tf.contrib.cudnn_rnn.CudnnRNNReluSaveable

Class **CudnnRNNReluSaveable**

Defined in `tensorflow/contrib/cudnn_rnn/python/ops/cudnn_rnn_ops.py`.

SaveableObject implementation handling Cudnn RNN Relu opaque params.

Methods

`__init__`

```
__init__(
    opaque_params,
    num_layers,
    num_units,
    input_size,
    input_mode=CUDNN_INPUT_LINEAR_MODE,
    direction=CUDNN_RNN_UNIDIRECTION,
    scope=None,
    name='cudnn_rnn_saveable'
)
```

Creates a CudnnOpaqueParamsSaveable object.

CudnnOpaqueParamsSaveable is saveable/restorable in a checkpoint file and is used to save/restore the weights and biases parameters in a canonical format which is directly consumable by platform-independent tf RNN cells. Parameters are saved as tensors layer by layer with weight tensors followed by bias tensors, and forward direction followed by backward direction (if applicable). When restoring, a user could name param_variables as desired, and restore weight and bias tensors to these variables.

For CudnnRNNRelu or CudnnRNNTanh, there are 2 tensors per weight and per bias for each layer: tensor 0 is applied to the input from the previous layer and tensor 1 to the recurrent input.

For CudnnLSTM, there are 8 tensors per weight and per bias for each layer: tensor 0-3 are applied to the input from the previous layer and tensor 4-7 to the recurrent input. Tensor 0 and 4 are for the input gate; tensor 1 and 5 the forget gate; tensor 2 and 6 the new memory gate; tensor 3 and 7 the output gate.

For CudnnGRU, there are 6 tensors per weight and per bias for each layer: tensor 0-2 are applied to the input from the previous layer and tensor 3-5 to the recurrent input. Tensor 0 and 3 are for the reset gate; tensor 1 and 4 the update gate; tensor 2 and 5 the new memory gate.

Args:

- `opaque_params` : a variable, Cudnn RNN opaque params.
- `num_layers` : the number of layers for the RNN model.
- `num_units` : the number of units within the RNN model.
- `input_size` : the size of the input, it could be different from the num_units.
- `input_mode` : indicate whether there is a linear projection between the input and the actual computation before the first layer. It could be 'linear_input', 'skip_input' or 'auto_select'. 'linear_input' (default) always applies a linear projection of input onto RNN hidden state. (standard RNN behavior). 'skip_input' is only allowed when input_size == num_units; 'auto_select' implies 'skip_input' when input_size == num_units; otherwise, it implies 'linear_input'.
- `direction` : the direction model that the model operates. Could be either 'unidirectional' or 'bidirectional'
- `scope` : string of VariableScope, the scope of equivalent subgraph consisting only platform-independent tf RNN cells.
- `name` : the name of the CudnnOpaqueParamsSaveable object.

restore

```
restore(
    restored_tensors,
    restored_shapes
)
```


Last updated November 2, 2017.