纵向DNN算法参数介绍文档

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数含义 | 默认值 | 可选值 |
| task\_type | 模型执行任务的类型 | "classification" | 为字符串变量，可选值包括："classification"、"regression" |
| config\_type | 参数类型，判断具体模型参数采用何种后端包进行实现，与横向不同，仅支持keras | "keras" | "keras" |
| bottom\_nn\_define  interactive\_layer\_define  top\_nn\_define | 描述三部分模型结构的字典型参数，在使用时应注意其被放置在role\_parameters中guest和host的hetero\_nn\_0中 | 三者依次定义，且定义的规则相同，下面介绍过程中统一使用nn\_define作为父类 |  |
| nn\_define.class\_name | 规定模型的类型参数 |  | "Sequential" |
| nn\_define.config | 规定模型的具体参数 |  |  |
| nn\_define.config.name | 模型的名称，具有唯一性 |  | 字符串值 |
| nn\_define.config.layers | 由模型中的层构成的数组，每一个层结构具有下述属性 |  |  |
| nn\_define.config.layers.class\_name | 规定模型中某一层的类型参数 |  | "Dense" |
| nn\_define.config.layers.config | 规定模型中某一层的具体参数 |  |  |
| nn\_define.config.layers.config.name | 表示模型中的层名，具有唯一性 |  | 字符串值 |
| nn\_define.config.layers.config.trainable | 标识该layer对象是否可训练，若不可训练则训练时该层的参数冻结不变 | True | bool值，可改为False |
| nn\_define.config.layers.config.batch\_input\_shape | 代表输入的数据的整体维度 |  | 数组形式输入 |
| nn\_define.config.layers.config.dtype | 输入数据的数值类型 |  | "float32" |
| nn\_define.config.layers.config.units | 全连接层神经元数量即输出维度 |  | 正整数 |
| nn\_define.config.layers.config.activation | 激活函数 | "relu" | keras支持的激活函数均可，常见如RELu、Softmax |
| nn\_define.config.layers.config.use\_bias | 是否使用偏置 | True | bool值，可设置为False |
| nn\_define.config.layers.config.kernel\_initializer | 模型权重初始化方法 |  |  |
| nn\_define.config.layers.config.kernel\_initializer.class\_name | 权重初始化方法名称 | "Constant" | keras支持的初始化方法均可 |
| nn\_define.config.layers.config.kernel\_initializer.config | 针对权重初始化方法的细节参数 |  |  |
| nn\_define.config.layers.config.kernel\_initializer.config.value | nn\_define.config.layers.config.kernel\_initializer.class\_name为Constant时需要设定的常数值 | 1 | 可以为任意常数，一般不建议过大 |
| nn\_define.config.layers.config.kernel\_initializer.config.dtype | 权重的数值类型 |  | "float32" |
| nn\_define.config.layers.config.bias\_initializer | 模型偏置初始化方法 |  |  |
| nn\_define.config.layers.config.bias\_initializer.class\_name | 偏置初始化方法名称 | "Zeros" | keras支持的初始化方法均可，常使用zeros |
| nn\_define.config.layers.config.bias\_initializer.config | 针对偏置初始化方法的细节参数 |  |  |
| nn\_define.config.layers.config.bias\_initializer.config.value | nn\_define.config.layers.config.bias\_initializer.class\_name为Constant时需要设定的常数值 | 1 | 可以为任意常数，一般不建议过大 |
| nn\_define.config.layers.config.bias\_initializer.config.dtype | 偏置的数值类型 |  | "float32" |
| nn\_define.config.layers.config.kernel\_regularizer | 模型权重正则化器 | null | keras支持的正则化方法均可，一般不使用 |
| nn\_define.config.layers.config.bias\_regularizer | 偏移量正则化器 | null | keras支持的正则化方法均可，一般不使用 |
| nn\_define.config.layers.config.activity\_regularizer | 激活值正则化器 | null | keras支持的正则化方法均可，一般不使用 |
| nn\_define.config.layers.config.kernel\_constraint | 权重的约束 | null | 不能设置，该参数对于多方计算不安全 |
| nn\_define.config.layers.config.bias\_constraint | 偏移量的约束 | null | 不能设置，该参数对于多方计算不安全 |
| nn\_define.keras\_version | 描述模型所使用的keras版本 | "2.2.4-tf" | 与使用对应 |
| nn\_define.backend | 描述模型后端的技术支持框架 | "tensorflow" | "tensorflow" |
| interactive\_layer\_lr | 交互层结构的学习率需要单独定义 | 0.9 | 与正常学习率的数值取值范围相同，一般取较大值 |
| optimizer | 优化器方法，为模型更新当中的基础设定 | "SGD" | 接收两种内容：  1）字符串：  "Adadelta"、"Adagrad"、"Adam"、"Adamax"、"Nadam"、"RMSprop"、"SGD"  2）字典结构，带有支持的优化器名及其所适配的属性键值对（如学习率等），具体结构可查阅keras文档<https://keras.io/api/optimizers/> |
| optimizer.optimizer | 字典结构的优化器方法中的必要字段，对应使用的优化器 |  | "Adadelta"、"Adagrad"、"Adam"、"Adamax"、"Nadam"、"RMSprop"、"SGD"等 |
| optimizer.learning\_rate | 字典结构的优化器方法中的必要字段，对应优化器的学习率 |  | 0-1之间的数字，一般不超过0.1 |
| loss | 描述模型损失函数的参数 | str=None | 字符串型变量，具体输入详见文档<https://keras.io/api/losses/> |
| epochs | 训练中的最大迭代轮次 | 100 | int型，可随意设置 |
| batch\_size | 训练更新模型时所使用的数据规模 | -1 | int型，默认-1为使用全部数据，即不使用mini-batch策略 |
| early\_stop | 判断训练是否收敛的方法 | "diff" | 字符串变量  仅支持"diff"：利用两轮迭代间的loss差分值判断 |
| tol | 残差收敛条件，即容忍出现错误的概率，误差项达到指定值时停止训练 | 1e-5 |  |
| validation\_freqs | 描述训练过程是否进行数据验证。 | None | None:不进行验证正数:每n轮进行一次验证container对象:在指定轮次进行验证注意正数要被epochs整除 |
| early\_stopping\_rounds | 判断训练过程当中的评估指标是否在最近的n轮当中没有提升，若是则停止训练。 | None | int值，可自定义轮次数 |
| metrics | 列表，用于判断模型训练过程中的评价指标，如果没有输入，按照默认参数进行 | None | 自定义参数或None  None时根据任务类型不同，使用的评价指标不同：  1）二分类 ['auc', 'ks']  2）回归 ['root\_mean\_squared\_error']  3）多分类 ['accuracy','precision','recall'] |
| use\_first\_metric\_only | 判断是否采用评估指标列表中的第一项作为唯一的早停判断指标 | True | bool值，可为False |
| predict\_param | 预测相关参数 |  |  |
| predict\_param.threshold | 用于区分正负样本的阈值 | 0.5 | 0到1之间的任意值 |
| cv\_param | 参数名:参数含义 |  |  |
| cv\_param.n\_splits | n\_splits:指定在KFold中使用多少个分割点 | 5 | 大于0任意正整数 |
| cv\_param.mode | mode:指明当前的任务模式 | Hetero | Homo |
| cv\_param.role | role:在当前任务中扮演什么角色 | Guest | Host,Arbiter |
| cv\_param.shuffle | shuffle:在KFold之前是否打乱顺序 | True | False |
| cv\_param.random\_seed | random\_seed:随机种子 | 1 | 任意整数 |
| cv\_param.need\_cv | need\_cv:是否要运行该模块 | False | True |
| cv\_param.output\_fold\_history | output\_fold\_history:是否输出每个fold使用的每个id表 | True | False |
| cv\_param.history\_value\_type | history\_value\_type:指明在output\_fold\_history中包括原始实例或预测分数 | score | instance |
| encrypt\_param | 加密参数 |  |  |
| encrypt\_param.method | 加密时使用的方法 | Paillier | None——横向LR  Paillier  IterativeAffine  Random\_IterativeAffine |
| encrypt\_param.key\_length | 指定加密方法的密钥长度 | 1024 | int值，默认为1024 |
| encrypted\_mode\_calculator\_param | 加密模式计算参数 |  |  |
| encrypted\_mode\_calculator\_param.mode | 定义加密模式 | "confusion\_opt" | "strict"  "fast"  "balance"  "confusion\_opt"  "only" |
| encrypted\_mode\_calculator\_param.re\_encrypted\_rate | 当mode参数为balance时使用 | 1 | int或float型数值，取值为[0, 1]之间 |