A Starting Point

After several days of research, training, and coding, I've completed the first adaptation of my AMP transformer model. The model itself is simple and unpolished, but it serves as the foundation on which I plan to expand and refine this project over the coming weeks and months.

Features

Multi-Headed Attention:

This version introduces several key changes compared to the original variational autoencoder architecture. The transformer design is more robust and is organized into five subclasses that together form the overall AMPTransformer model. The two most notable components are the Head and MultiHeadedAttention classes, which enable the model to effectively communicate with itself, an idea I've discussed in earlier blog posts.

Token and Positional Embedding Layers:

Beyond self-attention, this model gains a significant structural advantages from the token and positional embedding layers. These layers effectively act as a lookup table that embed each token in a higher dimensional space, considering both the amino acid and its relative position in the context window.

Layer Normalization

I implemented pre-norm layer normalization to stabilize training across the transformer architecture. Without this, gradients would explode when processing data with diverse compositions. The normalization maintains consistent activation scales, allowing the model to handle peptides of varying lengths and properties within the same batch, which proved very useful for convergence.

Dropout Regularization

I added 20% dropout to prevent overfitting on my limited dataset of ~3000 AMP sequences. This forces my model to learn robust, distributed representations rather than memorizing specific peptides. Without dropout, generated sequences would be near-copies of training data.

Residual Connections

I incorporated residual paths to make my model depth tractable. These skip connections solved the vanishing gradient problem I initially faced, where deeper models performed worse than shallow ones. The residuals allow gradients to flow directly to early layers while preserving both simple features throughout the forward pass.

Flaws and Next Steps

Training Data and Training Time

Moving forward, this project will requires lots of changes to become anything close to production grade. This projects main bottlenecks include the amount of training data, and the amount of time spent during the training phase. To establish a more thorough model that can make better predictions, I will greatly expand the quantity of data it is trained on by curating data from several databases. To develop better training speeds, I would like to use cloud computing, or potential computational resources at my school that can allow my model to refine its predictions.

Testability

The main component of this project that I would like to address is its ability to determine wether or not these AMP sequences are viable, rather than just creating pretty pictures. In the near future, I should be able to create a program that

can empirically test certain features of the protein sequences outside of the transformer architecture itself. This will require further research into the biological-side of the project which I am looking forward to.

pretty picture: