Introduction. This project trains the IMDB review dataset (50k) on two different architectures - the Gated Recurrent Unit (GRU) and a basic transformer architecture.

Architecture Details. Implemented a word-level tokenization of vocabulary size 20000 and minimum frequence 2. Sequence length was fixed to 256, and padded/truncated accordingly. Train/test split was 25k/25k each.

- GRU implementation is located in `gru.py`. The architecture is a simple recurrent neural network with a fully connected layer at the output. The embedding size is 200 and hidden size is 256.

  Padding is implented using the PyTorch function `pack_padded_sequence`, which requires a length array of size `(B,)`. Such array was calculated at the dataset level and passed through as an argument.

- The basic transformer implementation is located in `transformer.py`. A fixed sinusoidal positional embedding was used. The embedding size was 256. Six attention layers were used, each with four heads.

- A tweaked version of the basic transformer was also implemented. This version swaps the ReLU nonlinearity to a GELU nonlinearity, and also implements a learnable positional embedding.

Training Details. Due to time and compute contraints, a hyperparameter sweep was not performed. Batch size and number of epochs are fixed to 64 and 10.

- For the GRU, a learning rate of 10^{-3} was used, with a cosine annealing scheduler with no warmup. The AdamW optimizer was used with $\beta = (0.9, 0.99)$, and no weight decay was used. Dropout was set to 0.3, as RNNs reportedly benefit from high regularization.

- For the transformer, a learning rate of 3×10^{-4} was used, with a cosine annealing scheduler with warmup ratio 0.06. The AdamW optimizer was used with $\beta = (0.9, 0.98)$, and weight decay 0.01 was used. Dropout was set to 0.2.

Results. The following table lists the best training accuracy.

	GRU	Transformer	Tweaked Transformer
Best Test Accuracy	89.97%	89.09%	88.43%

Table 1: IMDB Training Results

Discussions. The transformer performed worse, and the tweaked version even worse. The lack of a hyperparameter sweep is the main issue. Moreover, this could be attributed to the small size of the IMDB dataset. I suspect that adding more flexibility through the transformer architecture / learned positional encodings / GELU nonlinearity decreased the inductive bias needed for such a small dataset.