The output of the second adapter module 22 is added to Z_i, supplied by a residual connection 24, to generate the output X_i of the adapted transformer block 20.

Thus, the output X_i of an ADA transformer block in the *i*-th layer, is given by:

$$\begin{aligned} Z_i &= X_{i-1} + A_i^1 MSA(LN(X_{i-1})) + b_i^1, \\ X_i &= Z_i + A_i^2 MLP(LN(Z_i)) + b_i^2 \end{aligned} \qquad (2)$$

The design of the ADA transformer block 20 is a "micro" design feature. We also consider "macro" design features, namely which of the transformer block layers 12 of the foundation model shown in Fig. 1 are replaced by ADA transformer blocks 20 to form the adapted neural network. Three possibilities, i.e. three respective possible adapted neural networks, are illustrated in Fig. 3.

The adapted neural network of Fig. 3(a) is formed from the foundation model of Fig. 1 by replacing the first layer of the sequence of layers (i.e. the transformer block 12 which receives the output of the embedder layer 11) of the foundation model by an ADA transformation block 20. That is, two adapter modules 21, 22 corresponding respectively to the MSA function and MLP function, are added to the first transformer block 12, and configured to apply two linear transforms defined respectively by A_1^1 and b_1^1 and by A_1^2 and b_1^2 to the results of the corresponding function.

The adapted neural network of Fig. 3(a) is trained based on a database of training examples of the second computational task. Each training example comprises input data and corresponding output data, where the output data is the result of performing the second computational task on the corresponding input data.

The training procedure includes repeatedly presenting the input data of one of the training examples to the input layer of the adapted neural network model and modifying the adapter matrices A_1^1 and A_1^2, and the bias vectors b_1^1 and b_1^2, to make an output of the foundation neural network closer to the corresponding output data of the training example. This is typically performed by a backpropagation algorithm, and is typically performed using batches of the training examples, rather than individual training examples. Optionally, before the training the adapter matrices A_1^1 and A_1^2 may be identity matrices, and the bias vectors b_1^1 and b_1^2 may be zero, so that the untrained adapted neural network is equal to the foundation model.

Note that in this training procedure the sets of numerical parameters of the foundation neural network defining the MSA function and the MLP function are preserved (i.e. not changed). Thus, the distributions of output vectors obtained by the MSA function and MLP function are not changed, except that they are subject to scaling by scaling factors defined by the adapter matrices A_1^1 and A_1^2, and to an adaptation of their mean values according to the bias vectors b_1^1 and b_1^2. Thus, there is an