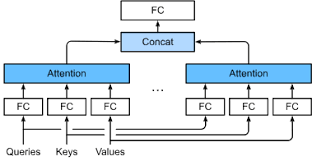
the most discriminative ones. This is achieved through a residual squeeze and excitation block, which improves the representation capability of the features and enhances classification performance.

**4. Temporal Context Encoder (TCE):**

The TCE leverages a multi-head attention mechanism with causal convolutions to capture temporal dependencies in the extracted features. This module is crucial for modeling the sequential nature of sleep stages and ensuring that the temporal context of the EEG data is adequately represented in the model. The causal convolutions ensure that the attention mechanism respects the temporal order of the data.

**5. Multi-Head Attention Mechanism:**

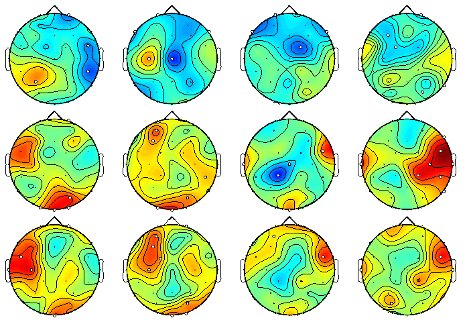
The multi-head attention mechanism is a key component of the TCE as mentioned diagrammatically in Figure 3.2. It allows the model to focus on different parts of the input sequence simultaneously, capturing a variety of temporal dependencies. By using multiple heads, the model can learn to attend to various aspects of the data, improving its ability to classify different sleep stages accurately.

**Figure 3.2: Multi – Head Attention Mechanism**

**6. Class-Aware Loss Function:**

To address the issue of data imbalance in sleep stage classification, the study introduced a class-aware loss function. This loss function assigns different weights to different classes, ensuring that the model pays more attention to underrepresented classes. This approach helps in improving the classification performance for all sleep stages, especially the less frequent ones.

**7. Public Datasets:**

The study utilized three public datasets for training and evaluating the proposed model. These datasets provided a diverse range of EEG signals, which were essential for testing the generalizability and robustness of the model. The use of public datasets also allows for reproducibility and comparison with other studies in the field as demonstrated in Figure 3.3.

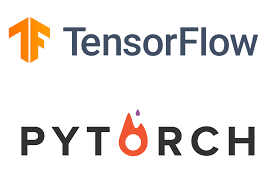
**Figure 3.3: EEG Dataset**

**8. Experimental Framework:**

An extensive experimental framework was implemented to evaluate the proposed AttnSleep model. This included a variety of evaluation metrics to assess the model's performance, such as accuracy, precision, recall, and F1 score. The experimental setup ensured a thorough analysis of the model's strengths and weaknesses, providing insights into its real-world applicability.

**9. Software and Libraries:**

The implementation of the AttnSleep model was carried out using various software tools and libraries. Deep learning frameworks like TensorFlow or PyTorch were likely used for building and training the neural network models. Additionally, libraries for data preprocessing, visualization, and statistical analysis played a crucial role in the research process.



**Figure 3.4: TensorFlow and PyTorch**