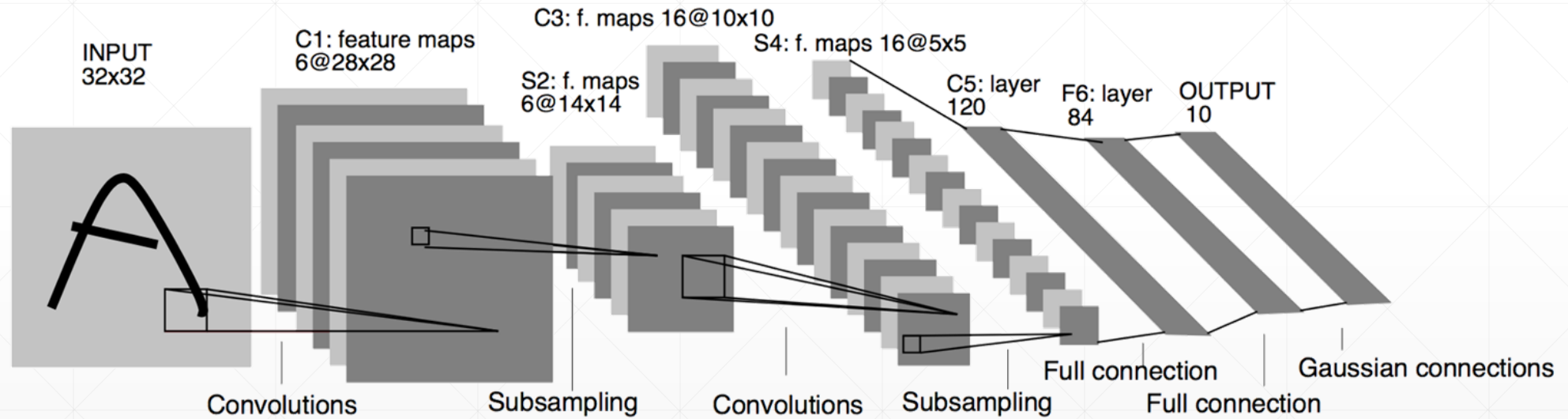




时间序列表示

主讲：龙良曲

Spatial signals



Temporal signals?



Text Message
Today 12:43 PM

Hey Caroline! This is Gerald :)
How are you?

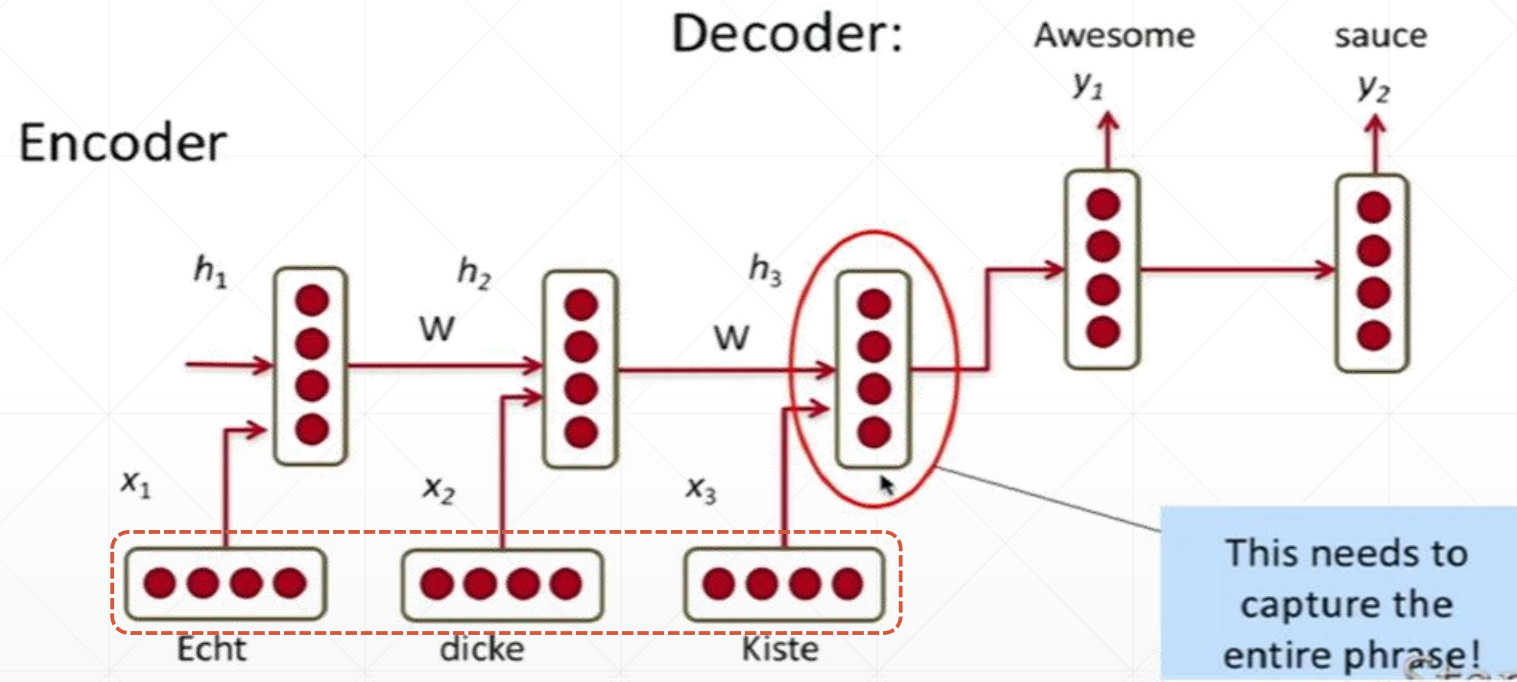
Hey Ger

I'm good! Super busy at work.
You?

What are you up too?


To*

Sequence

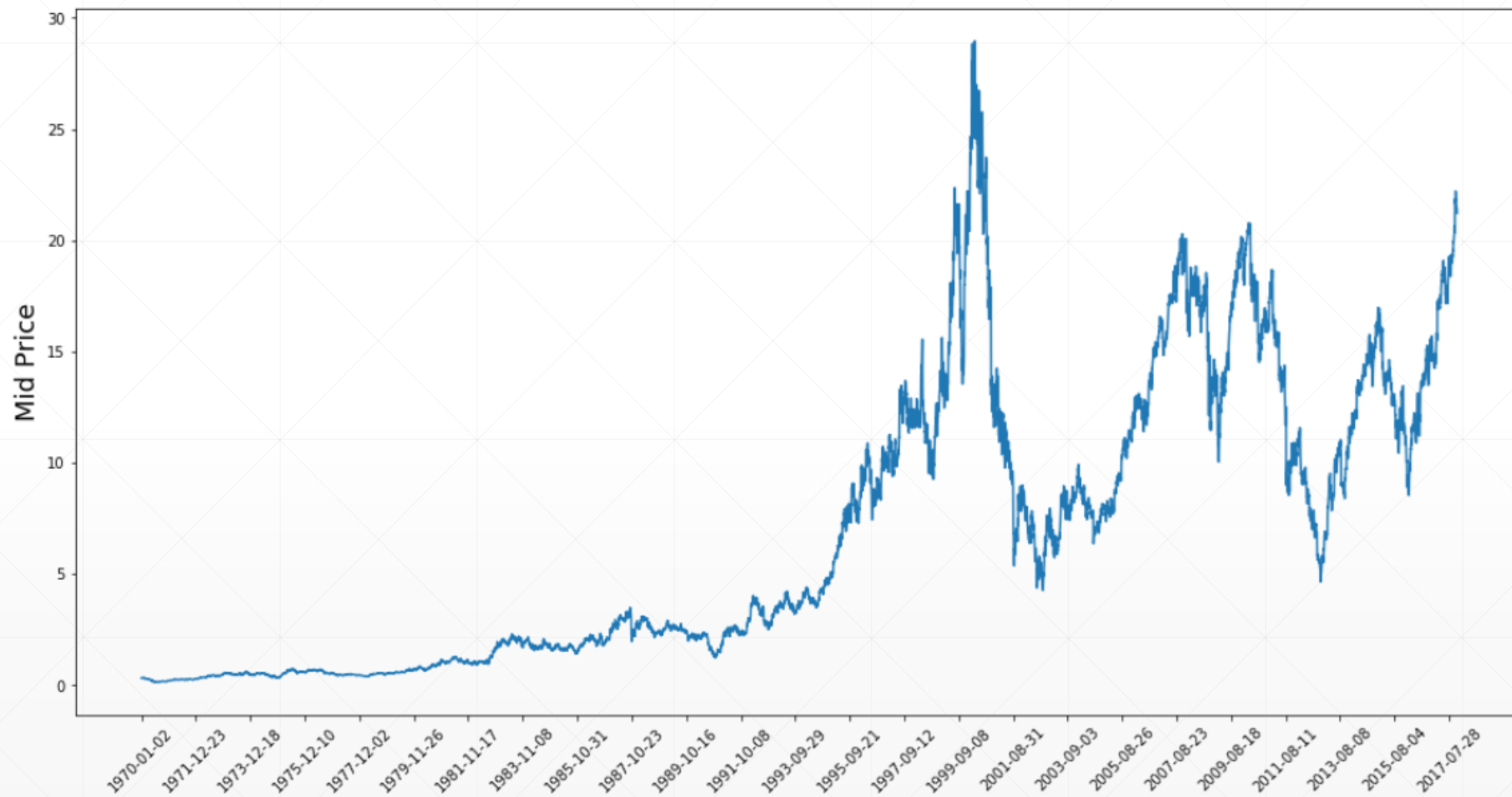


Sequence embedding

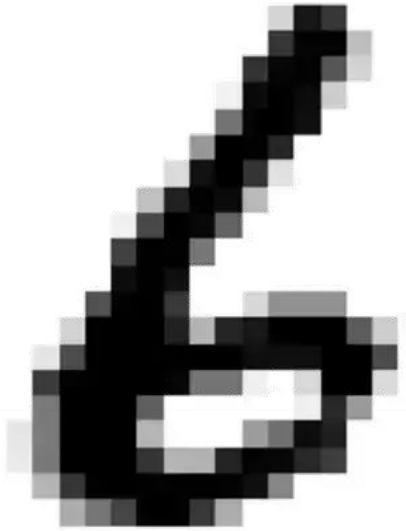
- `[b, seq_len, feature_len]`
- e.g. *I like it.*

```
  
<tf.Tensor: id=3, shape=(3, 3), dtype=float32, numpy=  
array([[1., 0., 0.],  
       [0., 1., 0.],  
       [0., 0., 1.]], dtype=float32)>
```

[b, 100, 1]

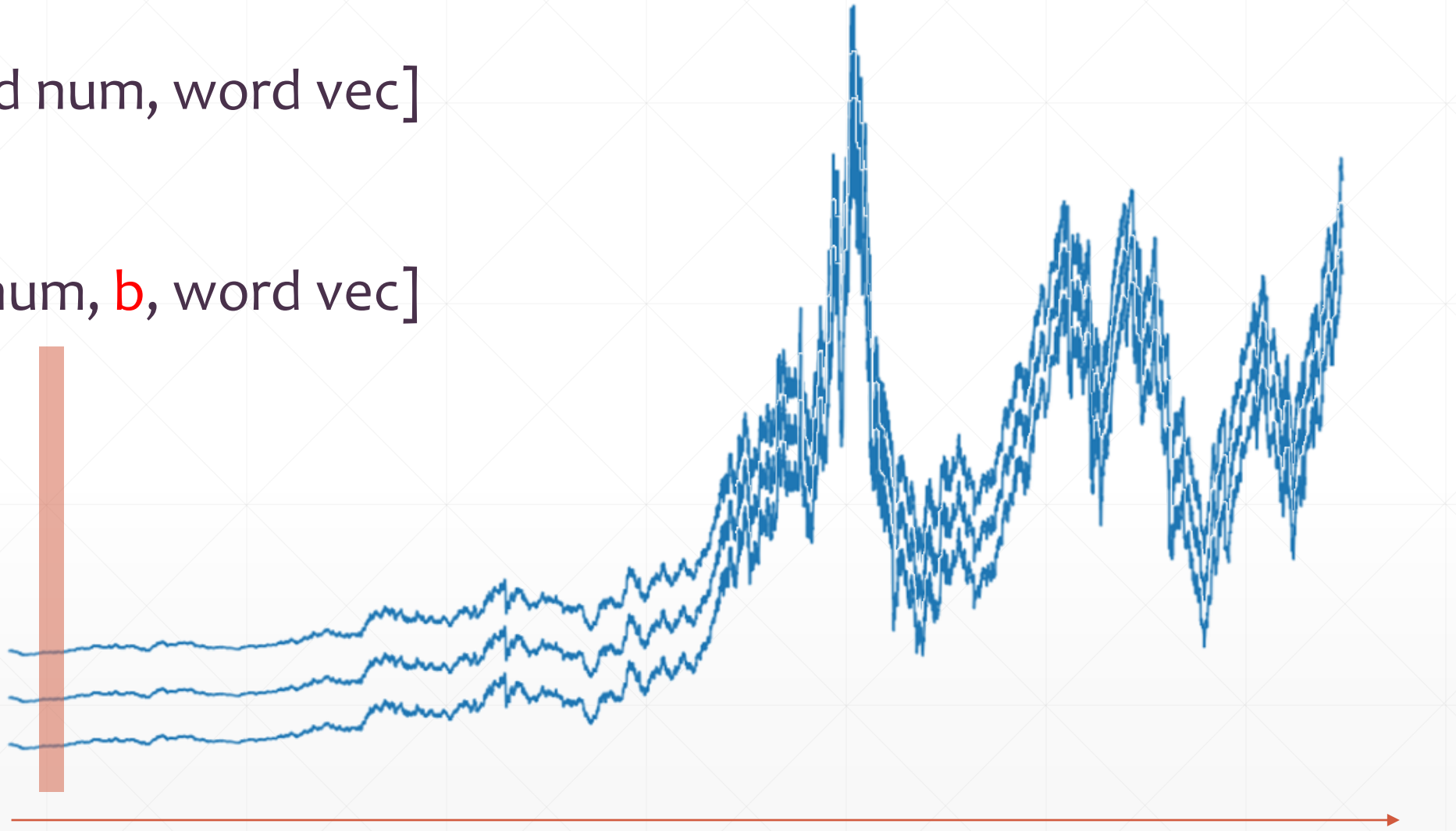


[b, 28, 28]



Batch

- [**b**, word num, word vec]
- [word num, **b**, word vec]



[words, word vec]

- How to represent a word

- [Rome, Italy, ...]

- one-hot

Rome = [1, 0, 0, 0, 0, 0, ..., 0]

Paris = [0, 1, 0, 0, 0, 0, ..., 0]

Italy = [0, 0, 1, 0, 0, 0, ..., 0]

France = [0, 0, 0, 1, 0, 0, ..., 0]

[words, word vec]

- sparse
- high-dim
- semantic similarity
- trainable

```
model.most_similar('king', topn=10)
```

(word, similarity with 'king')


('kings', 0.897245)
('baratheon', 0.809675)
('son', 0.763614)
('robert', 0.708522)
('lords', 0.698684)
('joffrey', 0.696455)
('prince', 0.695699)
('brother', 0.685239)
('aerys', 0.684527)
('stannis', 0.682932)

```
model.most_similar('queen', topn=10)
```

(word, similarity with 'queen')

('cersei', 0.942618)
('joffrey', 0.933756)
('margaery', 0.931099)
('sister', 0.928902)
('prince', 0.927364)
('uncle', 0.922507)
('varys', 0.918421)
('ned', 0.917492)
('melisandre', 0.915403)
('robb', 0.915272)

Word embedding



index	Embedding
0	[1.2, 3.1]
1	[0.1, 4.2]
2	[1.0, 3.1]
3	[0.3, 2.1]
4	[2.2, 1.4]
5	[0.7, 1.7]
6	[4.1, 2.0]

Word Embedding

- Word2Vec vs GloVe

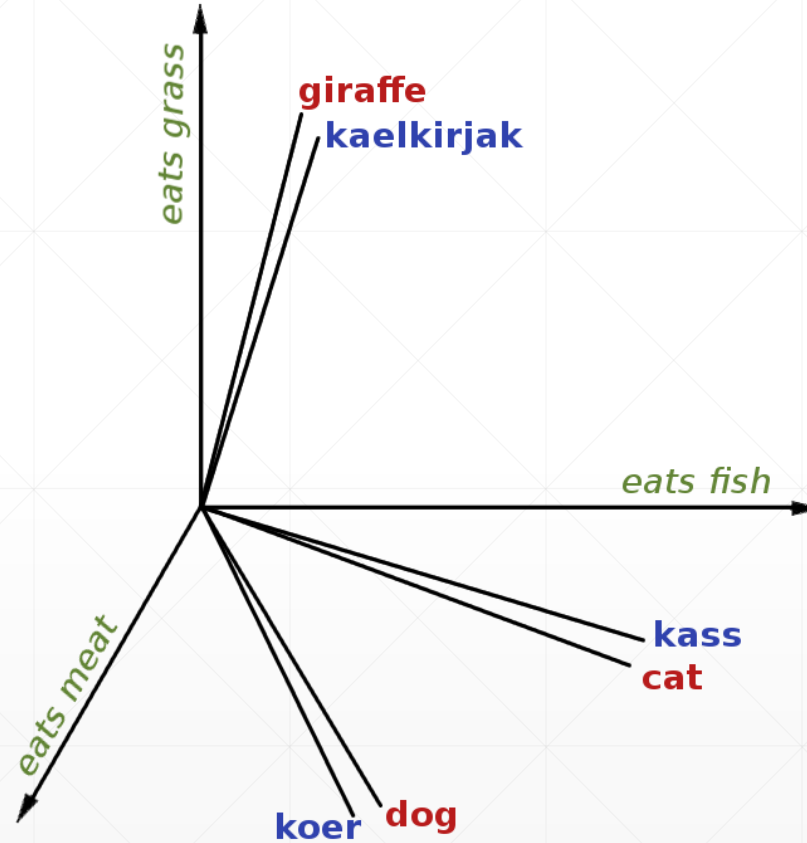


Fig. English VS Estonian

Embedding Layer

- Random initialized embedding

```
In [8]: from tensorflow.keras import layers
In [3]: x=tf.range(5)
In [5]: x=tf.random.shuffle(x)
Out[6]: <tf.Tensor: id=9, shape=(5,), dtype=int32, numpy=array([0, 4, 1, 3, 2])>

In [9]: net=layers.Embedding(10, 4)
In [10]: net(x)
<tf.Tensor: id=25, shape=(5, 4), dtype=float32, numpy=
array([[ -0.03535435,  0.01710499,  0.024379 , -0.010867 ],
       [ -0.03977622,  0.01753286,  0.00805125,  0.00836002],
       [ -0.0119989 ,  0.01030685, -0.01133521,  0.02052242],
       [  0.02230989, -0.02186236,  0.01720804, -0.03888531],
       [ -0.04997355,  0.00911248,  0.01886252,  0.01570504]],
      dtype=float32)>
```

Embedding Layer



```
In [11]: net.trainable
```

```
Out[11]: True
```

```
In [12]: net.trainable_variables
```

```
[<tf.Variable 'embedding/embeddings:0' shape=(10, 4) dtype=float32, numpy=
array([[ -0.03535435,  0.01710499,  0.024379   , -0.010867   ],
       [ -0.0119989 ,  0.01030685, -0.01133521,  0.02052242],
       [ -0.04997355,  0.00911248,  0.01886252,  0.01570504],
       [  0.02230989, -0.02186236,  0.01720804, -0.03888531],
       [ -0.03977622,  0.01753286,  0.00805125,  0.00836002],
       [ -0.03328228,  0.04350821, -0.03760867, -0.02835478],
       [  0.03658437, -0.00686272, -0.00069226,  0.02232203],
       [ -0.02224611, -0.00517293, -0.00249453,  0.00809779],
       [  0.00234641, -0.01876179, -0.03804705, -0.02393213],
       [ -0.01690916, -0.04798424, -0.04477951, -0.03771662]]),
 dtype=float32)>]
```

下一课时

循环神经网络

Thank You.
