tf.nn.fixed_unigram_candidate_sampler

```
fixed_unigram_candidate_sampler(
    true_classes,
    num_true,
    num_sampled,
    unique,
    range_max,
    vocab_file='',
    distortion=1.0,
    num_reserved_ids=0,
    num_shards=1,
    shard=0,
    unigrams=(),
    seed=None,
    name=None
)
```

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

See the guide: Neural Network > Candidate Sampling

Samples a set of classes using the provided (fixed) base distribution.

This operation randomly samples a tensor of sampled classes (`sampled_candidates`) from the range of integers `[0, range_max)`.

The elements of `sampled_candidates` are drawn without replacement (if `unique=True`) or with replacement (if `unique=False`) from the base distribution.

The base distribution is read from a file or passed in as an in-memory array. There is also an option to skew the distribution by applying a distortion power to the weights.

In addition, this operation returns tensors `true_expected_count` and `sampled_expected_count` representing the number of times each of the target classes (`true_classes`) and the sampled classes (`sampled_candidates`) is expected to occur in an average tensor of sampled classes. These values correspond to `Q(y|x)` defined in this document. If `unique=True`, then these are post-rejection probabilities and we compute them approximately.

Args:

- `true_classes`: A `Tensor` of type `int64` and shape `[batch_size, num_true]`. The target classes.
- `num_true`: An `int`. The number of target classes per training example.
- `num_sampled`: An `int`. The number of classes to randomly sample.
- `unique`: A `bool`. Determines whether all sampled classes in a batch are unique.
- `range_max`: An `int`. The number of possible classes.
- `vocab_file`: Each valid line in this file (which should have a CSV-like format) corresponds to a valid word ID. IDs are in sequential order, starting from num_reserved_ids. The last entry in each line is expected to be a value corresponding to the count or relative probability. Exactly one of `vocab_file` and `unigrams` needs to be passed to this operation.
- `distortion`: The distortion is used to skew the unigram probability distribution. Each weight is first raised to the

distortion's power before adding to the internal unigram distribution. As a result, `distortion = 1.0` gives regular unigram sampling (as defined by the vocab file), and `distortion = 0.0` gives a uniform distribution.

- `num_reserved_ids`: Optionally some reserved IDs can be added in the range `[0, num_reserved_ids)` by the users. One use case is that a special unknown word token is used as ID 0. These IDs will have a sampling probability of 0.
- `num_shards`: A sampler can be used to sample from a subset of the original range in order to speed up the whole computation through parallelism. This parameter (together with `shard`) indicates the number of partitions that are being used in the overall computation.
- `shard`: A sampler can be used to sample from a subset of the original range in order to speed up the whole computation through parallelism. This parameter (together with `num_shards`) indicates the particular partition number of the operation, when partitioning is being used.
- `unigrams`: A list of unigram counts or probabilities, one per ID in sequential order. Exactly one of `vocab_file` and `unigrams` should be passed to this operation.
- `seed`: An `int`. An operation-specific seed. Default is 0.
- `name`: A name for the operation (optional).

Returns:

- `sampled_candidates`: A tensor of type `int64` and shape `[num_sampled]`. The sampled classes.
- `true_expected_count`: A tensor of type `float`. Same shape as `true_classes`. The expected counts under the sampling distribution of each of `true_classes`.
- `sampled_expected_count`: A tensor of type `float`. Same shape as `sampled_candidates`. The expected counts under the sampling distribution of each of `sampled_candidates`.

Last updated November 2, 2017.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**