# Progress Report

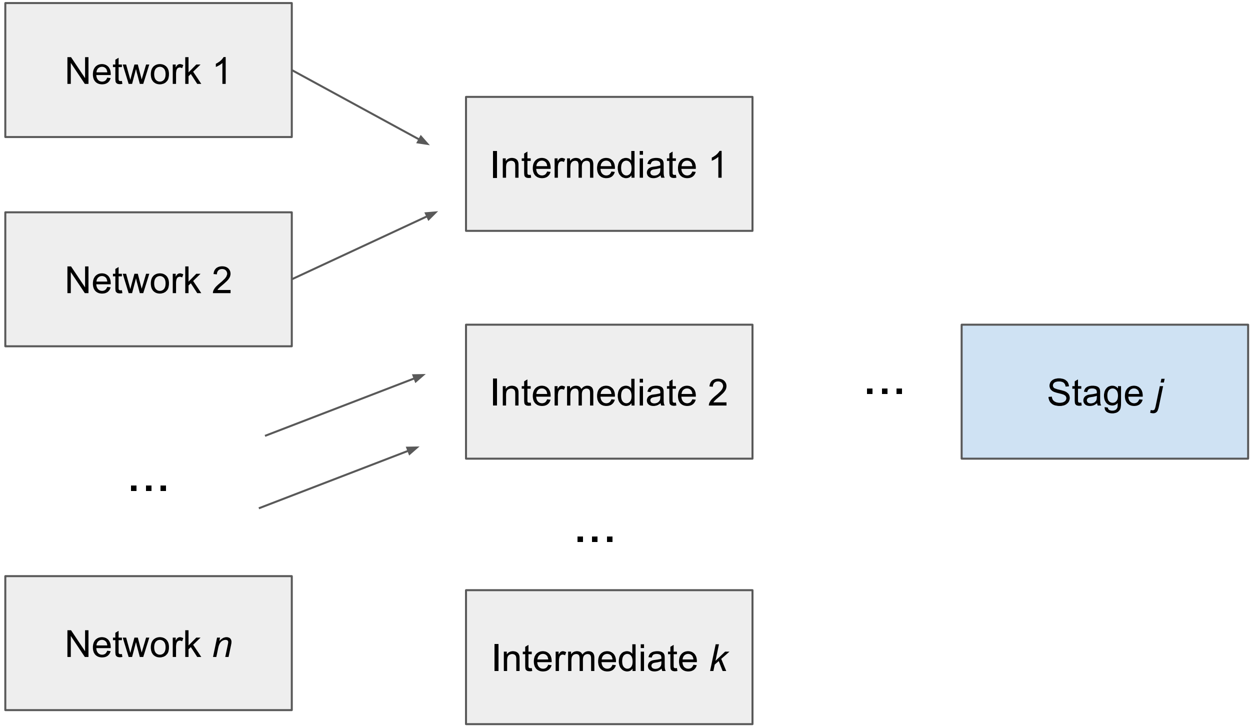
Pranav Bangarbale
11/16/2021

# Review of Last Week’s Ideas

- **Intermediate weight averaging/combination**
- Pruning intermediate networks
- Parallel training
- XGBoost

# Concept: Intermediate Weight Averaging

- At arbitrary epoch intervals, average weights of random “groups” of networks
- For example, if 500 replicas are generated, average the weights of 10 groups of 50 networks at halfway through the training period and then show the CFF distributions created by the resultant 10 networks through the end of training
- Should drastically reduce the training time required



Network 1

Network 2

…

Network *n*

Intermediate 1

Intermediate 2

…

Intermediate *k*

…

Stage *j*

Where *n* is the number of replicas

Where *k* is the number of intermediate networks

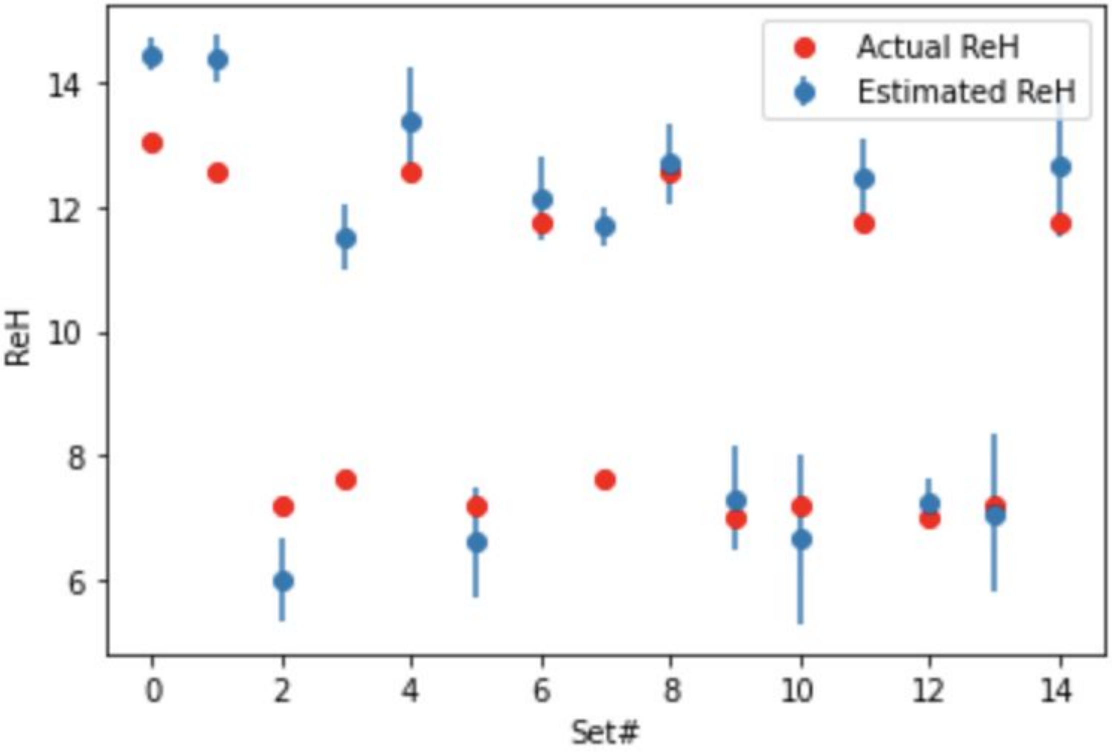
Where *j* is the number of times weight averaging occurs

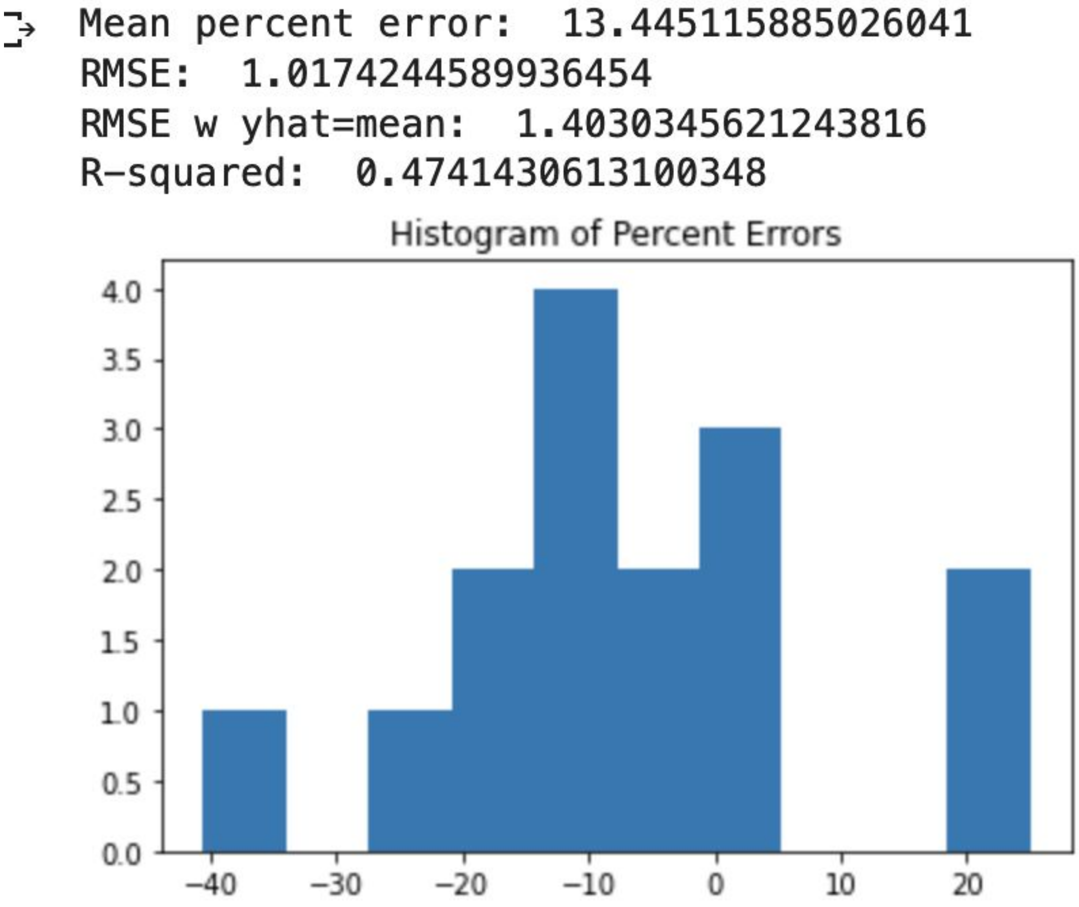
# Further Implementation Details

- As explained earlier, *n* is the number of networks/replicas, *k* is the number of intermediate networks, and *j* is the number of stages.
- For the purpose of trying this concept, I simply utilized *n*=250 *k*=25, *j*=1
- As such, 10 randomly generated groups' weights were averaged at halfway through the training process (750 epochs)
- The training then continued for another 750 epochs but only for these 10 networks
- Final CFF distributions of these networks were generated

# Distributions

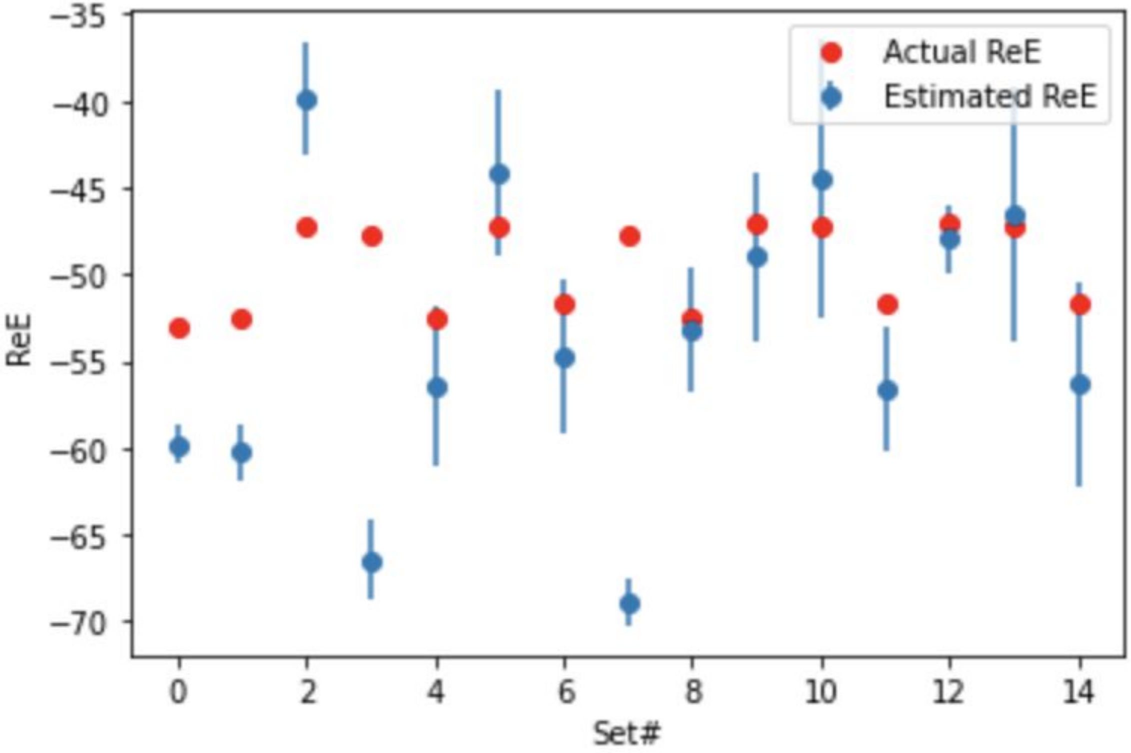
# ReH



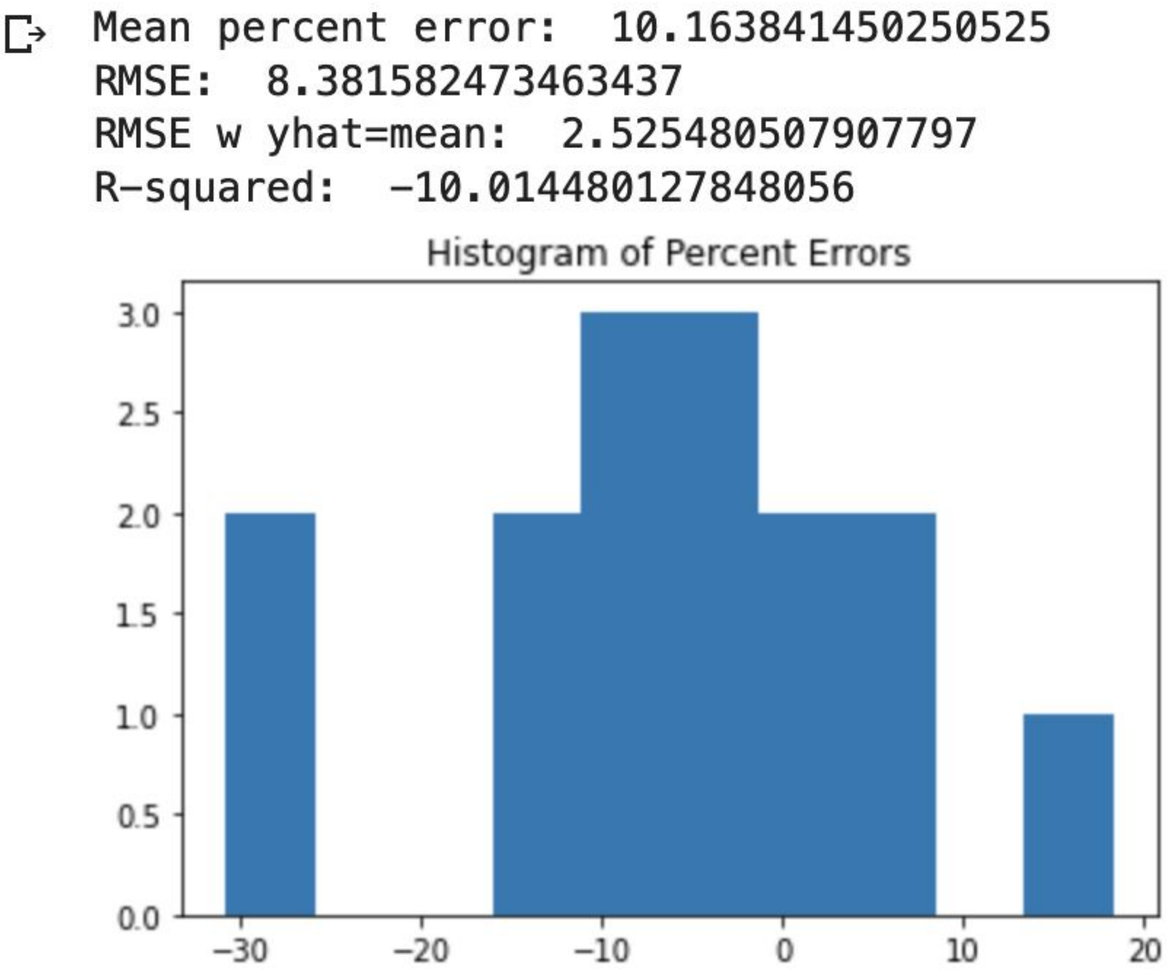


```
Mean percent error:  13.445115885026041
RMSE:  1.0174244589936454
RMSE w yhat=mean:  1.4030345621243816
R-squared:  0.4741430613100348
```

# ReE

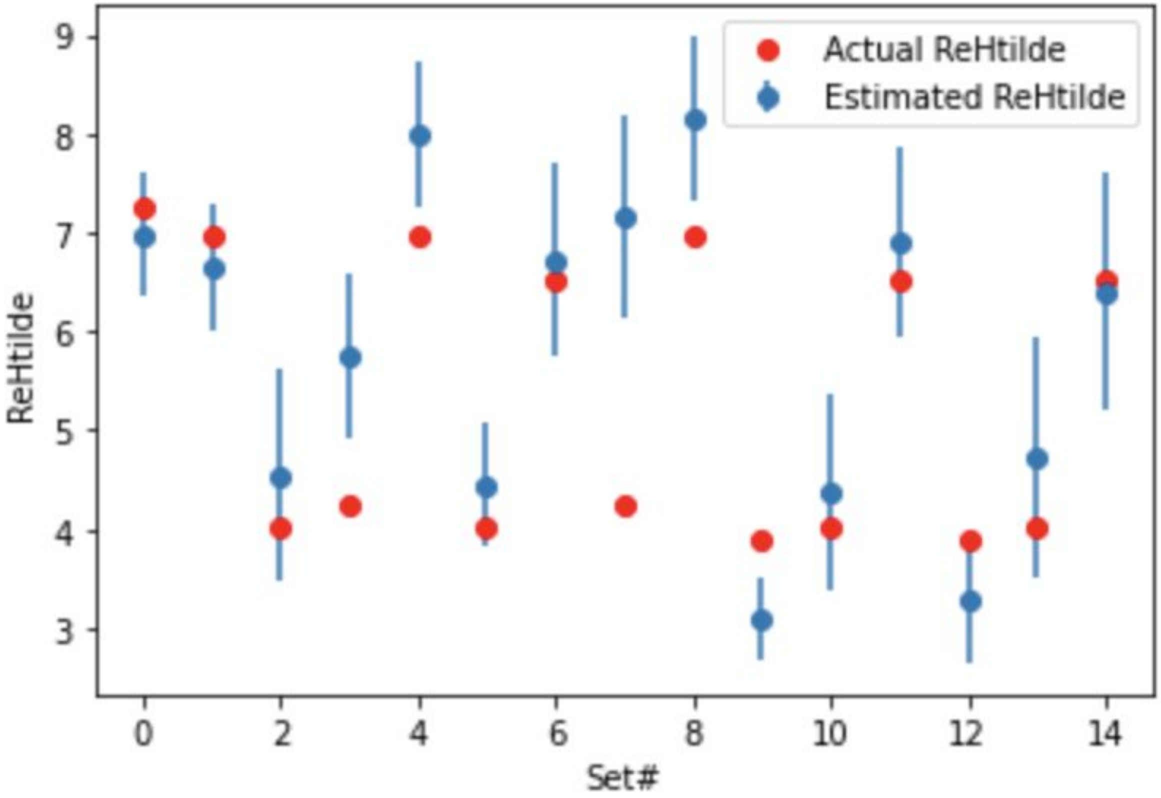


● Actual ReE
● Estimated ReE
ReE
−35
−40
−45
−50
−55
−60
−65
−70
0
2
4
6
8
10
12
14
Set#

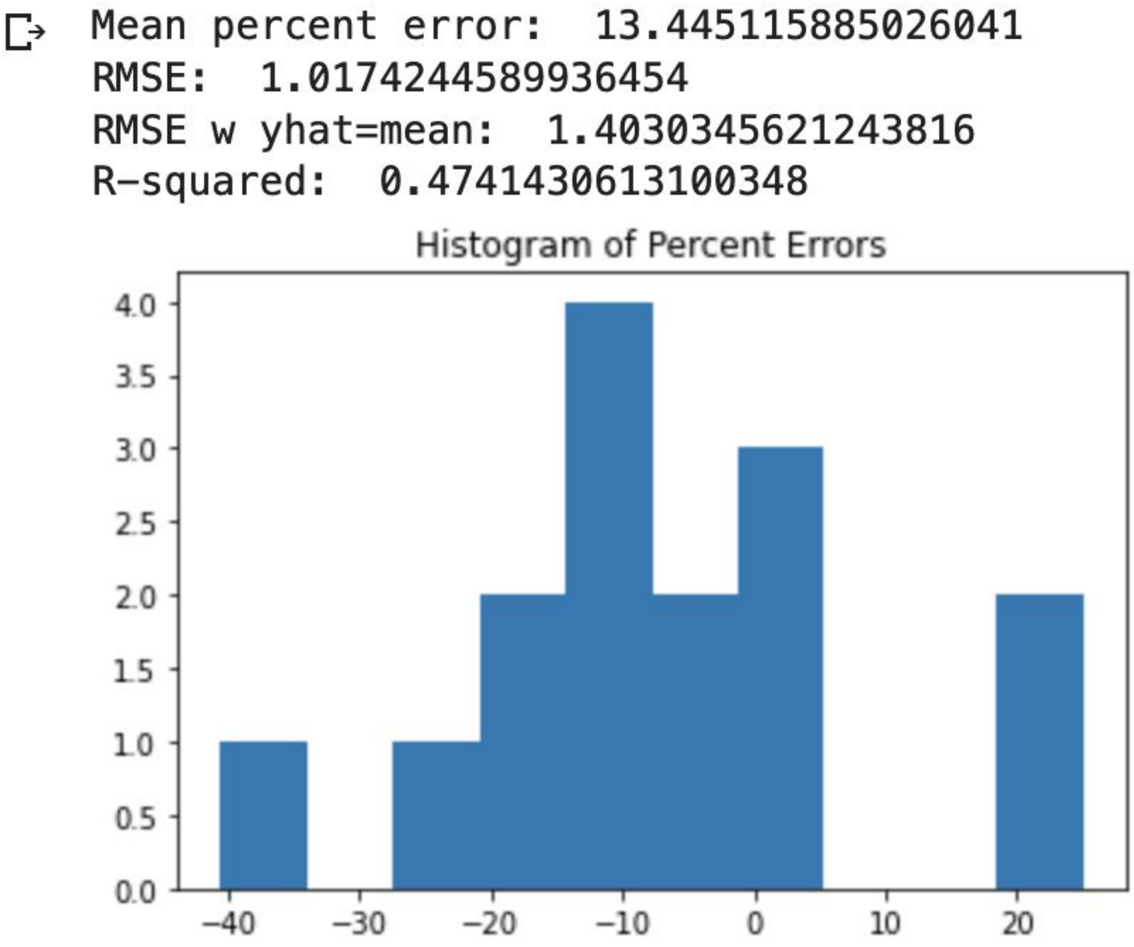


```
Mean percent error:  10.163841450250525
RMSE:  8.381582473463437
RMSE w yhat=mean:  2.525480507907797
R-squared:  -10.014480127848056
```

# ReHtilde



9
8
7
6
5
4
3
ReHtilde
0
2
4
6
8
10
12
14
Set#
● Actual ReHtilde
● Estimated ReHtilde



```
Mean percent error:  13.445115885026041
RMSE:  1.0174244589936454
RMSE w yhat=mean:  1.4030345621243816
R-squared:  0.4741430613100348
```

# Reasons for Performance

- Random groups are not necessarily making the combination of multiple networks “smarter”
- Networks should be combined in a more algorithmically sound way
- This could dramatically boost results of weight averaging
- **Overall takeaway: promising technique, just needs to be implemented correctly**

# Future work

- **Using other methods of deciding network groups (e.g. averaging those that are closer together?, etc)**
- Using other methods of combining weights
- Trying different combinations of n, k, j