Yunhao CHEN
yunhao.chen@polytechnique.edu

Question 1

Our objective is to make $\rho(\sum_{x \in X} \phi(x)) = \sum_{x \in X} x$, where the functions ρ, ϕ represent respectively the 2nd and 1st layer. Here, the 2nd layer is linear. Therefore, $\rho(\sum_{x \in X} \phi(x)) = \sum_{x \in X} \rho(\phi(x))$. Thus, the neural network could determine the sum of the inputs if the composition $\rho(\phi)$ is close to identity.

Question 2

Thanks to the non-linearity of $tanh$ in 1st layer, DeepSets model could easily embed the two sets into different vectors. For instance, we take weight matrix in 1st layer as [2,1] and no bias for simplicity, then the output of SUM operator for X_1 equals to $tanh(2.4 - 0.7) + tanh(-1.6 + 0.5) = tanh(1.7) + tanh(-1.1) \approx 0.135$; and the output of SUM operator for X_2 is $tanh(0.1) + tanh(0.5) \approx 0.562$. Then for the weight in 2nd layer, it is sufficient to take identity.

Question 3

First, we can employ a node embedding layer to embed the nodes in the graph. Then by taking a chosen aggregator function (for example the MEAN), we can obtain a vector representation of each graph. Now, we take previous DeepSets model to achieve the classification task (except we replace the last layer by a layer whose the number of output neurons equals to the number of classes, followed by a softmax activation function).

Question 4

The message passing layer satisfies permutation equivariant property. The readout function (SUM) also satisfies the permutation-invariance. One permutation inside the set does not change the aggregator behavior. The protein sequence ordering has been partially taken into account by the model (in the construction of graph, there would be an edge between two adjacent nodes in primary sequence). To better benefit from the sequential information, we can add a positional embedding, just like the transformer.