Deep Learning at Twitter's Scale

Cibele Montez Halasz
Machine Learning Engineer @ Twitter Cortex

October 11, 2018

Agenda

1 Background
2 Workflow/Platform
3 Modeling and Optimizations
4 Additional Performance Gains
5 GPU

Background

- Challenges: Characteristics of Platform
- Data Shift

Challenges

VERY

DATA

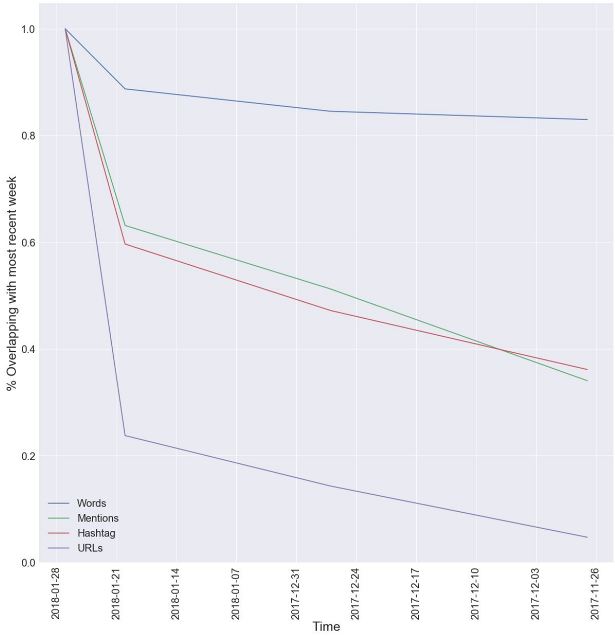
SPARSE

Source: Luca Belli/Dan Shiebler, June, 2018

Challenges

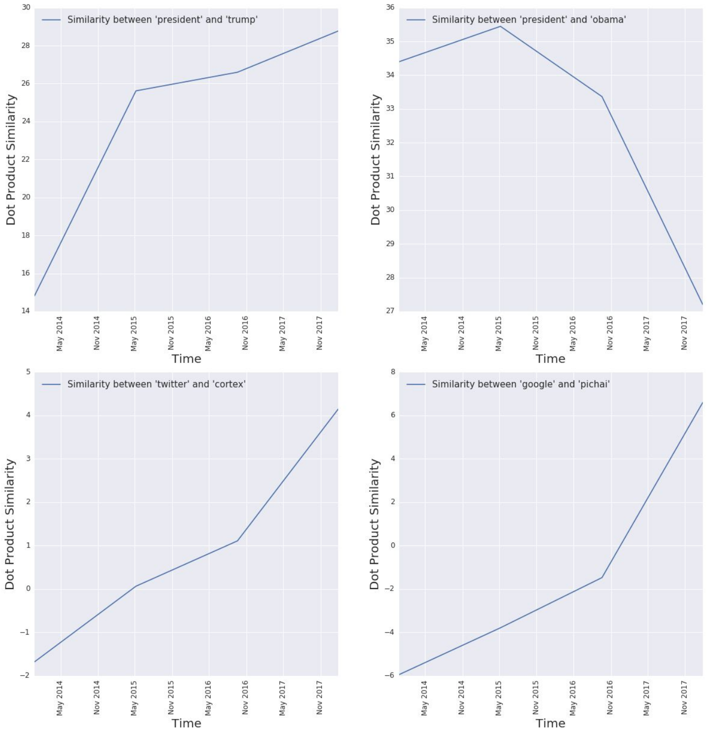
Source: Jacob Kastrenakes, The Verge, July, 2018

Data Shift



Source: Luca Belli/Dan Shiebler, June, 2018

Data Shift

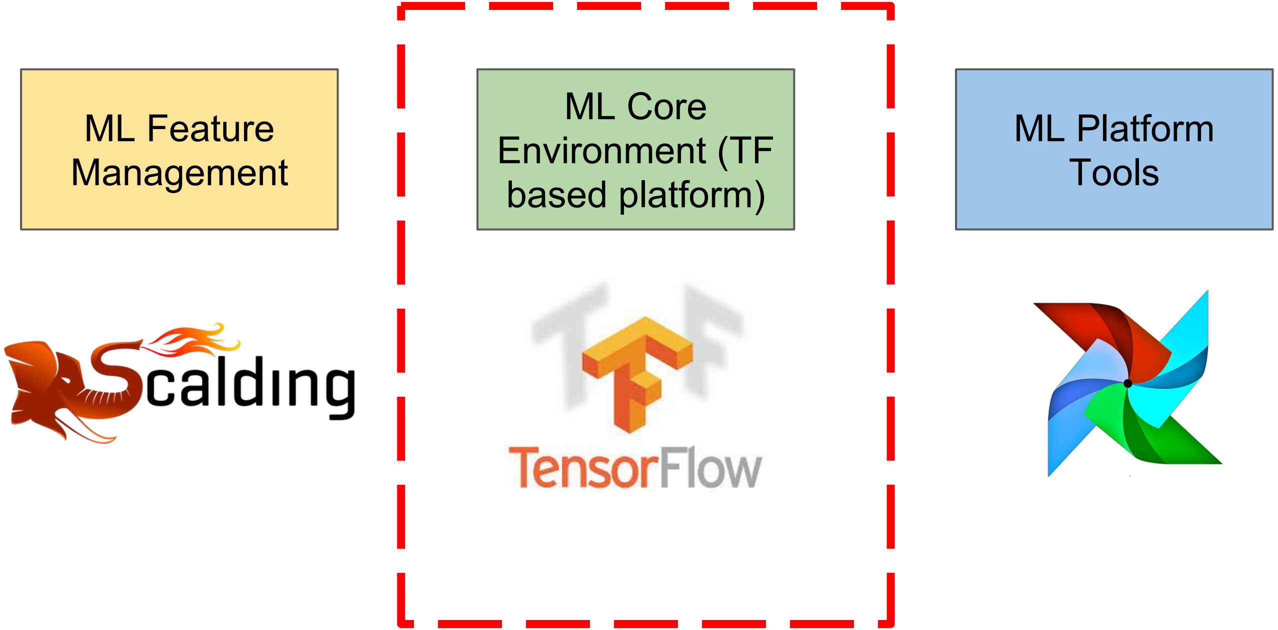


Source: Luca Belli/Dan Shiebler, June, 2018

Machine Learning at the Company

- Environment
- Modeling: some use cases

Environment: ML Platforms

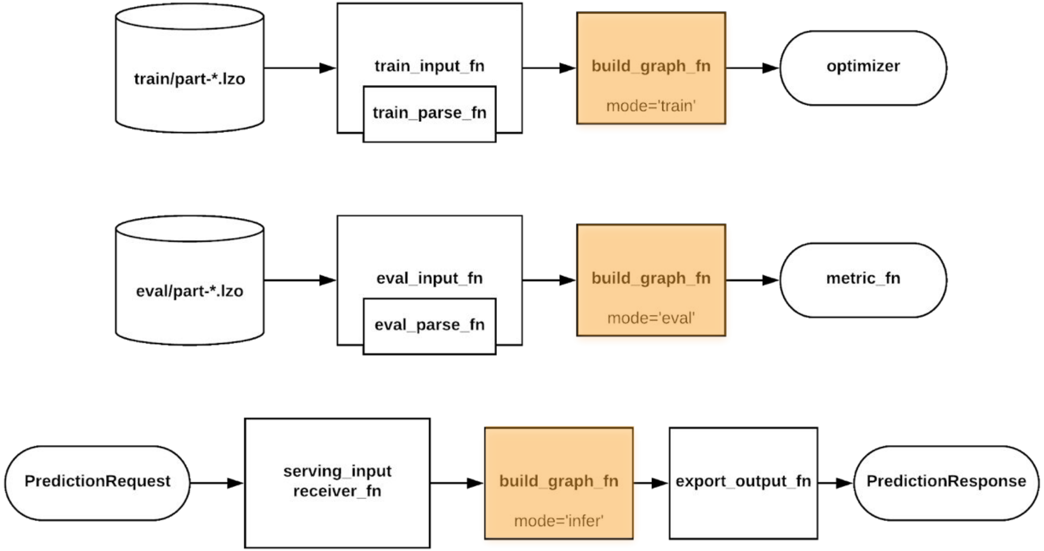


ML Feature Management

ML Core Environment (TF based platform)

ML Platform Tools

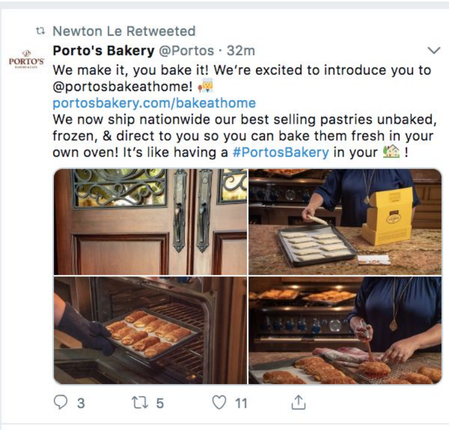
Environment: ML Training



Environment: Priorities

- Feature Addition → Scalable data
- Data Addition → Scalable data
- **Training → Fast, robust training engine**
- **Deployment → Seamless and tested ML services**
- A/B test → Good AB test environment

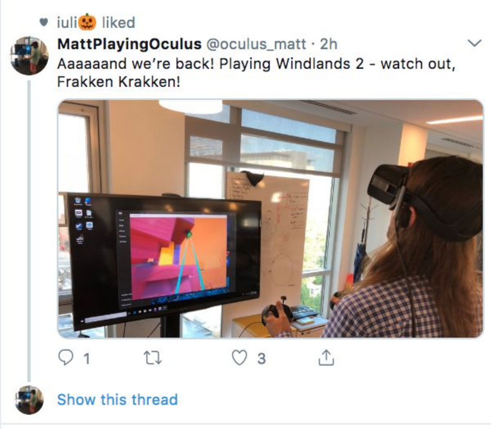
Modeling Use Cases: Timelines



Newton Le Retweeted

Porto's Bakery @Portos · 32m

We make it, you bake it! We're excited to introduce you to @portosbakeathome! 👩‍🍳
portosbakery.com/bakeathome
We now ship nationwide our best selling pastries unbaked, frozen, & direct to you so you can bake them fresh in your own oven! It's like having a #PortosBakery in your 🏡 !

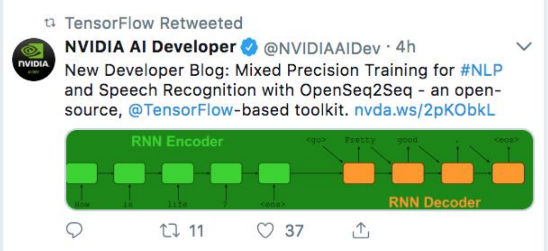


iuli🎃 liked

MattPlayingOculus @oculus_matt · 2h

Aaaaaand we're back! Playing Windlands 2 - watch out, Frakken Krakken!

Show this thread

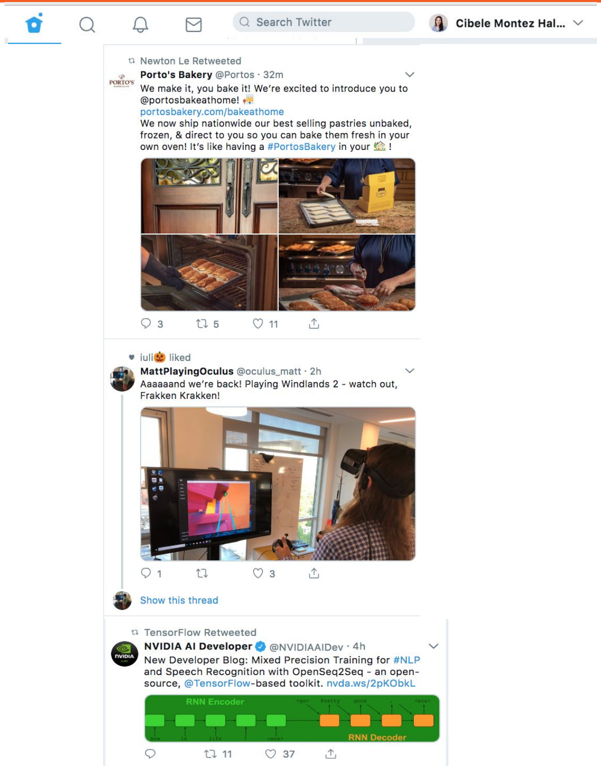


TensorFlow Retweeted

NVIDIA AI Developer @NVIDIAAIDev · 4h

New Developer Blog: Mixed Precision Training for #NLP and Speech Recognition with OpenSeq2Seq - an open-source, @TensorFlow-based toolkit. nvda.ws/2pKObkL

Modeling Use Cases: Timelines



Newton Le Retweeted

Porto's Bakery @Portos · 32m

We make it, you bake it! We're excited to introduce you to @portosbakeathome! 👐
portosbakery.com/bakeathome
We now ship nationwide our best selling pastries unbaked, frozen, & direct to you so you can bake them fresh in your own oven! It's like having a #PortosBakery in your 🏡 !

3 5 11

iuli🎃 liked

MattPlayingOculus @oculus_matt · 2h

Aaaaaand we're back! Playing Windlands 2 - watch out, Frakken Krakken!

1 3

Show this thread

TensorFlow Retweeted

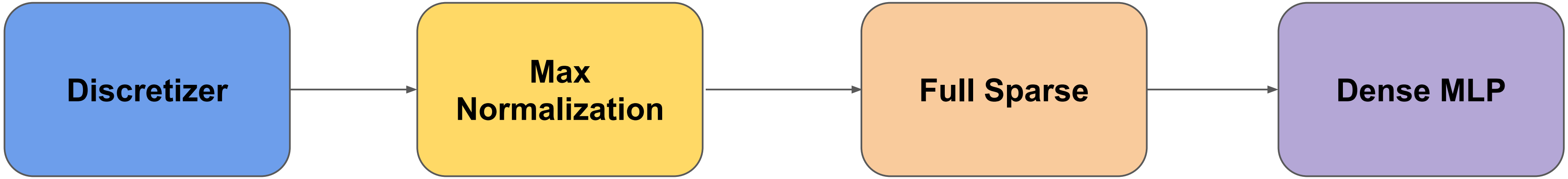
NVIDIA AI Developer @NVIDIAAIDev · 4h

New Developer Blog: Mixed Precision Training for #NLP and Speech Recognition with OpenSeq2Seq - an open-source, @TensorFlow-based toolkit. nvda.ws/2pKObkL

11 37

Modeling: Modeling and Optimizations with TensorFlow

Modeling

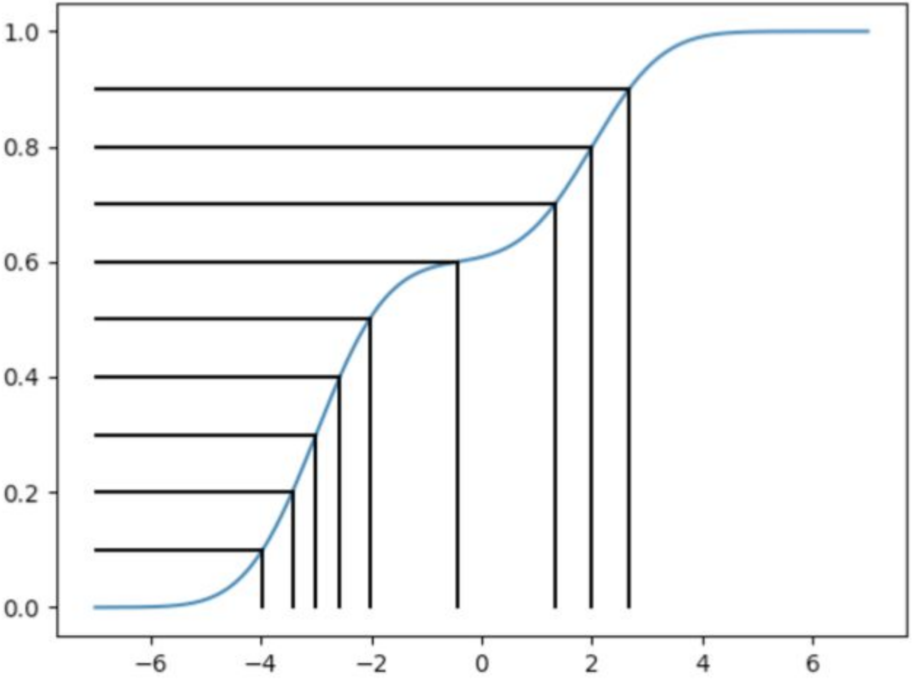

Discretizer

Max Normalization

Full Sparse

Dense MLP

Modeling: Discretizer



Sparse Linear Layer: Online Normalization

- Example:

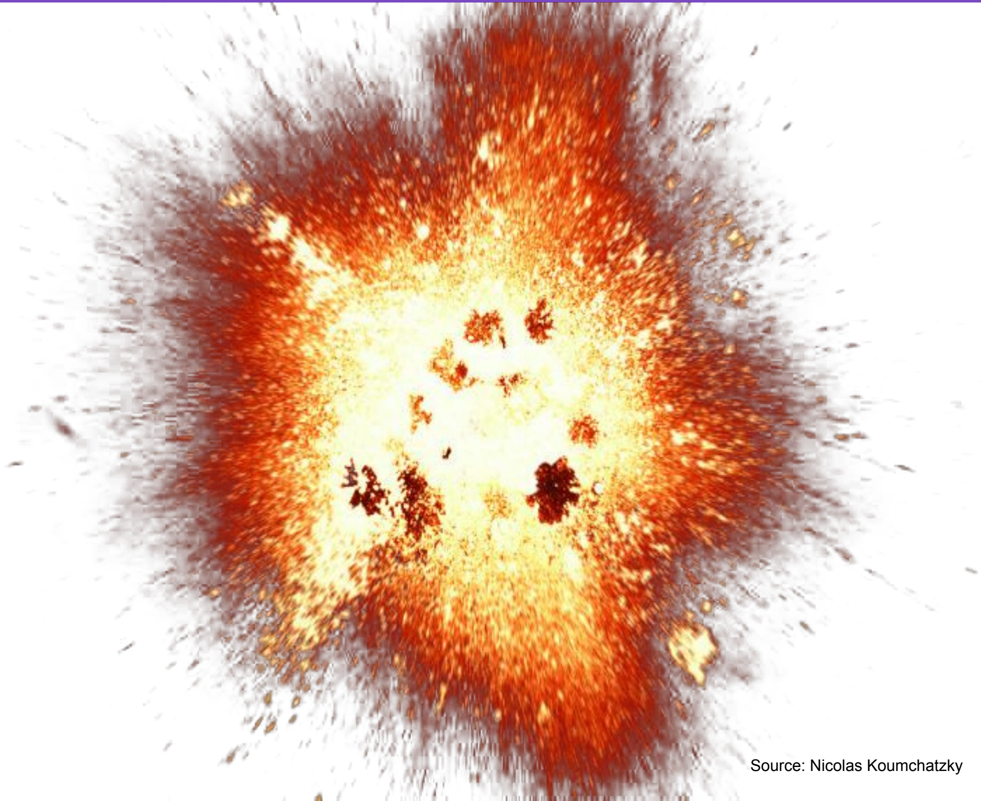
Input: input_feature (value == 1M)
⇒ weight_gradient == 1M
⇒ update = 1M * learning_rate
⇒ ?

Source: Nicolas Koumchatzky

Sparse Linear Layer: Online Normalization

- Example:

Input: input_feature (value == 1M)
⇒ weight_gradient == 1M
⇒ update = 1M * learning_rate
⇒



Source: Nicolas Koumchatzky

Sparse Linear Layer: Online Normalization

- Normalization of input values

$$norm(V_i) == \boxed{V_i / max(|V_i|)} + \boxed{b_i}$$

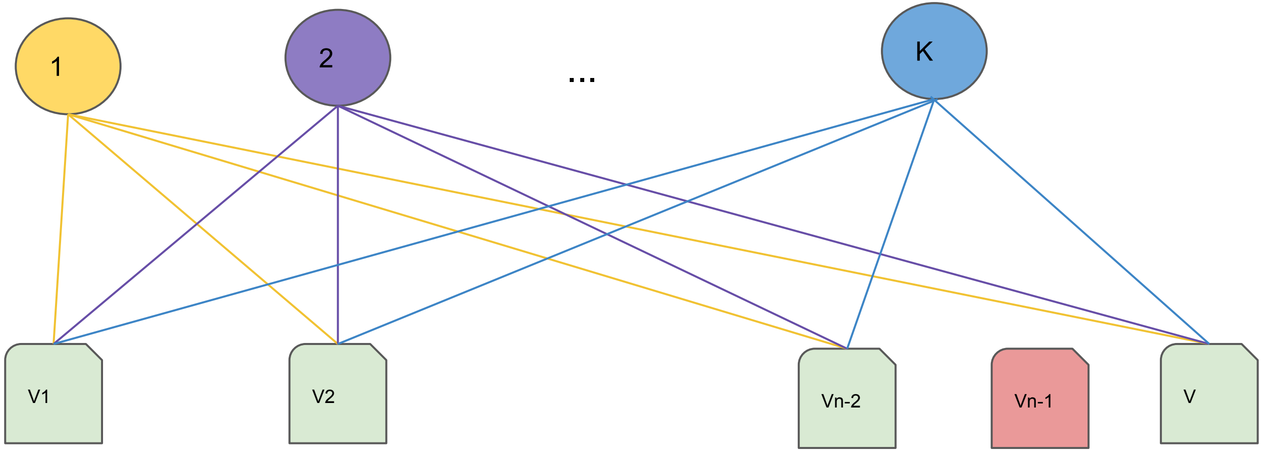
Belongs to [-1, 1]

Trainable per-feature bias: discriminate absence and presence of features

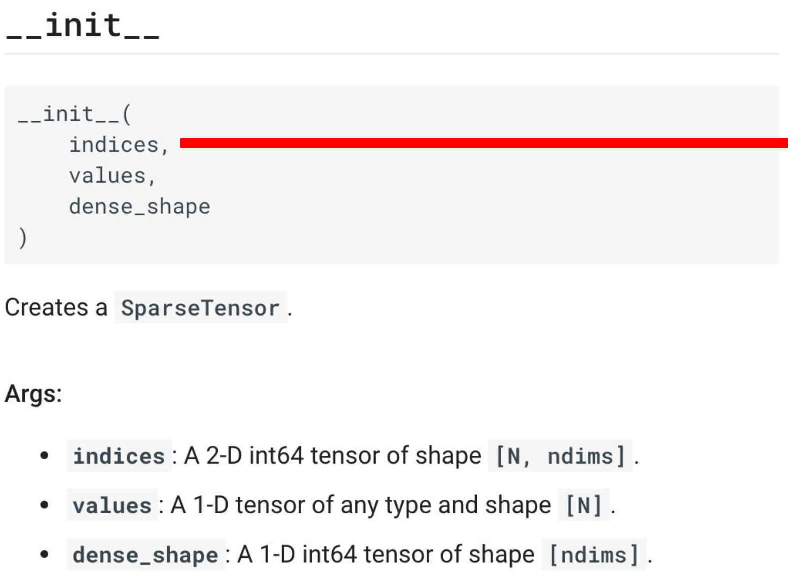
Source: Nicolas Koumchatzky

Modeling: Sparse Linear Layer

$$N_j = F(\sum Wi,j * norm(V_i) + B_j)$$



Sparse Linear Layer: Batching Data Records



__init__

```
__init__(
    indices,
    values,
    dense_shape
)
```

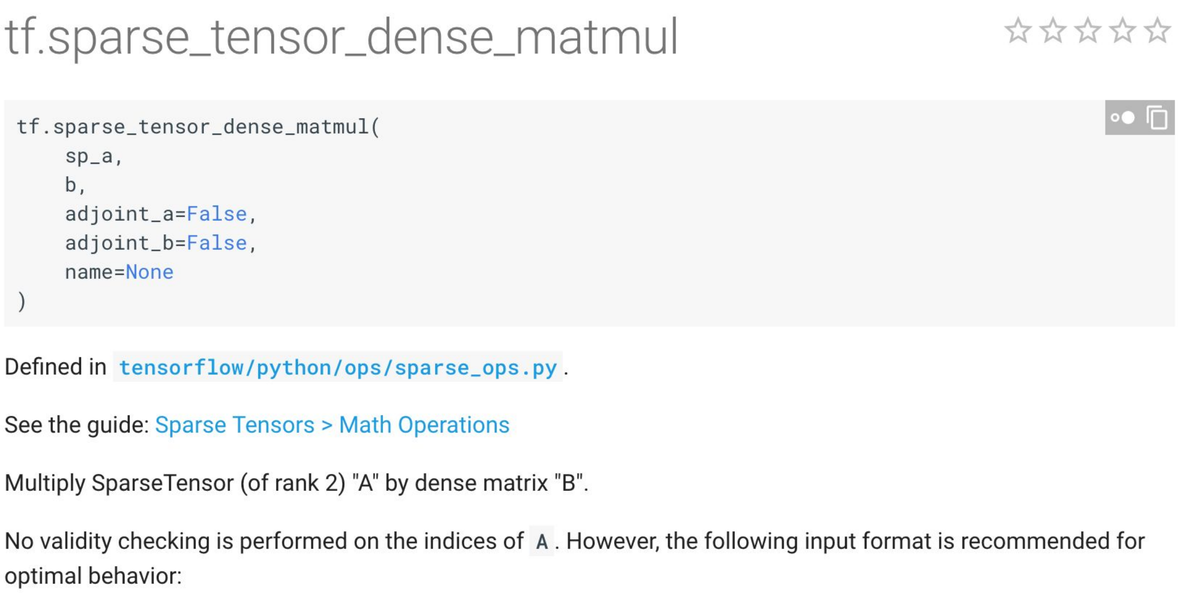
Creates a `SparseTensor`.

Args:

- `indices`: A 2-D int64 tensor of shape `[N, ndims]`.
- `values`: A 1-D tensor of any type and shape `[N]`.
- `dense_shape`: A 1-D int64 tensor of shape `[ndims]`.

Indices[:,0] = sample indices
Indices[:,1] = feature keys

Sparse Linear Layer: First Approach



tf.sparse_tensor_dense_matmul

```
tf.sparse_tensor_dense_matmul(
    sp_a,
    b,
    adjoint_a=False,
    adjoint_b=False,
    name=None
)
```

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

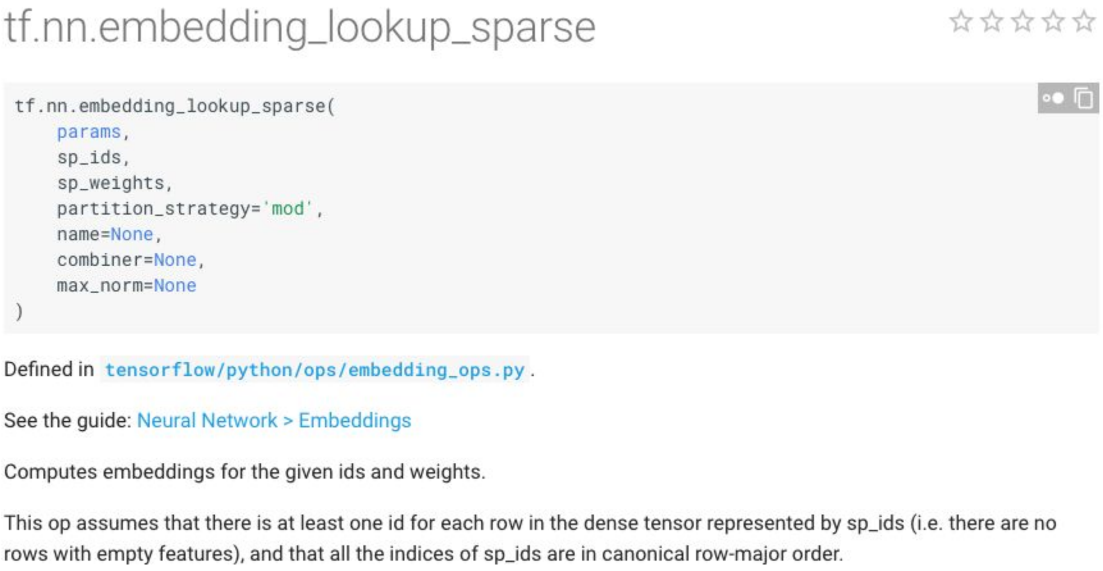
See the guide: Sparse Tensors > Math Operations

Multiply SparseTensor (of rank 2) "A" by dense matrix "B".

No validity checking is performed on the indices of `A`. However, the following input format is recommended for optimal behavior:

Source: Tensorflow

Sparse Linear Layer: Final Approach



tf.nn.embedding_lookup_sparse

```
tf.nn.embedding_lookup_sparse(
    params,
    sp_ids,
    sp_weights,
    partition_strategy='mod',
    name=None,
    combiner=None,
    max_norm=None
)
```

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

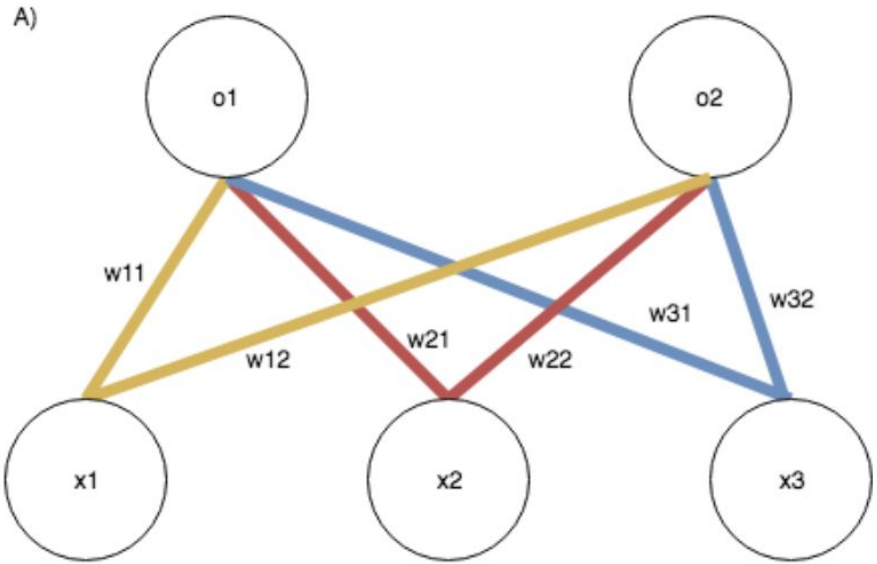
See the guide: Neural Network > Embeddings

Computes embeddings for the given ids and weights.

This op assumes that there is at least one id for each row in the dense tensor represented by sp_ids (i.e. there are no rows with empty features), and that all the indices of sp_ids are in canonical row-major order.

Source: Tensorflow

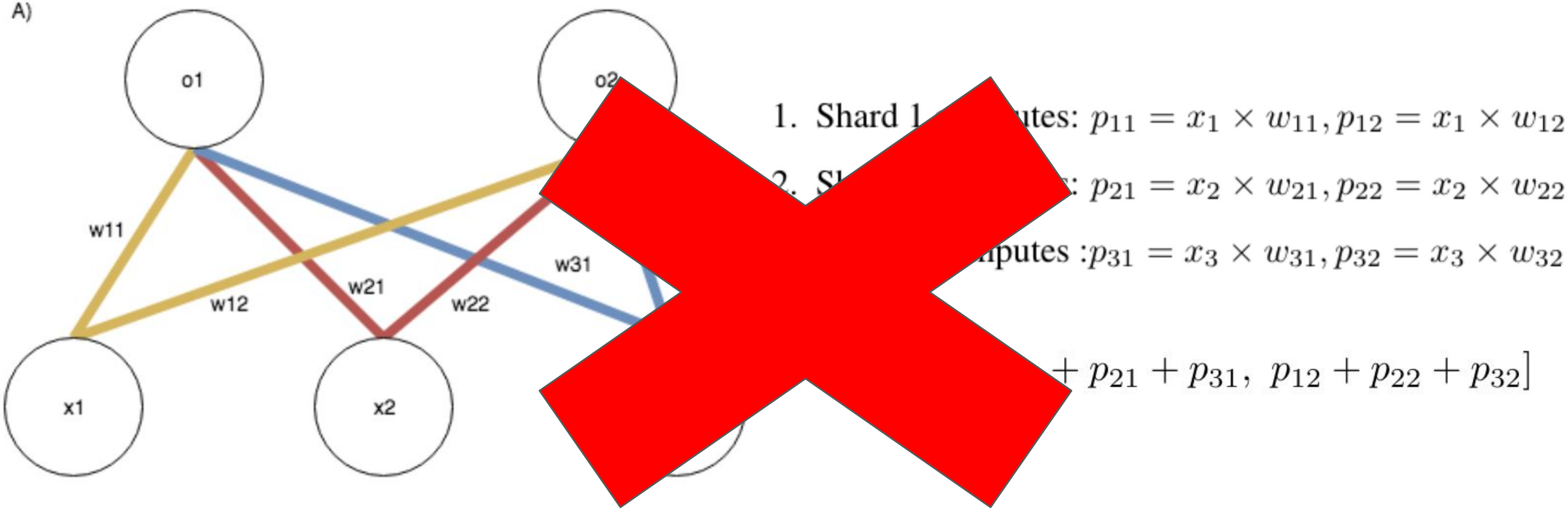
Sparse Linear Layer: Variable Partitioning



1. Shard 1 computes: $p_{11} = x_1 \times w_{11}, p_{12} = x_1 \times w_{12}$
2. Shard 2 computes: $p_{21} = x_2 \times w_{21}, p_{22} = x_2 \times w_{22}$
3. Shard 3 computes :$p_{31} = x_3 \times w_{31}, p_{32} = x_3 \times w_{32}$

output: $[p_{11} + p_{21} + p_{31},\ p_{12} + p_{22} + p_{32}]$

Sparse Linear Layer: Variable Partitioning



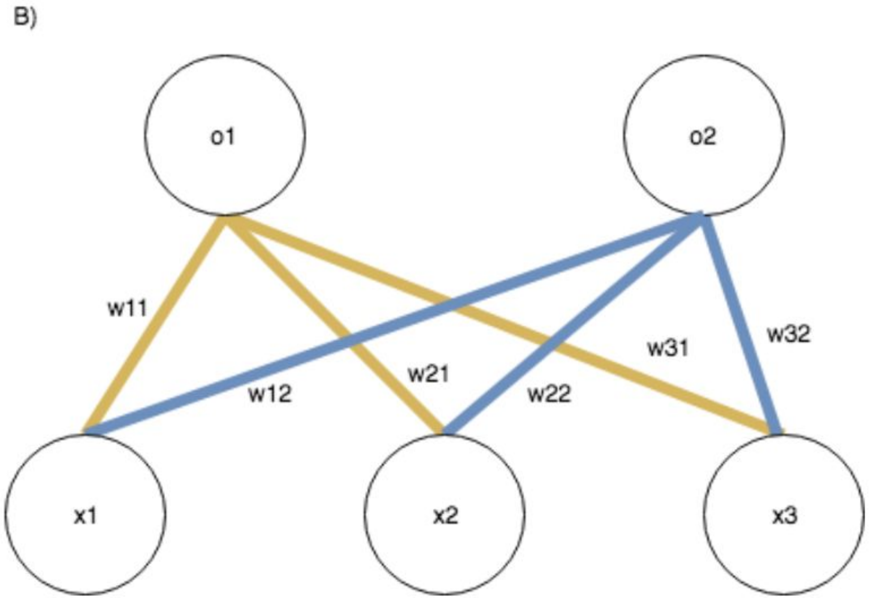
1. Shard 1 utes: $p_{11} = x_1 \times w_{11}, p_{12} = x_1 \times w_{12}$

2. Sl : $p_{21} = x_2 \times w_{21}, p_{22} = x_2 \times w_{22}$

nputes :$p_{31} = x_3 \times w_{31}, p_{32} = x_3 \times w_{32}$

$+ p_{21} + p_{31},\ p_{12} + p_{22} + p_{32}]$

Sparse Linear Layer: Variable Partitioning

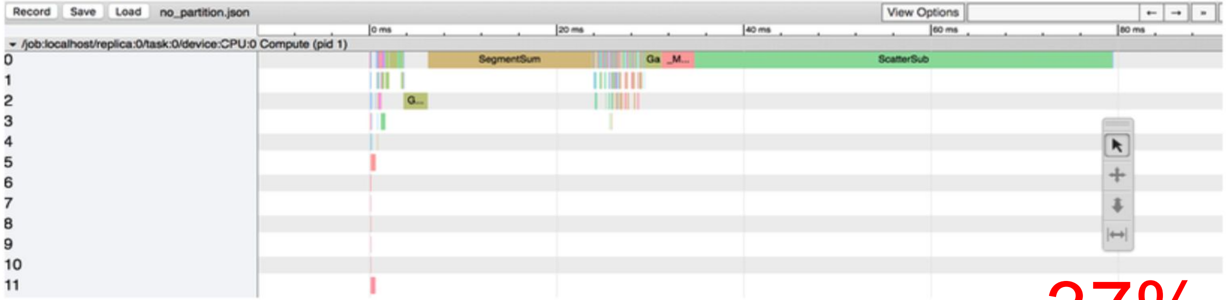


Shard 1 computes: $o_1 = x_1 \times w_{11} + x_2 \times w_{21} + x_3 \times w_{31}$

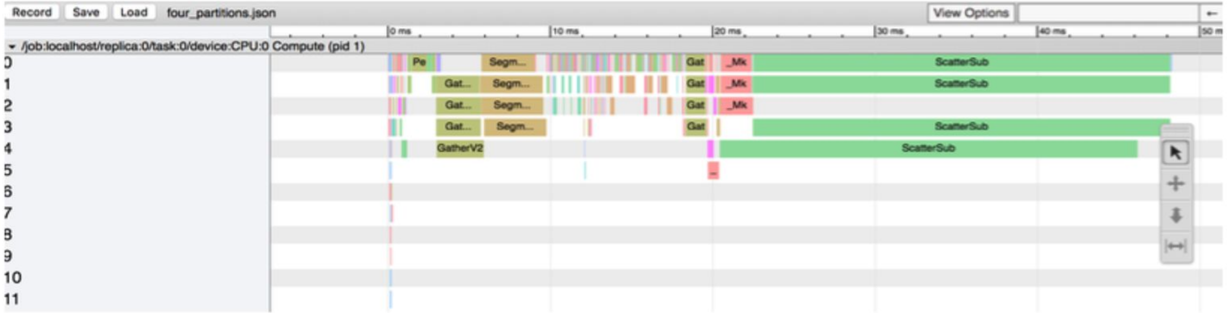
Shard 2 computes: $o_2 = x_1 \times w_{12} + x_2 \times w_{22} + x_3 \times w_{32}$

output: $[o_1 , o_2]$

Sparse Linear Layer: Variable Partitioning: Profiling



~37% reduction

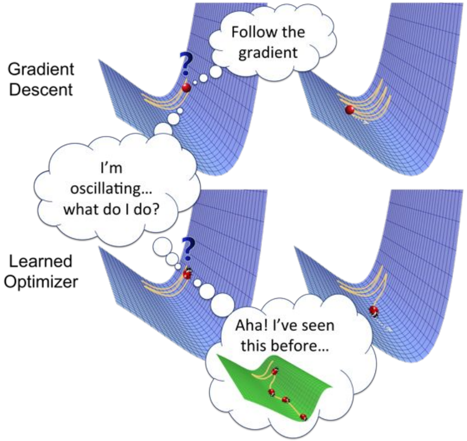


Sparse Linear Layer

- Optimizers
  - SGD
  - Lazy Adam

Adam:

“Velocity”
“Momentum”

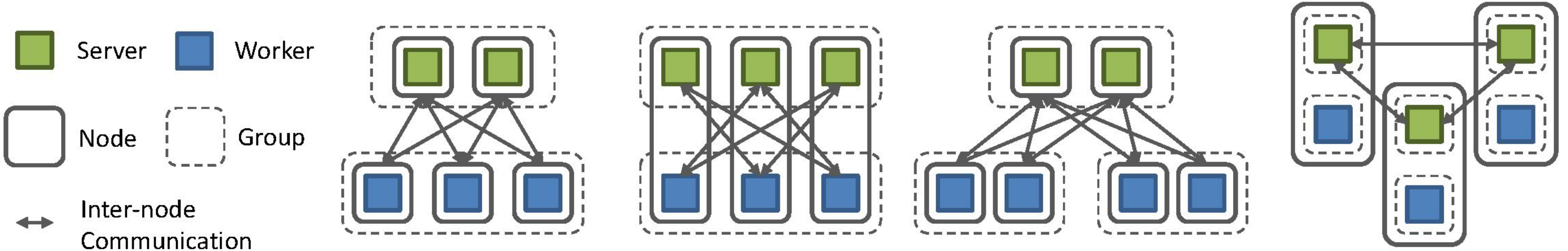

Gradient Descent
Follow the gradient
I’m oscillating... what do I do?
Learned Optimizer
Aha! I’ve seen this before...

Source: Berkeley Research Artificial Lab

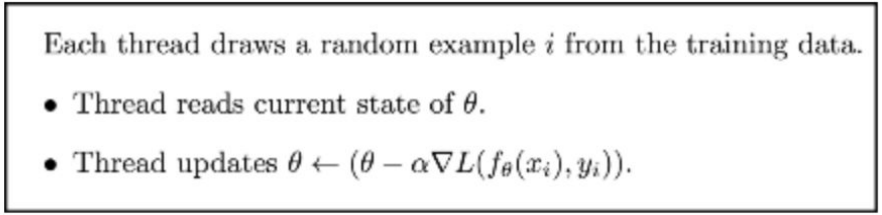
Additional Performance Gains

Hogwild

“Remove all thread locks from parallel SGD code.”



(a) Sandblaster. (b) AllReduce. (c) Downpour. (d) Distributed Hogwild.

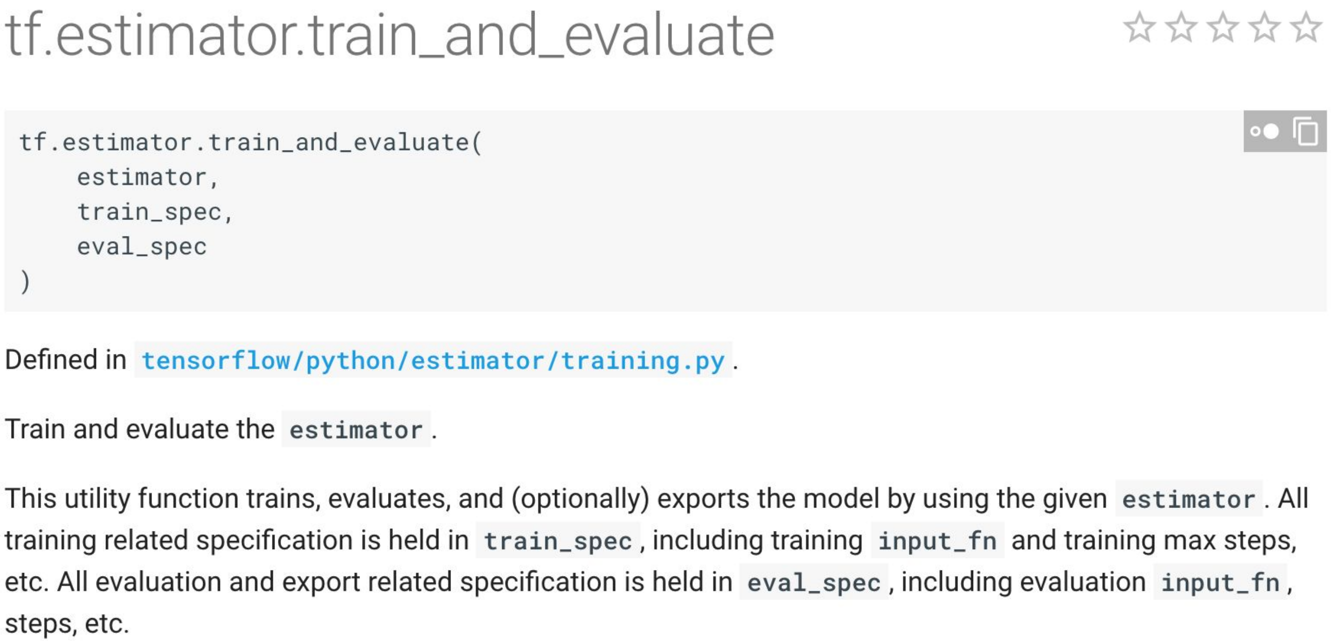


Each thread draws a random example i from the training data.

- Thread reads current state of θ.
- Thread updates $\theta \leftarrow (\theta - \alpha \nabla L(f_\theta(x_i), y_i))$.

Source: Hogwild!, UC Berkeley

Hogwild



tf.estimator.train_and_evaluate

```
tf.estimator.train_and_evaluate(
    estimator,
    train_spec,
    eval_spec
)
```

Defined in `tensorflow/python/estimator/training.py`.

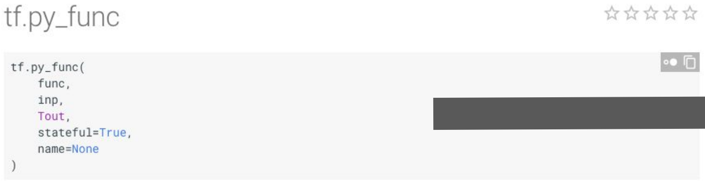
Train and evaluate the `estimator`.

This utility function trains, evaluates, and (optionally) exports the model by using the given `estimator`. All training related specification is held in `train_spec`, including training `input_fn` and training max steps, etc. All evaluation and export related specification is held in `eval_spec`, including evaluation `input_fn`, steps, etc.

Source: Tensorflow

Performance Gains

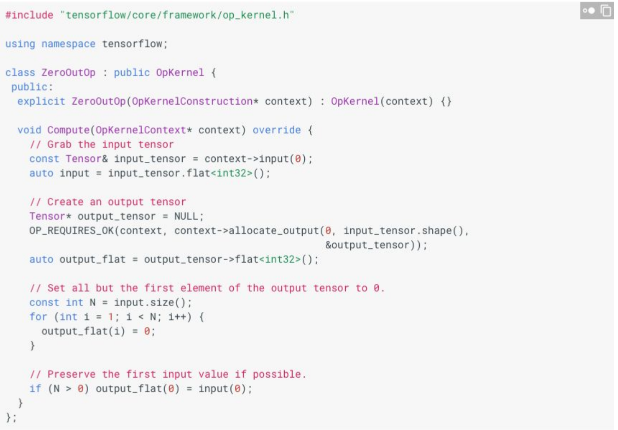
Custom Ops



tf.py_func

☆☆☆☆☆

```python
tf.py_func(
    func,
    inp,
    Tout,
    stateful=True,
    name=None
)
```



```cpp
#include "tensorflow/core/framework/op_kernel.h"

using namespace tensorflow;

class ZeroOutOp : public OpKernel {
 public:
  explicit ZeroOutOp(OpKernelConstruction* context) : OpKernel(context) {}

  void Compute(OpKernelContext* context) override {
    // Grab the input tensor
    const Tensor& input_tensor = context->input(0);
    auto input = input_tensor.flat<int32>();

    // Create an output tensor
    Tensor* output_tensor = NULL;
    OP_REQUIRES_OK(context, context->allocate_output(0, input_tensor.shape(),
                                                     &output_tensor));
    auto output_flat = output_tensor->flat<int32>();

    // Set all but the first element of the output tensor to 0.
    const int N = input.size();
    for (int i = 1; i < N; i++) {
      output_flat(i) = 0;
    }

    // Preserve the first input value if possible.
    if (N > 0) output_flat(0) = input(0);
  }
};
```

GPU x CPU metrics

CPU/GPU Benchmarks: Remarks

- TensorFlow 1.10 compiled with MKL enabled
- Disabled MKL’s multithreading by exporting OMP NUM THREADS=1
- Tweak parallelism: inter_op_parallelism_threads/intra_op_parallelism_threads

GPU Benchmarks

Batch Size/Model Optimization	CPU: Baseline (tf.sparse_tensor_dense_matmul)	CPU: After optimizations	GPU: Baseline (tf.sparse_tensor_dense_matmul)	GPU: After Optimizations
256	1024 samples/s	7372 samples/s	5504 samples/s	22528 samples/s
512	1638 samples/s	11264 samples/s	8448 samples/s	21504 samples/s
1024	2355 samples/s	13312 samples/s	10752 samples/s	22528 samples/s

GPU benchmarks were run with NVIDIA Tesla K80 Processors
CPU benchmarks were run with Intel Xenon Platinum 8180 Processors

Acknowledgements

- Andrew Bean
- Ricardo Cervera-Navarro
- Priyank Jain
- Ruhua Jiang
- Nicholas Léonard
- Briac Marcatté
- Mahak Patidar
- Tim Sweeney
- Pavan Yalamanchili
- Yi Zhuang

Thank you!
Questions?

Follow me on Twitter: @cibelemh
Email me at: cmontezhalasz@twitter.com