tf.nn.sampled_softmax_loss

```
sampled_softmax_loss(
    weights,
    biases,
    labels,
    inputs,
    num_sampled,
    num_classes,
    num_true=1,
    sampled_values=None,
    remove_accidental_hits=True,
    partition_strategy='mod',
    name='sampled_softmax_loss'
)
```

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

See the guide: Neural Network > Candidate Sampling

Computes and returns the sampled softmax training loss.

This is a faster way to train a softmax classifier over a huge number of classes.

This operation is for training only. It is generally an underestimate of the full softmax loss.

A common use case is to use this method for training, and calculate the full softmax loss for evaluation or inference. In this case, you must set `partition_strategy="div"` for the two losses to be consistent, as in the following example:

```
if mode == "train":
  loss = tf.nn.sampled_softmax_loss(
      weights=weights,
      biases=biases,
      labels=labels,
      inputs=inputs,
      ...,
      partition_strategy="div")
elif mode == "eval":
  logits = tf.matmul(inputs, tf.transpose(weights))
  logits = tf.nn.bias_add(logits, biases)
  labels_one_hot = tf.one_hot(labels, n_classes)
  loss = tf.nn.softmax_cross_entropy_with_logits(
      labels=labels_one_hot,
      logits=logits)
```

See our Candidate Sampling Algorithms Reference

Also see Section 3 of Jean et al., 2014 (pdf) for the math.

Args:

- `weights`: A `Tensor` of shape `[num_classes, dim]`, or a list of `Tensor` objects whose concatenation along dimension 0 has shape [num_classes, dim]. The (possibly-sharded) class embeddings.
- `biases`: A `Tensor` of shape `[num_classes]`. The class biases.

- `labels`: A `Tensor` of type `int64` and shape `[batch_size, num_true]`. The target classes. Note that this format differs from the `labels` argument of `nn.softmax_cross_entropy_with_logits`.
- `inputs`: A `Tensor` of shape `[batch_size, dim]`. The forward activations of the input network.
- `num_sampled`: An `int`. The number of classes to randomly sample per batch.
- `num_classes`: An `int`. The number of possible classes.
- `num_true`: An `int`. The number of target classes per training example.
- `sampled_values`: a tuple of (`sampled_candidates`, `true_expected_count`, `sampled_expected_count`) returned by a `*_candidate_sampler` function. (if None, we default to `log_uniform_candidate_sampler`)
- `remove_accidental_hits`: A `bool`. whether to remove "accidental hits" where a sampled class equals one of the target classes. Default is True.
- `partition_strategy`: A string specifying the partitioning strategy, relevant if `len(weights) > 1`. Currently `"div"` and `"mod"` are supported. Default is `"mod"`. See `tf.nn.embedding_lookup` for more details.
- `name`: A name for the operation (optional).

Returns:

A `batch_size` 1-D tensor of per-example sampled softmax losses.
