Understanding the significance and architecture of AlphaFold 2

Jiaxian (David) Li

lidavid174@gmail.com | 0488 019 219

The Protein Folding Problem

Proteins are large and complex molecules that form the basic building blocks of life. They are responsible for essential tasks such as forming antibodies and transporting and storing nutrients. Since a protein's three dimensional (3D) structure has a tangible impact on its function and utility, much of recent work in biology pertains to solving the Protein Folding Problem: to predict the 3D shape of any protein given its contiguous chain of amino acids.

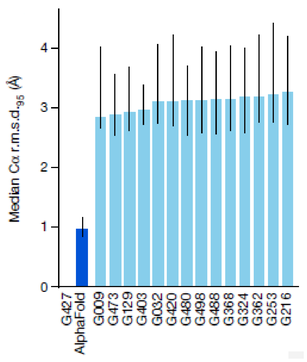

Figure 1: AlphaFold accuracy[1]

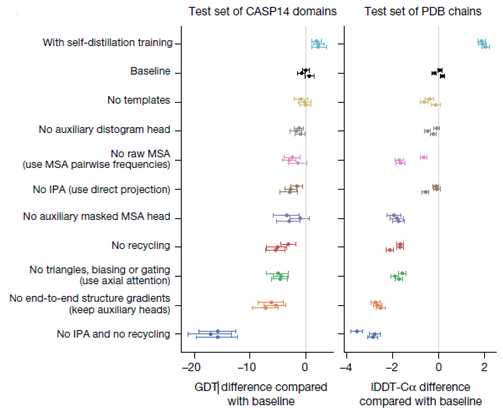

Figure 2a: ablation studies[1]

AlphaFold's Significance

Hence, AlphaFold's success stems from its ability to understand concepts, reason from them and create original knowledge. This builds upon the previous success achieved by DeepMind in narrow domains with man-made rules, such as Chess and Go. However, in solving the Protein Folding Problem, AlphaFold has allowed AI to make the jump to solving fundamental biological problems that could tangibly impact humanity in the years to come.

Therefore, AlphaFold's success is like a stepping-stone in the AI development journey, bridging the gap between narrow domains that demand strategy, and gaining brain-like capabilities to not only solve problems in nature, but identify them. This potential demonstrates that achieving intelligence more sophisticated than what we thought possible is now within arm's reach.

Adaptive Computation Time

Adaptive Computation Time (ACT), first introduced in [3], is a mechanism that allows Networks to dynamically learn how many repetitions to "ponder" its input before outputting the next state. ACT is achieved through Invariant Point Attention (IPA) and recycling.

IPA is a new type of attention that acts on a set of frames which are parameterised in the structure module as Euclidean transforms. **Recycling** is achieved through using the previous cycle's output as the new cycle's input.

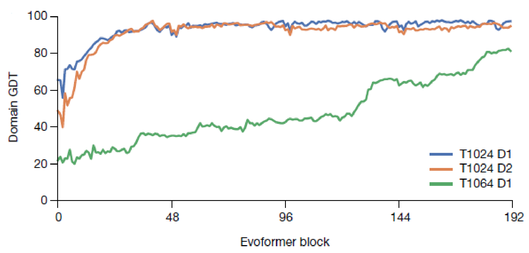

Figure 2b: interpreting ablation [1]

Although the ablation studies show neither recycling nor removing IPA independently affects accuracy greatly, the removal of both has a significant effect. Thus, it is reasonable to think this ablation causes a significant increase in the number of "wrong" models AF must compute.

Figure 2b supports the hypothesis that recycling during training is a variant of ACT that directs additional training resources to low-confidence segments of the structure.

Communication

The Evoformer has two avenues of communication.

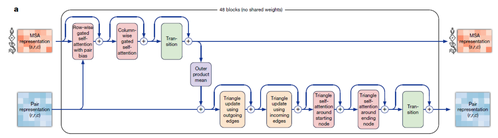

Figure 3: Evoformer structure [1]

def MSARowAttentionWithPairBias($\{\mathbf{m}_{si}\}, \{\mathbf{z}_{ij}\}, c = 32, N_{\text{head}} = 8$) :

 # Input projections

1: $\mathbf{m}_{si} \leftarrow \text{LayerNorm}(\mathbf{m}_{si})$

2: $\mathbf{q}_{si}^h, \mathbf{k}_{si}^h, \mathbf{v}_{si}^h = \text{LinearNoBias}(\mathbf{m}_{si})$ $\quad \mathbf{q}_{si}^h, \mathbf{k}_{si}^h, \mathbf{v}_{si}^h \in \mathbb{R}^c,\ h \in \{1, \ldots, N_{\text{head}}\}$

3: $b_{ij}^h = \text{LinearNoBias}(\text{LayerNorm}(\mathbf{z}_{ij}))$

4: $\mathbf{g}_{si}^h = \text{sigmoid}\left(\text{Linear}(\mathbf{m}_{si})\right)$ $\quad \mathbf{g}_{si}^h \in \mathbb{R}^c$

 # Attention

5: $a_{sij}^h = \text{softmax}_j\left(\frac{1}{\sqrt{c}}\, \mathbf{q}_{si}^h{}^\top \mathbf{k}_{sj}^h + b_{ij}^h\right)$

6: $\mathbf{o}_{si}^h = \mathbf{g}_{si}^h \odot \sum_j a_{sij}^h \mathbf{v}_{sj}^h$

 # Output projection

7: $\tilde{\mathbf{m}}_{si} = \text{Linear}\left(\text{concat}_h(\mathbf{o}_{si}^h)\right)$ $\quad \tilde{\mathbf{m}}_{si} \in \mathbb{R}^{c_m}$

8: **return** $\{\tilde{\mathbf{m}}_{si}\}$

Algorithm 1: Row attention with pair bias [1a]

Attention begins in line 5, where the pair bias is added to the standard argument ($\frac{\boldsymbol{q}_{si}^h \cdot \boldsymbol{k}_{sj}^h + b_{ij}^h}{\sqrt{c}}$) of the SoftMax function:

$$softmax(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

The bias term conveys prior hypotheses regarding the distance between residues i and j. Moreover, additive bias conveys a sense of Boolean logic. [4]

Gates

Gating is a concept that was developed to remedy Short-Term memory in RNN's. [5] These gates learn what information is relevant and thus should be remembered, and what should be discarded. This is achieved through the sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

which squeezes values between 0 and 1.

def TriangleAttentionStartingNode($\{\mathbf{z}_{ij}\}, c = 32, N_{\text{head}} = 4$) :

 # Input projections

1: $\mathbf{z}_{ij} \leftarrow \text{LayerNorm}(\mathbf{z}_{ij})$

2: $\mathbf{q}_{ij}^h, \mathbf{k}_{ij}^h, \mathbf{v}_{ij}^h = \text{LinearNoBias}(\mathbf{z}_{ij})$ $\quad \mathbf{q}_{ij}^h, \mathbf{k}_{ij}^h, \mathbf{v}_{ij}^h \in \mathbb{R}^c,\ h \in \{1, \ldots, N_{\text{head}}\}$

3: $b_{ij}^h = \text{LinearNoBias}(\mathbf{z}_{ij})$

4: $\mathbf{g}_{ij}^h = \text{sigmoid}\left(\text{Linear}(\mathbf{z}_{ij})\right)$ $\quad \mathbf{g}_{ij}^h \in \mathbb{R}^c$

 # Attention

5: $a_{ijk}^h = \text{softmax}_k\left(\frac{1}{\sqrt{c}}\, \mathbf{q}_{ij}^h{}^\top \mathbf{k}_{ik}^h + b_{jk}^h\right)$

6: $\mathbf{o}_{ij}^h = \mathbf{g}_{ij}^h \odot \sum_k a_{ijk}^h \mathbf{v}_{ik}^h$

 # Output projection

7: $\tilde{\mathbf{z}}_{ij} = \text{Linear}\left(\text{concat}_h(\mathbf{o}_{ij}^h)\right)$ $\quad \tilde{\mathbf{z}}_{ij} \in \mathbb{R}^{c_z}$

8: **return** $\{\tilde{\mathbf{z}}_{ij}\}$

Algorithm 2: triangular attention in the Evoformer [1a]

The element-wise multiplication by $\boldsymbol{g}_{\boldsymbol{di}}^h$ in algorithms 1 and 2 is strongly reminiscent of an output gate from LSTM's. The lack of forget and input gate implies that the authors do not necessarily want to sacrifice any prior information. .

def TriangleMultiplicationOutgoing($\{\mathbf{z}_{ij}\}, c = 128$) :

1: $\mathbf{z}_{ij} \leftarrow \text{LayerNorm}(\mathbf{z}_{ij})$

2: $\mathbf{a}_{ij}, \mathbf{b}_{ij} = \text{sigmoid}\left(\text{Linear}(\mathbf{z}_{ij})\right) \odot \text{Linear}(\mathbf{z}_{ij})$ $\quad \mathbf{a}_{ij}, \mathbf{b}_{ij} \in \mathbb{R}^c$

3: $\mathbf{g}_{ij} = \text{sigmoid}\left(\text{Linear}(\mathbf{z}_{ij})\right)$ $\quad \mathbf{g}_{ij} \in \mathbb{R}^{c_z}$

4: $\tilde{\mathbf{z}}_{ij} = \mathbf{g}_{ij} \odot \text{Linear}(\text{LayerNorm}(\sum_k \mathbf{a}_{ik} \odot \mathbf{b}_{jk}))$ $\quad \tilde{\mathbf{z}}_{ij} \in \mathbb{R}^{c_z}$

5: **return** $\{\tilde{\mathbf{z}}_{ij}\}$

Algorithm 3: outgoing multiplicative triangular updates [1a]

This layer utilises two gates: one in line and the other in line 3

The first gate is representative of [6]. This extracts information more efficiently due to the gradient advantages, although one difference between here and [6] is additional parameters.

The output gate performs a similar role to the attention output gate in regulating plausible structural predictions to ensure computation is not wasted.

Conclusion

The three features discussed contribute greatly to AlphaFold's success. However, they are also extremely general, and their interpretation links to reasoning quite naturally. Hence, these choices were made by DeepMind with the connection to thinking and reasoning in mind.

References

[1] Jumper, J., Evans, R., Pritzel, A. et al., 2021, 'Highly accurate protein structure prediction with AlphaFold', *Nature*, vol. 596, pp. 583-589

[1a] Supplementary Materials to 'Highly accurate protein structure prediction with AlphaFold'.

[2] AlQuraishi, M., 2019, 'End-to-End differentiable Learning of Protein Structure', *Cell Systems*, vol. 8, no. 4, pp. 292-301.

[3] Graves, A., 2016, 'Adaptive Computation Time for Recurrent Neural Networks', *CoRR*, 1603.08983.

[4] Boole, G., 1854, *An Investigation of the Laws of Thought*, Dover Publications, New York.

[5] Hochreiter, S. & Schmidhuber, J. 1997, 'Long Short-term Memory', *Neural Computation*, vol. 9, no. 8, pp. 1735-1780.

[6] Wu, Y., Zhang, S., Zhang, Y., et al., 2016, 'On Multiplicative Integration with Recurrent Neural Networks', *30th International Conference on Neural Information Processing Systems*, no. 9, pp 2864-2872.