HTX xData Technical Test (Data Scientist)

Task 2: Experiments on Training an ASR Model

This experiment involved fine-tuning an Automatic Speech Recognition (ASR) model wav2vec2-large-960h (from https://huggingface.co/facebook/wav2vec2-large-960h) to minimize the Word Error Rate (WER) when evaluated on a given validation dataset.

Due to computational and time constraints, the fine-tuning process was conducted on a dataset of 10,000 randomly selected data points. These data points constituted the primary dataset, which was subsequently divided into training and validation subsets.

The following experiments were performed:

Learning Rate Sweep

The model was trained for 100 steps using multiple learning rates (1e-4, 5e-5, 1e-5, 5e-6). Since the model performed evaluations every 20 steps, this provided five observations of the Word Error Rate (WER) on the test-validation dataset, along with the corresponding training and validation loss.

Here are the values of my initial observation:

lr = 1e-5

Step	Training Loss	Validation Loss	Wer
20	44.234000	26.779074	0.117262
40	24.918200	30.115297	0.139807
60	41.530600	28.069191	0.118282
80	39.042500	24.716057	0.111529
100	32.406900	24.728260	0.125211

lr = 5e-5

Step	Training Loss	Validation Loss	Wer
20	35.047700	26.725161	0.104565
40	17.051900	27.443743	0.111142
60	23.794600	24.192263	0.110685
80	23.517800	27.231407	0.104636
100	22.193600	26.689270	0.103264

lr = 1e-6

Step	Training Loss	Validation Loss	Wer
20	27.166600	25.991772	0.101541
40	10.557600	26.646088	0.101611
60	14.171100	27.343058	0.102138
80	13.479900	27.446886	0.099817
100	16.077500	27.883266	0.100872

lr = 5e-6

Step	Training Loss	Validation Loss	Wer
20	22.967200	27.966652	0.100134
40	6.746700	29.271669	0.099887
60	9.957400	30.138979	0.100661
80	10.554100	29.907749	0.098832
100	14.620500	29.766417	0.099465

Hence, my choice of learning rate at the final model was 5e-6. I also used a weight decay of 0.01 and warmup_steps of 10 (10% of total steps).

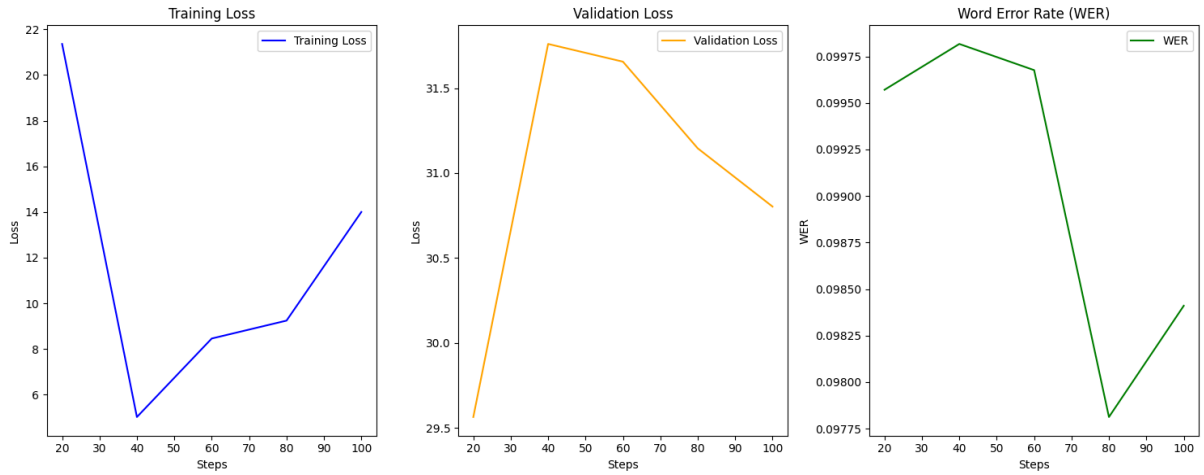
The finetuning for this Wav2Vec2 model for automatic speech recognition started with resampling all audio files to 16 kHz, in view the model's requirements. Audio was then normalised to achieve consistent volume levels across the dataset. This minimies variability caused by recording conditions. Additionally, augmentations such as time masking and frequency masking were used enhance the model's robustness to noise and improve generalization.

The Wav2Vec2 processor for tokenisation, which combines a tokenizer and feature extractor designed specifically for handling raw audio inputs. The tokenizer created labels based on the provided text transcriptions, allowing the model to adapt to the vocabulary present in the dataset. Feature extraction involved segmenting the audio into manageable chunks using the processor’s inbuilt functions, which extract Mel-frequency cepstral coefficients (MFCCs) or similar representations for input into the neural network,.

Hyperparameter optimizations were done through observation and the learning rate was set to 5e-6, a value determined through experimentation to balance convergence speed and stability as seen in the final table. Gradient accumulation steps were configured at 16, effectively increasing the batch size to 32 to GPU memory limitations. Weight decay was set to 0.01 to regularize the model and prevent overfitting, while warmup steps were established as 10% of the total training steps to ensure gradual learning rate increases during the initial phase of training. Mixed precision training (FP16) was employed to accelerate computations

and reduce memory usage, and checkpoints were saved every 20 steps to monitor progress and enable early stopping if necessary. Evaluation during training was performed at regular intervals using Word Error Rate (WER) as the primary metric, providing insights into the model's transcription accuracy.

Plot



The training process shows positive and negative trends, as illustrated in the above figure. The sharp initial decline in training loss suggests that the model quickly learns meaningful patterns; however, the subsequent increase indicates potential overfitting. Despite this, the validation loss exhibits a favorable trend, with a peak followed by a steady decline, aligning with a sharp reduction in Word Error Rate (WER) after step 70. This improvement in WER validates the model's ability to refine its transcription accuracy over time. The lowest WER achieved compared to other hyperparameters justifies the choice of the hyperparameters. However, admittedly, addressing overfitting through regularization and learning rate optimization could further enhance the model's performance. Additionally, with more computing resources, the model could be trained on the entire dataset (instead of just 10000 samples), adding to the model's ability to correctly transcribe.