Hugging Face Trainers

Lesson Downloads

Show TranscriptSummarize Video

Hugging Face trainers offer a simplified approach to training generative AI models, making it easier to set up and run complex machine learning tasks. This tool wraps up the hard parts, like handling data and carrying out the training process, allowing us to focus on the big picture and achieve better outcomes with our AI endeavors.

Technical Terms Explained:

Truncating: This refers to shortening longer pieces of text to fit a certain size limit.

Padding: Adding extra data to shorter texts to reach a uniform length for processing.

Batches: Batches are small, evenly divided parts of data that the AI looks at and learns from each step of the way.

Batch Size: The number of data samples that the machine considers in one go during training.

Epochs: A complete pass through the entire training dataset. The more epochs, the more the computer goes over the material to learn.

Dataset Splits: Dividing the dataset into parts for different uses, such as training the model and testing how well it works.

Quiz Question

Why do we use padding in machine learning mod [illegible]

- To protect data from unauthorized access.
- To increase the volume of data we have.
- To ensure that all input data has the same length.

Submit

Code Example

```
from transformers import (DistilBertForSe[illegible]ication,
    DistilBertTokenizer,
    TrainingArguments,
    Trainer
)
from datasets import load_dataset

model = DistilBertForSequenceClassification.from_pretrained(
    "distilbert-base-uncased", num_labels=2
)
tokenizer = DistilBertTokenizer.from_pretrained("distilbert-base-uncased")

def tokenize_function(examples):
    return tokenizer(examples["text"], padding="max_length", truncation=True)


dataset = load_dataset("imdb")
tokenized_datasets = dataset.map(tokenize_function, batched=True)

training_args = TrainingArguments(
    per_device_train_batch_size=64,
    output_dir="./results",
    learning_rate=2e-5,
    num_train_epochs=3,
)
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_datasets["train"],
    eval_dataset=tokenized_datasets["test"],
)
trainer.train()
```

Resources

Hugging Face Trainers documentation index
Hugging Face DistilBertForSequenceClassification documentation
Hugging Face DistilBertTokenizer documentation
distilbert-base-uncased Model documentation on Hugging Face
Hugging Face transformers.TrainingArguments documentation
Hugging Face transformers.Trainer documentation