

# Few-Shot Learning

Сабина Даянова

# План

Определение и  
мотивация



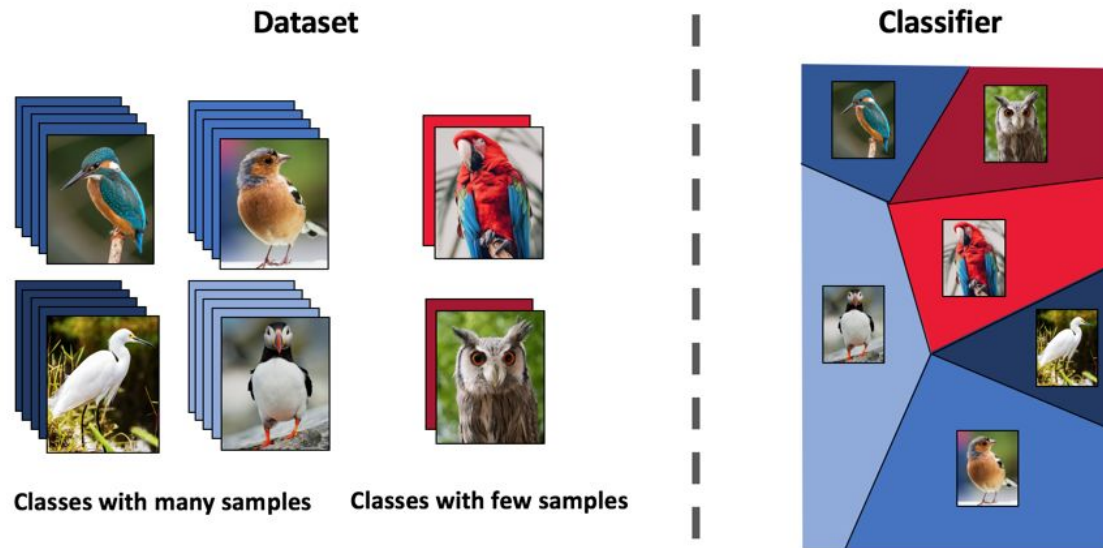
Постановка  
проблемы

Многочисленные  
решения



Примеры и  
результаты

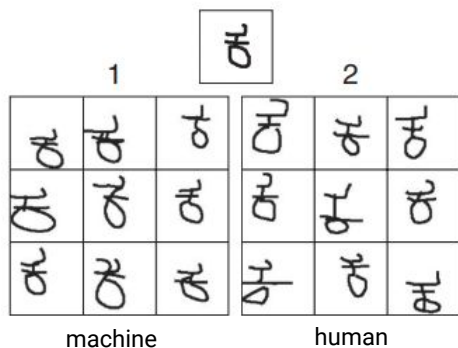
Few-Shot Learning - тип задач в машинном обучении, в котором доступно малое количество данных



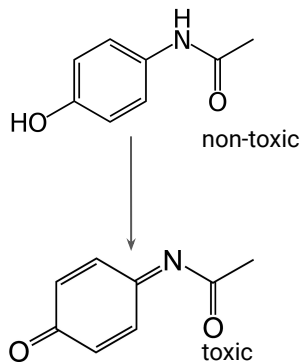
# Мотивация?

## Сценарии

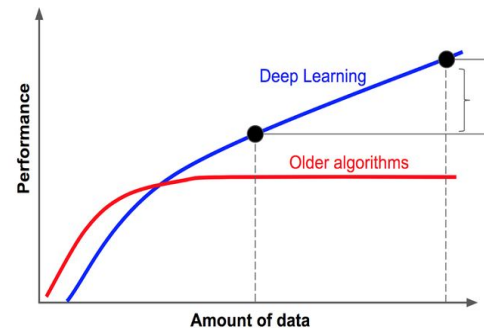
человеко-подобное обучение



работа с редкими случаями



уменьшение затрат при вычислении



# Постановка задачі

Expected risk  
(ожидаемый)

$$R(h) = \int \ell(h(x), y) dp(x, y) = \mathbb{E}[\ell(h(x), y)]$$

Empirical risk  
(наблюдаемый)

$$R_I(h) = \frac{1}{I} \sum_{i=1}^I \ell(h(x_i), y_i)$$


- $p(x, y)$  - распределение  $x, y$
- $\mathcal{H}$  - пространство гипотез, определяемое моделью
- $I$  - количество объектов в датасете

- $\hat{h} = \arg \min_h R(h)$
- $h^* = \arg \min_{h \in \mathcal{H}} R(h)$
- $h_I = \arg \min_{h \in \mathcal{H}} R_I(h)$


Разложение ошибки:

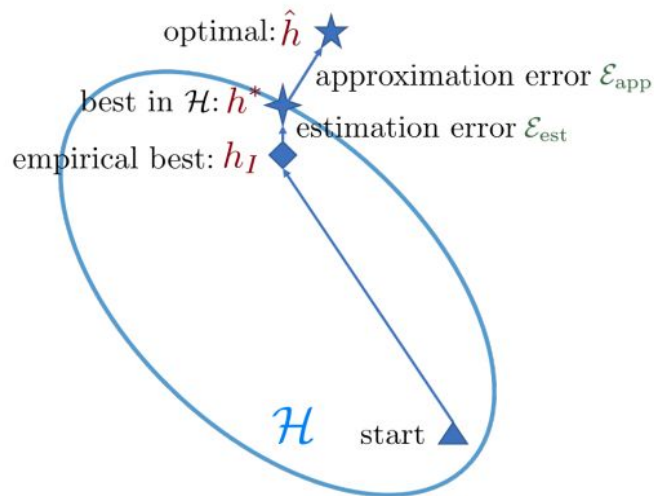
$$\mathbb{E}[R(h_I) - R(\hat{h})] = \underbrace{\mathbb{E}[R(h^*) - R(\hat{h})]}_{\mathcal{E}_{\text{app}}(\mathcal{H})} + \underbrace{\mathbb{E}[R(h_I) - R(h^*)]}_{\mathcal{E}_{\text{est}}(\mathcal{H}, I)}$$

насколько хорошо  
приближаемся к  
оптимальной гипотезе

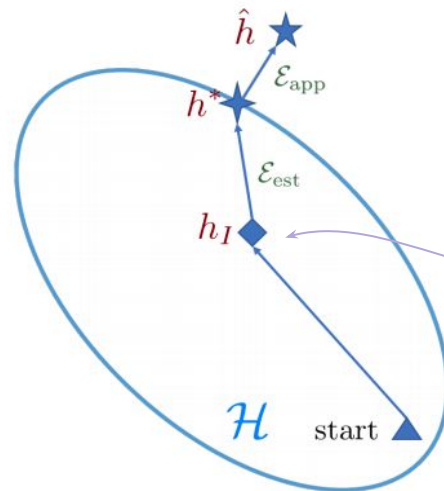


штраф за минимизацию  
эмпирического риска  
вместо ожидаемого





(a) Large  $I$ .



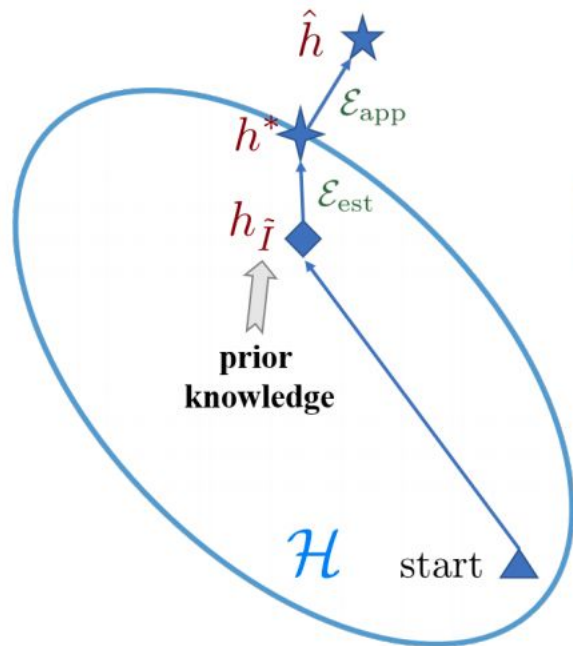
(b) Small  $I$ .

Проблема: минимизатор эмпирического риска ненадежен

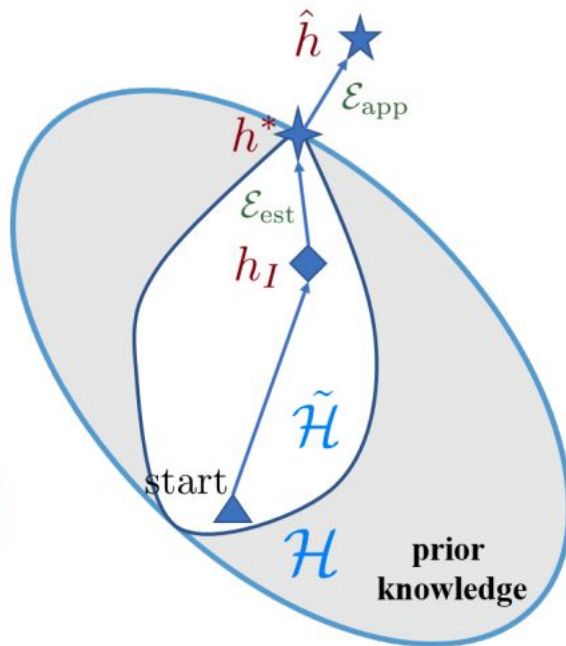
$$\mathbb{E}[R(h_I) - R(\hat{h})] = \underbrace{\mathbb{E}[R(h^*) - R(\hat{h})]}_{\mathcal{E}_{\text{app}}(\mathcal{H})} + \underbrace{\mathbb{E}[R(h_I) - R(h^*)]}_{\mathcal{E}_{\text{est}}(\mathcal{H}, I)}$$



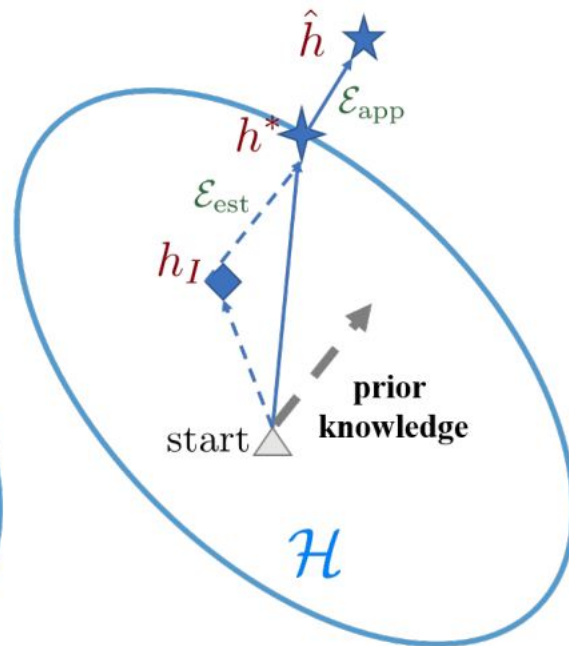
Решение: использовать априорные знания



(a) Data.

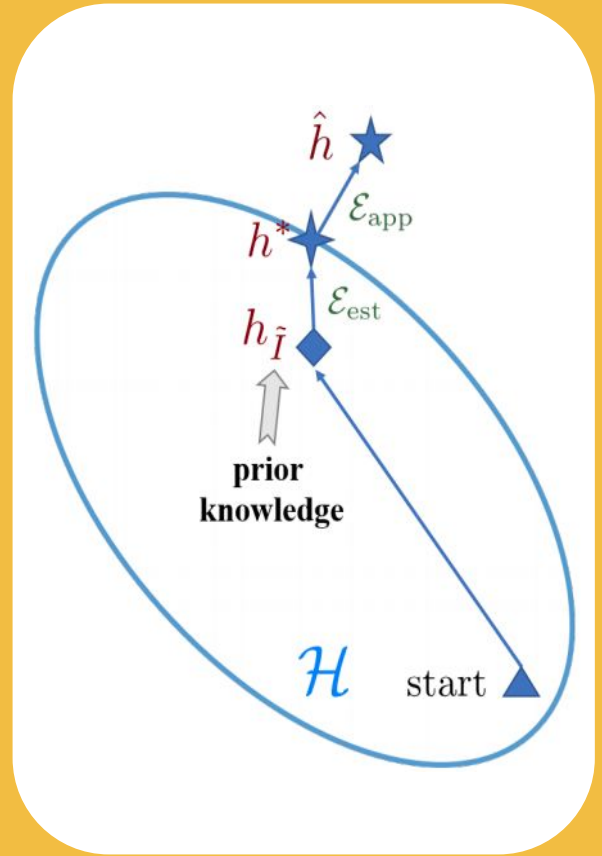


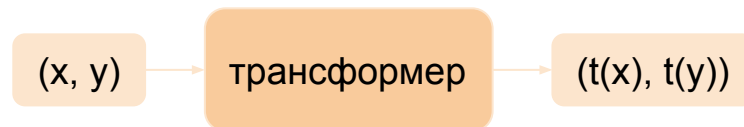
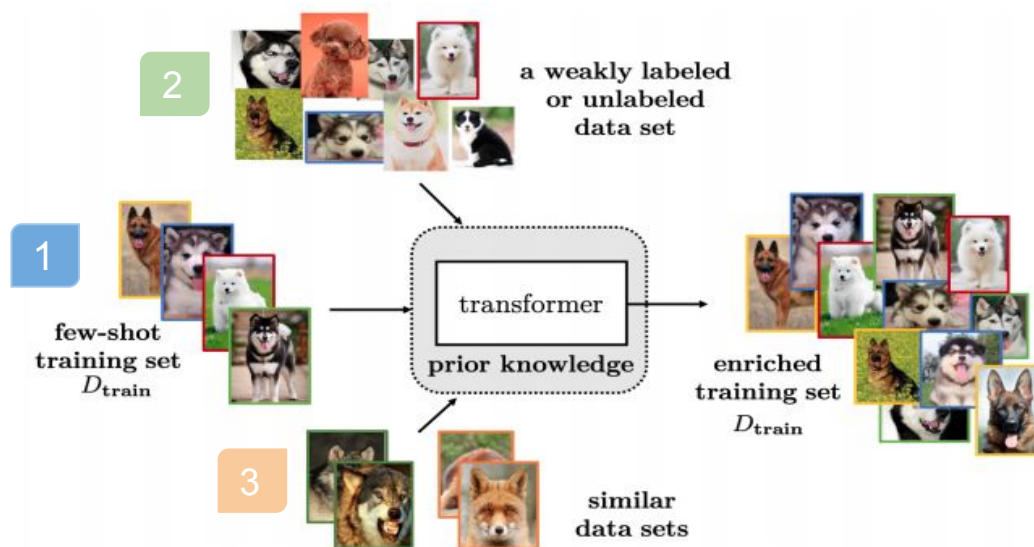
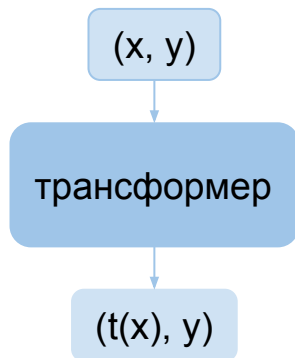
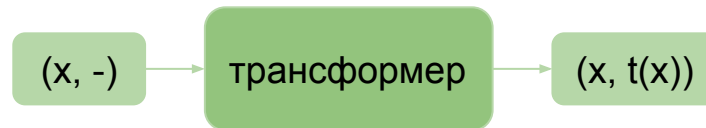
(b) Model.



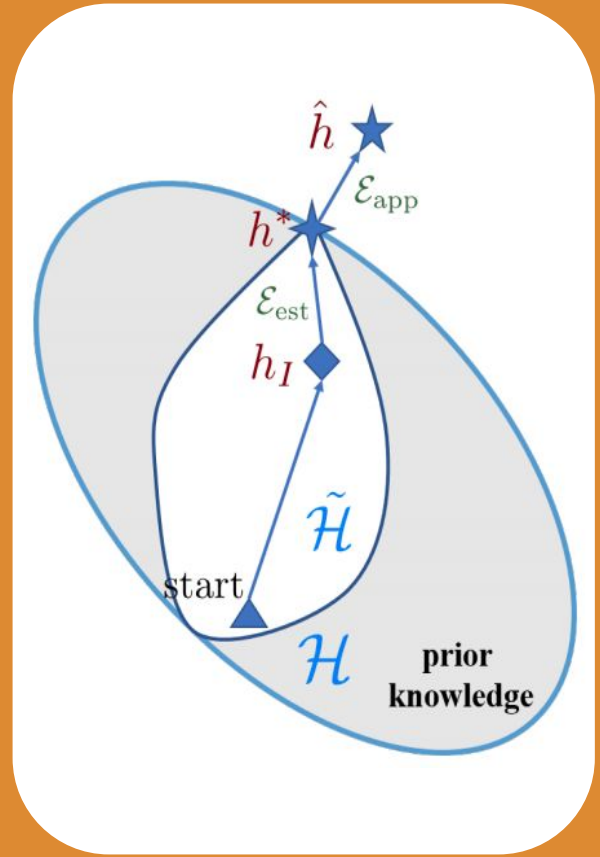
(c) Algorithm.

# Prior knowledge in Data



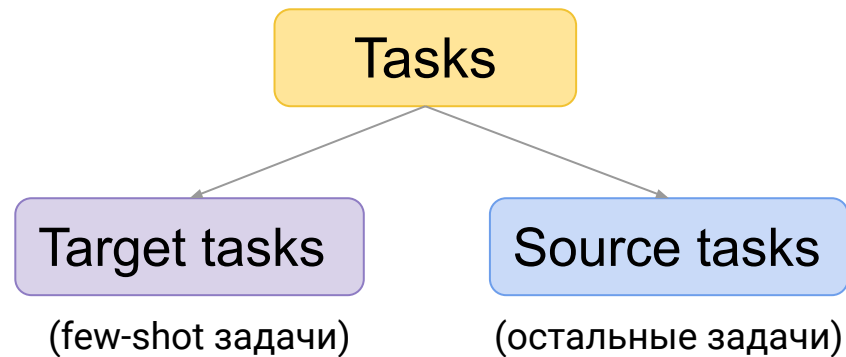
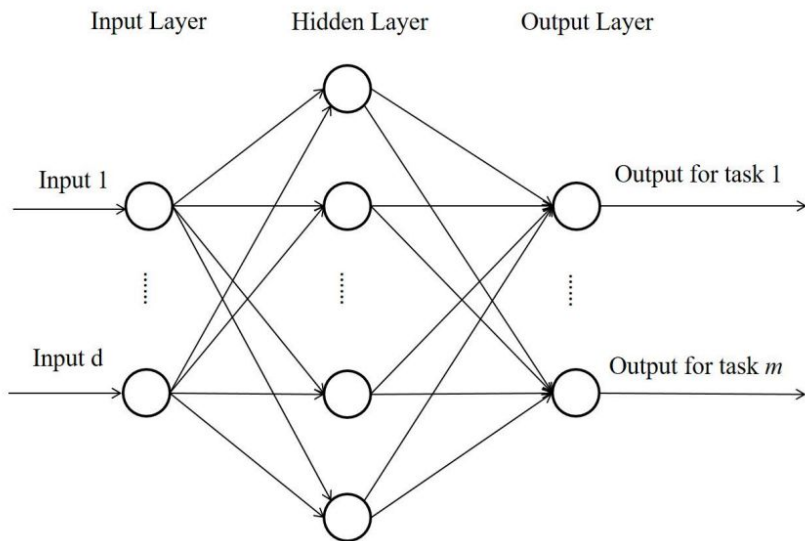


# Prior knowledge in Model



стратегия	априорные знания	как ограничить $\mathcal{H}$
multitask learning	датасеты других задач	делить/связывать параметры
embedding learning	эмбеддинги, обученные на/вместе с другими задачами	спроецировать выборку на пространство меньшей размерности
generative modeling	априорная модель, обученная на других задачах	делать ограничения на вид распределения

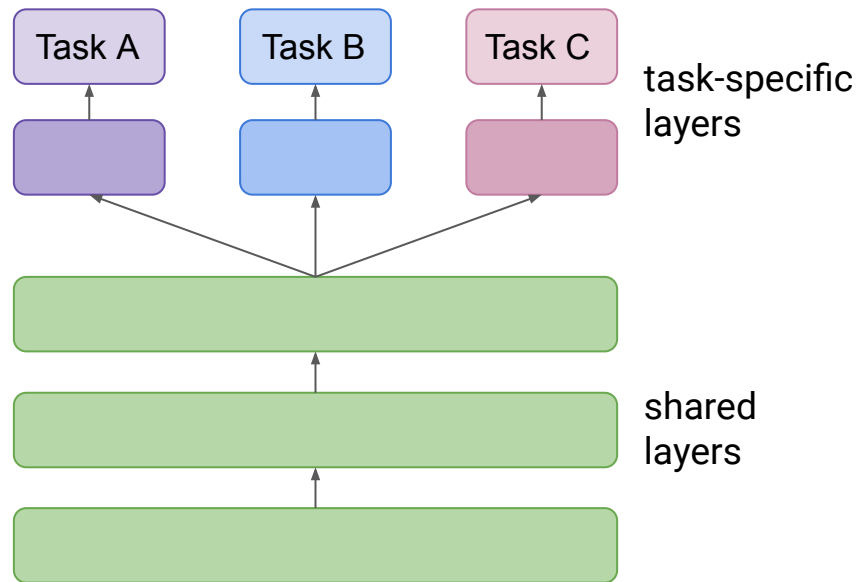
# Multitask learning



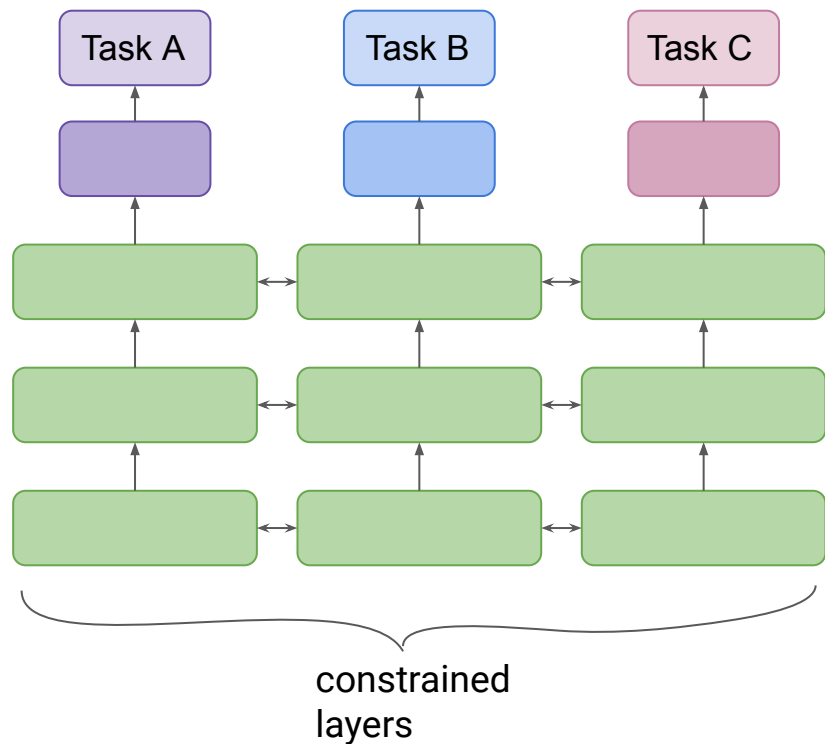
- задачи обучаются одновременно
- параметры каждой задачи ограничиваются другими задачами

## Hard parameter sharing

- Есть общие (shared) слои
- Есть личные (task-specific) слои для каждой задачи отдельно
- target задачи обновляют только task-specific слои
- source задачи обновляют task-specific и shared слои
- этот метод уменьшает риск переобучения



## Soft parameter sharing

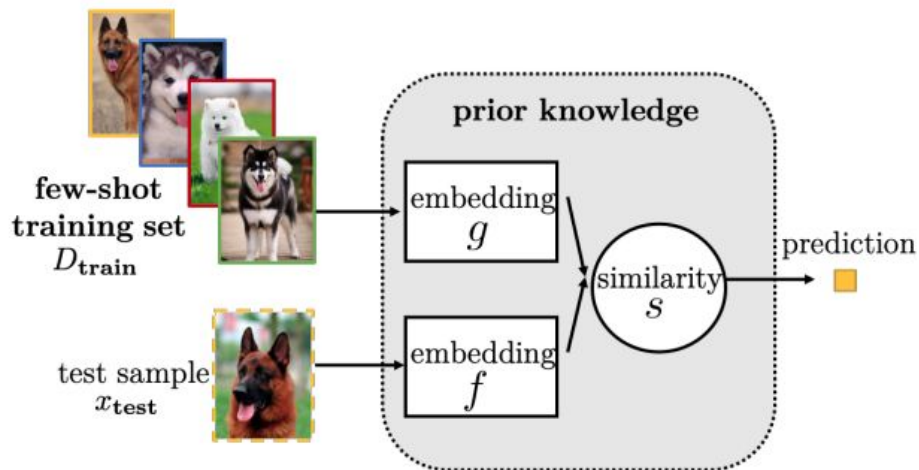


- у каждой задачи своя модель и свои параметры
- для схожести параметров регуляризуем расстояния между ними

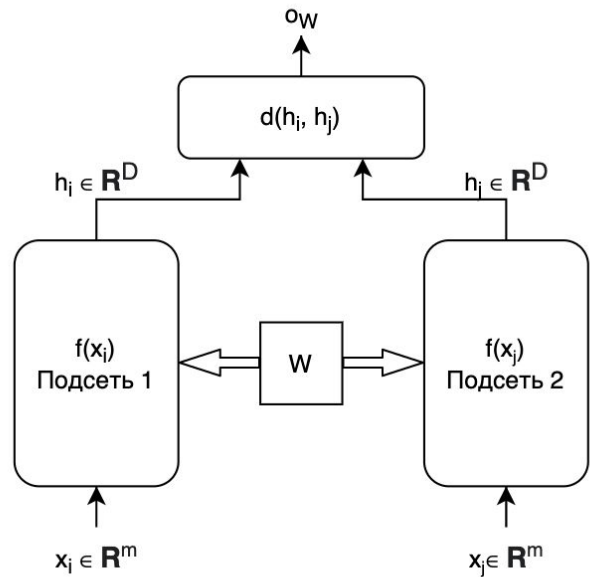


## Embedding learning

- проецируем данные на пространство меньшей размерности за счет эмбединга
- в уменьшенном пространстве гипотез похожие объекты находятся ближе, разные объекты более различимы



# Siamese networks



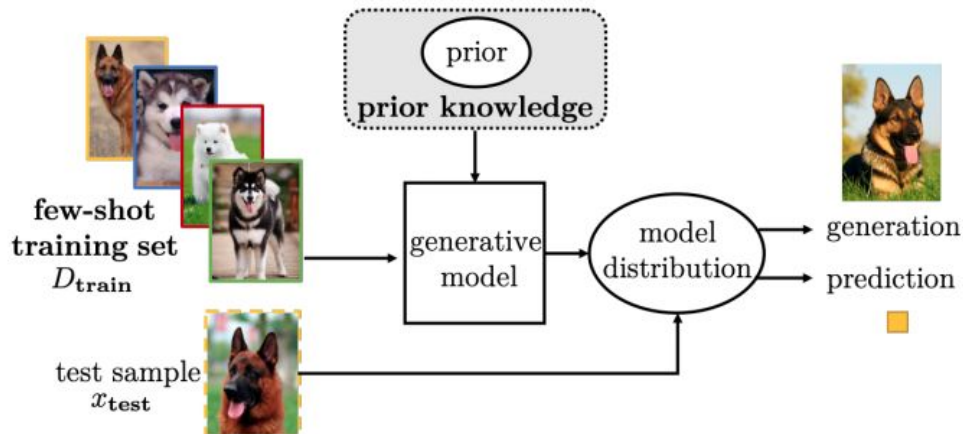
- 2 одинаковые сети
- в конце сети объединяются с помощью метрики расстояния
- веса связаны между сетями

# Generative modeling

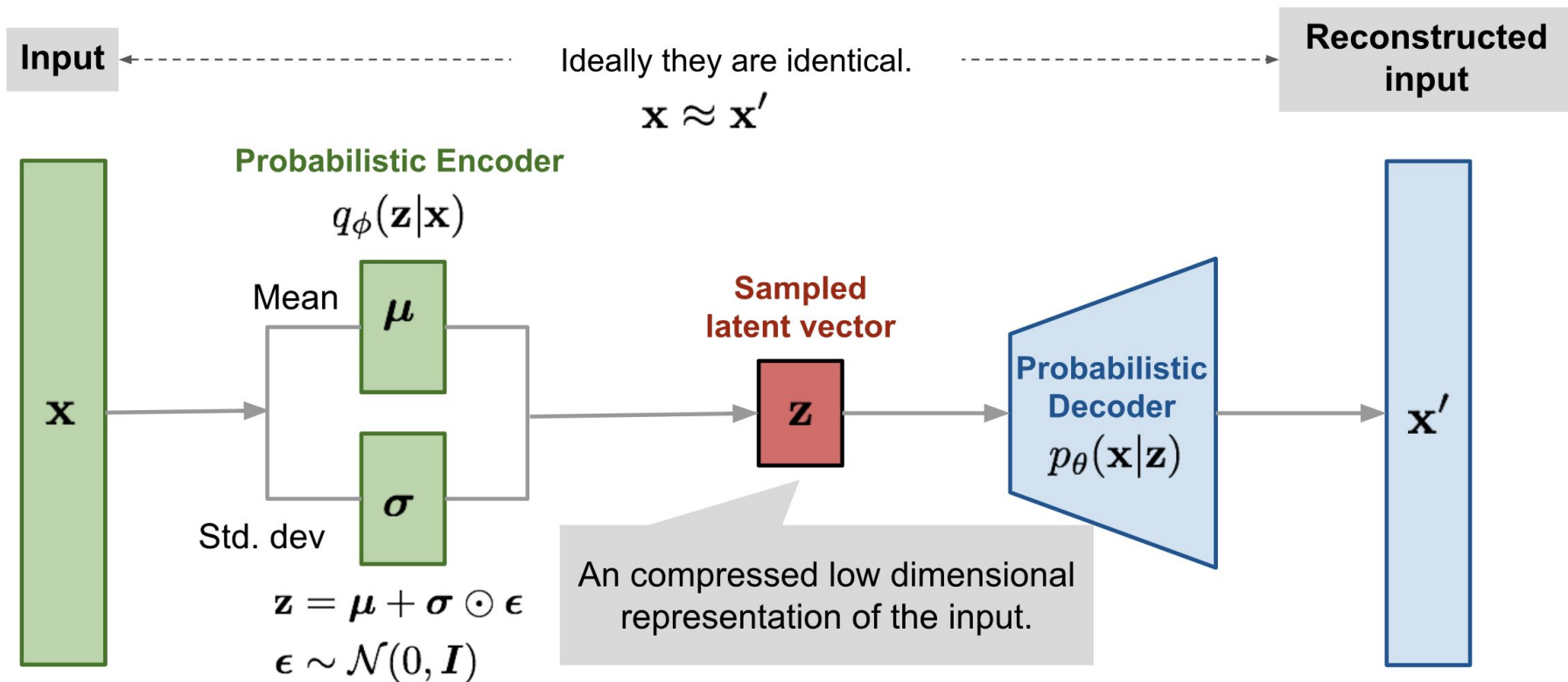
- используя априорные данные и наш датасет, хотим оценить распределение

$$x \sim \int p(x|z; \theta)p(z; \gamma)dz$$

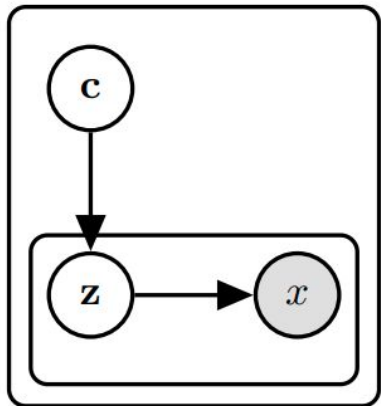
- априорные модели обучаются на других датасетах



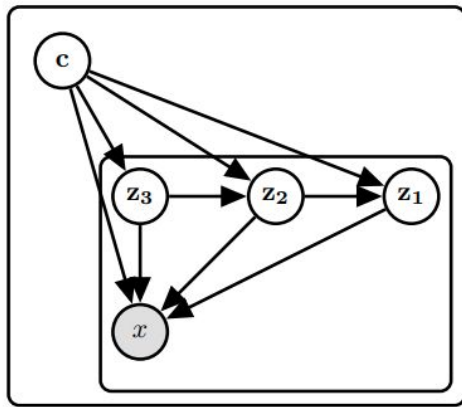
# VAE



## Neural statistician



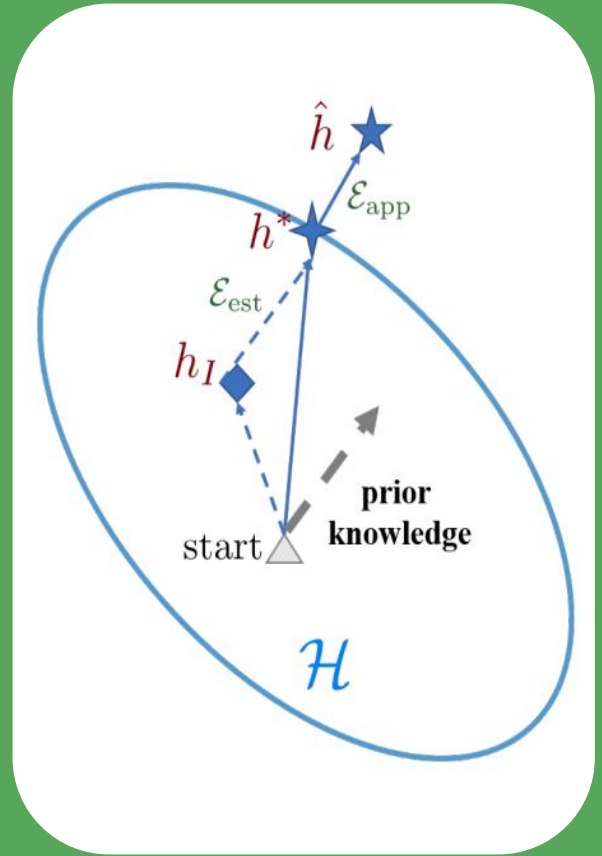
а). Базовая модель



б). Neural statistician

- есть глобальная латентная переменная  $c$  (*context*)
- каждая латентная переменная  $z$  декомпозирована на несколько слоев для работы с датасетами со сложной структурой
- при классификации объекта выбираем тот класс, чье распределение наиболее близко к распределению объекта

# Prior knowledge in Algorithm

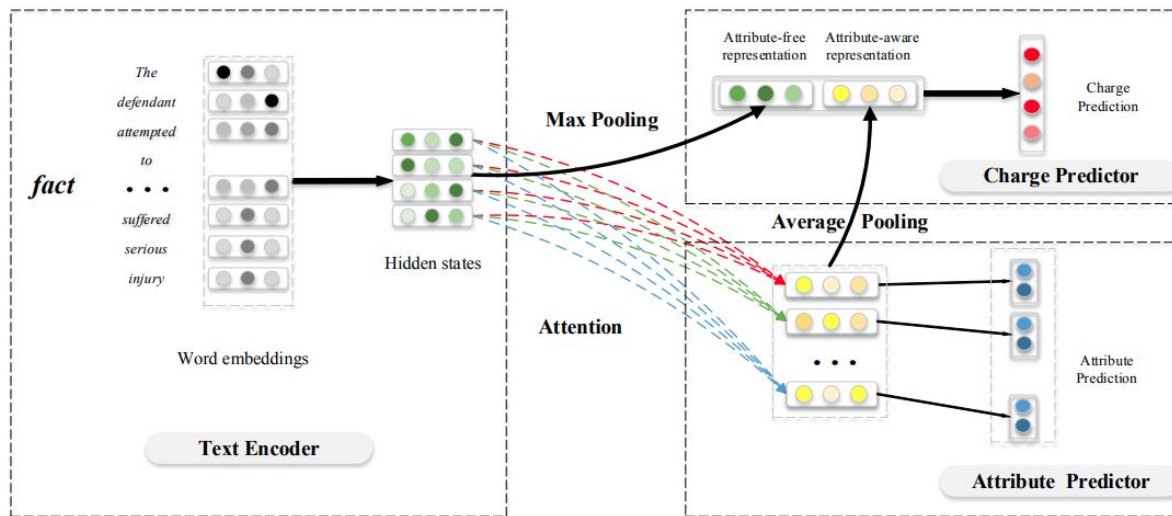


стратегия	априорные знания	как искать $\theta$ у $h^*$ в $\mathcal{H}$
улучшение существующих параметров	взять начальное приближение модели, обученной на похожей задаче	адаптировать к $\theta$ с помощью обучающего датасета
улучшение мета-параметров	использовать мета-обучатель	адаптировать к $\theta$ с помощью обучающего датасета
обучить оптимизатор	использовать мета-обучатель	использовать шаги поиска, полученные с помощью мета-обучателя

# Примеры

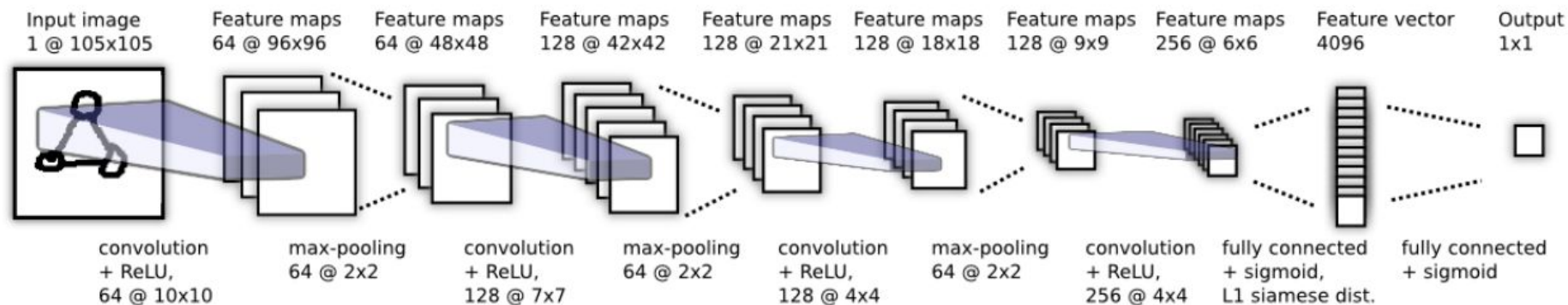


# Charge prediction (MTL)



Datasets	Criminal-S				Criminal-M				Criminal-L			
Metrics	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
TFIDF+SVM	85.8	49.7	41.9	43.5	89.6	58.8	50.1	52.1	91.8	67.5	54.1	57.5
CNN	91.9	50.5	44.9	46.1	93.5	57.6	48.1	50.5	93.9	66.0	50.3	54.7
CNN-200	92.6	51.1	46.3	47.3	92.8	56.2	50.0	50.8	94.1	61.9	50.0	53.1
LSTM	93.5	59.4	58.6	57.3	94.7	65.8	63.0	62.6	95.5	69.8	67.0	66.8
LSTM-200	92.7	60.0	58.4	57.0	94.4	66.5	62.4	62.7	95.1	72.8	66.7	67.9
Fact-Law Att.	92.8	57.0	53.9	53.4	94.7	66.7	60.4	61.8	95.7	73.3	67.1	68.6
<b>Our Model</b>	<b>93.4</b>	<b>66.7</b>	<b>69.2</b>	<b>64.9</b>	<b>94.4</b>	<b>68.3</b>	<b>69.2</b>	<b>67.1</b>	<b>95.8</b>	<b>75.8</b>	<b>73.7</b>	<b>73.1</b>

# Image recognition (siamese nets)



Method	Test
<b>Humans</b>	95.5
<b>Hierarchical Bayesian Program Learning</b>	95.2
<b>Affine model</b>	81.8
<b>Hierarchical Deep</b>	65.2
<b>Deep Boltzmann Machine</b>	62.0
<b>Simple Stroke</b>	35.2
<b>1-Nearest Neighbor</b>	21.7
<b>Siamese Neural Net</b>	58.3
<b>Convolutional Siamese Net</b>	92.0

# Video object segmentation



		Semi-Supervised								Unsupervised							Bounds		
Measure		Ours	OFL	BVS	FCP	JMP	HVS	SEA	TSP	FST	NLC	MSG	KEY	CVOS	TRC	SAL	COB SP	COB	MCG
$\mathcal{J}$	Mean $\mathcal{M} \uparrow$	<b>79.8</b>	68.0	60.0	58.4	57.0	54.6	50.4	31.9	<b>55.8</b>	55.1	53.3	49.8	48.2	47.3	39.3	<b>86.5</b>	79.3	70.7
	Recall $\mathcal{O} \uparrow$	<b>93.6</b>	75.6	66.9	71.5	62.6	61.4	53.1	30.0	<b>64.9</b>	55.8	61.6	59.1	54.0	49.3	30.0	<b>96.5</b>	94.4	91.7
	Decay $\mathcal{D} \downarrow$	14.9	26.4	28.9	<b>-2.0</b>	39.4	23.6	36.4	38.1	<b>0.0</b>	12.6	2.4	14.1	10.5	8.3	6.9	2.8	3.2	<b>1.3</b>
$\mathcal{F}$	Mean $\mathcal{M} \uparrow$	<b>80.6</b>	63.4	58.8	49.2	53.1	52.9	48.0	29.7	51.1	<b>52.3</b>	50.8	42.7	44.7	44.1	34.4	<b>87.1</b>	75.7	62.9
	Recall $\mathcal{O} \uparrow$	<b>92.6</b>	70.4	67.9	49.5	54.2	61.0	46.3	23.0	51.6	51.9	<b>60.0</b>	37.5	52.6	43.6	15.4	<b>92.4</b>	88.5	76.7
	Decay $\mathcal{D} \downarrow$	15.0	27.2	21.3	<b>-1.1</b>	38.4	22.7	34.5	35.7	<b>2.9</b>	11.4	5.1	10.6	11.7	12.9	4.3	2.3	3.9	<b>1.9</b>
$\mathcal{T}$	Mean $\mathcal{M} \downarrow$	37.6	21.7	34.5	29.6	15.3	35.0	<b>14.9</b>	41.2	34.3	41.4	29.1	25.2	<b>24.4</b>	37.6	64.1	<b>27.4</b>	44.1	69.8

## References

- <https://arxiv.org/pdf/1904.05046.pdf> - a survey on FSL
- <https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf> - Siamese Networks for One-shot Learning
- <https://arxiv.org/pdf/1606.02185.pdf> - Neural Statistician
- <https://www.aclweb.org/anthology/C18-1041.pdf> - Charge prediction
- [https://openaccess.thecvf.com/content\\_cvpr\\_2017/papers/Caelles\\_One-Shot\\_Video\\_Object\\_CVPR\\_2017\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2017/papers/Caelles_One-Shot_Video_Object_CVPR_2017_paper.pdf) - Video object segmentation