Processing Graphs With Neutral Networks

Cheng Guan

May 27, 2018

1 Processing Graphs With Neutral Networks

We now describe a deep neural network suitable for processing the question and scene graphs to infer an answer. See Fig. 1 for an overview. The two graphs representing the question and the scene are processed independently in a recurrent architecture. We drop the exponents S and Q for this paragraph as the same procedure applies to both graphs. Each node x_i is associated with a gated recurrent unit (GRU [1]) and processed over a xed number T of iterations (typically T=4):

$$h_i^0 = 0 \tag{1}$$

$$n_i = pool_j\left(e_{ij}{}' \circ x_j{}'\right) \tag{2}$$

$$h_i^t = GRU\left(h_i^{t-1}, [x_i{}'; n_i]\right) \tag{3}$$

Square brackets with a semicolon represent a concatenation of vectors, and $\circ$ the Hadamard (element-wise) product. The nal state of the GRU is used as the new representation of the nodes: $x_i" = h_i^T$. The pool operation transforms features from a variable number of neighbours (i.e. connected nodes) to a xed-size representation. Any commutative operation can be used (e.g. sum, maximum). In our implementation, we found the best performance with the average function, taking care of averaging over the variable number of connected neighbours. An intuitive interpretation of the recurrent processing is to progressively integrate context information from connected neighbours into each nodes own representation. A node corresponding to the word ball, for instance, might thus incorporate the fact that the associated adjective is red. Our formulation is similar but slightly different from the gated graph networks [3],as the propagation of information in our model is limited to the rst order. Note that our graphs are typically densely connected.

We now introduce a form of attention into the model, which constitutes an essential part of the model. The motivation is two-fold: (1) to identify parts of the input data most relevant to produce the answer and (2) to align specic words in the question with particular elements of the scene. Practically, we estimate the relevance of each possible pairwise combination of words and objects. More precisely, we compute scalar matching weights between node sets $\left\{x'^Q\right\}$ and $\left\{x'^S\right\}$. These weights are comparable to the attention weights in other models [2]. Therefore, $\forall i \in 1...N^Q, j \in 1...N^S$:

$$a_{ij} = \sigma\left(W_5\left(\frac{x_i'^Q}{\left\|x_i'^Q\right\|} \circ \frac{x_i'^Q}{\left\|x_i'^Q\right\|}\right) + b_5\right) \tag{4}$$

where $W_5 \in \mathbb{R}^{1\times h}$ and $b_5 \in \mathbb{R}$ are learned weights and biases, and σ the logistic function that introduces a nonlinearity and bounds the weights to (0,1). The formulation is similar to a cosine similarity with learned weights on the feature dimensions. Note that the weights are computed using the initial embedding of the node features (pre-GRU). We apply the scalar weights a_{ij} to the corresponding pairwise combinations of question and scene features, thereby focusing and giving more importance to the matched pairs (Eq.5). We sum the weighted features over the scene elements (Eq.6) then over the question elements (Eq.7), interleaving the sums with afne projections and non-linearities to obtain a nal prediction:

$$y_{ij} = a_{ij} \cdot \left[x_i''^Q; x_j''^S\right] \tag{5}$$

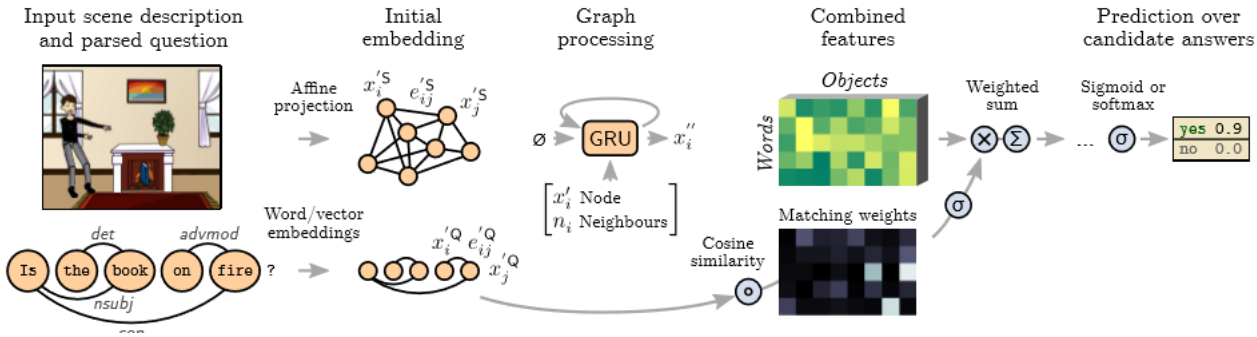

Figure 1: Architecture of the proposed neural network. The input is provided as a description of the scene (a list of objects with their visual characteristics) and a parsed question (words with their syntactic relations). The scene-graph contains a node with a feature vector for each object, and edge features that represent their spatial relationships. The question-graph reects the parse tree of the question, with a word embedding for each node, and a vector embedding of types of syntactic dependencies for edges. A recurrent unit (GRU) is associated with each node of both graphs. Over multiple iterations, the GRU updates a representation of each node that integrates context from its neighbours within the graph. Features of all objects and all words are combined (concatenated) pairwise, and they are weighted with a form of attention. That effectively matches elements between the question and the scene. The weighted sum of features is passed through a nal classier that predicts scores over a xed set of candidate answers.

$$y_i^{'} = f\left(W_6 \sum_j^{N^S} y_{ij} + b_6\right) \tag{6}$$

$$y_i^{''} = f^{'}\left(W_7 \sum_i^{N^Q} y_i^{'} + b_7\right) \tag{7}$$

with W_6, W_7, b_6, b_7 learned weights and biases, f a ReLU, and $f^{'}$ a softmax or a logistic function . The summations over the scene elements and question elements is a form of pooling that brings the variable number of features (due to the variable number of words and objects in the input) to a xed-size output. The nal output vector $y^{''} \in \mathbb{R}^T$ contains scores for the possible answers, and has a number of dimensions equal to 2 for the binary questions of the balanced dataset, or to the number of all candidate answers in the abstract scenes dataset. The candidate answers are those appearing at least 5 times in the training set (see supplementary material for details).

References

[1] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *In Proceedings of Conference Empirical Methods in Natural Language Processing*, 2014.

[2] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.0593*, 2015.

[3] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. *arXiv preprint arXiv:1606.00061*, 2016.