tf.accumulate_n

```
accumulate_n(
    inputs,
    shape=None,
    tensor_dtype=None,
    name=None
)
```

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

See the guide: Math > Reduction

Returns the element-wise sum of a list of tensors.

Optionally, pass `shape` and `tensor_dtype` for shape and type checking, otherwise, these are inferred.

NOTE: This operation is not differentiable and cannot be used if inputs depend on trainable variables. Please use `tf.add_n` for such cases.

Aside from differentiability, `tf.accumulate_n` performs the same operation as `tf.add_n`, but does not wait for all of its inputs to be ready before beginning to sum. This can save memory if inputs are ready at different times, since minimum temporary storage is proportional to the output size rather than the inputs size.

For example:

```
a = tf.constant([[1, 2], [3, 4]])
b = tf.constant([[5, 0], [0, 6]])
tf.accumulate_n([a, b, a])  # [[7, 4], [6, 14]]

# Explicitly pass shape and type
tf.accumulate_n([a, b, a], shape=[2, 2], tensor_dtype=tf.int32)  # [[7,  4],
                                                                 #  [6, 14]]
```

Args:

- `inputs`: A list of `Tensor` objects, each with same shape and type.
- `shape`: Shape of elements of `inputs`.
- `tensor_dtype`: The type of `inputs`.
- `name`: A name for the operation (optional).

Returns:

A `Tensor` of same shape and type as the elements of `inputs`.

Raises:

- `ValueError`: If `inputs` don't all have same shape and dtype or the shape cannot be inferred.

Last updated November 2, 2017.