Structure your model

TensorFlow for Deep Learning Research

Lecture 4

Agenda

Overall structure of a model in TensorFlow

word2vec

Name scope

Embedding visualization

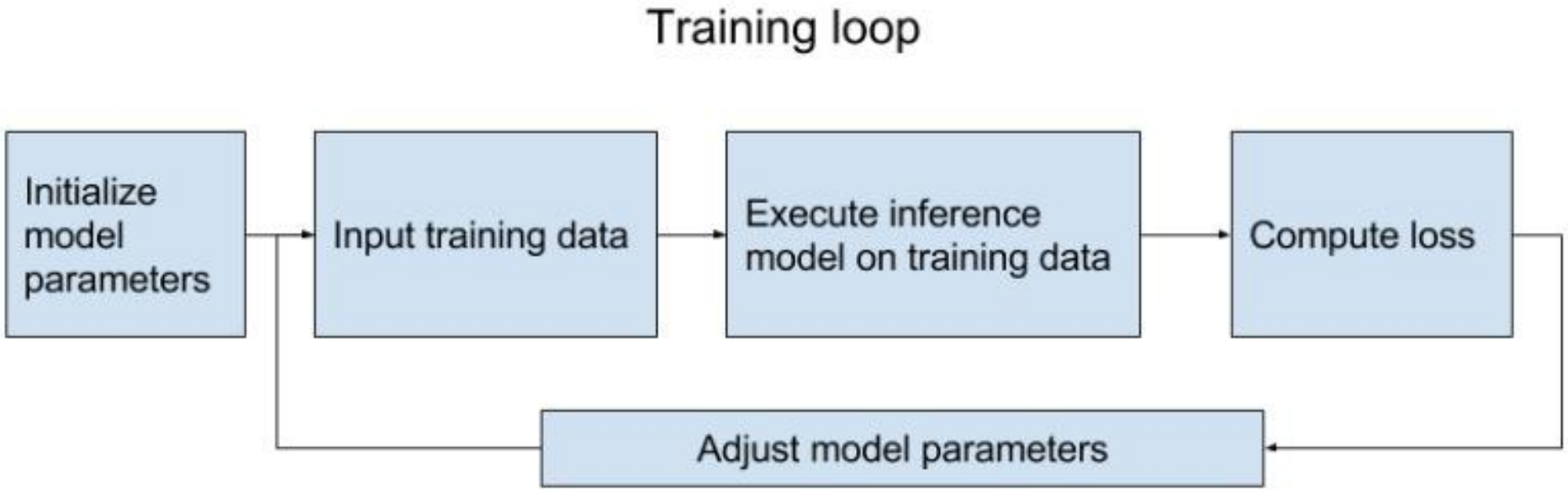
Interactive Coding!

Overall structure of a model in TensorFlow

Phase 1: Assemble graph

1. Define placeholders for input and output
2. Define the weights
3. Define the inference model
4. Define loss function
5. Define optimizer

Phase 2: Compute

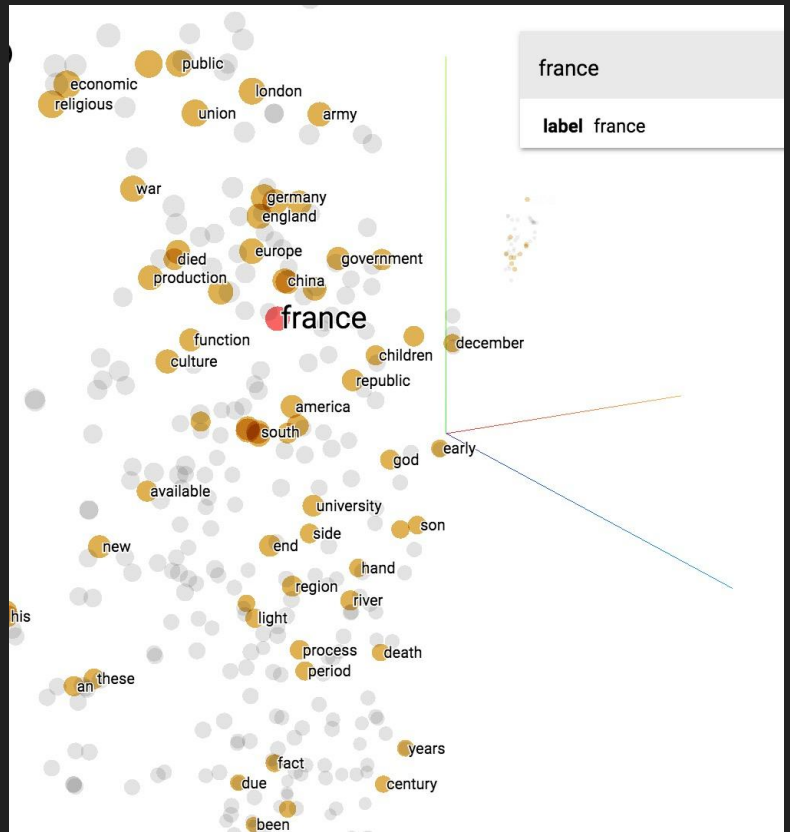


Graph from the book “TensorFlow for Machine Intelligence”

Word Embedding

Capture the semantic relationships between words

Word Embedding



Live visualization

Count vs Predict

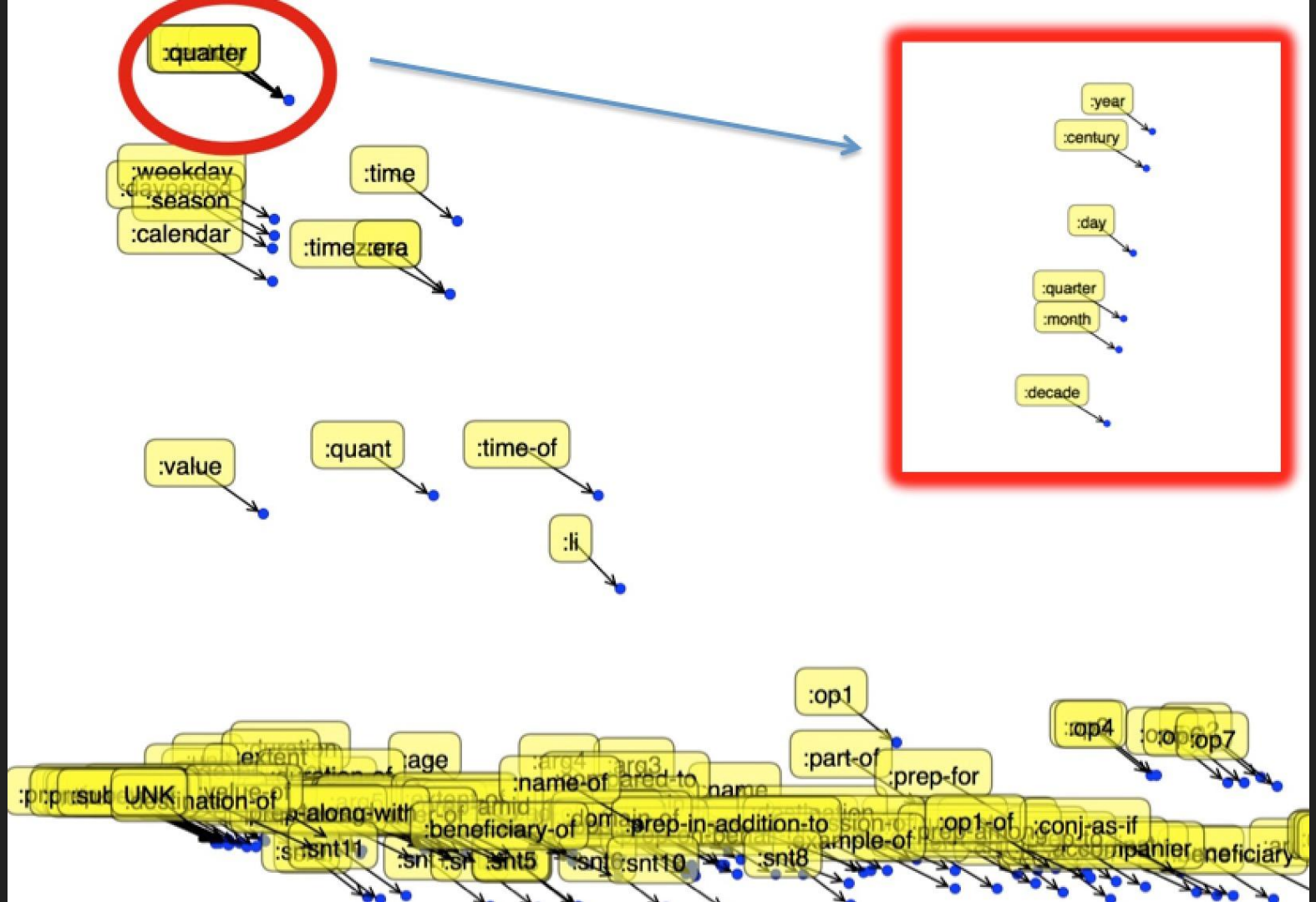
Counting

- Example corpus:
  - I like deep learning.
  - I like NLP.
  - I enjoy flying.

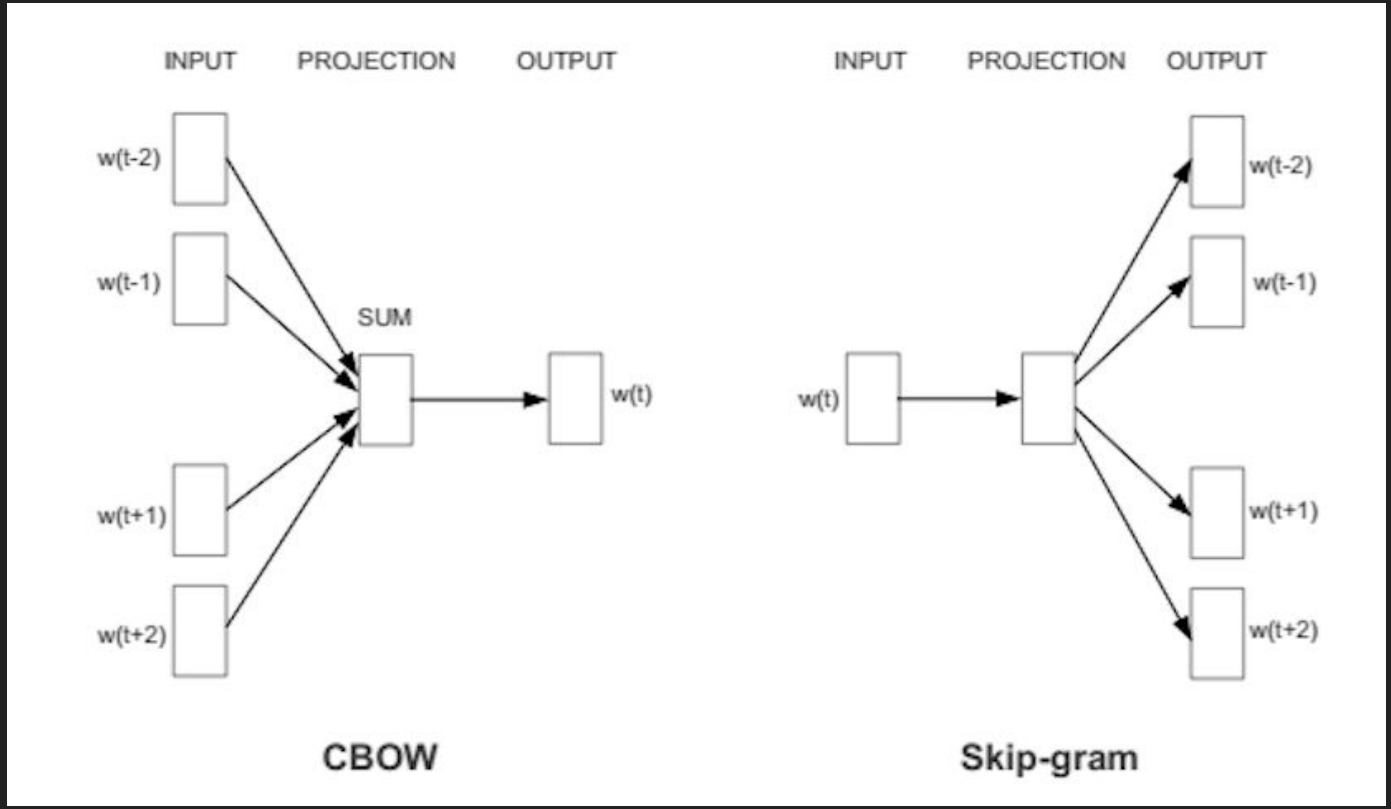
counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

15

Graph by Richard Socher for CS224N


:quarter
:weekday
:season
:calendar
:time
:era
:year
:century
:day
:quarter
:month
:decade
:value
:quant
:time-of
:li
:op1
:part-of
:prep-for
:op4
:op7
:name-of
:beneficiary-of
:prep-in-addition-to
:op1-of
:conj-as-if
:snt11
:snt5
:snt10
:snt8

Predicting



Graph by deeplearning4j.org

Implementing word2vec skip-gram

Softmax vs Sample-based Approaches

Softmax

$$p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)}$$

Computationally expensive

Handwritten formula from a slide for CS224N

Sample-based Approaches

Negative Sampling

is a simplified version of

Noise Contrastive Estimation

Sample-based Approaches

NCE guarantees approximation to softmax

Negative Sampling doesn't

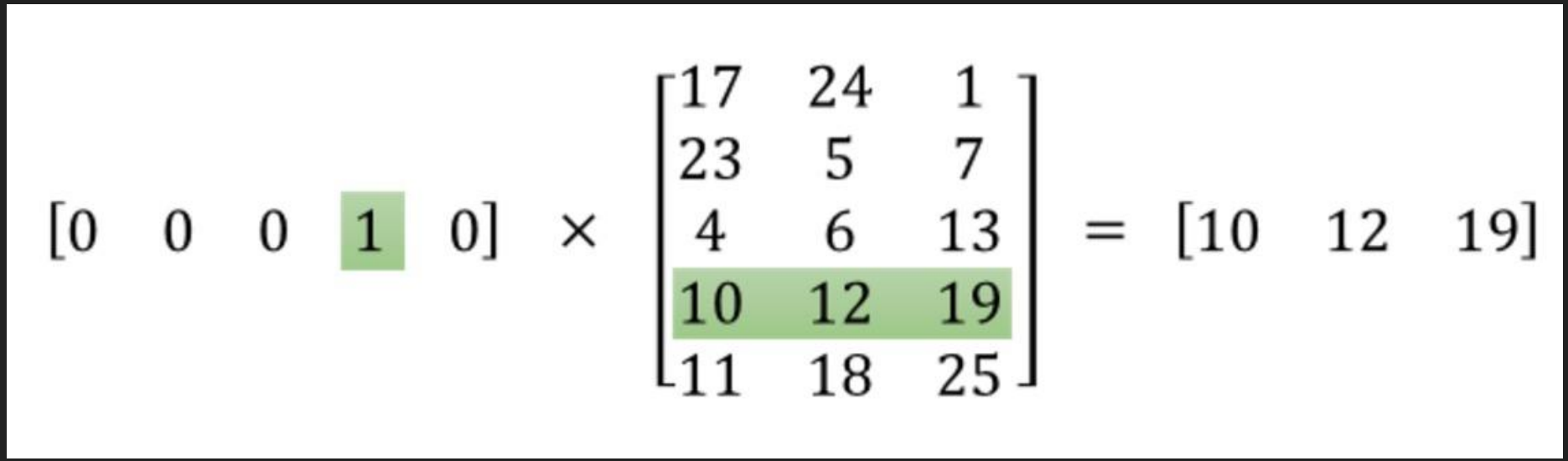
See lecture note for mathy stuff

Embedding Lookup

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = \begin{bmatrix} 10 & 12 & 19 \end{bmatrix}$$

Illustration by Chris McCormick

Embedding Lookup




```
tf.nn.embedding_lookup(params, ids, partition_strategy='mod', name=None,
                       validate_indices=True, max_norm=None)
```


Illustration by Chris McCormick

NCE Loss

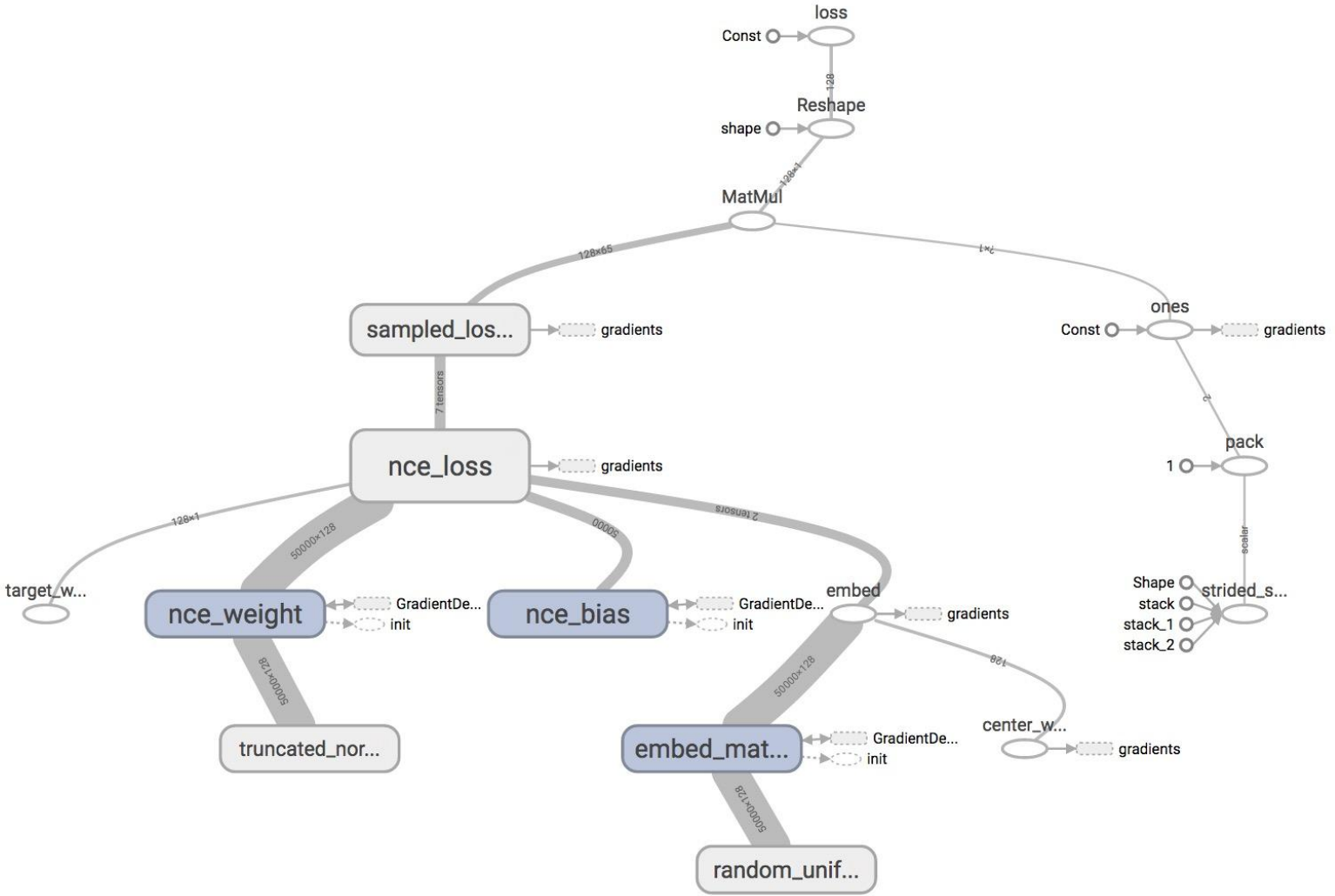
```
tf.nn.nce_loss(weights, biases, labels, inputs, num_sampled,
               num_classes, ...)
```

Illustration by Chris McCormick

Let’s write some code

Get process_data.py and 04_word2vec_starter.py
from GitHub
(in examples)

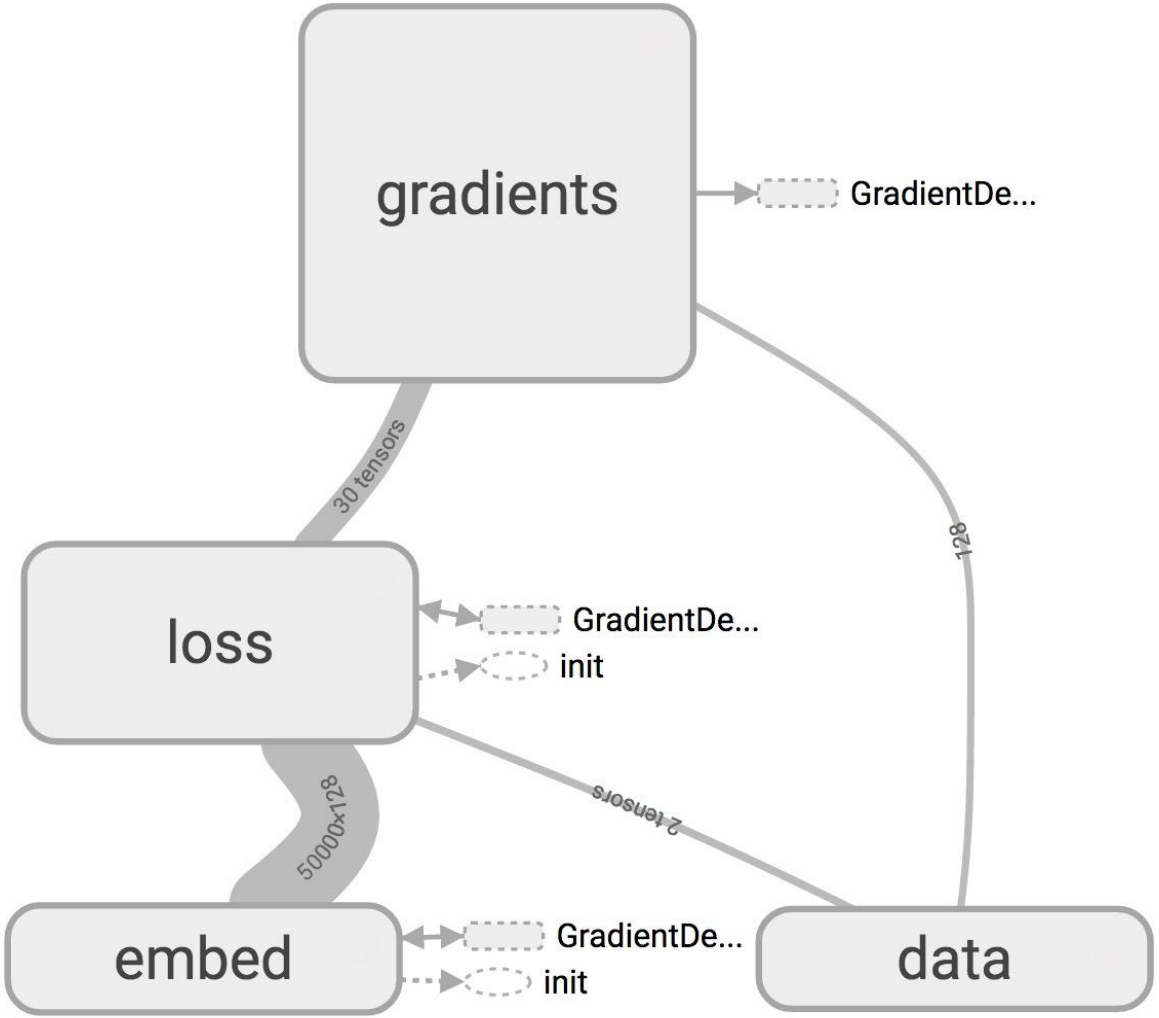
Illustration by Chris McCormick


loss
Const
128
Reshape
shape
128×1
MatMul
128×65
2×1
sampled_los...
gradients
ones
Const
gradients
7 tensors
2
pack
1
nce_loss
gradients
scalar
128×1
50000×128
50000
2 tensors
target_w...
nce_weight
GradientDe...
init
nce_bias
GradientDe...
init
embed
gradients
Shape
stack
stack_1
stack_2
strided_s...
50000×128
128
50000×128
truncated_nor...
embed_mat...
GradientDe...
init
center_w...
gradients
50000×128
random_unif...

Name scope

Group nodes together

```
with tf.name_scope(name)
```


gradients
GradientDe...
30 tensors
128
loss
GradientDe...
init
50000×128
2 tensors
embed
GradientDe...
init
data

Next class

Manage experiments

Example: word2vec

Feedback:

Thanks!