



# Structure your model

CS 20SI:  
TensorFlow for Deep Learning Research  
Lecture 4  
1/25/2017



**Assignment 1 is out! (due 1/31)**

**Questions?**

# Agenda

Overall structure of a model in TensorFlow

word2vec

Name scope

Embedding visualization



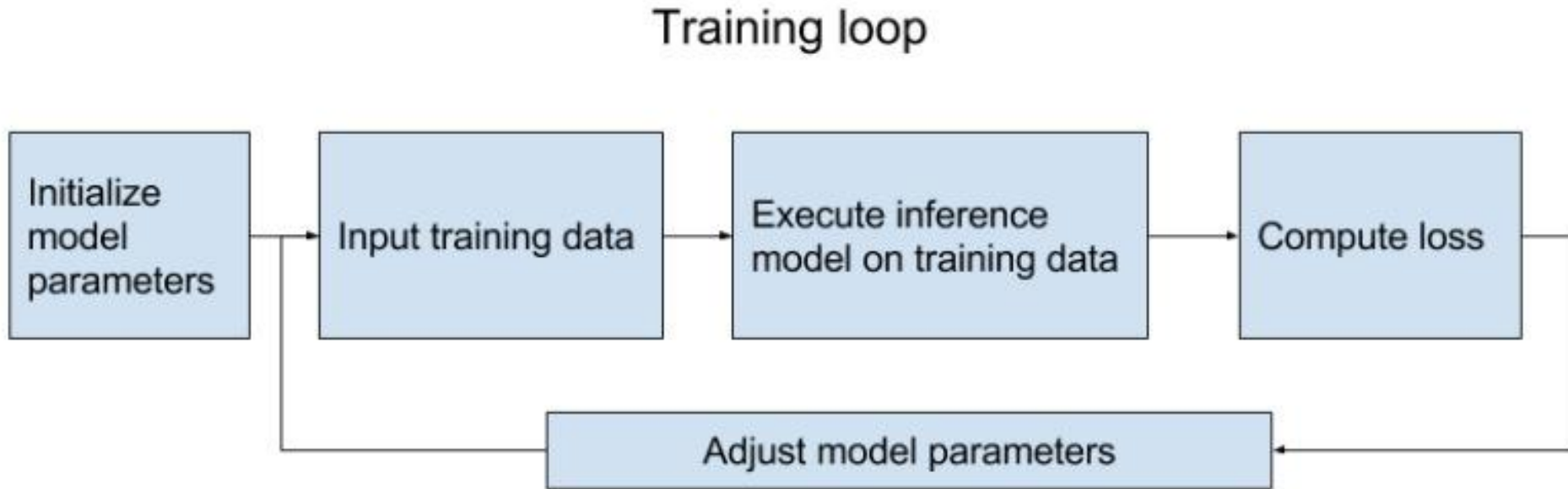
Interactive Coding!

# **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

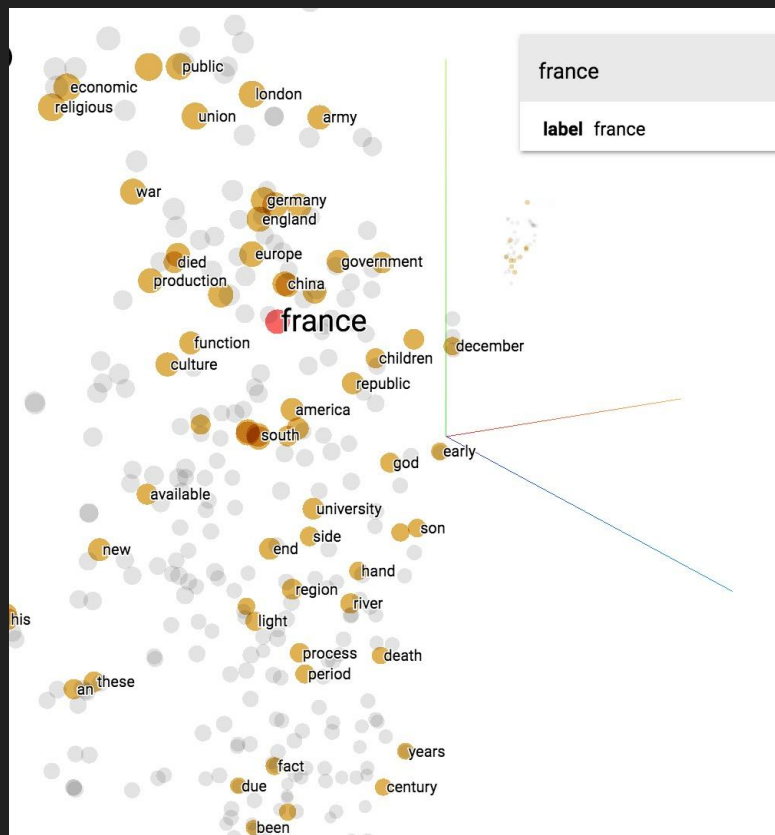


# Word Embedding

Capture the semantic relationships between words



# Word Embedding



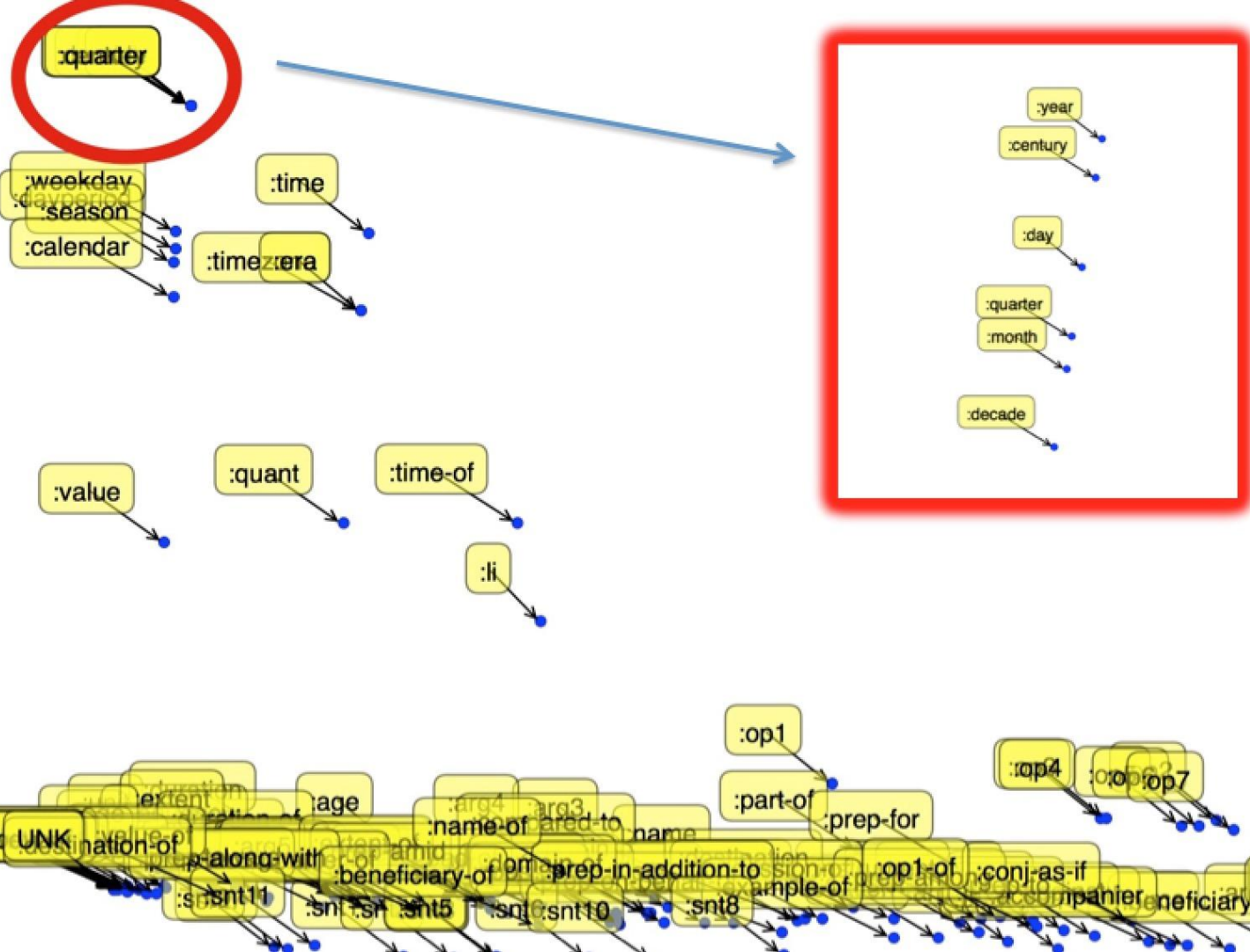
# Live visualization

# Count vs Predict

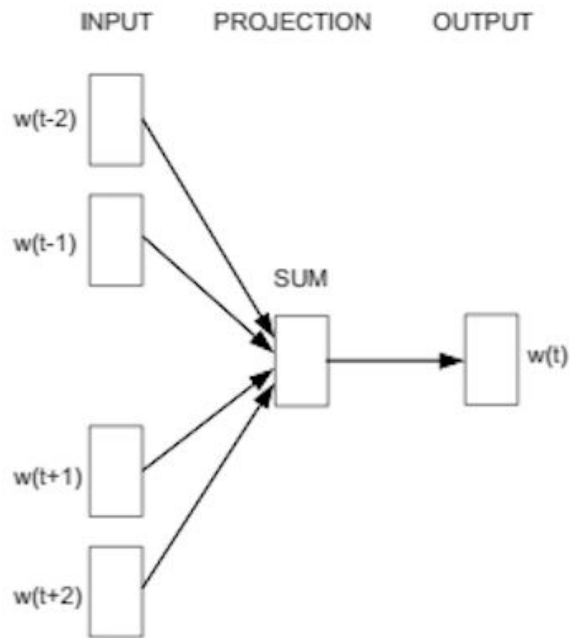
# Counting

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

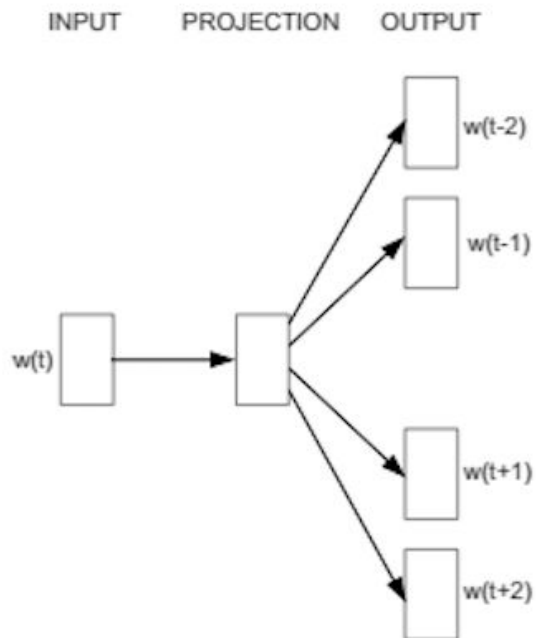
counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0



# Predicting



CBOW



Skip-gram

# 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

# 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

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

# Embedding Lookup

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

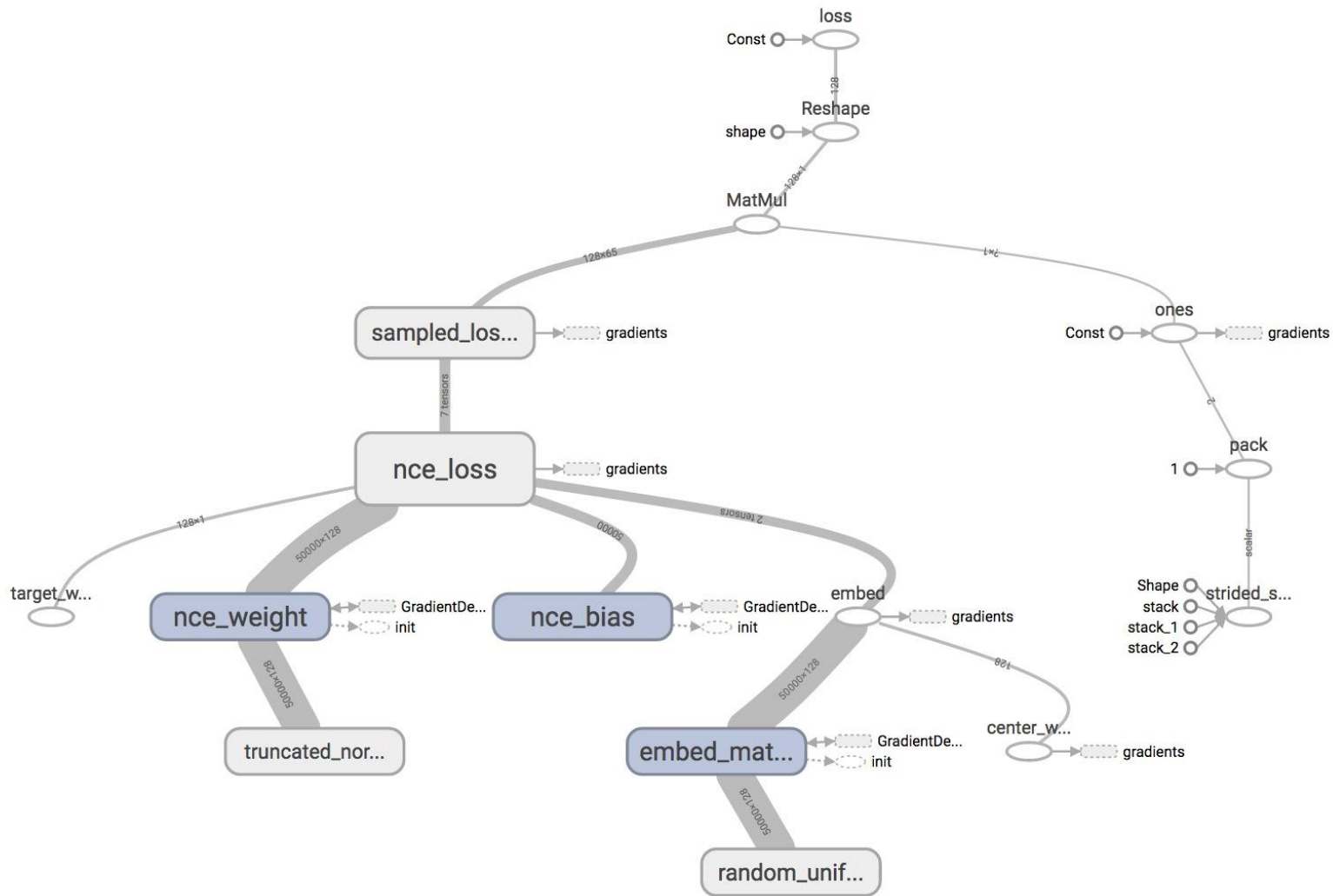
```
tf.nn.embedding_lookup(params, ids, partition_strategy='mod', name=None,  
                        validate_indices=True, max_norm=None)
```

# NCE Loss

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

# Let's write some code

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

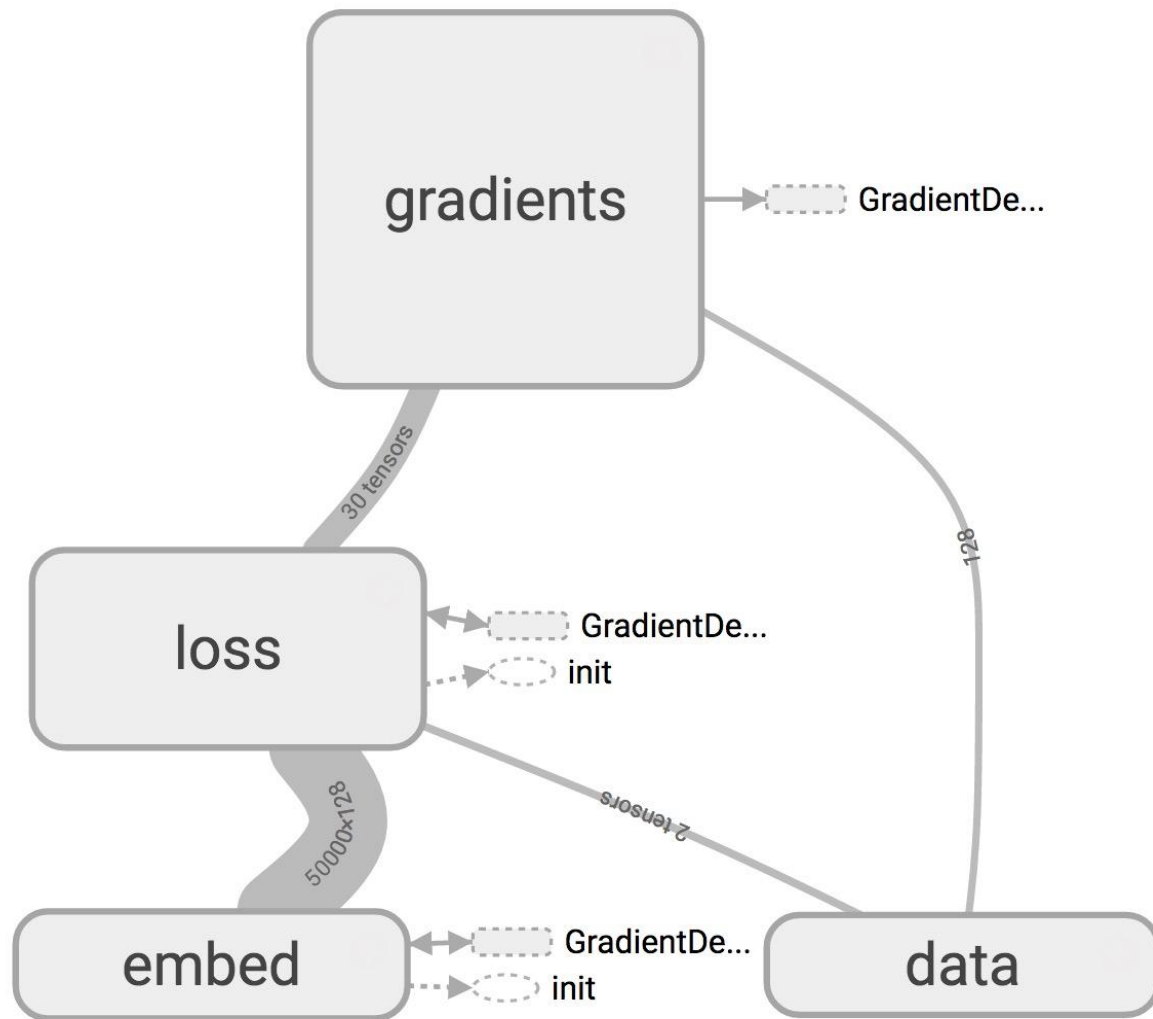




# Name scope

Group nodes together

with `tf.name_scope(name)`



# Next class

Manage experiments

Example: word2vec

Feedback: [huyenn@stanford.edu](mailto:huyenn@stanford.edu)

Thanks!