HW4: Attention

Joe Davison
jddavison@g.harvard.edu

Josh Feldman
joshua_feldman@g.harvard.edu

April 3, 2019

1 Introduction

Natural language inference (NLI) has been proposed as a task to demonstrate natural language understanding [1]. Given a premise statement p and a hypothesis statement h, our task is to evaluate whether the p entails h, p contradicts h, or neither. [2] argues that an effective strategy in NLI is to break this task into three sub-tasks: attending, comparing and aggregating. Attending involves using an attention mechanism to align relevant subphrases in p and h. Then we compare the aligned subphrases. Finally, we aggregate these comparisons into a final answer. In addition to implementing and evaluating the decomposable attention model proposed in [2], we will also use their architecture to power a latent variable mixture model. Rather than using a single decomposable attention model, we will train multiple instances of this model with the hopes that each instantiation will become an "expert" on a certain types of premise-hypothesis pairs.

2 Problem Description

We formalize the task of NLI as follows. Let $\mathbf{p} = (p_0, \dots, p_i)$ be the premise sentence and let $\mathbf{h} = (h_0, \dots, h_j)$ be the hypothesis sentence. Each token in the sentence $p_i, h_j \in \mathbb{R}^d$ is a d-dimensional word embedding. For each premise-hypothesis pair $\{\mathbf{p}^{(n)}, \mathbf{h}^{(n)}\}$ in our dataset, we are also provided a label $y^{(n)} \in \mathbb{N}$ that indicates whether the pairing is an entailment, contradiction, neutral, or unknown. Our goal is to learn a function $f(\mathbf{p}, \mathbf{h}) = \hat{p}(y \mid \mathbf{p}, \mathbf{h}; \theta)$ that predicts the class label from the premise-hypothesis pair.

3 Model and Algorithms

In this section, we describ the models we implemented for this assignment.

3.1 Decomposable Attention Model

This model can be described in terms of attend, compare, and aggregation submodules. The attend submodule is implemented via an attention mechanism. First, we calculate unnormalized attention weights

$$e_{ij} = F(p_i)^T F(h_j)$$

where F is a 2-layer feedforward neural network with ReLU activation functions and dropout applied after each layer. These attention weights are normalized for each word in both the premise and hypothesis,

$$\pi_i = \mathrm{softmax}_j(e_{ij})$$

$$\pi_j = \mathrm{softmax}_i(e_{ij})$$

where π_i is a distribution over the words in $\mathbf{h}$ corresponding to word $p_i \in \mathbf{p}$ and, likewise, π_j is a distribution over the words in $\mathbf{p}$ corresponding to word $h_j \in \mathbf{h}$. We then use these distributions to soft-align the hypothesis for each word in the premise and vice-versa as follows

$$\widetilde{h}_i = \mathbf{h}\pi_i$$

$$\widetilde{p}_j = \mathbf{p}\pi_j$$

For each word p_i, h_j in the premise and hypothesis, we have soft-aligned the relevant subphrase $\widetilde{h}_i, \widetilde{p}_j$ in the opposite statement.

The next submodule is the decomposable attention model performs a comparison between the word and the aligned subphrase. To do so, we concatenate the word and the aligned subphrase and pass it through a feedforward network G with the same architecture as F,

$$c_{\text{premise},i} = F([p_i, \widetilde{h}_i])$$

$$c_{\text{hypothesis},j} = F([h_j, \widetilde{p}_j])$$

where $[\cdot, \cdot]$ denotes concatenation. We perform this operation for all words in the premise and hypothesis

The final submodule aggregates these $i + j$ comparison vectors. First, we pool via summation:

$$c_{\text{premise}} = \sum_i c_{\text{premise},i}$$

$$c_{\text{hypothesis}} = \sum_i c_{\text{hypothesis},i}$$

We feed the two pooled comparison vectors through a final network H, with same architecture as F,

$$\hat{\mathbf{y}} = H([c_{\text{premise}}, c_{\text{hypothesis}}])$$

where $\hat{\mathbf{y}}$ are the logits of our prediction. The final prediction is therefore $\hat{y} = \arg\max(\hat{\mathbf{y}})$. To train this model, we use the multi-class cross-entropy loss function.

3.2 Decomposable Attention Model with Intra-Attention

In the decomposable attention model described above, we can only align words in one sentence to subphrases in the other. By adding intra-attention, we can align subphrases to one another. We apply intra-attention to both the premise and hypothesis before feeding the resulting vectors through the architecture described above. Without loss of generality, we describe the intra-attention architecture with respect to the premise $\mathbf{p}$. We define the unnormalized attention weights as

$$f_{ij} = F_{intra}(p_i)^T F_{intra}(p_j)$$

We also define distance biases $\mathbf{d} \in \mathbb{R}^{11}$, which will provide the model with some sequence information. These are biases are trained via backpropogation. We calculate the attention weights

$$\pi_{intra,i} = \text{softmax}_j(f_{ij} + d_{\min(|i-j|,11)})$$

Finally, each element of the self-aligned premise $\mathbf{p}_{intra}$ is defined as

$$p_{intra,i} = \mathbf{p}\pi_{intra,i}$$

We then feed $\mathbf{p}_{intra}$ and $\mathbf{h}_{intra}$ into the decomposable attention architecture.

3.3 Mixture of Models

We implement a mixture of models as latent variable model, as described in the homework specification. Instead of using a single model, we use K models, each with a parameter configuration θ_c, where $c \sim \text{Uniform}(1, \cdots, K)$ denotes which model produces a label y. Then the marginal log likelihood is the following:

$$p(y|\mathbf{p}, \mathbf{h}; \theta) = \sum_{c=1}^{K} p(c)p(y|\mathbf{p}, \mathbf{h}; \theta_c)$$

We explore two variations of this setup. In the first, we simply enumerate over our K models by computing the expectation. In the second, we define an inference network $q(c|y, \mathbf{p}, \mathbf{h})$. Then we can optimize by using the ELBO:

$$\log p(y|\mathbf{p}, \mathbf{h}; \theta) \geq \mathbb{E}_{c \sim q(c|y, \mathbf{p}, \mathbf{h})} \log p(y|\mathbf{p}, \mathbf{h}; \theta_c) - KL(q(c|y, \mathbf{p}, \mathbf{h})||p(c)),$$

Since c is discrete, we cannot optimize the first term using the reparameterization trick, so we instead get its gradient using policy gradients, as described in the homework description. We evaluate both methods in our experiments below.

4 Experiments

We tuned and evaluated each of these models on the SNLI dataset [1]. Models were implemented with PyTorch and trained according to multi-class cross entropy loss. In each case, we used the Adam optimizer and gradually decayed the learning rate. We report the test accuracy for each model in Table 1.

Model	Test Accuracy (%)
Attention	79.5
Intra-attention	78.7
Uniform Mixture Model	67.1
VAE Mixture Model	34.6

Table 1: Best model test performance

We can visualize both the intra-attention distributions (fig. 1) and attention distributions between premise and hypothesis (fig. 2) We note three interesting patterns. First, the keywords of the sentence have a much lower variance in their attention distributions and concentrate around the

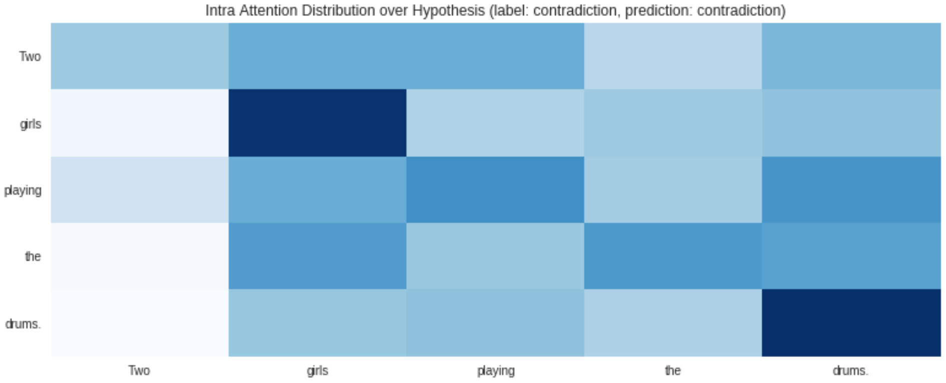


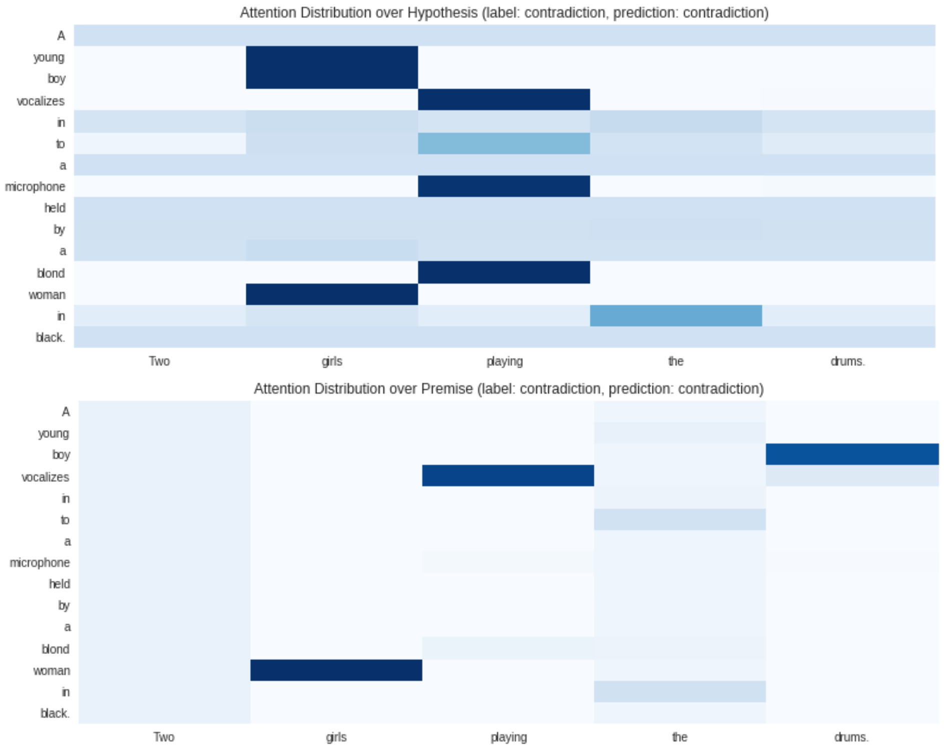
Figure 1: Sample intra-attention distribution (distributions are horizontal).

keywords in the other statement. This suggests that our model is at least comparing relevant items, although it is unclear whether these comparisons are meaningful. Second, the intra-attention distribution have a surprisingly high variance, given that relevant sub-phrases should only be couple words long. A regularizer on the intra-attention variance or adjusting the distance bias could help focus this component of the model. Finally, we note that the word drums is aligned almost entirely with the word boy. This suggests that our model might be leveraging gender stereotypes to create alignments.

5 Discussion and Conclusion

Our decomposable attention models achieved the highest performance. The model with intra-attention model performed slightly worse, which contradicts the results presented in [2]. Either we did not train the intra-attention model for long enough or the results in [2] do not generalize. Since intra-attention only increased accuracy by half a percent in the original paper, the latter could be the case. The visualizations of the intra-attention distributions illustrate that they have high variance and reducing this variance could improve performance.

In our experiments, our latent variable ensemble models did not perform as well on classification accuracy as the individual models did. The variational model did particularly poorly, apparently struggling to learn anything in depth. This difference of performance may simply be due to practical computational restraints – lack of resources and time to fine tune different models. The larger K is, the more models must be trained and converge to achieve a



Attention Distribution over Hypothesis (label: contradiction, prediction: contradiction)

Attention Distribution over Premise (label: contradiction, prediction: contradiction)

Figure 2: Sample attention distribution between premise and hypothesis.

high-performance model. The VAE model adds several additional layers of complexity by adding in the inference network, requiring variational training methods (policy gradients) rather than differential loss minimization. In short, we found better results from single models, but it is difficult to conclude that latent models were outperformed by single models without additional time and resources.

Due to the poor overall performance, it is difficult to determine whether the inference network learned meaningful relationships between data points and latent components. In evaluating the posterior distributions (with five components), however, we found that the model did favor some components for different types of tasks. For example, one component was frequently predicted by the network for neutral sentence pairs. Another component was almost never predicted. Interestingly, remaining three components were split between contradictions and entailments.

References

[1] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[2] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *CoRR*, abs/1606.01933, 2016.