Predictive Analytics using Neural Networks: Applications, Pitfalls and Beyond

Bratislava, MLMU, 24th January, 2018

Rudradeb Mitra
https://www.linkedin.com/in/mitrar/

Today’s talk (~50-60 mins)

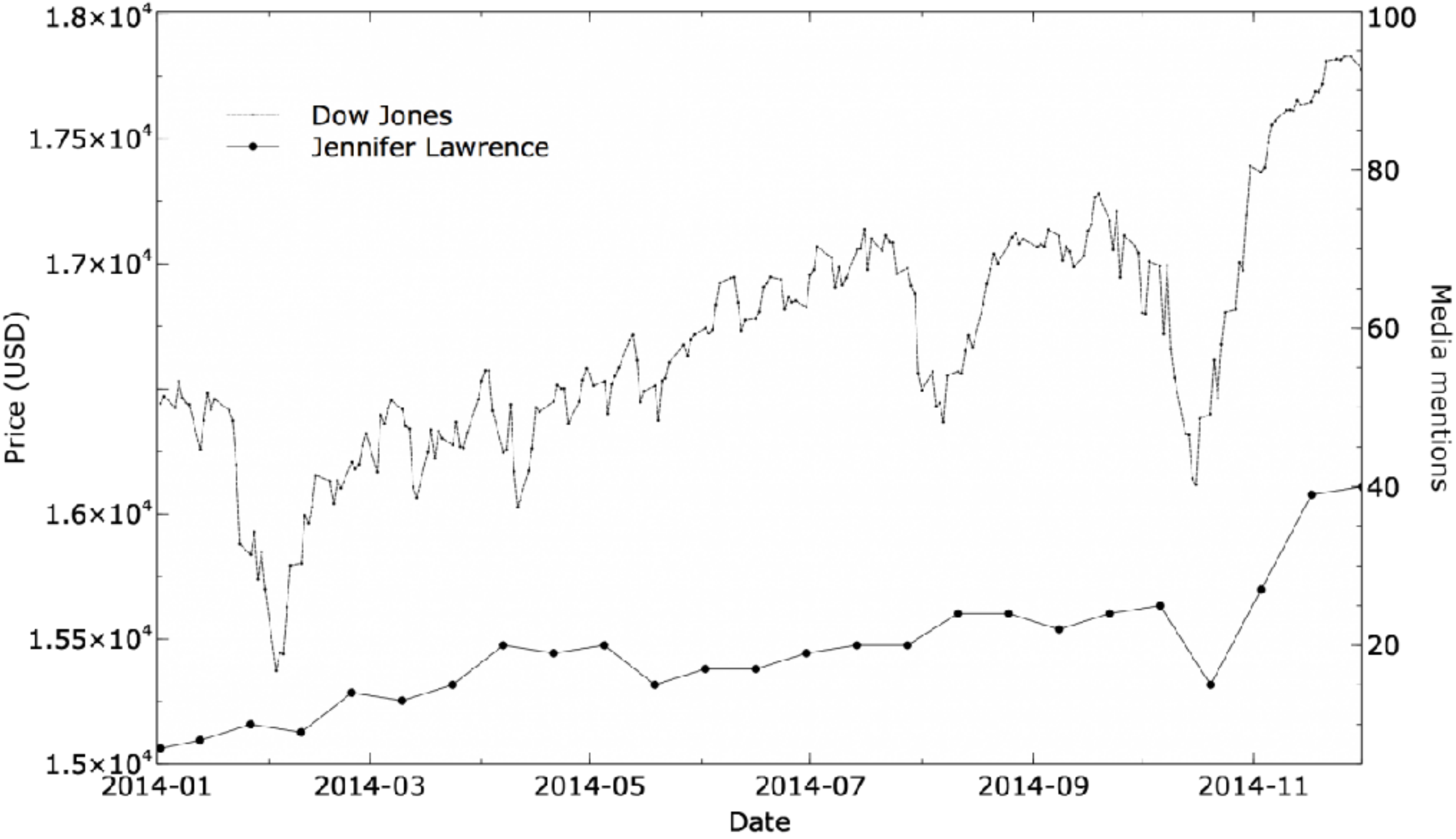
- What is Time-series data and Predictive Analytics?
- One case study - How Neural networks can be used for predicting next purchase
    - word2vec
    - Long Short-Term Memory (LSTM)
    - Neural networks using Reinforcement Learning
- Other applications and pitfalls

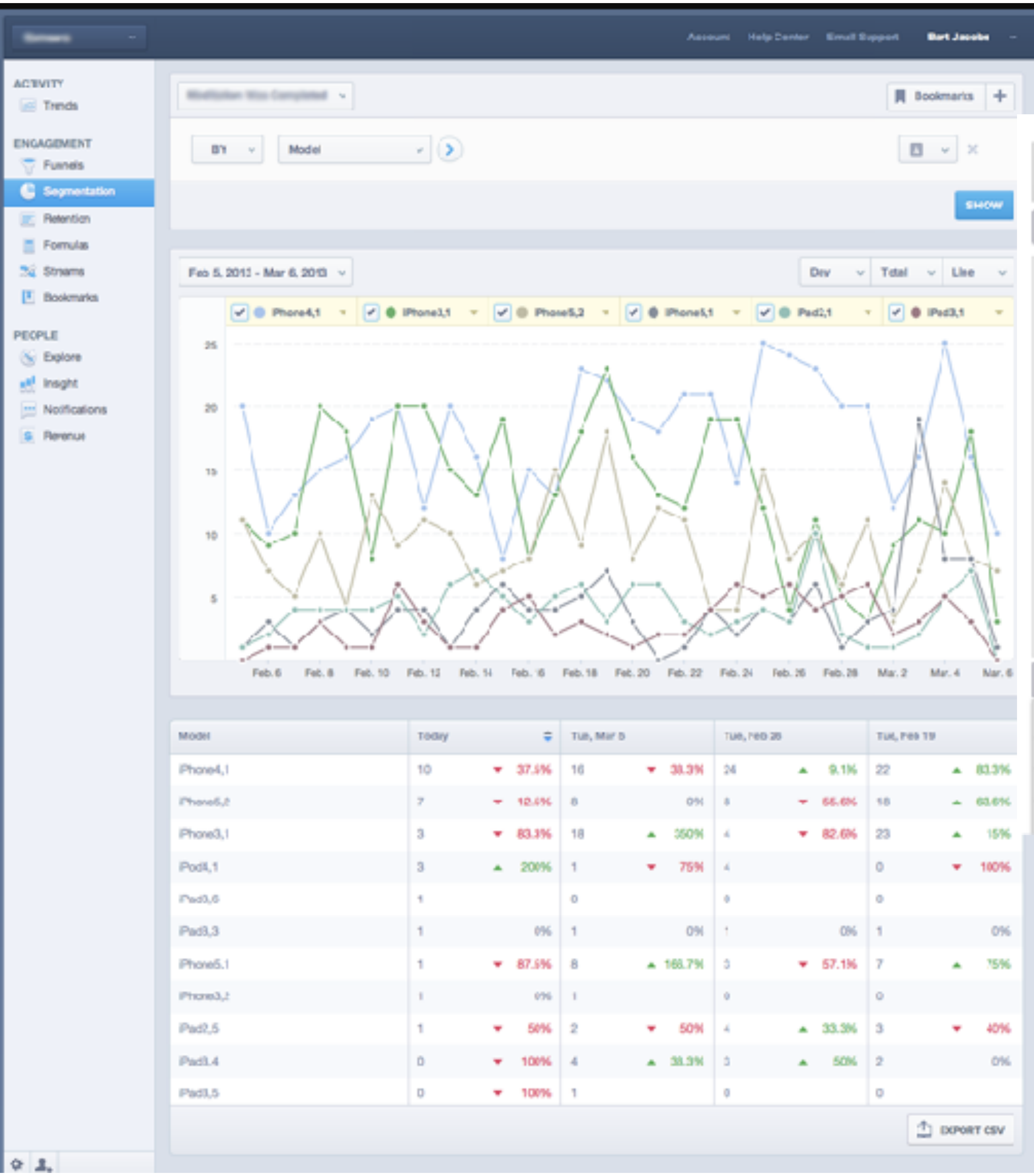
Why?

- Bridge gap between technical AI and practical AI
- You can build your own applications using Neural networks.

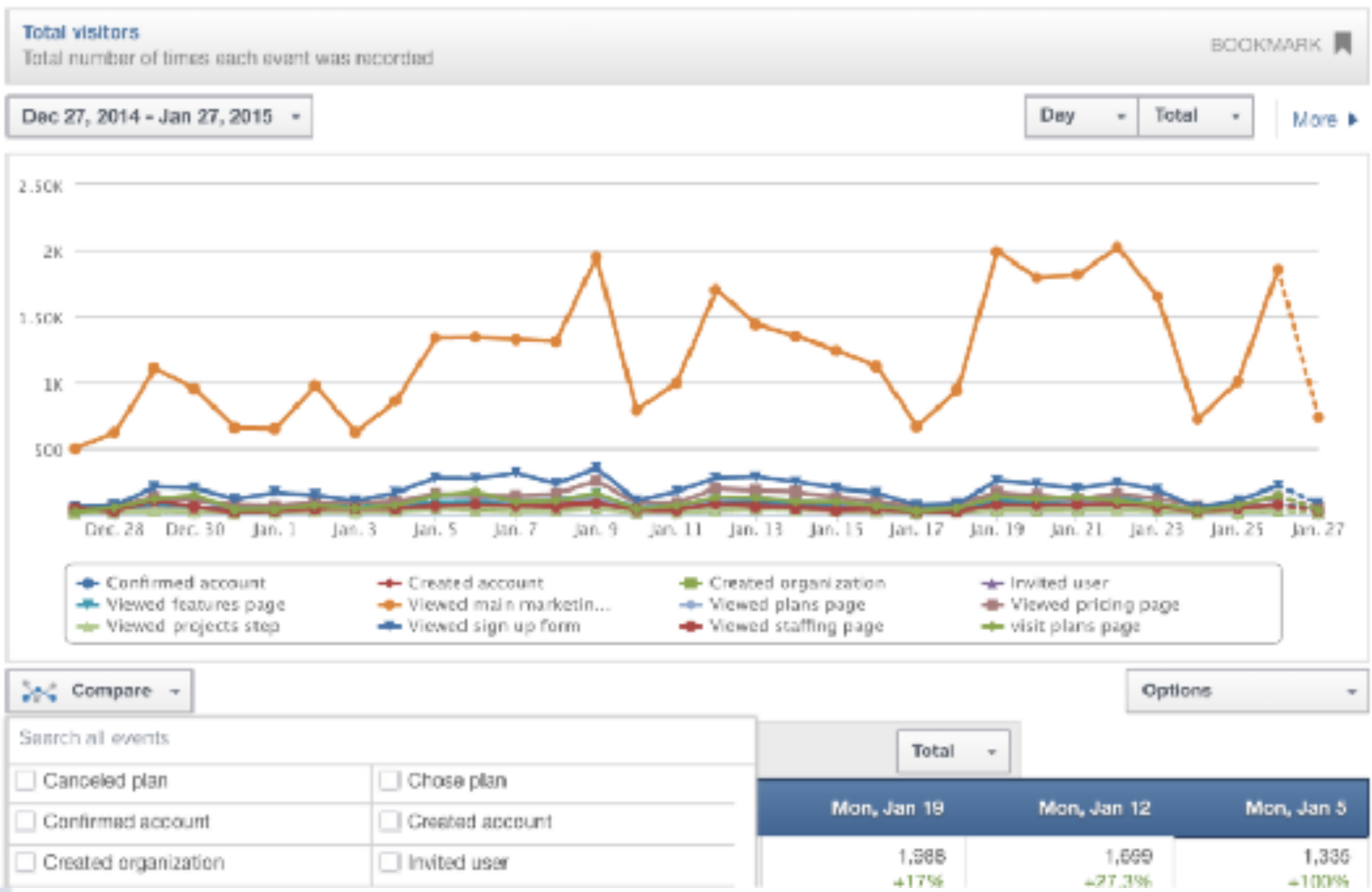
Time series data

Time series data





Model	Today		Tue, Mar 5		Tue, Feb 26		Tue, Feb 19	
iPhone4,1	10	▼ 37.5%	16	▼ 33.3%	24	▲ 9.1%	22	▲ 83.3%
iPhone5,2	7	▼ 12.5%	8	0%	8	▼ 55.6%	18	▲ 63.6%
iPhone3,1	3	▼ 83.3%	18	▲ 350%	4	▼ 82.6%	23	▲ 15%
iPod4,1	3	▲ 200%	1	▼ 75%	4		0	▼ 100%
iPad3,6	1		0		0		0	
iPad3,3	1	0%	1	0%	1	0%	1	0%
iPhone5,1	1	▼ 87.5%	8	▲ 166.7%	3	▼ 57.1%	7	▲ 75%
iPhone3,2	1	0%	1		0		0	
iPad2,5	1	▼ 50%	2	▼ 50%	4	▲ 33.3%	3	▼ 40%
iPad3,4	0	▼ 100%	4	▲ 33.3%	3	▲ 50%	2	0%
iPad3,5	0	▼ 100%	1		0		0	



Total visitors

Total number of times each event was recorded

☐ Canceled plan ☐ Chose plan
☐ Confirmed account ☐ Created account
☐ Created organization ☐ Invited user

Mon, Jan 19	Mon, Jan 12	Mon, Jan 5
1,988 +17%	1,699 +27.3%	1,335 +100%

What is missing?

Predictive Analytics

Case study

Predicting next purchase

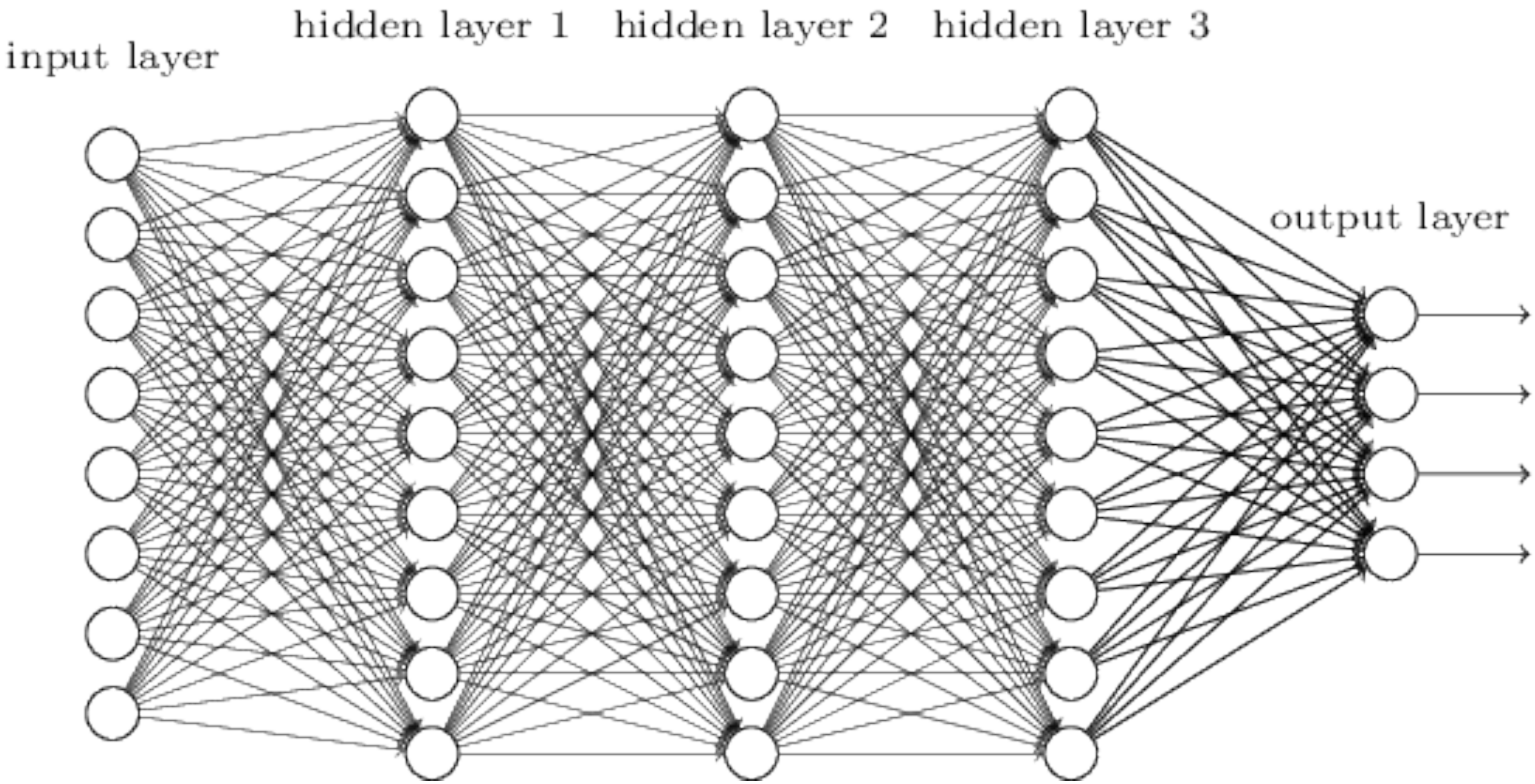
Kale&
Avocados&
Quinoa&
instacart
Groceries Delivered From Local Stores

Neural Networks

Neural Network

Approximation function

Learning via Back propagation



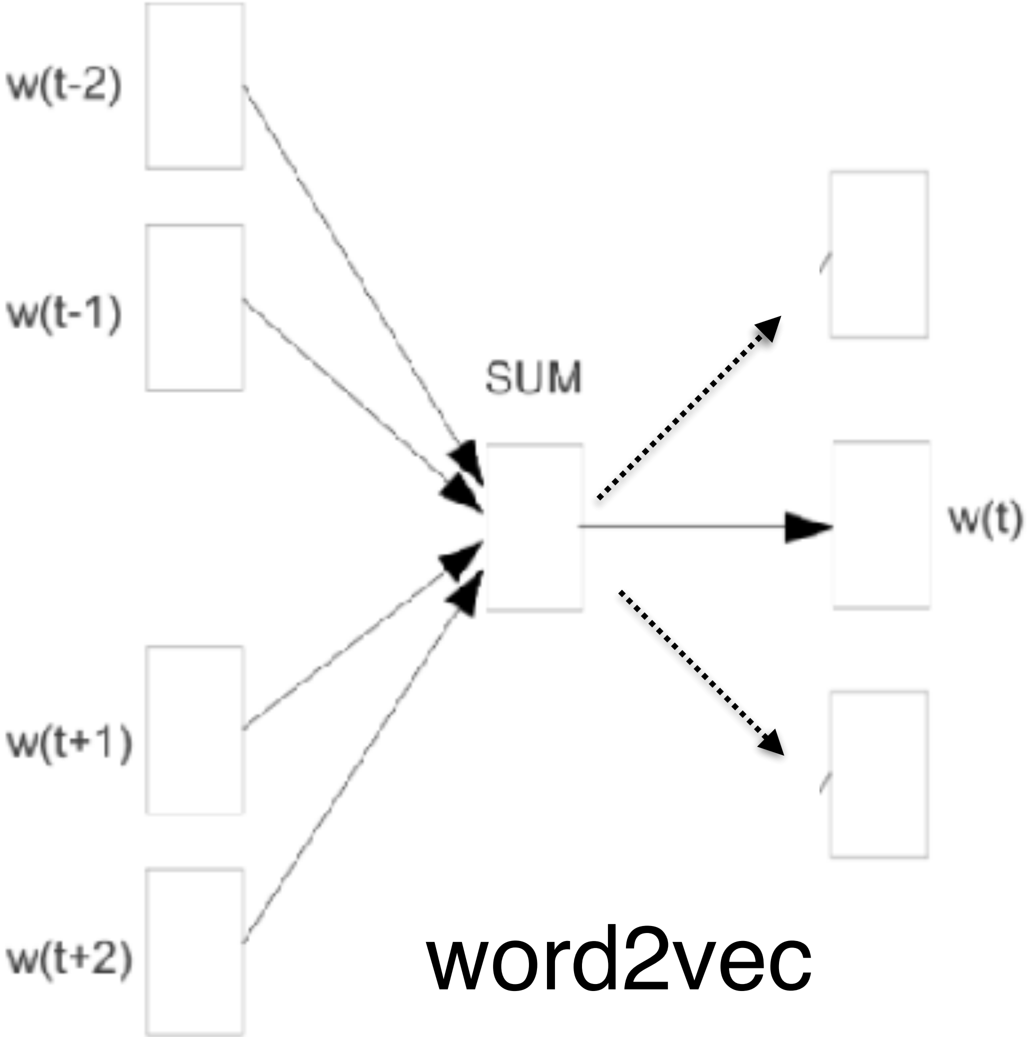
But how Neural Network can help in predicting?

word2vec- Predicting next word

Input words embeddings

Hidden Layer

Output word embedding



How

Are

You

Feeling

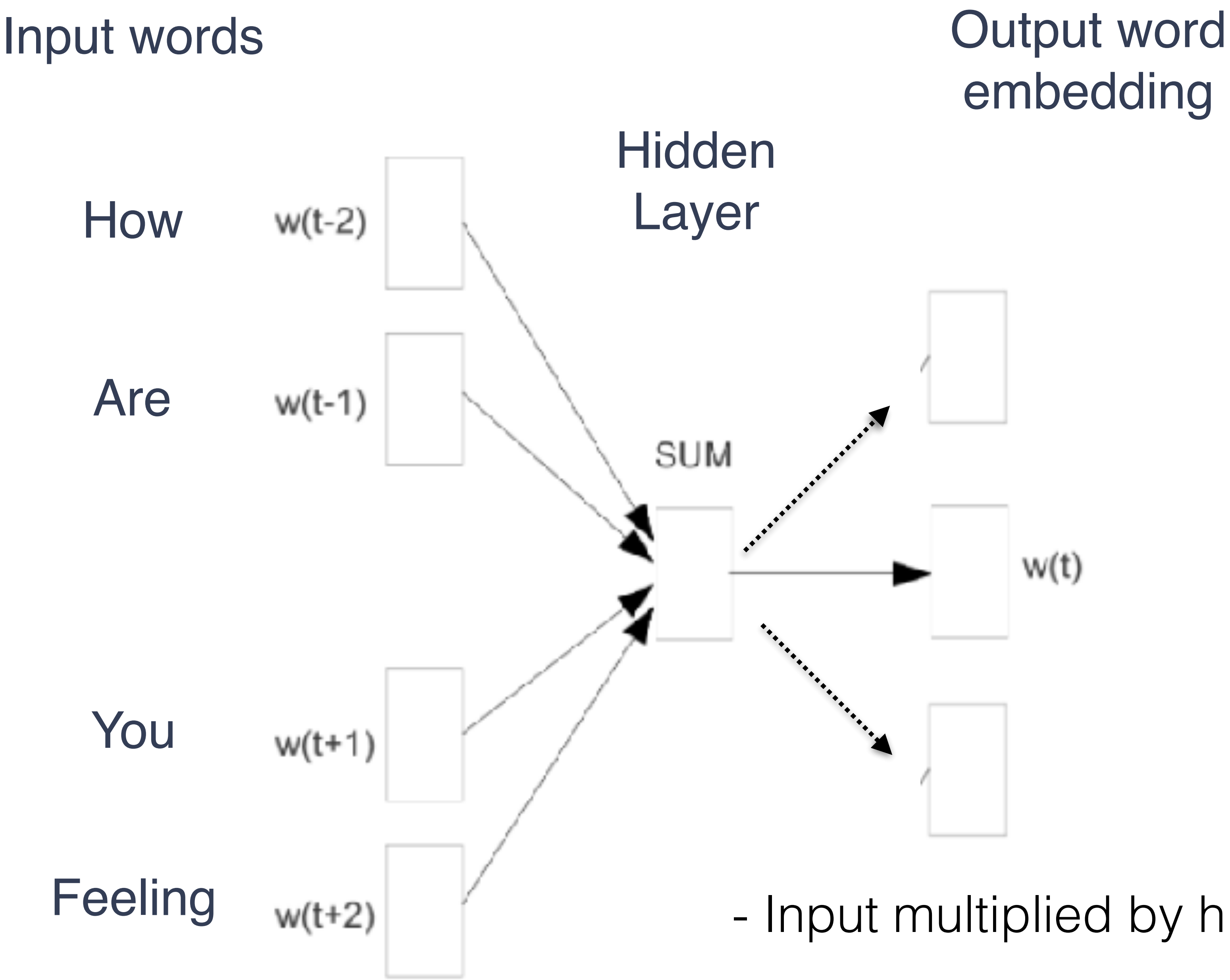
Predicted next word

Today

word2vec

- Does not understand words or grammar

Semi supervised learning



Input words

Output word embedding

Hidden Layer

How

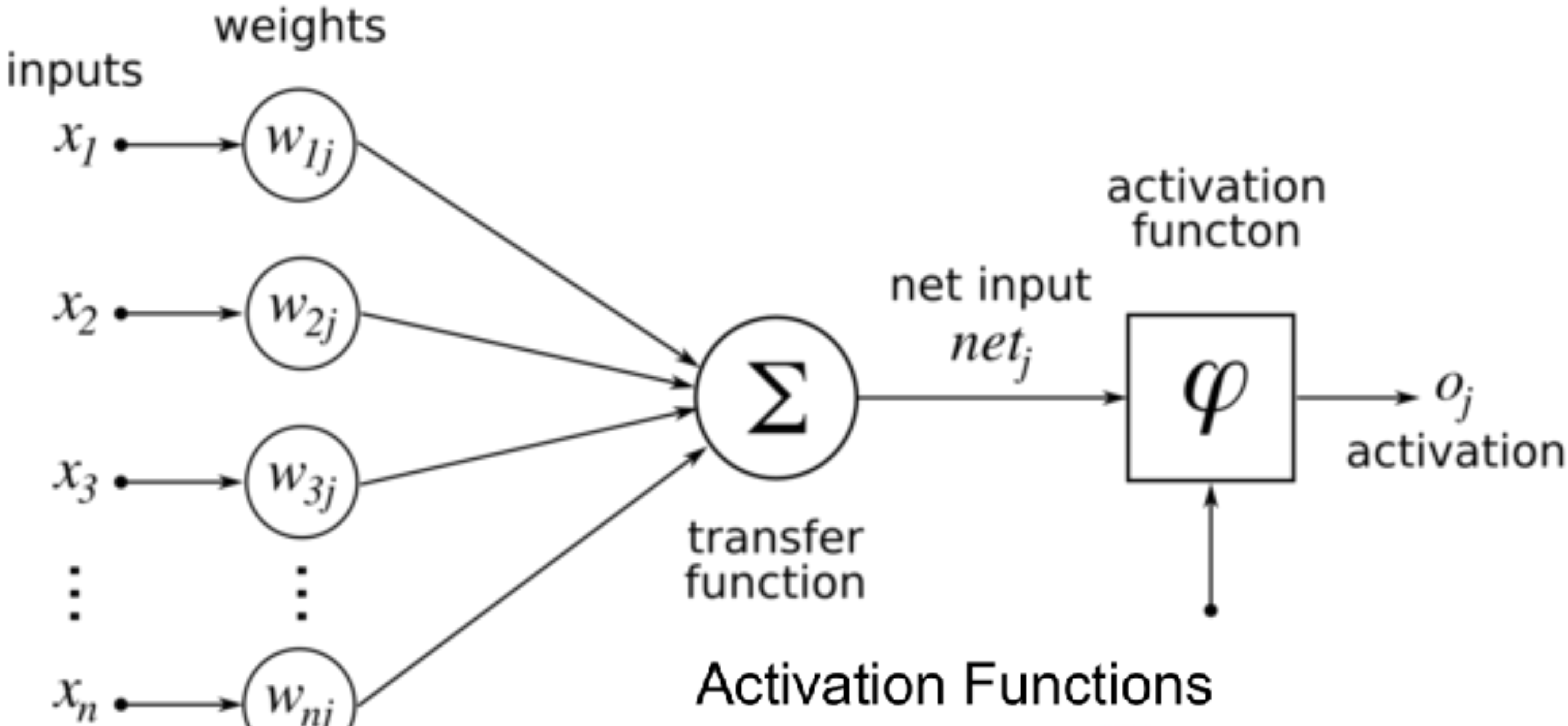
Are

You

Feeling

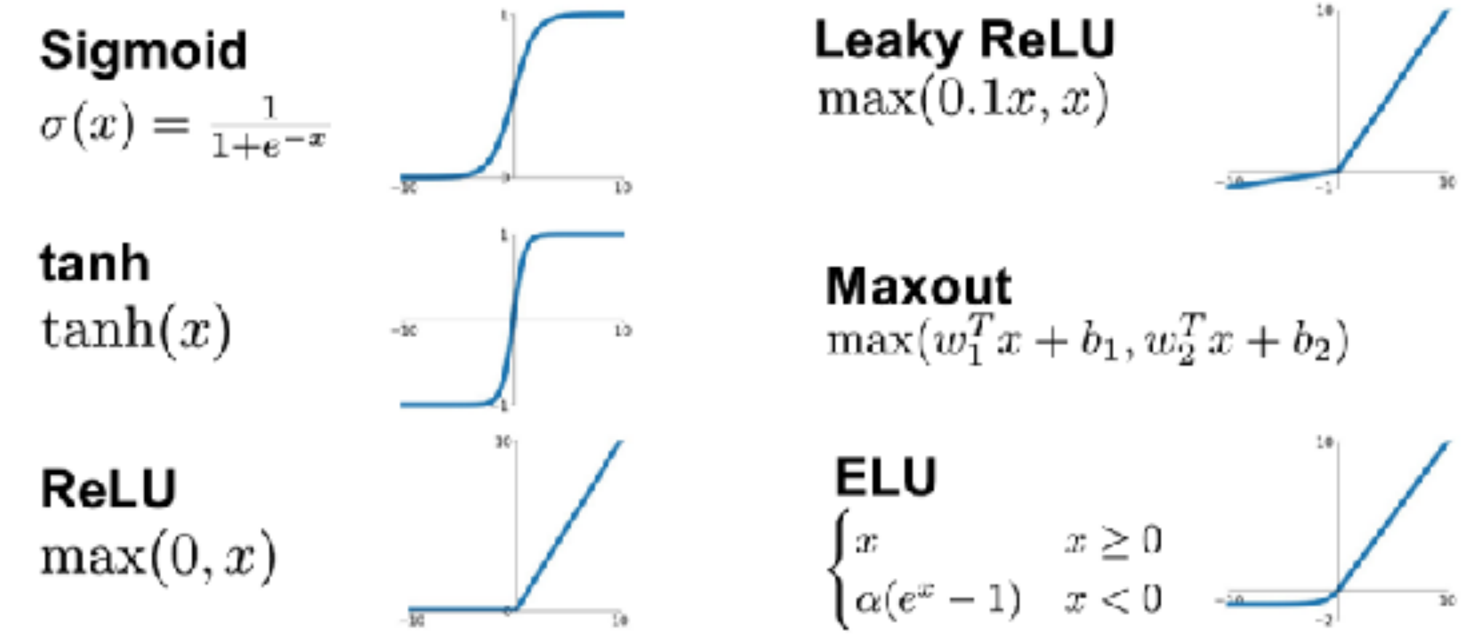
- Input multiplied by hidden input weights
- Hidden input multiplied by hidden output weights
- Converts output to probabilities through softmax
- Error back propagated

Activation function



Softmax
- Multiclass regression
- Extension of sigmoid to multiclass

Activation Functions



Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$

tanh
$\tanh(x)$

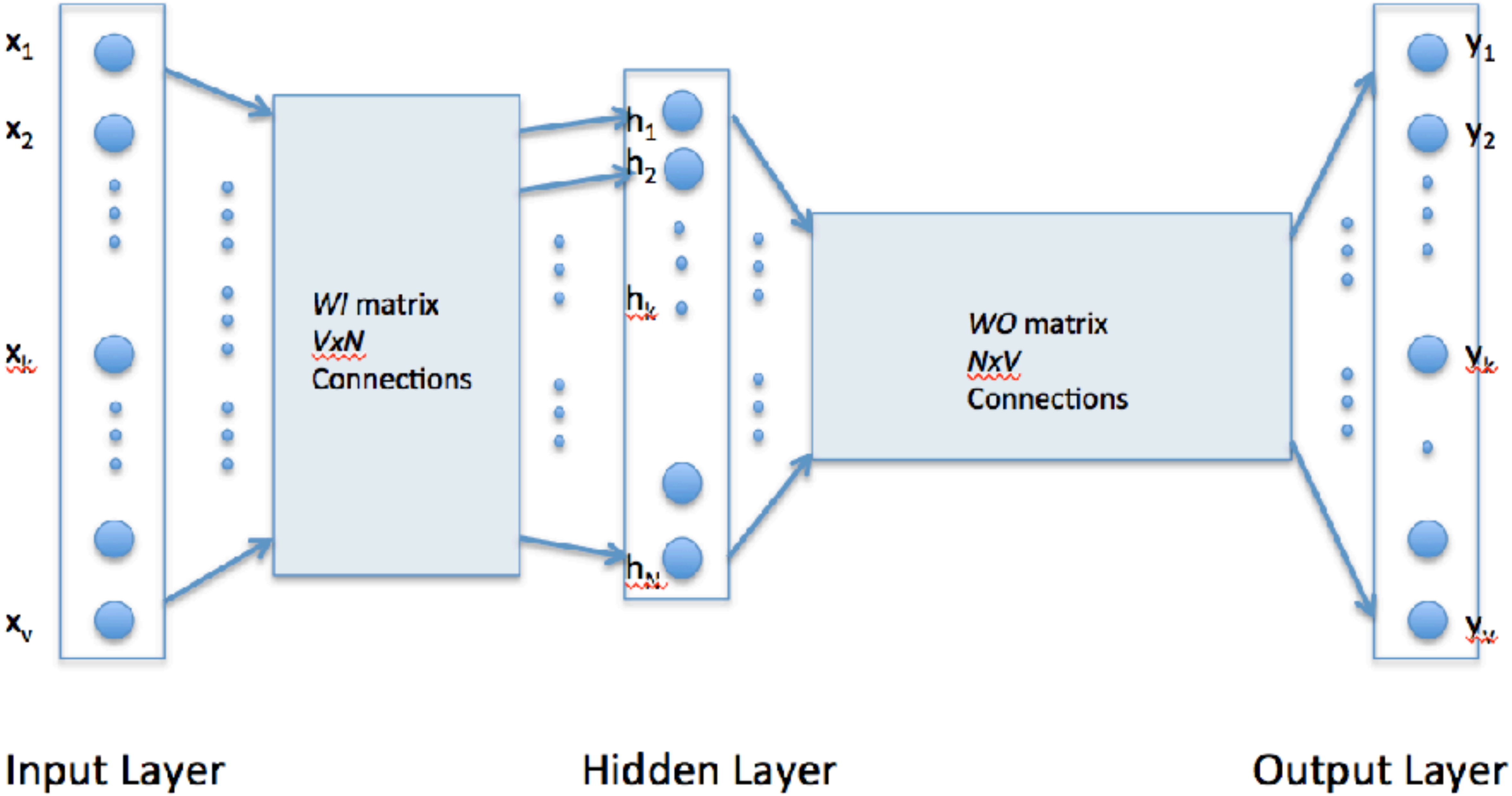
ReLU
$\max(0, x)$

Leaky ReLU
$\max(0.1x, x)$

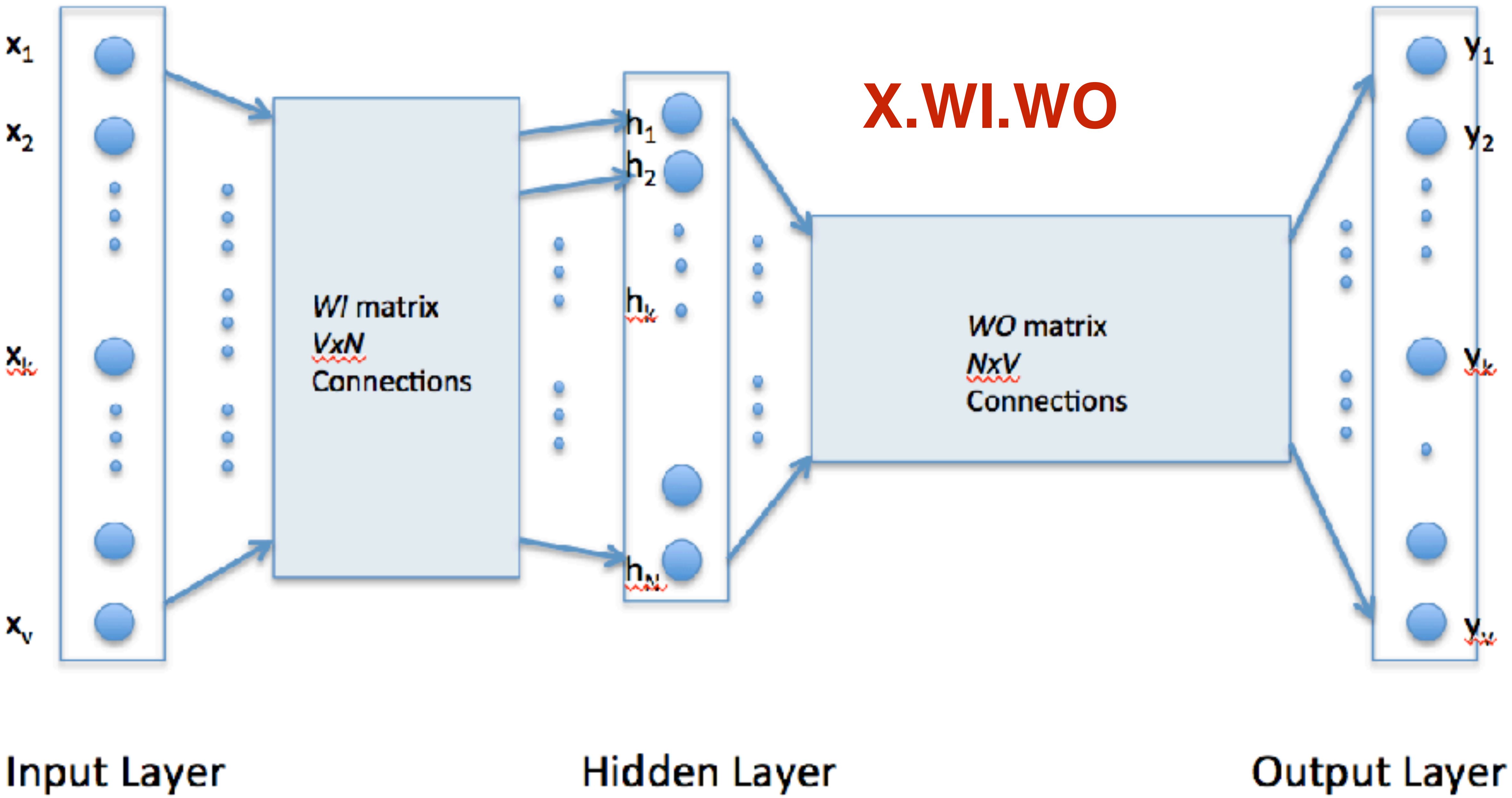
Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

word2vec - Something more interesting?



Word2vec Word embeddings



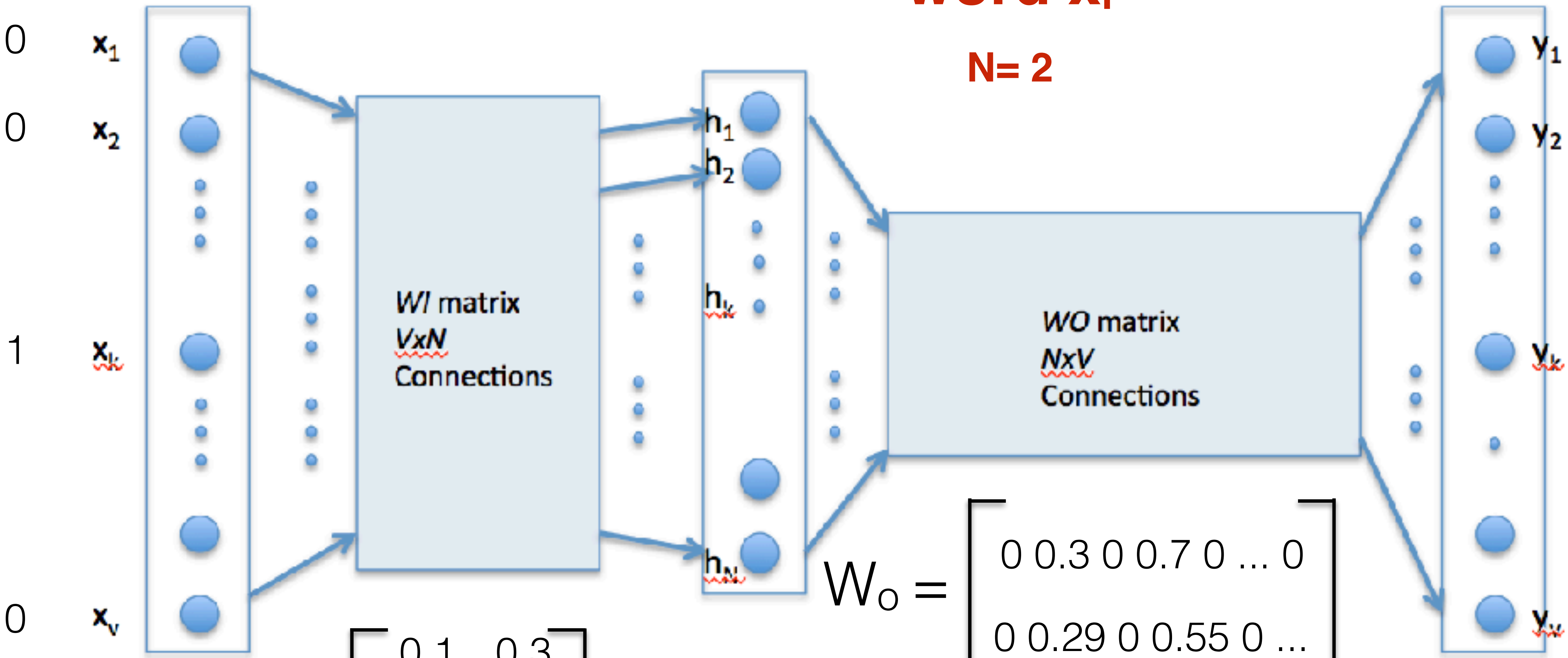
X.WI.WO

$$X.W_i = \begin{bmatrix} X_1W_1 & X_1W_i & \\ & X_lW_i & \\ & X_vW_i & X_vW_n \end{bmatrix}$$

Example

word x_i

N= 2



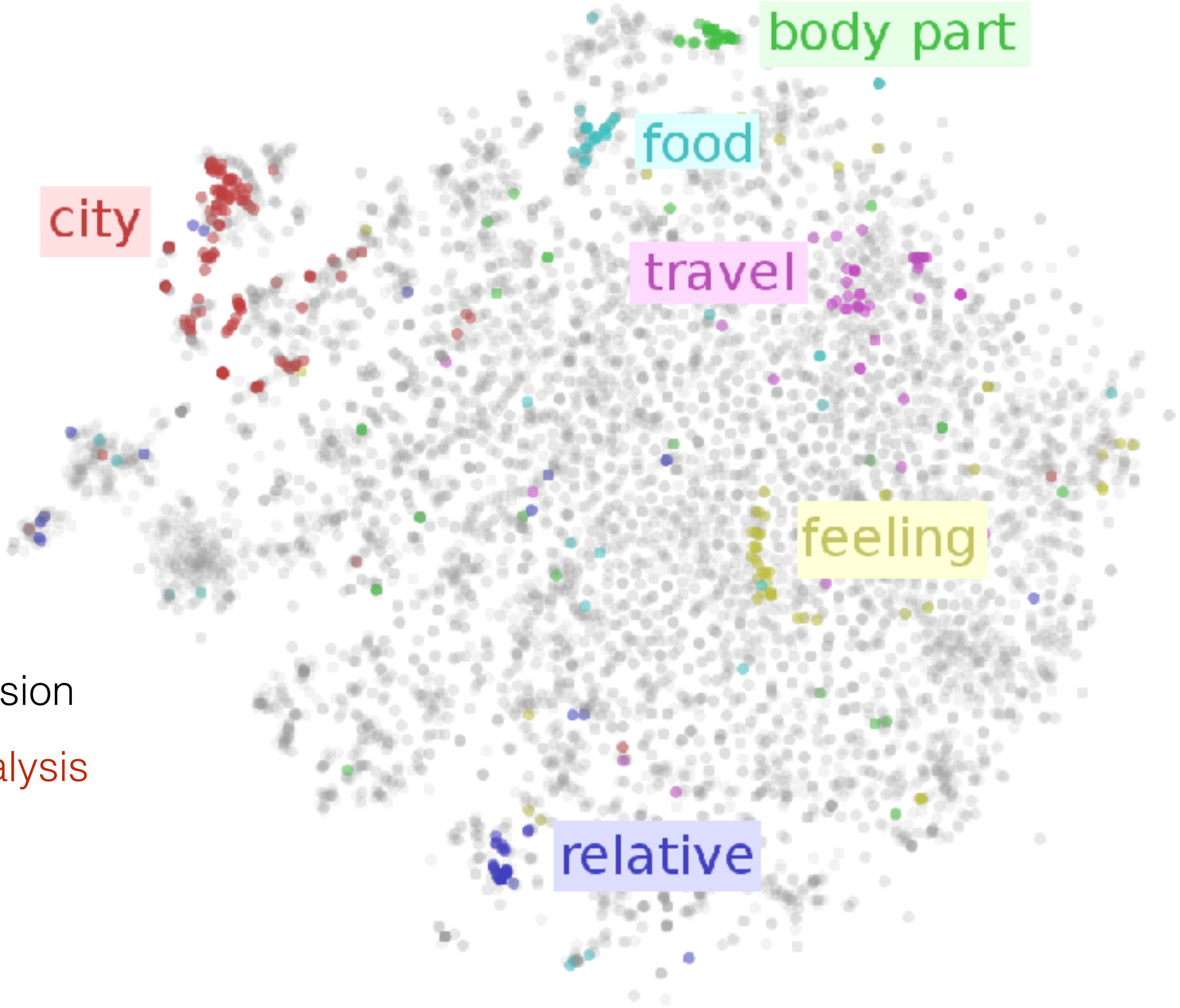
$$W_o = \begin{bmatrix} 0\ 0.3\ 0\ 0.7\ 0\ \ldots\ 0 \\ 0\ 0.29\ 0\ 0.55\ 0\ \ldots \\ \\ \end{bmatrix}$$

$$W_i = \begin{bmatrix} 0.1 & 0.3 \\ 0.77 & 0.5 \\ \vdots & \\ 0.39 & 0 \end{bmatrix}$$

Word embedding X_i = [0.33 ... 0.64]

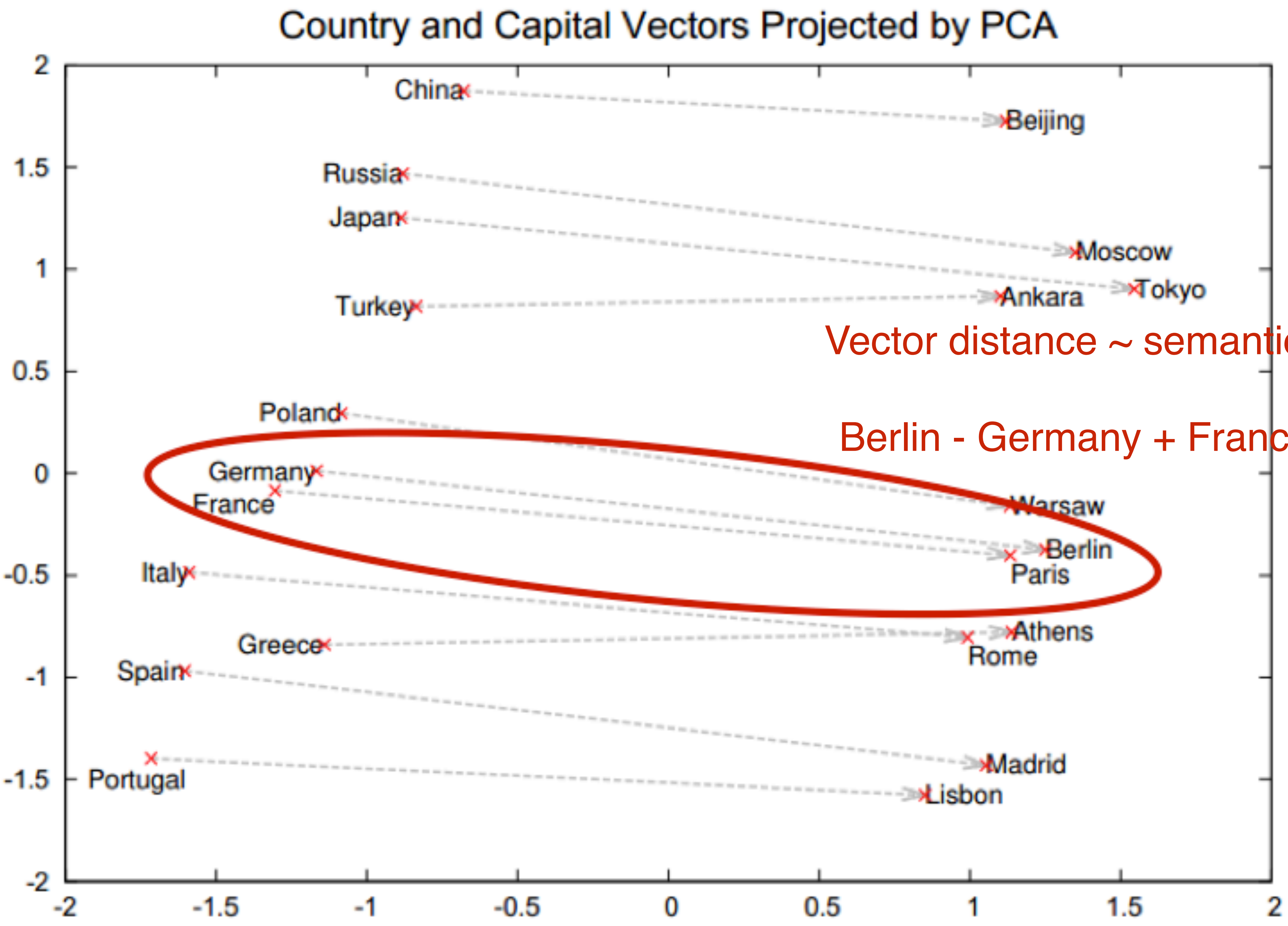
Word embedding = $X.\ W_i\ .\ W_o$

word2vec - 2d space



- Output matrix to 2 dimension
- Principal component Analysis

word2vec



Vector distance ~ semantic distance

Berlin - Germany + France = Paris

Picture taken from https://www.lucypark.kr/courses/2015-ba/text-mining.html

word2vec - Predicting next order

Kale&
Avocados&
Quinoa&
instacart
Groceries Delivered From Local Stores

Analyzing 3M Instacart Orders

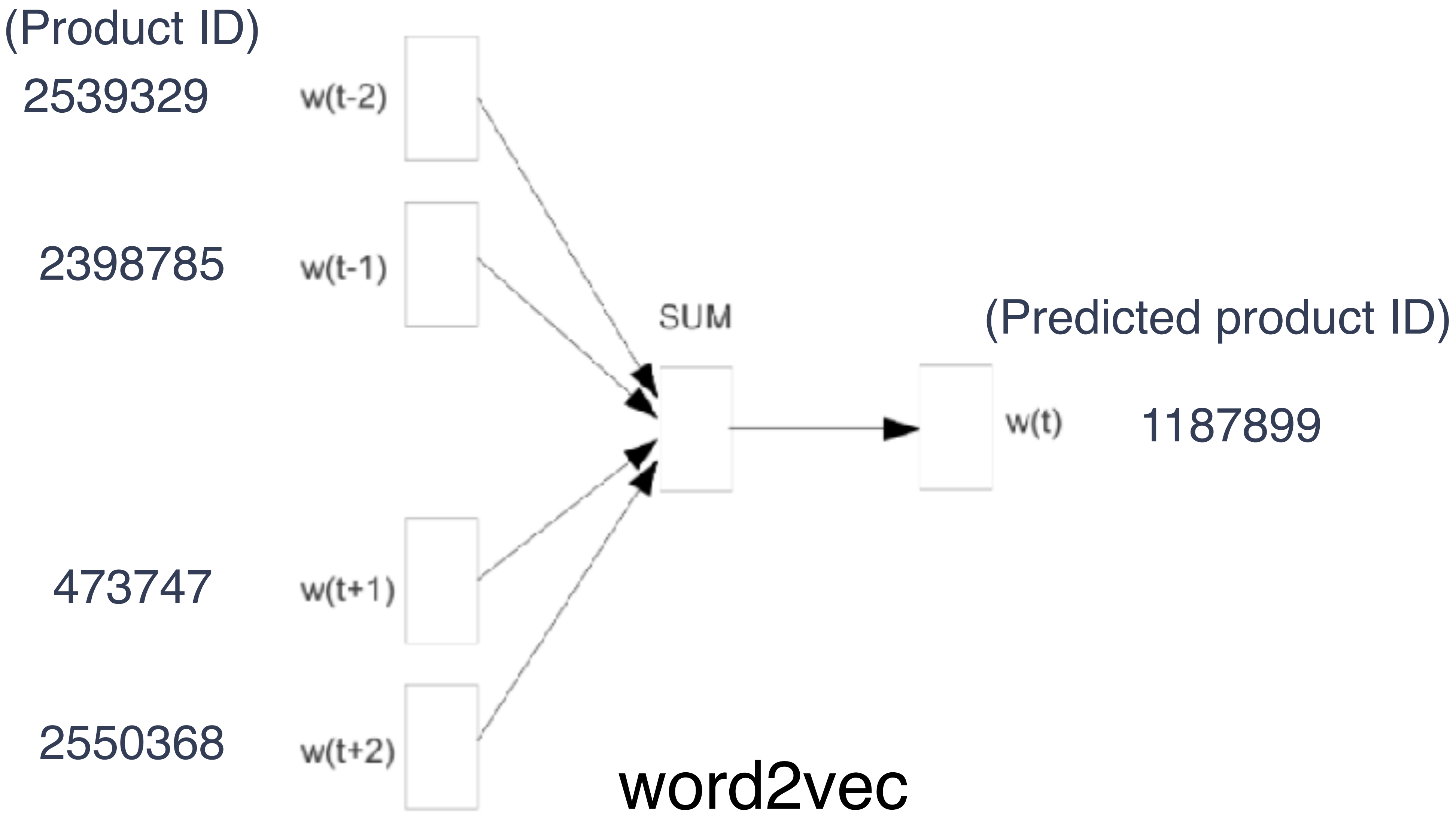
- **Prior: ~3.2m orders**
- **Train:~131k orders**
- **Test: ~75k orders**

Instacart Data

product_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
2539329	1	prior	1	2	8	7
2398795	1	prior	2	5	7	15
473747	1	prior	3	7	12	20
22544786	1	prior	4	1	7	21
4215438	1	prior	5	3	15	28
2295261	1	prior	6	2	7	19
2295261	1	prior	7	6	20	20
2550362	1	prior	8	5	14	14
1187899	1	prior	9	2	16	0
2168274	1	prior	10	2	8	30
1501582	1	train	11	1	11	10

Data taken from https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2

Current approach to prediction for instacart



(Product ID)

2539329

2398785

473747

2550368

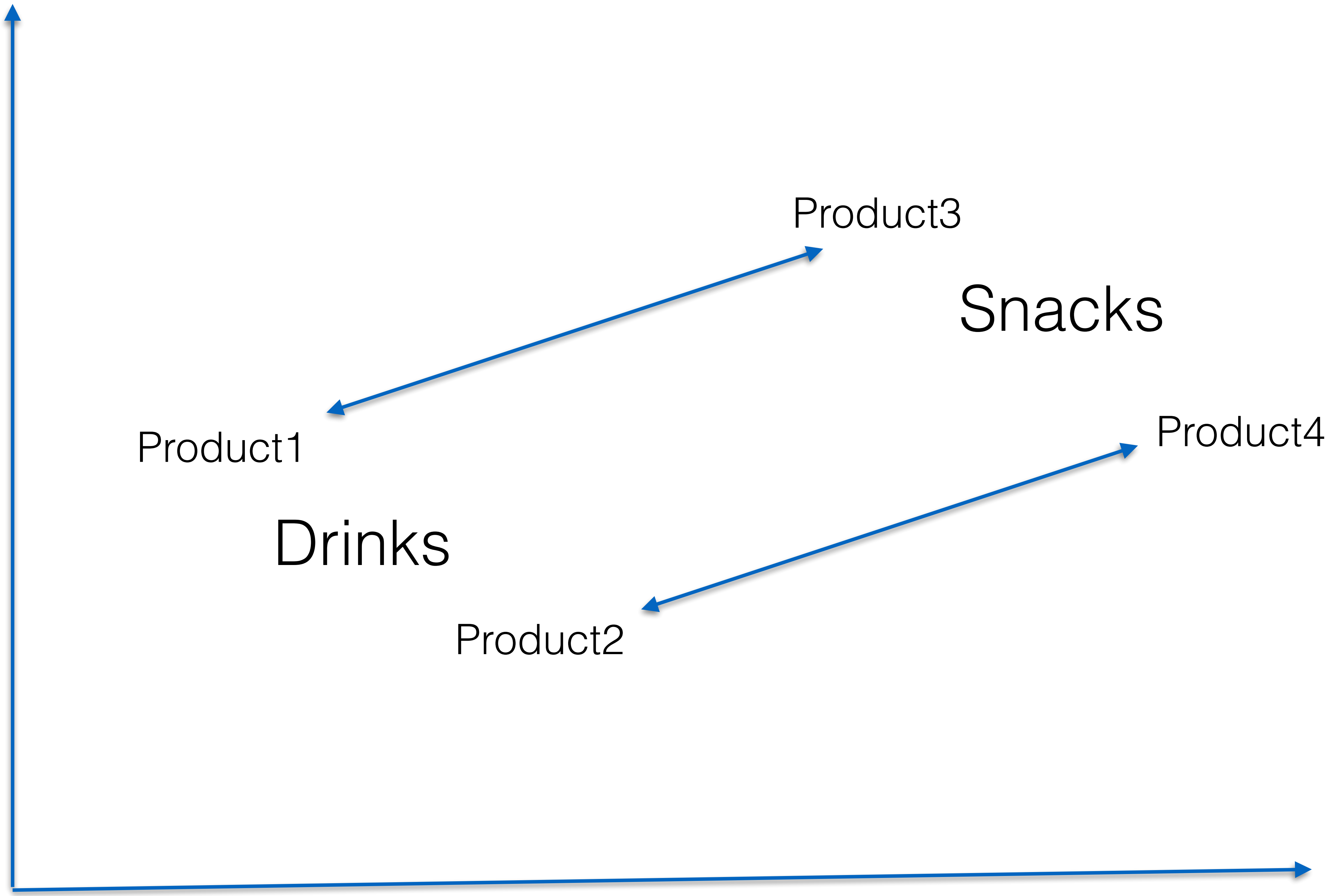
(Predicted product ID)

1187899

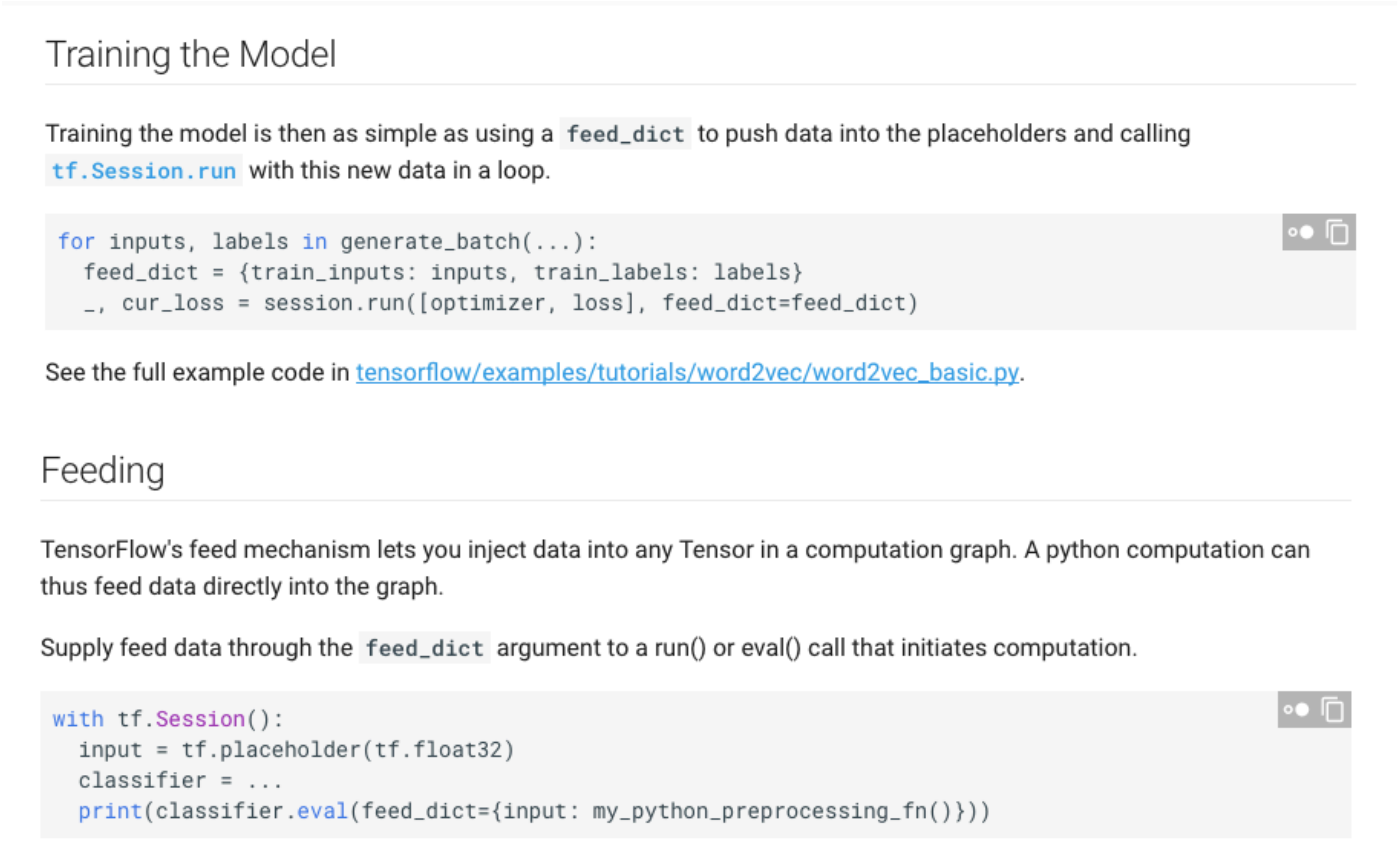
word2vec

- Using the semantic association between orders

Vector space of products



word2vec in tensorflow



Training the Model

Training the model is then as simple as using a `feed_dict` to push data into the placeholders and calling `tf.Session.run` with this new data in a loop.

```
for inputs, labels in generate_batch(...):
  feed_dict = {train_inputs: inputs, train_labels: labels}
  _, cur_loss = session.run([optimizer, loss], feed_dict=feed_dict)
```

See the full example code in tensorflow/examples/tutorials/word2vec/word2vec_basic.py.

Feeding

TensorFlow's feed mechanism lets you inject data into any Tensor in a computation graph. A python computation can thus feed data directly into the graph.

Supply feed data through the `feed_dict` argument to a run() or eval() call that initiates computation.

```
with tf.Session():
  input = tf.placeholder(tf.float32)
  classifier = ...
  print(classifier.eval(feed_dict={input: my_python_preprocessing_fn()}))
```

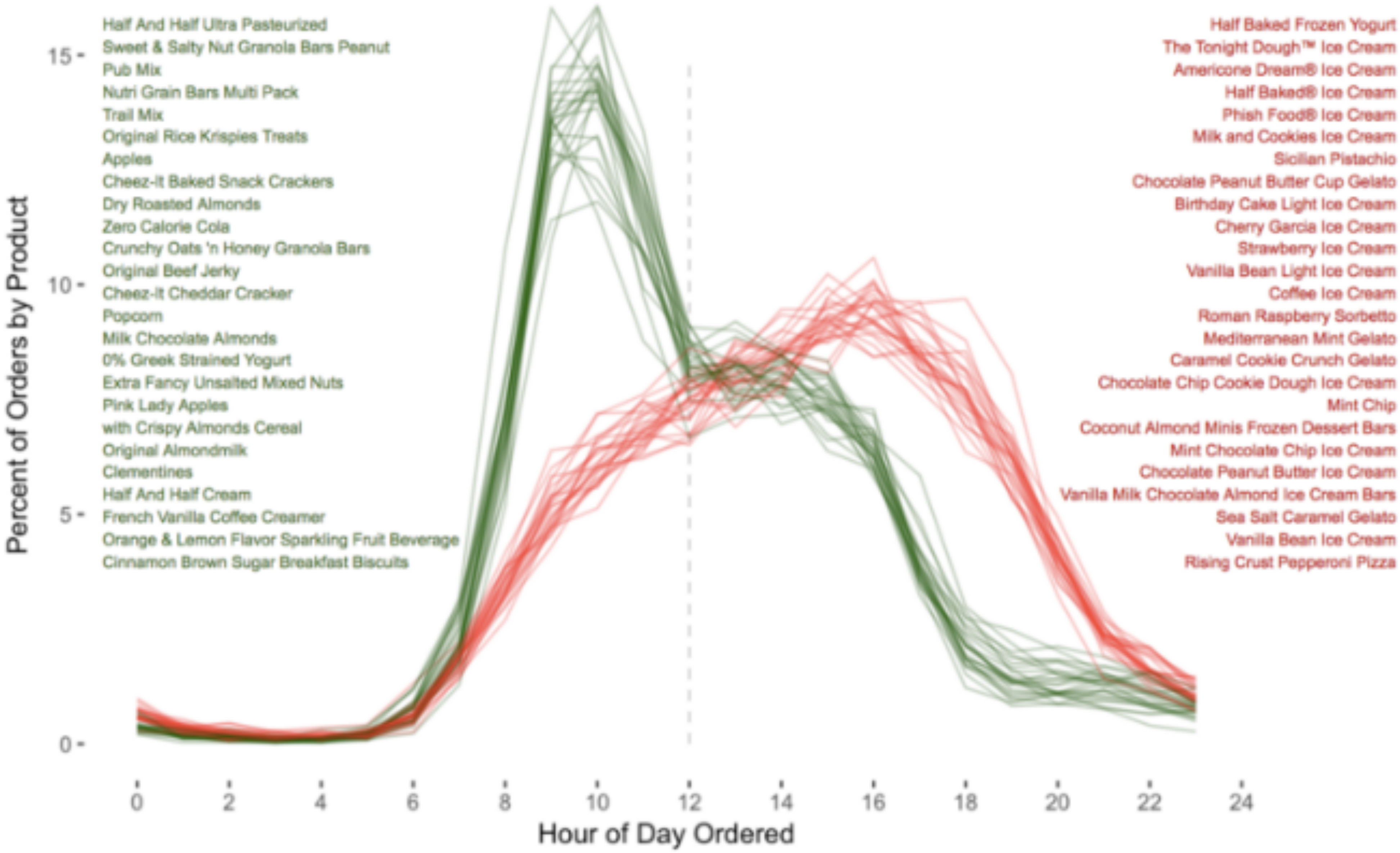
Use-case feature for using word2vec

- **Repeated patterns of human action**

Any other use case?

What is missing?

Sequence of order is important!



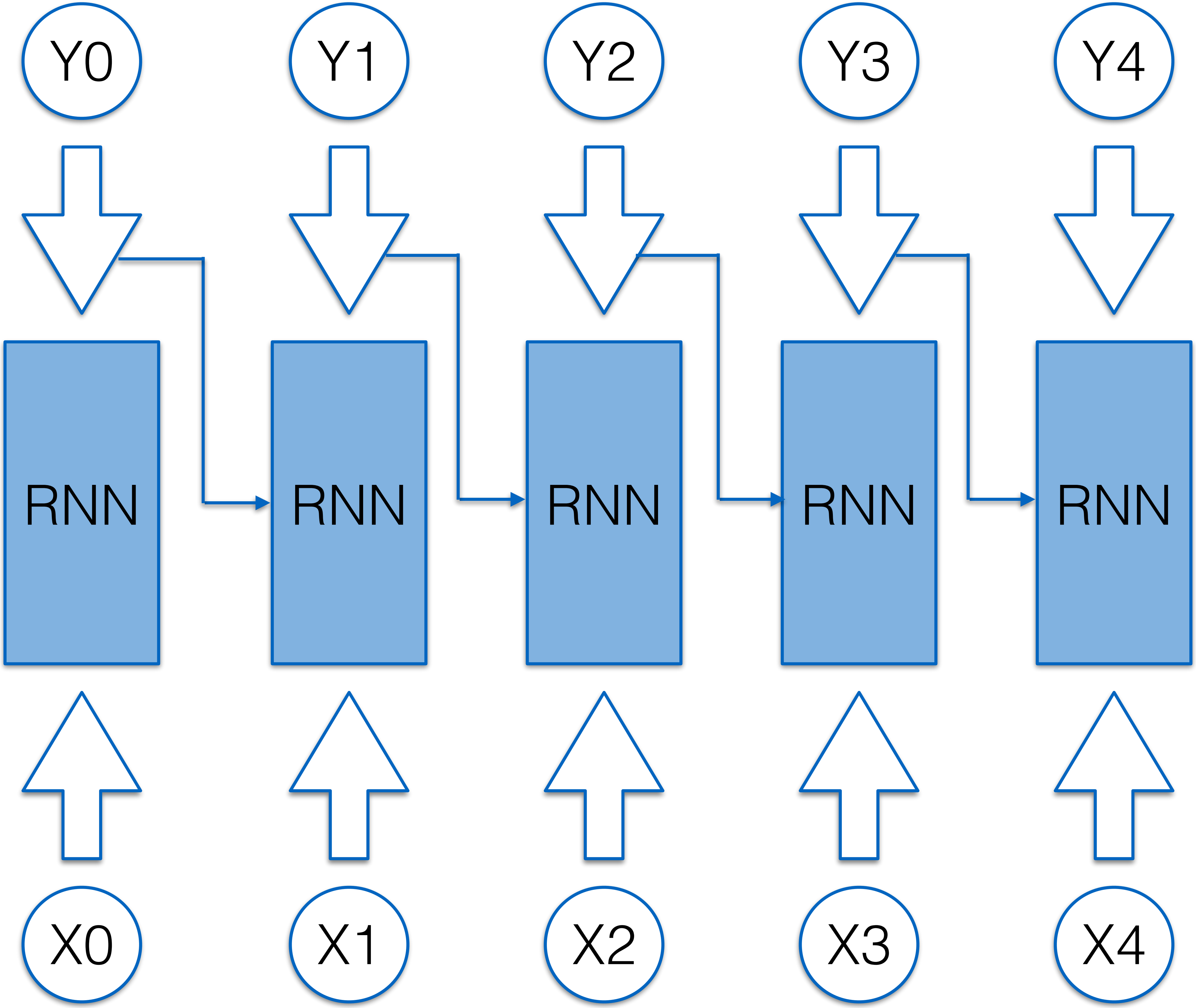
Popular products purchased earliest in the day (green) and latest in the day (red).

Data taken from https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2

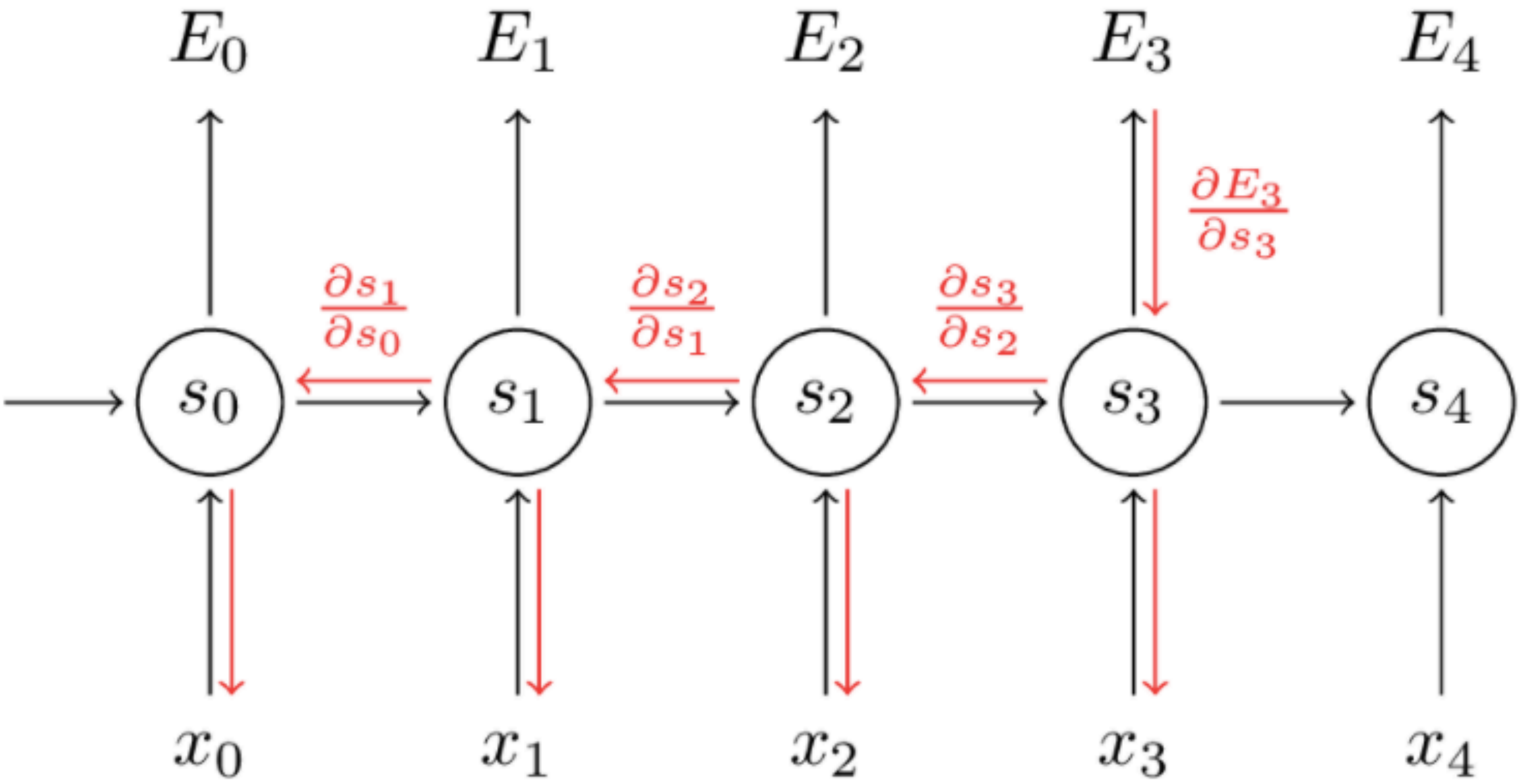
Instacart Data

product_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
2539329	1	prior	1	2	8	
2398795	1	prior	2	5	7	15
473747	1	prior	3	7	12	20
22544786	1	prior	4	1	7	21
4215438	1	prior	5	3	15	28
2295261	1	prior	6	2	7	19
2295261	1	prior	7	6	20	20
2550362	1	prior	8	5	14	14
1187899	1	prior	9	2	16	0
2168274	1	prior	10	2	8	30
1501582	1	train	11	1	11	10

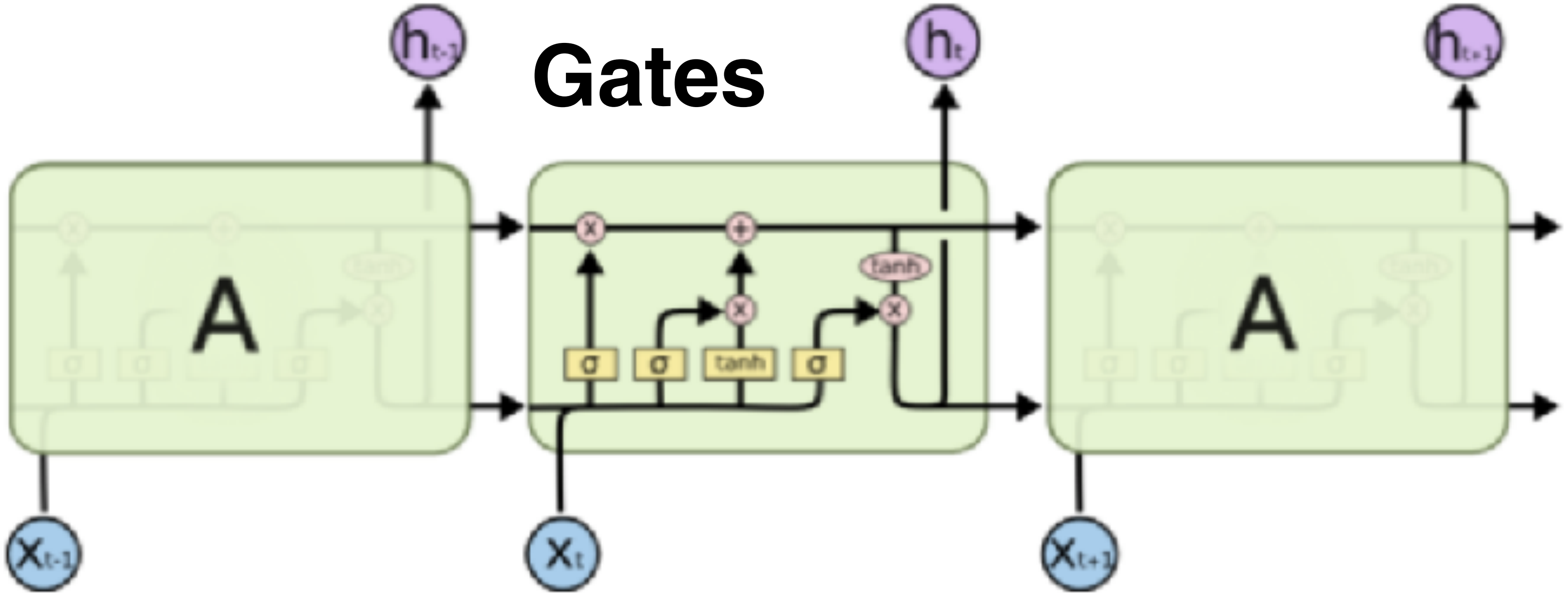
Data taken from https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2


Y0
Y1
Y2
Y3
Y4
RNN
RNN
RNN
RNN
RNN
X0
X1
X2
X3
X4

Vanishing gradient problem


E_0
E_1
E_2
E_3
E_4
$\frac{\partial E_3}{\partial s_3}$
$\frac{\partial s_1}{\partial s_0}$
$\frac{\partial s_2}{\partial s_1}$
$\frac{\partial s_3}{\partial s_2}$
s_0
s_1
s_2
s_3
s_4
x_0
x_1
x_2
x_3
x_4

Long Short Term Memory

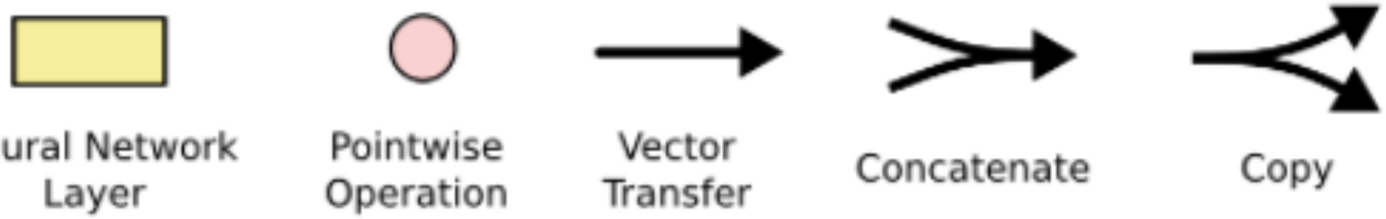


Gates

Time 1 Time 2 Time 3

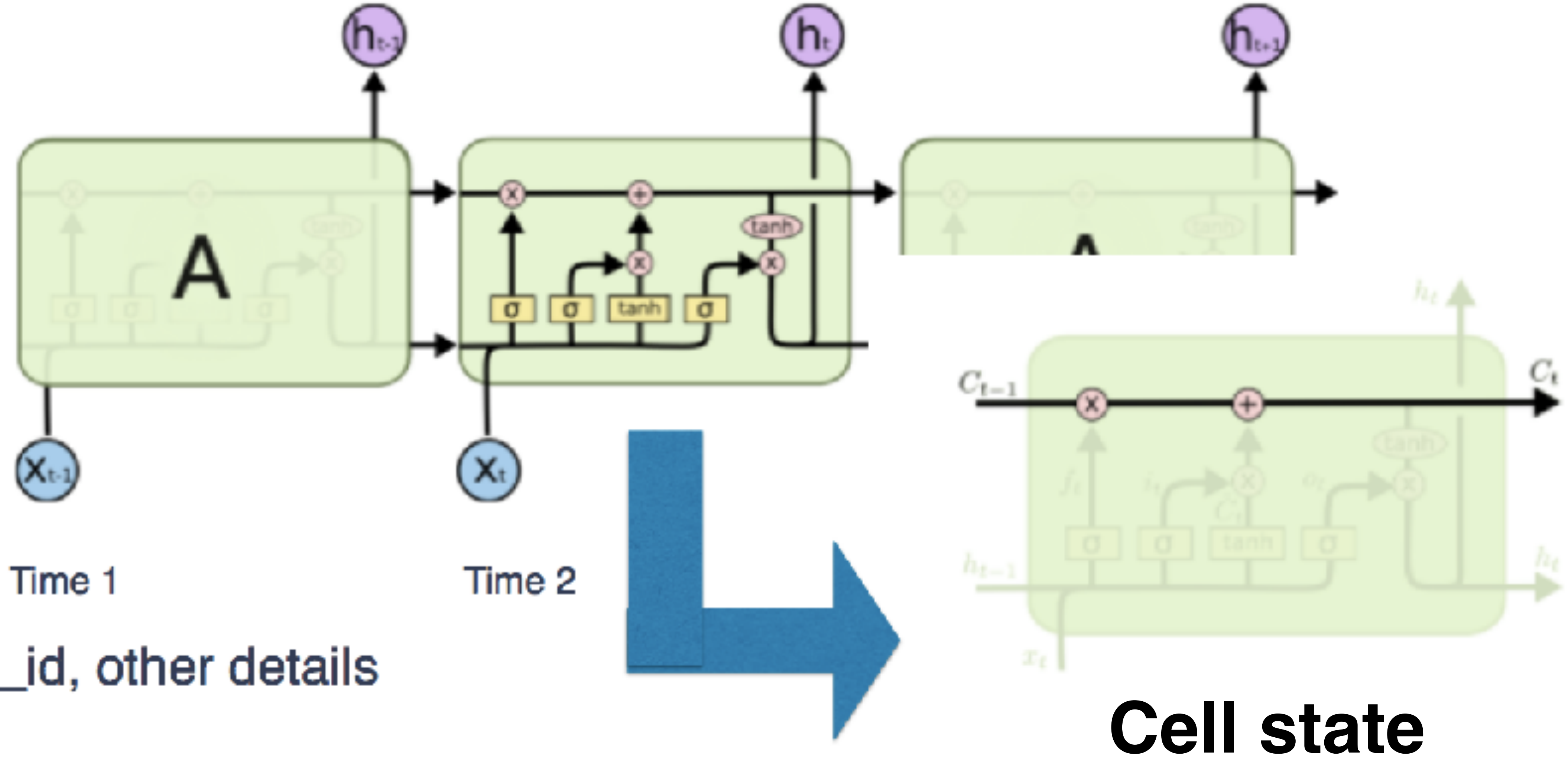
time, order_id, other details

time, order_id, other details



Picture taken from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory



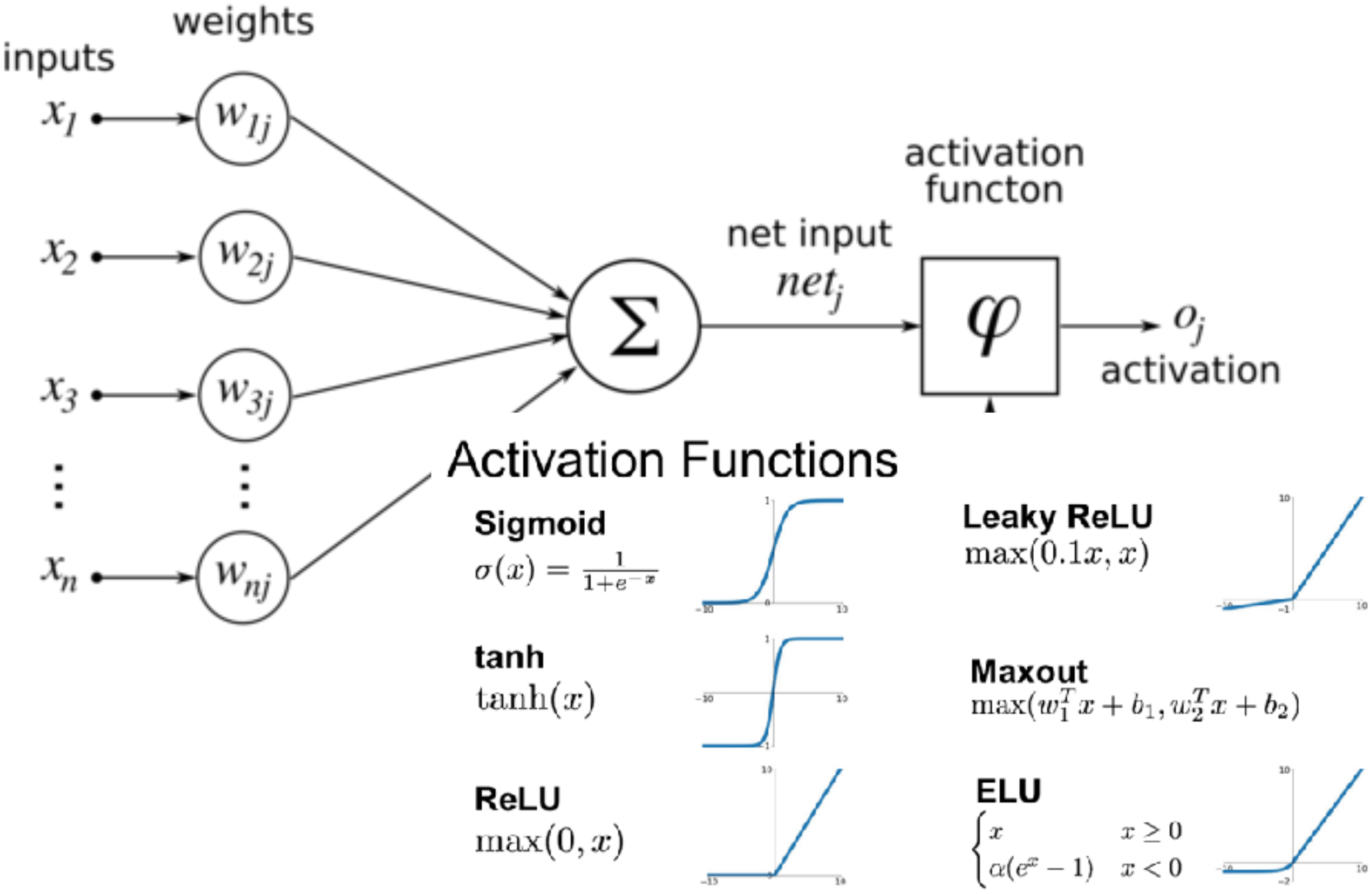
time, order_id, other details

Cell state

Forget, Update and Output

Picture taken from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Activation function



Activation Functions

Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$

tanh
$\tanh(x)$

ReLU
$\max(0, x)$

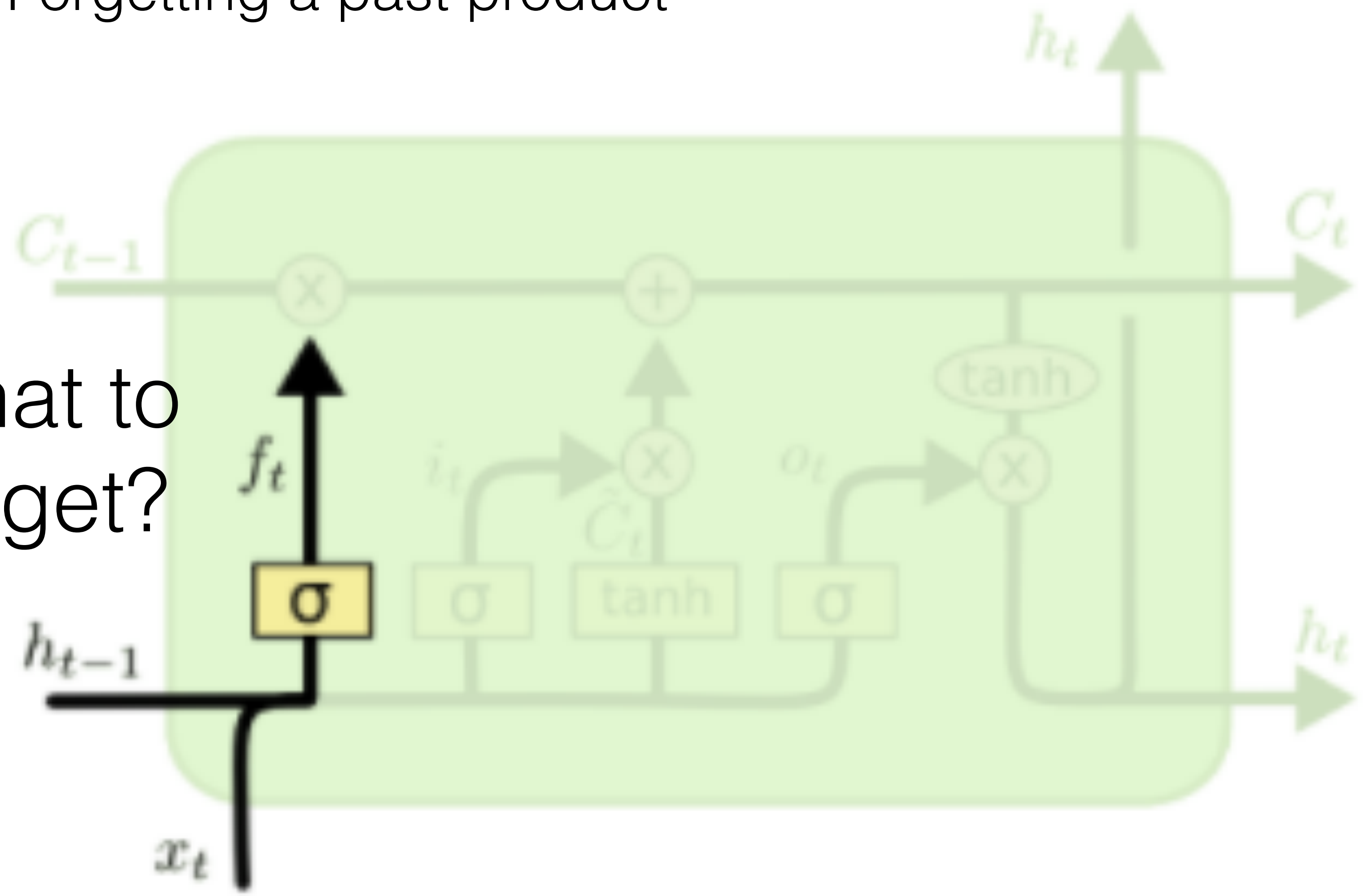
Leaky ReLU
$\max(0.1x, x)$

Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

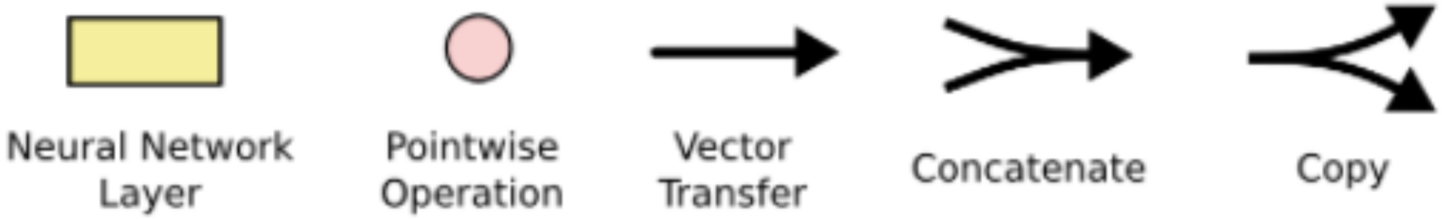
ELU
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

Long Short Term Memory cell - Forget gate

Forgetting a past product



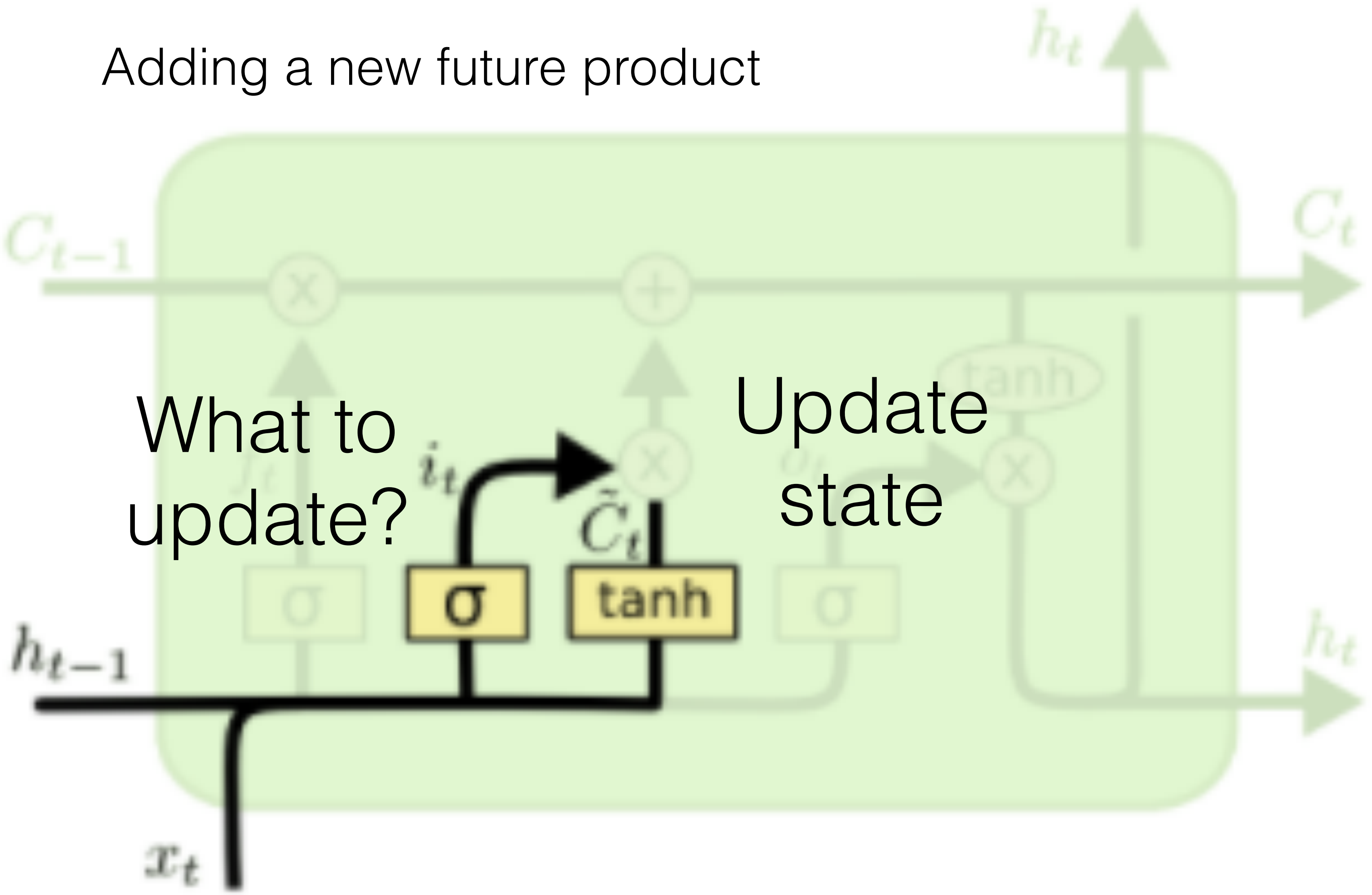
What to forget?



Picture taken from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

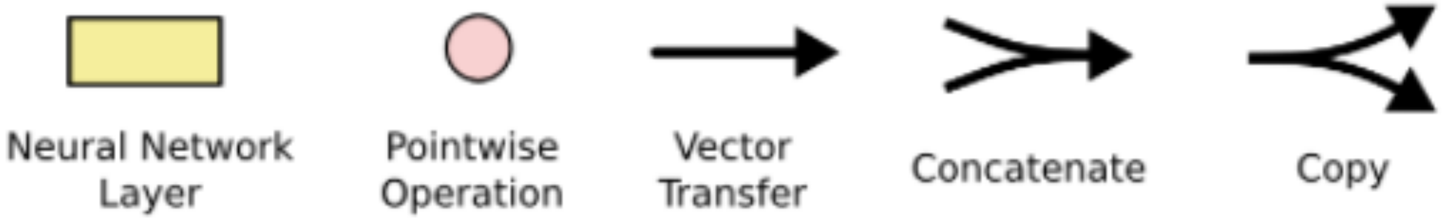
Long Short Term Memory cell - Input gate

Adding a new future product



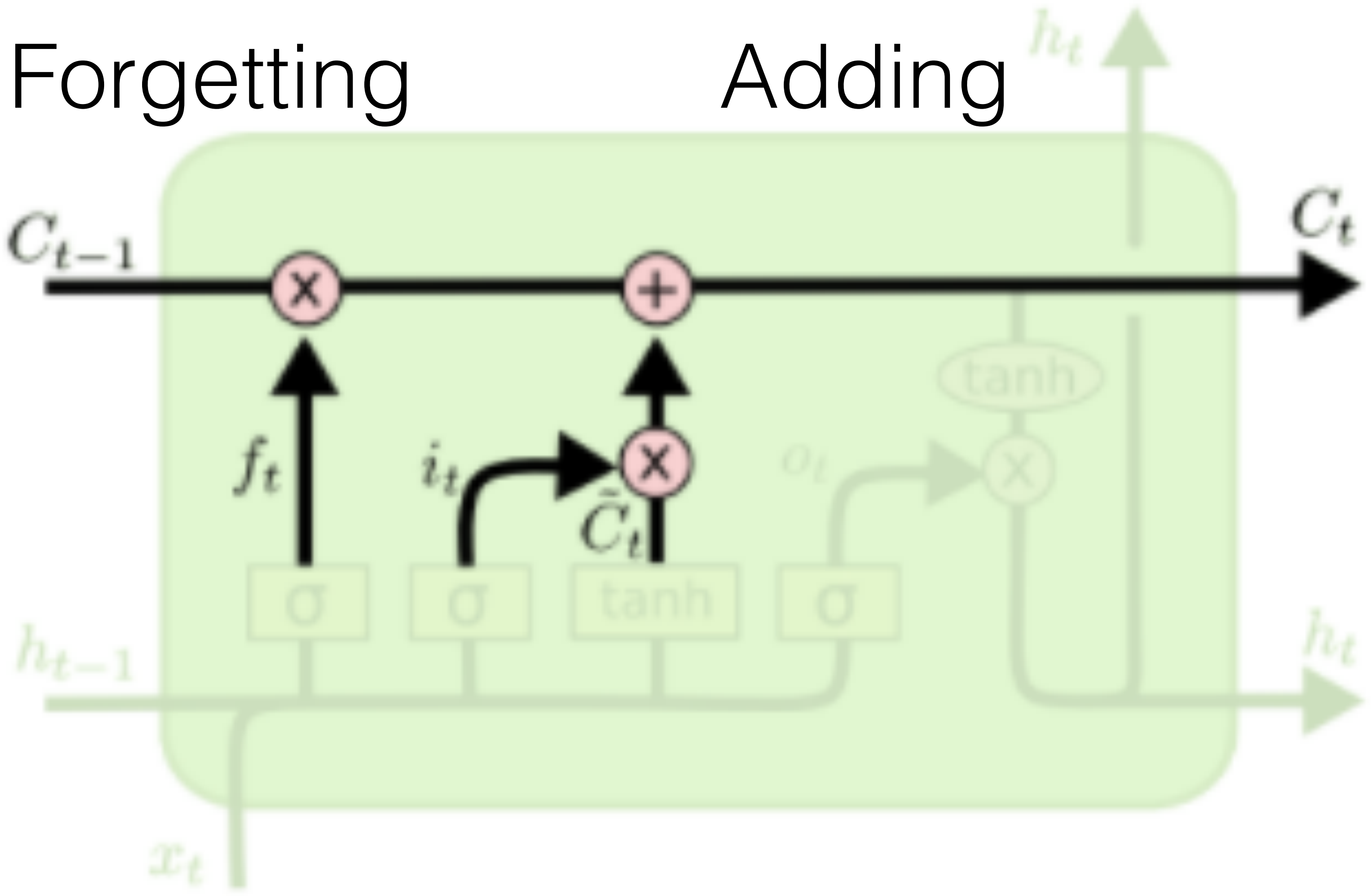
What to update?

Update state



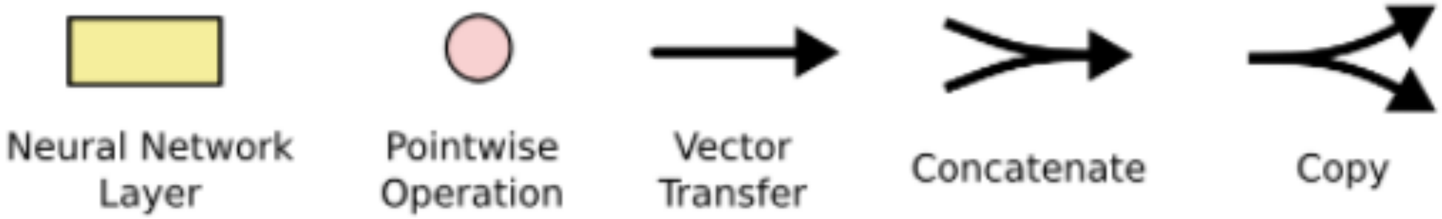
Picture taken from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory cell - New internal state



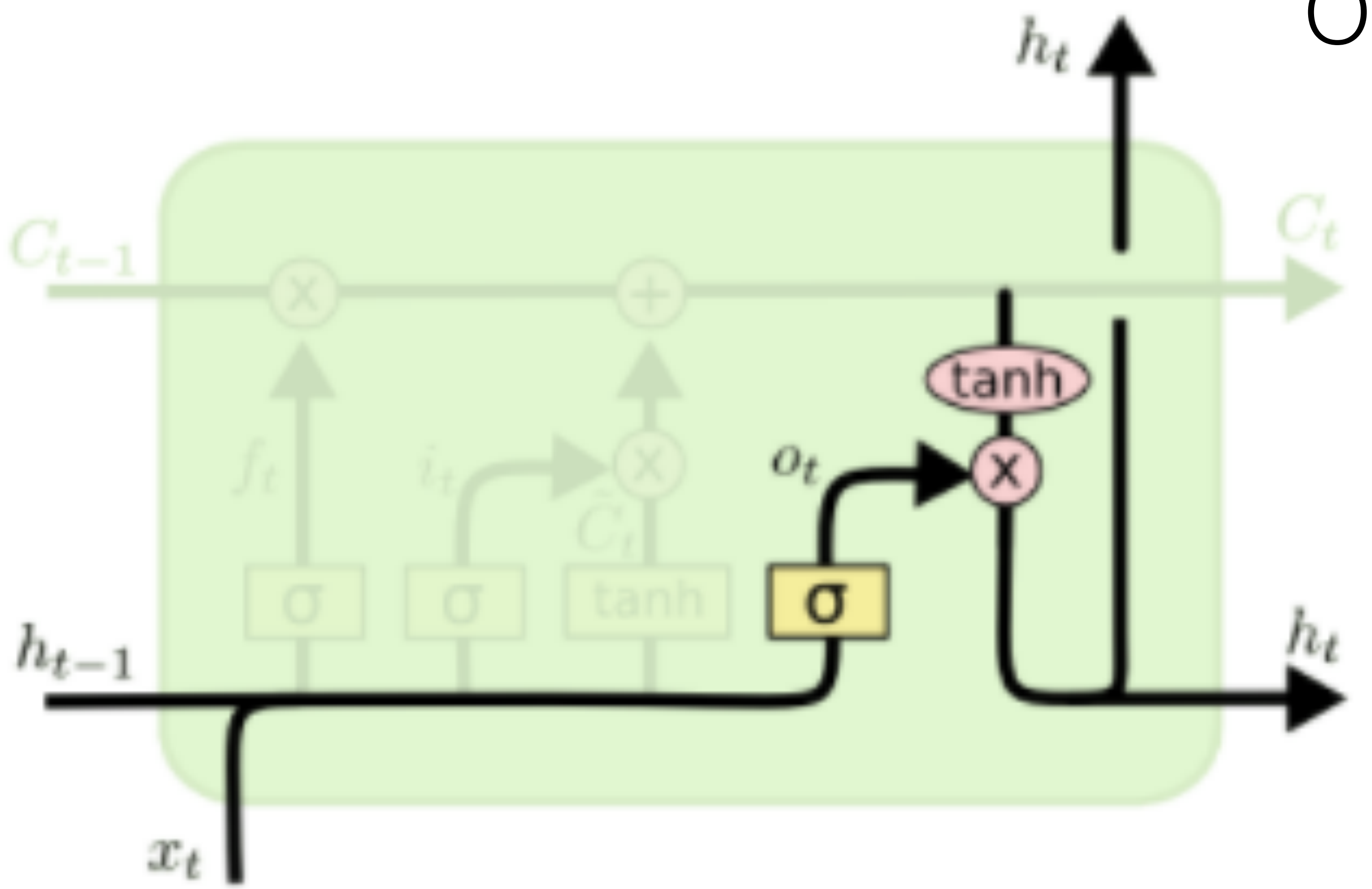
Forgetting

Adding



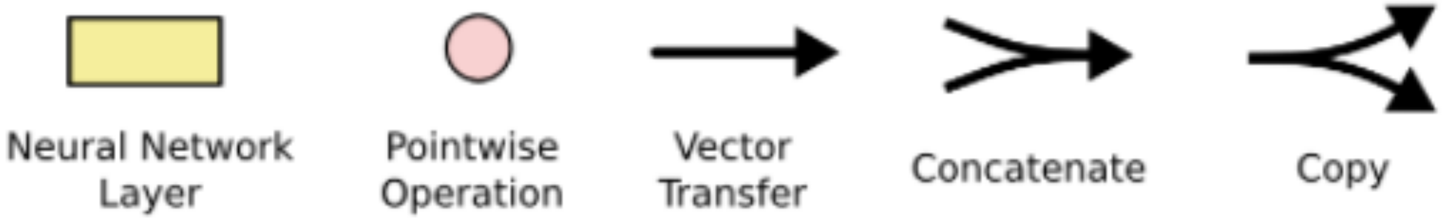
Picture taken from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory cell - Output gate

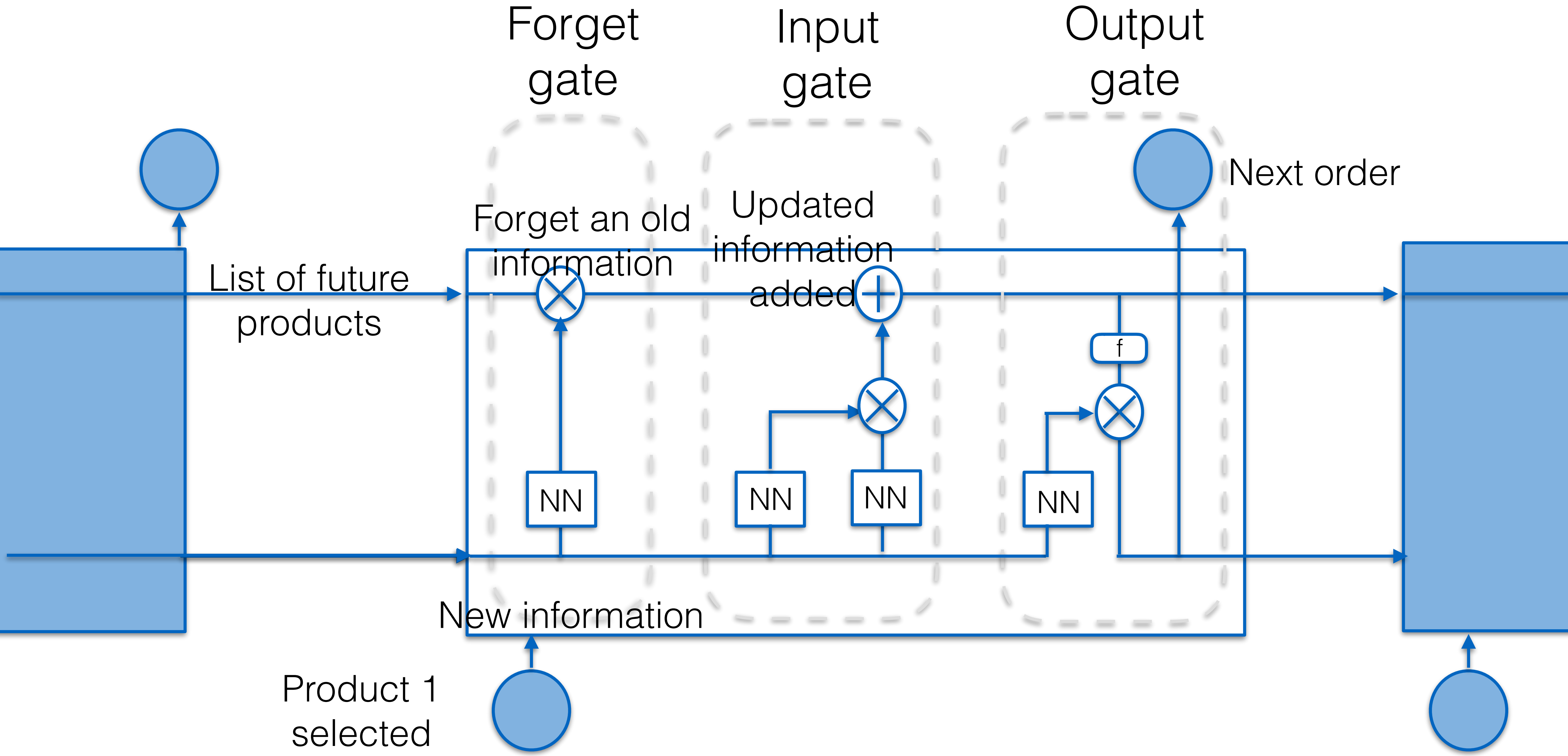


Next product to order

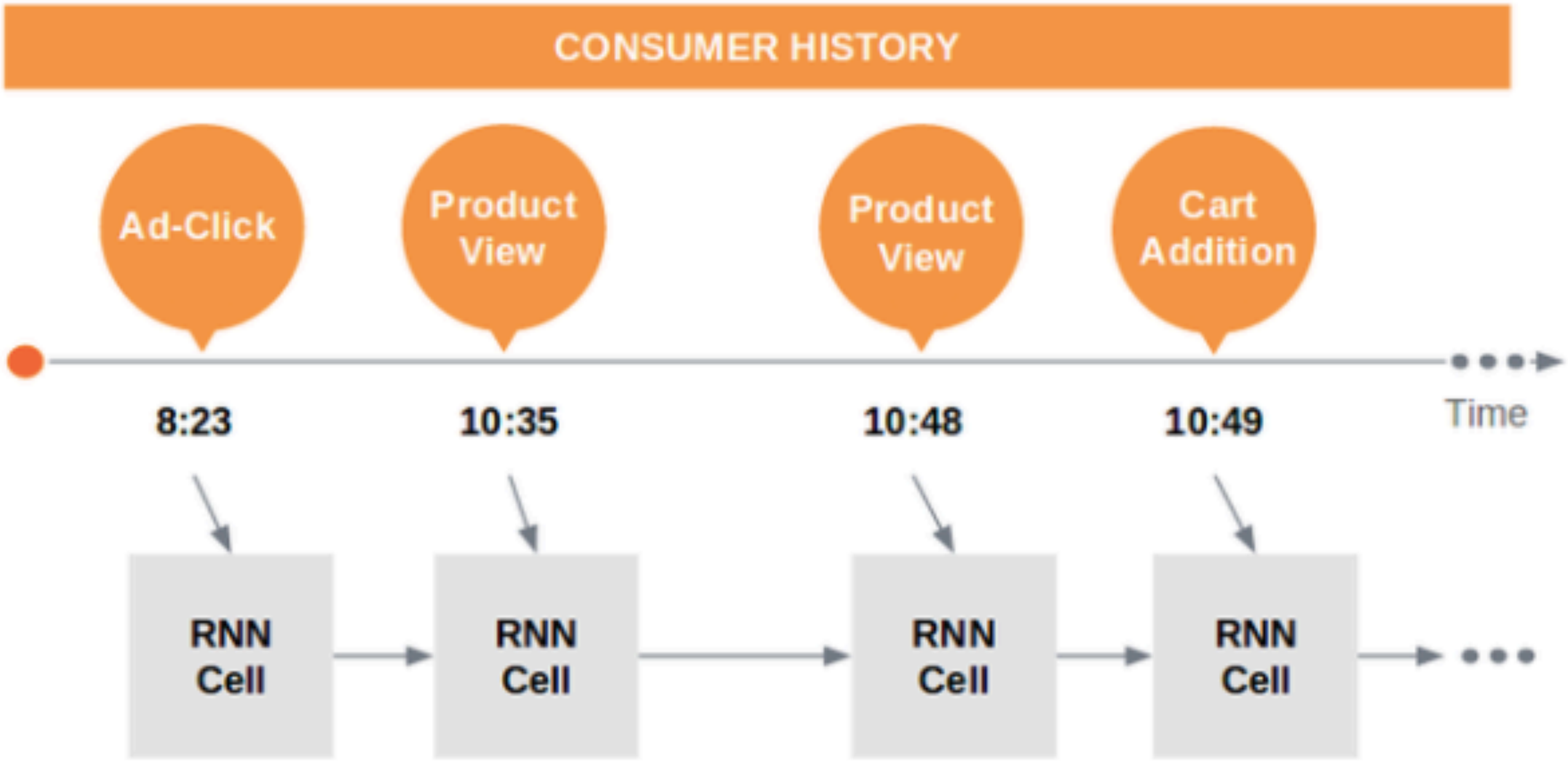
Output



Picture taken from http://colah.github.io/posts/2015-08-Understanding-LSTMs/


Forget gate
Input gate
Output gate
Next order
Forget an old information
Updated information added
List of future products
f
NN
NN
NN
NN
New information
Product 1 selected

Zolando’s approach to Prediction

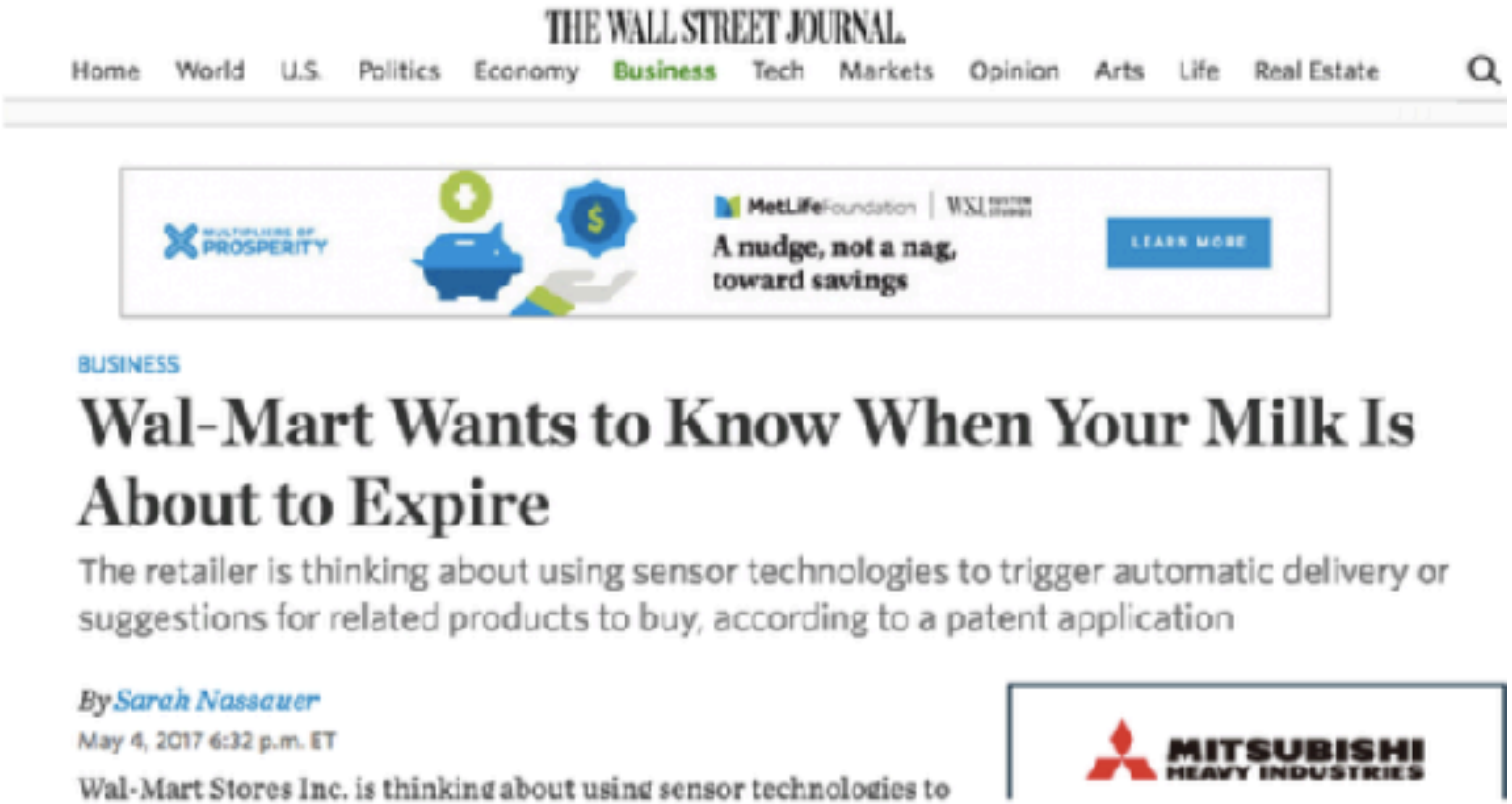


Picture taken from https://jobs.zalando.com/tech/blog/deep-learning-for-understanding-consumer-histories/?gh_src=4n3gxh1

Future Products

Amazon Knows What You Want Before You Buy It

4.7k SHARES



THE WALL STREET JOURNAL.
Home World U.S. Politics Economy Business Tech Markets Opinion Arts Life Real Estate

BUSINESS

Wal-Mart Wants to Know When Your Milk Is About to Expire

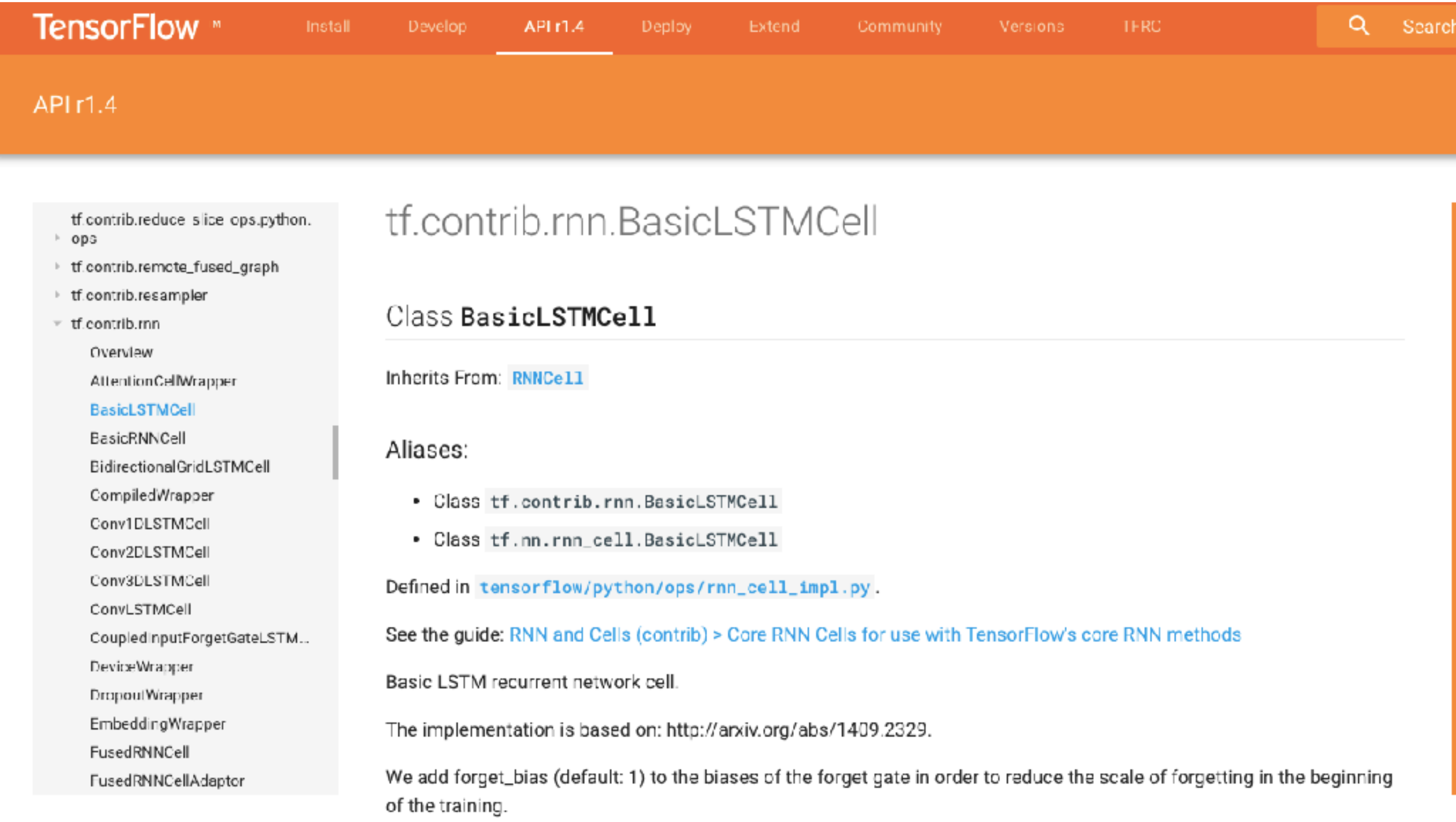
The retailer is thinking about using sensor technologies to trigger automatic delivery or suggestions for related products to buy, according to a patent application

By Sarah Nassauer

May 4, 2017 6:32 p.m. ET

Wal-Mart Stores Inc. is thinking about using sensor technologies to

Tensorflow LSTM API



tf.contrib.rnn.BasicLSTMCell

Class `BasicLSTMCell`

Inherits From: `RNNCell`

Aliases:

- Class `tf.contrib.rnn.BasicLSTMCell`
- Class `tf.nn.rnn_cell.BasicLSTMCell`

Defined in `tensorflow/python/ops/rnn_cell_impl.py`.

See the guide: RNN and Cells (contrib) > Core RNN Cells for use with TensorFlow's core RNN methods

Basic LSTM recurrent network cell.

The implementation is based on: http://arxiv.org/abs/1409.2329.

We add forget_bias (default: 1) to the biases of the forget gate in order to reduce the scale of forgetting in the beginning of the training.

Use-case feature for using LSTM

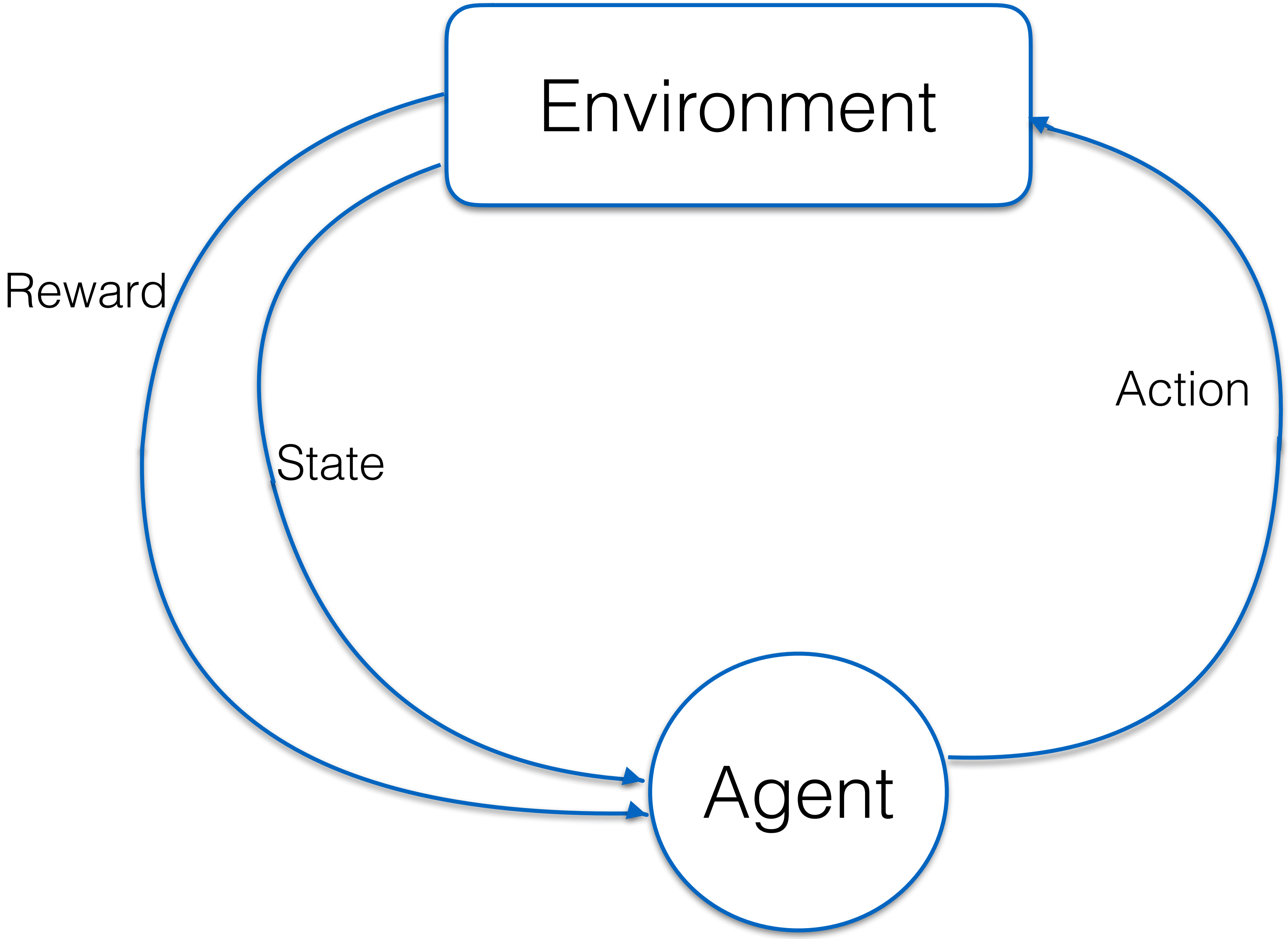
- **Repeated patterns of human action done over longer duration**

Any other use case?

What is missing?

Training Data

Reinforcement learning



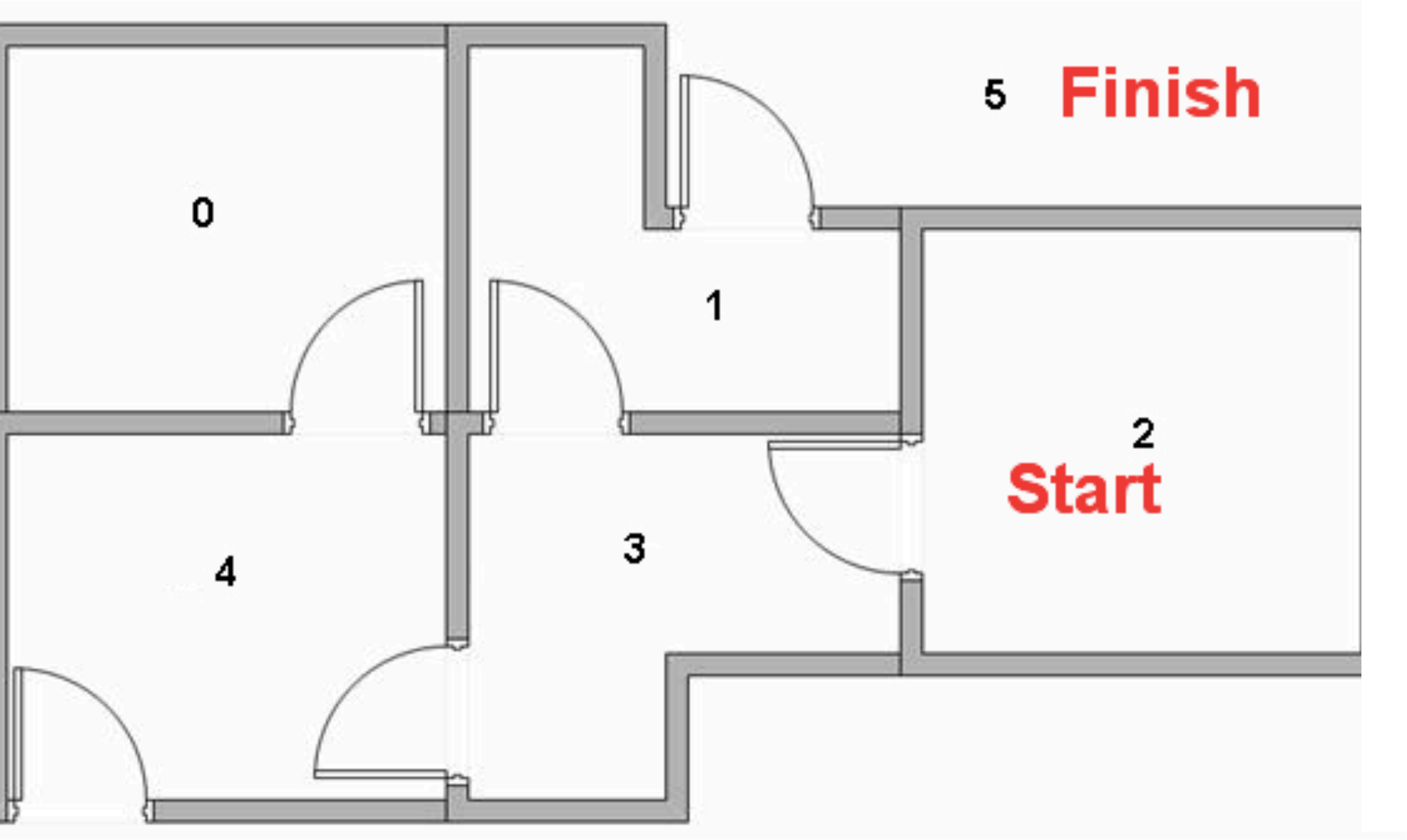
Q-learning

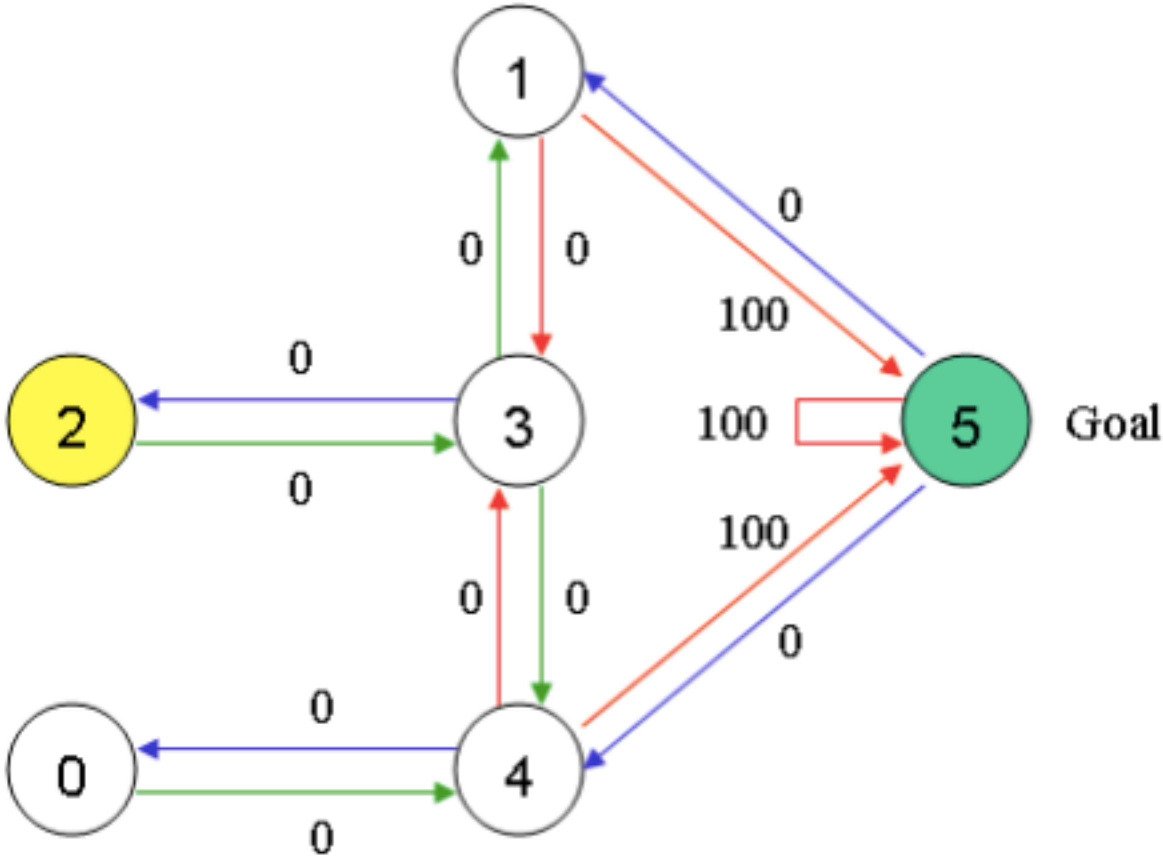
- Reward function for each action in a given state

- *Q(s,a)=R(s,a) + γ * max [Q(next states, all actions)]*

- Initial Q is 0

- Value of γ is (0,1) depends how much you want future actions to influence current learning.





$$Q(s,a)=R(s,a) + \gamma * \max [Q(\text{next states, all actions})]$$

$$\begin{array}{c} \\ \\ R= \end{array}
\begin{array}{c|cccccc}
 & & & \text{Action} & & & \\
\text{State} & 0 & 1 & 2 & 3 & 4 & 5 \\
\hline
0 & -1 & -1 & -1 & -1 & 0 & -1 \\
1 & -1 & -1 & -1 & 0 & -1 & 100 \\
2 & -1 & -1 & -1 & 0 & -1 & -1 \\
3 & -1 & 0 & 0 & -1 & 0 & -1 \\
4 & 0 & -1 & -1 & 0 & -1 & 100 \\
5 & -1 & 0 & -1 & -1 & 0 & 100
\end{array}$$

If $\gamma = 0.8$

$Q(s,a)=R(s,a) +$

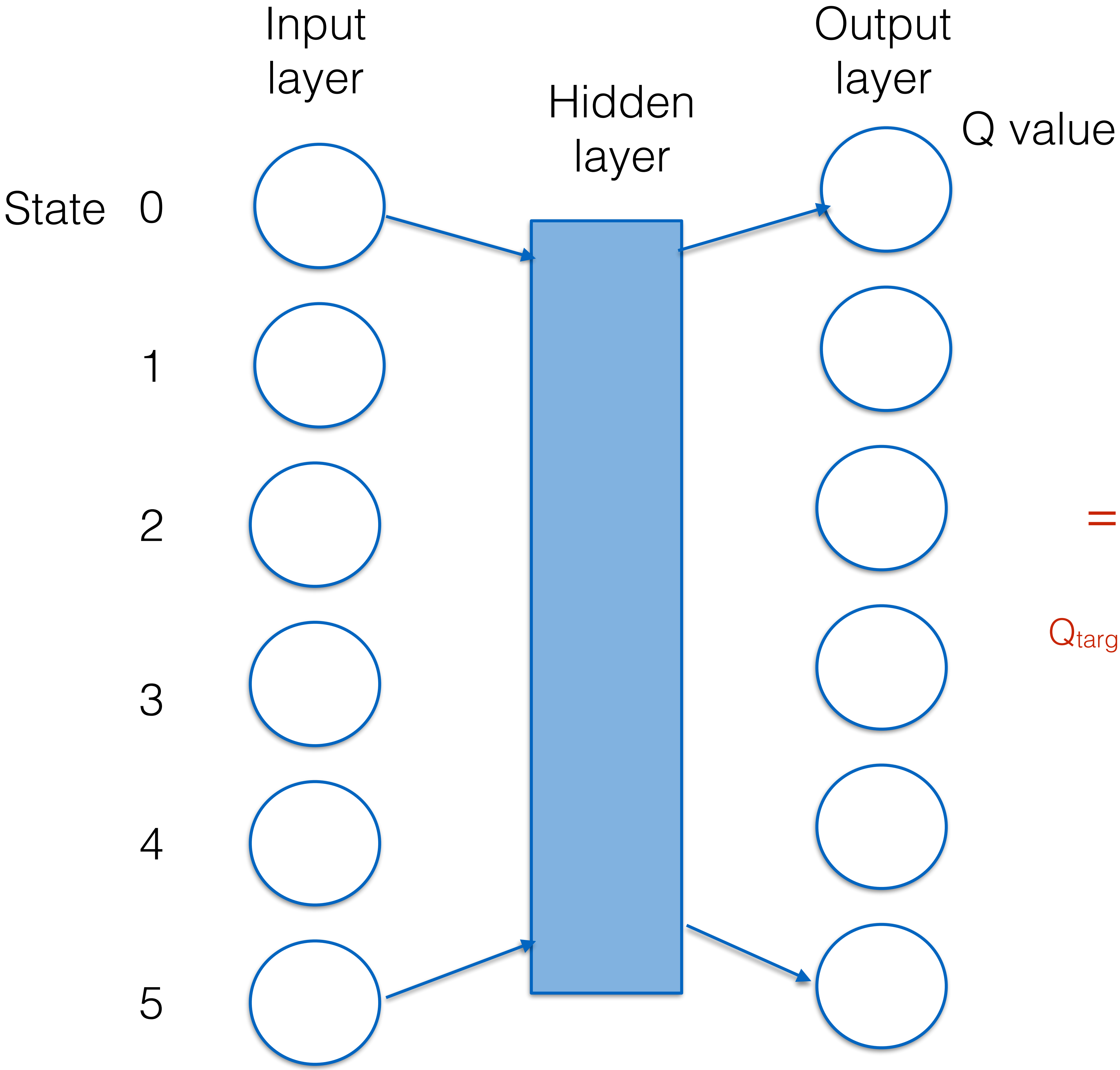
$0.8 * \max [Q(\text{next states, actions})]$

eg

$Q(3,1)=0 + 0.8 [\max(Q(1,3),Q(1,5))] = 400$

$$Q= \begin{array}{c|cccccc}
 & 0 & 1 & 2 & 3 & 4 & 5 \\
\hline
0 & 0 & 0 & 0 & 0 & 400 & 0 \\
1 & 0 & 0 & 0 & 320 & 0 & \boxed{500} \\
2 & 0 & 0 & 0 & \boxed{320} & 0 & 0 \\
3 & 0 & \boxed{400} & 256 & 0 & 400 & 0 \\
4 & 320 & 0 & 0 & 320 & 0 & 500 \\
5 & 0 & 400 & 0 & 0 & 400 & 500
\end{array}$$

Example taken from
http://mnemstudio.org/path-finding-q-learning-tutorial.htm



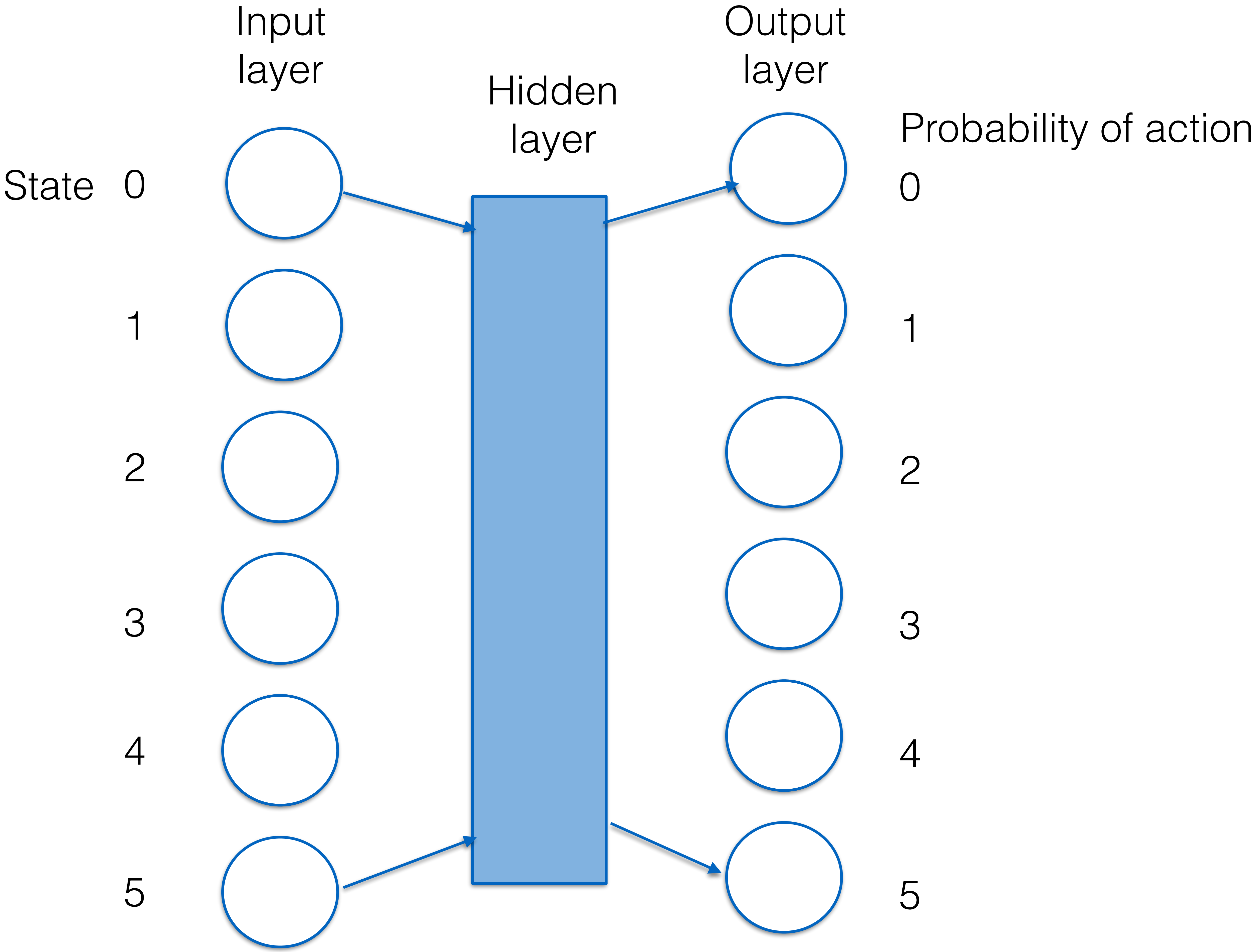
Loss

$= \Sigma\ (Q_{targeti} - Q_i)$

$Q_{targeti}=r + \gamma * \max(Q(s',a'))$

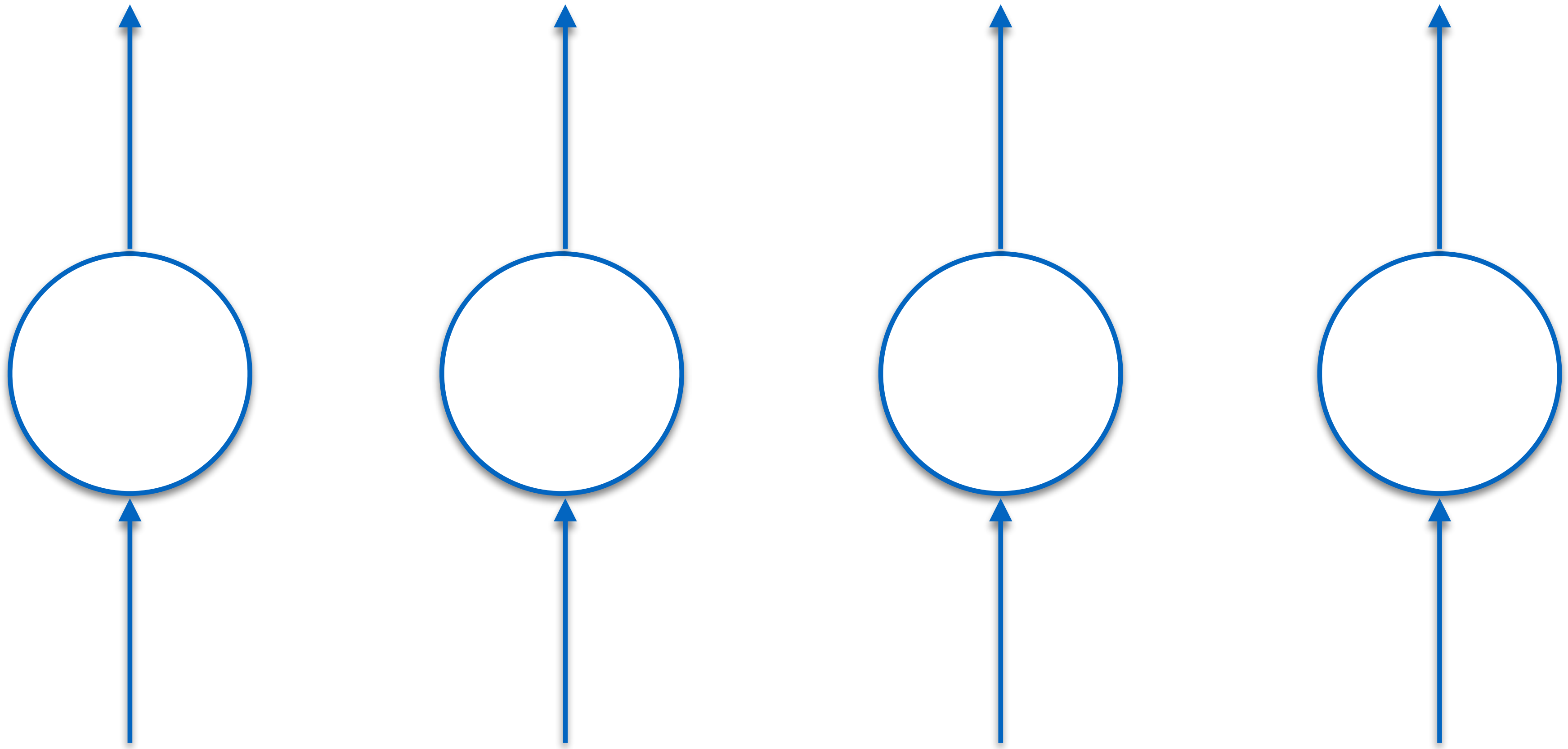
Policy Gradient

- Optimizes the policy space so the Neural network directly models the action space
- More complex problems esp. continuous action space

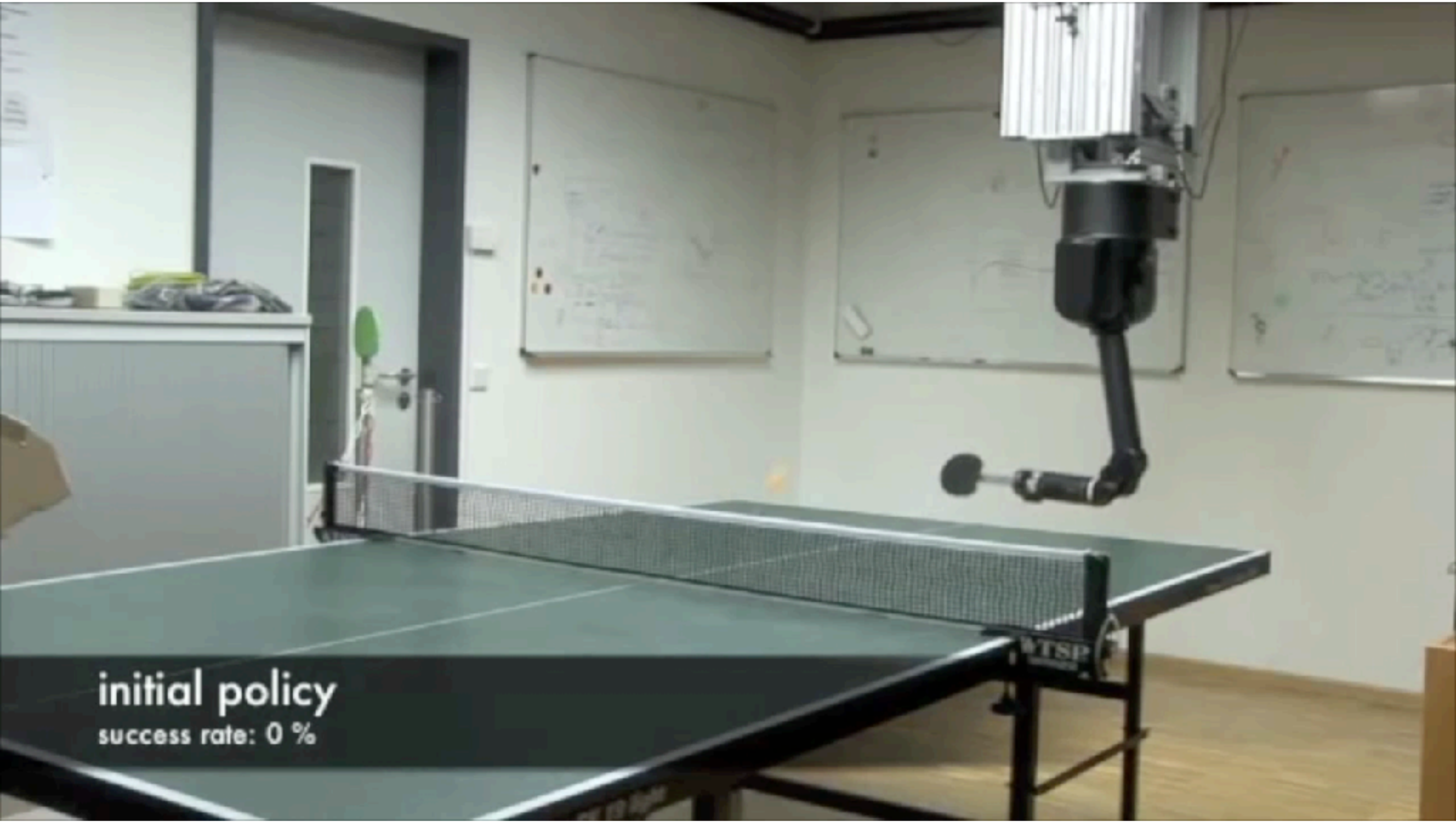

Input layer
Hidden layer
Output layer
Probability of action
State 0
1
2
3
4
5
0
1
2
3
4
5

Predicting next order

Action Product 5



State Product 1 Product 3



initial policy

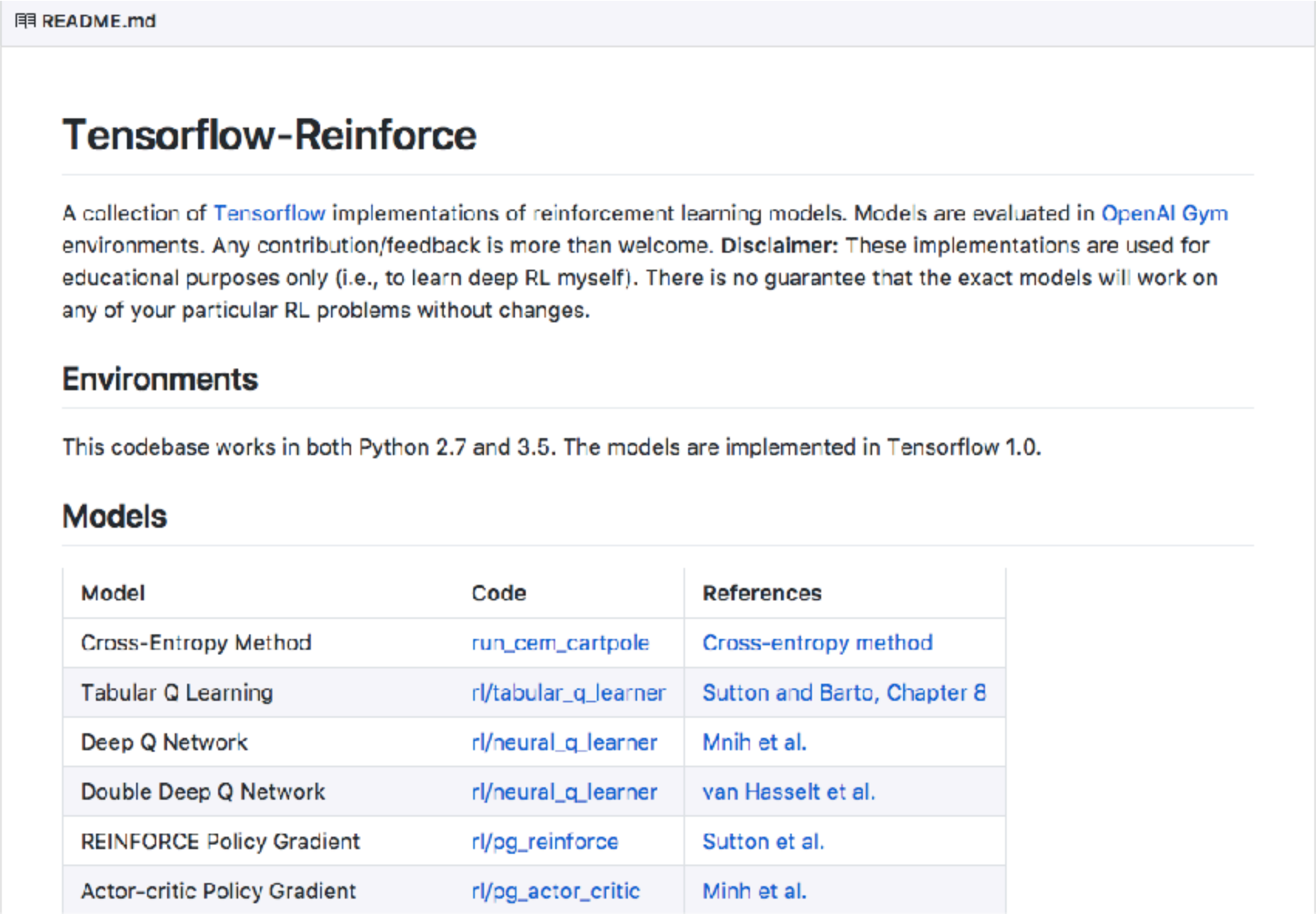
success rate: 0 %

Use-case feature for using RL

- **Hard to get past data**

Any other use case?

Implementation



README.md

Tensorflow-Reinforce

A collection of Tensorflow implementations of reinforcement learning models. Models are evaluated in OpenAI Gym environments. Any contribution/feedback is more than welcome. **Disclaimer:** These implementations are used for educational purposes only (i.e., to learn deep RL myself). There is no guarantee that the exact models will work on any of your particular RL problems without changes.

Environments

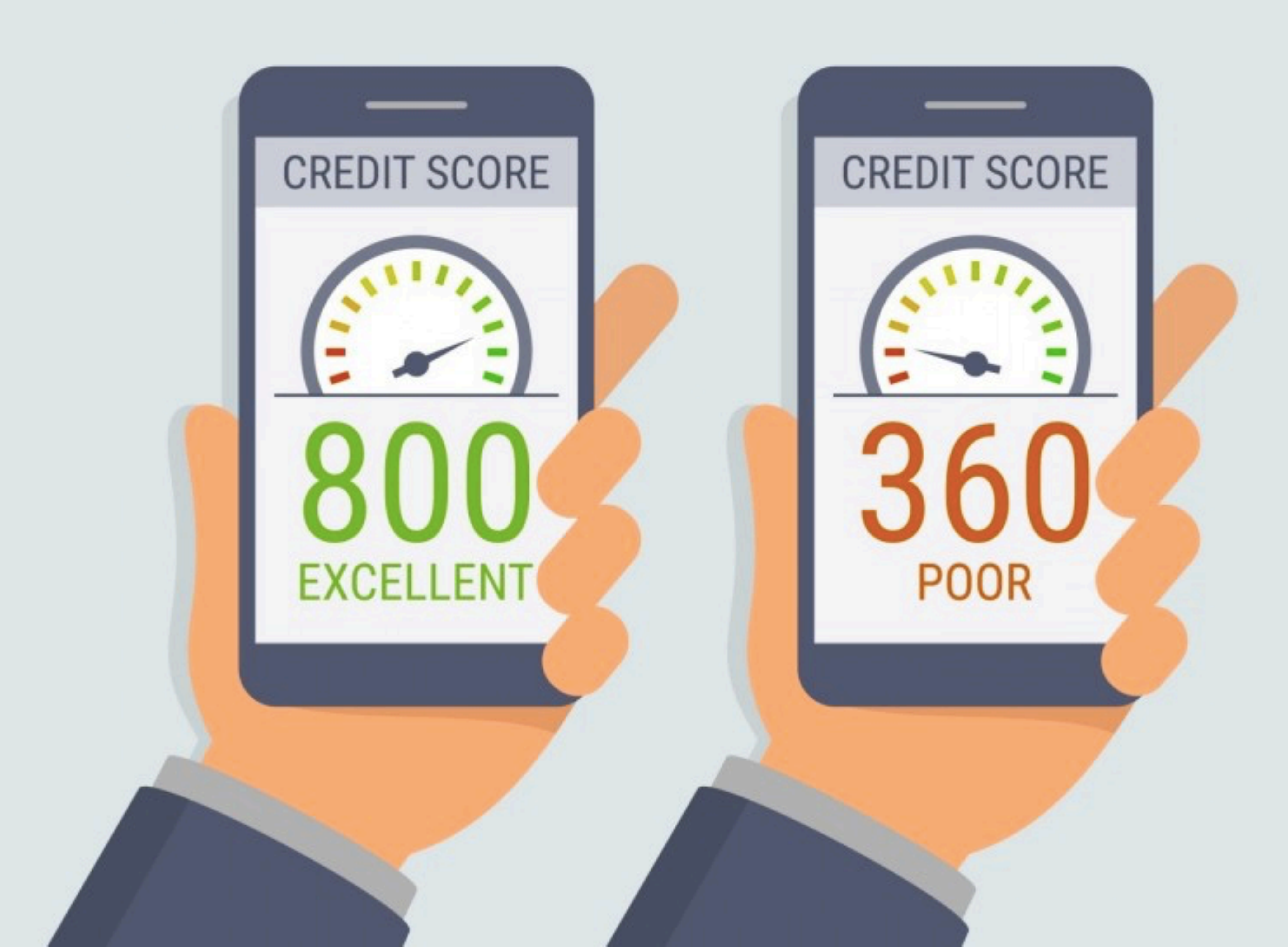
This codebase works in both Python 2.7 and 3.5. The models are implemented in Tensorflow 1.0.

Models

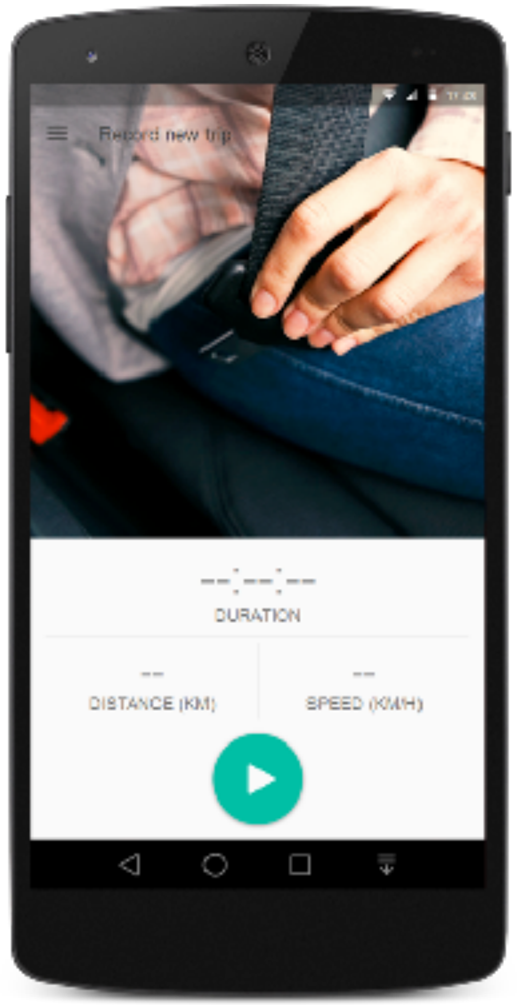
Model	Code	References
Cross-Entropy Method	run_cem_cartpole	Cross-entropy method
Tabular Q Learning	rl/tabular_q_learner	Sutton and Barto, Chapter 8
Deep Q Network	rl/neural_q_learner	Mnih et al.
Double Deep Q Network	rl/neural_q_learner	van Hasselt et al.
REINFORCE Policy Gradient	rl/pg_reinforce	Sutton et al.
Actor-critic Policy Gradient	rl/pg_actor_critic	Minh et al.

Other applications?

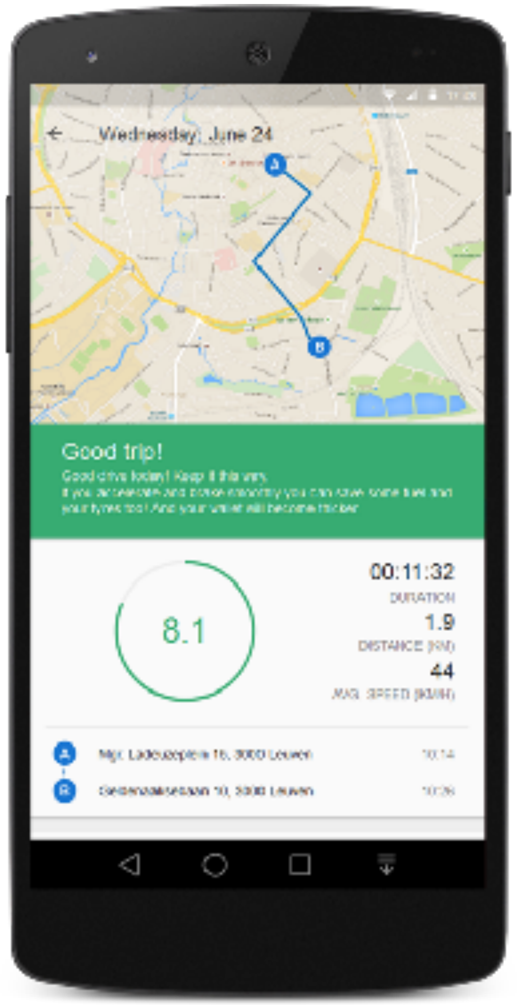
Finance



Insurance



Record a trip

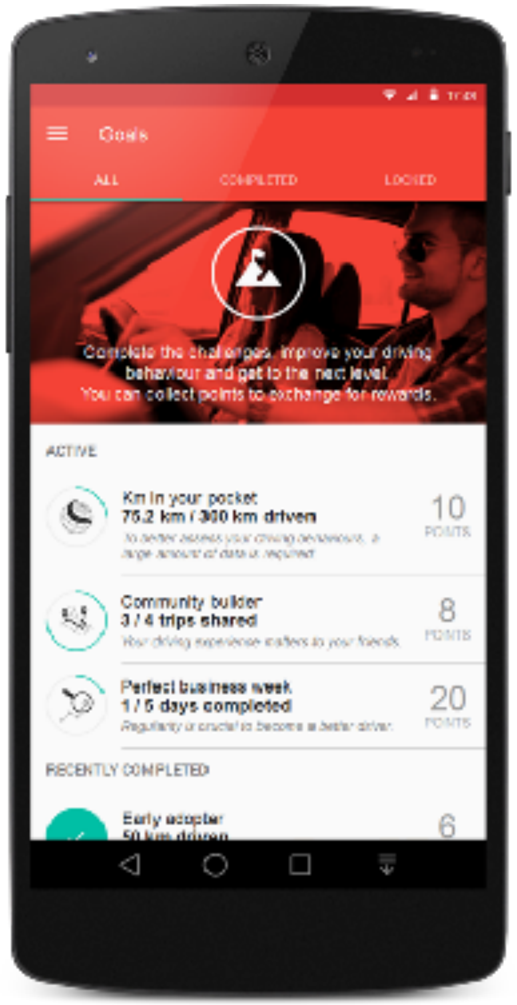

Wednesday, June 24

Good trip!

Good drive today! Keep it this way.
If you accelerate and brake smoothly you can save some fuel and your tyres too! And your wallet will become thicker.

A	Mgr. Ladeuzeplein 15, 3000 Leuven	10:14
B	Geldenaaksebaan 10, 3000 Leuven	10:26

Trip feedback



Complete the challenges, improve your driving behaviour and get to the next level.
You can collect points to exchange for rewards.

ACTIVE

Km in your pocket
75.2 km / 300 km driven
To better assess your driving behaviour, a large amount of data is required.
10 POINTS

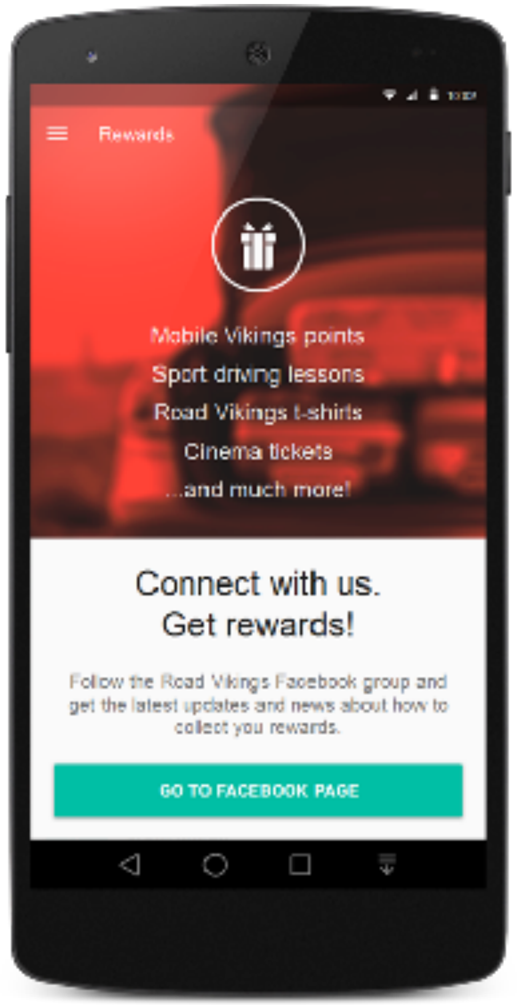
Community builder
3 / 4 trips shared
Your driving experience matters to your friends.
8 POINTS

Perfect business week
1 / 5 days completed
Regularity is crucial to become a better driver.
20 POINTS

RECENTLY COMPLETED

Early adopter
50 km driven
6

Goals & challenges



Mobile Vikings points
Sport driving lessons
Road Vikings t-shirts
Cinema tickets
...and much more!

Connect with us.
Get rewards!

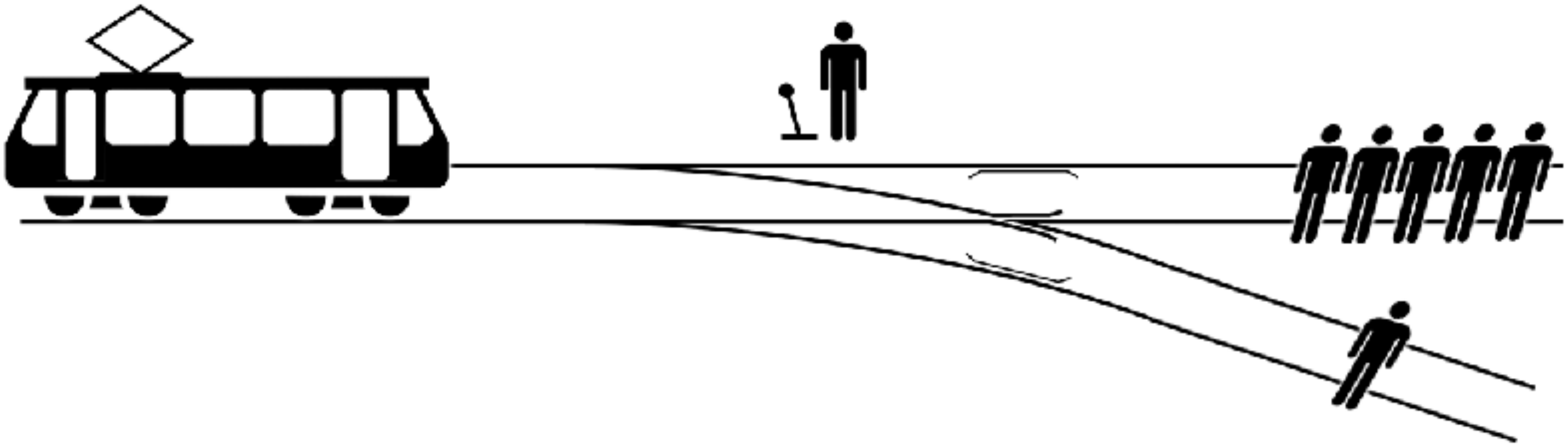
Follow the Road Vikings Facebook group and get the latest updates and news about how to collect you rewards.

Rewards

Healthcare

Pitfalls?

Trolley problem



Ethical dilemma

Guidelines

ETHICALLY ALIGNED DESIGN

A Vision for Prioritizing Human Wellbeing with Artificial Intelligence and Autonomous Systems

- AI systems should be designed so that they always are able to show the process which led to their actions (Government decision-making)

Ethically Aligned design

- How can we assure that AI/AS are accountable?
- How can we ensure that AI/AS are transparent?
- How can we extend the benefits and minimize the risks of AI/AS technology being misused?

Thank you for listening!
Questions?

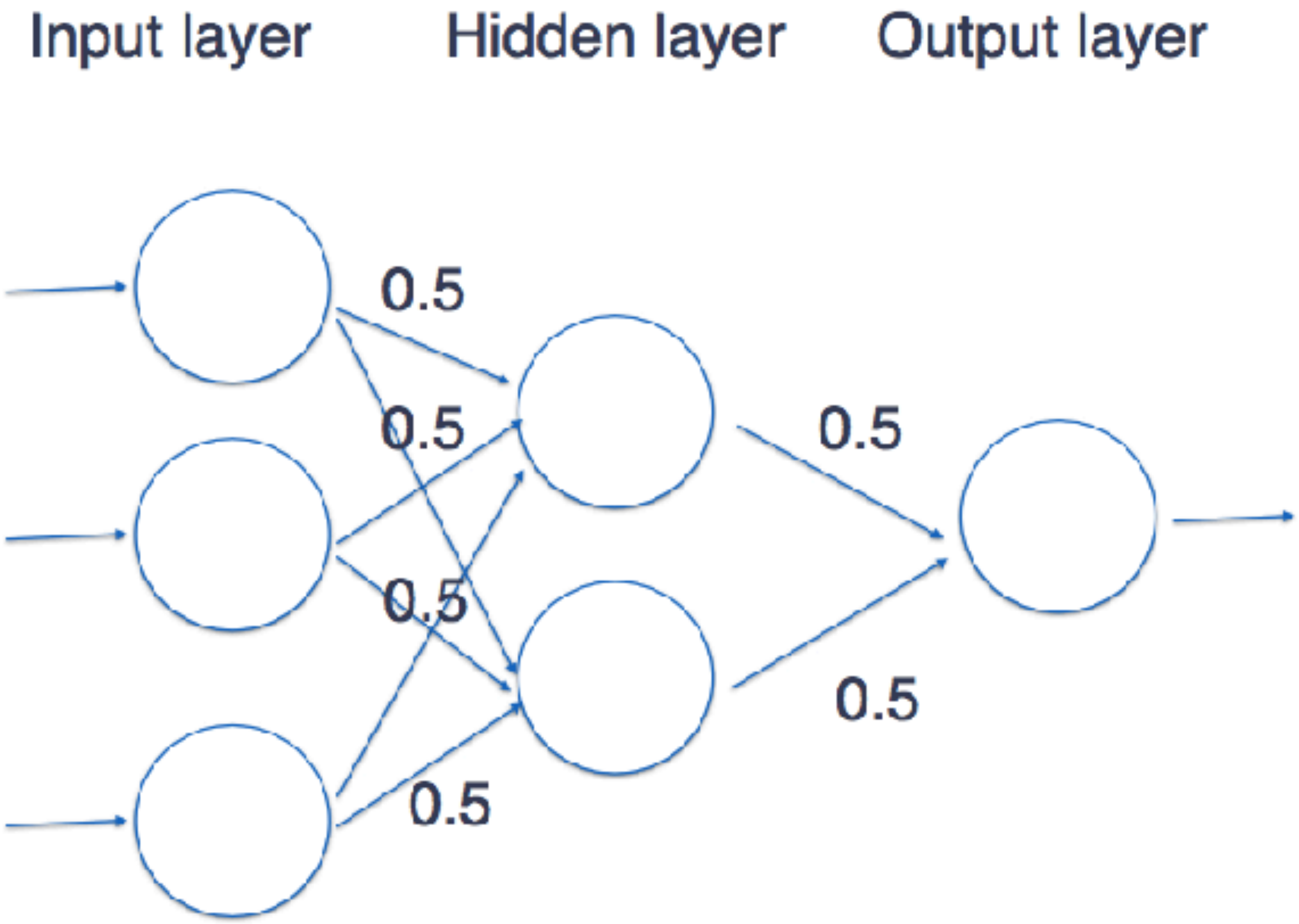
Feel free to add me at

https://www.linkedin.com/in/mitrar/

An example

Training

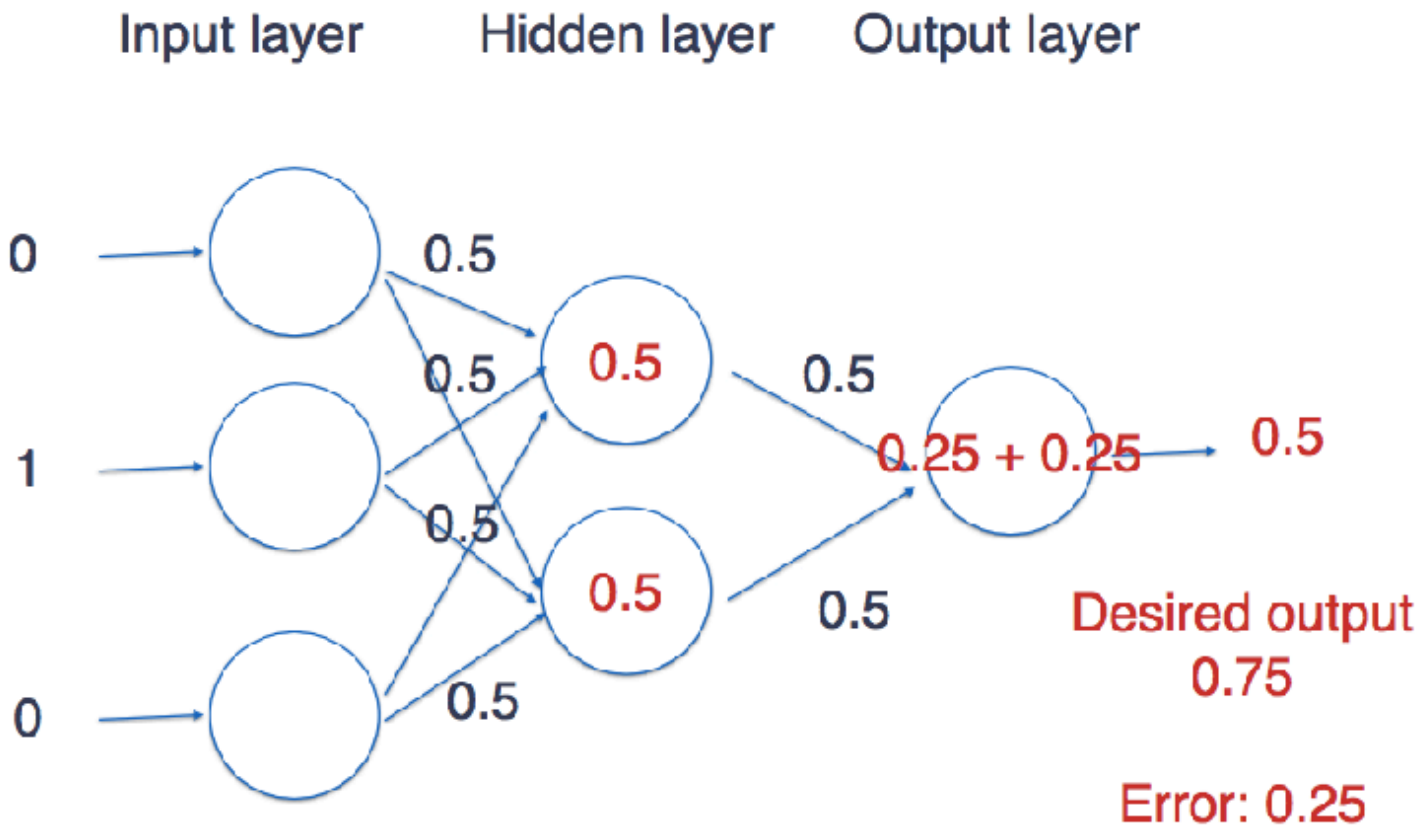
Input: 0 1 0
Output: 0.75



An example

Training

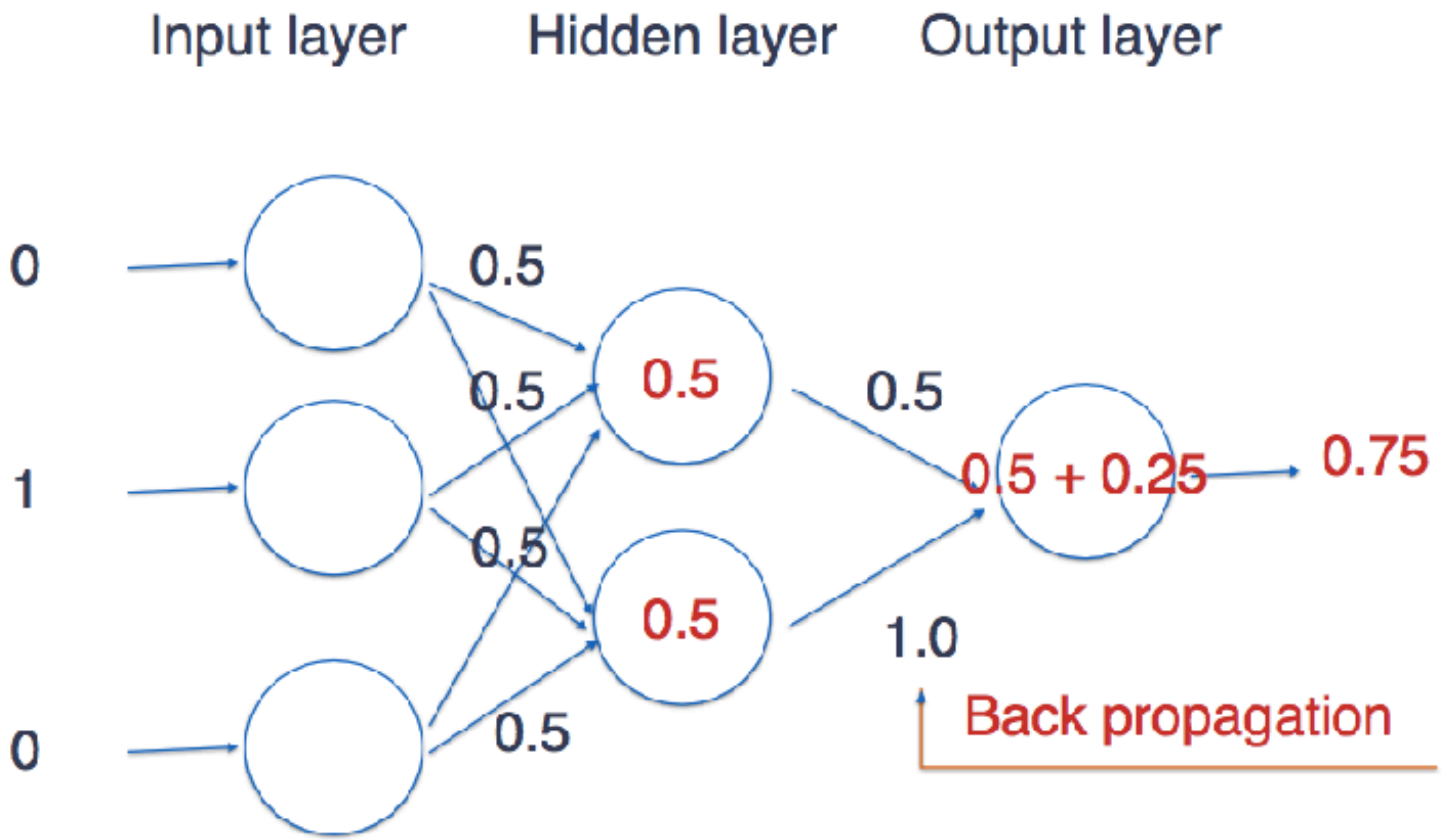
Input: 0 1 0
Output: 0.75

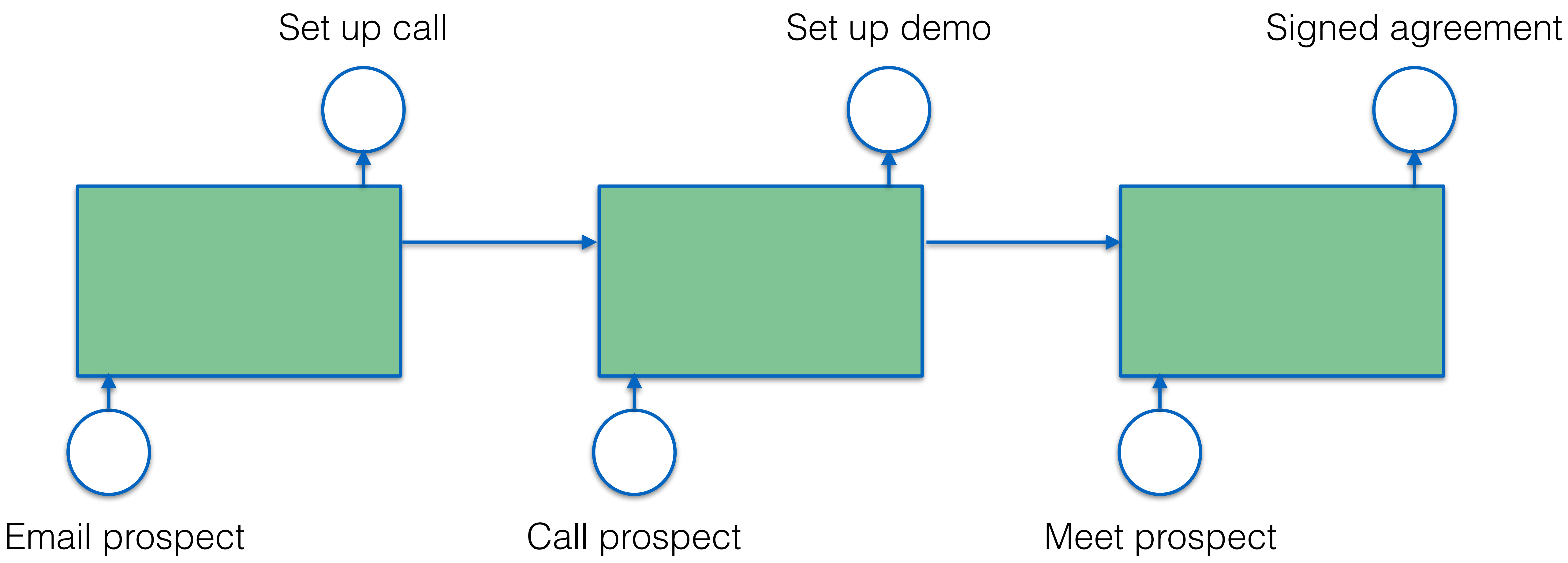


An example

Training

Input: 0 1 0
Output: 0.75





Training LSTM