算法 1 特征抽取

输入: $data$ 时间序列对应的数据，$label$ 时间序列对应的标签，$window\_len$ 窗口长度

输出: 时间序列抽取的特征矩阵和标签

function EXTRACT_FEATURES($data, label, window\_len$)
    $features \leftarrow 0$
    $targets \leftarrow 0$
    $i \leftarrow 0$
    $num\_windows \leftarrow len(data)/(window\_len/2)$
    for $i = 0 \rightarrow num\_windows$ **do**
        $target \leftarrow int(label[i \rightarrow i + window\_len].mode())$
        $targets \leftarrow targets \cup target$
        for $c = 0 \rightarrow data.columns$ **do**
            $features \leftarrow features \cup$ FEATURIZE($data[i \rightarrow i + window\_len]$)
        end for
        $i \leftarrow i + window\_len/2$
    end for
    return $features, targets$
end function

$$Precision = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{(TP + FN)}$$

$$F1 - Score = \frac{2 * Precision * Recall}{(Recall + Precision)}$$

$$Accuracy = \frac{TP}{(TP + FP + FN + TN)}$$

算法 2 FEATURIZE

$rms\_val = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$

$min\_max\_mean = \frac{1}{len}\sum_{i=1}^{len} |max_i - min_i|$ 其中 $len = \min(max, min)$ ；min 和 max 为序列中的极大值和极小值点

$peak = \max max - \min min$

$peaknum = len(max) + len(min)$

$mean = \frac{1}{n}\sum_{i=1}^{n} x_i$

$standarddeviation = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i - mean}$

$coefficientsofvariation = \frac{standarddeviation}{mean}$

$Skewness = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - mean)^3}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - mean)^2)^{\frac{3}{2}}}$

$Kurtosis = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - mean)^4}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - mean)^2)^3} - 3$

$log - energy = \frac{1}{n}\sum_{i=1}^{n} log(x_i^2)$

算法 3 单一活动类别识别

function TASK1

    $features \leftarrow 0$

    $targets \leftarrow 0$

    for $i = 0 \rightarrow 12$ **do**

        $data \leftarrow data \cup$ PREPROCESSING(i)

        for $i = 1 \rightarrow 7$ **do**

            $d \leftarrow data[label = i]$

            $feature, target =$ EXTRACT_FEATURES$(d[x, y...], d[label], 10)$

            $features \leftarrow features \cup feature$

            $targets \leftarrow targets \cup target$

        end for

    end for

    $classifiers = Train(features, targets)$

    for $i = 13 \rightarrow 15$ **do**

        $data \leftarrow data \cup$ PREPROCESSING(i)

        for $i = 1 \rightarrow 7$ **do**

            $feature, target =$ EXTRACT_FEATURES$(data[label == i][x, y...], data[label = i][label], 10)$

            TEST(classifiers,feature,target)

        end for

    end for

end function

算法 4 多活动类别识别

```
function TASK3
    features ← 0
    targets ← 0
    for i = 0 → 12 do
        data ← data∪PREPROCESSING(i)
        feature, target = EXTRACT__FEATURES(data[x, y...], data[label], 3)
        features ← features ∪ feature
        targets ← targets ∪ target
    end for
    classifiers = Train(features, targets)
    for i = 13 → 15 do
        data ← data∪PREPROCESSING(i)
        feature, target = EXTRACT__FEATURES(data[x, y...], data[label], 3)
        TEST'(classifiers,feature,target)
    end for
end function
```

算法 5 多活动类别识别

```
function TASK3
    features ← 0
    targets ← 0
    for i = 0 → 12 do
        data ← data∪PREPROCESSING(i)
        feature, target = EXTRACT__FEATURES(data[x, y...], data[label], 3)
        features ← features ∪ feature
        targets ← targets ∪ target
    end for
    classifiers = Train(features, targets)
    for i = 13 → 15 do
        data ← data∪PREPROCESSING(i)
        feature, target = EXTRACT__FEATURES(data[x, y...], data[label], 3)
        TEST'(classifiers,feature,target)
    end for
end function
```

算法 6 梯度下降法

输入: $data$ 数据矩阵，$label$ 标签

输出: 权重

```
function gradAscent(data, label)
    alpha ← 0.001
    maxCycles ← 500
    weights ← ones((n, 1))
    for k = 0 → maxCycles do
        h ← sigmoid(data * weights)
        error ← (label − h)
        weights ← weights + alpha * data * error
    end for
    return weights
end function
```

算法 7 决策树

输入: $data$ 数据矩阵,$Target\_attribute$ 要预测的目标属性，$Attributes$ 除目标属性外供学习到的决策树测试属性列表

输出: $root$ 一颗能正确分类给定 data 的决策树

function DecisionTreeClassifier($data, label$)
    创建 $root$ 节点
    if 所有 data 都为正 **then**
        return label = + 的单节点树 root
    end if
    if 所有 data 都为负 **then**
        return label = -的单节点树 root
    end if
    if attributes 为空 **then**
        return 单节点树 root，$label = data$ 中最普遍的 $Target\_attributes$
    end if
    $A \leftarrow attributes$ 中分类 $data$ 能力最好的属性
    root 的决策属性 $\leftarrow A$
    for $i \rightarrow len(A)$ **do**
        $a \leftarrow A[i]$
        在 root 下加一个新的分支对应测试 A=a
        令 data(a) 为 data 中满足 A 属性值为 a 的子集
        在这个新分支下一个子树 DecisionTreeClassifier($data(a), Target\_attribute, Attributes - A$)
    end for
    return $root$
end function