1. \*\*Environment Configuration\*\*: Install the required dependencies based on the `requirements.txt` file to ensure that the running environment matches the project requirements.

\*\*Code Structure\*\*:

- `data/`: Scripts for data processing and dataset-related files.

- `deepspeech\_pytorch/`: Model code, including baseline models, comparison models, and ablation models.

- `configs/`: Configuration files.

2. \*\*Data Processing\*\*:

After configuring the environment, run the `an4.py` script in the `data` folder to process the AN4 dataset. The command to run is as follows:

```bash

cd data/ && python an4.py && cd ..

```

3. \*\*Model Training\*\*:

Run the following command to start the training:

```bash

python train.py +configs=an4

```

4. \*\*Model Validation\*\*:

The `deepspeech\_pytorch` folder contains code for the baseline model and ablation models. To validate a specific model, rename the model file to `model.py`, then run `train.py` to train the model.

5. \*\*Best Model\*\*:

`Model\_best.py` is the best-performing model after tuning and serves as the reference baseline. The design of the ablation models is described below:

- \*\*Model\_gai.py (model1)\*\*: This model reduces the network structure of `model\_best.py` to demonstrate that `model\_best.py` achieves the best performance. It removes the fusion of BiLSTM features and retains only two layers of `DoubleConv` features. The input to the JPU is limited to the features extracted by `DoubleConv`, and the adjustment layer for BiLSTM features is removed, leaving only the adjustment layer for `DoubleConv` features. A DualAttention module is added after the JPU layer to enhance feature extraction by combining position and channel attention.

- \*\*Model\_gai1.py (model2)\*\*: The network structure is modified to a U-Net style. The stride of the two `DoubleConv` layers is changed to 2, adding two downsampling operations. In the JPU module, features extracted from different scales of `DoubleConv` and BiLSTM features are fused. The upsampling stage uses two `UP` modules to restore feature dimensions: first by bilinear interpolation, then using `DoubleConv` layers to adjust the features.

- \*\*Model\_gai2.py (model3)\*\*: This model is a reduced version of `model\_gai.py`. Due to overfitting in `model\_gai.py`, the network size is reduced by removing the final DualAttention structure, leaving only two `DoubleConv` layers and the JPU module. The JPU module still fuses features from the two `DoubleConv` modules. In this version of the JPU module, the adjustment layers for BiLSTM features and the two feature decoupling layers (dilation) in the JPU module are removed. This aims to validate whether a smaller network structure impacts performance.

6. Additionally, to view the training graphs, input the following command in the terminal:

```bash

tensorboard --logdir=path\_to\_saved\_logs

```

where `path\_to\_saved\_logs` is the path containing the logs from the training session.