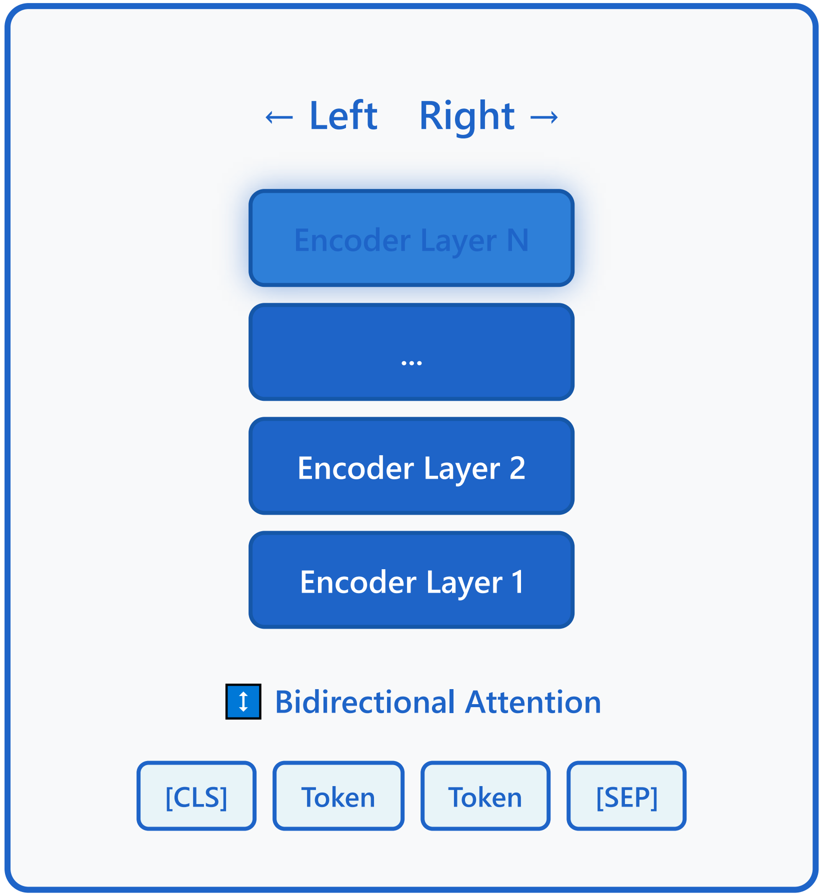
BERT: Bidirectional Encoder Representations from Transformers

Google, 2018 - Revolutionary NLP Model



Base: 110M params **Large:** 340M params

Dataset: BooksCorpus + Wikipedia (3.3B words)

🔄 Bidirectional Context

Deep **bidirectional** understanding from both directions

🗂️ Transformer Encoder

Stack of **Transformer encoders** with self-attention

📚 Pre-training Corpus

Trained on **BooksCorpus** and **Wikipedia**

⚡ Two Model Sizes

BERT-Base (110M) and BERT-Large (340M) parameters

🏆

State-of-the-Art Performance

Achieved SOTA on **11 NLP benchmarks**
Revolutionized NLP in 2018

🎓 BERT Pre-Training Process

1 Input Preparation

Tokenize the input text and add special tokens. [CLS] marks the start of the sentence, and [SEP] marks the end.

📝 Example Input:

```
"The cat sat on the mat"
→ [CLS] The cat sat on the mat [SEP]
```

2 Masked Language Modeling (MLM)

Randomly mask 15% of input tokens. The model learns to predict masked words using bidirectional context.

🎭 Masking Example:

```
Input: [CLS] The cat sat on the mat [SEP]
Masked: [CLS] The [MASK] sat on the mat [SEP]
Target: Predict "cat"
```

💡 Key Points

- 80% replaced with [MASK]
- 10% replaced with random word
- 10% keep original word

3 Next Sentence Prediction (NSP)

Learn to determine whether two sentences are consecutive. This improves the ability to understand relationships between sentences.

🔗 NSP Example:

```
Case 1 (IsNext):
[CLS] The cat sat on the mat [SEP] It was sleeping [SEP]
Label: IsNext ✓

Case 2 (NotNext):
[CLS] The cat sat on the mat [SEP] Paris is beautiful [SEP]
Label: NotNext ✗
```

4 Training Objective

Train the model by simultaneously optimizing both MLM and NSP losses.

MLM Loss

Cross-Entropy Loss aiming for accurate prediction of masked tokens

NSP Loss

Binary Classification Loss for correctly judging sentence pair continuity

🎯 Total Loss

Loss = MLM Loss + NSP Loss

🎯 BERT Fine-tuning Process

1 Load Pre-trained Model

Load the pre-trained BERT model. It has already learned general patterns of language.

✨ Pre-trained Knowledge

- Understanding grammatical structure
- Capturing semantic relationships
- Bidirectional context comprehension

2 Add Task-Specific Layer

Add an output layer specific to the task. Different structures are used depending on the task.

📊 Classification

[CLS] token output + Softmax Layer
(Sentiment analysis, topic classification, etc.)

🏷️ Token Classification

Each token output + Classification Layer
(Named entity recognition, POS tagging, etc.)

❓ Question Answering

Start/End Position Prediction Layers
(SQuAD, reading comprehension, etc.)

🔄 Sequence Pairing

[CLS] token + Binary Classifier
(Natural language inference, sentence similarity, etc.)

3 Fine-tune on Task Data

Fine-tune the entire model with task-specific training data. Typically, 2-4 epochs are sufficient.

📝 **Sentiment Analysis Example:**

```
Input: [CLS] This movie is amazing [SEP]
→ BERT Encoding →
→ Classification Layer →
Output: Positive (95% confidence)
```

⚙️ Training Settings

- Learning Rate: 2e-5 ~ 5e-5
- Epochs: 2-4
- Batch Size: 16 or 32

4 Inference & Prediction

Use the fine-tuned model to make predictions on new data.

🔮 **Prediction Pipeline:**

```
Step 1: Tokenize new input
Step 2: Pass through fine-tuned BERT
Step 3: Apply task-specific head
Step 4: Generate prediction with confidence score
```

🎯 Final Output

Generate results in the appropriate format for the task:
Classification labels, token tags, answer spans, etc.

BERT revolutionized NLP by introducing bidirectional pre-training and transfer learning to the field.