Overcoming Obstacle

In considering how to best present my open-ended research and how I overcome its challenges, I've decided to tell you about my most recent hurdle.

First, you will need a little background about me and my research. My ultimate goal is to make speech technologies available for all languages, and my research is a direct extention of that goal. Working with low-resource languages, my thesis topic is Multi-Task Learning for deep neural net acoustic modeling in Automatic Speech Recognition (ASR).

Acoustic models in ASR are statistical classifiers, accepting as input some speech signal (ie. time chunks of an audio file), and returning a probabilities over possible phonetic transcriptions. To accomplish this translation of audio → phonetic transcription, I train a feedforward neural net with multiple output layers. Each output layer represents a different task the net is required to perform. The logic behind Multi-Task Learning is that if two tasks are related, they will contain overlapping concepts. Learning related tasks in unison should improve performance on any one of the tasks, because the weights in the net will be biased towards general, task-independent representations of the data.

The Multi-Task approach offers an elegant way to exploit small datasets, as long as you can come up with the right tasks. Coming from a lingusitics background, I first saw each task as a way to inject phonetic knowledge into a neural net, via soft parameter biases as opposed to hard architectural constraints.

My early experiments show that by adding additional tasks along dimensions relevant to linguistics (place, voicing, manner), we can lower Word Error Rates for smaller data sets. I spent a lot of time thinking about how to best define each tasks, consulting machine learning and linguistic literature. I tried many different configurations until I found something that reliably showed improvement over my baseline.

However, if I want to be working towards techniques which can be used to quickly train an ASR system for a new language, then defining each extra task by hand is not an option. The expert-linguist approach is not scalable. Since those experiments, my time has been devoted to finding scalable solutions to task creation. In defining tasks by hand, the human is the bottleneck, so I've decided to take the human out of the picture. The auxiliary tasks need to be automatically generated, not hand-crafted by experts.

A task in supervised learning can be defined as a set of (label,data) pairs along with some function to map the data to the labels and an evaluation function of the performance of the mapping.

In Multi-Task Learning, usually tasks are added by finding a new set of labels for the data. However, it is possible to change the labels, the data, the mapping function or evaluation function to define a new task.

It is very easy to view any classifier trained via Multi-Task Learning as a kind of ensemble model. Typically, ensemble methods are trained separately, but in MTL, they are trained in unison with one feature extractor. My favorite ensemble method is the Random Forest, because it yields an unbiased classifier given the data, by training a multitude of separate trees on subsamples of the data, and then combining their votes. The Random Forest creates separate tasks not by changing labels, but by merely resampling the data. Each subsample will have its own decision plane, specific not only to the data but to the noise in that particular sample. By averaging the votes of all trees, the noise of each sample is ignored, and the generalities remain.

Taking inspiration from Random Forests, I am training Multi-Task feedforward neural nets, where the different tasks are independent subsets of the data. I use the Kaldi Speech Recognition toolkit to run experiments.

Recently I ran into an obstacle running my experiments.

Typically, I perform my experiments in this way: After I've defined a new training procedure, I first run a toy model on a small subset of the data, to quickly make sure there's no bug in the code.

Then, I run some pratice rounds with fewer parameters and fewer epochs on a larger subset of the data, to get a ball-park idea of model performance.

Finally, I define an architecture with a real number of parameters and epochs on all the data, and set a loop to train multiple versions of the same model (to get a standard deviation for the differences in random initializations).

My most recent problem came during the "practice round" stage. I was testing different sized random re-samples of the data as auxiliary tasks, and it seemed like the size of the re-sample made no difference at all. I tried re-samples all the way from 50% to 5%, but the performance was identical. Not only was the performance identical regardless of the size of the re-sample, but the baseline did just as well as all the other models. This was strange to me because in my expert-crafted tasks, I never got identical performance to the baseline.

So, I plotted the performance of the different tasks over time, and realized a major issue. The smaller tasks were training for less time, and as such, the main task was allowed to overfit the data after all the smaller tasks ran out. I was getting sequential learning instead of parallel learning.

I went to the `kaldi-help` Google group to see if I could get some help there. I posted my problem on the forum, and Dan Povey got back with some pointers. In a follow-up email he suggested that there might be a bug in my scripts resulting in a error for the vizualizations. I double checked my scripts, and the original data I used to plot, and there was no bug. He said in that case, it might be as simple as files being overwritten without any error messages to notify.

As it stands, he suggested I clear out all my generated experiment files and retrain from scratch, to make sure there's no file overwrite happening. I wiped my log files and the experiment is running on an Amazon instance as I write this.

This issue does not have a clear solution to it yet. I decided to share it because it is fairly typical of the issues I run into during my research. I spend all my work day either reading, writing, or running experiments in Kaldi. With my experiments, I will try to debug myself, search the official documentation for answers, and then go to the Google group. I usually have a back and forth with Dan, and he points me in the right direction to fix the problem.

My Research Passion & the Google AI Residency

Everyone should have access to speech technology in their native language.

This is the conviction that drives my research. I want the world to be a place where someone born blind doesn't have to learn a second language just to read Moby Dick. I've personally met very smart, dedicated people who couldn't attend university because they are blind, and the audiobooks they need to study don't exist in their native language. Working on technologies for the Kyrgyz language, I've played a small role to make speech synthesis available for free, but there is a hard limit as to what I can accomplish alone.

Google is hands down the world's leader in accessible language technologies, and I want to be a part of the team to make these technologies possible.

The Google AI residency is the ideal first step to accomplish my goals. I want to publish influential work and get more people caring about low-resource languages. With my knowledge of linguistics and passion for machine learning, the Google Brain and AI teams are the perfect place for me to flourish, and share my enthusiasm for research. Collaborating with researchers at Google, I will be able to help create new approaches and algorithms, testing theories on multi-task learning, and working towards better methods for low-resource languages. I have many ideas for multilingual experiments in particular, extracting information from big datasets to transfer knowledge to smaller domains. With the guidance of Google researchers, I will be able to efficiently run more experiments.

I see the list of languages *not* in Google Translate as boxes waiting to be checked off, and I want to help make that happen.

Thank you for your time and consideration.