Suggestions Page

We put this together as a list of specific suggestions about problems we frequently see. These are outcomes that can be difficult to identify as problems when working with dendrites for the first time. We are hoping this page will minimize folks submitting a final project which is just a few adjustments away from being successful, without realizing the project is not quite finished yet.

Try/Excepts with Dendrites

If you are running into an error it means something needs to be debugged. Most likely something is being used incorrectly within the API, but it is also very possible that there is something wrong on our side. Either way, do not just wrap it with a try/except and risk a core function not being used. If you can't debug something, ping us. We will guide you on what might be going wrong, or ask you for more details to see if it might be something we need to debug on our side.

How to do Compression Experiments

When you are running a compression experiment you <u>MUST</u> run a minimum of 3 experiments.

- Traditional Network baseline - this is the baseline network you are compressing
- Network compressed with dendrites - this is the new architecture of the smaller model you are adding dendrites to
- Traditional Network compressed without dendrites - once you have determined the optimal smaller network, it should also be run without dendrites.

The reason to do this is because we have seen projects before where the smaller network without dendrites is also equally accurate and it's not the dendrites enabling it to be better. For compression experiments we recommend the following protocol:

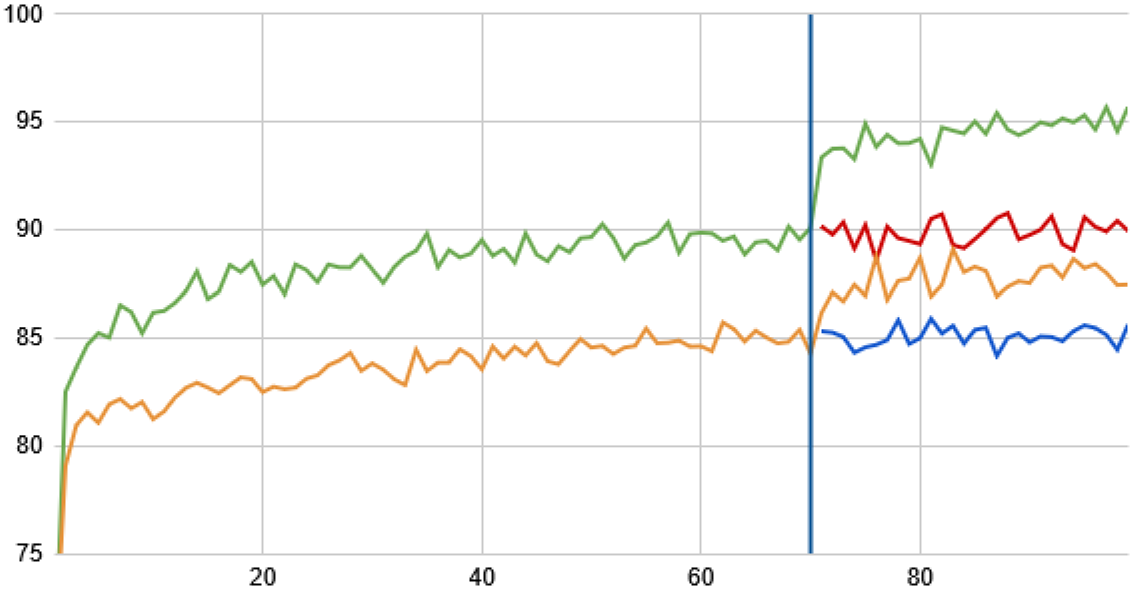
- Sweep over a handful of architectures both with dendrites and without dendrites
- For the network baseline choose the highest accuracy non-dendritic model
- For the dendritic model choose the smallest model that has a better score than the best traditional model
- For the third model chose the traditional model with the same settings as the dendritic model.

Feel free to make minor adjustments in your choices if various scores are close to each other. But for whichever dendritic model you chose be sure to include a non dendritic model with all of the other hyperparameters the same.

Interpreting the Output Graphs

We frequently see submissions where scores look like the project is working but once we see the graphs that are generated it is clear settings are not being used properly. These are some frequent types of graphs that are output into the folder when training with dendrites. With the default save_name these graphs will be located at PAI/PAI.png

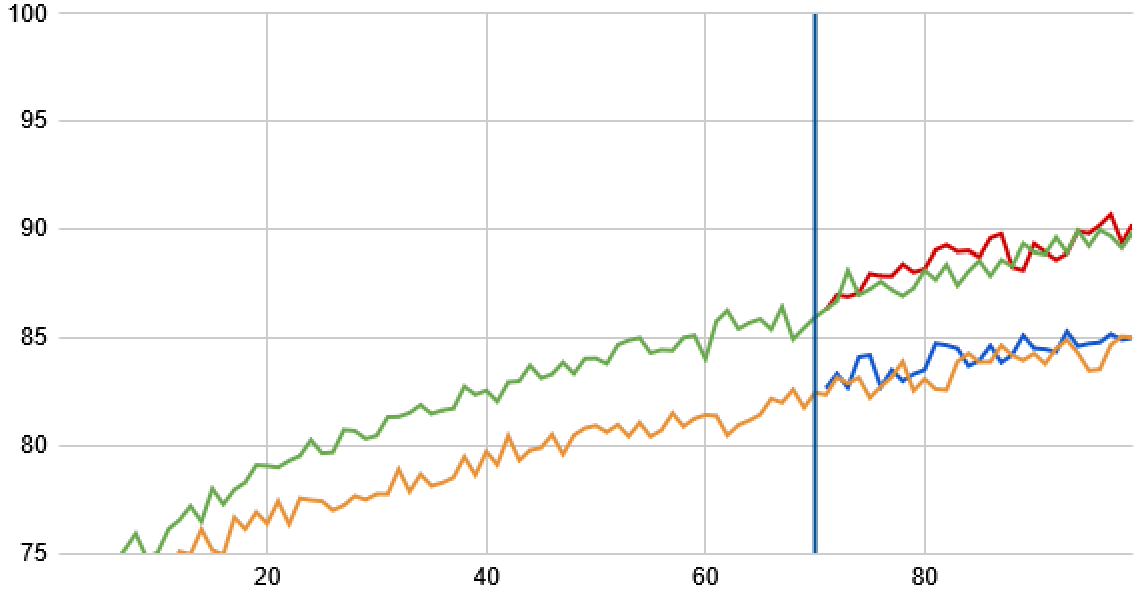
What a Good Graph Should look like



This is what a good graph should look like. The X axis here is epochs and the Y axis is the accuracy scores. If you are minimizing loss rather than maximizing a score this should just be flipped vertically.

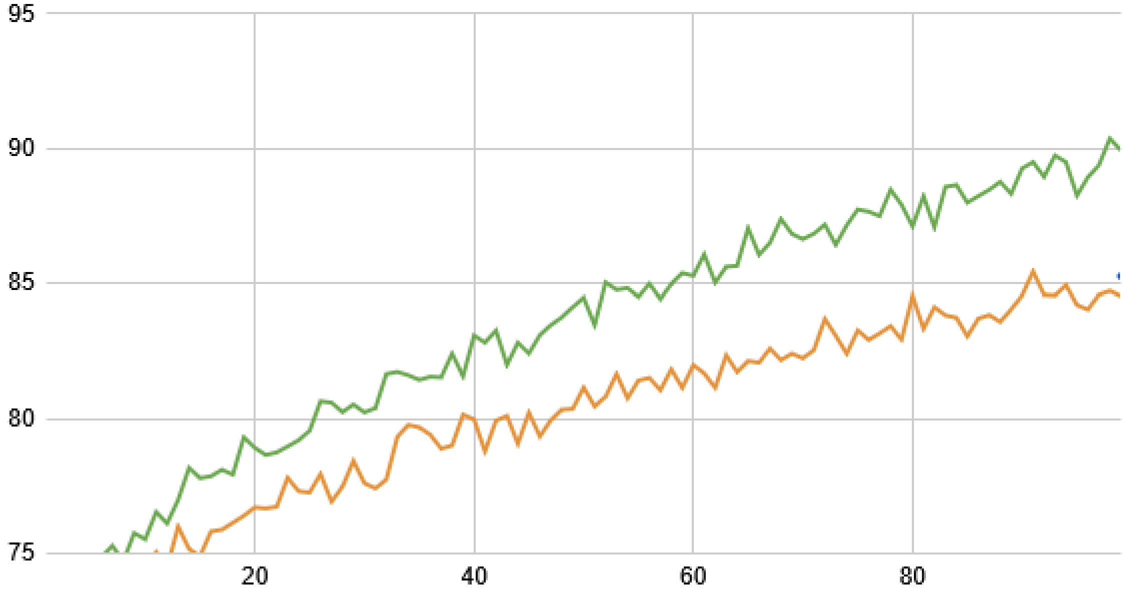
- The vertical blue bar represents the epoch where dendrites were added.
- The green line is the training scores, the orange line is the validation scores.
    - If things are working well what you will see is scores getting better over time before starting to plateau. Once plateauing has started dendrites will be added. After this blue line the training and validation scores should both get a spike before reaching a second plateau. Sometimes there will be a dip initially before getting better in future epochs
- On this graph the red and blue lines represent what would have happened if dendrites were not added. When dendrites are added the best model is loaded first, so these are epochs that were actually run before loading the best model. There should be a clear difference between what would have happened without dendrites and what actually happens with dendrites.

Graph Problems 1 - Early Addition



If you see a graph like this, it could mean that you are adding dendrites too early and just more training would also cause the gain. This can be seen when the blue and red also continue to go up after the blue line and there is not a clear difference between blue and orange or red and green.

Graph Problems 2 - No Dendrites



If you see a graph like this it means dendrites have not been added at all. The vertical lines show where dendrites have been added so if there are no vertical lines it means one of two things. One is that you just need to train for more epochs. When using dendrites you should **always** train until training_complete returns true. We recommend editing your for loop from a

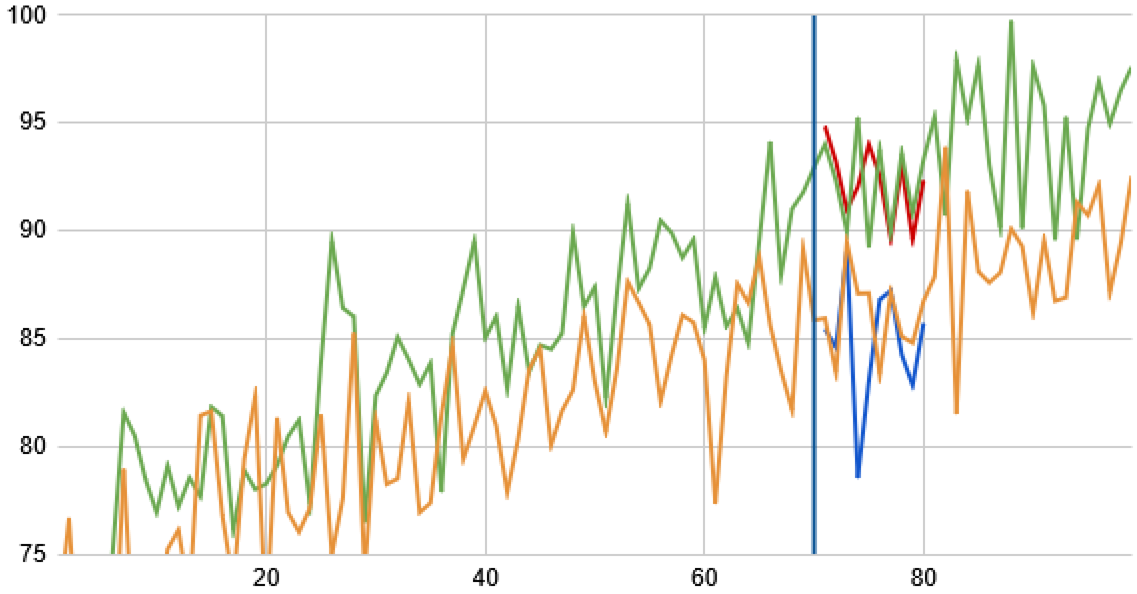
```
For epoch in range(max_epochs):
```

Into a

```
Epoch = -1
While True:
   Epoch += 1
```

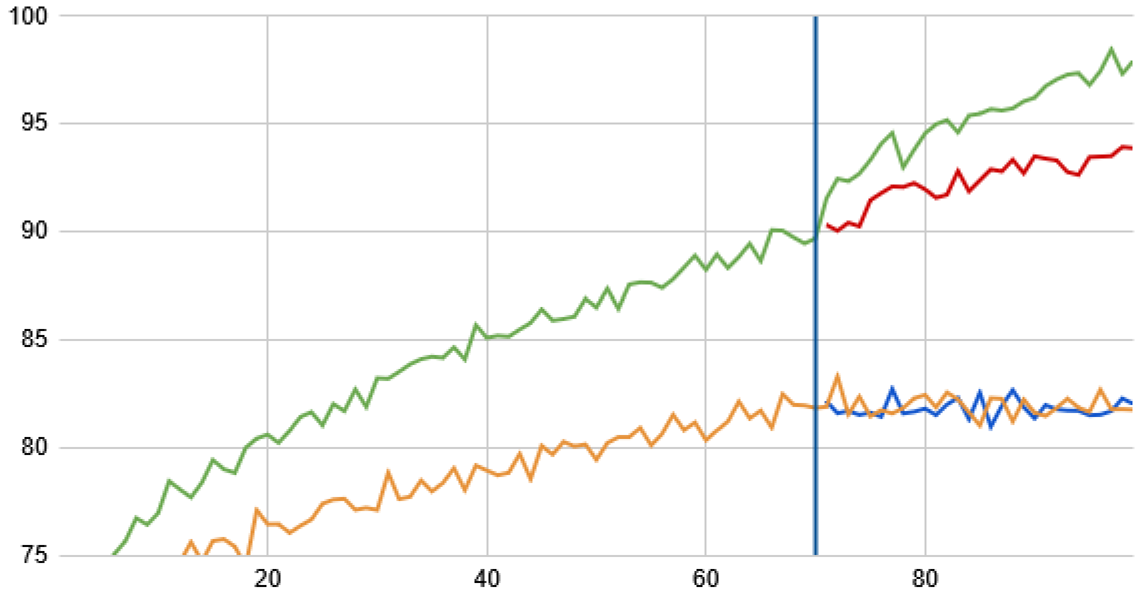
Then break this loop when training_compelte is true. A second thing that could be happening is that dendrites aren't actually improving anything. If that folder contains files with a no_improve in the name that means that the system did try to add dendrites but they didn't help. If this is the case for you feel free to reach out for more guidance or check the suggestions in customization.md.

Graph Problems 3 - Just Luck



If your scores have high oscillations like this, it can often trigger early stopping like in the first problem example. When you have higher oscillations you need to wait longer before switching to be 100% sure that more epochs wouldn't have caused the gains. This can be done with GPA.pc.set_n_epochs_to_switch(higher number)

Graph Problems 4 - Overfitting



Finally a common problem with dendrites is overfitting. In this graph take note of the red and blue lines. As you can see over time the red line keeps getting better and better while the blue line has flatlined. This means that you are already overfitting without dendrites and better training scores are not improving your validation scores. If this is the case, improving your training scores with dendrites won't help any more than improving them without dendrites.

If you are in this situation try various methods to counteract overfitting, including regularization techniques like higher dropout rates. Additionally, if there is a clear difference between red and green, but not blue and orange, this could be a hint that the architecture is compatible, but the overfitting is already to strong. Often, this means this could be a great case for a compression experiment rather than an accuracy experiment. See if you can make a smaller model where overfitting isn't happening where dendrites can make it as accurate as the original model.