MicroFold Project Report

This project report outlines the design and testing of a simplified AlphaFold implementation, aimed at predicting how proteins fold based on their amino acid sequence. The project is based on a basic framework provided by James Atlas, and all the associated code is part of the submission.

Experimental Design and Methods

The core idea of this implementation is to map the amino acid sequence into a latent space, where a decoder is trained to predict the distance matrix. A precomputed distance matrix is used, and an attention matrix is applied to it. The attention-weighted matrix serves as the latent space representation for the decoder.

The simplest approach uses a neural network that processes the one-hot encoded amino acid sequence and converts it into an attention matrix through two fully connected layers with a leaky ReLU activation, as shown in Figure 1.

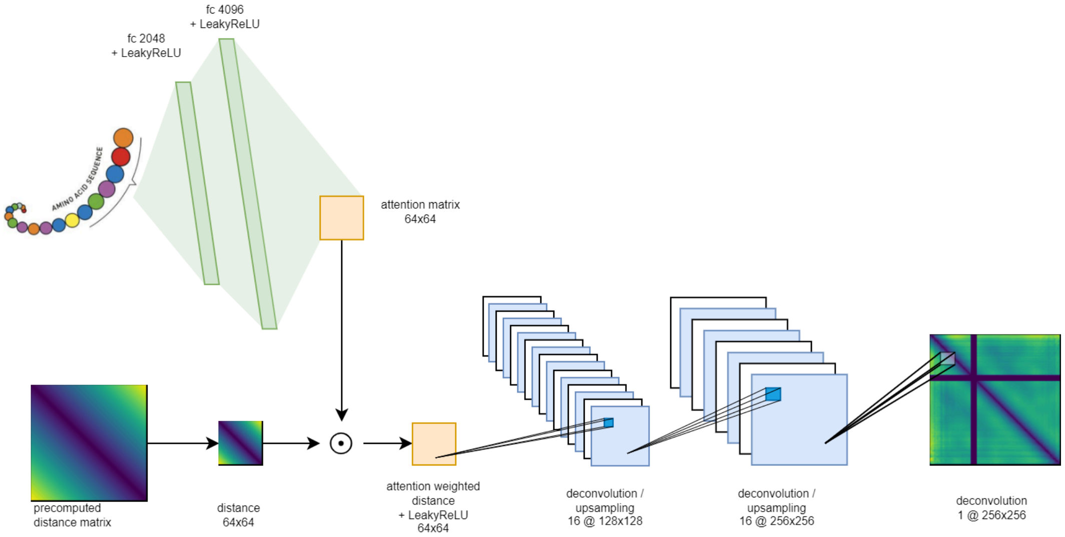

Figure 1: Architecture of the first prediction model

Training this network reveals signs of overfitting, as seen in the recorded losses during training (see Figure 3, solid lines). While the training loss decreases with each epoch, the validation and test losses either remain unchanged or increase, clear indicators of overfitting.

To address this, the network complexity is reduced by replacing the first fully connected layer with two convolutional layers (see Figure 2). This not only improves the models ability to generalise through the use of partially connected neurons but also significantly reduces the network size from 74MB to 16MB.

Training this modified network shows that although the training loss remains higher, the validation and test losses decrease over time, ultimately outperforming the previous implementation (see Figure 3, dashed lines). This demonstrates that the more lightweight model generalises better.

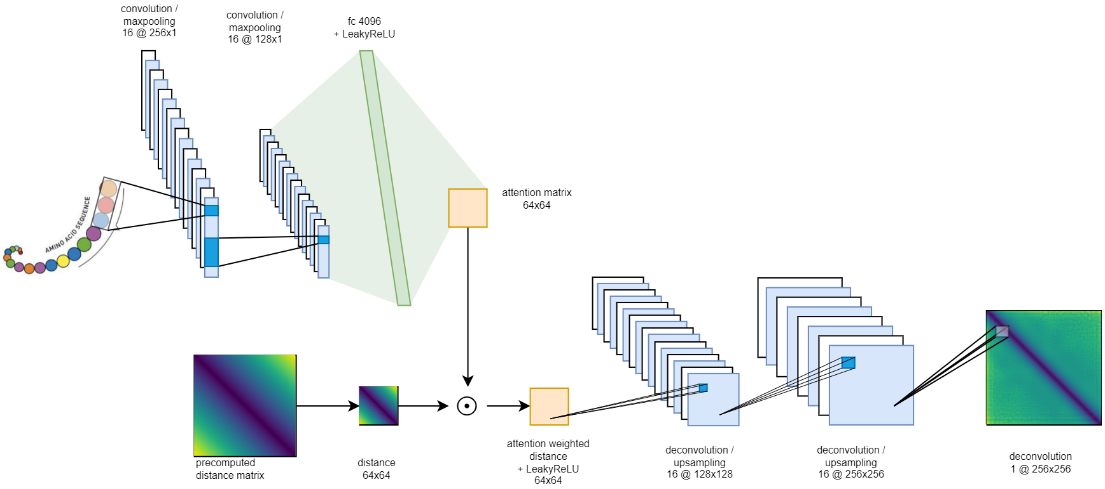

Figure 2: Architecture of the second prediction model

Results

As discussed earlier, Figure 3 shows that the initial model tends to overfit, while the training loss drops, the validation and test losses remain constant. In contrast, the second model, although initially performing worse, sees both training, validation, and test losses decrease over the course of the epochs. Both networks are trainable within 20 minutes on a Google T4 TPU via Colab.

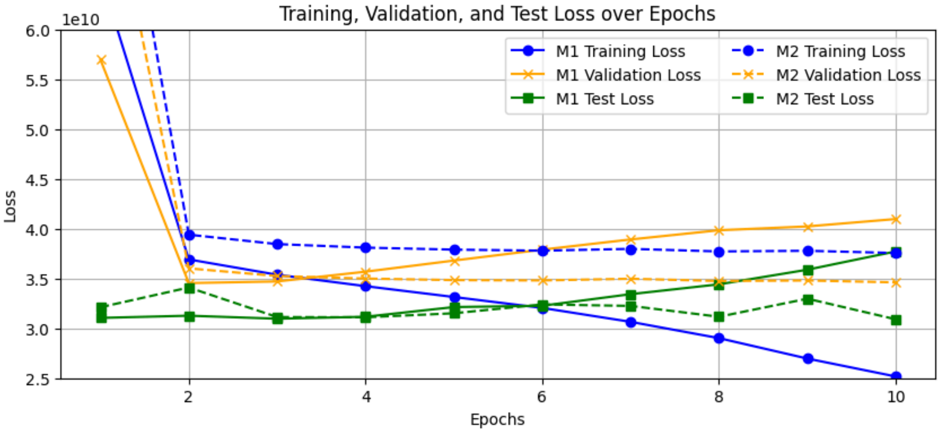

Figure 3: Training, Validation, and Test Loss over Epochs

Conclusion

The introduction of convolutional layers to simplify the model led to the expected improvements, the model no longer overfits at the current training scale. This approach is commonly used in computer vision and also introduces translational invariance.

Further investigation could focus on optimising kernel sizes, filter bank numbers, and layer configurations. The current setup was based on heuristic choices, which may have unforeseen impacts on the network's performance.

While this implementation is far from capable of predicting protein folding with the accuracy of AlphaFold, a future direction could involve incorporating a more advanced transformer-based architecture to map the amino acid sequence to the latent space.