}} = f(\mathbf{c}), \quad \mathbf{c} = \sum_{t=1}^{T} \alpha_t \mathbf{h}_t, \quad \mathbf{h_t} = LSTM(\mathbf{h_{t-1}}, \mathbf{x}_t)$$

where α_t is attention weight, and it indicates to what extent the t-th state influences the output $\mathbf{y}$. Vector $\mathbf{c}$ is the output of attention unit, and it is a weighted average over all states. Attention weight α_t is learned through a Multilayer perceptron (MLP) and normalized over all time steps via a softmax function:

$$\begin{aligned} e_t &= MLP(\mathbf{h}_t), \\ \alpha_t &= softmax_t(\mathbf{e}), \quad \sum_{t=1}^{T} \alpha_t = 1. \end{aligned} \tag{1}$$

The softmax function makes the sum of attention weights to be 1. Therefore, attention weight can also be seen as probabilities that a state is related to the output.

Since each state is assigned a weight, this kind of attention mechanism is called soft-attention and global-attention. There is also hard-attention and local-attention (Luong, Pham, and Manning 2015). In hard-attention, targets attend to only one state each time. This is computational efficient. However, it loses much information from inputs. Usually, not only one input has effects on targets. Local attention is a balance between soft and hard attention. Targets only attend to a window of its neighbors. The limitation is obvious, non-neighbors can also have influences on targets. Therefore, we proposed a sparse attention mechanism, GA-Net, which can dynamically select important inputs to attend to. It not only improves the computational efficiency of soft-attention, but also retains more information and achieves better interpretability than hard-attention and local-attention.

Gated Attention Network

We call our model Gated Attention Network (GA-Net) because it has an auxiliary network to generate binary gates to dynamically select elements to pay attention to for a backbone attention network. Theoretically, the backbone attention network can be any neural network with attention mechanism, such as bidirectional LSTMs and other sequence-to-sequence models. There are also a range of choices of auxiliary network, as long as it can produce a series of probabilities. Next, we describe the mechanism of GA-Net in details in the context of text classification.

Architecture of GA-Net

Figure 3 is the architecture of an GA-Net in classification tasks. The backbone attention network on the right is similar to that in Figure 2. The input to the model is a sequence of features $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]$ and the output is the classification target $\mathbf{y}$. Different from traditional attention network, the backbone network has additional gates

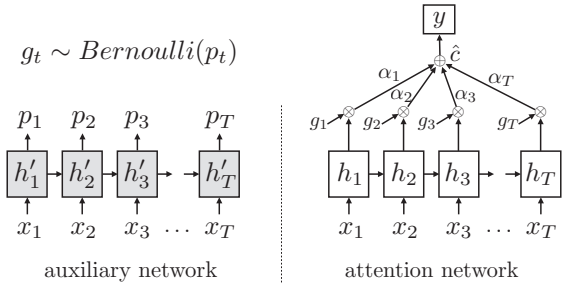

Figure 3: Architecture of GA-Net for classification tasks. Backbone attention network is on the right. The left one is a small auxiliary network producing a series of probabilities. Gate g_t is binary. It is sampled from the output of auxiliary network. Gate is open when $g_t = 1$, otherwise closed.

$\mathcal{G} = \{g_1, g_2, \ldots, g_T\}, g_t \in \{0, 1\}$ associated with each time step. The t-th gate is open when $g_t = 1$ and is closed when $g_t = 0$. It controls whether the information from current state should flow into targets. Let S be the set of positions t where $g_t = 1$. The attention weights in GA-Net are nonzeros for those positions with open gates:

$$\begin{aligned} e_t &= MLP(\mathbf{h}_t) \quad \text{for } t \in S, \\ \alpha_t &= \frac{\exp(e_t)}{\sum_{t' \in S} \exp(e_{t'})}, \quad \sum_{t \in S} \alpha_t = 1. \end{aligned} \tag{2}$$

On the other hand, for the positions with closed gates, we have $\alpha_t = 0$ and those hidden states $\mathbf{h}_t$ are not included into the aggregation. The attention output and the prediction of $\mathbf{y}$ are then computed as follows:

$$\hat{\mathbf{y}} = f(\hat{\mathbf{c}}), \quad \hat{\mathbf{c}} = \sum_{t \in S} \alpha_t \mathbf{h}_t. \tag{3}$$

The binary gates act as controllers to selectively activate part of the network. This resembles the dynamic network configuration mentioned earlier. To generate the binary gates, we introduce a dedicated auxiliary network. The auxiliary network takes a glimpse of the input sentence, and generate the binary gates for each position to determine whether the position needs to be paid attention to. The auxiliary network is on the left of Figure 3. It shares same input features with backbone attention network, but generally has a much smaller network size. The output of this auxiliary network is a set of probabilities $\mathbf{p} = \{p_1, p_2, \ldots, p_T\}$:

$$\begin{aligned} \mathbf{h}'_t &= LSTM(\mathbf{h}'_{t-1}, \mathbf{x}_t) \\ p_t &= sigmoid(\mathbf{U}\mathbf{h}'_t). \end{aligned} \tag{4}$$

The probability p_t determines the probability of the gate being open, and it is used to parameterize a Bernoulli distribution. A binary gate is then a sample generated from the Bernoulli distribution:

$$g_t \sim Bernoulli(p_t). \tag{5}$$

Though the example in Figure 3 shows an LSTM network as the auxiliary network, there are also other choices, such as

a feed forward network and a self-attention network. However, a sequence model considering the dependencies among words in the sentence, such as an RNN or a self-attention network might be a better choice in this situation.

Training GA-Net

We describe the end-to-end training method for the proposed GA-Net in the following. It is hard to train because the gates have discrete values of 0 and 1. Thus, errors cannot be back-propagated through gradient descent. Some papers (Mnih et al. 2014; Luong, Pham, and Manning 2015) utilized reinforcement learning to solve this problem. However, it is computational expensive and has limited performance. A recently proposed categorical reparameterization, named Gumbel-Softmax (Jang, Gu, and Poole 2017; Maddison, Mnih, and Teh 2017), is a potential method. Gumbel-Softmax aims at approximating a categorical distribution by a Gumbel-Softmax distribution with continuous relaxation. Optimizing an objective over an architecture with discrete stochastic nodes can be accomplished by gradient descent on the samples of the corresponding Gumbel-Softmax relaxation. Thus, it is a potential silver bullet to the back-propagation problem in our model.

In our model, each gate is a sample of value 0 or 1 from Bernoulli distribution. It can be taken as a binary 'classifier'. Each classifier g_t produce a two-element one-hot vector $\mathbf{g}_t = [g_{t,i}]_{i=0,1}$, where $g_{t,i} = 1$ means $g_t = i$. Similarly, $p_{t,i}$ is the probability that $g_t = i$:

$$\mathbf{g}_t = \text{one\_hot}\left(\arg\max_i p_{t,i},\ i = 0, 1\right), \tag{6}$$

$$p_{t,0} = 1 - p_t,\ p_{t,1} = p_t.$$

To let the auxiliary network differentiable during training, we can apply Gumbel-Softmax distribution as a surrogate of Bernoulli distribution to each gate. The Gumbel-Softmax distribution makes a softmax approximation to the one-hot vector $\mathbf{g}_t$:

$$\hat{\mathbf{g}}_t = [\hat{p}_{t,i}]_{i=0,1}, \tag{7}$$

$$\hat{p}_{t,i} = \frac{\exp((\log(p_{t,i}) + \epsilon_i)/\tau)}{\sum_{j=0}^{1} \exp((\log(p_{t,j}) + \epsilon_j)/\tau)}, \tag{8}$$

where ϵ_i is a random sample from Gumbel$(0, 1)$. When temperature τ approaches 0, Gumbel-Softmax distribution approaches to be one-hot. The attention weights with soft gates can be computed by

$$\alpha_t = \frac{g_t \odot \exp e_t}{\sum_{t'=1}^{T} g_{t'} \odot \exp e_{t'}}, \quad \sum_{t=1}^{T} \alpha_t = 1. \tag{9}$$

We use the gradients of Gumbel-Softmax as the surrogate gradients during backpropagation. During testing, however, the surrogate is not necessary. The generated gates are binary. Only selected elements are used for the computation of attention weights as in Eq(2). And we directly use the probabilities generated by the auxiliary network to parameterize the Bernoulli distribution and obtain the binary gates.

Table 1: Datasets Statistics. This table provides average sequence length (l), number of classes (K), number of training samples (Train) and testing samples (Test), and type of task in each dataset.

Dataset	l	K	Train	Test	Types
IMDB	231	2	25k	25k	sentiment analysis
AG	44	4	120k	7.6k	topic categorization
SST-1	20	5	12k	2.2k	sentiment analysis
SST-2	20	2	10k	1.8k	sentiment analysis
TREC	10	6	6k	500	question classification

To facilitate training procedure, we define the loss of this joint network as follow:

$$\mathcal{L} = -\sum_k y_k \log \hat{y}_k + \frac{\lambda \|\mathcal{G}\|_1}{T}. \tag{10}$$

The first term in loss function is cross-entropy loss, where y_k is the ground-truth label for k-th class. The second term is an l_1 norm regularizer over all gates, where λ is a hyper-parameter to make a trade-off between the cross-entropy loss and l_1 norm and T is input sequence length. The l_1 norm term aims at encouraging the network to turn off more gates and generate more sparse attention connections.

Experiments

In this section, we evaluate the performance of GA-Net on various datasets for sentence classification tasks. We did both quantitative and qualitative analysis on experiment results. Our model got better results compared with all baseline models in our experiments consistently.

Text Classification

We ran a series of experiments on various datasets for sentence classification task. Table 1 provides a summary of each dataset, and they are:

IMDB Large Movie Review Dataset (IMDB) is a binary sentiment classification dataset. IMDB provides a set of 25,000 highly polar movie reviews for training, and 25,000 for testing.

AG's News AG's News corpus contains 496,835 categorized news articles from more than 2000 news sources. It is constructed into a topic classification dataset with 4 main categories by Xiang Zhang (Zhang, Zhao, and LeCun 2015). Each category contains 30,000 training samples and 1,900 testing samples.

SST-1 Stanford Sentiment Treebank is a collection of movie reviews but with train/dev/test splits provided and fine-grained labels (very positive, positive, neutral, negative, very negative), re-labeled by (Socher et al. 2013)[1]. It has 11,855 training samples and 2,210 testing samples.

[1]https://github.com/harvardnlp/sent-conv-torch/tree/master/data

Table 2: Results of classification accuracy on various datasets for text classification. Bold numbers indicate best performance. Numbers in brackets indicate density of attention connections.

	IMDB	AG's News	SST-1	SST-2	TREC
BiLSTM	0.8509	0.9139	0.4125	0.8035	0.8876
BiLSTM+localAtt	0.8578	0.9234	0.4343	0.8154	0.8944
BiLSTM+softAtt	0.8863	**0.9264**	0.4445	0.8246	0.9008
GA-Net	**0.8941**	**0.9263**	**0.4464**	**0.8262**	**0.9124**
(density)	(0.1999)	(0.4310)	(0.4722)	(0.6005)	(0.4431)

SST-2 It is same as SST-1 but with neutral reviews removed and binary labels. It contains 9,613 training samples and 1,821 testing samples.

TREC This dataset is a collection of questions (Li and Roth 2002). The task is to classify a question into 6 question types (whether the question is about entity, human, location information, etc.) It contains 6,000 training samples and 500 testing samples.

For all the experiments, we chose a 2-layer bidirectional LSTM with attention as backbone network and another 1-layer bidirectional LSTM as auxiliary network (GA-Net). We applied the attention mechanism in (Zhou et al. 2016) as benchmark for text classifications. We compared our GA-Net with:

BiLSTM It is a vanilla bidirectional LSTM without attention. In this BiLSTM, we take the last hidden state as input to classifier;

BiLSTM+localAtt It is a vanilla bidirectional LSTM with local attention . The architecture we used as BiLSTM+localAtt is similar to that in (Luong, Pham, and Manning 2015). The original architecture in (Luong, Pham, and Manning 2015) is designed for sequence-to-sequence model: a local attended position in an encoder is predicted by the current hidden state in a decoder. To adapt it to our classification tasks, we take the last hidden state to predict a local position in a sequence to attend to.

BiLSTM+softAtt This is a vanilla bidirectional LSTM with global soft attention. It is common used attention mechanism which computes an attention weight for each element in a sequence.

In all three cases, we use the same BiLSTM configurations as those in backbone network of our GA-Net.

We download pre-trained 100-dimensional GloVe word vectors (Pennington, Socher, and Manning 2014)[2] to initialize all models. The hidden dimensions of backbone BiLSTM and auxiliary BiLSTM are both 100 in our experiments. During training, Gumbel-Softmax approximations are used as gates for all samples in both forward and backward propagation. During testing, only sampled discrete values are used. We use Adam (Kingma and Ba 2014) to optimize all models, with learning rate choosing from the set $[0.0001, 0.0002, 0.0005, 0.001, 0.002, 0.005, 0.01]$, batch size choosing from the set $[8, 16, 32, 64, 128]$.

[2]glove.6B, pretrained from Wikipedia 2014 and Gigaword 5. https://nlp.stanford.edu/projects/glove/

Table 3: The number of floating point operations of attention computations during testing (FLOPs).

	IMDB	AG's News	SST-1	SST-2	TREC
BiLSTM+softAtt	2.4G	131M	17M	14.1M	1.4M
GA-Net	0.4G	59M	6M	5.6M	0.7M

In GA-Net, we choose the temperature τ in Gumbel-Softmax from the set $[0.5, 1.0, 1.5, 2.0]$ to balance the sharpness of gradients and difficulty of training. We choose λ in loss function from the set $[0.4 \times 10^{-5}, 0.5 \times 10^{-5}, 1.0 \times 10^{-5}, 0.4 \times 10^{-4}, 0.5 \times 10^{-4}, 1.0 \times 10^{-4}, 1.0 \times 10^{-3}]$. We adapt cross-validation to select hyper-parameters for each dataset and task.

The results of classification accuracy are shown in Table 2. As it can be seen that, the proposed GA-Net consistently achieves the best performance on all datasets. For example, in TREC dataset, GA-Net achieves accuracy of 91.24%, and outperforms the baselines by at least 1.16%. At the same time, for the density of the resulting attention connections, we can see that it achieves this with much sparser attention structures. Especially for long sequences as in IMDB dataset where the average sequence length is 231, only 19.99% gates are switched on for each input. It demonstrates the fact that not all attention is needed especially for long sequences, and that GA-Net indeed has the ability of selecting those important units to attend to. In BiLSTM+localAtt, we choose a window size of 40, 16, 8, 8, 4 for each dataset respectively. This aim at making BiLSTM+localAtt have similar sparsity with GA-Net and achieve good performance at the same time.

We also report the improvement of attention computation in GA-Net and BiLSTM+softAtt in Table 3. We measure the number of floating point operations (FLOPs) of attention computations during testing for both models. GA-Net with sparse attention has lower FLOPs than BiLSTM+softAtt.

Interpretability Since our GA-Net only attends to part of units in a given sequence, its capacity of selecting related units in the sequence is important. Though GA-Net has good performance, we still want to know what elements in a sequence they actually attend to. To explore this, we did several case studies.

Figure 4 gives two examples. The two sentences in Figure 4 are drawn from SST-2 test dataset. Attention weights are computed for each token by GA-Net, soft attention and local attention respectively. Attention weights in Figure 4a and Figure 4d are assigned by GA-Net. The distributions

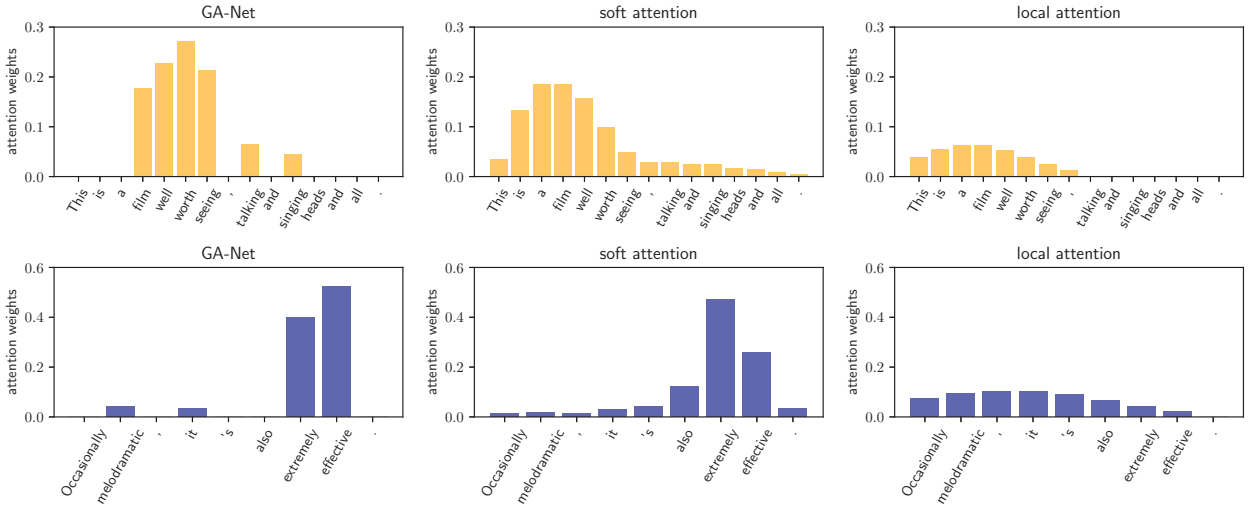

Figure 4: Case study in short sequences. Each row is an example sentences with computed attention weights for each token. The first sentence is a positive statement: *This is a film well worth seeing, talking and singing heads and all.* The second sentence is also a positive statement: *Occasionally melodramatic, it 's also extremely effective.* The attentions from left column to right column are provided by GA-Net, soft attention and local attention respectively.

Table 4: Case study in long sequences. The following are two paragraphs in IMDB test dataset. Bold texts are focused tokens selected by GA-Net.

I admire Deepa Mehta and **this movie is a masterpiece . I** 'd recommend to buy **this** movie on DVD because it 's a **movie you** might want to watch more often than **just** once . And trust me , you 'd still find little meaningful details after watching it several times.   The **characters** - except for the grandmother perhaps - are all very balanced , no black and white . Even though you follow the story from the perspective of the two protagonists , there is also empathy for the other characters.   I think the IMDb rating for the movie **is** far too low - probably due to **its politically controversial content** .
... means " take up and read " , which is precisely what I felt like doing after having seen **this marvelous film. Von** Ancken stimulates and inspires with **this breathtaking and superbly executed adaptation of** Tobias Wolff 's 1995 New Yorker article of the same name . The **incredible performance by** Tom Noonan **is brilliant and provocative and the editing** , sound design , cinematography and directing are **truly inspired . The nuanced** changes and embellishments on the original story are subtle , clever , and make the film cinematically more dynamic . It 's lyrical pacing **is mesmerizing and begs you to** watch it again. Watch out for this young director ... he 's going places .

of attention weights are sparse and compact. At the same time, attentions accurately focus on important token related to sentiment classification and shut off gates for meaningless tokens. For examples, token '*extremely*' and '*effective*' have very high weights in the second sentence "*Occasionally melodramatic, it 's also extremely effective.*". Meaningless punctuations, token ''*s*' and '*Occasionally*' are shut off. Figure 4b and Figure 4e are attentions computed by soft attention mechanism. Although it can identify related tokens, the distribution of its weights is more smooth than GA-Net and show less interpretability. Local attention in Figure 4c and Figure 4f is much more smooth compared with the other two models. It exhibits a weak capacity of finding out related tokens. This also implies the reason why BiLSTM+localAtt has similar performance with raw BiLSTM especially for long sequences where attention is helpful. Table 4 are examples from IMDB dataset with attention computed by GA-Net. The bold texts are tokens being selected to attend. The attention results show that the GA-Net successfully identifies related keywords for classification.

Table 5: Experiment results on IMDB Reviews with different auxiliary networks in GA-Net.

GA-Net	Accuracy	Density
BiLSTM+AUX$_{FNN}$	0.8890	0.2178
BiLSTM+AUX$_{ATT}$	0.8892	0.5326
BiLSTM+AUX$_{LSTM}$	**0.8941**	**0.1999**

Auxiliary Network To investigate the impact from the size of hidden dimension in auxiliary network, we did several experiments on IMDB, SST-1, TREC with GA-Net using the same structure but with different hidden dimensions in auxiliary BiLSTM. The hidden dimensions range from 20 to 100 with a step of 20. Figure 5 provides the variations in accuracy and density. we can observe that the performance of auxiliary LSTM with 20d hidden states is competitive to that with 100d hidden states. It also outperforms all baselines, and retains a low attention density. This implies that just a small auxiliary network is able to make attention sparse.

Moreover, we also curious about the impact from different choices of auxiliary networks. In stead of LSTM,

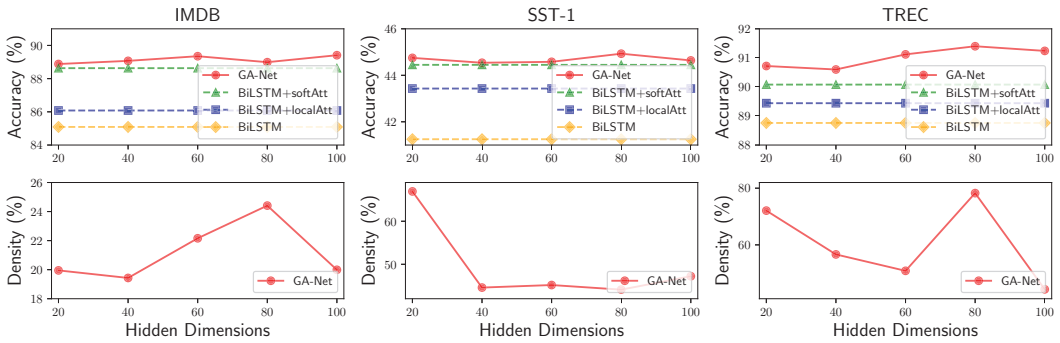

Figure 5: IMDB, SST-1 and TREC classification accuracy and attention density with different sizes of auxiliary networks. X-axis: size of LSTM hidden state; Y-axis: classification accuracy (upper row) and GA-Net attention density (bottom row).

we also chose an 1-hidden layer feed forward neural network (FNN), a self-attention network as auxiliary network to make comparison. We name them as BiLSTM+AUX$_{FNN}$, BiLSTM+AUX$_{ATT}$ respectively. We name the above GA-Net with LSTM as BiLSTM+AUX$_{LSTM}$. The backbone attention networks are same in all models. We use the same dimensions of hidden states as BiLSTM+AUX$_{LSTM}$ for BiLSTM+AUX$_{FNN}$ and BiLSTM+AUX$_{ATT}$. We did experiments on IMDB dataset.

Table 5 gives the results. Both BiLSTM+AUX$_{FNN}$ and BiLSTM+AUX$_{ATT}$ outperform all baseline models. This proves again that not all attention is needed, and the ability of auxiliary network in selecting meaningful units. Compared with BiLSTM+AUX$_{FNN}$ and BiLSTM+AUX$_{ATT}$, BiLSTM+AUX$_{LSTM}$ is still the one who achieves the best performance considering both accuracy and density. Therefore, LSTM is a relatively good choice for auxiliary network in dealing with similar tasks in which the inputs are sequences.

Conclusions

In this paper, we propose a novel method called Gated Attention Network (GA-Net) for sequence data. GA-Net dynamically selects a subset of elements to attend to using an auxiliary network, and computes attention weights to aggregate the selected elements. It combines two input-dependent dynamic mechanisms, attention mechanism and dynamic network configuration, and has a dynamically sparse attention structure. Experiments show that the proposed method achieves the best results consistently while requiring less computation and achieving better interpretability.

Acknowledgments

We would like to acknowledge the anonymous reviewers for their insightful comments. Research on this article was supported by Hong Kong Research Grants Council under grants 16202118 and 16212516.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*.

Bengio, E.; Bacon, P.-L.; Pineau, J.; and Precup, D. 2016. Conditional computation in neural networks for faster models. In *4rd International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings*.

Bengio, Y. 2013. Deep learning of representations: Looking forward. In *Statistical Language and Speech Processing*. Springer Berlin Heidelberg.

Bengio, Y. 2014. Deep sequential neural network. In *Deep Learning and Representation Learning Workshop*. NIPS.

Chen, Z.; Li, Y.; Bengio, S.; and Si, S. 2019. Gaternet: Dynamic filter selection in convolutional neural network via a dedicated global gating network. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*.

Child, R.; Gray, S.; Radford, A.; and Sutskever, I. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Cho, K.; Courville, A. C.; and Bengio, Y. 2015. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Trans. Multimedia* 17(11):1875–1886.

Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; and Bengio, Y. 2015. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems 2015*, 577–585.

Deng, Y.; Kim, Y.; Chiu, J.; Guo, D.; and Rush, A. 2018. Latent alignment and variational attention. In *Advances in Neural Information Processing Systems 2018*, 9712–9724.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Guo, Q.; Qiu, X.; Liu, P.; Shao, Y.; Xue, X.; and Zhang, Z. 2019. Star-transformer. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technolo-*

gies, NAACL-HLT 2019, Volume 1 (Long and Short Papers), 1315–1325.

Hermann, K. M.; Kociský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 2015*, 1693–1701.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*.

Kim, Y.; Denton, C.; Hoang, L.; and Rush, A. M. 2017. Structured attention networks. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, X., and Roth, D. 2002. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, 1–7. Association for Computational Linguistics.

Liu, Y., and Lapata, M. 2018. Learning structured text representations. *TACL* 6:63–75.

Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances in Neural Information Processing Systems 2016*, 289–297.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1412–1421. Lisbon, Portugal: Association for Computational Linguistics.

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2017. The concrete distribution: A continuous relaxation of discrete random variables. *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*.

Martins, A. F. T., and Astudillo, R. F. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning, ICML 2016, Volume 48*, 1614–1623.

Mensch, A., and Blondel, M. 2018. Differentiable dynamic programming for structured prediction and attention. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 3459–3468.

Mnih, V.; Heess, N.; Graves, A.; et al. 2014. Recurrent models of visual attention. In *Advances in neural information processing systems*, 2204–2212.

Niculae, V., and Blondel, M. 2017. A regularized framework for sparse and structured neural attention. In *Advances in Neural Information Processing Systems 2017*, 3338–3348.

Niculae, V.; Martins, A. F. T.; Blondel, M.; and Cardie, C. 2018. Sparsemap: Differentiable sparse structured inference. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 3796–3805.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language understanding paper.pdf*.

Rush, A. M.; Chopra, S.; and Weston, J. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, 379–389.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Veit, A., and Belongie, S. J. 2018. Convolutional networks with adaptive inference graphs. In *Proceedings of ECCV*. Association for Computational Linguistics.

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 *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, 2048–2057.

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

Ye, Z.; Guo, Q.; Gan, Q.; and Zhang, Z. 2019. Segtree transformer: Iterative refinement of hierarchical features. In *ICLR 2019 Workshop on" Representation Learning on Graphs and Manifolds*.

Yuan, Y.; Lyu, Y.; Shen, X.; Tsang, I. W.; and Yeung, D. 2019. Marginalized average attentional network for weakly-supervised learning. In *7th International Conference on Learning Representations, ICLR 2019*.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, 649–657.

Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; and Xu, B. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 207–212.

Zhu, C.; Zhao, Y.; Huang, S.; Tu, K.; and Ma, Y. 2017. Structured attentions for visual question answering. In *IEEE International Conference on Computer Vision, ICCV 2017*, 1300–1309.