#include "tensorflow/core/framework/op.h"

#include "tensorflow/core/framework/shape\_inference.h"

#include "tensorflow/core/framework/op\_kernel.h"

#include <twml.h>

#include "tensorflow\_utils.h"

using namespace tensorflow;

REGISTER\_OP("FeatureId")

.Attr("feature\_names: list(string)")

.Output("output: int64")

.SetShapeFn([](::tensorflow::shape\_inference::InferenceContext\* c) {

return Status::OK();

}).Doc(R"doc(

A tensorflow OP that hashes a list of strings into int64. This is used for feature name hashing.

Attr

feature\_names: a list of string feature names (list(string)).

Outputs

ouput: hashes corresponding to the string feature names (int64).

)doc");

class FeatureId : public OpKernel {

private:

std::vector<string> input\_vector;

public:

explicit FeatureId(OpKernelConstruction\* context) : OpKernel(context) {

OP\_REQUIRES\_OK(context, context->GetAttr("feature\_names", &input\_vector));

}

void Compute(OpKernelContext\* context) override {

// Get size of the input\_vector and create TensorShape shape

const int total\_size = static\_cast<int>(input\_vector.size());

TensorShape shape = {total\_size};

// Create an output tensor

Tensor\* output\_tensor = nullptr;

OP\_REQUIRES\_OK(context, context->allocate\_output(0, shape,

&output\_tensor));

auto output\_flat = output\_tensor->flat<int64>();

// Transform the input tensor into a int64

for (int i = 0; i < total\_size; i++) {

output\_flat(i) = twml::featureId(input\_vector[i]);

}

}

};

REGISTER\_KERNEL\_BUILDER(

Name("FeatureId")

.Device(DEVICE\_CPU),

FeatureId);