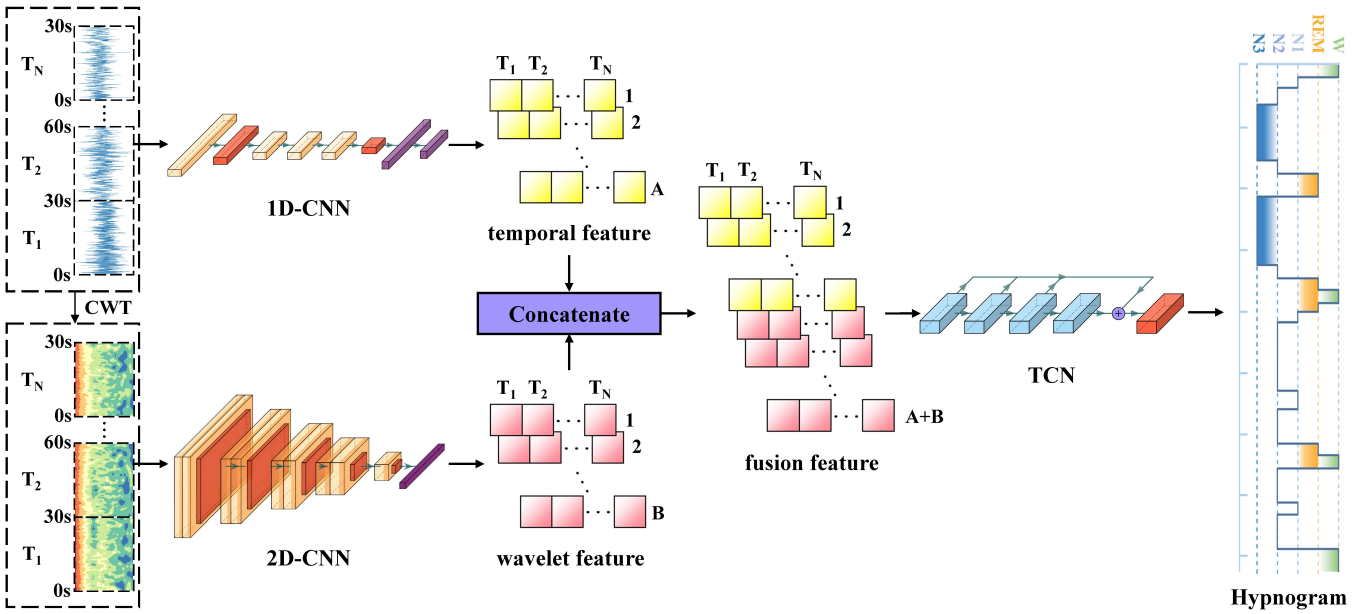
FFTCN

This project provids a straightforward and promising method for improving the accuracy of automatic sleep staging using only single-channel EEG by applying a hybrid model with a 1D-CNN, a 2D-CNN and a TCN.

This project is a re-release of the original project from the Gitee repository on GitHub. The original project is available at https://gitee.com/abmoon/waveletnet/tree/master, and the original contributor of this project is https://gitee.com/abmoon?utm_source=poper_profile.

Introduction

We proposed a novel deep learning algorithm named feature fusion temporal convolutional network (FFTCN) for automatic sleep staging using single-channel EEG data. This algorithm employed a one-dimensional convolutional neural network (1D-CNN) to extract temporal features from raw EEG, and a two-dimensional CNN (2D-CNN) to extract time-frequency features from spectrograms generated through continuous wavelet transform (CWT) at the epoch level. These features were subsequently fused and further fed into a temporal convolutional network (TCN) to classify sleep stages at the sequence level. Moreover, a two-step training strategy was used to enhance the model's performance on an imbalanced dataset. Our proposed method exhibits superior performance in the 5-class classification task for healthy subjects, as evaluated on the SHHS-1, Sleep-EDF-153, and ISRUC-S1 datasets. This work provided a straightforward and promising method for improving the accuracy of automatic sleep staging using only single-channel EEG, and the proposed method exhibited great potential for future applications in professional sleep monitoring, which could effectively alleviate the workload of sleep technicians. The architecture of FFTCN is illustrated in the figure as follows.



1.Preprocess EEG data

`FFTCN/data/preprocessor.py`

Firstly, the S3 and S4 stages were merged into N3 to achieve compatibility with AASM criteria. Secondly, only a continuous Wake epoch of 30 minutes was retained at the beginning and end in each data to reduce redundancy and discard the MOVEMENT and UNKNOWN stages. Additionally, the EEG signals were down sampled to 100 Hz to reduce computational complexity.

2.Extraction and Compression of Wavelet Maps

`FFTCN/data/wavelet_torch.py`

`cwt` performs CWT to transfom 30-s EEG segment to time-frequency image.

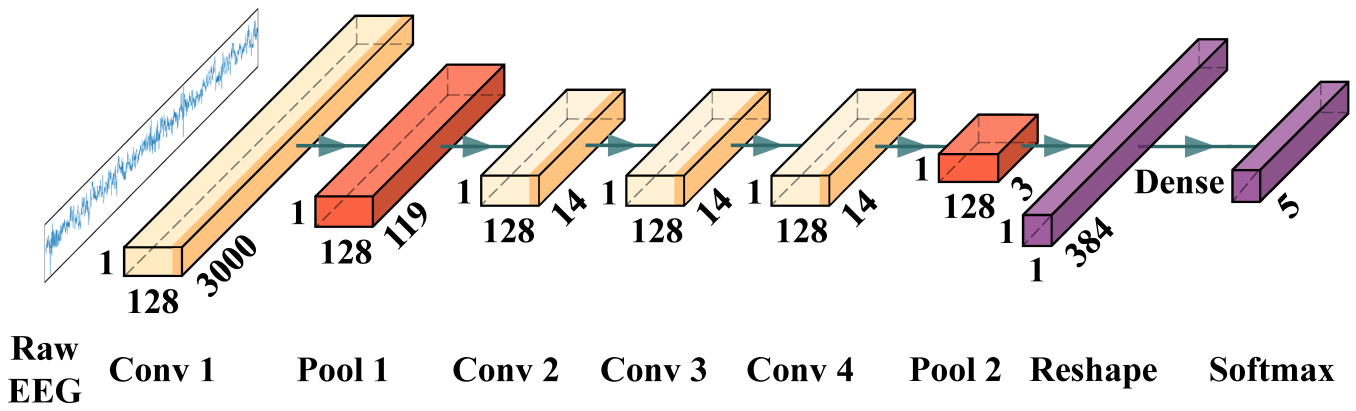
Convolutional network is not capable to deal with 3000*30 long-strip image, so call `compress` to copress it on the time-scale.

3.1D-CNN

`FFTCN/models/raw/network.py`

In this study, a 1D-CNN structure was designed to extract the temporal feature from the raw EEG signal at the epoch level. As depicted in Figure as follows, the network primarily consisted of 4 convolutional blocks and 2 max-pooling blocks. Each convolutional block comprised a

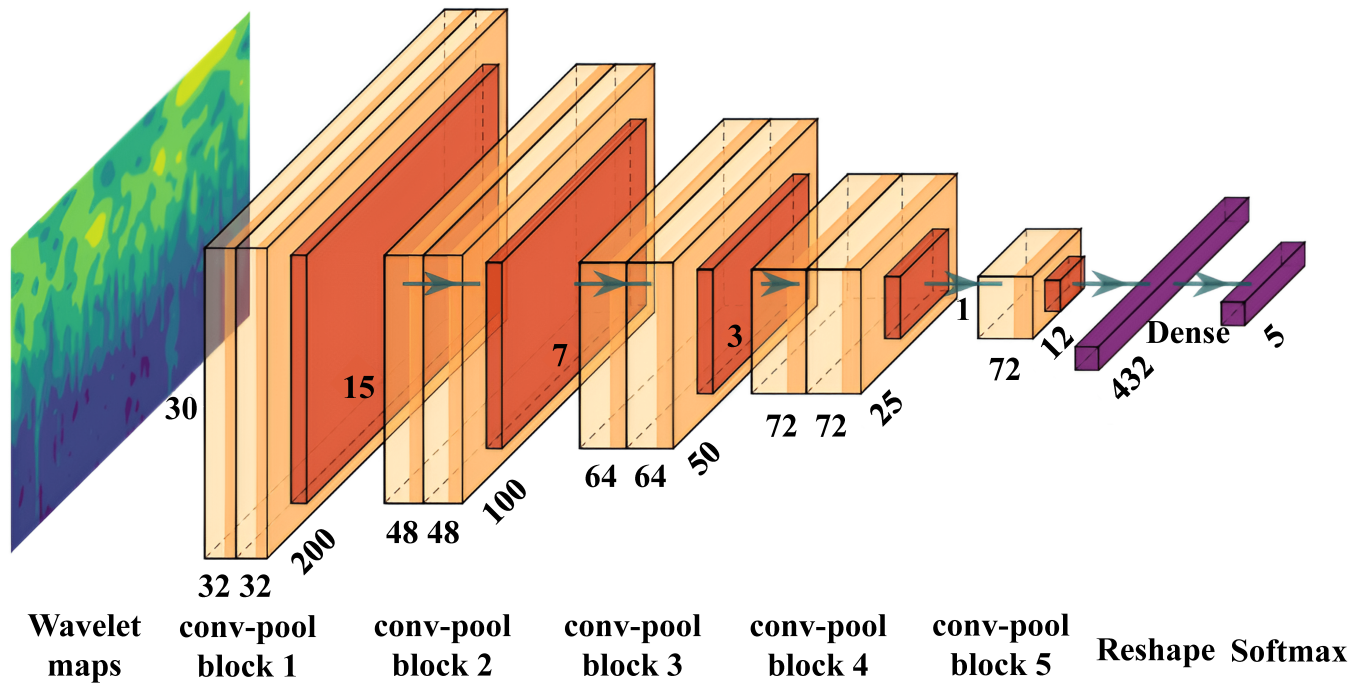
convolutional layer, a batch normalization (BN) layer, and applied Rectified Linear Unit (ReLU) activation.



4.2D-CNN

`FFTCN/models/wavelet/network.py`

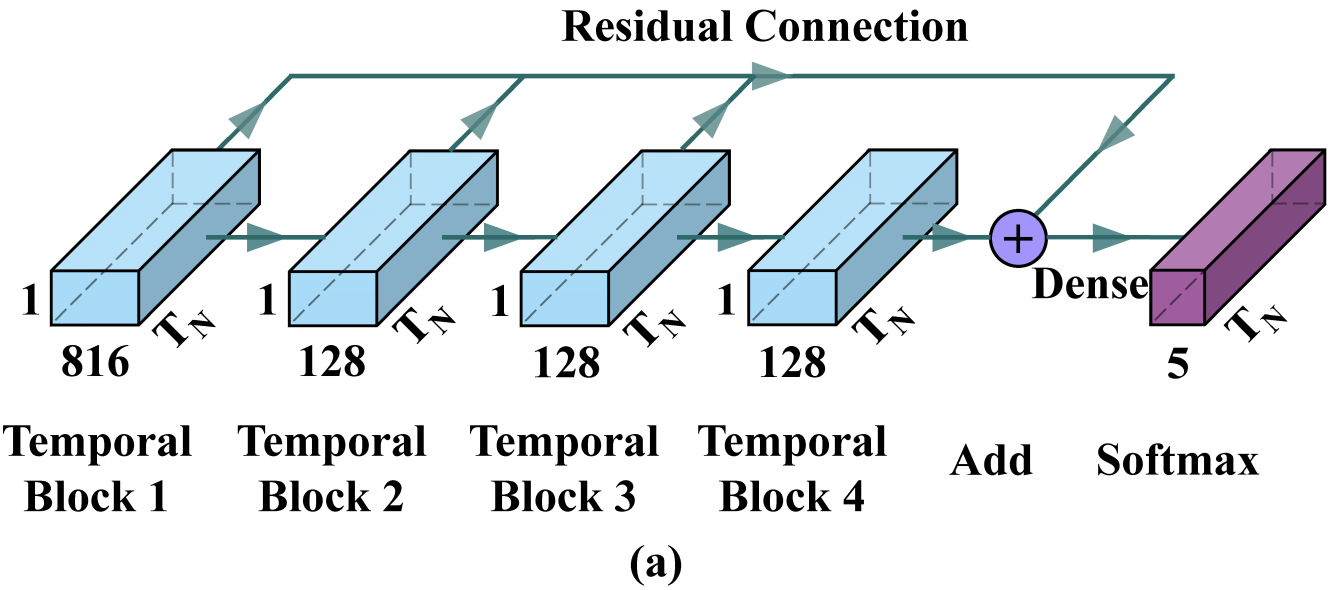
We designed a 2D-CNN structure to extract time-frequency features from the wavelet maps at the epoch level. The architecture is illustrated in Figure as follows. The network consisted of 4 convolutional pooling (conv-pool) blocks, each of which included 2 convolution operations and 1 pooling operation, and an additional conv-pool block was performed to further reduce the feature dimension.



5.TCN

`FFTCN/models/base/tcn.py`

Our TCN model consisted of four Temporal Blocks. Each block was comprised of 2 dilated convolution operations and a convolutional layer for residual connections.



(a)

`FFTCN/models/merge/network.py`

Combine 1D-CNN, 2D-CNN and TCN to obtain FFTCN.

6.Dataloader, Train and Validate

We provide a dataloader for [https://sleepdata.org/datasets/shhshttps://sleepdata.org/datasets/shhs](SHHS dataset).

Run `FFTCN/models/raw/trainer.py` to pretrain 1D-CNN, and run `FFTCN/models/wavelet/trainer.py` to pretrain 2D-CNN. Finally, run `FFTCN/models/merge/trainer.py` to finetune FFTCN. The results of validation and test set will be saved in the same directory.