Text Classification

Lena Voita

Lecture-blog and lots of additional materials are here:
https://lena-voita.github.io/nlp_course/text_classification.html

What is going to happen:

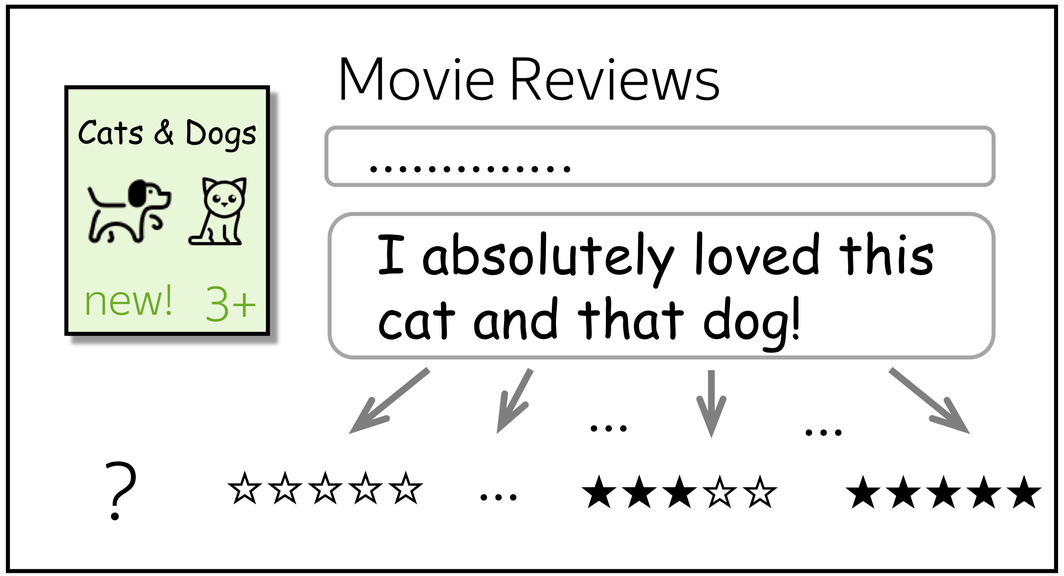
- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

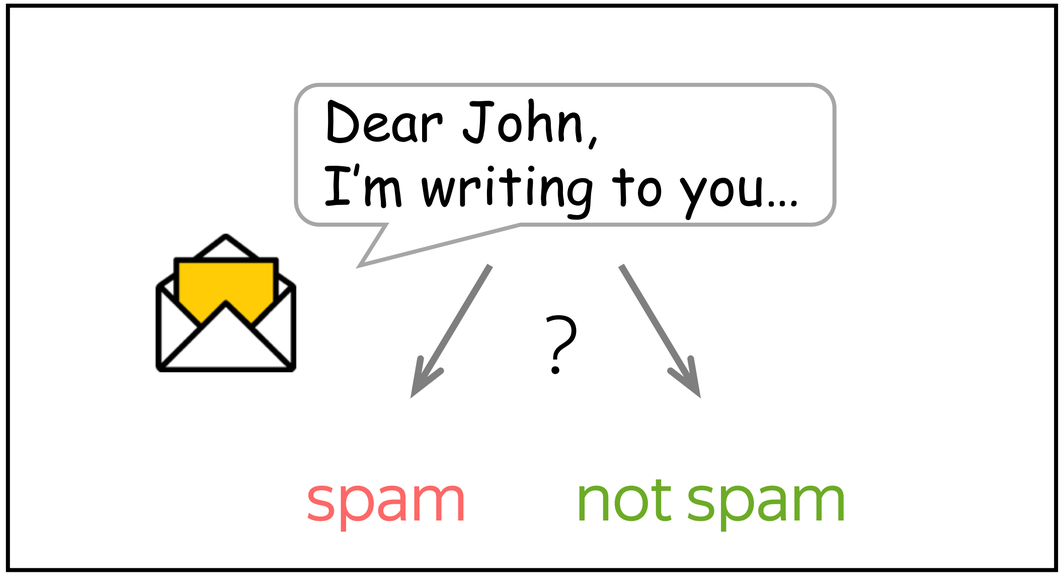
Text Classification

- Multi-class: many labels
- Single label: only one label is correct



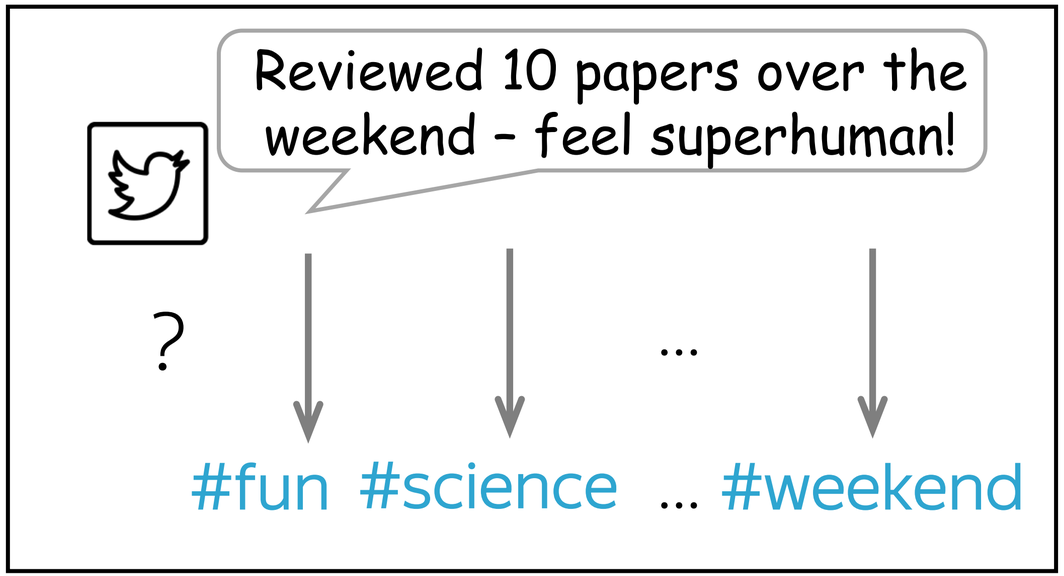
Text Classification

- Binary: two labels
- Single label: only one label is correct



Text Classification

- Multi-class: many labels
- Multi-label: several labels can be correct



Datasets for Text Classification

Datasets vary a lot in:

Dataset	Type	Number of labels	Size (train/test)	Avg. length (tokens)
SST	sentiment	5 or 2	8.5k / 1.1k	19
IMDb Review	sentiment	2	25k / 25k	271
Yelp Review	sentiment	5 or 2	650k / 50k	179
Amazon Review	sentiment	5 or 2	3m / 650k	79
TREC	question	6	5.5k / 0.5k	10
Yahoo! Answers	question	10	1.4m / 60k	131
AG's News	topic	4	120k / 7.6k	44
Sogou News	topic	6	54k / 6k	737
DBPedia	topic	14	560k / 70k	67

https://lena-voita.github.io/nlp_course/text_classification.html#dataset_examples

Datasets for Text Classification

Datasets vary a lot in:

- Type

Dataset	Type	Number of labels	Size (train/test)	Avg. length (tokens)
SST	sentiment	5 or 2	8.5k / 1.1k	19
IMDb Review	sentiment	2	25k / 25k	271
Yelp Review	sentiment	5 or 2	650k / 50k	179
Amazon Review	sentiment	5 or 2	3m / 650k	79
TREC	question	6	5.5k / 0.5k	10
Yahoo! Answers	question	10	1.4m / 60k	131
AG's News	topic	4	120k / 7.6k	44
Sogou News	topic	6	54k / 6k	737
DBPedia	topic	14	560k / 70k	67

https://lena-voita.github.io/nlp_course/text_classification.html#dataset_examples

Datasets for Text Classification

Datasets vary a lot in:

- Type
- Number of labels

Dataset	Type	Number of labels	Size (train/test)	Avg. length (tokens)
SST	sentiment	5 or 2	8.5k / 1.1k	19
IMDb Review	sentiment	2	25k / 25k	271
Yelp Review	sentiment	5 or 2	650k / 50k	179
Amazon Review	sentiment	5 or 2	3m / 650k	79
TREC	question	6	5.5k / 0.5k	10
Yahoo! Answers	question	10	1.4m / 60k	131
AG's News	topic	4	120k / 7.6k	44
Sogou News	topic	6	54k / 6k	737
DBPedia	topic	14	560k / 70k	67

https://lena-voita.github.io/nlp_course/text_classification.html#dataset_examples

Datasets for Text Classification

Datasets vary a lot in:

- Type
- Number of labels
- Dataset size

Dataset	Type	Number of labels	Size (train/test)	Avg. length (tokens)
SST	sentiment	5 or 2	8.5k / 1.1k	19
IMDb Review	sentiment	2	25k / 25k	271
Yelp Review	sentiment	5 or 2	650k / 50k	179
Amazon Review	sentiment	5 or 2	3m / 650k	79
TREC	question	6	5.5k / 0.5k	10
Yahoo! Answers	question	10	1.4m / 60k	131
AG's News	topic	4	120k / 7.6k	44
Sogou News	topic	6	54k / 6k	737
DBPedia	topic	14	560k / 70k	67

https://lena-voita.github.io/nlp_course/text_classification.html#dataset_examples

Datasets for Text Classification

Datasets vary a lot in:

- Type
- Number of labels
- Dataset size
- Example length

Dataset	Type	Number of labels	Size (train/test)	Avg. length (tokens)
SST	sentiment	5 or 2	8.5k / 1.1k	19
IMDb Review	sentiment	2	25k / 25k	271
Yelp Review	sentiment	5 or 2	650k / 50k	179
Amazon Review	sentiment	5 or 2	3m / 650k	79
TREC	question	6	5.5k / 0.5k	10
Yahoo! Answers	question	10	1.4m / 60k	131
AG's News	topic	4	120k / 7.6k	44
Sogou News	topic	6	54k / 6k	737
DBPedia	topic	14	560k / 70k	67

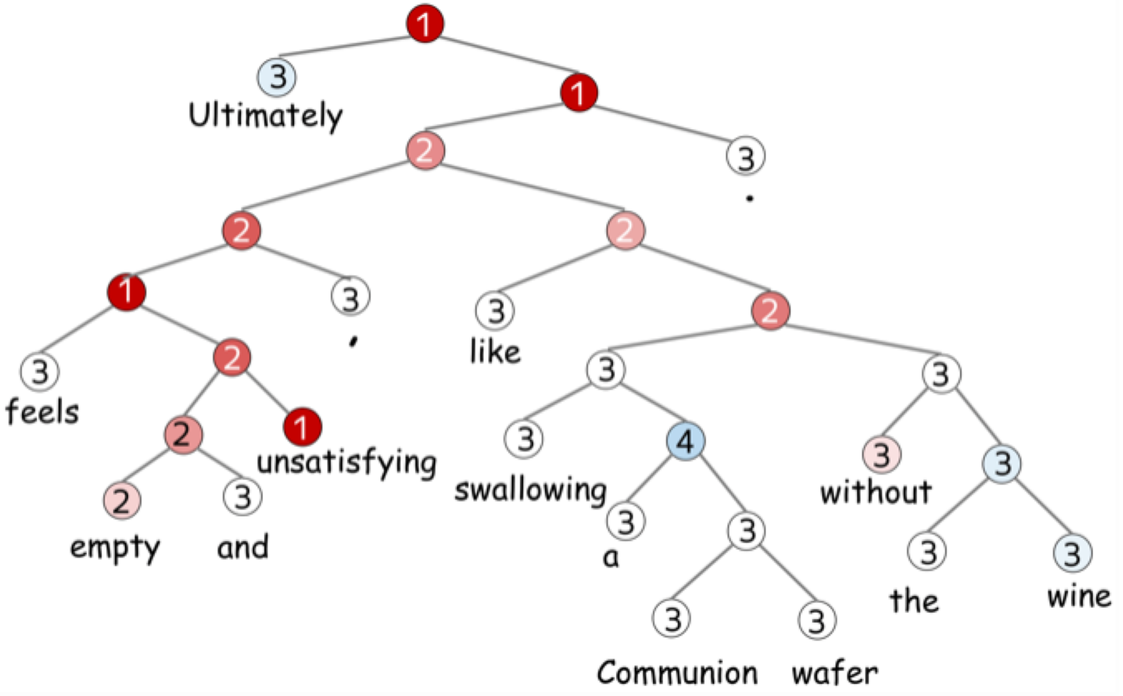
https://lena-voita.github.io/nlp_course/text_classification.html#dataset_examples

Datasets for Text Classification: Sentiment

Pick a dataset

SST | IMDb Review | Yelp Review | Amazon Review | TREC | Yahoo! Answers | AG's News | Sogou News | DBPedia

Label: 1
Review:
Ultimately feels empty and unsatisfying , like swallowing a Communion wafer without the wine .



https://lena-voita.github.io/nlp_course/text_classification.html#dataset_examples

Datasets for Text Classification: Sentiment

Pick a dataset

- SST
- (•) IMDb Review
- Yelp Review
- Amazon Review
- TREC
- Yahoo! Answers
- AG's News
- Sogou News
- DBPedia

Label: negative

Review
Hobgoblins .... Hobgoblins .... where do I begin?!?

This film gives Manos - The Hands of Fate and Future War a run for their money as the worst film ever made . This one is fun to laugh at , where as Manos was just painful to watch . Hobgoblins will end up in a time capsule somewhere as the perfect movie to describe the term : " 80 's cheeze " . The acting (and I am using this term loosely) is atrocious , the Hobgoblins are some of the worst puppets you will ever see , and the garden tool fight has to be seen to be believed . The movie was the perfect vehicle for MST3 K , and that version is the only way to watch this mess . This movie gives Mike and the bots lots of ammunition to pull some of the funniest one - liners they have ever done . If you try to watch this without the help of Mike and the bots ..... God help you ! !

https://lena-voita.github.io/nlp_course/text_classification.html#dataset_examples

Datasets for Text Classification: Sentiment

Pick a dataset

○ SST ○ IMDb Review ◉ Yelp Review ○ Amazon Review
○ TREC ○ Yahoo! Answers ○ AG's News ○ Sogou News ○ DBPedia

Label: 4

Review
I had a serious craving for Roti. So glad I found this place. A very small menu selection but it had exactly what I wanted. The serving for $8.20 after tax is enough for 2 meals. I know where to go from now on for a great meal with leftovers. This is a noteworthy place to bring my Uncle T.J. who's a Trini when he comes to visit.

https://lena-voita.github.io/nlp_course/text_classification.html#dataset_examples

Datasets for Text Classification: Sentiment

Pick a dataset

- SST
- IMDb Review
- Yelp Review
- **Amazon Review**
- TREC
- Yahoo! Answers
- AG's News
- Sogou News
- DBPedia

Label: 3

Review Title: Simple

Review Content:
This book was not anything special. Although I love romances, it was too simple. The symbolism was spelled out to the readers in a blunt manner. The less educated readers may appreciate it. The wording was quite beautiful at times and the plot was enchanting (perfect for a movie) but it is not heart wrenching like the movie Titantic (which was a must see!) ;)

https://lena-voita.github.io/nlp_course/text_classification.html#dataset_examples

Datasets for Text Classification: Questions

Pick a dataset

○ SST ○ IMDb Review ○ Yelp Review ○ Amazon Review
○ TREC ◉ Yahoo! Answers ○ AG's News ○ Sogou News ○ DBPedia

Label: Society & Culture

Question Title: Why do people have the bird, turkey for thanksgiving?

Question Content: Why this bird? Any Significance?

Best Answer
It is believed that the pilgrims and indians shared wild turkey and venison on the original Thanksgiving.

Turkey's "Americanness" was established by Benjamin Franklin, who had advocated for the turkey, not the bald eagle, becoming the national bird.

https://lena-voita.github.io/nlp_course/text_classification.html#dataset_examples

Datasets for Text Classification: Topic

Pick a dataset

○ SST ○ IMDb Review ○ Yelp Review ○ Amazon Review
○ TREC ○ Yahoo! Answers ◉ AG's News ○ Sogou News ○ DBPedia

Label: Sports

Title: Schumacher Triumphs as Ferrari Seals Formula One Title

Description
BUDAPEST (Reuters) - Michael Schumacher cruised to a record 12th win of the season in the Hungarian Grand Prix on Sunday to hand his Ferrari team a sixth successive constructors' title.

https://lena-voita.github.io/nlp_course/text_classification.html#dataset_examples

Datasets for Text Classification: Topic

Pick a dataset

○ SST ○ IMDb Review ○ Yelp Review ○ Amazon Review
○ TREC ○ Yahoo! Answers ○ AG's News ○ Sogou News ◉ DBPedia

Label: Artist

Title: Esfandiar Monfaredzadeh

Abstract
Esfandiar Monfaredzadeh (Persian : اسفندیار منفردزاده) is an Iranian composer and director. He was born in 1941 in Tehran His major works are Gheisar Dash Akol Tangna Gavaznha. He has 2 daughters Bibinaz Monfaredzadeh and Sanam Monfaredzadeh Woods (by marriage).

https://lena-voita.github.io/nlp_course/text_classification.html#dataset_examples

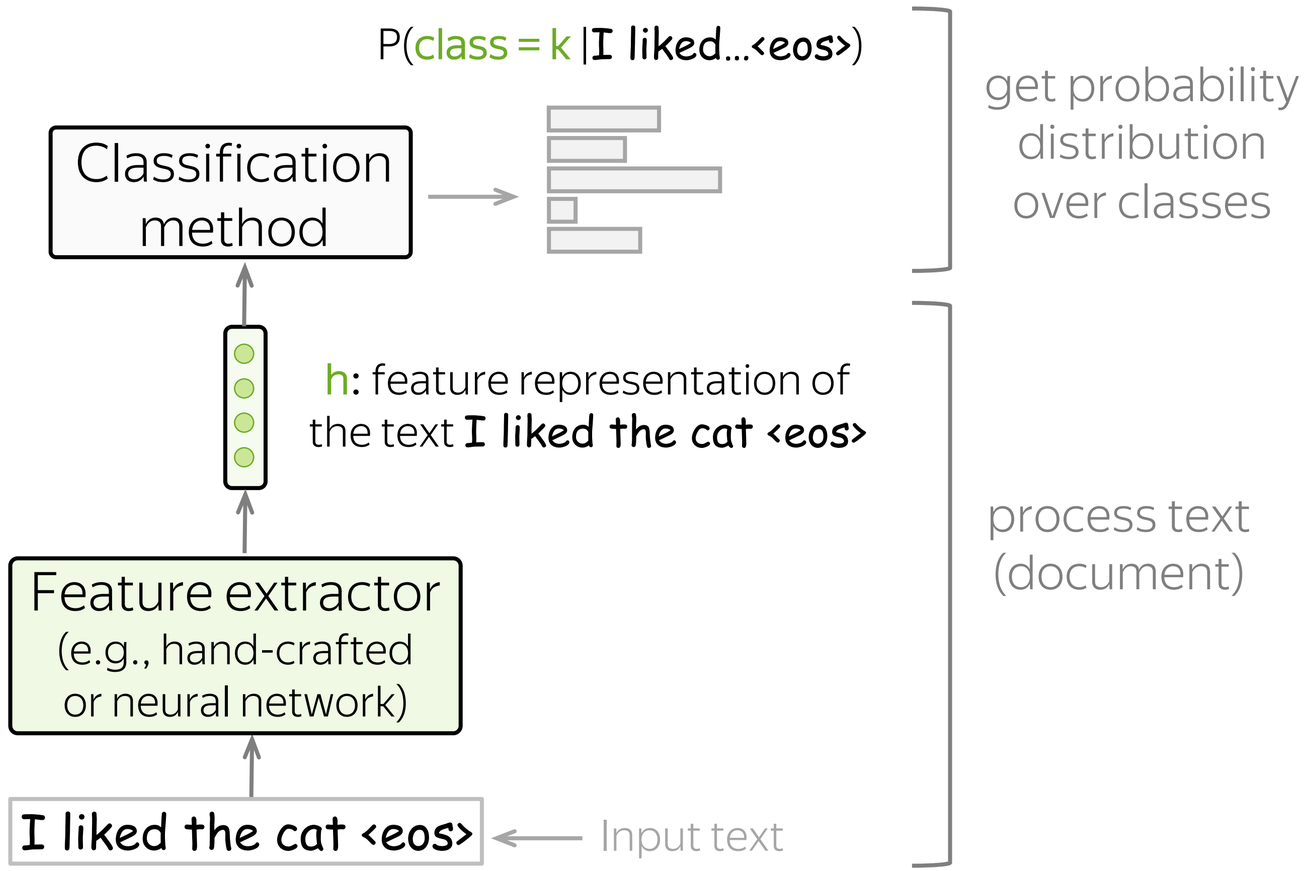
What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

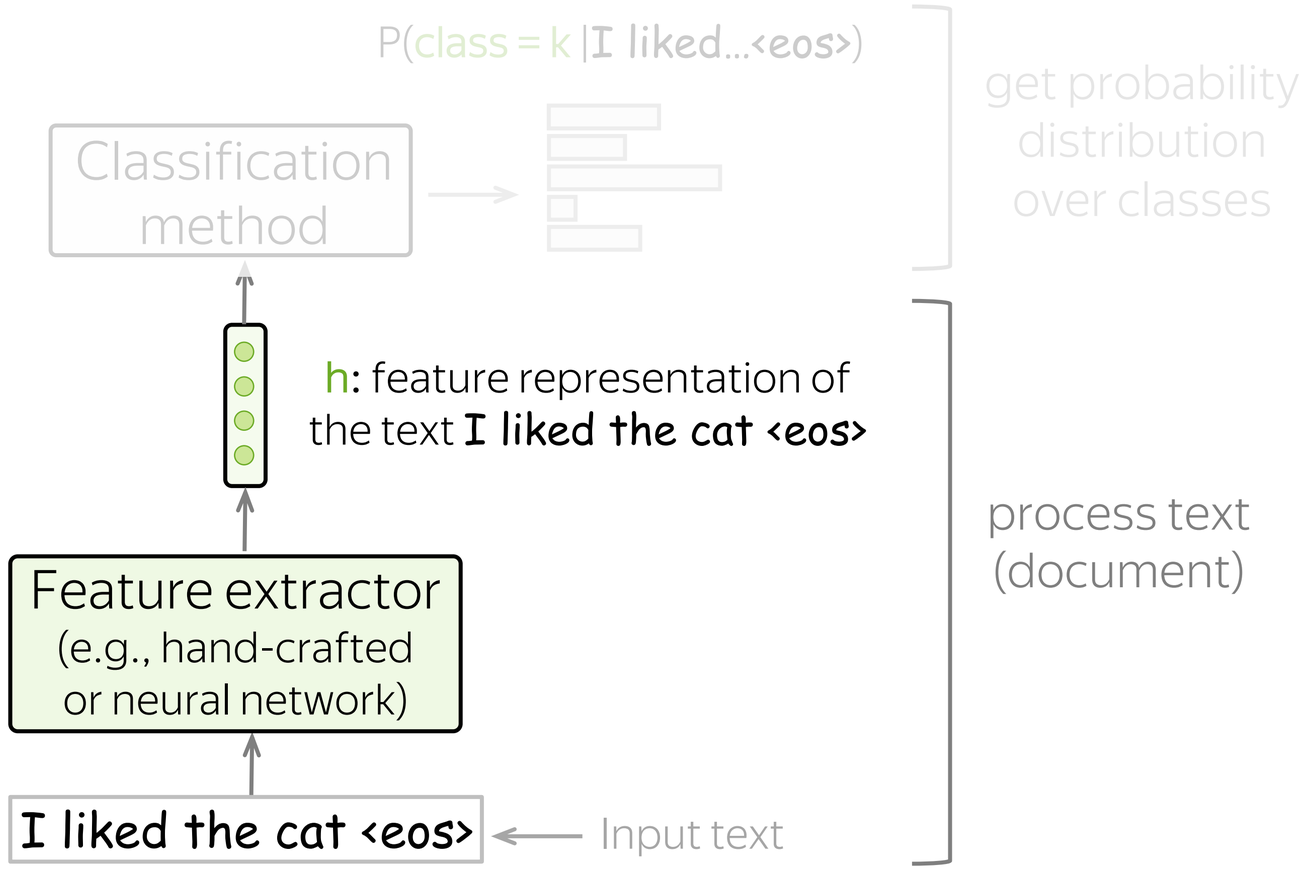
Get Feature Representation and Classify



Get Feature Representation and Classify

Feature extractor:

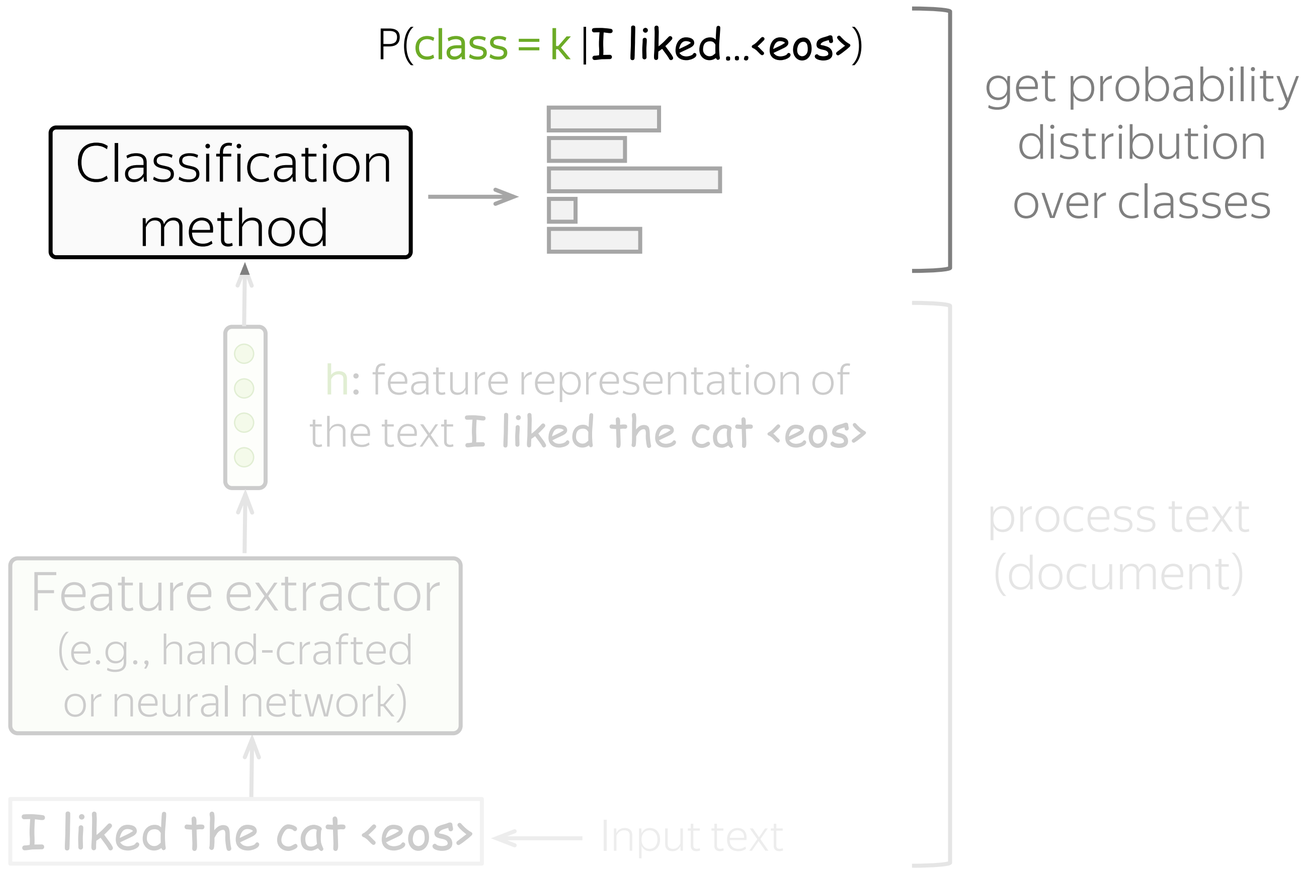
- **classical methods** – features are extracted manually by a human
- **neural methods** – features are extracted by a network: a network **learns** what is important



Get Feature Representation and Classify

Classification method:

- generative – ?
- discriminative – ?



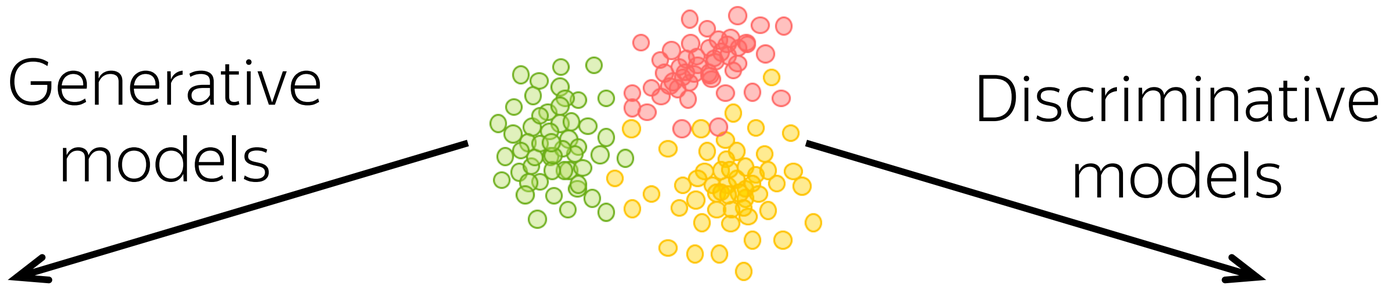
What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

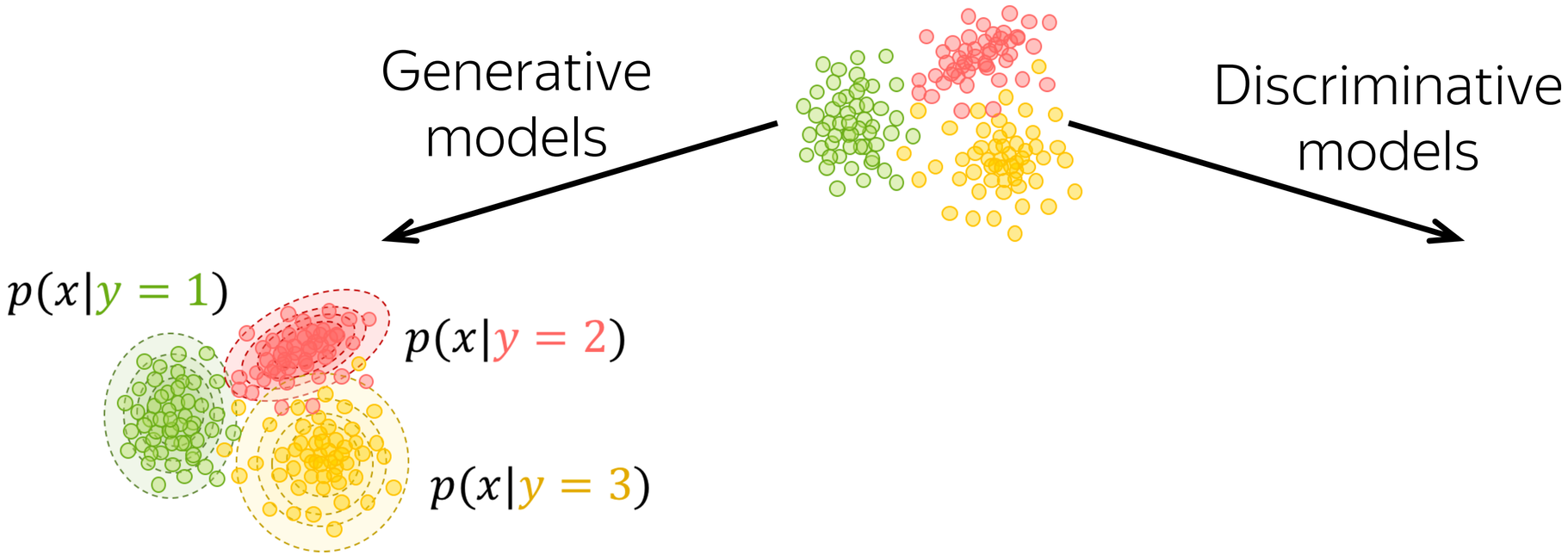
What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

Generative and Discriminative Models



Generative and Discriminative Models



Learn: data distribution

$$p(x, y) = p(x|y) \cdot p(y)$$

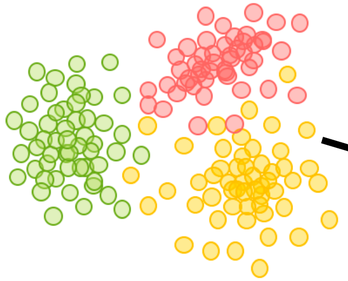
How predict:

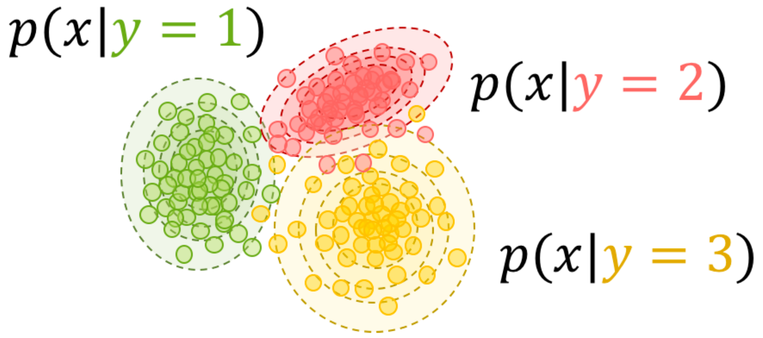
$$y = \arg\max_k p(x, y) = \arg\max_k p(x|y) \cdot p(y)$$

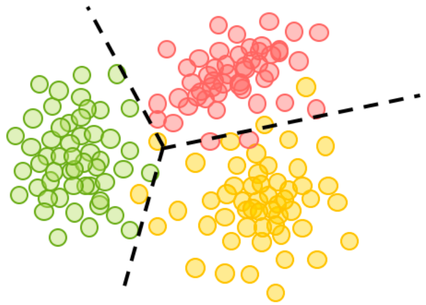
Generative and Discriminative Models

Generative models

Discriminative models







Learn: data distribution

$$p(x, y) = p(x|y) \cdot p(y)$$

How predict:

$$y = \arg\max_k p(x, y) = \arg\max_k p(x|y) \cdot p(y)$$

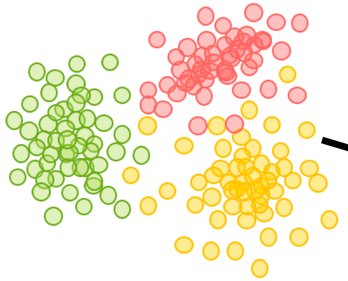
Learn: boundary between classes $p(y|x)$

How predict:

$$y = \arg\max_k p(y|x)$$

Generative and Discriminative Models

Generative models



Discriminative models

$p(x|y=1)$

$p(x|y=2)$

$p(x|y=3)$

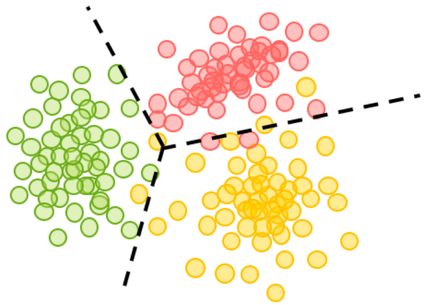
Most of the models are discriminative

Learn: data distribution

$p(x,y) = p(x|y) \cdot p(y)$

How predict:

$$y = \arg\max_k p(x,y) = \arg\max_k p(x|y) \cdot p(y)$$



Learn: boundary between classes $p(y|x)$

How predict:

$$y = \arg\max_k p(y|x)$$

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
  - Naïve Bayes
  - Logistic Regression
  - SVM
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

Naïve Bayes: High-Level Idea

$$y^* = \arg\max_k P(y = k|x)$$

Naïve Bayes: High-Level Idea

Bayes' rule
(hence Naïve Bayes)

$$y^* = \arg\max_k P(y = k|x) =$$

Naïve Bayes: High-Level Idea

Bayes' rule
(hence Naïve Bayes)

$$y^* = \arg\max_k P(y = k|x) = \arg\max_k \frac{P(x|y = k) \cdot P(y = k)}{P(x)}$$

Naïve Bayes: High-Level Idea

Bayes' rule (hence Naïve Bayes)

Ignore P(x) – it does not influence the argmax

$$y^* = \arg\max_k P(y = k|x) = \arg\max_k \frac{P(x|y = k) \cdot P(y = k)}{P(x)} =$$

Naïve Bayes: High-Level Idea

Bayes' rule (hence Naïve Bayes)

Ignore P(x) – it does not influence the argmax

$$y^* = \arg\max_k P(y = k|x) = \arg\max_k \frac{P(x|y = k) \cdot P(y = k)}{P(x)} = \arg\max_k P(x|y = k) \cdot P(y = k)$$

Naïve Bayes: High-Level Idea

Bayes' rule (hence Naïve Bayes)

Ignore P(x) – it does not influence the argmax

$$y^* = \arg\max_k P(y = k|x) = \arg\max_k \frac{P(x|y = k) \cdot P(y = k)}{P(x)} = \arg\max_k P(x|y = k) \cdot P(y = k)$$

need to define this

Naïve Bayes: High-Level Idea

$$y^* = \arg\max_k P(y = k|x) = \arg\max_k P(x|y = k) \cdot P(y = k) = \arg\max_k P(x, y = k)$$

Naïve Bayes: High-Level Idea

$$y^* = \arg\max_k P(y = k|x) = \arg\max_k P(x|y = k) \cdot P(y = k) = \arg\max_k P(x, y = k)$$

posterior probability:
after looking at data
(i.e., we know x)

Naïve Bayes: High-Level Idea

$$y^* = \arg\max_k \underbrace{P(y = k|x)}_{\text{posterior}} = \arg\max_k P(x|y = k) \cdot \underbrace{P(y = k)}_{\text{prior}} = \arg\max_k P(x, y = k)$$

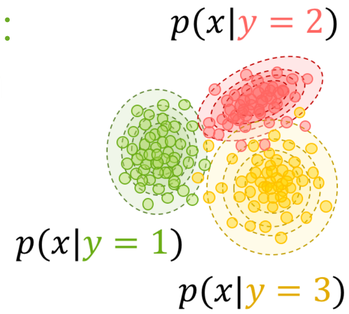
posterior probability:
after looking at data
(i.e., we know **x**)

prior probability:
before looking at data
(i.e., we don't know **x**)

Naïve Bayes: High-Level Idea

$$y^* = \arg\max_k \underbrace{P(y = k|x)}_{\text{posterior}} = \arg\max_k \underbrace{P(x|y = k)}_{\text{likelihood}} \cdot \underbrace{P(y = k)}_{\text{prior}} = \arg\max_k P(x, y = k)$$

posterior probability:
after looking at data
(i.e., we know **x**)

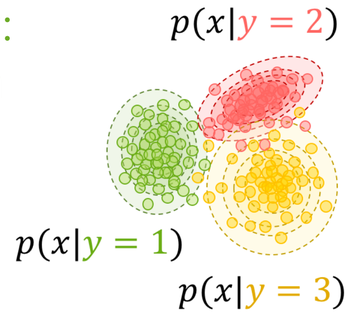


prior probability:
before looking at data
(i.e., we don't know **x**)

Naïve Bayes: High-Level Idea

$$y^* = \arg\max_k \underbrace{P(y = k|x)}_{\text{posterior}} = \arg\max_k \underbrace{P(x|y = k)}_{\text{likelihood}} \cdot \underbrace{P(y = k)}_{\text{prior}} = \arg\max_k \underbrace{P(x, y = k)}_{\text{joint}}$$

posterior probability:
after looking at data
(i.e., we know **x**)



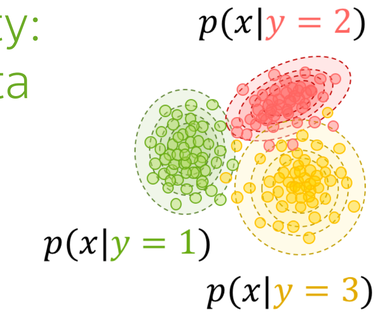
prior probability:
before looking at data
(i.e., we don't know **x**)

joint probability

Naïve Bayes: High-Level Idea

$$y^* = \arg\max_k \underbrace{P(y = k|x)}_{\text{posterior}} = \arg\max_k \underbrace{P(x|y = k)}_{\text{likelihood}} \cdot \underbrace{P(y = k)}_{\text{prior}} = \arg\max_k \underbrace{P(x, y = k)}_{\text{joint}}$$

posterior probability:
after looking at data
(i.e., we know **x**)



prior probability:
before looking at data
(i.e., we don't know **x**)

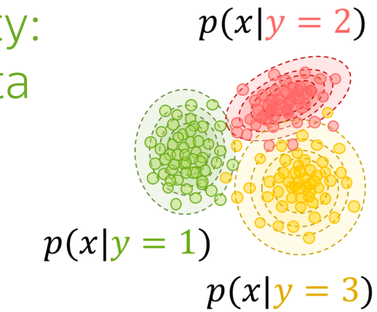
joint probability

Is this a generative or a discriminative model?

Naïve Bayes: High-Level Idea

$$y^* = \arg\max_k \underbrace{P(y = k|x)}_{\text{posterior}} = \arg\max_k \underbrace{P(x|y = k)}_{\text{likelihood}} \cdot \underbrace{P(y = k)}_{\text{prior}} = \arg\max_k \underbrace{P(x, y = k)}_{\text{joint}}$$

posterior probability:
after looking at data
(i.e., we know **x**)



prior probability:
before looking at data
(i.e., we don't know **x**)

joint probability
(hence the model
is **generative**)

This is a generative model!

Naïve Bayes: High-Level Idea

$$y^* = \arg\max_k P(y = k|x) = \arg\max_k \underbrace{P(x|y = k)} \cdot \underbrace{P(y = k)} = \arg\max_k P(x, y = k)$$

need to define this

How to define P(y=k)?

Easy! Just count labels:

$$P(y = k) = \frac{N(y = k)}{\sum_{i=1}^{K} N(y = i)}$$

P(x|y=k): Use the “naive” assumptions

$$P(x|y = k) = P(x_1, x_2, \dots, x_n|y = k)$$

P(x|y=k): Use the “naive” assumptions

$$P(x|y = k) = P(x_1, x_2, \dots, x_n|y = k)$$

The Naive Bayes assumptions:

- Bag of Words assumption - word order does not matter,
- Conditional Independence assumption - features (words) are independent given the class.

P(x|y=k): Use the “naive” assumptions

$$P(x|y=k) = P(x_1, x_2, \ldots, x_n|y=k) = \prod_{i=1}^{n} P(x_i|y=k)$$

The Naive Bayes assumptions:

- Bag of Words assumption - word order does not matter,
- Conditional Independence assumption - features (words) are independent given the class.

P(x|y=k): Use the “naive” assumptions

$$P(x|y=k) = P(x_1, x_2, \ldots, x_n|y=k) = \prod_{i=1}^{n} P(x_i|y=k)$$

The Naive Bayes assumptions:

- Bag of Words assumption - word order does not matter,
- Conditional Independence assumption - features (words) are independent given the class.

P(This film is awesome !| $y = +$) =

- $P(\text{This}|y = +)$
- $P(\text{film}|y = +)$
- $P(\text{is}|y = +)$
- $P(\text{awesome}|y = +)$
- $P(!|y = +)$

P(x|y=k): Use the “naive” assumptions, then count

$$P(x|y=k) = P(x_1, x_2, \dots, x_n|y=k) = \prod_{i=1}^{n} P(x_i|y=k)$$

- use the naïve assumptions

$$P(x_i|y=k) = \ ?$$

P(x|y=k): Use the “naive” assumptions, then count

$$P(x|y=k) = P(x_1, x_2, \dots, x_n|y=k) = \prod_{i=1}^{n} P(x_i|y=k)$$ - use the naïve assumptions

$$P(x_i|y=k) = \frac{N(x_i, y=k)}{\sum_{t=1}^{|V|} N(x_t, y=k)}$$ - count!

What if $N(x_i, y = k) = 0$?

$$P(x_i|y = k) = \frac{N(x_i, y = k)}{\sum_{t=1}^{|V|} N(x_t, y = k)}$$

Imagine we haven't seen **pterodactyl** or **abracadabra** in training texts $\Longrightarrow$ $N(x_i, y = k) = 0$

What if $N(x_i, y = k) = 0$?

$$P(x_i|y = k) = \frac{N(x_i, y = k)}{\sum_{t=1}^{|V|} N(x_t, y = k)}$$

Imagine we haven't seen **pterodactyl** or **abracadabra** in training texts $\Longrightarrow$ $N(x_i, y = k) = 0$

$N(x_i, y = k)$

In training data, haven't seen token x_i in documents of class k

What if $N(x_i, y = k) = 0$?

$$P(x_i|y = k) = \frac{N(x_i, y = k)}{\sum_{t=1}^{|V|} N(x_t, y = k)}$$

Imagine we haven't seen **pterodactyl** or **abracadabra** in training texts $\implies N(x_i, y = k) = 0$

nulls out token prob.

$$\underbrace{N(x_i, y = k)}_{\text{In training data, haven't seen token } x_i \text{ in documents of class } k} \implies P(x_i|y = k) = \frac{N(x_i, y = k)}{\sum_{t=1}^{|V|} N(x_t, y = k)} = 0$$

What if $N(x_i, y = k) = 0$?

$$P(x_i|y = k) = \frac{N(x_i, y = k)}{\sum_{t=1}^{|V|} N(x_t, y = k)}$$

Imagine we haven't seen **pterodactyl** or **abracadabra** in training texts $\Longrightarrow$ $N(x_i, y = k) = 0$

nulls out token prob. $\Longrightarrow$ nulls out document probability $\Longrightarrow$ Bad!

$$\underbrace{N(x_i, y = k)}_{\text{In training data, haven't seen token } x_i \text{ in documents of class } k} \Longrightarrow P(x_i|y = k) = \frac{N(x_i, y = k)}{\sum_{t=1}^{|V|} N(x_t, y = k)} = 0 \Longrightarrow P(x|y = k) = \prod_{i=1}^{n} P(x_i|y = k) = 0$$

What if $N(x_i, y = k) = 0$?

$$P(x_i|y = k) = \frac{N(x_i, y = k)}{\sum_{t=1}^{|V|} N(x_t, y = k)}$$

Imagine we haven't seen **pterodactyl** or **abracadabra** in training texts $\Longrightarrow$ $N(x_i, y = k) = 0$

A simple trick: add a small δ to all counts:

$$P(x_i|y = k) = \frac{\delta + N(x_i, y = k)}{\sum_{t=1}^{|V|} (\delta + N(x_t, y = k))} = \frac{\delta + N(x_i, y = k)}{\delta \cdot |V| + \sum_{t=1}^{|V|} N(x_t, y = k)}$$

What if $N(x_i, y = k) = 0$?

$$P(x_i|y = k) = \frac{N(x_i, y = k)}{\sum_{t=1}^{|V|} N(x_t, y = k)}$$

Imagine we haven't seen **pterodactyl** or **abracadabra** in training texts $\Longrightarrow$ $N(x_i, y = k) = 0$

A simple trick: add a small δ to all counts:

$$P(x_i|y = k) = \frac{\delta + N(x_i, y = k)}{\sum_{t=1}^{|V|} (\delta + N(x_t, y = k))} = \frac{\delta + N(x_i, y = k)}{\delta \cdot |V| + \sum_{t=1}^{|V|} N(x_t, y = k)}$$

Note: this is Laplace (add-one) smoothing – we'll see it in the text lecture!

Making a Prediction

Data: $x =$ This film is awesome !

x_1 x_2 x_3 x_4 x_5

Making a Prediction

Data: $x =$ This film is awesome !

x_1 x_2 x_3 x_4 x_5

$$y^* = \arg\max_k P(x, y = k) = \arg\max_k P(y = k) \cdot P(x|y = k)$$

Making a Prediction

Data: $x =$ This film is awesome !

x_1 x_2 x_3 x_4 x_5

$$y^* = \arg\max_k P(x, y = k) = \arg\max_k P(y = k) \cdot P(x|y = k)$$

Positive class

$$P(x, y = +) = P(y = +) \cdot P(x|y = +)$$

Negative class

$$P(x, y = -) = P(y = -) \cdot P(x|y = -)$$

Making a Prediction

Data: $x =$ This film is awesome !

x_1 x_2 x_3 x_4 x_5

$$y^* = \arg\max_k P(x, y = k) = \arg\max_k P(y = k) \cdot P(x|y = k)$$

Positive class

$$P(x, y = +) = P(y = +) \cdot P(x|y = +) = P(y = +) \cdot$$

Negative class

$$P(x, y = -) = P(y = -) \cdot P(x|y = -) = P(y = -) \cdot$$

Making a Prediction

Data: $x =$ This film is awesome !

x_1 x_2 x_3 x_4 x_5

$$y^* = \arg\max_k P(x, y = k) = \arg\max_k P(y = k) \cdot P(x|y = k)$$

Positive class

$$P(x, y = +)$$
$$= P(y = +) \cdot P(x|y = +)$$
$$= P(y = +) \cdot$$
$$\cdot P(\text{This}|y = +)$$
$$\cdot P(\text{film}|y = +)$$
$$\cdot P(\text{is}|y = +)$$
$$\cdot P(\text{awesome}|y = +)$$
$$\cdot P(!|y = +)$$

Negative class

$$P(x, y = -)$$
$$= P(y = -) \cdot P(x|y = -)$$
$$= P(y = -) \cdot$$
$$\cdot P(\text{This}|y = -)$$
$$\cdot P(\text{film}|y = -)$$
$$\cdot P(\text{is}|y = -)$$
$$\cdot P(\text{awesome}|y = -)$$
$$\cdot P(!|y = -)$$

Making a Prediction

Data: $x =$ This film is awesome !

x_1 x_2 x_3 x_4 x_5

$$y^* = \arg\max_k P(x, y = k) = \arg\max_k P(y = k) \cdot P(x|y = k)$$

Positive class

$$P(x, y = +)$$
$$= P(y = +) \cdot P(x|y = +)$$
$$= P(y = +) \cdot$$
$$\cdot P(\text{This}|y = +)$$
$$\cdot P(\text{film}|y = +)$$
$$\cdot P(\text{is}|y = +)$$
$$\cdot P(\text{awesome}|y = +)$$
$$\cdot P(!|y = +)$$

Negative class

$$P(x, y = -)$$
$$= P(y = -) \cdot P(x|y = -)$$
$$= P(y = -) \cdot$$
$$\cdot P(\text{This}|y = -)$$
$$\cdot P(\text{film}|y = -)$$
$$\cdot P(\text{is}|y = -)$$
$$\cdot P(\text{awesome}|y = -)$$
$$\cdot P(!|y = -)$$

Making a Prediction

Data: $x =$ This film is awesome !

x_1 x_2 x_3 x_4 x_5

$$y^* = \arg\max_k P(x, y = k) = \arg\max_k P(y = k) \cdot P(x|y = k)$$

Positive class

$$P(x, y = +) = P(y = +) \cdot P(x|y = +)$$

$$= P(y = +) \cdot \left[\cdot P(\text{This}|y = +) \cdot P(\text{film}|y = +) \cdot P(\text{is}|y = +) \cdot P(\text{awesome}|y = +) \cdot P(!|y = +) \right]$$

Prior class probability (often 0.5)

Negative class

$$P(x, y = -) = P(y = -) \cdot P(x|y = -)$$

$$= P(y = -) \cdot \left[\cdot P(\text{This}|y = -) \cdot P(\text{film}|y = -) \cdot P(\text{is}|y = -) \cdot P(\text{awesome}|y = -) \cdot P(!|y = -) \right]$$

Making a Prediction

Data: $x =$ This film is awesome !

x_1 x_2 x_3 x_4 x_5

$$y^* = \arg\max_k P(x, y = k) = \arg\max_k P(y = k) \cdot P(x|y = k)$$

Positive class

$$P(x, y = +) = P(y = +) \cdot P(x|y = +)$$

$$= P(y = +) \cdot \left[\begin{array}{l} \cdot P(\text{This}|y = +) \\ \cdot P(\text{film}|y = +) \\ \cdot P(\text{is}|y = +) \\ \cdot P(\text{awesome}|y = +) \\ \cdot P(!|y = +) \end{array}\right]$$

Prior class probability (often 0.5)

Neutral words – not much difference in probabilities for classes

Negative class

$$P(x, y = -) = P(y = -) \cdot P(x|y = -)$$

$$= P(y = -) \cdot \left[\begin{array}{l} \cdot P(\text{This}|y = -) \\ \cdot P(\text{film}|y = -) \\ \cdot P(\text{is}|y = -) \\ \cdot P(\text{awesome}|y = -) \\ \cdot P(!|y = -) \end{array}\right]$$

Making a Prediction

Data: $x =$ This film is awesome !

x_1 x_2 x_3 x_4 x_5

$$y^* = \arg\max_k P(x, y = k) = \arg\max_k P(y = k) \cdot P(x|y = k)$$

Positive class

$$P(x, y = +) = P(y = +) \cdot P(\text{This}|y = +) \cdot P(\text{film}|y = +) \cdot P(\text{is}|y = +) \cdot P(\text{awesome}|y = +) \cdot P(!|y = +)$$

Prior class probability (often 0.5)

Neutral words – not much difference in probabilities for classes

This is where we expect to see the difference!

Negative class

$$P(x, y = -) = P(y = -) \cdot P(\text{This}|y = -) \cdot P(\text{film}|y = -) \cdot P(\text{is}|y = -) \cdot P(\text{awesome}|y = -) \cdot P(!|y = -)$$

Making a Prediction

Data: $x =$ This film is awesome !
x_1 x_2 x_3 x_4 x_5

$$y^* = \arg\max_k P(x, y = k) = \arg\max_k P(y = k) \cdot P(x|y = k)$$

Positive class

$$P(x, y = +) = P(y = +) \cdot P(x|y = +)$$

$$= P(y = +) \cdot P(\text{This}|y = +) \cdot P(\text{film}|y = +) \cdot P(\text{is}|y = +) \cdot P(\text{awesome}|y = +) \cdot P(!|y = +)$$

Prior class probability (often 0.5)

Neutral words – not much difference in probabilities for classes

This is where we expect to see the difference!

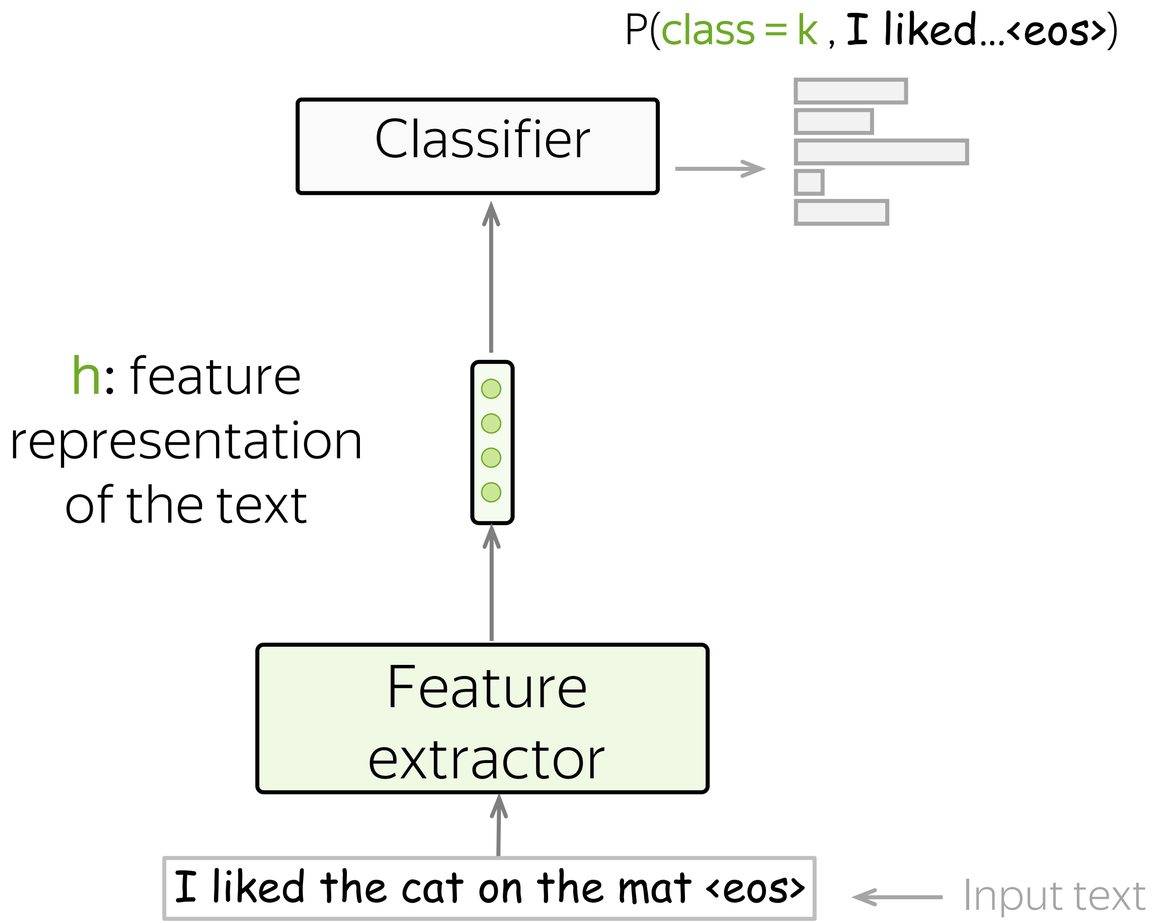
Negative class

$$P(x, y = -) = P(y = -) \cdot P(x|y = -)$$

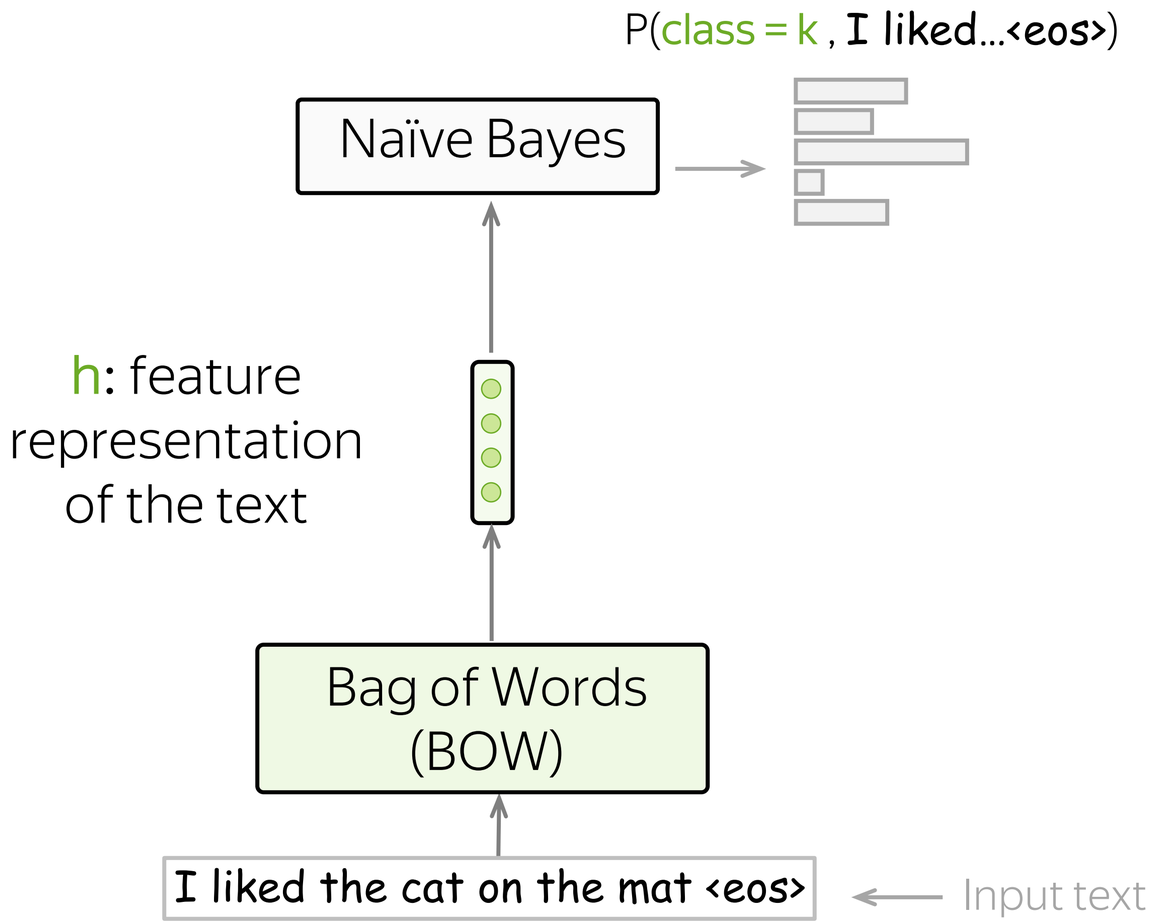
$$= P(y = -) \cdot P(\text{This}|y = -) \cdot P(\text{film}|y = -) \cdot P(\text{is}|y = -) \cdot P(\text{awesome}|y = -) \cdot P(!|y = -)$$

$$P(\text{awesome}|y = +) \gg P(\text{awesome}|y = -)$$

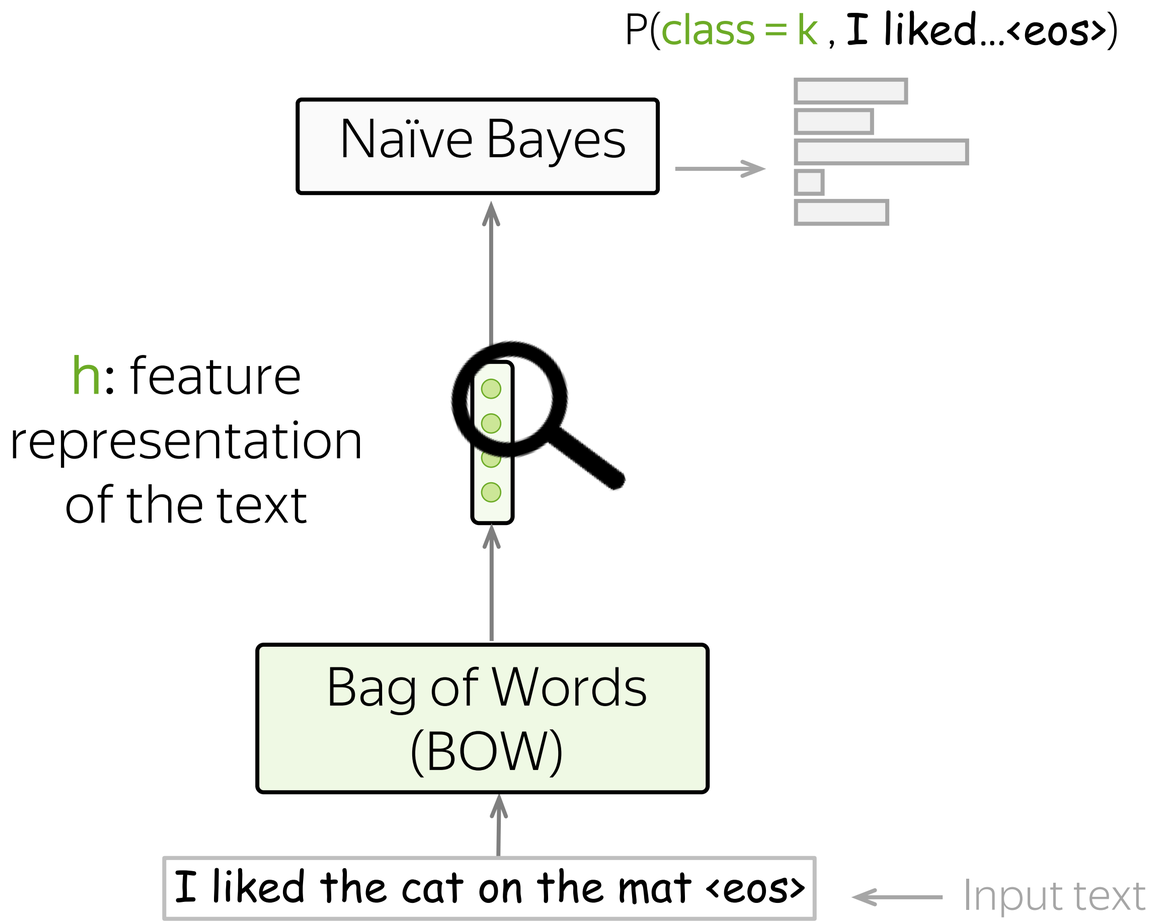
Naïve Bayes: View in our General Framework



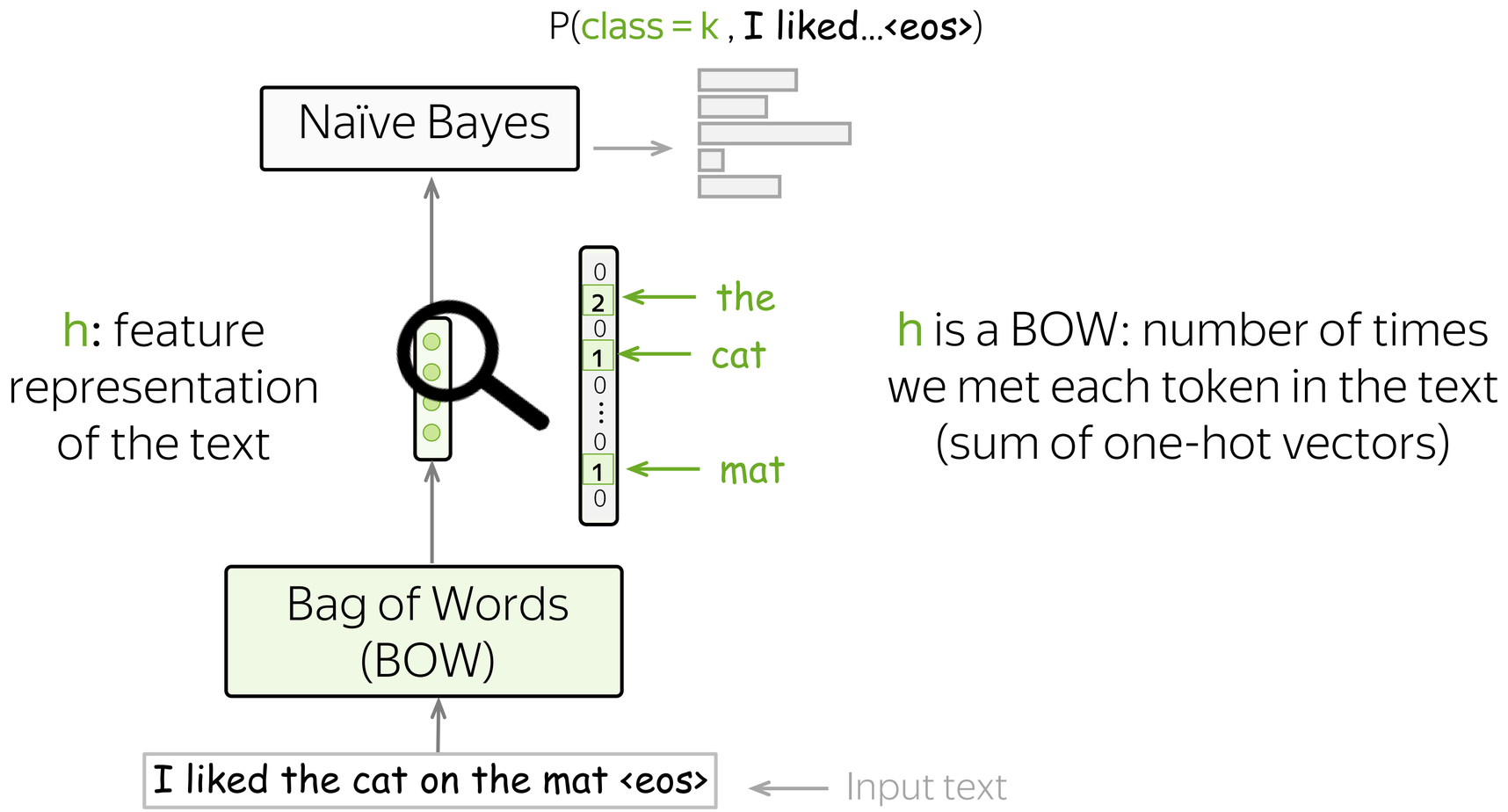
Naïve Bayes: View in our General Framework



Naïve Bayes: View in our General Framework



Naïve Bayes: View in our General Framework



Problems of Naïve Bayes

This is rather good:
not bad at all!

This is rather bad:
not good at all!

this, is, good, bad,
rather, not, at, all, !

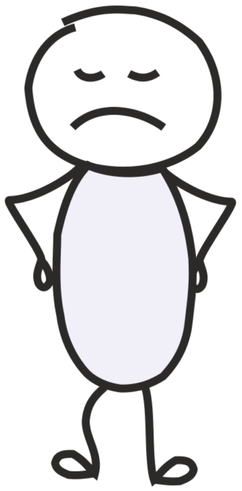
Naïve Bayes features

NLP Course For You

Improve Naïve Bayes!

The method is old, then why do we need this?

Obviously, Naïve Bayes is not the best method, and now we have all these cool huge neural networks.



Lena, then why you are telling us all this?

Students (aka “некоторые редиски”)

We have to explain model decisions

In 2018, the European Union General Data Protection Regulation (GDPR) regulated any significant or legally related decision to be explainable. The subject can require human intervention to challenge the decision.

We have to explain model decisions

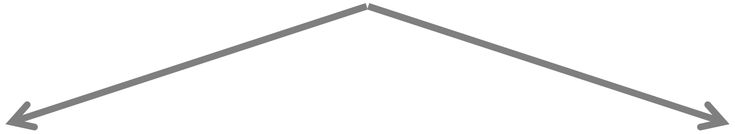
In 2018, the European Union General Data Protection Regulation (GDPR) regulated any significant or legally related decision to be explainable. The subject can require human intervention to challenge the decision.

We need to provide human-understandable explanation of why a model made some decision.

We have to explain model decisions

In 2018, the European Union General Data Protection Regulation (GDPR) regulated any significant or legally related decision to be explainable. The subject can require human intervention to challenge the decision.

We need to provide human-understandable explanation of why a model made some decision.



Neural Networks:

- not interpretable (unless we do something specifically for this).

Classical approaches:

- often interpretable by design
- may win in terms of “quality+interpretability” trade-off

What is going to happen:

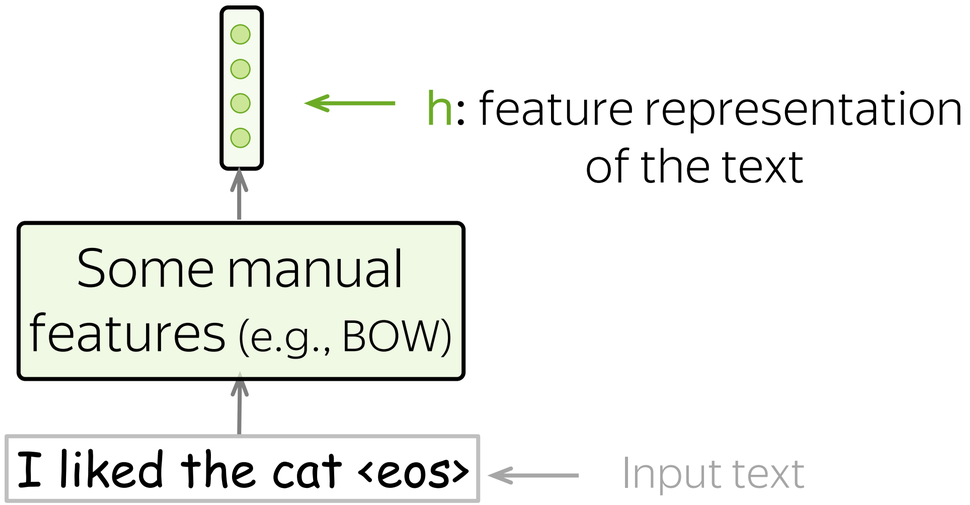
- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
  - Naïve Bayes
  - Logistic Regression
  - SVM
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
  - Naïve Bayes
  - Logistic Regression
  - SVM
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

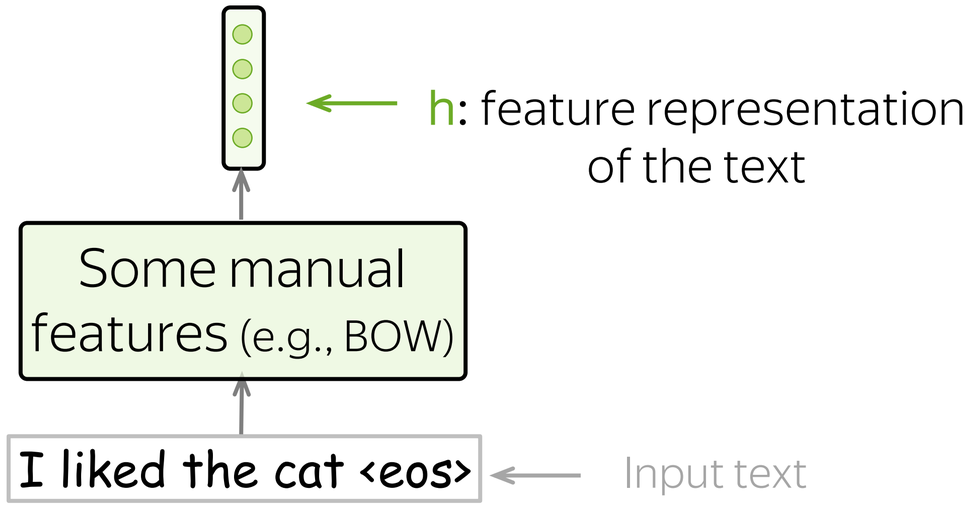
Logistic Regression

- $h = (f_1, f_2, \ldots, f_n)$ – features of an input text



Logistic Regression

- $h = (f_1, f_2, \ldots, f_n)$ – features of an input text
- $w^{(k)} = (w_1^{(k)}, w_2^{(k)}, \ldots, w_n^{(k)})$ – feature weights for class k

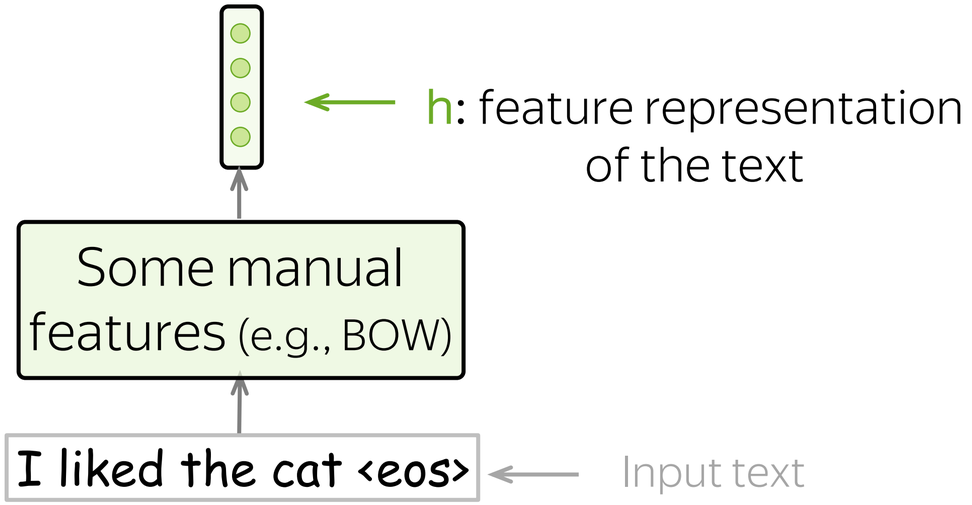


Logistic Regression

- $h = (f_1, f_2, \ldots, f_n)$ – features of an input text
- $w^{(k)} = (w_1^{(k)}, w_2^{(k)}, \ldots, w_n^{(k)})$ – feature weights for class k

$$w^{(k)} h = w_1^{(k)} \cdot f_1 + \cdots + w_n^{(k)} \cdot f_n, \qquad k = 1, \ldots, K$$

$$P(class = k|h) = \frac{\exp(w^{(k)} h)}{\sum_{i=1}^{K} \exp(w^{(i)} h)}$$

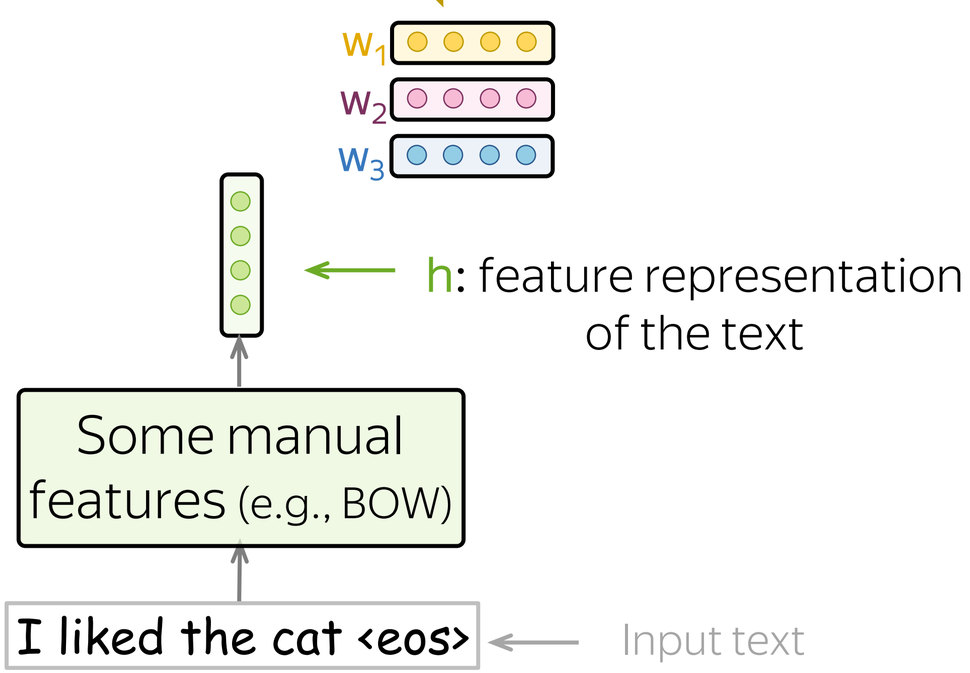


Logistic Regression

- $h = (f_1, f_2, \ldots, f_n)$ – features of an input text
- $w^{(k)} = (w_1^{(k)}, w_2^{(k)}, \ldots, w_n^{(k)})$ – feature weights for class k

$$w^{(k)} h = w_1^{(k)} \cdot f_1 + \cdots + w_n^{(k)} \cdot f_n, \qquad k = 1, \ldots, K$$

$$P(class = k|h) = \frac{\exp(w^{(k)} h)}{\sum_{i=1}^{K} \exp(w^{(i)} h)}$$

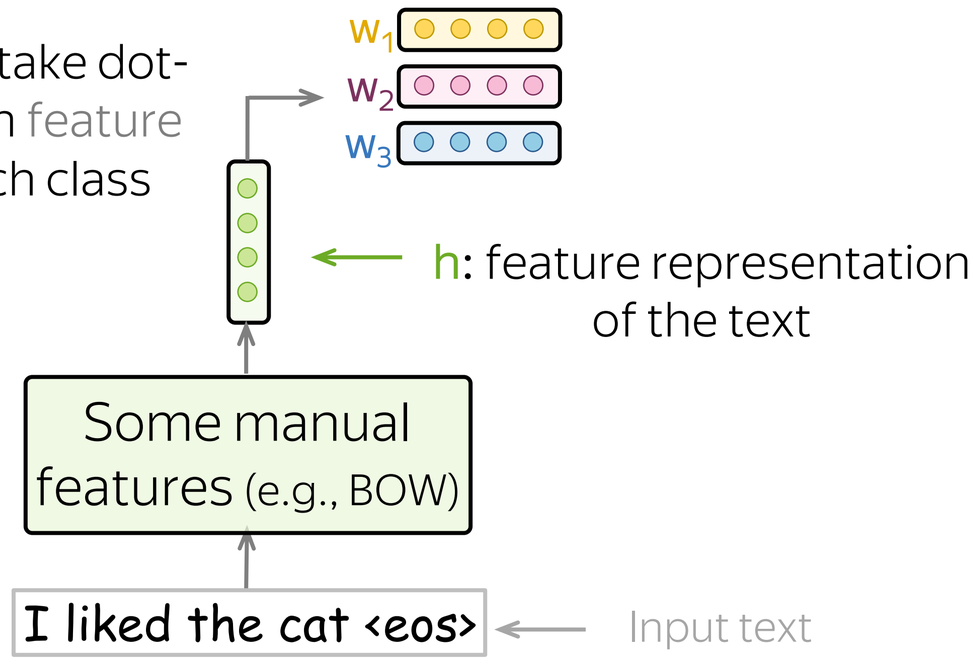


Logistic Regression

- $h = (f_1, f_2, \ldots, f_n)$ – features of an input text
- $w^{(k)} = (w_1^{(k)}, w_2^{(k)}, \ldots, w_n^{(k)})$ – feature weights for class k

$$w^{(k)} h = w_1^{(k)} \cdot f_1 + \cdots + w_n^{(k)} \cdot f_n, \qquad k = 1, \ldots, K$$

$$P(class = k|h) = \frac{\exp(w^{(k)} h)}{\sum_{i=1}^{K} \exp(w^{(i)} h)}$$

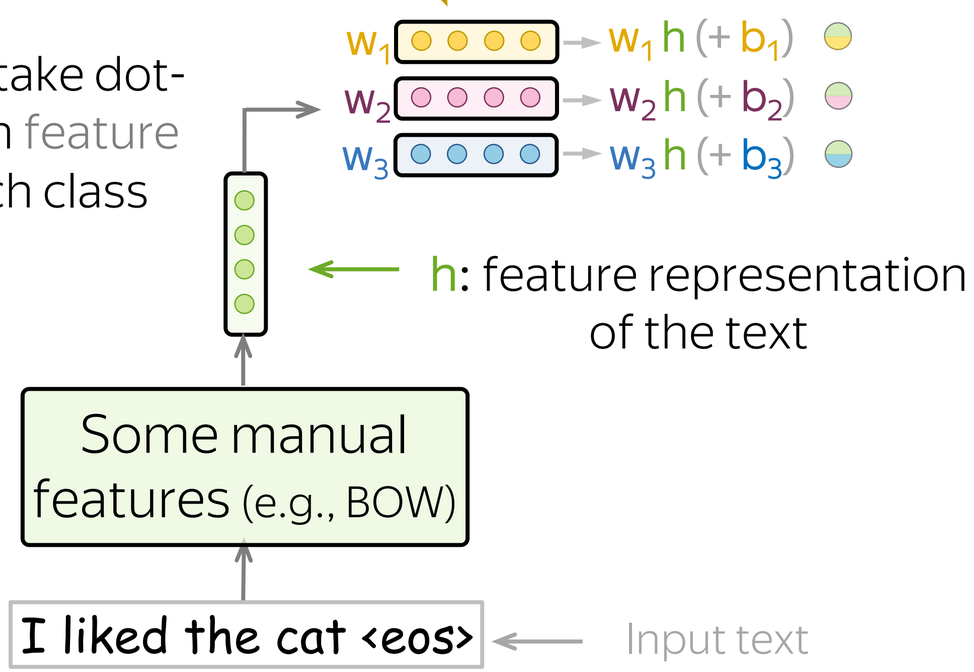


Logistic Regression

- $h = (f_1, f_2, \ldots, f_n)$ – features of an input text
- $w^{(k)} = (w_1^{(k)}, w_2^{(k)}, \ldots, w_n^{(k)})$ – feature weights for class k

$$w^{(k)} h = w_1^{(k)} \cdot f_1 + \cdots + w_n^{(k)} \cdot f_n, \qquad k = 1, \ldots, K$$

$$P(class = k|h) = \frac{\exp(w^{(k)} h)}{\sum_{i=1}^{K} \exp(w^{(i)} h)}$$

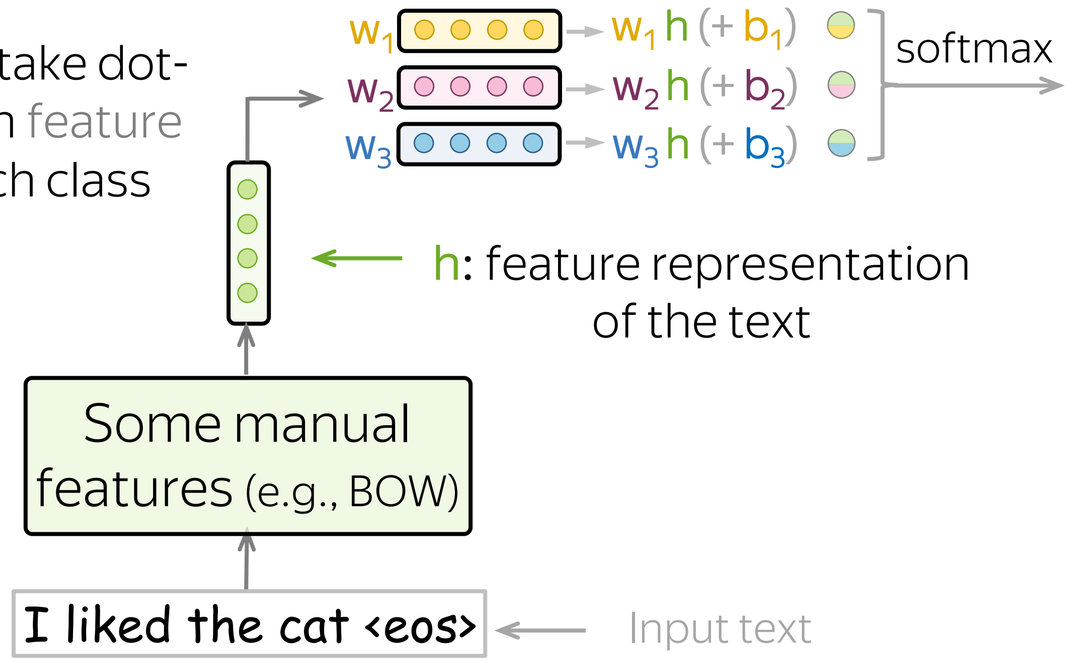


Logistic Regression

- $h = (f_1, f_2, \ldots, f_n)$ – features of an input text
- $w^{(k)} = (w_1^{(k)}, w_2^{(k)}, \ldots, w_n^{(k)})$ – feature weights for class k

$$w^{(k)}h = w_1^{(k)} \cdot f_1 + \cdots + w_n^{(k)} \cdot f_n, \qquad k = 1, \ldots, K$$

$$P(class = k|h) = \frac{\exp(w^{(k)}h)}{\sum_{i=1}^{K} \exp(w^{(i)}h)}$$

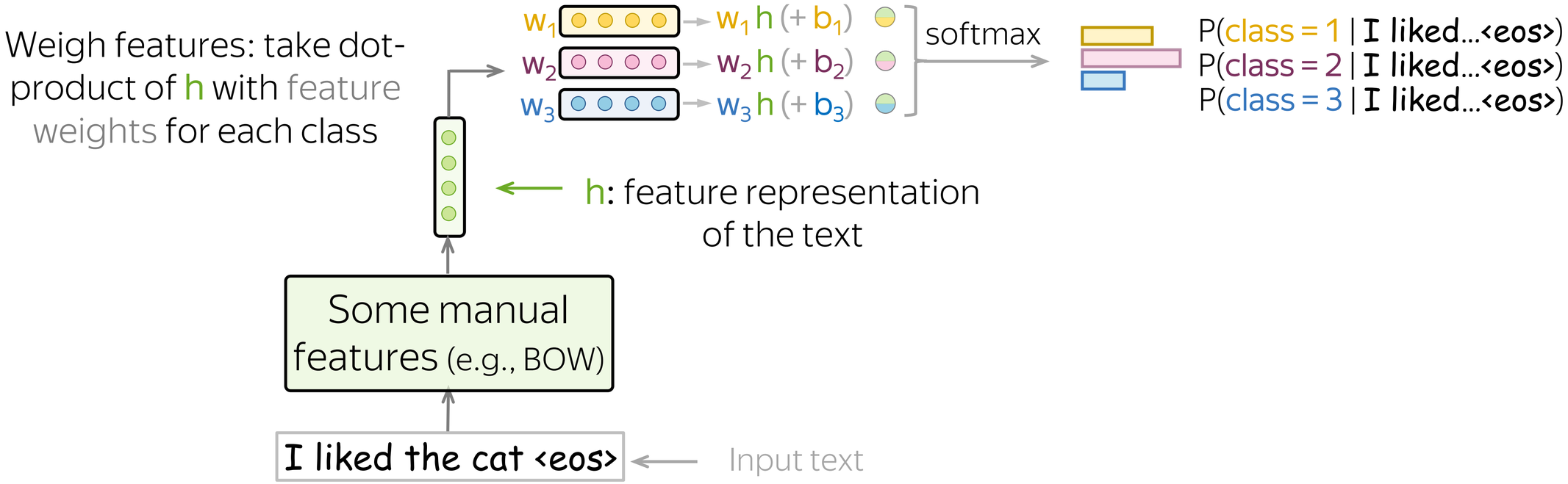


Logistic Regression

- $h = (f_1, f_2, \ldots, f_n)$ – features of an input text
- $w^{(k)} = (w_1^{(k)}, w_2^{(k)}, \ldots, w_n^{(k)})$ – feature weights for class k

$$w^{(k)} h = w_1^{(k)} \cdot f_1 + \cdots + w_n^{(k)} \cdot f_n, \qquad k = 1, \ldots, K$$

$$P(class = k|h) = \frac{\exp(w^{(k)} h)}{\sum_{i=1}^{K} \exp(w^{(i)} h)}$$



Training: Maximizing Likelihood

Training example: **I liked the cat on the mat <eos>** Label: k

target

$\log P(y = k|x) \rightarrow max$ Maximizing log-likelihood of the correct class

Training: Maximizing Likelihood

Training example: **I liked the cat on the mat <eos>** Label: k

target

$$\log P(y = k|x) \rightarrow max$$

$$\Updownarrow$$

$$-\log P(y = k|x) \rightarrow min$$

Maximizing log-likelihood of the correct class

Minimizing negative log-likelihood of the correct class

Training: Maximizing Likelihood

Training example: **I liked the cat on the mat <eos>** Label: k

target

$$\log P(y = k|x) \rightarrow max$$

Maximizing log-likelihood of the correct class

$$\Updownarrow$$

$$-\log P(y = k|x) \rightarrow min$$

Minimizing negative log-likelihood of the correct class

$$\Updownarrow$$

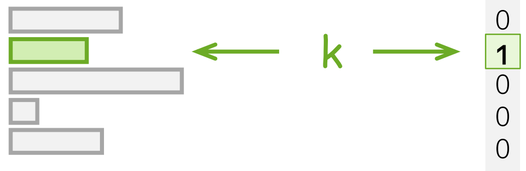
$$-\sum_{i=1}^{K} p_i^* \cdot \log P(y = i|x) \rightarrow min$$

Minimizing cross-entropy loss

$$(p_k^* = 1, p_i^* = 0, i \neq k\ )$$

Model prediction: P(class = i |**I liked…<eos>**)

Target: p^*



Naïve Bayes vs Logistic Regression

Naïve Bayes:

- very simple
- very fast
- interpretable
- assumes that features are conditionally independent
- text representation: manually defined (and often too simple, e.g. BOW)

Logistic Regression:

- quite simple
- interpretable
- does **not** assume that features are conditionally independent
- not so fast (multiple iterations of gradient ascent)
- text representation: manually defined

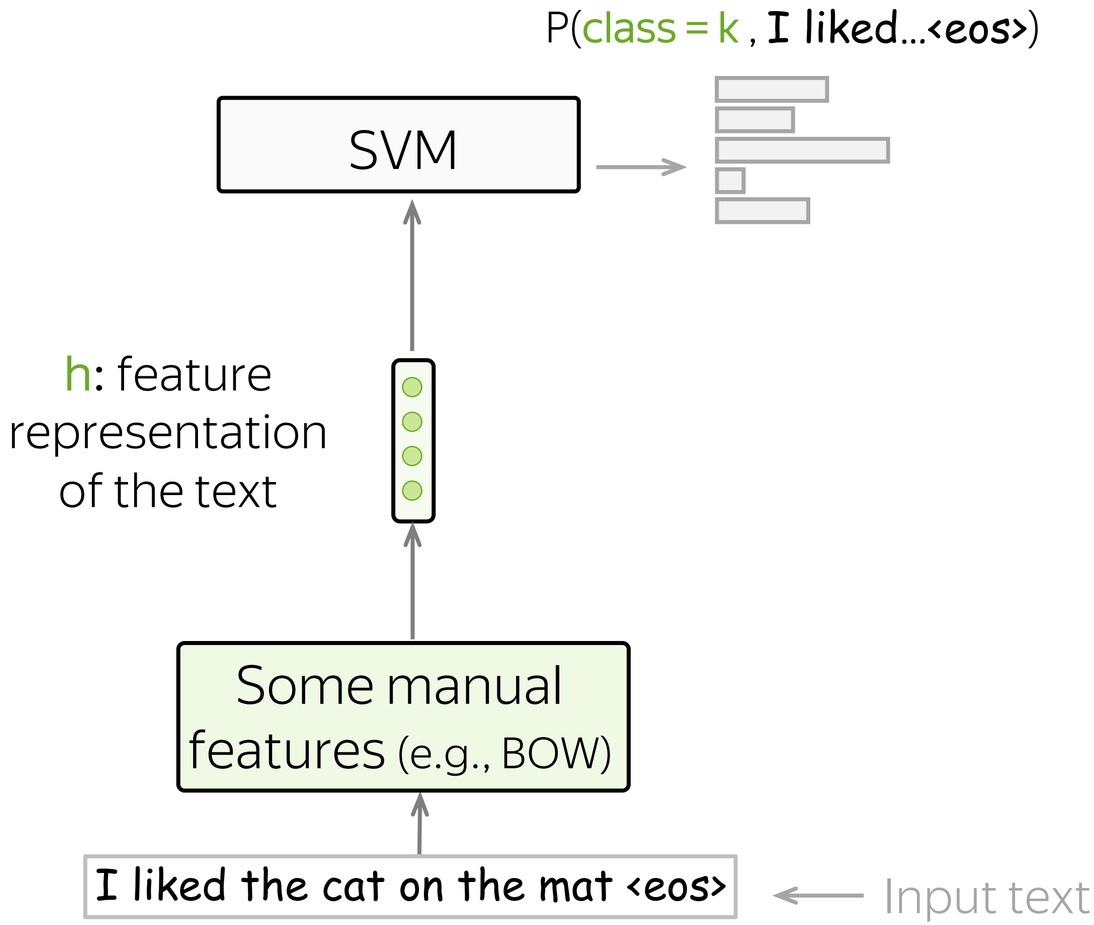
What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
  - Naïve Bayes
  - Logistic Regression
  - SVM
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

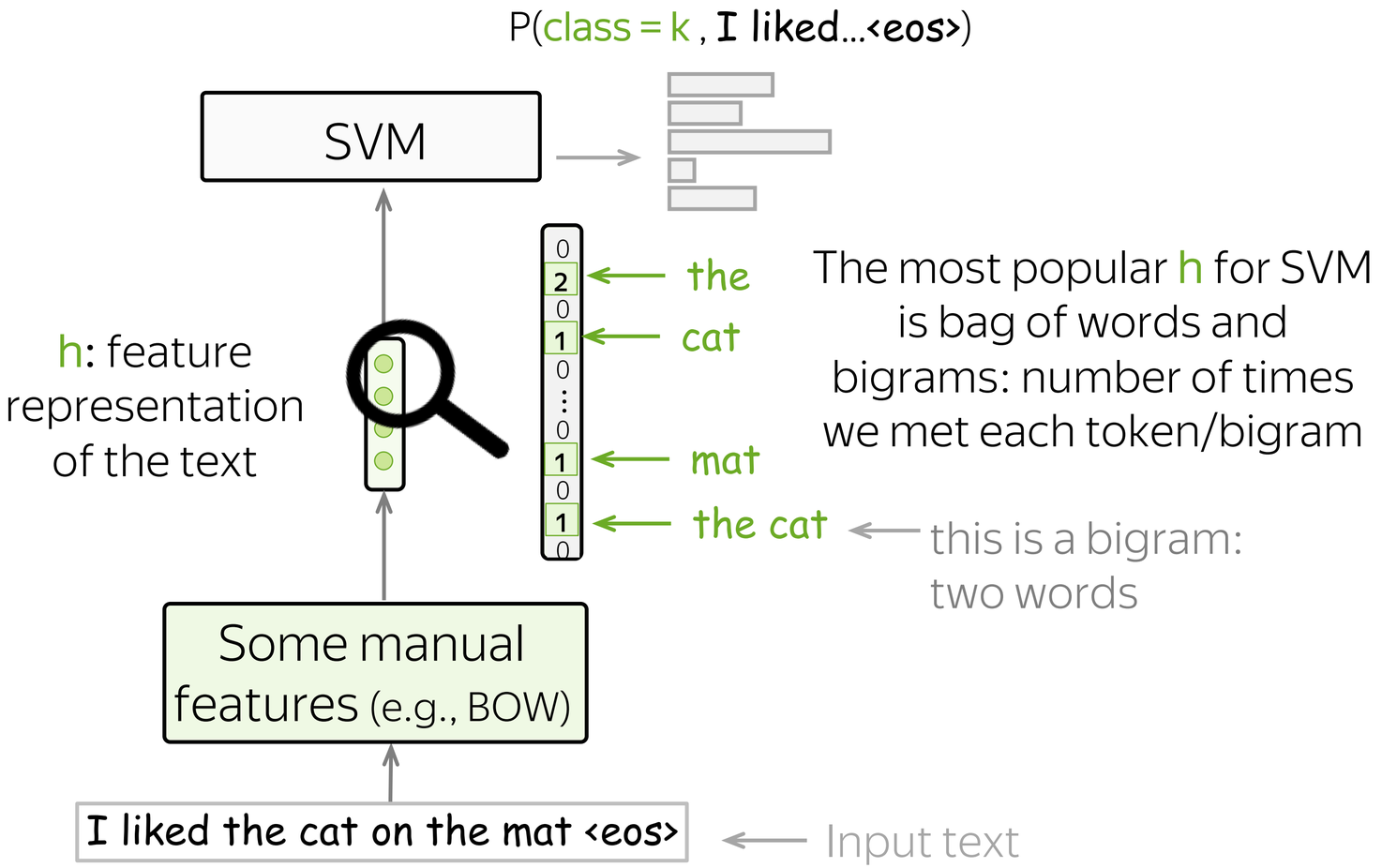
- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
  - Naïve Bayes
  - Logistic Regression
  - SVM
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

SVM: Get Features and Apply SVM



SVM: Get Features and Apply SVM

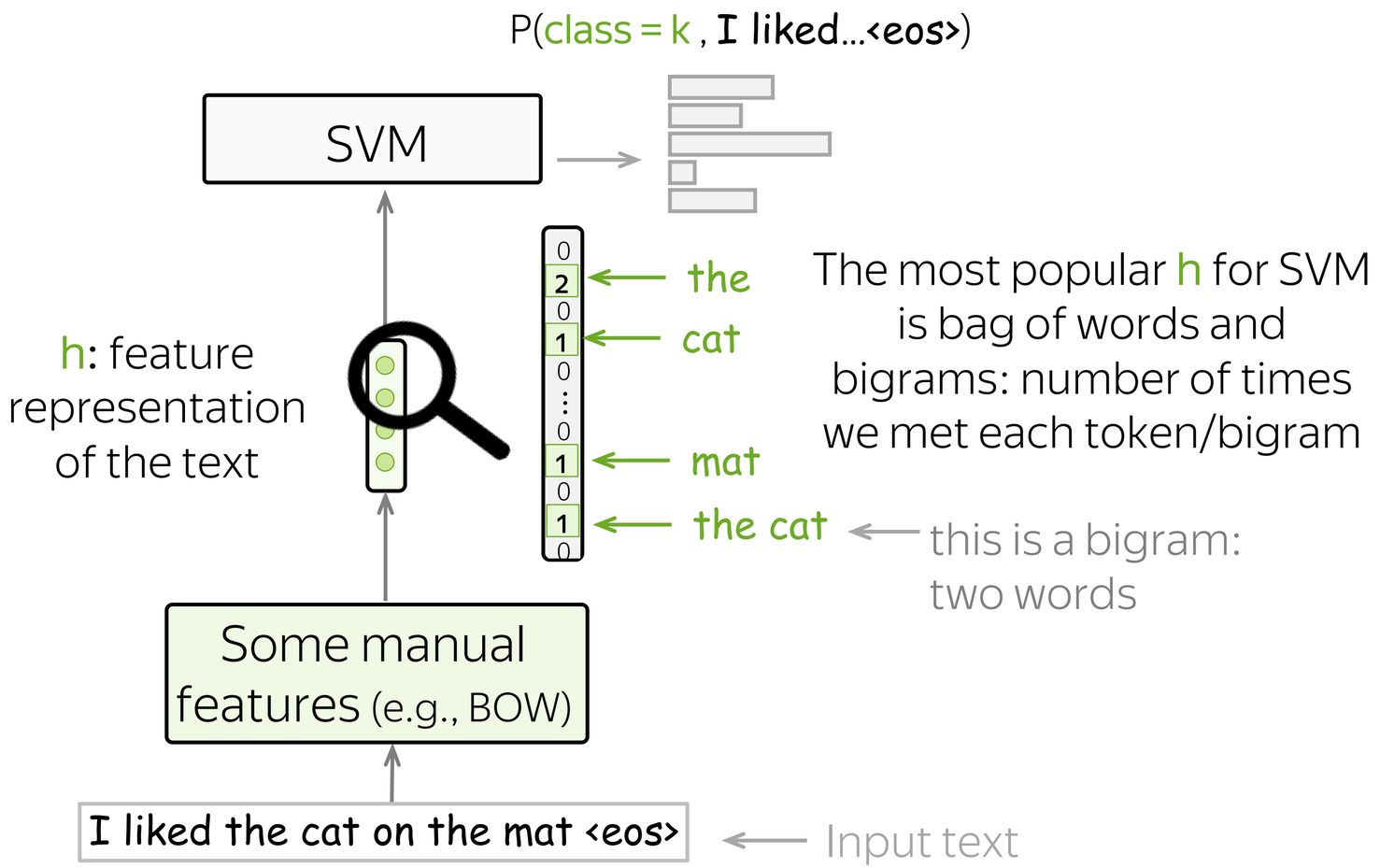
- Features: bag-of-words and ngrams



SVM: Get Features and Apply SVM

- Features: bag-of-words and ngrams
- Result: SVMs are better than Naïve Bayes

  (e.g., see the paper Question Classification using Support Vector Machines)



What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

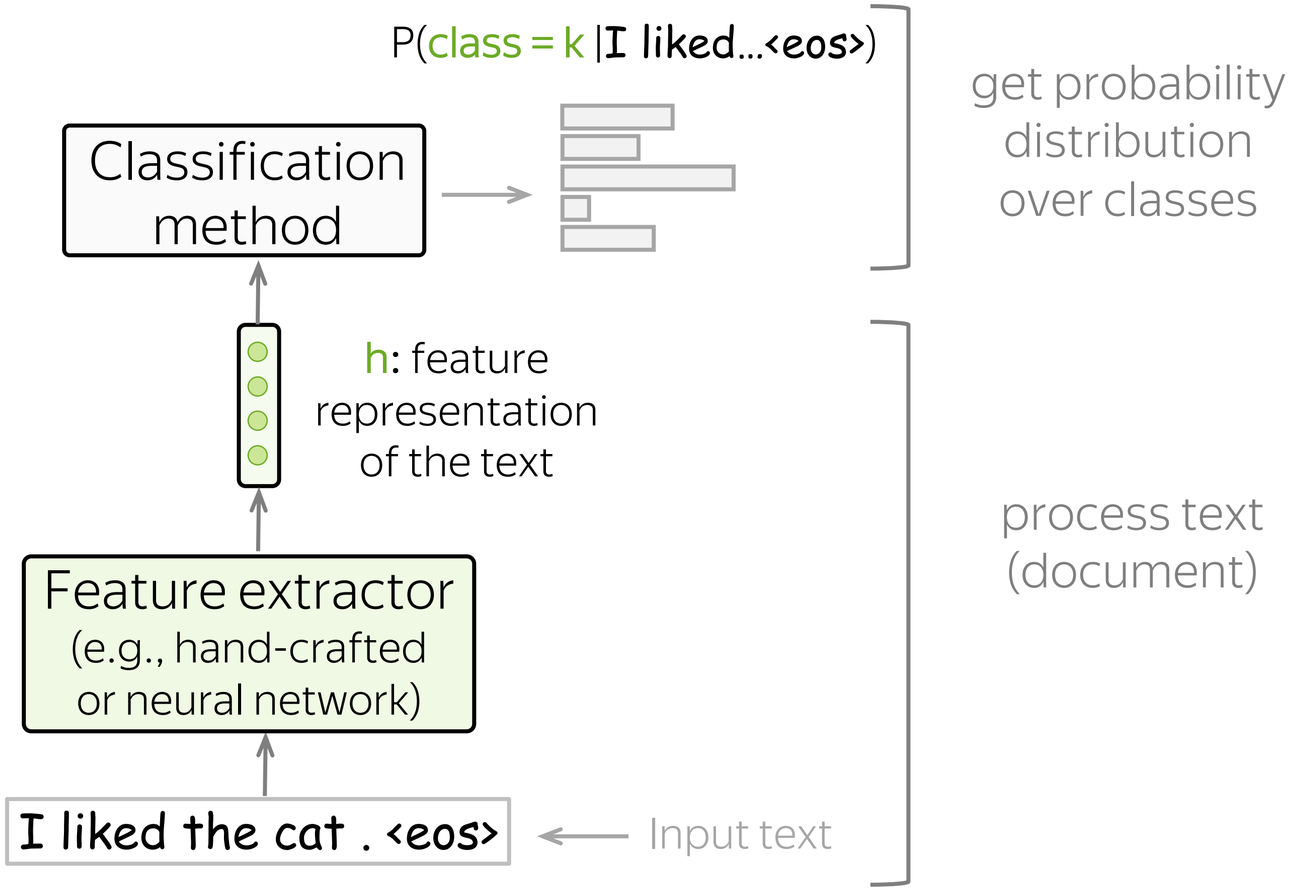
- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
  - High-Level View
  - Training: Cross-Entropy
  - Models: (Weighted) BOW
  - Models: Convolutional
  - Models: Recurrent
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

Classification with Neural Networks

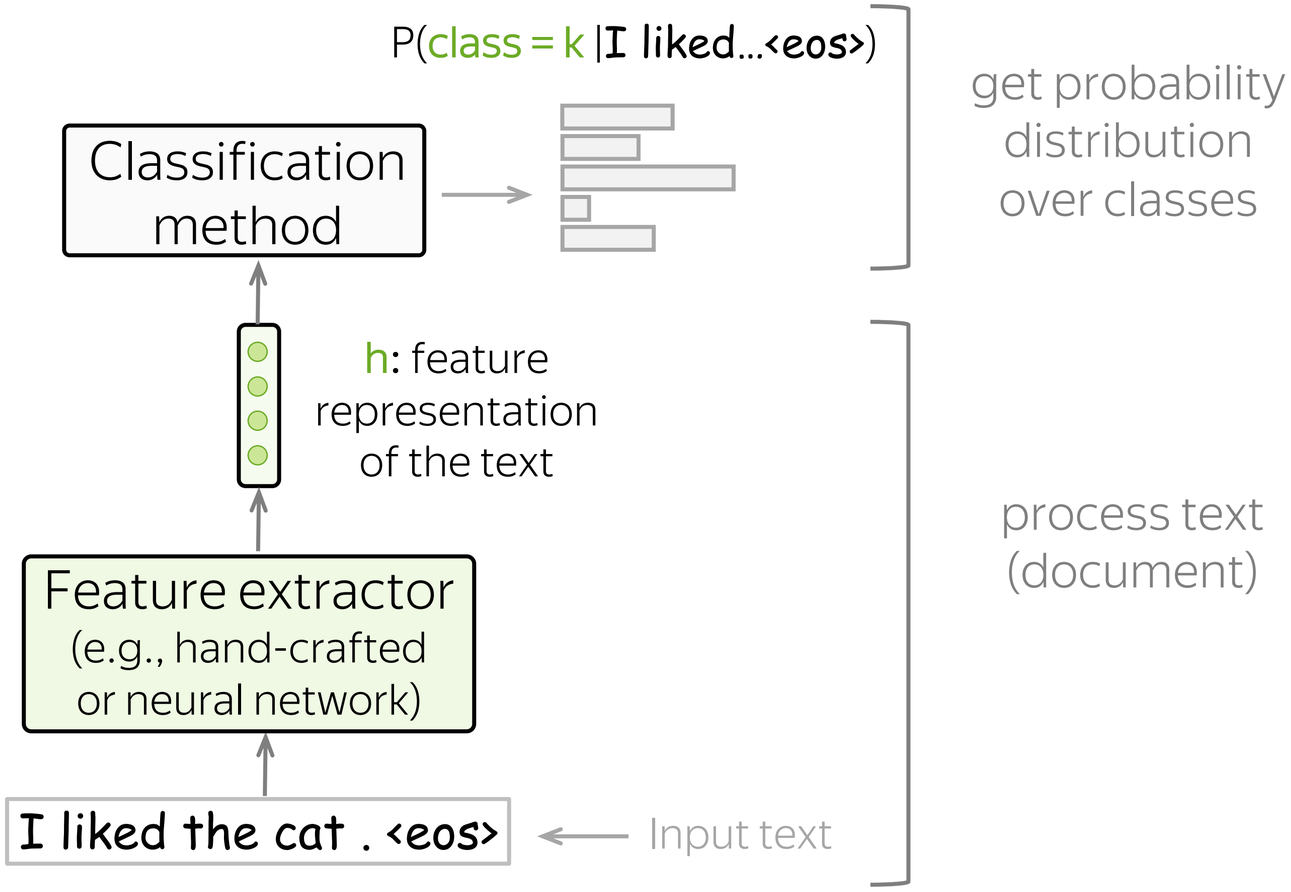
- General Classification Pipeline



Classification with Neural Networks

> Instead of manually defined features, let a neural network to **learn useful features**.

- General Classification Pipeline

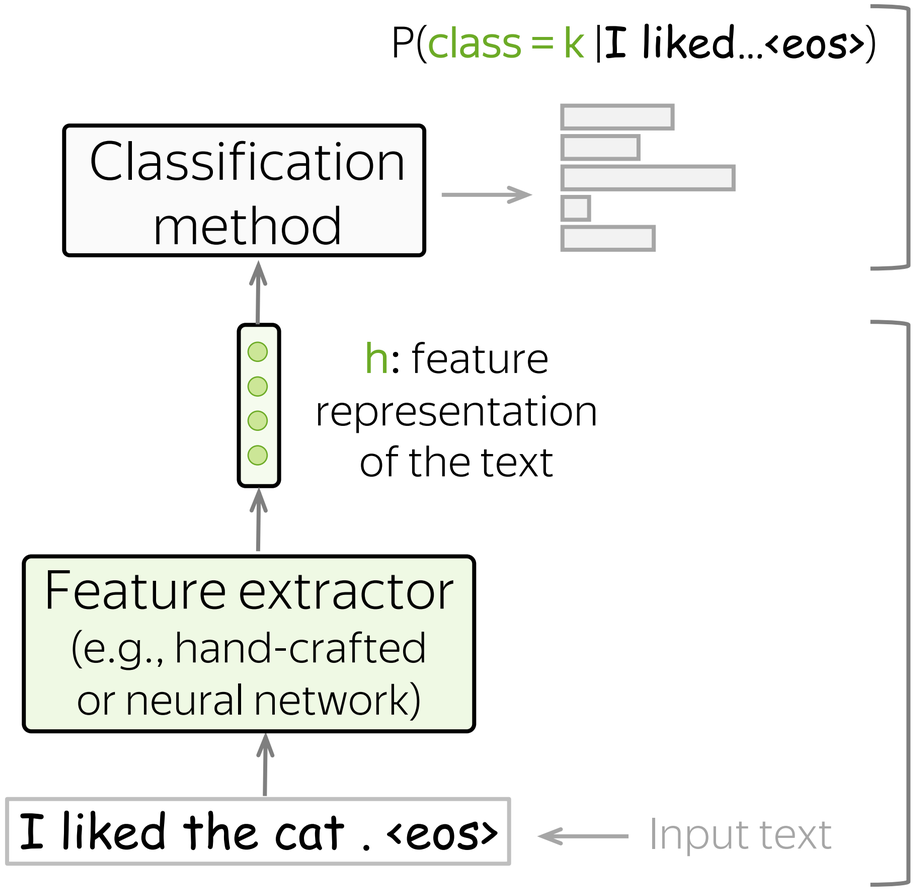


- Classification with Neural Networks

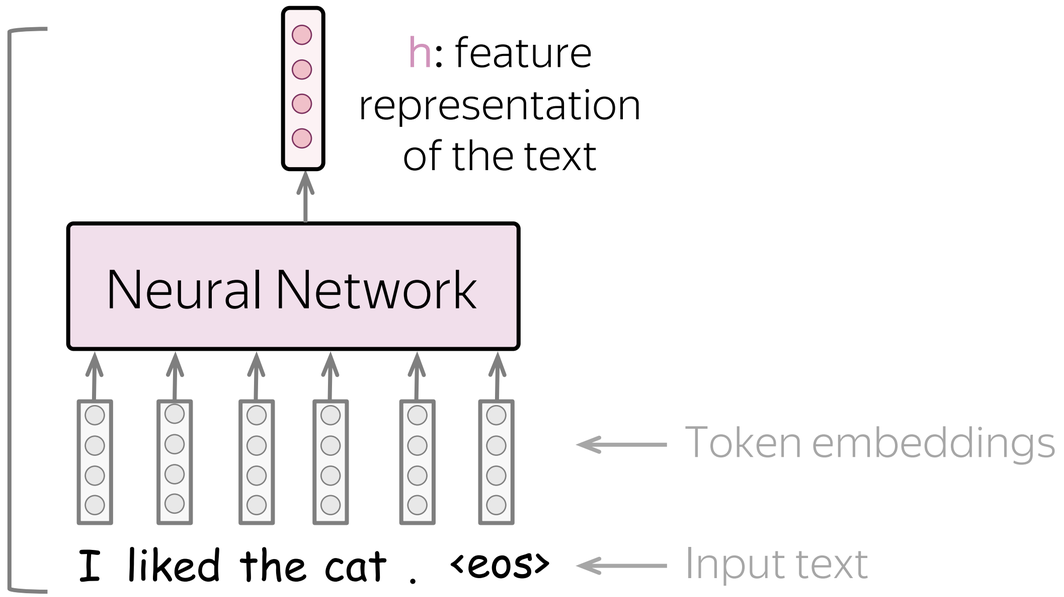
Classification with Neural Networks

> Instead of manually defined features, let a neural network to **learn useful features**.

- General Classification Pipeline



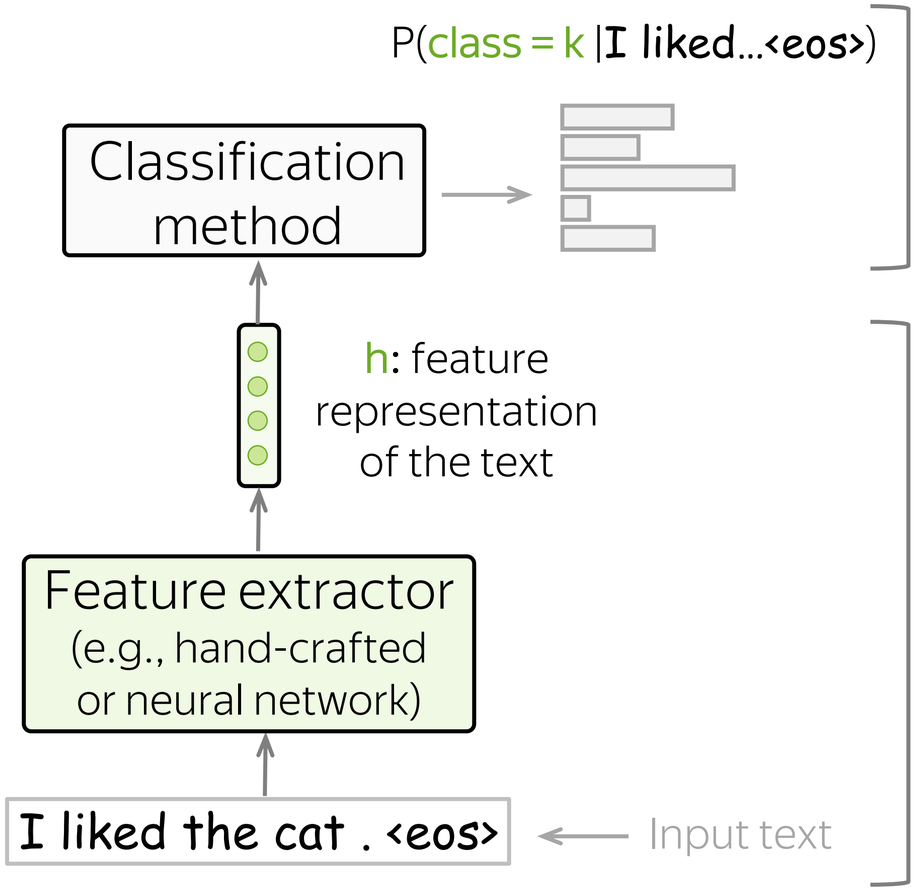
- Classification with Neural Networks



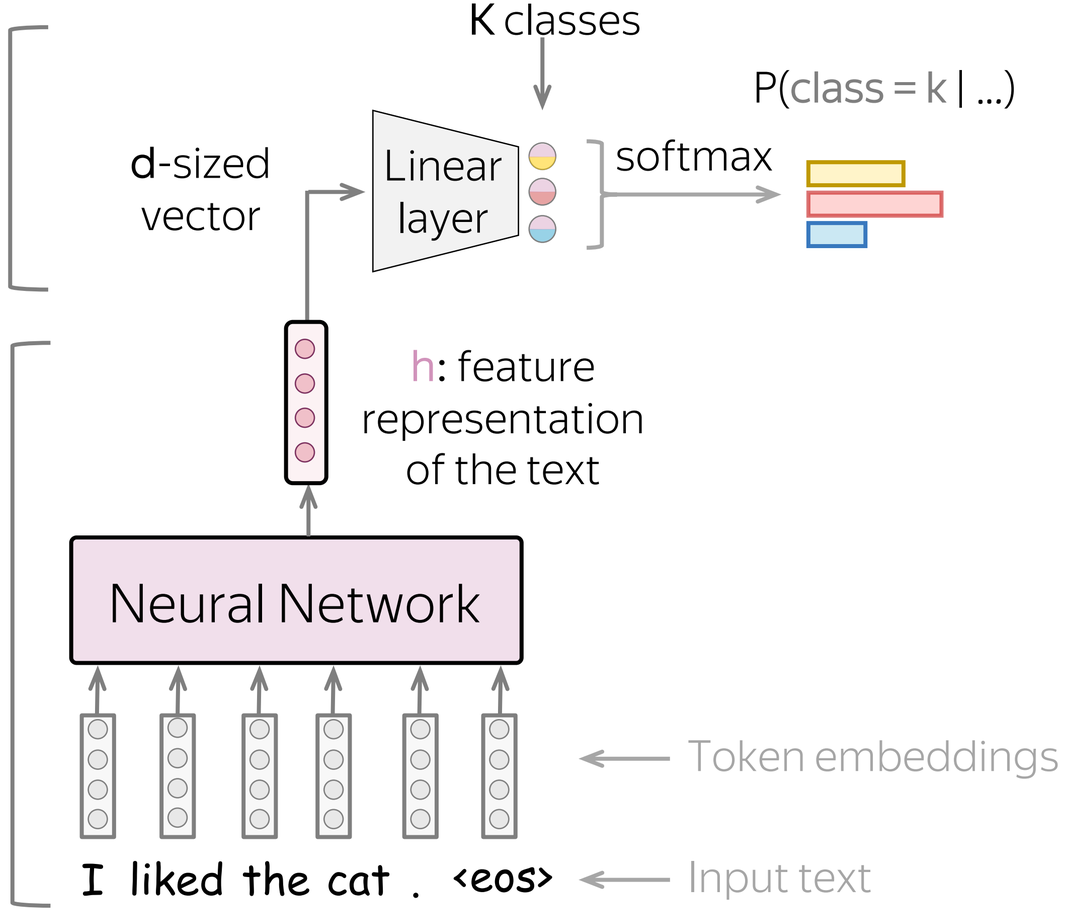
Classification with Neural Networks

> Instead of manually defined features, let a neural network to **learn useful features**.

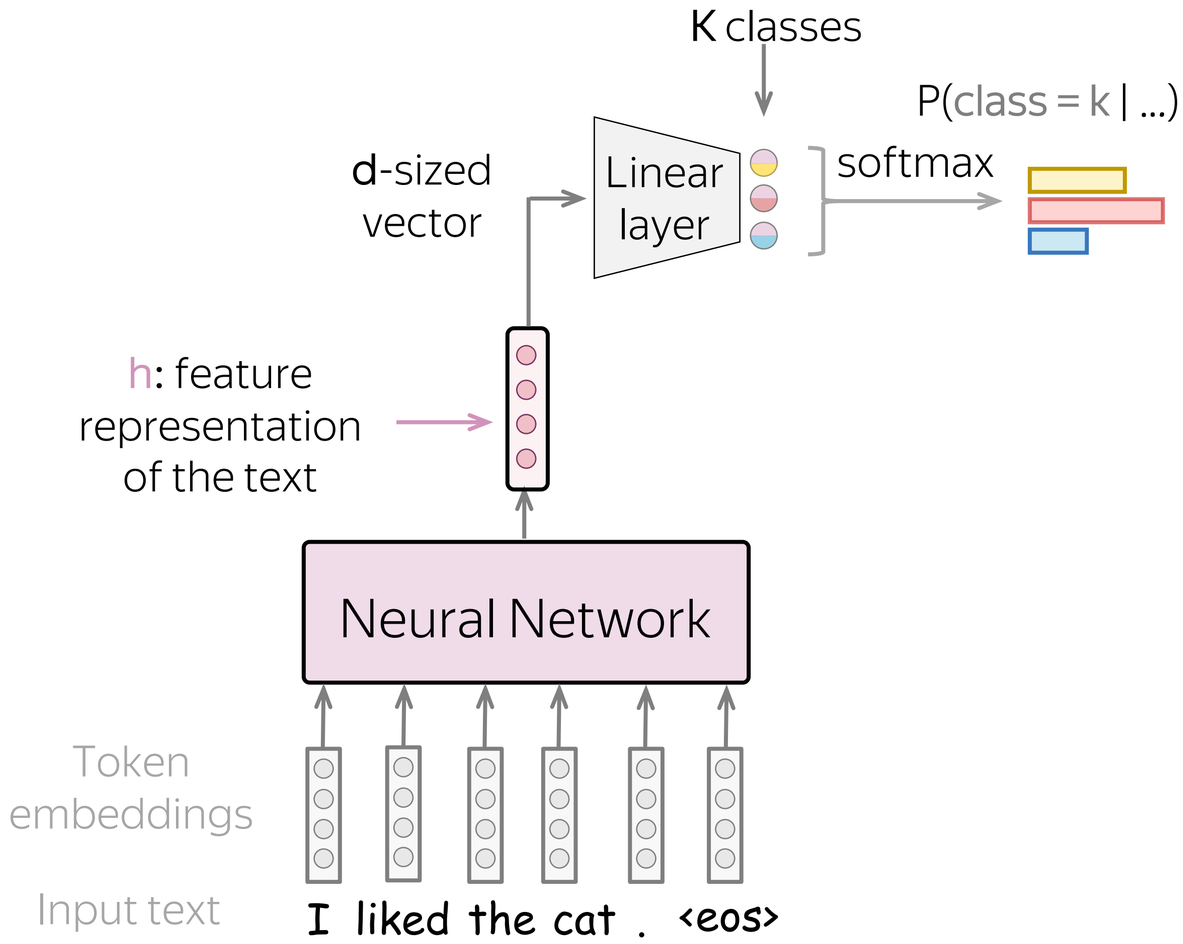
- General Classification Pipeline



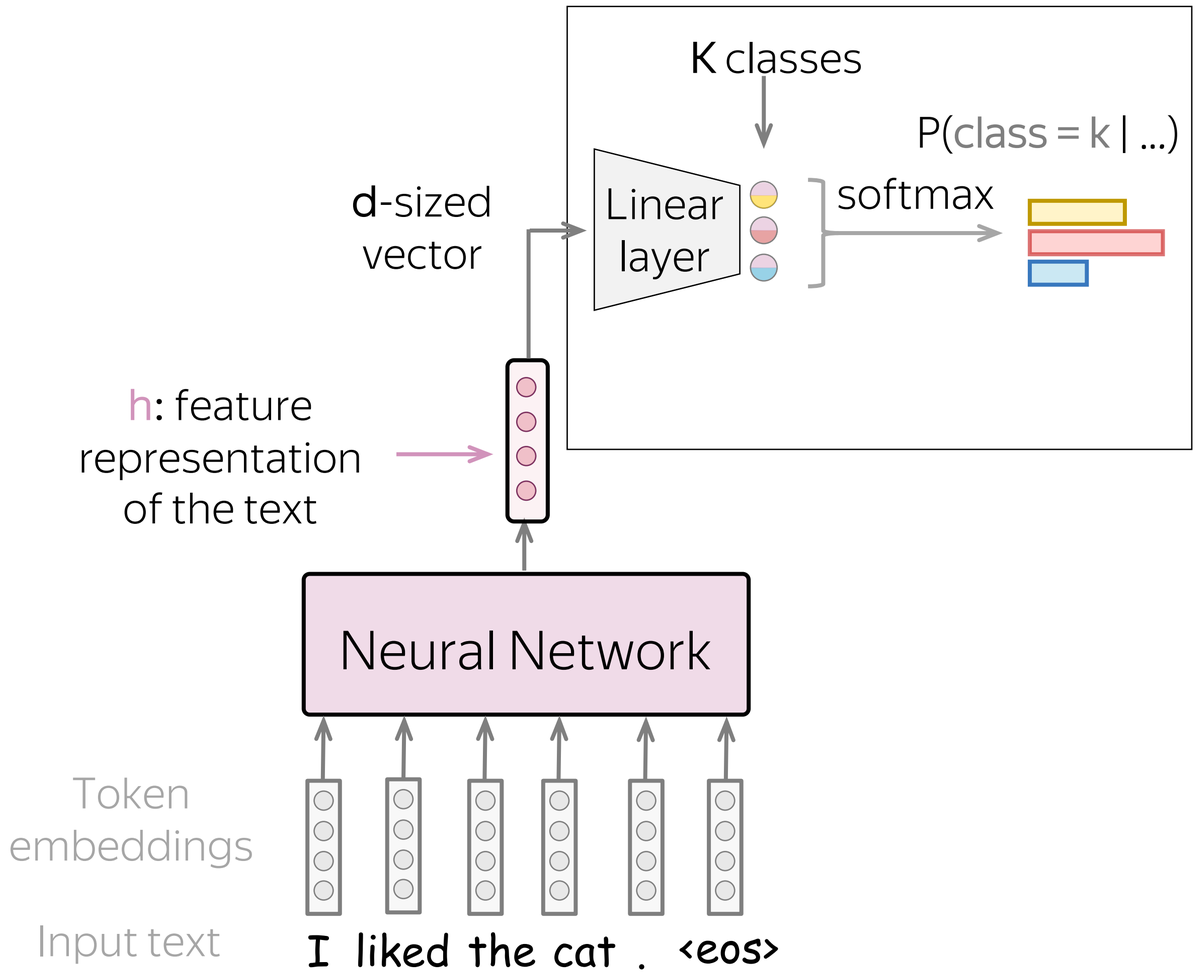
- Classification with Neural Networks



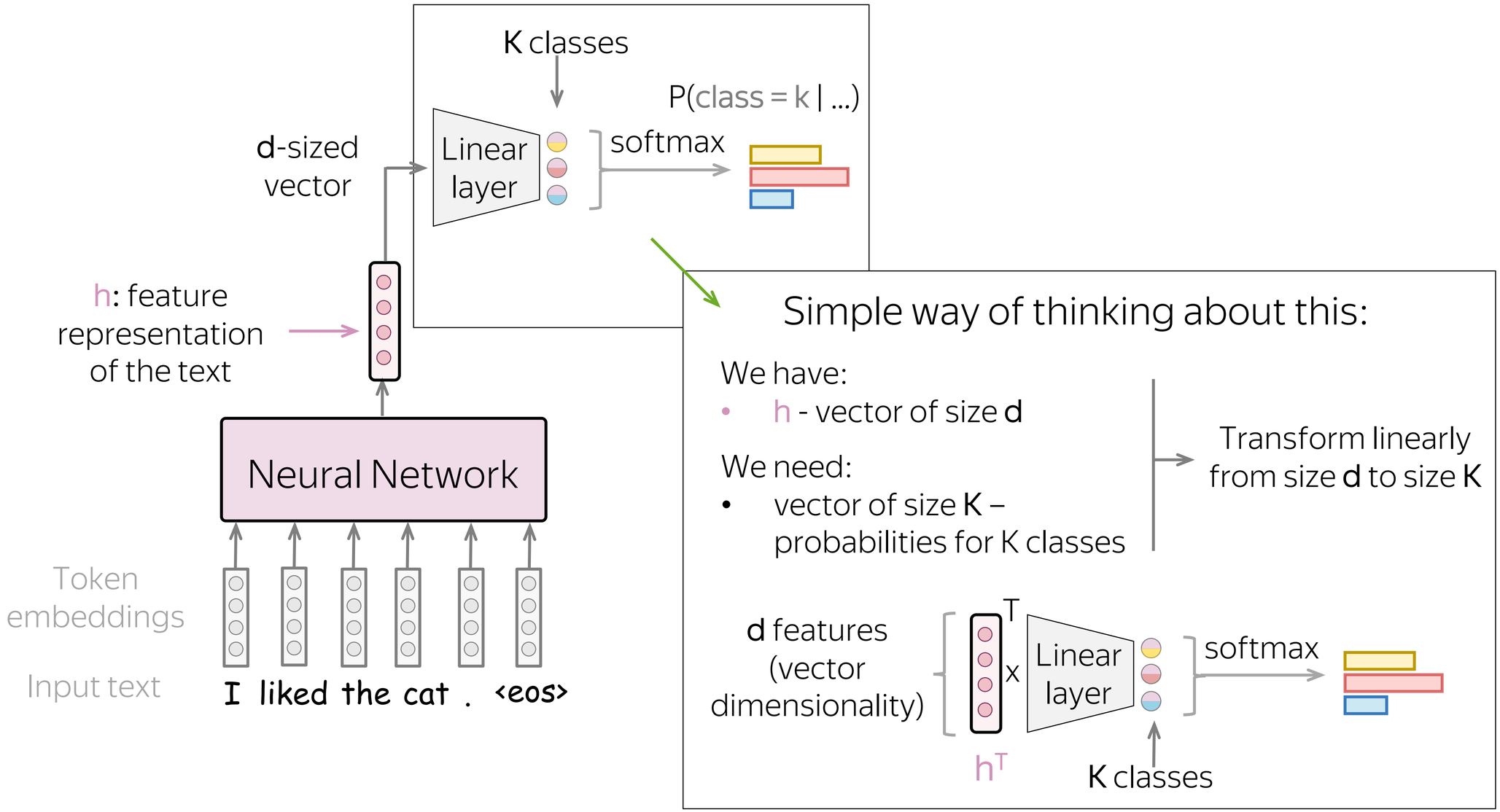
Classification with Neural Networks



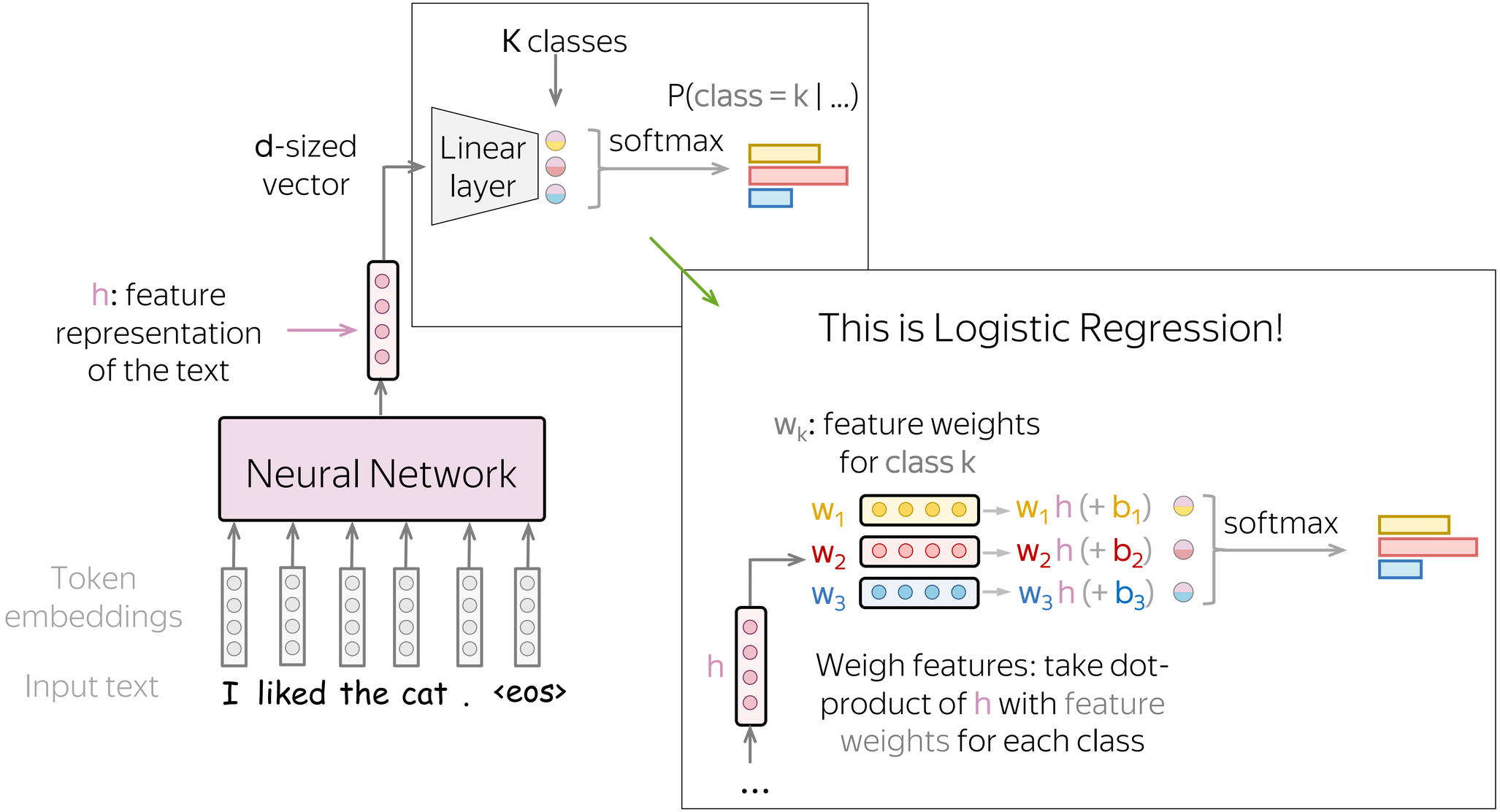
Classification with Neural Networks



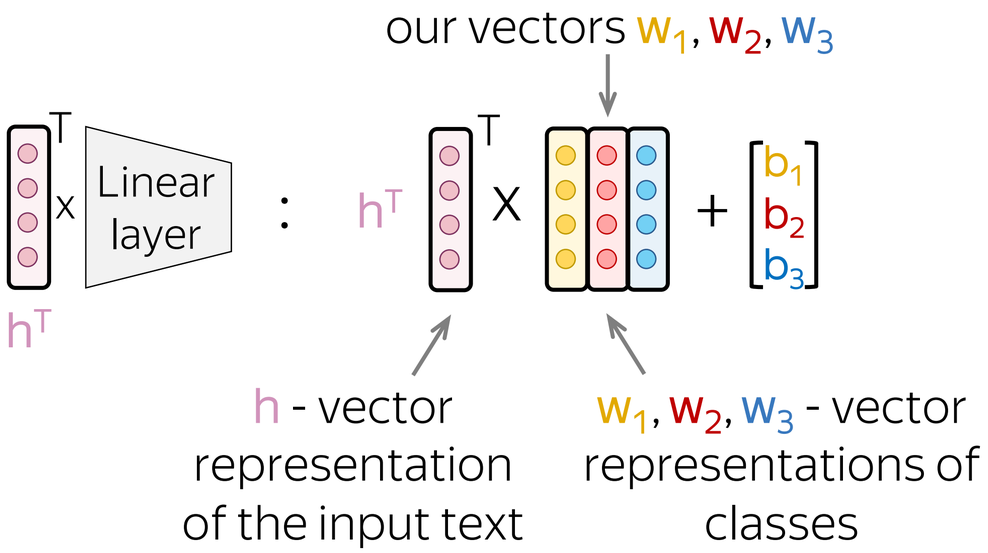
Classification with Neural Networks



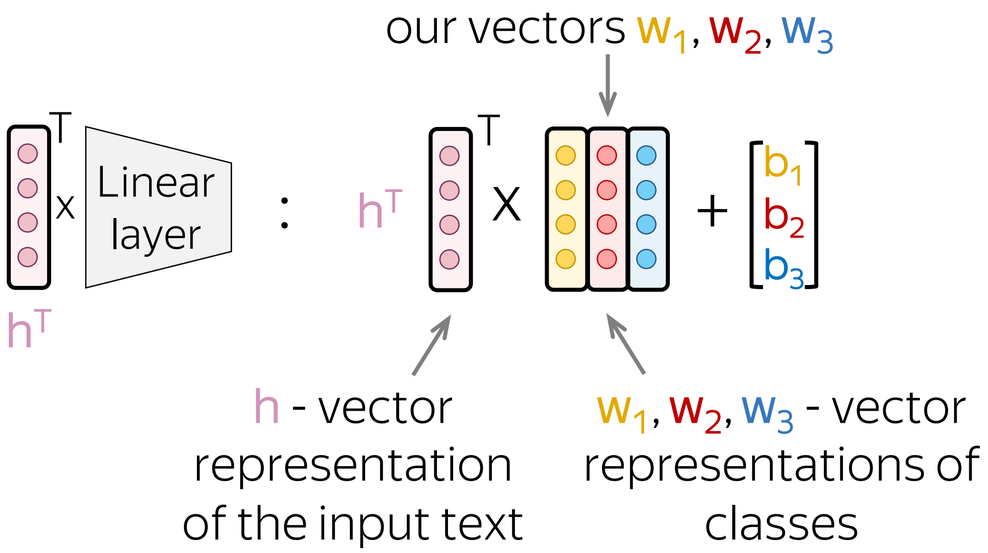
Classification with Neural Networks



Text Representation and Class Representation

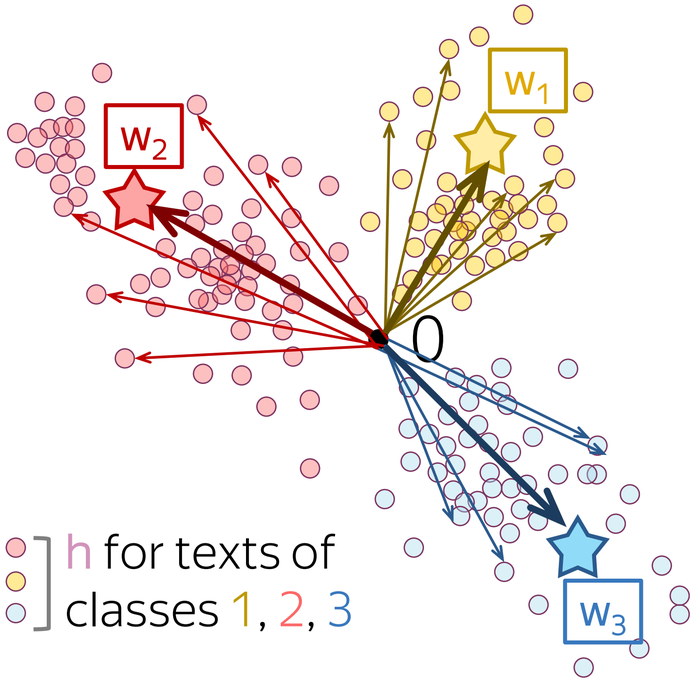


Text Representation and Class Representation



What NN learns (hopefully):

Text vectors point in the direction of the corresponding class vectors



What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
  - High-Level View
  - Training: Cross-Entropy
  - Models: (Weighted) BOW
  - Models: Convolutional
  - Models: Recurrent
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
  - High-Level View
  - Training: Cross-Entropy
  - Models: (Weighted) BOW
  - Models: Convolutional
  - Models: Recurrent
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

Training: Cross-Entropy

Training example: I liked the cat on the mat <eos>
Label: k

Training: Cross-Entropy

Training example: I liked the cat on the mat <eos>

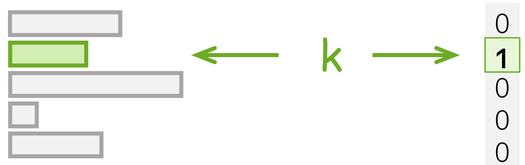
Label: k

Model prediction:

P(class = i |I liked...<eos>)

Target:

p^*



k

Training: Cross-Entropy

Training example: I liked the cat on the mat <eos>

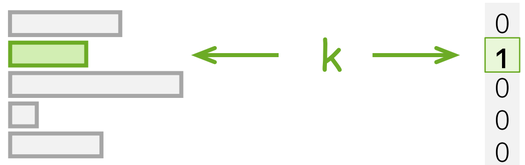
Label: k

Model prediction:

P(class = i |I liked…<eos>)

Target:

p^*



Cross-entropy loss:

$$-\sum_{i=1}^{K} p_i^* \cdot \log P(y = i|x) \to min \quad (p_k^* = 1, p_i^* = 0, i \neq k)$$

For one-hot targets, this is equivalent to

$$-\log P(y = k|x) \to min$$

Training: Cross-Entropy

Training example: I liked the cat on the mat <eos>

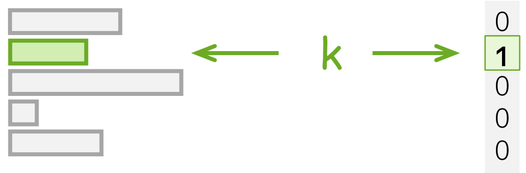
Label: k

Model prediction:

P(class = i |I liked...<eos>)

Target:

p^*

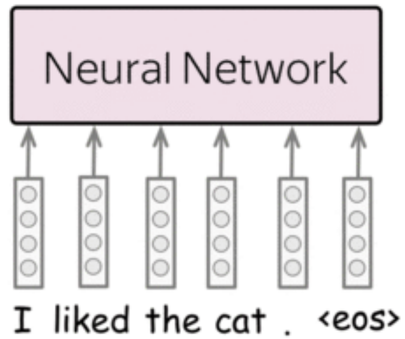


Cross-entropy loss:

$$-\sum_{i=1}^{K} p_i^* \cdot \log P(y=i|x) \to min \quad (p_k^* = 1, p_i^* = 0, i \neq k)$$

For one-hot targets, this is equivalent to

$$-\log P(y=k|x) \to min$$



Feed a text to the network

Correct label: 4 ← we want the model to predict this

What is going to happen:

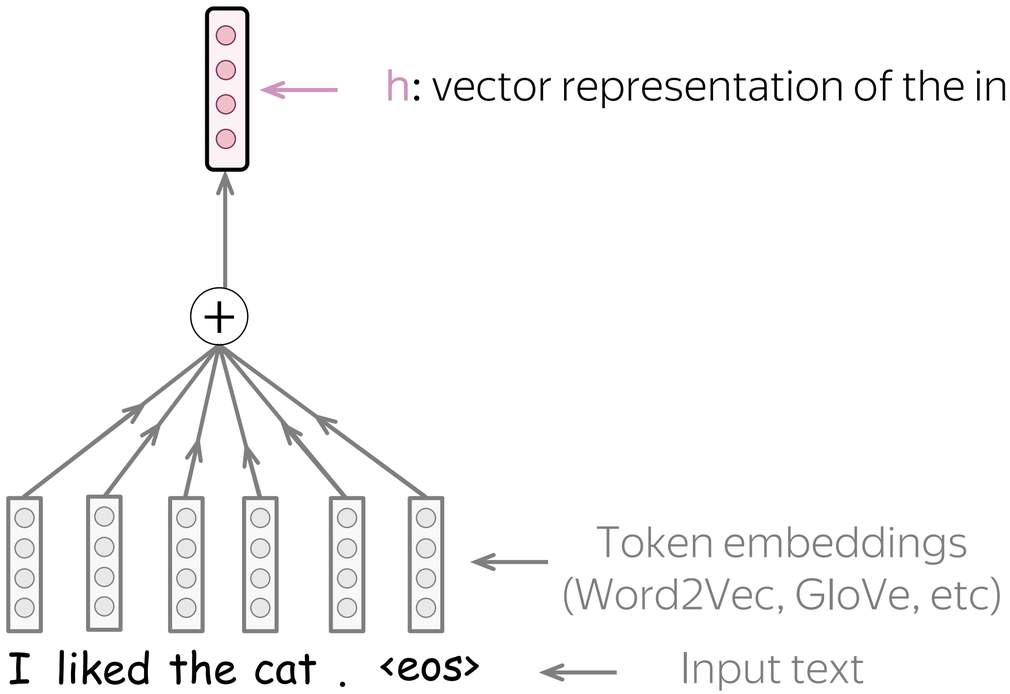
- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
  - High-Level View
  - Training: Cross-Entropy
  - Models: (Weighted) BOW
  - Models: Convolutional
  - Models: Recurrent
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

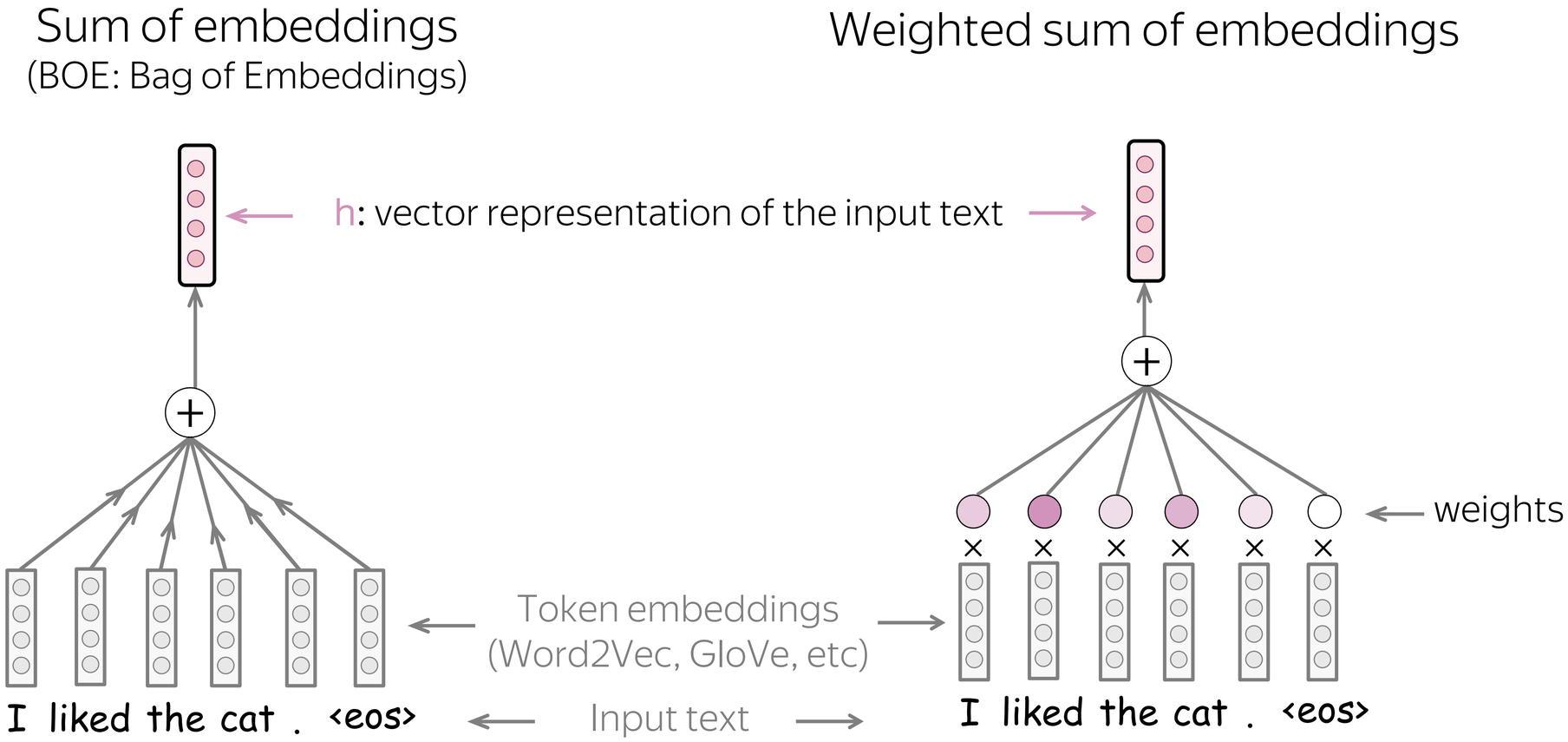
- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
  - High-Level View
  - Training: Cross-Entropy
  - Models: (Weighted) BOW
  - Models: Convolutional
  - Models: Recurrent
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

The Simplest Models: BOE and Weighted BOE

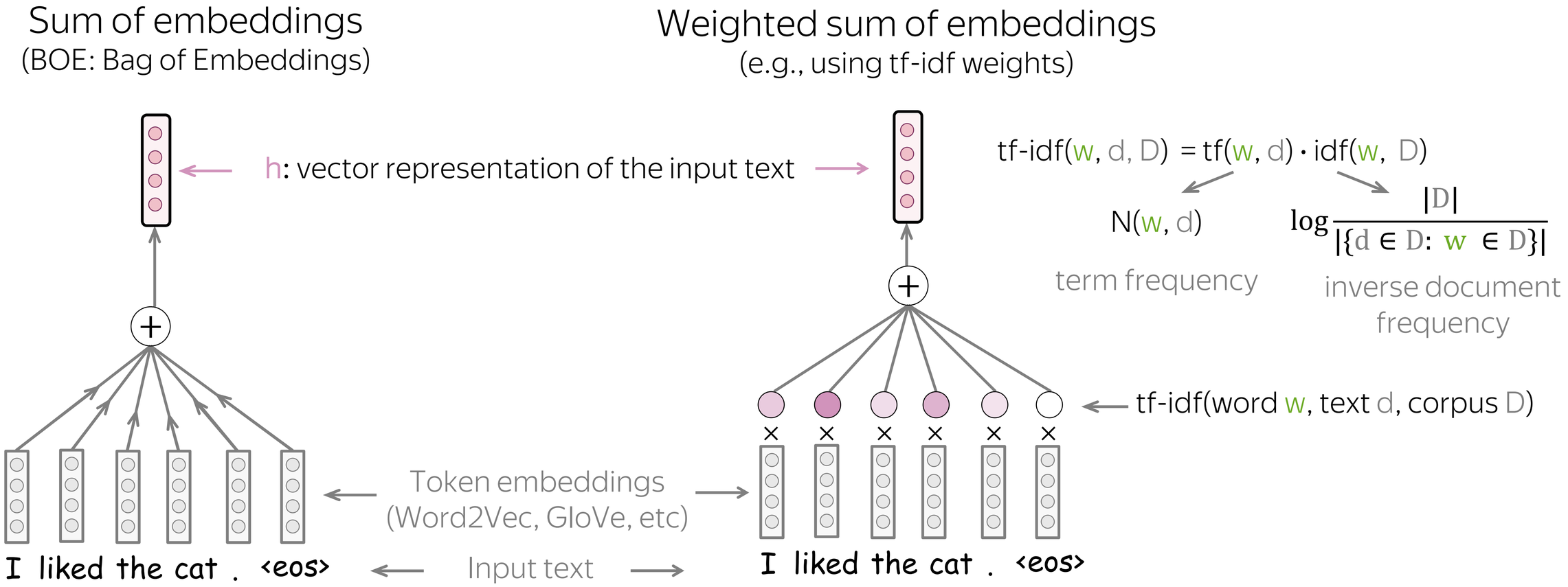
Sum of embeddings
(BOE: Bag of Embeddings)



The Simplest Models: BOE and Weighted BOE



The Simplest Models: BOE and Weighted BOE



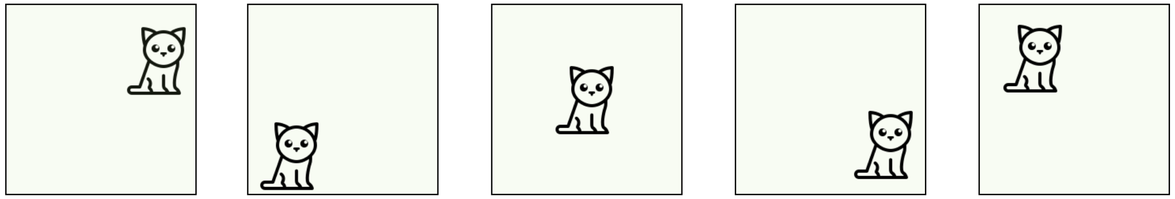
What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
  - High-Level View
  - Training: Cross-Entropy
  - Models: (Weighted) BOW
  - Models: Convolutional
  - Models: Recurrent
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

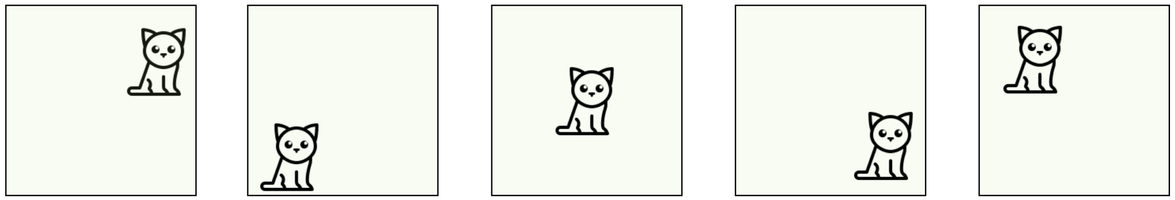
- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
  - High-Level View
  - Training: Cross-Entropy
  - Models: (Weighted) BOW
  - Models: Convolutional
  - Models: Recurrent
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

Convolutions for Images and Translation Invariance



Label: cat Label: cat Label: cat Label: cat Label: cat

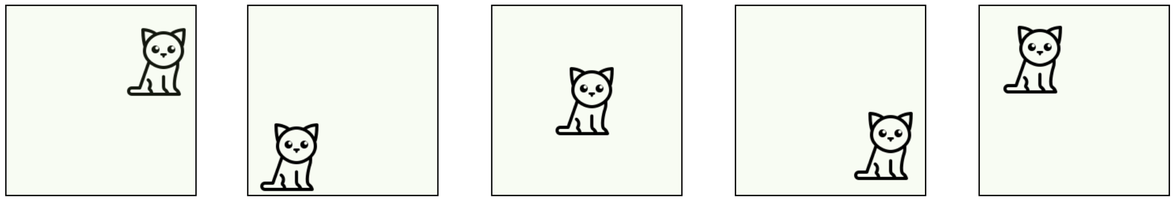
Convolutions for Images and Translation Invariance



Label: **cat** Label: **cat** Label: **cat** Label: **cat** Label: **cat**

We don’t care where the cat is, we
care that it is somewhere.

Convolutions for Images and Translation Invariance

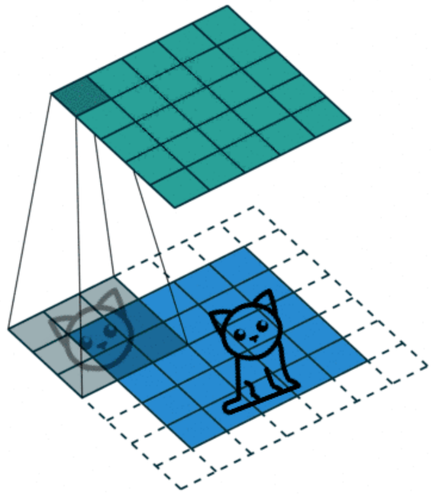


Label: cat Label: cat Label: cat Label: cat Label: cat

We don’t care where the cat is, we care that it is somewhere.

Then why don’t we process all these cats similarly?

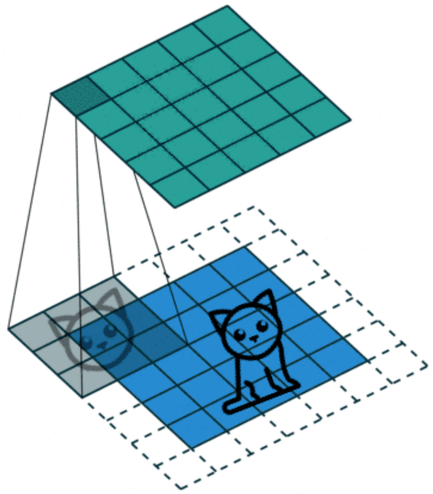
Convolutions for Images and Translation Invariance



The gif is adapted from the one taken from the repo https://github.com/vdumoulin/conv_arithmetic

- apply the same operation to small parts of an input
- find “matches” with patterns

Convolutions for Images and Translation Invariance

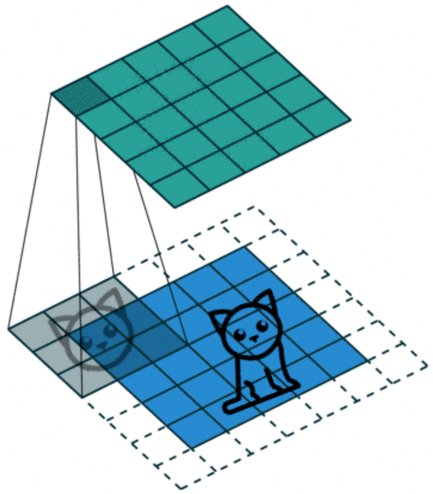


The gif is adapted from the one taken from the repo https://github.com/vdumoulin/conv_arithmetic

- apply the same operation to small parts of an input
- find “matches” with patterns

this is how CNNs extract features

Convolutions for Images and Translation Invariance



The gif is adapted from the one taken from the repo https://github.com/vdumoulin/conv_arithmetic

- apply the same operation to small parts of an input
- find “matches” with patterns

this is how CNNs extract features

- a network learns which patterns are useful
- from bottom to top of a network, patterns evolve from simple to complicated

We’ll see this in the analysis section

What About Texts?

An absolutely great movie! I watched the premiere with my friends.

The movie about cats was absolutely great, and the cats were cute.

The movie is about cats running around, and it is absolutely great.

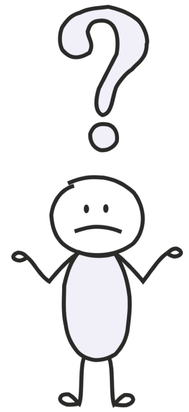
What About Texts?

An absolutely great movie! I watched the premiere with my friends.

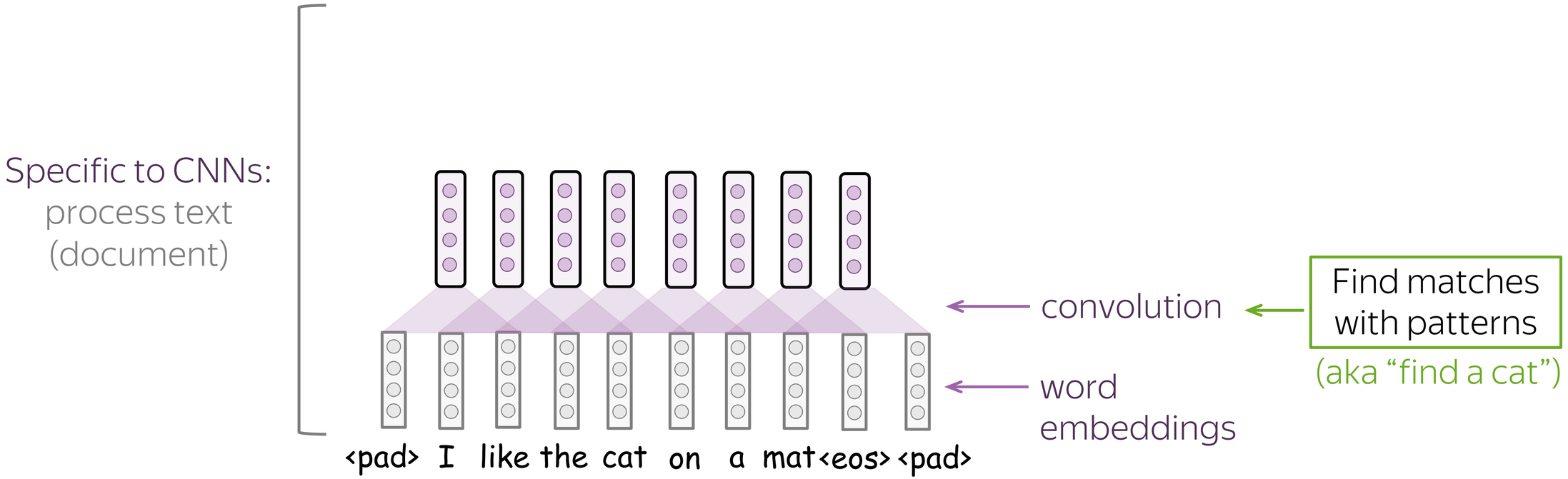
The movie about cats was absolutely great, and the cats were cute.

The movie is about cats running around, and it is absolutely great.

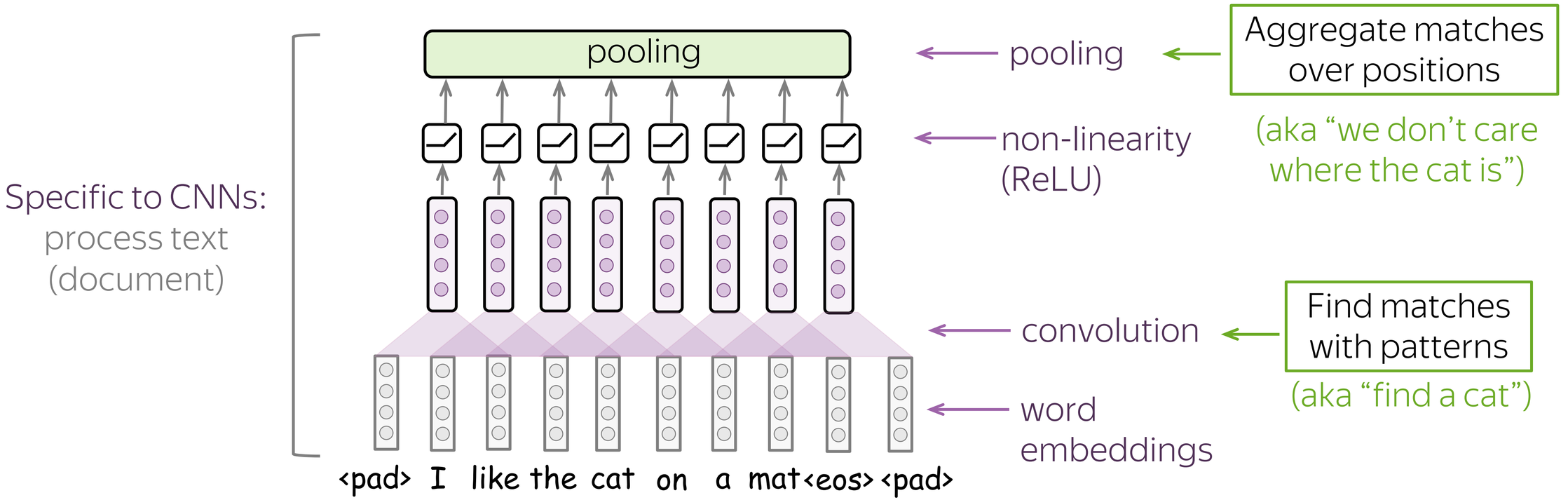
If a clue is very informative, maybe we don't care much where in a text it appears?



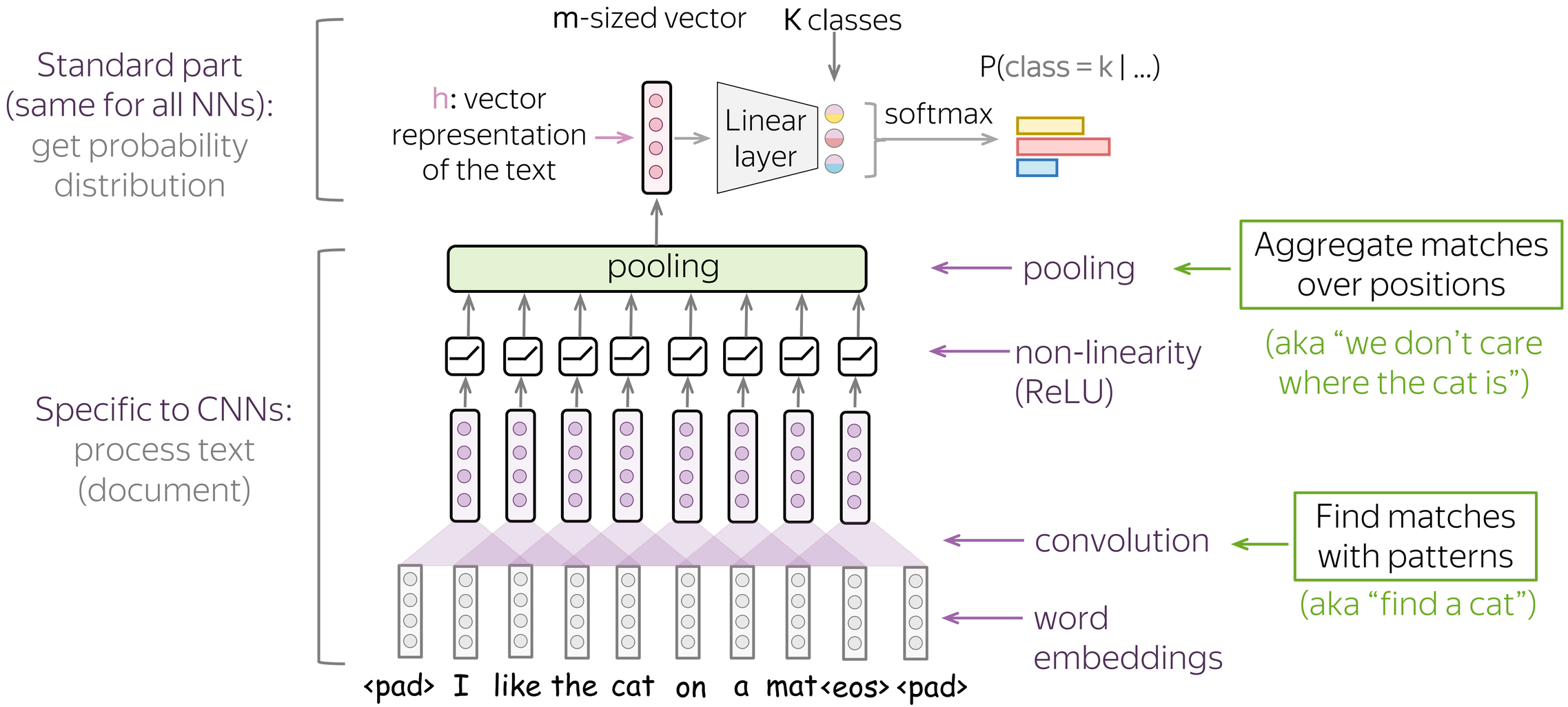
A Typical Model: Convolution + Pooling



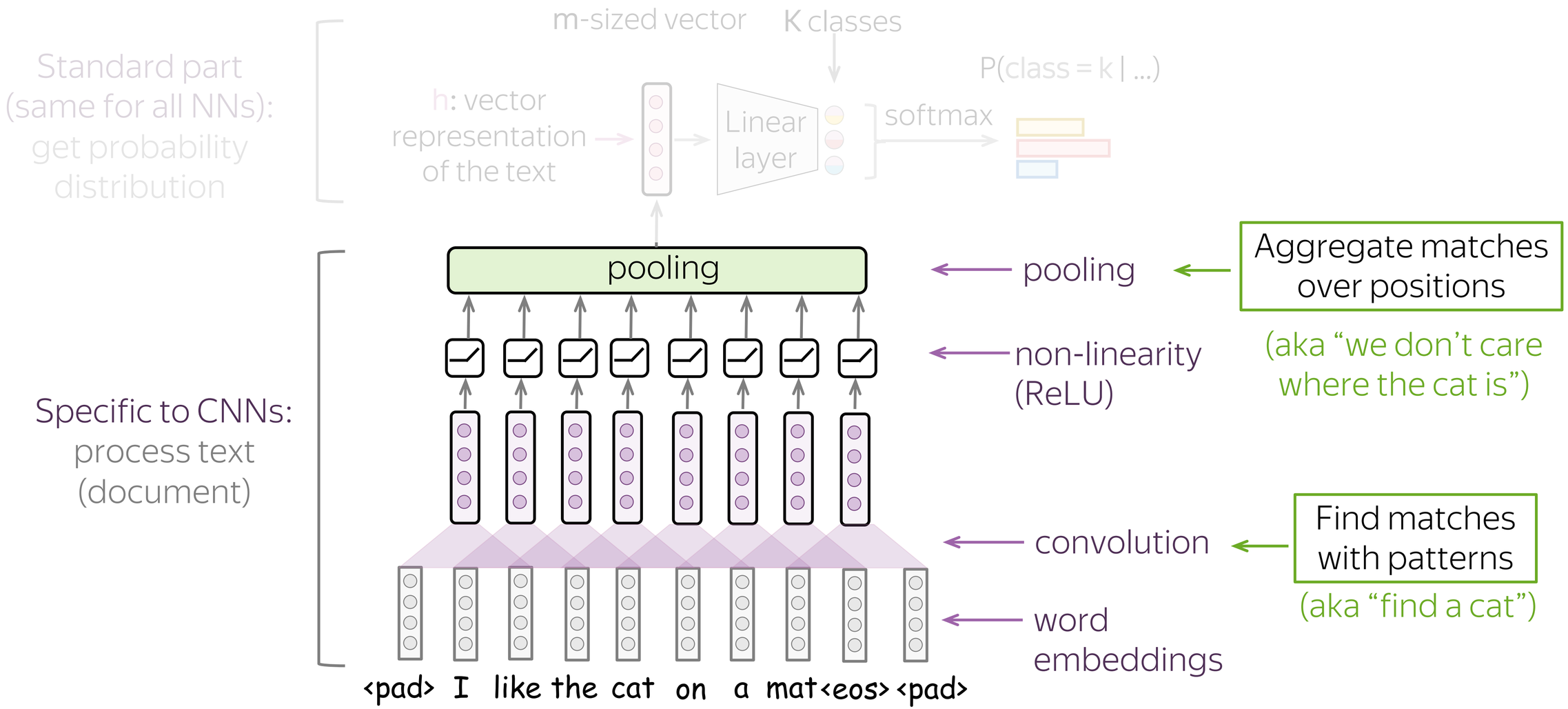
A Typical Model: Convolution + Pooling



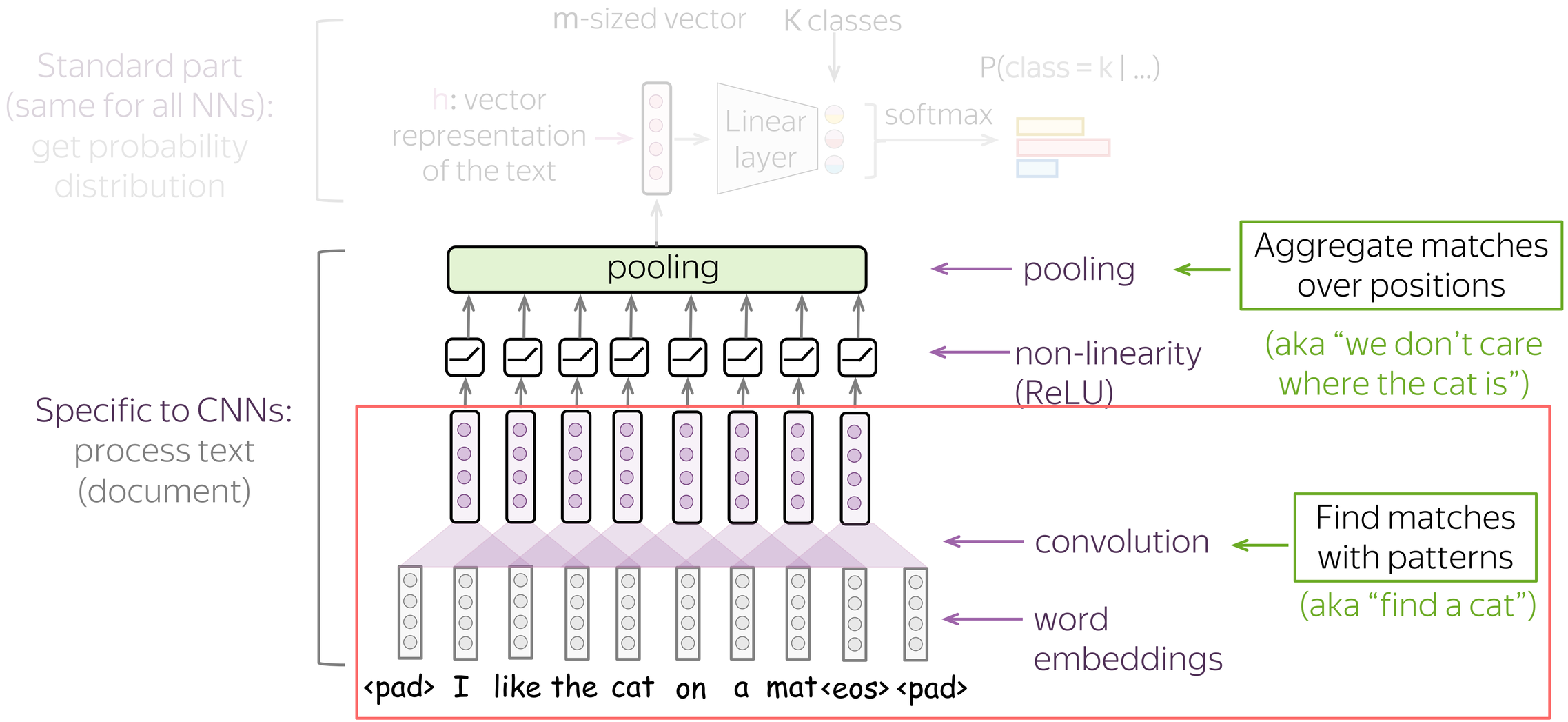
A Typical Model: Convolution + Pooling



A Typical Model: Convolution + Pooling

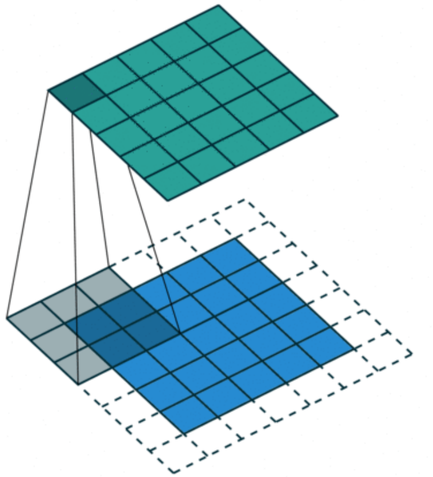


A Typical Model: Convolution + Pooling



Building Blocks: Convolution

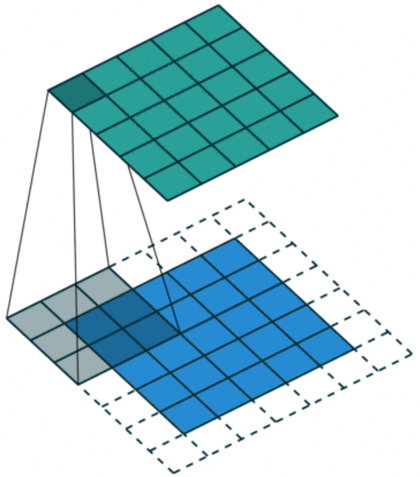
Convolution filter for an image



This gif is from the repo
https://github.com/vdumoulin/conv_arithmetic

Building Blocks: Convolution

Convolution filter for an image

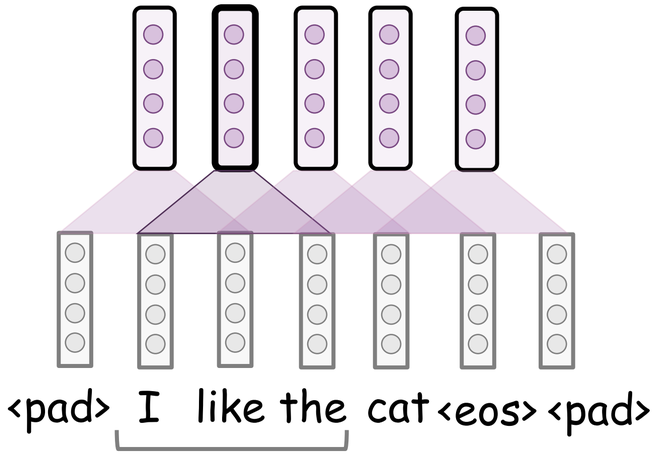


This gif is from the repo
https://github.com/vdumoulin/conv_arithmetic

Convolution filter for a text

Convolution is a Linear Operation Applied to Each Window

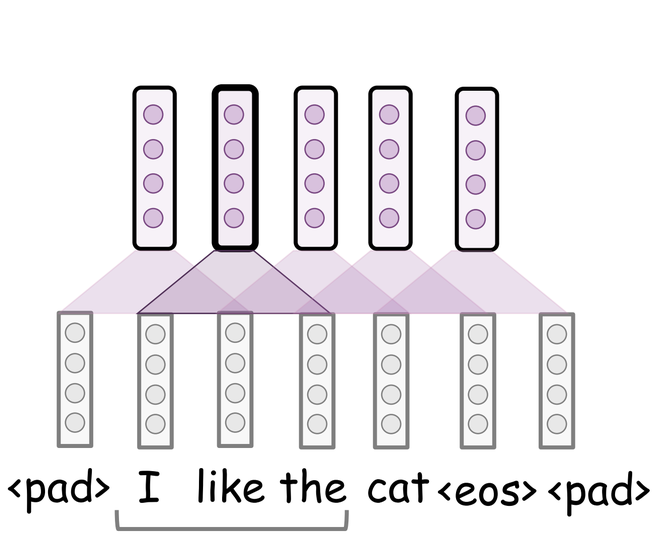
(except for a non-linearity)

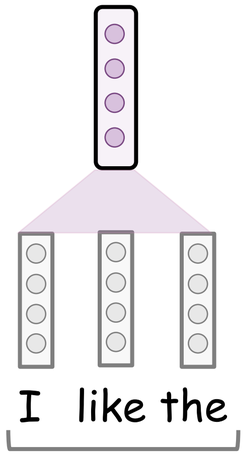


- $x_1, x_2, \ldots, x_n$ - input vectors (e.g., word embeddings)

Convolution is a Linear Operation Applied to Each Window

(except for a non-linearity)

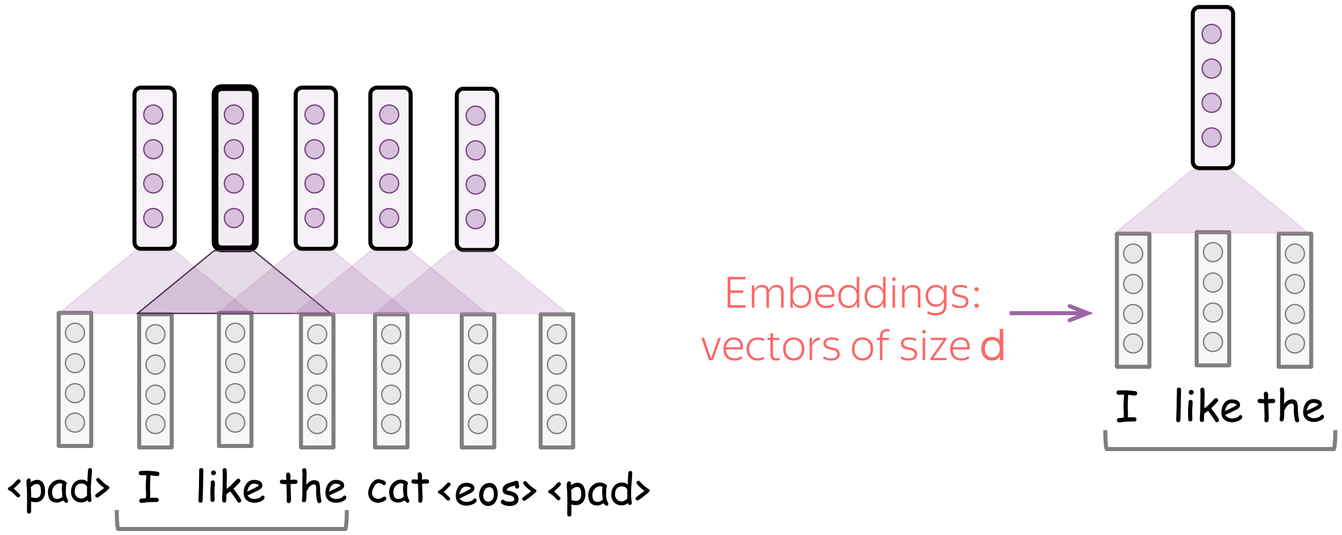




- $x_1, x_2, \ldots, x_n$ - input vectors (e.g., word embeddings)

Convolution is a Linear Operation Applied to Each Window

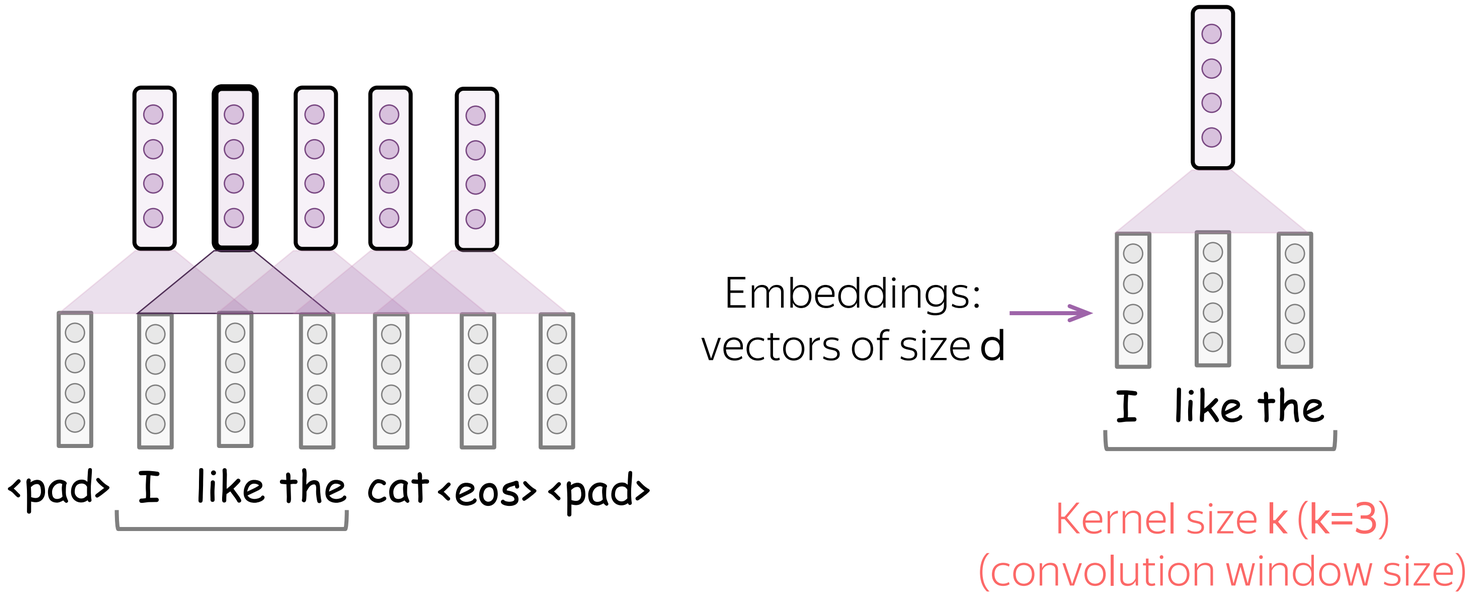
(except for a non-linearity)



- $x_1, x_2, \ldots, x_n$ - input vectors (e.g., word embeddings)
- d (input channels) – input vector size

Convolution is a Linear Operation Applied to Each Window

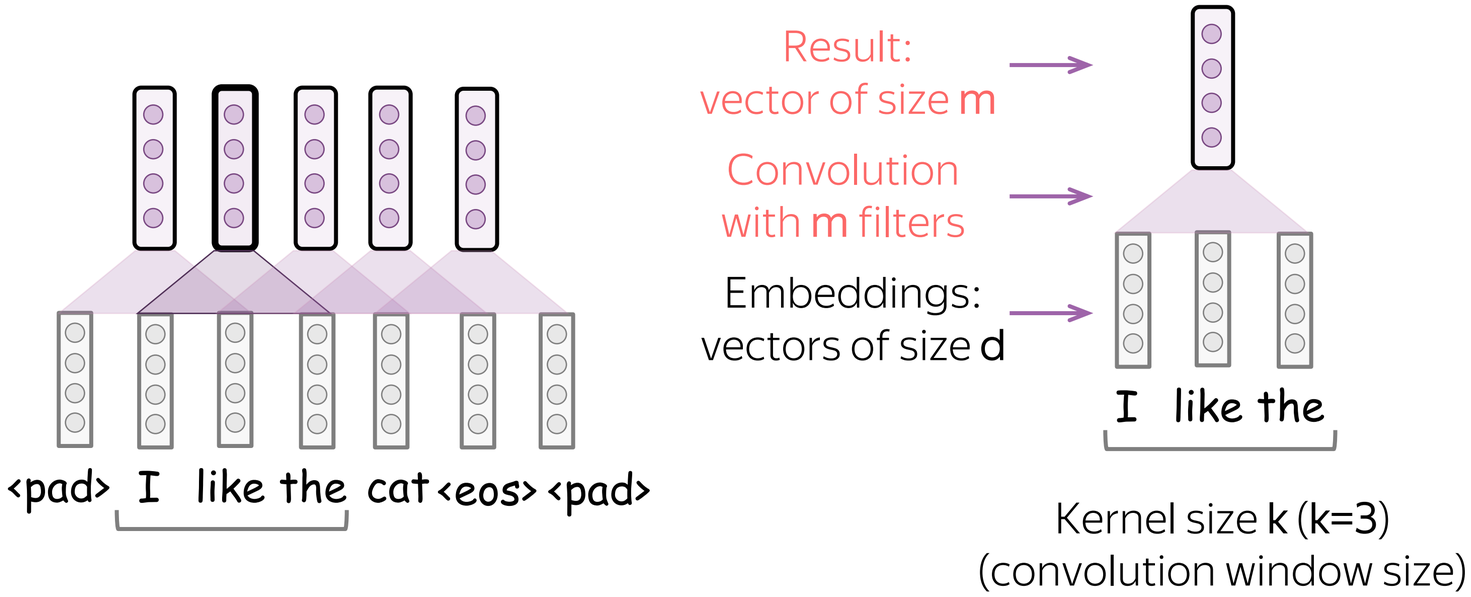
(except for a non-linearity)



- $x_1, x_2, \ldots, x_n$ - input vectors (e.g., word embeddings)
- d (input channels) – input vector size
- k (kernel size) – conv. window length

Convolution is a Linear Operation Applied to Each Window

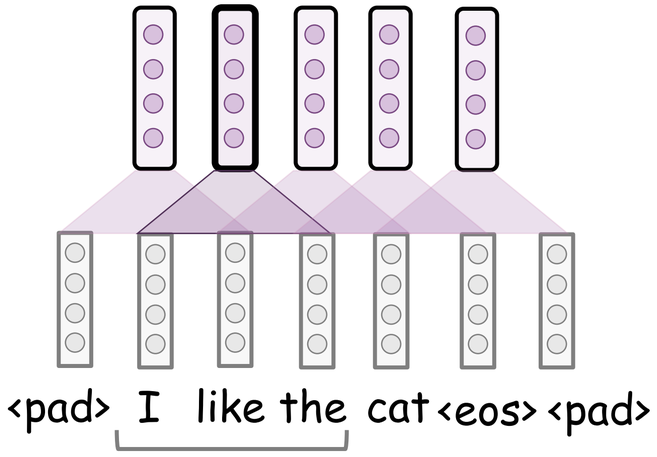
(except for a non-linearity)

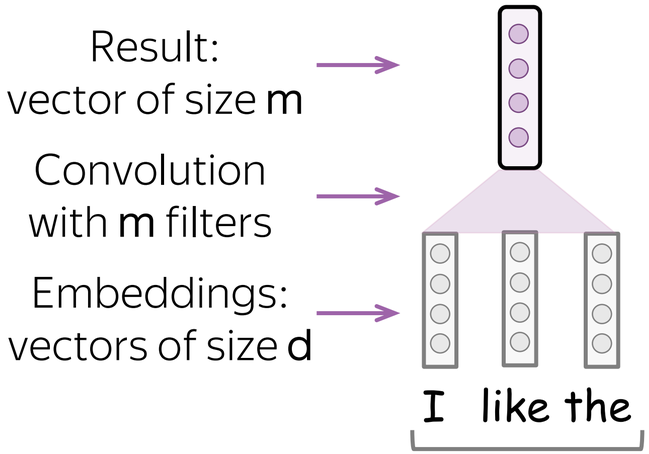


- $x_1, x_2, \ldots, x_n$ - input vectors (e.g., word embeddings)
- d (input channels) – input vector size
- k (kernel size) – conv. window length
- m (output channels) – number of filters

Convolution is a Linear Operation Applied to Each Window

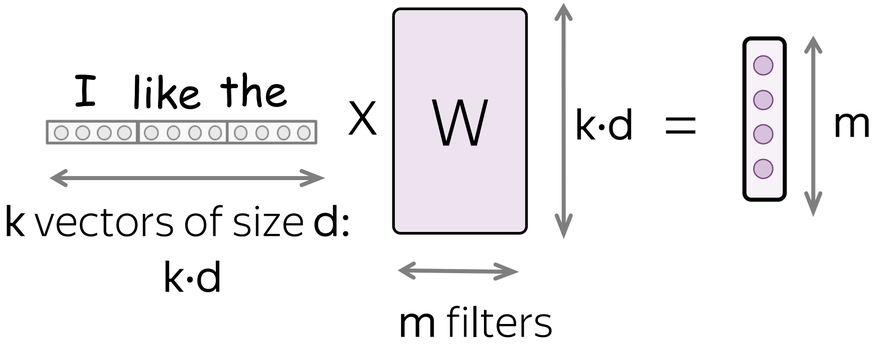
(except for a non-linearity)





Kernel size k (k=3)
(convolution window size)

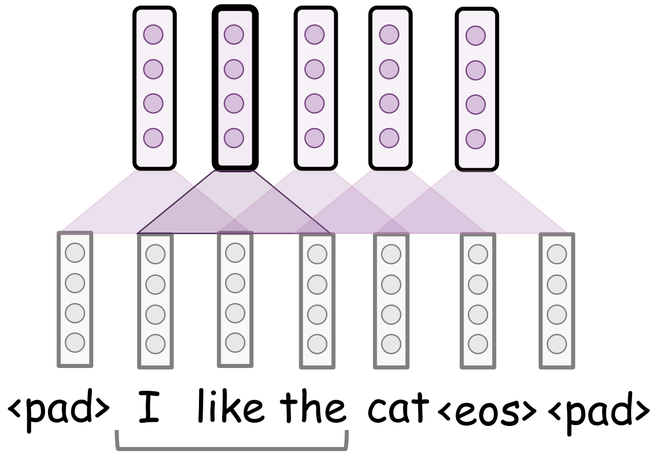
Convolution is a linear layer
mapping from k·d to m

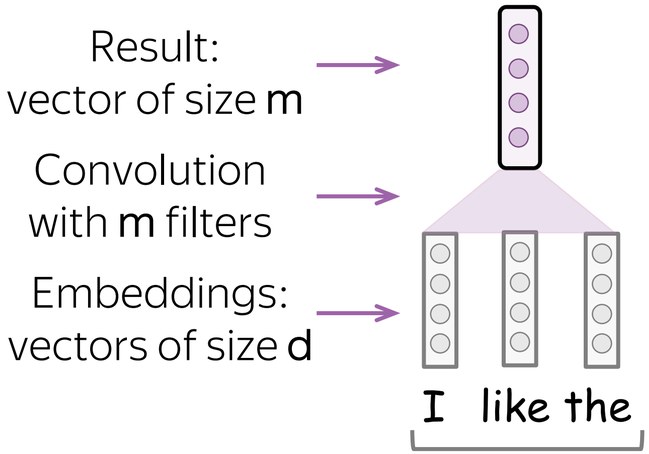


- $x_1, x_2, \ldots, x_n$ - input vectors (e.g., word embeddings)
- d (input channels) – input vector size
- k (kernel size) – conv. window length
- m (output channels) – number of filters

Convolution is a Linear Operation Applied to Each Window

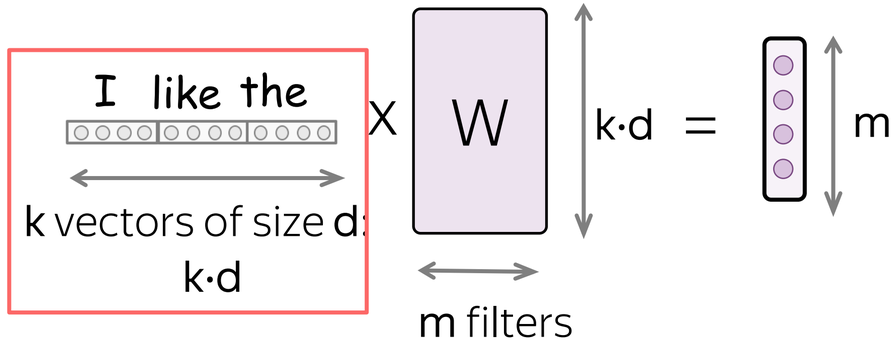
(except for a non-linearity)





Kernel size k (k=3)
(convolution window size)

Convolution is a linear layer
mapping from k·d to m

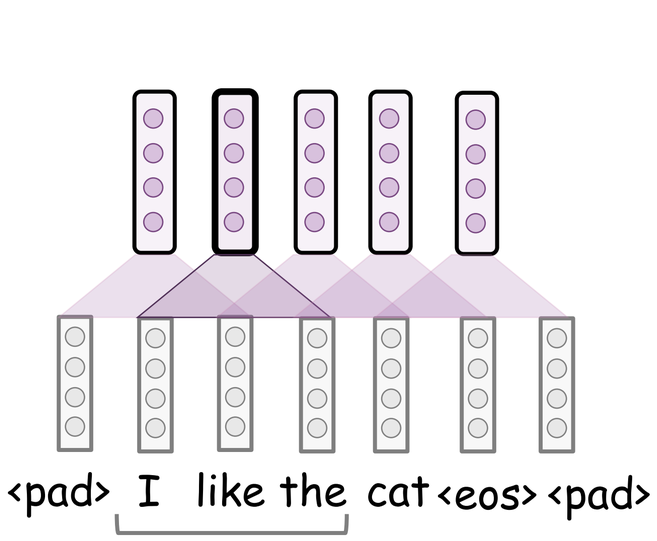


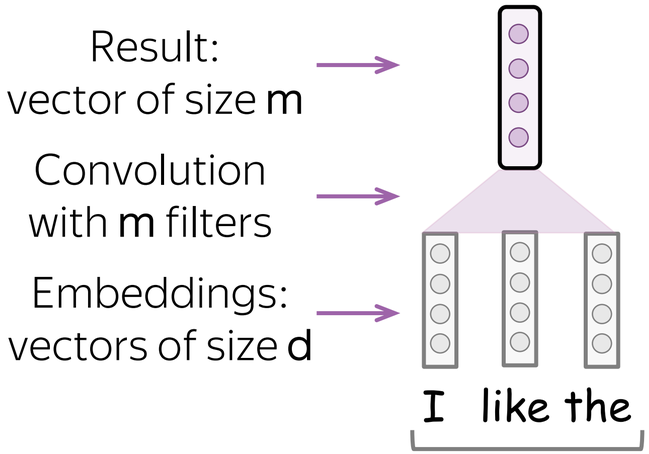
- $x_1, x_2, \dots, x_n$ - input vectors (e.g., word embeddings)
- d (input channels) – input vector size
- k (kernel size) – conv. window length
- m (output channels) – number of filters

$u_i = [x_i, \dots, x_{i+k-1}] \in \mathbb{R}^{k \cdot d}$ - concatenate representations in the i-th window

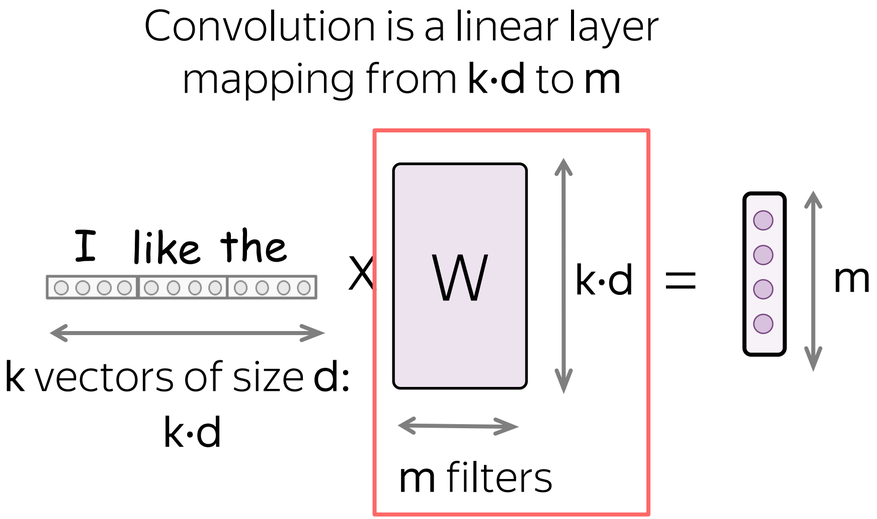
Convolution is a Linear Operation Applied to Each Window

(except for a non-linearity)





Kernel size k (k=3)
(convolution window size)



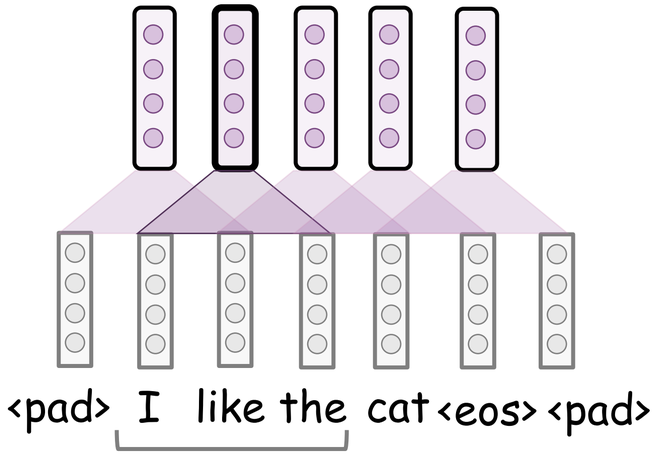
- $x_1, x_2, \ldots, x_n$ - input vectors (e.g., word embeddings)
- d (input channels) – input vector size
- k (kernel size) – conv. window length
- m (output channels) – number of filters

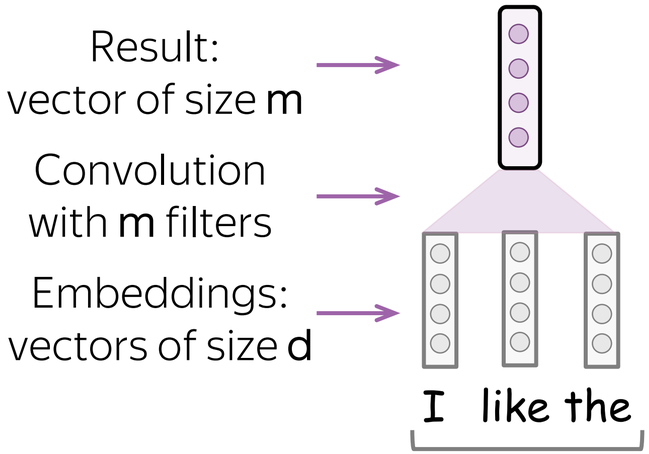
$u_i = [x_i, \ldots, x_{i+k-1}] \in \mathbb{R}^{k \cdot d}$ - concatenate representations in the i-th window

$W \in \mathbb{R}^{(k \cdot d) \times m}$ - convolution

Convolution is a Linear Operation Applied to Each Window

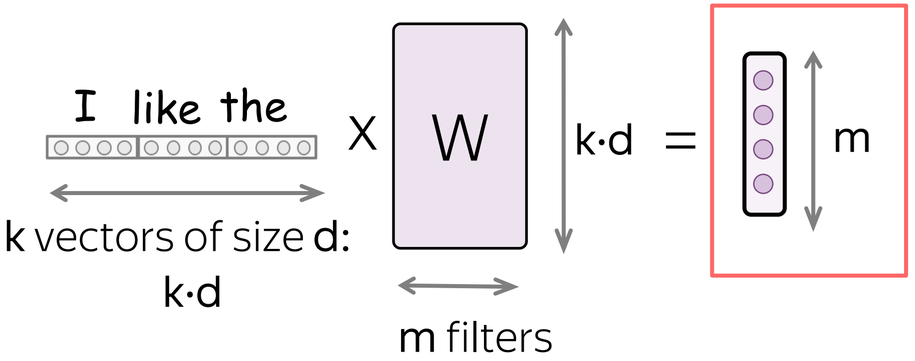
(except for a non-linearity)





Kernel size k (k=3)
(convolution window size)

Convolution is a linear layer
mapping from k·d to m



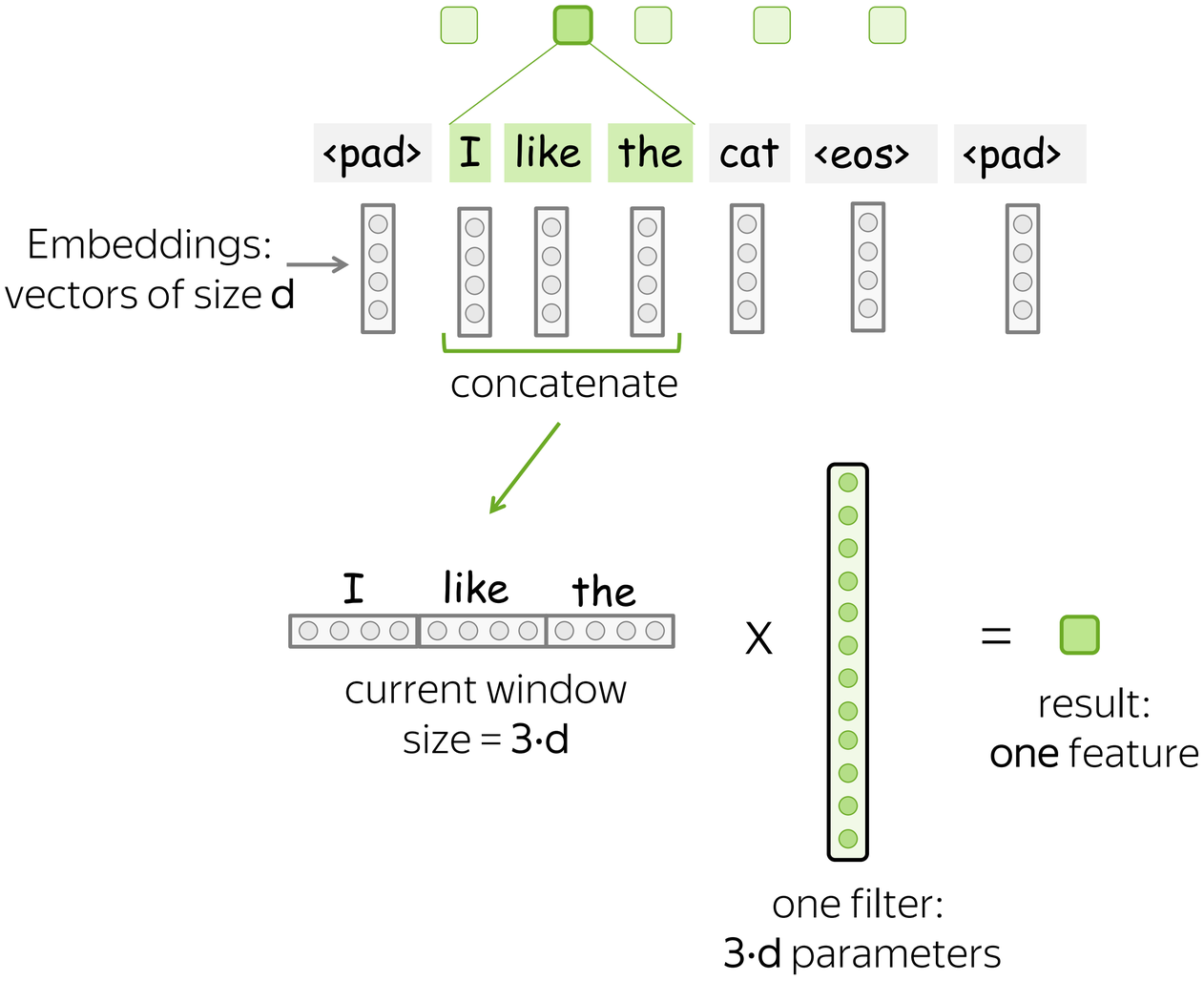
- $x_1, x_2, \ldots, x_n$ - input vectors (e.g., word embeddings)
- d (input channels) – input vector size
- k (kernel size) – conv. window length
- m (output channels) – number of filters

$u_i = [x_i, \ldots, x_{i+k-1}] \in \mathbb{R}^{k \cdot d}$ - concatenate representations in the i-th window

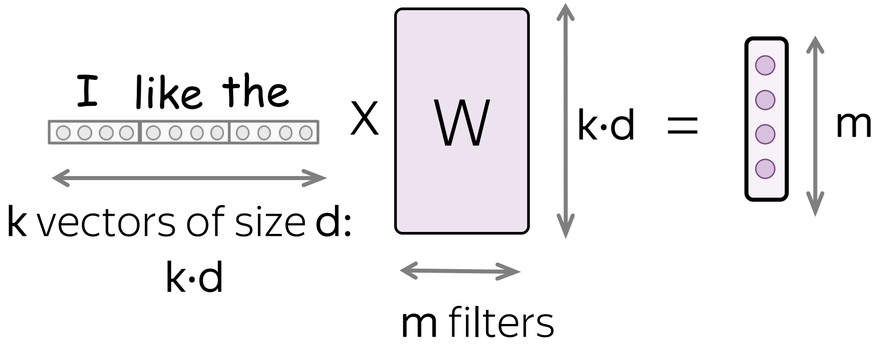
$W \in \mathbb{R}^{(k \cdot d) \times m}$ - convolution

$F_i = u_i \times W$ – convolution applied to the i-th window

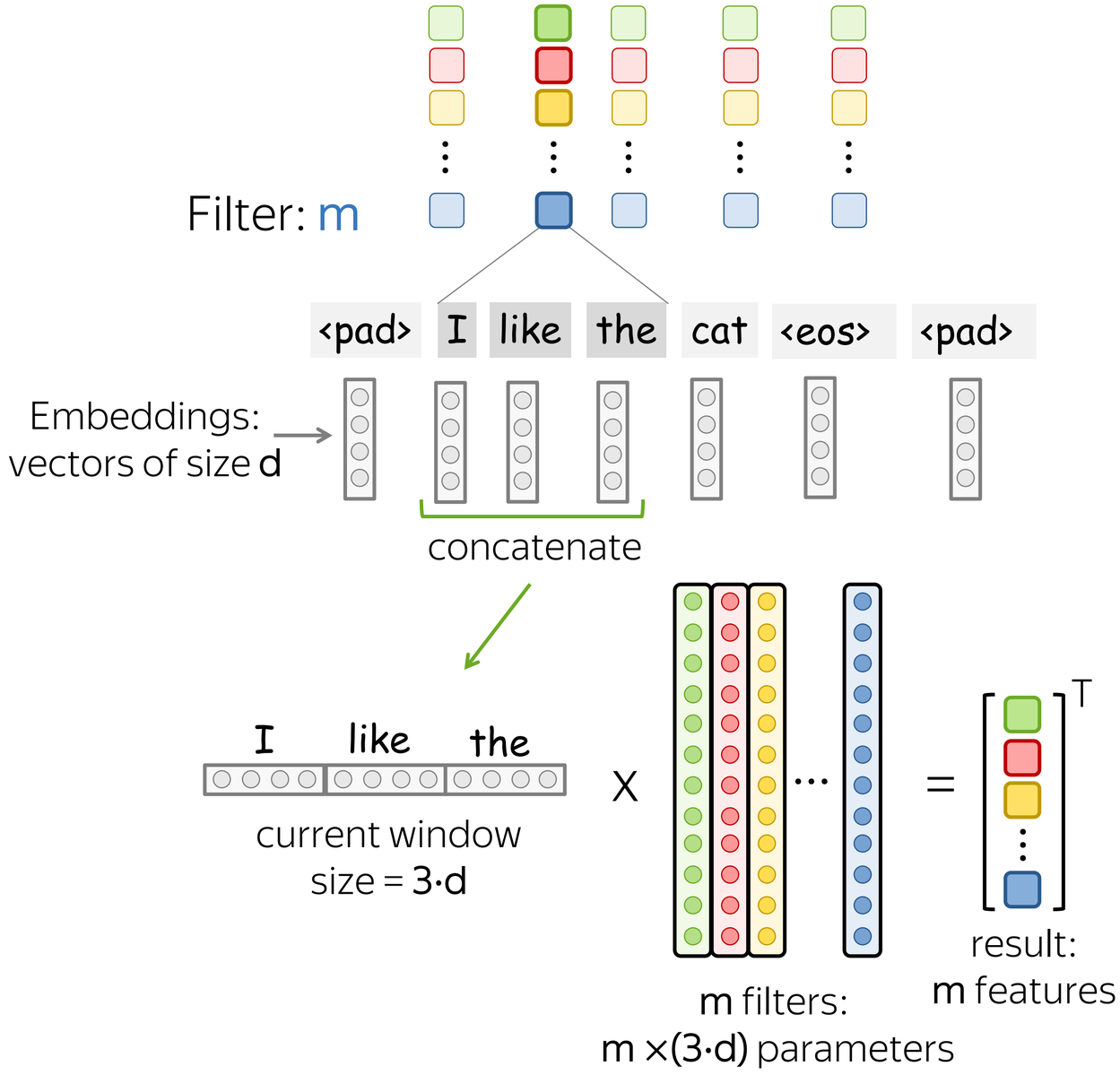
Intuition: A Filter is a Feature Extractor



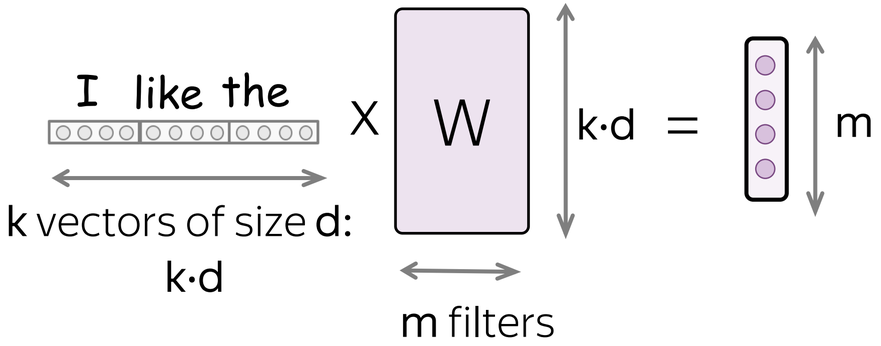
Convolution is a linear layer
mapping from **k·d** to **m**



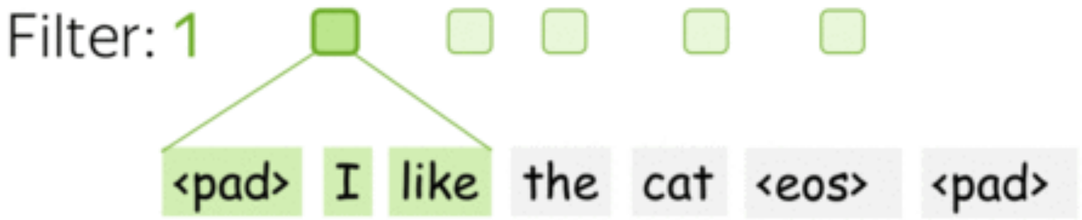
Several Filters – Several Features



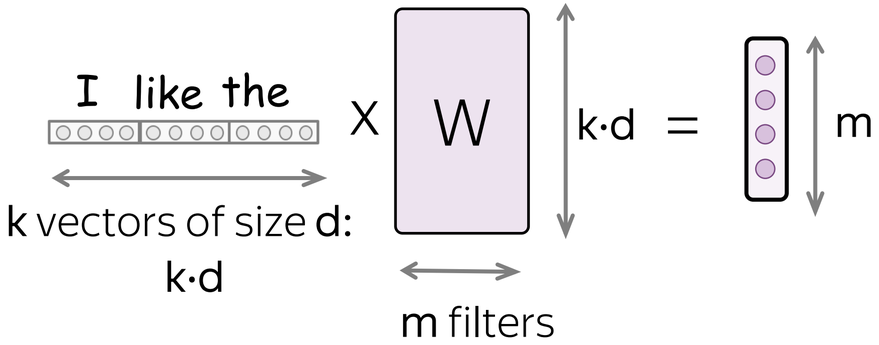
Convolution is a linear layer
mapping from **k·d** to **m**



Several Filters – Several Features



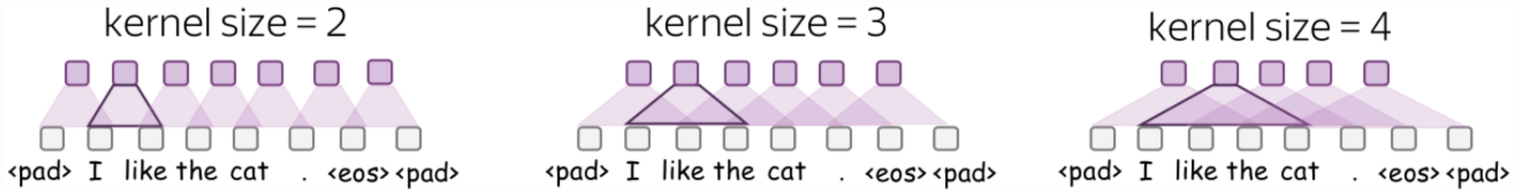
Convolution is a linear layer
mapping from **k·d** to **m**



Convolution: Parameters

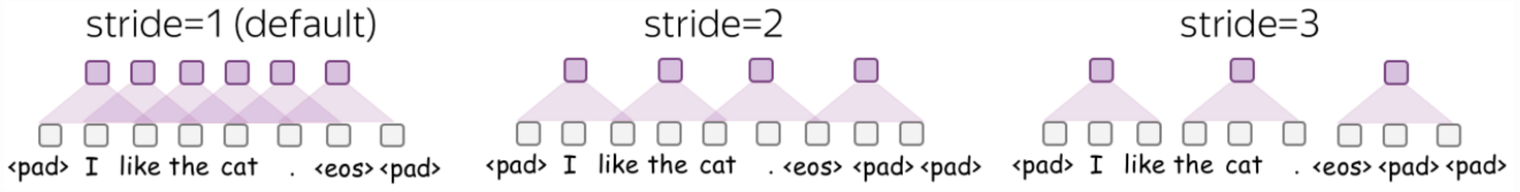
- **Kernel size**: How far to look

Kernel size is the number of input elements (tokens) a convolution looks at each step. For text, typical values are 2-5.

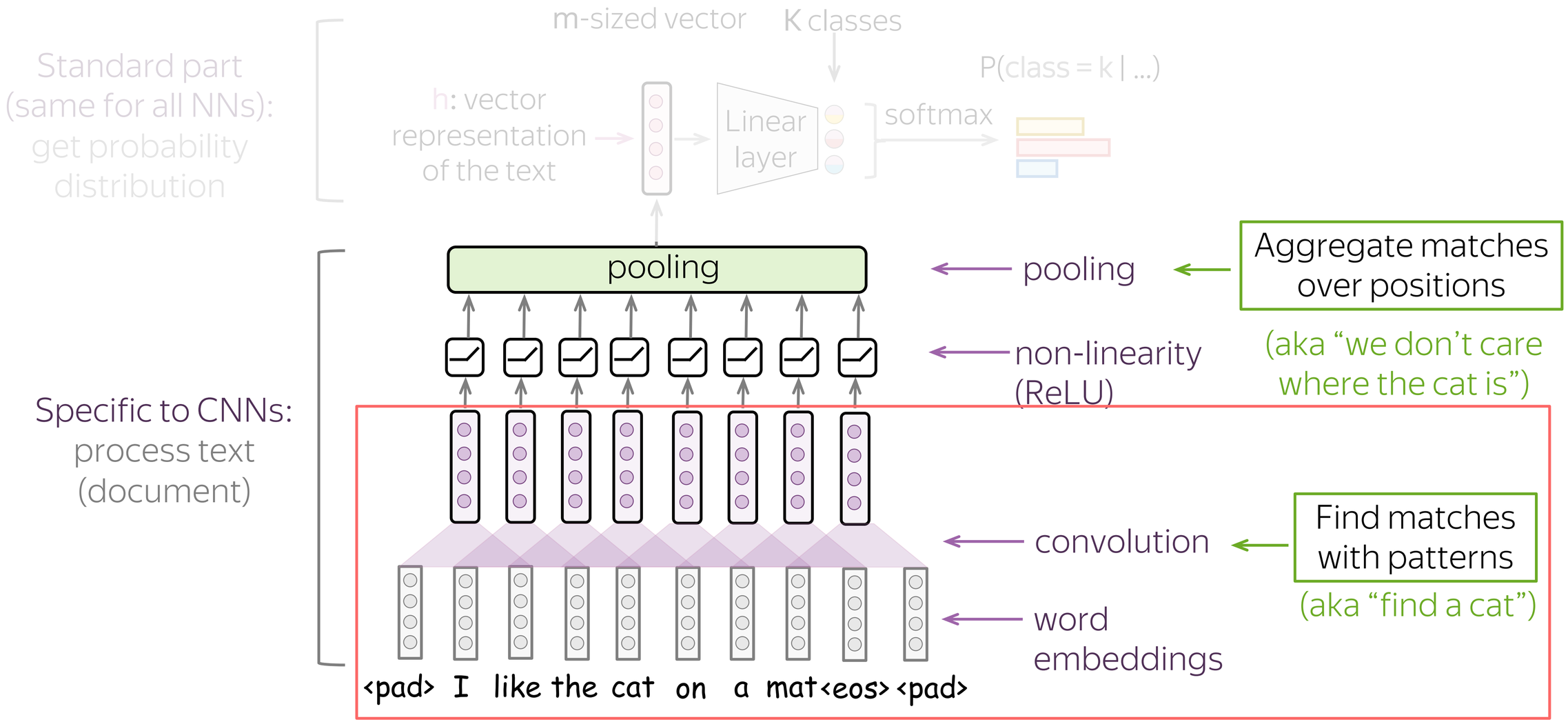


- **Stride**: How much move a filter at each step

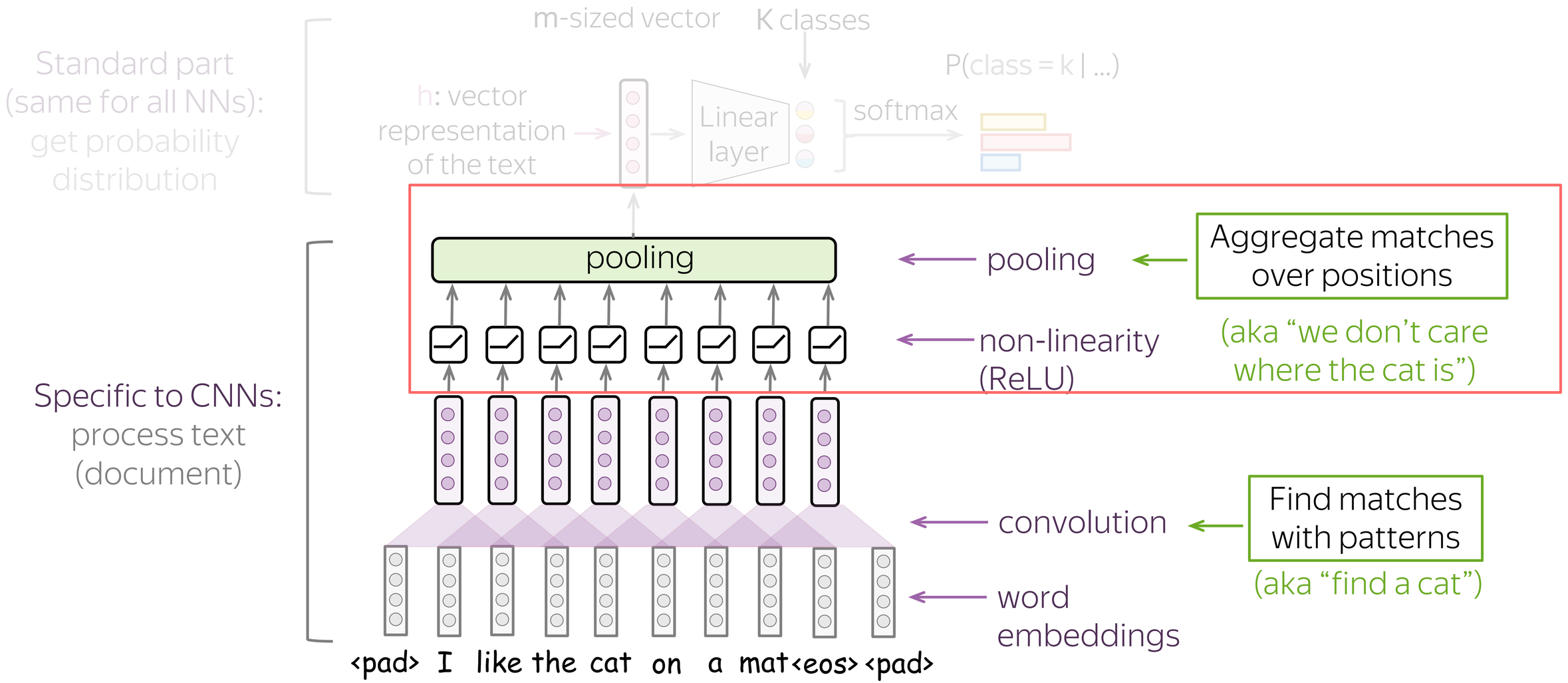
Stride tells how much to move filter at each step. For example, stride equal to 1 means that we move the filter by 1 input element (pixel for images, token for texts) at each step.



A Typical Model: Convolution + Pooling

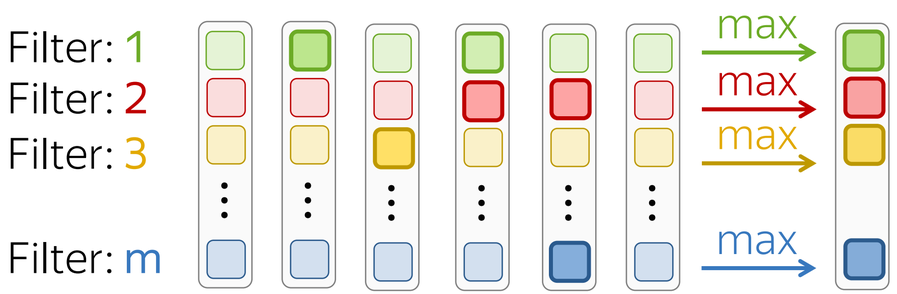


A Typical Model: Convolution + Pooling



Building Blocks: Pooling (max, mean, etc)

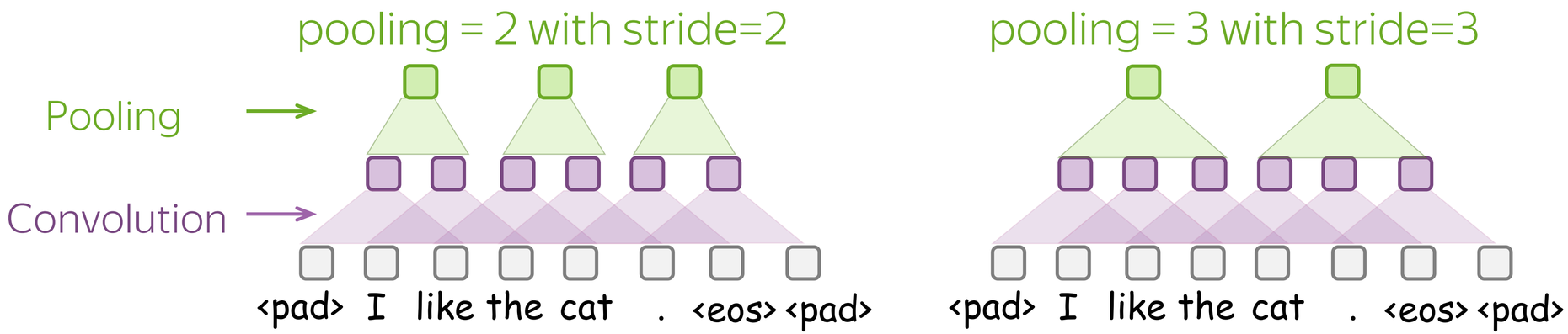
- **Max pooling**: maximum for each dimension (feature)
- **Mean pooling**: mean value for each dimension (feature)



Max pooling: maximum for each dimension (feature)

Building Blocks: Pooling and Global Pooling

- **Pooling**: aggregate features in some area

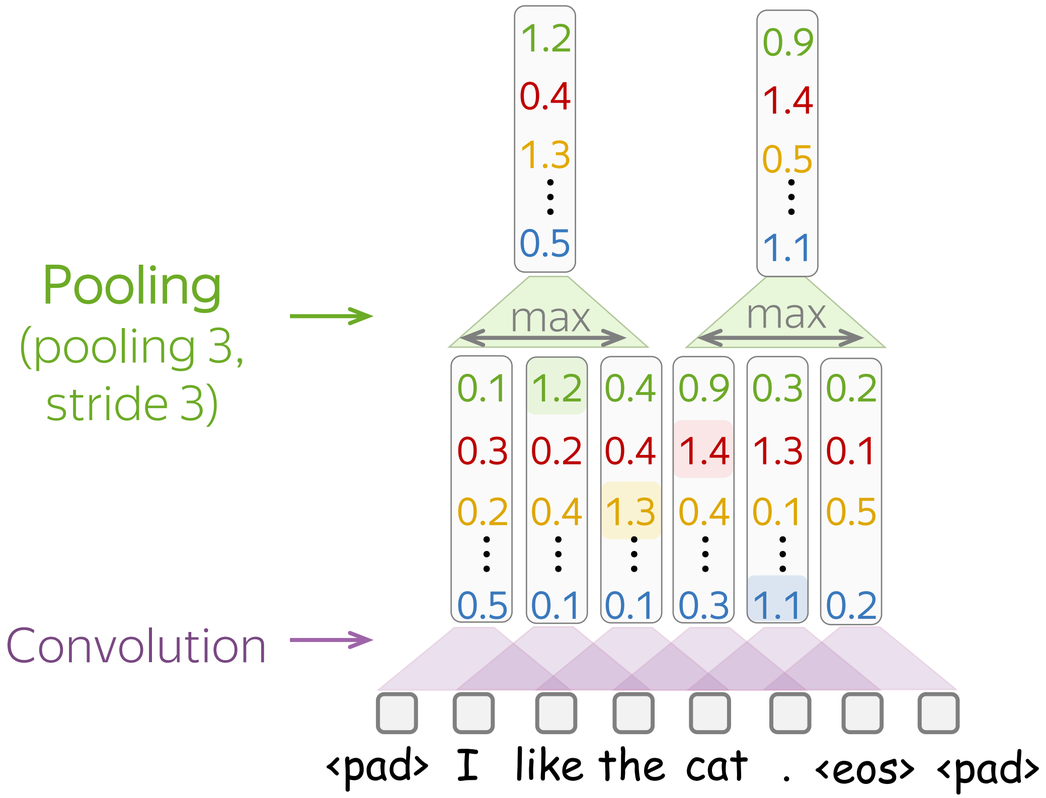


Building Blocks: Pooling and Global Pooling

- Pooling: aggregate features in some area
- Global pooling: aggregate features over all input

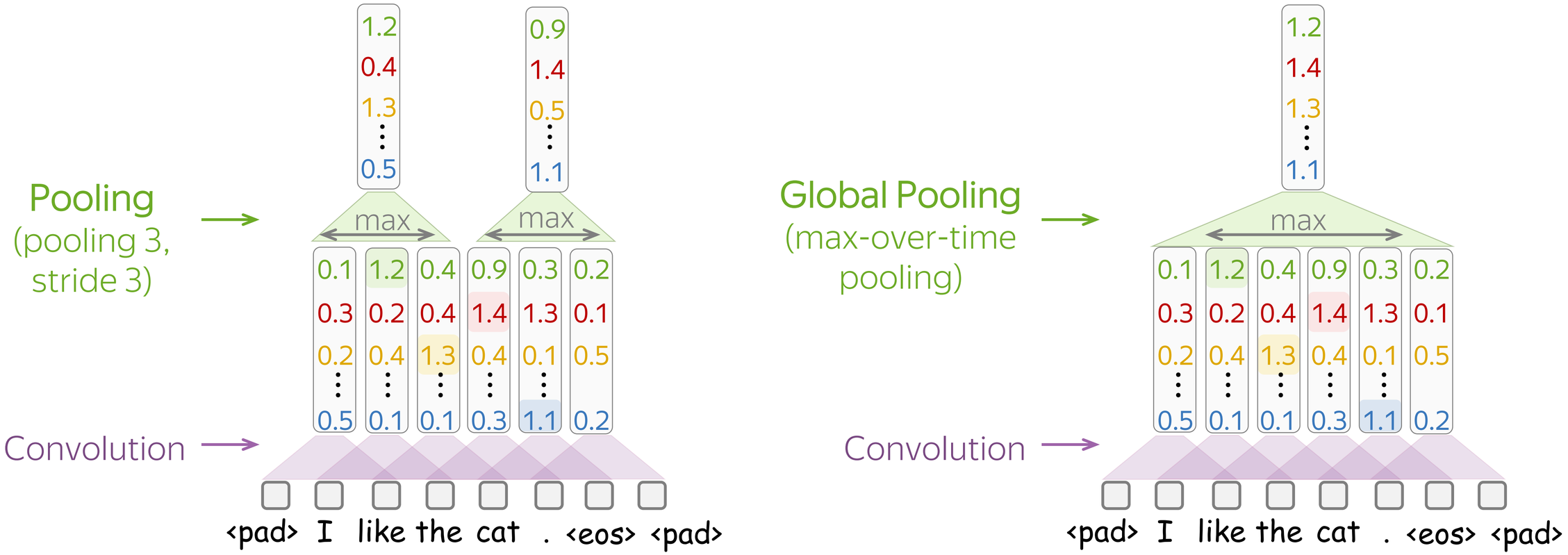
Building Blocks: Pooling and Global Pooling

- **Pooling**: aggregate features in some area
- **Global pooling**: aggregate features over all input

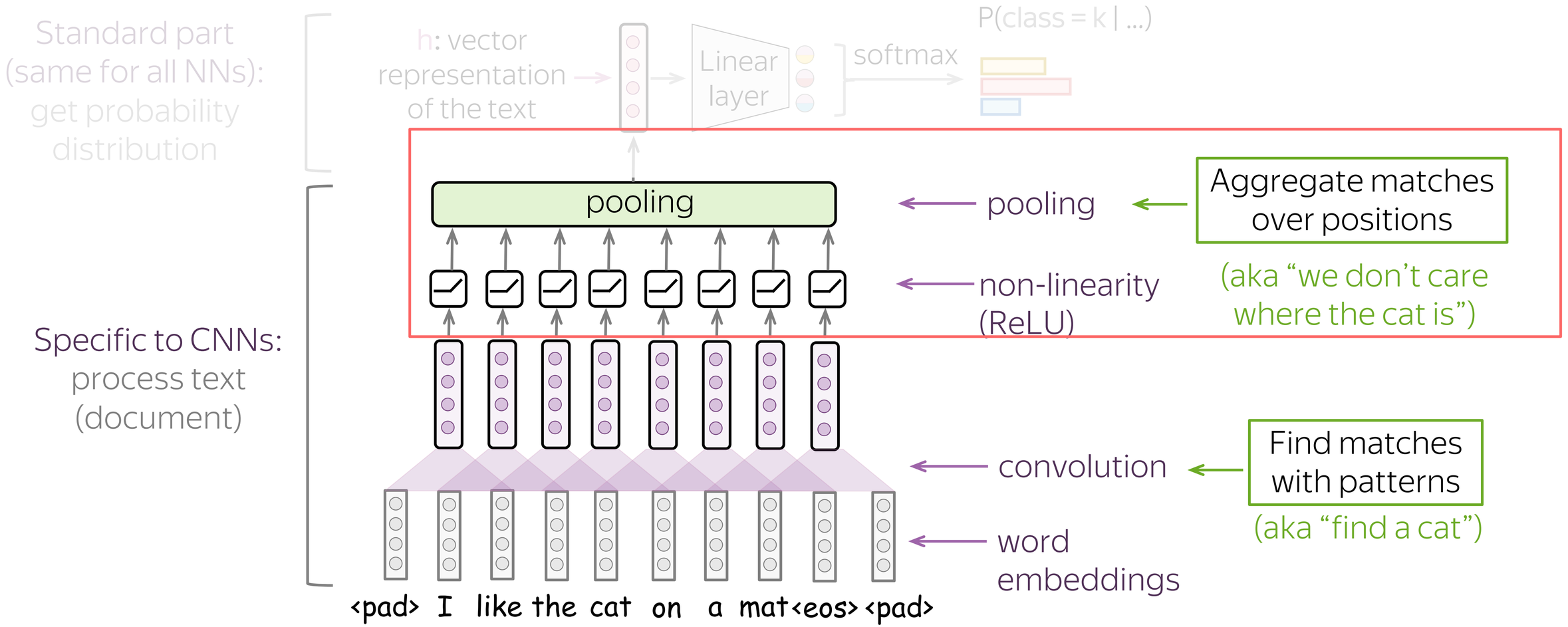


Building Blocks: Pooling and Global Pooling

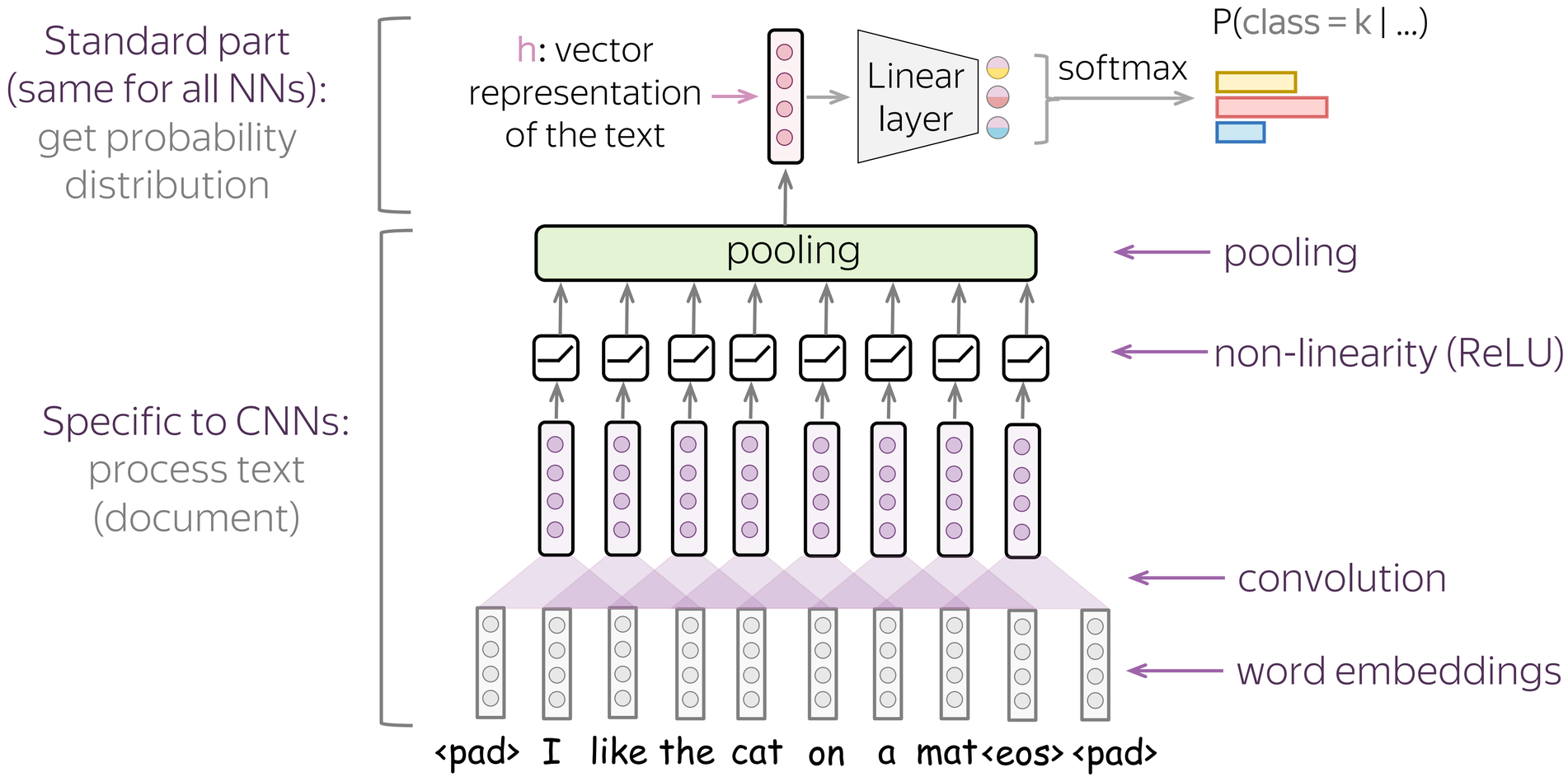
- **Pooling**: aggregate features in some area
- **Global pooling**: aggregate features over all input



A Typical Model: Convolution + Pooling

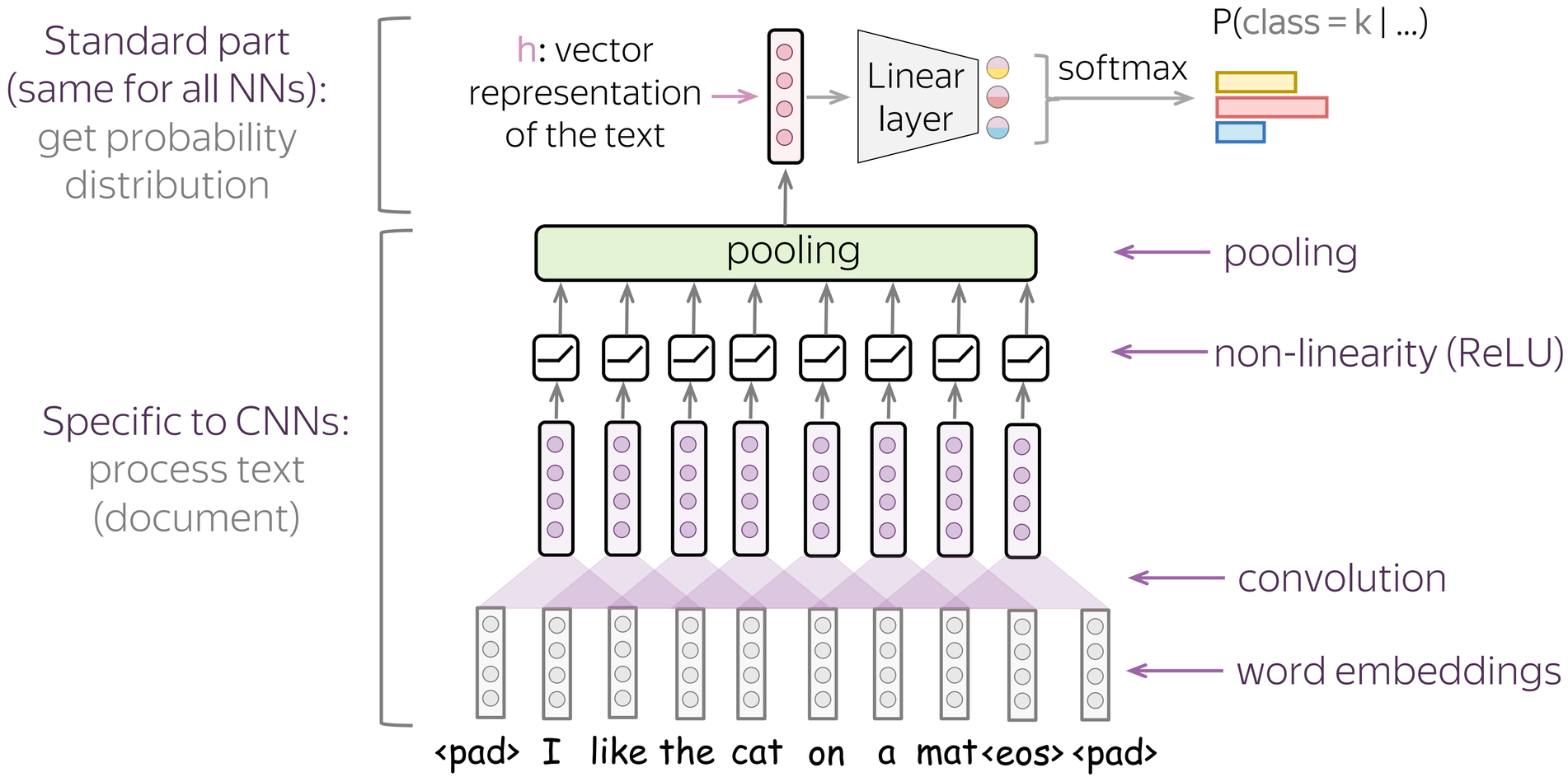


A Typical Model: Convolution + Pooling



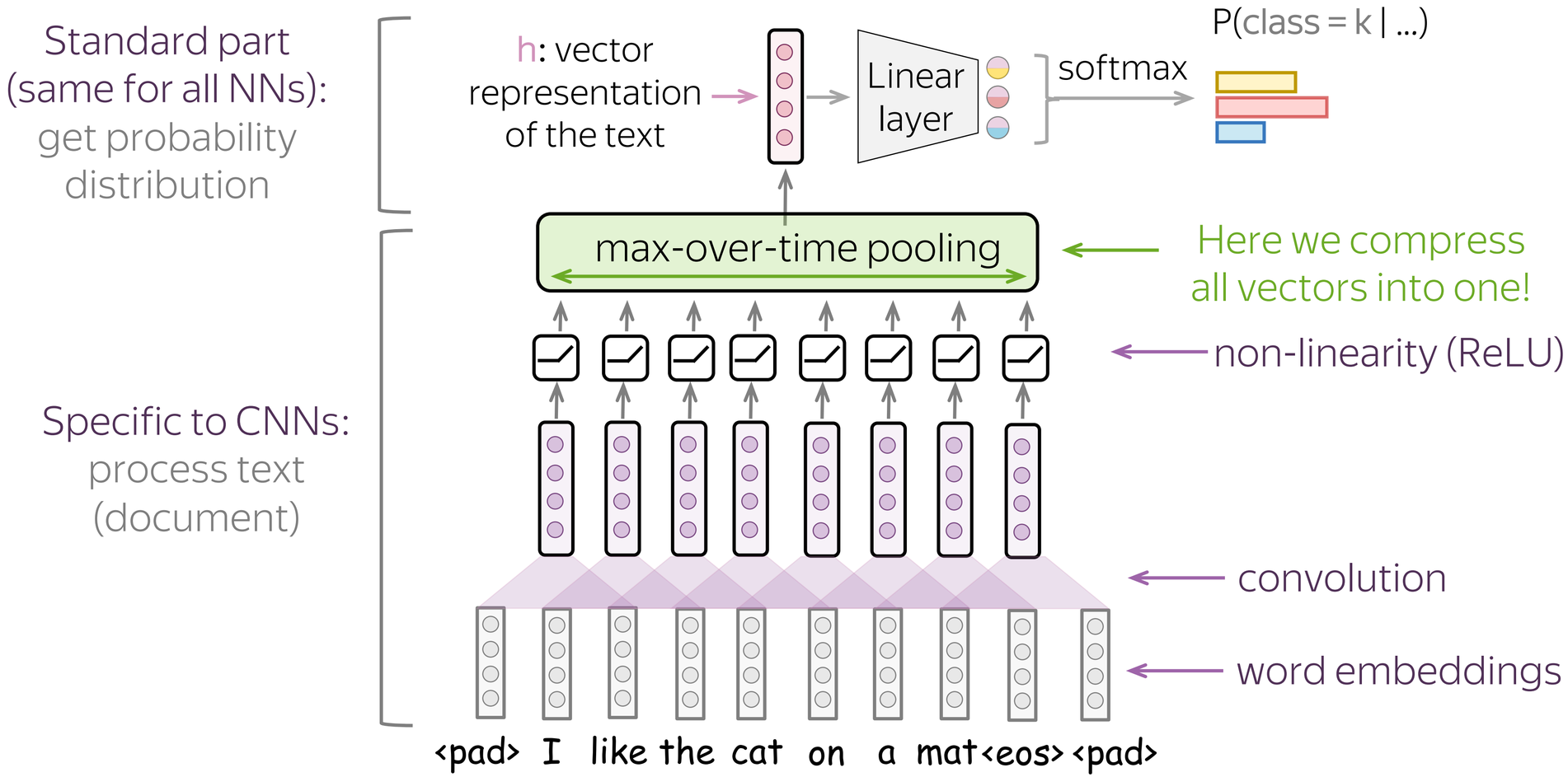
A Typical Model: Convolution + Pooling

> We need a model that can produce a **fixed-sized** vector for inputs of **different** lengths.



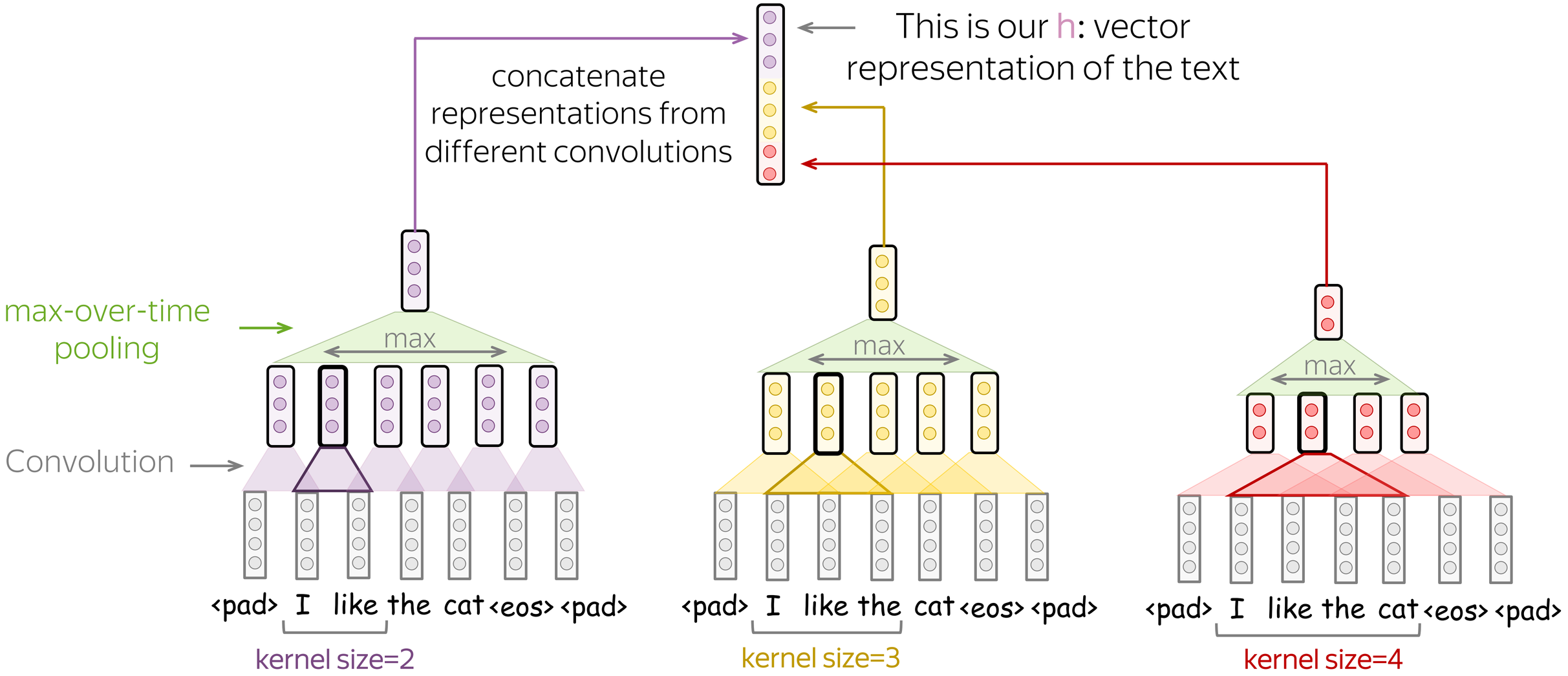
A Typical Model: Convolution + Pooling

> We need a model that can produce a **fixed-sized** vector for inputs of **different** lengths.



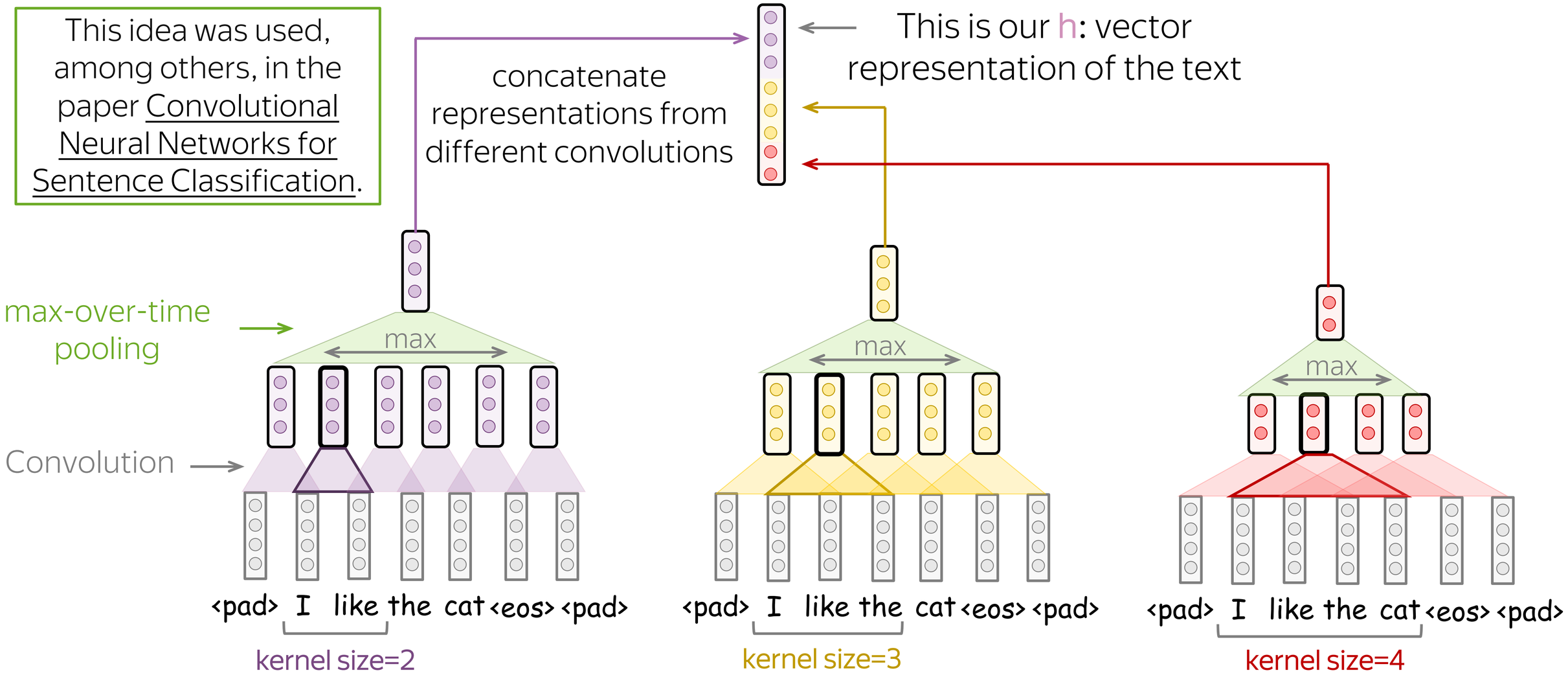
Convolutional Models for Text Classification

- Several Convolutions with Different Kernel Sizes



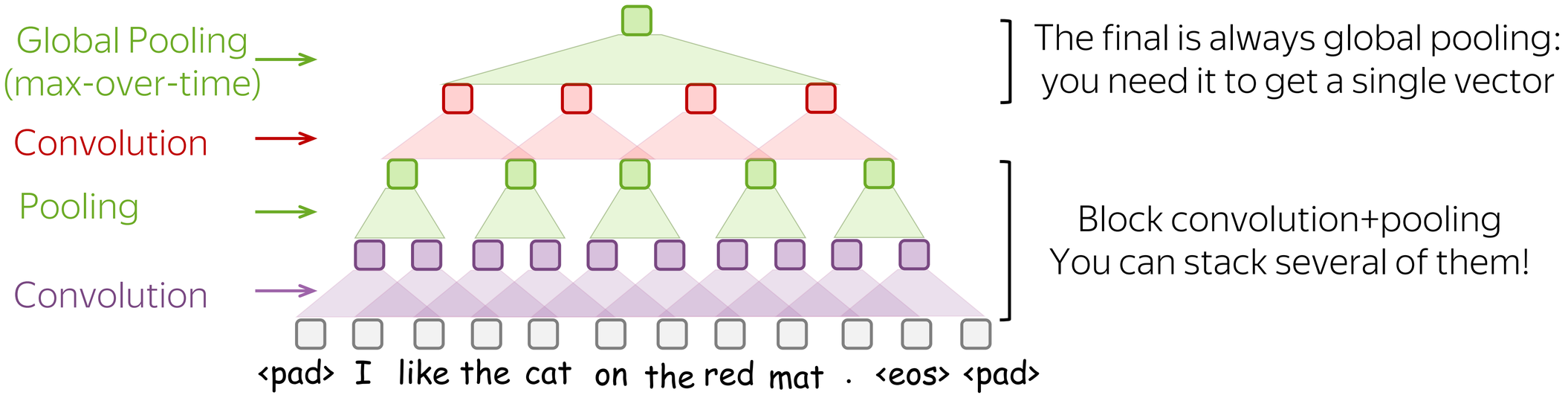
Convolutional Models for Text Classification

- Several Convolutions with Different Kernel Sizes



Convolutional Models for Text Classification

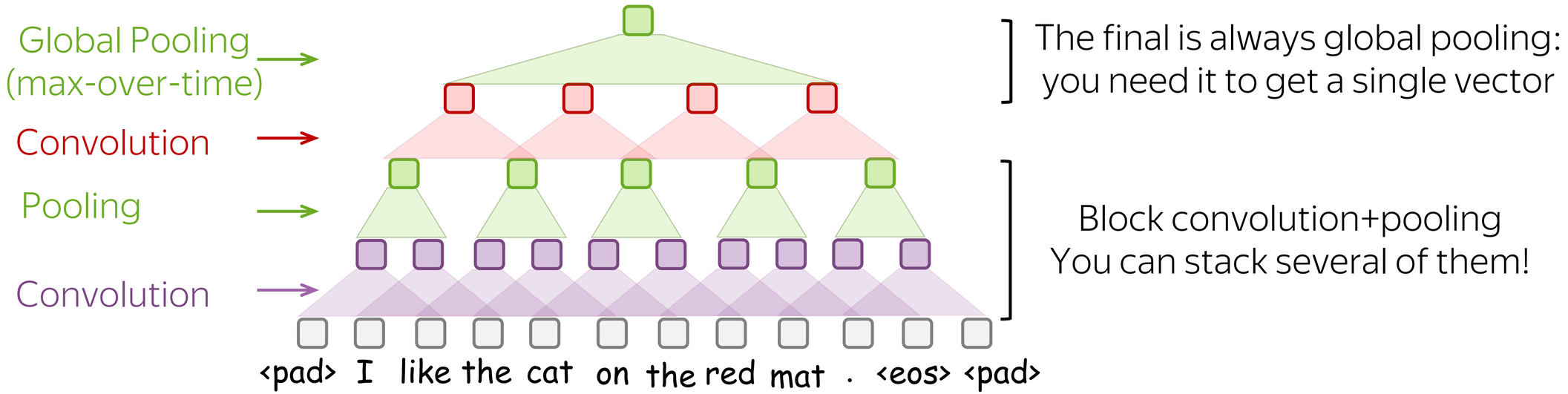
- Stack Several Blocks Convolution+Pooling



Convolutional Models for Text Classification

- Stack Several Blocks Convolution+Pooling

This idea was used, among others, in the paper
Character-level Convolutional Networks for Text Classification.



What is going to happen:

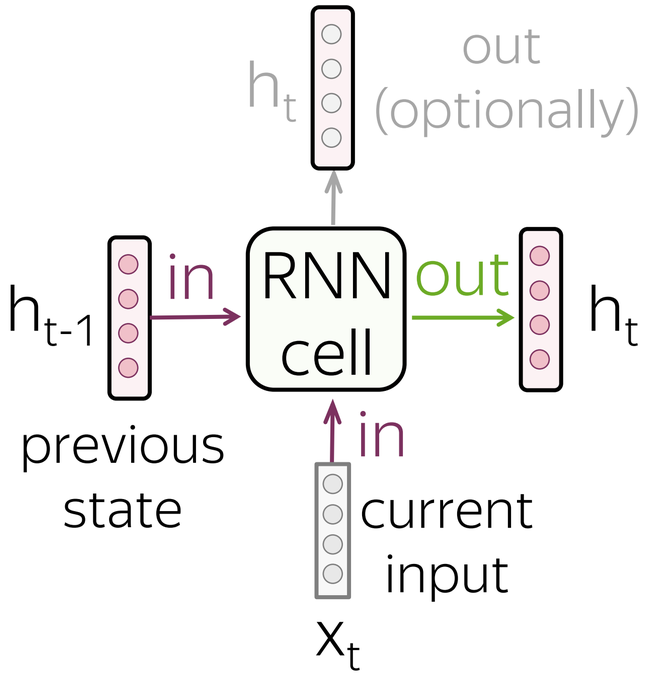
- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
  - High-Level View
  - Training: Cross-Entropy
  - Models: (Weighted) BOW
  - Models: Convolutional
  - Models: Recurrent
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
  - High-Level View
  - Training: Cross-Entropy
  - Models: (Weighted) BOW
  - Models: Convolutional
  - Models: Recurrent
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

Basics: Recurrent Neural Networks

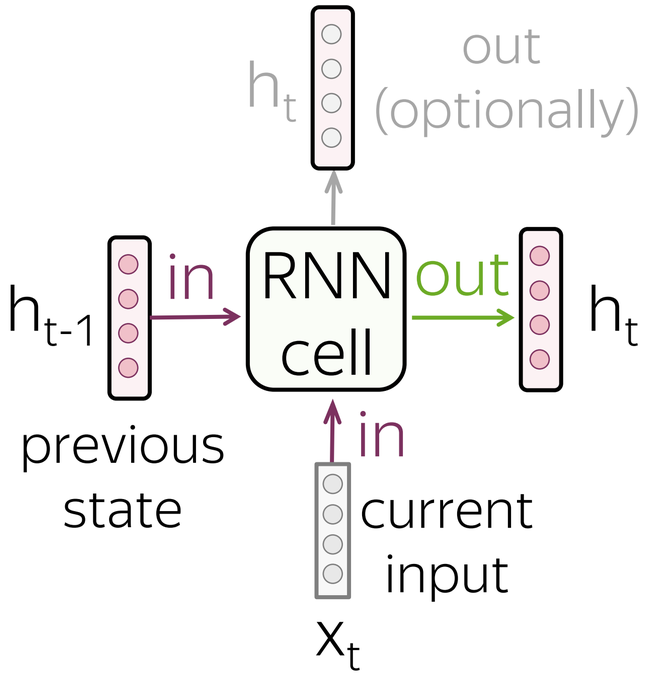
- recurrent cell



For more details of RNN basics, look at the Colah's blog post.

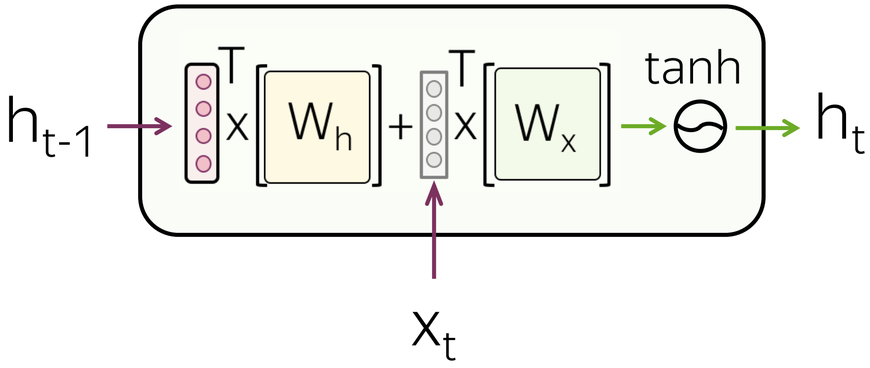
Basics: Recurrent Neural Networks

- recurrent cell



- vanilla RNN

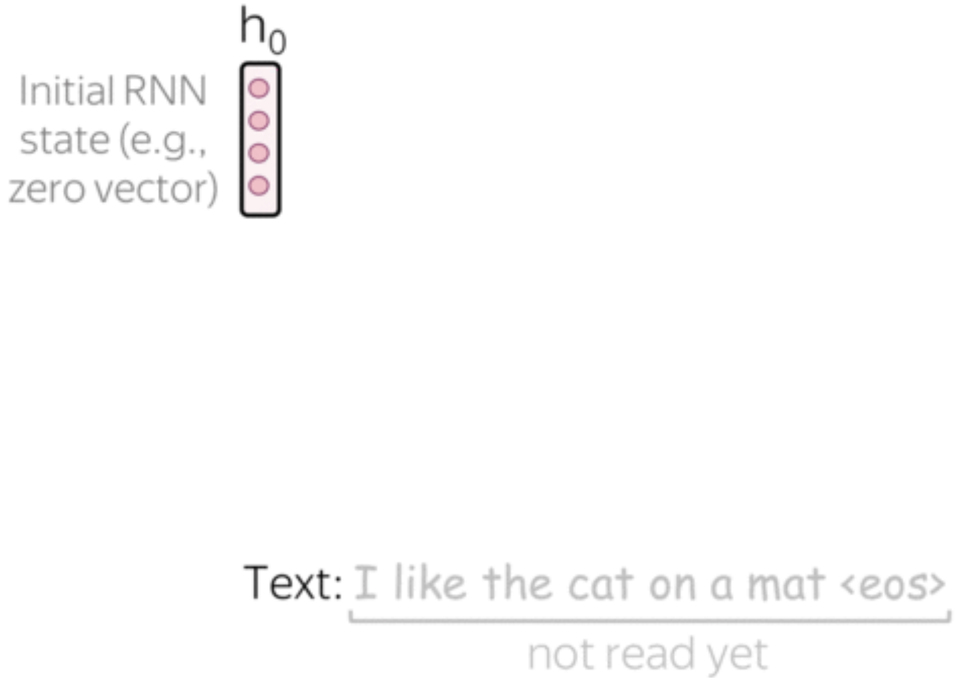
$$h_t = \tanh(h_{t-1} W_h + x_t W_x)$$



For more details of RNN basics, look at the Colah's blog post.

Basics: Recurrent Neural Networks

- RNN reads a text



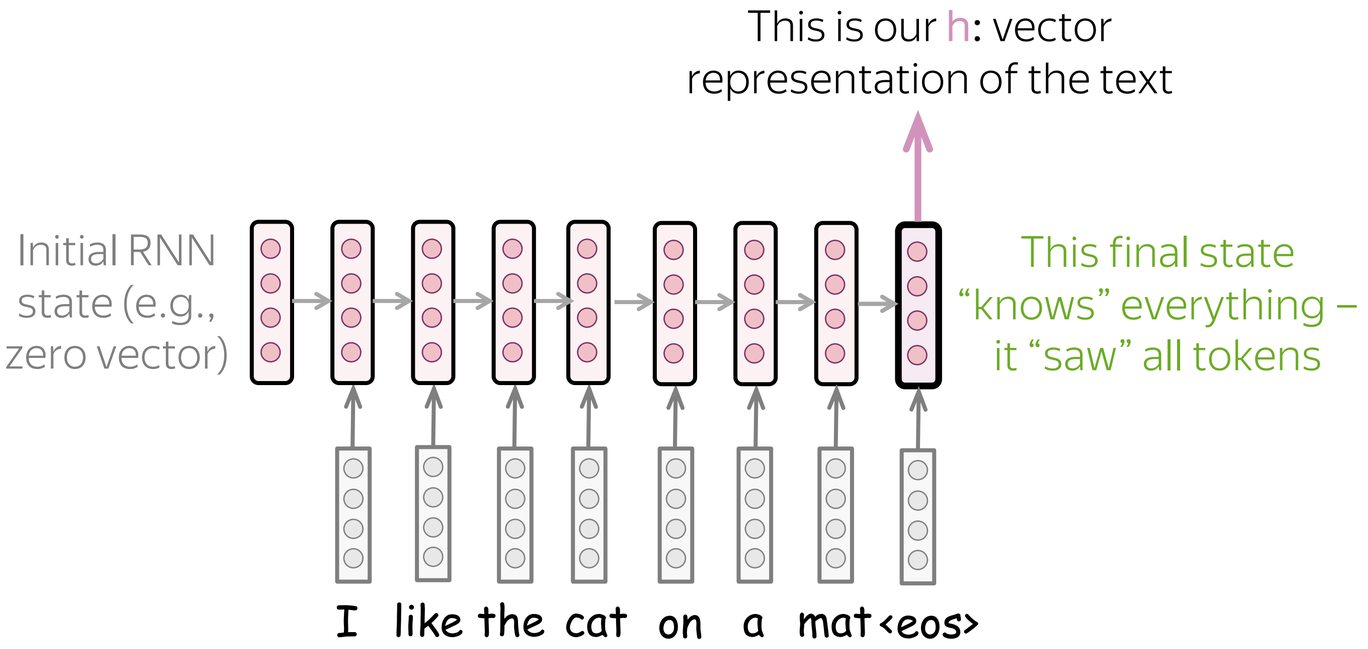
Recurrent Models for Text Classification

> We need a model that can produce a **fixed-sized** vector for inputs of **different** lengths.

Recurrent Models for Text Classification

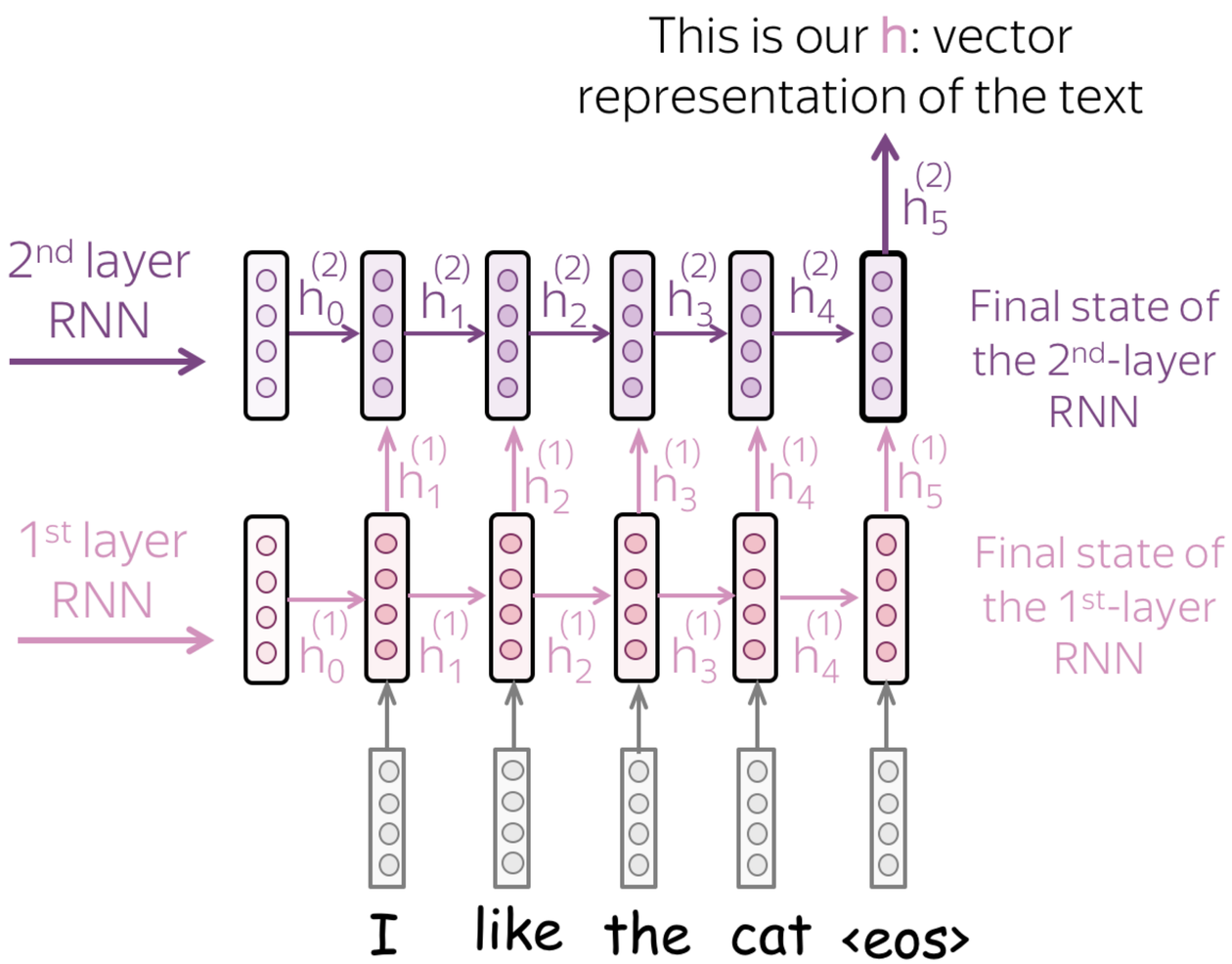
> We need a model that can produce a **fixed-sized** vector for inputs of **different** lengths.

- **simple**: read a text, take the final state



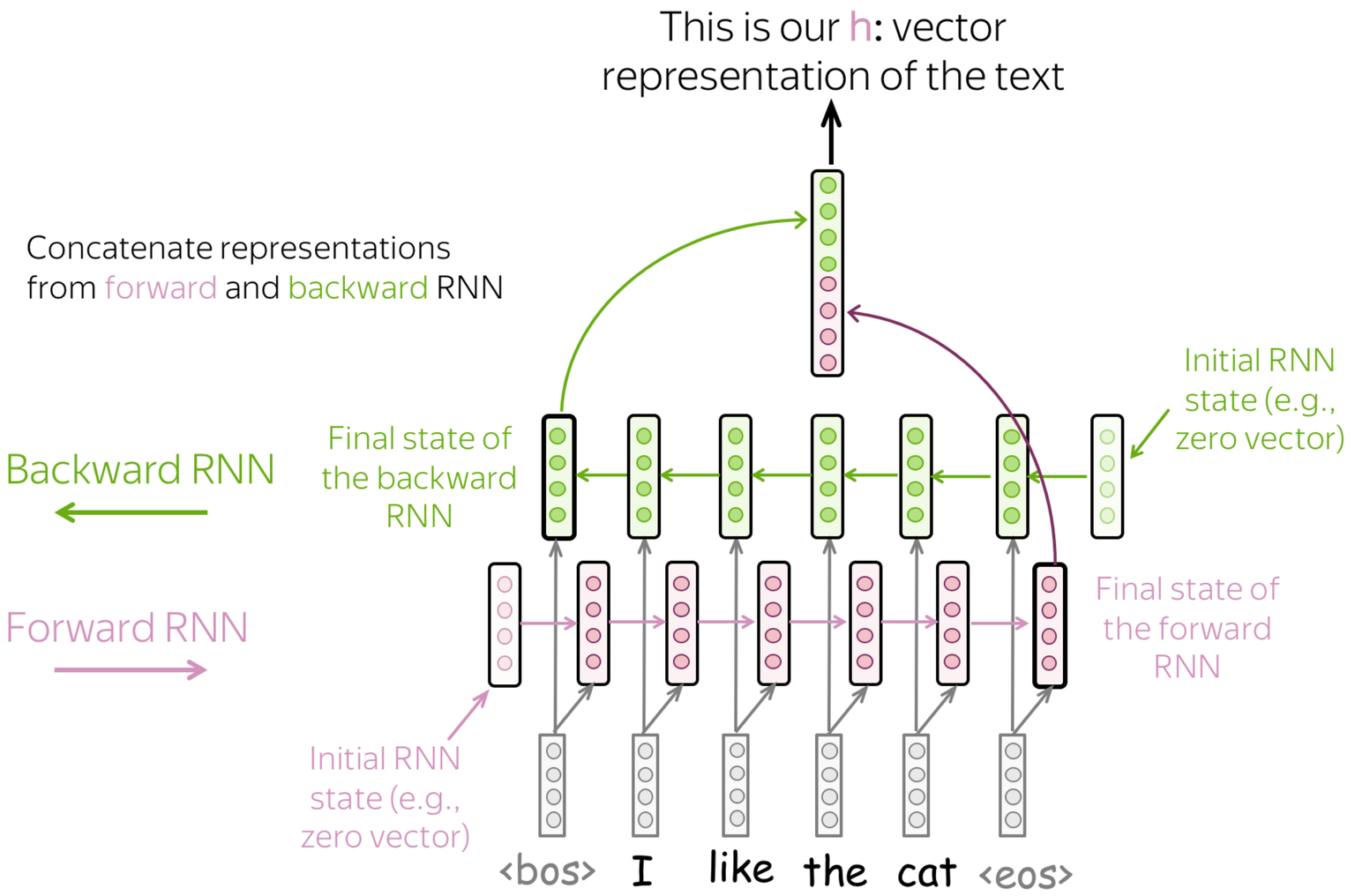
Recurrent Models for Text Classification

- **Multiple layers**: feed the states from one RNN to the next



Recurrent Models for Text Classification

- **bidirectional**: use final states from forward and backward RNNs



What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
  - High-Level View
  - Training: Cross-Entropy
  - Models: (Weighted) BOW
  - Models: Convolutional
  - Models: Recurrent
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification → NLP Course For You: if you want, read here
- Practical Tips
-  Analysis and Interpretability

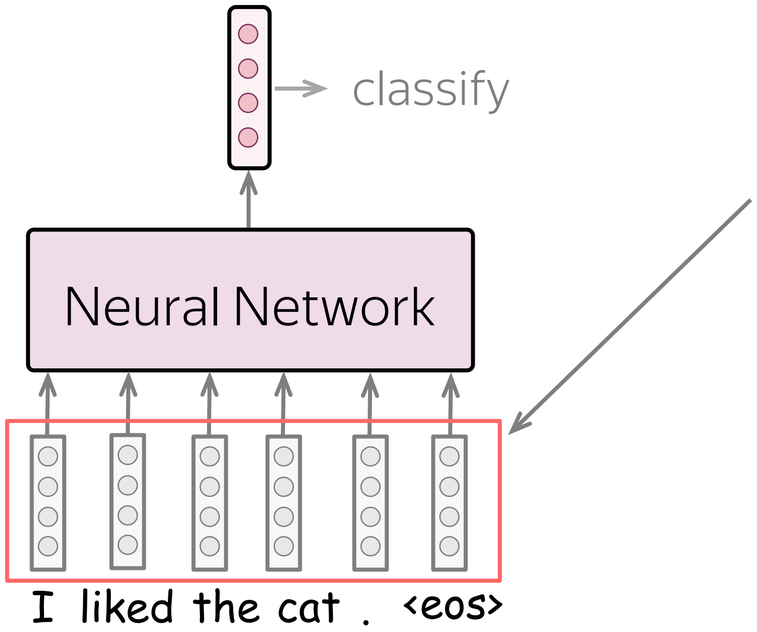
What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
  - Word Embeddings
  - Data Augmentation
-  Analysis and Interpretability

Word Embeddings: How to Deal with them?

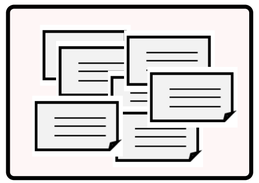


Input word embeddings:

- Train from scratch
- Take pretrained (Word2Vec, GloVe)
- Initialize with pretrained, then fine-tune

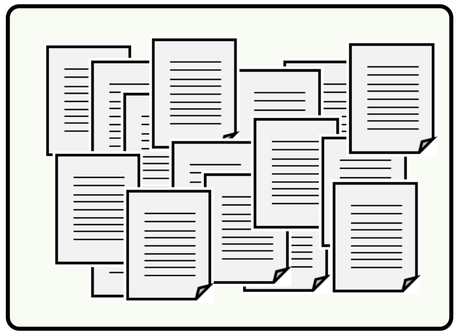
Which data do we have?

Training data for text classification (labeled)



- Not huge, or not diverse, or both
- Domain: task-specific

Training data for word embeddings (unlabeled)

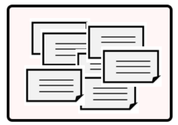


- Huge diverse corpus (e.g., Wikipedia)
- Domain: general

Word Embeddings: How to Deal with them?

- Train from scratch

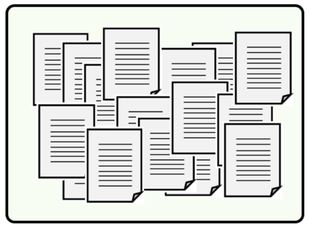
What they will know:



May be not enough to learn relationships between words

- Take pretrained (Word2Vec, GloVe)

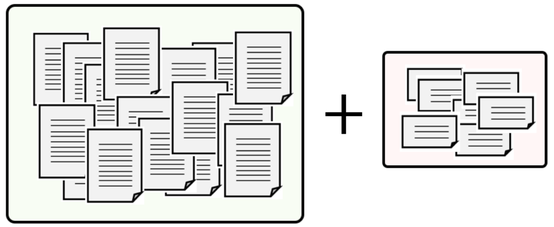
What they will know:



Know relationships between words, but are **not** specific to the task

- Initialize with pretrained, then fine-tune

What they will know:

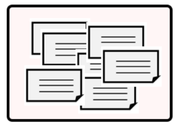


Know relationships between words and adapted for the task

Word Embeddings: How to Deal with them?

- Train from scratch

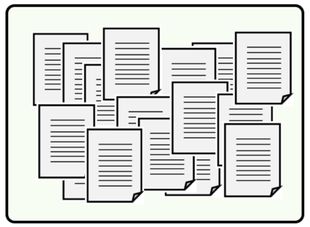
What they will know:



May be not enough to learn relationships between words

- Take pretrained (Word2Vec, GloVe)

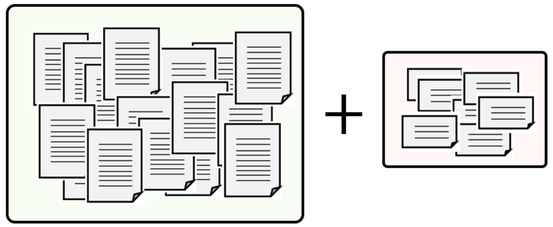
What they will know:



Know relationships between words, but are **not** specific to the task

- Initialize with pretrained, then fine-tune

What they will know:



Know relationships between words and adapted for the task

"**Transfer**" knowledge from a huge unlabeled corpus to your task-specific model

We'll learn more about this later in the course!

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
  - Word Embeddings
  - Data Augmentation
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
  - Word Embeddings
  - Data Augmentation
-  Analysis and Interpretability

Data Augmentation: Get More Data for Free

Data augmentation alters your dataset in different ways to get alternative versions of the same training example.

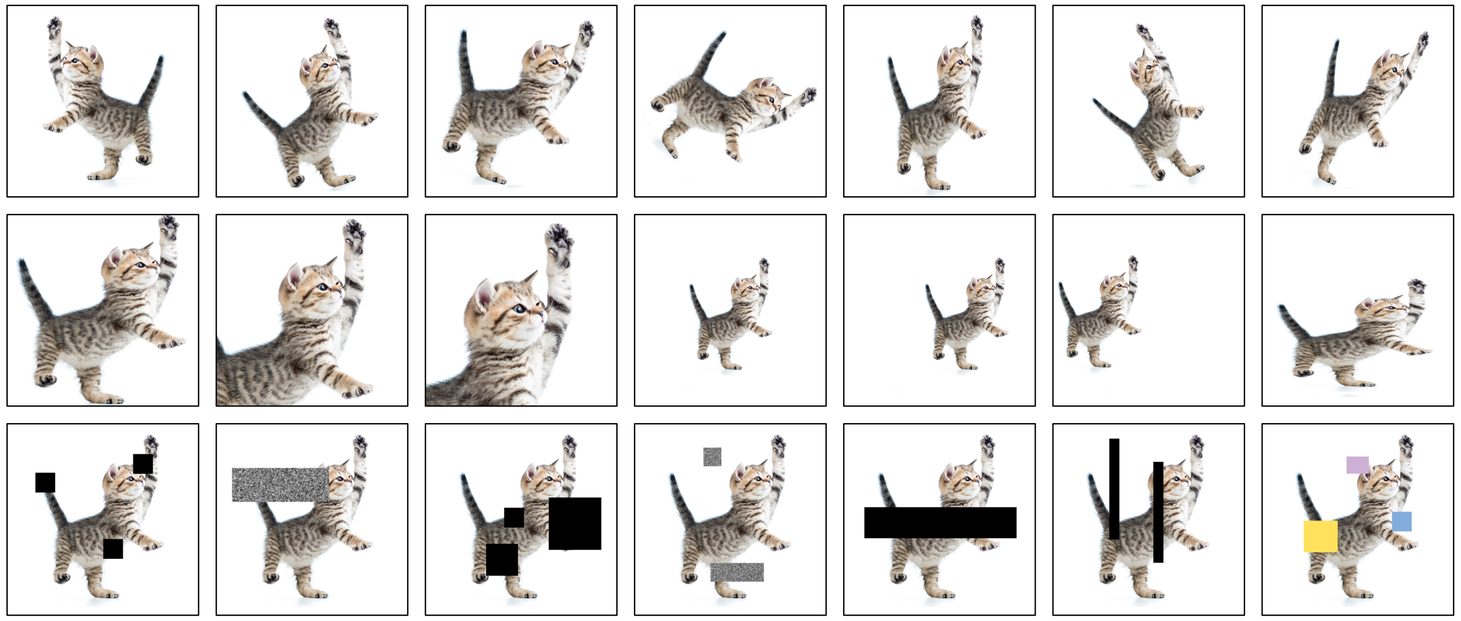
It can increase:

- the amount of data
- diversity of data

Data Augmentation for Images

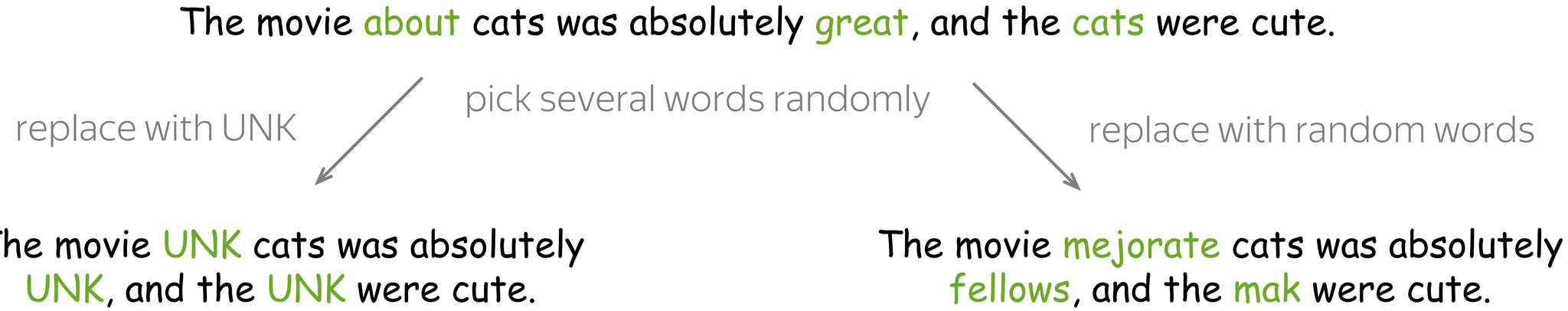
- Flipping an image
- Geometrical transformations (rotation, stretching, zoom in/out)
- Covering some parts with patches

original image



Data Augmentation for Text

- **word dropout** - the most simple and popular



Data Augmentation for Text

- **word dropout** - the most simple and popular

The movie about cats was absolutely great, and the cats were cute.

pick several words randomly

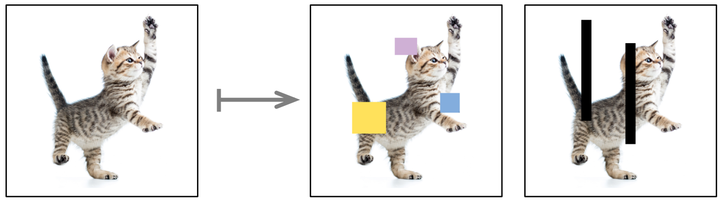
replace with UNK

replace with random words

The movie UNK cats was absolutely UNK, and the UNK were cute.

The movie mejorate cats was absolutely fellows, and the mak were cute.

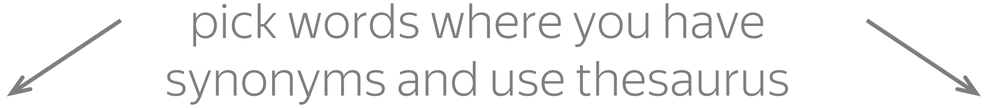
For images, this is similar to covering some parts: these parts are “dropped” from an image



Data Augmentation for Text

- use external resource (e.g., thesaurus) - a bit more complicated

The movie about cats was absolutely great, and the cats were cute.



The film about cats was certainly great, and the cats were nice.

The video about cats was completely great, and the cats were charming.

Data Augmentation for Text

- use separate models for paraphrasing - if you really care

The movie about cats was absolutely great, and the cats were cute.

En-Ru ↓ Yandex Translate

Фильм о кошках был замечательный, и кошки были милые.

Ru-En ↓ Yandex Translate

The cat movie was just great, and the cats were cute.

En-Zh ↓ Google Translate

關於貓的電影絕對很棒，而且貓很可愛。

Zh-En ↓ Google Translate

Movies about cats are absolutely great, and cats are cute.

En-Ja ↓ Google Translate

猫の映画は本当に素晴らしく、猫はかわいい。

Ja-En ↓ Yandex Translate

The Cat movie is really nice and the cat is cute.

Data Augmentation for Text

- use separate models for paraphrasing - if you really care

The movie about cats was absolutely great, and the cats were cute.

En-Ru ↓ Yandex Translate

Фильм о кошках был замечательный, и кошки были милые.

Ru-En ↓ Yandex Translate

The cat movie was just great, and the cats were cute.

En-Zh ↓ Google Translate

關於貓的電影絕對很棒，而且貓很可愛。

Zh-En ↓ Google Translate

Movies about cats are absolutely great, and cats are cute.

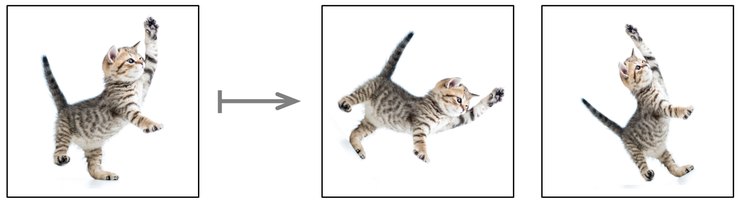
En-Ja ↓ Google Translate

猫の映画は本当に素晴らしく、猫はかわいい。

Ja-En ↓ Yandex Translate

The Cat movie is really nice and the cat is cute.

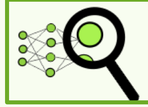
For images, this is similar to geometric transformations: we change text, but want to preserve all meaning.



What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

Analyzing Convolutional Filters

Lower layers

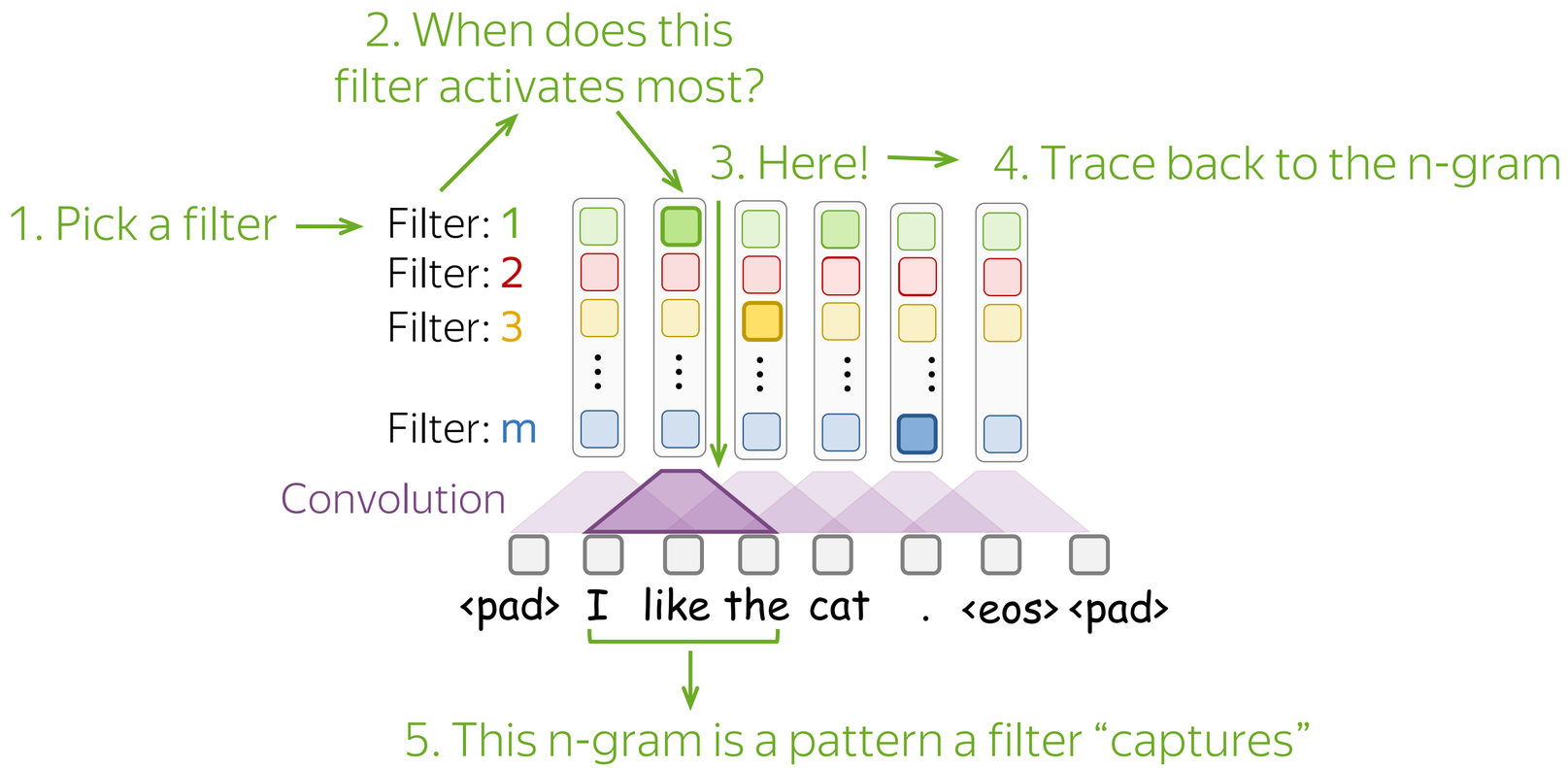
Middle layers

Upper layers

Examples of patterns captured by convolution filters for images.
The examples are from Activation Atlas from distill.pub.

Analyzing Convolutional Filters

- Find which patterns activate neurons



Analyzing Convolutional Filters

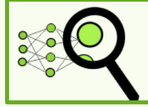
filter	Top n-gram	Score
1	poorly designed junk	7.31
2	simply would not	5.75
3	a minor drawback	6.11
4	still working perfect	6.42
5	absolutely gorgeous .	5.36
6	one little hitch	5.72
7	utterly useless .	6.33
8	deserves four stars	5.56
9	a mediocre product	6.91

Top n-grams for filter 4	Score
still working perfect	6.42
works - perfect	5.78
isolation proves invaluable	5.61
still near perfect	5.6
still working great	5.45
works as good	5.44
still holding strong	5.37

A filter activates for a family of n-grams with similar meaning

The example is from the paper Understanding Convolutional Neural Networks for Text Classification.

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

What is going to happen:

- Examples of classification tasks
- General View: Features + Classifier
- Models: Generative vs Discriminative
- Classical Methods
- Neural Methods
- Multi-Label Classification
- Practical Tips
-  Analysis and Interpretability

 Learn more in the NLP Course For You → This is up to You!

Thank you!

Lena Voita

PhD student, Uni Edinburgh & Uni Amsterdam

Facebook PhD Fellow in NLP

lena-voita@hotmail.com

https://lena-voita.github.io

@lena_voita

lena-voita