

分类号\_\_\_\_\_密级\_\_\_\_\_  
UDC \_\_\_\_\_



# 本 科 毕 业 论 文

## 基于 Transformer 的地震方位预测

学生姓名 李云帆 学号 19130051014

指导教师 于紫凝

院、系、中心 电子工程学院

专业年级 2019 级电子信息工程

论文答辩日期 2023 年 5 月 15 日

中 国 海 洋 大 学

# 基于 Transformer 的地震方位预测

完成日期: \_\_\_\_\_

指导教师签字: \_\_\_\_\_

答辩小组成员签字: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

# 基于 Transformer 的地震方位预测

## 摘 要

在各种公共紧急事件中，地震最为我们熟知。与一般的灾害和事故相比，地震灾害的表现通常具有全域性、复杂性、突发性、选择性等诸多特点。目前对于地震方位的预测也存在多种方法，其中一种就是利用高精度的钻孔应变仪来对震前应力数据的微观变化进行异常检测与提取，并结合宏观的科学考察与研究分析震源与地下应力场的关系。本文有别于传统的研究方法，针对应力数据的时变性、多维性，从深度学习的角度出发，基于 Transformer 模型中的词编码与位置编码机制、多头注意力机制以及编码器结构，将其改进从而实现对多维时序数据时间和空间信息的捕捉。然后配合不同的模型训练方法，从而得到最优的 Transformer 地震方位预测模型。最后利用 python 的图形开发界面 tkinter 库设计地震方位预测的 UI 界面实现地震方位预测的可视化。

**关键词：**地震方位预测；多维时序应力数据；Transformer 分类预测；tkinter 可视化

---

# Earthquake azimuth prediction based on Transformer

## Abstract

Among various public emergencies, earthquakes are the most familiar to us. Compared with general disasters and accidents, the performance of earthquake disasters usually has many characteristics such as globality, complexity, suddenness, and selectivity. At present, there are many methods for the prediction of earthquake azimuth, one of which is to use high-precision borehole strain gauges to detect and extract the anomaly of microscopic changes in pre-earthquake stress data, and combine macroscopic scientific investigation and research to analyze the source and Subsurface stress field relations. This paper is different from the traditional research method, aiming at the time-varying and multi-dimensional nature of stress data, from the perspective of deep learning, based on the word encoding and position encoding mechanism, multi-head attention mechanism and encoder structure in the Transformer model, it is improved In this way, the time and space information of multi-dimensional time series data can be captured. Then cooperate with different model training methods to obtain the optimal Transformer earthquake azimuth prediction model. Finally, the UI interface of earthquake azimuth prediction is designed by using the tkinter library of python's graphical development interface to realize the visualization of earthquake azimuth prediction.

**Keywords:** Earthquake azimuth prediction; Multivariate time series stress data; Transformer multi-category prediction; Tkinter visualization

## 目 录

摘 要 .....	III
ABSTRACT .....	II
目 录 .....	III
1 绪论 .....	5
1.1 选题背景及研究意义 .....	5
1.2 国内外研究现状 .....	6
1.3 本文的主要工作 .....	8
2 Transformer 分类预测模型设计 .....	10
2.1 Transformer 基本结构 .....	10
2.2 Transformer 结构改进 .....	12
2.3 分类评价指标 .....	14
2.4 本章小结 .....	15
3 基于 Transformer 的地震方位预测分析 .....	16
3.1 数据预处理 .....	16
3.1.1 钻孔应变数据简介 .....	16
3.1.2 时间戳处理 .....	18
3.1.3 极端值处理 .....	19
3.1.4 噪声处理 .....	19
3.2 地震区域划分 .....	20
3.2.1 地震事件的空间位置分析 .....	20
3.2.2 基于 Kmeans 聚类的地震空间区域分类 .....	20
3.3 地震方位预测结果分析 .....	23
3.3.1 基于机器学习分类器的二分类验证 .....	23
3.3.2 基于 Transformer 的二分类验证 .....	25
3.3.3 Transformer 地震方位预测模型的训练超参数调试 .....	29
3.3.4 Transformer 地震方位预测对比验证 .....	40
3.4 本章小结 .....	42
4 地震方位预测 UI 界面设计 .....	43
4.1 编程库选择 .....	43
4.2 UI 界面的功能设计与实现 .....	43
4.3 地震定位 UI 界面应用 .....	46
4.4 本章小结 .....	48
5 总结与展望 .....	48
5.1 研究成果总结 .....	49
5.2 不足与展望 .....	50

参考文献 .....	51
致谢 .....	53

# 1 绪论

## 1.1 选题背景及研究意义

在各种公共紧急事件中，地震最为我们熟知。与一般的灾害和事故相比，地震灾害的表现通常具有全域性、复杂性、突发性、选择性等诸多特点，因此也是国际上公认的重大灾害之一。地震发生的过程中常常伴随着建筑物、生命线工程等基础设施的直接破坏，并进一步引发包括火灾、爆炸、危险物质泄露等次生灾害，以上种种加重了地震的破坏程度并引起更大的人员伤亡和财产损失，使人们产生恐慌心理，严重的干扰了正常的社会秩序以及影响了当地的经济的发展。

为了降低灾害所带来的损失，我国一直以来也致力于地震预测领域方面的研究，在近近年来对一次次过往地震进行研究，总结经验归纳方法。近 50 年来，我国地震学者在地震预测领域所取得的进展着实令人兴奋，成功预测了发生在我国的多次地震<sup>[1]</sup>，如海城地震，为我国的防震减灾事业做出巨大贡献，激励并推动了我国防震减灾事业的发展。这些年来，我国也总结出了许多方法经验，其中一种就是通过利用高精度的仪器进行微观异常检测并结合宏观的科学考察研究从而达到预测地震的目的。而这其中，往往离不开对于地震前应力数据的观测研究。

应力，即地球内部的受力，它是由地球内部的形变所产生的，它所包含来自地球内部不同角度的信息，描述了地球内部的状态<sup>[2]</sup>。由于我国位于世界上最主要的两大地震带之间，即太平洋-亚欧板块边界和亚欧-印度洋板块边界，而板块频繁的运动也会使得地球内部的应力、应变场时刻变化。不仅如此，作为施加于地球外部的力的一种，人类的活动如核实验也会对应力产生变化，因此通过应变仪等仪器所观测到的地球应力数据，对于地震研究预测有着举足轻重的作用。

对于地震预测，离不开对地震发生的时间、地点、强度等数据的观测。而对于地震强度的研究，也离不开对地震前后发震地点周围的地壳应力数据的分析。我国的陆上地震高发区具有明显的东西分区性，大致以贺兰山-龙门山-横断山为界，与我国一二级地势阶梯的分界线东段高度重合。在界线两边，中国东西部地震高风险区的成因却很不相同，对于人类生命财产的威胁也不一样。西部地震高风险区连片性强，主要位于塔里木陆块南北两侧、青藏高原地区所在的诸多造山系和对接带内。而东部地震高风险区面积总体较小，相对孤立，位于华北陆块内部。由于地处地势分界线，周围如四川、云南、甘肃、宁夏等地也不可避免地经历不同类型的数次地震。为了解决该地区地震方位预测的问题，就需要捕捉动态的地壳应力数据，分析应力数据与地震方位之间的非线性关系，即利用时序数据来进行地震方位分类预测。

目前利用时序数据进行分类预测的算法大致可分为机器学习和深度学习两种。在机器学习算法中，常用树模型<sup>[3]</sup>来对时序数据进行预测。该方法建立方式和人的

思维方式很相近, 非常便于理解。同时对于数据要求较低, 可以处理连续数据、离散数据以及缺失值等, 不需要预先对输入数据进行正则化、归一化等处理。但是它的缺点是对训练数据比较敏感, 容易受到噪声影响, 对于数据之间的时序关系处理能力不佳, 因此机器学习算法并不适用。目前, 主流的深度学习中, 利用时序进行分类预测的模型包括卷积神经网络(CNN)、循环神经网络(RNN)、长短期记忆神经网络(LSTM)、Transformer 等, 本文则使用的是当下广泛运用于自然语言处理任务(NLP)的 Transformer 模型进行方位预测。

本文利用贺兰山-龙门山-横断山分界线周围八个台站(甘肃张掖高台观测站、宁夏银川观测站, 宁夏海原小山观测站, 甘肃临夏观测站, 四川康定姑咱镇观测站, 云南昭通观测站, 云南丽江永胜观测站, 云南腾冲观测站)自 2010 至 2017 年的时序数据, 通过大量实验训练出当前最优的 Transformer 模型, 来实现预测地震方位的目的。为识别震前异常、预测地震方位提供较为科学的定量估计, 有助于帮助人们更好地了解如何进行地震方位预测, 也为地震方位预测的领域提供一定的研究基础。

## 1.2 国内外研究现状

地震预测一直以来都是地震学中非常重要的问题, 许多学者也已经提出过各种方法技术来解决这个问题。现有的地震预测, 从方向上看大致可以分为四类, 即数学分析、地震前兆信号、机器学习以及深度学习。第一类方法使用数学工具来解释地震预测问题, 例如 FDL 方法<sup>[4]</sup>(Fibonacci 斐波那契数、对偶数、卢卡斯数)。第二类方法常研究地震前兆信号以帮助进行地震预测。例如电磁信号<sup>[5]</sup>、气溶胶光学深度(AOD)<sup>[6]</sup>、岩石圈-大气电离层<sup>[7]</sup>以及动物的异常行为<sup>[8]</sup>。第三类方法主要利用机器学习的预测方式, 常用的有支持向量机(SVM)、k 邻近(kNN)以及多目标信息模糊网络 (M-IFN)<sup>[9]</sup>。第四种则是用深度学习的神经网络, 目前针对地震预测的方向, 主要的有卷积神经网络、递归神经网络、长短期记忆神经网络及 Transformer 等。Dario Jozinovic 等人<sup>[10]</sup>研究了一种基于深度卷积神经网络的技术来预测地震前后地面震动强度的变化趋势。他们选取地震目录中震级大于 3.0 的事件, 使用距地震时间前 10 秒的三分量地震波形数据输入模型。结果表明深度卷积网络能够准确预测附近站点的震动强度, 可以提供在地震发生后 15-20 秒内的地面运动估计。Yu Huang 等人<sup>[11]</sup>将递归神经网络应用于地震前边坡动态响应的预测。他们引入了移动步长策略并建立了三个递归神经网络模型: RNN、LSTM 和 GRU 模型。同时他们训练了多层感知器预测模型进行比较验证。结果表明, 在数据量不大的情况下, 递归神经网络在分析斜坡的地震动力响应方面表现良好, 并提供比多层感知器网络更好的预测, 适用于地震载荷动态响应的时间序列预测。Qianlong Wang 等人<sup>[12]</sup>设计了一种能够处理二维输入的长短期记忆网络来学习不同地点地震之间的时空关系, 并利用该关系进行预测。结果表明, 具有二维输入的 LSTM 网络能



够发现和利用地震之间的时空相关性，从而比以前做出更好的预测。Jannes Münchmeyer 等人<sup>[13]</sup>使用基于注意力机制的 Transformer 网络进行地震震级和位置的估计，他们使用来自一组站点动态变化的波形数据，结果表明 Transformer 在具有足够训练数据的区域估计非常精确，同时在预估大震级事件上效果良好。

因为本文研究的数据为应力时间序列，而上述不同的模型算法在时间序列的处理上也各有特点。从吴国栋等人研究中<sup>[14]</sup>可以发现 CNN 可以通过定长滤波器提取输入序列局部信息，但其没办法捕捉长时数据特征。从陈聪等人研究中<sup>[15]</sup>可以发现 RNN 具有记忆历史信息的特点，并且适用于处理任意输入、输出的长度，较适合处理时序信息。但是 RNN 在处理长时数据时经过多轮梯度迭代会发生梯度消失和爆炸的问题。从杨少波研究中<sup>[16]</sup>可以发现 LSTM 作为 RNN 的一种优秀变种模型，继承了 RNN 大部分模型的特性并解决了 RNN 中梯度爆炸和消失的问题，能很好地挖掘时间维度上的关联性。但其仍存在的缺点是，它仍是以递归神经网络为基础的串行结构，在并行处理上存在劣势，模型中每个状态都依赖前一步状态，计算费时训练速度较慢。Ashish Vaswani 指出<sup>[17]</sup>，Transformer 模型是完全基于注意力机制的深度学习模型，具有编码器-解码器（encoder-decoder）结构。同时 Transformer 舍弃了类似 LSTM 模型的递归结构，添加位置编码模块对时序信息建模，结合多头注意力机制实现了并行处理输入序列的效果，训练速度更快，并且能够捕捉长时序列数据特征。这些特点，也是本文选取 Transformer 进行研究对象的原因。

近年来，Transformer 在计算机视觉(CV)、自然语言处理(NLP)等领域取得了较大成就。但是，Transformer 在处理时间序列的分类问题上作用也十分明显。Jeeheh Oh 等人<sup>[18]</sup>探索了 Transformer 在临床死亡分类上的应用，他们使用 Transformer 网络用来学习识别和解释临床时间序列数据中的不变性，并将其应用于院内死亡类别的分类预测。结果表明，Transformer 可以直接从数据中学习各种有价值的不变性，适用于院内死亡类别预测分类的任务。Marc Rußwurm 等人<sup>[19]</sup>将 Transformer 运用在遥感研究中，分别用 CNN、RNN、Transformer 对原始卫星时间序列数据和预处理的数据进行类型分类。结果表明，Transformer 的自注意力机制对于数据中隐式特征的提取效果明显，在原始卫星时间序列分类上优于其余两种模型。Jaderberg 等人<sup>[20]</sup>将 Transformer 改成了 Spatial Transformer。该模型允许对网络中的多维时间数据进行空间操纵，对数据的特征部分进行显式空间转换。结果表明改进模型在插入原网络结构中可以提高分类任务的准确率。M Liu 等人<sup>[21]</sup>改进发明了 Gated Transformer，其通过编码器分别对多维时间序列从时间空间上进行编码，后以门控的形式计算各输出权重并合并输出，并证明了该模型相比目前主流模型在处理多维时序分类问题上的优越性。

通过地震预测的相关工作可知，深度神经网络的方法有重要的作用。通过主流模型处理时间序列的相关工作可知，Transformer 对于时间序列的问题更为合适。

通过 Transformer 或其改进结构处理时序分类的相关工作可知, Transformer 的自身结构在改进后, 能够进一步提高时序分类的效果。从以上学者的研究工作中, 进行总结学习, 本文将使用的 Transformer 模型结构, 运用来自于四川、云南、甘肃、宁夏四个地区八个台站的时间应力数据来训练模型从而实现地震方位分类预测。又因为每个台站包含四个检测方向, 且各台站的四个方向不尽相同, 因此处理的时间序列属于多维时间数据, 需要改进 Transformer 结构使其能完成多维时序数据的分类预测任务。

### 1.3 本文的主要工作

本文使用 Transformer 模型完成地震方位的分类预测任务, 模型输入数据来自多个地震前兆数据检测台站的多维时序应力数据。通过 K-means 聚类对某区域内震源方位进行划分将其作为输出类别。同时改进 Transformer 模型, 利用模型中的多头注意力机制以及编码器结构分别提取多维时序数据中的时间和空间信息, 获取多维时间应力序列与震源方位的非线性关系。同时改变模型训练参数来获取最佳模型, 同时进行二分类和多分类对比实验来验证模型的效果。最后用 Python 的 tkinter 库实现对地震方位预测的可视化呈现。

具体工作如下:

第一章: 绪论。介绍地震预测和应力观测数据的背景及研究意义, 并对国内外针对地震预测、深度学习模型对时序数据、Transformer 对时序数据的分类预测任务三方面的发展动态进行介绍。根据学者们使用的研究方法, 引出本文选用 Transformer 用于地震方位预测的方法。

第二章: Transformer 分类预测模型的设计。设计的目的是使模型能够对输入的多维时序应力数据, 预测输出地震的方位类别。具体的设计方式是利用模型的编码器结构, 分别编码提取序列的时间信息和空间信息, 将其合并输入 softmax 层来计算各编码器输出权重。并将权重与各编码结果相乘最后合并输出。同时介绍在分类任务中所需的评价指标如混淆矩阵、准确率、精确率、召回率、f1score、ROC 曲线、ROC 曲线下面积。

第三章: 基于 Transformer 的地震方位预测分析。介绍了基于 Transformer 模型进行地震方位预测的过程以及训练最优模型的过程。首先进行时序数据的预处理并划分地震方位。然后基于 Transformer 的结构进行多分类模型的设计。之后使用该模型进行二分类任务来验证模型对于多维时序应力数据进行分类的可行性。再利用该模型进行大量实验并修改对应实验参数获取最佳模型, 最后通过改变应力数据中震级的范围进行训练来探究震级大小与模型预测准确率大小的相关性。

第四章: 地震方位预测 ui 界面设计。基于 Python 的 tkinter 库, 设计针对于该文章框架下的 ui 界面, 包括应力数据的预处理、台站应力数据与地震目录的呈现、模型预测以及结果呈现等功能。

第五章为总结与展望。对此次研究成果进行总结，提出本文研究内容的意义，并分析其中存在的问题和不足，提出对于未来的思考和展望。

## 2 Transformer 分类预测模型设计

### 2.1 Transformer 基本结构

Transformer 网络结构主要由编码组件 Encoder 与解码组件 Decoder 构成，结构如图 2-1 所示。编码组件是由  $N$  个编码器组成的堆栈，每个编码器有两个子层，分别是多头自注意力机制层及全连接前馈网络层。每个子层上方还包括一个 Add&Norm 层，Add 表示残差连接，用于防止网络退化，Norm 表示对每一层的激活值进行归一化。解码组件包含  $N$  个解码器，相比于编码器多增加一层带掩码 Masked 的注意力层。Masked 的注意力层。

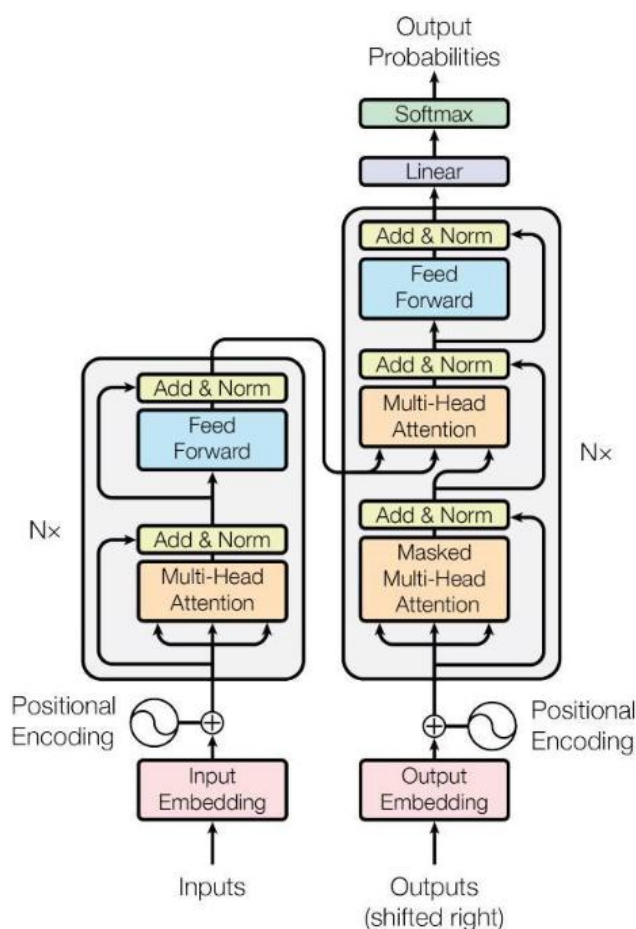


图 2-1 Transformer 结构图

图片来源：《LANGUAGE MODELING WITH NN.TRANSFORMER AND TORCHTEXT》

[https://pytorch.org/tutorials/beginner/transformer\\_tutorial.html](https://pytorch.org/tutorials/beginner/transformer_tutorial.html)

Transformer 模型广泛运用于 NLP（自然语言处理）任务，该任务下它的工作流程为：对于输入的句子，假设其单词有  $n$  个，编码嵌入维度为  $d$ 。那么第一步对其每一个单词进行词编码和位置编码再相加得到每个单词的表示向量  $X(n \times d)$ 。第二步将得到的单词表示向量矩阵传入编码器结构中，经过  $N$  个编码组件后可以得到句子所有单词的编码信息矩阵  $C(n \times d)$ ，此时输出的矩阵维度与输入完全一致。

第三步则将输出的编码信息矩阵  $C$  传递到解码器结构中，解码器依次会根据当前翻译过的第  $1 \sim i$  个单词翻译下一个单词  $i+1$ 。并且在使用的过程中，翻译到第  $i+1$  单词时，需要通过 Mask (掩盖) 操作遮盖住  $i+1$  之后的单词。

在 Transformer 结构中，对输入向量进行维度转换的编码层、自注意力机制层较为重要。

编码层，分为输入单词的直接编码和前后位置编码。对于一个大小为  $(\text{num}, \text{step}, \text{channel})$  的输入序列  $X$ ，经过直接编码后大小变为  $(\text{num}, \text{step}, \text{hidden\_dim})$ ， $d\_embed$  通常为 512。而对于位置编码，常用的方法是利用三角函数位置编码公式：

$$p_{(2i, \text{pos})} = \sin(\text{pos}/10000^{2i/\text{hidden\_dim}}) \quad (2-1)$$

$$p_{(2i+1, \text{pos})} = \cos(\text{pos}/10000^{2i/\text{hidden\_dim}}) \quad (2-2)$$

其中  $i$  指的词向量第  $i$  维， $\text{pos}$  是当前词的绝对位置。采取如此正余弦编码的原因是，正弦和余弦函数具有周期性，对于固定长度偏差  $k$ ， $p_{(\text{pos}+k)}$  可以表示为  $p_{(\text{pos})}$  的一个线性变化，方便模型学习词与词之间的一个相对位置关系。

自注意力机制的核心过程就是通过矩阵  $Q$  和  $K$  计算得到注意力权重；然后再作用于  $V$  得到整个权重和输出。具体的，对于输入  $Q$ 、 $K$  和  $V$  来说，其输出向量的计算公式为：

$$\text{Attention}(Q, K, V) = \text{soft max}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (2-3)$$

其中  $d_k$  是  $Q$ 、 $K$ 、 $V$  矩阵的列数，即向量维度。一般  $d_q=d_k=d_v$  时表示自注意力机制， $d_q=d_v \neq d_k$  时为注意力机制。对于  $Q$ ， $K$ ， $V$  三矩阵，它们是通过编码后序列  $X$  分别乘以 3 个不同的矩阵计算而来。图 2-2 所示。

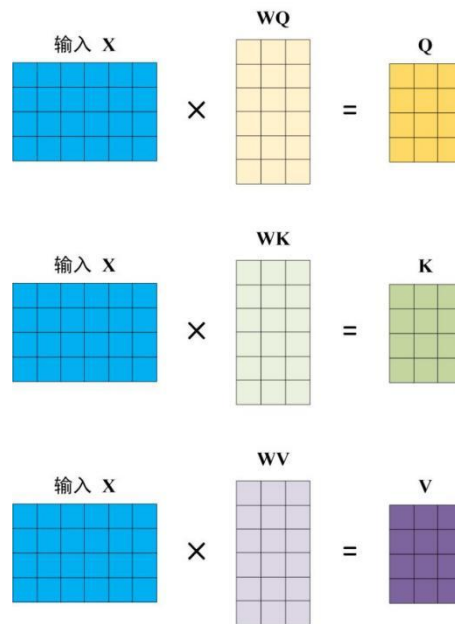


图 2-2 自注意力机制 QKV 矩阵计算图

图片来源：<https://blog.csdn.net/pxiongw/article/details/122656623>

由图 2-2 所示, 可以看到对于同一个嵌入序列  $X$  进行 3 次不同的线性变换后, 会得到 QKV 不同的 3 种矩阵。有了自注意力机制后, 仅仅只需要对原始输入进行几次矩阵变换便能够得到最终包含有不同位置注意力信息的编码向量。而多头注意力机制, 也就是包括多个自注意力机制层, 数据在并行输入后再进行合并。

## 2.2 Transformer 结构改进

首先, Transformer 结构在运用到一维时间序列的分类任务时, 并不涉及解码这一过程, 通常只需将时间序列输入到编码器中, 所得的向量再输入到 softmax 分类器中输出各种概率即可完成分类。因此针对多维时间序列的时间信息, 可以直接采用解码器结构。

其次, 多维时间序列相比于一维时间序列, 在空间上通常含有更多的特征。因此解决此问题可以借助 Transformer 结构中的自注意力机制(Self-Attention)来发掘不同特征之间的关系。

不仅如此, 多维时间序列中的空间特征信息, 并非如时间维度上具有位置的绝对相关或相对相关。即使切换时间序列中不同特征的顺序, 序列原始的信息也不会变化。因此, 在将时间序列特征信息输入到编码器时, 并不需要添加位置编码。

甚至, 处理特征信息的多头注意力层, 也不需要采用掩码(Masked)操作。因为在 NLP 任务的解码器中, 翻译解码的过程是按顺序的, 即翻译完第  $i$  个单词, 才可以翻译第  $i+1$  个单词。有了掩码操作则可以防止第  $i$  个单词知道  $i+1$  个单词之后的信息。但对于特征信息并不存在前后位置关系, 因此计算注意力得分时让其知道前后特征并无影响。

最后, 从两种编码器得到编码输出后不能只进行简单合并, 这样会使得时间和空间编码器所得的编码结果相互影响损害性能。为了使两种编码结果相互关联, 先将编码结果进行合并, 后输入线性层改变形状再经过 softmax 层获得两种编码器输出各自所占的权重。完成此步骤后, 将各自权重与之前编码器输出相乘后即可进行合并, 并通过线性层改变形状, 最后所得的, 即各类别的概率。

综上所述, 针对多维时序数据的分类任务, 设计改进后的 Transformer 网络结构如图 2-3 所示。

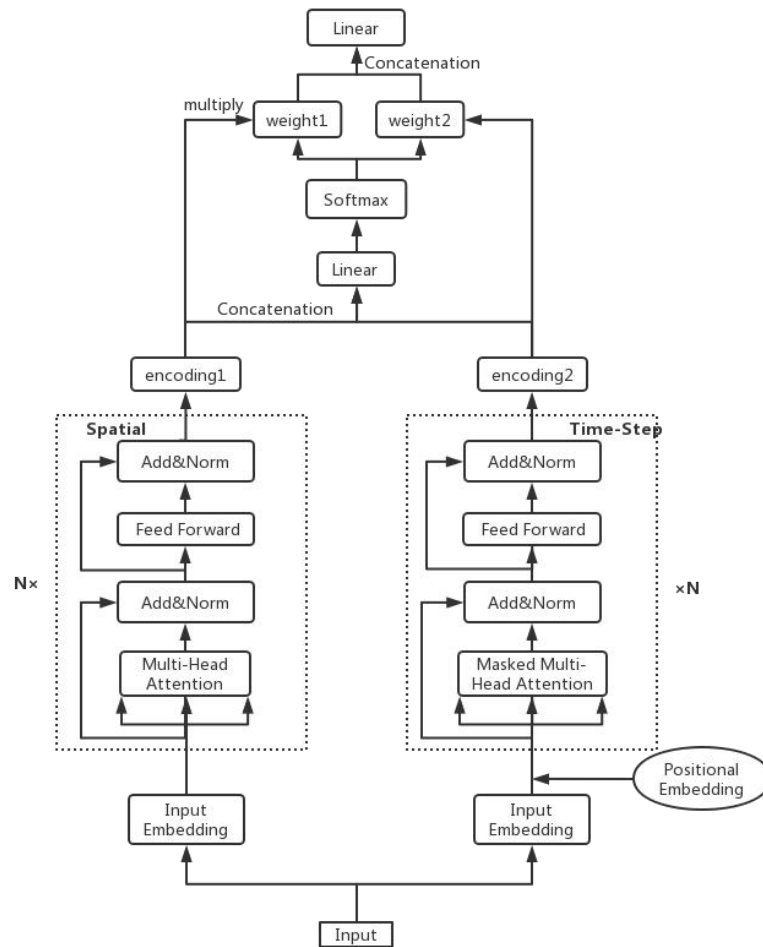


图 2-3 改进后 Transformer 结构

由图 2-3，假定输入的多维时间序列  $X$  大小为  $(n, \text{time\_step}, \text{feature})$ ， $n$  表示时间序列的数量， $\text{time\_step}$  表示每个序列选取的时间长度， $\text{feature}$  表示每个序列包含的特征数量。假定输出的类别为 9 类。对于输入的一条多维时间序列，当输入进不同的编码器时，对应地将时间轴和特征轴进行转换。一个编码器用于处理序列时间信息，一个用于处理序列特征信息。

由图 2-3 右处 **temporal** 时间编码器所示，其功能是对时间信息进行编码，使用 Transformer 原始的编码结构，带掩码(Masked)的自注意力机制来计算相同特征信息下不同时间点之间的注意力权重。与原始的模型架构一样，数据经过多头注意力层后经过前馈全连接层，以增强特征提取。并保留了原始编码器结构中周围的残差连接层以引导信息和梯度流，并进行层归一化。

由图 2-3 左处 **spatial** 空间特征编码器所示，不同于前者的是，其对空间特征信息进行编码。不含有位置编码结构以及掩码结构，用于计算相同时间信息上不同特征点的注意力权重。其余结构同于时间编码器。

由图 2-3 上半部分,当两类编码器分别得到编码结果 `encoding1`、`encoding2` 后,进行合并处理。为了防止时间、空间编码器所计算的注意力权重在合并后受到损害,需要加上每个编码器输出的权重信息。因此将 `encoding1`、`encoding2` 进行合并、线性变换后通过 `softmax` 函数层得到各自权重值 `weight1`、`weight2`。此时通过相乘的方式结合权重与编码输出,再合并所得的结果即为体现该数据所属各类别的概率特征向量。

## 2.3 分类评价指标

在分类任务中,各种指标的意义十分重要,首要的分类指标为混淆矩阵,其中的各个参数,如图 2-4 所示。

混淆矩阵		真实值	
		Positive	Negative
预测值	Positive	TP	FP Type I
	Negative	FN Type II	TN

图 2-4 混淆矩阵指标图

图片来源: [https://blog.csdn.net/xys430381\\_1/article/details/88198167](https://blog.csdn.net/xys430381_1/article/details/88198167)

在混淆矩阵中,当真实值是 `positive`,模型认为是 `positive` 的数量称为 TP(True Positive)。当真实值是 `positive`,模型认为是 `negative` 的数量称为 FN(False Negative)。这也被称为统计学上的第一类错误(Type I Error)。当真实值是 `negative`,模型认为是 `positive` 的数量称为 FP(False Positive),这也就是统计学上的第二类错误(Type II Error)。当真实值是 `negative`,模型认为是 `negative` 的数量称为 TN(True Negative)。

准确率:表示的是算法分类正确的数据个数与输入算法的数据的个数之比,公式如下:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{T} + \text{F}) \quad (2-4)$$

使用准确率评价算法有一个问题,就是在数据的类别不均衡,特别是有极偏的数据存在的情况下,准确率这个评价指标是不能客观评价算法的优劣的。

精确率:模型预测为正的样本中预测对的比重,公式如下:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (2-5)$$

召回率:真实值为正的样本中模型预测对的比重,公式如下:

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) = \text{TP} / \text{T} \quad (2-6)$$

精准率和召回率是此消彼长的,即精准率高了,召回率就下降,在一些场景下要兼顾精准率和召回率,于是就有 F1 score。



**F1 值**：精确率和召回率的兼顾指标，是精确率和召回率的调和平均数。调和平均数的性质是，只有当精确率和召回率二者都非常高的时候，它们的调和平均才会高。如果其中之一很低，调和平均就会被拉得接近于那个很低的数。

**ROC (Receiver operating characteristic)**：接收者操作特征曲线，是反映敏感性和特异性连续变量的综合指标，roc 曲线上每个点反映着对同一信号刺激的感受性。其横坐标表示伪正类率(False positive rate, FPR)，预测为正但实际为负的样本占所有负例样本的比例。纵坐标表示真正类率(True positive rate, TPR)，预测为正且实际为正的样本占所有正例样本的比例。

**AUC(Area Under Curve)**：定义为 ROC 曲线下的面积，这个面积的数值不会大于 1，大小类似于 F1 score，都是综合评价精准率和召回率的指标。AUC 的取值范围一般在 0.5 和 1 之间，使用 AUC 值作为评价标准是因为很多时候 ROC 曲线并不能清晰的说明哪个分类器的效果更好，而作为一个数值，对应 AUC 更大的分类器效果更好。当  $AUC = 1$ ，是完美分类器，采用这个预测模型时，存在至少一个阈值能得出完美预测。当  $0.5 < AUC < 1$ ，优于随机猜测。此分类器若妥善设定阈值的话，能有预测价值。 $AUC = 0.5$ ，效果类似于随机猜测，模型没有预测价值。

## 2.4 本章小结

本章从 Transformer 神经网络的基本原理展开叙述，介绍 Transformer 的工作流程以及其中的编码层、自注意力层的原理和作用。根据多维时序数据的特点，设计出两种编码器分别编码时间信息和空间信息。不同编码器的结构也有所不同，时间编码器沿用 Transformer 原编码器结构，拥有输入编码、位置编码、带掩码机制的自注意力机制、前馈全连接层、残差结构。而因为序列的空间特征信息不存在前后位置关系，因此不需要添加位置编码以及带掩码的自注意力机制。对于两个编码器的输出，还需要计算各自的权重占比再进行合并。介绍完改进的模型结构后，并最后介绍了分类任务需要研究的指标。

### 3 基于 Transformer 的地震方位预测分析

#### 3.1 数据预处理

时间序列数据包含大量信息，但通常是较难直接获得的。与时间序列相关的常见问题是无序时间戳、异常值、数据中的噪声。

##### 3.1.1 钻孔应变数据简介

针对钻孔应力数据，需要了解其整体大小、变化、形状、等特点以及地震前后数据变化的情况，从而后续进行针对性处理。

选取了海原小山地震台四个方向的应力数据，其范围从2010年1月1日至2017年9月4日共2804天4037760条数据，通过python的pandas库进行数据读取并分析，先查看数据的基本情况，如图3-1所示。

各字段情况	direction_first	direction_second	direction_third	direction_fourth
count	4.037760e+06	4.037760e+06	4.037760e+06	4.037760e+06
mean	1.936593e+05	1.855911e+05	3.084658e+05	2.415418e+05
std	3.893599e+04	4.619515e+04	5.220395e+04	4.183128e+04
min	1.632457e+05	1.197715e+05	3.011400e+04	1.957815e+05
25%	1.869396e+05	1.727361e+05	2.768145e+05	2.206320e+05
50%	1.966212e+05	1.868686e+05	3.128195e+05	2.421047e+05
75%	1.986004e+05	1.984985e+05	3.423111e+05	2.568416e+05
max	9.999990e+05	9.999990e+05	9.999990e+05	9.999990e+05

图 3-1 海原台四方向应力数据指标

由图3-1可知，数据中四个方向应力数据中存在异常值999999。

查看数据中是否含有缺失值、重复值，如图3-2所示。

```

缺失的数据:
  Unnamed: 0      0
direction_first  0
direction_second  0
direction_third   0
direction_fourth  0
dtype: int64
具体缺失查看
  Unnamed: 0      False
direction_first  False
direction_second  False
direction_third   False
direction_fourth  False
dtype: bool
data_shape
(4037760, 5)

```

图 3-2 海原台四方向缺失值、重复值指标

由图3-2可知，数据中不含有缺失值、重复值。

以数据中方向一(方向角  $111^{\circ}$ )为例, 查看其中数据整体的分布直方图, 如图 3-3 所示。

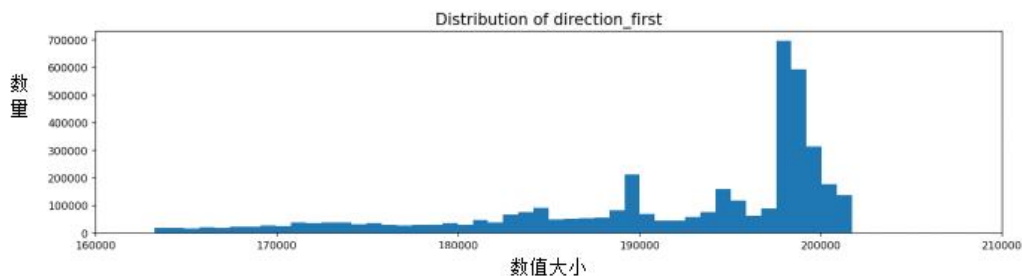


图 3-3 海原台方向 1 分布直方图

由数据分布图 3-3 可知, 海原台方向一的应力数据大小大多集中在 20000 附近。绘制查看方向一的折线图 and 散点图, 如图 3-4 所示。

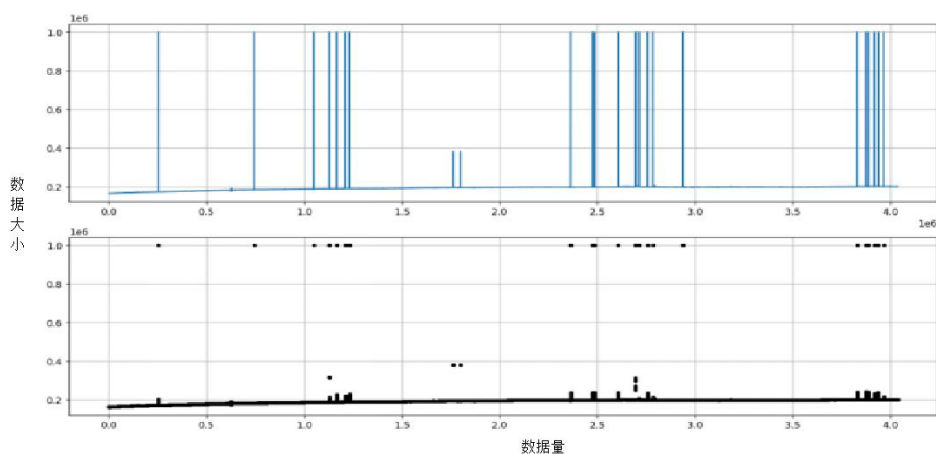


图 3-4 海原台方向 1 折线图/散点图

由图 3-4 中折线图可知, 数据中存在较多的突变, 突变范围很大。由图 3-4 中散点图则可以清晰看到突变的峰值点都跳变异常, 且值都接近于  $1e-6$ , 属于数据中的极端值, 是需要去除的。

用四分位法获取上述方向一数据中的中位数、上四分位数、下四分位数、上边缘以及下边缘并查看数据的四分位图, 如图 3-5 所示。

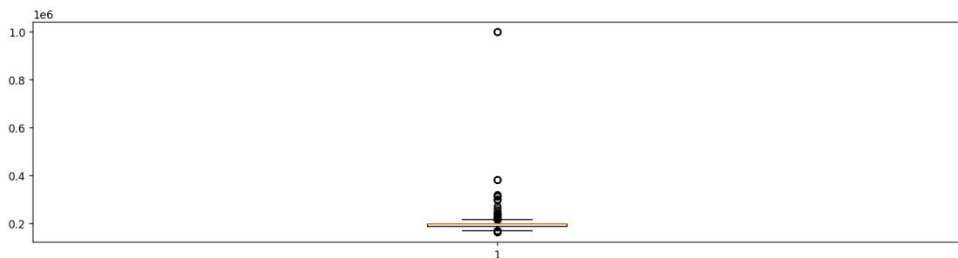


图 3-5 海原台方向 1 箱线图

由图 3-5 可知, 数据中确实存在相当多的极端值, 需要进行替换或去除。这里先采用替换的方法, 使用盖帽法, 将方向一中极端值替换为数据中的边缘值, 替换后的数据分布直方图、折线图和散点图如图 3-6 所示。

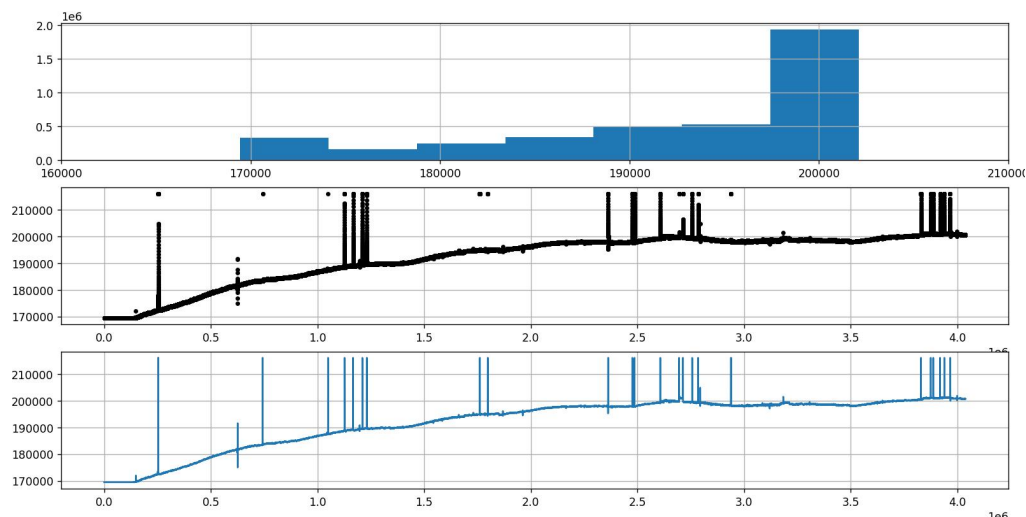


图 3-6 盖帽法处理后

由图 3-6 可知，经四分位法处理后，数据中不存在极端值，但因上述使用的盖帽法，其存在的缺陷是原来的极端值现在都变成了上边缘值，使得数据中的异常突变情况仍未改变，只是突变的峰值降低了，甚至可能会误导地震的突变观测。而从图中散点图也可以看到‘正常’数据中依旧存在着较大的突变，突变值大多在 20000 以上，但无法确定此类突变是否为地震所造成的，因此需要先查看地震相关波形从而进行判断。

通过反复查看比对地震时间范围内的波形，从而做出判断：地震波形最大突变范围为 2000~3000，突变值远不到 10000 甚至 20000 以上。因此可以断定上述‘正常’数据中的较大突变值属于时间序列中的噪声，是需要处理的。海原台方向 1(方向角  $111^\circ$ ) 突变值为 2000 左右的的地震波形图如图 3-7 所示。

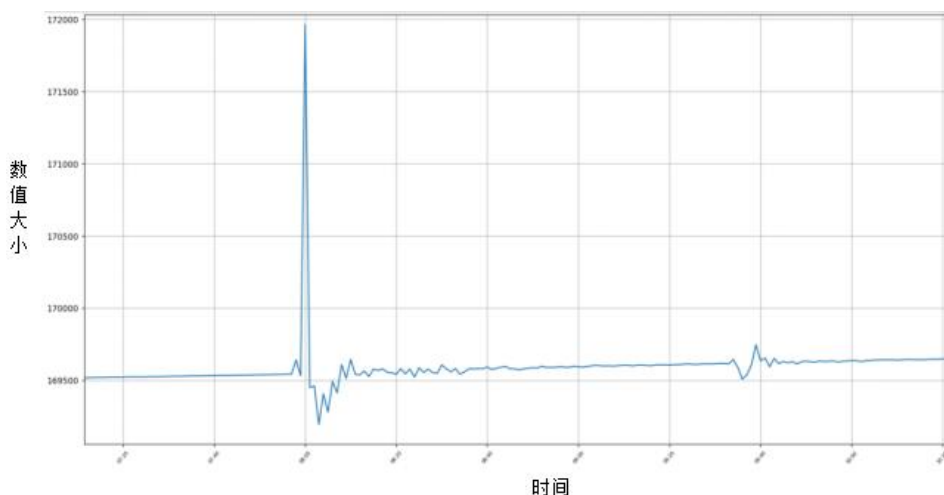


图 3-7 海原台方向 1 地震波形图

### 3.1.2 时间戳处理

利用 pandas 库中的 `pd.to_datetime` 方法以及 `strftime('%Y-%m-%d-%H:%M')` 方法规定数据中的时间戳格式，如图 3-8 所示。

	datetime	direction_first	...	direction_third	direction_fourth
0	2010-01-01-00:00	163531.1	...	216990.0	195843.0
1	2010-01-01-00:01	163531.5	...	216993.2	195843.8
2	2010-01-01-00:02	163533.7	...	216989.7	195846.2
3	2010-01-01-00:03	163532.9	...	216994.9	195843.6
4	2010-01-01-00:04	163531.5	...	216990.7	195843.3
...	...	...	...	...	...
4037755	2017-09-04-23:55	200733.7	...	369059.2	272781.0
4037756	2017-09-04-23:56	200733.7	...	369061.9	272782.9
4037757	2017-09-04-23:57	200732.2	...	369057.6	272783.2
4037758	2017-09-04-23:58	200732.3	...	369062.6	272782.1
4037759	2017-09-04-23:59	200730.8	...	369063.6	272781.8

[4037760 rows x 5 columns]

图 3-8 时间戳处理后

3.1.3 极端值处理

因为应力具有拉伸和压缩两个方向，所以数据会存在正负值。由“数据了解”部分可知，数据中的异常值即大的极端值 999999。所以同上述，首先通过四分位法计算得出上四分位数、下四分位数、上界、下界，从而区分出异常值和正常值。不同于上述盖帽法操作的是，本次操作将位于上界之外的值去除并置为 NAN，视为缺失值处理。又因为缺失值都属于一个个独立的点，其周围的点大多保持不变，则可用一次线性插值的方式利用周围的点来模拟缺失值。

3.1.4 噪声处理

针对数据中存在的噪声部分，即“数据了解”部分中所谈及的大于 10000 的突变。对其处理方式是找出数据中的所有突变点，并筛选出突变值（峰值-谷值）大于某一阈值（这里取 10000）的所有点。之后用一定大小的窗口（这里取该突变点前后一天长度，即 1440 个点），在此类点附近移动，每次将窗口中心的值替换为窗口内的平均值，从而实现滚动平均去噪。海原台四个方向的去噪前后对比如图 3-9 至图 3-12 所示。

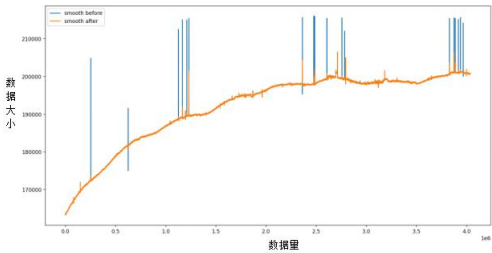


图 3-9 方向 1 去噪前后

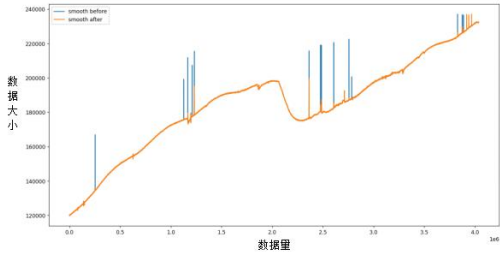


图 3-10 方向 2 去噪前后

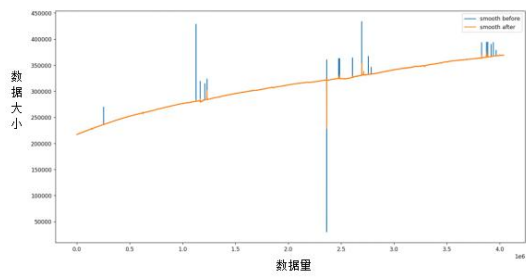


图 3-11 方向 3 去噪前后

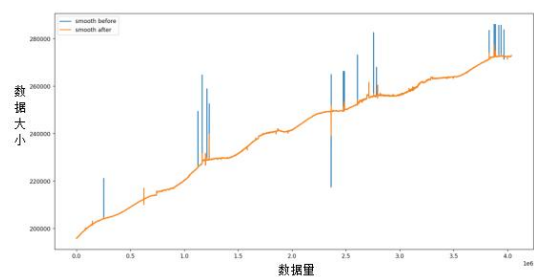


图 3-12 方向 4 去噪前后

由图 3-9 至 3-12 可知，数据去噪后整体走势上突变减少，同时在不改变数据细节的情况下现有突变更符合数据走势，贴身数据本身特点。

### 3.2 地震区域划分

#### 3.2.1 地震事件的空间位置分析

在地图上观察已有的八个台站分布，如图 3-13 所示。

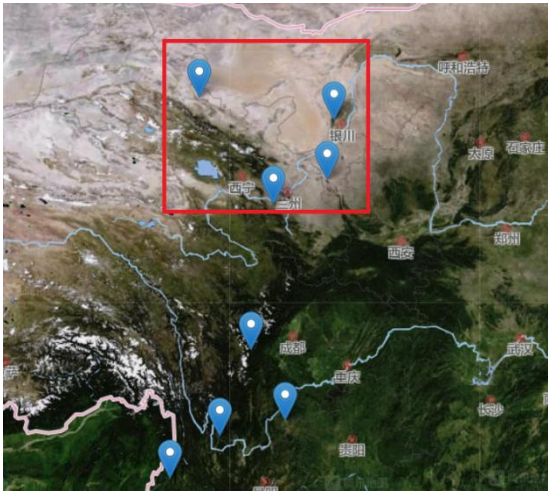


图 3-13 台站分布卫星图

由图 3-13 可知，八个台站，整体上以四个为一群呈现南北分布，本文选取分布在北边的四个台站海原、银川、临夏、高台的数据以及周边区域进行分析。对于周边区域，本文划定的讨论范围为东经  $96^{\circ}$  至东经  $107^{\circ}$ ，北纬  $33^{\circ}$  至北纬  $41^{\circ}$ 。对于数据的时间范围，选取自 2010 年 1 月 1 日至 2017 年 1 月 1 日。对于地震的震级，选取为震级 3 至震级 10。对于深度，选取 0 至 10000 米，地点选取国内。该范围内地震数目为 240 个。

#### 3.2.2 基于 Kmeans 聚类地震空间区域分类

与分类、序列标注等任务不同，聚类是在事先并不知道任何样本标签的情况下，通过数据之间的内在关系把样本划分为若干类别，使得同类别样本之间的相似度高，不同类别之间的样本相似度低，即增大类内聚，减少类间距。

聚类属于非监督学习，K 均值聚类是最基础常用的聚类算法。它的基本思想是，通过迭代寻找 K 个簇的一种划分方案，使得聚类结果对应的损失函数最小。其中，损失函数可以定义为各个样本距离所属簇中心点的误差平方（SSE）和：

$$J(c, \mu) = \sum_{i=1}^M \|x_i - \mu_{c_i}\|^2 \quad (3-1)$$

其中  $x_i$  代表第 i 个样本， $c_i$  是  $x_i$  所属的簇， $\mu_{c_i}$  代表簇对应的中心点， $M$  是样本总数。

其中核心在于划定聚类数量 K 值，可以将簇内误差平方和、轮廓系数和戴维斯-博尔丁指标相结合<sup>[22]</sup>。

簇内误差平方和（SSE）指的是数据集中所有样本点到其簇中心的距离之和的平方。选取的是误差平方和斜率突然变小时，即转折点的 k 值，象征着聚类的好坏，K 越大 SSE 越低。

轮廓系数（Silhouette Coefficient）表明每个点与同一聚类中的其他点的相似程度，如果大多数点有一个高的轮廓系数，那么聚类方案是合适的。其公式如下：

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (3-2)$$

其中  $a(i)$  表示样本点 i 到与其属于同一个簇的其他样本点的距离的平均值。 $a(i)$  越小，说明该样本 i 属于该类的可能性越大。 $b(i)$  表示样本点 i 到其他簇  $c(j)$  中的所有样本的平均距离  $b_{ij}$  的最小值。而所有样本点 i 的轮廓系数的平均值，即为该聚类结果总的轮廓系数 S，S 越接近与 1，聚类效果越好。

戴维斯-博尔丁指标基于聚类内距离和聚类间距离的比率，其公式如下：

$$DB = \frac{1}{k} \sum_{i=1}^k \max \{D_{i,j}\}_{j \neq i} \quad (3-3)$$

其中， $D_{i,j}$  是第 i 个和第 j 个簇的簇内到簇间距离比， $D_{ij}$  的最大值表示簇 i 的最坏簇内-簇间比率。简单来说最优聚类解决方案具有最小的 Davies Bouldin 值。

三种指标图如图 3-14 至图 3-16 所示。



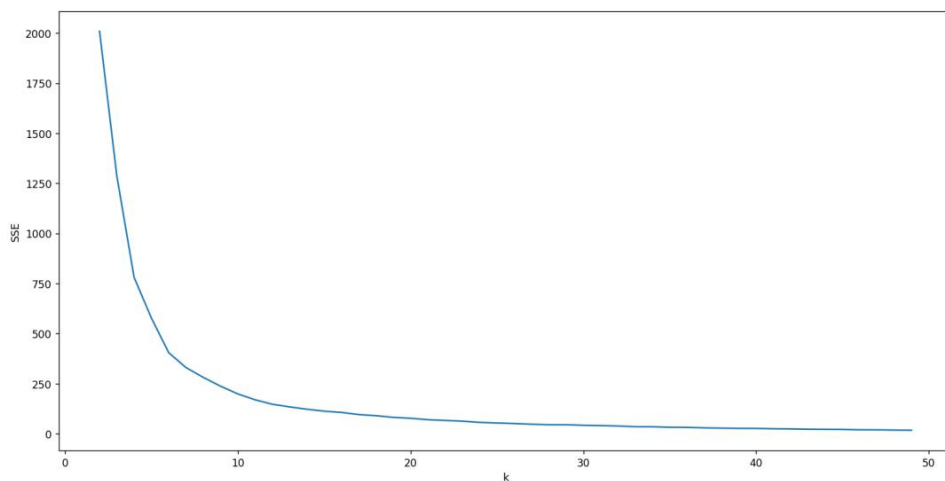


图 3-14 簇数-簇内误差平方和图

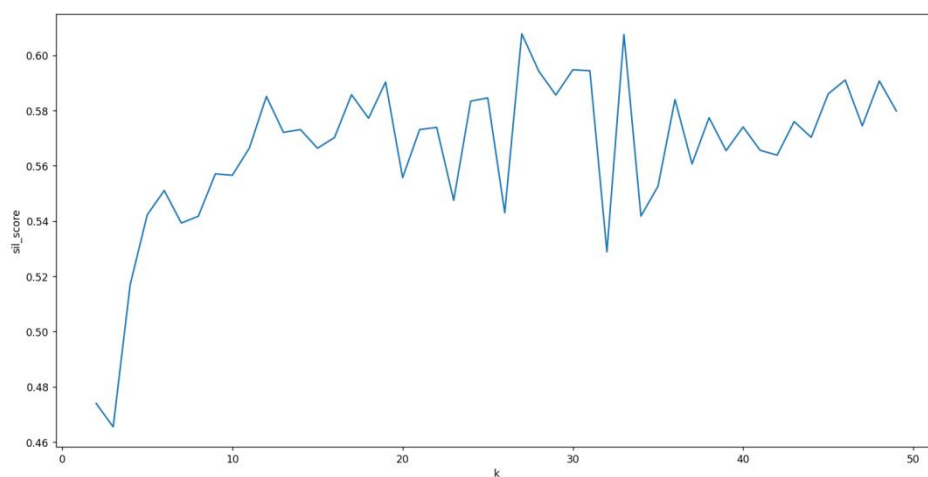


图 3-15 簇数-轮廓系数图

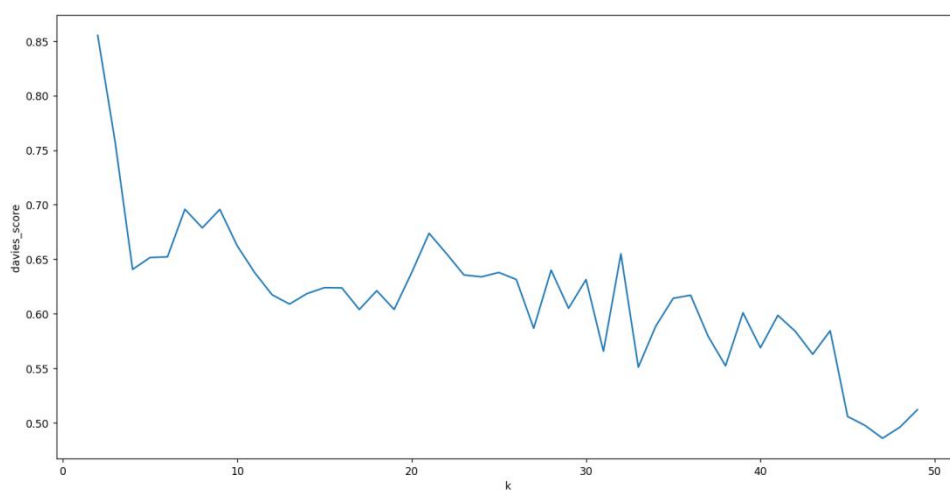


图 3-16 簇数-戴维斯博尔丁指标图

因为地震目录中地震数为 240 个，因此划分的区域  $k$  不适合多，规定  $k$  最高只能取 10。由图 3-14 所示，可知在  $k > 5$  与  $k < 10$  之间簇内误差平方和图像满足。



由图 3-15 所示，在  $k>5$  与  $k<10$  之间  $k=10$  附近轮廓系数较大。由图 3-16 可知，戴维斯-博尔丁指标随着  $k$  增大整体下降趋势，在  $k>5$  且  $k<10$  之间同样是  $k=10$  附近指标较低。因此结合三种指标，选取  $k$  值 9 为簇类数。

综上所述，对东经  $96^\circ$  至东经  $107^\circ$ ，北纬  $33^\circ$  至北纬  $41^\circ$ ，自 2010 年 1 月 1 日至 2017 年 1 月 1 日，震级为震级 3 至震级 10，深度 0-1000 的地震进行区域分块，分成 9 块如图 3-17 所示。

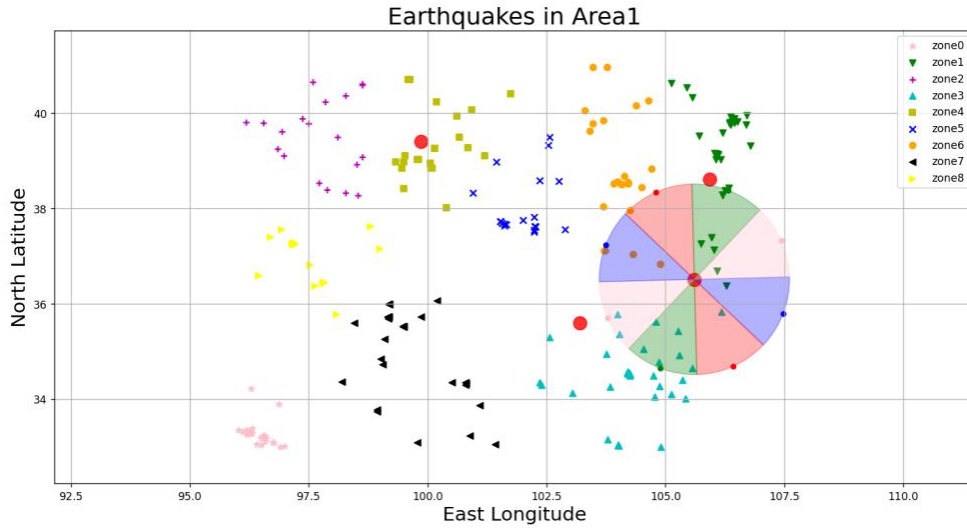


图 3-17 九类区域与台站分布图

由图 3-17，图中红色四点代表四个台站，扇形区域表示的以台站方向为中线，左右延伸  $22.5^\circ$  共  $45^\circ$  的检测范围，对顶角方向表示同一方向，蓝红绿粉扇形区域分别表示方向 1234。

### 3.3 地震方位预测结果分析

#### 3.3.1 基于机器学习分类器的二分类验证

##### 1、实验方案

选取“地震区域划分”部分所划定的区域，区域参数为东经  $96^\circ$  至东经  $107^\circ$ ，北纬  $33^\circ$  至北纬  $41^\circ$ ，自 2010 年 1 月 1 日至 2017 年 1 月 1 日，震级为震级 3 至震级 10，深度 0-1000 米。使用的应力数据为海原台自 2010 年 1 月 1 日至 2017 年 1 月 1 日的四个方向数据。在所划分区域的地震目录中包含 240 条地震，根据地震目录的时间找到应力数据中对应的时间节点，并相应地取该节点前 10 分钟后 30 分钟数据共 40 条，将其当作正样本表示为有地震。再随机从整个应力数据中选出除了该地震目录时间节点下的其余 240 个地震节点，同前面相应地取前 10 分钟后 30 分钟数据共 40 条，将其当作负样本表示为无地震。正负样本大小都为 (240, 40, 4)，240 表示数据条数，40 表示每条数据的时间范围，4 表述每条数据的特征数量。正负样本整体比例为 1:1，训练集测试集比例为 8:2。

依次使用朴素贝叶斯、逻辑回归、k 邻近、支持向量机 SVM、带核的支持向量机 kernel\_svm 五种机器学习的分类器进行实验。

## 2、实验结果

分别通过五种分类器训练后测试获得准确率、精确率、召回率、f1score、AUC（ROC 曲线下面积）等指标。如表 3-1 所示。

表 3-1 五种机器学习分类器测试指标

指标 分类器	准确率 Accuracy	精确率 Precision	召回率 Recall	F1score	AUC
Naive_bayes	0.5089	0.5085	0.5292	0.5186	0.4925
Logistic	0.4776	0.4744	0.4729	0.4751	0.4923
Knn	0.5391	0.5392	0.5375	0.5383	0.5402
Svm	0.4849	0.4611	0.1792	0.2581	0.4901
Kernel_svm	0.5411	0.5406	0.5479	0.5442	0.5549

选取 AUC 评分最高的 Kernel\_svm 分类器的 ROC 曲线查看，如图 3-18 所示。

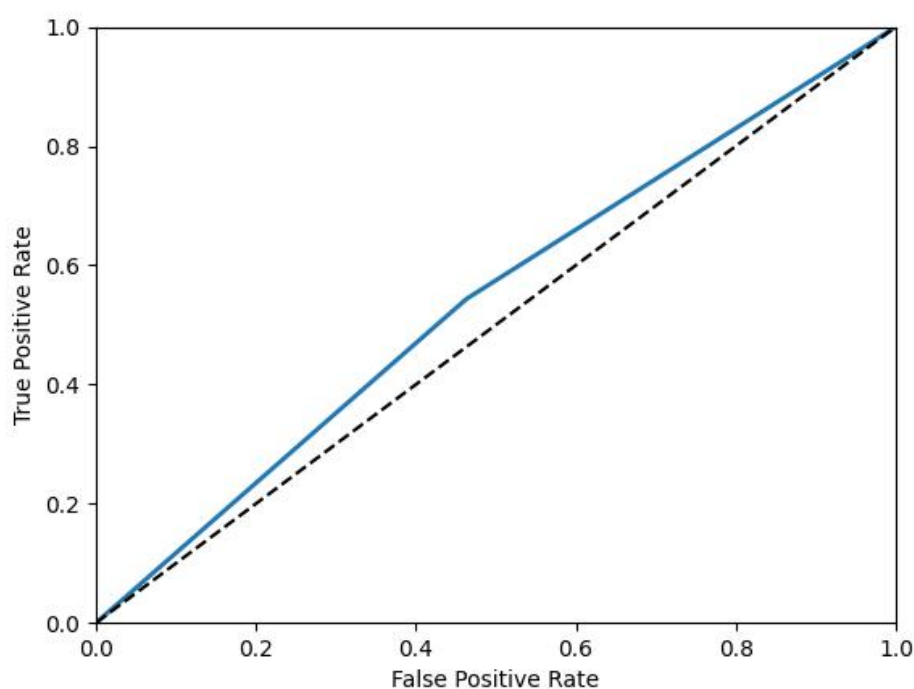


图 3-18 Kernel\_svm 分类器 ROC 曲线下面积

从表 1、图 3-18 整体上看，五个分类器效果并未达到预期，准确率、精确率、召回率都在 50% 上下说明模型可能并未学习到特征，可能全猜 0 或 1。从 F1score 和 AUC 综合指标来看，五种机器学习的分类器难以胜任当前多维时间序列的二分类任务，对于该任务分类器的结构不够复杂，泛化能力不足。因此需要利用设计好的 Transformer 分类模型进行实验。

### 3.3.2 基于 Transformer 的二分类验证

#### 1、实验方案

选取“地震区域划分”部分所划定的区域，区域参数为东经  $96^{\circ}$  至东经  $107^{\circ}$ ，北纬  $33^{\circ}$  至北纬  $41^{\circ}$ ，自 2010 年 1 月 1 日至 2017 年 1 月 1 日，震级 3 至震级 10，深度 0-1000 米。使用的应力数据为海原台自 2010 年 1 月 1 日至 2017 年 1 月 1 日的四个方向数据。在所划分区域的地震目录中包含 240 条地震，根据地震目录的时间找到应力数据中对应的时间节点，并相应地取该节点前 10 分钟后 30 分钟数据共 40 分钟时间序列，将其合成一条数据，且当作正样本表示为有地震。再随机从整个应力数据中选出除了该地震目录时间节点下的其余 240 个地震节点，与前面操作相同。正负样本数量都为 240 条，每条时间步长是 40，表示共 40 分钟数据。特征步长为 4，表示四个方向特征。整体比例为 1:1，训练集测试集比例为 8:2。

将其输入设计好的 Transformer 模型进行训练，输入数据大小为(240, 40, 4)，输出数据类别为 2，即 0 和 1。训练参数中，训练轮次为 1000，batchsize 为 128，学习率 lr 为  $1e-4$ 。

#### 2、实验结果

得到轮次-训练损失、轮次-训练准确率/测试准确率、轮次-训练平均损失、轮次-测试平均损失（每隔 5 轮绘制）、轮次-精确率、轮次-召回率、轮次-f1score 等曲线如图 3-19 至图 3-23 所示。

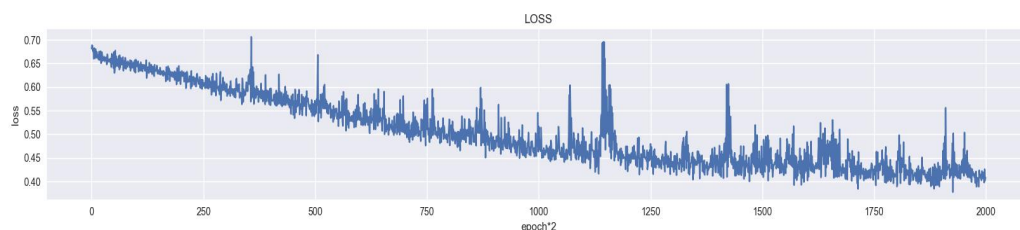


图 3-19 轮次(1000)-训练损失

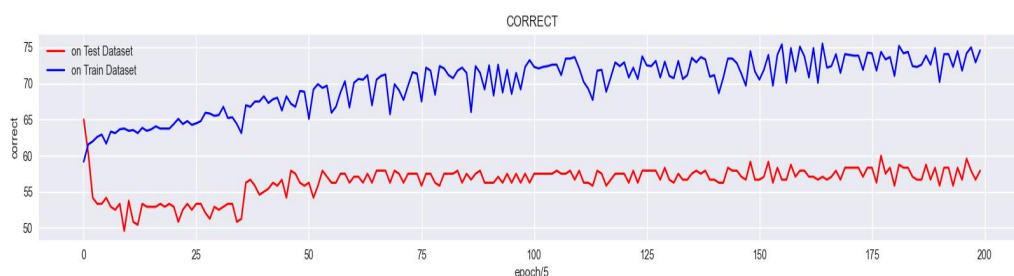


图 3-20 轮次(1000)-训练/测试准确率

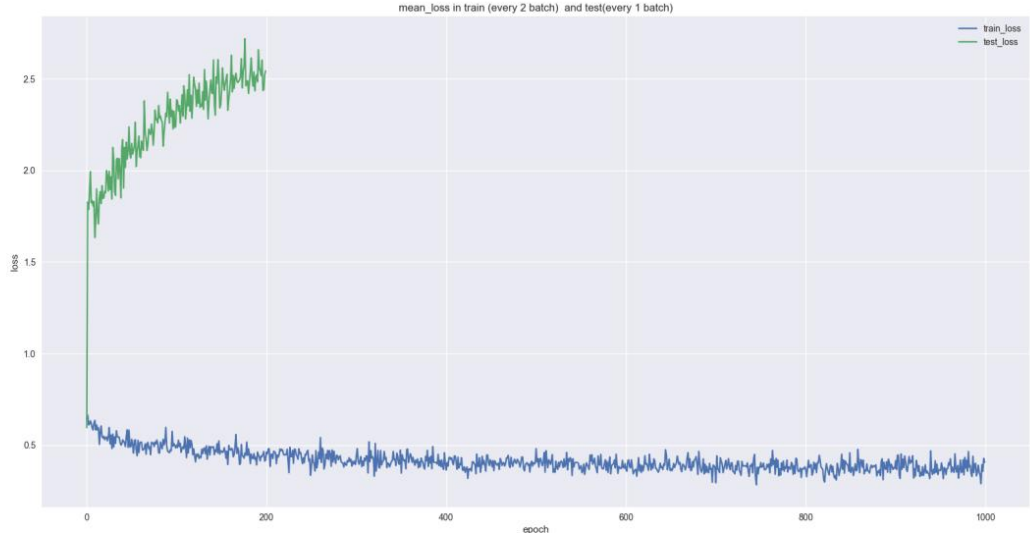


图 3-21 轮次(1000)-训练平均损失

最小loss: 0.377602219581604    最小loss对应的epoch数:964    最后一轮loss:0.408203661441803  
 最大correct: 测试集:65.0% 训练集:75.52%    最大correct对应的已训练epoch数:5    最后一轮correct: 57.92%  
 d\_model=512    q=8    v=8    h=8    N=8    drop\_out=0.2  
 共耗时36.25分钟

图 3-22 训练信息

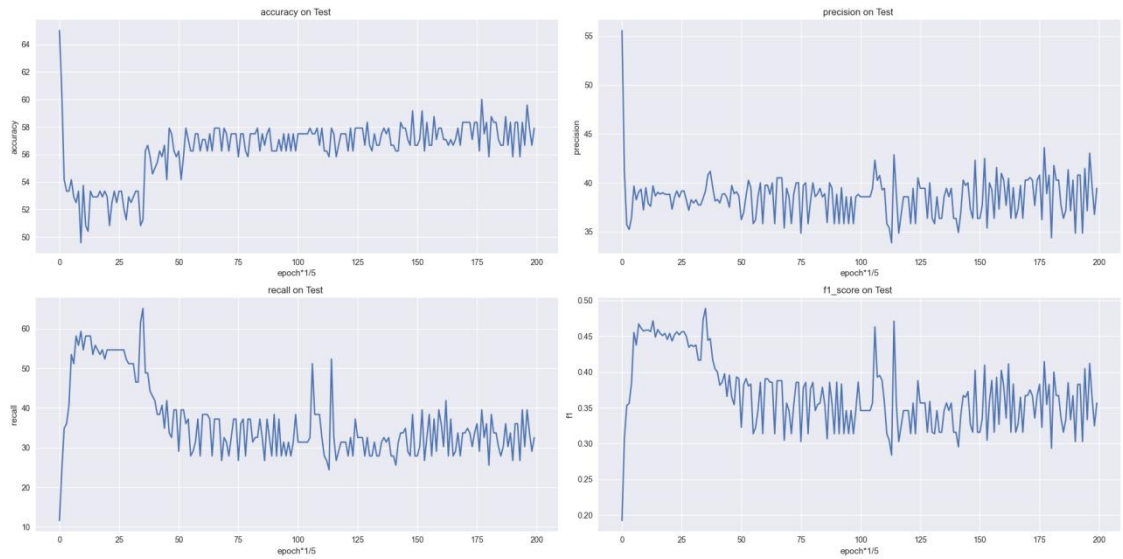


图 3-23 轮次(1000)-分类指标

根据图 3-19 至 3-23 可知，将数据输入设计好的 Transformer 模型训练，来进行二分类的效果相比机器学习分类器略有提升。据图 3-19、图 3-20、图 3-22 得知，测试准确率在很早轮次就已达到最大，后随着训练 loss 减小基本不变，并且保持在 55%附近。据图 3-21 可知，存在过拟合的现象，测试损失值不降反增。据图 3-23 可知，精确率、召回率、f1score 等参数也基本不变且低于 50%。分析原因为两点：一是当前数据集样本不够丰富，数据集数量太少。二是当前模型对于该任务结构

较为复杂。为了解决这个问题，在不更改模型结构的情况下，对数据集进行改进，使用数据增强的方法<sup>[23]</sup>。

数据增强通常包括加噪、换序、缩放、翻转、幅度扭曲、时间扭曲等操作，其中换序、幅度扭曲、时间扭曲等操作会对原有序列结构上改变，破坏原序列时间上的前后关系特征，如图 3-24 至图 3-27 所示。

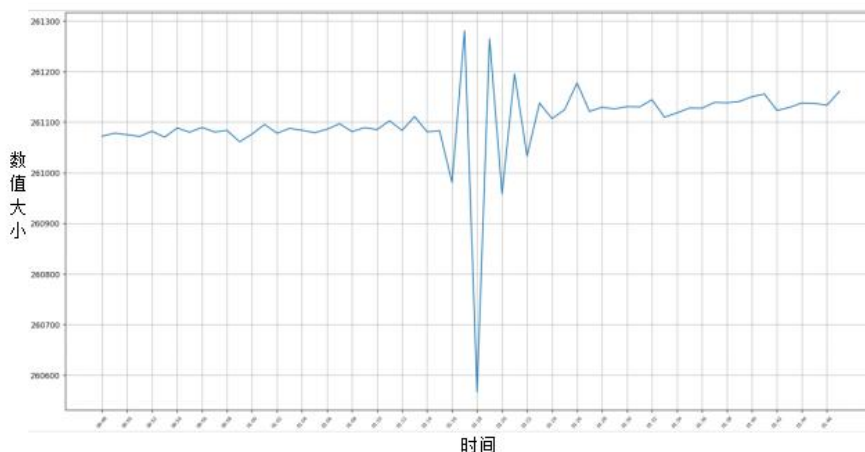


图 3-24 数据增强前时间序列波形

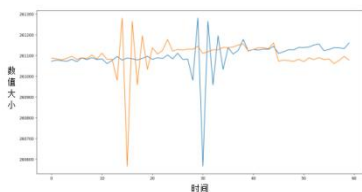


图 3-25 换序

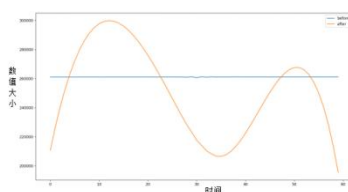


图 3-26 幅度扭曲

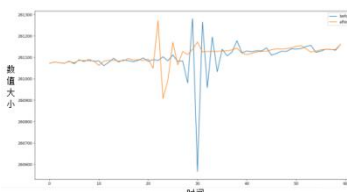


图 3-27 时间扭曲

由图 3-25 至图 2-27 可知，对于换序、幅度扭曲、时间扭曲的操作，原本的地震序列将变为非地震序列，应归为负样本。而其余的加噪、缩放、翻转等操作对于原本序列并无结构上改变，都体现在整体幅值上，因此原本样本所属标签保持不变。后再次进行实验，总的训练样本变为 10549 条，时间步长 40，特征步长 4，正负样本比例为 1: 1.1。总的测试样本变为 2639 条，正负样本比例为 1: 1。训练测试集 8: 2。训练轮次为 500，batchsize 为 512，学习率 lr 为 1e-4，得到轮次-训练损失、轮次-训练准确率/测试准确率、轮次-训练平均损失、轮次-测试平均损失（每隔 5 轮绘制）、轮次-精确率、轮次-召回率、轮次-f1score 等曲线如图 3-28 至图 3-32 所示。

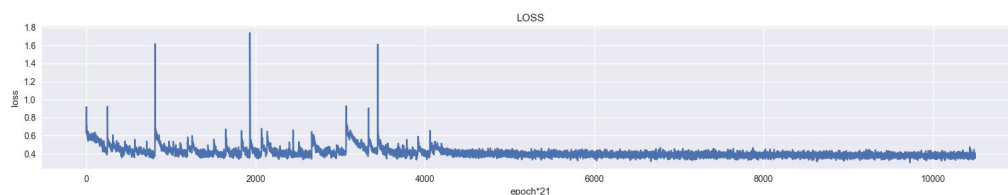


图 3-28 轮次(500)-训练损失

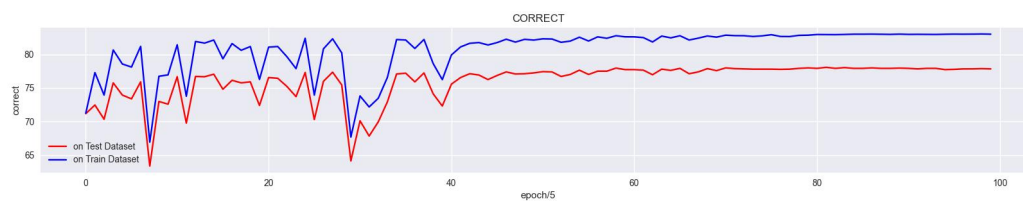


图 3-29 轮次(500)-训练/测试准确率

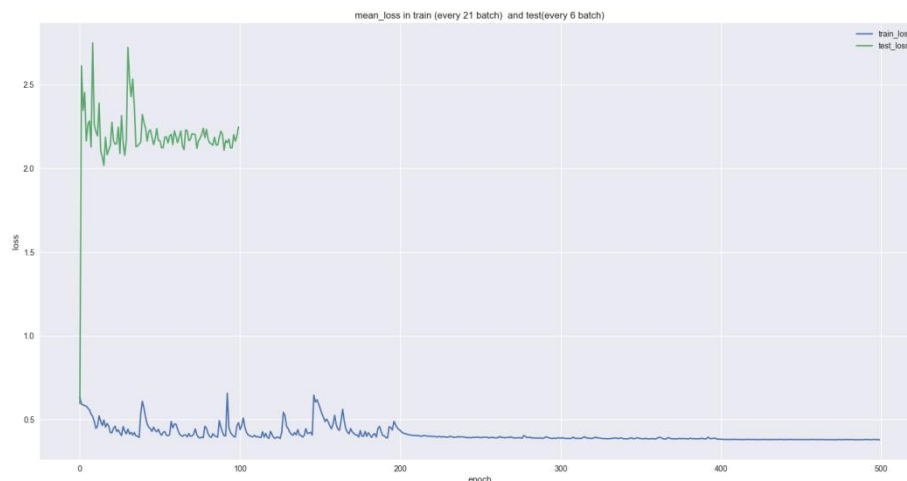


图 3-30 轮次(500)-训练/测试平均损失

最小loss: 0.3017188608646393    最小loss对应的epoch数:464    最后一轮loss:0.3482274115085602  
 最大correct: 测试集:78.06% 训练集:83.03%    最大correct对应的已训练epoch数:410    最后一轮correct: 77.83%  
 d\_model=512    q=8    v=8    h=8    N=8    drop\_out=0.2  
 共耗时238.4分钟

图 3-31 训练信息图

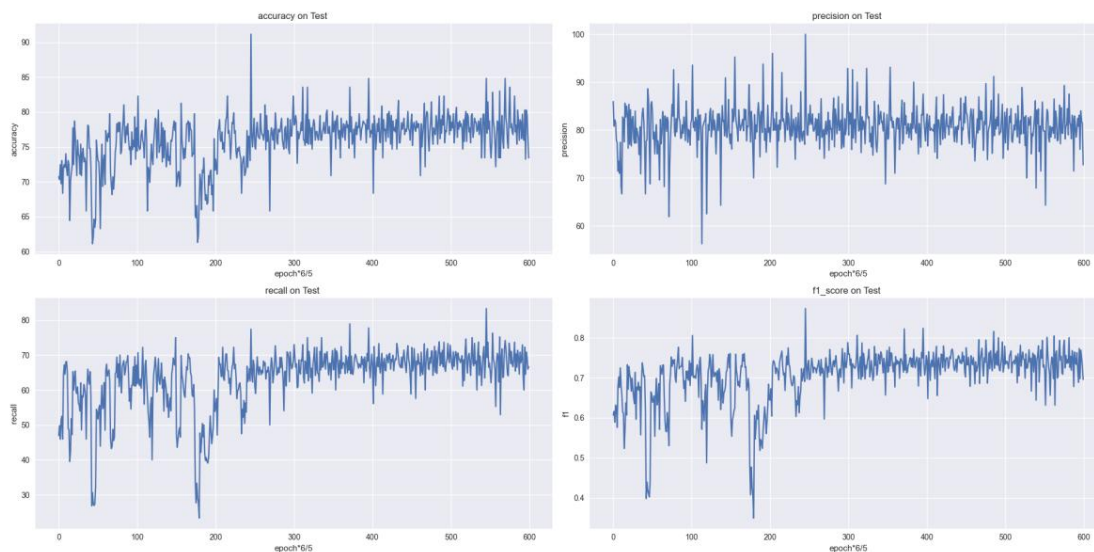


图 3-32 轮次(500)-分类指标图



从图 3-29 与图 3-31 可知,测试准确率明显得到提升,保持在了 78%,并且从图 3-30 可知测试 loss 曲线(每隔 5 轮绘制)随着轮次不再上升,和训练 loss 曲线一样保持下降趋势。同时从图 3-32 分类指标来看,精确率、召回率、f1score 都较之前明显的提升,精确率稳定在 90%,召回率稳定在 75%,f1 综合指标稳定在 0.8。从图 3-28 至图 3-32 整体可知,通过数据增强的手段可以提高模型识别的效果,即使效果并未达到最好,但一定程度上说明在保证数据集丰富度的前提下该 Transformer 模型具备获取多维时间序列时间空间特征的能力,能够通过输入多维时序应力数据来进行地震方位的分类预测。

### 3.3.3 Transformer 地震方位预测模型的训练超参数调试

为了训练出较好的模型,需要调节好模型训练相关的超参数在深度学习神经网络中,可以从两个方面来看。一是网络设计相关的参数,包括神经网络的网络层数、不同层的类别和搭建顺序、隐藏层神经元的参数设置、LOSS 层的选择、正则化参数和训练过程相关的参数,包括学习率使用策略、迭代次数、数据批尺寸 batchsize 的大小、输入数据的结构、优化器的选择等。

本节将以训练过程相关参数的角度出发,进行实验,寻找该问题下较好的学习率、批尺寸、优化器、轮数以及输入数据的结构。

#### 1、实验数据准备

选取“地震区域划分”部分所划定的区域,区域参数为东经  $96^{\circ}$  至东经  $107^{\circ}$ ,北纬  $33^{\circ}$  至北纬  $41^{\circ}$ ,自 2010 年 1 月 1 日至 2017 年 1 月 1 日,震级为震级 3 至震级 10,深度 0-1000 米。使用的应力数据为海原台自 2010 年 1 月 1 日至 2017 年 1 月 1 日的四个方向数据。在所划分区域的地震目录中包含 240 条地震,根据地震目录的时间找到应力数据中对应的时间节点,并相应地取该节点前 10 分钟后 30 分钟数据共 40 条,将其合成一条数据。数据每条时间步长是 40,表示共 40 分钟数据。特征为 4,表示四个方向特征。训练集测试集比例为 8:2。训练输入数据大小为 (192, 40, 4),输出类别数量为 9,类别从 0~9。每条数据所属类别由“地震区域划分”部分 kmeans 聚类所确定。

#### 2、寻找最优初始学习率

学习率决定了在每步参数更新中,模型参数有多大程度的调整。学习率的调整在模型训练中非常重要,学习率太小,则收敛得慢。学习率太大,则损失会震荡甚至变大。而对于训练过程中,通常会先设置一个初始的学习率,使得损失尽可能快地降低,经过一定轮次后再按照一定的 scheduler 降低学习率。因此,设置初始学习率的方式就尤为重要。初始学习率的范围一般在  $10^{-6}$  量级到 1 量级之间,为了更高效地训练出最好的模型,需要实验找到最优的初始学习率。

具体操作是<sup>[24]</sup>：以学习率为自变量，在单个轮次内，随着迭代的次数增加，学习率从小到大的增加，并画出损失随学习率变化的曲线，选择损失下降最快以及损失爆炸前的作为初始学习率。

改变 batchsize 从 4 到 32 获得 lr-loss 曲线如图 3-33 至图 3-36 所示。

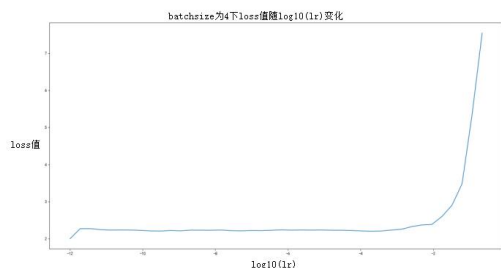


图 3-33 lr 量级-loss(batchsize=4)

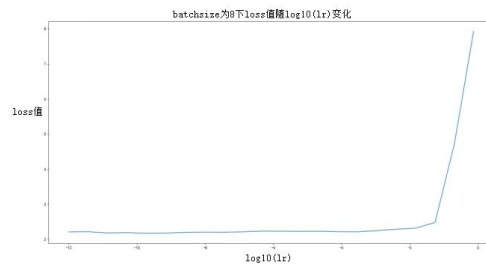


图 3-34 lr 量级-loss(batchsize=8)

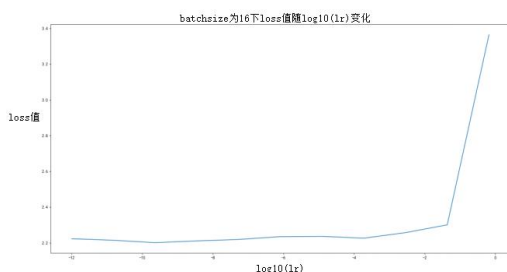


图 3-35 lr 量级-loss(batchsize=16)

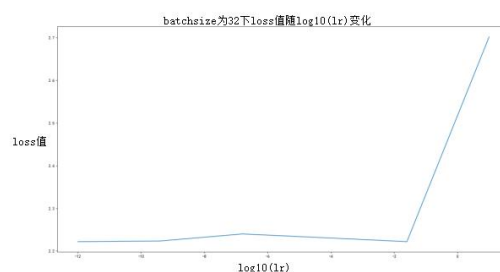


图 3-36 lr 量级-loss(batchsize=32)

由图 3-33 至图 3-36 可以看到不同 batchsize 情况下，lr-loss 曲线中 loss 值都基本在 -2~-3 量级开始平缓甚至上升，为了使得 loss 值保持稳定且下降的趋势，因此应该选取 -4 量级作为初始的学习率量级。

### 3、寻找最优批大小(batchsize)

批尺寸 batchsize 表示模型一次性读入多少批量的数据，它决定的是模型训练时下降的方向。当选的太小时，容易修正方向导致不收敛，或者需要经过很大的轮次才能收敛。当选的太大时，显卡的显存可能因此不够，还可能会因为迭代次数的减少而造成参数修正变的缓慢。

考虑到学习率与 batchsize 具有相互影响的关系，不能只调节 batchsize 不调节学习率。因此为了获取当前任务最优的批尺寸，在七种不同初始学习率基础上，选取五种不同的 batchsize 进行实验。学习率分别为  $5e-6$   $1e-5$   $2.5e-5$   $5e-5$   $1e-4$   $2e-4$   $1e-3$ 。batchsize 分别为 16、32、64、128、256。训练轮次为 2000 轮，优化器选择 Adam。最终将不同 lr 与 batchsize 下相应的准确率绘制如表 3-2 所示。表格中括号表示的是横向排名，准确率越高排名越前：

表 3-2 不同 lr、batchsize 下测试准确率及其排名(2000 轮)

Batchsize \ Lr	16	32	64	128	256
5e-6	47.92 (1)	43.75 (2)	39.58 (3)	37.50 (4)	39.58 (3)



1e-5	56.25 (2)	50.00 (3)	68.75 (1)	68.75 (1)	47.92 (4)
2.5e-5	43.75 (3)	41.67 (4)	45.83 (2)	47.92 (1)	41.67 (4)
5e-5	64.58 (4)	72.92 (1)	70.83 (2)	66.67 (3)	66.67 (3)
1e-4	75.00 (2)	68.75 (4)	87.50 (1)	72.92 (3)	72.92 (3)
2e-4	43.75 (3)	43.75 (3)	47.92 (1)	47.92 (1)	45.83 (2)
1e-3	27.08 (1)	20.83 (2)	27.08 (1)	18.75 (3)	16.67 (4)

根据表 2, 从学习率的角度来看, 当学习率逐渐增大时, 准确率是先升高后下降的趋势, 在  $lr$  为  $1e-4$  左右整体的准确率较高。从  $batchsize$  角度, 当  $batchsize$  逐渐增大时, 往往中间的  $batchsize$  如 32、64、128 的准确率更高。而表格中当  $lr$  为  $1e-4$ ,  $batchsize$  为 64 时准确率甚至达到 87.5%, 并且  $batchsize$  为 64 时的准确率相比其他  $batchsize$  情况下排名都靠前。因此整体来看,  $batchsize$  为 64 时比较适合。

不同  $lr$  下训练 loss 值随轮次变化如图 3-37 至图 3-40 所示。

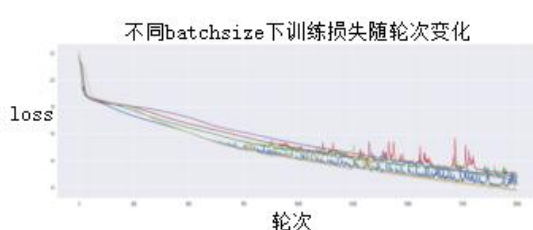


图 3-37 轮次-损失值( $lr=5e-6$ )

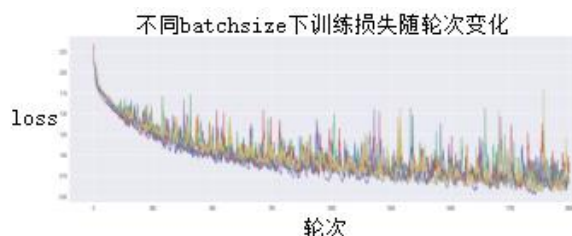


图 3-38 轮次-损失值( $lr=5e-5$ )

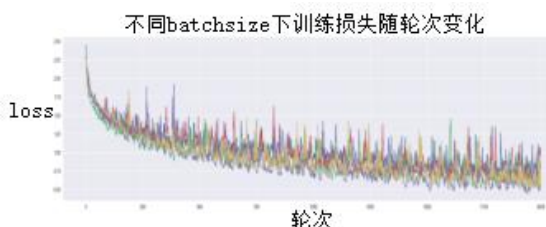


图 3-39 轮次-损失值( $lr=1e-4$ )

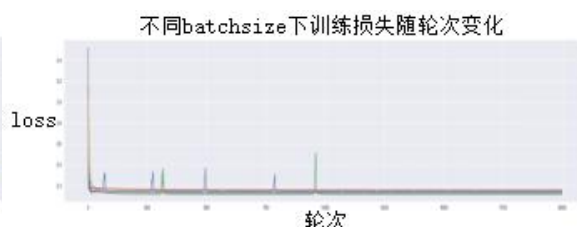


图 3-40 轮次-损失值( $lr=1e-3$ )

从图 3-37 至图 3-40 可知, 随着  $lr$  增大,  $loss$  值曲线下下降斜率越来越大, 下降的速度越来越快。当  $lr=1e-3$  时,  $loss$  值在短时间内下降到 2.2 左右并不再下降, 说明学习率此时已经不适宜, 这也从另一方面证实了上述初始学习率选择  $1e-4$  的合理性。

#### 4、寻找最优优化器

目前主流的五大优化器有 SGD、SGDM、Adagrad、RMSProp、Adam<sup>[25]</sup>。SGD 随机梯度下降优化器, 它每次选择一个 mini-batch, 而不是全部样本, 使用梯度下降来更新模型参数。其优点是解决了随机小批量样本的问题, 缺点是存在学习率自适应、易卡在梯度较小点等问题。SGDM 为加入了动量机制的 SGD, 相比于 SGD 的差别就在于, 参数更新时受当前迭代的梯度, 和之前迭代的累积梯度影响。其优点是相比 SGD 参数的更新可以保持之前的趋势, 避免卡在当前梯度最小点, 缺

点是仍存在学习率自适应更新的问题。Adagrad 相比前面 SGD 和 SGDM，它利用迭代次数和累积梯度，对学习率进行自动衰减。刚开始迭代学习率较大可以快速收敛，而后逐渐减小，使模型达到最优点。但其缺点是未考虑迭代衰减。若刚开始的梯度特别大，而后面的比较小，则学习率基本不会变化。RMSProp 类似于 Adagrad，不同点在于加入了迭代衰减，解决了当迭代梯度过大，导致自适应梯度无法变化的问题。Adam 是 SGDM 和 RMSProp 的结合，它基本解决了之前提到的梯度下降的一系列问题，比如随机小样本、自适应学习率、容易卡在梯度较小点等问题。

为了找到更适合当前任务的优化器，分别利用这五种优化器进行实验，参数设定  $lr=1e-4$ 、 $batchsize=64$ ， $epoch=500$  进行实验，得到轮次-训练损失、轮次-测试损失、轮次-训练准确率、轮次-测试准确率如图 3-41 至图 3-44 所示。

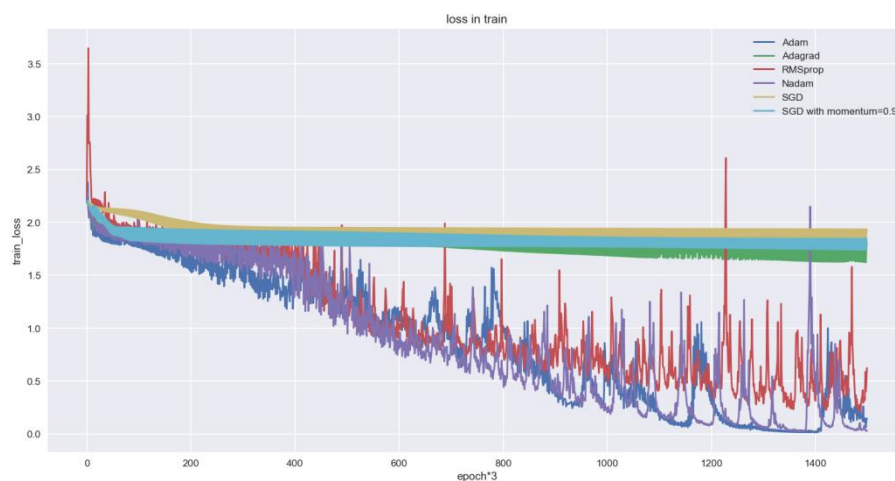


图 3-41 轮次(500)-训练损失

从图 3-41 可知 Adam、Nadam、RMSprop 在 500 轮次后还有继续收敛的趋势，loss 值一直下降接近 0，其中 Adam 和 Nadam 损失值下降的效果较好。同时 Adam 相较 Nadam 与 RMSprop 损失值下降较为稳定，损失值瞬间突变升高较少。而 SGD、SGD+momentum 与 Adagrad 在很早轮次 loss 值已保持稳定，达到收敛状态。

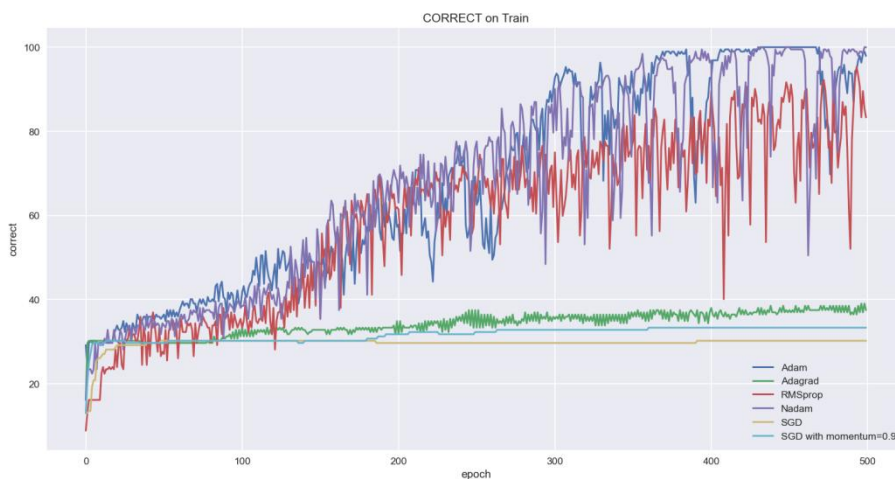


图 3-42 轮次(500)-训练准确率

从图 3-42 可知, Adam、Nadam、RMSprop 三种优化器训练准确率较高, 其中 Adam、Nadam 训练准确率接近 100%, 也对应着前面损失值曲线。

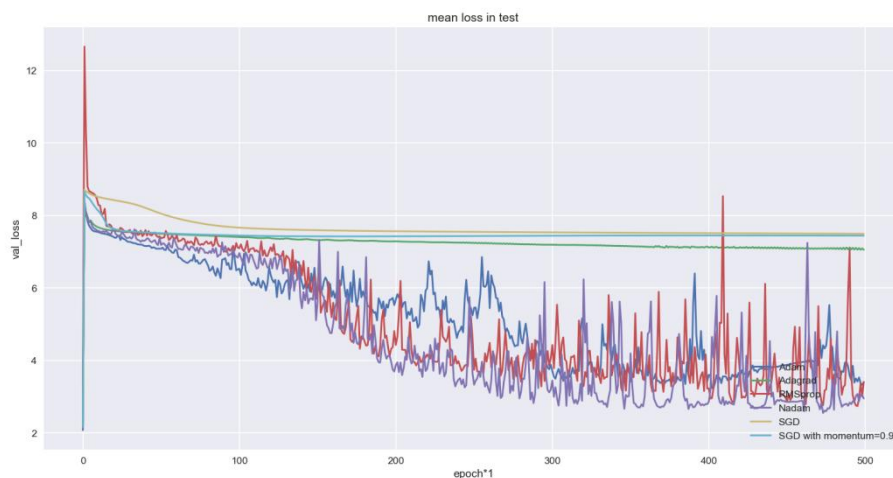


图 3-43 轮次(500)-测试损失

从图 3-43 可知, 依旧是 Adam、Nadam、RMSprop 三种优化器损失值一直下降趋近收敛状态。而剩下几种优化器损失值一直保持较高状态。

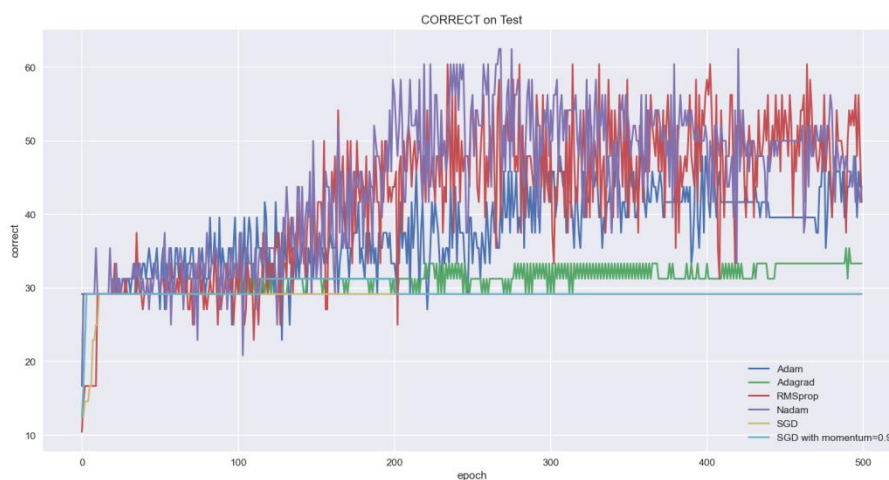


图 3-44 轮次(500)-测试准确率

从图 3-44 可知, 依旧是 Adam、Nadam、RMSprop 三种优化器效果较好, 其中 Nadam 与 RMSprop 优化器接近, 略优于 Adam。

由图 3-41 至图 3-44 可知, 整体考虑到训练阶段的收敛速度、损失值以及测试阶段准确率, 选出适合于当下任务的优化器 Adam。

### 5、训练最优模型

学习率、batchsize、优化器找到最优后, 进行训练, 轮次为 20000, 轮次-损失值、轮次-训练/测试准确率、轮次-训练/测试平均损失值图如图 3-45 至图 3-46 所示。

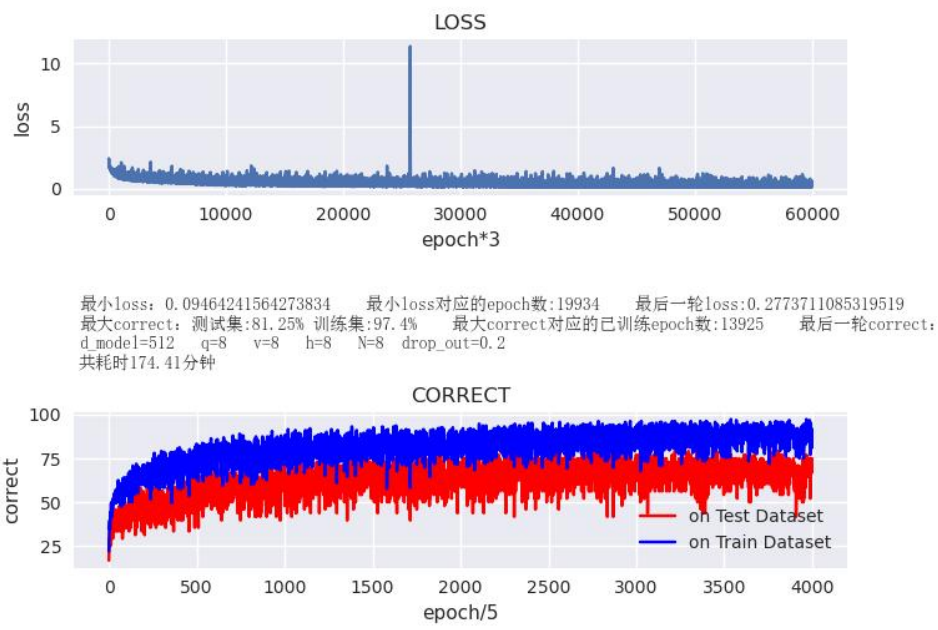


图 3-45 轮次(20000)-训练损失/训练测试准确率

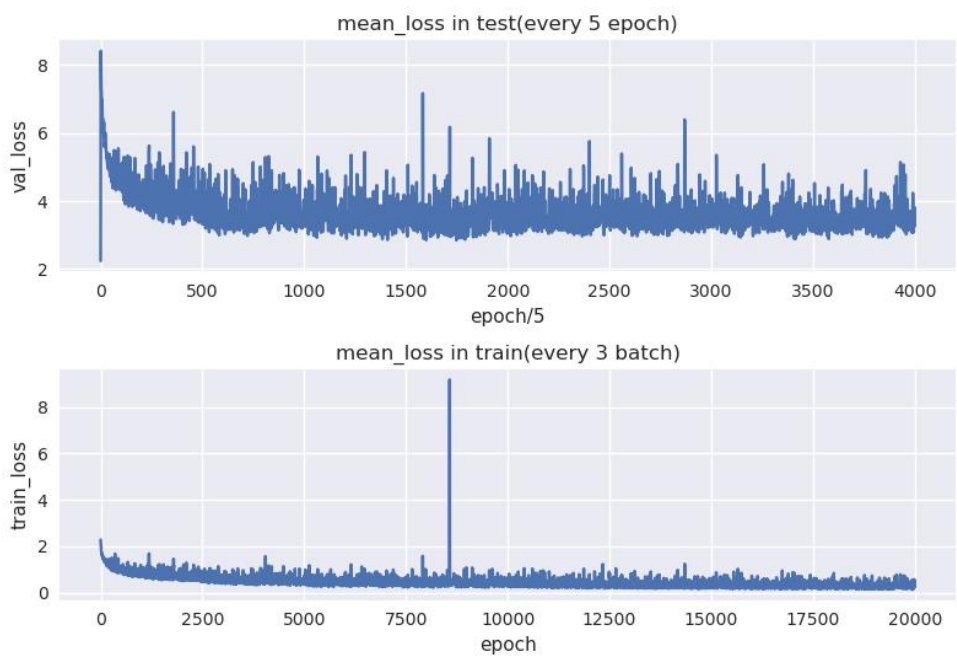


图 3-46 轮次(20000)-训练/测试平均损失

从图 3-45 至图 3-46 可知，在足够大的轮次配合上该类超参数下，训练准确率已经可以突破 95%，测试准确率也相应达到 81%。

为了进一步提升准确率，考虑到输入的数据中空间特征太少，只含有四个方向，为了使数据集结构更加复杂，后计算所选区域内各地震与临夏台的距离和角度添加至原数据集中，此时输入数据大小为（192，40，6），重复上述实验，并



配合学习率固定步长衰减策略（每隔 3000 轮衰减 0.1）获得轮次-损失值、轮次-训练/测试准确率、轮次-训练/测试平均损失值图结果如图 3-47 至图 3-48 所示。

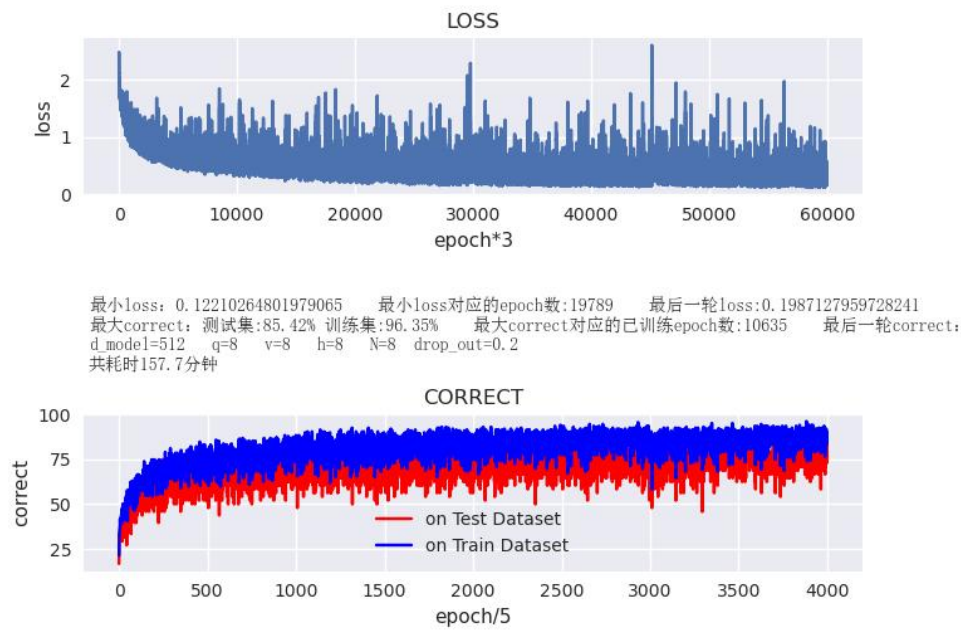


图 3-47 增加特征后轮次(20000)-训练损失/训练测试准确率

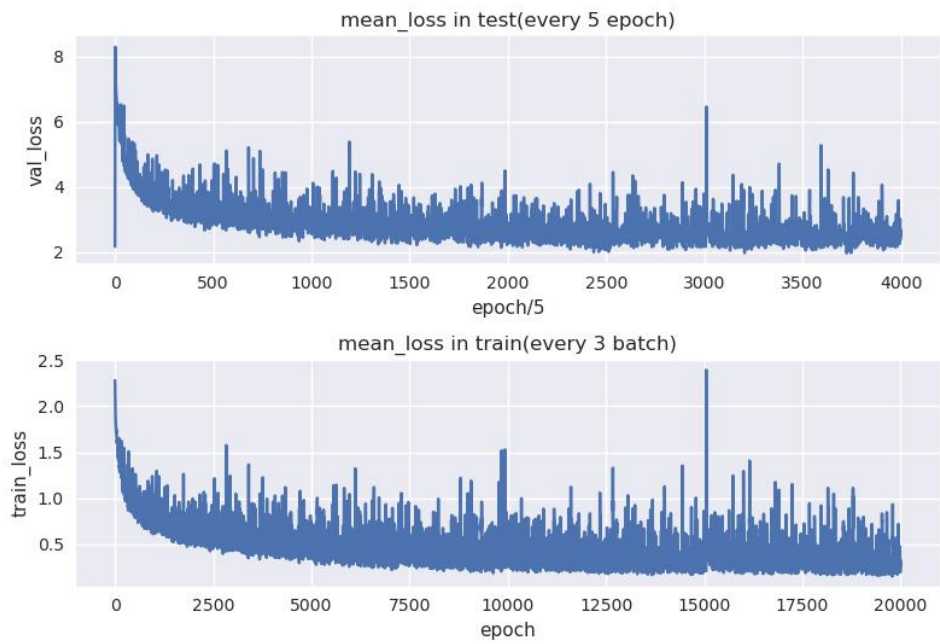


图 3-48 增加特征后轮次(20000)-训练/测试平均损失

从图 3-47 至图 3-48 可知，通过增加数据特征以及配合学习率衰减策略，模型的准确率也已经进一步提升。

因配合了学习率衰减策略，为了找到最优的衰减步长，多次改变步长进行实验，结果如表 3-3 所示。

表 3-3 不同衰减步长下最大测试准确率/训练最小损失值

指标 \ 步长	11000 1e-4~1e-5	7000 1e-4~1e-6	6000 1e-4~1e-7	5000 1e-4~1e-7	3500 1e-4~1e-9
最大测试准确率	85.42	89.58	87.5	87.5	85.42
训练最小 loss 值	0.00961	0.00089	0.12078	0.004629	0.02806

由表 3-3 可知, 当衰减步长为 5000 到 7000 之间, lr 衰减两到三次时测试准确率提升。且当步长为 7000 时测试准确率最大、训练 loss 值最小。使用衰减步长为 7000 后轮次-训练损失、轮次-训练/测试准确率、轮次-训练/测试平均损失如图 3-49 至图 3-50 所示。

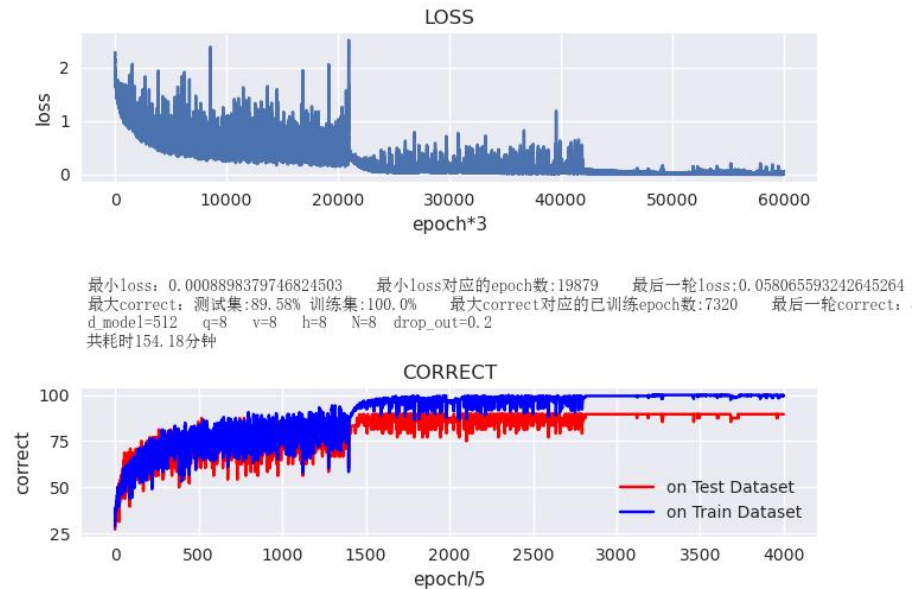


图 3-49 最优衰减步长下训练图

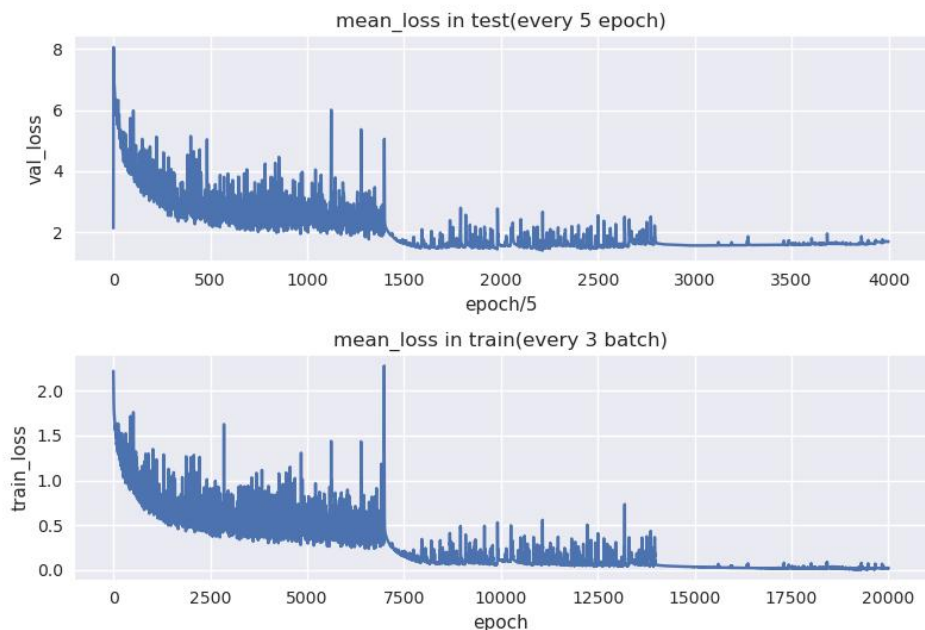


图 3-50 最优衰减步长下训练/测试平均损失

从地图上观察台站位置的分布，可知临夏台相比海原台位置更优越，四周环绕着较多区域，如图 3-51 所示。

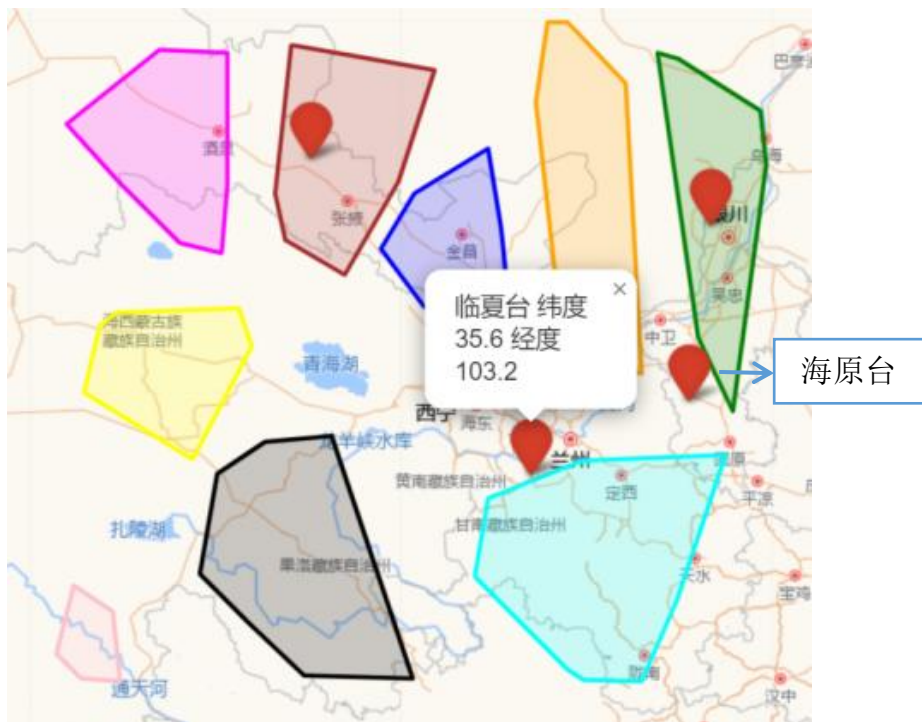


图 3-51 高德地图下区域台站显示

因此将上述海原应力数据换成临夏应力数据，添加距离方向特征，使用上述所得最优超参数， $lr=1e-4$ 、 $batchsize=64$ 、 $optimizer=Adam$ 、 $scheduler3000$  ( $lr$  衰减 3 次) 进行 10000 轮训练，结果如图 3-52 至图 3-53 所示。

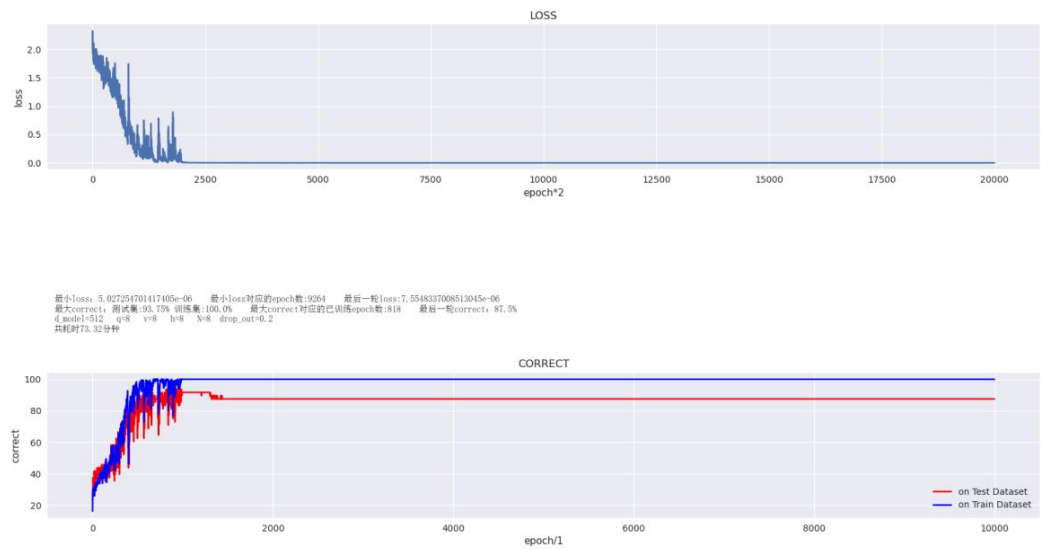


图 3-52 临夏台最优模型训练图

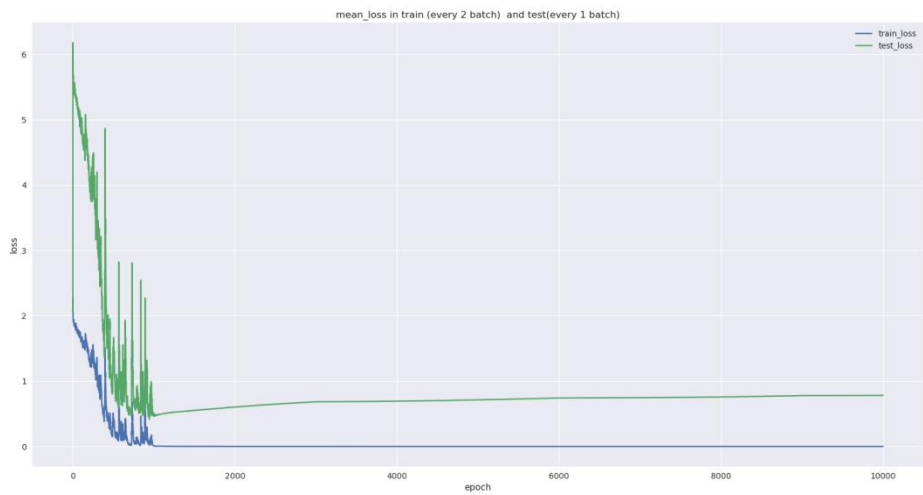


图 3-53 临夏台最优模型训练/测试平均损失图

由图 3-52 至图 3-53 可知，此时模型的测试准确率已经达到 93.75%，训练中模型也达到收敛。将该区域内 3~10 级地震 240 条数据输入模型，大小为(240,40,6)，获取模型输出的多分类指标如混淆矩阵、精确率、召回率、f1score 如图 3-54 至图 3-55 所示。



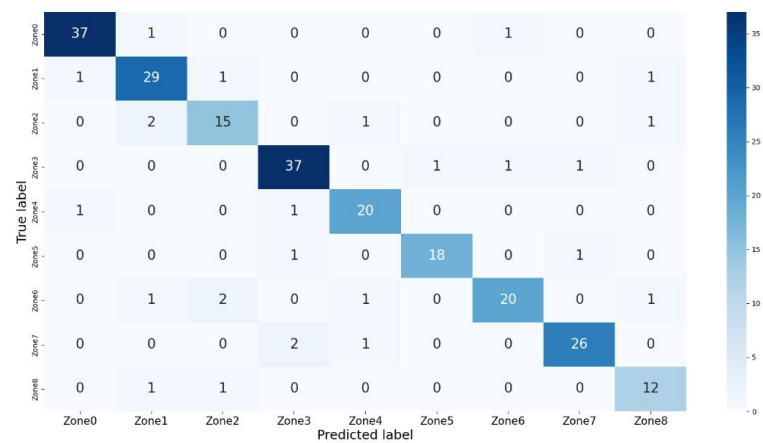


图 3-54 临夏台最优模型混淆矩阵

表 3-4 模型测试各分类指标

测试准确率	测试召回率	测试精确率	F1 值
89.17%	89.17%	89.30%	0.8917

由图 3-54 以及表 3-4 可知，预测结果的精确率、召回率、f1score 都比较高，在 89%。尤其是召回率。对于地震预测，真正关心的其实是实际发生的地震中模型预测对的数量，所以召回率的效果好一定程度上体现了模型的效果。

表 3-5 模型对各区域预测准确率

区域	区域 1	区域 2	区域 3	区域 4	区域 5	区域 6	区域 7	区域 8	区域 9
类别样本数	39	32	19	40	22	20	25	29	14
类别预测正确数量	37	29	15	37	20	18	20	26	12
类别预测准确率	94.87	90.62	78.95	92.5	90.91	90.0	80.0	89.6	85.71

由表 3-5 可知，模型对于区域 1、2、4、5、6、8 的预测准确率高，都在 90% 及以上，其中区域 1、2、4、8 的原始样本数量都在 25 以上。而区域 3、7、9 的预测准确率相比之下较低，其中 3、9 样本数量都在 20 以下，其中区域 3 的预测准确率最低，为 78.95%。区域 3、7、9 可视化方位如图 3-56 所示。

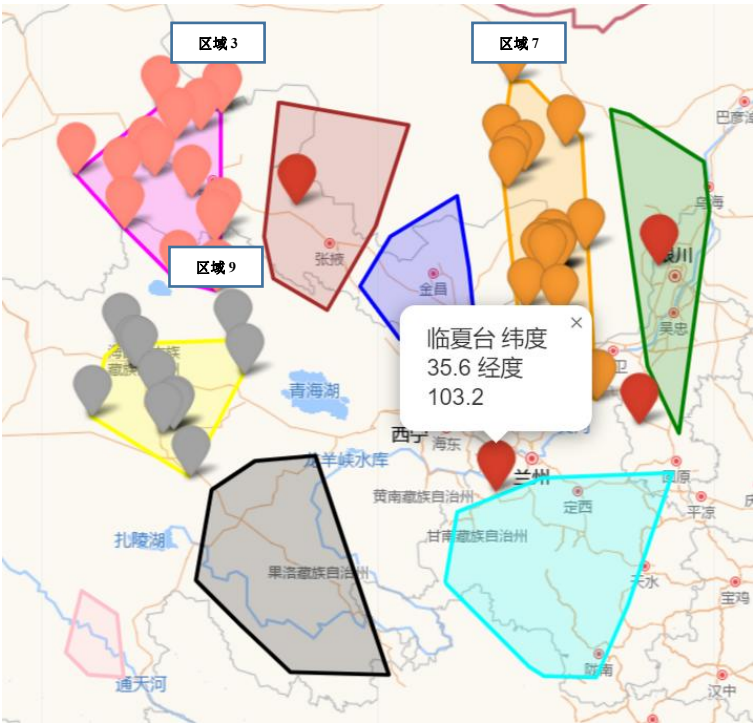


图 3-55 区域 3、7、9 方位

由图 3-55 可知，区域 3、9 方向位于临夏台西部与西北部且距离较远，并且 3、9 的原始数据量都不够多。总结得出：模型训练样本各类别的数量会影响模型对该类别的预测准确率，并且模型对距台站较远的区域的地震预测效果较差。

3.3.4 Transformer 地震方位预测对比验证

上节说明了如何调节各种训练参数获得较好的模型，本节将要进一步探究模型好坏是否与训练数据中震级程度相关。假设为：若数据中震级程度越高，应力数据在地震节点的变化越明显，那么模型也就更容易学习到其特征获得更好的效果。

在所研究区域下将样本震级划为五个范围：2-10、3-10、3.5-10、4-10、4.5-10，分别在每种震级数据集下，改变学习率和批大小进行多次实验。

表 3-6 2~10 震级下改变 Lr、Batchsize 得到模型测试准确率实验

Batchsize \ Lr	128	256	512
5e-6	36.94	35.99	37.83
1e-5	40.13	38.54	43.95
5e-5	48.73	57.64	51.59
1e-4	49.04	56.59	53.82

由表 3-4，2~10 震级下实验，样本数量：1570 训练：1256 测试：314

训练标签各类别出现个数: {1: 149, 2: 104, 3: 101, 4: 271, 5: 142, 6: 152, 7: 77, 8: 180, 9: 80}

测试标签各类别出现个数: {1: 47, 2: 26, 3: 23, 4: 76, 5: 43, 6: 38, 7: 19, 8: 33, 9: 9}

最大测试准确率: 57.64 平均: 45.91

表 3-7 3~10 震级下改变 Lr、Batchsize 得到模型测试准确率实验

Batchsize \ Lr	32	64	128	256
5e-6	43.75	39.58	37.5	39.58
1e-5	50.0	68.75	68.75	47.92
5e-5	72.92	70.83	66.67	66.67
1e-4	68.75	87.5	72.92	72.92

由表 3-5, 3~10 震级下实验, 样本数量: 240 训练: 192 测试: 48

最大测试准确率: 87.5 平均: 60.94

表 3-8 3.5~10 震级下改变 Lr、Batchsize 得到模型测试准确率实验

Batchsize \ Lr	16	32	64	128
5e-6	40.0	36.0	44.0	36.0
1e-5	48.0	40.0	48.0	32.0
5e-5	48.0	48.0	48.0	44.0
1e-4	52.0	36.0	48.0	48.0

由表 3-6, 3.5~10 震级下实验, 样本量: 125 训练 100 测试 25

训练标签各类别出现个数: {1: 22, 2: 9, 3: 11, 4: 4, 5: 8, 6: 14, 7: 8, 8: 9, 9: 15}

测试标签各类别出现个数: {1: 3, 2: 2, 3: 7, 5: 2, 6: 1, 7: 2, 8: 5, 9: 3}

最大测试准确率: 52.0 平均: 43.5

表 3-9 4~10 震级下改变 Lr、Batchsize 得到模型测试准确率实验

Batchsize \ Lr	8	16	32	64
5e-6	57.14	35.71	50.0	42.86
1e-5	57.14	57.14	57.140	42.86
5e-5	64.29	71.43	71.43	57.14
1e-4	57.14	57.14	64.29	57.14

由表 3-7, 4~10 震级下实验, 样本量: 67 训练 53 测试 14

训练标签各类别出现个数: {1: 6, 2: 8, 3: 13, 4: 4, 5: 6, 6: 3, 7: 3, 8: 5, 9: 5}

测试标签各类别出现个数: {1: 2, 2: 1, 3: 3, 4: 2, 5: 4, 6: 2}

最大测试准确率: 71.43 平均: 56.25

表 3-10 4.5~10 震级下改变 Lr、Batchsize 得到模型测试准确率实验

Batchsize \ Lr	4	8	16	32
5e-6	85.71	57.14	71.43	71.43
1e-5	85.71	85.71	71.43	85.71
5e-5	85.71	85.71	85.71	85.71
1e-4	85.71	85.71	85.71	85.71

由表 3-8, 4.5~10 震级下实验, 样本量: 35 训练 28 测试 7

训练标签各类别出现个数: {1: 9, 2: 5, 3: 4, 4: 1, 5: 2, 6: 2, 7: 1, 8: 2, 9: 2}

测试标签各类别出现个数: {1: 3, 2: 1, 3: 1, 4: 1, 8: 1}

最大测试准确率: 85.71 平均: 81.25

综上表格, 总结出随着震级升高, 各种 lr、batchsize 参属下模型最大、平均准确率变化如表 3-9 所示。

表 3-11 不同震级范围下实验所得最大测试准确率及平均准确率

震级 \ 指标	2.5-10	3-10	3.5-10	4-10	4.5-10
最大测试准确率	57.64	87.5	52.0	71.43	85.71
平均准确率	45.91	60.94	43.5	56.25	81.25

由表 3-9 可知, 随着震级升高, 数据减少, 模型一定程度上可以获得更好的效果, 但两者不属于线性关系。

得到的结论是: 当模型的训练数据所在的震级越集中越显著时, 模型的训练效果会得到提升。

### 3.4 本章小结

本章介绍了基于 Transformer 地震预测分析过程。首先介绍了数据的预处理, 包括时间戳处理、四分位法找出极端值并置空、滑动窗口去噪。之后利用 Kmeans 聚类算法, 通过比对多个指标找出最好的分类数为 9 类来进行地震区域划分。有了数据和方位的先决条件后, 为了验证应力数据特征的可学习性, 分别进行机器学习与深度学习方法进行二分类任务。再利用该模型进行大量实验, 找出适合当前任务下模型最好的训练参数包括学习率、批大小、优化器、轮次、数据集结构、学习率衰减策略等, 从而获取最佳模型。最后通过改变应力数据中震级的范围进行训练来探究震级大小与模型预测准确率大小的相关性。

## 4 地震方位预测 UI 界面设计

### 4.1 编程库选择

Python 提供了多个图形开发界面的库,其中 Tkinter 与 TK 模块是 Python 常用 GUI 工具包的接口。TK 和 Tkinter 可以在大多数 Unix 平台下使用,同样可以应用在 Windows 系统里。其优点一是 API 比较稳定,版本更新较慢,程序不存在短期内无法使用的情况。二是简洁,适合小型应用的开发,其控件都较基础,不会过于复杂。

### 4.2 UI 界面的功能设计与实现

基于 Transformer 的地震预测 UI 界面的流程图如图 4-1 所示。

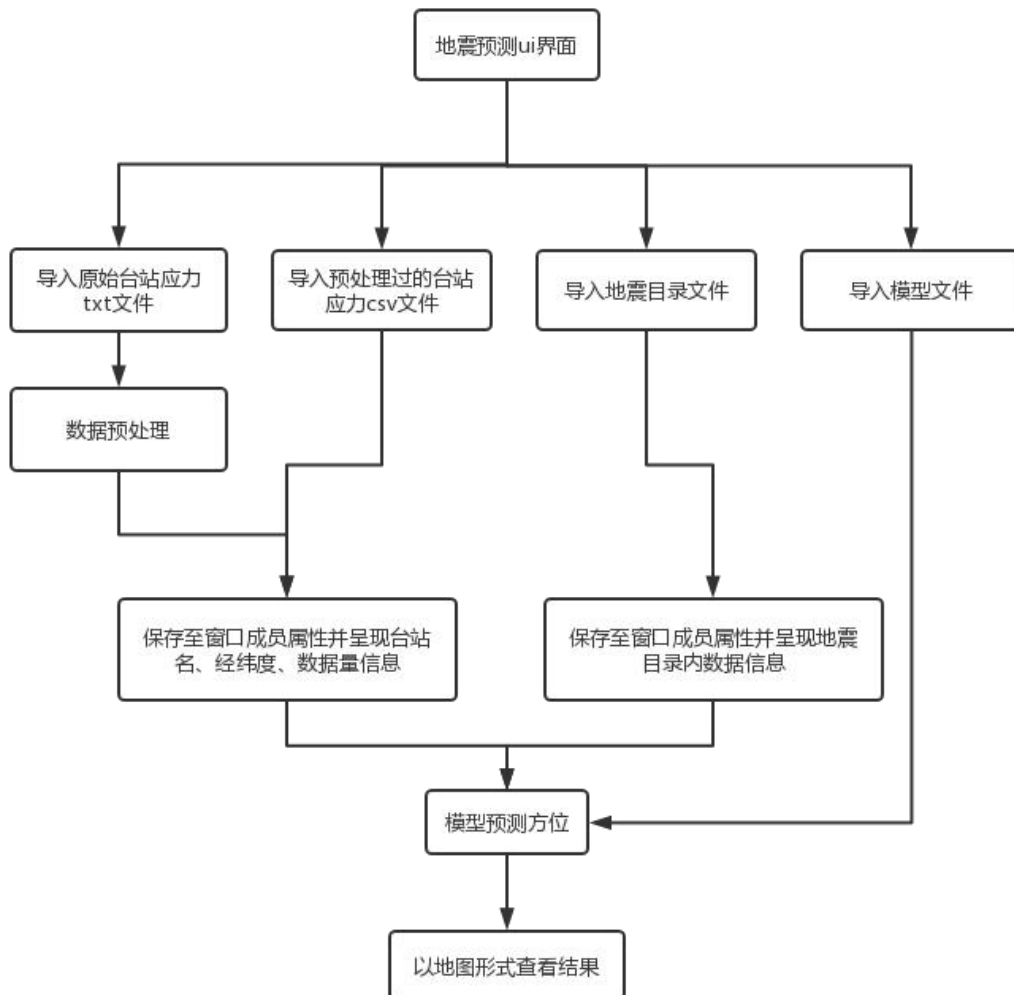


图 4-1 地震预测 ui 界面流程图

由图 4-1, 具体实现步骤为:

步骤一：导入台站应力数据、地震目录、模型文件，若台站应力数据为原始文本，则经过数据预处理后将处理后的台站应力数据的具体信息如导入时间、台站名、台站经纬度、台站数据量呈现在界面上。同理地震目录导入后将地震目录内导入时间、数据区域、经纬度范围、深度范围、地震网信息呈现在界面上。模型文件信息不呈现。

步骤二：若三种文件都已导入方可点击预测按钮进行地震预测，预测结果保留在网页的高德地图上，可自行选择查看。

具体设计思路：先利用 `tkinter` 自带的控件，将界面进行布局呈现。首先是菜单栏的控件布局，代码如图 4-2 所示。

```
# 创建菜单栏并绑定相应事件函数
self.main_menu = Menu(self.root)
self.main_menu.add_command(label="导入文件", command=lambda: self.upload_file())
self.main_menu.add_command(label="导入文本", command=lambda: self.upload_txt())
self.main_menu.add_command(label="导入模型", command=lambda: self.load_model())
self.main_menu.add_command(label="查看结果", command=lambda: self.get_result2(self.map_url))
self.main_menu.add_command(label="查看用法", command=lambda: self.watch_usage())
self.root.config(menu=self.main_menu)
self.main_menu.add_separator()
```

图 4-2 菜单控件布局

由图 4-2，代码实现的功能是利用 `tkinter` 的 `Menu` 控件依次添加子菜单。其中 `command` 表示的是绑定功能函数，菜单点击后触发。

主要信息框，其中又包含上下信息框，上信息框中包含台站文件导入信息框和地震目录文件导入信息框。下信息框包含地震目录具体信息框和台站具体信息框，而台站具体信息框中又包含四个具体的台站信息框。按钮则与主信息框并列排布，排布在主信息框下。整体的呈现结构图如图 4-3 所示。

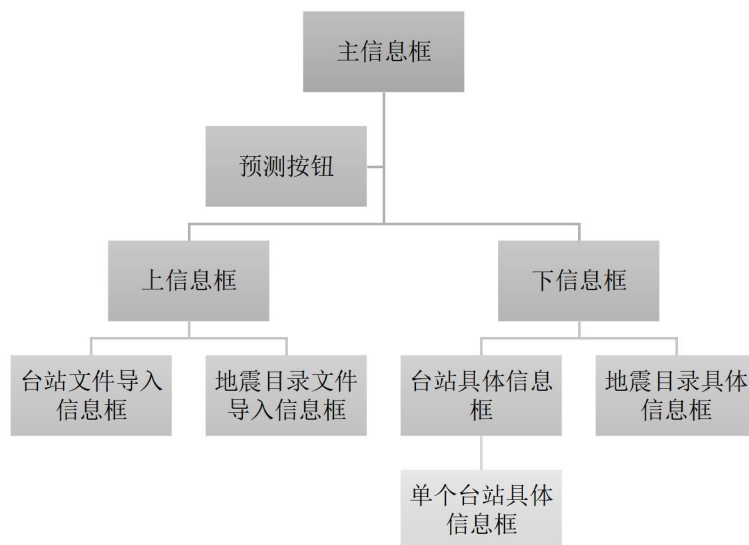


图 4-3 信息框结构

图 4-3 中每个子信息框中的内容，则通过各个标签实现，如图 4-4 所示。

```
# 下方框
self.details_frame = ttk.LabelFrame(self.mainFrame)
self.details_frame.grid(row=1)
self.summary_frame = ttk.LabelFrame(self.details_frame, text="地震目录")
self.summary_frame.grid(row=0, column=0, sticky="W")
self.station_frame = ttk.LabelFrame(self.details_frame, text='台站信息')
self.station_frame.grid(row=1, column=0, sticky="W")

# 地震目录信息
# 纬度
ttk.Label(self.summary_frame, text="纬度范围:").grid(column=0, row=0, sticky="E")
self.latEntry = ttk.Label(self.summary_frame, width=20, textvariable=self.lat)
self.latEntry.grid(column=1, row=0, sticky="W")
```

图 4-4 框内标签布局

由图 4-4，通过 ttk 的 LabelFrame 来实例化为 details\_frame 下方框，并指定父控件为主信息框。同时用 grid 方法进行布局，由此来确定位置。之后具体的信息内容通过 ttk 的 Label 来实例化，并指定父控件和布局。

网页地图的呈现，用到了 Python 绘制地图的第三方库 folium，是常用的可视化处理工具。要使用前提是获得地图基底，如图 4-5 所示。

```
m = folium.Map(location=centerLoc,
               zoom_start=5,
               control_scale=True,
               control=False,
               tiles=None
            )

folium.TileLayer(
    tiles='http://webrd02.is.autonavi.com/appmaptile?lang=zh_cn&size=1&scale=1&style=8&x={x}&y={y}&z={z}',
    attr="&copy; <a href=http://ditu.amap.com/>高德地图</a>",
    min_zoom=0,
    max_zoom=19,
    control=True,
    show=True,
    overlay=False,
    name=baseLayerTitle
).add_to(m)
```

图 4-5 获取底图

由图 4-5，folium.Map 表示得到一张地图，location 表示地图的中心位置、zoom\_start 表示比例尺、control\_scale 表示是否在地图上添加比例尺、tiles 表示底图样式。

区域的绘制，核心是绘制点再连线。区域的点可根据“地震区域划分”部分 Kmeans 聚类获得。但并不是所有的点都参与绘制，而是需要找到边界点。边界点寻找利用 scipy.spatial 中计算凸包的函数 convexHull，即寻找包围一组散点的最小凸边形。如图 4-6 所示。

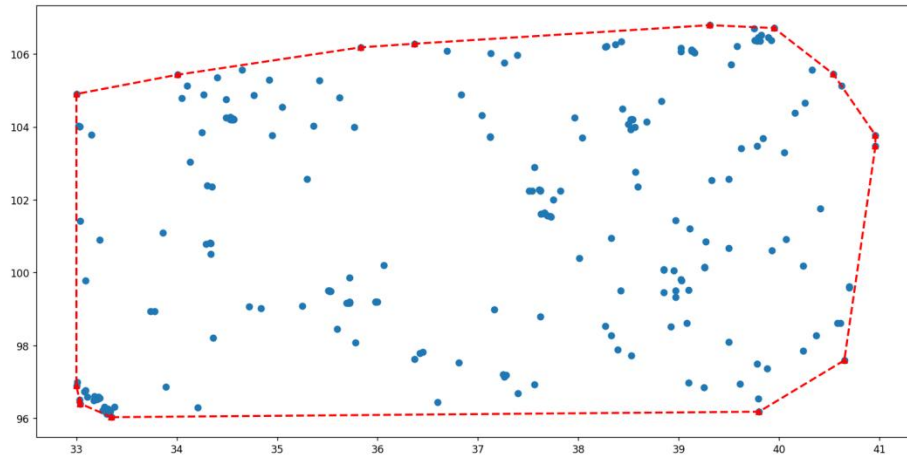


图 4-6 散点边界获取

由图 4-6 方法，找到边界点后用 `folium.Polygon` 方法即可通过点连成多边形从而形成区域。

### 4.3 地震定位 UI 界面应用

首先程序运行后界面如图 4-7 所示。



图 4-7 ui 运行界面

界面分为两大展示块，顶部块呈现的是台站文件数量、台站导入时间、地震目录条数、地震目录导入时间。

当导入地震目录、台站文件、模型文件后如图 4-8 所示。





图 4-8 文件上传后信息展示

当文件齐全后，点击“开始预测”按钮，预测完成后将会跳出网页，这里拿临夏台数据进行预测，预测前后结果如图 4-9 至图 4-10 所示。

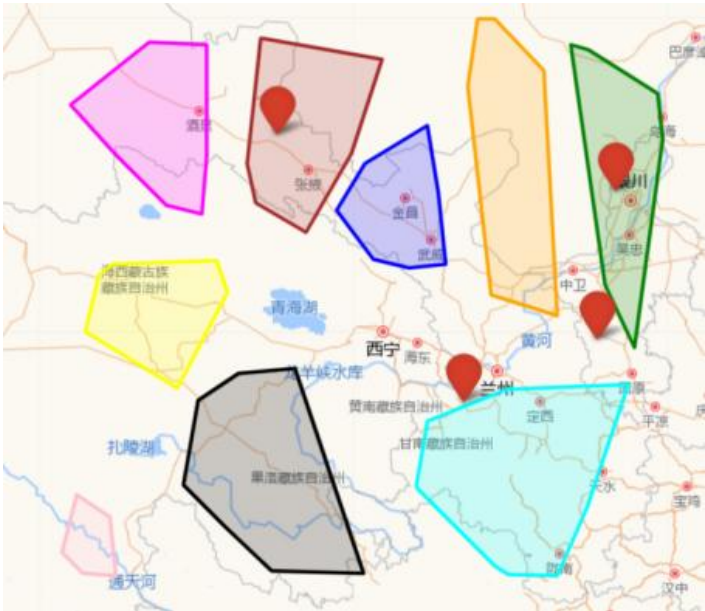


图 4-9 预测前

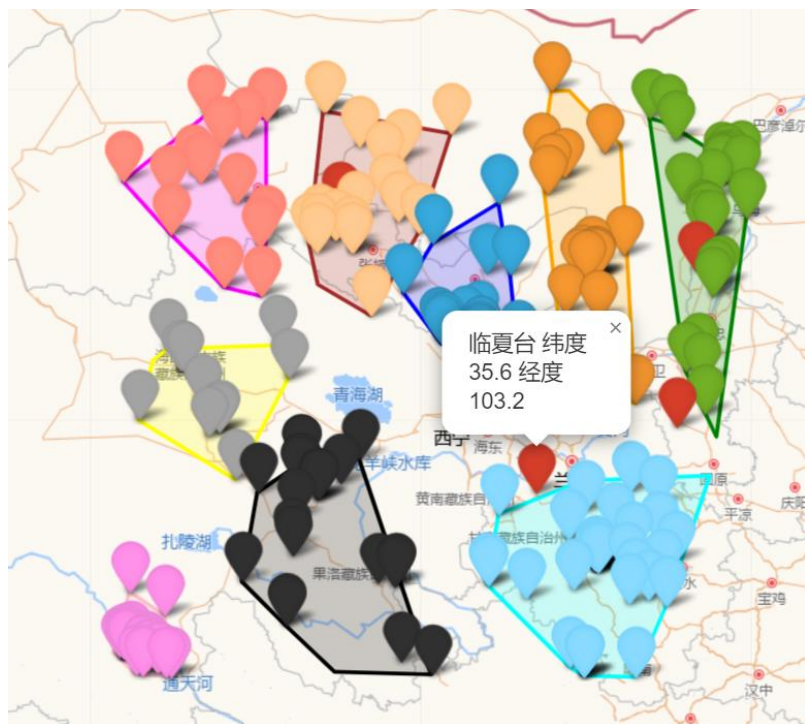


图 4-10 预测后

由图 4-9, 表示预测前九个区域以及四个台站分布。有图 4-10 表示在真实的地震中, 各区域所预测对的地震。

#### 4.4 本章小结

本章介绍了基于 Tkinter 的地震分类预测 UI 界面的结构、实现方式与功能。结合上一章节所训练出的最好的模型, 实现了完整的模型预测功能。

## 5 总结与展望

### 5.1 研究成果总结

本文通过总结学者们的经验,对当下广泛运用于 NLP 任务的 Transformer 模型进行改进,使之能够适应多维时间应力序列的分类任务,实现地震的分类预测。并且通过二分类、多分类等实验来探究训练最优模型所需的参数。最后通过 Python 内置的 Tkinter 库将整个分类过程将 UI 界面形式呈现出来,实现完整的地震方位预测流程。

首先从 Transformer 神经网络的基本原理展开,根据多维时序应力数据的特点,设计出用于编码时间和空间信息的编码器结构。时间编码器沿用 Transformer 原编码器结构,拥有输入编码、位置编码、带掩码机制的自注意力机制、前馈全连接层、残差结构。而因为时间序列的空间特征信息不存在前后位置关系,因此不需要添加位置编码以及带掩码的自注意力机制。对于两个编码器的输出,计算各自输出的权重占比再进行合并。

然后以 Transformer 网络为基础,介绍地震方位预测分析的流程。首先分析数据的预处理,包括时间戳处理、四分位法找出极端值并置空、滑动窗口去噪。之后分析地震区域的划分,利用 Kmeans 聚类算法,通过比对簇内误差平方和、轮廓系数、戴维斯-博尔丁指标,从而找出最好的分类数为 9 类,进而对区域进行划分。有了数据和方位的先决条件后,分别利用机器学习五种分类器,包括(朴素贝叶斯、逻辑回归、K 邻近、支持向量机、带核的支持向量机)与深度学习方法(改进后的 Transformer 结构)进行二分类任务,得到的结论是应力时序数据特征具有可学习性,能够用于 Transformer 的地震分类预测。再利用改进后的 Transformer 模型进行实验,找出适合当前任务下模型最好的训练参数包括初始学习率为( $1e-4$ )、批大小(64)、优化器(Adam)、轮次(10000)、数据集结构(原四个方向基础上加入距离和角度)、学习率衰减策略(scheduler7000)等,从而获取最佳模型,模型的准确率、召回率能达到 89%,能很好地完成区域分类预测地任务。模型对于训练样本中数量多的类别的分类预测准确率高,对于离台站较近区域的地震的预测准确率高。后通过改变应力数据中震级的范围进行训练来探究震级大小与模型预测准确率大小的相关性,得到的结论是随着训练应力数据震级程度越集中越显著,模型的训练效果会更好,一定程度上给训练更佳的模型带来指导。

最后介绍了基于 Tkinter 的地震预测 UI 界面的结构、实现方式与功能。结合上一章节所训练出的最好的模型,实现了完整的模型预测功能。

## 5.2 不足与展望

本文的不足之处在于模型训练的应力数据较少、所划定的区域单一不够大、数据只应用了单一台站，整体上数据集不够复杂。通过后续测试发现模型并不具备震级分类预测的普适性，模型对于所划定区域内的数据的预测性能高，但对于所划区域外的应力数据预测性能降低。

另一个是训练最佳模型时并未综合考虑网络结构参数和训练参数，仅选择了训练参数进行实验探究，考虑不够全面。

今后可以寻找更加适合于地震分类预测的 Transformer 结构，并尝试更多的调节训练参数和网络参数的方法将之结合，模型的训练数据丰富化，不仅局限于单区域单台站。

## 参考文献

- [1] 李思宇,吴忠良.海城地震预报作为一个科学史事件[J].自然辩证法通讯,2019,41(10):64-68.DOI:10.15994/j.1000-0763.2019.10.009.
- [2] 陆洲. 地壳应力应变变化与地球动力学[D].中国科学技术大学,2019.
- [3] 郑温冬. 面向时序数据关联特性的预测方法研究[D].湖南大学,2020.DOI:10.27135/d.cnki.ghudu.2020.003423.
- [4] Boucouvalas A C, Gkasios M, Tselikas N T, et al. Modified-Fibonacci-Dual-Lucas method for earthquake prediction[C]//Third International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2015). SPIE, 2015, 9535: 400-410.
- [5] Hayakawa M. Earthquake prediction with electromagnetic phenomena[C]//AIP Conference Proceedings. AIP Publishing LLC, 2016, 1709(1): 020002.
- [6] Akhoondzadeh M, Chehrebargh F J. Feasibility of anomaly occurrence in aerosols time series obtained from MODIS satellite images during hazardous earthquakes[J]. Advances in Space Research, 2016, 58(6): 890-896.
- [7] Korepanov V. Possibility to detect earthquake precursors using cubesats[J]. Acta Astronautica, 2016, 128: 203-209.
- [8] Hayakawa M, Yamauchi H, Ohtani N, et al. On the precursory abnormal animal behavior and electromagnetic effects for the Kobe earthquake (M~ 6) on April 12, 2013[J]. Open Journal of Earthquake Research, 2016, 5(03): 165.
- [9] Last M, Rabinowitz N, Leonard G. Predicting the maximum earthquake magnitude from seismic data in Israel and its neighboring countries[J]. PloS one, 2016, 11(1): e0146101.
- [10] Jozinović D, Lomax A, Štajduhar I, et al. Rapid prediction of earthquake ground shaking intensity using raw waveform data and a convolutional neural network[J]. Geophysical Journal International, 2020, 222(2): 1379-1389.
- [11] Huang Y, Han X, Zhao L. Recurrent neural networks for complicated seismic dynamic response prediction of a slope system[J]. Engineering Geology, 2021, 289: 106198.
- [12] Wang Q, Guo Y, Yu L, et al. Earthquake prediction based on spatio-temporal data mining: an LSTM network approach[J]. IEEE Transactions on Emerging Topics in Computing, 2017, 8(1): 148-158.
- [13] Münchmeyer J, Bindi D, Leser U, et al. Earthquake magnitude and location estimation from real time seismic waveforms with a transformer network[J]. Geophysical Journal International, 2021, 226(2): 1086-1104.
- [14] 吴国栋,宋福根,涂立静等.基于改进 CNN 的局部相似性预测推荐模型[J].计算机工程与科学,2019,41(06):1071-1077.

- [15]陈聪,候磊,李乐乐等.基于 GRU 改进 RNN 神经网络的飞机燃油流量预测[J].科学技术与工程,2021,21(27):11663-11673.
- [16]杨少波,刘道伟,安军等.基于长短期记忆网络的电网动态轨迹趋势预测方法[J].中国电机工程学报,2020,40(09):2854-2866.DOI:10.13334/j.0258-8013.pcsee.190570.
- [17]Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [18]Oh J, Wang J, Wiens J. Learning to exploit invariances in clinical time-series data using sequence transformer networks[C]//Machine learning for healthcare conference. PMLR, 2018: 332-347.
- [19]Rußwurm M, Körner M. Self-attention for raw optical satellite time series classification[J]. ISPRS journal of photogrammetry and remote sensing, 2020, 169: 421-435.
- [20]Jaderberg M, Simonyan K, Zisserman A. Spatial transformer networks[J]. Advances in neural information processing systems, 2015, 28.
- [21]Liu M, Ren S, Ma S, et al. Gated transformer networks for multivariate time series classification[J]. arXiv preprint arXiv:2103.14438, 2021.
- [22]Liu Y, Li Z, Xiong H, et al. Understanding of internal clustering validation measures[C]//2010 IEEE international conference on data mining. IEEE, 2010: 911-916.
- [23]Um T T, Pfister F M J, Pichler D, et al. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks[C]//Proceedings of the 19th ACM international conference on multimodal interaction. 2017: 216-220.
- [24]Smith L N. Cyclical learning rates for training neural networks[C]//2017 IEEE winter conference on applications of computer vision (WACV). IEEE, 2017: 464-472.
- [25]Wilson A C, Roelofs R, Stern M, et al. The marginal value of adaptive gradient methods in machine learning[J]. Advances in neural information processing systems, 2017, 30.

## 致谢

到这里，我的毕业论文就完成了，大学生涯也即将结束。回首过去四年，不禁感慨，这段时光给与了我太多。

首先要感谢我的毕设指导老师于紫凝老师，从选题到定稿，这其中的每一步都离不开您的悉心指导。每当我一筹莫展之际，您为我指点迷津、指引方向。每当我自信满满之时，您为我道出不足，发掘问题的本质。在撰写论文的过程中，也是您一遍又一遍地校对，对我的不足进行指正，以提高文章质量。您的耐心和对于学术的认真严谨，让我由衷地感激和敬佩。

同时也要感谢我的母校中国海洋大学，在这里获得的不仅是书本中的知识，更有生活中各方面的启发。感谢电子系的任课老师授予我专业知识，让我在未来的专业道路上有了些许认知。也感谢我的辅导员老师对我的关怀与帮助，感谢这里的一花一草、一水一木。与此同时还要感谢我的室友、朋友们，他们陪伴我成长，共度欢乐时光。

最后，我要感谢我的父母，感谢他们对我成长道路上一以贯之的指引、教导、包容、精神鼓励与物质支持。