1.模型训练网络的基本实现与创新性介绍

XXXXXXXXXXXXXX

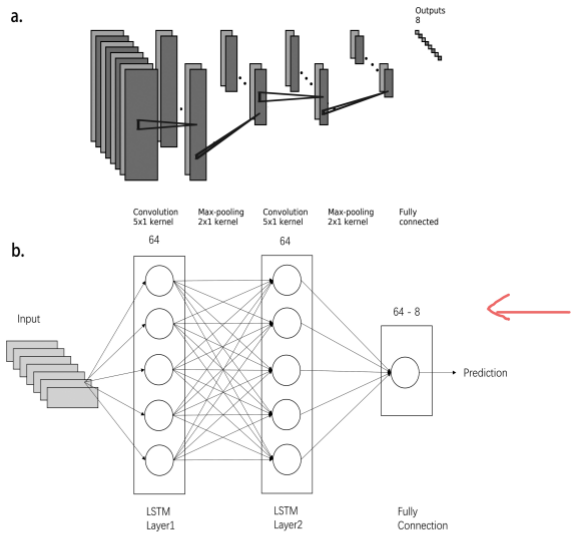


Figure 1. Deep learning model architecture. (a) First, we designed a CNN that consists of three layers. The first two layers are used to extract the features of the Raman data, and each layer is a combination of a convolution layer and a pooling layer. The last layer is a fully connected neural network layer for classification. The convolution layers contain 16 and 32 neurons, and the convolution filter of each layer is 1 × 5. Then, after being activated by the ReLU function, the network enters the maximum pooling layer and pools by 1 × 2. Finally, the eight-dimensional data are output through the fully connected layer. The cross-entropy loss function is used to calculate the loss value, the backpropagation algorithm is used to adjust the model parameters, and the Adam optimizer is selected for optimization. (b) RNN model uses the LSTM method and consists of three layers with 64 neurons in each layer, yielding a final output of eight dimensions through the fully connected layer. Tanh is used as the activation function, and the cross entropy loss function and Adam optimizer are used to calculate the loss values and perform backpropagation optimization, respectively. The other construction and training processes are the same as those in the CNN training. The Raman signal data of each kind of bacteria include 1200 data points, which are used as convolution layer input of the CNN model and LSTM layer input of the long-term and short-term memory model. Both models have two layers, but each layer of the CNN model includes a convolution layer and a pooling layer, while each layer of the LSTM model includes three gating units. Then, the model initializes the parameters of each layer randomly. Through the calculation of the model, the CNN and LSTM models obtain eight output values. Then, the eight values of the model are compared with the real species tag to obtain the error, and the parameters of the whole model are updated by the back propagation algorithm. Finally, through multiple rounds of training, the error between the eight output values of the CNN and LSTM models and the real species tag is minimized.

AM-CNN（基于注意力改进的卷积神经网络）模型是一种用于处理细菌拉曼图谱数据的新型深度学习算法。该模型在输入数据特征组合时，考虑了细菌拉曼图谱的波长向量和强度向量，通过滑动窗口方式获取目标词与周围词的综合向量。首先，通过第一次的注意力机制捕获实体与序列中每个词的相关性，并将其与输入的综合词向量矩阵相乘。接着，对卷积结果使用第二次注意力机制捕获视窗与关系的相关性。最终，将卷积结果与相关性矩阵相乘，得到最后的输出结果。

（这个模型的核心在于将细菌拉曼图谱的波长向量和强度向量与输入数据进行组合。首先，将这两种向量进行拼接，构成了最初的输入向量。接着，使用滑动窗口的方式将目标词与周围词组合在一起，形成综合向量。第一次的注意力机制应用在实体与序列中每个词的相关性。将相关性矩阵与输入的综合词向量矩阵相乘，得到一个二维矩阵。然后，使用卷积提取特征，并对卷积结果使用第二次注意力机制捕获视窗与关系的相关性。最后，将卷积结果与相关性矩阵相乘，得到最终的输出结果。通过这种方式，模型能够充分考虑细菌拉曼图谱的波长向量和强度向量在输入数据中的关联关系。）

网络构建：（需要修改）

1. 输入层：将细菌拉曼图谱的波长向量和强度向量作为输入数据。波长向量和强度向量可以分别作为两个输入通道。
2. 注意力机制1：使用注意力机制1捕获输入数据中实体与序列中每个词的相关性。可以采用自注意力（self-attention）机制或全局平均池化（global average pooling）等方式。
3. 综合词向量矩阵：将注意力机制1得到的相关性矩阵与输入的综合词向量矩阵相乘，得到一个二维矩阵，用于提取特征。
4. 卷积层：使用卷积层对综合词向量矩阵进行特征提取，可以使用不同的卷积核大小和数量，以捕获不同尺度的特征。
5. 注意力机制2：使用注意力机制2对卷积结果进行进一步的特征选择，捕获视窗与关系的相关性。
6. 全连接层：将经过注意力机制2的卷积结果展平，并通过全连接层进行特征融合和映射，得到最终的输出。
7. 输出层：根据任务需求，可以添加合适的输出层，如softmax层用于分类任务，sigmoid层用于二分类任务等。

8. 损失函数：选择合适的损失函数用于模型的训练和优化。

2.训练数据的预处理准备以及训练流程的介绍

（数据的分组->训练集、验证集、测试集的分配比例：0.8：0.1：0.1->接入神经网络进行epoches=400的预训练->调整训练窗口、学习率等超参数提高模型的泛化性与鲁棒性->.....

同时为了保证模型的稳定性，我们设定初始学习率为0.0001以及设定Dropout层值为0.1

(描述好数据集的大小、特征数量、标签数量、样本分布信息)

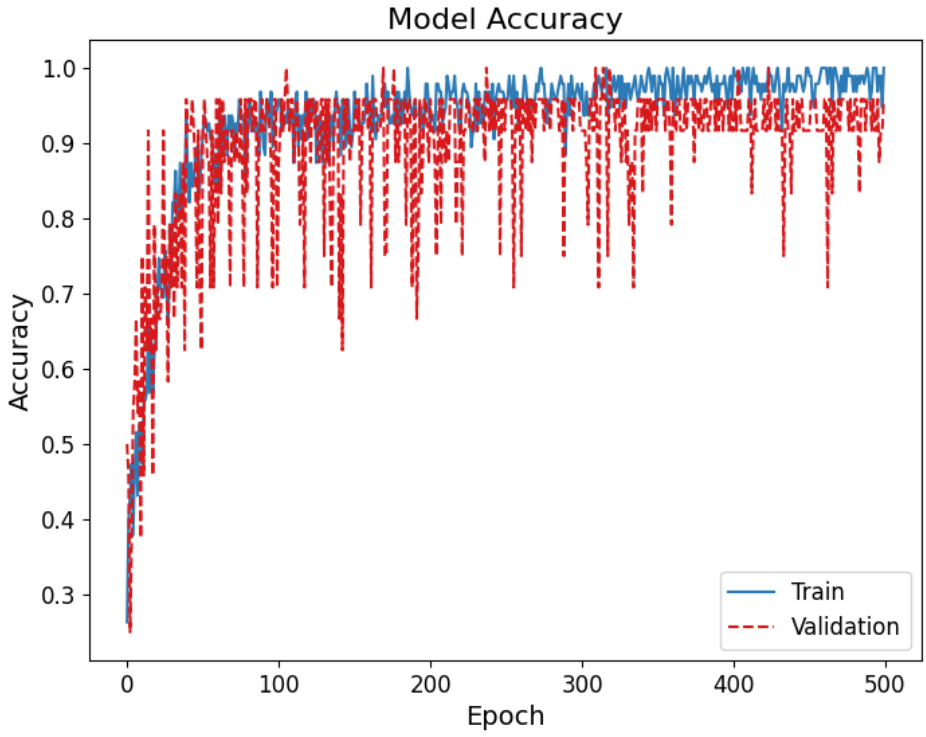
我们在训练网络时，为了使得模型可以更快更准确的训练，加入了学习率的自适应调整函数，可以根据训练的数据情况以及已有的训练量来自动调整学习率，使训练效果达到最优。

具体模型构架如下：

1. 我们首先将训练数据集按照4：1划分成训练集与验证集。
2. 构建AM-CNN网络框架
3. 将训练数据输入AM-CNN网络进行1000轮训练
4. 待模型训练好后，使用测试数据测试模型预测结果
5. 调整模型参数，待模型结构最优后，测试模型最终的分类准确度，并记录训练期间 Loss 值的变动情况。

3.训练结果展示+结果解读

- **准确率图**:



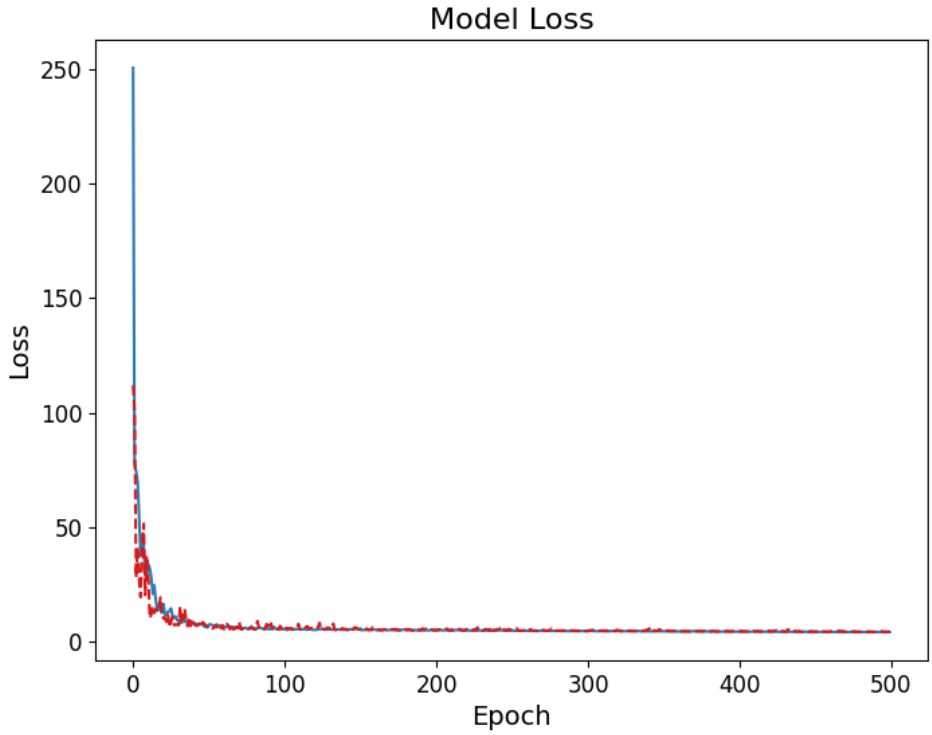
我们模型训练的情况可以通过训练加验证准确率图来对于模型的好坏进行综合评估，训练准确率代表模型在当前训练数据上的表现。训练多轮后，训练准确率会逐渐提高，这表明模型学到了更多的数据分类特征。但是，如果训练准确率开始变得非常高，而验证准确率却不再提高，这说明模型开始过拟合训练数据。

通过不断优化代码的网络结构，将训练的准确率情况效果达到最优

- 在第100次训练后模型的**训练准确率**维持在90%以上
- 在第334次训练后模型的**验证准确率**基本维持在90%以上

这说明AM-CNN机制对于细菌的拉曼光谱数据拟合效果非常理想，与此同时，我们在选取模型的优化器时选取了Adagrad自适应学习率优化器，它可以根据梯度的历史信息自适应地调整学习率，可以对于整体模型提供更好的性能和泛化能力。

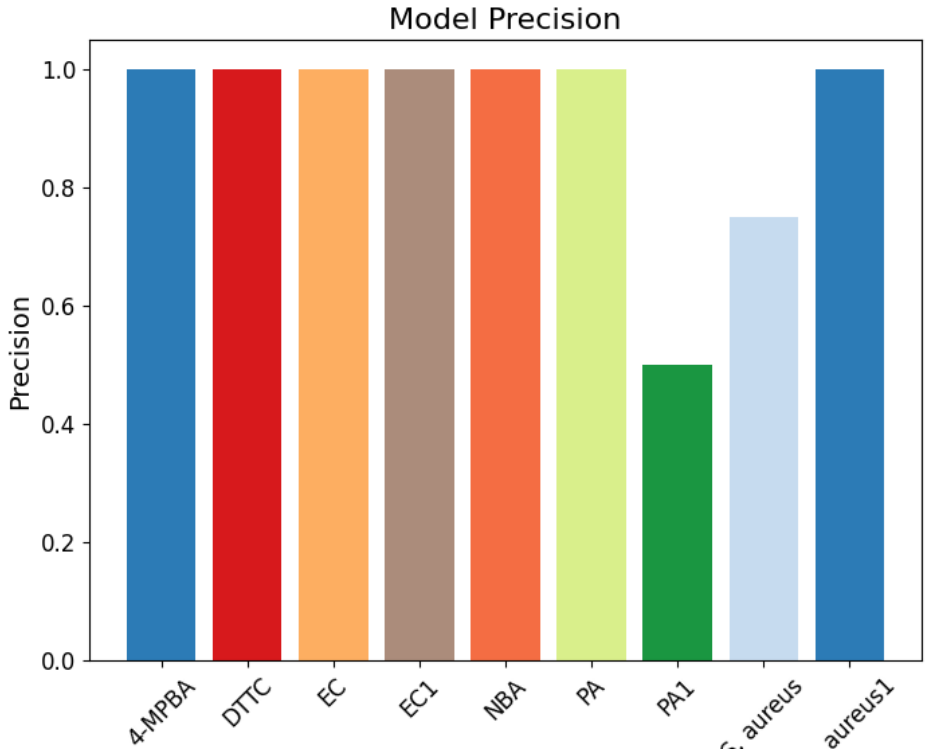
- **LOSS图**



通过比较该模型训练过程中的验证损失率与训练损失率情况，我们可以看出：

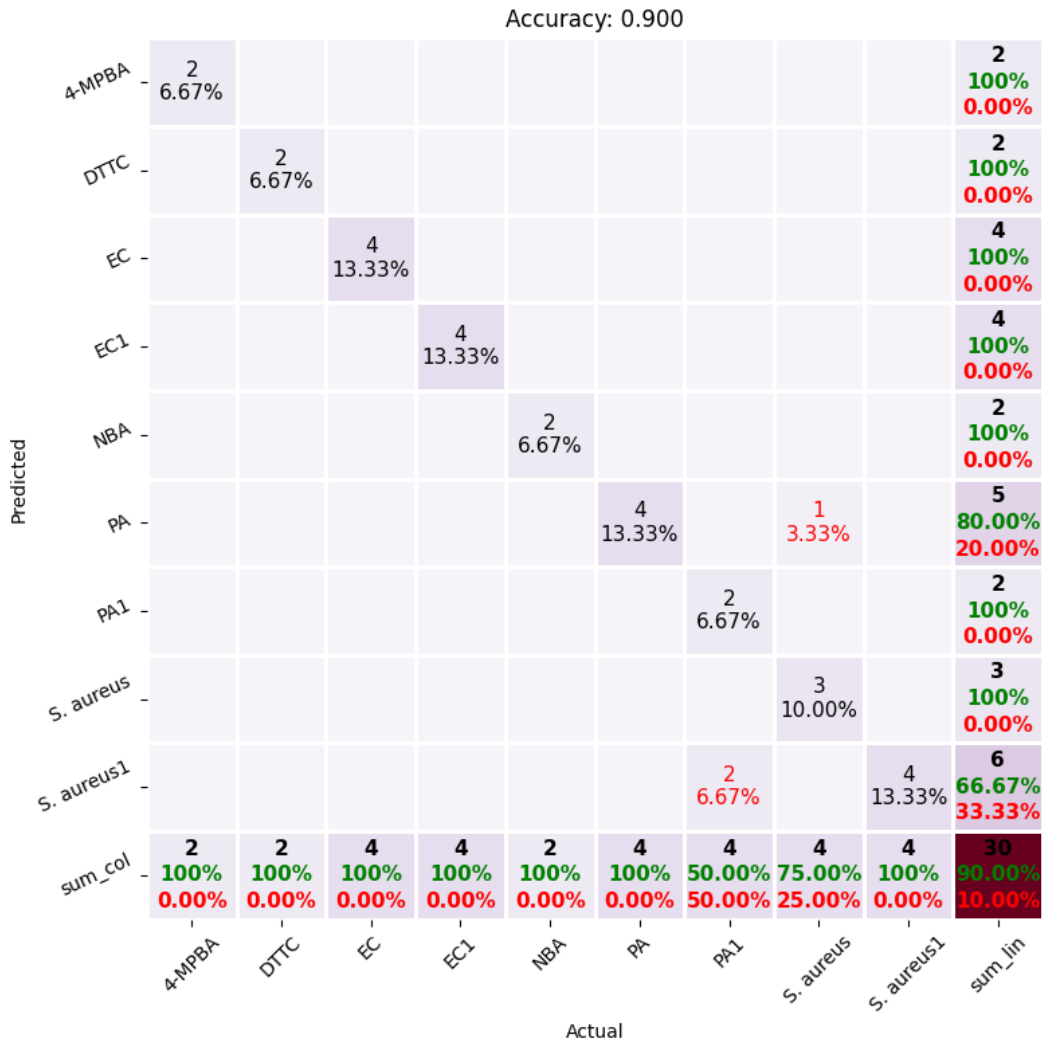
1. **模型的泛化性能**：模型在训练集和验证集上都表现良好，能够很好地泛化到新的数据。这可以提高模型在实际应用中的性能和鲁棒性。
2. **减少模型的调试和优化时间**：如果模型的训练损失和验证损失相差很大，那么我们就需要花费更多的时间来调试和优化模型。相反，当训练损失和验证损失相差很小时，我们可以更快地确定最佳的超参数和模型架构，从而减少调试和优化的时间。因此该模型框架对于后期调试与优化来说非常方便。
3. **提供更加可靠的实验结果**：训练损失和验证损失相差很小时，模型在验证集上的表现和在训练集上的表现非常接近，这可以提高实验结果的可靠性和可重复性。

- **模型分别对于n（9）种细菌数据具体分类情况与召回情况**



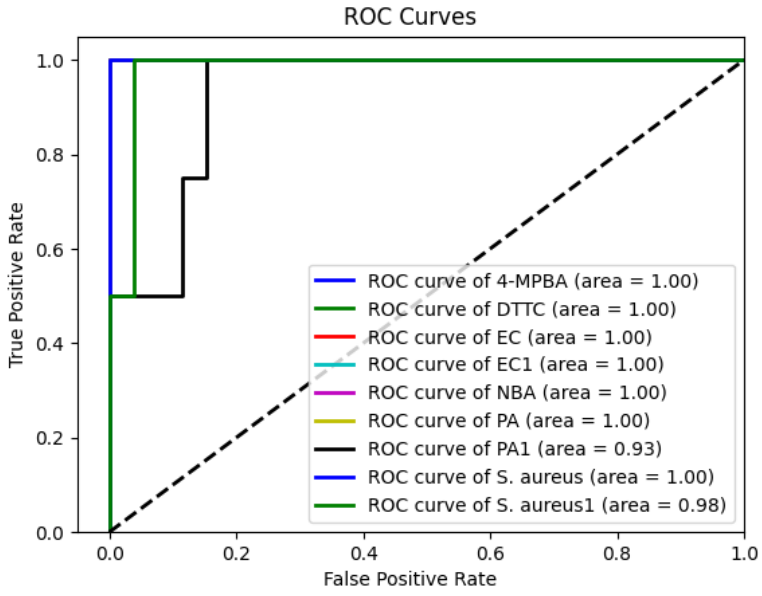
模型在分类除PA1和S. aureus细菌以外的其他细菌时，表现出色，整体准确率保持在100%。然而，对于PA1细菌，模型的训练准确率仅为50%，而对于S. aureus细菌，训练准确率略高，约为78%。

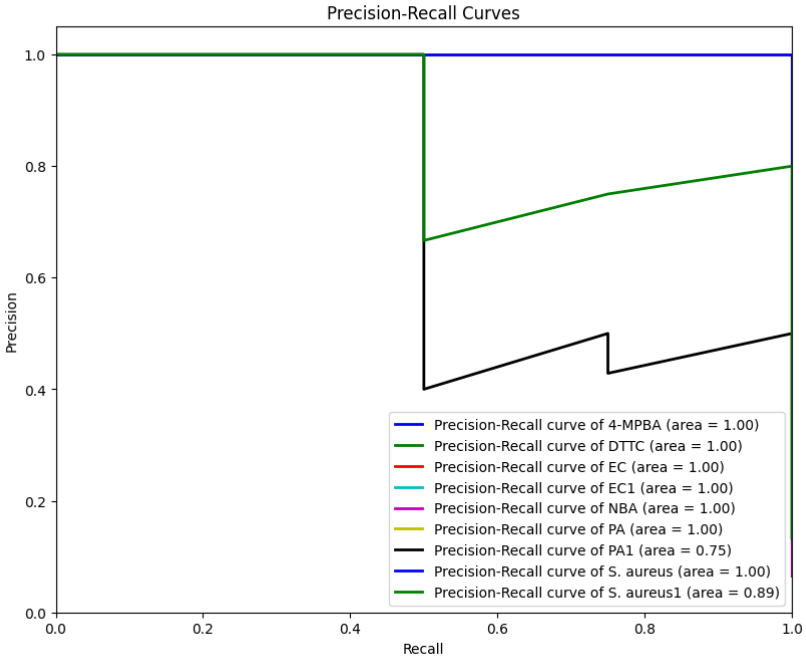
- **模型对于测试集的验证情况**



从图中可以看出，完成训练的注意力改进的卷积神经网络模型在测试表现中，对 7 种菌 实现 100%的分类结果，对其余2种菌均有1株左右的错误结果，卷积神经网络 模型最终的分类准确度达到90%。

- **ROC曲线**





通过查阅资料可知：

PR曲线观察方法

PR曲线是一种直观且有效的工具，用于评估分类模型在不同阈值下的性能，特别是在数据不平衡的情况下。通过分析PR曲线，我们可以更好地理解模型的精度和召回率之间的权衡，从而为实际应用中的决策提供依据。

1. 模型性能的整体评估：计算PR曲线下的面积（称为平均精度，AP），我们可以得到一个总体衡量模型性能的指标。AP值越高，表示模型的性能越好。精度和召回率通常是相互影响的，很难同时达到很高的精度和召回率。
2. 精度和召回率的权衡：通过观察PR曲线，我们可以了解在不同阈值下精度和召回率之间的权衡情况，以便根据实际应用场景选择合适的阈值。

ROC曲线观察方法

ROC曲线用于评估分类模型在不同阈值下的性能。通过分析ROC曲线，我们可以更好地了解模型的预测能力和鲁棒性

1. 模型性能的整体评估：通过计算ROC曲线下的面积（称为AUC，Area Under the Curve），我们可以得到一个总体衡量模型性能的指标。AUC值越接近1，表示模型的性能越好。而AUC值为0.5，表示模型性能等同于随机猜测。
2. 阈值选择：通过观察ROC曲线，我们可以根据实际应用场景和需求，选择合适的阈值以达到最佳的预测性能。例如，在某些场景中，我们可能希望最大化召回率（TPR）而不太关心假正例率（FPR），此时我们可以通过ROC曲线选择合适的阈值。

分析

为了综合评估我们训练出的模型的稳定性与泛化性，我们利用ROC曲线以及PR曲线交叉分析我们训练的模型的稳定性。

我们首先通过ROC曲线可以看出PA1与S. aureus1细菌对应ROC值分别为0.93与0.98，其余种类的细菌值均为1。说明上文建立的分类模型在这九种细菌的识别和分类上表现非常出色。ROC值为1表示模型能够完美地区分正类和负类样本，即模型对所有正类样本的预测概率高于负类样本的预测概率。

接着我们通过PR曲线来交叉验证来评估模型在新数据上的泛化性能，从图中我们不难看出：该模型对于绝大多数训练细菌的PR值维持在1，意味着分类器在各种阈值下的Precision和Recall都非常高，表现出极佳的分类性能。这意味着分类器在预测正例时的准确性很高，同时也能找到几乎所有的实际正例。

而对于PA1与S. aureus1细菌的AUC（Area Under Curve）分别为0.75和0.89，这表明S. aureus1的分类器性能整体优于PA1。在Recall值为0.51之前，两者的Precision均维持在1，说明