All for One: Multi-Modal, Multi-Tasking

Bryan McCann
bmccann@stanford.edu

Nat Roth
natusroth@gmail.com

Abstract

We introduce Attention Teams, a unified framework for training multiple models end-to-end for multiple tasks. This paper explores banding together simple Gated Recurrent Units in teams coordinated by a high-level attention mechanism. The attention mechanism is operated by a GRU leader that determines how relevant the output of each GRU teammate is to the current input. This kind of coordination shows promise for more efficiently using parameters partitioned over small models for multi-tasking across Visual Question Answering and Sentiment Classification tasks. We hope that the idea of Attention Teams will inspire others to try more complex units within the teams, as the attention gives the architecture both the ability to communicate structure as well as the ability to intelligently ensemble units.

1 Introduction

This paper addresses multi-task, multi-modal modeling across visual and linguistic tasks. While unified architectures have proven successful on a variety of standard visual and non-visual question answering tasks [3,4,7], little work has been done to share the learning of one instantiation of an architecture with another [7]. No results to date have shown whether or not a single instantiation of an architecture can perform well across many tasks. Catastrophic forgetting remains a troublesome phenomenon for transfer learning done sequentially on many tasks [8], but this paper side-steps the issue of catastrophic forgetting by preferring a multi-task approach to a transfer learning one. Rather than focus on architectures that perform well when trained on specific tasks, this paper pushes towards using single models – with only one set of weights – for tackling both visual and non-visual question answering tasks. We propose to solve these problems by explicitly partitioning our architecture into a set of smaller models, with a single attention layer which chooses when to use each of these smaller models. This separation should help the model distinguish between different types of tasks in the multimodal and multi task setting, and have many smaller models which learn to specialize for subtasks.

2 Related Work

2.1 Visual Question Answering

Visual Question Answering has recently become a popular task with the introduction of the Visual Question Answering dataset [1]. The jury is still out on whether the dataset can really provide the kind of visual Turing test that so many hope for. Language models have seen tremendous success handling the linguistic information in the dataset with both simple BOW-based baselines [6] as well as simple deep-learning architectures like LSTMs [1]. Such models can achieve nearly 50% accuracy using only the linguistic inputs, viz. the question itself, on a task that uses completely open-ended questions about images. This has surprised many, but the community has not yet figured out how to tap the visual inputs in an effective way. Current results only have the visual input boosting performance by 10% [4]. As a multi-modal task, VQA fits nicely with a push for multi-tasking across modes.

2.2 Sentiment Classification

Sentiment Classification has been a standard task in Natural Language Processing for many years, but with the introduction of the Stanford Sentiment Treebank [5], language models finely began to break decades old barriers. Now, this task seems a standard for inclusion in papers on unified architectures [3] in Natural Language Processing, as progress continues to be made at a regular rate. Fortunately, newer recurrent networks have made the parse trees less relevant, opening up the field for less computationally expensive approaches.

2.3 Neural Attention Models

Attention teams naturally rely on an attention mechanism. These have proven successful in tasks ranging from image classification to machine translation and algorithmic tasks. Rather than have attention on words within a sentence or on representations of facts, we apply this idea to model-level attention by having a single attention mechanism that calculates a score for each model in the Attention Team based on their state and the current input. This allows for high-level coordination of models.

3 Data and Evaluation

3.1 Visual Question Answering

The Visual Question Answering [1] dataset builds off of the Microsoft COCO image dataset. It includes 82,783 training images, 40,504 validation images, and 81,434 test images. Each image comes with three questions, totaling the Visual Quesiton Answering dataset at 248.349 training examples, 121,512 validation examples, and 244,302 testing examples. Questions based on abstract scenes crafted by humans are also available, but this paper only makes use of the real images. We attempt to predict the most common answer of the ten answers proffered by the annotators. Answers need not be a single word, though we do not generate muliple word answers. We simply create unique tokens for such answers and include that in the vocabulary for our models.

3.2 Stanford Sentiment Treebank

The Stanford Sentiment Treebank [5] has fine-grained sentiment labels at each node in parse trees for 11,855 sentences. This paper only focuses on the binary classification task for full sentences. Our model must distinguish between positive and negative sentiment only at the root node, but fine-grained labels are used during the training process to boost the performance. Removing the neutral sentences leaves nearly 10,000 sentences for training, validation, and testing. We use the standard splits with 8,544 train sentences, 1,101 validation sentences, and 2,210 test sentences.

3.3 Mixing datasets

We chose to frame the task as question answering, which required some manipulation of the Stanford Sentiment Treebank. Each sentence was extracted from the examples, and the parse trees were discarded. The sentences were then treated as questions, to which the model was required to respond. Our version of the tasks is slightly more difficult in that we open up the vocabulary to be any word rather than restraining the model to only positive and negative classifications.

4 Methods

4.1 Baselines

For our baselines, we used single Gated Recurrent Units of varying embedding and hidden sizes.

$$z_t = \sigma(W_z[h_{t-1}, L[x_t]])$$

$$r_t = \sigma(W_r[h_{t-1}, L[x_t]])$$

$$\hat{h}_t = \tanh(W[r_t \circ h_{t-1}, L[x_t]])$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \hat{h}_t$$

where L is an embedding matrix accessed by the index for the input at time t: x_t. This provided a solid baseline for how the Attention Teams makes use of the parameters more effectively while simultaneously achieving better generalization results.

4.2 Attention Teams

An Attention Team can be described as follows. Let us denote a unit within the A-Team as U. We may have up to N units; each denoted as U_i. We use a single shared word embedding matrix L that is accessed by all U_i. $L[X]$ is the sequence of embeddings for the input.

We give each unit U_i access to $L[X]$, and calculate $m_i = U_i(L[X])$. This is the state of U_i after having seen the input. In the case of a GRU unit, this corresponds to the final state of the GRU after having read the entire sequence of input embeddings.

A lead unit $\hat{U}$, encodes an attention state for the team based on the input, which is used in conjunction with the state of each unit to get an attention score for that unit.

$$e = U(L[X])$$

$$a_i = \text{softmax}(\tanh(W_a[m_i, e]) + b_a)$$

These attention scores are used to blend the contributions from each unit and subsequently produce an output.

$$\tilde{m} = \sum_i a_i \circ m_i$$

$$o = f(W\tilde{m} + b)$$

where f is some non-linearity.

4.3 Images

Note, our tasks included answering questions about images. We used an improved Inception architecture [9] to extract feature vectors for all of the images. We extracted these R^{2048} vectors by running each image through the deep convolutional neural network, and we then embedded this representation in R^D, the size of a word embedding. For all models, we simply append this embedded image to the beginning of the word sequence for the question. If no image was available, e.g. for sentiment analysis questions, we simply used the zero vector of approach dimension. This has the effect of conditioning all attention units U_i as well as the lead unit U on the image. While a naive approach, decent success has been made using GoogleNet and VGG-19 [1,4,6].

Figure 1: A single GRU

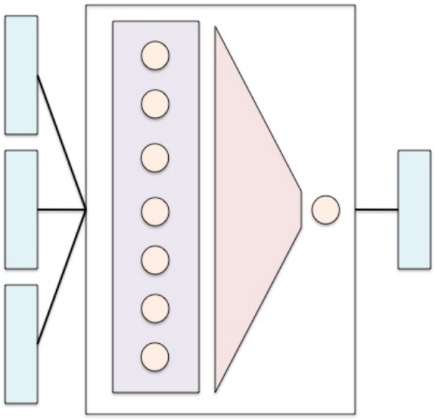

Figure 2: A GRU Attention Team

4.4 Intuition

Attention teams might be understood as ensembles of models that are trained together, end-to-end, and share a single embedding space. One might also experiment with pre-training units and then training the lead unit to choose between the pre-trained models in an approach that would more closely resemble ensembling methods. The benefit of Attention Teams are, in principle, two-fold. First, the partitioned parameter space coordinated by attention should allow different parts of the model to specialize on specific structures or patterns in the data, thereby dividing up the task of learning and making it feasible to conquer with smaller models and many fewer parameters. Second, if one prefers an ensembling perspective, Attention Teams are a generalized method for taking a combination of models during the learning process. Rather than just averaging together a collection of models that seemed to do well after they are all trained separately, Attention Teams take a principled approach to applying different models on different sub-tasks that it learns to find within the data.

5 Experiments

Since all of our tasks were classification based, we used the standard cross entropy loss across the board. The entire Attention Team is trained end-to-end using an Adam Optimizer with a learning rate that begins at 0.001 and is cut in half each time the validation accuracy does not improve after an epoch. Both lower and higher learning rates yielded extremely poor results, which exclude for the sake of space.

We found that dropout was absolutely essential for Attention teams to learn. After experimenting with several magnitudes of dropout, we discovered that large dropout was necessary to ensure that the attention mechanism allows all of the models to train in the beginning stages. Without this, only one model would be selected as it would continue to improve over the others with each example it saw. A dropout rate of 0.5 appeared to provide each unit access to enough attention in the beginning stages, but we found that leaving the dropout at 0.5 would sometimes detract too much from the specializing properties of the Attention Team. To mitigate this, we begin with a dropout rate of 0.5, but we slowly increase the probability of keeping the activation, viz. we 'anneal' the dropout rate towards 1 with each epoch.

We experimented with a broad array of hidden dimensions and word embedding dimensions. The size of the vocabulary was unlimited, but some results do suggest that capping the vocabulary between 6000 and 8000 can make learning easier and boost performance. We also experimented with different non-linearities for the function f, and we discovered that there was no significant difference between tanh and ReLU.

Single GRUs varying over the same hyperparameters served as our baseline for the Attention Teams.

6 Results

The Stanford Sentiment Treebank is significantly smaller than the Visual Question Answering dataset, which makes overall validation accuracies misleading when in the multi-tasking setting. For this reason, and for direct comparison against other results, we report the accuracy achieved on the VQA dataset, the accuracy achieved on the SSTB, and the combined accuracy on VQA+SSTB.

We were not concerned with approach state-of-the-art numbers, but instead aimed at proving the relative effectiveness of Attention Teams in three key areas: first, Attention Teams can make use of fewer parameters than larger models while achieving better performance; second, Attention Teams can successfully survive multi-tasking across VQA and SSTB; and third, having the coordinated attention gives valuable insight into both the model and the data. All three of these advantages we expect to hold with more complicated units, and so a trek towards state-of-the-art with Attention Teams should be fairly low-hanging fruit for future work.

D	H	N	VQA	SSTB	VQA+SSTB
300	10	1	27.87	78.56	28.23
300	10	2	27.99	75.80	27.65
600	10	2	28.45	77.87	28.81
300	10	5	28.75	50.92	28.90
600	10	5	28.43	76.72	28.77
300	10	10	28.56	76.23	28.89
600	10	10	29.42	75.92	29.74
300	100	1	39.24	82.22	39.54
600	100	1	38.98	81.08	39.28
300	100	2	40.21	82.11	40.50
600	100	2	40.73	82.00	41.01
300	100	5	41.29	81.42	41.57
600	100	5	*42.07*	80.16	*42.33*
300	100	10	41.44	80.50	41.72
600	100	10	**42.14**	82.00	**42.42**
300	1000	1	40.90	81.42	41.19
600	1000	1	40.48	**84.86**	40.78
300	1000	2	40.38	83.83	40.67

Table 1: Results of joint-training.

6.1 Trends in the Results

The first thing to note when looking through the above results table is that there appears to be a bit of a curve in the effectiveness of adding additional parameters to a model without imposing further structure. Our Attention Team imposes this structure on itself, as will be discussed below, which allows it to make use of smaller units with a shared embedding space. Teams of 5 and 10 units with only 100 parameters in the hidden dimension of each unit outperform the swollen GRUs with 1000 dimensional hidden spaces, which is an order of magnitude of savings in the parameter space even with the larger ten unit team.

This does not seem to hold true for the Stanford Sentiment Treebank alone despite the fact that it holds for the joint results. In fact, there appears to be a trade-off between fitting the two datasets; larger models tend to outperform even the small teams on the sentiment task, but they sacrifice too much on Visual Question Answering in order to do so.

On the other hand, analogous experiments training on the datasets separately suggest that training jointly improves the language model both tasks. The highest results on the Sentiment Treebank without joint training was 83.00% and the best result on the Visual Question Answering dataset we achieved without joint training was 40.02, both delivered by a single 1000 dimensional hidden state GRU with 300 dimensional word embeddings. Though much higher accuracies have been published using more complex methods, this appears to demonstrate the relative effectiveness of Attention Teams while also ssuggesting that they may be strengthened with stronger units.

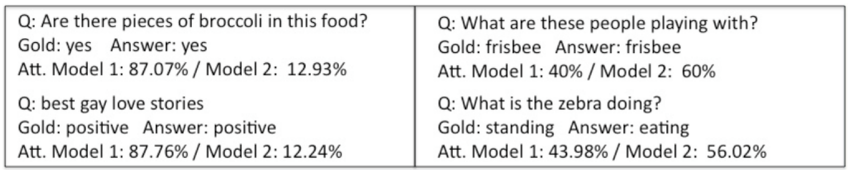

Figure 3: One model gets more attention for binary classification questions, while the other gets open-ended questions.

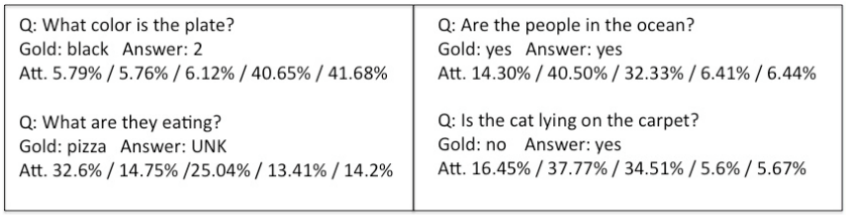

Figure 4: When there appear to be more models than sub-tasks identified by the lead unit, the team units ensemble and fire in distinct groups: color, yes/no, other.

6.2 Attention Analysis

Our experiments produced a few interesting results and behaviors. In particular, the models learned to specialize to different sub-tasks within the data. We expected the model to learn to split the joint dataset into the original constituents, this was not the case in general.

When using two units, we noticed that often one model would learn to answer the binary questions, namely yes/no for VQA, and the sentiment questions (though this was not universally the case across runs and model sizes), while the second would handle the other VQA question.

Questions that are more open-ended tend to be more difficult for the architecture. In these cases, the attention wieghts are also much closer to uniform. In this case it seems like the model is learning less to explicitly specialize, and instead learns to average over both different models. Anecdotally, we noticed that the attentions were more split on categories of VQA questions where we had more examples, such as color, and more uniform on those where data was sparser. This makes sense, as it seems plausible neither model would specialize as heavily to handle these harder, rarer cases, simply because other data points would outweigh them.

This trend held when we used more models than just two as well, though the patterns changed a little. When using five units, rather than seeing a single model get all the attention for a given type of question, we observed that groups of models would team up and fire together to tackle a single sub-task. For example, for yes/no questions, two models would be granted most of the attention, but split between them. For color questions, two other models, mutually disjoint from the first group, fired together. We had originally expected to see attention scores that were heavily tied to a single model, which would indicate that each model had learned to specialize to a unique part of the data. We are not completely sure why attention chose to ensemble models rather than continue to pick out smaller and smaller sub-tasks within the data. It is possible that by training all the units with a shared word embedding matrix, we introduce some correlation between how separate units are learning, which in turn inextricably links them. This connection may make it harder for a model to isolate all units. It would be interesting to see how results changed if we tried partitioning the total dimensionality of the word-embedding matrix across the different units.

7 Future Work

Our work has provided us with a good deal of insight and further questions we would like to examine. In particular, we would like to experiment with having each unit have its own word embedding matrix, and see if this affects the way in which the units learn or the way attention is distributed. We would like to continue to mix in more datasets, such as Facebook's BABI dataset, and see if

the trends we observe here all hold or if any others arise. If untying the word embeddings does not increase isolation, one might try hard attention, rather than a soft attention; this could be done by choosing to only consider the model with the highest attention score or by sampling models based on the attention scores. Preliminary results suggested that this would not help much, but our investigation was not thorough enough to rule out the possibility. Continuing in that vein, we would also like to see experiments with attention on the past, so that we don't need to run each model over the inputs. One might simply give the lead unit the most recent final states of each unit, and then only run over the current input with the unit that gets the most attention. Significant work could be done by substituting the GRU with any other of a number of candidate units, tapping into the beauty of deep-learning's constitutive properties. Aside from all of this, we would like to mimic similar methods on the visual input of the data, using small CNNs rather than a large one that has been pre-trained for image classification. This seems particularly urgent, as the current features appear to encode little information useful for color and number questions.

8 References

[1] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. VQA: Visual Question Answering. ICCV, arXiv:1505.00468, 2015.

[2] Weston, J., Bordes, A., Chopra, S., and Mikolov, T. Towards ai-complete question answering: A set of prerequisite toy tasks. arXiv:1502.05698, 2015a.

[3] Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., and Socher, R. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. arXiv:1506.07285, 2015.

[4] Xiong, C., Merity, S., and Socher, R. Dynamic Memory Networks for Visual and Textual Question Answering. arXiv:1506.07285, 2015.

[5] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. EMNLP, 2013b.

[6] Zhou, B., Tian, Y., Sukhbaatar, S., Szlam, A., and Fergus, R. Simple baseline for visual question answering. arXiv:1512.02167, 2015.

[7] Collobert, R. and Weston, J. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. ICML, 2008.

[8] Goodfellow, I., Mirza, M., Xiao, D., Courville, A. and Bengio, Y. An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks. arXiv: 1312.6211, 2015.

[9] Szegedy, C., Vanhoucke, V., Ioffe, S., and Schlens, J. Rethinking the Inception Architecture for Computer Vision. arXiv: 1512.00567, 2015.