tf.contrib.data.read_batch_features

```
read_batch_features(
    file_pattern,
    batch_size,
    features,
    reader,
    reader_args=None,
    randomize_input=True,
    num_epochs=None,
    capacity=10000
)
```

Defined in `tensorflow/contrib/data/python/ops/readers.py`.

Reads batches of Examples.

Example:

```
serialized_examples = [
  features {
    feature { key: "age" value { int64_list { value: [ 0 ] } } }
    feature { key: "gender" value { bytes_list { value: [ "f" ] } } }
    feature { key: "kws" value { bytes_list { value: [ "code", "art" ] } } }
  },
  features {
    feature { key: "age" value { int64_list { value: [] } } }
    feature { key: "gender" value { bytes_list { value: [ "f" ] } } }
    feature { key: "kws" value { bytes_list { value: [ "sports" ] } } }
  }
]
```

We can use arguments:

```
features: {
  "age": FixedLenFeature([], dtype=tf.int64, default_value=-1),
  "gender": FixedLenFeature([], dtype=tf.string),
  "kws": VarLenFeature(dtype=tf.string),
}
```

And the expected output is:

```
{
  "age": [[0], [-1]],
  "gender": [["f"], ["f"]],
  "kws": SparseTensor(
    indices=[[0, 0], [0, 1], [1, 0]],
    values=["code", "art", "sports"]
    dense_shape=[2, 2]),
}
```

Args:

- `file_pattern`: List of files or patterns of file paths containing `Example` records. See `tf.gfile.Glob` for pattern

rules.

- `batch_size`: An int representing the number of consecutive elements of this dataset to combine in a single batch.
- `features`: A `dict` mapping feature keys to `FixedLenFeature` or `VarLenFeature` values. See `tf.parse_example`.
- `reader`: A function or class that can be called with a `filenames` tensor and (optional) `reader_args` and returns a `Dataset` of serialized Examples.
- `reader_args`: Additional arguments to pass to the reader class.
- `randomize_input`: Whether the input should be randomized.
- `num_epochs`: Integer specifying the number of times to read through the dataset. If None, cycles through the dataset forever.
- `capacity`: Capacity of the ShuffleDataset. A large capacity ensures better shuffling but would increase memory usage and startup time.

Returns:

A dict from keys in features to Tensor or SparseTensor objects.
