Supplementary File 2. Description of transformer architecture

1. **Input Projection Layer**

* **Purpose:** Projects segments of the original input into a higher-dimensional space (hidden dimension).
* **Details:**
* The original input dimension (1376) must be divisible by the predefined sub-embedding sequence length (8), so it can be reshaped into a sequence of feature vectors.
* The reshaped input tensor has the shape: (batch size, sub-embedding sequence length, input dimension // sub-embedding sequence length).
* A linear layer maps this into a hidden dimension size:



1. **Transformer Encoder**

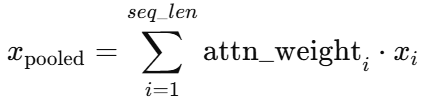
* **Purpose:** Learns contextual relationships between sequential input segments.
* **Details:**
* Built using PyTorch's nn.TransformerEncoder.
* Each encoder layer uses:
* Multi-head self-attention (8 attention heads, each one for one sub-embedding sequence),
* Feedforward network with hidden dimension \*2,
* GELU activation,
* Dropout (dropout probability, 0.3),
* Layer normalization.
* Operates on inputs of shape (batch size, sub-embedding sequence length, hidden dimension) with batch first.

1. **Attention Pooling Mechanism**

* **Purpose:** Dynamically aggregates encoded sequence information into a single vector using learnable attention weights.
* **Details:**
* A small neural network computes attention scores for each sequence position:



* The scores are normalized using softmax across the sequence dimension.
* A weighted sum of the sequence outputs produces a fixed-size context vector:



1. **Classification Head**

* **Purpose:** Maps the pooled vector to the final output logits.
* **Details:**
* Consists of:
* A linear layer reducing dimensionality to hidden dimension // 2,
* ReLU activation,
* Dropout (0.3),
* A final linear layer producing category logits.

1. **Focal Loss Function**

* **Purpose:** Designed to tackle class imbalance by focusing more on hard-to-classify examples.
* **Details:**
* Parameters:
* α (default: 0.75): Weighting factor for positive samples,
* Γ (default: 2): Focusing parameter to reduce the loss contribution from easy examples,
* reduction: 'mean'.
* Computation:

, where:

图示

AI 生成的内容可能不正确。, ,

