CMSC 473/673
Natural Language Processing

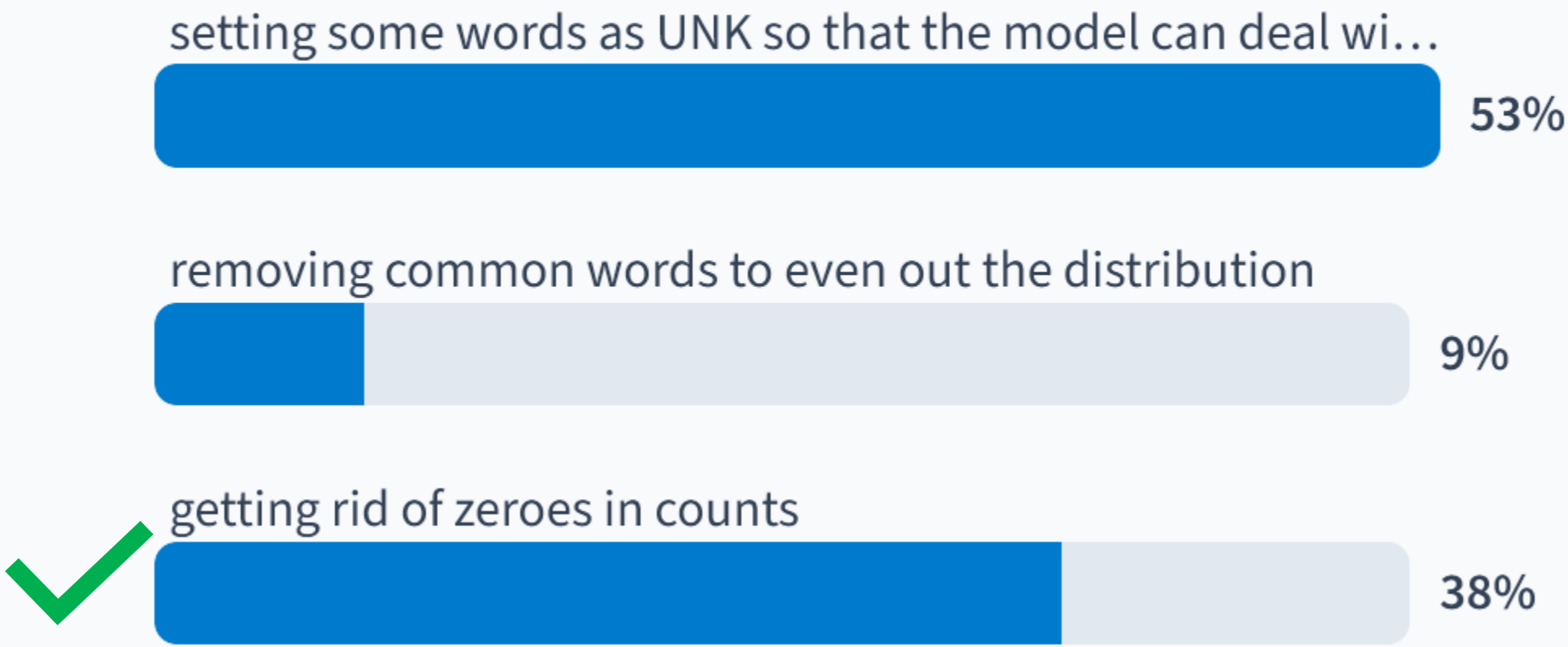
Instructor: Lara J. Martin (she/they)

TA: Duong Ta (he)

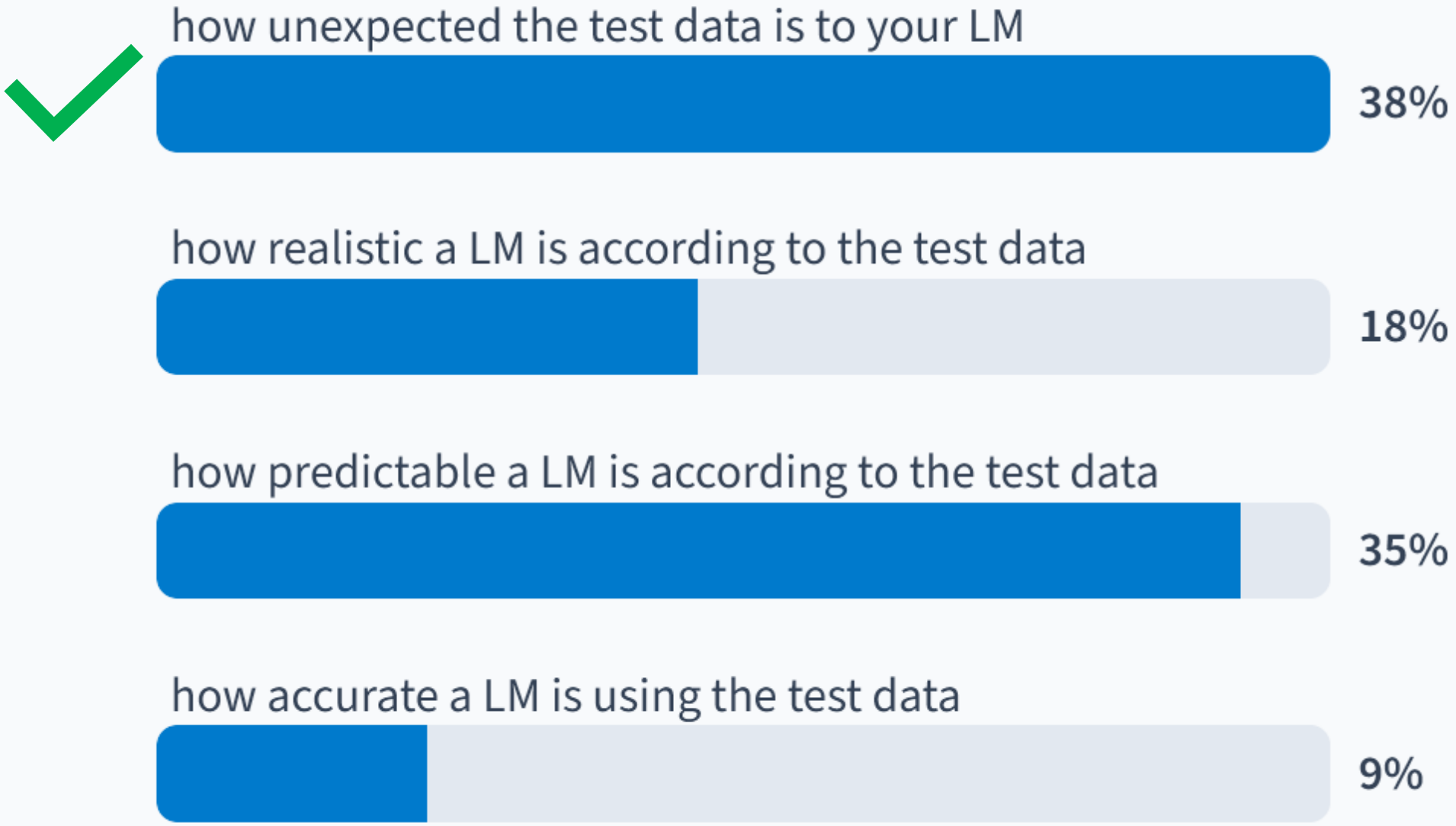
Slides modified from Dr. Frank Ferraro

What is smoothing (in language modeling)?

setting some words as UNK so that the model can deal with out-of-vocabulary words



Perplexity is a measure of...



Review: Add-λ estimation

Other names: Laplace smoothing, Lidstone smoothing

Pretend we saw each word λ more times than we did

Add λ to all the counts

$$p(\mathrm{z}) \cong count(\mathrm{z}) + \lambda$$
$$= \frac{count(\mathrm{z}) + \lambda}{\sum_v (count(\mathrm{v}) + \lambda)}$$

Review: An Extended Trigram Example

The film got a great opening and the film went on to become a hit .

Context: x y	Word (Type): z	Raw Count	Add-1 count	Norm.	Probability p(z | x y)
The film	The	0	1	17 (=1+16*1)	1/17
The film	film	0	1		1/17
The film	got	1	2		2/17
The film	went	0	1		1/17
…					…
The film	OOV	0	1		1/17
The film	EOS	0	1		1/17
…					
a great	great	0	1	17	1/17
a great	opening	1	2		2/17
a great	and	0	1		1/17
a great	the	0	1		1/17
…					

Review: Language Model with Maxent n-grams

$$p_n(\text{📄}|y) = \prod_{i=1}^{M} \text{maxent}(y, x_{i-n+1:i-1}, x_i)$$

label

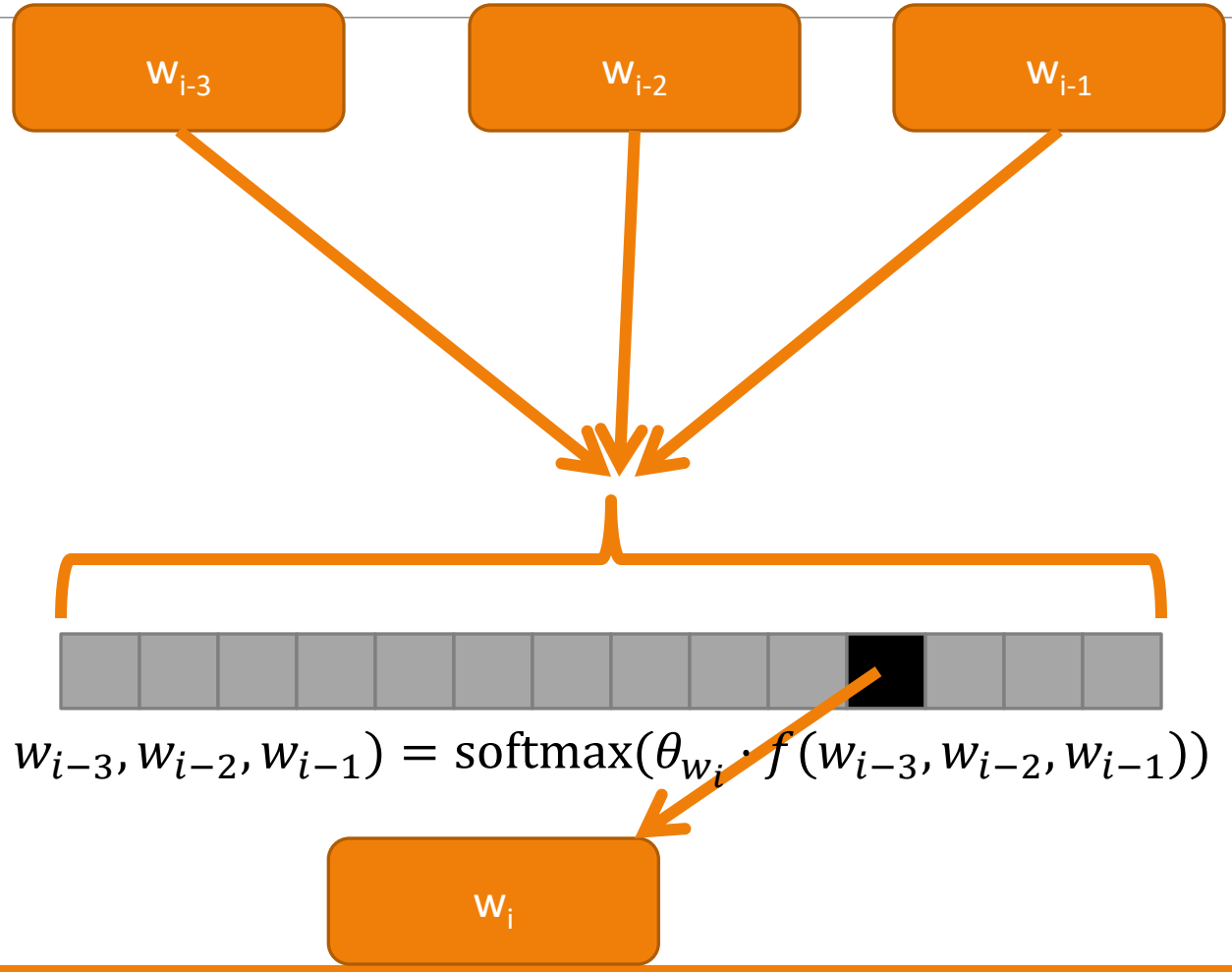
n-gram

$$= \prod_{i=1}^{M} \frac{\exp(\theta_{x_i}^T f(y, x_{i-n+1:i-1}))}{\sum_{x'} \exp(\theta_{x'}^T f(y, x_{i-n+1:i-1}))}$$

Iterate through all possible output vocab types x'---just like in count-based LMs

Review: Maxent Language Models

given some context…



compute beliefs about what is likely…

predict the next word

Review: A Closer Look at Maxent $p($ Won't you please donate? $|$ Primary $)$

This is a *class-based* language model, but incorporate the label into the features

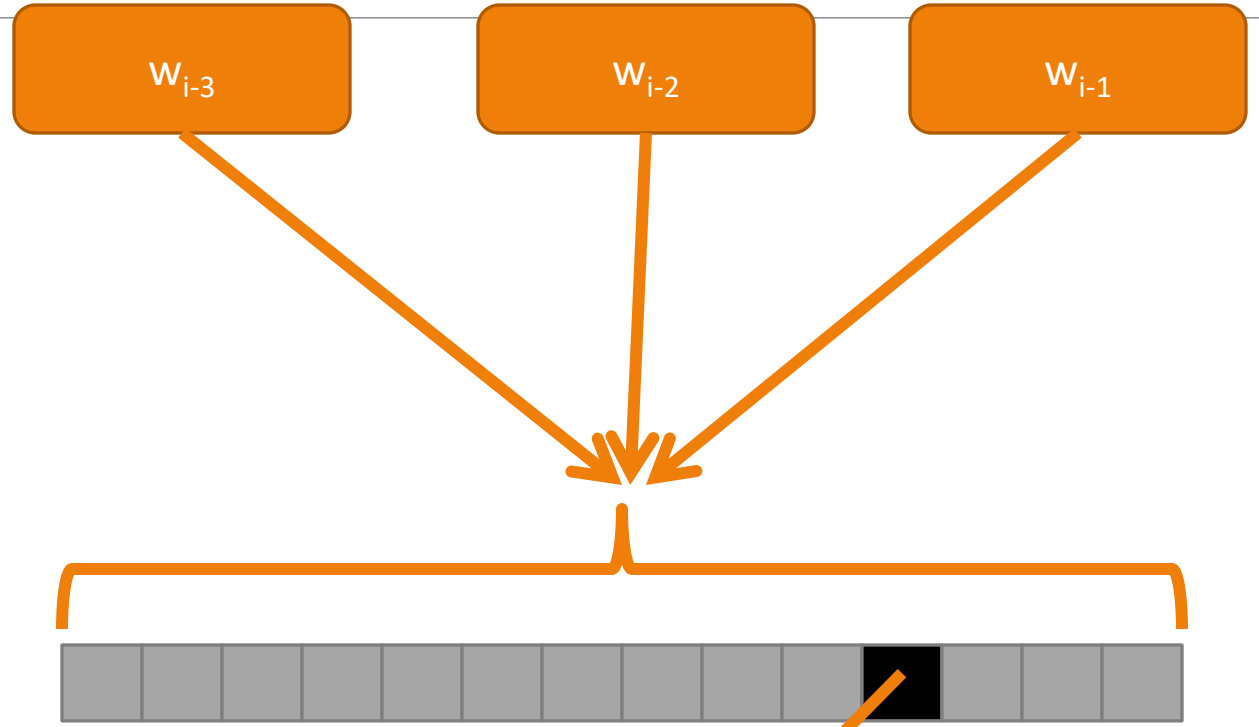
To learn $p($ Won't you please donate? $|$ Class $)$:

Define features f that make use of the specific label Class

Unlike count-based models, you don't *need* "separate" models here

Maxent Language Models

given some context…



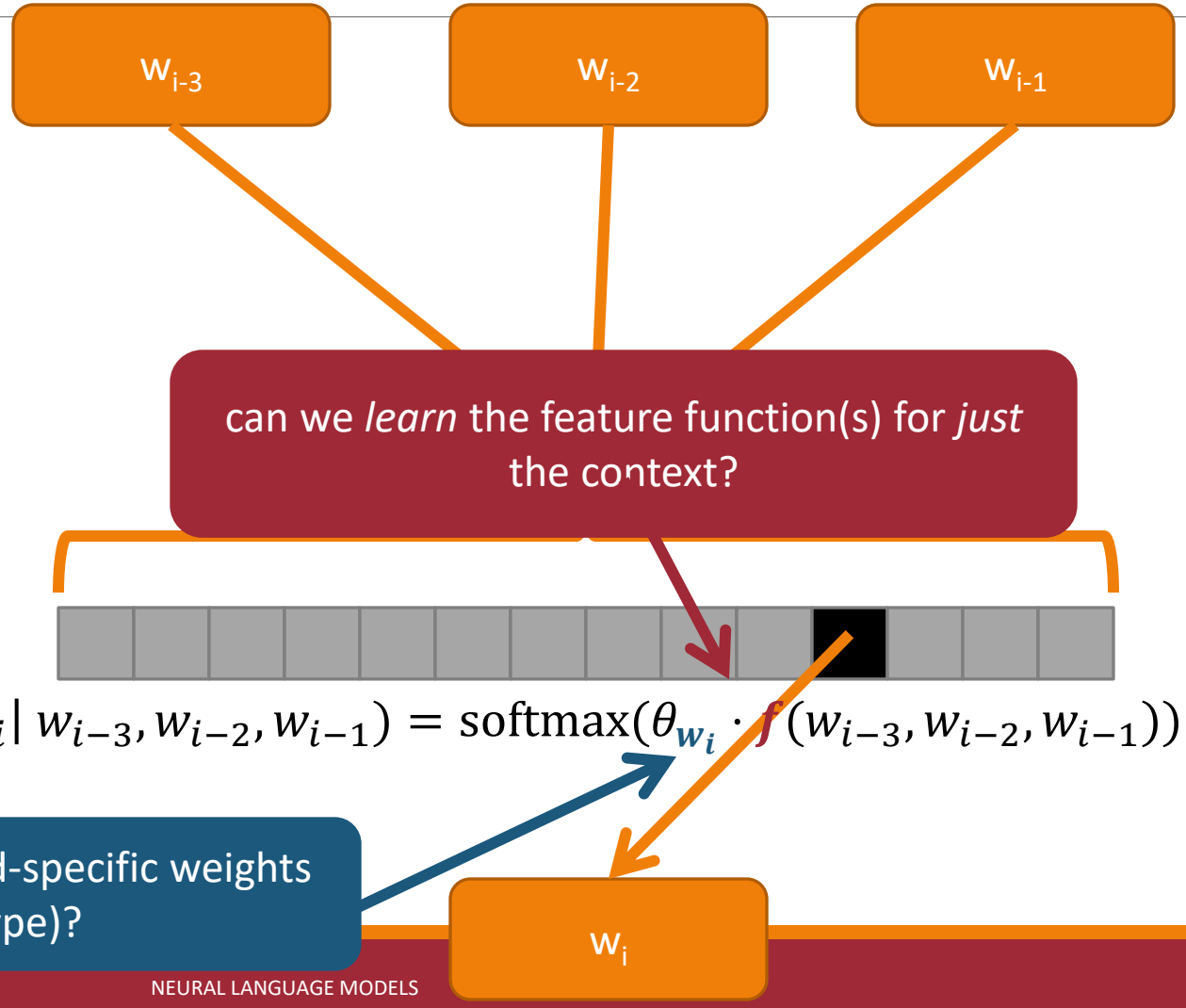
compute beliefs about what is likely…

predict the next word

can we learn word-specific weights (by type)?

Neural Language Models

given some context…



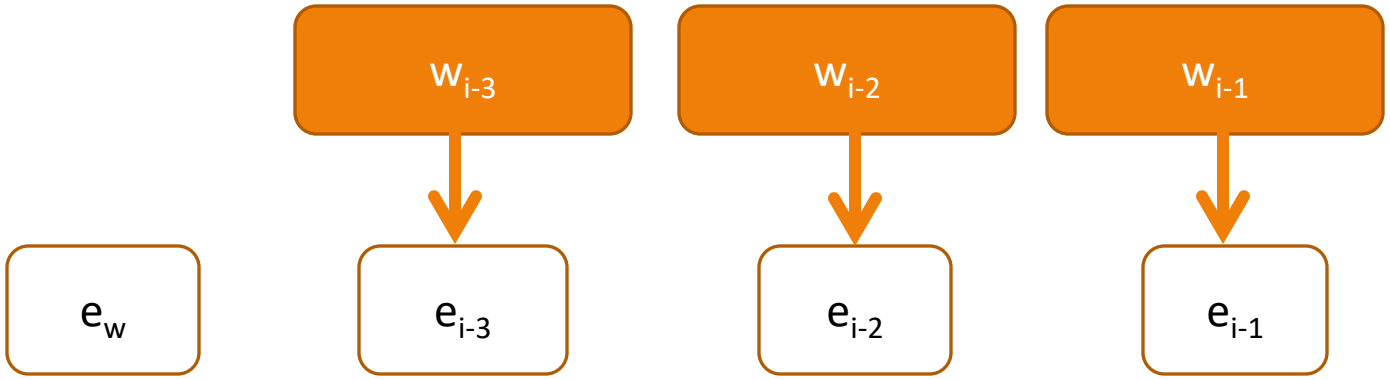
compute beliefs about what is likely…

predict the next word

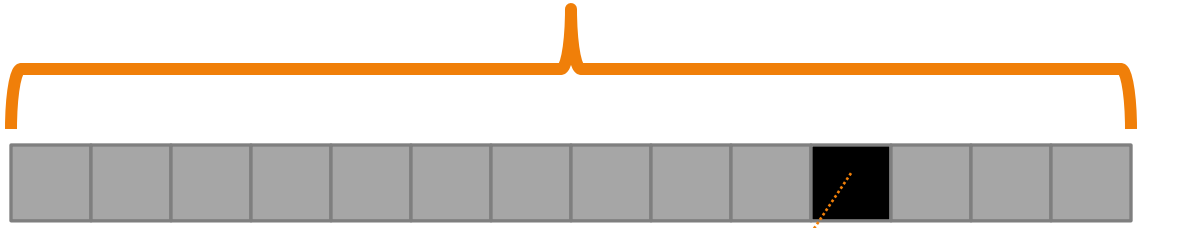
Neural Language Models

given some context…

create/use "distributed representations"…



compute beliefs about what is likely…



$$p(w_i | w_{i-3}, w_{i-2}, w_{i-1}) = \text{softmax}(\theta_{w_i} \cdot \boldsymbol{f}(w_{i-3}, w_{i-2}, w_{i-1}))$$

predict the next word

Neural Language Models

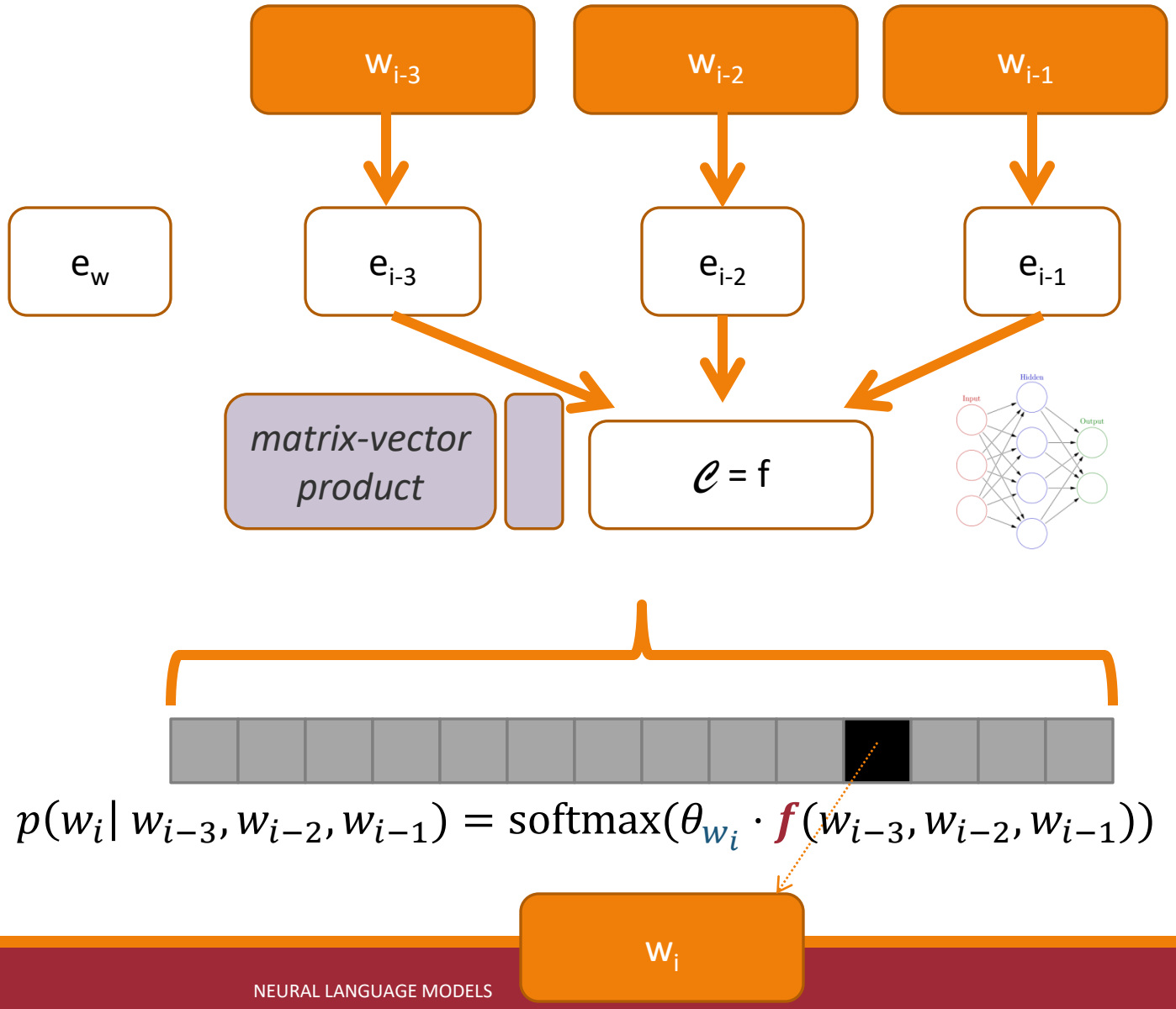
given some context…

create/use "distributed representations"…

combine these representations…

compute beliefs about what is likely…

predict the next word



Neural Language Models

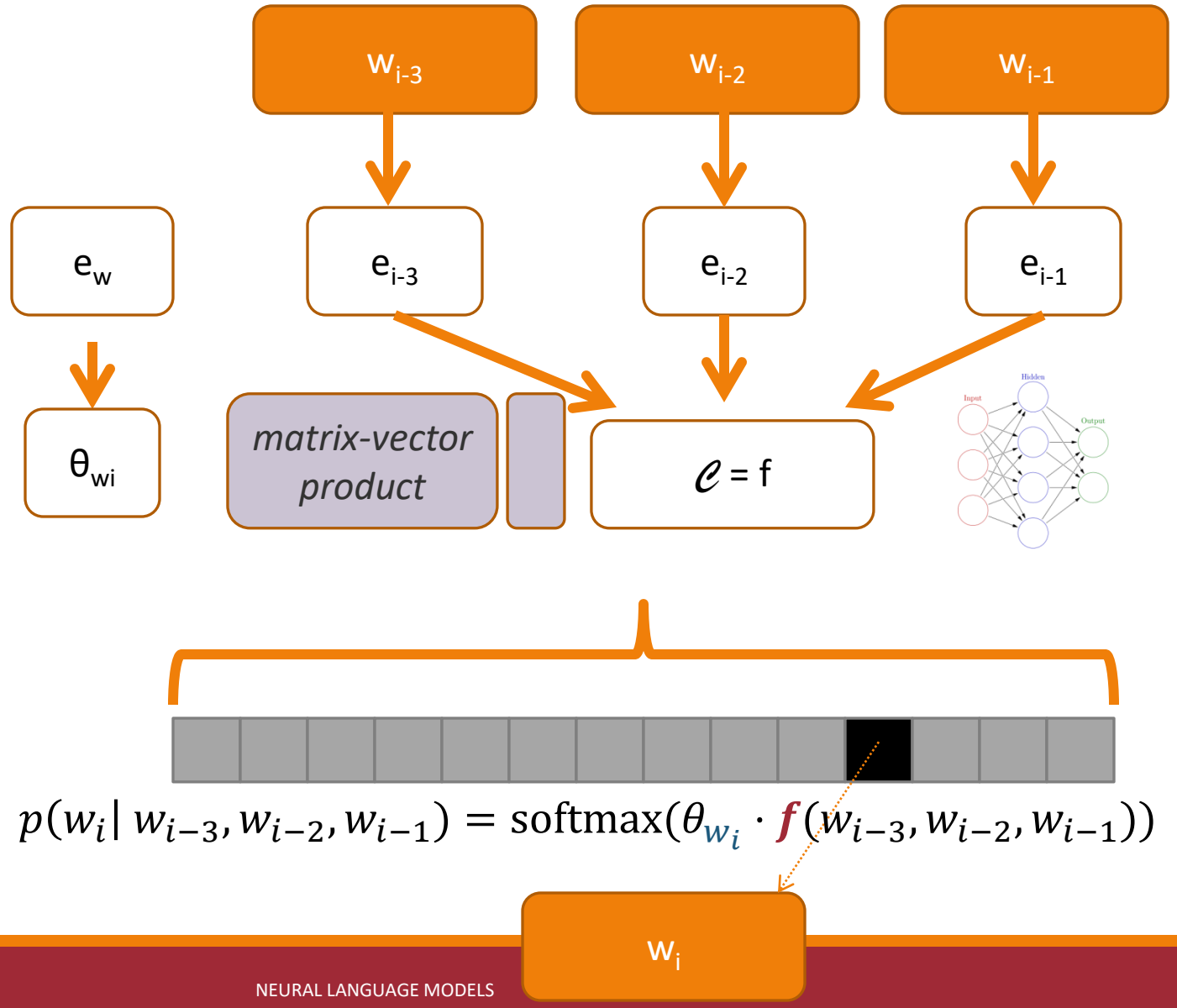
given some context…

create/use "distributed representations"…

combine these representations…

compute beliefs about what is likely…

predict the next word



Neural Language Models

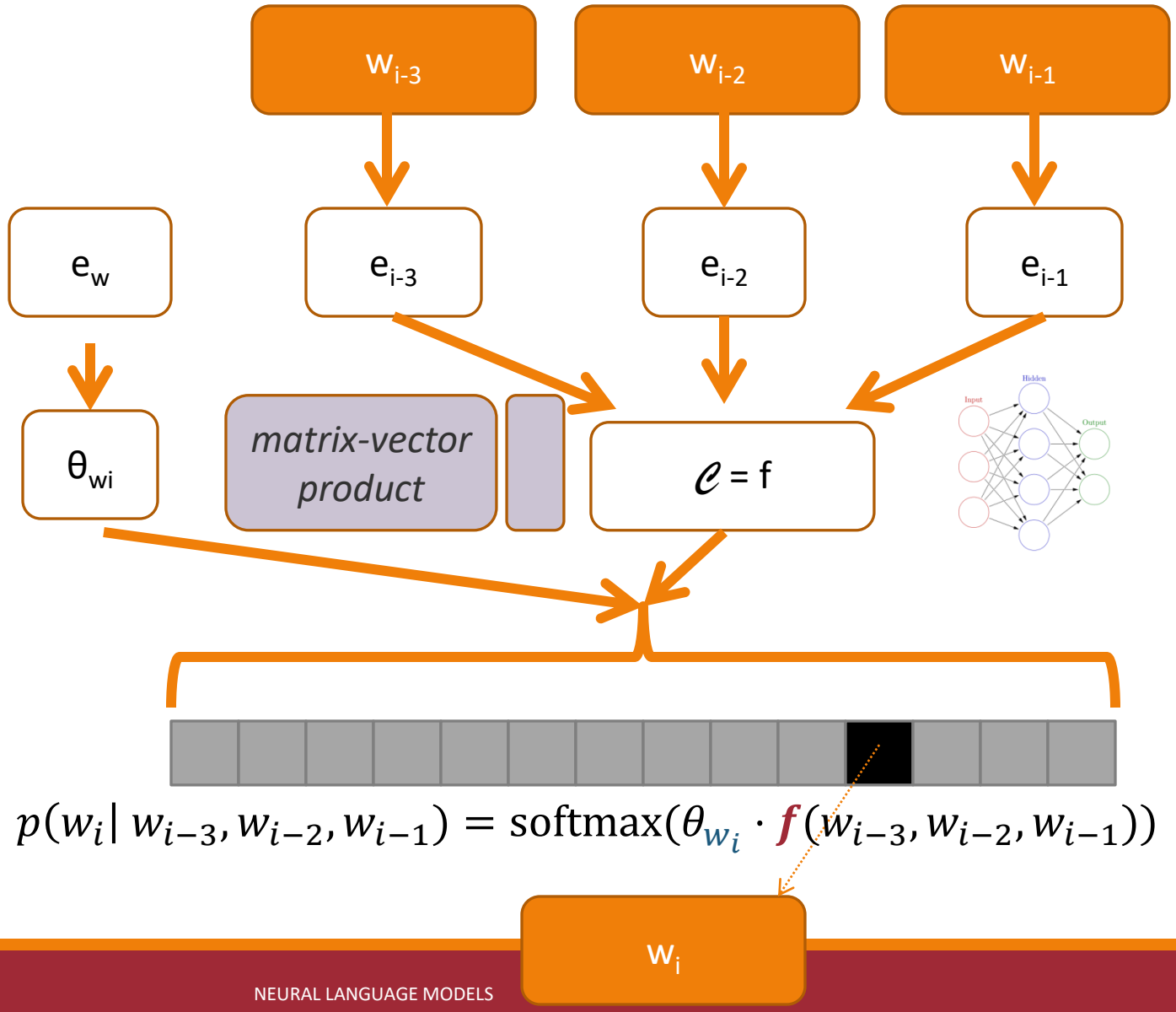
given some context…

create/use "distributed representations"…

combine these representations…

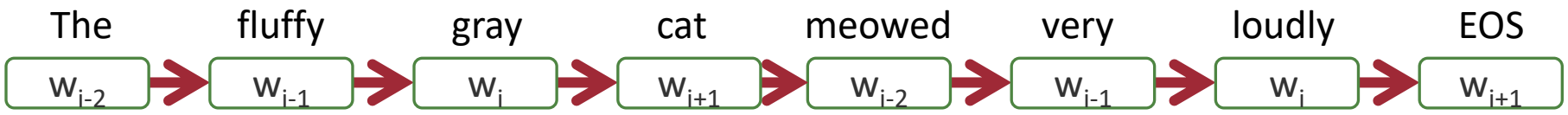
compute beliefs about what is likely…

predict the next word



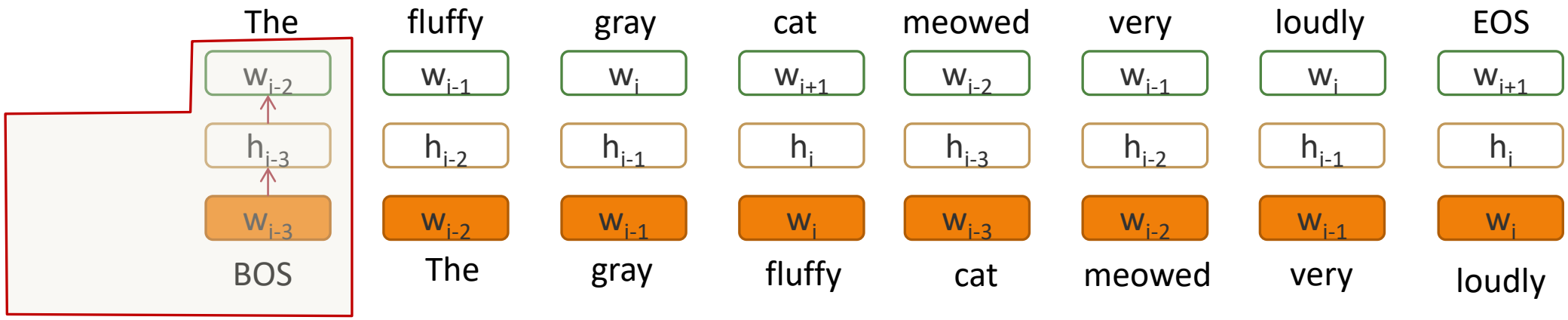
A Neural N-Gram Model

The fluffy gray cat meowed very loudly



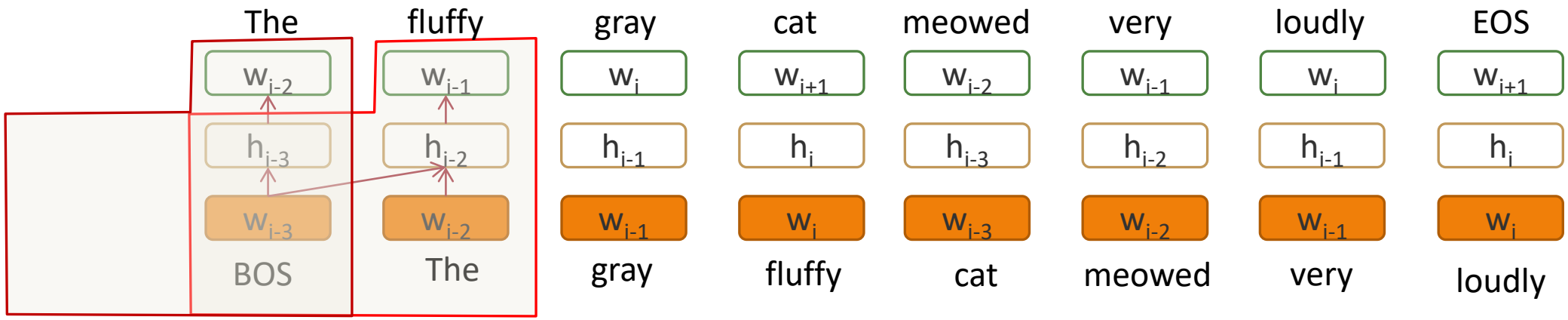
A Neural N-Gram Model (N=3)

The fluffy gray cat meowed very loudly



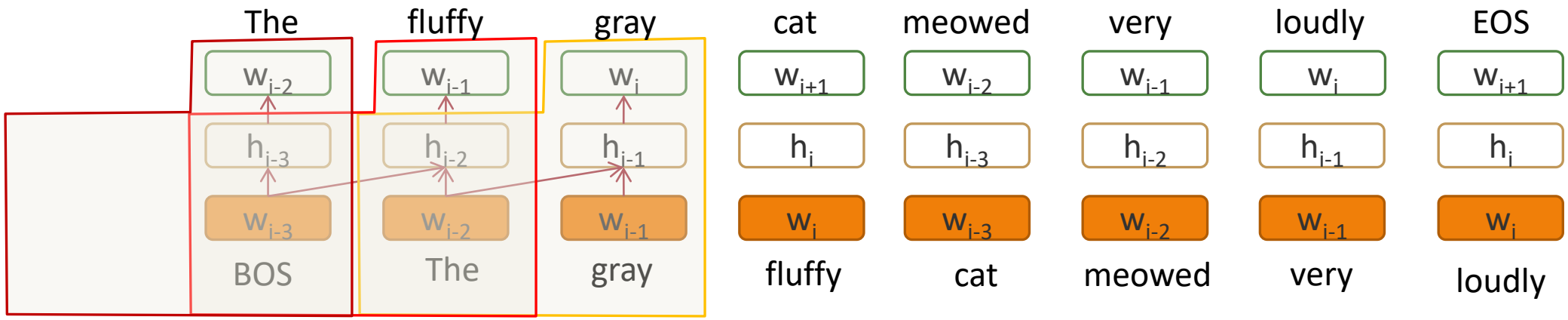
A Neural N-Gram Model (N=3)

The fluffy gray cat meowed very loudly



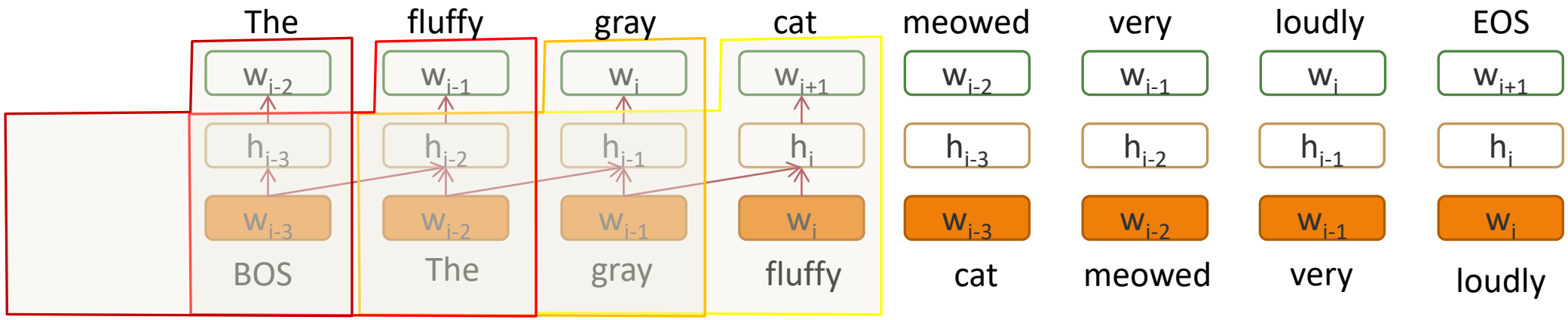
A Neural N-Gram Model (N=3)

The fluffy gray cat meowed very loudly



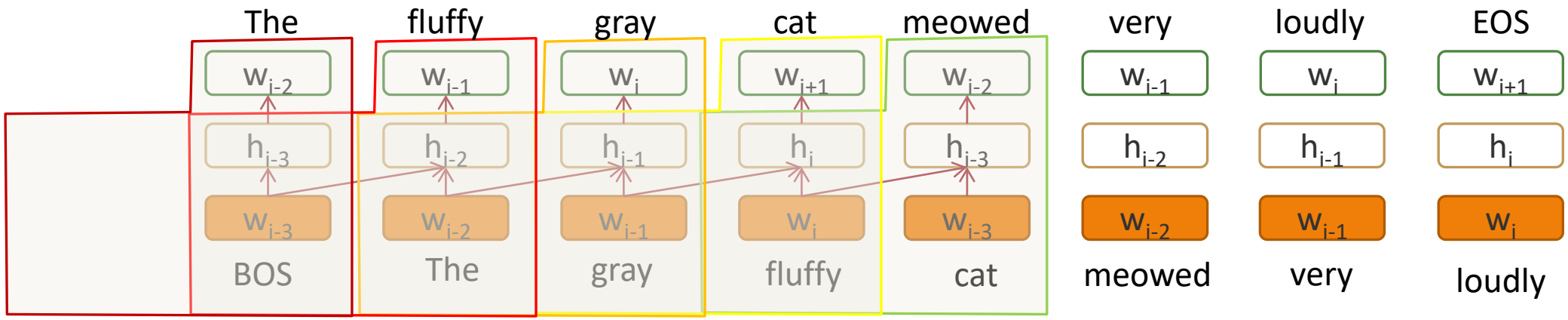
A Neural N-Gram Model (N=3)

The fluffy gray cat meowed very loudly



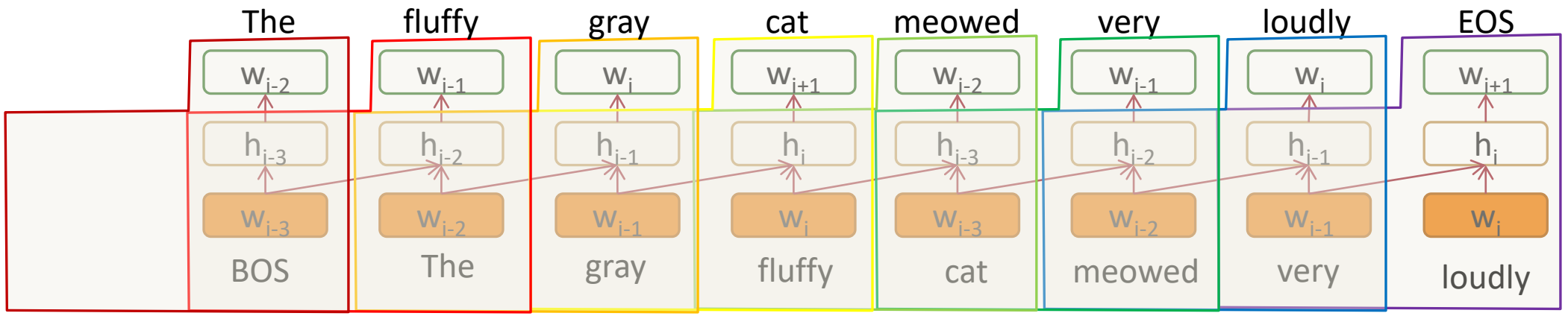
A Neural N-Gram Model (N=3)

The fluffy gray cat meowed very loudly

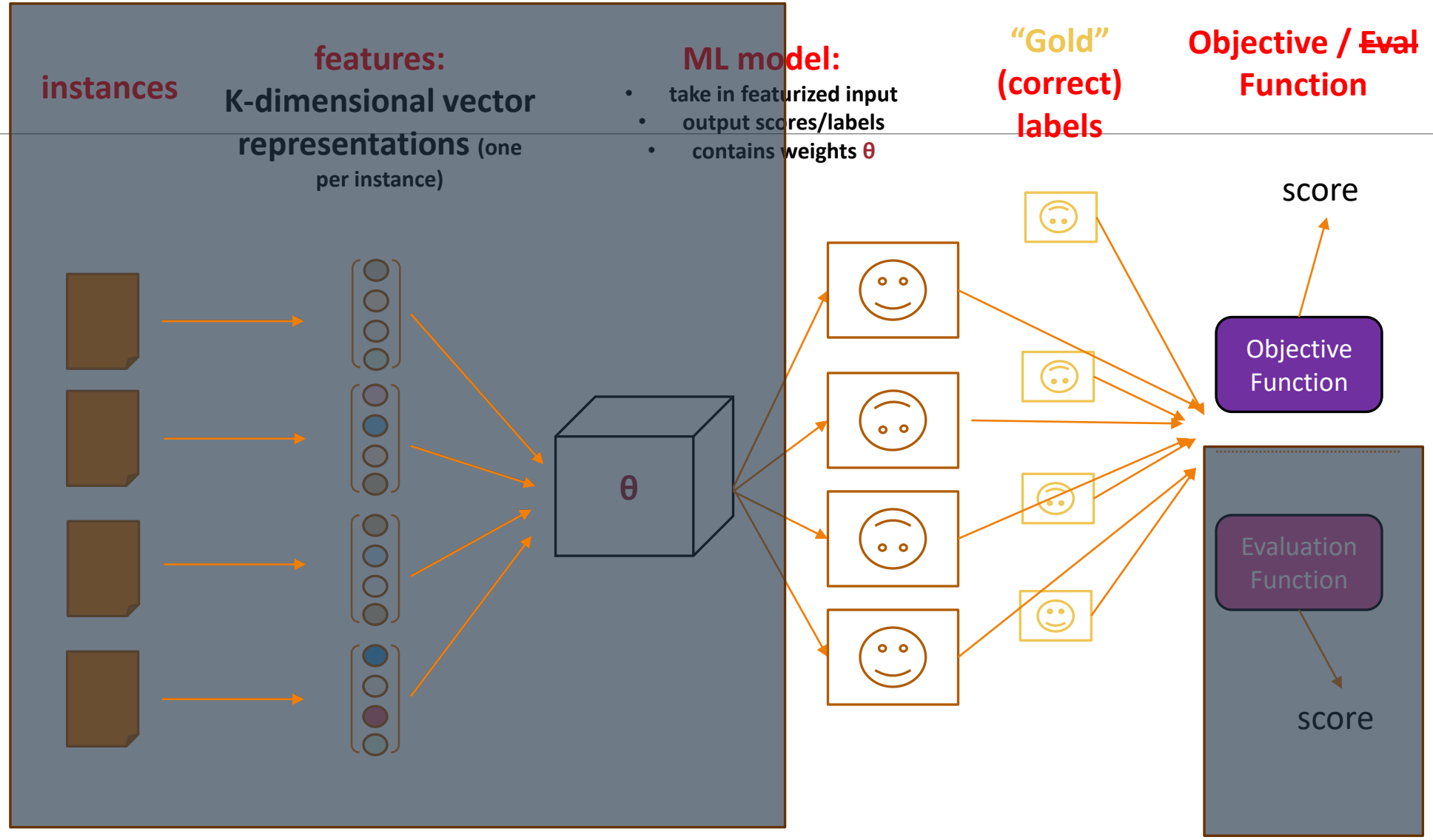


A Neural N-Gram Model (N=3)

The fluffy gray cat meowed very loudly



Defining the Objective



Review: *Maximize* Log-Likelihood (Classification)

Original maxent equation

$$\frac{\exp(\theta_y^T f(x))}{\sum_{y'} \exp(\theta_{y'}^T f(x))}$$

$$\log \prod_i p_\theta(y_i|x_i) = \sum_i \log p_\theta(y_i|x_i)$$

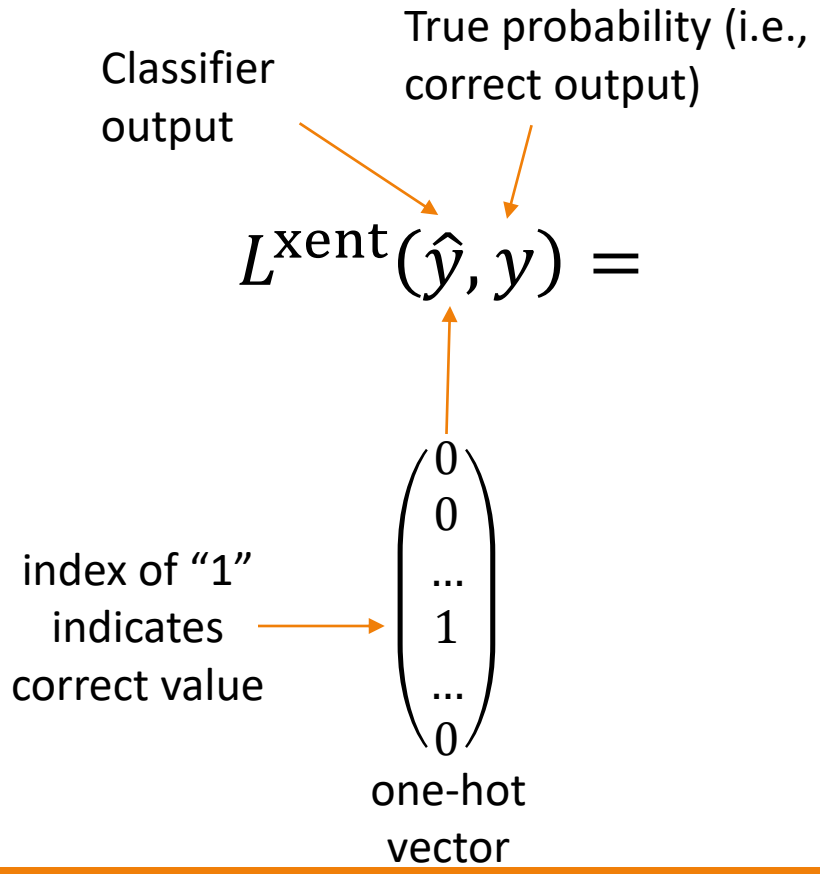
Differentiating this becomes nicer (even though Z depends on θ)

Inverse of exp
log(exp(x)) = x

$$= \sum_i \theta_{y_i}^T f(x_i) - \log Z(x_i)$$

$$= F(\theta)$$

Review: *Minimize* Cross Entropy Loss



Cross entropy:
How much $\hat{y}$ differs from the true y

objective is convex
(when f(x) is not learned)

“A Neural Probabilistic Language Model,” Bengio et al. (2003)

BASELINES

LM Name	N-gram	Params.	Test PPL
Interpolation	3	---	336
Kneser-Ney backoff	3	---	323
Kneser-Ney backoff	5	---	321
Class-based backoff	3	500 classes	312
Class-based backoff	5	500 classes	312

“A Neural Probabilistic Language Model,” Bengio et al. (2003)

BASELINES

LM Name	N-gram	Params.	Test PPL
Interpolation	3	---	336
Kneser-Ney backoff	3	---	323
Kneser-Ney backoff	5	---	321
Class-based backoff	3	500 classes	312
Class-based backoff	5	500 classes	312

NPLM

N-gram	Word Vector Dim.	Hidden Dim.	Mix with non-neural LM	PPL
5	60	50	No	268
5	60	50	Yes	257
5	30	100	No	276
5	30	100	Yes	252

“we were not able to see signs of over- fitting (on the validation set), possibly because we ran only 5 epochs (over 3 weeks using 40 CPUs)” (Sect. 4.2)

A Closer Look at Neural $p($ Won't you please donate? $|$ Primary $)$

This is a *class-based* language model, but incorporate the label into the *embedding representation*

To learn $p($ Won't you please donate? $|$ Class $)$:

Define an embedding method that makes use of the specific label Class

Unlike count-based models, you don't *need* "separate" models here

LM Comparison for $p($  $|$  $)$

N-GRAM/COUNT-BASED

Class-specific

MAXENT/LR

Class-based

Uses features

NEURAL

Class-based

Uses *embedded* features