package com.twitter.product\_mixer.component\_library.scorer.tensorbuilder

import inference.GrpcService.InferTensorContents

import inference.GrpcService.ModelInferRequest.InferInputTensor

case object SparseMapInferInputTensorBuilder

extends InferInputTensorBuilder[Option[Map[Int, Double]]] {

private final val batchFeatureNameSuffix: String = "batch"

private final val keyFeatureNameSuffix: String = "key"

private final val valueFeatureNameSuffix: String = "value"

def apply(

featureName: String,

featureValues: Seq[Option[Map[Int, Double]]]

): Seq[InferInputTensor] = {

val batchIdsTensorContents = InferTensorContents.newBuilder()

val sparseKeysTensorContents = InferTensorContents.newBuilder()

val sparseValuesTensorContents = InferTensorContents.newBuilder()

featureValues.zipWithIndex.foreach {

case (featureValueOption, batchIndex) =>

featureValueOption.foreach { featureValue =>

featureValue.foreach {

case (sparseKey, sparseValue) =>

batchIdsTensorContents.addInt64Contents(batchIndex.toLong)

sparseKeysTensorContents.addInt64Contents(sparseKey.toLong)

sparseValuesTensorContents.addFp32Contents(sparseValue.floatValue)

}

}

}

val batchIdsInputTensor = InferInputTensor

.newBuilder()

.setName(Seq(featureName, batchFeatureNameSuffix).mkString("\_"))

.addShape(batchIdsTensorContents.getInt64ContentsCount)

.addShape(1)

.setDatatype("INT64")

.setContents(batchIdsTensorContents)

.build()

val sparseKeysInputTensor = InferInputTensor

.newBuilder()

.setName(Seq(featureName, keyFeatureNameSuffix).mkString("\_"))

.addShape(sparseKeysTensorContents.getInt64ContentsCount)

.addShape(1)

.setDatatype("INT64")

.setContents(sparseKeysTensorContents)

.build()

val sparseValuesInputTensor = InferInputTensor

.newBuilder()

.setName(Seq(featureName, valueFeatureNameSuffix).mkString("\_"))

.addShape(sparseValuesTensorContents.getFp32ContentsCount)

.addShape(1)

.setDatatype("FP32")

.setContents(sparseValuesTensorContents)

.build()

Seq(batchIdsInputTensor, sparseKeysInputTensor, sparseValuesInputTensor)

}

}