**به نام خدا**

**چیست؟ BERT**

**BERT (Bidirectional Encoder Representations from Transformers)** یک مدل پیشرفته پردازش زبان طبیعی (NLP) است که توسط گوگل در سال ۲۰۱۸ معرفی شد. در ادامه به طور کامل BERT را توضیح می‌دهم:

* یک مدل **زبان مبتنی بر Transformer** است
* از معماری **Encoder-only** استفاده می‌کند برخلاف GPT که Decoder-only است
* **دوطرفه (Bidirectional)** است یعنی هم زمینه قبل و هم بعد از هر کلمه را در نظر می‌گیرد

## ویژگی‌های کلیدی

### 1. **دوطرفه بودن (Bidirectional)**

* برخلاف مدل‌های قدیمی که فقط چپ به راست یا راست به چپ پردازش می‌کردند
* همزمان از **هر دو جهت** متن را تحلیل می‌کند

### 2.  **پیش‌آموزی (Pre-training)**

* روی حجم عظیمی از داده‌های متنی آموزش دیده
* دو وظیفه اصلی پیش‌آموزی:
  + - **MLM (Masked Language Model)**: پیش‌بینی کلمات پوشیده شده
    - **NSP (Next Sentence Prediction)**: تشخیص جمله بعدی

### **Fine-tuning** 3.**آسان**

* برای کارهای مختلف قابل تنظیم است
* نیاز به داده کمی برای Fine-tuning دارد

## 🎯 کاربردهای اصلی

### 1. **درک متن (Text Understanding)**

* تحلیل احساسات (Sentiment Analysis)
* طبقه‌بندی متن (Text Classification)

### 2. **پرسش و پاسخ (Question Answering)**

* استخراج پاسخ از متن
* سیستم‌های چت‌بات

### 3. **پردازش زبان**

* خلاصه‌سازی متن
* ترجمه ماشینی
* تولید متن

## 🏗️ معماری BERT

### انواع معماری:

* **BERT-Base**: 12 لایه، 110 میلیون پارامتر
* **BERT-Large**: 24 لایه، 340 میلیون پارامتر

### ساختار:

Input → Tokenization → Embedding → Transformer Encoders → Output

## مقایسه با دیگر مدل‌ها

|  |  |  |  |
| --- | --- | --- | --- |
| مدل | نوع | جهت‌گیری | کاربرد اصلی |
| BERT | Encoder | دوطرفه | درک متن |
| GPT | Decoder | یکطرفه | تولید متن |
| T5 | Encoder-Decoder | دوطرفه | ترجمه و خلاصه‌سازی |

## مثال عملی :

*# استفاده از BERT برای تحلیل احساسات*

from transformers import BertTokenizer, BertForSequenceClassification

import torch

*# بارگذاری مدل از پیش آموزش دیده*

tokenizer = BertTokenizer.from\_pretrained('bert-base-uncased')

model = BertForSequenceClassification.from\_pretrained('bert-base-uncased')

*# پردازش متن*

text = "I love this movie! It's amazing."

inputs = tokenizer(text, return\_tensors="pt", padding=True, truncation=True)

*# پیش‌بینی*

outputs = model(\*\*inputs)

predictions = torch.softmax(outputs.logits, dim=-1)

print(predictions) *# احتمال مثبت/منفی بودن*

## 🌟 مزایای BERT

### ✅ نقاط قوت:

1. **دقت بالا** در کارهای NLP
2. **چندمنظوره** بودن
3. **جامعه بزرگ** توسعه‌دهندگان
4. **پیش‌آموزی قوی**

### ❌ محدودیت‌ها:

1. **حجم بزرگ** مدل
2. **مصرف محاسباتی** بالا
3. **نیاز به سخت‌افزار** قدرتمند

## 🚀 نسخه‌های بهبودیافته

* **RoBERTa**: نسخه بهینه‌شده BERT
* **DistilBERT**: نسخه سبک‌تر
* **ALBERT**: نسخه با پارامترهای کمتر
* **BERT Multilingual**: پشتیبانی از چند زبان

## 🎯 جمع‌بندی

BERT انقلابی در حوزه NLP ایجاد کرد زیرا:

* **درک عمیق‌تر** از متن ارائه می‌دهد
* **قابل تنظیم** برای کارهای مختلف است
* **پایه‌ای** برای بسیاری از مدل‌های جدید شد

امروزه BERT و مدل‌های مبتنی بر آن در بسیاری از محصولات استفاده می شوند مانند:

* جستجوی گوگل
* دستیارهای صوتی
* سیستم‌های ترجمه
* چت‌بات‌های هوشمند