Bidirectional Encoder Representations from Transformers (BERT)

Shusen Wang

What is BERT?

- BERT [1] is for pre-training Transformer's [2] encoder.
- How?
- Predict masked word.
- Predict next sentence.

Reference

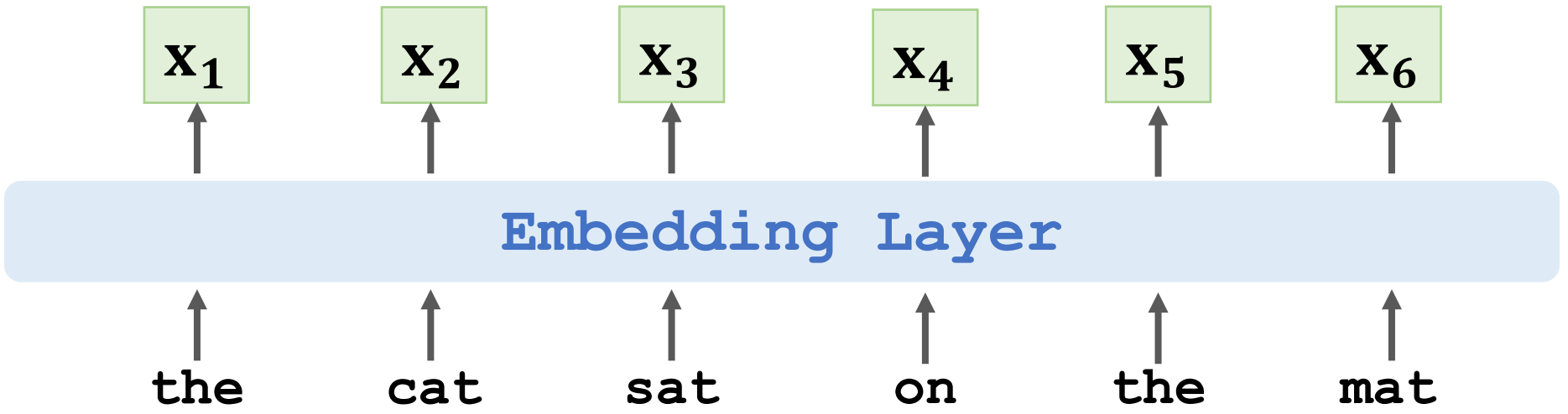
1. Devlin, Chang, Lee, and Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019.
2. Vaswani and others. Attention is all you need. In *NIPS*, 2017.

Task 1: Predict Masked Words

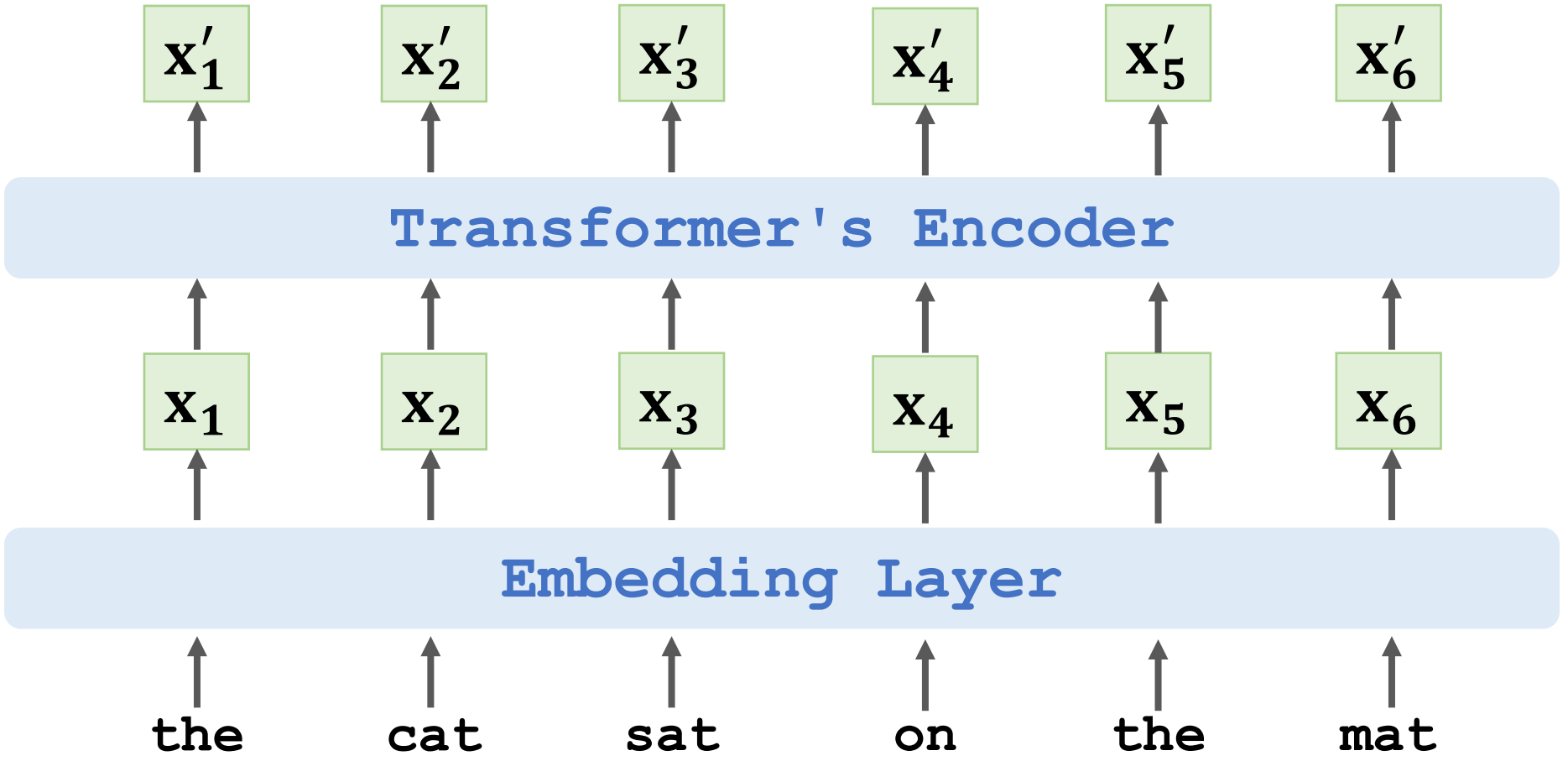
Revisit Transformer's Encoder

the cat sat on the mat

Revisit Transformer's Encoder



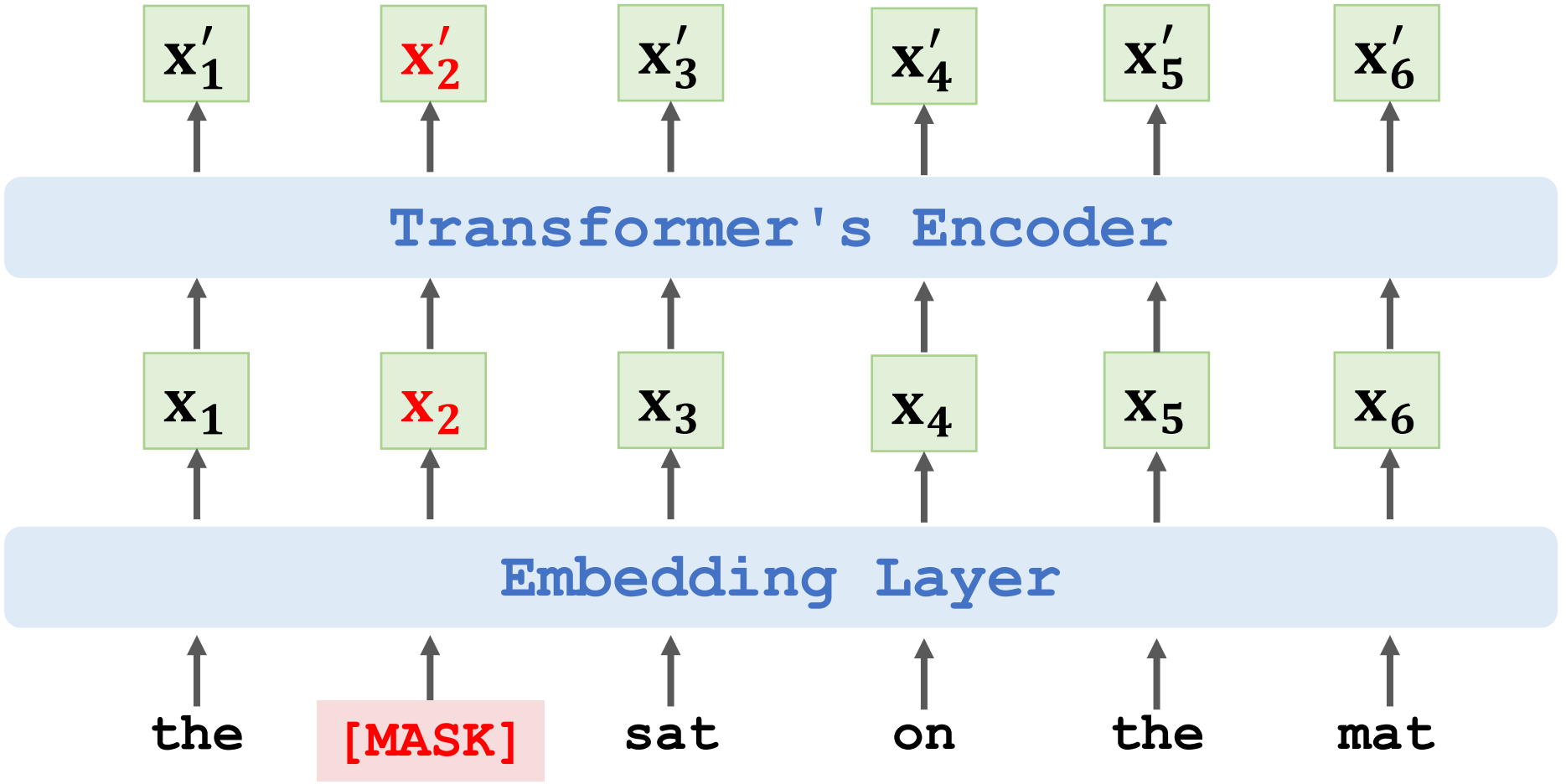
Revisit Transformer's Encoder

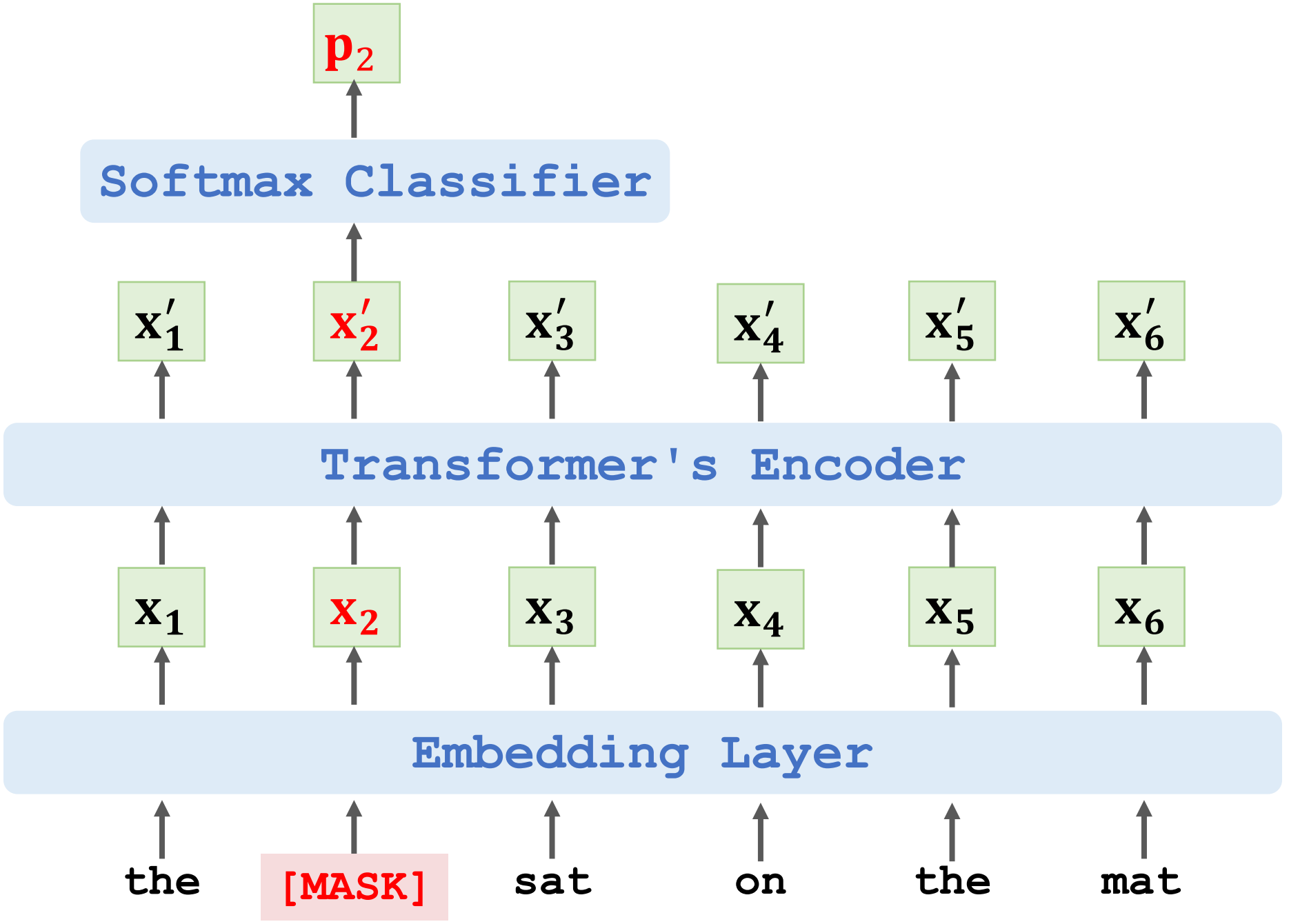


Randomly mask a word

- "The _____ sat on the mat"
- What is the masked word?

Randomly mask a word




p_2
Softmax Classifier
x'_1
x'_2
x'_3
x'_4
x'_5
x'_6
Transformer's Encoder
x_1
x_2
x_3
x_4
x_5
x_6
Embedding Layer
the
[MASK]
sat
on
the
mat

Predict the masked word

- $\mathbf{e}_2$: one-hot encode of the masked word "cat".
- $\mathbf{p}_2$: output probability distribution at the masked position.
- $\text{Loss} = \text{CrossEntropy}(\mathbf{e}_2, \mathbf{p}_2)$.
- Performing one gradient descent to update the model parameters.

Task 2: Predict the Next Sentence

Predict the next sentence

- Given the sentence:

  "calculus is a branch of math".

- Is this the next sentence?

  "it was developed by newton and leibniz"

- Is this the next sentence?

  "panda is native to south central china"

Input Representation

- **Input:**

[CLS] “calculus is a branch of math”

[SEP] “it was developed by newton and leibniz”

- **[CLS]** is a token for classification.
- **[SEP]** is for separating sentences.

Input Representation

- **Input:**

  [CLS] “calculus is a branch of math”
  [SEP] “it was developed by newton and leibniz”

- **Target:** true

Input Representation

- **Input:**

  [CLS] “calculus is a branch of math”

  [SEP] “it was developed by newton and leibniz”

- **Target:** true

Input Representation

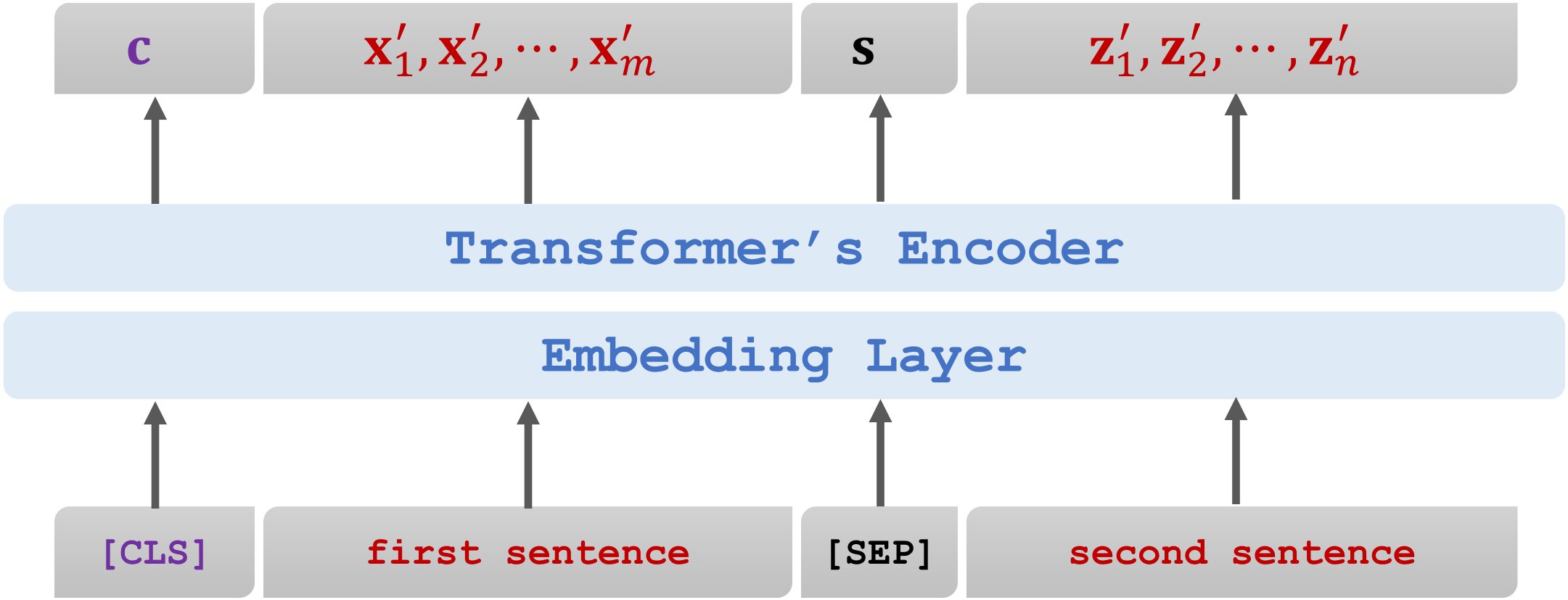
- **Input:**

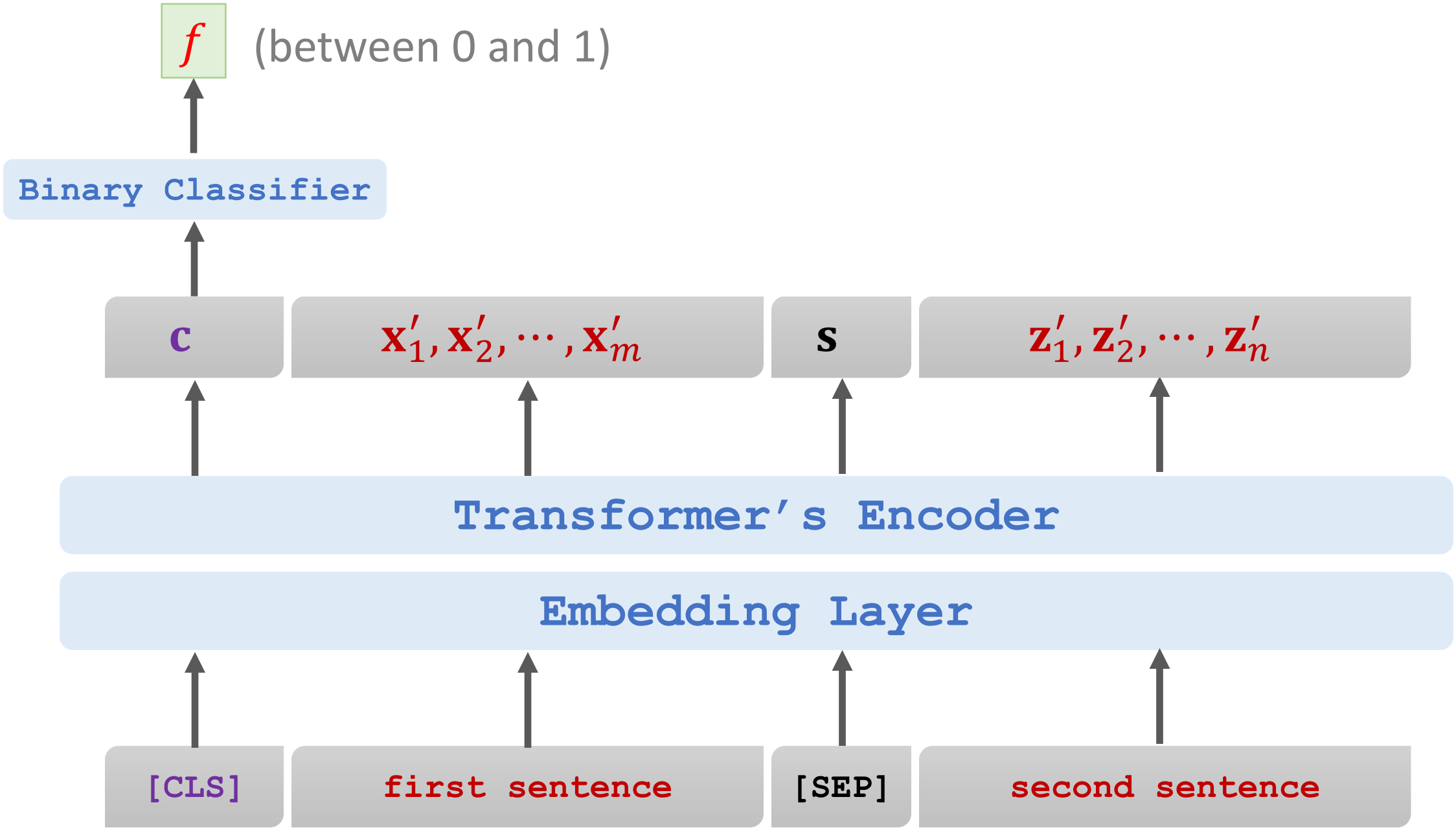
  [CLS] “calculus is a branch of math”

  [SEP] “panda is native to south central china”

- **Target:** false

Predict the next sentence




f
(between 0 and 1)
Binary Classifier
$\mathbf{c}$
$\mathbf{x}'_1, \mathbf{x}'_2, \cdots, \mathbf{x}'_m$
$\mathbf{s}$
$\mathbf{z}'_1, \mathbf{z}'_2, \cdots, \mathbf{z}'_n$
Transformer's Encoder
Embedding Layer
[CLS]
first sentence
[SEP]
second sentence

Combining the two methods

Input Representation

- **Input:**

"[CLS] calculus is a **[MASK]** of math [SEP] it **[MASK]** developed by newton and leibniz".

- **Targets:** true, "branch", "was".

Input Representation

- **Input:**

"[CLS] **[MASK]** is a branch of math [SEP] panda is native to **[MASK]** central china".

- **Targets:** false, "calculus", "south".

Training

- Loss 1 is for binary classification (i.e., predicting the next sentence.)
- Loss 2 and Loss 3 are for multi-class classification (i.e., predicting the masked words.)
- Objective function is the sum of the three loss functions.
- Update model parameters by performing one gradient descent.

Data

- Use large-scale data, e.g., English Wikipedia (2.5 billion words.)
- BERT does not need manually labeled data. (Nice! Manual labeling is expensive.)
- Randomly mask words (with some tricks.)
- 50% of the next sentences are real. (The other 50% are fake.)

Cost of Computation

- BERT Base
  - 110M parameters.
  - 16 TPUs, 4 days of training (without hyper-parameter tuning.)
- BERT Large
  - 235M parameters.
  - 64 TPUs, 4 days of training (without hyper-parameter tuning.)

Thank you!