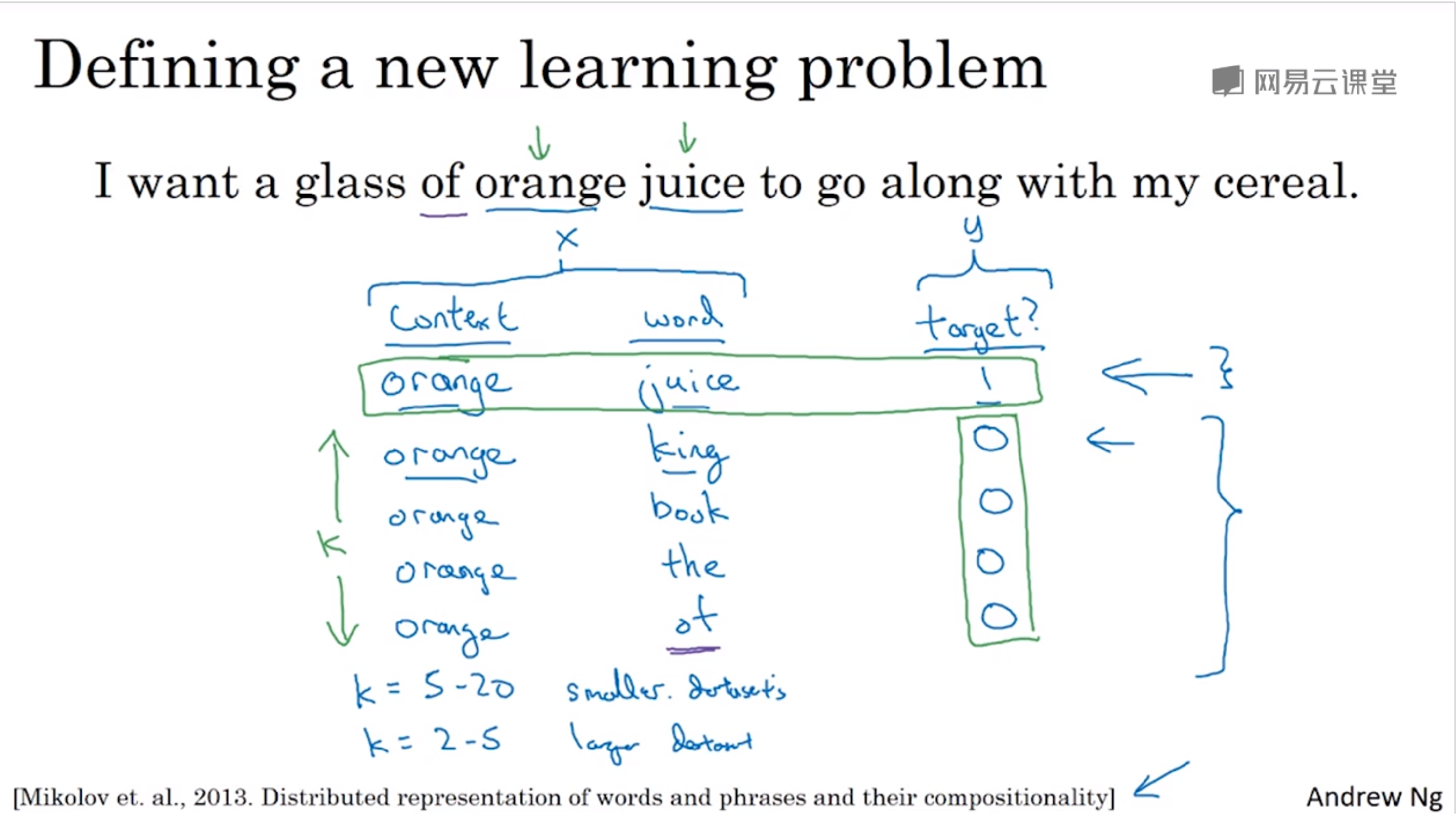
NLP and Word Embeddings

Negative sampling

deeplearning.ai



Defining a new learning problem

I want a glass of orange juice to go along with my cereal.

Context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

k = 5-20 smaller datasets

k = 2-5 larger dataset

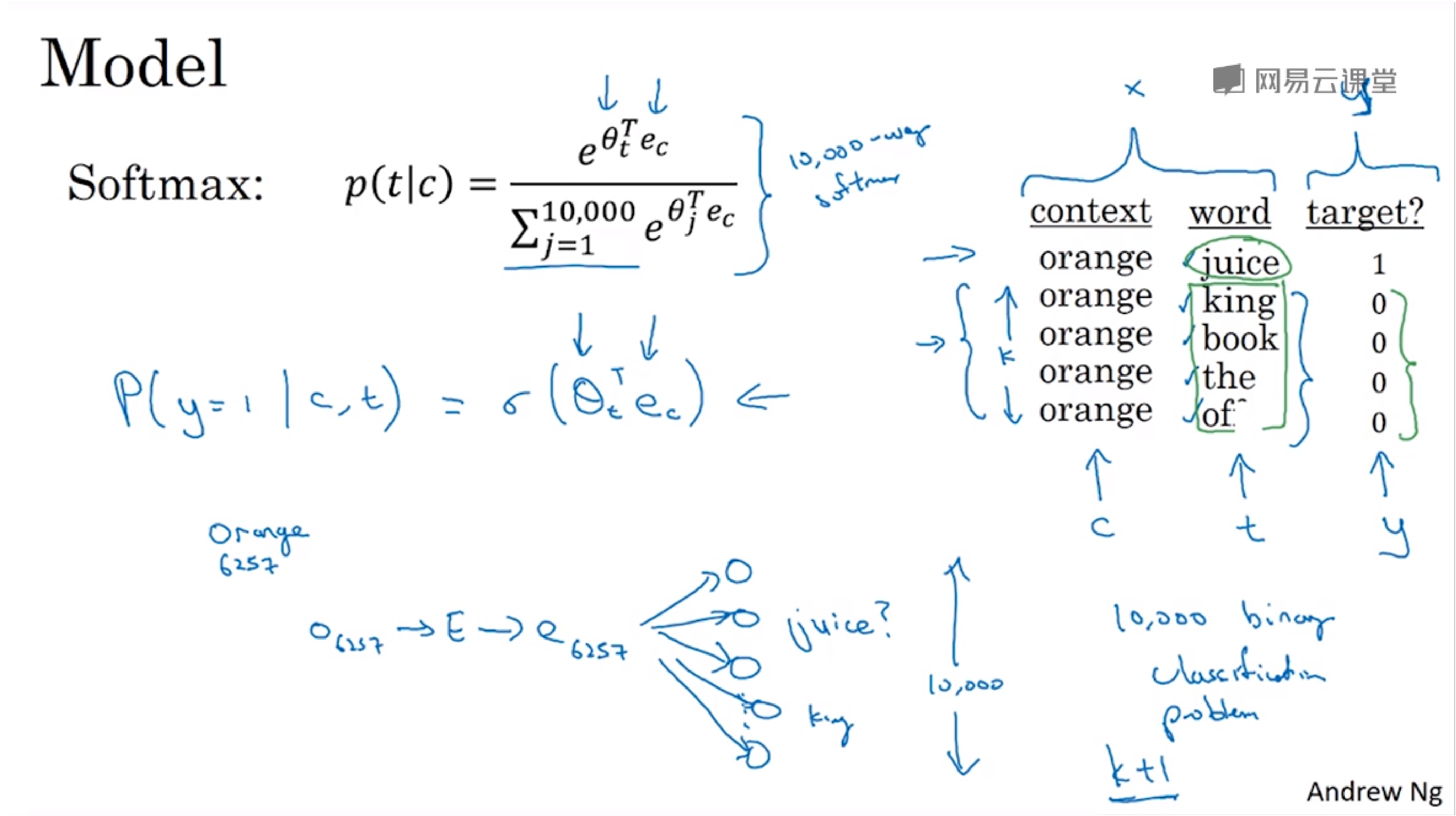
[Mikolov et. al., 2013. Distributed representation of words and phrases and their compositionality]

Andrew Ng

It's really to try to distinguish between these two types of distributions from which you might sample a pair of words.

larger training dataset smaller k

Next, let's describe the supervised learning model for learning a mapping from x to y.



Model

Softmax: $$p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$$

10,000-way softmax

$$P(y=1 | c,t) = \sigma(\theta_t^T e_c)$$

context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

x
y
k
c
t
y
Orange 6257
$o_{6257} \to E \to e_{6257}$
juice?
king
10,000
10,000 binary classification problem
k+1

Andrew Ng

k to 1 ratio of negative to positive examples

And on every iteration, we're only going to train k+1 of them.

So the parameters are similar as before, you have one parameter vector theta for each possible target word.
And a separate parameter vector, really the embedding vector for each possible context word.
And we're going to use this formula to estimate the probability that y is equal to 1.

Selecting negative examples

x		y
context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

↑ k

the, of, and, ...

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=1}^{10,000} f(w_j)^{3/4}}$$

$$\frac{1}{|V|}$$

↑

f empirical frequency of word in your corpus