周报

本周的工作包括：

1. 阅读论文
2. 学习注意力机制模型

阅读《SSAST: Self-Supervised Audio Spectrogram Transformer》

背景：基于`self-Attention`机制的神经网络，例如视觉转换器，在自然语言处理、视觉等领域都表现出了优于CNNs（卷积神经网络）的性能。因此研究界将该方法应用到音频领域，设计了Audio Spectrogram Transformer（AST），发现也有很好的性能。

问题：然而，直接将`self-Attention`应用到AST中，与CNN相比，存在以下问题：

- 需要更多的训练数据；
- 性能依赖于基于大量标签数据的有监督预训练；
- 复杂的训练过程。

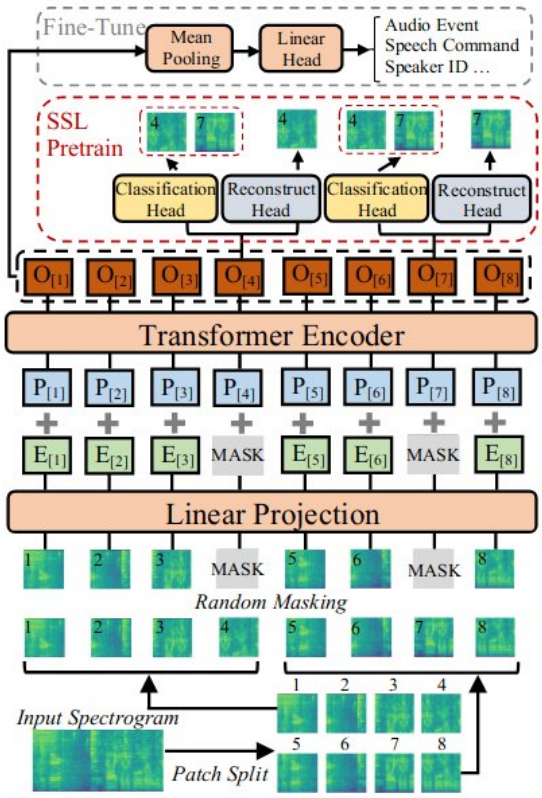
这篇文章的方案：通过使用无标签数据进行自监督学习以减少对标签数据的需求，进一步提出了一种使用联合判别和生成掩蔽频谱图patch的方法来预训练AST模型。

这篇文章的创新点有下面两点：

1. 提出MSPM模型——一种基于联合判别和生成自监督学习的掩蔽频谱图patch的方法。这种模型与CNN相比，它在训练上可以并行执行。
2. 与其他基于`self-Attention`的音频任务相比，这篇文章同时使用了音频和语音数据来训练，这比单独用上述其中一种数据集来进行预训练所取得的性能更好。

方案的具体构建流程图如下。

1. 首先将一段`t`秒的音频进行线性变换，转换到特征空间中的特征数据，在音频处理中对应为频谱图，这个也是AST模型的输入。
2. 然后将频谱图按照一定的patch分块，对应图中1～8个patch。
3. 使用随即掩蔽遮住某两个patch，然后将剩下的patch使用Linear Projection作为embedding layer，的得到E。
4. 由于AST模型使用的是这段音频的全局信息，但是patch本身不带有它在原始音频中的顺序信息，因此需要加上位置信息，对应P。
5. 将E+P的结果输入Encoder层进行训练，得到步骤1中的线性变换矩阵、步骤3和步骤5中的变换矩阵。
6. 将输出进行Decoder，最后得到任务的输出。
7. 下游分析将原始数据与训练得到的变换矩阵进行运算，便可得到任务结果。

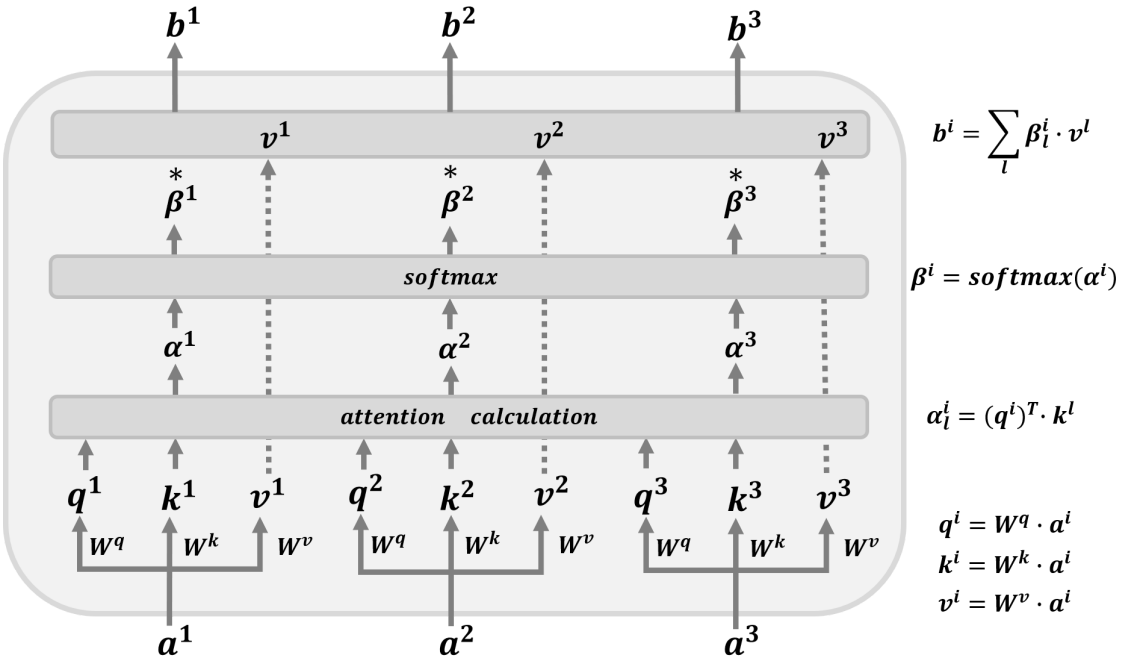


其中，上述模型的核心是Transformer，而Transformer的核心是注意力机制。

注意力机制

Transformer中的核心机制就是`self-Attention`。`self-Attention`机制的本质来自于人类视觉注意力机制。当人视觉在感知东西时候往往会更加关注某个场景中显著性的物体，为了合理利用有限的视觉信息处理资源，人需要选择视觉区域中的特定部分，然后集中关注它。注意力机制主要目的就是对输入进行注意力权重的分配，即决定需要关注输入的哪部分，并对其分配有限的信息处理资源给重要的部分。

`self-Attention`原理



`self-Attention`工作原理如上图。给定输入word embedding向量$a^1, a^2, a^3 \in R^{d_l \times 3}$，然后对于输入向量$a^i \in \{1,2,3\}$通过矩阵$W^q \in R^{d_l \times d_t}$，$W^k \in R^{d_l \times d_t}$，$W^v \in R^{d_l \times d_t}$进行线性变换得到$Query$向量$q^i \in R^{d_k \times 1}$，$Key$向量$k^i \in R^{d_k \times 1}$，以及$Value$向量$v^i \in R^{d_k \times 1}$，即

$$\begin{cases} q^i = W^q \cdot a^i \\ k^i = W^q \cdot a^k, i \in \{1,2,3\} \\ v^i = W^q \cdot a^v \end{cases}$$

如果令矩阵$A = (a^1, a^2, a^3) \in R^{d_l \times 3}$，$Q = (q^1, q^2, q^3) \in R^{d_k \times 3}$，$K = (k^1, k^2, k^3) \in R^{d_k \times 3}$，$V = (v^1, v^2, v^3) \in R^{d_l \times 3}$，则此时有

$$\begin{cases} Q = W^q \cdot A \\ K = W^q \cdot A \\ V = W^q \cdot A \end{cases}$$

接着再利用得到的$Query$向量和Key向量计算注意力得分，采用的注意力计算公式为点积缩放公式

$$\alpha_l^i = \frac{(q^i)^T \cdot k^l}{\sqrt{d^k}} = \frac{\sqrt{d^k}}{d^k} \sum_{n=1}^{d^k} k_n^l \cdot q_n^i \quad i,l \in \{1,2,3\}$$

假设Key向量$k^l = (k_1^l, k_2^l, k_3^l)$的元素和$Query$向量$q^i = (q_1^i, q_2^i, q_3^i)$的元素独立同分布，且令均值为0，方差为1。令注意力分数矩阵$\Lambda = (\alpha^1, \alpha^2, \alpha^3) \in R^{3 \times 3}$，则有

$$\Lambda = \frac{K^T \cdot Q}{\sqrt{d^k}}$$

然后对注意力分数向量经过`softmax`层进行归一化，得到归一化后的注意力分布 β^i ，即

$$\beta_j^i = \frac{e^{\alpha_j^i}}{\sum_{n=1}^3 e^{\alpha_n^i}} \quad i,j \in \{1,2,3\}$$

最后利用得到的注意力分数向量β^i和$Value$矩阵V获得最后的输出$b^i \in R^{d_l \times 1}$，则有

$$b^i = \sum_{l=1}^3 \beta_l^i \cdot v^l \quad i \in \{1,2,3\}$$

令输出矩阵$B = (b^1, b^2, b^3) \in R^{d_l \times 3}$，则有

$$B = Attention(Q,K,V) = V \cdot softmax(\frac{K^T \cdot Q}{\sqrt{d^k}})$$

以上就是`self-Attention`的原理和计算流程。