

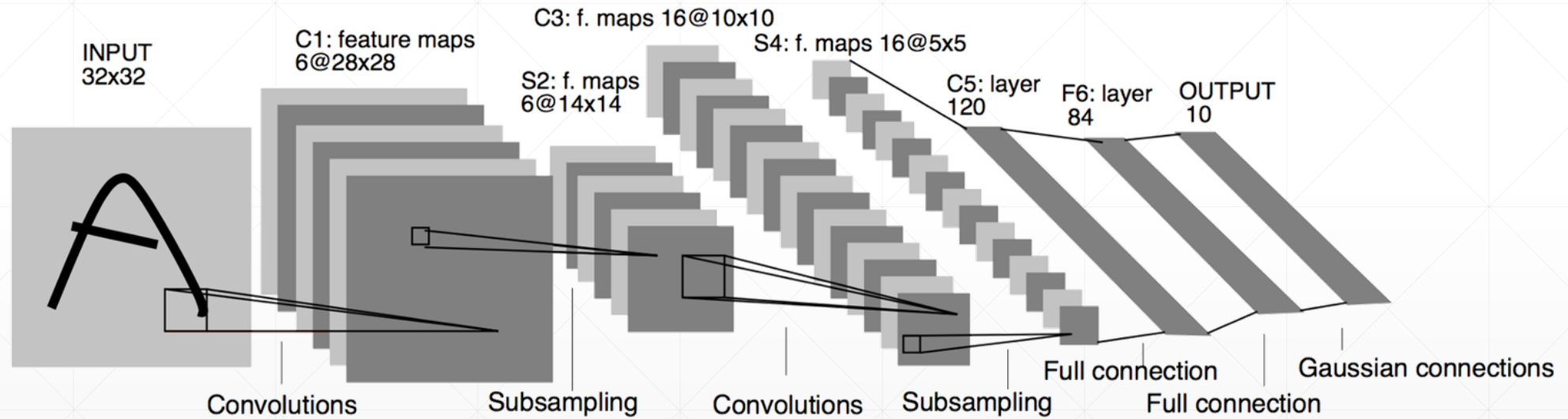


# 时间序列表示

---

主讲人：龙良曲

# Spatial Signals



# Temporal Signals?



Text Message  
Today 12:43 PM

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

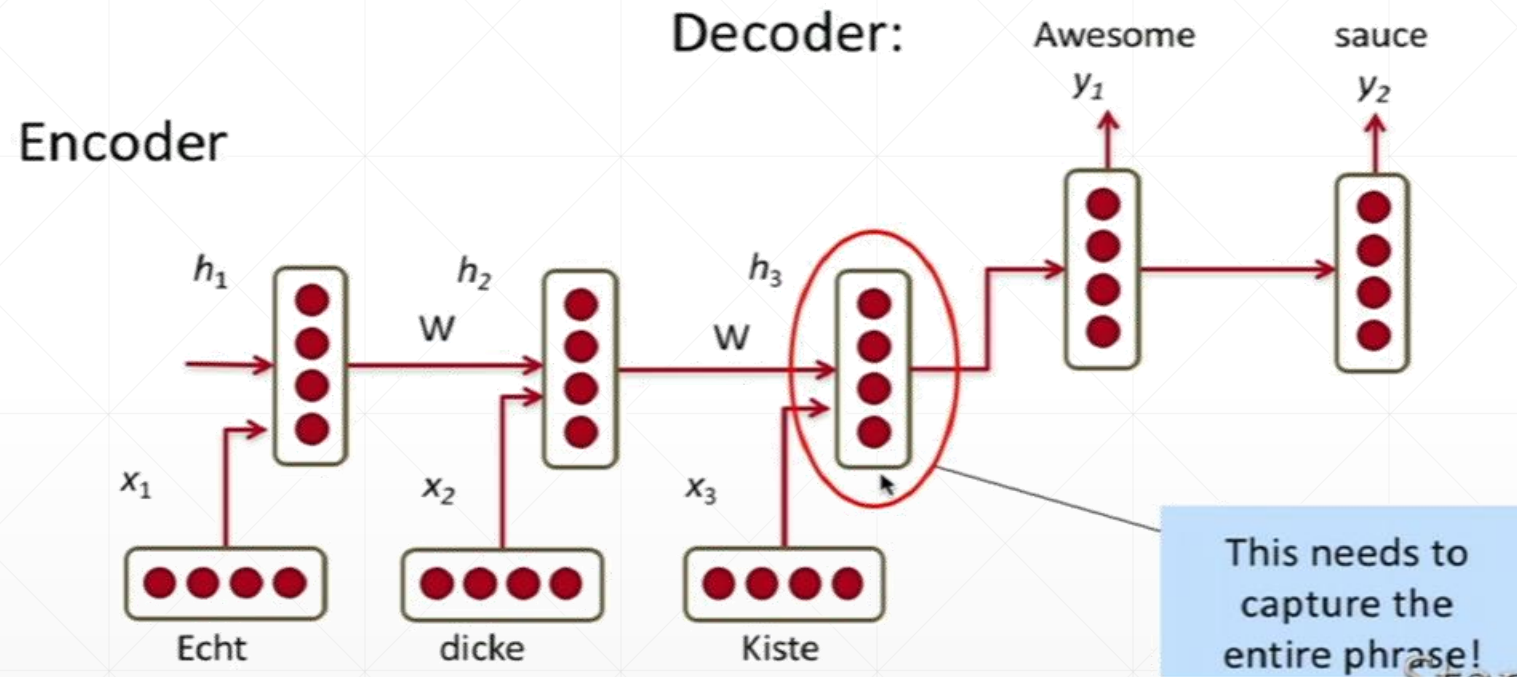
Hey Ger

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

What are you up too?

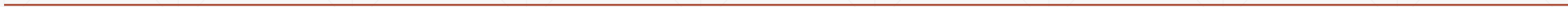
To\*

# Sequence

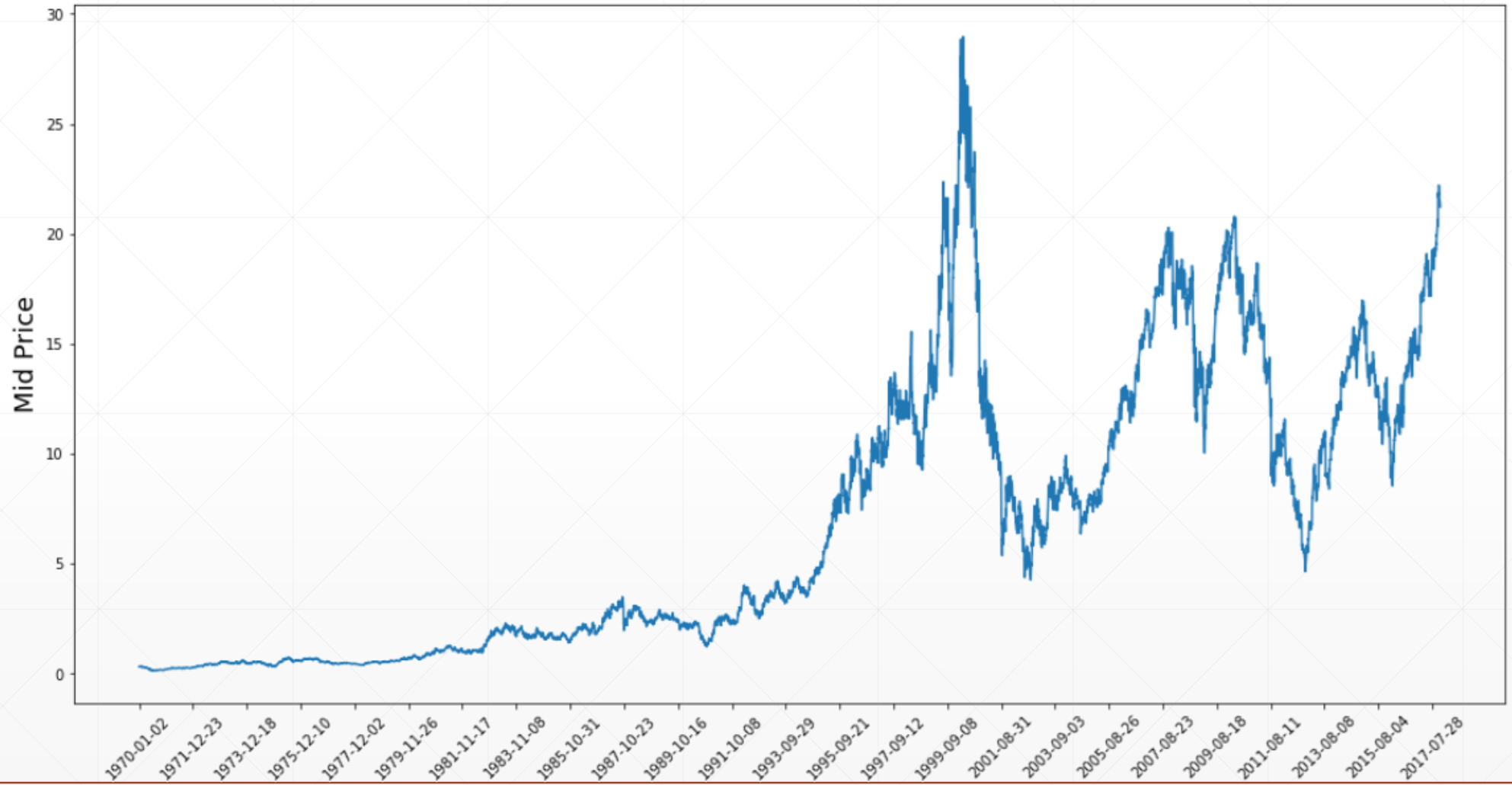


# Sequence representation

- [seq\_len, feature\_len]



[100, 1]



**[28, 28]**



# [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

```
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)

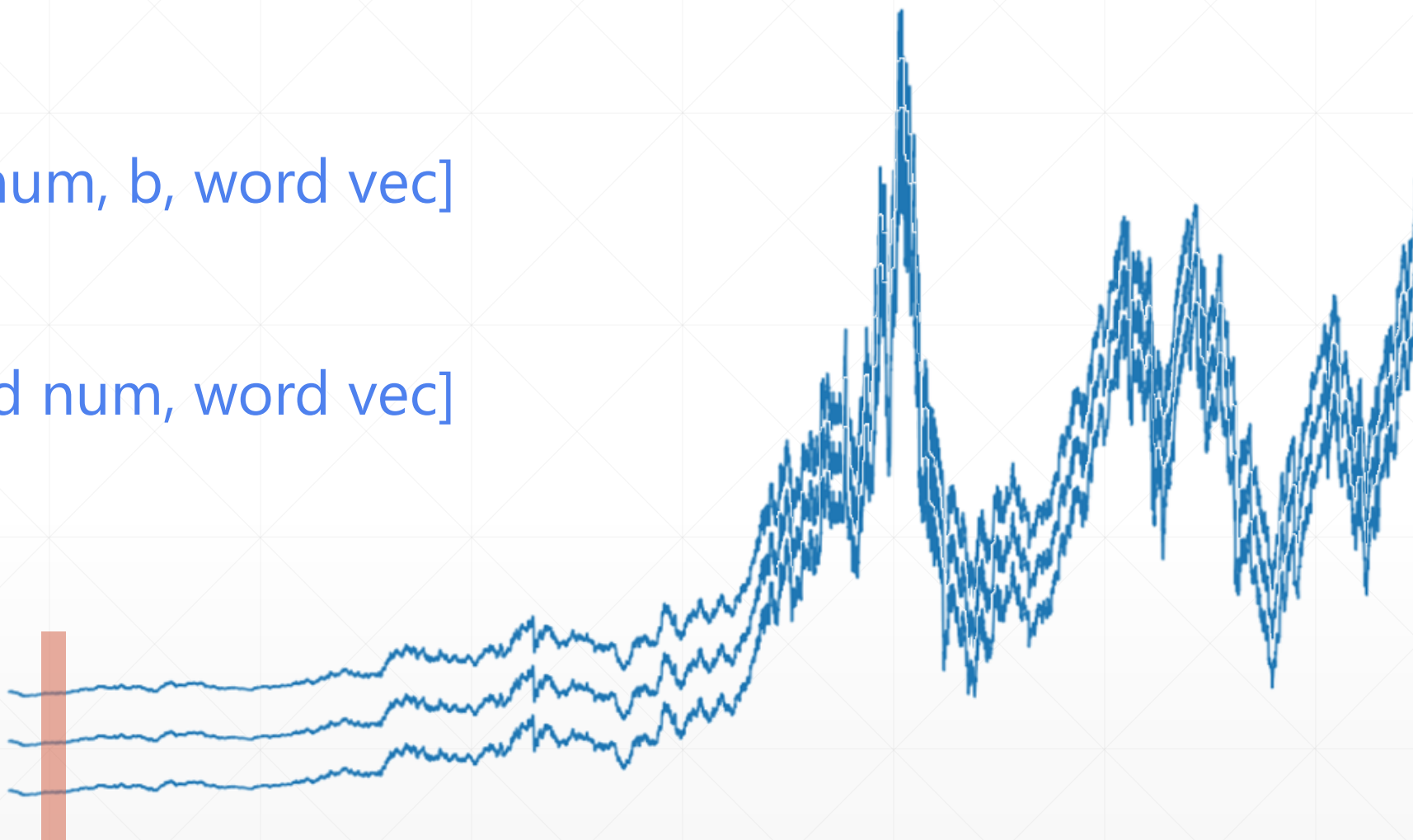
```
model.most_similar('queen', topn=10)
```

(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)

# Batch

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



# word2vec vs GloVe

```
word_to_ix = {"hello": 0, "world": 1}

lookup_tensor = torch.tensor([word_to_ix["hello"]], dtype=torch.long)

embeds = nn.Embedding(2, 5) # 2 words in vocab, 5 dimensional embeddings
hello_embed = embeds(lookup_tensor)
print(hello_embed)
tensor([[ 0.6614,  0.2669,  0.0617,  0.6213, -0.4519]],
       grad_fn=<EmbeddingBackward>)
```

*Word Vector  
Lookup Table!*

*300 features*

*10,000 words*



# PyTorch



```
1 from torch.nn import GloVe
2 vectors = GloVe()
3
4 vectors['hello']
5 -1.7494
6 0.6242
7 ...
8 -0.6202
9 2.0928
10 [torch.FloatTensor of size 100]
```

# 下一课时

---

RNN原理

**Thank You.**

---