

*Лекция 2*

# **Языковое моделирование**

**Марковские модели со сглаживанием  
Нейросетевые языковые модели**

*Потапенко Анна Александровна*

*12 сентября 2018*

# Языковое моделирование

This is the ...

house

rat

did

malt

Какова вероятность следующего слова?

$$p(\textit{house} \mid \textit{this is the}) = ?$$

# Игрушечный корпус

*This is the house that Jack built.  
This is the malt  
That lay in the house that Jack built.  
This is the rat,  
That ate the malt  
That lay in the house that Jack built.  
This is the cat,  
That killed the rat,  
That ate the malt  
That lay in the house that Jack built.*

$p(\text{house} \mid \text{this is the}) =$

# Игрушечный корпус

*This is the house that Jack built.*

*This is the malt*

*That lay in the house that Jack built.*

*This is the rat,*

*That ate the malt*

*That lay in the house that Jack built.*

*This is the cat,*

*That killed the rat,*

*That ate the malt*

*That lay in the house that Jack built.*

$p(\text{house} \mid \text{this is the}) =$

# Игрушечный корпус

*This is the house that Jack built.*

*This is the malt*

*That lay in the house that Jack built.*

*This is the rat,*

*That ate the malt*

*That lay in the house that Jack built.*

*This is the cat,*

*That killed the rat,*

*That ate the malt*

*That lay in the house that Jack built.*

$p(\text{house} \mid \text{this is the}) =$

# Игрушечный корпус

*This is the house that Jack built.*

*This is the malt*

*That lay in the house that Jack built.*

*This is the rat,*

*That ate the malt*

*That lay in the house that Jack built.*

*This is the cat,*

*That killed the rat,*

*That ate the malt*

*That lay in the house that Jack built.*

$$p(\text{house} \mid \text{this is the}) = \frac{c(\text{this is the house})}{c(\text{this is the ...})} = \frac{1}{4}$$

# Игрушечный корпус

*This is the house that Jack built.*

*This is the malt*

*That lay in the house that Jack built.*

*This is the rat,*

*That ate the malt*

*That lay in the house that Jack built.*

*This is the cat,*

*That killed the rat,*

*That ate the malt*

*That lay in the house that Jack built.*

4-граммы

$$p(\text{house} \mid \text{this is the}) = \frac{c(\text{this is the house})}{c(\text{this is the ...})} = \frac{1}{4}$$

# Игрушечный корпус

*This is the house **that Jack** built.*

*This is the malt*

***That** lay in the house **that Jack** built.*

*This is the rat,*

***That** ate the malt*

***That** lay in the house **that Jack** built.*

*This is the cat,*

***That** killed the rat,*

***That** ate the malt*

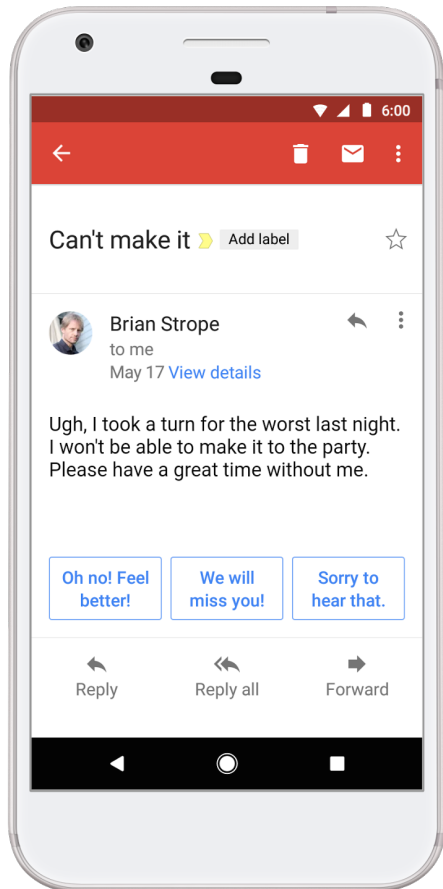
***That** lay in the house **that Jack** built.*

биграммы

$$p(\text{Jack} \mid \text{that}) = \frac{c(\text{that Jack})}{c(\text{that...})} = \frac{4}{10}$$



# Где нужны языковые модели?



- Исправление опечаток
- Автоматические ответы
- Машинный перевод
- Распознавание речи
- Распознавание рукописного текста
- ...

# Языковое моделирование

This is the ...

house

rat

did

malt

Какова вероятность всей последовательности?

$$p(\textit{this is the house}) = ?$$

# Немного математики

Дана последовательность слов:

$$\mathbf{w} = (w_1 w_2 w_3 \dots w_k)$$

# Немного математики

Дана последовательность слов:

$$\mathbf{w} = (w_1 w_2 w_3 \dots w_k)$$

- **Правило условной вероятности:**

$$p(\mathbf{w}) = p(w_1)p(w_2|w_1) \dots p(w_k|w_1 \dots w_{k-1})$$

# Немного математики

Дана последовательность слов:

$$\mathbf{w} = (w_1 w_2 w_3 \dots w_k)$$

- **Правило условной вероятности:**

$$p(\mathbf{w}) = p(w_1)p(w_2|w_1) \dots p(w_k | \cancel{w_1 \dots w_{k-1}})$$

- **Предположение Маркова:**

$$p(w_i | w_1 \dots w_{i-1}) = p(w_i | w_{i-n+1} \dots w_{i-1})$$

# Биграммная языковая модель

Для  $n = 2$ :

$$p(\mathbf{w}) = p(w_1)p(w_2|w_1) \dots p(w_k|w_{k-1})$$

# Биграммная языковая модель

Для  $n = 2$ :

$$p(\mathbf{w}) = p(w_1)p(w_2|w_1) \dots p(w_k|w_{k-1})$$

**Корпус:**

*This is the malt*

*That lay in the house that Jack built.*

$$p(\text{this is the house}) = p(\text{this}) p(\text{is} | \text{this}) p(\text{the} | \text{is}) p(\text{house} | \text{the})$$

# Биграммная языковая модель

Для  $n = 2$ :

$$p(\mathbf{w}) = p(w_1)p(w_2|w_1) \dots p(w_k|w_{k-1})$$

Корпус:

*This is the malt*

*That lay in the house that Jack built.*

$\frac{1}{12}$     1    1     $\frac{1}{2}$

$$p(\text{this is the house}) = p(\text{this}) p(\text{is} | \text{this}) p(\text{the} | \text{is}) p(\text{house} | \text{the})$$



# Биграммная языковая модель

Для  $n = 2$ :

$$p(\mathbf{w}) = \cancel{p(w_1)} p(w_2|w_1) \dots p(w_k|w_{k-1}) \\ p(w_1|start)$$

Корпус:

*This is the malt*

*That lay in the house that Jack built.*

$$p(\text{this is the house}) = \overset{1/2}{p(\text{this})} \overset{1}{p(\text{is}|\text{ this})} \overset{1}{p(\text{the}|\text{ is})} \overset{1/2}{p(\text{house}|\text{ the})}$$

# Биграммная языковая модель

Для  $n = 2$ :

$$p(\mathbf{w}) = \cancel{p(w_1)} p(w_2|w_1) \dots p(w_k|w_{k-1}) \\ p(w_1|start)$$

# Биграммная языковая модель

Для  $n = 2$ :

$$p(\mathbf{w}) = \cancel{p(w_1)} p(w_2|w_1) \dots p(w_k|w_{k-1}) \\ p(w_1|start)$$

**Неверная нормировка: отдельно по каждой длине!**

$$p(this) + p(that) = 1.0$$

$$p(this this) + p(this is) + \dots + p(built built) = 1.0$$

...

# Биграммная языковая модель

Для  $n = 2$ :

$$p(\mathbf{w}) = \cancel{p(w_1)} p(w_2|w_1) \dots p(w_k|w_{k-1})$$

$p(\text{end} | w_k)$

**Неверная нормировка: отдельно по каждой длине!**

$$p(\text{this}) + p(\text{that}) = 1.0$$

$$p(\text{this this}) + p(\text{this is}) + \dots + p(\text{built built}) = 1.0$$

...

# Проверим нормировку

\_ dog \_

\_ dog cat tiger \_

\_ cat dog cat \_

$p(\textit{cat dog cat}) =$



# Проверим нормировку

\_ dog \_

\_ dog cat tiger \_

\_ cat dog cat \_

$$p(\textit{cat dog cat}) = p(\textit{cat} \mid \_)$$

*dog*

*cat*

# Проверим нормировку

\_ dog \_

\_ dog cat tiger \_

\_ cat dog cat \_

$$p(\textit{cat dog cat}) = p(\textit{cat} \mid \_)$$

*dog*

*cat*

# Проверим нормировку

\_ dog \_

\_ dog **cat tiger** \_

\_ **cat dog cat** \_

$$p(\text{cat dog cat}) = p(\text{cat} \mid \_) p(\text{dog} \mid \text{cat})$$

<i>dog</i>	<i>cat tiger</i>
	<i>cat dog</i>
	<i>cat</i> _



# Проверим нормировку

\_ dog \_

\_ dog cat tiger \_

\_ cat dog cat \_

$$p(\textit{cat dog cat}) = p(\textit{cat} \mid \_) p(\textit{dog} \mid \textit{cat})$$

<i>dog</i>	<i>cat tiger</i>
	<i>cat dog</i>
	<i>cat</i> _

# Проверим нормировку

\_ dog \_

\_ dog cat tiger \_

\_ cat dog cat \_

$$p(\text{cat dog cat}) = p(\text{cat} \mid \_) p(\text{dog} \mid \text{cat}) p(\text{cat} \mid \text{dog})$$

<i>dog</i>	<i>cat tiger</i>	
	<i>cat dog cat</i>	<i>cat dog_</i>
	<i>cat</i> _	

# Проверим нормировку

\_ dog \_

\_ dog cat tiger \_

\_ cat dog cat \_

$$p(\textit{cat dog cat}) = p(\textit{cat} \mid \_) p(\textit{dog} \mid \textit{cat}) p(\textit{cat} \mid \textit{dog})$$

<i>dog</i>	<i>cat tiger</i>	
	<i>cat dog cat</i>	<i>cat dog_</i>
	<i>cat</i> _	

# Проверим нормировку

\_ dog \_

\_ dog **cat tiger** \_

\_ **cat dog cat** \_

$$p(\text{cat dog cat}) = p(\text{cat} \mid \_) p(\text{dog} \mid \text{cat}) p(\text{cat} \mid \text{dog}) p(\_ \mid \text{cat})$$

<i>dog</i>	<i>cat tiger</i>
	<i>cat dog cat tiger</i>   <i>cat dog _</i>
	<i>cat dog cat dog</i>
	<i>cat dog cat _</i>
	<i>cat _</i>

# Проверим нормировку

\_ dog \_

\_ dog cat tiger \_

\_ cat dog cat \_

$$p(\text{cat dog cat}) = p(\text{cat} \mid \_) p(\text{dog} \mid \text{cat}) p(\text{cat} \mid \text{dog}) p(\_ \mid \text{cat})$$

<i>dog</i>	<i>cat tiger</i>
	<i>cat dog cat tiger</i>   <i>cat dog _</i>
	<i>cat dog cat dog</i>
	<i>cat dog cat _</i>
	<i>cat _</i>

# Обобщение: n-граммная модель


$$\mathbf{w} = (w_1 w_2 w_3 \dots w_k)$$

**Биграммная модель:**

$$p(\mathbf{w}) = \prod_{i=1}^{k+1} p(w_i | w_{i-1})$$

**N-граммная модель:**

$$p(\mathbf{w}) = \prod_{i=1}^{k+1} p(w_i | w_{i-n+1}^{i-1})$$

$(w_{i-n+1}, \dots, w_{i-1})$   


# Обучение

**Максимизация логарифма правдоподобия:**

$$\log p(\mathbf{w}_{\text{train}}) = \sum_{i=1}^{N+1} \log p(w_i | w_{i-n+1}^{i-1}) \rightarrow \max$$

**Оценки на параметры:**

$$p(w_i | w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^i)}{\sum_{w_i} c(w_{i-n+1}^i)} = \frac{c(w_{i-n+1}^i)}{c(w_{i-n+1}^{i-1})}$$

# Генерация Шекспира

## Униграммы:

*To him swallowed confess hear both. Which. Of save on trail  
for are ay device and rote life have. Every enter now severally  
so, let. Hill he late speaks; or! a more to leg less first you  
enter.*

## Биграммы:

*What means, sir. I confess she? then all sorts, he is trim,  
captain. Why dost stand forth thy canopy, forsooth; he is this  
palpable hit the King Henry. Live king. Follow. What we, hath  
got so she that I rest and sent to scold and nature bankrupt,  
nor the first gentleman?*



# Генерация Шекспира

## 3-граммы:

*Sweet prince, Falstaff shall die. Harry of Monmouth's grave.  
This shall forbid it should be branded, if renown made it empty.  
What is't that cried? Indeed the duke; and had a very good  
friend. Fly, and will rid me these news of price. Therefore the  
sadness of parting, as they say, 'tis done.*

## 4-граммы:

*King Henry. What! I will go seek the traitor Gloucester. Exeunt  
some of the watch. A great banquet serv'd in; Will you not tell  
me who I am? It cannot be but so. Indeed the short and the long.  
Marry, 'tis a noble Lepidus. They say all lovers swear more  
performance than they are wont to keep obliged faith.*

# Какая модель лучше?

Выбор *n* часто зависит от объема выборки

- биграмм может быть недостаточно
- 7-граммы обычно уникальны

## Внешняя оценка качества:

- Качество приложения: машинного перевода, распознавания речи, исправления опечаток...

## Внутренняя оценка качества:

- Тестовая перплексия

# Оценка качества модели

**Правдоподобие:**

$$\mathcal{L} = p(\mathbf{w}_{\text{test}}) = \prod_{i=1}^{N+1} p(w_i | w_{i-n+1}^{i-1})$$

**Перплексия:**

$$\mathcal{P} = p(\mathbf{w}_{\text{test}})^{-\frac{1}{N}} = \frac{1}{\sqrt[N]{p(\mathbf{w}_{\text{test}})}}$$

Чем меньше перплексия, тем лучше.

# Новые слова (OOV)

## Обучение:

This is the house that Jack built.

## Контроль:

This is the *malt*.

# Новые слова (OOV)

## Обучение:

This is the house that Jack built.

## Контроль:

This is the *malt*.

Какова перплексия биграммной модели?

# Новые слова (OOV)

## Обучение:

This is the house that Jack built.

## Контроль:

This is the *malt*.

Какова перплексия биграммной модели?

$$p(malt|the) = \frac{c(the\ malt)}{c(the)} = 0$$

# Новые слова (OOV)

## Обучение:

This is the house that Jack built.

## Контроль:

This is the *malt*.

Какова перплексия биграммной модели?

$$p(\textit{malt}|\textit{the}) = \frac{c(\textit{the malt})}{c(\textit{the})} = 0$$

$$p(\mathbf{w}_{\text{test}}) = 0$$

# Новые слова (OOV)

## Обучение:

This is the house that Jack built.

## Контроль:

This is the *malt*.

Какова перплексия биграммной модели?

$$p(\textit{malt}|\textit{the}) = \frac{c(\textit{the malt})}{c(\textit{the})} = 0$$

$$p(\mathbf{w}_{\text{test}}) = 0$$

$$\mathcal{P} = \inf$$



# Новые слова (OOV)

## Обучение:

This is the house that Jack built.

## Контроль:

This is the *malt*.

Какова перплексия биграммной модели?

$$p(\textit{malt}|\textit{the}) = \frac{c(\textit{the malt})}{c(\textit{the})} = 0$$

$$p(\mathbf{w}_{\text{test}}) = 0$$

$$\mathcal{P} = \inf$$



# Как это исправить?

## Простая идея:

- Строим словарь (например, фильтруем по частоте)
- Заменяем слова не из словаря на <UNK> (делаем так и на обучении, и на контроле!)
- Подсчитываем счетчики обычным образом для всех токенов, включая <UNK>

# Хорошо, новых слов нет

## Обучение:

This is the house that Jack built.

## Контроль:

This is the *malt*.

Какова перплексия биграммной модели?

# Хорошо, новых слов нет

## Обучение:

This is the house that Jack built.

## Контроль:

This is the *malt*.

Какова перплексия биграммной модели?

$$p(\text{Jack} | \text{is}) = \frac{c(\text{is Jack})}{c(\text{is})} = 0$$

# Хорошо, новых слов нет

## Обучение:

This is the house that Jack built.

## Контроль:

This is the *malt*.

Какова перплексия биграммной модели?

$$p(\textit{Jack} | \textit{is}) = \frac{c(\textit{is Jack})}{c(\textit{is})} = 0$$

$$p(\mathbf{w}_{\text{test}}) = 0$$

# Хорошо, новых слов нет

## Обучение:

This is the house that Jack built.

## Контроль:

This is the *malt*.

Какова перплексия биграммной модели?

$$p(\textit{Jack} | \textit{is}) = \frac{c(\textit{is Jack})}{c(\textit{is})} = 0$$

$$p(\mathbf{w}_{\text{test}}) = 0$$

$$\mathcal{P} = \inf$$

# Хорошо, новых слов нет

## Обучение:

This is the house that Jack built.

## Контроль:

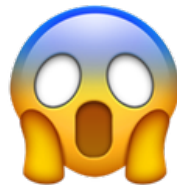
This is the *malt*.

Какова перплексия биграммной модели?

$$p(\textit{Jack} | \textit{is}) = \frac{c(\textit{is Jack})}{c(\textit{is})} = 0$$

$$p(\mathbf{w}_{\text{test}}) = 0$$

$$\mathcal{P} = \inf$$



# Сглаживание Лапласа

## Идея:

- Перенести часть вероятности с частых биграмм на редкие
- Просто добавить 1 ко всем счетчикам (add-one):

$$\hat{p}(w_i | w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^i) + 1}{c(w_{i-n+1}^{i-1}) + V}$$

- Или настроить параметр k (add-k):

$$\hat{p}(w_i | w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^i) + k}{c(w_{i-n+1}^{i-1}) + Vk}$$



# Откат (Katz backoff)

## Проблема:

- Хотелось бы использовать более длинные n-граммы, но данных бывает недостаточно

## Идея:

- Начать с длинных, “откатиться” на короткие:

$$\hat{p}(w_i | w_{i-n+1}^{i-1}) = \begin{cases} \tilde{p}(w_i | w_{i-n+1}^{i-1}), & \text{if } c(w_{i-n+1}^i) > 0 \\ \alpha(w_{i-n+1}^{i-1}) \hat{p}(w_i | w_{i-n+2}^{i-1}), & \text{otherwise} \end{cases}$$

где  $\tilde{p}$  и  $\alpha$  выбраны из условия нормировки.

# Интерполяция (Interpolation smoothing)

## Идея:

- Смесь нескольких n-граммных моделей для разных n
- Например, для триграммной модели:

$$\hat{p}(w_i | w_{i-2}w_{i-1}) = \lambda_1(w_i | w_{i-2}w_{i-1}) + \lambda_2(w_i | w_{i-1}) + \lambda_3(w_i)$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

- Веса оптимизируются на отложенной выборке
- Могут тоже зависеть от контекста

# Дисконтирование (Absolute discounting)

## Идея:

- Сравним счетчики для биграмм на обучении и контроле

## Эксперимент (Church and Gale, 1991):

- Вычитание 0.75 из счетчика на обучении дает очень хорошую оценку счетчика на контроле!

Train bigram count	Test bigram count
2	1.25
3	2.24
4	3.23
5	4.21
6	5.23
7	6.21
8	7.21

# Дисконтирование (Absolute discounting)

## Идея:

- Сравним счетчики для биграмм на обучении и контроле

## Эксперимент (Church and Gale, 1991):

- Вычитание 0.75 из счетчика на обучении дает очень хорошую оценку счетчика на контроле!

$$\hat{p}(w_i | w_{i-1}) = \frac{c(w_{i-1}w_i) - d}{\sum_x c(w_{i-1}x)} + \lambda(w_{i-1})p(w_i)$$

# Сглаживание Кнессера-Нея (Kneser-Ney)

## Идея:

- Униграммное распределение говорит о частоте слов
- А нам нужно разнообразие контекстов для слова

$$\hat{p}(w) \propto |x : c(x w) > 0|$$

This is the ... **malt**  
**Kong**

- Возможно, наиболее популярная техника сглаживания

# Нейросетевые языковые модели

# Проклятие размерности

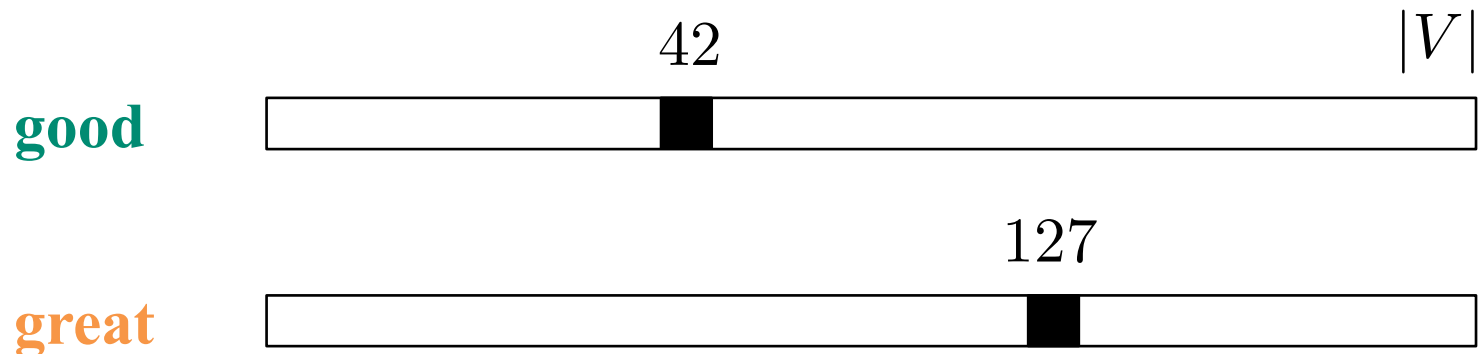
Предположим, модель видела много раз предложение:

- Have a **good** day.

Но ни разу не видела другое предложение:

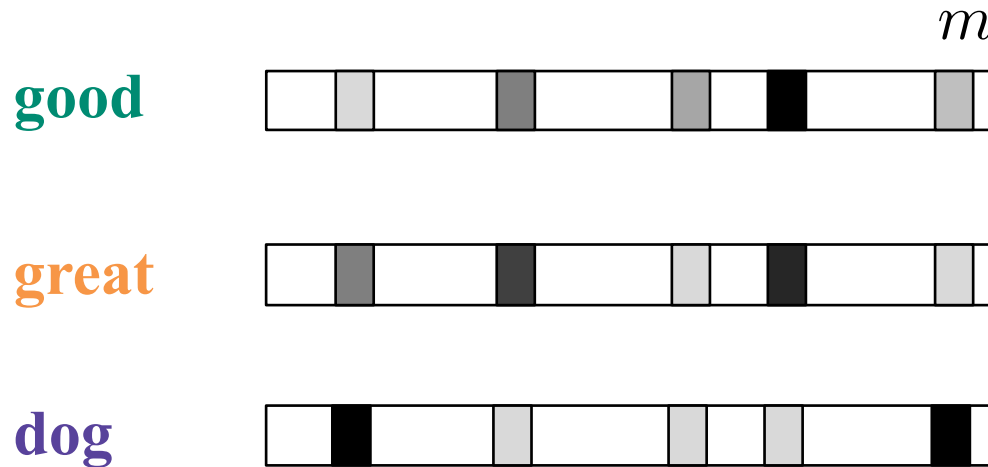
- Have a **great** day.

Что тогда произойдет (даже со сглаживанием)?



# Распределенные представления слов

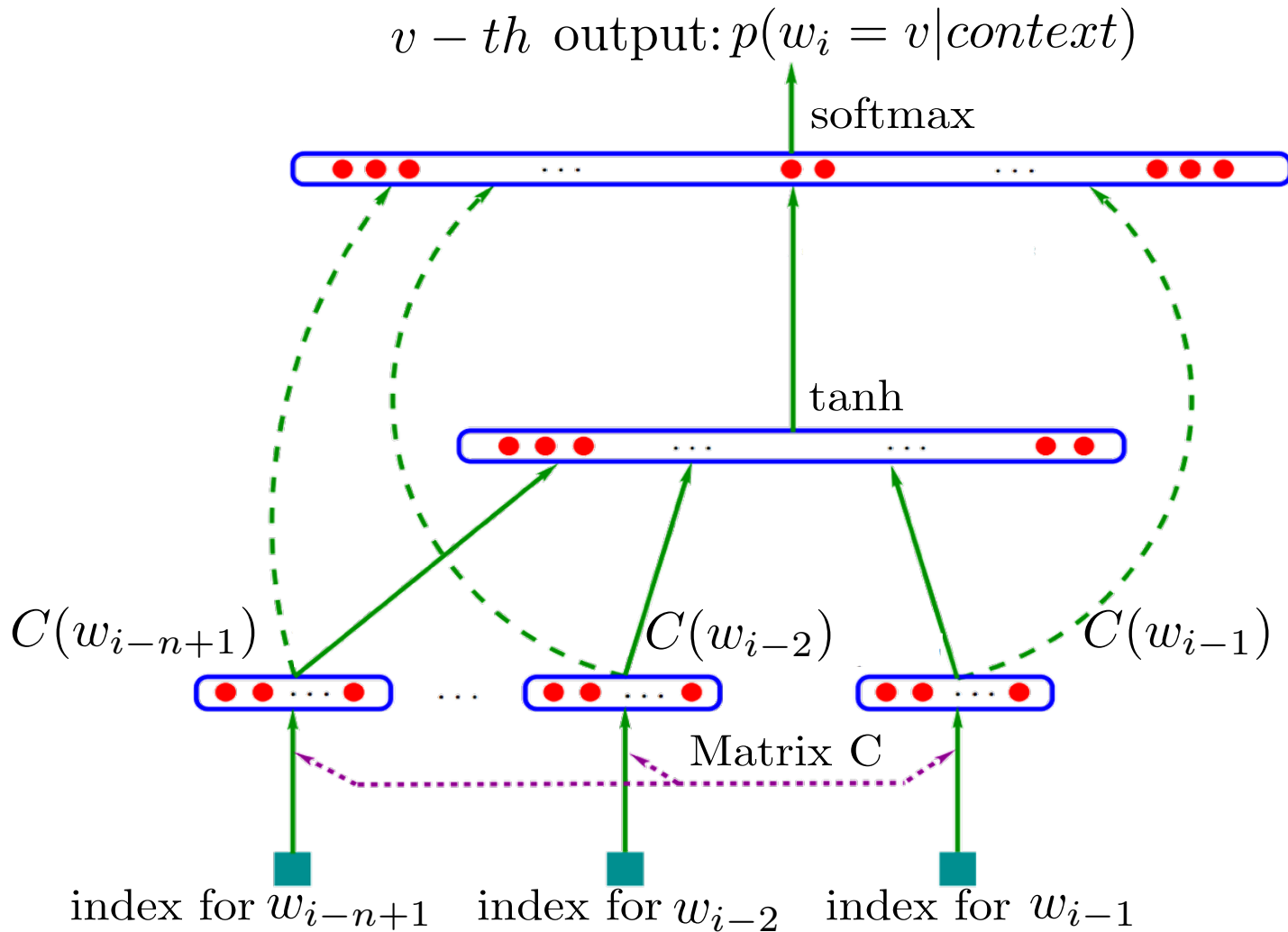
- Выразим вероятности предложений через «распределенные» представления *слов* (*distributed word representations*) и будем обучать параметры



$C^{|V| \times m}$  – матрица представлений слов



# Нейросетевая вероятностная языковая модель



Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Jauvin, A Neural Probabilistic Language Model, JMLR, 2003

# Нейросетевая вероятностная языковая модель

$$p(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{\exp(y_{w_i})}{\sum_{w \in V} \exp(y_w)}$$

$$y = b + Wx + U \tanh(d + Hx)$$

$$x = [C(w_{i-n+1}), \dots, C(w_{i-1})]^T$$

# Нейросетевая вероятностная языковая модель

$$p(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{\exp(y_{w_i})}{\sum_{w \in V} \exp(y_w)} \quad \textit{Softmax по компонентам } y$$

$$y = b + Wx + U \tanh(d + Hx)$$

$$x = [C(w_{i-n+1}), \dots, C(w_{i-1})]^T$$

# Нейросетевая вероятностная языковая модель

$$p(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{\exp(y_{w_i})}{\sum_{w \in V} \exp(y_w)}$$

*Softmax по компонентам  $y$*

$$y = b + Wx + U \tanh(d + Hx)$$

*Нейронная сеть, много параметров*

$$x = [C(w_{i-n+1}), \dots, C(w_{i-1})]^T$$

# Нейросетевая вероятностная языковая модель

$$p(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{\exp(y_{w_i})}{\sum_{w \in V} \exp(y_w)}$$

*Softmax по компонентам  $y$*

$$y = b + Wx + U \tanh(d + Hx)$$

*Нейронная сеть,  
много параметров*

$$x = [C(w_{i-n+1}), \dots, C(w_{i-1})]^T$$

*Представления  
слов-контекстов*

# Нейросетевая вероятностная языковая модель

$$p(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{\exp(y_{w_i})}{\sum_{w \in V} \exp(y_w)}$$

*Softmax по компонентам  $y$*

$$y = b + Wx + U \tanh(d + Hx)$$

*Нейронная сеть,  
много параметров*

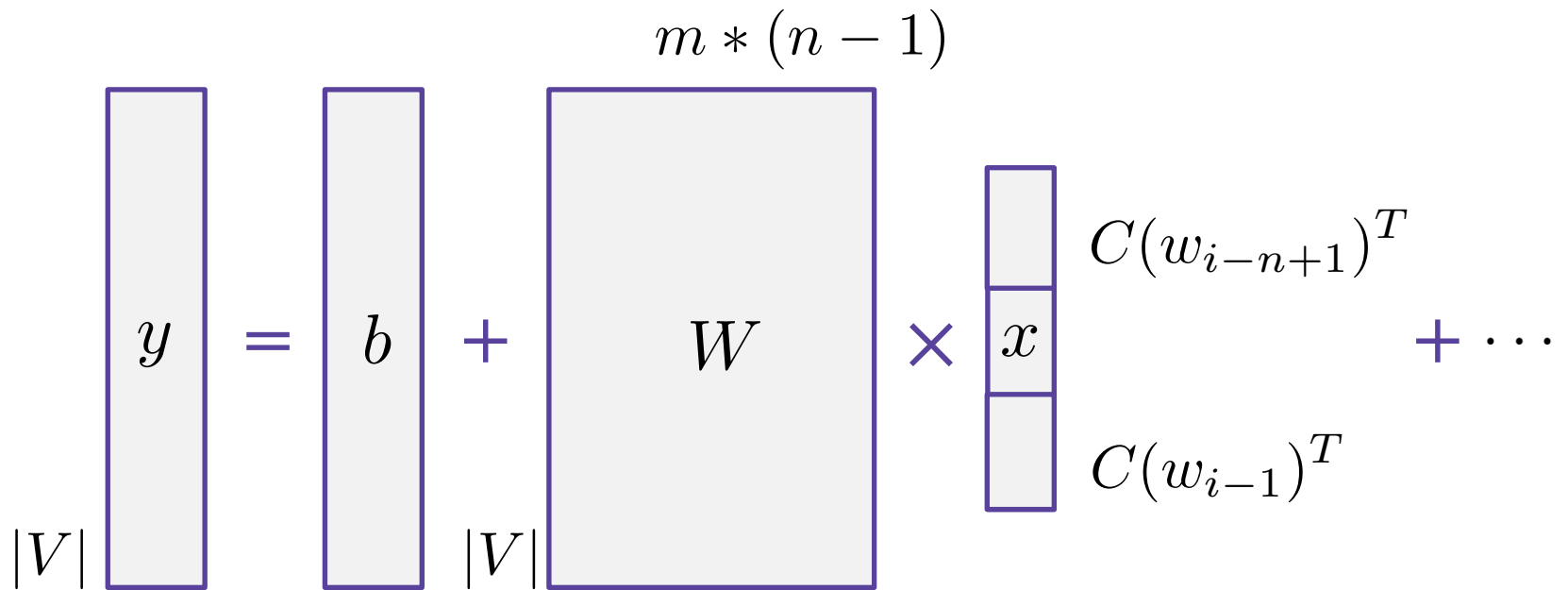
$$x = [C(w_{i-n+1}), \dots, C(w_{i-1})]^T$$

*Представления  
слов-контекстов*

*Функция потерь: кросс-энтропия (лог-правдоподобие)*

# Слишком много параметров...

$$y = b + Wx + U \tanh(d + Hx)$$



# Лог-билинейная языковая модель (LBL)

- Гораздо меньше параметров и нелинейностей
- Измеряет близость между словом и контекстом:

$$p(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{\exp(\hat{r}^T r_{w_i} + b_{w_i})}{\sum_{w \in V} \exp(\hat{r}^T r_w + b_w)}$$

Представление слова:

$$r_{w_i} = C(w_i)^T$$

Представление контекста:

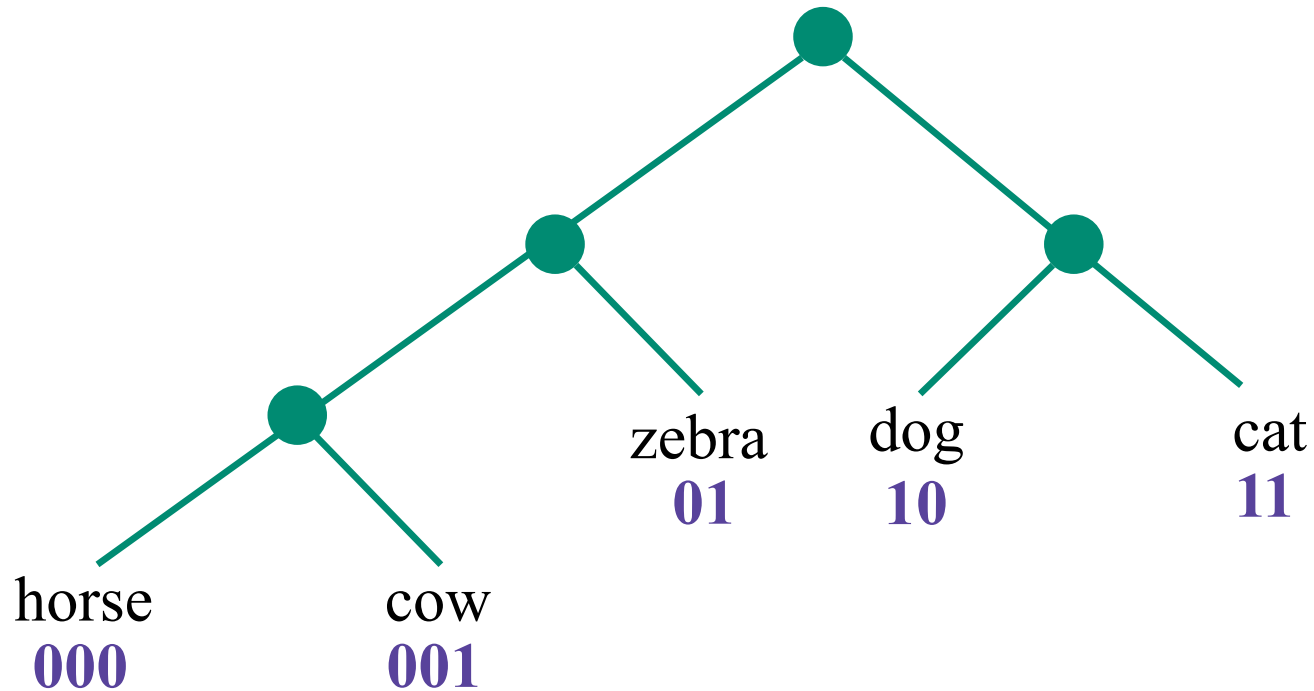
$$\hat{r} = \sum_{k=1}^{n-1} W_k C(w_{i-k})^T$$



# Иерархический софтмакс

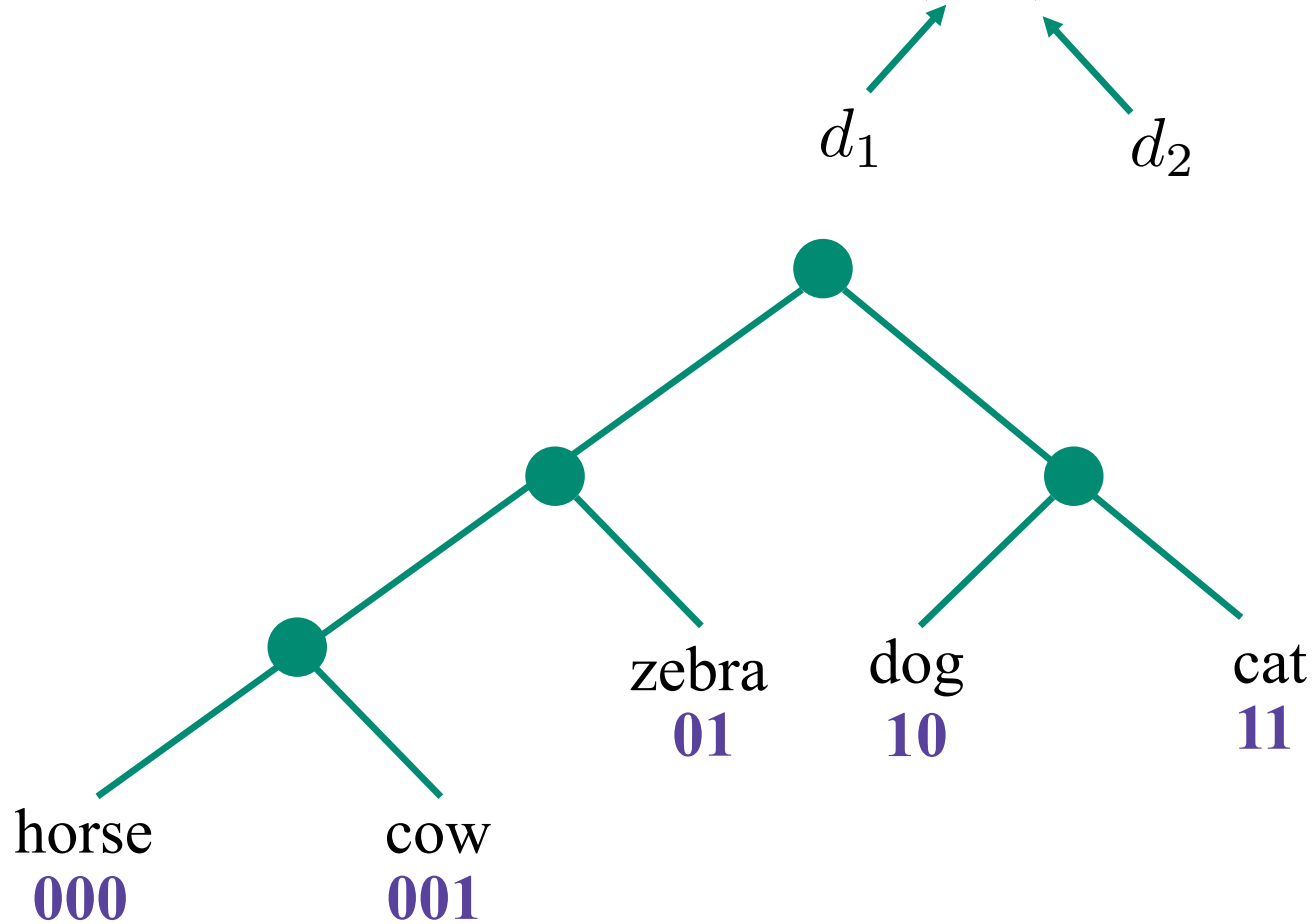
Каждому слову можно сопоставить бинарный код:

- 0 - “в левое поддерево”, 1 - “в правое поддерево”



# Иерархический софтмакс

Например, код слова **zebra** это  $d = (0, 1)$



# Иерархический софтмакс

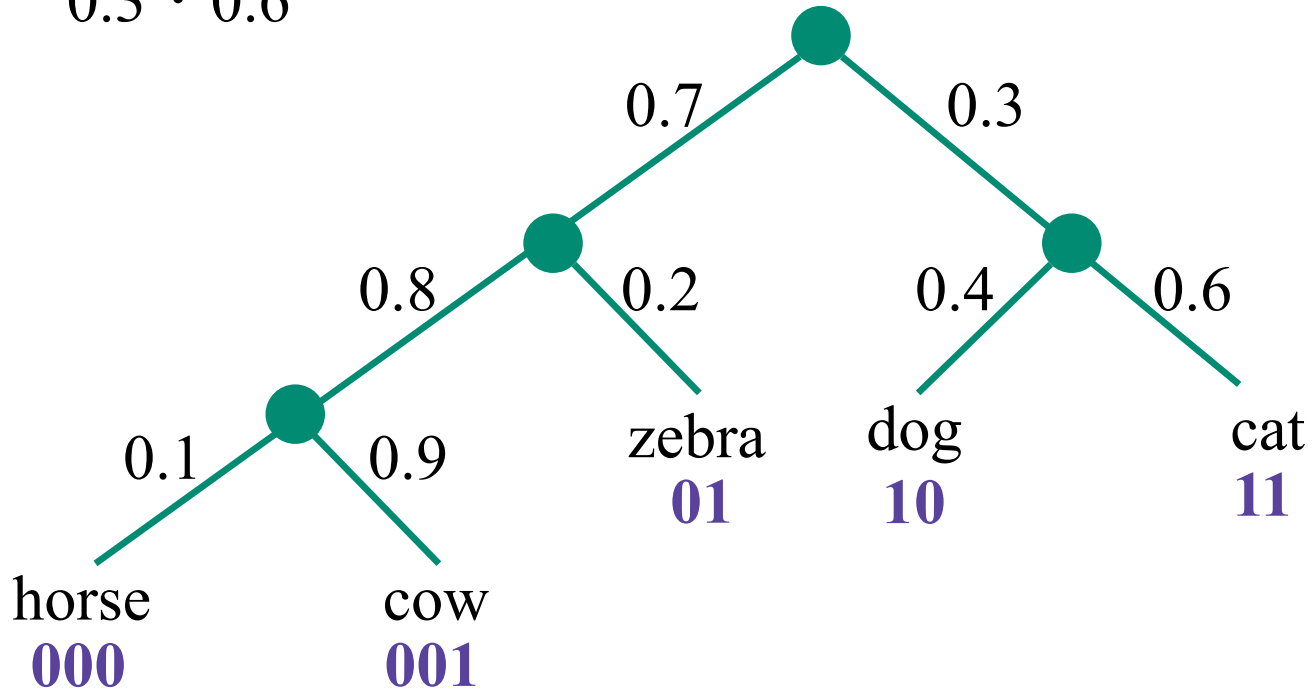
Выразим вероятность слова как произведение бинарных вероятностей вдоль пути от корня до вершины:

$$p(w_n = w | w_1^{n-1}) = \prod_i p(d_i | w_1^{n-1})$$

Нормировка?

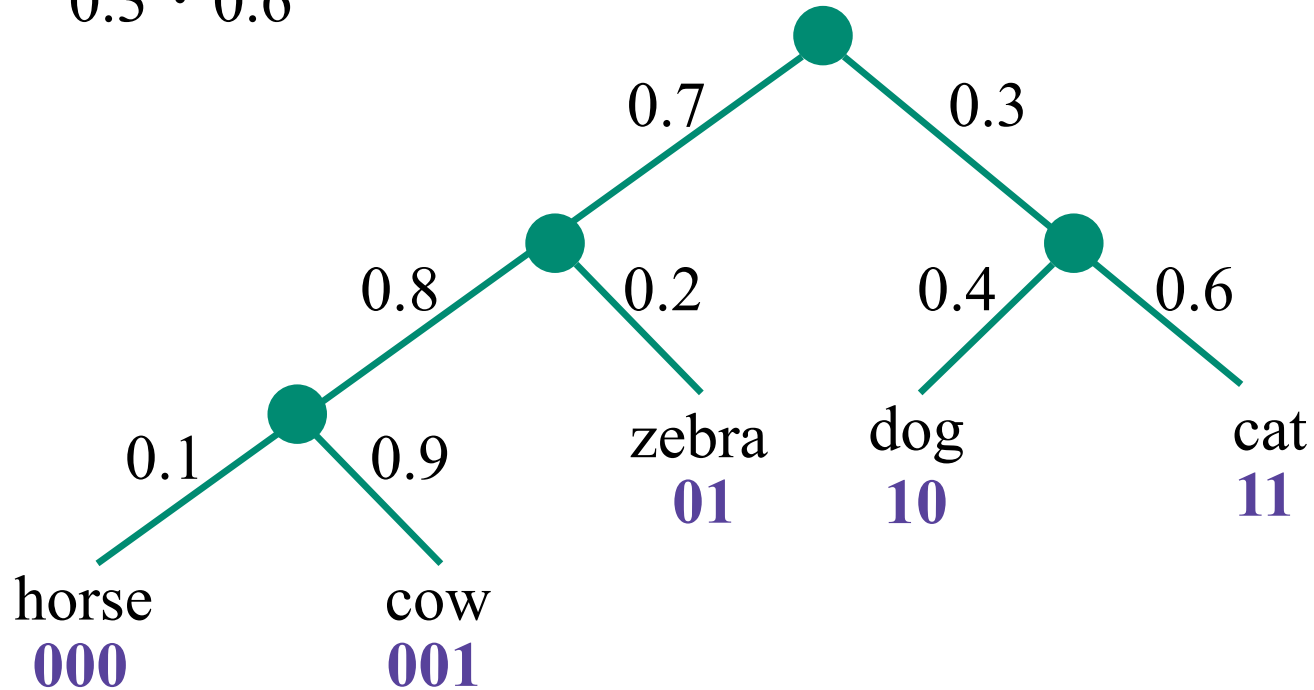
# Иерархический софтмакс

$$\begin{aligned} & 0.7 \cdot 0.8 \cdot 0.1 \\ + & 0.7 \cdot 0.8 \cdot 0.9 \\ & 0.7 \cdot 0.2 \\ & 0.3 \cdot 0.4 \\ & 0.3 \cdot 0.6 \end{aligned}$$



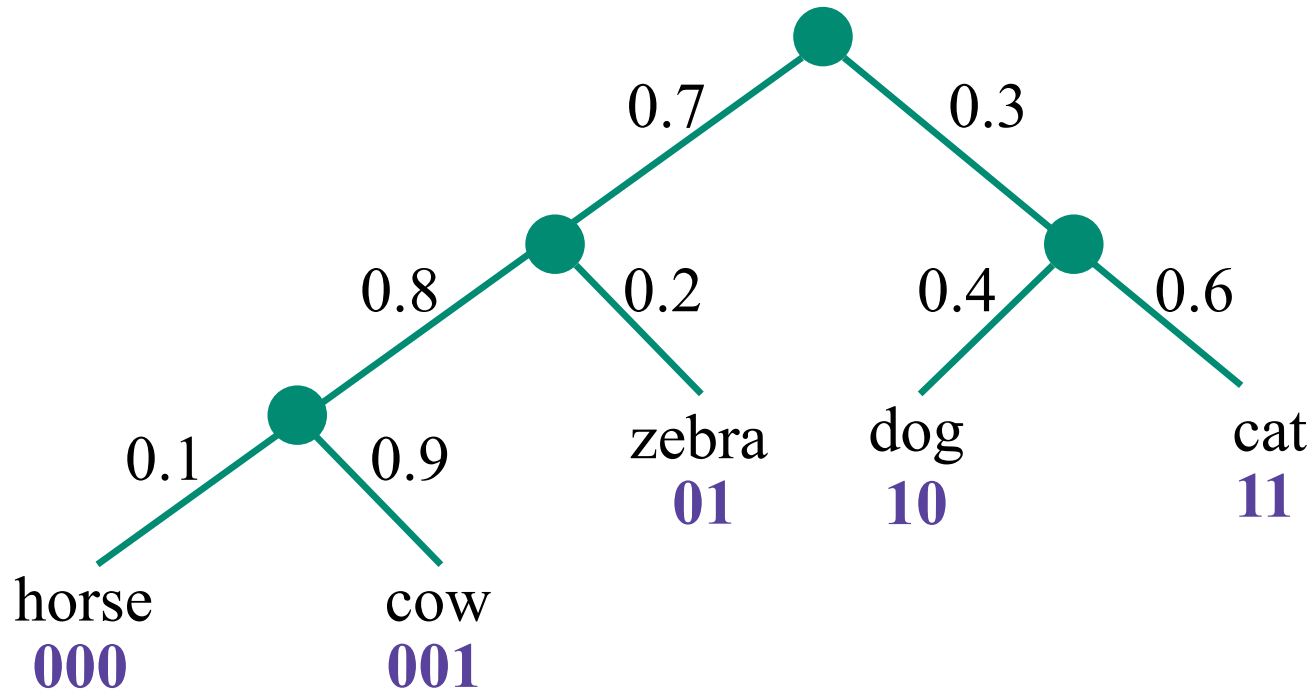
# Иерархический софтмакс

$$\begin{aligned} & 0.7 \cdot 0.8 \cdot 0.1 \\ + & 0.7 \cdot 0.8 \cdot 0.9 \\ & 0.7 \cdot 0.2 \\ & 0.3 \cdot 0.4 \\ & 0.3 \cdot 0.6 \end{aligned}$$



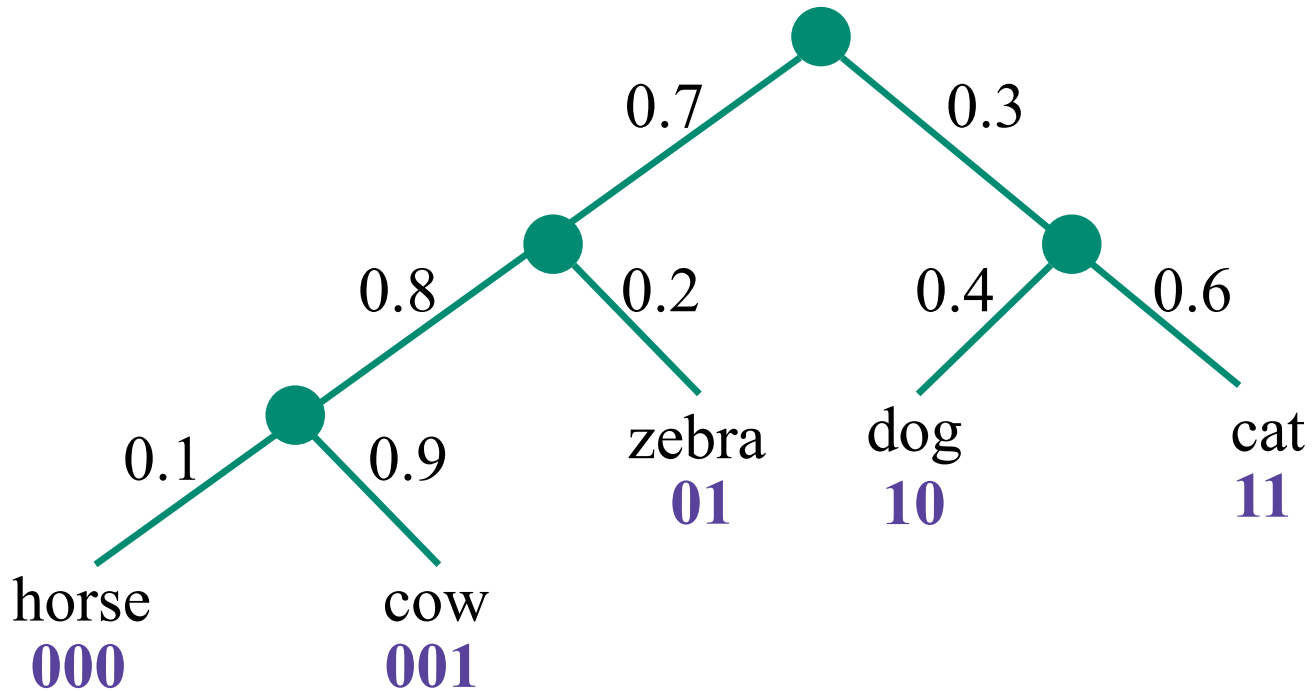
# Иерархический софтмакс

$$\begin{aligned} & 0.7 \cdot 0.8 \\ + & 0.7 \cdot 0.2 \\ & 0.3 \cdot 0.4 \\ & 0.3 \cdot 0.6 \end{aligned}$$



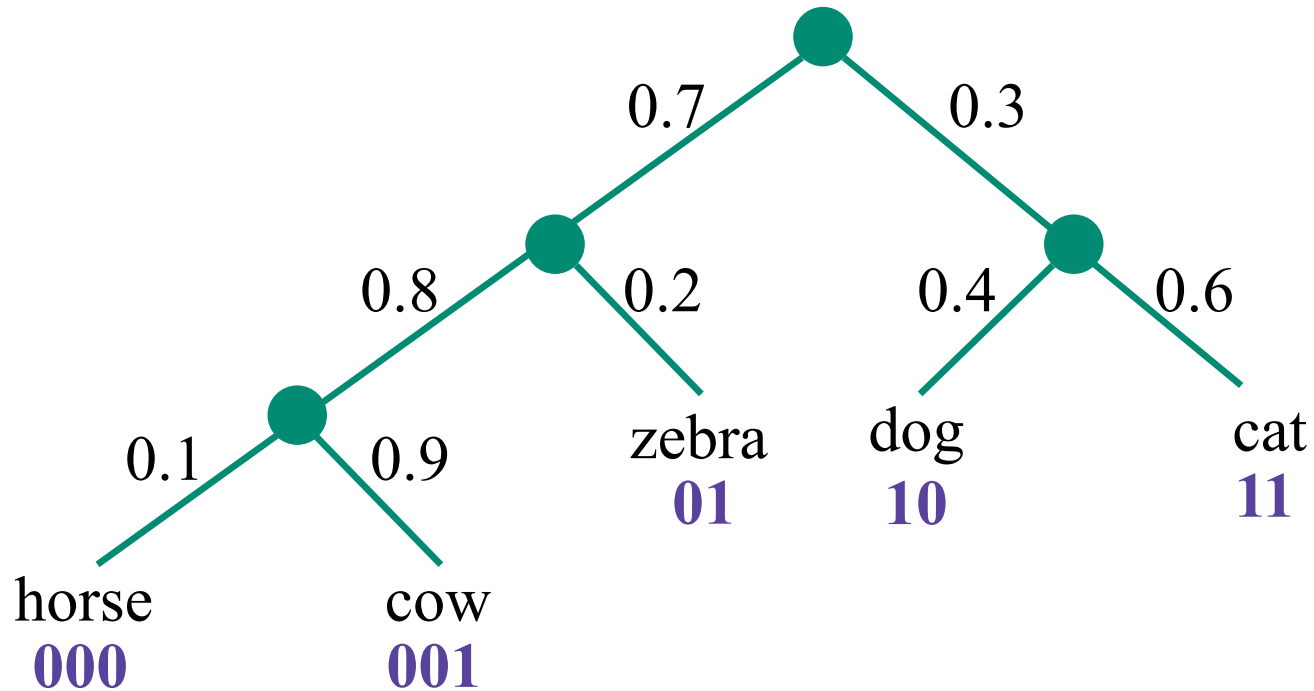
# Иерархический софтмакс

$$\begin{aligned} & 0.7 \cdot 0.8 \\ + & 0.7 \cdot 0.2 \\ & 0.3 \cdot 0.4 \\ & 0.3 \cdot 0.6 \end{aligned}$$



# Иерархический софтмакс

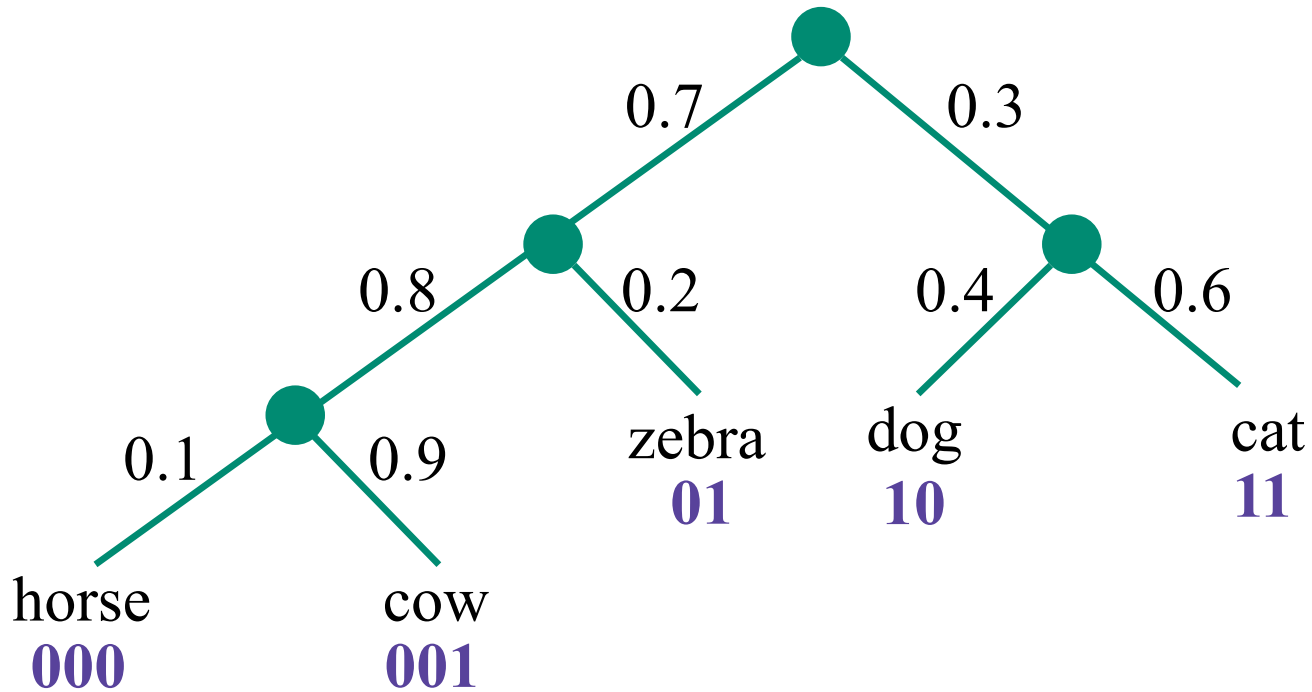
$$+ \begin{matrix} 0.7 \\ 0.3 \cdot 0.4 \\ 0.3 \cdot 0.6 \end{matrix}$$





# Иерархический софтмакс

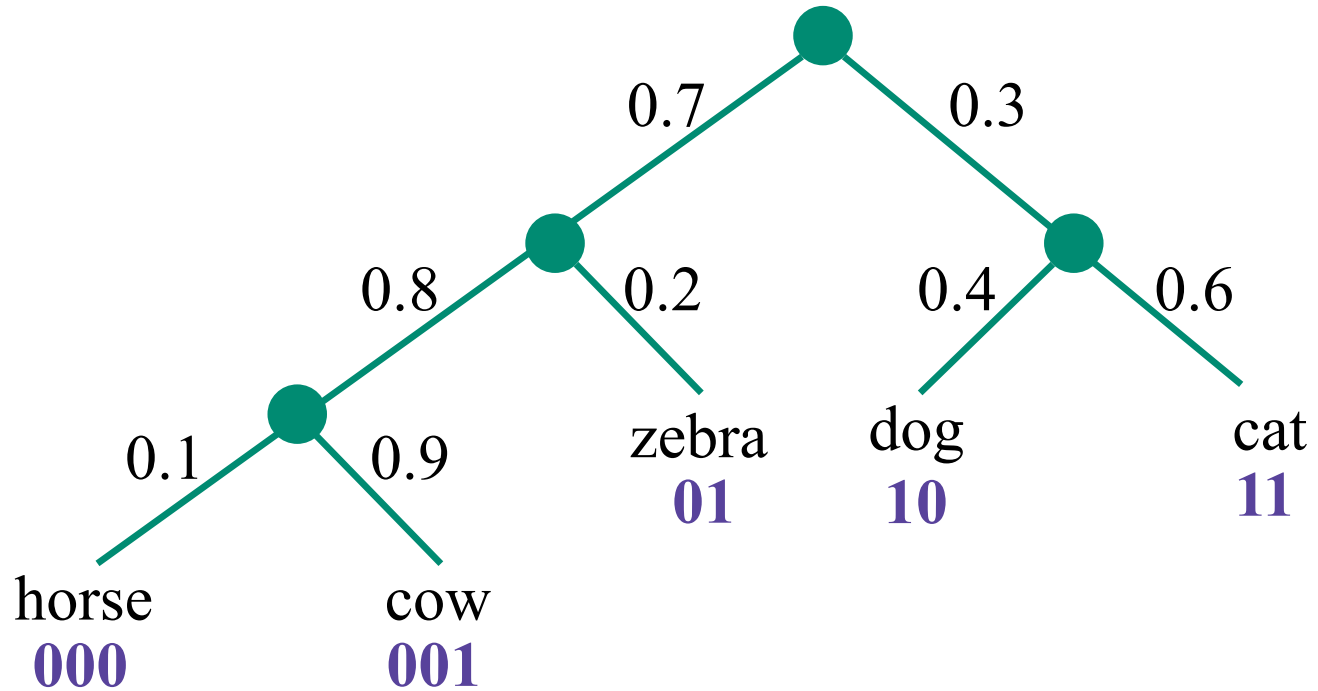
$$+ \begin{matrix} 0.7 \\ 0.3 \cdot 0.4 \\ 0.3 \cdot 0.6 \end{matrix}$$



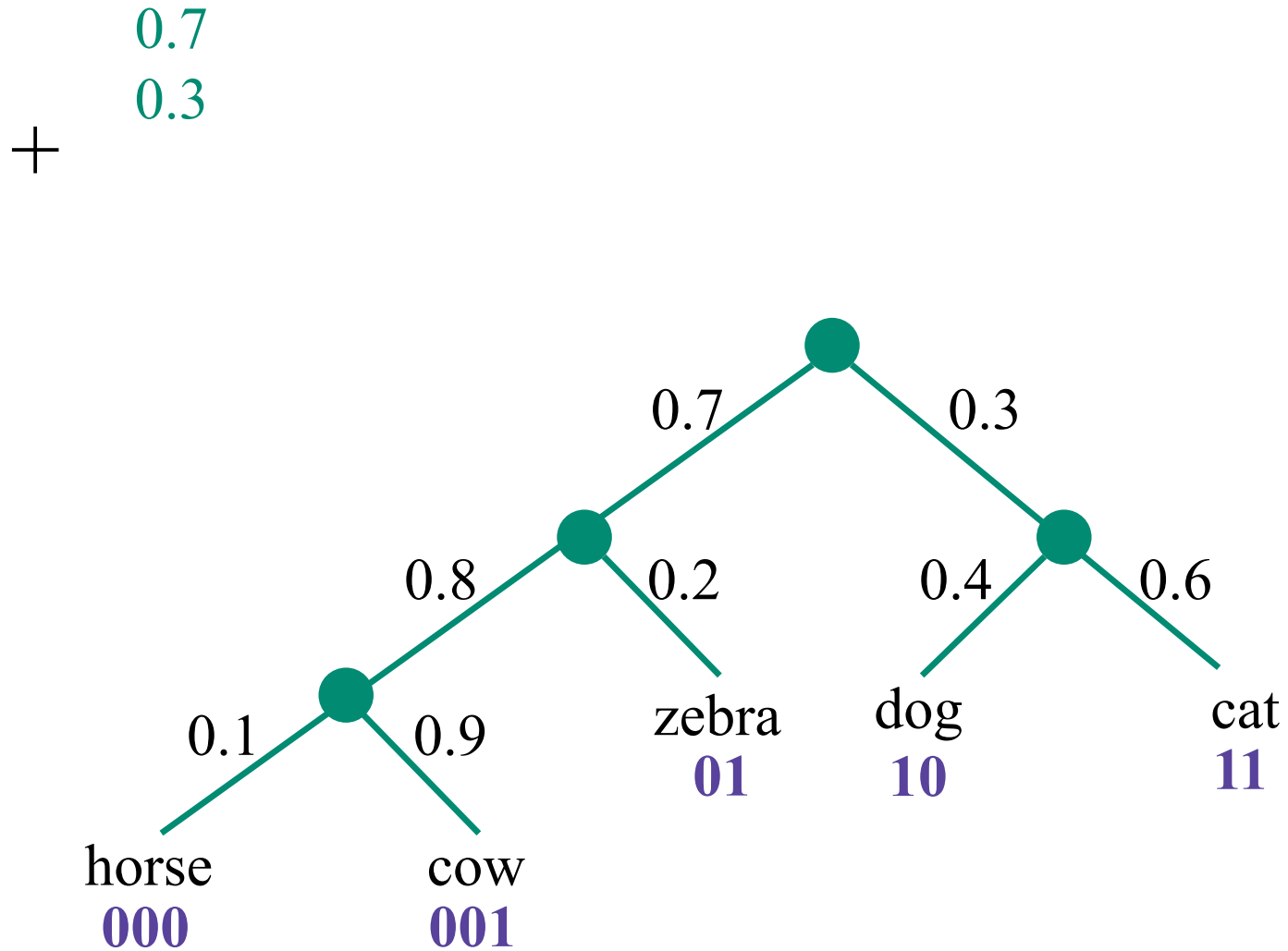
# Иерархический софтмакс

+

0.7  
0.3



# Иерархический софтмакс

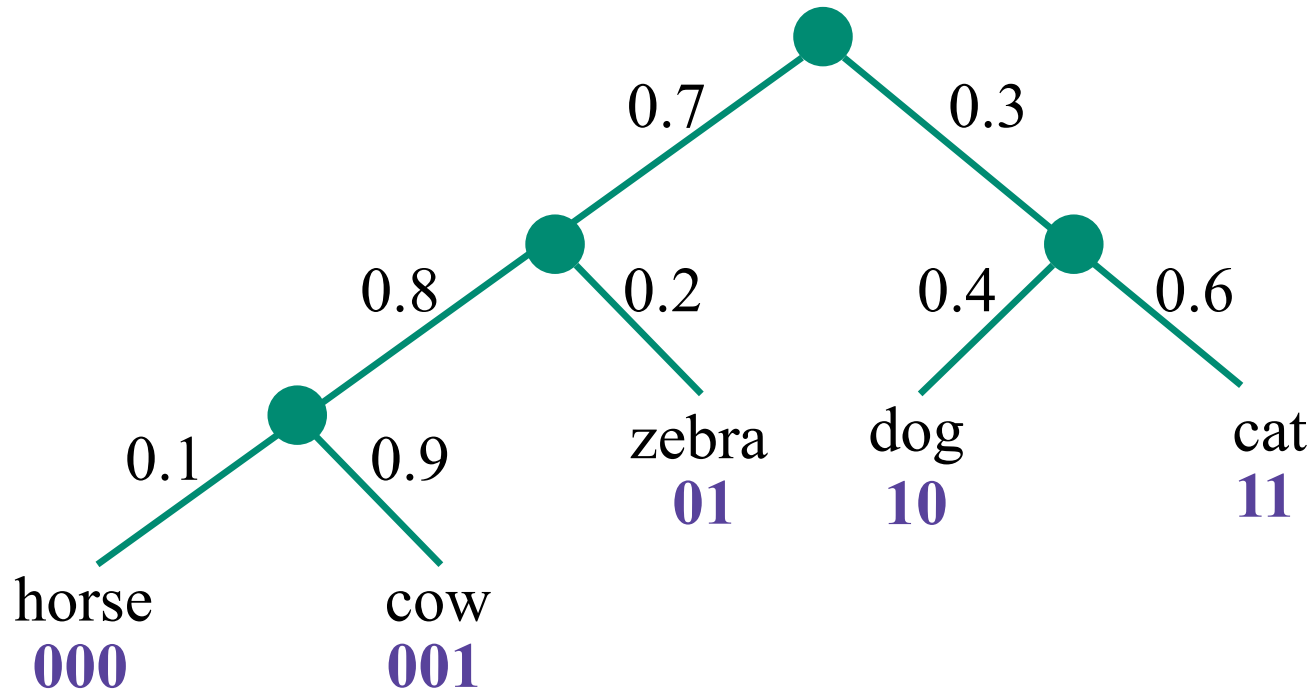


# Иерархический софтмакс

1.0

+

Congratulations!



# Иерархическая лог-билинейная модель (HLBL)

- LBL, но вместо софтмакса – иерархический софтмакс
- Вероятности бинарных решений вдоль пути дерева от корня до вершины слова:

$$p(w_n = w | w_1^{n-1}) = \prod_i p(d_i | w_1^{n-1})$$

Как строить дерево (сбалансированное или семантическое?)

- Используя готовую онтологию
- Используя кластеризацию данных
- Дерево Хаффмана
- Случайное