Progress Indication for Deep Learning Model Training

Gang Luo

Department of Biomedical Informatics and Medical Education

University of Washington

luogang@uw.edu

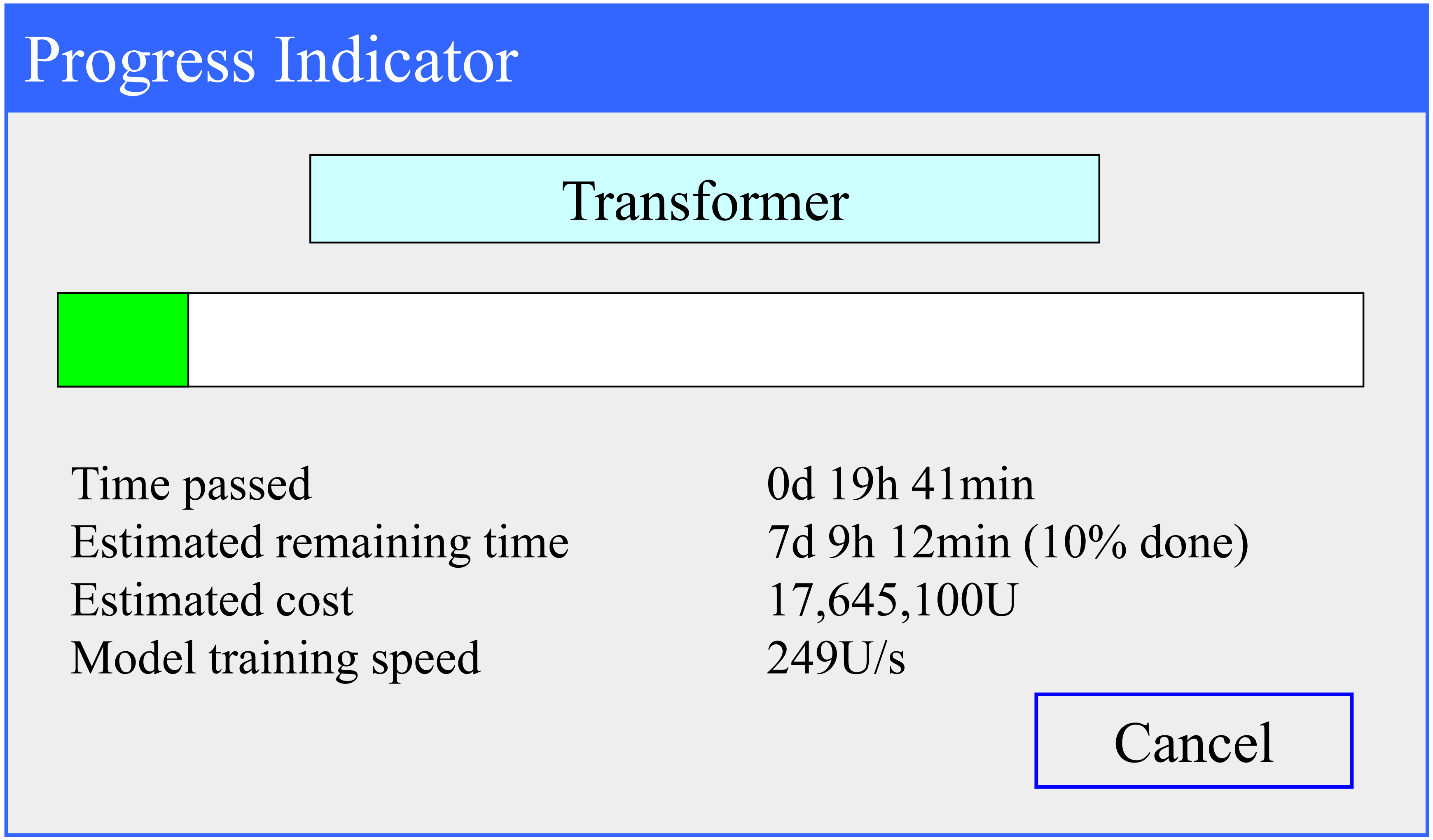
A Challenge for Using Deep Learning

- Training a deep learning model on a large data set can take several days or even months, while the model builder has no idea when model training will finish and could easily become frustrated
  - Using 50 graphics processing units, a Google team spent two months training a deep neural network on 300 million images
  - The model builder could mistakenly think that the machine learning software has stopped working

Need for Automatic Administration

- Yasser M. Ibrahim, the head of distributed machine learning at Amazon
  - Using a large computer cluster, his team took several months to train a deep neural network to support speech recognition in Alexa
  - Every so often, his team retrains this neural network and would like to finish the re-training in a given amount of time
  - Need a method to find an appropriate cluster configuration for each round of re-training
    - The amount of training data, the neural network's hyper-parameter values, and the server capacity keep changing over time

Our Solution: Progress Indicator for Deep Learning Model Training



Progress Indicator

Transformer

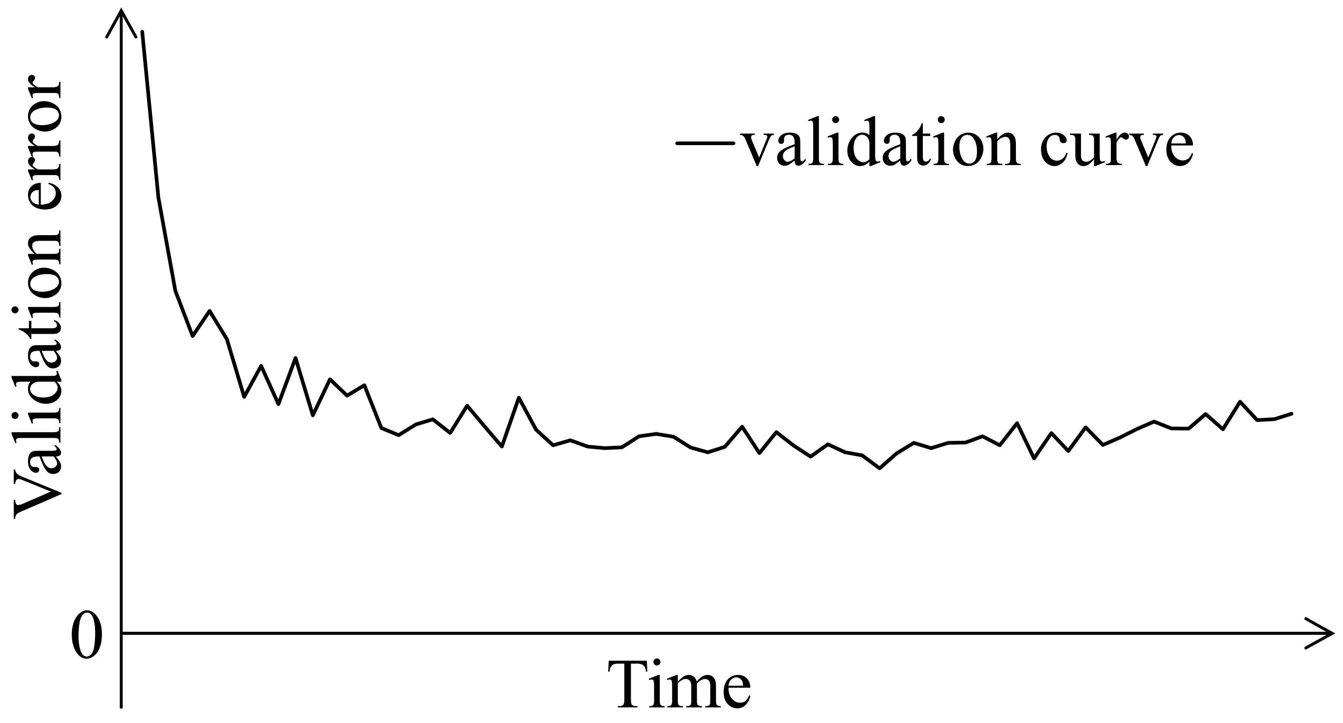
Time passed	0d 19h 41min
Estimated remaining time	7d 9h 12min (10% done)
Estimated cost	17,645,100U
Model training speed	249U/s

Main Idea

- A deep learning model is trained in batches
  - In each batch, a fixed # of training instances are used to compute the updates to the model's parameters
- Each batch's running cost is relatively stable and can be quickly measured
- Key to estimating the progress of model training: project the # of batches needed for model training
  - Particularly when early stopping is allowed

Main Idea – Cont.

- Project the # of batches needed for model training using the validation curve
  - The model's error rates on the validation set, i.e., the validation errors, obtained over time

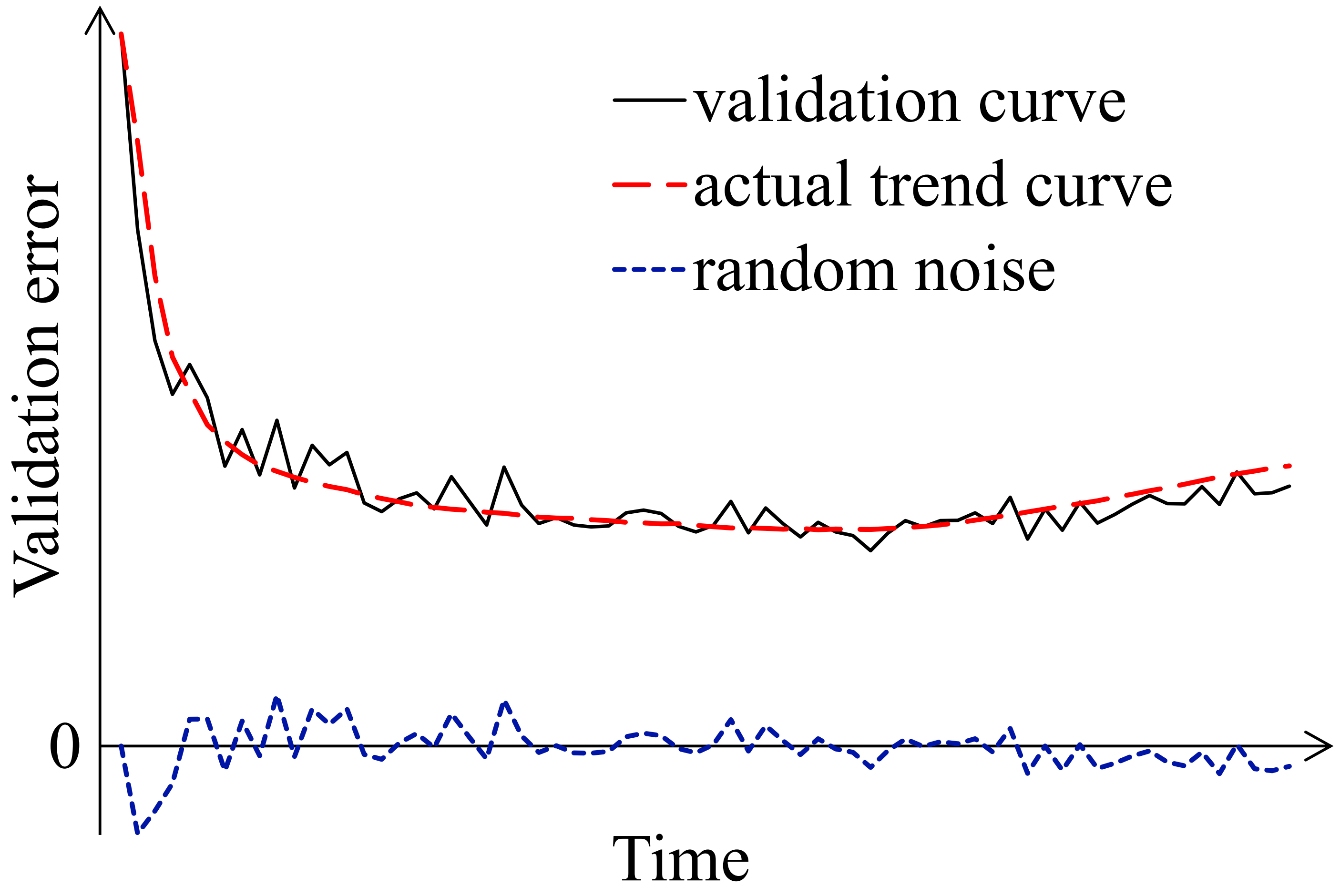


A Technical Difficulty

- The validation error tends to reduce over time before early stopping occurs and also oscillates over time
- Can seldom obtain a good estimate of the # of batches needed for model training if we
  - Use a monotonically decreasing function to model the validation curve without accommodating the oscillations
  - Directly apply the early stopping criterion to the projected curve

Solution to the Difficulty

- Regard the validation curve = a smooth trend curve + some 0-mean random noise



Solution to the Difficulty – Cont.

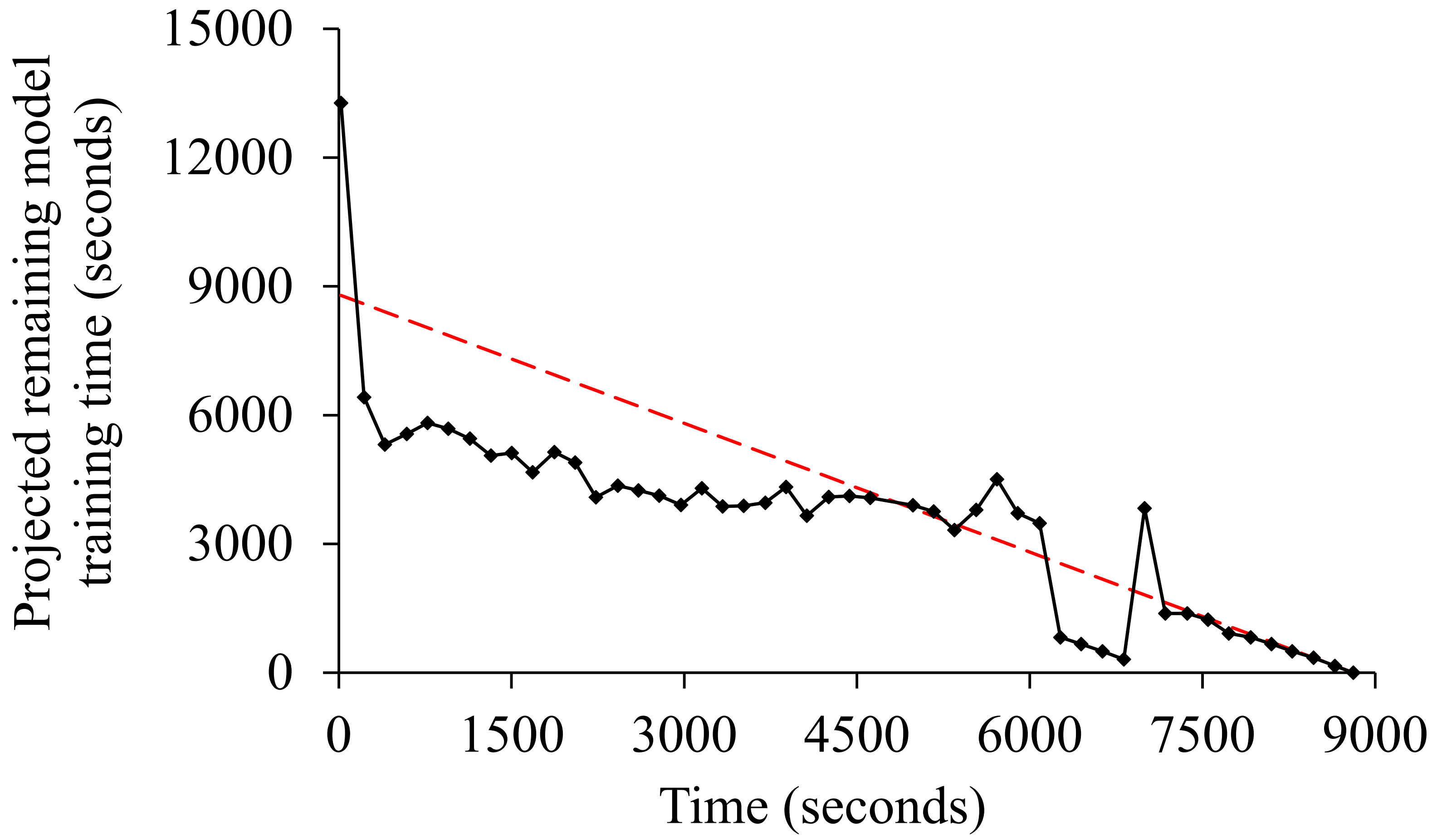
- Use a regression function to estimate the trend curve
- Use historical data to gauge the random noise's variance
  - If the learning rate changes over time, also model the change's impact on the random noise's variance
- Use Monte Carlo simulation to project the # of batches needed for model training

Monte Carlo Simulation

- Generate several synthetic validation curves
  - By adding simulated random noise to the projected trend curve
- On each of them, apply the early stopping criterion to obtain a simulated # of batches needed for model training
- The estimated mode of these simulated numbers forms the basis for the projected # of batches needed for model training

Some Results

- Estimated remaining model building time becomes more precise over time



Conclusions

- Our progress indication method can make deep learning more user friendly and accessible
- Welcome collaboration opportunities

Thank you