**代码需求**

**一、远程或者本地跑通之后搭环境**

https://github.com/ahenkes1/HENKES\_SNN

**二、替代梯度的修改和集成**

原文：

spike\_grad = surrogate.atan()

https://snntorch.readthedocs.io/en/latest/snntorch.surrogate.html

修改需求：首先用五种**替代梯度**分别跑一遍，用**BIC**方法**集成**结果。(可能需要适当调参)

（1）五种替代梯度

spike\_grad\_atan = surrogate.atan(alpha=2.0)

spike\_grad\_sigmoid = surrogate.sigmoid(slope=25)

spike\_grad\_straight\_through = surrogate.straight\_through\_estimator()

spike\_grad\_triangular = surrogate.triangular(threshold=1.0)

spike\_grad\_spike\_rate\_escape = surrogate.spike\_rate\_escape(beta=1.0, slope=25)

（2）BIC

<1-对于每种替代梯度函数，代码在训练过程中记录了预测与实际输出之间的均方误差（Mean Squared Error, MSE），存储在 losses 数组中。

<2-n\_params 计算了网络中所有可训练参数的总数，包括输入权重、循环权重、输出权重以及 SLSTM 和 RLIF 模块的参数。

<3-首先，对每个模型的损失取自然对数。然后，减去所有模型中最小的对数损失值，以便使得最佳模型的归一化对数损失为零，其余模型的归一化损失为正值。

log\_losses = log(losses);

min\_log\_loss = min(log\_losses);

normalized\_log\_losses = log\_losses - min\_log\_loss;

<4-使用以下公式计算 BIC：

BIC = time\_steps \* normalized\_log\_losses + n\_params \* log(time\_steps);

其中：

* time\_steps 是时间步数，即数据点的数量。
* normalized\_log\_losses 反映了模型拟合的好坏，值越小表示拟合越好。
* n\_params 是模型的参数数量，log(time\_steps) 是对时间步数的自然对数，用于惩罚模型复杂度。

（3）集成方法

<1-计算 BIC 的相对差异：

min\_BIC = min(BIC);

delta\_BIC = BIC - min\_BIC;

max\_delta = max(delta\_BIC);

if max\_delta > 0

delta\_BIC = delta\_BIC / max\_delta \* 10;

end

<2-计算权重：

weights = exp(-0.5 \* delta\_BIC);

weights = weights / sum(weights);

<3-计算预测值和方差：

% 计算预测

bma\_prediction = zeros(size(outputs));

for i = 1:n\_gradients

bma\_prediction = bma\_prediction + weights(i) \* predictions{i};

end

% 计算方差

bma\_variance = zeros(size(outputs));

for i = 1:5

model\_var = mean((predictions{i} - outputs).^2, 2);

bma\_variance = bma\_variance + weights(i) \* (model\_var + (predictions{i} - bma\_prediction).^2);

end

**三、将输入数据命令改为如下内容：**

% 从文件加载实际数据

arr6\_data = load('arr6.txt');

tau\_arr6\_data = load('tau\_arr6.txt');

% 提取关节角度数据（第4、5列）和对应的力矩数据

q1 = arr6\_data(:, 4); % 第一个活动关节角度

q2 = arr6\_data(:, 5); % 第二个活动关节角度

% 计算关节速度（使用差分）

dq1 = [0; diff(q1)]; % 第一个关节角速度

dq2 = [0; diff(q2)]; % 第二个关节角速度

% 提取实际力矩数据（使用前两列力矩）

tau1 = tau\_arr6\_data(:, 4); % 第一个关节力矩

tau2 = tau\_arr6\_data(:, 5); % 第二个关节力矩

% 整理输入和输出数据

inputs = [q1'; q2'; dq1'; dq2'];

outputs = [tau1'; tau2'];

% 归一化输入和输出

inputs = (inputs - mean(inputs, 2)) ./ std(inputs, 0, 2);

outputs = (outputs - mean(outputs, 2)) ./ std(outputs, 0, 2);

将其前70%数据作为训练集，后30%作为测试集。

**[arr6.txt]**

**[tau\_arr6.txt]**

**四、对比实验和消融实验**

相关对比实验和消融实验也需要补充。（参考第五部分，所需的输出表格和图片）

原始代码中的对比实验和消融实验：

**[代理梯度实现解析.pdf]**

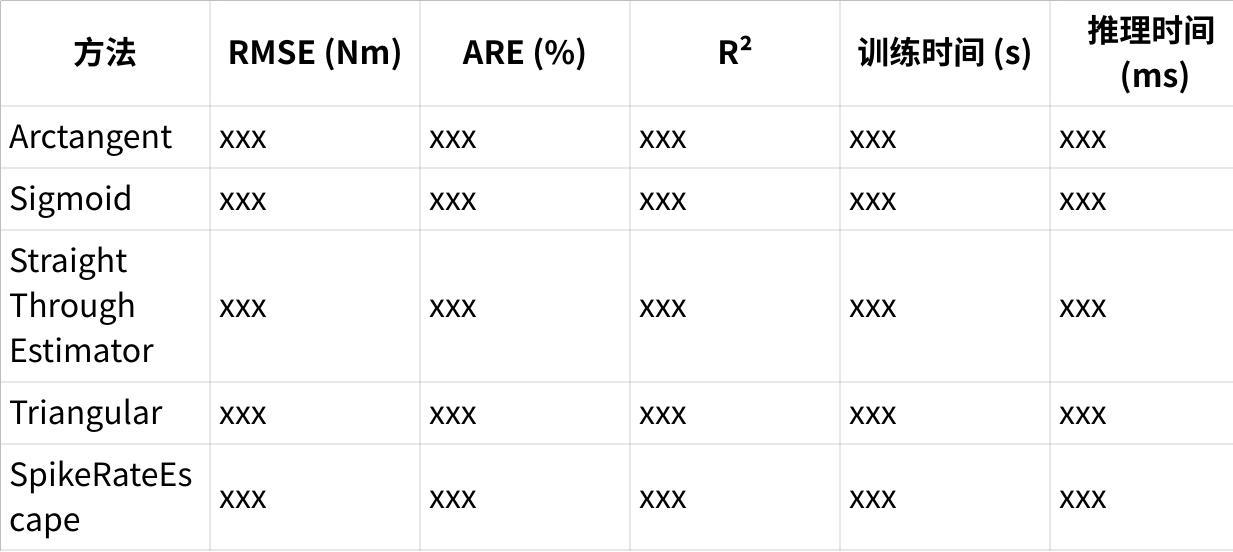
**五、所需的输出表格和图片**

（1）**表格1：不同替代梯度方法的性能比较**

**主要内容：**

* 比较五种替代梯度方法（Arctangent、Sigmoid、直通估计器（Straight Through Estimator）、Triangular 和 SpikeRateEscape）的性能指标。
* 指标包括：均方根误差（RMSE）、平均相对误差（ARE）、决定系数（R²）、训练时间（秒）、推理时间（毫秒）。

**示例：**



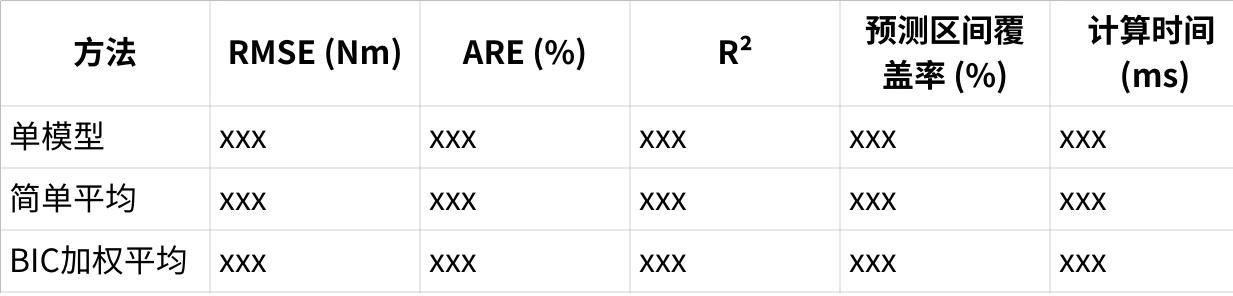
**点击图片可查看完整电子表格**

（2）**表格2：单模型、简单平均和BIC加权平均的性能比较**

**主要内容：**

* 比较单个模型、简单平均和基于BIC的加权平均三种方法的性能指标。
* 指标包括：均方根误差（RMSE）、平均相对误差（ARE）、决定系数（R²）、预测区间覆盖率（%）、计算时间（毫秒）。

**示例：**



**点击图片可查看完整电子表格**

（3）**表格3：BIC加权平均中各模型的权重**

**主要内容：**

* 列出参与BIC加权平均的各模型及其对应的权重。

**示例：**



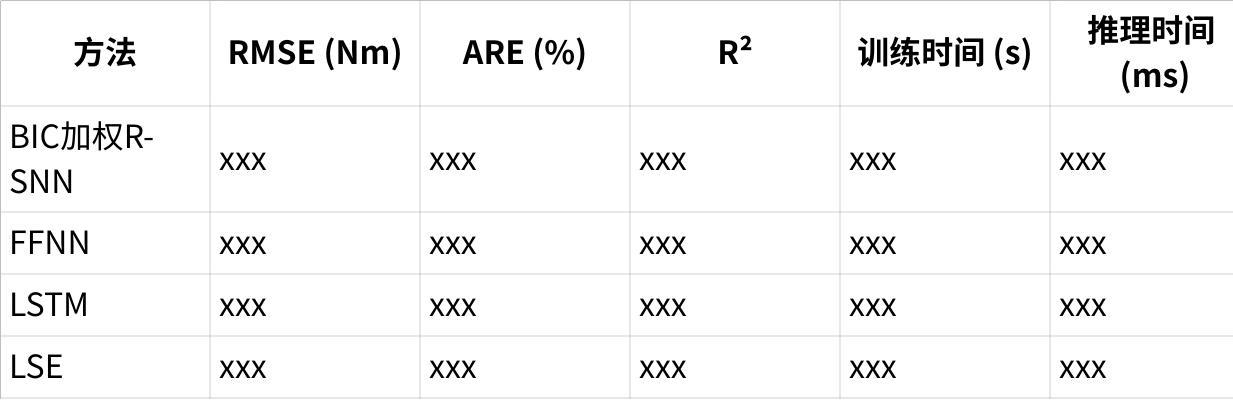
**点击图片可查看完整电子表格**

（4）**表格4：BIC加权平均R-SNN与基线模型的性能比较**

**主要内容：**

* 比较BIC加权平均R-SNN与传统基线模型（FFNN、LSTM、LSE）在多个性能指标上的表现。
* 指标包括：均方根误差（RMSE）、平均相对误差（ARE）、决定系数（R²）、训练时间（秒）、推理时间（毫秒）、参数数量。

**示例：**



**点击图片可查看完整电子表格**

（5）**表格5：BIC R-SNN与基线模型之间RMSE差异的显著性检验结果**

**主要内容：**

* 展示BIC R-SNN与各基线模型在RMSE指标上的显著性差异。
* 包含比较对象、p值和显著性标记。

**示例：**



**点击图片可查看完整电子表格**

（6）图1 **BIC-R-SNN的训练损失曲线**

**主要内容：**

* 展示bic-R-SNN在训练过程中的训练损失和验证损失随训练轮数的变化曲线。
* X轴为训练轮数，Y轴为损失值。

(7) 图2 **典型测试轨迹上的扭矩预测比较**

**主要内容：**

* 比较BIC加权R-SNN、FFNN、LSTM和LSE在一个典型测试轨迹上的扭矩预测结果。
* 展示真实扭矩值与各模型预测值随时间的变化曲线。

（8）图3 **不同替代梯度方法的收敛曲线**

**主要内容：**

* 展示五种替代梯度方法在训练过程中的损失函数收敛曲线。
* X轴为训练轮数，Y轴为损失值。

（9） 示意图：BIC-R-SNN神经网络结构图