Insemble: A Shared-Parameter Ensembling Technique for Question-Answering

Justin Lim Luke Hansen

Computer Science, Stanford

Goals and Motivations

Goals:

- Improve upon BiDAF's performance on a reading comprehension task with modifications such as using a GRU RNN, character-level embeddings, self-attention encoder, and using a QANet model.
- Create a model inspired by the Intra-Ensemble neural network (an end-to-end ensemble strategy with submodels which share some parameters) that uses our best performing model as a submodel
- Explore this model's effectiveness, and investigate how performance scales with number of parameters.

Motivations: Ensembling is effective but significantly increases number of parameters. How can we get the performance benefits of ensembling, with a less significant increase in parameter size?

Approach

- Insemble: a language processing architecture that was heavily inspired by the layer sharing technique proposed by the Intra-ensemble paper.
- Insemble has three submodels, with the first and second model sharing parameters for the first encoder block and the first and third model sharing parameters for the second encoder block.
- The submodels will use slightly different hyper-parameters and architectural components to increase the diversity of representations captured by each submodel.

Insemble will use a traditional QA architecture as a submodel. We experimented with the following candidate submodels:

1. **BiDAF**: Baseline BiDAF model provided by CS224n class. Does not include character-level embeddings
2. **GRU and Character Embeddings**: Replaced the Bi-directional LSTM with GRU (Gated Recurrent Unit).
3. **Self-Attention Encoder**: replaced the first encoder layer of the BiDAF model with a self-attention layer that adds context-to-context attention and query-to-query attention
4. **QANet**: new model that discards the recurrent concept of BiDAF. Uses convolutions and multi-headed self-attention as encoder blocks.

We will deploy Insemble on a reading comprehension task using the SQuAD 2.0 dataset (a collection of context, question, answer triples).

After training the 4 candidate models we will:

- Evaluate each model's EM, F1, AvnA scores
- Choose best models for Insemble
- Diversify submodels through using different hyper-parameters, and architectures within the encoder blocks.

Experiments and Results

- Trained candidate submodels until losses, F1, and EM scores plateaued
- Trained QANet with a learning rate of 0.1, hidden size of 100, drop probability of 0.1
- All other submodels were trained with a learning rate of 0.5, hidden size of 100, drop probability of 0.2

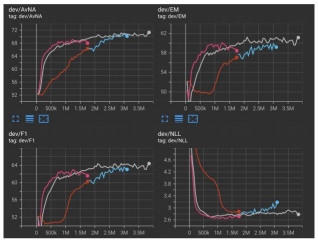

Figure 1. AvNA, EM, F1, and NLL training trajectories: GRU (grey), self attention encoder (pink), and QANet (red and blue)

Our 4 submodel candidates achieved the following EM, F1, and AvNA scores:

Model	NLL	F1	EM	AvNA
BiDAF	3.31	60.18	56.53	67.23
GRU	3.11	60.76	57.22	67.47
GRU + Char-Embed	2.70	64.32	61.1	71.1
Self-Attention Encoder	2.65	62.97	60.01	69.06
QANet	2.75	63.63	60.85	70.98

Table 1. Comparison of submodel performance on dev Set

- Model with GRU encoder performed best so chose it as submodel
- For our Insemble, the third submodel used an LSTM for its first encoder, increasing submodel diversity. Trained our Insemble with different hidden state sizes
- As a control, one of the submodels (225-GRU) was trained with a similar number of parameters as smallest Insemble network.

Model	NLL	F1	EM	AvNA
GRU-225 (included as a control)	2.92	61.34	57.67	70.14
Insemble-75	2.76	64.87	61.84	70.14
Insemble-100	2.73	66.59	63.59	71.35
Insemble-150	2.93	66.03	62.95	71.27

Table 2. Comparison of Insemble models with different number of parameters (number to to the left of Insemble represents hidden size value; and performance of a submodel with a comparable number of parameters.

The Insemble model with a hidden size of 100 achieved the best dev set score, so we ran it on the test set. It achieved scores 65.46 (F1) and 62.40 (EM).

Analysis

- Using GRU RNN increase performance and decreased training time.
- Using character-level embeddings improved the model's handling of unencountered words.
- QAnet had an unexpectedly low performance. Given the limited the credit count and the extensive time needed to train the QANet (14 hours), we decided to focus on testing the Insemble model instead of debugging the QANet.
- Smallest Insemble model allowed for better representations of the data than using a single model with a comparable number of parameters (This single model quickly overfit the data).
- Increasing the number of model parameters increased the performance of the model, to an extent– increasing the hidden size from 75 to 100 increased the F1 and EM scores by 1.72 and 1.75, respectively.
- However increasing the hidden size from 100 to 150 hurt the models performance, lowering the F1 and EM scores by 0.56 and 0.64, respectively. We expect this was due to the model's increased representational capacity not being utilized because there was an insufficient amount of data.

Furthermore, we split the Insemble results by question type (How, What, Why, Which, Who, Where, When, Other). All three models performed the best on "when" questions and performed poorly on "how" and "why" questions. This observation makes sense intuitively, because "when" questions often have more straightforward answers than "how" and "why" questions (Figure 2) All models perform poorly on "other" questions, which seems to be a direct result of the low number of training examples in this category.

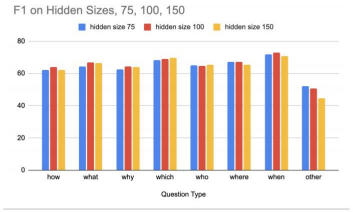

Figure 2. Insemble score by question type

Conclusion

- The following strategies performed better than a vanilla BiDAF model: replacing the LSTM RNN with GRU RNN; including character embeddings; using self-attention instead of the LSTM RNN; and using QANet model instead of the BiDAF model.
- Insemble performed better than any of the models individually, even with models with approximately equal parameter sizes.
- Insemble's performance improved with more parameters to a certain extent. After increasing the size too much, performance began to decrease.