tf.einsum

Aliases:

- `tf.einsum`
- `tf.linalg.einsum`

```
einsum(
    equation,
    *inputs,
    **kwargs
)
```

Defined in `tensorflow/python/ops/special_math_ops.py`.

See the guide: Math > Reduction

A generalized contraction between tensors of arbitrary dimension.

This function returns a tensor whose elements are defined by `equation`, which is written in a shorthand form inspired by the Einstein summation convention. As an example, consider multiplying two matrices A and B to form a matrix C. The elements of C are given by:

```
  C[i,k] = sum_j A[i,j] * B[j,k]
```

The corresponding `equation` is:

```
  ij,jk->ik
```

In general, the `equation` is obtained from the more familiar element-wise equation by 1. removing variable names, brackets, and commas, 2. replacing "*" with ",", 3. dropping summation signs, and 4. moving the output to the right, and replacing "=" with "->".

Many common operations can be expressed in this way. For example:

```
# Matrix multiplication
>>> einsum('ij,jk->ik', m0, m1)  # output[i,k] = sum_j m0[i,j] * m1[j, k]

# Dot product
>>> einsum('i,i->', u, v)  # output = sum_i u[i]*v[i]

# Outer product
>>> einsum('i,j->ij', u, v)  # output[i,j] = u[i]*v[j]

# Transpose
>>> einsum('ij->ji', m)  # output[j,i] = m[i,j]

# Batch matrix multiplication
>>> einsum('aij,ajk->aik', s, t)  # out[a,i,k] = sum_j s[a,i,j] * t[a, j, k]
```

This function behaves like `numpy.einsum`, but does not support:

- Ellipses (subscripts like `ij...,jk...->ik...`)
- Subscripts where an axis appears more than once for a single input (e.g. `ijj,k->ik`).
- Subscripts that are summed across multiple inputs (e.g., `ij,ij,jk->ik`).

Args:

- `equation`: a `str` describing the contraction, in the same format as `numpy.einsum`.
- `*inputs`: the inputs to contract (each one a `Tensor`), whose shapes should be consistent with `equation`.
- `name`: A name for the operation (optional).

Returns:

The contracted `Tensor`, with shape determined by `equation`.

Raises:

- `ValueError`: If
  - the format of `equation` is incorrect,
  - the number of inputs implied by `equation` does not match `len(inputs)`,
  - an axis appears in the output subscripts but not in any of the inputs,
  - the number of dimensions of an input differs from the number of indices in its subscript, or
  - the input shapes are inconsistent along a particular axis.
