

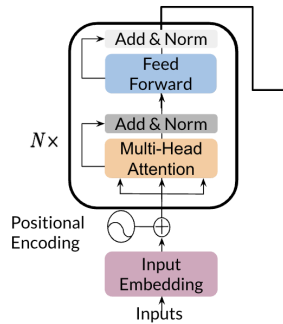
Question Answering

- ✓ **Video:** Week 3 Overview
6 min
 - ✓ **Reading:** Week 3 Overview
10 min
 - ✓ **Video:** Transfer Learning in NLP
7 min
 - ✓ **Reading:** Transfer Learning in NLP
10 min
 - ✓ **Video:** ELMo, GPT, BERT, T5
7 min
 - ✓ **Reading:** ELMo, GPT, BERT, T5
10 min
 - ✓ **Video:** Bidirectional Encoder Representations from Transformers (BERT)
4 min
 - ✓ **Reading:** Bidirectional Encoder Representations from Transformers (BERT)
10 min
 - ✓ **Video:** BERT Objective
2 min
 - ✓ **Reading:** BERT Objective
10 min
 - ✓ **Video:** Fine tuning BERT
2 min
 - ✓ **Reading:** Fine tuning BERT
10 min
 - ✓ **Video:** Transformer: T5
3 min
 - ✓ **Reading:** Transformer T5
10 min
 - ✓ **Video:** Multi-Task Training Strategy
5 min
 - ✓ **Reading:** Multi-Task Training Strategy
10 min
 - ✓ **Video:** GLUE Benchmark
2 min
 - ✓ **Reading:** GLUE Benchmark
10 min
 - ✓ **Video:** Question Answering
2 min
 - 📖 **Reading:** Question Answering
10 min
 - 📄 **Lab:** SentencePiece and BPE
2h
 - 📖 **Reading:** Content Resource
10 min
- ### Assignment
- 📄 **Programming Assignment:** Question Answering
3h

Question Answering

You will be implementing an encoder this week. Last week you implemented the decoder. So here it is:

Transformer encoder



Feedforward:

```
[  
  LayerNorm,  
  dense,  
  activation,  
  dropout_middle,  
  dense,  
  dropout_final  
]
```

Encoder block:

```
[  
  Residual(  
    LayerNorm,  
    attention,  
    dropout_,  
  ),  
  Residual(  
    feed_forward,  
  )  
]
```

You can see there is a feedforward and the encoder-block above. It makes use of two residual connections, layer normalization, and dropout.

The steps you will follow to implement it are:

- Load a pre-trained model
- Process data to get the required inputs and outputs: "question: Q context: C" as input and "A" as target
- Fine tune your model on the new task and input
- Predict using your own model

[Mark as completed](#)