1 Question 1

The optimal parameters depend on the embedding that is considered. We give two examples below.

- Let us assume that the embedding of the digit k is a vector of dimension $h_1 = 10$, whose first k components are ones, and the other $d_1 - k$ are zeros.

  Set $W_1 = tI_{10}$, W_2 the 1×10 matrix filled with ones, and $b_1 = b_2 = 0$. Then:

  $f(X) = \sum_{i=1}^{m} \sum_{j=1}^{10} \tanh(t\mathbf{1}_{j \leq x_i}) \underset{t \to +\infty}{\longrightarrow} \sum_{i=1}^{m} \sum_{j=1}^{10} \mathbf{1}_{j \leq x_i} = \sum_{i=1}^{m} x_i$, using that $\tanh(x) \underset{x \to +\infty}{\longrightarrow} 1$.

  So for t large enough, the obtained DeepSets architecture works well.

- Let us assume that the embedding is the identity (so $h_1 = 1$). Then, for $t > 0$, set $W_1 = t$, $W_2 = 1/t$, $b_1 = b_2 = 0$.

  We then have $f(X) = \frac{1}{t} \sum_{i=1}^{m} \tanh(tx_i) \underset{t \to 0^+}{\sim} \frac{1}{t} \sum_{i=1}^{m} tx_i = \sum_{i=1}^{m} x_i$, using that $\tanh(x) \underset{x \to 0}{\sim} x$.

  So for $t > 0$ small enough, the obtained DeepSets architecture works well.

If we do not use embedding nor $\tanh$, we could simply have biases equal to 0, and weights equal to identity.

2 Question 2

Once again, this depends on the embedding that is used.
We see that when summing the two vectors in X_1, and summing the two vectors in X_2, we get the same results.
However, due to the embedding, and the non-linearity, we can easily have different results.

For instance, if the embedding is the identity, $W_1 = I_2$, $b_1 = 0$, then:

- For X_1, $\phi(x_1) = \tanh([1.2, -0.7]^T) \approx [0.83, -0.60]^T$, $\phi(x_2) = \tanh([-0.8, 0.5]^T) \approx [-0.66, 0.46]^T$, so $\phi(x_1) + \phi(x_2) \approx [0.17, -0.14]^T$.
- For X_2, $\phi(x_1) = \tanh([0.2, -0.3]^T) \approx [0.20, -0.29]^T$, $\phi(x_2) = \tanh([0.2, 0.1]^T) \approx [0.20, 0.10]^T$, so $\phi(x_1) + \phi(x_2) \approx [0.39, -0.19]^T$.

Then, we can simply use $W_2 = (1\ 1), b_2 = 0$, and then $f(X_1) \approx 0.03, f(X_2) \approx 0.20$.

3 Question 3

DeepSets takes sets as inputs, while a GNN takes graphs as inputs.
However, a set can simply be seen as a graph without edges.

Therefore, given a set S, we set the adjacency matrix $A = 0$, then $\tilde{A} = I_n$, and this can be fed to a GNN (with the feature of each node equal to the vector this node represents).
Because non-diagonal coefficients of $\tilde{A}$ are 0, the result of each node is only computed using itself (and not the other features), as is the case with DeepSets when applying ϕ.

Therefore, DeepSets architecture can indeed be seen as a submodule of GNN.

4 Question 4

In Erdős–Rényi random graphs with n nodes, the maximum number of edges is $\binom{n}{2} = \frac{n(n-1)}{2}$.
Each edge has a probability p of effectively being present in the graph, so the expected number of edges is $\frac{pn(n-1)}{2}$.

As for the variance, by independence it is the sum of the variances for each edge, which is $p(1-p)$, so the total variance is $\frac{p(1-p)n(n-1)}{2}$.

For $n = 15, p = 0.2$, we have an expected number of edges of 21, and a variance of 16.8.
For $n = 15, p = 0.4$, we have an expected number of edges of 42, and a variance of 25.2.