Transformer from Scratch

Implementation of the Transformer architecture.

Files

- `transformer.py` — full implementation and example

Architecture

The model follows the standard encoder-decoder Transformer:

- **LayerNorm** — normalizes across the feature dimension
- **PositionalEncoding** — sinusoidal position signals added to embeddings
- **MultiHeadAttention** — scaled dot-product attention across multiple parallel heads
- **FeedForward** — two-layer MLP applied after attention
- **EncoderBlock / Encoder** — self-attention + FFN, stacked N times
- **DecoderBlock / Decoder** — masked self-attention + cross-attention + FFN, stacked N times
- **Transformer** — full encoder-decoder model with embedding lookup and output projection

Running

```
python transformer.py
```

This runs a small example with a 12-word vocabulary. The model is randomly initialized so predicted words will be nonsense — no training is included.

Example Output

```
Source:           the cat sat on the mat
Target:           a dog ran fast
Output shape:     (1, 6, 12)
Predicted words:  ran ran ran ran ran ran
```