# Text classification

Vlad Shakhuro

MISIS UNIVERSITY

2 October 2025

# Outline

1. Classification tasks

2. General pipeline

3. Generative and discriminative models

4. Classical methods

5. Neural methods

# Classification tasks

- Binary
- Multi-class
- Multi-label

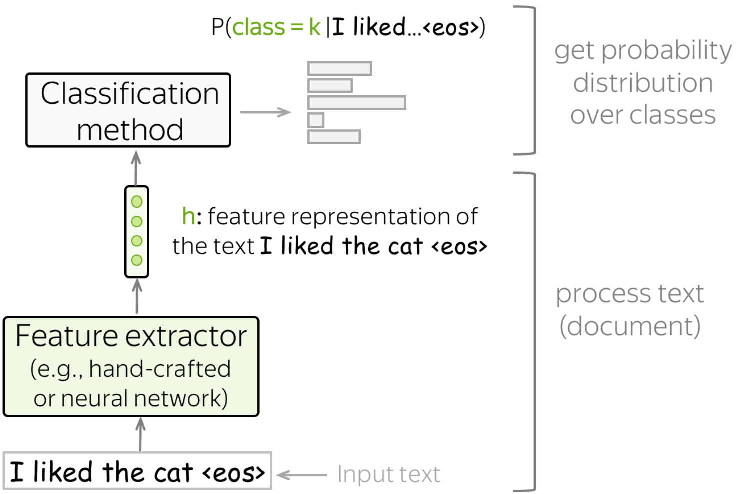
# Classification datasets

| 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 |

# Outline

1. Classification tasks
2. General pipeline
3. Generative and discriminative models
4. Classical methods
5. Neural methods

# Obtain features and classify them



# Outline

1. Classification tasks

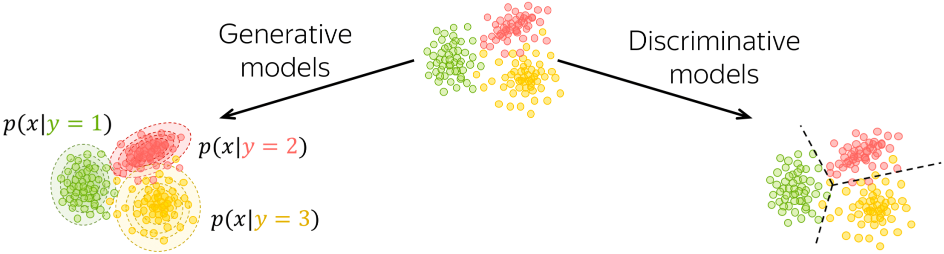
2. General pipeline

3. Generative and discriminative models

4. Classical methods

5. Neural methods

# Generative and discriminative models




Learn: data distribution $p(x,y) = p(x|y) \cdot p(y)$

How predict: $y = \arg\max_k P(x, y = k) =$

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

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

How predict: $y = \arg\max_k P(y=k|x)$

# Outline

1. Classification tasks
2. General pipeline
3. Generative and discriminative models
4. Classical methods
5. Neural methods

# Naive Bayes

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

$$\arg\max_k P(x \mid y = k)P(y = k)$$

# Naive Bayes

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

# Naive Bayes

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

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

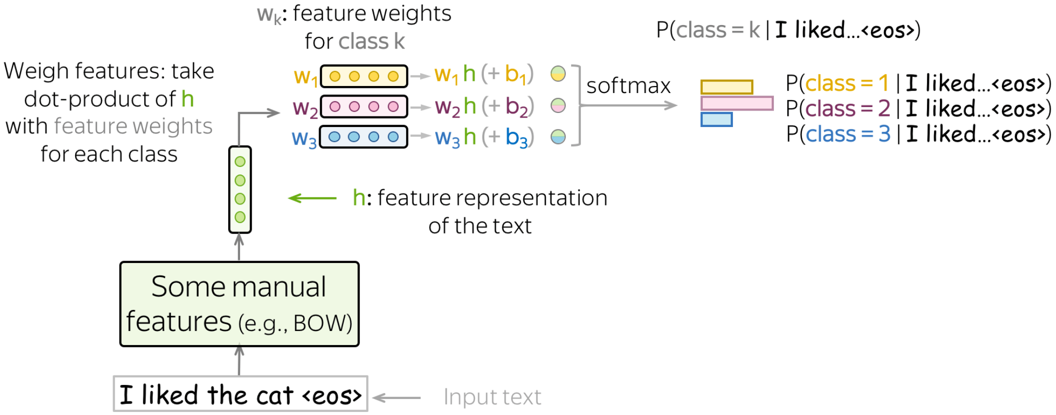
Naive assumption:

- **bag of words** – word order doesn't matter
- **conditional independence** – features (words) are independent given the class

Now we can compute probabilities simply by counting frequences in the training data

# Logistic regression

$$y^* = \arg\max_k \frac{\exp(w_k h)}{\sum_{i=1}^{K}(w_i h)}$$



# Training: maximizing likelihood

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

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

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

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

# Outline

1. Classification tasks

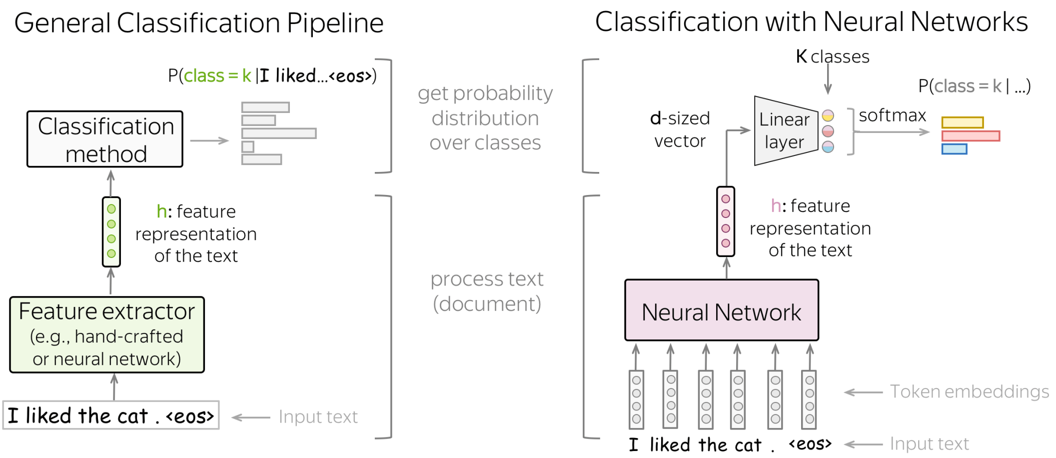
2. General pipeline

3. Generative and discriminative models

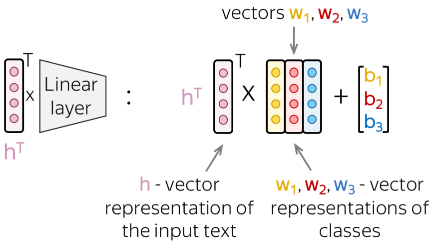
4. Classical methods

5. Neural methods

# Neural networks in general

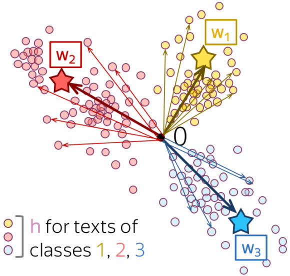


# Neural networks in general



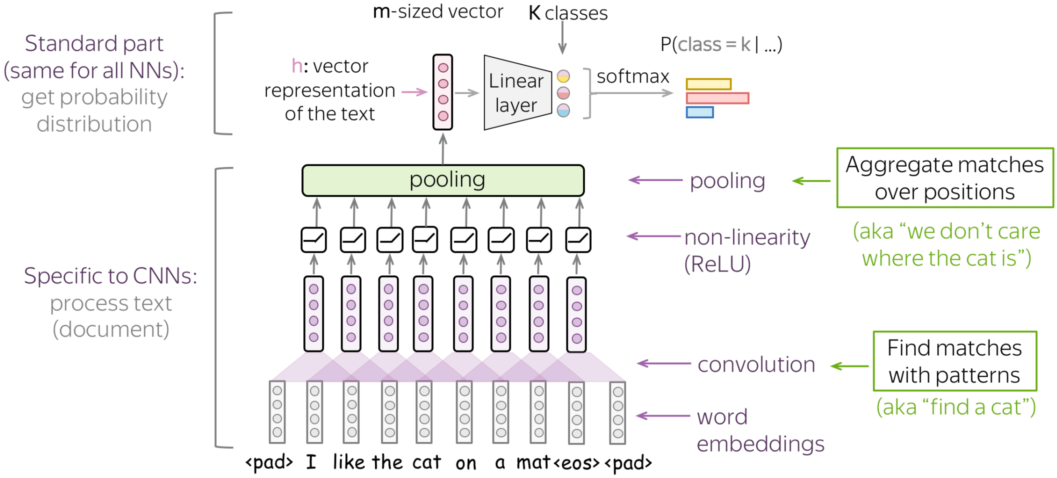

What NN learns (hopefully):

Text vectors point in the direction of the corresponding class vectors

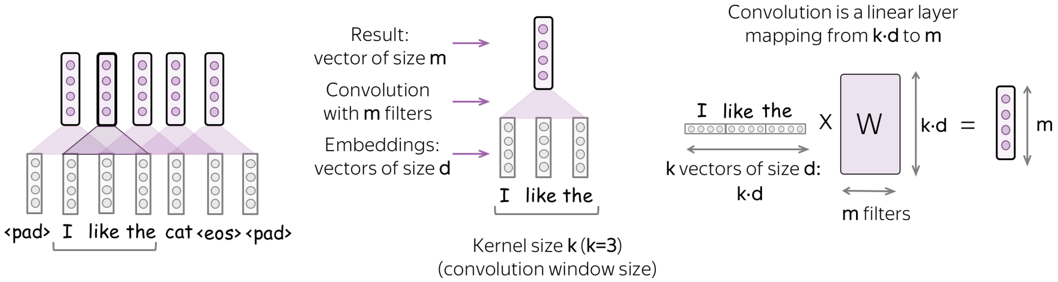


# Convolutional networks in CV

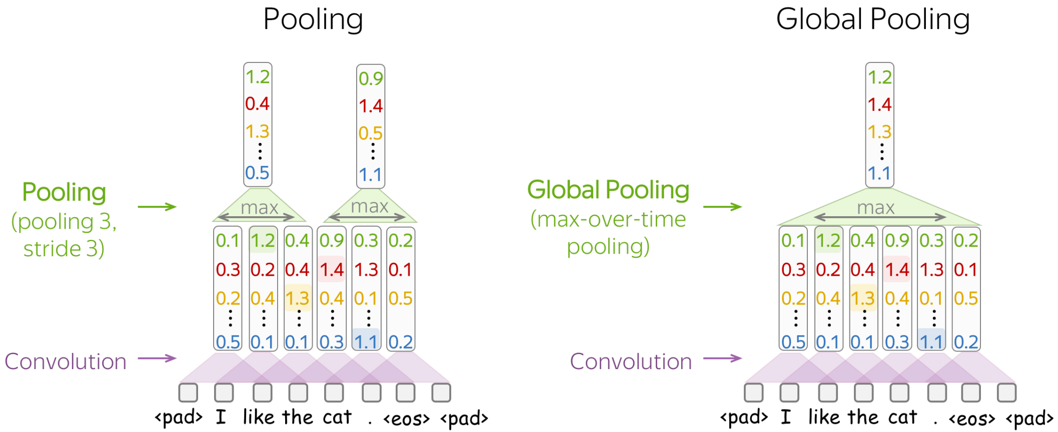
# Convolutional networks for text



# Convolution operation



# Pooling



Pooling
Global Pooling
Pooling
(pooling 3, stride 3)
Convolution
max
1.2
0.4
1.3
0.5
0.9
1.4
0.5
1.1
0.1 1.2 0.4 0.9 0.3 0.2
0.3 0.2 0.4 1.4 1.3 0.1
0.2 0.4 1.3 0.4 0.1 0.5
0.5 0.1 0.1 0.3 1.1 0.2
<pad> I like the cat . <eos> <pad>
Global Pooling
(max-over-time pooling)
Convolution
max
1.2
1.4
1.3
1.1
0.1 1.2 0.4 0.9 0.3 0.2
0.3 0.2 0.4 1.4 1.3 0.1
0.2 0.4 1.3 0.4 0.1 0.5
0.5 0.1 0.1 0.3 1.1 0.2
<pad> I like the cat . <eos> <pad>

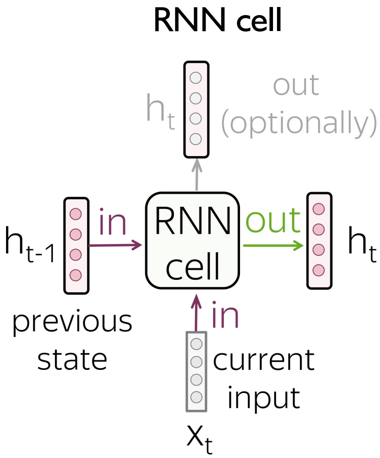
# What convolutional networks learn?

| 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 |
|---|---|---|
| 1 | still working perfect | 6.42 |
| 2 | works - perfect | 5.78 |
| 3 | isolation proves invaluable | 5.61 |
| 4 | still near perfect | 5.6 |
| 5 | still working great | 5.45 |
| 6 | works as good | 5.44 |
| 7 | still holding strong | 5.37 |

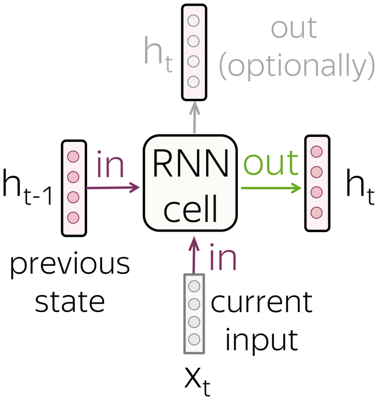
A filter activates for a family of n-grams with similar meaning

# Recurrent networks



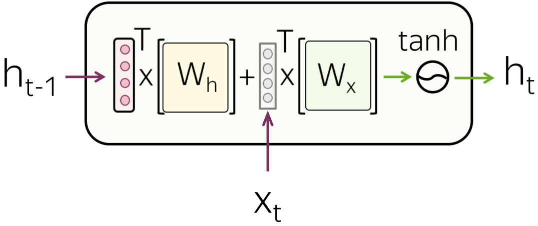
# Recurrent networks

## RNN cell

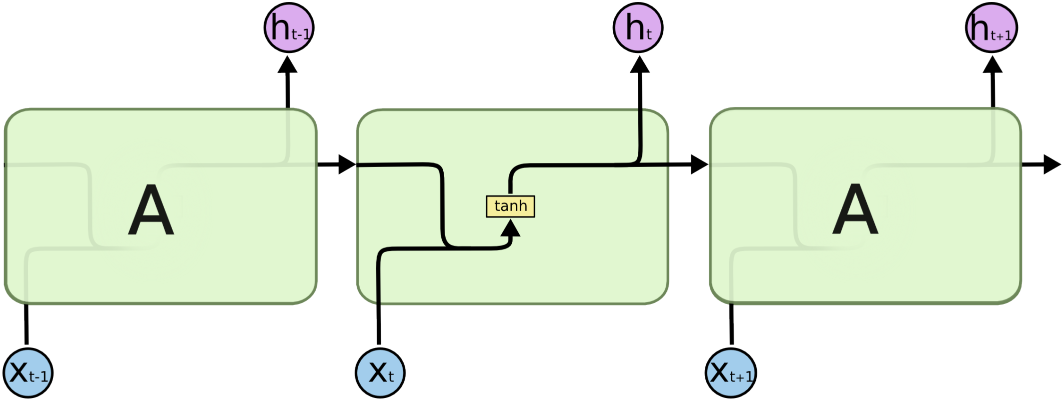



## vanilla RNN

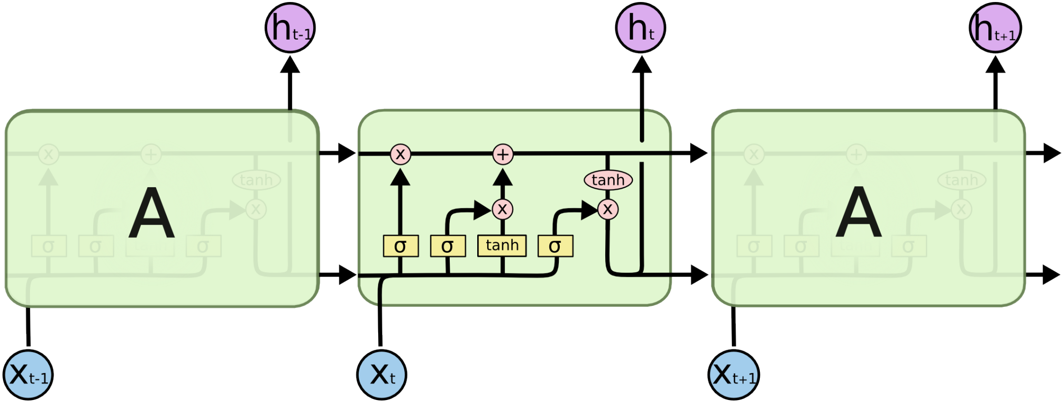
$$h_t = \tanh(h_{t-1} W_h + x_t W_x)$$



# RNN

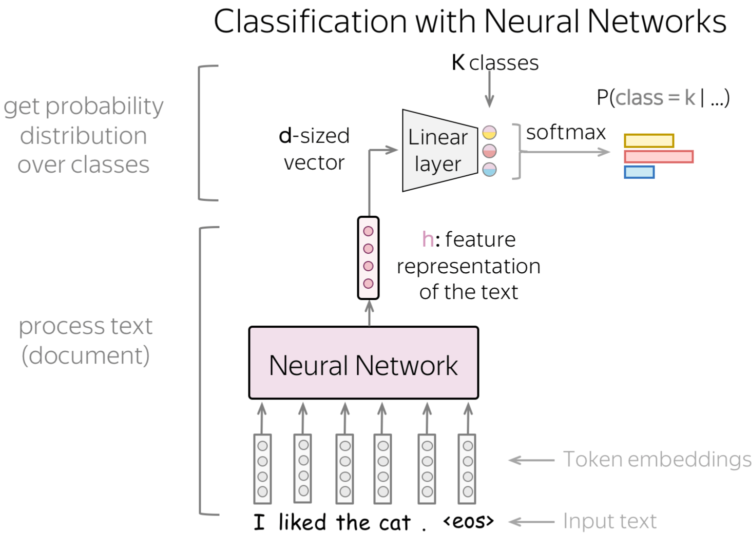


# Long short-term memory (LSTM)




https://colah.github.io/posts/2015-08-Understanding-LSTMs/

# Training from scratch and finetuning



# Conclusion

We reviewed following topics:

- classification task and datasets
- general classification pipeline
- generative and discriminative models
- classical methods: naive bayes and logistic regression
- neural networks: convolutional, recurrent