Simple Book Example

TeXstudio Team

October 26, 2022

Contents

Chapter 1

The First Chapter

Main Task:

Es müssen mathematische und algorithmische Grundlagen für neurosymbolische KI-Systeme entwickelt werden. Ein ganzheitlicher Ansatz sollte die Perspektive der Verwendung verallgemeinerbarer und robuster neuronaler Netzmodelle mit der Ausdrucks- und Argumentationskraft symbolischer Systeme im Entwurfsprozess verbinden. Zu diesem Zweck sollen neuartige neurosymbolische Modelle und dynamische Architekturen entwickelt werden, um in Gegenwart von strukturierten, verrauschten und heterogenen Daten effektiv zu lernen.

The goal of neural-symbolic AI(NeSy) is to integrate symbolic reasoning and neural networks, where the first topic focus on the task of **reasoning** and the second topic focus on the task of **learning**.

NeSy is different from statistical relational learning and artificial intelligence (StarIAI), which is also a domain that focus on the integration of learning and reasoning.

[5] talked about neural symbolic AI.

Image question answering often requires multiple steps of reasoning. [7]

Chapter 2

Dataset

2.1 CLEVR

CLEVR[2] is a dataset for *image reasoning.* CLEVRER is a dataset for *video reasoning.*

Objects and Relationships

- Three object shapes: cube, sphere, and cylinder.
- Two absolute sizes: small and large.
- Two materials: shiny and matte.
- Eight colors.
- Four relationships: left, right, behind, in front

Scene representation A scene can be represented by a scene graph.

Image generation Images are generated by randomly sampling a scene graph and render it using Blender.

Question representation Each question is associated with a *functional program* that can be executed on an image's scene graph.

Functional programs are built from simple basic building blocks, which are elementary operations of visual reasoning, such as querying object attributes, counting sets of objects, or comparing values.

Question size is defined as the number of functions in its program. Effective question is defined as the smallest equivalent program. Effective size is the size of effective question. Large effective size leads to long reasoning chains, which gives a higher error rate.

Complex questions can be represented by compositions of simple building blocks.

Question families A question family contains a template for constructing functional programs and several text templates providing multiple ways of expressing these programs in natural language.

Question generation

1. choose a question family (depth-first search)
2. select values for each of its template parameters
3. execute the resulting program on the image's scene graph to find the answer
4. use one of the text templates from the question family to generate the final natural-language question

Question topology

1. chain-structured, *what color is the cube to the right of the yellow sphere.*
2. tree-structured, maybe more difficult, require models to perform two sub-tasks in parallel before fusing their results. *How many cylinders are in front of the small thing and on the left side of the green object.*

2.2 DAQUAR

This dataset is proposed in [3].

- 6795 training questions on 795 images
- 5673 test questions on 654 images
- question types
  - object
  - color
  - number

2.3 COCO-QA

This dataset is proposed in [4].

- 78736 training questions on 8000 images
- 38948 test questions on 4000 images
- question types
  - object
  - color
  - number
  - location

2.4 VQA

This dataset is proposed in [1].

- 0.25M images
- 0.76M questions
- 10M answers

Chapter 3

Language Models

3.1 LSTM

LSTM is used for question analyzing tasks. It can reserve the state of a sequence.

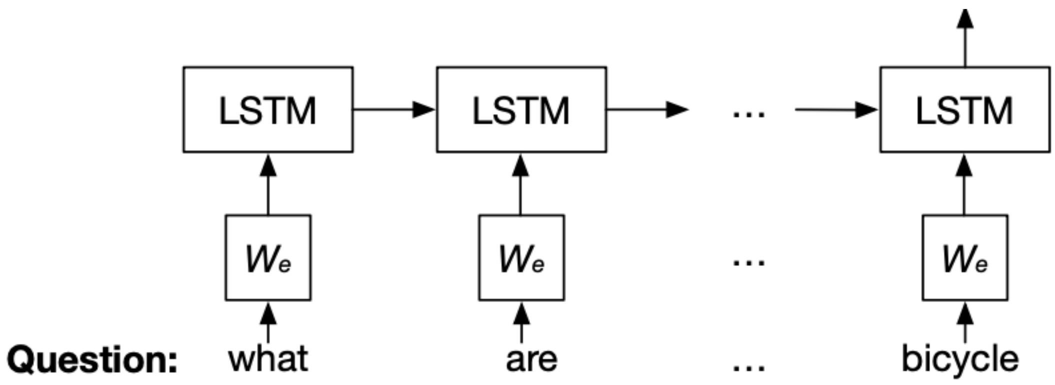

Figure 3.1: LSTM based question model *what are sitting in the basket on a bicycle.* [7]

Given the question

$$q = [q_1, ..., q_T]$$

where q_t is the one hot vector representation of word at position t. We embedded the words to a vector space through an embedding matrix and get the input vector x_t,

$$x_t = W_e q_t, t \in \{1, 2, ..., T\}$$

Then for every time step, we feed the embedding vector of words in the question to LSTM, then output a hidden state h_t,

$$h_t = LSTM(x_t), t \in \{1, 2, ..., T\}$$

The LSTM update process uses the gate mechanism. An input gate i_t controls how much the current input x_t updates the memory cell.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

The forget gate f_t controls how much information from past state c_{t-1} is preserved.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$

The output gate o_t controls how much information of the memory is fed to the output as hidden state

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

The memory cell c_t reserves the state of a sequence

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

The output hidden state h_t is updated as

$$h_t = o_t \tanh(c_t)$$

The final hidden layer is taken as the representation vector for the question, i.e.

$$v_Q = h_T$$

3.2 CNN

Similar to LSTM based model, CNN model embeds words to **word vectors**

$$x_t = W_e q_t$$

Then it concatenates the word vectors to **question vectors**

$$x_{1:T} = [x_1, x_2, ..., x_T]$$

Let c denotes the size of convolution filters, t denotes the t-th convolution output, W_c is the convolution weight, b_c is the bias. The convolution output is given by

$$h_{c,t} = \tanh(W_c x_{t:t+c-1} + b_c)$$

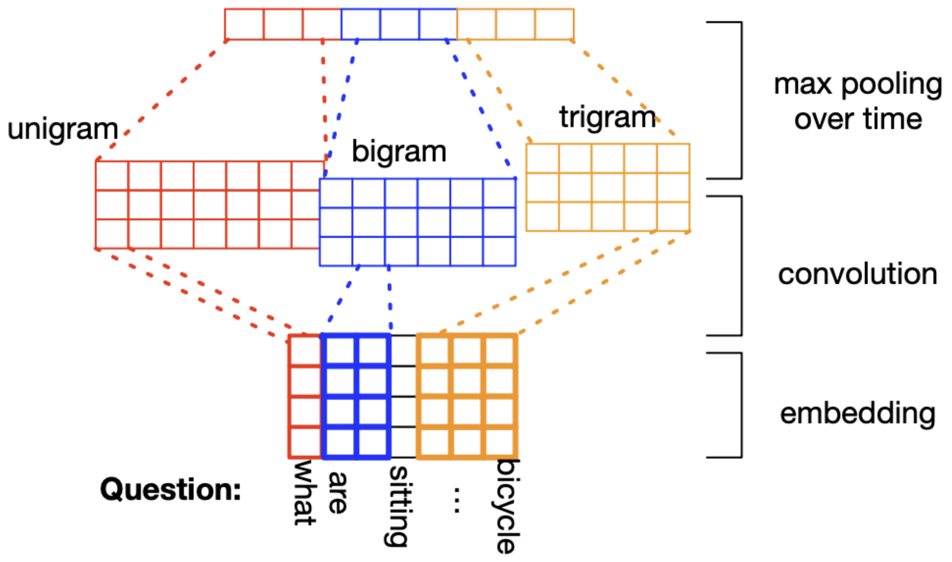

Figure 3.2: CNN based question model *what are sitting in the basket on a bicycle.* [7]

The **feature map** with convolution filter size c is given by

$$h_c = [h_{c,1}, h_{c,2}, ..., h_{c,T-c+1}]$$

Then we apply max-pooling over the feature maps of the convolution size c and denote it as

$$\tilde{h}_c = \max_t [h_{c,1}, h_{c,2}, ..., h_{c,T-c+1}]$$

For convolution feature maps of different sizes c, we concatenate them to form the feature representation vector of the whole question sentence

$$h = [\tilde{h}_1, \tilde{h}_2, \tilde{h}_3]$$

Hence the CNN based question vector is

$$v_Q = h$$

3.3 Stacked Attention Networks (SAN)

Let $v_I \in \mathbb{R}^{d \times m}$ denotes the image feature matrix, m is the number of image regions. $v_Q \in \mathbb{R}^d$ is a d dimensional vector. denotes the question feature vector. Let $W_{I,A}, W_{Q,A} \in \mathbb{R}^{k \times d}$, $\oplus$ denotes the addition of a matrix and a vector, which is performed by adding each column of the matrix by the vector. Then the two vectors first go through a single layer neural network

$$h_A = \tanh(W_{I,A}v_I \oplus (W_{Q,A}v_Q + b_A))$$

Let $W_P \in \mathbb{R}^{1\times k}$, then a softmax function is used to generate the attention distribution $p_I \in \mathbb{R}^m$ over the regions of the image, which corresponds to the attention probability of each image region given v_Q.

$$p_I = \text{softmax}(W_P h_A + b_P)$$

Based on the attention distribution we calculate the weighted sum of the image vectors $\tilde{v}_i$.

$$\tilde{v}_I = \sum_i p_i v_i$$

Then $\tilde{v}_i$ is combined with the question vector v_Q to form a refined **query vector** u.

$$u = \tilde{v}_I + v_Q$$

The k-th attention layer is calculated as

$$\begin{aligned} h_A^k &= \tanh(W_{I,A}^k v_I \oplus (W_{Q,A}^k u^{k-1} + b_A^k)) \\ p_I^k &= \text{softmax}(W_P^k h_A^k + b_P^k) \end{aligned} \tag{3.1}$$

where u^0 is initialized to be v_Q. The aggregated image feature vector is added to the previous query vector to form a new query vector:

$$\begin{aligned} \tilde{v}_I^k &= \sum_i p_i^k v_i \\ u^k &= \tilde{v}_I^k + u^{k-1} \end{aligned} \tag{3.2}$$

It will be repeat for K times and then use the final u^K to infer the answer:

$$p_{ans} = \text{softmax}(W_u u^K + b_u)$$

3.4 Convolutional bottleneck attention module (CBAM)

The CBAM[6] module is designed to learn what and where to emphasize or suppress and refines intermediate features effectively.

Given an intermediate feature map $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$ as input, CBAM sequentially infers a 1D channel attention map $\mathbf{M}_c \in \mathbb{R}^{C\times 1\times 1}$ and a 2D spatial

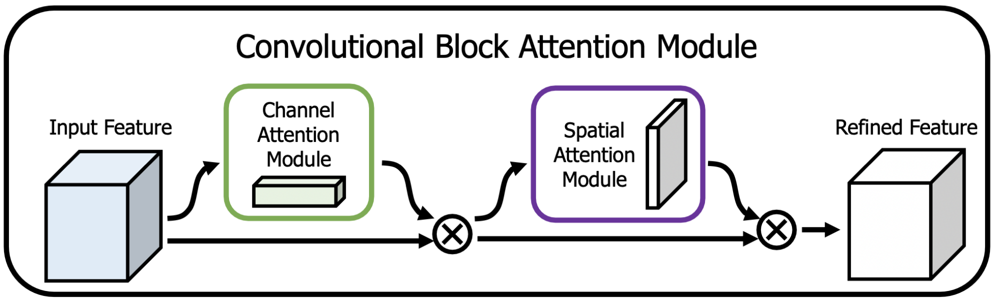

Figure 3.3: The overview of CBAM [6]

attention map $\mathbf{M}_s \in \mathbb{R}^{1\times H\times W}$ as illustrated in Figure 3.3. The overall attention process can be summarized as

$$\begin{aligned} \mathbf{F}' &= \mathbf{M}_c(\mathbf{F}) \otimes \mathbf{F} \\ \mathbf{F}'' &= \mathbf{M}_s(\mathbf{F}') \otimes \mathbf{F}' \end{aligned} \tag{3.3}$$

where $\otimes$ denotes element-wise multiplication.

The channel attention module and spatial attention module are illustrated in figure 3.4.

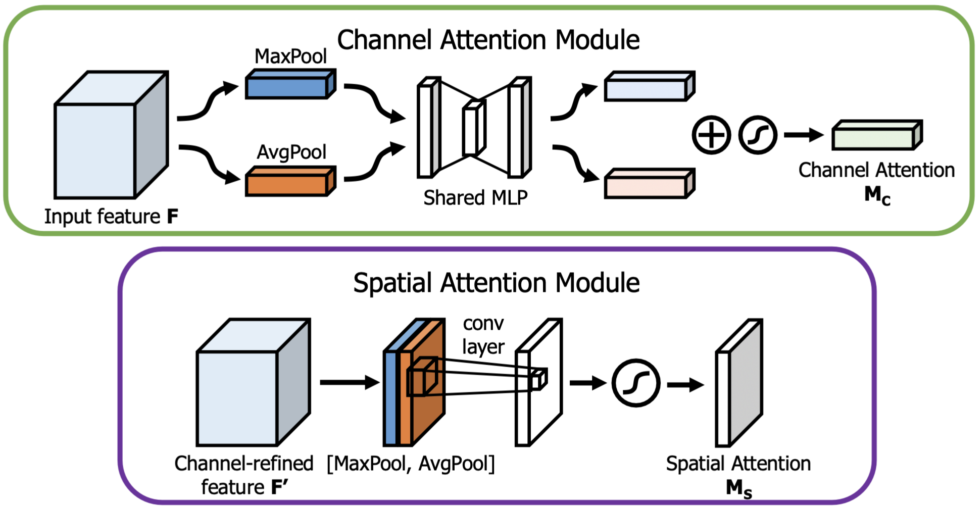

Figure 3.4: Diagram of each attention sub-module.[6]

3.4.1 Channel attention module

The channel attention module focus on what is meaningful given an input image.

Chapter 4

Evaluation

4.1 Metrics

4.1.1 MIN

For the open-ended task, [1] uses following metrics

$$\text{accuracy} = \min(\frac{\text{\# humans that provided that answer}}{3}, 1)$$

Thus if at least 3 workers provided the answer, then the answer is 100% accurate.

4.2 Baselines

4.2.1 per Q-type prior

For the open-ended task, pick the most popular answer per question type. [1]

4.2.2 nearest neighbor

Given a test image, question pair, first find the K nearest neighbor questions and associated images from the training set. Then, for the open-ended task, pick the most frequent ground truth answer from this set of nearest neighbor question, image pairs.

Bibliography

[1] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Batra, and D. Parikh. Vqa: Visual question answering, 2015. URL `https://arxiv.org/abs/1505.00468`.

[2] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning, 2016. URL `https://arxiv.org/abs/1612.06890`.

[3] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input, 2014. URL `https://arxiv.org/abs/1410.0210`.

[4] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering, 2015. URL `https://arxiv.org/abs/1505.02074`.

[5] Z. Susskind, B. Arden, L. K. John, P. Stockton, and E. B. John. Neuro-symbolic ai: An emerging class of ai workloads and their characterization, 2021. URL `https://arxiv.org/abs/2109.06133`.

[6] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cbam: Convolutional block attention module, 2018. URL `https://arxiv.org/abs/1807.06521`.

[7] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering, 2015. URL `https://arxiv.org/abs/1511.02274`.