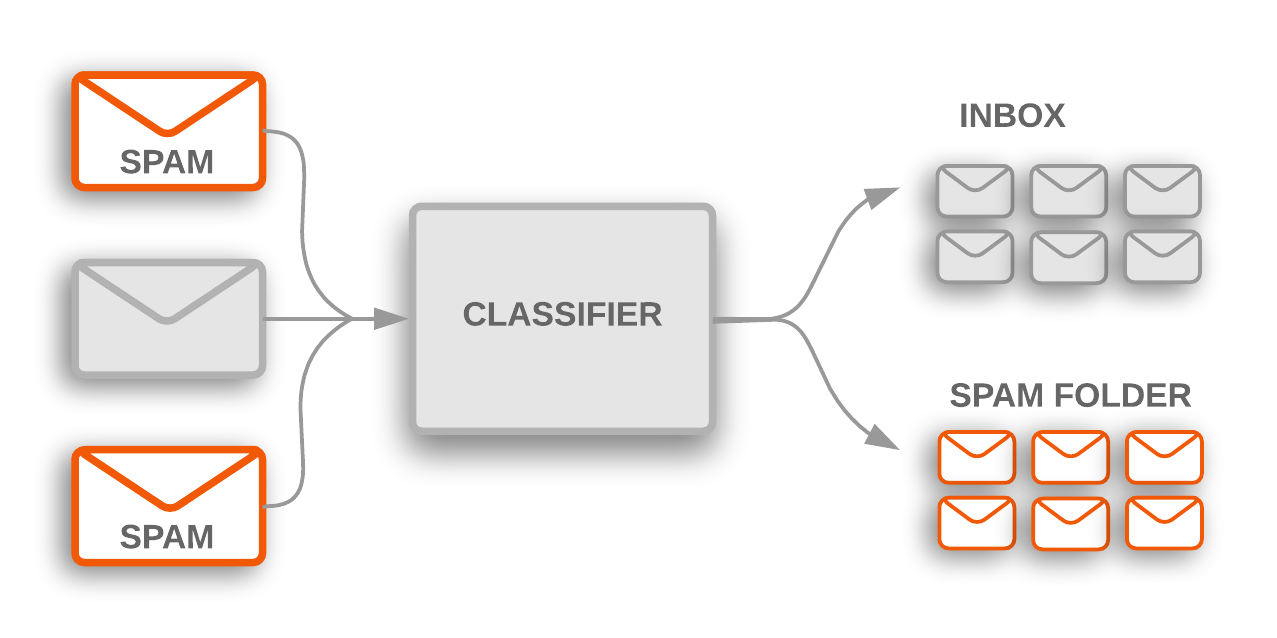
# 文本分类最佳实践

大纲

**什么是文本分类？**

文本分类作为文本处理和挖掘的重要算法，也是各类文本处理系统的中心算法。具体的使用场景包括：垃圾邮件分拣和过滤；用户评论的情感分析；论坛不当言论检测，文章按分类自动聚合等。



图一：邮件的主题分类可以将用户收到的邮件进行分拣过滤

**目标**

1.分享一个端到端的基于机器学习文本分类方案

2.怎样为手上的文本分类问题选择一个合适的分类算法

3.怎样通过Tensorflow和Keras实现文本分类

**深度学习掌握的关键知识：**

有监督学习(SupervisdLearning)

输入(Input)

权重(Weight)

偏置(Bias)

输出(Output)

模型(Model)

激活函数(Activation)

损失函数(Loss)

优化器(Optimizer)

反向传播（BackPropagation）

梯度下降（GradientDescent）

训练迭代(Training /Batch Training)

性能指标(Metrics)

准确率(Accuracy)

文本分类的基本步骤（workflow）

1.收集数据

2.数据理解和探索、模型选择

3.数据清洗和准备

4.建模、训练、验证和评估模型

5.调参

6.模型部署上线

Step1：收集数据

途径：网络采集、积累数据源、开源数据集、各大公司数据API等

值得注意的点：

1.如果采用接口方式获取的外部数据源则需要遵守API的基本规范，部分公司的开放接口在提供数据查询的同时也严格限制了访问和查询的速度

2.理论上越“多”的数据将会对后续建模更有帮助

3.确保收集的各个类别的样本数量是平衡的，建模的时候需要对类别下的样本数进行比较。

4.确保收集的样本能充分表征未来预测的输入空间

Step2: 数据理解和探索

1.检查数据：

对收集的数据需要做一些抽样检查，确保数据符合预期。例如：需要对某个类别的数据进行随机抽样并查看该数据的描述与类别是否符合。

2.关键指标的分析：

样本总数

类别总数

每个类别的样本总数

每个类别的词数

词频的分布情况

样本文本长度的分布情况

3.模型的选择

怎样将现有的文本数据转化为模型算法需要的数值数据？

如何为目标文本选择一个合适的分类模型？

应该为模型配置怎样的超参数？

google采用12个数据集，总体样本数接近450K（问题涉及情感分析和主题分类的问题），通过对每个数据集的采用不同的预处理和不同的模型结构做了实验，得到了下图的结论：

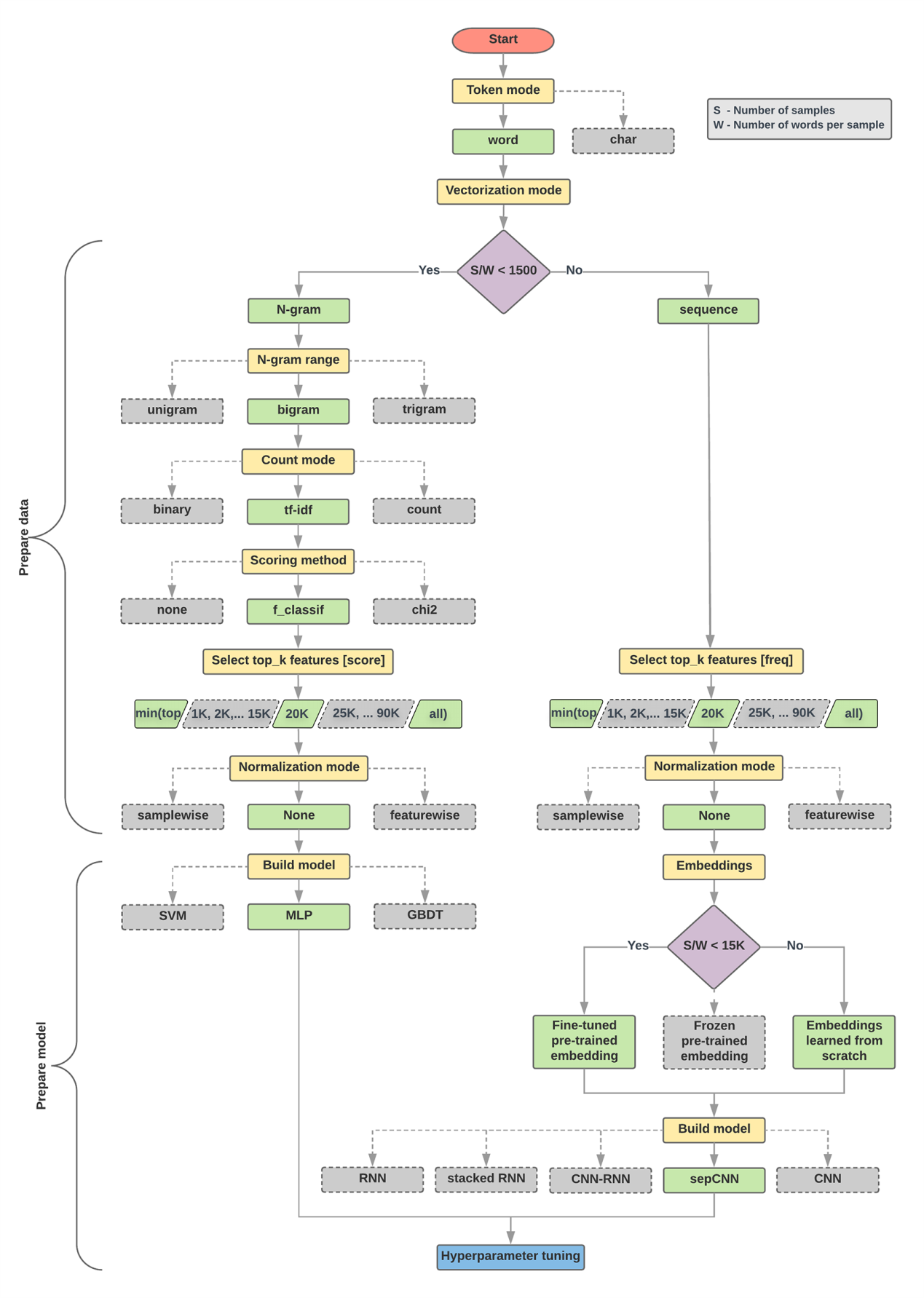


图2.本分类流图

Step3：数据清洗和准备

1.数据顺序

在预处理数据之前需要对数据的顺序进行打乱，确保不是按照类别来进行排列，因为在划分训练集和验证集的时候会导致分割的数据集合不能完整表征数据的分布

2.预处理（分词、向量化表示）：

分词：利用自定义词库和分词软件进行分词

向量化：将单词的文本形式转化为算法需要的数值向量形式

Step4. 建模、训练、验证和评估模型

建模：利用Tensorflow或Keras建模（sklearn中的传统机器学习算法、pytorch等）

神经网络建模流程：

定义输入(Input)、权重(Weight)、偏置(Bias)、输出变量(Output)

算法公式构造建模(Model)选择激活函数(Activation)：y^ = activation(W\*X +b)

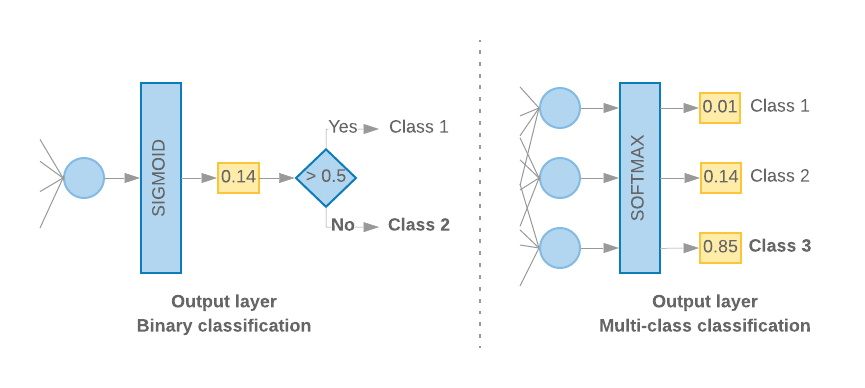
定义损失函数(Loss)：通常用来衡量预测的类别y^和真实类别Y之间的差距

选择优化器(Optimizer)：采用反向传播（BackPropagation）对算法的权重和偏置进行调整，其中调整的方向是利用梯度下降（GradientDescent）算法进行计算的

训练迭代(Training /Batch Training)

度量分类性能(Metrics)：利用验证集和测试集对训练好的分类器进行验证，其中验证指标包括：正确率(Accuracy), AUC等

不同任务输出层定义：定义分类输出是二分类（输出的结果应该是单个概率值表征正样本类别的概率）还是多分类（输出应该是和类别数目相同的概率值，比如10个目标类别，输出就是保证10个类别的概率，其中他们的加和为1）还是多标签分类（一个样本可能对应多个类别，输出为样本对应每一个标签的概率，加和不为1）。根据结果不同输出层采用的激活函数也不一样，通常二分类和多标签分类采用sigmoid激活函数，多分类采用softmax激活函数



Step5. 调参

调参是为了最大化提升模型的性能（每个节点的最佳参数的强强联合使得网络预测分类准确性更高）

调参手段：

网络的层数

每层神经元的数量

Droupout的比例

学习率

卷积核大小（CNN）

嵌入维度大小（Embedding）

6.模型部署上线

途径：Tensorflow serving

TF model + flask/tornado

云服务类似GoogleCloud

值得注意的点：

确保预测的数据和训练数据是在一个分布下的

做好监控和纠错

如果数据分布改变（例如调整分类数）则需要重新训练模型

7.高级话题(DeepLearning Pipeline For Classification)