Topic:

Question Answering on Mathematics Dataset

SAPIENZA
UNIVERSITÀ DI ROMA

student:

Caterina Borzillo, 1808187

23/03/2023

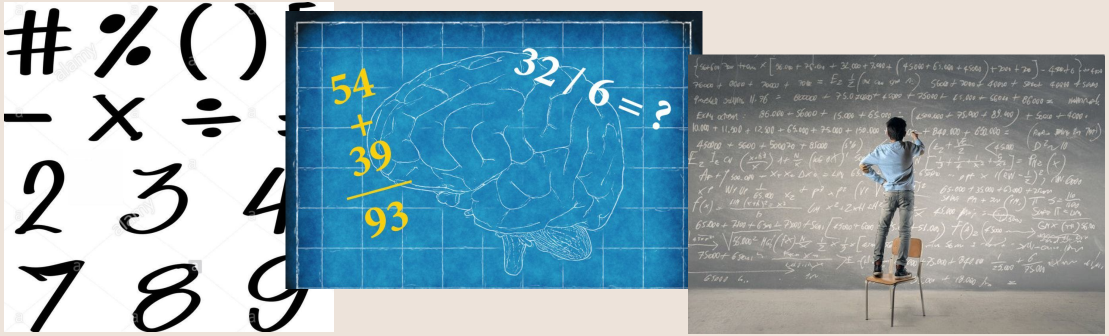
1. Task
2. Dataset
3. First baseline
4. Additional baseline
5. SOTA approach
6. Results
7. Final considerations

Task

The task is: question answering on a dataset that contains mathematics questions at school-level difficulty.

Question: Solve -42*r + 27*c = -1167 and 130*r + 4*c = 372 for r.

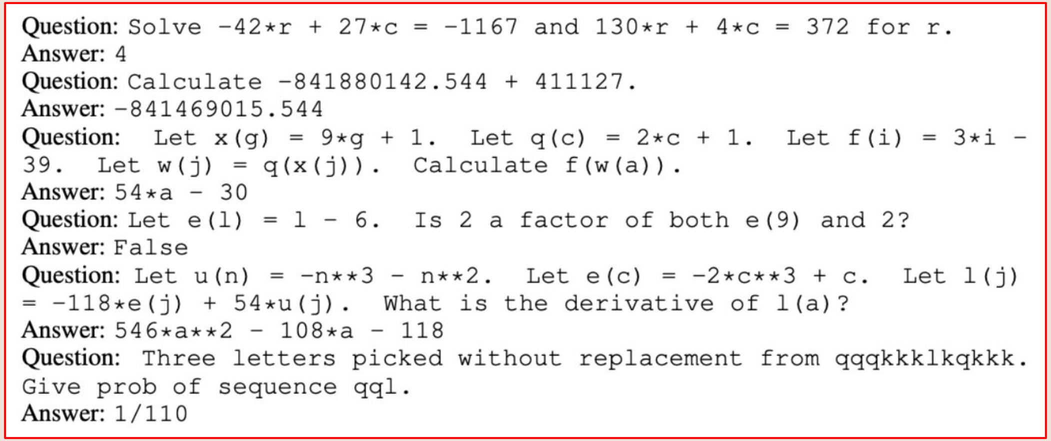
Answer: 4



1. Task
2. Dataset
3. First baseline
4. Additional baseline
5. SOTA approach
6. Results
7. Final considerations

Dataset

Mathematics dataset: set of math problems written in a free-form textual format.

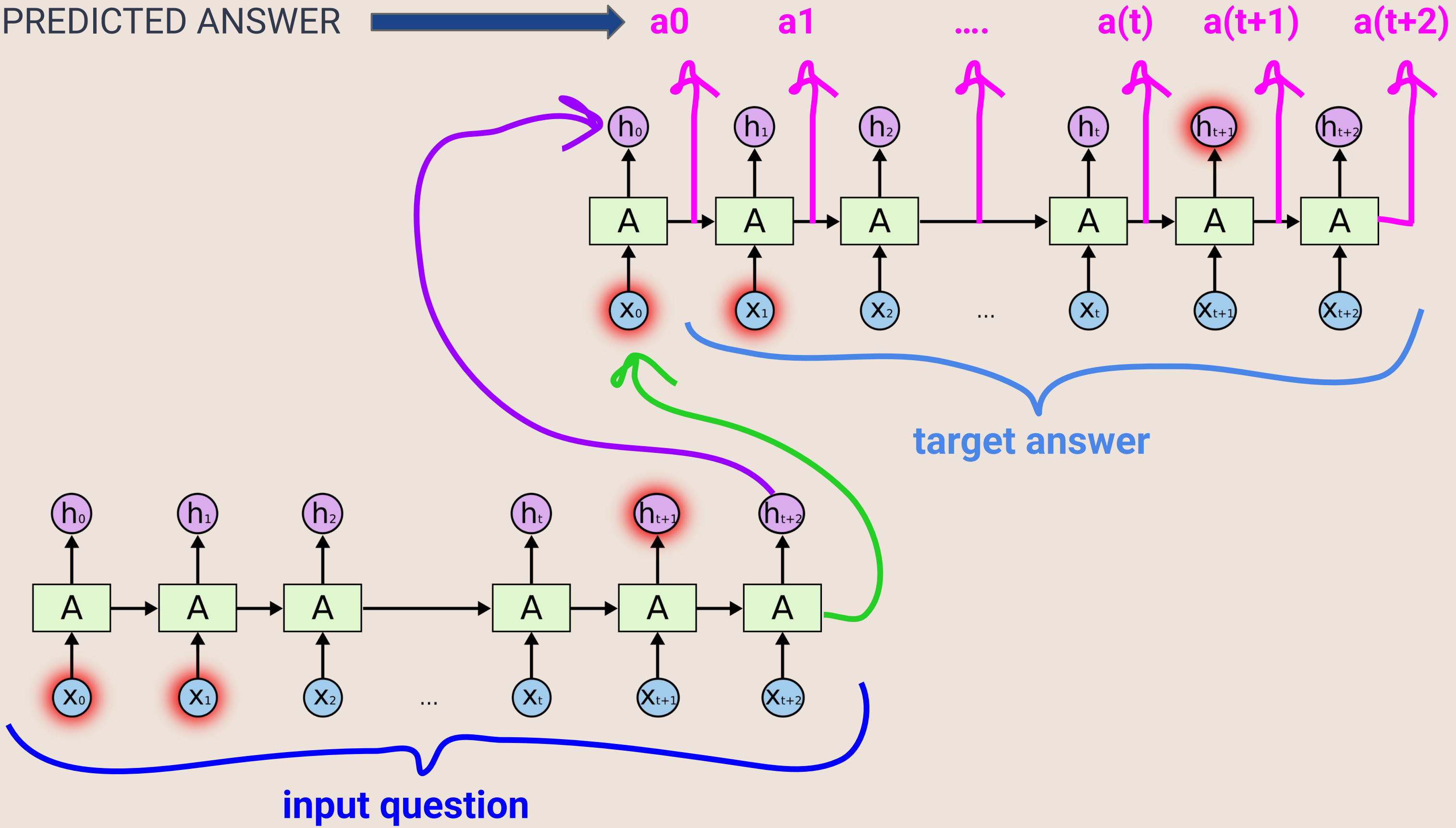


Question: Solve -42*r + 27*c = -1167 and 130*r + 4*c = 372 for r.
Answer: 4
Question: Calculate -841880142.544 + 411127.
Answer: -841469015.544
Question: Let x(g) = 9*g + 1. Let q(c) = 2*c + 1. Let f(i) = 3*i - 39. Let w(j) = q(x(j)). Calculate f(w(a)).
Answer: 54*a - 30
Question: Let e(l) = l - 6. Is 2 a factor of both e(9) and 2?
Answer: False
Question: Let u(n) = -n**3 - n**2. Let e(c) = -2*c**3 + c. Let l(j) = -118*e(j) + 54*u(j). What is the derivative of l(a)?
Answer: 546*a**2 - 108*a - 118
Question: Three letters picked without replacement from qqqkkklkqkkk. Give prob of sequence qql.
Answer: 1/110

Among all the mathematics areas, I solved the task on 5 sub-problems: algebra__linear_1d, arithmetic__add_or_sub, numbers__place_value, numbers__round_number, calculus__differentiate

1. Task
2. Dataset
3. First baseline
4. Additional baseline
5. SOTA approach
6. Results
7. Final considerations

First baseline: Simple LSTM



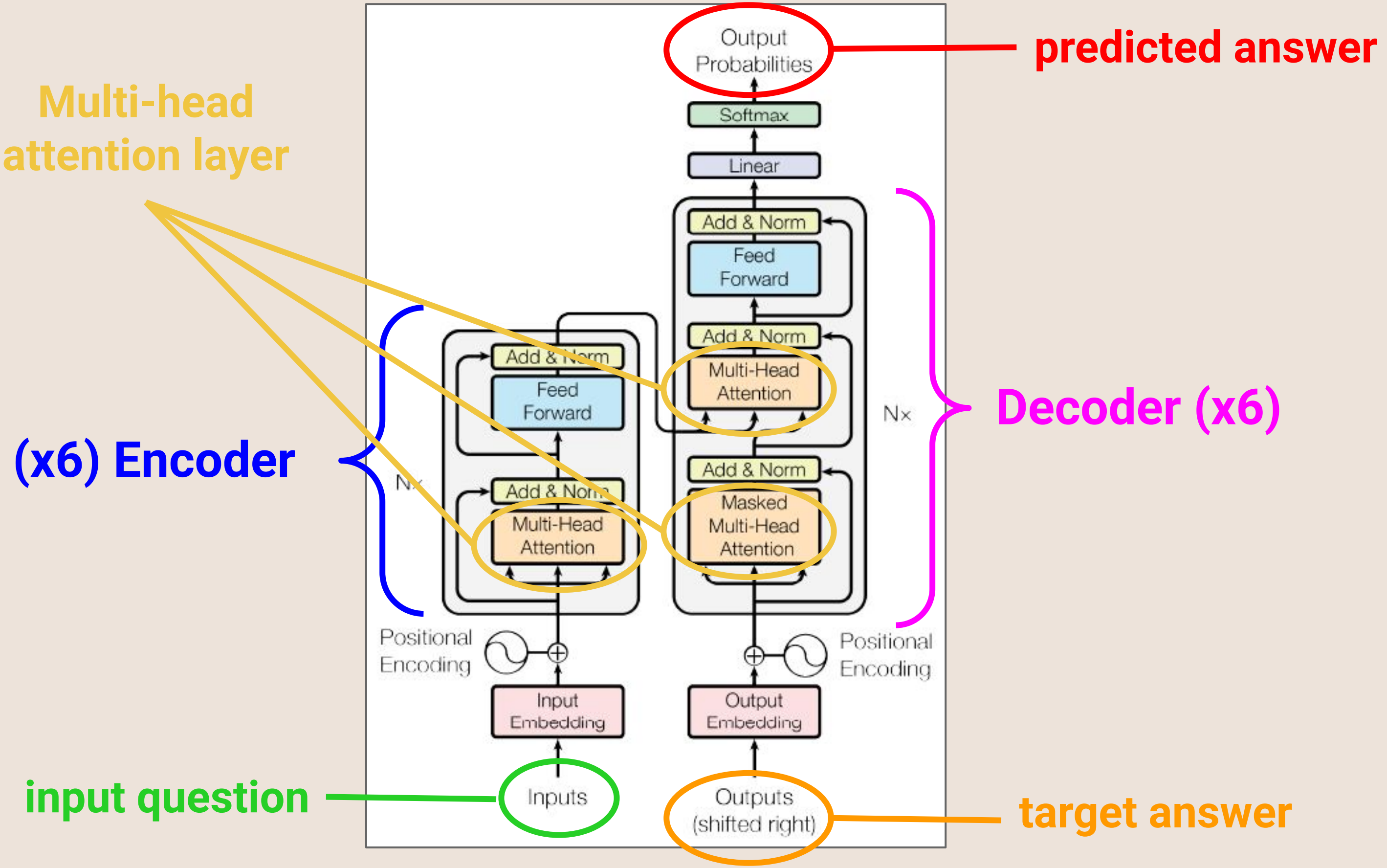
PREDICTED ANSWER

target answer

input question

1. Task
2. Dataset
3. First baseline
4. **Additional baseline**
5. SOTA approach
6. Results
7. Final considerations

Additional baseline: Transformer

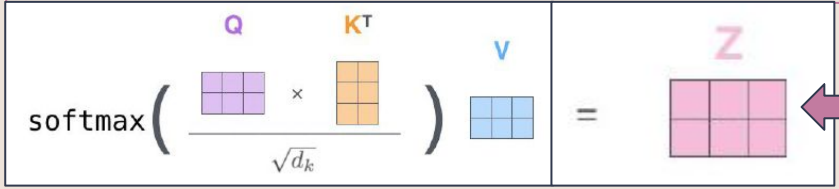


1. Task
2. Dataset
3. First baseline
4. Additional baseline
5. SOTA approach
6. Results
7. Final considerations

Multi-head attention (Transformer)

The Multi-Head attention mechanism is based on 3 parameters: **Query**, **Key** and **Value**.

Q, K and V are combined together to produce an **Attention Score** for each token in the sequence in this way (it is called the **scaled dot-product attention**) :



final attention scores related to each token in the sequence

The Multi-Head attention layer is used in three places:

- **Self-attention** in the Encoder
- **Self-attention** + **Encoder-Decoder attention** in the Decoder

Q=output of the decoder self-attention (target ans); K,V=output of the Encoder stack (input quest)
why? Because it is the **target answer** that pays attention to the **input question**.

Q,K,V are the same (Q,K,V=input quest or Q,K,V=target answer) why? Because:
-in the Encoder there is the **input question** that is focusing on itself
-in the Decoder there is the **target answer** that is focusing on itself

1. Task
2. Dataset
3. First baseline
4. Additional baseline
5. SOTA approach
6. Results
7. Final considerations

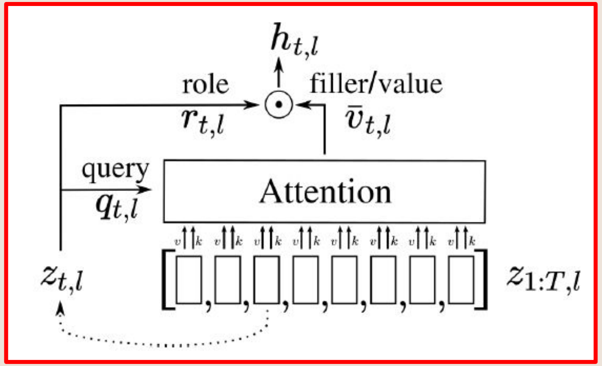
SOTA approach: TP-Transformer

The general architecture of the TP-Transformer is exactly the same of the Transformer BUT:

TP-Transformer introduces a **novel attention mechanism** called **TP-Attention**.

Novel because it aims to find and learn connections, relations between the tokens of the input question.

Here, **4** components are used: Query, Key, Value + **Role Vector** (or **Relation Vector**) where:
- the Role Vector serves as a representational space for the relations that the TP-Transformer has to capture between tokens in the sequence.

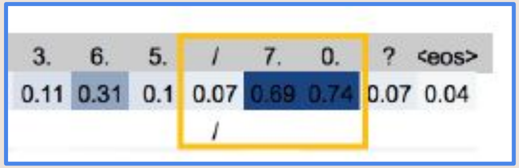


EXAMPLE: this attention can be interpreted as encoding a **relation** ***second-argument-to*** holding between the querying digits and the '/' operator:

INPUT QUESTION:

ATTENTION SCORES:

(on one head)



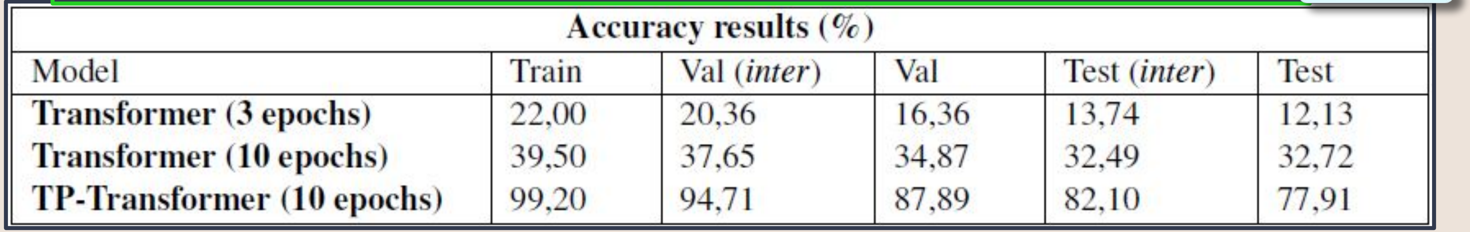
3.	6.	5.	/	7.	0.	?	<eos>
0.11	0.31	0.1	0.07	0.69	0.74	0.07	0.04
			/				

1. Task
2. Dataset
3. First baseline
4. Additional baseline
5. SOTA approach
6. Results
7. Final considerations

Results

Simple LSTM results: the accuracy obtained on 3 mathematics sub-problems is very low, about 3,7% during training phase and about 2,6% and 1,9% in validation and test phases.
I also tried to solve the task with the Simple LSTM on 1 mathematics problem only and the accuracy on the training set was about 5%.

Transformer and TP-Transformer results: there is a very noticeable increase in accuracy, considering also that the results here are obtained on 5 mathematics different sub-problems.



Accuracy results (%)

Model	Train	Val (*inter*)	Val	Test (*inter*)	Test
Transformer (3 epochs)	22,00	20,36	16,36	13,74	12,13
Transformer (10 epochs)	39,50	37,65	34,87	32,49	32,72
TP-Transformer (10 epochs)	99,20	94,71	87,89	82,10	77,91

! The Validation and Test step are not only computed on the validation and test sets of data but also on the training set (interpolation).

1. Task
2. Dataset
3. First baseline
4. Additional baseline
5. SOTA approach
6. Results
7. Final considerations

Final Considerations

- Simple LSTM method solves the problem in the most straightforward way but, it’s evident that the network **makes it very difficult** to capture the **meaning** of the input sequence and thus it is not able to generate a correct answer.

- Furthermore, the “help” that the LSTM recevices during training (target answer is given in input to the lstm) seems to be not enough especially in those cases where the **input sequences are long** and therefore the model struggles to remember the first part of the sequence.

1. Task
2. Dataset
3. First baseline
4. Additional baseline
5. SOTA approach
6. Results
7. Final considerations

Final Considerations

- the number of trainable parameters of the Simple LSTM is about an half with respect to the Transformer (due in particular to the attention mechanism), thus Simple LSTM learns much less information.
- Also, the total number of trainable parameters in the SOTA approach is higher with respect to the Transformer (due to the **role-vector computation**) and this could explain the **difference** in the results.
- However, as we can notice from the results, the **self-attention mechanism plays a fundamental role** in examining and analyzing sequence data.
- Finally, the novel attention mechanism, with the additional computation of the role vector, allows the model to learn a representational space in which the relations among characters is highlighted.

Thank you!

Caterina Borzillo, 1808187

23/02/2023