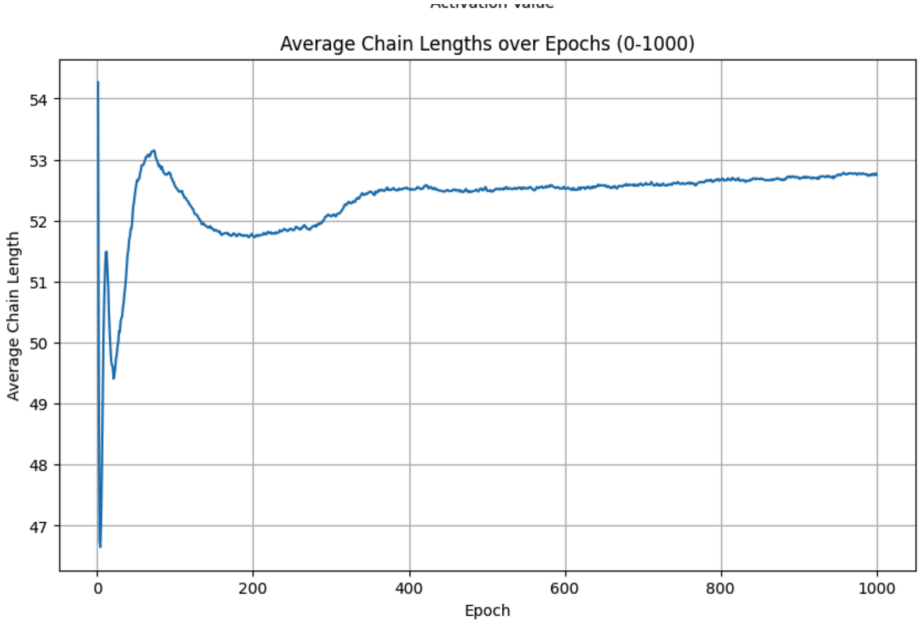
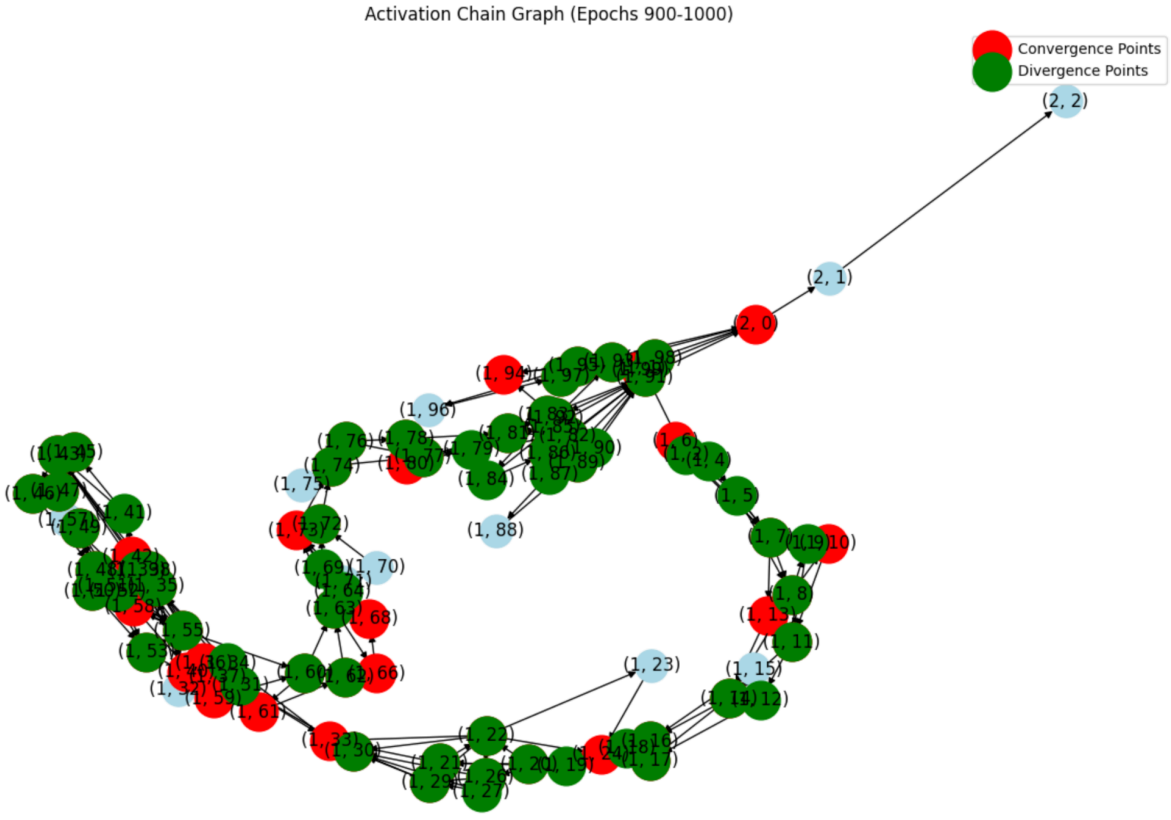
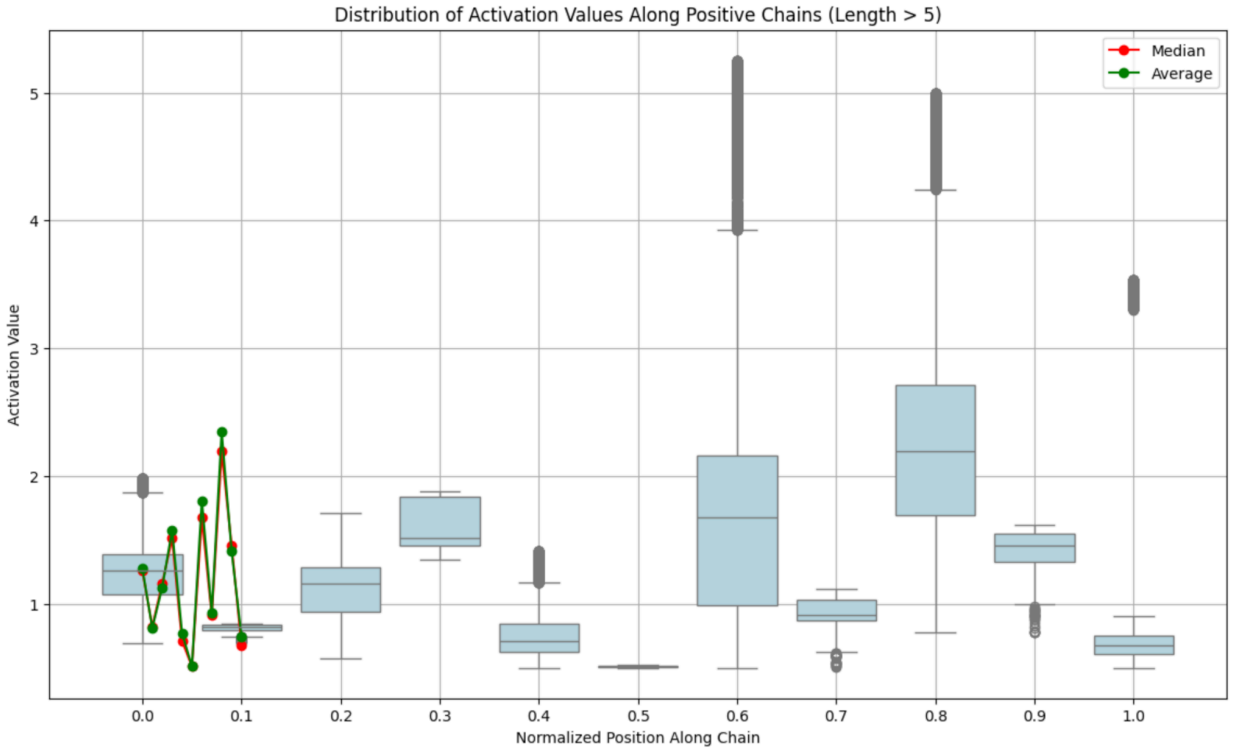
I’m analyzing the chaining of activations in NN specifically my from scratch model. Trying to learn what chaining is like in a model to see if I can build a new model training out of it.

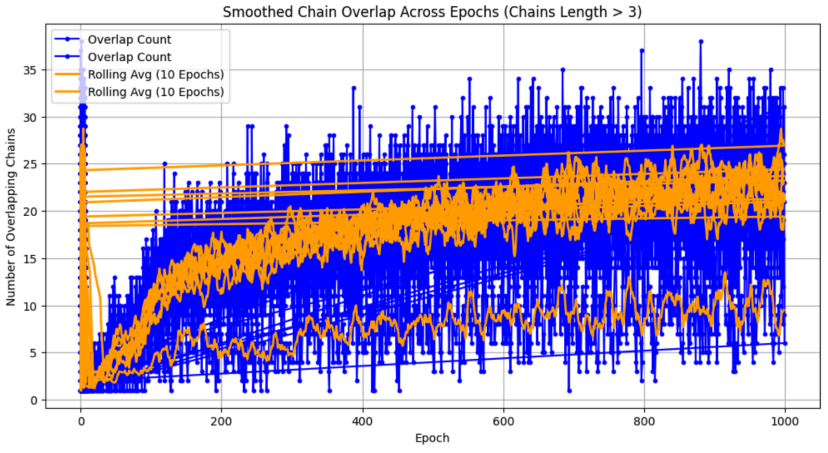


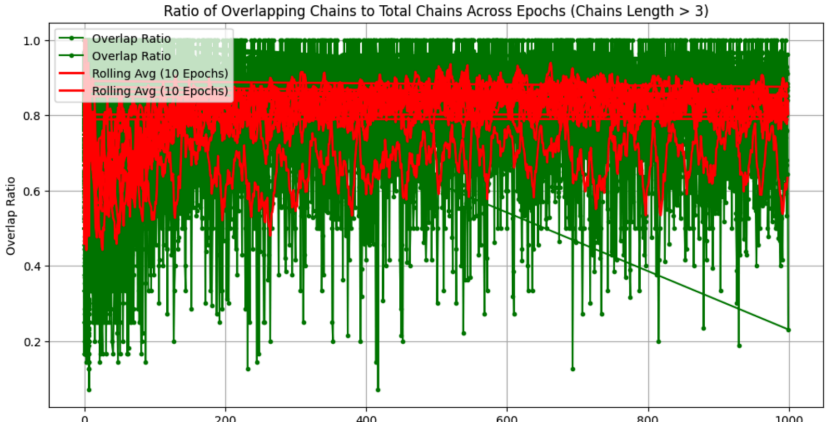


Median avg is stupidly plotted but it is right



Bad plots again but it does show the overlap interesting that it increases over time and plateaus near then end. These are the chains I'm interested in targeting.





So for a version 0:

Need a warmup period before I start connecting the chains. I want it to be more stable chains like it is in the later training epochs.

Choosing to build chains only over a certain length is probably critical.

No connections:overfits
Epoch 309/1000, Loss (Train): 0.0244, Loss (Val): 0.0132
Epoch 999/1000, Loss (Train): 0.0007, Loss (Val): 0.1789
Epoch 1000/1000, Loss (Train): 0.0007, Loss (Val): 0.1789

Random connections: 0.3 add, 0.2 remove, no overfitting, way better result?
Epoch 1000/1000, Loss (Train): 0.0036, Loss (Val): 0.0007

But it isn't reproducible, sometimes better sometimes worse in training. However somewhere in the training it is seemingly always better than the standard method.

Kind of acts like a regularization method but less stable.

1000 hidden layers:
No randoms:
Epoch 1000/1000, Loss (Train): 0.0000, Loss (Val): 0.0028

0.3 add 0.2 remove
Epoch 926/1000, Loss (Train): 0.0001, Loss (Val): 0.0015
Epoch 1000/1000, Loss (Train): 0.0001, Loss (Val): 0.0030

0.8 add 0.4 remove
Epoch 864/1000, Loss (Train): 0.0001, Loss (Val): 0.0001

Basically ~same but will still have better results somewhere within the training and quicker. Not the most stable, sometimes it's better sooner but sometimes it's near the end.

Even add and remove never trains as well.

Standard training can get very similar to the low level but it takes much longer and often will never reach it
Standard:Epoch 15477/100000, Loss (Train): 0.0000, Loss (Val): 0.0009

I like it. would like to add smarter methods of connection

Even with a new line added I still see a decrease of accuracy on word prediction as training continues?

9-25-24: Thoughts

This wasn't a bad idea I should go back and try to implement a training algorithm that the model itself can determine where to place chains. This would look like a chain connecting across the connections and continual training. I should implement something like this in a more complex and high level method.

Low level algo design

Long training runs in test.

Analyze evolution of the connections over time.