- #paper
  - https://arxiv.org/abs/1803.03635
- The technique described in this paper may be useful to create the Speech-to-API Framework.

Summary

- The general idea is to find efficiently trainable subnetworks — *winning tickets* — in a Neural Network by Pruning it during the Training Process.
  - Evidence shows that the *winning ticket* weights change by a larger amount than the rest of the network.
  - Since a *winning ticket* learns faster than the rest of the network, we remove a percentage of the weights with the lowest magnitude. Pruning happens iteratively.
    - The initialization of *winning tickets* is crucial — if we keep the mask of a *winning ticket* (to have its structure) but reinitialize all the weights and prune outside of the mask, the *winning ticket* will learn progressively slower.
- Hence **the Lottery Ticket Hypothesis**:
  - > *A randomly-initialized, dense neural network contains a subnetwork that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations.*
- *Winning tickets* are easier to find while training a dense network as there are more possible subnetworks than in a sparse network.
- The resulting *winning tickets* can contain up to ~97% less parameters without losing in Accuracy (more than that, a certain increase in accuracy is possible for moderately pruned models) and training speed (generally, they can be trained even faster).

        - The improved ability of *winning tickets* to generalize is reflected in an Occam's Hill in the charts as the test accuracy increases and then decreases as pruning intensifies.
    - The authors draw parallels with Dropout, mentioning that it's often characterized as *simultaneously training the Ensemble of all subnetworks* while with pruning, we are moving toward the most promising one from the beginning.
    - *Winning tickets* are robust to Gaussian Noise: adding that of 0.5σ barely affects Accuracy.
    - It is possible to employ One-shot pruning to make the process more computationally efficient.
    - ### Drawbacks
        - The structure of *winning tickets* may be affected by training data.
        - The process of finding *winning tickets* is computationally costly — it may be hard to apply to the models with an excessively vast number of parameters.
    - ### Negative hypothesis testing
        - Random reinitialization or sparsity don't show much improvement in the model performance.
        - Pruning before training impairs the performance of the network.

- # Questions
    - May an adversarial attack on a *winning ticket* be easier than that on the whole network?
    - What if we train a winning ticket on different data? How much difference in the data can we have without ruining the performance of the network?
    - May there be a kind of a "winning ticket" Subgraph in the Graph of all scientific papers? Meaning: a Subset of the pieces of Research that are most helpful for studying.

- # Ideas

- Pruning based on how much each weight after an iteration of training differs from itself before that iteration. That is, we may try to mask out the weights that change too slowly *locally*.
