CS533: Natural Language Processing
Project Milestone Report
Gated Attention Network for Sequence Data

Keya Desai kd706 | **Twisha Naik** tn268 | **Prakruti Joshi** phj15

1 Problem Description

Attention mechanisms have proved to be very useful for improving the performance of deep learning models across various domains specifically in language modelling. A lot of recent research focuses on the dynamic mechanism that adapts itself based on the input provided. In our project, we aim to implement one such dynamic method for attention called **gated attention**. Our main goal is to re-implement the technically challenging NLP paper "Not All Attention Is Needed: Gated Attention Network for Sequence Data" (Xue et al., 2019) and test the model on text classification tasks of SST-2, TREC and IMDB datasets. We aim to compare the GANet proposed in the paper with baseline models such as LSTM, BiLSTM with soft and global attention for classification task. The model proposed in the paper claims to be a sparse network as the number of attention computations are reduced since it selectively computes the part of input sequence to be processed. We also aim to analyze the density of the attention networks proposed in the GANet with soft attention and local attention networks for small and longer sequences.

2 Model

The architecture of the proposed GA-Net for classification tasks is as shown in Fig. 2.

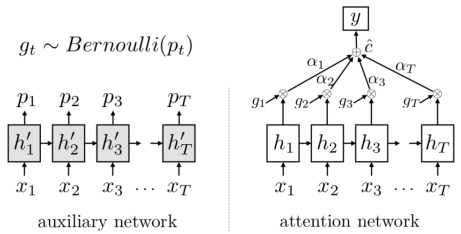

Figure 1: Proposed Architecture
On the right is the Backbone attention network which processes the input embedding sequence and produces an output classification target label. In the paper, the backbone network is a BiLSTM network with attention mechanism. The network on the left side is a small BiLSTM auxiliary network receives the same input sequence. This auxiliary network basically controls the gates associated with each timestep of the backbone network. This gates are binary and controls whether the information from current state should flow to the target. The gates g_t are sampled from the output of auxiliary network which is a sequence of probabilities. Gate is open when g_t = 1, otherwise closed.

In the original paper, the hidden dimensions of backbone BiLSTM and auxiliary BiLSTM are both 100. We have implemented the model with the same parameters as mentioned. In the auxiliary model, the gates are binary. This makes the normal backpropagation method invalid due to discontinuity. To overcome this problem, the training is done by implementing Gumbel Softmax idea proposed in 2016 (Jang et al., 2016). We have used Adam optimiser as suggested in the paper.

Currently the hyperparameters used are: To do We will be fine-tuning the hyperparameters such as learning rate, batch size, temperature to be used in Gumbel-Softmax and λ used in the loss function based on our experiments and different datasets. For the inputs, pre-trained 100-dimensional GloVe word vectors [1] are used to generate a word embedding.

Loss and Supervision setting:
We are using labeled datasets for training and supervised learning technique to train the model.

$$Loss = \sum_k y_k log(\hat{y}_k) + \frac{\lambda \left\| G \right\|_1}{T}$$

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

Model Description	Train Accuracy	Train Loss	Test Accuracy	Test Loss
LSTM	97.61%	1.136	77.54%	1.136
BiLSTM	97.26%	1.059	80.27%	1.112
GA-Net (LSTM + LSTM)	98.08%	1.051	78.91%	1.124
GA-Net (BiLSTM + BiLSTM)	97.46%	1.057	83.01%	1.088

Table 1: Results

3 Data Description

TREC-6: This dataset is for question classification. The task is to classify each question into 6 categories - Abbreviation, Description and abstract concepts, Entities, Human beings, Locations and numeric values. [2]
Training samples- 5452
Testing samples - 500
Distribution of classes in the training data:
'ENTY': 1250, 'HUM': 1223, 'DESC': 1162, 'NUM': 896, 'LOC': 835, 'ABBR': 86

4 Tasks done

We have used Python with PyTorch to implement the network. We are currently running our experiments on Google Colab.

- Loading and preprocessing data
  We are using the torchtext module to preprocess the data. Each data sample here is a question to be classified into one of the 6 classes. It is encoded into **data.Field()** and the corresponding label is stored as a **data.LabelField()**.
  By using data.Field(), the input is converted to a lower sequence and padded so that all the sequences are of same length. The data is split into training and testing.
  Each question is converted into an embedding using the pretrained Glove embeddings of length 100.

- Model variations
  1. LSTM and BiLSTM networks
  Implemented a LSTM and BiLSTM network independently without attention for text classification. We will use these models as a baseline for evaluating the results of GA-Net.
  2. GA-Net Implementation As described in the previous section, GA-Net has one auxiliary network and a backbone network. We have tried the following combinations of these two networks.
  - Backbone = LSTM, Auxiliary = LSTM
  - Backbone = BiLSTM, Auxiliary = BiLSTM

  3. Gumbel Softmax Integrated the Gumbel softmax in the auxiliary network to facilitate training.

- Training and Evaluation
  Implemented the cross entropy loss for training. Calculated the accuracy of classification by evaluating it with the ground-value of the target label. We have also measured the density of the attention network by calculating the ratio of attention networks activated during evaluation to total number of attention networks. This is to estimate the "sparseness" of the network.

5 Results

The results of the models is as described in the table 1. All the models are trained for **100 epochs**. The BiLSTM models perform better as compared to the LSTM models.

The hyperparameters used in the model are as follows:

```
learning_rate = 2e-5
batch_size = 32
output_size = 2
hidden_size = 256
embedding_length = 100
num_classes = 6
mlp_out_size = 32
weights = word_embeddings
aux_hidden_size = 100
batch_hidden_size = 100
tau = 0.5
```

Figure 2: Hyperparameters

[2] https://rdrr.io/cran/textdata/man/dataset$_trec.html$

GA-Net models perform better than the normal LSTM and BiLSTM models.

For TREC-6 data, we have kept the fixed sequence length of the input to be **10**.

- **Different configurations of Auxiliary Network**
  The auxiliary network that we have currently implemented is a LSTM and Bi-directional LSTM network. In the paper, the BiLSTM auxiliary network gives the best results. We will validate the results by changing the auxiliary network and implementing a simple Feed Forward Network and Attention network as compared to LSTM and BiLSTM.

- **Train the models till convergence**
  Currently, the stopping criterion for training is the predefined number of epochs. We will change it the condition and stop the training with the loss does not decrease for 10 epochs.

- **Train and test the models on different datasets**
  Current results are on the TREC data for classification of questions. We will train and test all the models for other datasets like SST-2 and IMDB.

- Compare the results of this sparse GANet with soft and local attention mechanism.

- Interpretability of the network in terms of attention weights assigned for shorter and longer sequences.

References

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Lanqing Xue, Xiaopeng Li, and Nevin L Zhang. 2019. Not all attention is needed: Gated attention network for sequence data. *arXiv preprint arXiv:1912.00349*.