G2Net Kaggle Competition Winner Talk

Dzianis Kanonik - denisbsu@gmail.com

Challenge

- Short clips with rather low sampling rate (2 sec @ 2048 sps)
- Unknown parameters distribution
- Classification only, no hints about target signal location
- Extremely low SNR
- Unclear detection criteria
- External data unavailability
- Relatively low number of examples
- Portion of signals were undetectable by design

(Un)viability of conventional approach

Competition was designed to foil conventional approach in number of ways:

- Clips were too short to apply any meaningful form of signal whitening, only simple filters where short enough to work. Inability to properly standardize signals from different sources greatly reduced both matching filter performance and combined SNR calculations accuracy.
- Proper matching filters method require long signals for large parts of parameters space as signal evolution is just too slow.
- Advanced SNR calculations (newsnr) works better with longer signals, 2 seconds is not enough to filter out false positives.

Common G2Net competitors approach

- Filtering: high-pass + notches
- Augmentations
- 1D->2D: Various spectrograms
- Usual Image Processing Module: EfficientNet
- Adam optimizer
- Ensembling
- Later in competition focus shifted to learnable 1D->2D representations

Our approach

- Quickly realized that learnable representation approach yields better results
- Heavy use of augmentations
- SGD optimizer to combat overfitting
- Simpler Image Processing Network (ResNet34) to combat overfitting
- Identified lack of data as main roadblock
- Aggressive search for additional data sources
- Experiments with conventional approaches and pipelines

Data generation

- We were able to generate correct noise for each of detectors rather early in competition.
- Artificial noise helped to combat overfitting, but score increase was quite modest.
- After few tries we were able to generate artificial signals, but parameters distribution was still a mystery for us.
- Direct mixing of datasets yield worse results as parameters distribution was completely off.
- Using generated dataset as pretrain and provided as finetune got us another modest score boost.
- Finding correct parameters distribution was paramount.

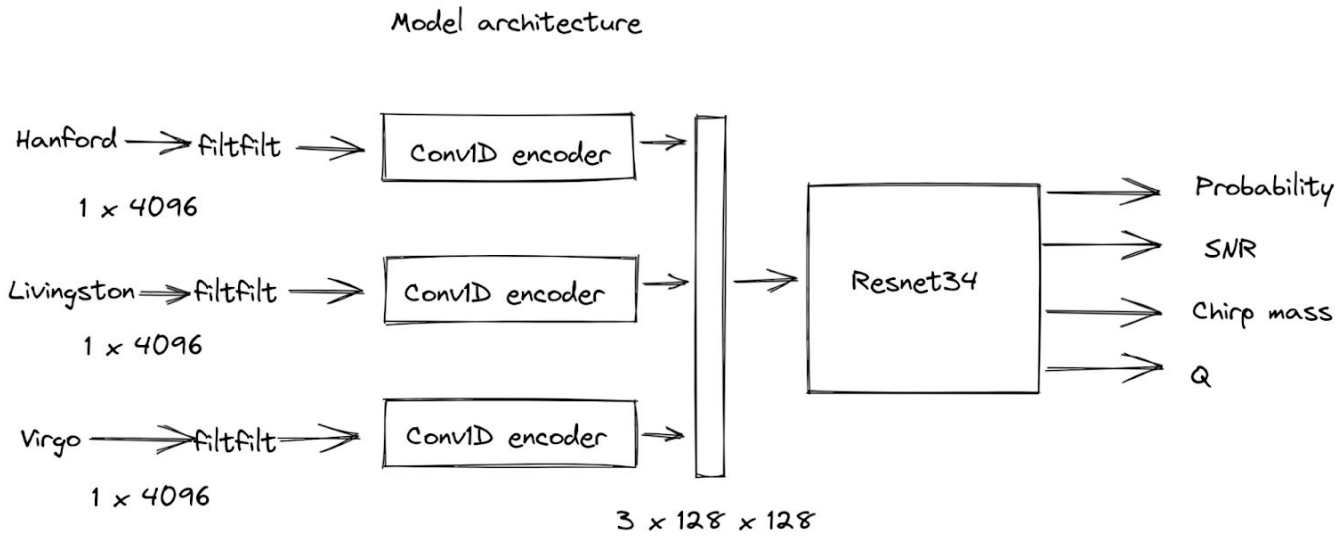
Parameters Distribution recovery

- Parameters in question: Chirp Time, Chirp Mass, Q, Distance, Spins
- Started with reasonable assumptions about limits and uniform distributions
- Select one parameter to optimize
- Generate dataset
- Train network on generated dataset with parameter as regression target and default classification target
- Predict provided dataset and update distribution
- Repeat, using classification target as “distribution closiness” score

Putting it all together

- For Chirp Time, Q and Distance we found more precise boundaries and uniform distribution
- Chirp Mass distribution was non-uniform and correct distribution boosted our score greatly
- Spins distribution did not matter
- Pre-training network with all regressions enabled gave us another significant boost. Chirp Time was the most influential auxiliary target.
- Using Chirp Time as mask to guide 2D features generation yielded interesting results, but did not improve score.

Final architecture



Interesting Findings

- Around 30% of signals in the train data hit SNR wall (were indistinguishable from noise even if the exact signal was used as a filter)
- 5% of noise samples contained something perceived by NN as strong signals, true for both generated and competition noise
- When training on these samples alone the model was able to generalize and predicted signal samples as noise and vice versa, giving 0.86 AUC (with 1-P)
- Learnable filters can do better than conventional frequency-domain representation
- Results suggests that better noise source needed for deep learning approach

Thank you