There are TWO models in total, corresponding to 2 datasets.

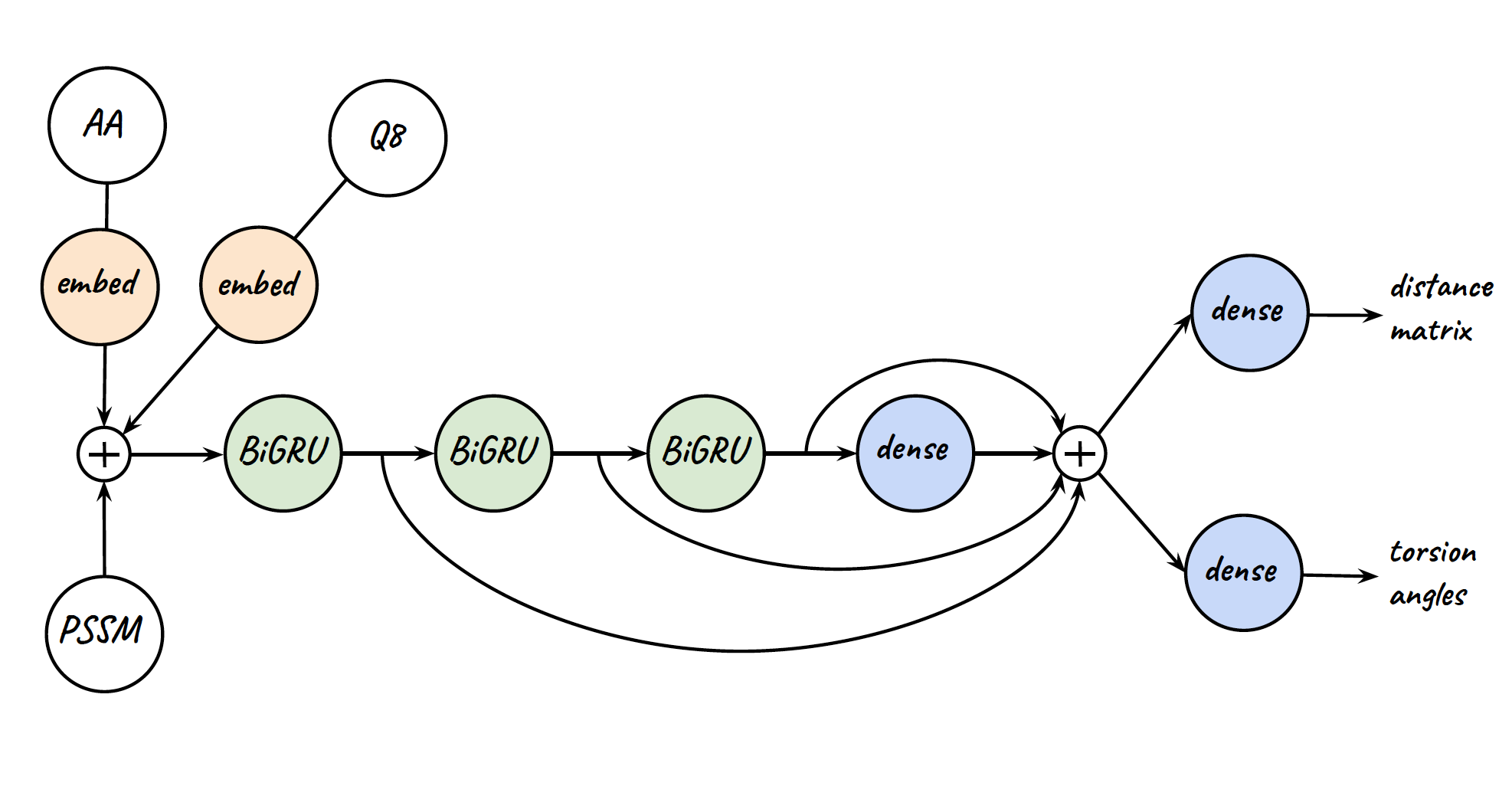
**model\_2\_1 (with 100k dataset as input)**

Preprocessing:

1. Remove proteins of extreme lengths (max=320, min=80)
2. Standard tokenization (AA and Q8/ss) and padding (AA, Q8, and PSSM).
3. Rescale training torsion angles (range from -180° to 180°) to [-1,1].

Model architecture (where Q8 = ‘ss’):

1. AA and Q8 sequences are embedded into 128-dimensinoal and 64-dimensional vectors, respectively, using the Keras Embedding Layer.
2. Concatenate the embedded AA and Q8 with PSSM features and feed to three BiGRU layers with 512, 256, and 128 units respectively and a time distributed (TD) dense layer (with ReLU activation) of output dimension 64. All intermediate outputs are concatenated as illustrated in the figure below.
3. Feed the concatenated intermediate output separately into
   1. a TD dense layer (without activation) of output dimension 3 for (coordinates →) distance matrix output.
   2. a time distributed dense layer (with Tanh activation) of output dimension 3 for torsion angles output.

****

(coordinates)

Training hyperparameters:

1. 30 epochs with batch size 32.
2. RMSProp with default hyperparameters (tf.keras.optimizers.RMSprop).
3. Initial 20 epoch learning rate = 0.001, subsequent epochs learning rate = 0.0005.

Files:

1. The training script is model\_2\_1.py
2. The prediction script is model\_2\_1\_pred.py
3. The output file is model\_2\_1.pkl

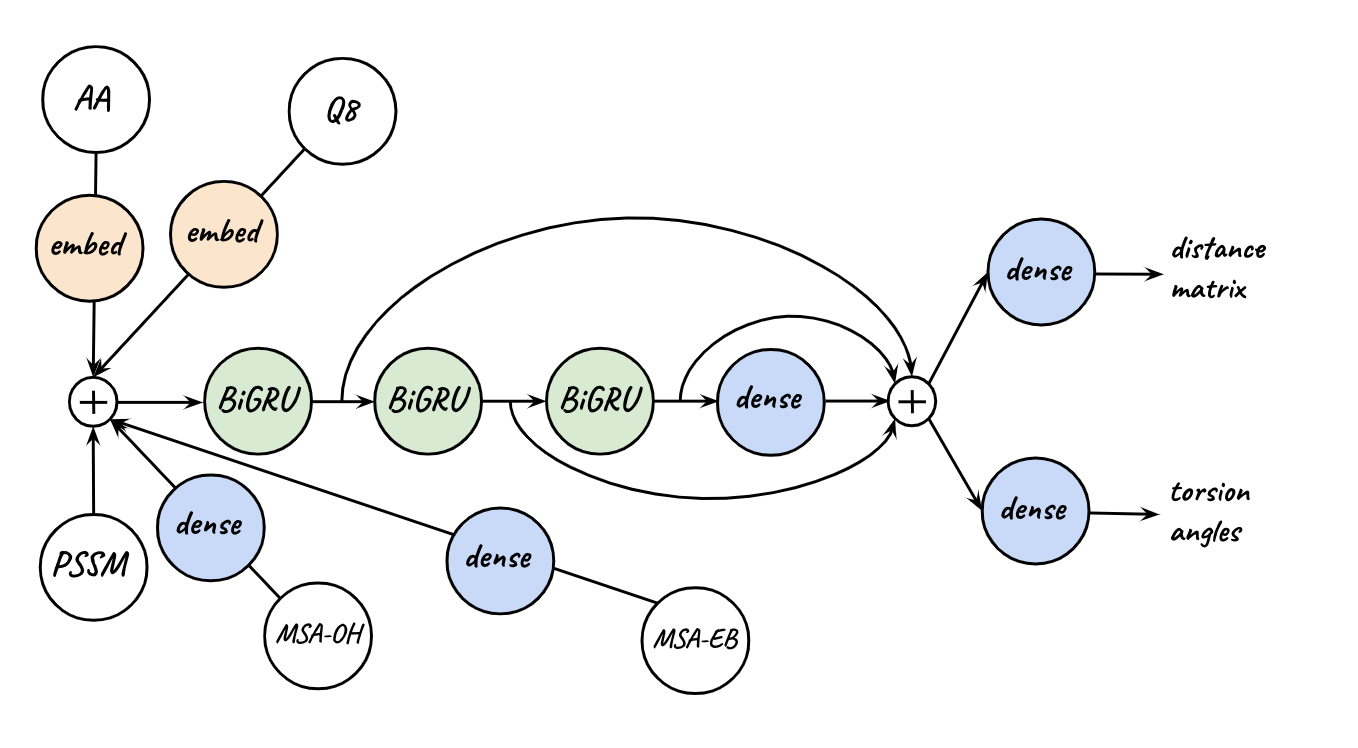
**model\_2\_2 (with the “cuprotein” dataset as input)**

Preprocessing:

1. Remove proteins of extreme lengths (max=320, min=80)
2. Standard tokenization (AA and Q8/ss) and padding (AA, Q8, and PSSM, MSA-OH, MSA-EB).
3. Rescale MSA one-hot and embedded feature values to [0,1].
4. Rescale training torsion angle values (range from -180° to 180°) to [-1,1].

Model architecture (identical to model\_2\_1 except input data)

1. AA and Q8 sequences are embedded into 128-dimensinoal and 64-dimensional vectors, respectively, using the Keras Embedding Layer.
2. Feed One-hot MSA features (MSA-OH) and embedded MSA features (MSA-EB) into
3. Concatenate everything and feed to three BiGRU layers with 512, 256, and 128 units respectively and a time distributed (TD) dense layer (with ReLU activation) of output dimension 64. All intermediate outputs are concatenated as illustrated in the figure below.
4. Feed the concatenated intermediate output separately into
   1. a TD dense layer (without activation) of output dimension 3 for (coordinates →) distance matrix output.
   2. a TD dense layer (with Tanh activation) of output dimension 3 for torsion angles output.



(coordinates)

Training hyperparameters:

1. 50 epochs with batch size 32.
2. RMSProp with default hyperparameters (tf.keras.optimizers.RMSprop).
3. Initial 20 epoch learning rate = 0.001, subsequent epochs learning rate = 0.0005.

Files

1. The training script is model\_2\_2.py
2. The prediction script is model\_2\_2\_pred.py
3. The output file is model\_2\_2.pkl