Competition Overview and Postmortem

Monday, September 14, 2020 9:56 AM

- [NOTE 9/13/20: To be addended after competition winners publish their code and I can review it. At this point I've only seen the one solution from the end of competition webinar from leading teams.]

Competition Overview

- This was a very straightforward image segmentation challenge. Remote sensing and analyzing satellites like this is an important use case of these techniques, so it was fun to apply them to it. It was a pretty easy challenge and I think that it would have been better if there was something more challenging or interesting to the competition.
- https://www.aicrowd.com/challenges/ai-for-good-ai-blitz-3/problems/lndst
- Due to time constraints I really never had any time to explore the data and the mistakes that the model was making. I suspect that with a little bit more time I might have found some useful preprocessing which would have helped the models do much better.
    - It occurs to me that there has been a ton of ML work on remote sensing applications like this and some quick Googling would likely have uncovered some useful approaches.
- I was having memory issues on Colab, so I stuck with small models. If I had more time to figure out and deal with the memory issues, or just run them on a different platform, I would have tried more and/or bigger models.

Model Overview

- All I pretty much did was put together fastai's factory methods for image segmentation using unets built on transfer learning from cnns trained on imagenet
    - Using the standard freeze - train - unfreeze - train approach (called fine_tune in fastai's library)
- This worked super well. I was getting f1 (dice) in the mid 90's on validation sets during training. However, despite those good scores and also a variety of stacking approaches the best I did on the test set was f1=0.874
- I trained [resnet18, xresnet18, resnet34, xresnet34], I would have used more and/or more complex models, but I was having a lot of memory issues with the unet on Colab, so I stuck with these small models
- Different submissions on the test set I tried included:
    - each model individually
    - each model architecture cv fold stacked together (mean)
    - Each resnet depth (ie resnet18 and xresent18 together)
    - All models
    - Also tried a weighted average where resnet34 had a higher weighting than resnet18

Postmortem

- I did well on this competition and built a perfectly good segmentation model. I placed 14th the leaderboard with the leaders doing about f1=0.1 better than me, which is significant and suggests that there were quite a few things I left on the table.
- I just ran out of time for the overall Blitz competition and so because I had a model which was working very well I didn't spend much time optimizing it and was relying on stacking to help (which it did) and hoping that my validation set results would correspond a little more closely to the test set (which they didn't)