tf.matrix_triangular_solve

Aliases:

- `tf.linalg.triangular_solve`
- `tf.matrix_triangular_solve`

```
matrix_triangular_solve(
    matrix,
    rhs,
    lower=True,
    adjoint=False,
    name=None
)
```

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

See the guide: Math > Matrix Math Functions

Solves systems of linear equations with upper or lower triangular matrices by

backsubstitution.

`matrix` is a tensor of shape `[..., M, M]` whose inner-most 2 dimensions form square matrices. If `lower` is `True` then the strictly upper triangular part of each inner-most matrix is assumed to be zero and not accessed. If `lower` is False then the strictly lower triangular part of each inner-most matrix is assumed to be zero and not accessed. `rhs` is a tensor of shape `[..., M, K]`.

The output is a tensor of shape `[..., M, K]`. If `adjoint` is `True` then the innermost matrices in `output` satisfy matrix equations `matrix[..., :, :] * output[..., :, :] = rhs[..., :, :]`. If `adjoint` is `False` then the strictly then the innermost matrices in `output` satisfy matrix equations `adjoint(matrix[..., i, k]) * output[..., k, j] = rhs[..., i, j]`.

Args:

- `matrix`: A `Tensor`. Must be one of the following types: `float64`, `float32`, `complex64`, `complex128`. Shape is `[..., M, M]`.
- `rhs`: A `Tensor`. Must have the same type as `matrix`. Shape is `[..., M, K]`.
- `lower`: An optional `bool`. Defaults to `True`. Boolean indicating whether the innermost matrices in `matrix` are lower or upper triangular.
- `adjoint`: An optional `bool`. Defaults to `False`. Boolean indicating whether to solve with `matrix` or its (block-wise) adjoint.
- `name`: A name for the operation (optional).

Returns:

A `Tensor`. Has the same type as `matrix`. Shape is `[..., M, K]`.

numpy compatibility

```
Equivalent to np.linalg.triangular_solve
```
