DATA SCIENCE RETREAT®
SINCE 2014

Deep Learning for Symbolic Math

Data Science Retreat Batch 21
Wenjuan Yang

What is Symbolic Math?

- Simple example: $\int x dx = \frac{x^2}{2}$ $\int_a^b$ ✗
- Complex example: $\int (x^2(\tan(x)^2 + 1) + 2x\tan(x) + 1)dx = x^2\tan(x) + x$
- More complex example:

$$\int \frac{16x^3 - 42x^2 + 2x}{(-16x^8 + 112x^7 - 204x^6 + 28x^5 - x^4 + 1)^{1/2}} dx = \sin^{-1}(4x^4 - 14x^3 + x^2)$$

Ref: Lample, Guillaume, and François Charton. "Deep learning for symbolic mathematics."

Can Deep Learning Learn Math?

Symbolic Math

- Language vs Math

- Statistic learning vs Rule-based Inference

Can Deep Learning Learn Math?

Symbolic Math

- Difference

  Language: Statistic way of learning

  Math: Rule base inference

- Similarity

  Pattern recogination

Background

Symbolic Math

DEEP LEARNING FOR SYMBOLIC MATHEMATICS

Guillaume Lample*
Facebook AI Research
glample@fb.com

François Charton*
Facebook AI Research
fcharton@fb.com

ABSTRACT

Neural networks have a reputation for being better at solving statistical or approximate problems than at performing calculations or working with symbolic data. In this paper, we show that they can be surprisingly good at more elaborated tasks in mathematics, such as symbolic integration and solving differential equations. We propose a syntax for representing mathematical problems, and methods for generating large datasets that can be used to train sequence-to-sequence models. We achieve results that outperform commercial Computer Algebra Systems such as Matlab or Mathematica.

2 Dec 2019

Facebook GitHub: https://github.com/facebookresearch/SymbolicMathematics

Code released on Mar 15

Outline

1. *Data Generation*
2. *Transformer Model*
3. *Results*
4. *Outlook*

Outline

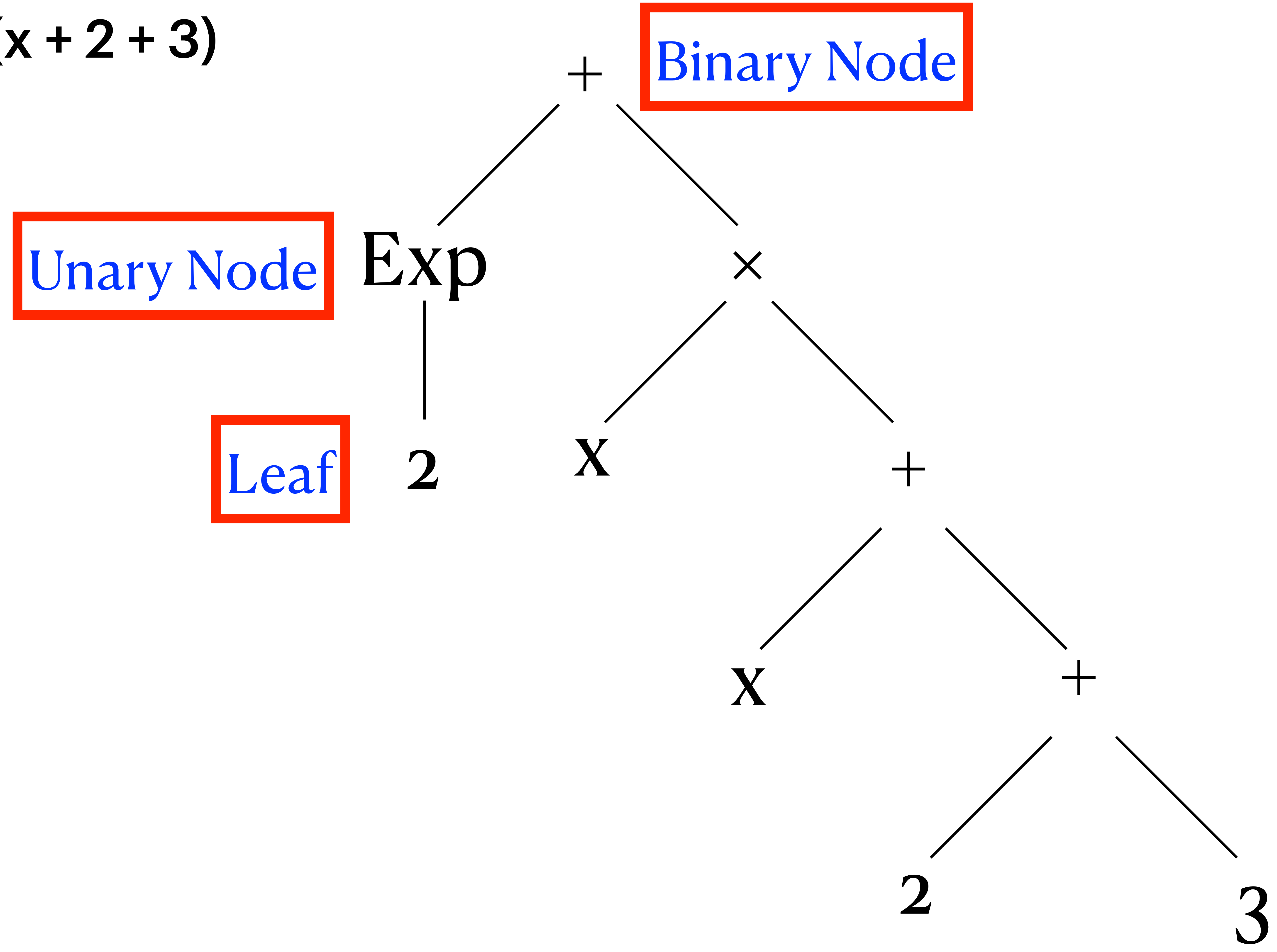
1. Data Generation
2. Transformer Model
3. Results
4. Outlook

How to Generate Data?

6 STEPS

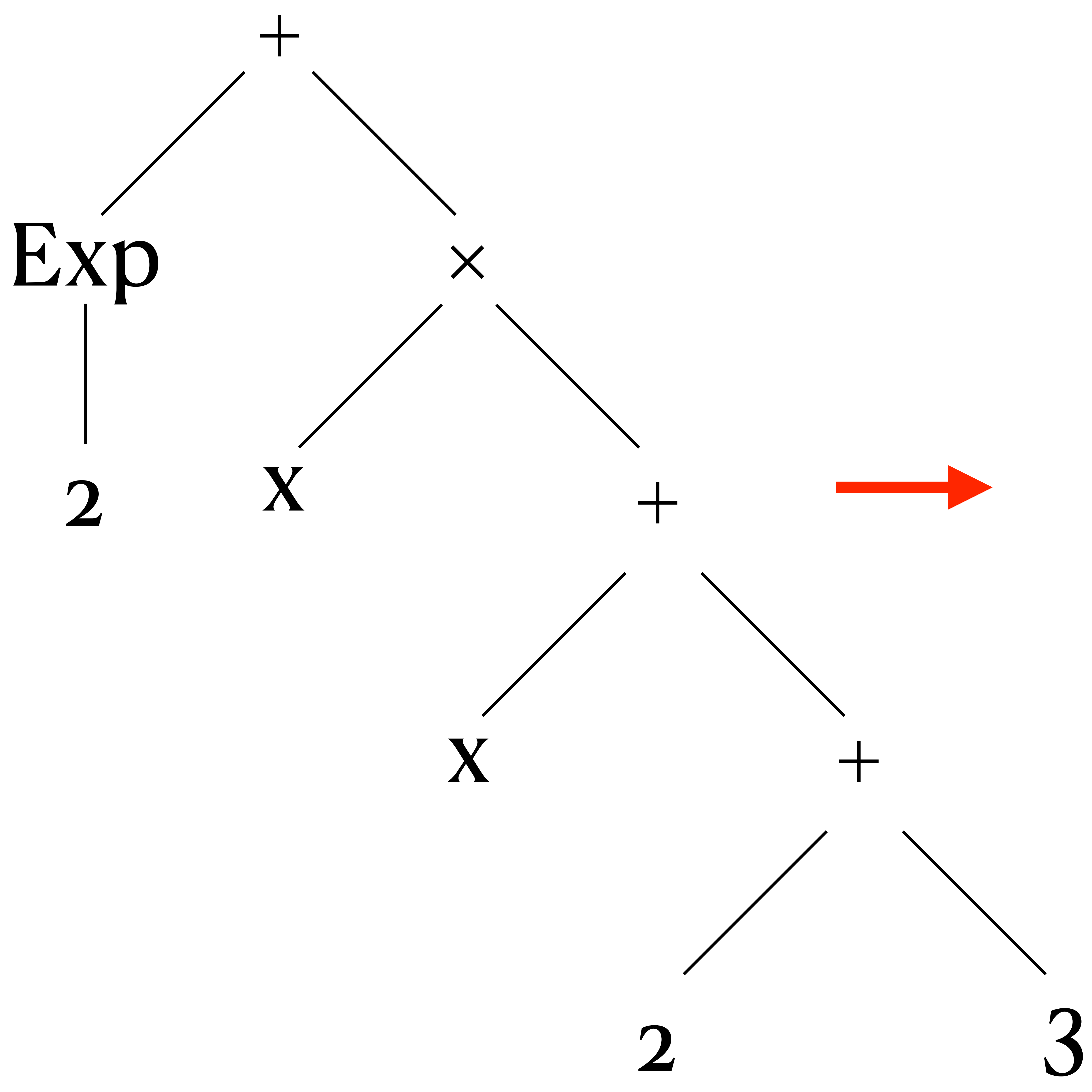
Random Tree

Example: Exp(2) + x (x + 2 + 3)



1

Random Tree → Prefix



Prefix

+ Exp 2 × x+ x + 2 3

Infix

Exp(2) + x (x + 2 + 3)

Prefix → Infix & Clean Data

$+ \,\mathrm{Exp}\, 2 \times x + x + 2\; 3 \longrightarrow \mathrm{Exp}(2) + x \times (x + 2 + 3)$

$\infty \quad -\infty \quad i$ ✗

Simplify

$\text{Exp}(2) + x \times (x + 2 + 3) \longrightarrow \text{Exp}(2) + x \times (x + 5)$ Infix

$\text{Exp}(2) + x \times (x + 5) \longrightarrow + \ \text{Exp} \ 2 \ \times \ x + x \ 5$ Prefix

Output

Differentiate

$$\mathrm{Exp}(2) + x \times (x + 5) \longrightarrow 2 \times x + 5$$

Infix → Prefix

DATA SCIENCE RETREAT®
SINCE 2014

$2 \times x + 5 \longrightarrow + \times 2\ x\ 5$

Input

6

Generate Answer before Question

DATA SCIENCE RETREAT
SINCE 2014

Features(X)

+ × 2 x 5

Input

Target(Y)

+ Exp 2 × x + x 5

Output

Parallel Data Generation

DATA SCIENCE RETREAT®
SINCE 2014

Two ways to generate data:

- Multiprocess thread pool

```
from multiprocessing.pool import ThreadPool
_FINISH = False
start = time.time()
with ThreadPool(processes=14) as p:
    out = []
    r = p.map_async(generate_bwd,
[sequences_per_process]*process_runs, callback=out.append)
    r.wait()
    time.sleep(10)
    _FINISH = True    p.terminate()
```

- Ray

```
ray.init(num_cpus=cpu)
dataset = []
for _ in range(process_runs*cpu):
    try:
        out = ray_generate_bwd.remote(sequences_per_process)
        out = ray.get(out, timeout=sequences_per_process)
        dataset.extend(out)
```

Outline

DATA SCIENCE RETREAT®
SINCE 2014

1. Data Generation
2. Transformer Model
3. Results
4. Outlook

Transformer Model

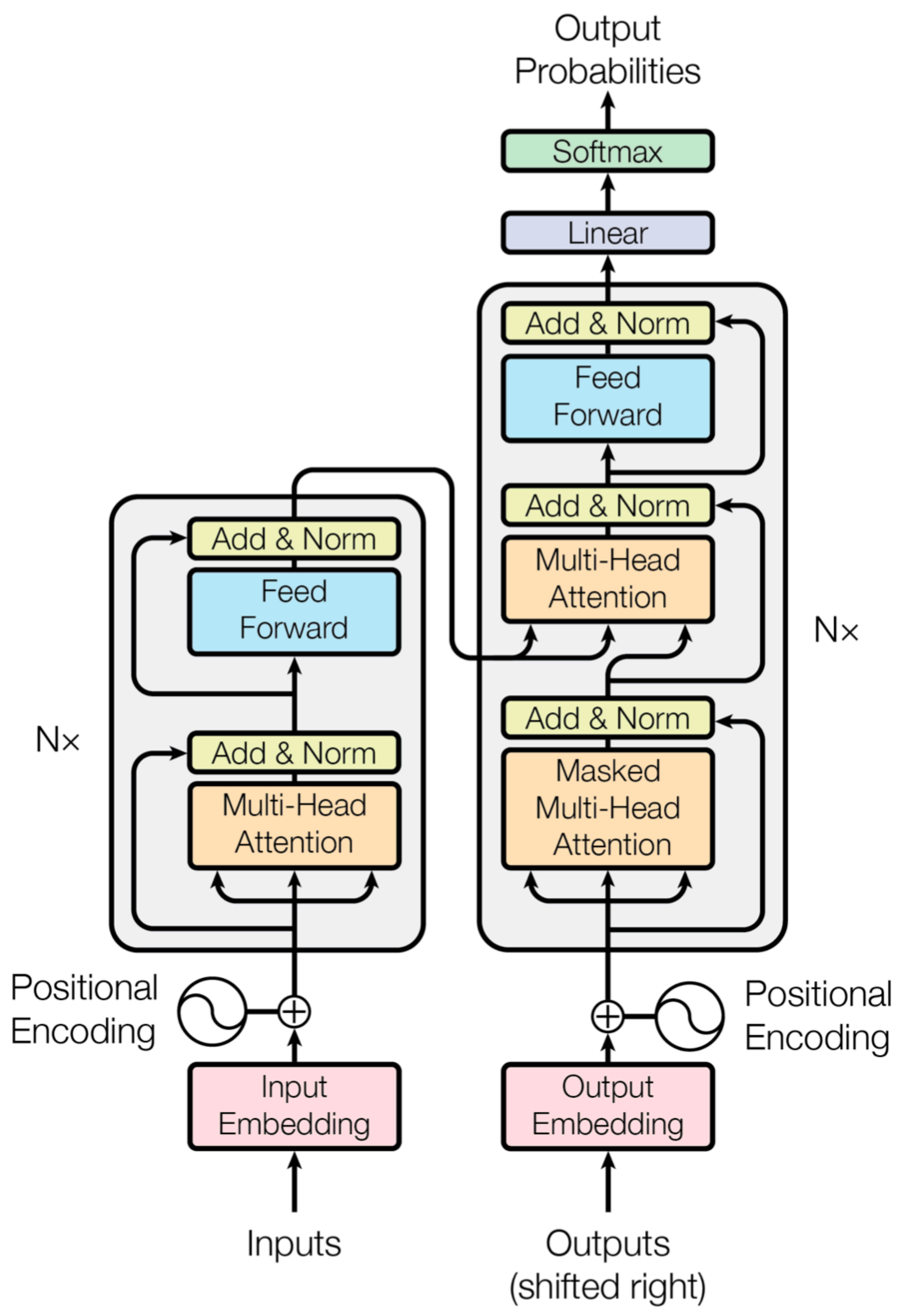


Figure from Vaswani, Ashish, et al. "Attention is all you need."(2017)

Transformer Model

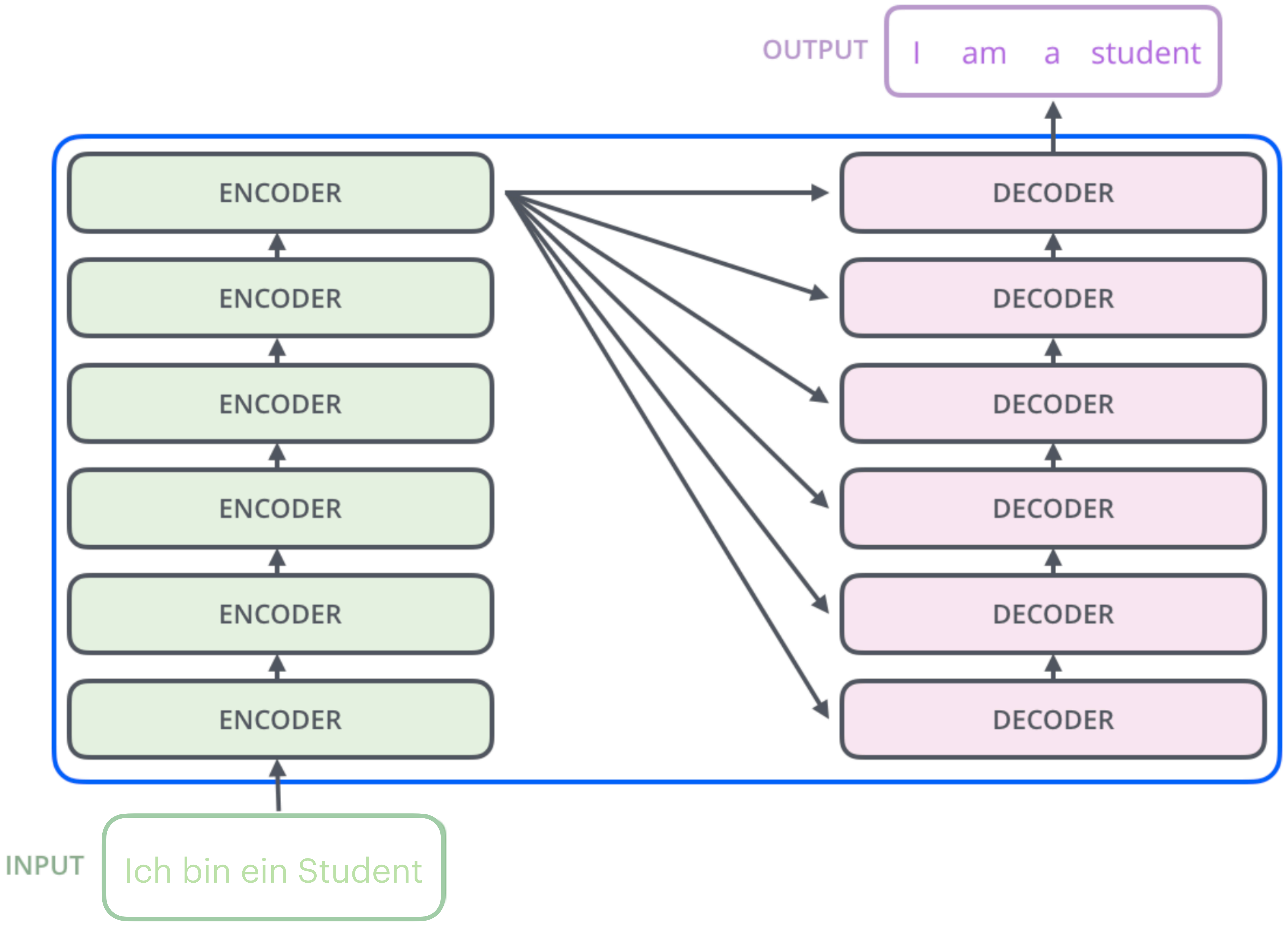


Figure from Jay Alammar http://jalammar.github.io/illustrated-transformer/

Transformer Model

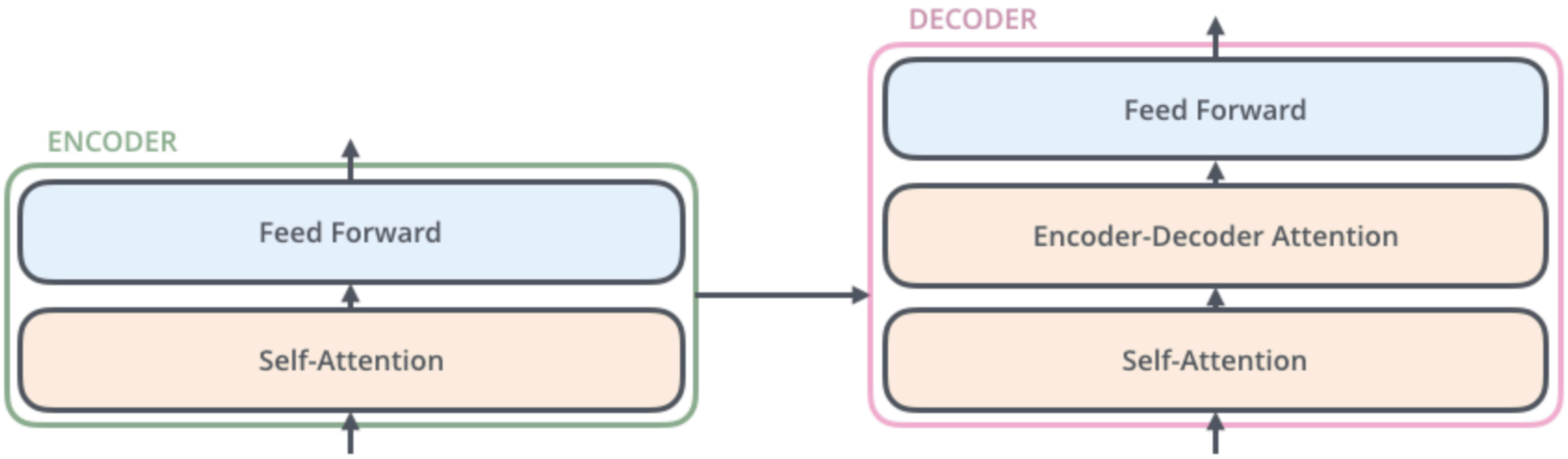


Figure from Jay Alammar http://jalammar.github.io/illustrated-transformer/

Transformer Model

Hyperparameters

batch_size = 32 d_model = 512

sequence_length = 512 num_layers = 6

num_heads = 8 dropout_rate = 0.1

optimizer = Adam learning_rate =CustomSchedule

loss, accuracy = CustomSchedule

Outline

DATA SCIENCE RETREAT®
SINCE 2014

1. Data Generation
2. Transformer Model
3. Results
4. Outlook

Paper Results

Training set size = 40M, Test set size = 5000

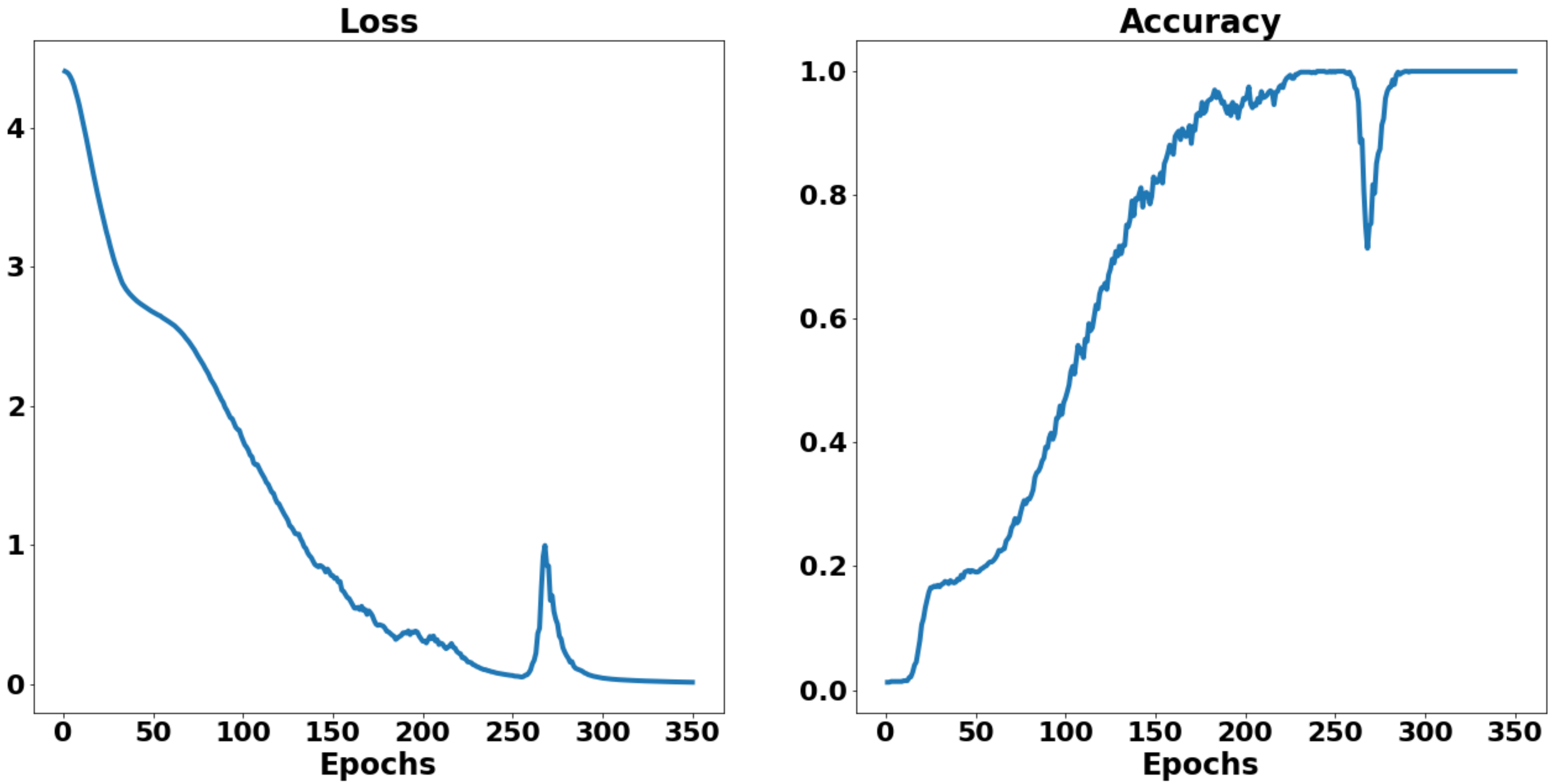
Compare with commercial software

	Integration (BWD)
Mathematica (30s)	84.0
Matlab	65.2
Maple	67.4
Beam size 1	98.4
Beam size 10	99.6
Beam size 50	99.6

Ref: Lample, Guillaume, and François Charton. "Deep learning for symbolic mathematics."

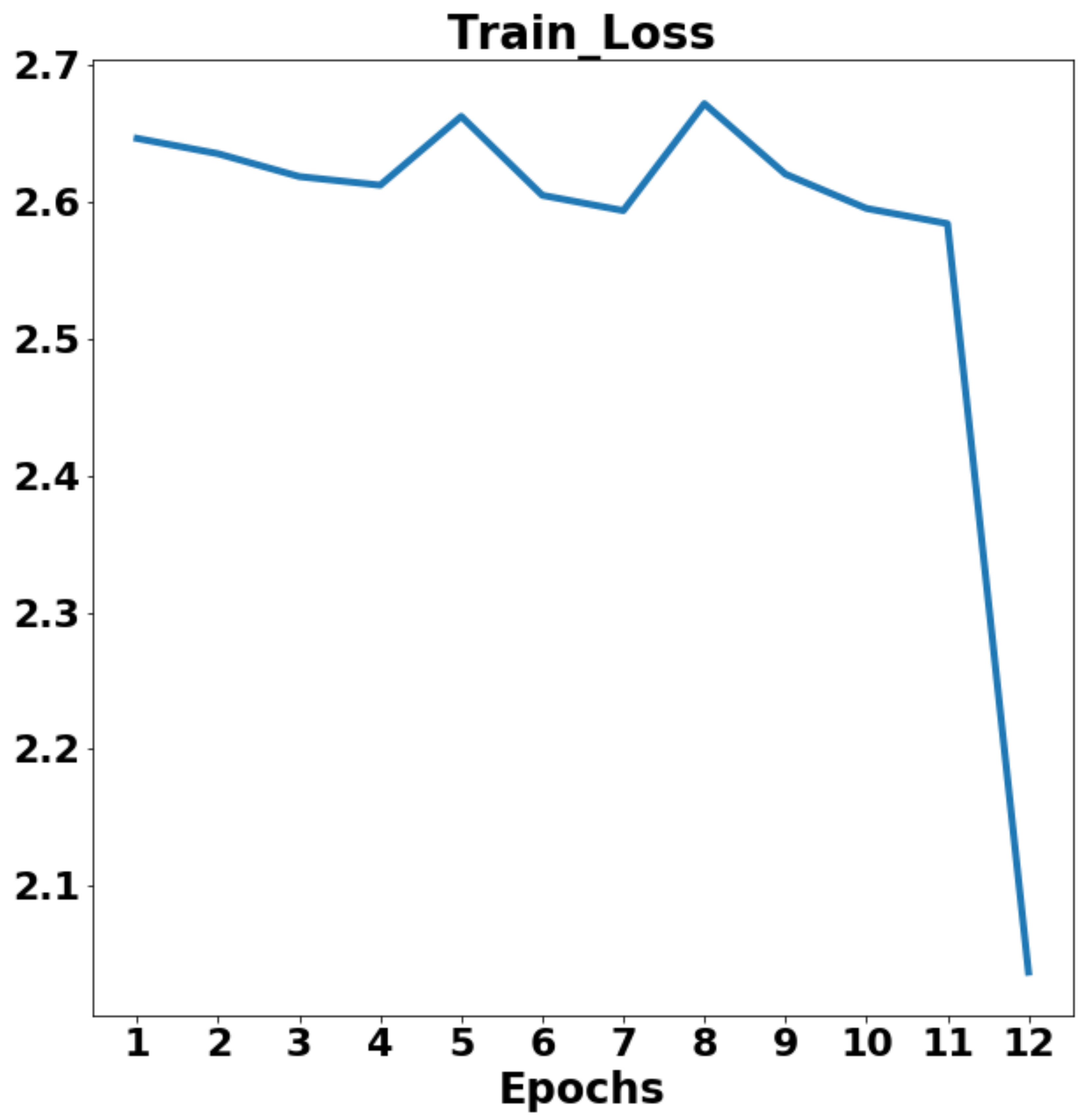
Results - Stage 1

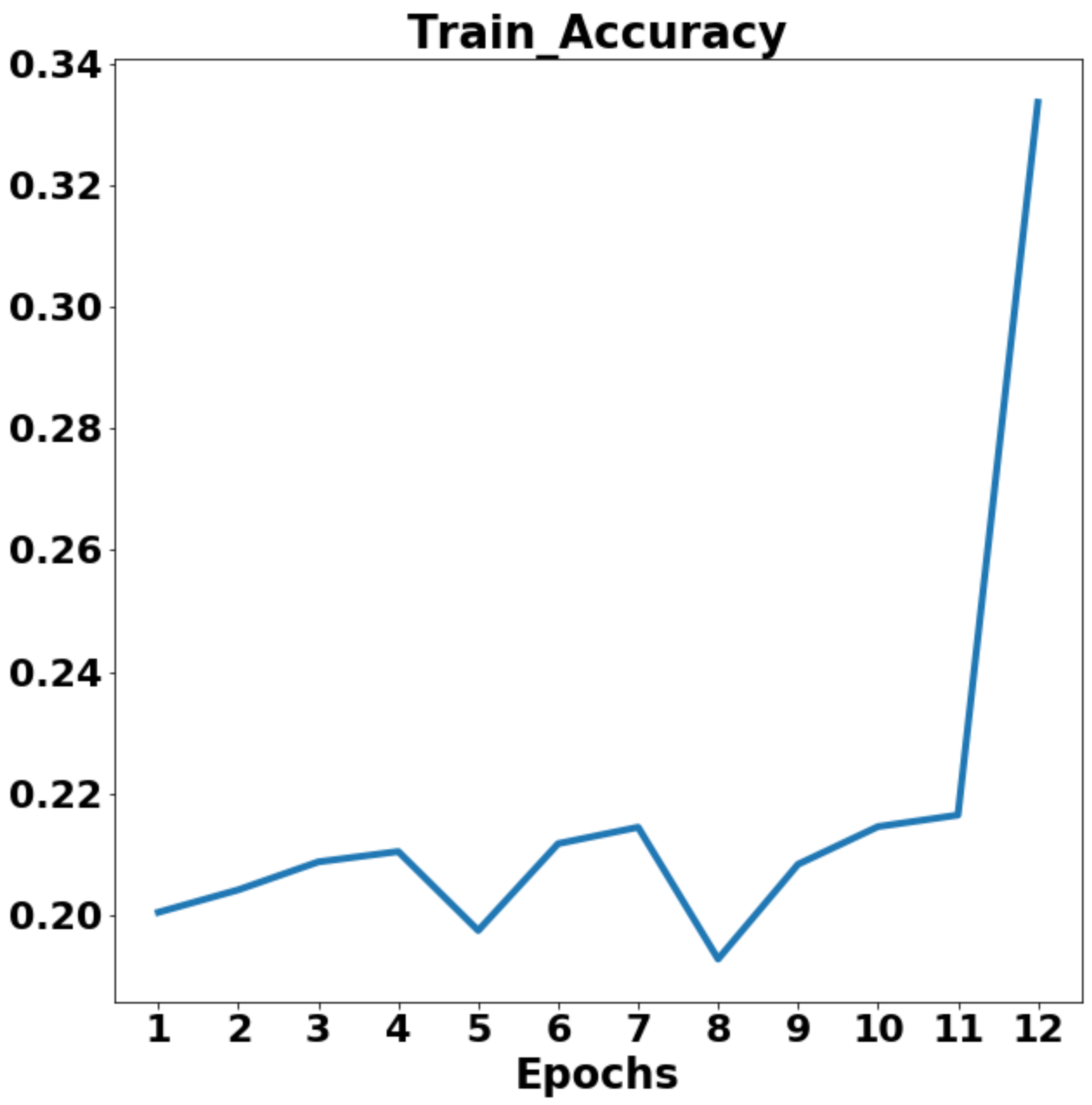
Training set size = 64



Results - Stage 2

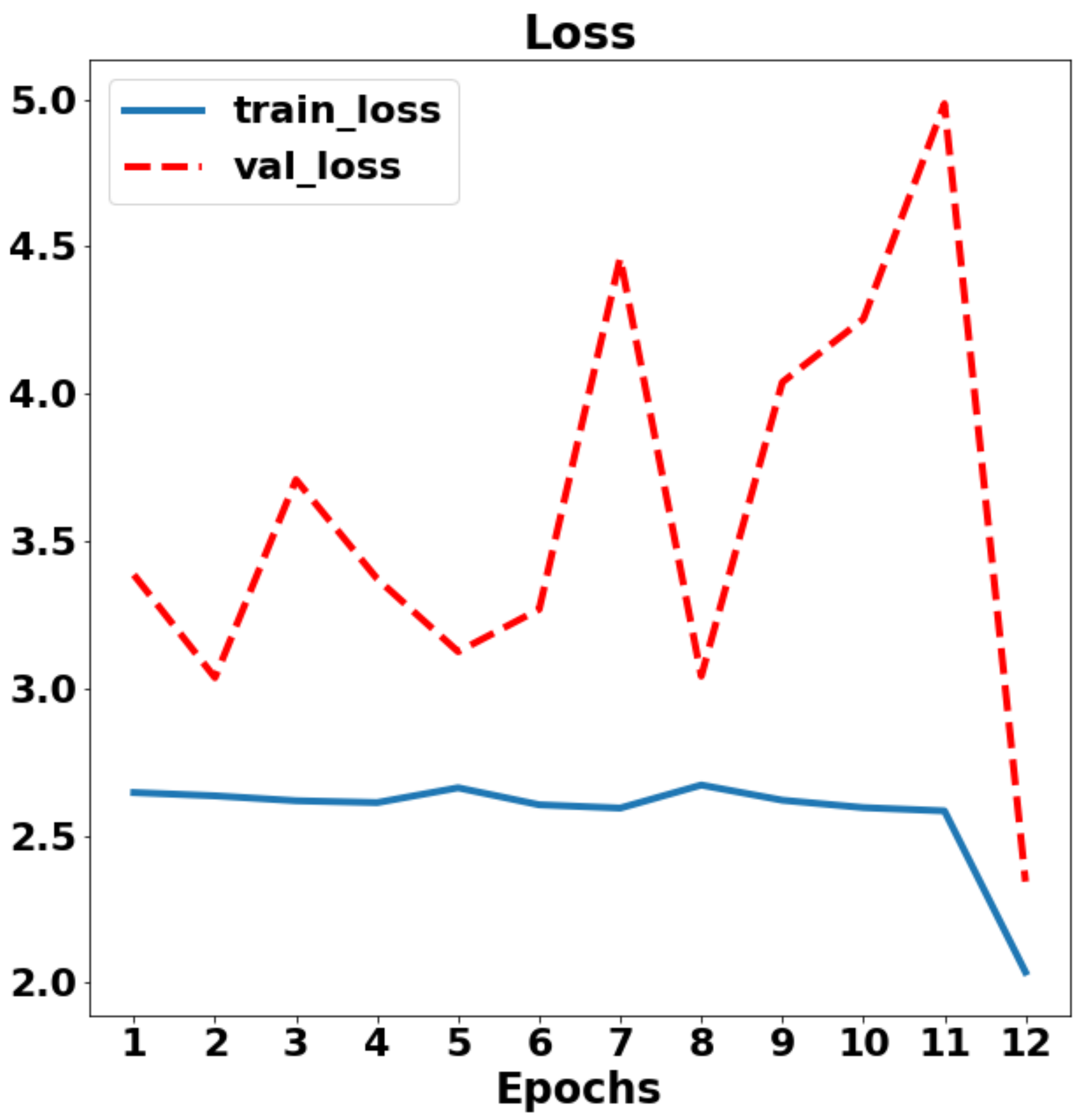
Training set size=100,000, Validation set size=9263

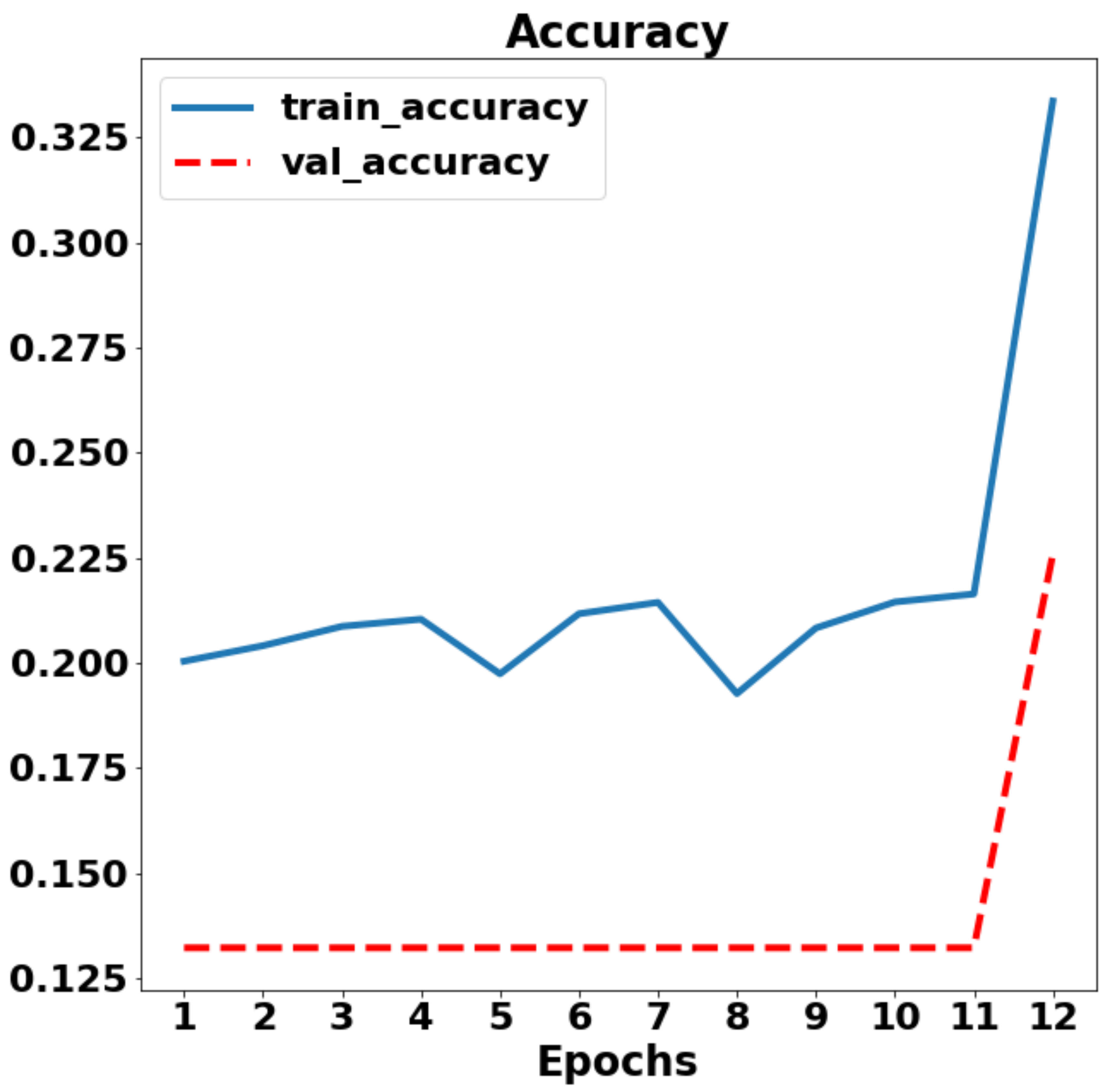




Results - Stage 2

Training set size=100,000, Validation set size=9263





Outline

1. Data Generation
2. Transformer Model
3. Results
4. Outlook

Outlook

- Generate more data and use more GPUs to train the model
- Compare performance of different dataset size
- Give a lower limit of dataset size
- Generalize to complex number

Reference

1. Lample, Guillaume, and François Charton. "Deep learning for symbolic mathematics." *arXiv preprint arXiv:1912.01412* (2019).
2. Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017.
3. Lample, Guillaume, and François Charton. "Deep learning for symbolic mathematics." Spotlight, Lunch&Learn, Author Speaking. https://aisc.ai.science/events/2020-02-18/
4. Jay Alammar "The Illustrated Transformer." http://jalammar.github.io/illustrated-transformer/

Acknowledgement

Thank you

Email: yangwj2011@gmail.com

Github: https://github.com/janeyoung2018/symbolic-math

LinkedIn: https://www.linkedin.com/in/wenjuan-yang-3664405a/