Advanced EEGNet Model Design - Explanation & Walkthrough

1. Overview

This document explains the design and training flow of the Advanced EEGNet model used for EEG classification across 80 classes. The model builds on convolutional and attention-based neural components, coupled with Focal Balanced Loss, SWA, and other robust training strategies.

2. FocalBalancedLoss

The FocalBalancedLoss is a modified cross-entropy loss that handles class imbalance and hard-to-classify examples. It adjusts the standard loss using:

- `alpha` to reduce the weight of easy examples.
- `gamma` to focus learning on harder misclassified examples.
- Class weights are derived from the label distribution in the training set to balance learning across all classes.

3. Model Architecture - AdvancedEEGNet

The model is composed of 3 main blocks and a classifier:

- Block 1: Captures frequency-based features using depthwise convolution, followed by GELU activation and dropout.
- Block 2: Applies a convolutional attention mechanism to enhance class-relevant channels and suppress noise.
- Block 3: Aggregates temporal-spatial features using adaptive pooling and projects to high-level feature space.
- Classifier: Three fully connected layers refine features and output logits for 80 classes, with dropout to reduce overfitting.

All convolutions use batch normalization and modern activation (GELU) for stability.

4. AttentionBlock

A lightweight channel attention module:

- Learns a dynamic attention mask across the spatial-temporal feature maps.
- Combines attention-weighted inputs with a residual convolution layer.
- Helps focus the network on salient brain regions and time points.

Advanced EEGNet Model Design - Explanation & Walkthrough

5. Training Process

The training loop includes:

- Mixed precision training using `torch.amp` to save memory and improve speed.
- Gradient clipping (`clip_grad_norm_`) to stabilize learning.
- AdamW optimizer with CosineAnnealingWarmRestarts scheduler.
- Scaler for dynamic loss scaling.

Model checkpoints are saved whenever test accuracy improves.

6. Stochastic Weight Averaging (SWA)

SWA maintains an averaged version of the model weights starting from epoch 80.

- `AveragedModel` tracks moving averages.
- `SWALR` uses a small constant LR to fine-tune.
- At the end, batch norm stats are recomputed using training data.

SWA improves generalization and prevents convergence to sharp local minima.

7. Evaluation

The model is evaluated after each epoch on the test set.

Final model accuracy is reported based on the SWA model, giving a robust estimate of generalization performance.

Accuracy and loss are printed after each epoch for monitoring training dynamics.

8. Final Notes

The model is robust for fine-grained EEG classification and includes components like:

- GELU activations for nonlinearity
- Weight normalization for stable gradients
- Focal loss for class imbalance
- SWA for ensemble-like generalization

This makes the setup ideal for complex, real-world EEG tasks.