A MULTIMODAL APPROACH FOR MATHEMATICAL REASONING VIA SYMBOLIC UNDERSTANDING

Abhishek Sinha* **& Kumar Ayush** *
{abhsinha, kayush}@adobe.com

ABSTRACT

This paper presents a new direction for the visual question answering task. Given an image with a simple linear algebraic equation system and a question in natural language based on the variables in the image, we propose an end-to-end deep learning model that produces accurate answers to questions pertaining to the value of the variables and other related questions. Modeling the problem of solving simple linear equations as a VQA task makes it interesting as the system now requires three kinds of understanding a) visual understanding to recognize digits, variables, operators and equal sign b) conceptual understanding of the symbolic meanings of coefficients, constants, variables, operators and equality and realizing the role of numbers as mathematical entities which can undergo mathematical operations and c) high level understanding of the interaction between the image and the questions in order to accurately answer them. We also create an open-source dataset for the same and compare the performance of our model with different baselines.

1 INTRODUCTION

Solving simple linear algebraic equations requires understanding of the meaning behind important concepts such as coefficients, constants, variables, operators and the equal sign. Symbolic understanding (Magruder (2012)) is essential to solve such equations because of the multiple meanings and roles the mathematical symbols hold, such as the differing roles of 5 in the two expressions, 5 and $5x$ (in the first example, 5 is a constant, in the second, 5 is a coefficient), the role of minus sign in the following cases: unary sign (-7), a binary sign $(2x-7y)$ or an operation sign $(7-3)$. Humans with such symbolic understanding of coefficients, constants, variables, operators and the equal sign are easily able to solve simple linear algebraic equations.

In case of very simple linear equations like (1) and (2)

$$2x + y = 6 \tag{1}$$

$$y = 4 \tag{2}$$

humans with symbolic understanding usually follow simple substitution steps and moving variables/constants to either side of the equality sign to solve them. This paper is specifically targeted towards solving simple linear algebraic equations from images and answering questions based on the equations using deep learning.

Deep learning models have performed remarkably well in several domains such as computer vision, natural language processing and speech recognition. These models are able to achieve high accuracy in various tasks and hence their recent popularity. Evans & Grefenstette (2018) have proposed a Differentiable Inductive Logic framework, a model hybridized with neural networks, and demonstrated its working on induction tasks such as *less than* where the model is given a pair of images representing numbers, and has to output a label (0 or 1) indicating whether the number of the left image is less than the number of the right image.

One complex mutlimodal task which has garnered a lot of attraction is Visual Question Answering (Antol et al. (2015), Ben-younes et al. (2017)) which aims at answering a question about an image. In this work, we aim to solve a simple linear equation system by modelling them as a visual question

*Equal Contribution

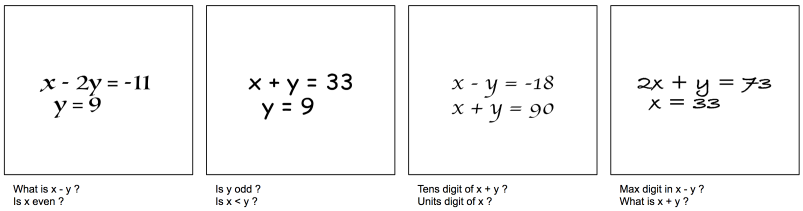

Figure 1: Examples of simple linear algebraic equations in the form of images with varying fonts along with the questions.

answering task wherein the equations are provided in the form of an image and questions are based on the values of the variables involved in the equation system and other complex questions such as a) Is $x < y$?, b) What is the value of $x + y$?, c) Is x odd ? d) Units digit of y ?, e) Max digit in $x - y$?, etc.

Our main contributions in this work are summarized below:

- To the best of our knowledge, this is the first work which aims at solving simple linear equations from images in an end-to-end fashion. The motivation stems from the fact that humans with symbolic understanding find it very easy to solve such equations when presented as images.
- We model this problem as a VQA task wherein we ask questions which are based not only on the final numerical values of the variables but other complex questions. We create an end-to end model for the task and compare our model with various baselines and show that our model performs better than them.

2 PROPOSED METHOD

The network architecture is visualized in Fig. 2. In this section, we explain our approach in more detail.

2.1 IMAGE EMBEDDING

We use a convolutional neural network (CNN) model to compute a high level representation of the input image I. The CNN module receives a 70×70 image and passes it through a series of ReLU activated convolutional layers. At the last layer, it produces a 5×5×64 three dimensional tensor. Let CR(f,r,s) denote a convolutional layer with Relu activation. The layer operates with a receptive field of $r \times r$ with stride of s along height and width and yields f filters. The architecture of the CNN network is as follows: CR(32,5,2)-CR(32,5,1)-CR(64,5,2)-CR(64,4,1)-CR(64,3,1).

2.2 QUESTION EMBEDDING

A single layered GRU(Chung et al. (2014)) is used to obtain a 64-dim embedding for the question. Each question word is encoded as a 21-dim one hot vector (input question vocabulary size is 21). We tokenize and encode a given question q into one hot word vectors $O_q = \{o_1, o_2, ..., o_P\}$ where P is the number of words in the question. The word encodings are then fed to the GRU.

2.3 FUSION SCHEME AND ANSWER COMPUTATION

The image and the question embeddings are fused to obtain a single embedding. The 64-dim question embedding is treated as a 1×1 kernel of depth 64. Then, the 5×5×64 dimensional image embedding is convolved with the above kernel to obtain a 5×5×1 embedding. This combined embedding is then flattened to a 25-dim embedding which is then passed through two fully connected layers of size 128 each to obtain two 128-dim embeddings. These embeddings are then used as the initial hidden state and cell state of a LSTM(Hochreiter & Schmidhuber (1997)) network which is used as a decoder.

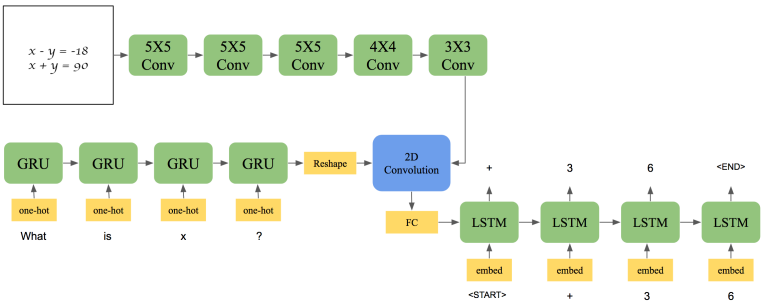

Figure 2: An overview of our model. The three dimensional image embedding is convolved with the reshaped question embedding to obtain the combined embedding.

The LSTM network produces the answer by generating one word/digit at every time step conditioned on the previous hidden state and the previously generated word/digit.

$$p(y_t) \propto exp(L_o(Ey_{t-1} + L_h h_t)) \tag{3}$$

The output vocabulary consists of the following words: +, -, True, False, 0-9, <START>, <END> and <PAD>. Therefore, the output vocabulary size is 17 and the words in output vocabulary are represented as 17-dim one-hot vectors.

Order of digits: During simple addition and subtraction of numbers, humans first find out the sign of the numerical result and then compute the result from unit's digit to ten's digit and so on. In order to investigate this intuition, we model two variants (See Fig. 3) of the decoder LSTM network with respect to the order of digits in the case of numerical answers i.e. a) sign-ten's digit-unit's digit (left to right - LTR) and b)sign-unit's digit-ten's digit (right to left - RTL).

3 Experiments

3.1 Dataset

We create a dataset[1] containing image-question pairs based on 10 simple linear equations and 20 different question types. We randomly vary the fonts while creating the images. While creating the dataset, we consider only integer values of x and y between 0-99. Also, the values of expressions like $x + y$, $x - y$ (in the questions) are considered between 0-99. Such a restriction has been done to keep the unrolling of the decoder LSTM network to a maximum of 4 time steps (for example, an output of 75 will unroll as +, 7, 5, <END> and an output of True will unroll as True, <END>.). Fig.1 shows a few examples from our dataset (image-question pairs). The train dataset consists of 3,55,000 image-question pairs and the testset comprises 60,000 image-question pairs.

3.2 Baselines

In this section, we discuss the various baselines. For the purpose of comparison, we keep the modules for generating image embedding, question embedding and LSTM decoder network (LTR) same and vary only the fusion scheme based on previous works for combining the image and the question embeddings.

- **Concatenation (Antol et al. (2015)):** The image embedding is flattened and concatenated with the question embedding.
- **Multiplication (Antol et al. (2015), Lu et al. (2015)):** The image embedding is flattened and passed through fully connected layers to get a 64-dim embedding. This embedding is then multiplied in an element-wise fashion with the question embedding .
- **Neural VQA fusion Scheme (Ren et al. (2015)):** The image embedding is flattened and passed through a FC layer to get a 32-dim embedding. Similarly, the words in the question

[1]Dataset and code can be found at this link.

are embedded into a 32 dimensional embedding. The image embedding is then passed as input to the single layered GRU at t_0 and then the remaining embeddings of question words are passed subsequently.

Table 1: Comparison with baseline fusion schemes on test dataset. The first three columns represent percentage accuracy and the final column represents the mean error in case of numerical outputs.

	T/F	Num	All	Mean Error
Ours (LTR)	85.7	**80.3**	82.1	**5.9**
Ours (RTL)	**97**	79.7	**86**	11.2
Concatenation	67.6	34.9	40.8	–
Mutliplication	50.3	3.1	20.3	–
Neural VQA	50.5	3.2	20.1	–

3.3 RESULTS

We compare our model with the baselines in terms of percentage accuracy on the test dataset. The answers fall under two categories: True/False and Numbers. Table 1 shows that our model outperforms the baseline models by huge margins for both answer types. It can be seen that the accuracy for True/False questions for the Multiplication and Neural VQA schemes are 50% which is as good as randomly answering the T/F questions. We would like to add that we hypothesized better performance from the RTL variant than the LTR one for numerical answers. However, this is not case and we would like to explore it further in future work.

4 CONCLUSION AND FUTURE WORK

In this paper, we proposed an end-to-end multimodal deep learning model to solve simple linear algebraic equations from images. We show that our fusion scheme outperforms other baseline schemes by a huge margin. We also create an open-source dataset for this task. We see that the model learns to treat numbers as mathematical entities which can be added and subtracted i.e. the model is able to understand the symbolic meaning of numbers that can undergo mathematical operations, which is the same way as we humans treat numbers. Our preliminary work advocates future research towards such mathematical reasoning from images using deep neural networks. Currently, our model fails to generalize over question types never seen during the training, like "What is $y - x$?" instead of "What is $x - y$"?. There are some immediate extensions possible from the presented work. We restricted the values of x and y in our dataset to be integers with atmost two digits; such restriction can be relaxed to three or more digits. Another possibility is the use of attention based model to address the task.

REFERENCES

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010. 6

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2425–2433, 2015. 1, 3

Hedi Ben-younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 1, pp. 3, 2017. 1

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2

Richard Evans and Edward Grefenstette. Learning explanatory rules from noisy data. In *Journal of Artificial Intelligence Research*, volume 61, pp. 1–64, 2018. 1

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997. 2

Jiasen Lu, Xiao Lin, Dhruv Batra, and Devi Parikh. Deeper lstm and normalized cnn visual question answering model. `https://github.com/VT-vision-lab/VQA_LSTM_CNN`, 2015. 3

Robin Lee Magruder. *Solving linear equations: A comparison of concrete and virtual manipulatives in middle school mathematics*. University of Kentucky, 2012. 1

Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. In *Advances in neural information processing systems*, pp. 2953–2961, 2015. 3

5 APPENDIX

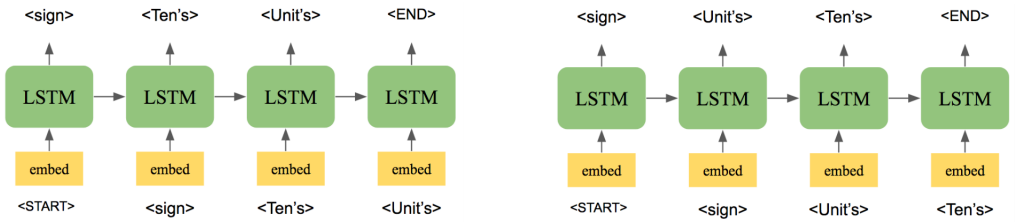

Figure 3: Different order of digits in the case of numerical outputs. Such a modeling is based on the intuition that in case of simple subtraction and addition of numbers, humans first find out the sign of the result and then compute the unit's digit, ten's digit and so on.

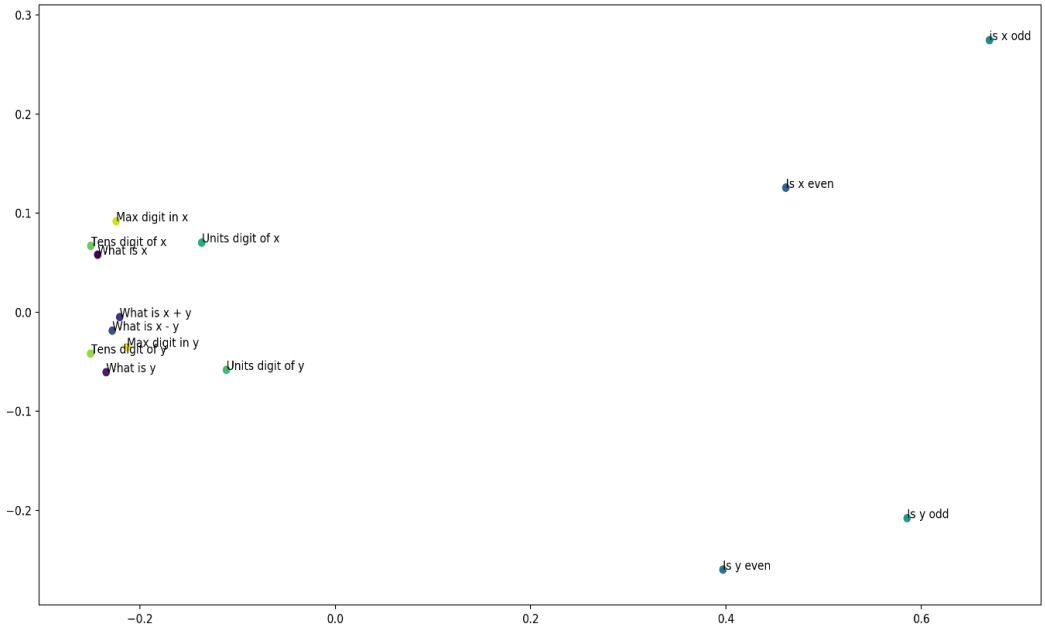

Figure 4: Visualization of question embeddings using Principal Component AnalysisAbdi & Williams (2010). It can be seen that questions related to a particular expression form a cluster. For example, questions pertaining to only x like Ten's digit of x?, Max digit in x?, What is x?, etc., are lying in a distinct cluster. Also, questions with True/False answers are lying far away from questions with numerical answers. This shows that the network is able to understand the difference between questions with T/F and numerical answers. Also, amongst questions with numerical answers, the network is able to learn the difference between different expressions like only x, only y, $x + y$, etc.