

# DS 207: Introduction to Natural Language Processing



## Text Classification II

---

Danish Pruthi

Slides courtesy: Graham Neubig

# Housekeeping

- Classes in January will happen in the MRC auditorium
- Teams link: <https://tinyurl.com/iisc-nlp-teams>
- Assignment 1 to be out next Monday ...



# Example: topic classification

- **Sports:** "Kohli scores another remarkable hundred"
- **Politics:** "Minister announces new metro plans ahead of elections"
- **Entertainment:** "A sleeper hit, 12th fail, is praised by many Bollywood actors"
- **Finance:** "Stocks for Indian delivery startups plummet"

# Generative Naive Bayes

*i<sup>th</sup> example*

$$P(X^{(i)}, y^{(i)}) = P(y^{(i)}) P(X^{(i)} | y^{(i)})$$

$P(X | y)$   $\equiv$   $P(w_1, w_2, w_3, \dots w_t | y)$

$$P(X | y) = \prod_i^t P(w_i | y)$$

*assumed independence*

$$P(w_i = "Kohli" | y = "sports")$$

# Estimating parameters

$$P(w_i = \text{"Kohli"} | y = \text{"sports"})$$

$$= \frac{\text{count}(w_i = \text{Kohli} \in y = \text{sports})}{\sum_{w \in |V|} \text{count}(w \in y = \text{sports})}$$

# Add- $\alpha$ smoothing (or Laplace smoothing)

$$P(w_i = \text{"Kohli"} \mid y = \text{"finance"})$$

$$= \frac{\text{count}(w_i = \text{Kohli} \in y = \text{finance}) + \alpha}{\sum_{w \in |V|} (\text{count}(w \in y = \text{finance}) + \alpha)}$$

# Evaluation

→ 2 classes

$TP + TN$

$P + N$

- Accuracy

$\approx$

- Precision =  $TP / (\text{Predicted}) P$

$= TP / (\text{Actual}) P$

- Recall

$\approx$   $2 \cdot P \cdot R / (P + R)$

		Predicted condition	
		Total population = $P + N$	Predicted positive (PP)
		Positive (P) [a]	True positive (TP), hit [b]
Actual condition	Positive (P) [a]		✓
Negative (N) [d]		False positive (FP), false alarm, overestimation	True negative (TN), correct rejection [e]

# What metrics are best suited

- Diagnosing rare type of cancer
- Criminal punishment ~~A~~
- Detecting spam
- Recruitment/Filtering based on text in the CVs
- Recommending products/songs/movies

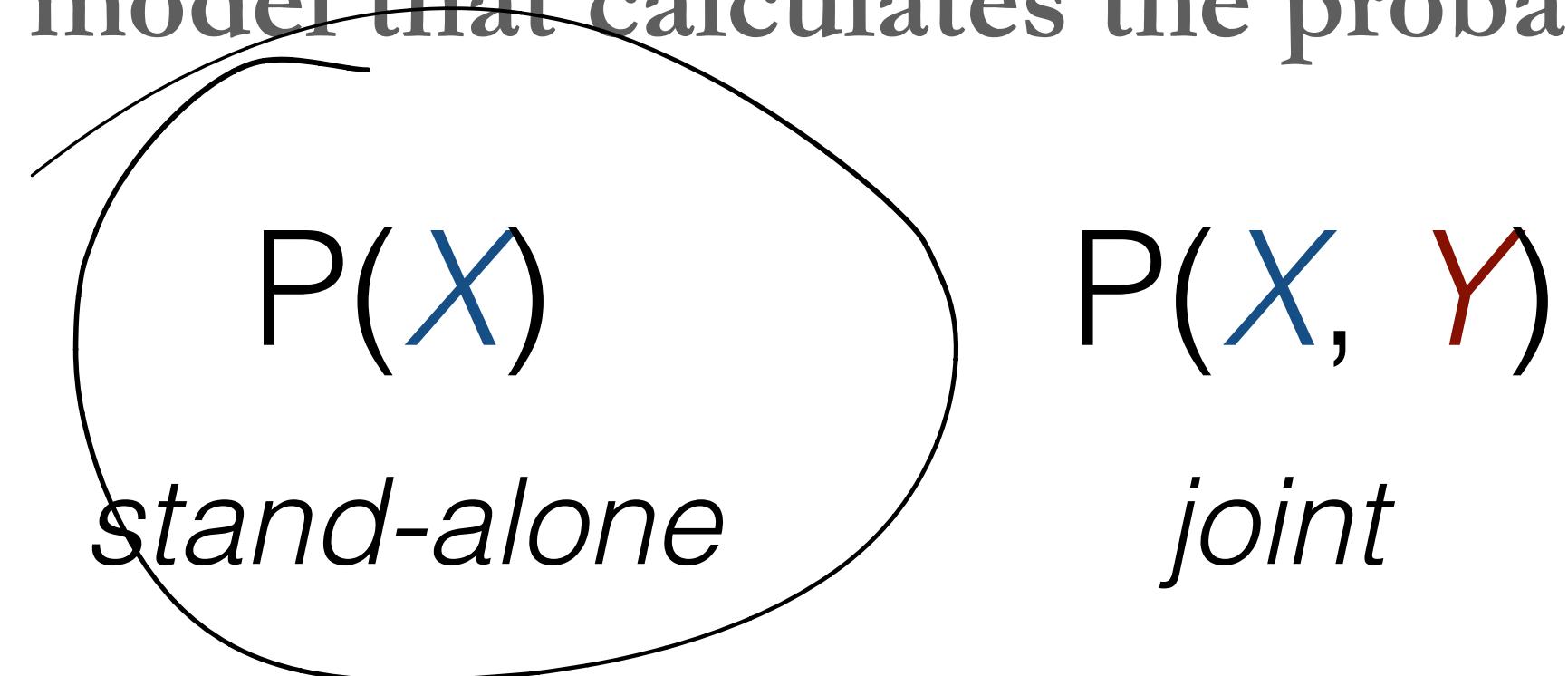
recall

f

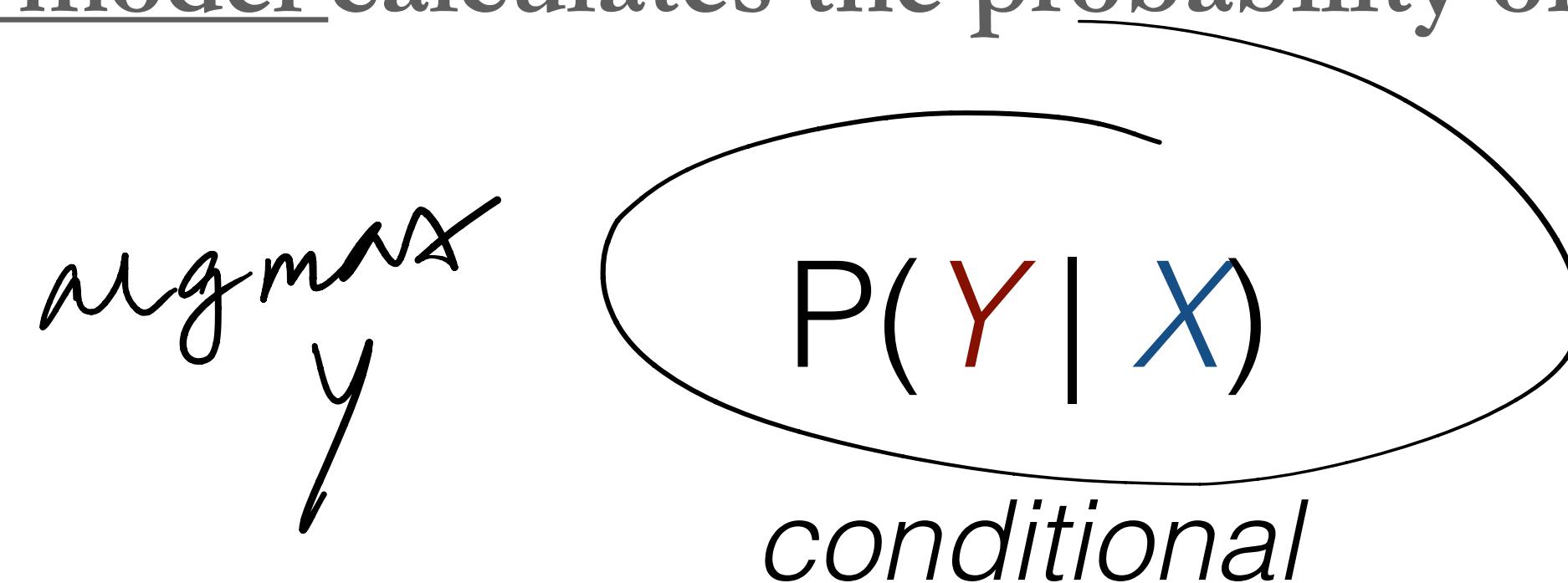
P

# Generative vs Discriminative models

- Generative model: a model that calculates the probability of the input data itself



- Discriminative model calculates the probability of a class (or trait) given the data



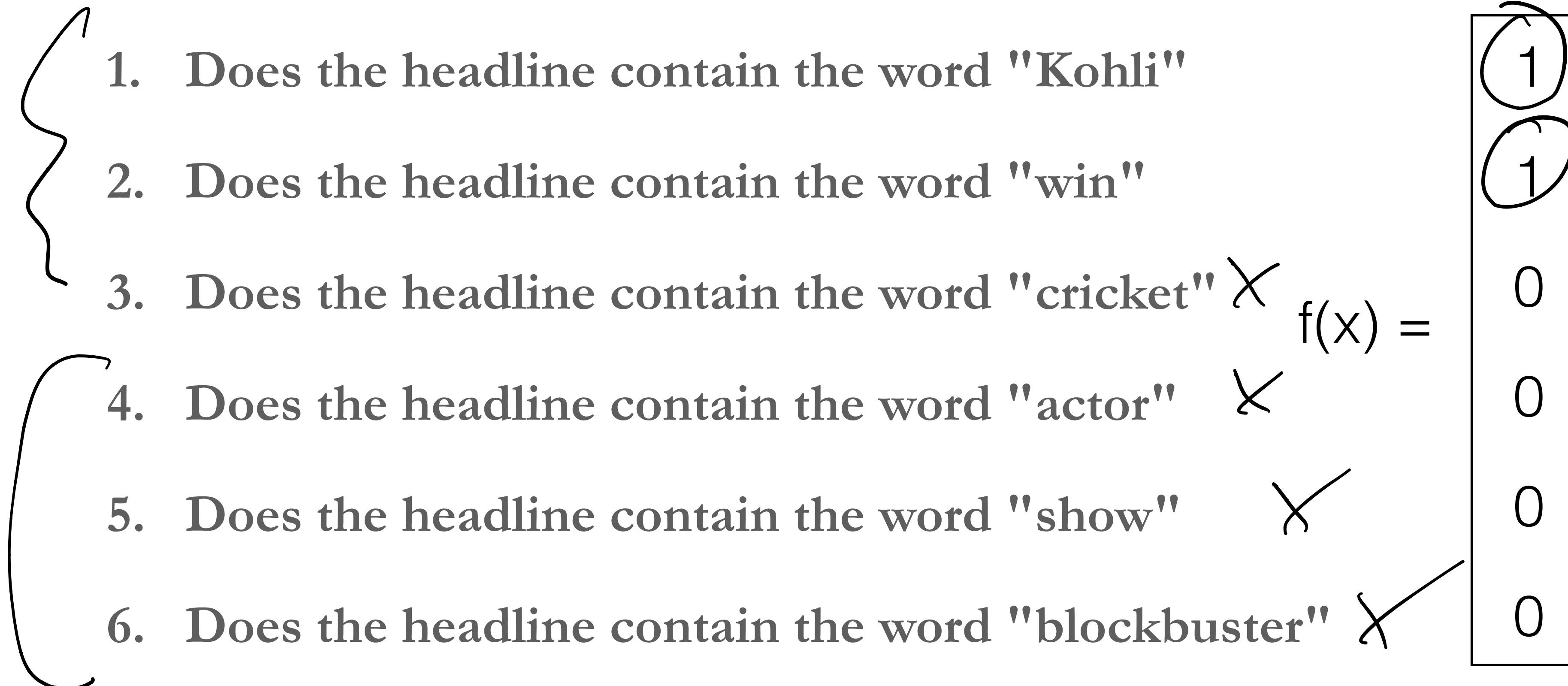
# Rule based systems: is the headline sports or entertainment?

$x = \text{Kohli Scores a ton. India Win.}$

- **Feature extraction:** Extract the salient features for making the predictions from text
  1. Does the headline contain the word "Kohli"
  2. Does the headline contain the word "win"
  3. Does the headline contain the word "cricket"
  4. Does the headline contain the word "actor"
  5. Does the headline contain the word "show"
  6. Does the headline contain the word "blockbuster"

# Rule based systems: is the headline sports or entertainment?

- **Feature extraction:** Extract the salient features for making the predictions from text



# Rule based systems: is the headline sports or entertainment?

$$f(\mathbf{x}) \cdot \mathbf{w}$$

- **Feature extraction:** Extract the salient features for making the predictions from text

1. Does the headline contain the word "Kohli"
2. Does the headline contain the word "win"
3. Does the headline contain the word "cricket"
4. Does the headline contain the word "actor"
5. Does the headline contain the word "show"
6. Does the headline contain the word "blockbuster"

$$f(\mathbf{x}) =$$

1
1
0
0
0
0

$$\mathbf{W} =$$

+1
+1
+1
-1
-1
-1

# A three step process for making predictions

- Feature extraction: Extract the salient features for making the decision from text

$\mathbb{R}^d$

$$\mathbf{h} = f(\mathbf{x})$$

text

$4 \times d$

- Score calculation: Calculate a score for one or more possibilities

$\mathbb{R}^d$

$$s = \mathbf{w} \cdot \mathbf{h}$$

binary

$$s = \mathbf{W} \mathbf{h}$$

multi-class

$\mathbb{R}$

$\mathbb{R}^4$

- Decision function: Choose one of the several possibilities

$\mathbb{R}$

$$\hat{y} = \text{decide}(s)$$

$\wedge$

HAT

$s > t$

$y^*$

$+$

$-$

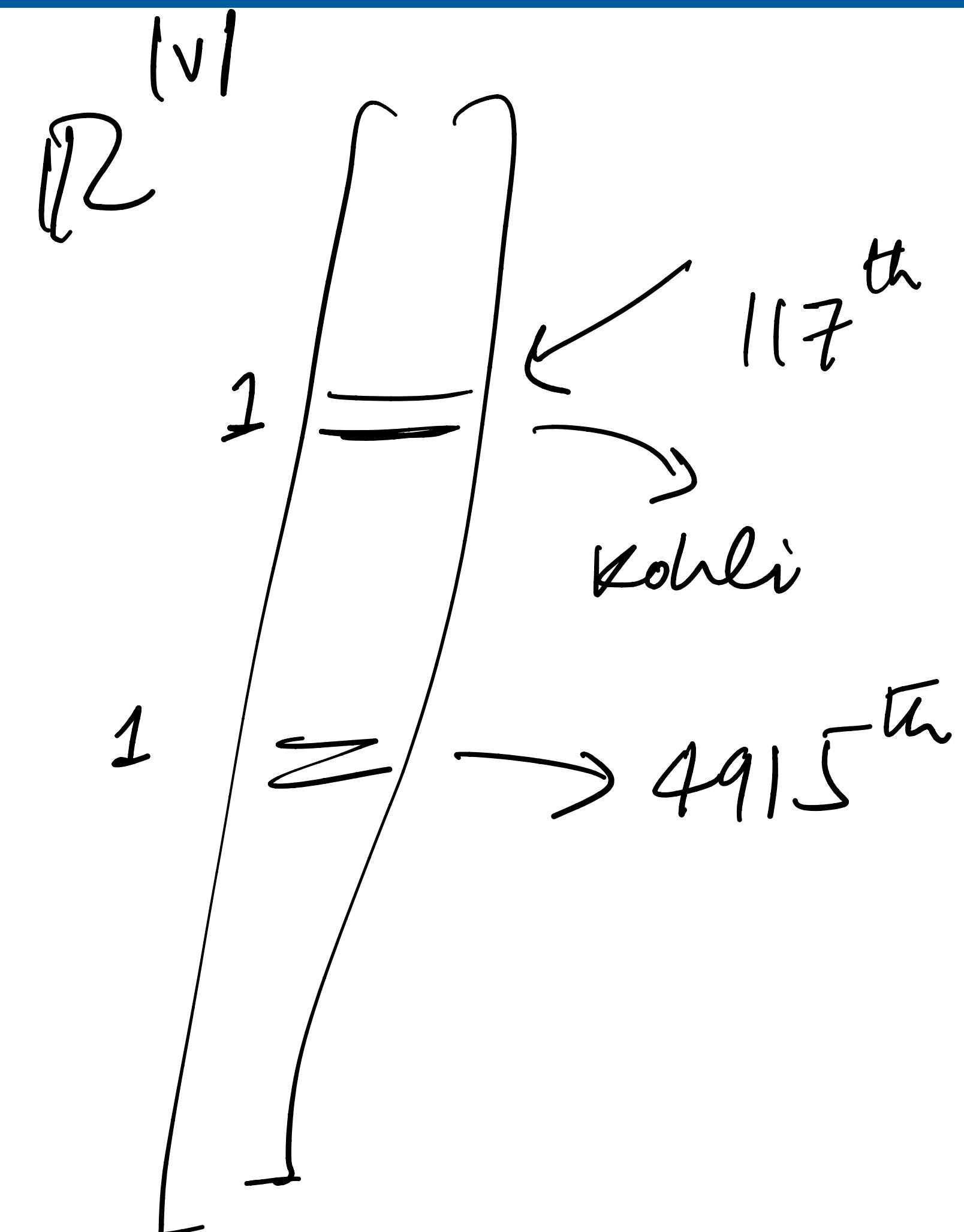
# Another discriminative classifier

# Another discriminative classifier

- Feature Extraction:  $h = f(x)$

One hot vectors ("bag of words")

Count      veetori 201



# Another discriminative classifier

- **Feature Extraction:**  $h = f(x) \rightarrow \mathbb{R}^{|V|}$

One hot vectors ("bag of words")

- **Score Calculation:** binary or multi-class

$$s = \mathbf{w} \cdot \mathbf{h}$$

The diagram shows a dot product between two vectors. On the left, a vector  $\mathbf{w}$  is shown with a circled 'w' at its head and a vertical arrow pointing down to a circled '12' at its tail. On the right, a vector  $\mathbf{h}$  is shown with a circled 'h' at its head and a vertical arrow pointing up to a circled '|V|' at its tail. A horizontal line connects the two vectors, with a dot at the center indicating their multiplication.

$$s = \mathbf{W} \mathbf{h}$$

The diagram shows a matrix multiplication. On the left, a large oval contains the letter 'W'. An arrow points from the bottom of this oval to a circled 'n' at the top of a vertical arrow pointing down. Another arrow points from the right side of the oval to a circled '|V|' at the top of a vertical arrow pointing up. To the right of the oval, there is a circled 'x' with arrows pointing to both the 'n' and '|V|' labels. A horizontal line connects the oval to a second smaller oval containing the letter 'h'. An arrow points from the bottom of this second oval to a circled '1' at the top of a vertical arrow pointing up.

# Another discriminative classifier

- **Feature Extraction:**  $h = f(x)$

One hot vectors ("bag of words")

- **Score Calculation:** binary or multi-class

$$s = \mathbf{w} \cdot \mathbf{h} \quad s = \mathbf{W}\mathbf{h}$$

- **Decision:** Convert to a probability:

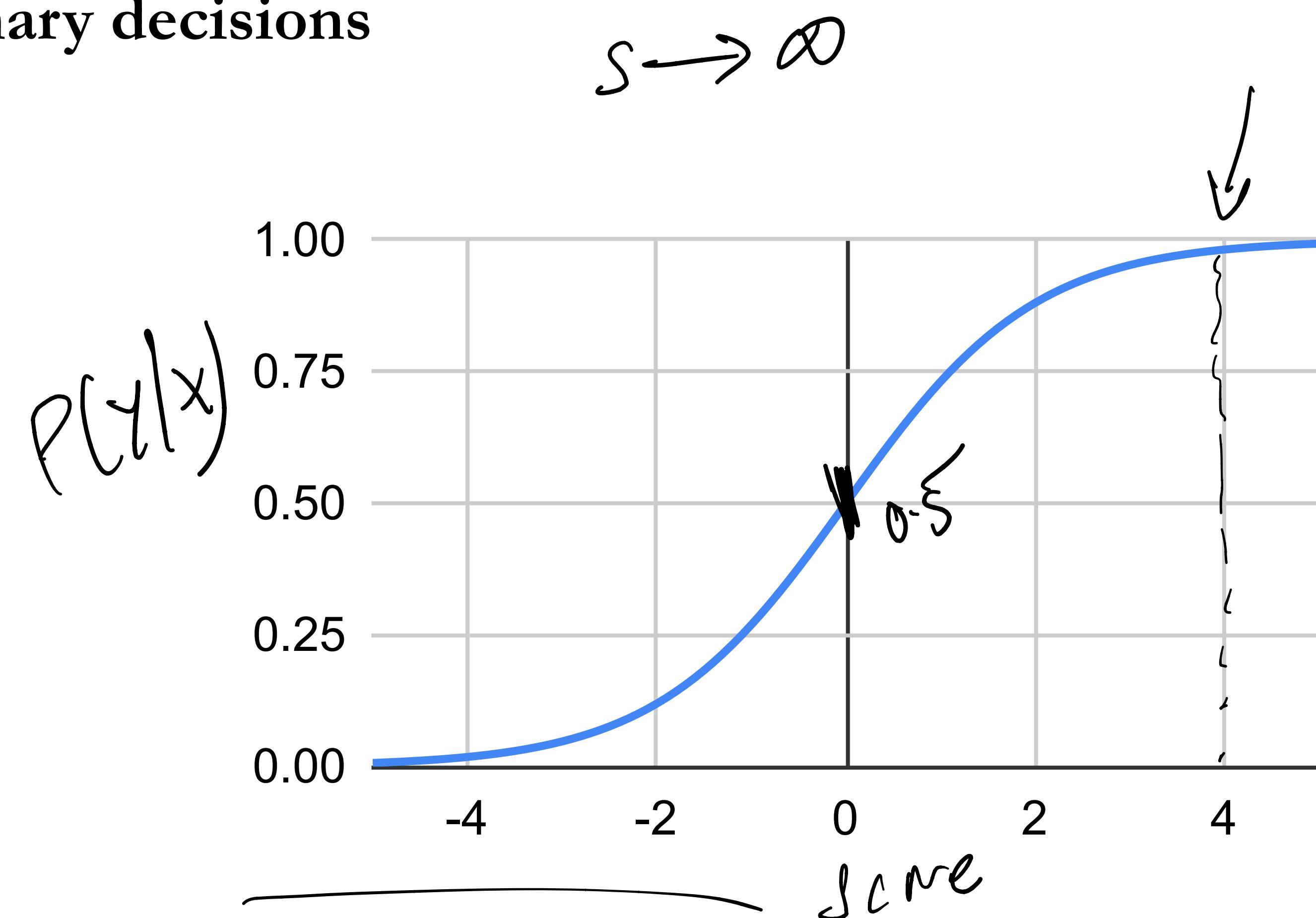
$$P(y|x) = \underbrace{\sigma(s)}_{\sim} \text{ or softmax}(\mathbf{s})$$

$$\cancel{k \times |V| \times |N|}$$

# Sigmoid function

- Sigmoid can be used for binary decisions

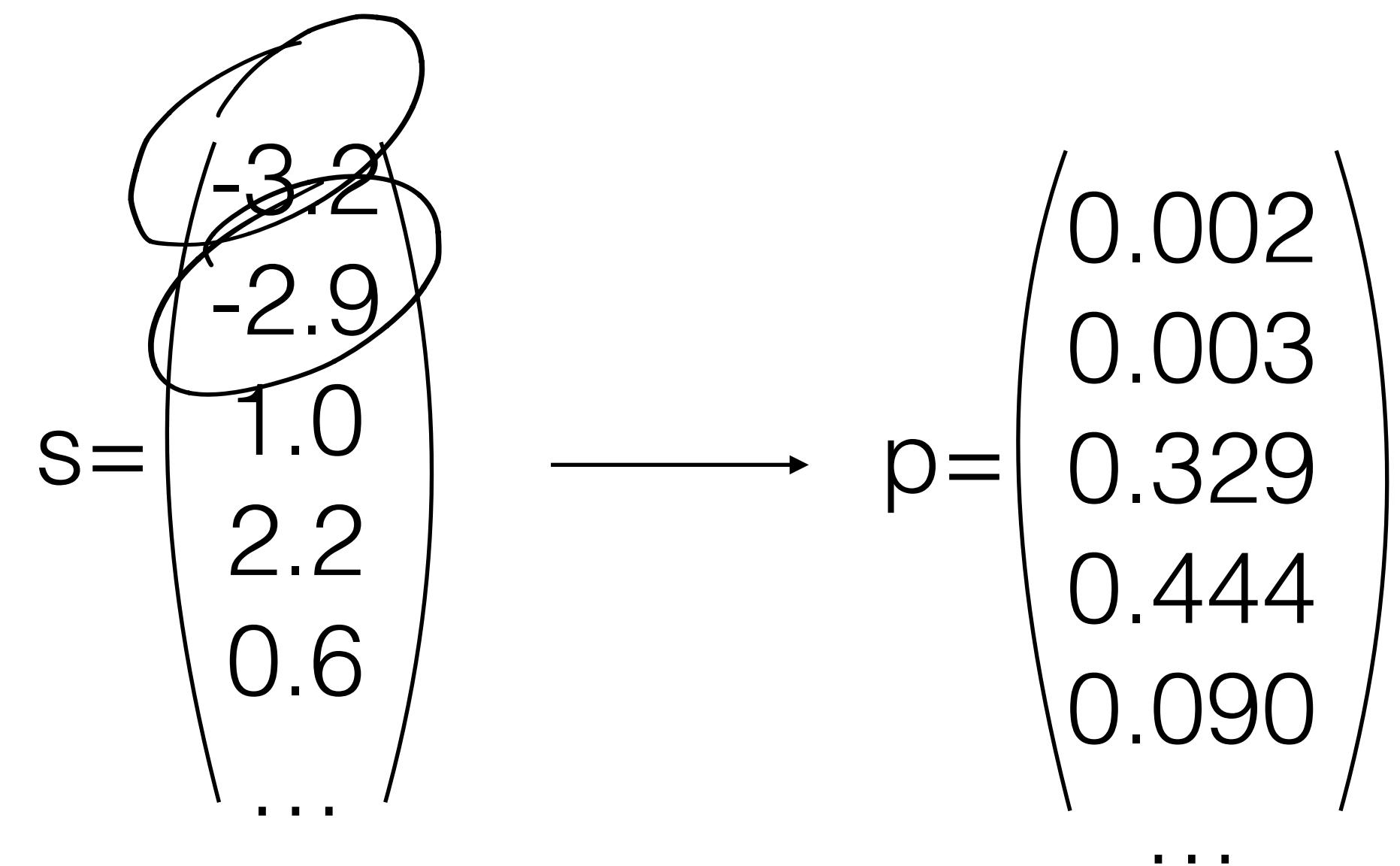
$$\sigma(s) = \frac{1}{1 + e^{-s}}$$



# Softmax

- Softmax is used for multi-label classification

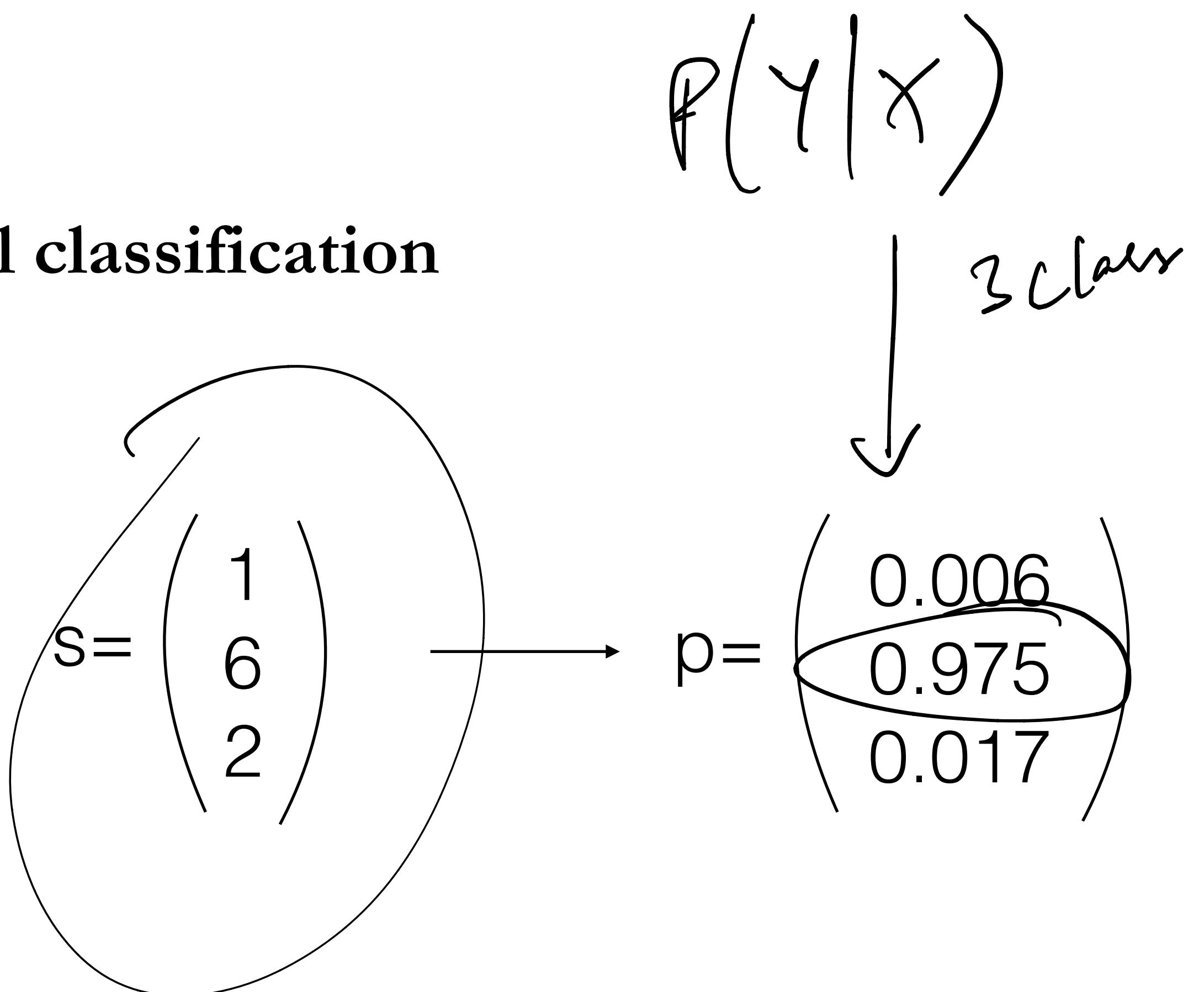
$$\text{softmax}(s) = \frac{e^{s_i}}{\sum_i^d e^{s_i}}$$



# Softmax

- Softmax is used for multi-label classification

$$\text{softmax}(s) = \frac{e^{s_i}}{\sum_i^d e^{s_i}}$$



# Recap: Discriminative models

- Define a model that calculates probability directly based on parameters  $\mathbf{W}$

$$\begin{aligned} P(Y|X; W) \\ = \sigma(w^T f(x)) \\ = \text{softmax}(w \cdot f(x)) \end{aligned}$$

# Computing loss (or error/cost) function

# Computing loss (or error/cost) function

Define a loss function (a function which is lower for better models):

# Computing loss (or error/cost) function

Define a loss function (a function which is lower for better models):

# Computing loss (or error/cost) function

Define a loss function (a function which is lower for better models):

$$P(Y, X) = \underbrace{P(Y|X)}_{Z}$$



$$L(W) = - \sum_{\substack{x, y \in D \\ 1/1}} \log P(y|x; W) \quad [\text{negative log likelihood}]$$

— training

# Computing loss (or error/cost) function

Define a loss function (a function which is lower for better models):

$$L(W) = - \sum_{x, y \in D} \log P(y | x; W) \quad [\text{negative log likelihood}]$$

# Computing loss (or error/cost) function

Define a loss function (a function which is lower for better models):

$$L(W) = - \sum_{x, y \in D} \log P(y | x; W) \quad [\text{negative log likelihood}]$$

$\arg\min_W L(W) = - \sum_i^n (y_i \log P(y = 1 | x_i; W) + (1 - y_i) \log(P(y = 0 | x_i; W))$

[binary cross entropy]

$y_i = 1$  *+ class*

$y_i = 0$  *- ve class*

$$(1 - y_i) \log(-P(y=1 | x_i; w))$$

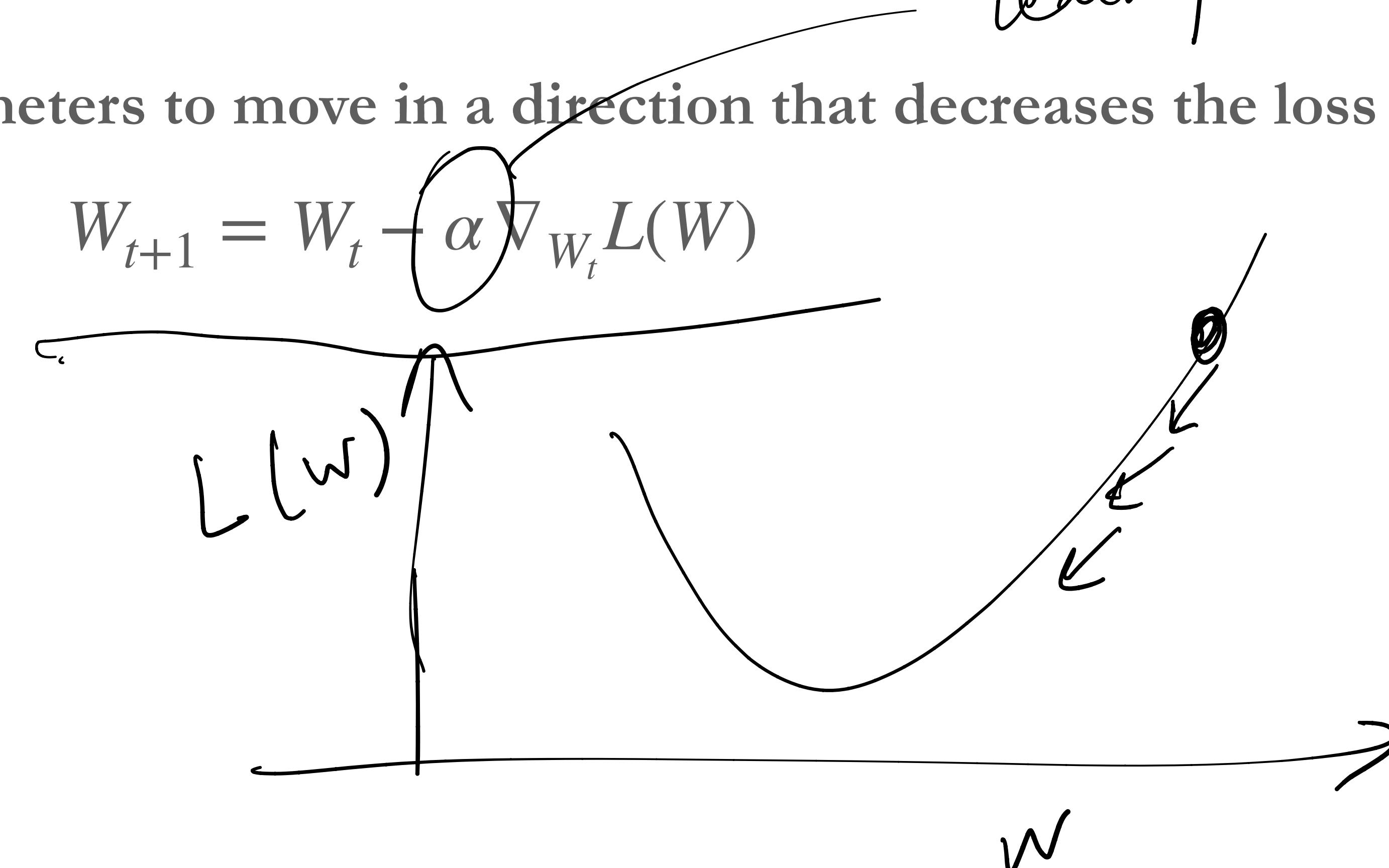
# Learn parameters through gradient descent

GREAT!

A big win for India!  
great  
learning rate

- Compute the gradient of the loss function with respect to the parameters
- Keep updating the parameters to move in a direction that decreases the loss

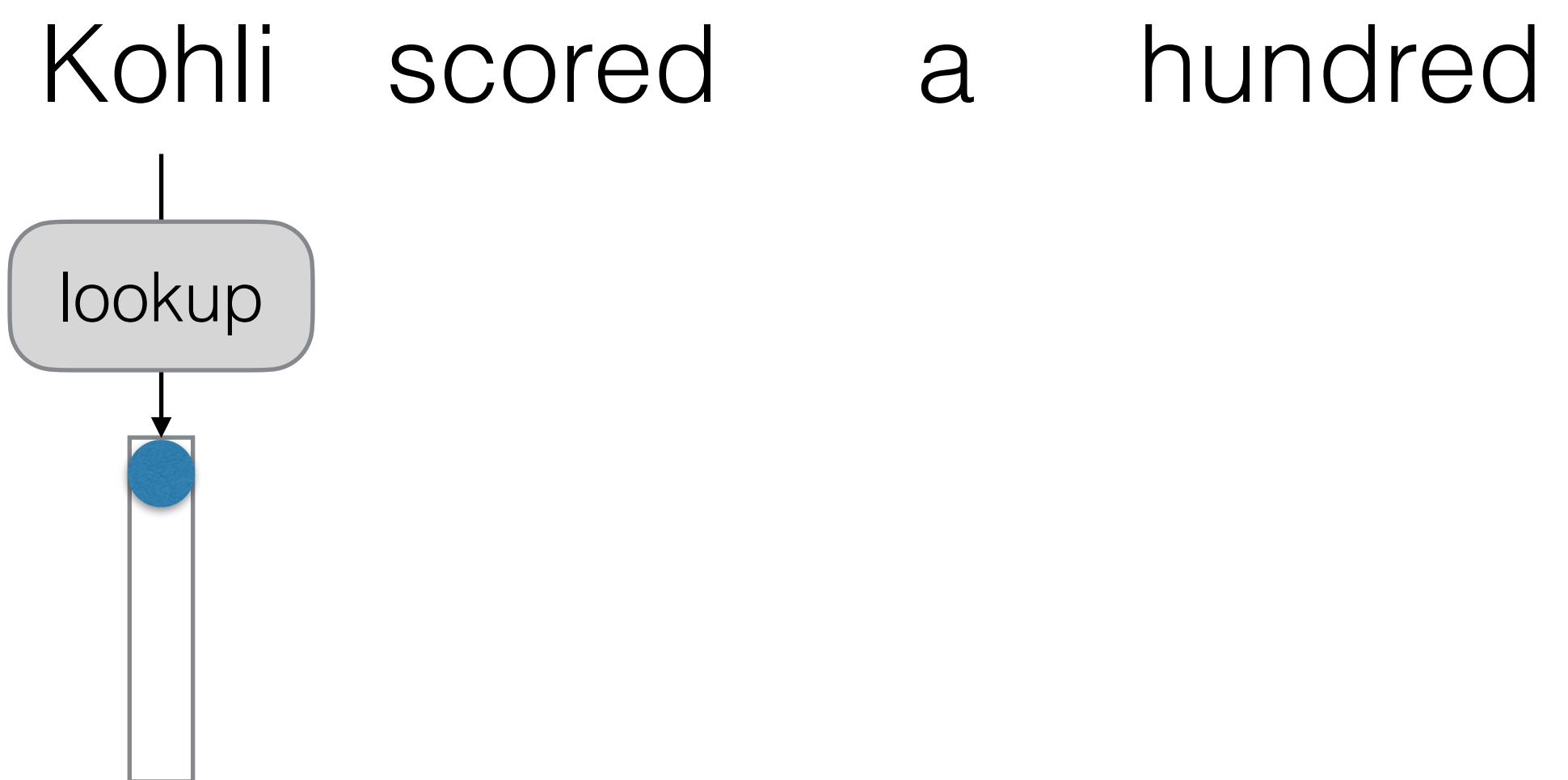
Kohli US  
Kohli US  
hundred 100 ton



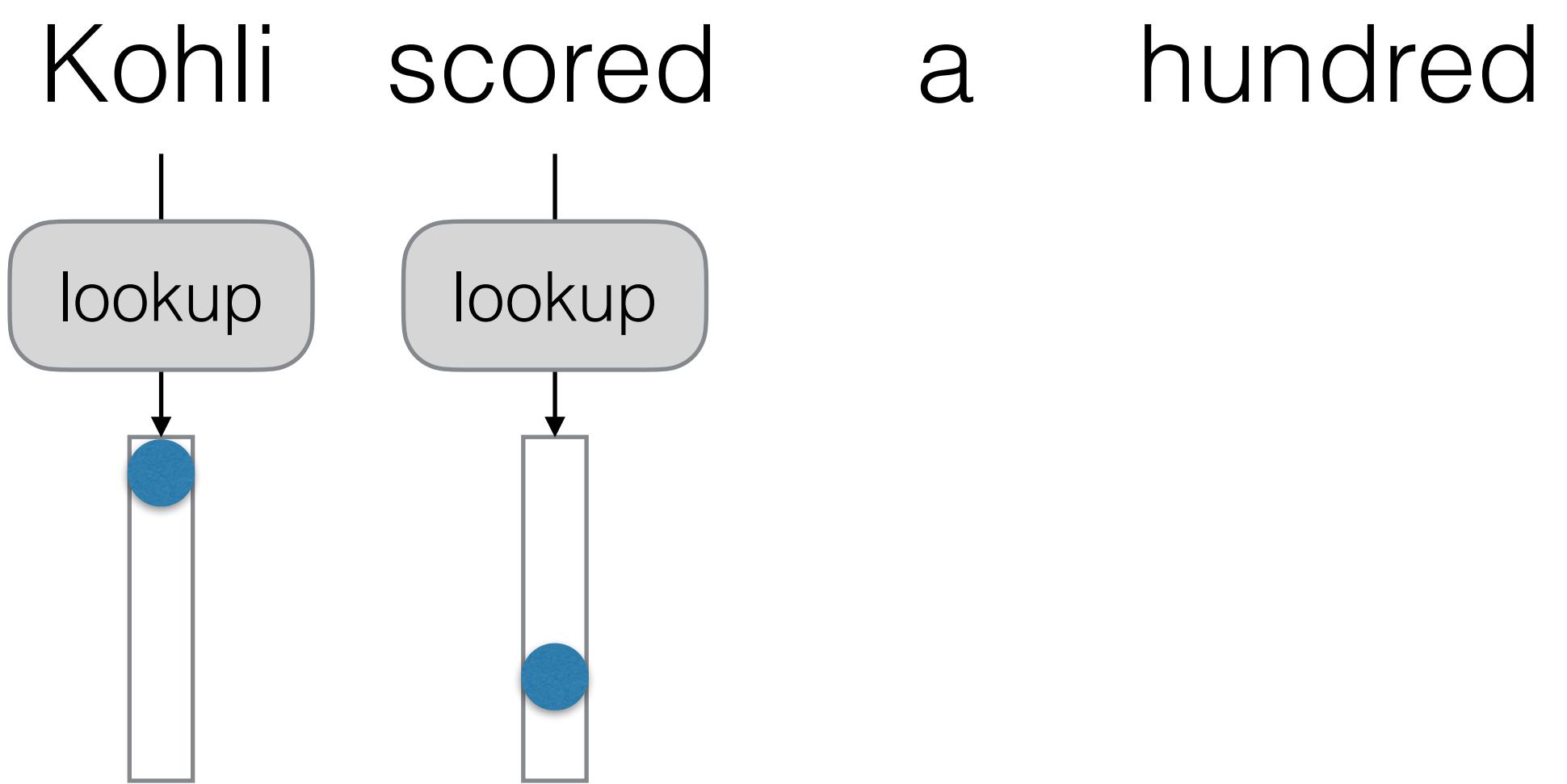
# Computational Graph View

Kohli scored a hundred

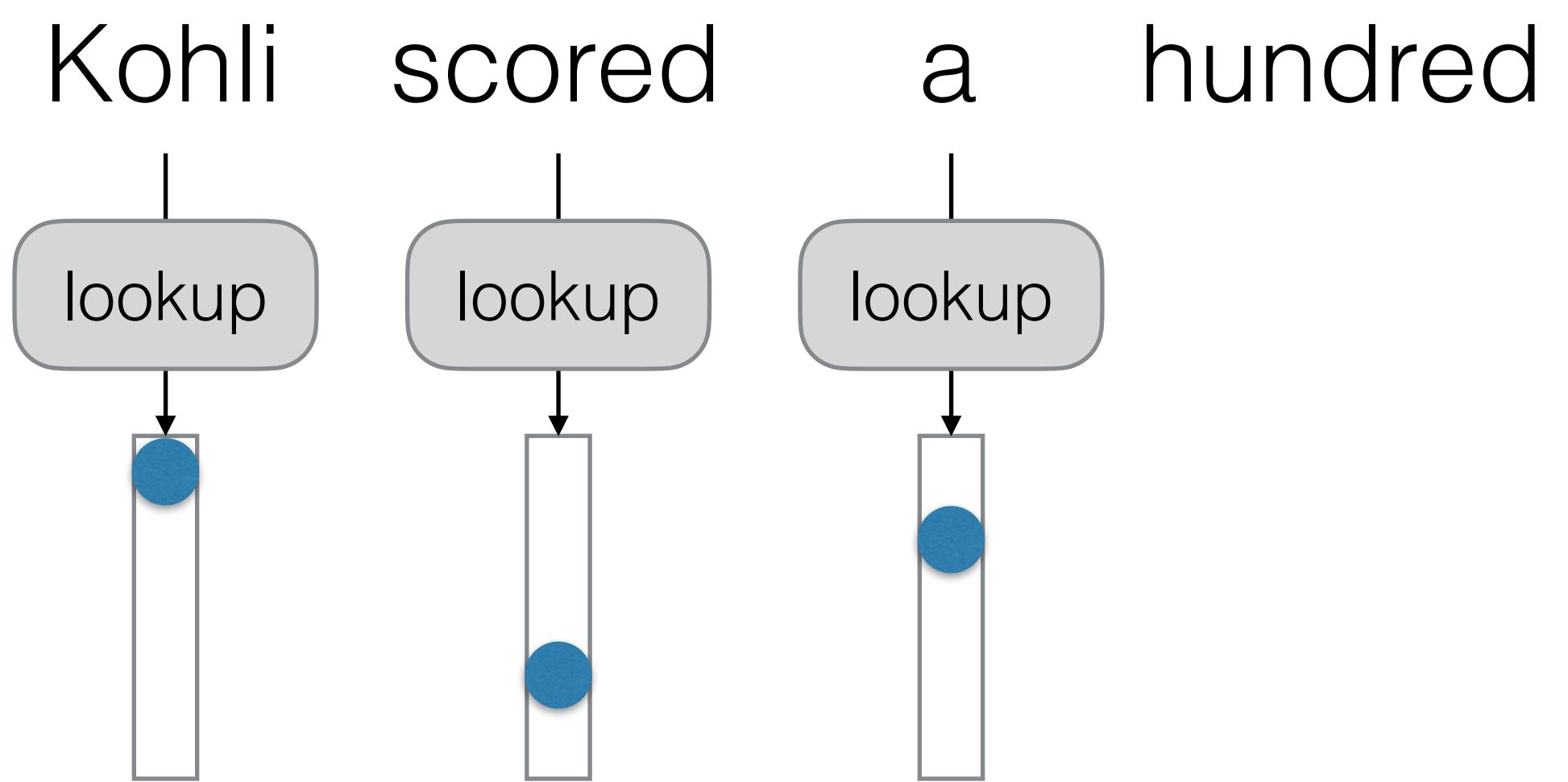
# Computational Graph View



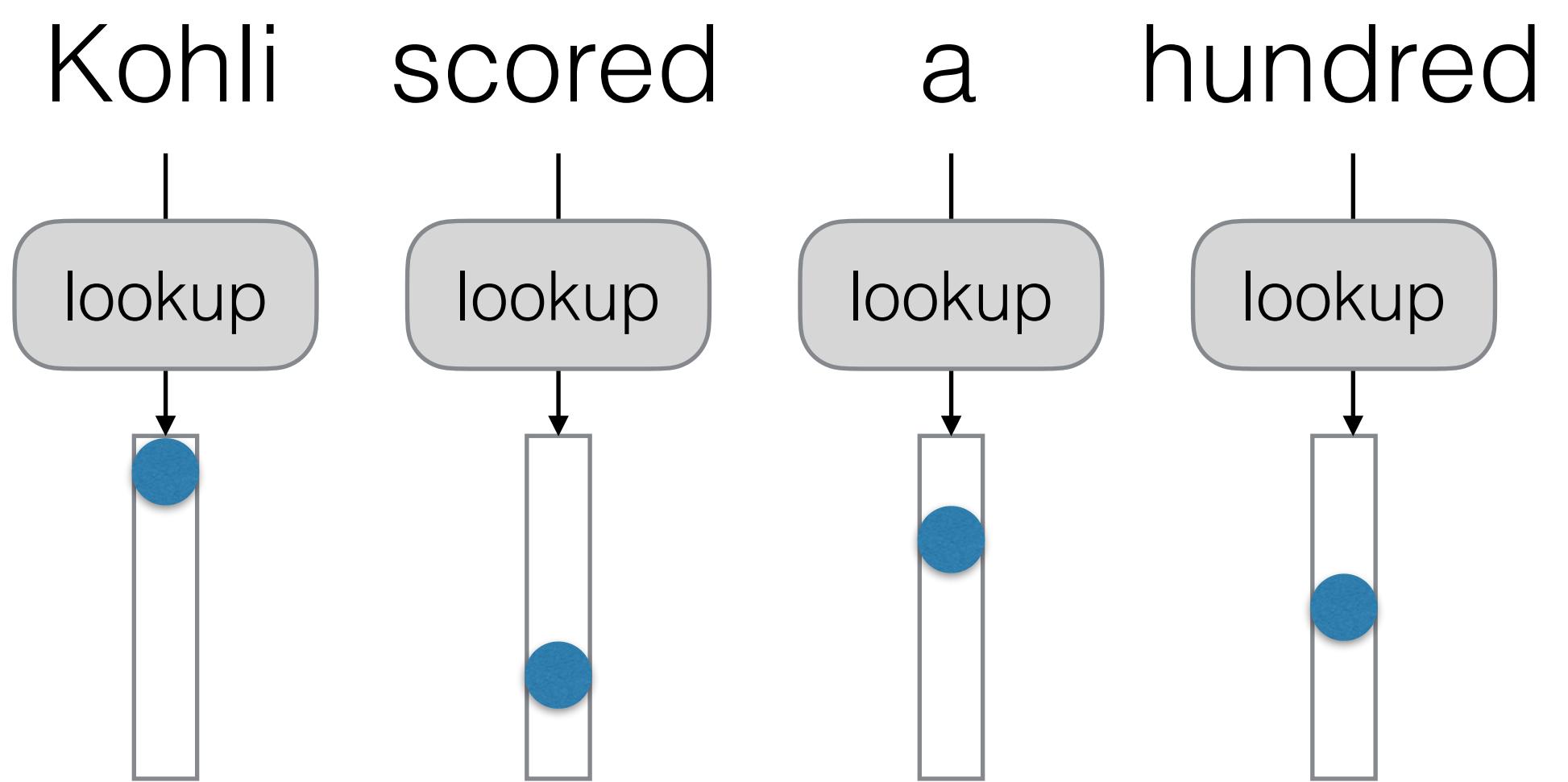
# Computational Graph View



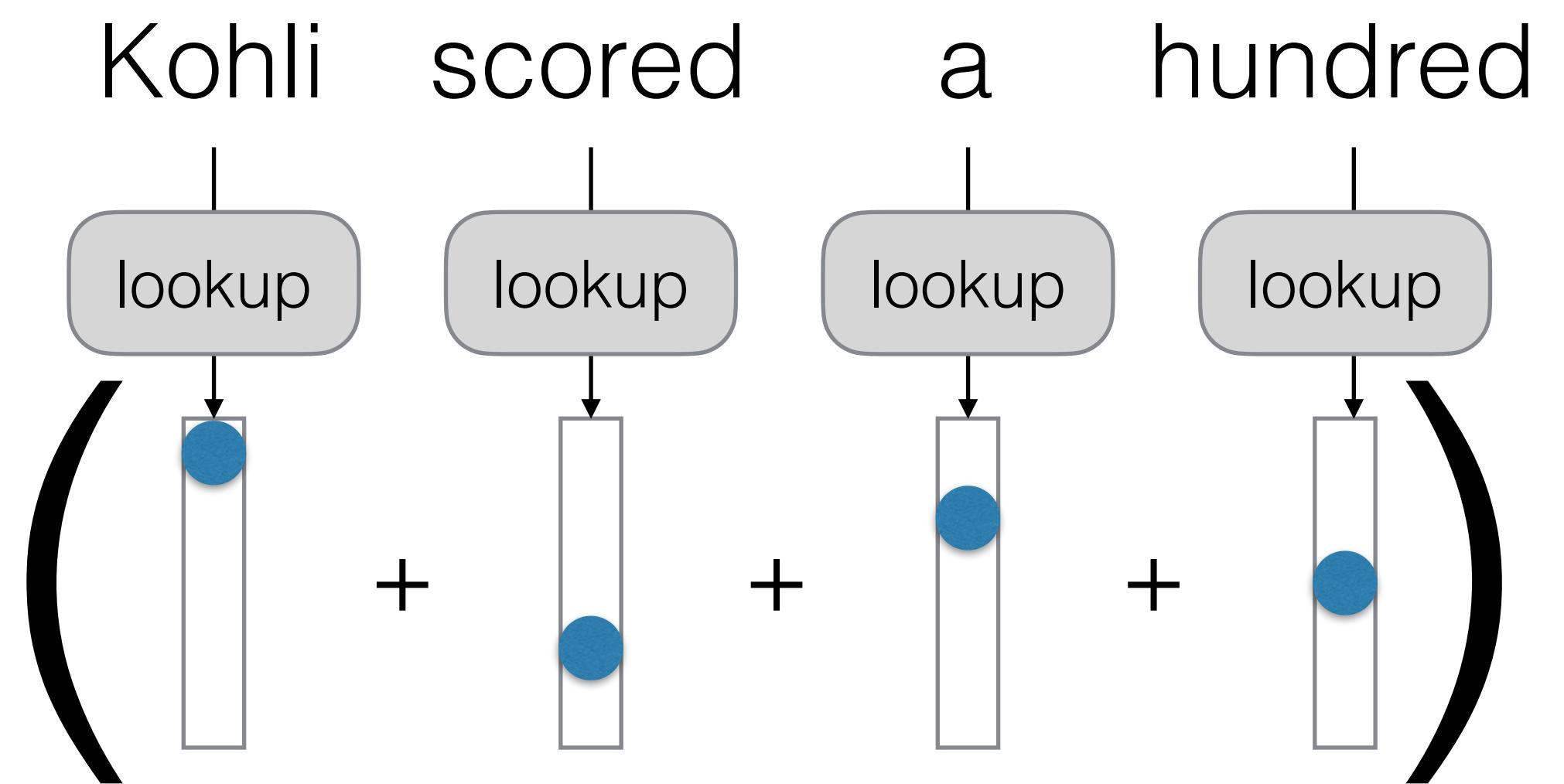
# Computational Graph View



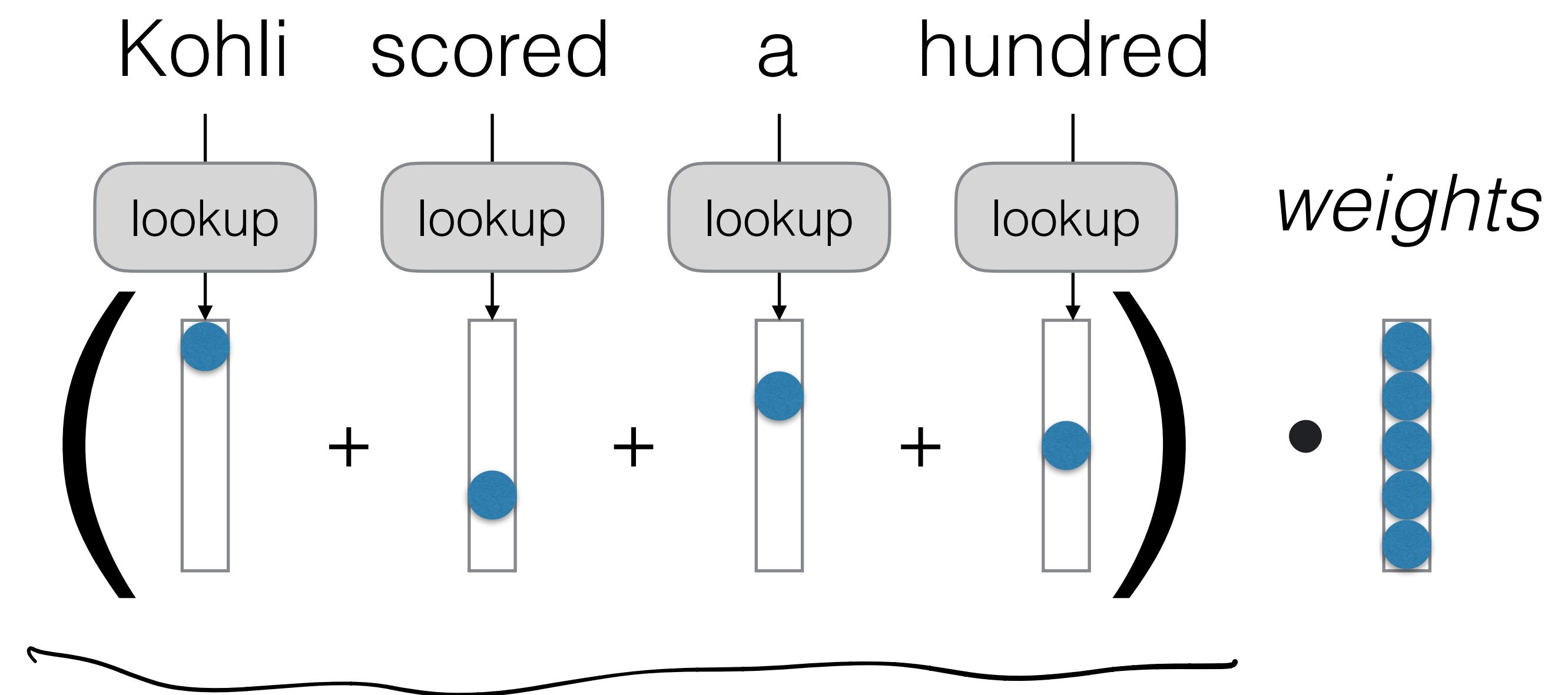
# Computational Graph View



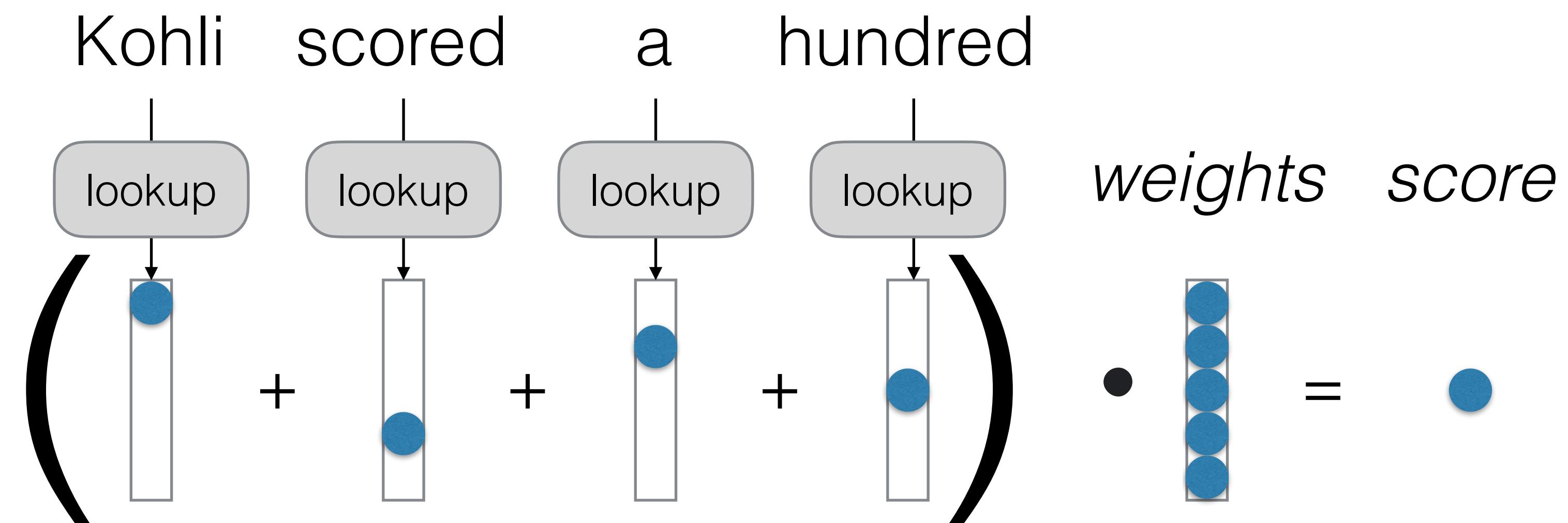
# Computational Graph View



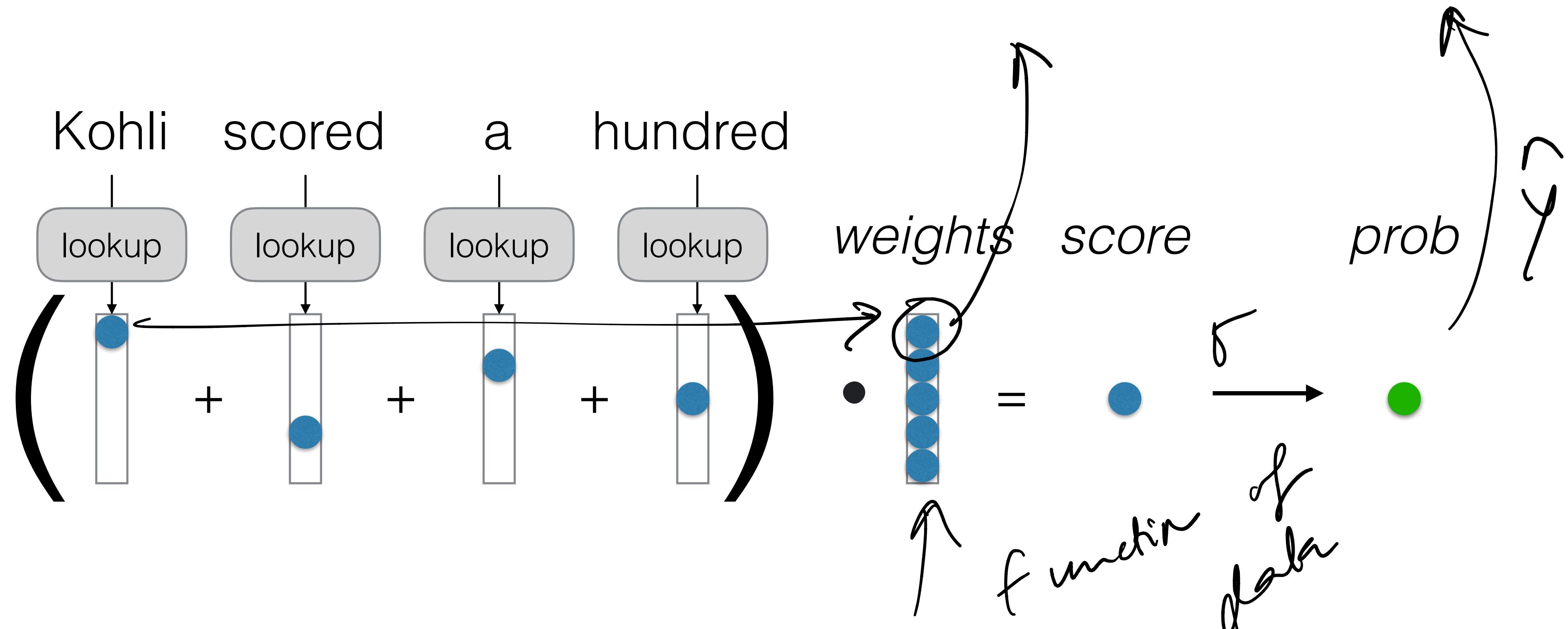
# Computational Graph View



# Computational Graph View



# Computational Graph View



"wooded" "fore"

Questions?

Next class: Word2vec