CNN for Text-Based Multiple Choice Question Answering

Team 1: Michelle Foo, Dong Hwan Kim, Jeongmin Park

Overview

Back in 2017, the idea of using CNN for text-based multiple choice question answering is a novel idea. And surprisingly the CNN model outperformed traditional LSTM models in this task. Our task in this course is to reimplement the model from the paper[1] and carry out some modifications in order to investigate how we may further improve the model.

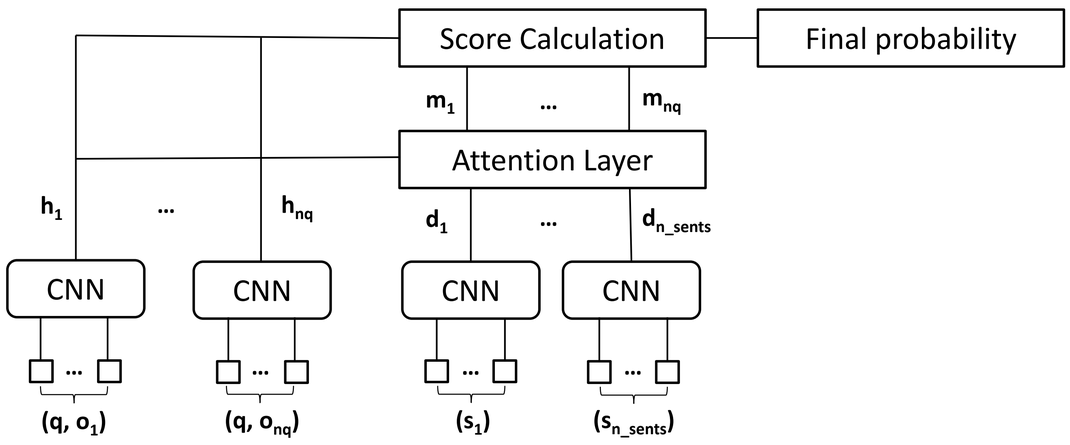

Figure 1: Architecture of the model. Attention layer attends on sentence embeddings d_j's using question-option tuple embeddings h_i's. Score calculation layer calculates cosine similarity between m_i and h_i which is passed through softmax to get the final probability distribution.

Dataset

SciQ Dataset[2]

- 13,678 crowdsourced science exam questions
- Questions are in multiple-choice format with 4 answers each
- Train data example:
  Question:
  What is the least dangerous radioactive decay?
  Options:
  1) alpha decay 2) beta decay 3) gamma decay 4) zeta decay
  Support text:
  All radioactive decay is dangerous to living things, but alpha decay is the least dangerous.

Contributions

Modification 1: Replace word embeddings model
Elaboration:

- Word2Vec[3] embeddings that are fed into the CNN models are replaced with FastText embeddings and BERT embeddings.

Justification:

1. FastText[4] is an extension of Word2Vec model. Each word is composed of character n-grams. For example the word "apple" with n=3: <ap, app, ppl, ple, le>. < and > are boundary symbols.
   Improvement over Word2Vec:
   a. Ability to generate better word embeddings for rare words because character n-grams are shared with other common words.
   b. Vector of Out Of Vocabulary (OOV) words can be constructed using character n-grams.
2. BERT[5]
   Improvement over Word2Vec:
   a. Context dependent

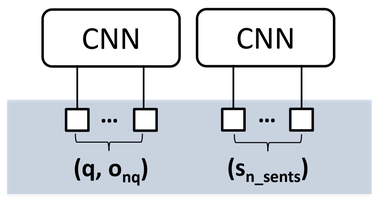

Figure 2: The highlighted region shows the part that is modified

Modification 2: Increase/decrease the number of CNN layers
Elaboration:

- Increasing/ decreasing the number of Convolution layers used for question-option and sentence embedding extractions.

Justification:

- Slightly augmenting model size at the cost of more training parameter could maybe lead to faster convergence of the model
- Reducing the number of trainable parameters to investigate the impact on the model's accuracy. Maybe not that many parameters are needed to achieve the same accuracy

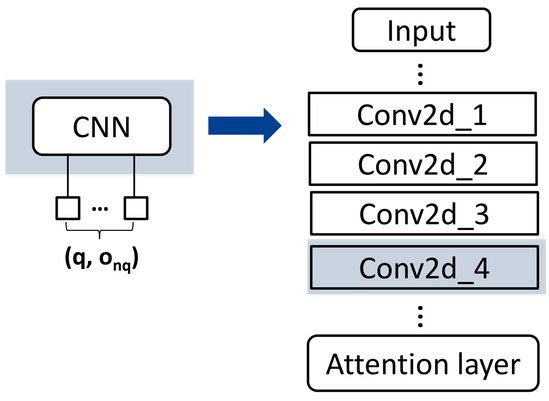

Figure 3: Additional Conv2D layer is added to the model

Experiments, Results & Discussions

Global model parameters:
Word embeddings size: 300 dimensions
Optimizer: Stochastic Gradient Descent (SGD)

Baseline model

- Word2Vec embeddings from model trained using Google News data

Modification 1: Replace word embeddings model

1. FastText embeddings from model trained using Wikipedia News data
2. BERT embeddings from model trained using Wikipedia data
   a. Since the pretrained BERT model only output word vectors with size 784, we need to retrain the BERT model in order to obtain embeddings with dimensions of 300.
   b. But due to computational power limitation, we are unable to carry out this step. And it is not feasible to use the existing BERT model for embeddings extraction because it takes too long without GPU support.

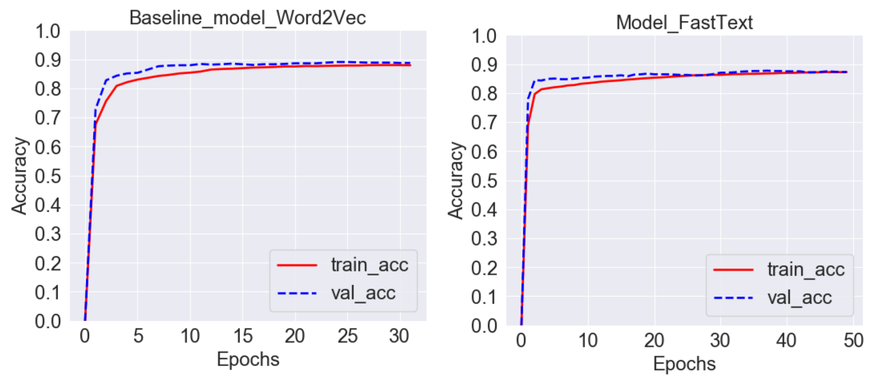

Figure 4: Train and validation accuracy curves

- The figures above show that the models' accuracy plateaued around 20 epochs.

Model	Test Accuracy
Baseline (Word2Vec)	**0.888**
FastText	0.874

Figure 5: Test accuracy of the trained models

- The model trained with FastText did not perform better than the baseline model. An accuracy difference of 1.4% is observed between the different models.

Modification 2: Increase/decrease the number of CNN layers

1. An extra Conv2D layer with filter size 5 is added
2. Layer with filter size 4 is removed

- The models are trained for 30 epochs, on average one model is trained for ~9 hours.

Results:

- The model with only 2 layers of Conv2D performed with similar accuracy compared to the baseline model, but it is able to reach a higher accuracy faster.
- With a smaller model a speed up of 1.13 times is possible.

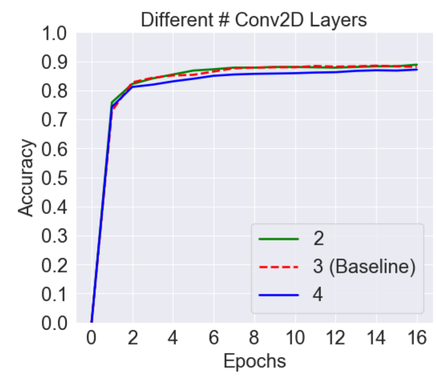

Figure 6: Validation accuracy of models with different # of Conv2D layers

# Conv2D layers	# Trainable Parameters	Duration per Epoch	Speed up	Test Accuracy
2	**225,300**	**~15 minutes**	**1.13x**	**0.888**
3 (Baseline)	270,300	~18 minutes	1x	0.888
4	315,300	~20 minutes	0.85x	0.877

Figure 7: Test accuracy of the various models

Conclusions

- We learned that **using FastText embeddings over Word2Vec has little impact on the resulting accuracy of the model**. This shows that the use of character n-grams embeddings did not improve the embeddings that were further generated by the CNN model.
- We proved that **only two layers of Conv2D layers is needed for the model to reach the same accuracy as the baseline model**. This means that the model can be scaled down which leads to faster training.
- Lack of computational power hindered our progress in the project. And the initial steps of getting the baseline model to run also consumed a significant amount of time. We are not able to carry out more model modifications and testing due to those issues.

References

1. Chaturvedi A., Pandit O., Garain U. CNN for Text-Based Multiple Choice Question Answering. ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Jul 2018, Melbourne, Australia. pp.272 - 277.
2. Welbl J., Nelson F. L., Gardner M. Crowdsourcing Multiple Choice Science Questions. CoRR 2017
3. Mikolov, T. et al. (2013). Distributed Representations of Words and Phrases and their Compositionality. Advances in NIPS 26 (pp. 3111--3119)
4. Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2016). Enriching Word Vectors with Subword Information
5. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In .