# Transformer 模型在故障检测中的应用

## 简介

本项目中的 Transformer 模型主要用于工业过程中的故障检测，特别是在半导体制造过程（SECOM数据集）的异常检测任务中。Transformer 模型因其强大的特征提取和表示学习能力，被用于改进传统的统计过程控制方法（如 T² 和 SPE/Q 统计量）。

## 模型架构

项目中实现了多种基于 Transformer 的模型架构：

### 1. 基本 Transformer 自编码器 (TransformerAutoencoder)

这是最基础的实现，使用 Transformer 编码器提取特征，然后通过简单的线性层重建输入：

def \_\_init\_\_(self, input\_dim, hidden\_dim, nhead=4, num\_layers=2, dropout=0.1):  
 # 输入投影层  
 self.input\_linear = nn.Linear(input\_dim, hidden\_dim)  
   
 # 位置编码  
 self.pos\_encoder = PositionalEncoding(hidden\_dim)  
   
 # Transformer 编码器  
 encoder\_layers = nn.TransformerEncoderLayer(  
 d\_model=hidden\_dim,   
 nhead=nhead,   
 dim\_feedforward=hidden\_dim\*4,   
 dropout=dropout,  
 batch\_first=True  
 )  
 self.transformer\_encoder = nn.TransformerEncoder(encoder\_layers, num\_layers=num\_layers)  
   
 # 输出层（重建）  
 self.output\_linear = nn.Linear(hidden\_dim, input\_dim)

### 2. 增强版 Transformer 自编码器 (EnhancedTransformerAutoencoder)

针对 SPE 性能优化的改进版本：

def \_\_init\_\_(self, input\_dim, hidden\_dim, nhead=4, num\_layers=3, dropout=0.1):  
 # 增强型输入嵌入层（双层MLP）  
 self.input\_embedding = nn.Sequential(  
 nn.Linear(input\_dim, hidden\_dim\*2),  
 nn.ReLU(),  
 nn.Linear(hidden\_dim\*2, hidden\_dim),  
 nn.LayerNorm(hidden\_dim)  
 )  
   
 # Transformer 编码器（增加层数）  
 encoder\_layers = nn.TransformerEncoderLayer(  
 d\_model=hidden\_dim,   
 nhead=nhead,   
 dim\_feedforward=hidden\_dim\*4,   
 dropout=dropout,  
 batch\_first=True  
 )  
 self.transformer\_encoder = nn.TransformerEncoder(encoder\_layers, num\_layers=num\_layers)  
   
 # 多层解码器（替代单一线性层）  
 self.decoder = nn.Sequential(  
 nn.Linear(hidden\_dim, hidden\_dim\*2),  
 nn.ReLU(),  
 nn.Dropout(0.1),  
 nn.Linear(hidden\_dim\*2, hidden\_dim\*3),  
 nn.ReLU(),  
 nn.Dropout(0.1),  
 nn.Linear(hidden\_dim\*3, input\_dim)  
 )

### 3. 改进的 T² Transformer 自编码器 (ImprovedTransformerAutoencoder)

专门针对霍特林 T² 统计量性能的优化版本：

def \_\_init\_\_(self, input\_dim, hidden\_dim, num\_heads=4, num\_layers=2, dropout=0.1):  
 # 特征嵌入层  
 self.feature\_embedding = nn.Sequential(  
 nn.Linear(input\_dim, hidden\_dim),  
 nn.LayerNorm(hidden\_dim),  
 nn.GELU()  
 )  
   
 # Transformer 编码器  
 encoder\_layers = nn.TransformerEncoderLayer(  
 d\_model=hidden\_dim,   
 nhead=num\_heads,  
 dim\_feedforward=hidden\_dim\*4,  
 dropout=dropout,  
 activation="gelu",  
 batch\_first=True  
 )  
 self.transformer\_encoder = nn.TransformerEncoder(encoder\_layers, num\_layers=num\_layers)  
   
 # Transformer 解码器层  
 decoder\_layers = nn.TransformerEncoderLayer(  
 d\_model=hidden\_dim,   
 nhead=num\_heads,  
 dim\_feedforward=hidden\_dim\*4,  
 dropout=dropout,  
 activation="gelu",  
 batch\_first=True  
 )  
 self.transformer\_decoder = nn.TransformerEncoder(decoder\_layers, num\_layers=num\_layers)

## 关键参数

在上述模型中，关键参数包括：

1. **input\_dim**：输入特征的维度，根据数据集变化（SECOM数据集为590个特征）
2. **hidden\_dim**：隐藏层维度，通常设置为较小的值（如27或32）用于降维
3. **num\_heads**：多头注意力机制中的头数，一般为4或8
4. **num\_layers**：Transformer编码器/解码器的层数，基础模型为2层，增强模型为3层或更多
5. **dropout**：丢弃率，一般设为0.1，用于防止过拟合
6. **dim\_feedforward**：前馈网络的维度，通常设置为hidden\_dim的4倍

## 训练方法

模型通常采用以下训练策略：

1. **两阶段训练**：先优化重建误差，再优化特征表示
2. **损失函数**：主要使用MSE（均方误差）作为重建损失
3. **批量大小**：一般为32
4. **学习率**：通常设为0.001，并配合学习率调度器（ReduceLROnPlateau）
5. **正则化**：使用权重衰减（weight\_decay）和Dropout防止过拟合
6. **早停**：当验证损失不再显著改善时停止训练

## 故障检测方法

使用Transformer模型进行故障检测的主要步骤：

1. 使用正常运行数据训练Transformer自编码器
2. 提取编码器部分的特征表示
3. 基于这些特征计算T²统计量和SPE（Q）统计量
4. 设置合适的阈值，当新样本的统计量超过阈值时判定为故障

## 模型优势

1. **自动特征提取**：无需手动设计特征
2. **捕捉复杂关系**：通过注意力机制捕捉变量间的复杂关系
3. **降噪能力**：自编码器架构有助于过滤噪音
4. **可解释性增强**：通过注意力权重可以分析变量重要性

## 不同模型的应用场景

1. **基本Transformer自编码器**：一般用途的异常检测
2. **增强版Transformer**：专注于SPE（Q统计量）性能，适用于敏感度要求高的场景
3. **改进的T²Transformer**：优化T²统计量性能，适用于需要更好协方差建模的情况
4. **两阶段优化模型**：结合了T²和SPE的优势，适用于复杂环境

## 实现细节

所有模型都实现了批处理、GPU加速和提前停止等技术，以提高训练效率和模型性能。模型参数会自动调整以确保兼容性（如注意力头数需要能被隐藏维度整除）。