tf.contrib.training.parse_values

```
parse_values(
    values,
    type_map
)
```

Defined in `tensorflow/contrib/training/python/training/hparam.py`.

Parses hyperparameter values from a string into a python map..

`values` is a string containing comma-separated `name=value` pairs. For each pair, the value of the hyperparameter named `name` is set to `value`.

If a hyperparameter name appears multiple times in `values`, a ValueError is raised (e.g. 'a=1,a=2', 'a[1]=1,a[1]=2').

If a hyperparameter name in both an index assignment and scalar assignment, a ValueError is raised. (e.g. 'a=[1,2,3],a[0] = 1').

The `value` in `name=value` must follows the syntax according to the type of the parameter:

- Scalar integer: A Python-parsable integer point value. E.g.: 1, 100, -12.
- Scalar float: A Python-parsable floating point value. E.g.: 1.0, -.54e89.
- Boolean: Either true or false.
- Scalar string: A non-empty sequence of characters, excluding comma, spaces, and square brackets. E.g.: foo, bar_1.
- List: A comma separated list of scalar values of the parameter type enclosed in square brackets. E.g.: [1,2,3], [1.0,1e-12], [high,low].

When index assignment is used, the corresponding type_map key should be the list name. E.g. for "arr[1]=0" the type_map must have the key "arr" (not "arr[1]").

Args:

- `values`: String. Comma separated list of `name=value` pairs where 'value' must follow the syntax described above.
- `type_map`: A dictionary mapping hyperparameter names to types. Note every parameter name in values must be a key in type_map. The values must conform to the types indicated, where a value V is said to conform to a type T if either V has type T, or V is a list of elements of type T. Hence, for a multidimensional parameter 'x' taking float values, 'x=[0.1,0.2]' will parse successfully if type_map['x'] = float.

Returns:

A python map mapping each name to either: *A scalar value.* A list of scalar values. * A dictionary mapping index numbers to scalar values. (e.g. "x=5,L=[1,2],arr[1]=3" results in {'x':5,'L':[1,2],'arr':{1:3}}")

Raises:

- `ValueError`: If there is a problem with input.
- If `values` cannot be parsed.

- If a list is assigned to a list index (e.g. 'a[1] = [1,2,3]').
- If the same rvalue is assigned two different values (e.g. 'a=1,a=2', 'a[1]=1,a[1]=2', or 'a=1,a=[1]')

Last updated November 2, 2017.