Machine Learning Course at MIPT

Modern DL Frameworks, CNN in Practice

Valentin Malykh

ml-mipt.github.io, val.maly.hk

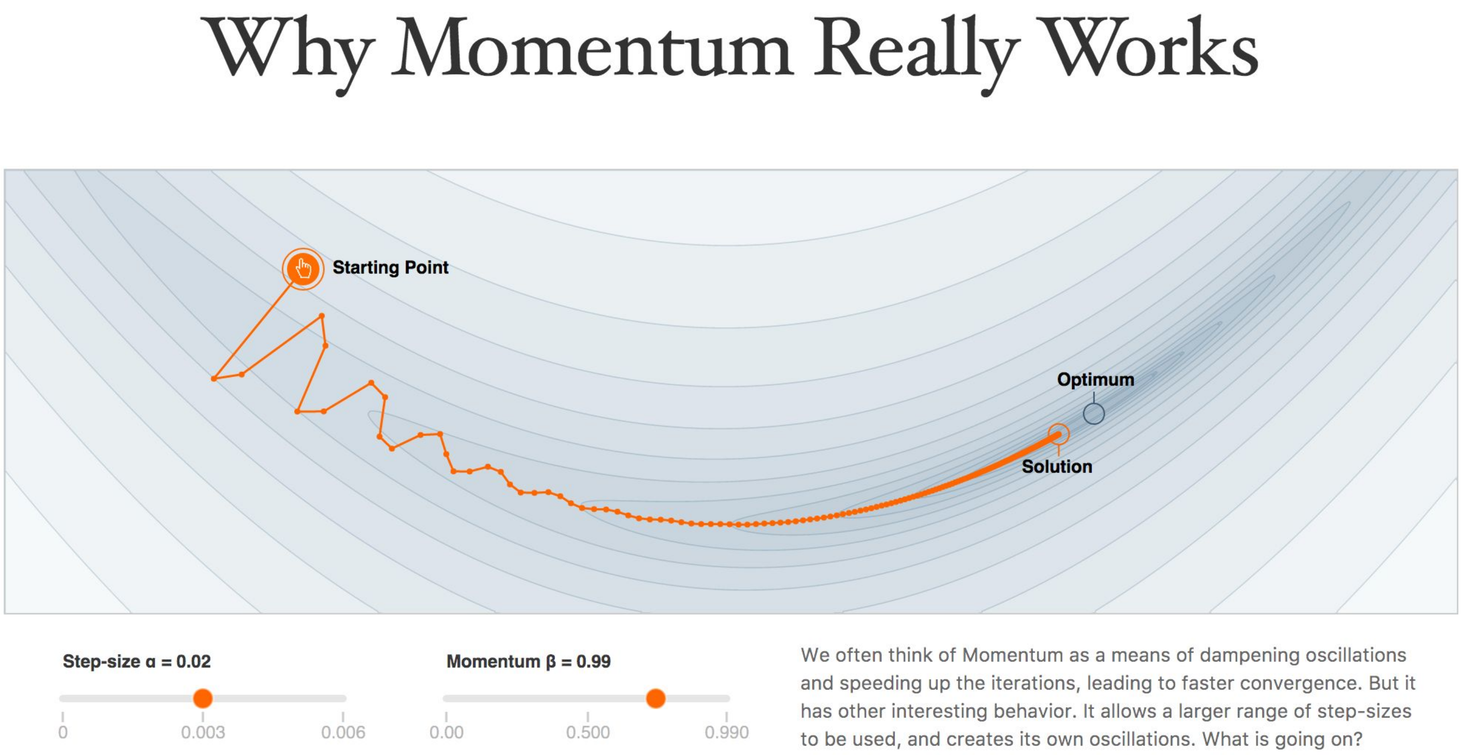
MIPT
HIGHER SCHOOL OF SYSTEMS ENGINEERING

iPavlov.ai

April 17th, 2018

Credit A. Karpathy, Lempitsky

Just Fun



Why Momentum Really Works

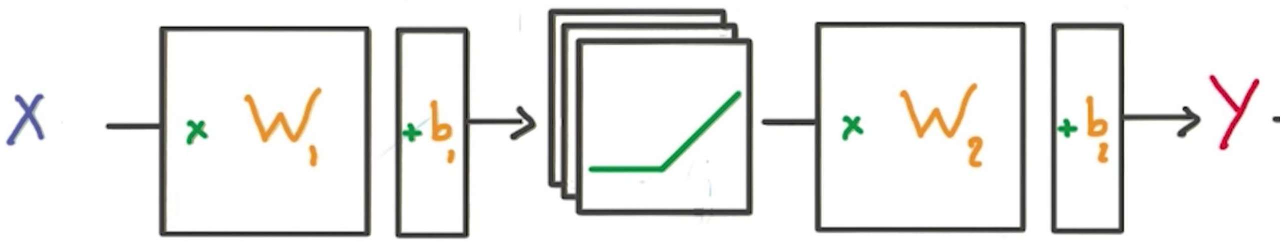
Step-size α = 0.02

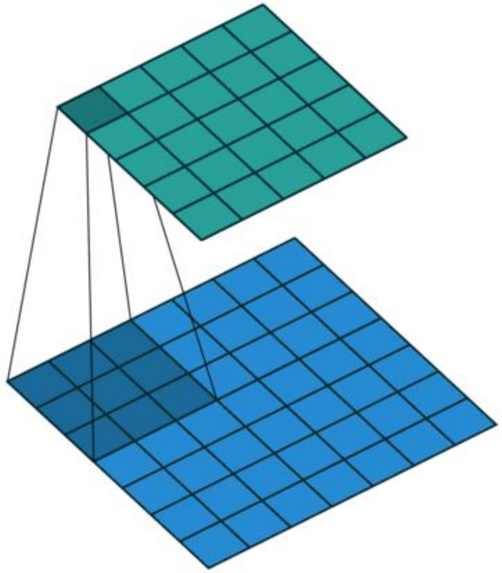
Momentum β = 0.99

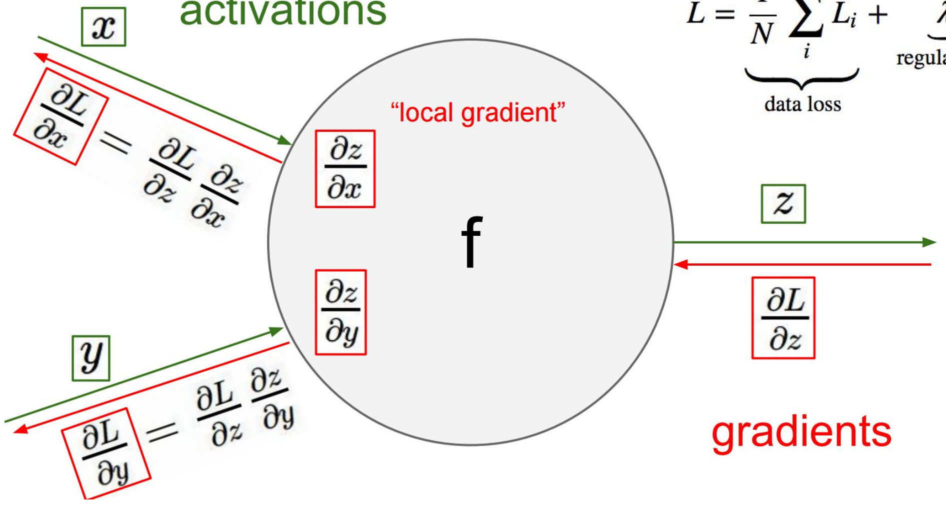
We often think of Momentum as a means of dampening oscillations and speeding up the iterations, leading to faster convergence. But it has other interesting behavior. It allows a larger range of step-sizes to be used, and creates its own oscillations. What is going on?

http://distill.pub/2017/momentum/

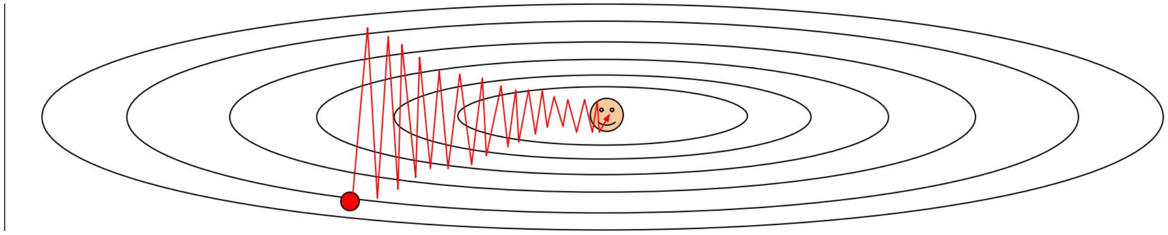
Last Time

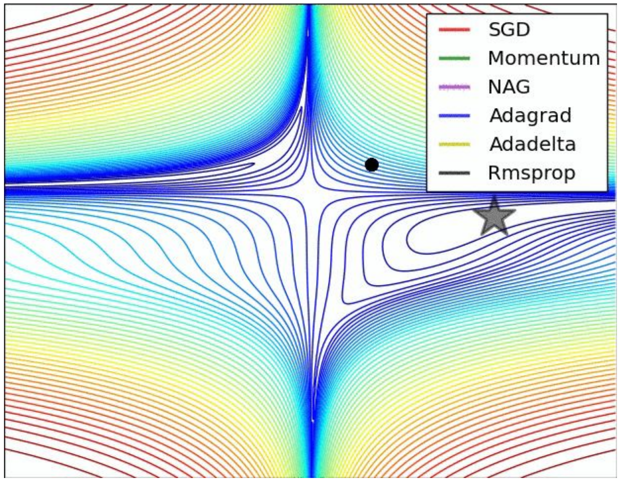






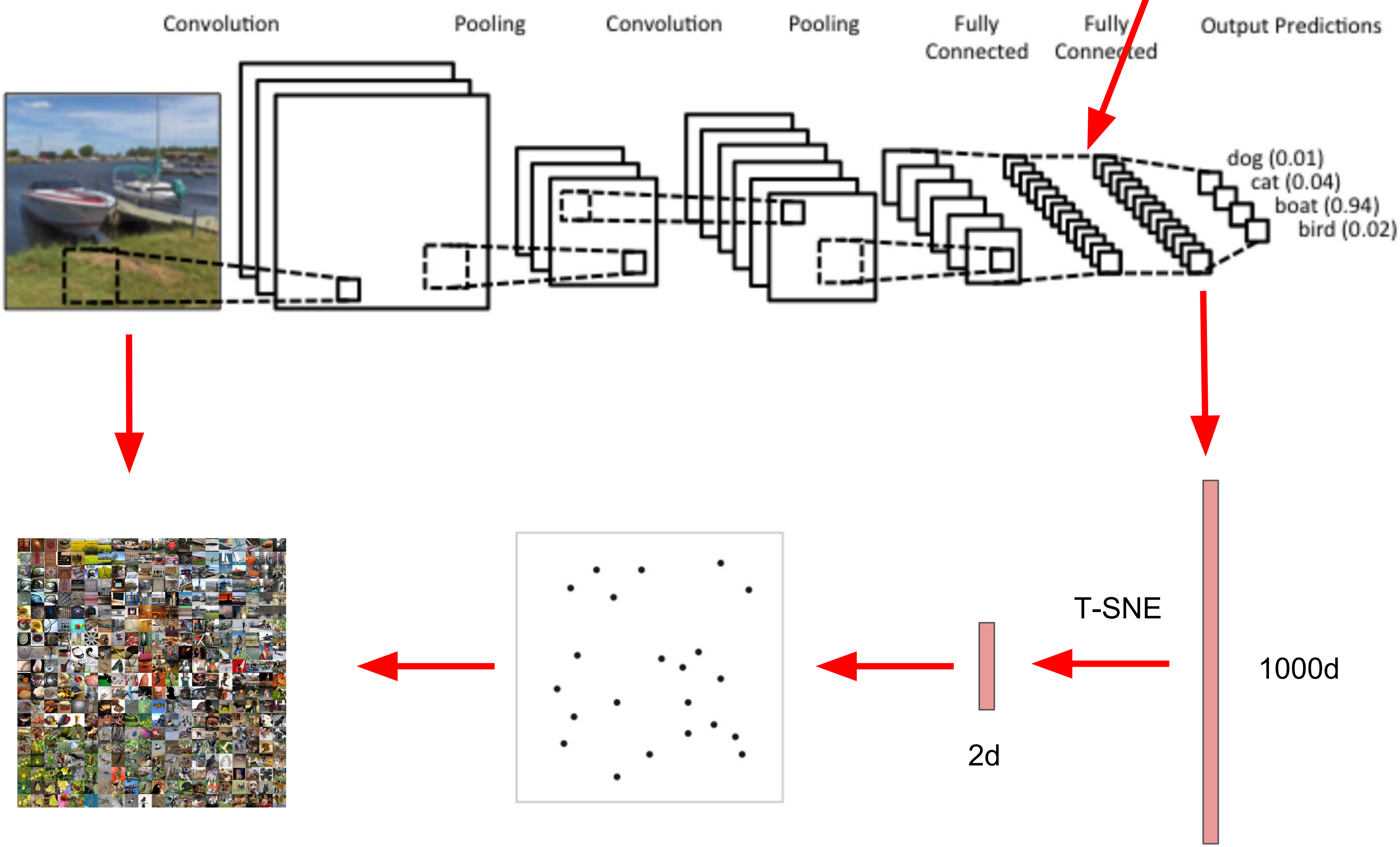
$$L = \underbrace{\frac{1}{N}\sum_i L_i}_{\text{data loss}} + \underbrace{\lambda R(W)}_{\text{regularization loss}}$$





Why classification?

High Level Representation



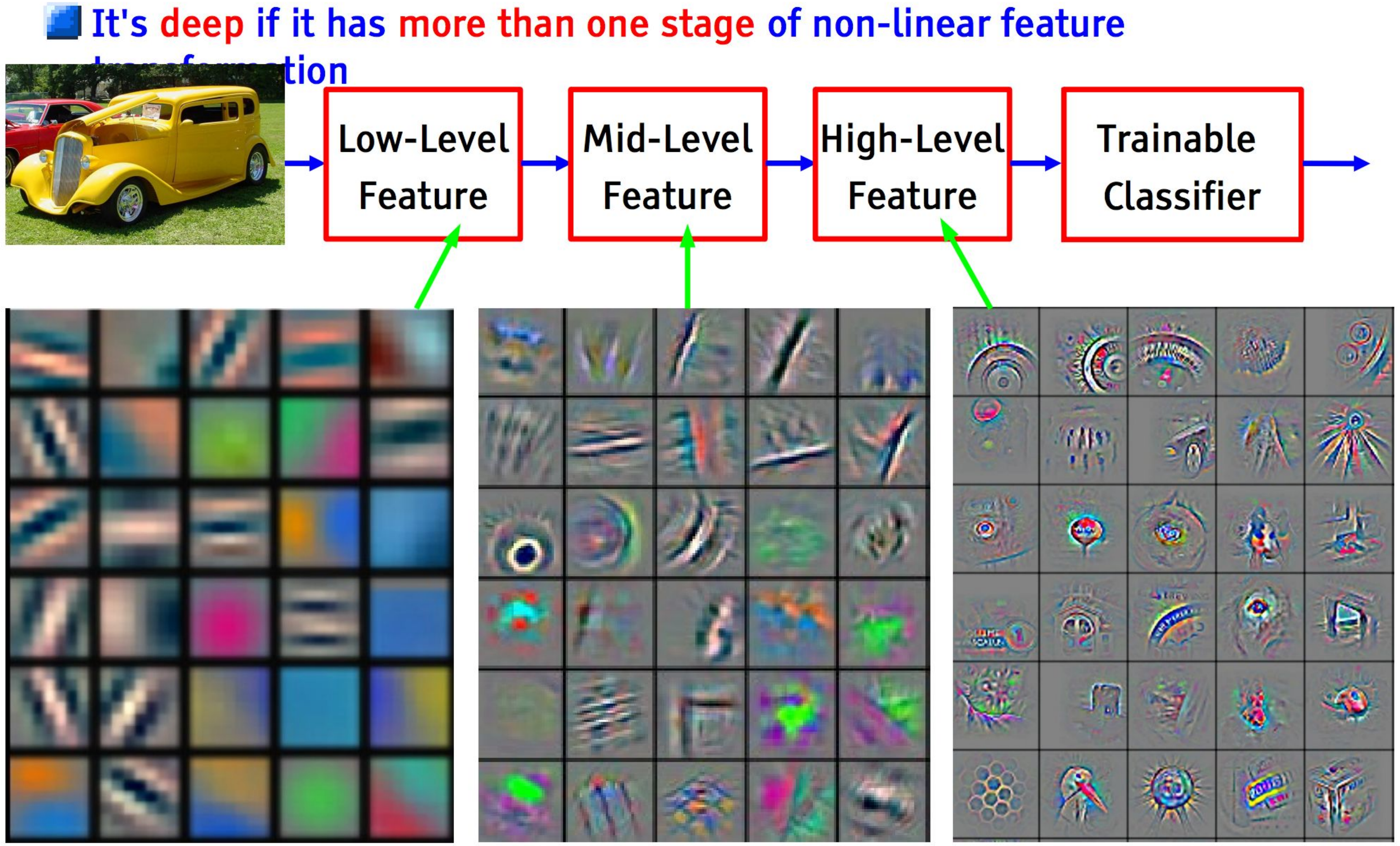
T-SNE

1000d

2d

http://cs.stanford.edu/people/karpathy/cnnembed/

Representation and Trainable



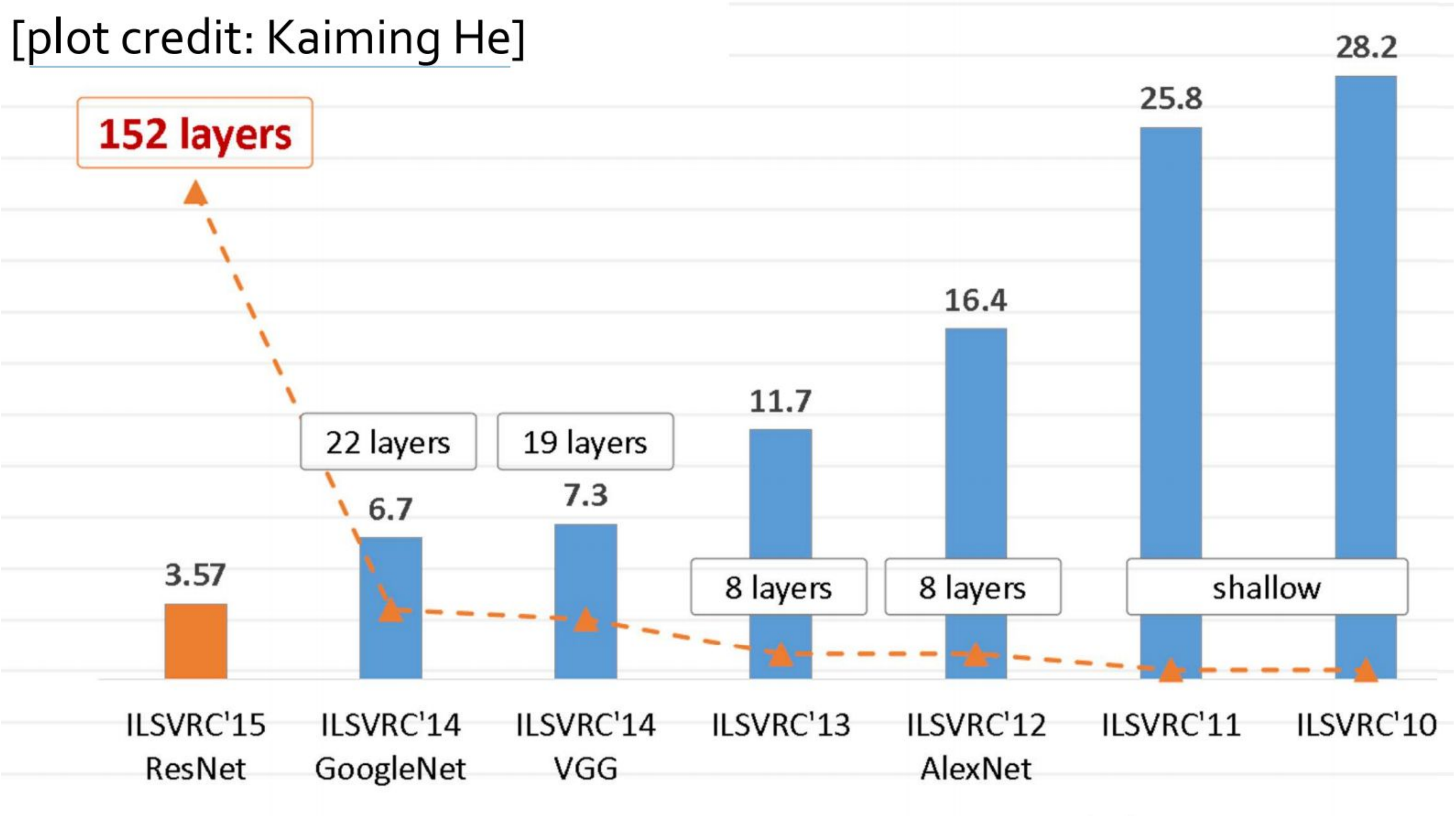
- It's deep if it has more than one stage of non-linear feature transformation

Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

http://www.cs.nyu.edu/~yann/talks/lecun-ranzato-icml2013.pdf

Modern Conv Arch

[plot credit: Kaiming He]



PYTORCH

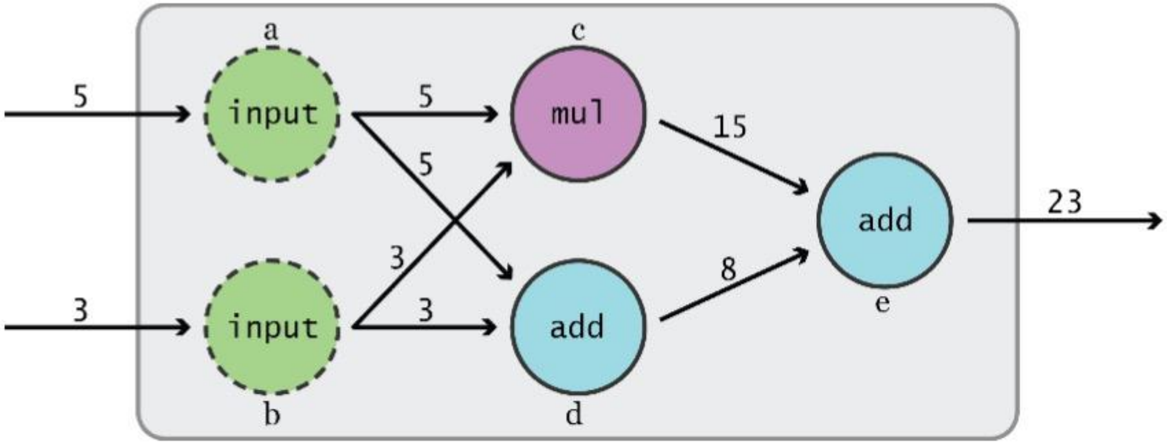
TensorFlow

mxnet

Graphs

Data Flow Graph

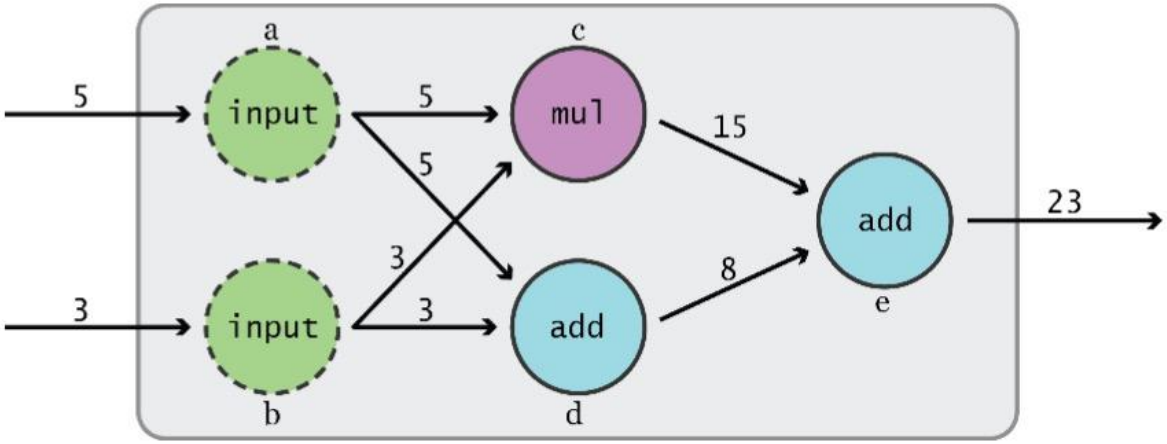
TensorFlow separates definition of computations from their execution



Data Flow Graph

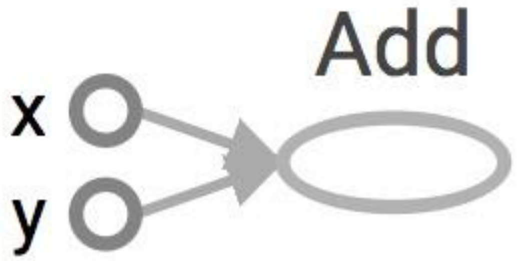
Phase 1: assemble a graph

Phase 2: use a session to execute operations in the graph.



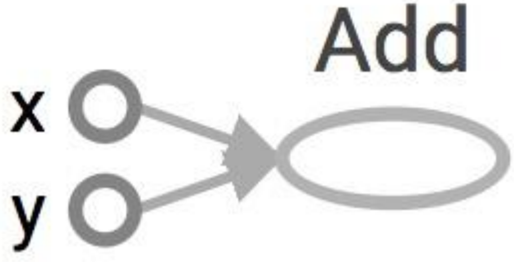
Data Flow Graph

```
import tensorflow as tf
a = tf.add(3, 5)
```



Data Flow Graph

```
import tensorflow as tf
a = tf.add(3, 5)
```



Why x, y?

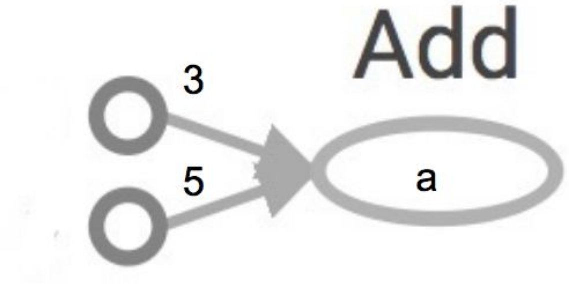
TF automatically names the nodes when you don’t explicitly name them.

x = 3

y = 5

Data Flow Graph

```
import tensorflow as tf
a = tf.add(3, 5)
```

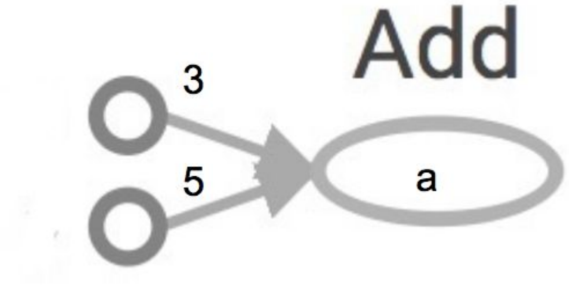


Nodes: operators, variables, and constants

Edges: tensors

Data Flow Graph

```
import tensorflow as tf
a = tf.add(3, 5)
```



Nodes: operators, variables, and constants

Edges: tensors

Tensors are data.

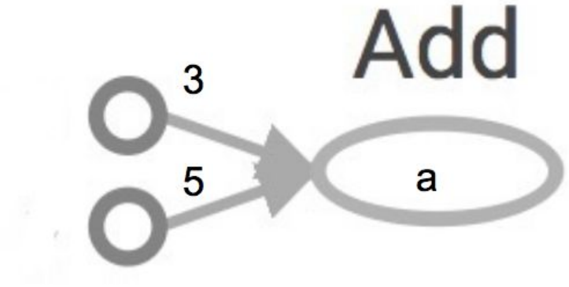
TensorFlow = tensor + flow = data + flow

(I know, mind=blown)

Data Flow Graph

```
import tensorflow as tf
a = tf.add(3, 5)
```

print(a)



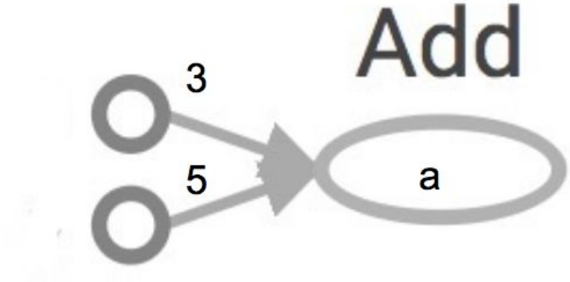
```
>> Tensor("Add:0", shape=(), dtype=int32)
```

(Not 8)

How to get the value of a?

Create a `session`, assign it to variable `sess` so we can call it later

Within the `session`, evaluate the graph to fetch the value of a



How to get the value of a?

Create a `session`, assign it to variable `sess` so we can call it later

Within the `session`, evaluate the graph to fetch the value of a

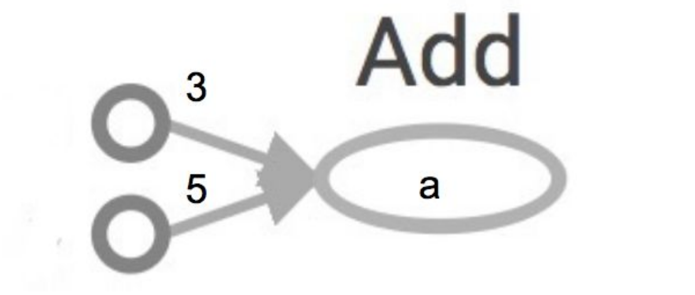
```
import tensorflow as tf

a = tf.add(3, 5)

sess = tf.Session()

print(sess.run(a))

sess.close()
```



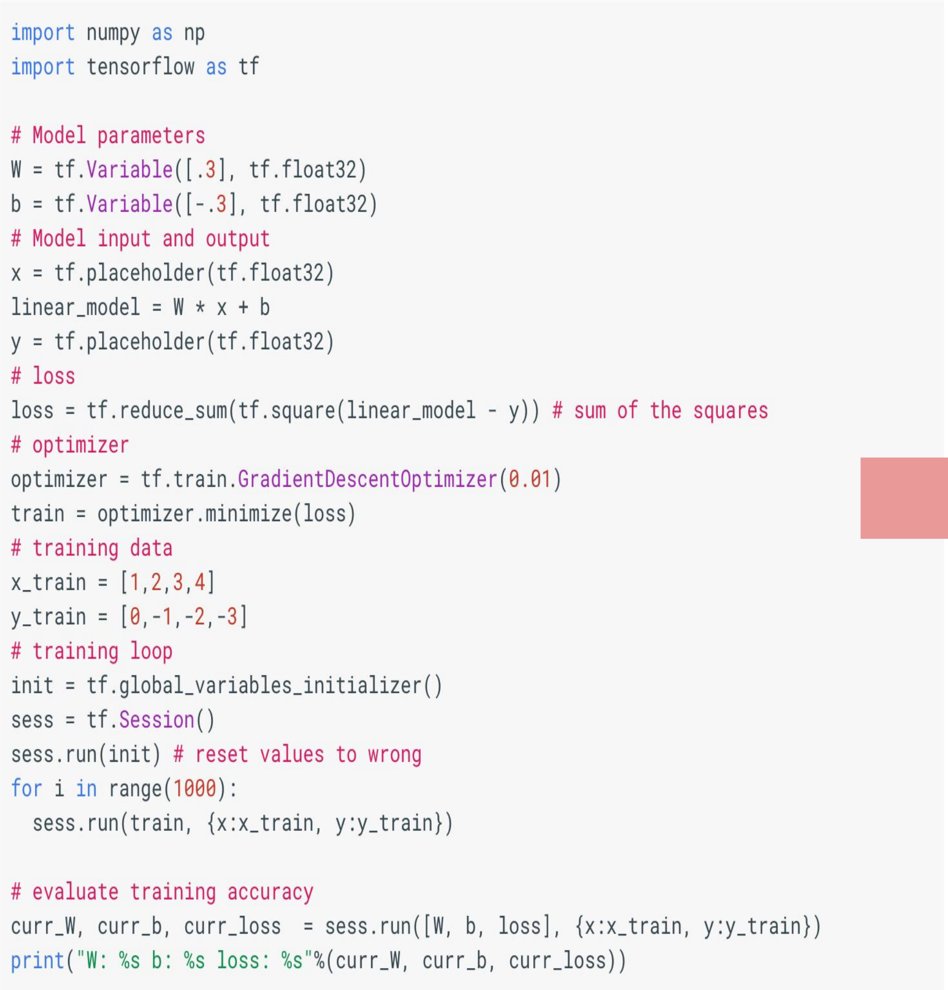
tf.Session()

A Session object encapsulates the environment in which Operation objects are executed, and Tensor objects are evaluated.

Session will also allocate memory to store the current values of variables.

TensorFlow

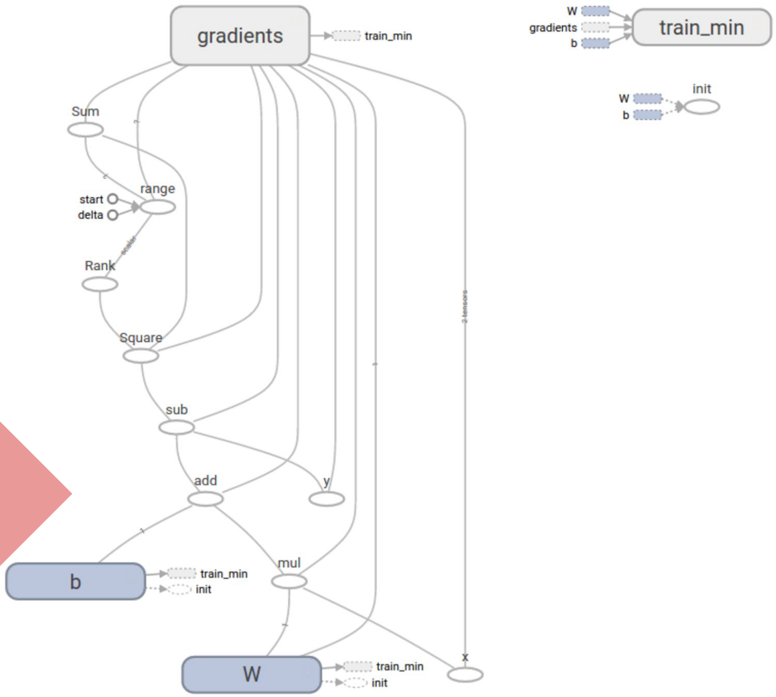
TensorFlow Today: Declarative (Graphs)



```
import numpy as np
import tensorflow as tf

# Model parameters
W = tf.Variable([.3], tf.float32)
b = tf.Variable([-.3], tf.float32)
# Model input and output
x = tf.placeholder(tf.float32)
linear_model = W * x + b
y = tf.placeholder(tf.float32)
# loss
loss = tf.reduce_sum(tf.square(linear_model - y)) # sum of the squares
# optimizer
optimizer = tf.train.GradientDescentOptimizer(0.01)
train = optimizer.minimize(loss)
# training data
x_train = [1,2,3,4]
y_train = [0,-1,-2,-3]
# training loop
init = tf.global_variables_initializer()
sess = tf.Session()
sess.run(init) # reset values to wrong
for i in range(1000):
  sess.run(train, {x:x_train, y:y_train})

# evaluate training accuracy
curr_W, curr_b, curr_loss  = sess.run([W, b, loss], {x:x_train, y:y_train})
print("W: %s b: %s loss: %s"%(curr_W, curr_b, curr_loss))
```



Graphs are ...

Optimizable

- automatic buffer reuse
- constant folding
- inter-op parallelism
- automatic trade-off between compute and memory

Deployable

- the Graph is an intermediate representation for models

Rewritable

- experiment with automatic device placement or quantization

But graphs are also ...

Difficult to debug

- errors are reported long after graph construction
- execution cannot be debugged with `pdb` or print statements

Un-Pythonic

- writing a TensorFlow program is an exercise in metaprogramming
- control flow (e.g., `tf.while_loop`) differs from Python
- can't easily mix graph construction with custom data structures

```
Traceback (most recent call last):
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1350, in _do_call
    return fn(*args)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1329, in _run_fn
    status, run_metadata)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/framework/errors_impl.py", line 473, in __exit__
    c_api.TF_GetCode(self.status.status))
tensorflow.python.framework.errors_impl.InvalidArgumentError: indices[0] = 3081 is not in [0, 128)
         [[Node: loss/nce_loss/embedding_lookup_1 = Gather[Tindices=DT_INT64, Tparams=DT_FLOAT, _class=["loc:@nce_bias"], validate_indices=true, _device="/job:localhost/replica:0/task:0/device:CPU:0"](nce_bias/read, loss/nce_loss/concat)]]

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "04_word2vec.py", line 102, in <module>
    main()
  File "04_word2vec.py", line 99, in main
    word2vec(dataset)
  File "04_word2vec.py", line 82, in word2vec
    loss_batch, _ = sess.run([loss, optimizer])
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 895, in run
    run_metadata_ptr)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1128, in _run
    feed_dict_tensor, options, run_metadata)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1344, in _do_run
    options, run_metadata)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1363, in _do_call
    raise type(e)(node_def, op, message)
tensorflow.python.framework.errors_impl.InvalidArgumentError: indices[0] = 3081 is not in [0, 128)
         [[Node: loss/nce_loss/embedding_lookup_1 = Gather[Tindices=DT_INT64, Tparams=DT_FLOAT, _class=["loc:@nce_bias"], validate_indices=true, _device="/job:localhost/replica:0/task:0/device:CPU:0"](nce_bias/read, loss/nce_loss/concat)]]

Caused by op 'loss/nce_loss/embedding_lookup_1', defined at:
  File "04_word2vec.py", line 102, in <module>
    main()
  File "04_word2vec.py", line 99, in main
    word2vec(dataset)
  File "04_word2vec.py", line 65, in word2vec
    num_classes=VOCAB_SIZE), name='loss')
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/nn_impl.py", line 1212, in nce_loss
    name=name)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/nn_impl.py", line 1046, in _compute_sampled_logits
    biases, all_ids, partition_strategy=partition_strategy)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/embedding_ops.py", line 325, in embedding_lookup
    transform_fn=None)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/embedding_ops.py", line 150, in _embedding_lookup_and_transform
    result = _clip(_gather(params[0], ids, name=name), ids, max_norm)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/embedding_ops.py", line 54, in _gather
    return array_ops.gather(params, ids, name=name)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/array_ops.py", line 2585, in gather
    params, indices, validate_indices=validate_indices, name=name)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/gen_array_ops.py", line 1864, in gather
    validate_indices=validate_indices, name=name)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/framework/op_def_library.py", line 787, in _apply_op_helper
    op_def=op_def)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/framework/ops.py", line 3160, in create_op
    op_def=op_def)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/framework/ops.py", line 1625, in __init__
    self._traceback = self._graph._extract_stack()  # pylint: disable=protected-access

InvalidArgumentError (see above for traceback): indices[0] = 3081 is not in [0, 128)
         [[Node: loss/nce_loss/embedding_lookup_1 = Gather[Tindices=DT_INT64, Tparams=DT_FLOAT, _class=["loc:@nce_bias"], validate_indices=true, _device="/job:localhost/replica:0/task:0/device:CPU:0"](nce_bias/read, loss/nce_loss/concat)]]
```

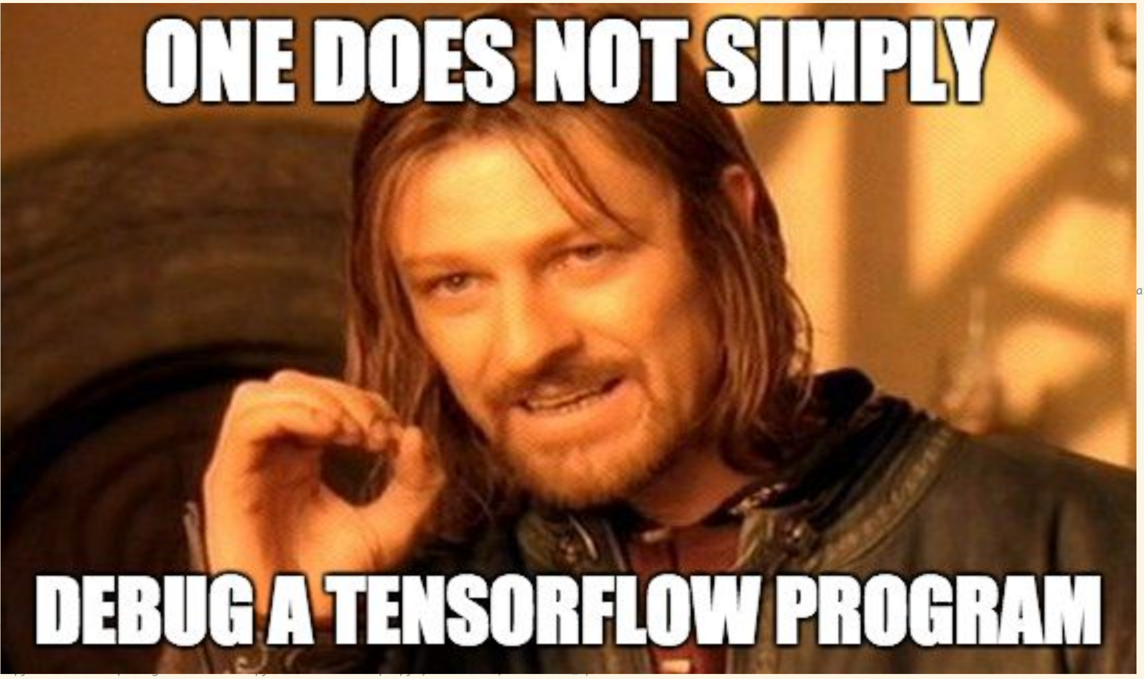
```
Traceback (most recent call last):
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1350, in _do_call
    return fn(*args)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1329, in _run_fn
    status, run_metadata)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/framework/errors_impl.py", line 473, in __exit__
    c_api.TF_GetCode(self.status.status))
tensorflow.python.framework.errors_impl.InvalidArgumentError: indices[0] = 3081 is not in [0, 128)
         [[Node: loss/nce_loss/embedding_lookup_1 = Gather[Tindices=DT_INT64, Tparams=DT_FLOAT, _class=["loc:@nce_bias"], validate_indices=true, _device="/job:localhost/replica:0/task:0/device:CPU:0"](nce_bias/read, loss/nce_loss/concat)]]

During handling of the above exception, an

Traceback (most recent call last):
  File "04_word2vec.py", line 102, in <mod
    main()
  File "04_word2vec.py", line 99, in main
    word2vec(dataset)
  File "04_word2vec.py", line 82, in word2
    loss_batch, _ = sess.run([loss, optimi
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    run_metadata_ptr)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    feed_dict_tensor, options, run_metadat
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    options, run_metadata)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    raise type(e)(node_def, op, message)
tensorflow.python.framework.errors_impl.In
         [[Node: loss/nce_loss/embedding_l                                                                                                                                                                   as/read, loss/nce_loss/concat)]]

Caused by op 'loss/nce_loss/embedding_look
  File "04_word2vec.py", line 102, in <mod
    main()
  File "04_word2vec.py", line 99, in main
    word2vec(dataset)
  File "04_word2vec.py", line 65, in word2
    num_classes=VOCAB_SIZE), name='loss')
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    name=name)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    biases, all_ids, partition_strategy=pa
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    transform_fn=None)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    result = _clip(_gather(params[0], ids,
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    return array_ops.gather(params, ids, n
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    params, indices, validate_indices=vali
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    validate_indices=validate_indices, nam
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    op_def=op_def)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/li
    op_def=op_def)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/framework/ops.py", line 1625, in __init__
    self._traceback = self._graph._extract_stack()  # pylint: disable=protected-access

InvalidArgumentError (see above for traceback): indices[0] = 3081 is not in [0, 128)
         [[Node: loss/nce_loss/embedding_lookup_1 = Gather[Tindices=DT_INT64, Tparams=DT_FLOAT, _class=["loc:@nce_bias"], validate_indices=true, _device="/job:localhost/replica:0/task:0/device:CPU:0"](nce_bias/read, loss/nce_loss/concat)]]
```



What if...

You could execute TensorFlow operations imperatively, *directly from Python*?

Eager Execution

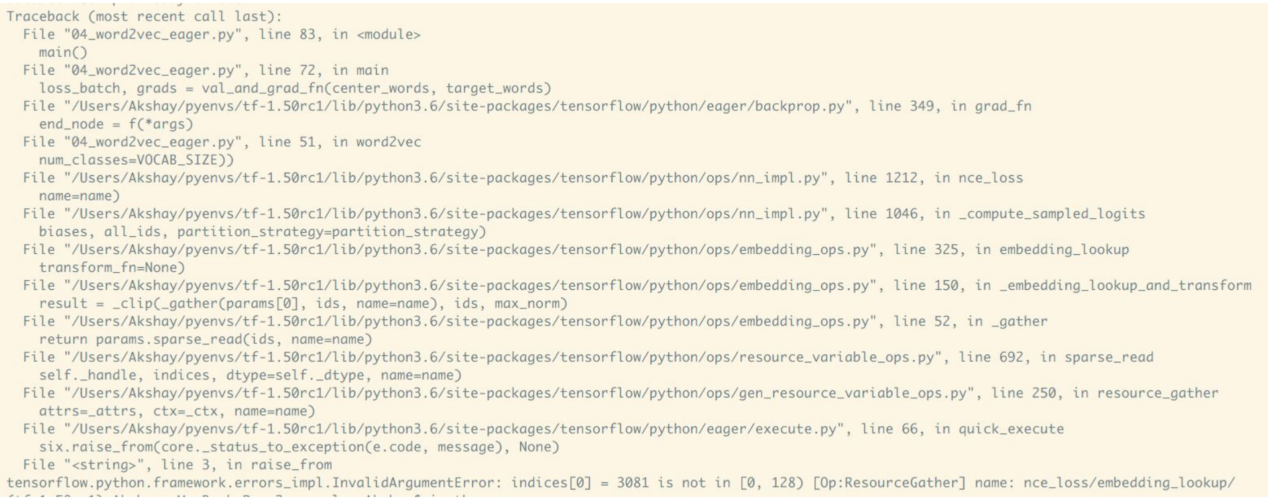
"A NumPy-like library for numerical computation with support for GPU acceleration and automatic differentiation, and a flexible platform for machine learning research and experimentation."

- the eager execution user guide

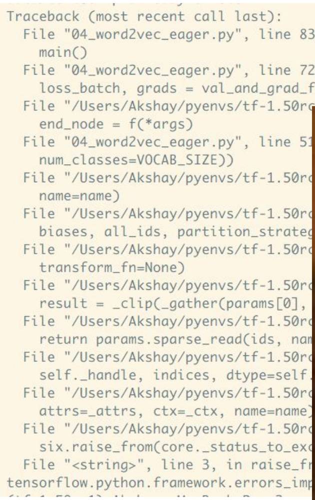
Key Advantages

- Compatible with Python debugging tools
    - `pdb.set_trace()` to your heart's content!
- Provides immediate error reporting
- Permits use of Python data structures
    - e.g., for structured input
- Enables easy, Pythonic control flow
    - `if` statements, `for` loops, recursion, oh my!

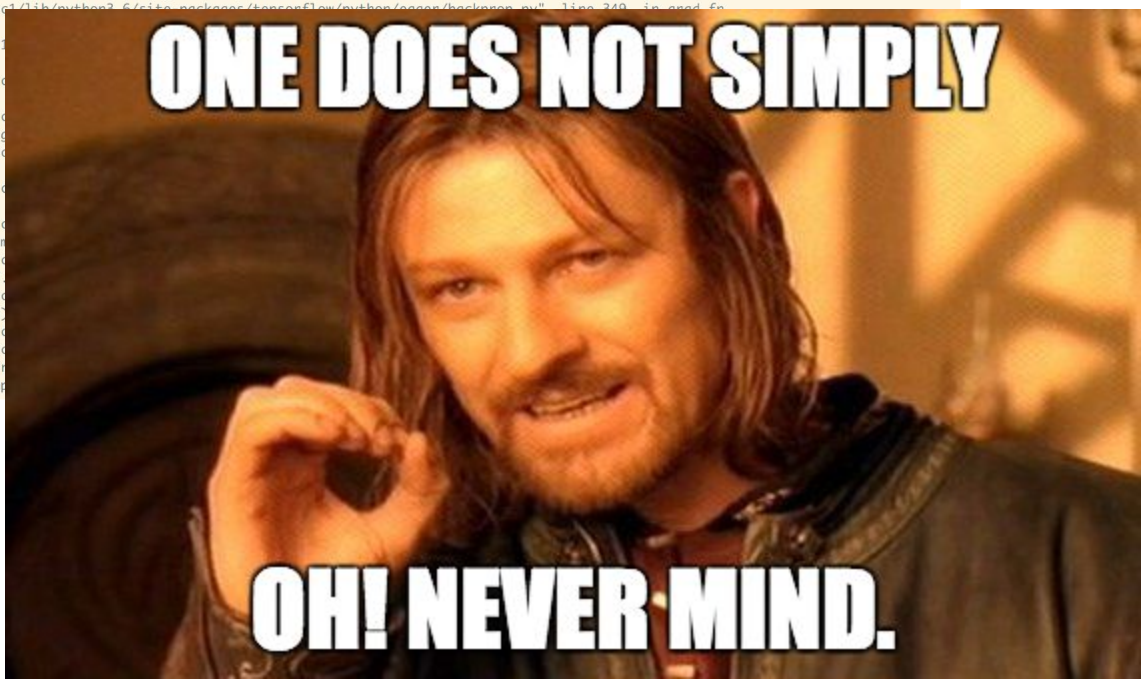
```
i = tf.constant(0)
while i < 1000:
  i = tf.add(i, 1)
  print("I could do this all day! %d" % i)
```



```
Traceback (most recent call last):
  File "04_word2vec_eager.py", line 83, in <module>
    main()
  File "04_word2vec_eager.py", line 72, in main
    loss_batch, grads = val_and_grad_fn(center_words, target_words)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/eager/backprop.py", line 349, in grad_fn
    end_node = f(*args)
  File "04_word2vec_eager.py", line 51, in word2vec
    num_classes=VOCAB_SIZE))
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/nn_impl.py", line 1212, in nce_loss
    name=name)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/nn_impl.py", line 1046, in _compute_sampled_logits
    biases, all_ids, partition_strategy=partition_strategy)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/embedding_ops.py", line 325, in embedding_lookup
    transform_fn=None)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/embedding_ops.py", line 150, in _embedding_lookup_and_transform
    result = _clip(_gather(params[0], ids, name=name), ids, max_norm)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/embedding_ops.py", line 52, in _gather
    return params.sparse_read(ids, name=name)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/resource_variable_ops.py", line 692, in sparse_read
    self._handle, indices, dtype=self._dtype, name=name)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/ops/gen_resource_variable_ops.py", line 250, in resource_gather
    attrs=_attrs, ctx=_ctx, name=name)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/eager/execute.py", line 66, in quick_execute
    six.raise_from(core._status_to_exception(e.code, message), None)
  File "<string>", line 3, in raise_from
tensorflow.python.framework.errors_impl.InvalidArgumentError: indices[0] = 3081 is not in [0, 128) [Op:ResourceGather] name: nce_loss/embedding_lookup/
```



```
Traceback (most recent call last):
  File "04_word2vec_eager.py", line 83, in <module>
    main()
  File "04_word2vec_eager.py", line 72, in main
    loss_batch, grads = val_and_grad_fn(center_words, target_words)
  File "/Users/Akshay/pyenvs/tf-1.50rc1/lib/python3.6/site-packages/tensorflow/python/eager/backprop.py", line 349, in grad_fn
    end_node = f(*args)
  File "04_word2vec_eager.py", line 51
    num_classes=VOCAB_SIZE))
  File "/Users/Akshay/pyenvs/tf-1.50rc
    name=name)
  File "/Users/Akshay/pyenvs/tf-1.50rc
    biases, all_ids, partition_strateg
  File "/Users/Akshay/pyenvs/tf-1.50rc
    transform_fn=None)
  File "/Users/Akshay/pyenvs/tf-1.50rc
    result = _clip(_gather(params[0],
  File "/Users/Akshay/pyenvs/tf-1.50rc
    return params.sparse_read(ids, nam
  File "/Users/Akshay/pyenvs/tf-1.50rc
    self._handle, indices, dtype=self.
  File "/Users/Akshay/pyenvs/tf-1.50rc
    attrs=_attrs, ctx=_ctx, name=name)
  File "/Users/Akshay/pyenvs/tf-1.50rc
    six.raise_from(core._status_to_exc
  File "<string>", line 3, in raise_fr
tensorflow.python.framework.errors_imp
```



Eager execution simplifies your code

You no longer need to worry about ...

1. placeholders
2. sessions
3. control dependencies
4. "lazy loading"
5. {name, variable, op} scopes

Boilerplate

```
x = tf.placeholder(tf.float32, shape=[1, 1])
m = tf.matmul(x, x)

print(m)
# Tensor("MatMul:0", shape=(1, 1), dtype=float32)

with tf.Session() as sess:
  m_out = sess.run(m, feed_dict={x: [[2.]]})
print(m_out)
# [[4.]]
```

Code like this...

~~Boilerplate~~

```
x = [[2.]]  # No need for placeholders!
m = tf.matmul(x, x)

print(m)  # No sessions!
# tf.Tensor([[4.]], shape=(1, 1), dtype=float32)
```

Becomes this

"Lazy Loading"

```
x = tf.random_uniform([2, 2])

with tf.Session() as sess:
  for i in range(x.shape[0]):
    for j in range(x.shape[1]):
      print(sess.run(x[i, j]))
```

Each iteration adds nodes to the graph

~~"Lazy Loading"~~

```
x = tf.random_uniform([2, 2])

for i in range(x.shape[0]):
  for j in range(x.shape[1]):
    print(x[i, j])
```

Tensors Act Like NumPy Arrays

```
x = tf.constant([1.0, 2.0, 3.0])

# Tensors are backed by NumPy arrays
assert type(x.numpy()) == np.ndarray
squared = np.square(x) # Tensors are compatible with NumPy functions

# Tensors are iterable!
for i in x:
  print(i)


for i in range(x.shape[0]):
  for j in range(x.shape[1]):
    print(x[i, j])
```

Caveat: use tf.equal to compare Tensors, not ==

Gradients

Gradients

Automatic differentiation is built into eager execution

Under the hood ...

- Operations are recorded on a **tape**
- The tape is **played back** to compute gradients
    - This is reverse-mode differentiation (backpropagation).

Gradients

```
def square(x):
  return x ** 2

grad = tfe.gradients_fu

print(square(3.))     # tf.Tensor(9., shape=(), dtype=float32)
print(grad(3.))       # [tf.Tensor(6., shape=(), dtype=float32)]
```

Differentiate w.r.t. input of square

Gradients

Use **tfe**.`Variable` when eager execution is enabled.

```
x = tfe.Variable(2.0)
def loss(y):
  return (y - x ** 2) ** 2

grad = tfe.implicit_gradient
```

Differentiate w.r.t. variables used to compute `loss`

```
print(loss(7.))  # tf.Tensor(9., shape=(), dtype=float32)
print(grad(7.))  # [(<tf.Tensor: -24.0, shape=(), dtype=float32>,
                     <tf.Variable 'Variable:0' shape=()
                      dtype=float32, numpy=2.0>)]
```

Gradients

APIs for computing gradients work even when eager execution is not enabled

- `tfe.gradients_function()`
- `tfe.value_and_gradients_function()`
- `tfe.implicit_gradients()`
- `tfe.implicit_value_and_gradients()`

See the user guide for documentation

It's not *that* different

A Collection of Operations

TensorFlow = Operation Kernels + Execution

- Graph construction: Execute compositions of operations with Sessions
- Eager execution: Execute compositions with Python

A Collection of Operations

Majority of TF API works regardless of whether eager execution is enabled.

- But, when eager execution is enabled …
    - prefer **`tfe`**`.Variable` under eager execution (compatible with graph construction)
    - manage your own variable storage — variable collections are not supported!
    - use `tf.contrib.summary`
    - use **`tfe`**`.Iterator` to iterate over datasets under eager execution
    - prefer object-oriented layers (e.g., `tf.layers.Dense`)
        - functional layers (e.g., `tf.layers.dense`) only work if wrapped in **`tfe`**`.make_template`
    - prefer **`tfe`**`.py_func` over `tf.py_func`

- See the user guide for details and updates

What if I like graphs?

Graphs are ...

- Optimizable
  - automatic buffer reuse
  - constant folding
  - inter-op parallelism
  - automatic trade-off between compute and memory
- Deployable
  - the Graph is an *intermediate representation* for models
- Rewritable
  - experiment with automatic device placement or quantization

Imperative to declarative and back

- **Write model definition code once**
    - The same code can execute operations in one Python process and construct graphs in another (see user guide/examples)
- **Checkpoints are compatible**
    - Train eagerly, checkpoint, load in a graph, or vice-versa
- **Create graphs while eager execution is enabled**:
    - `tfe.defun:` "Compile" computation into graphs and execute them.

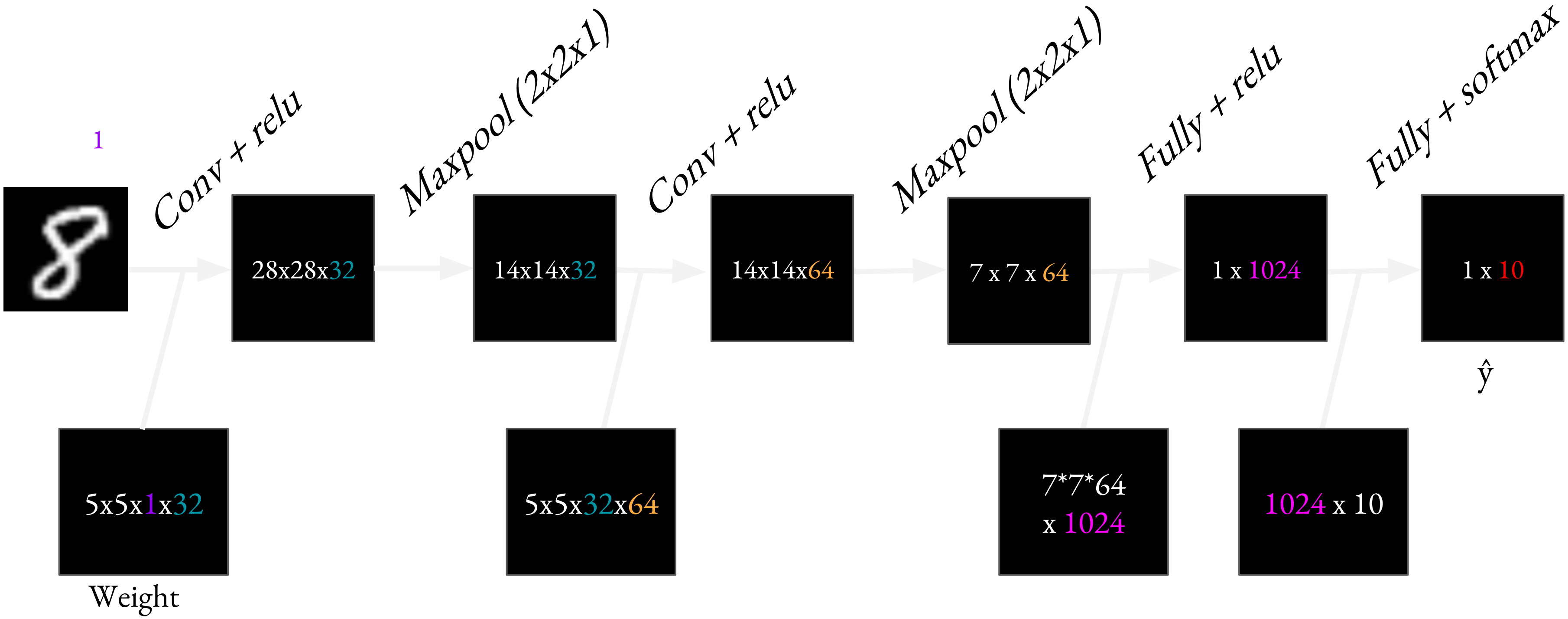
So when should I use eager execution?

Use eager if you're ...

- **a researcher and want a flexible framework**
  - python control flow and data structures enable experimentation
- **developing a new model**
  - immediate error reporting simplifies debugging
- **new to TensorFlow**
  - eager execution lets you explore the TF API in the Python REPL

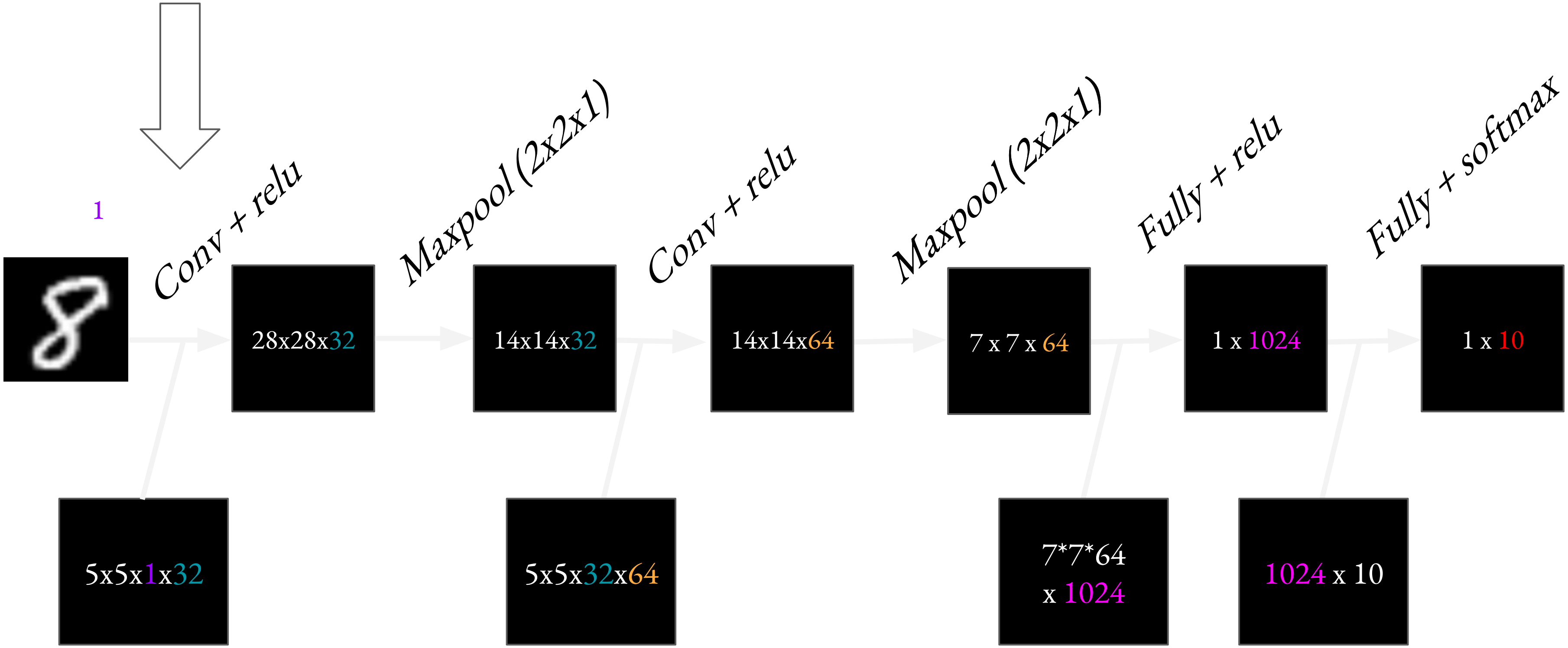
Now Back to Graphs

Model



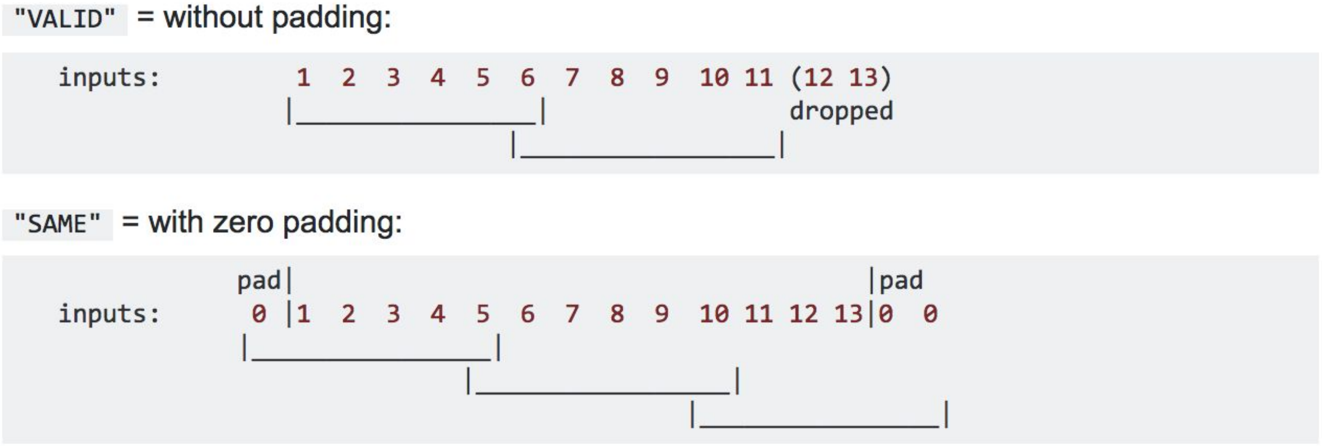
Strides for all convolutional layers: [1, 1, 1, 1]

Convolutional layer



```
conv = tf.nn.conv2d(images,
                    kernel,
                    strides=[1, 1, 1, 1],
                    padding='SAME')
```

Convolutional layer: padding



"VALID" = without padding:

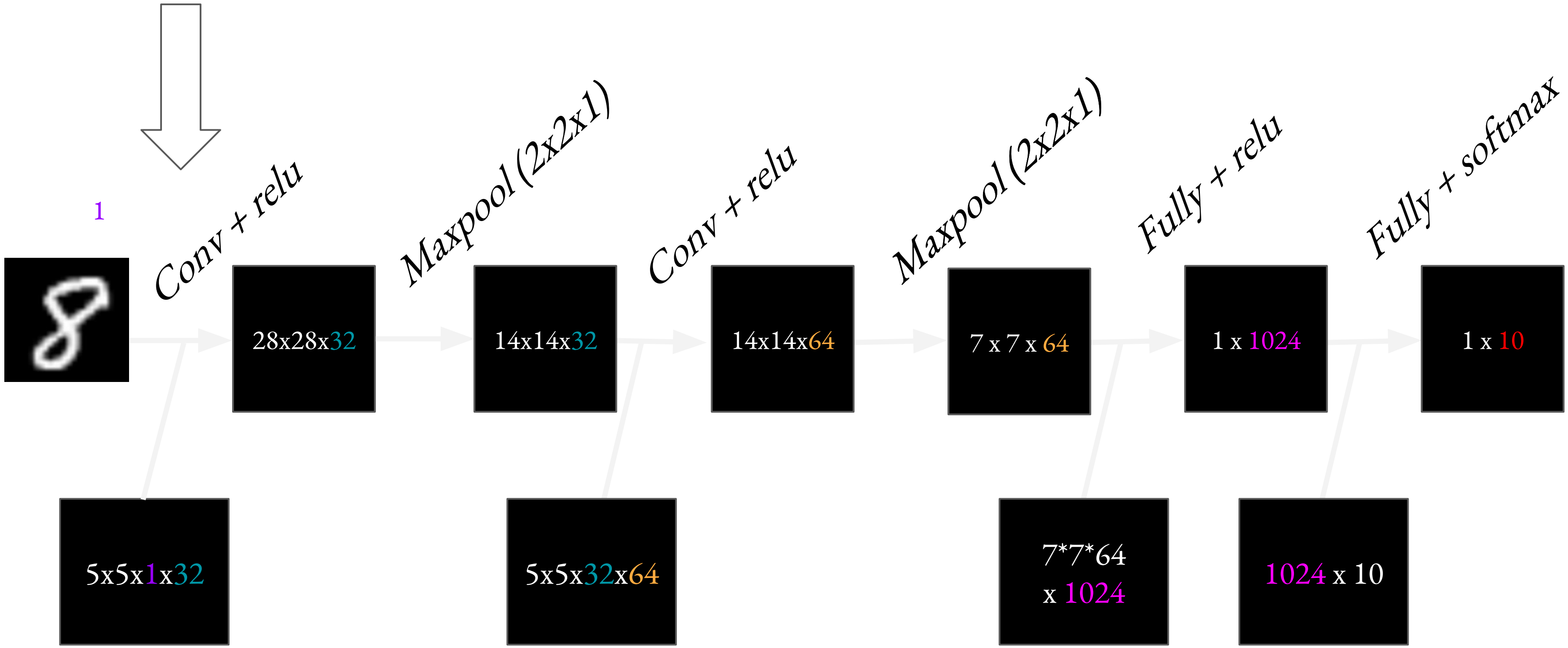
```
inputs:         1  2  3  4  5  6  7  8  9  10 11 (12 13)
               |________________|                dropped
                              |_________________|
```

"SAME" = with zero padding:

```
               pad|                                      |pad
inputs:         0 |1  2  3  4  5  6  7  8  9  10 11 12 13|0  0
               |________________|
                              |_________________|
                                             |________________|
```

Input width = 13
Filter width = 6
Stride = 5

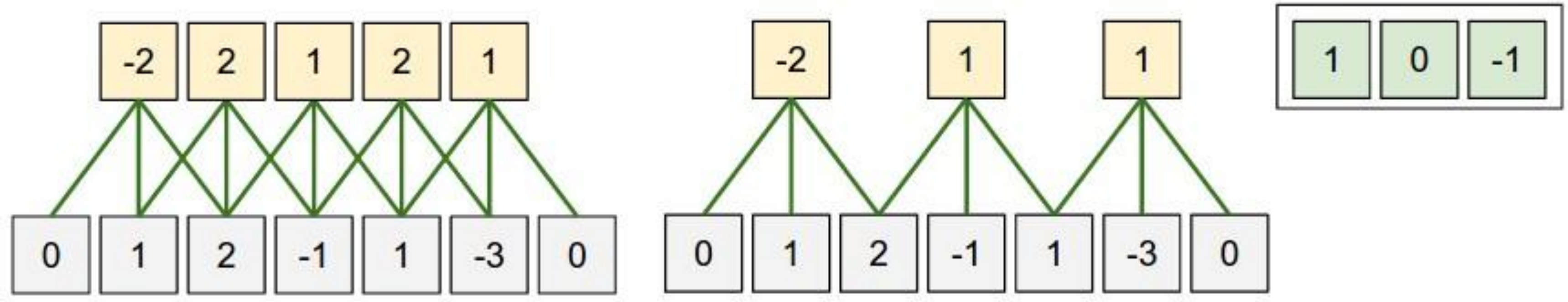
Convolutional layer: Dimension



(W−F+2P)/S+ 1

W: input width/depth F: filter width/depth
P: padding S: stride

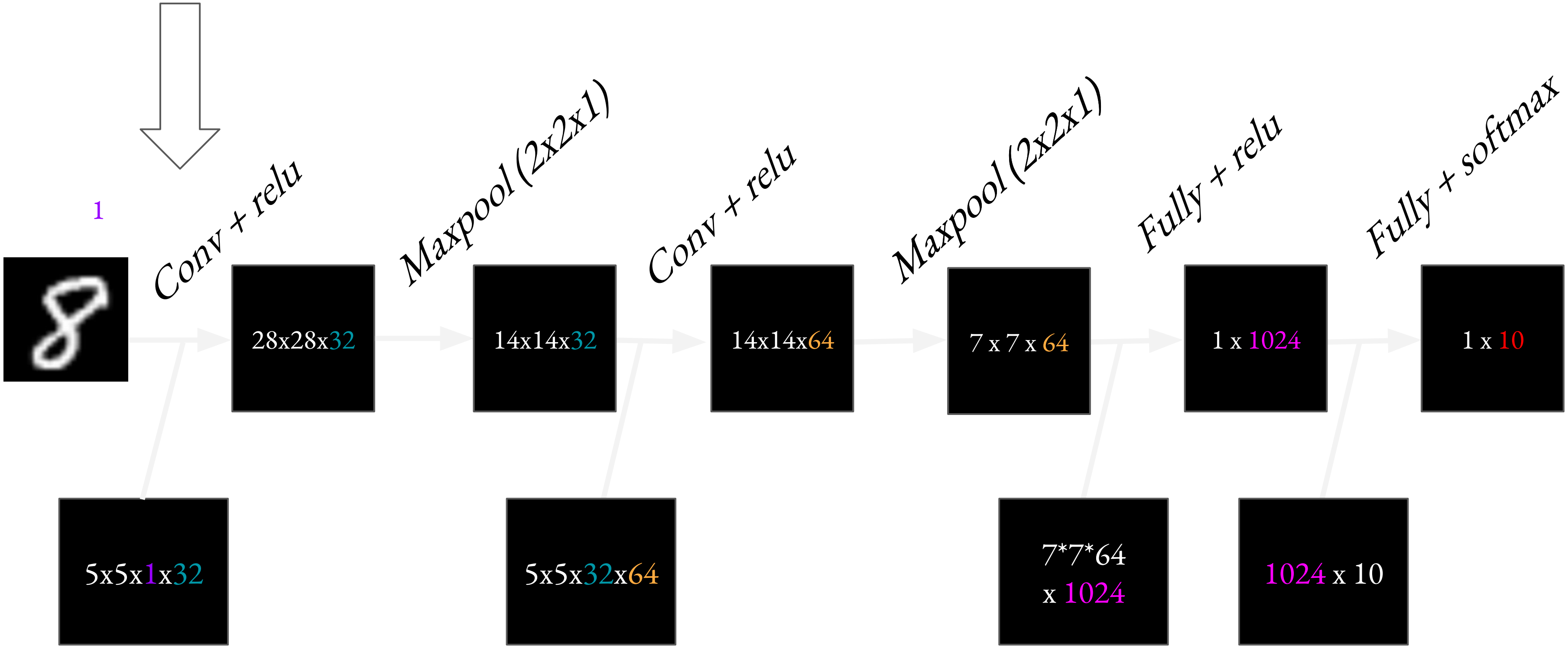
Convolutional layer: Dimension



$$\mathbf{(W-F+2P)/S+1}$$

W: input width/depth F: filter width/depth
P: padding S: stride

Convolutional layer: Dimension



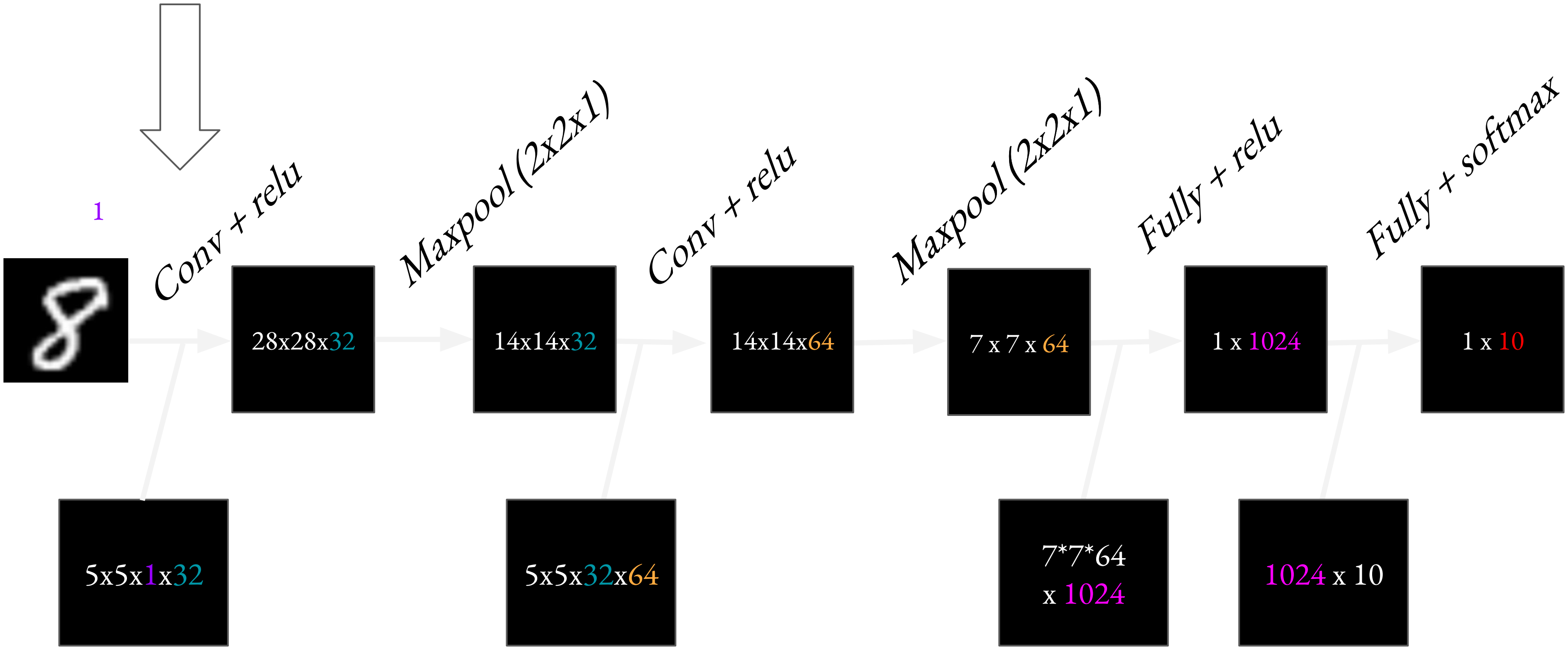
(W−F+2P)/S+ 1

(28 - 5 + 2*2)/1 + 1 = 28

W: input width/depth F: filter width/depth

P: padding S: stride

Convolutional layer: Dimension



(W−F+2P)/S+ 1

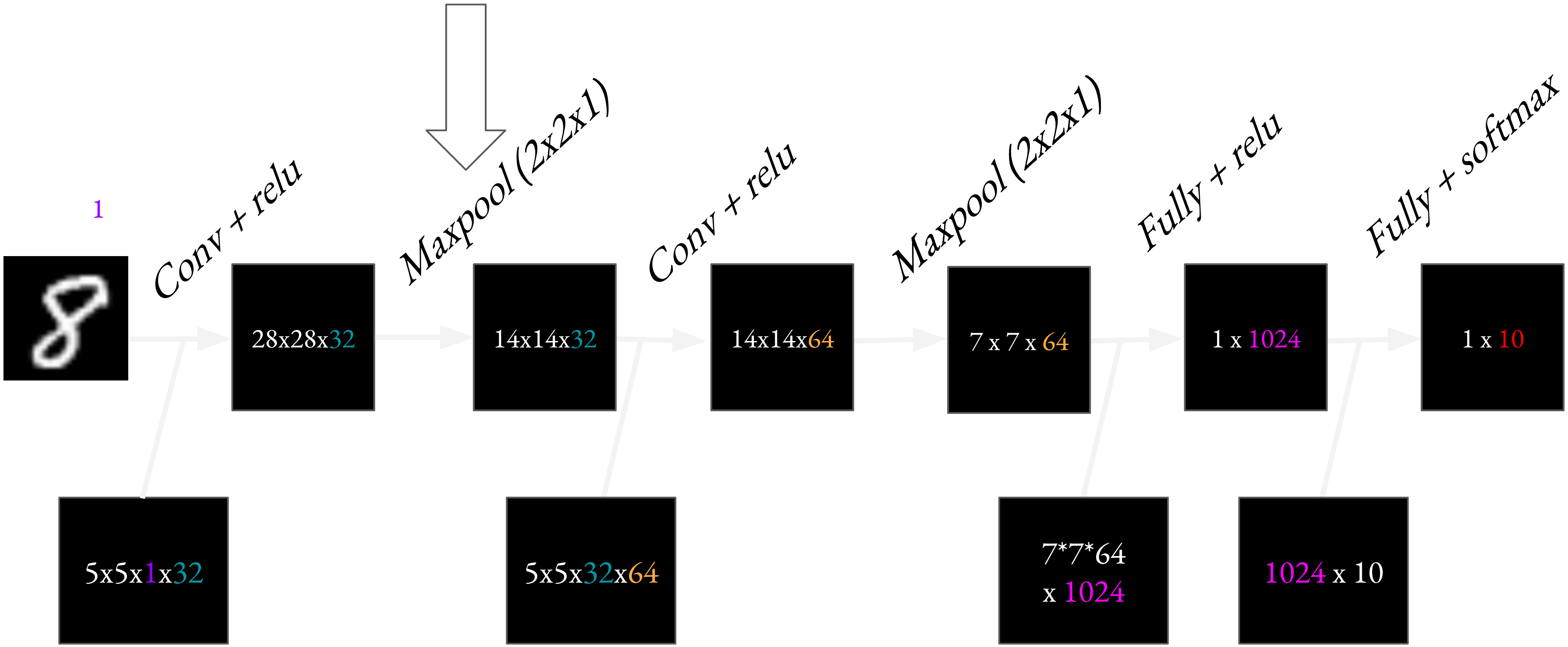
(28 - 5 + 2*2)/1 + 1 = 28

W: input width/depth F: filter width/depth

P: padding S: stride

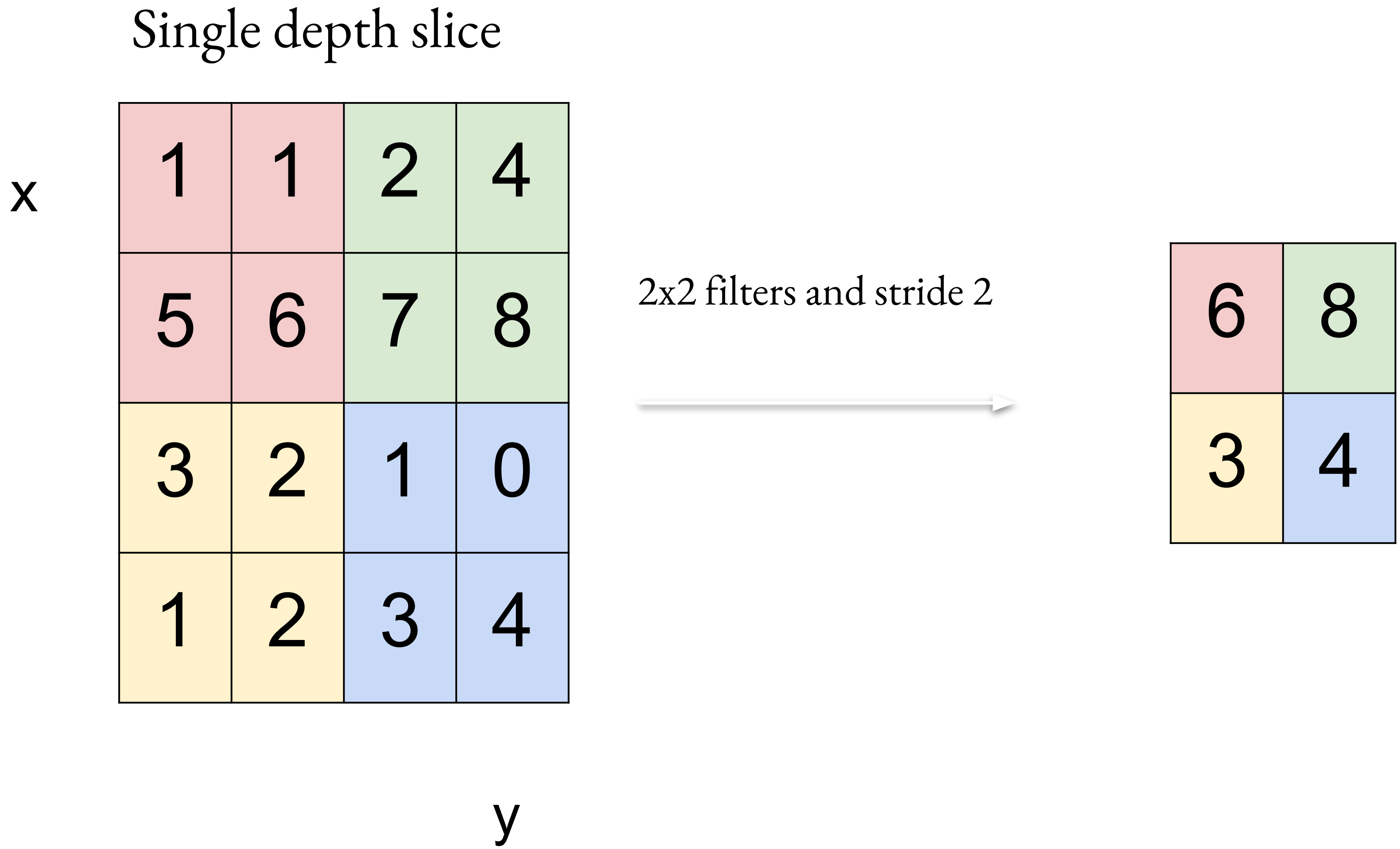
TF computes padding for us!

Maxpooling



```
pool1 = tf.nn.max_pool(conv1,
                       ksize=[1, 2, 2, 1],
                       strides=[1, 2, 2, 1],
                       padding='SAME')
```

Maxpooling



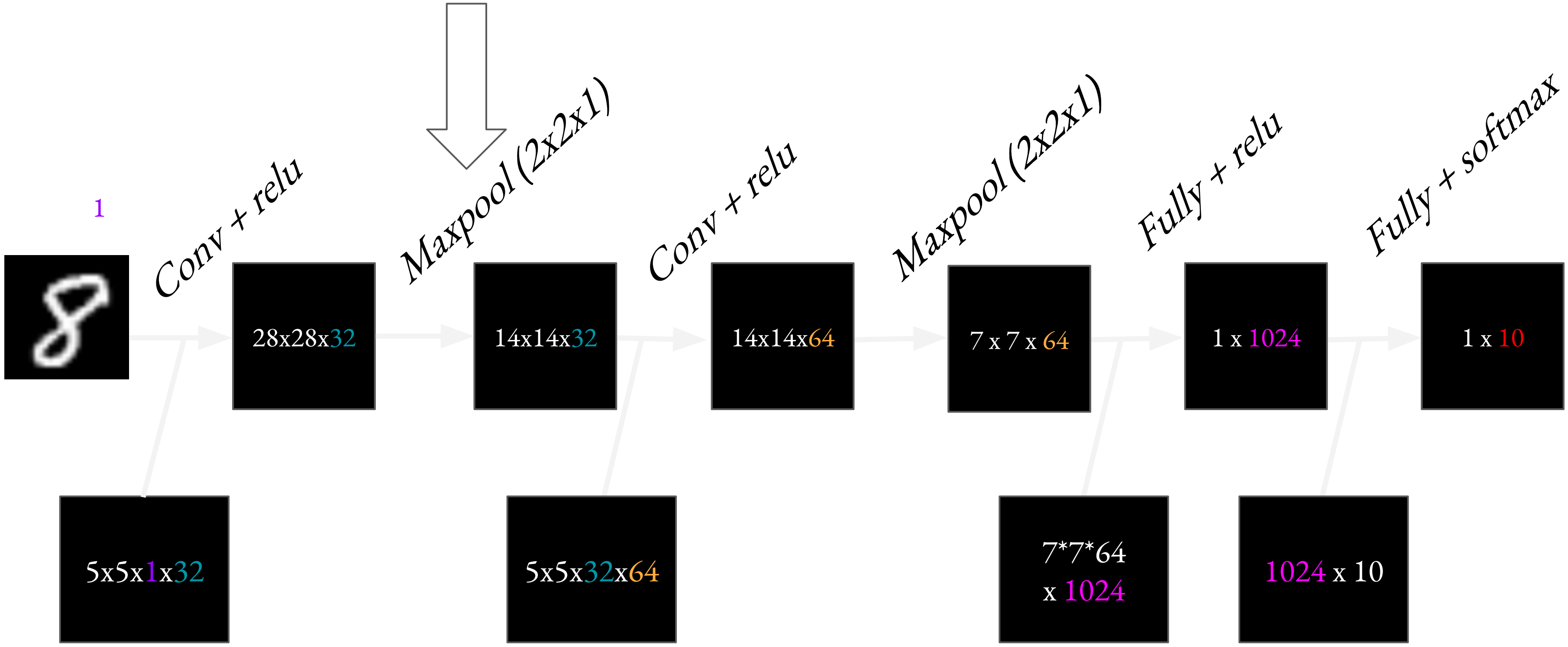
Single depth slice

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

2x2 filters and stride 2

6	8
3	4

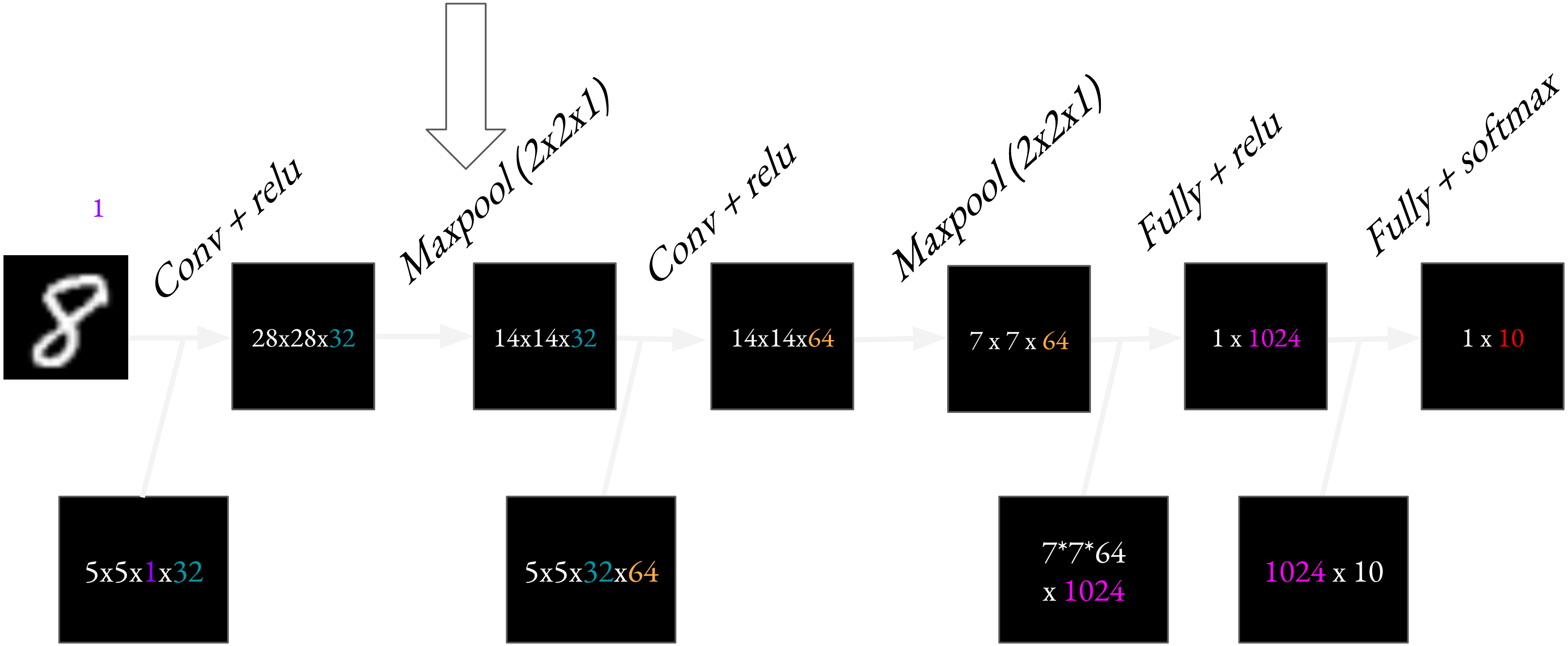
Maxpooling: Dimension



(W−K+2P)/S+ 1

W: input width/depth   K: window width/depth
P: padding   S: stride

Maxpooling: Dimension



1

Conv + relu

Maxpool (2x2x1)

Conv + relu

Maxpool (2x2x1)

Fully + relu

Fully + softmax

28x28x32

14x14x32

14x14x64

7 x 7 x 64

1 x 1024

1 x 10

5x5x1x32

5x5x32x64

7*7*64
x 1024

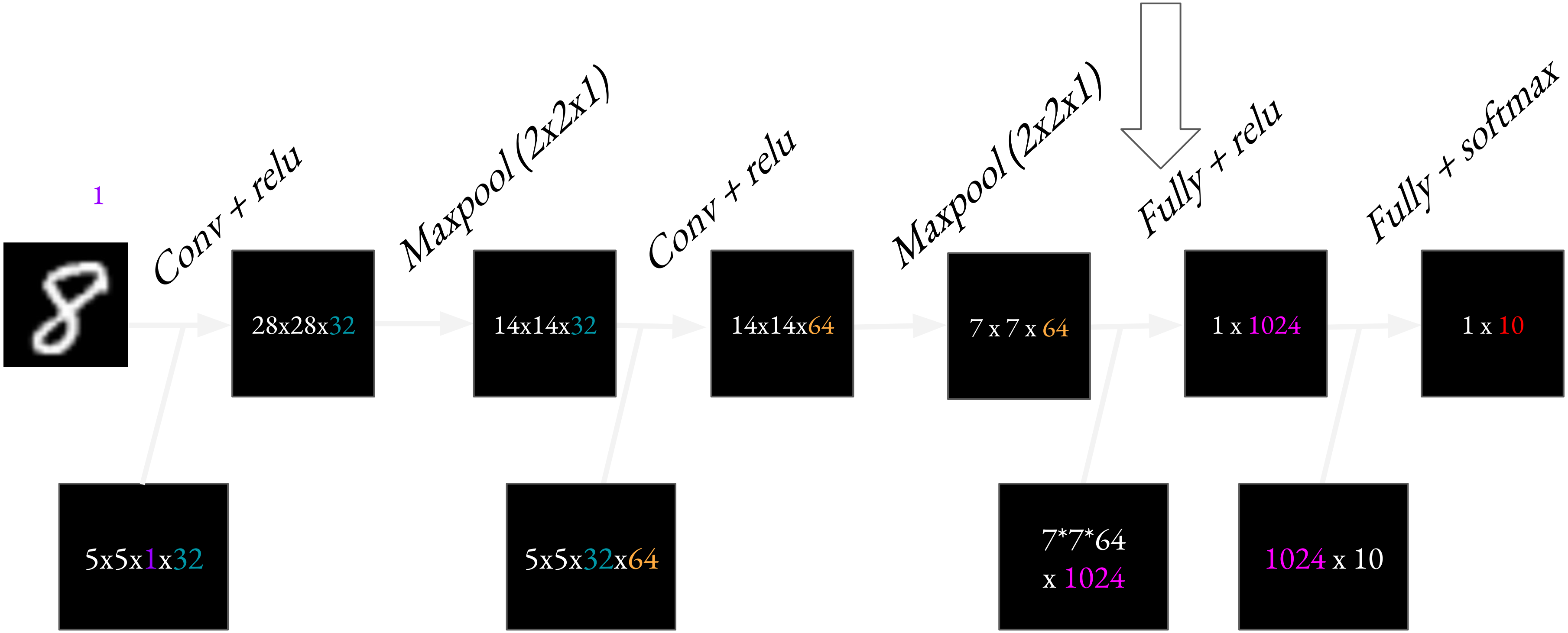
1024 x 10

(W−K+2P)/S+ 1

(28 - 2 + 2*0) / 2 + 1 = 14

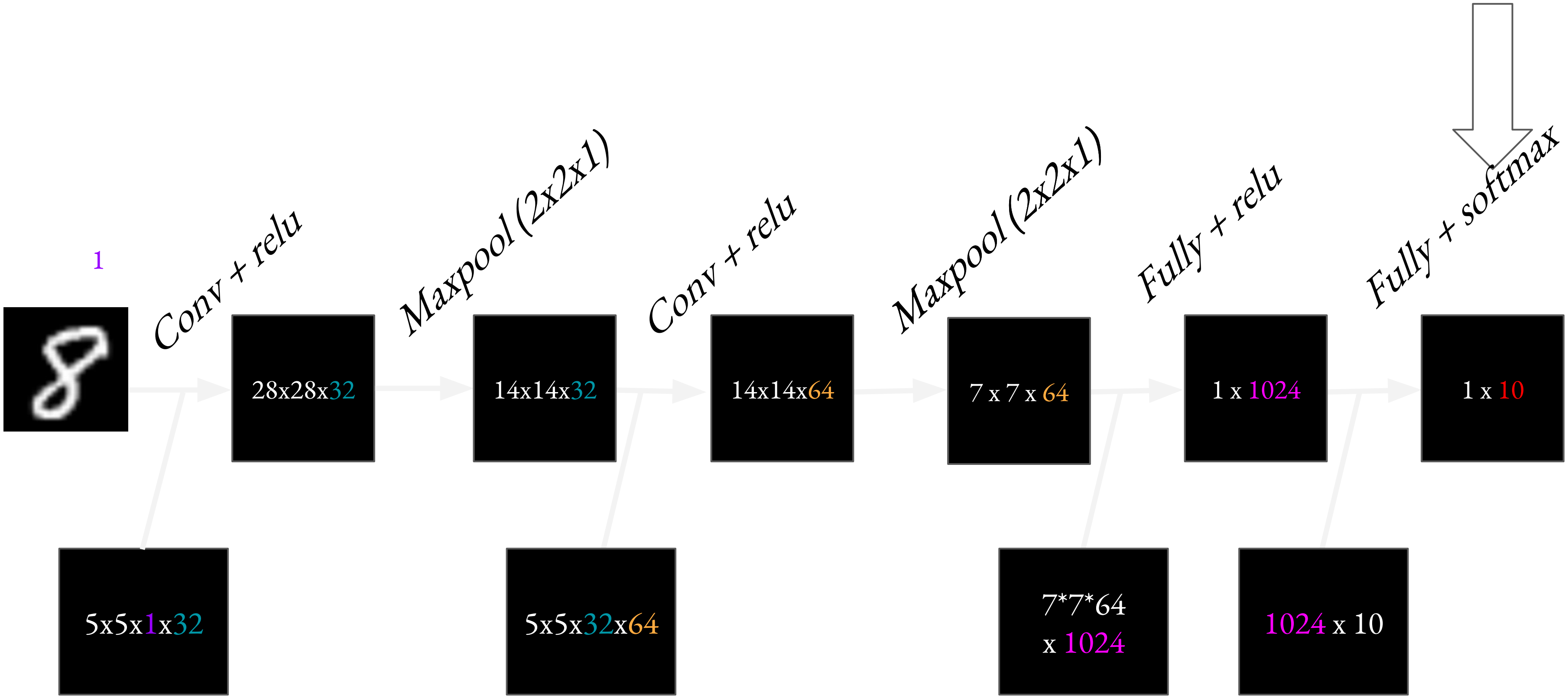
W: input width/depth	K: window width/depth
P: padding	S: stride

Fully connected



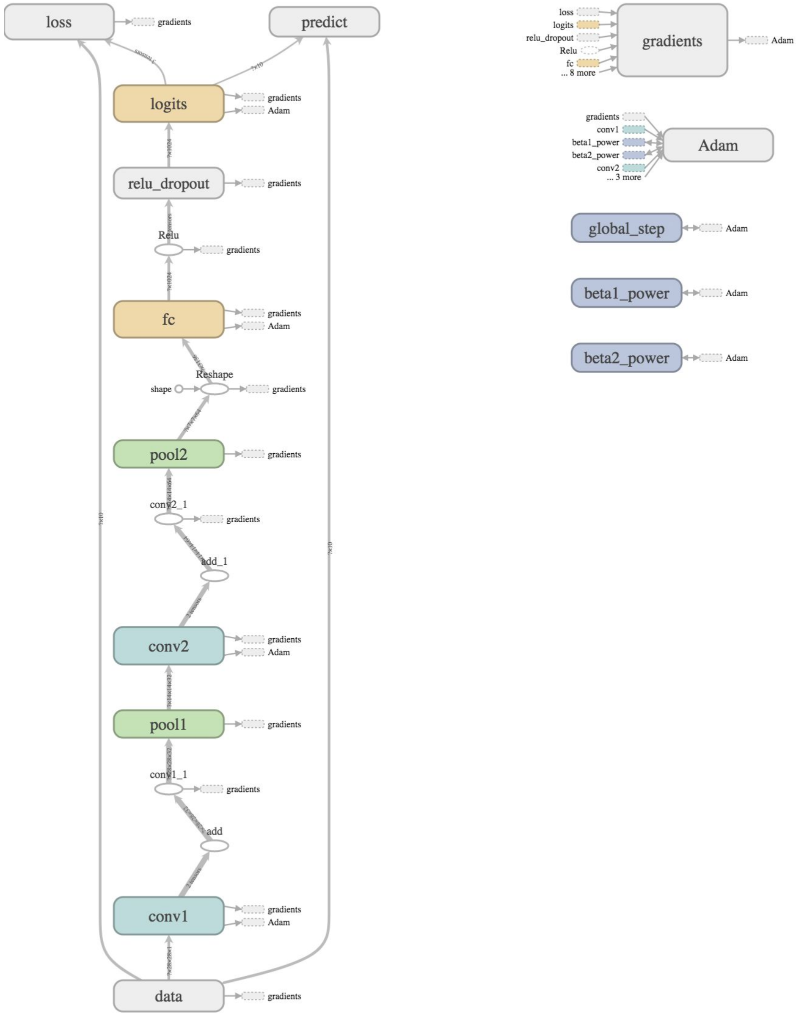
fc = tf.matmul(pool2, w) + b

Softmax

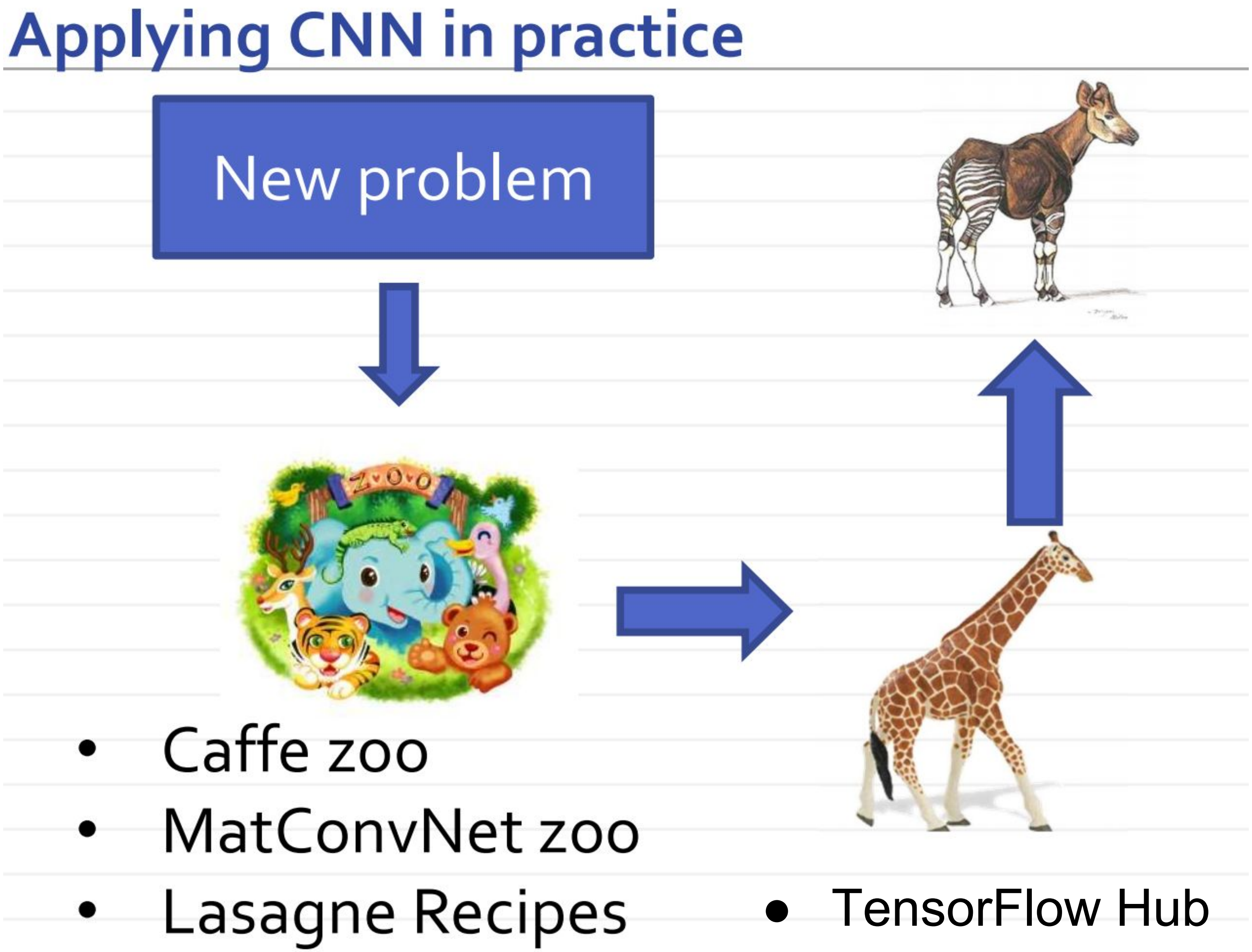


softmax_cross_entropy_with_logits

softmax


loss
gradients
predict
logits
gradients
Adam
relu_dropout
gradients
Relu
gradients
fc
gradients
Adam
Reshape
shape
gradients
pool2
gradients
conv2_1
gradients
add_1
conv2
gradients
Adam
pool1
gradients
conv1_1
gradients
add
conv1
gradients
Adam
data
gradients
?x10
?x10
loss
logits
relu_dropout
Relu
fc
... 8 more
gradients
Adam
gradients
conv1
beta1_power
beta2_power
conv2
... 3 more
Adam
global_step
Adam
beta1_power
Adam
beta2_power
Adam

DNN In Practice: Model Zoo + FT



Applying CNN in practice

New problem

- Caffe zoo
- MatConvNet zoo
- Lasagne Recipes

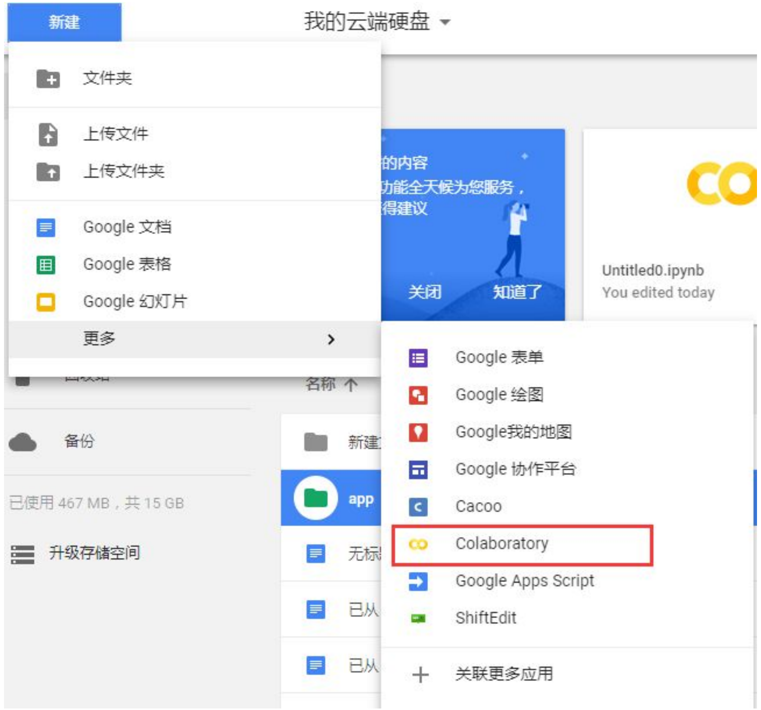
- TensorFlow Hub

DNN In Practice

1. GPU, GPU, GPU, GPU, GPU, GPU, GPU, ….
2. Fine Tuning with pretrained models
3. Weight Initialization, Learning rate -- really sensitive
4. Data Preprocessing and Augmentation
    a. Subtract mean, Divide on variance, ZCA whitening
    b. Rotation, Shifting, Noise, …
5. Debugging
    a. Nan -> div(x, 0), log(0), sqrt(0), Initialization, LR, Bad Optimizer, Grad Clipping, Max Norm, L2
    b. NoFit -> Initialization, LR, Data Preprocessing, Bad Optimizer, Gradient Vanishing (add another loss, remove sigmoid), Pretraining, Batch Norm
    c. OverFit -> dropout, Batch Norm, l2, ….
    d. Memory error -> less conv :(less batch, ...

Dark Magic

DNN In Practice: Google Colab



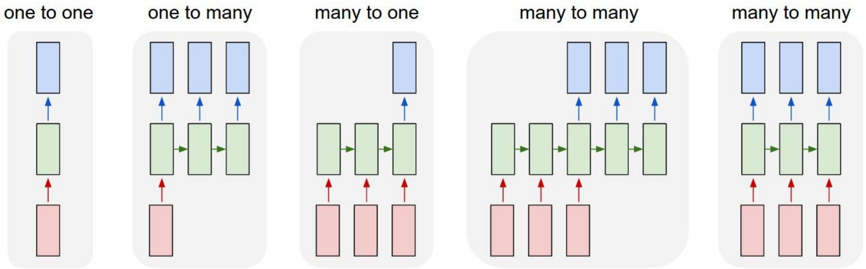
- You have to register for 2nd assignment
- ~ 100x Faster Training than CPU

Administrative

- A1 is due April 20, ~6 days left
- A2, **Deep ConvNets on CIFAR-10** will come out at the end of week at worst **(you need GPU!)**
- **!!!** Lecture Feedback https://goo.gl/forms/zeZiu1fSgrpPGp6T2 **!!!**

Next Time

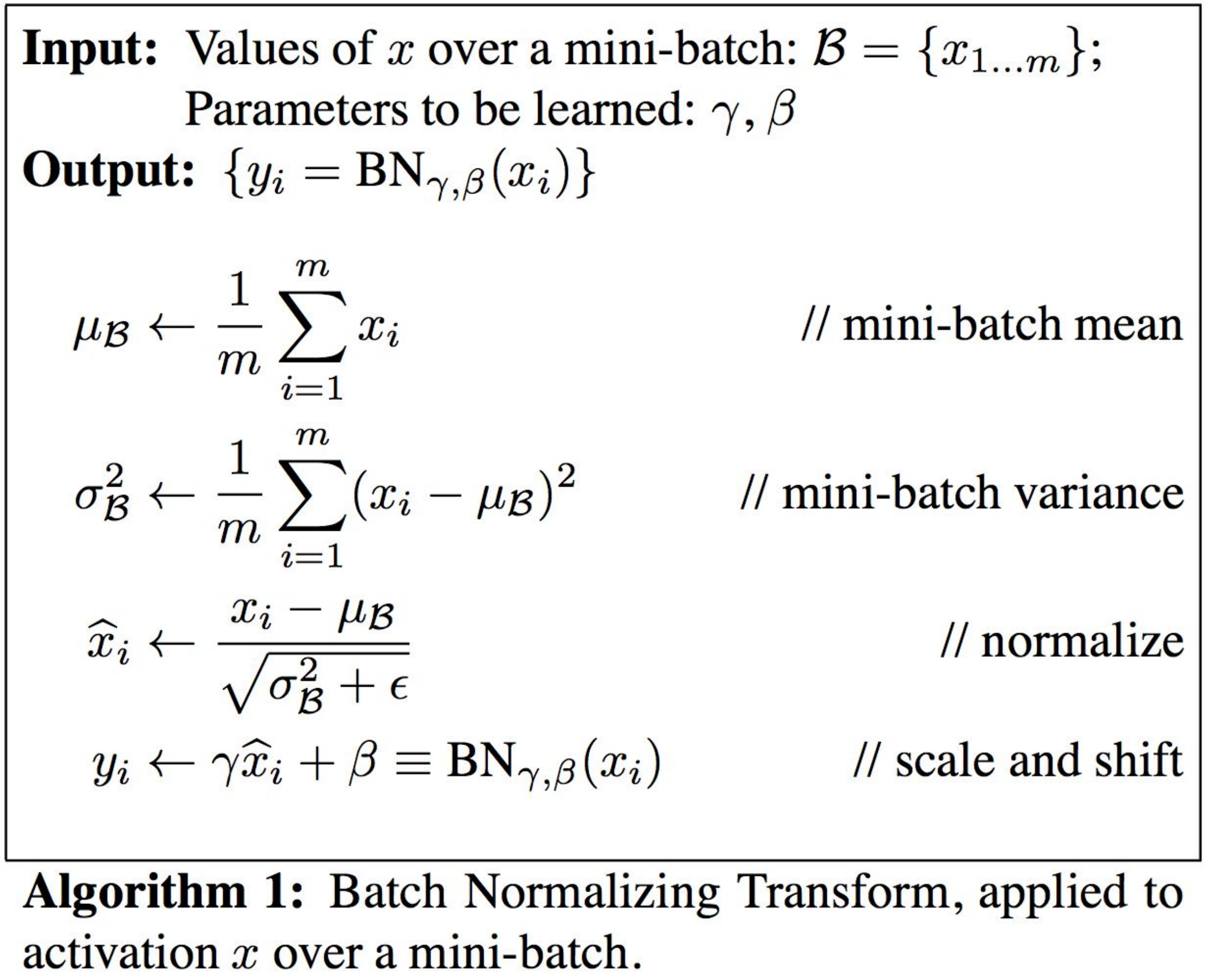
- Recurrent Neural Nets
- Text, Speech, Image Captioning, etc.



Good courses/books

- cs231n.stanford.edu, https://goo.gl/75Zi5m
- udacity.com/course/deep-learning–ud730
- deeplearningbook.org

ResNet (2015): Batch Norm



```
Input: Values of $x$ over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
       Parameters to be learned: $\gamma, \beta$
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

    $\mu_\mathcal{B} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$                              // mini-batch mean
    $\sigma_\mathcal{B}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\mathcal{B})^2$          // mini-batch variance
    $\widehat{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}}$                      // normalize
    $y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$               // scale and shift
```

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

https://arxiv.org/pdf/1512.03385.pdf