Speech autoencoder using deep fully-connected networks

Andrew Seagraves

1/3/19

Implementation Summary

- Vanilla autoencoder built using Keras/TensorFlow
- Custom pipeline for pre-processing and on-the-fly batch generation
- Specialized utility functions for training on LibriSpeech corpuses
- Mini-batches generated by traversing the raw audio signal for each speaker and breaking it into chunks of a specified size which become the feature vectors.
- Overlap allowed on input chunks as a means for incorporating temporal structure into the autoencoder. The corresponding non-overlapping output chunks are used as labels.

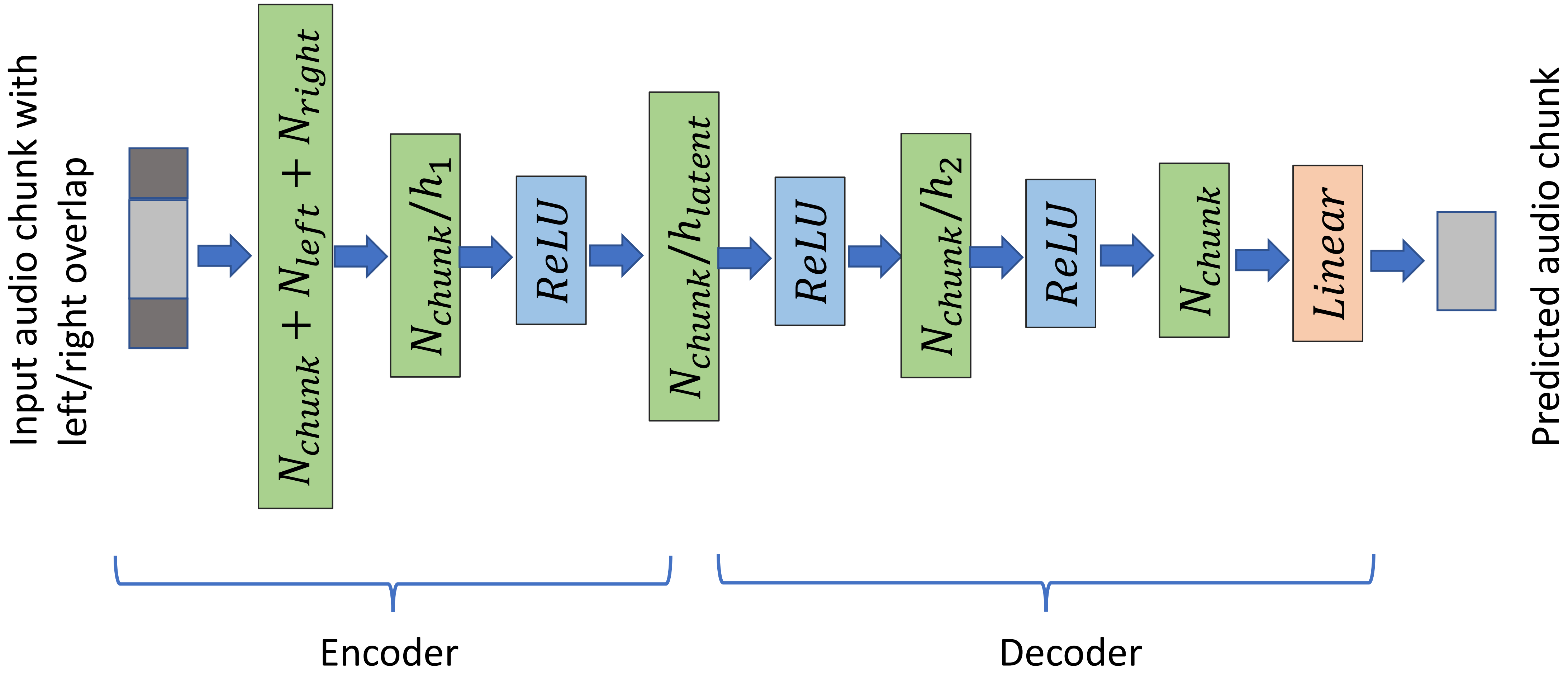
Pre-processing Workflow

build_speech_dict.py

- Generates a “speech_dict” data structure from the corpus
  - Reflects the internal structure of the LibriSpeech corpus. Useful for traversing the speakers, chapters, and utterances.

- Concatenates each speaker’s utterances into a master 1D numpy array for the speaker. Writes array to disk.
  - The sequence of the utterances is preserved in the array.
  - Speaker array used for fast on-the-fly batch generation

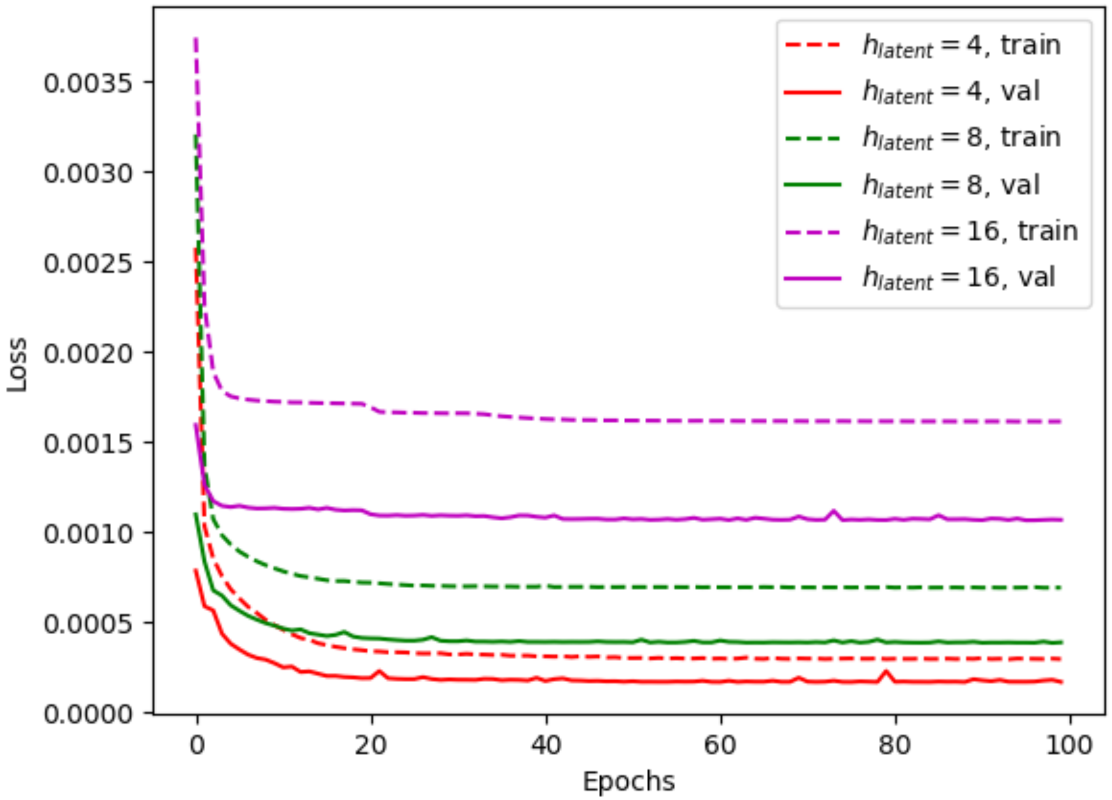
Example Network Architecture

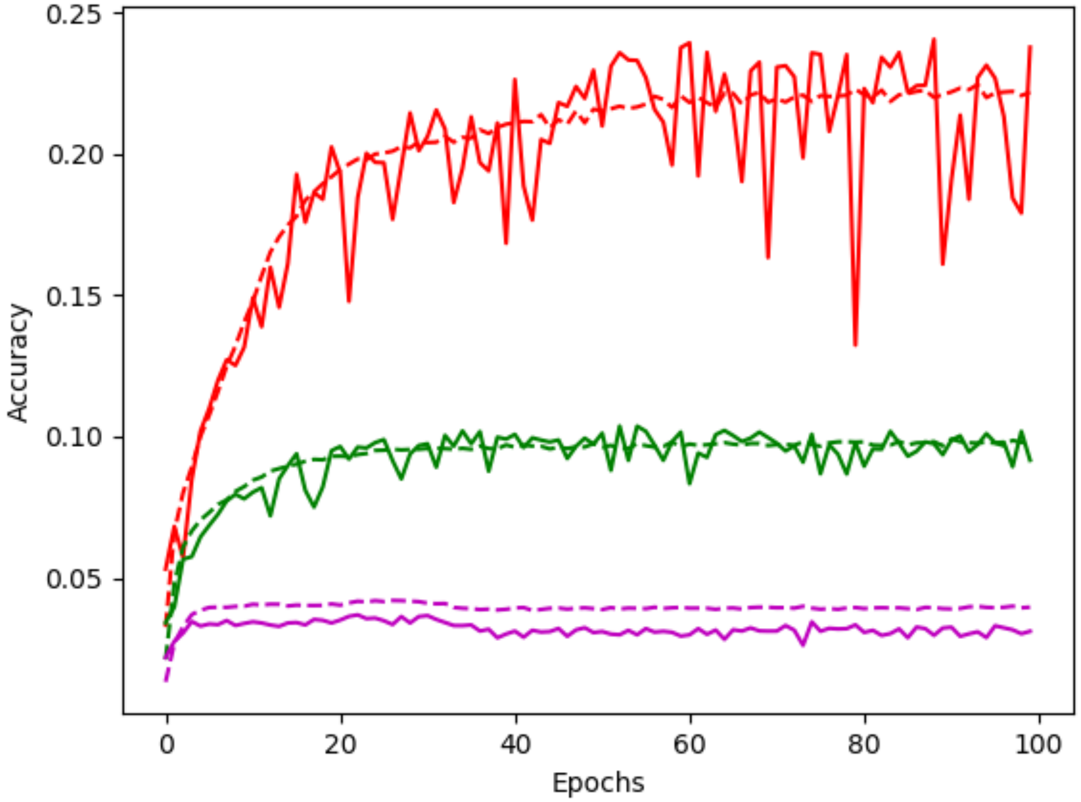


Training Experiments

- dev-clean dataset (40 speakers, 5.1 hrs total, 294M samples @16kHz)
- Tried chunk size, batch size from Chorowski, et al. (2019)
  - $N_{chunk} = 5120\ (\Delta t_{chunk} = 320ms,\ 16\mathrm{kHz}),$ batch size = 64
  - This chunk size is too large for the vanilla autoencoder and training fails with a non-decreasing loss function
- Reducing chunk size and increasing batch size proved successful
  - $N_{chunk} = 800\ (\Delta t_{chunk} = 50ms,\ 16\mathrm{kHz}),$ batch size = 128
- Training parameters:
  - 100 epochs, learning rate = 1e-4, Adam optimizer, MSE loss

3-layer networks

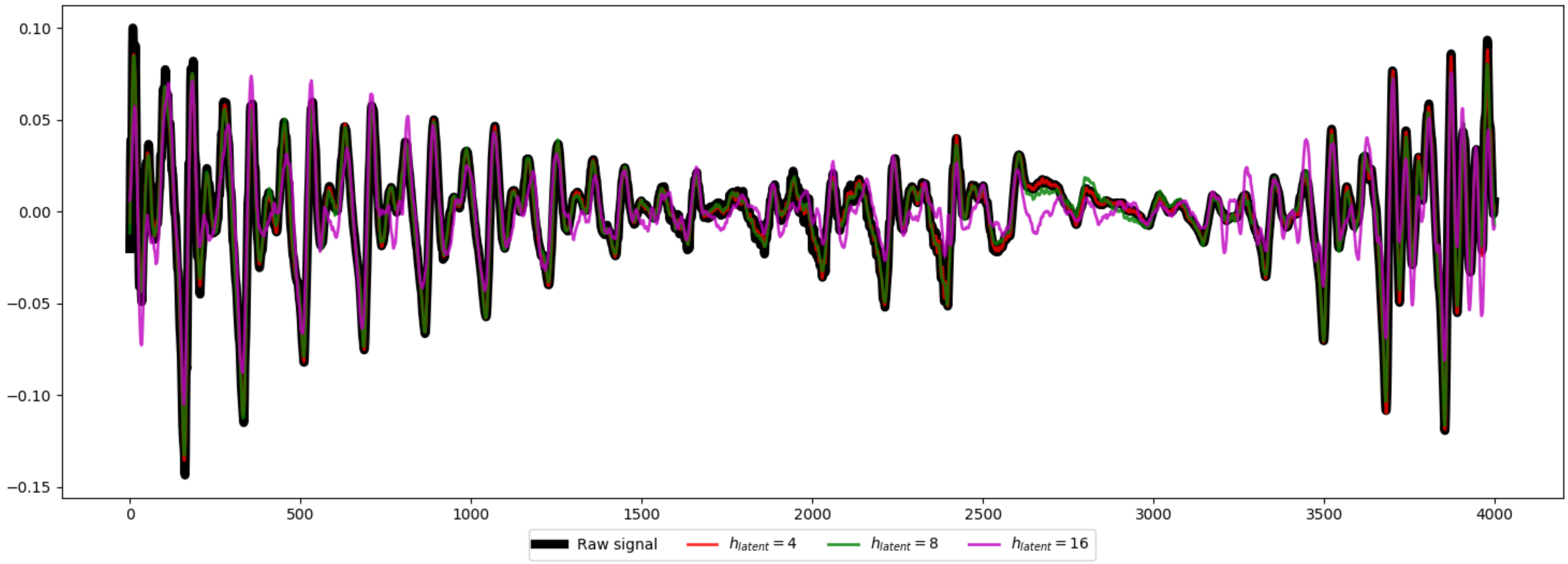




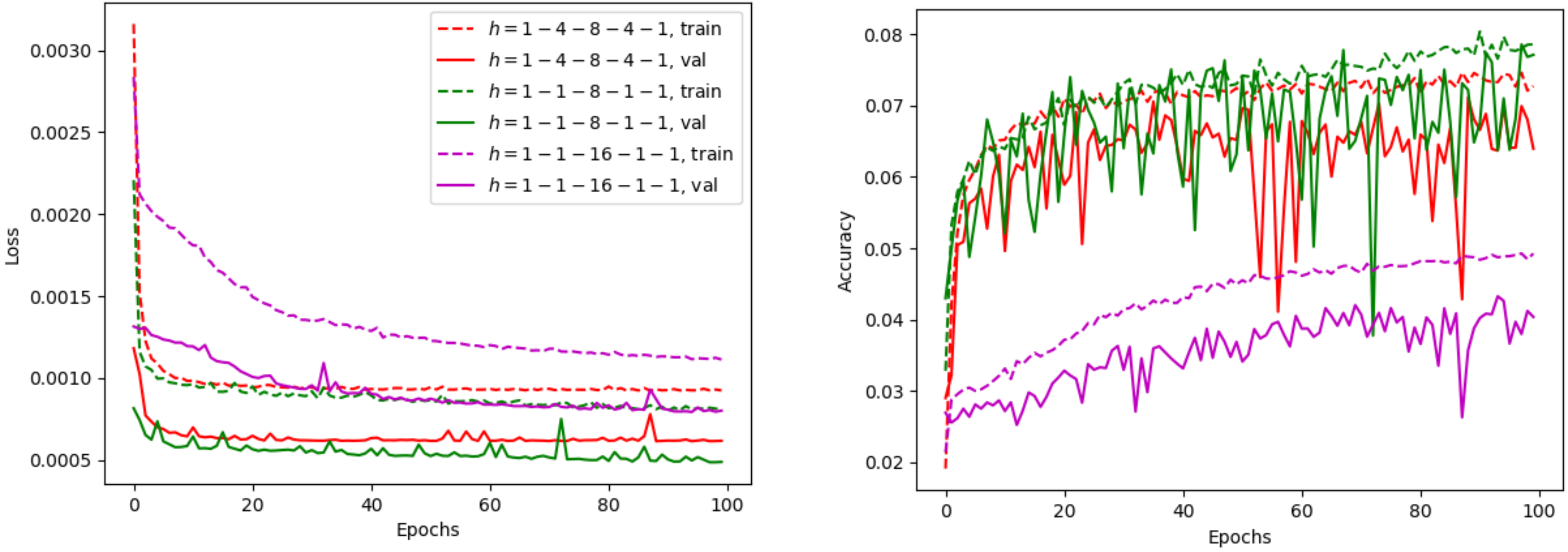
$$N_{chunk} \rightarrow \frac{N_{chunk}}{h_{latent}} \rightarrow N_{chunk}$$

3-layer networks

Speaker 2803
250ms audio sample (5 chunks)



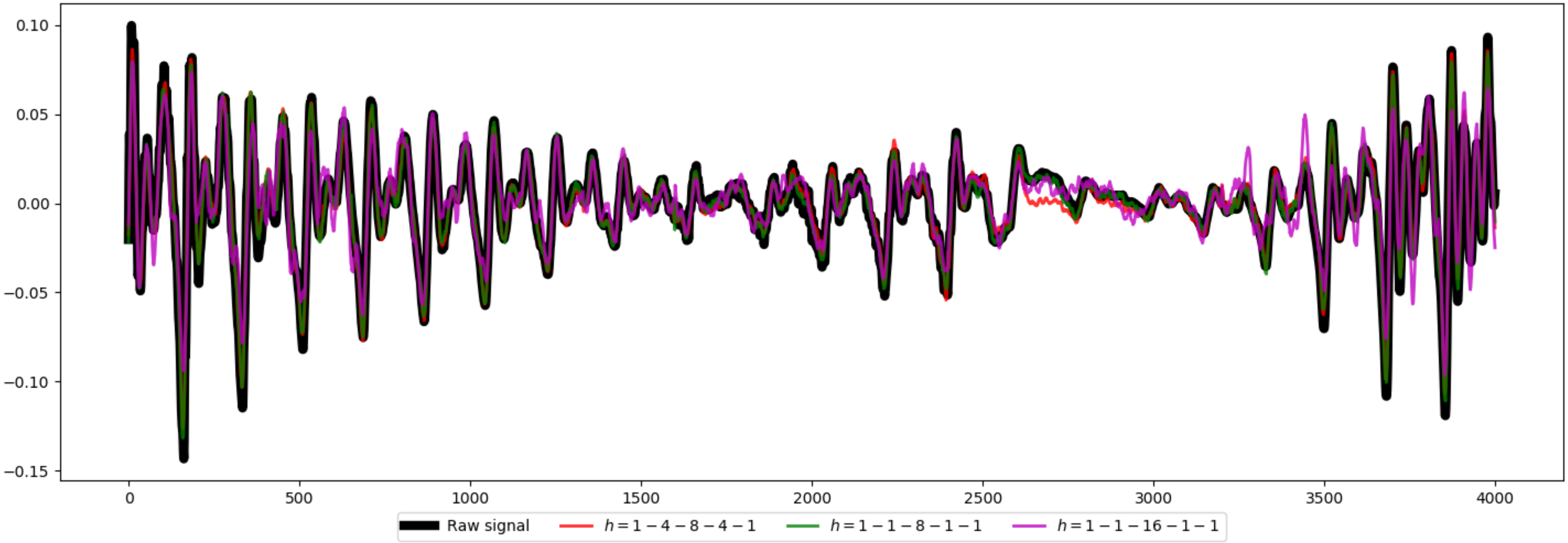
5-layer networks



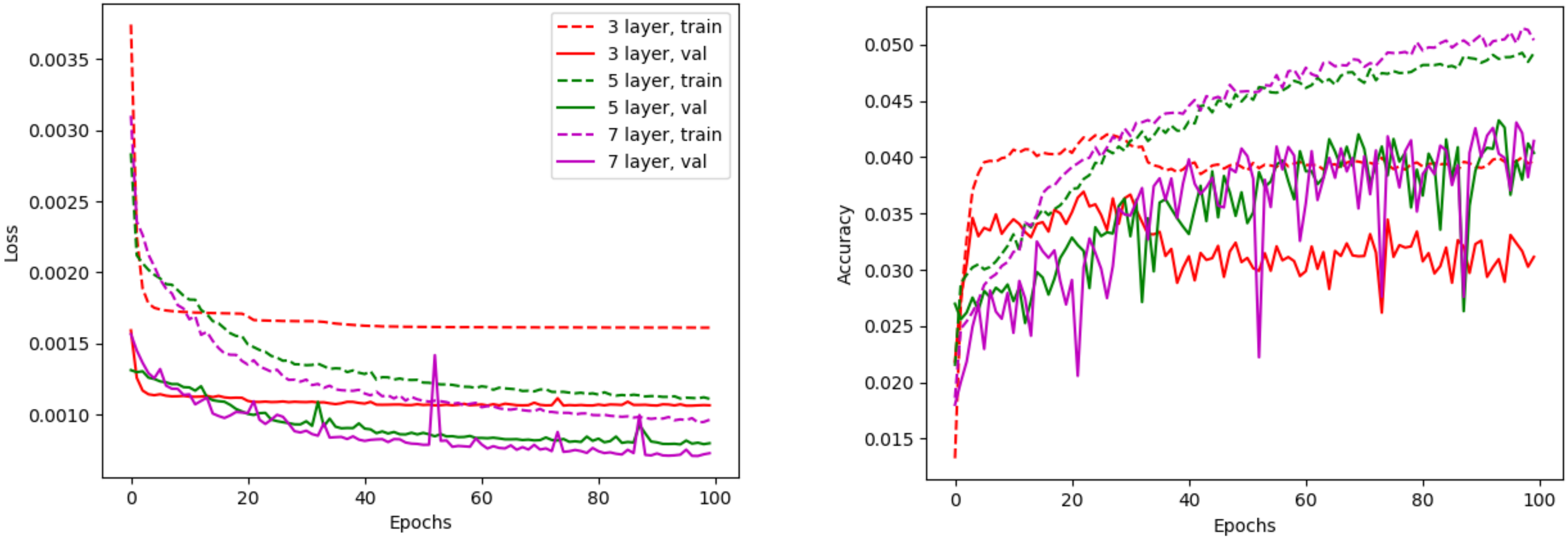
$$N_{chunk} \rightarrow \frac{N_{chunk}}{h_1} \rightarrow \frac{N_{chunk}}{h_{latent}} \rightarrow \frac{N_{chunk}}{h_1} \rightarrow N_{chunk}$$

5-layer networks

Speaker 2803
250ms audio sample (5 chunks)



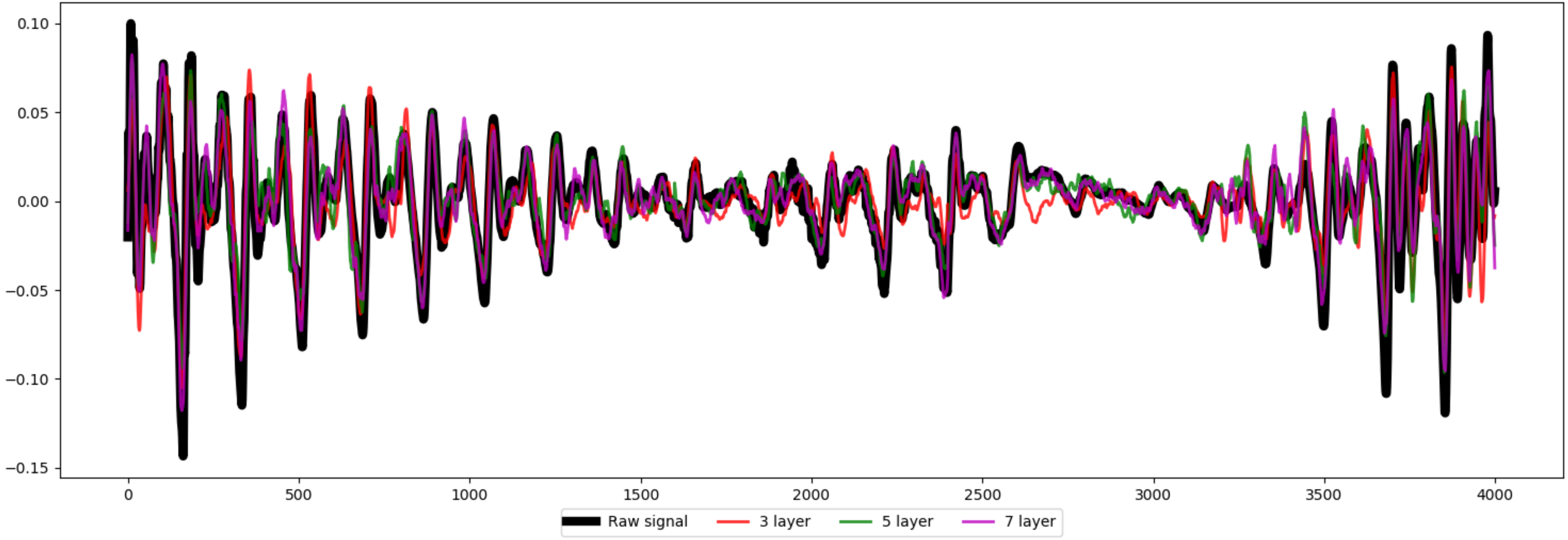
Effect of network depth



Symmetric architectures, h_{latent} =16, size reduction only in the latent layer

Effect of network depth

Speaker 2803
250ms audio sample (5 chunks)



Symmetric architectures, h_{latent} =16, size reduction only in the latent layer