Глубинное обучение
Лекция 11: Недифференцируемые нейросети

Лектор: Антон Осокин

ФКН ВШЭ, 2020

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Недифференцируемые модели?

- Часто недифференцируемые функции – backpropable (“нейро-дифференцируемые“)
  – Примеры: max, ReLu, медиана
- Совсем-недифференцируемые функции
  – Кусочно-постоянные функции
    - argmax
    - f(x) = 0 if x <0 else 1
  – Сложные индексы
    - Позиции прямоугольников на изображении
  – Ответы внешних систем:
    - Программа, среда, человек

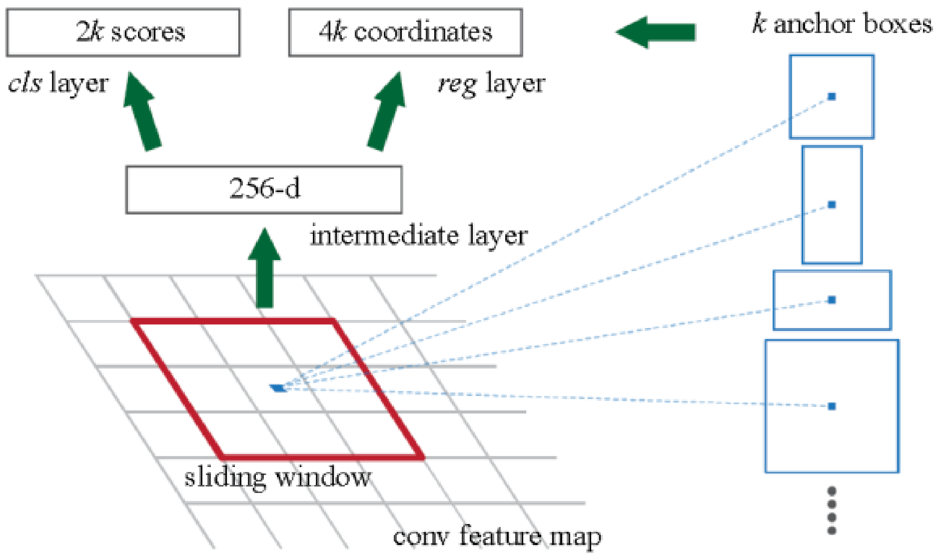
Что делать?

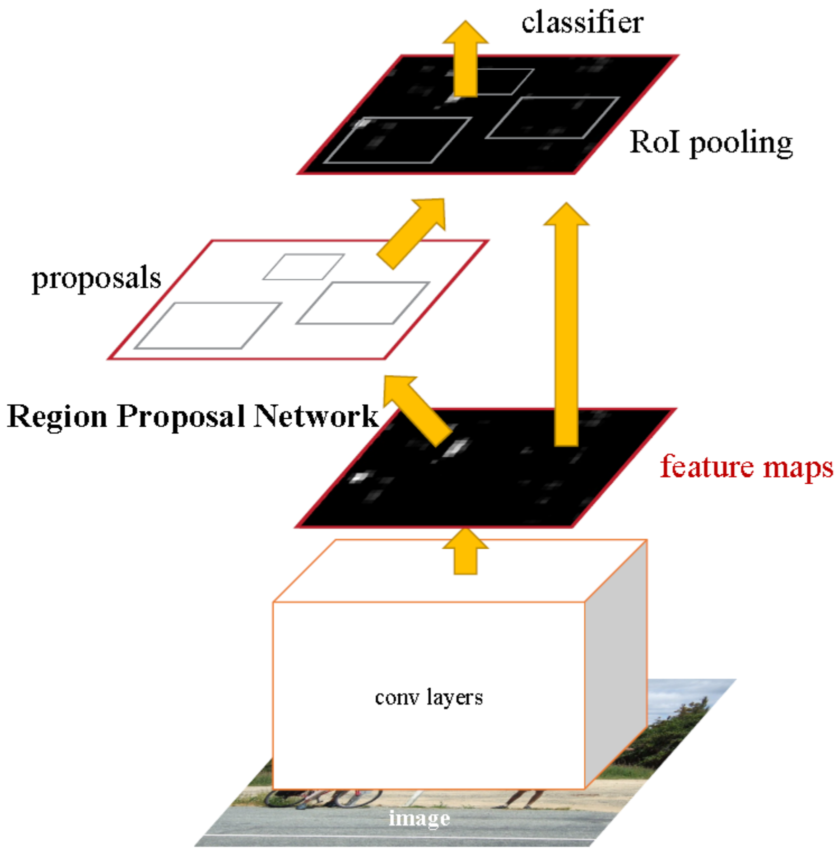
- Игнорировать ☺
  - max, комбинаторный пулинг
  - координаты пропоузалов в детекторе Faster R-CNN
- Сглаживать через стохастичность
  - Стохастические активации
  - Жёсткое внимание
  - Deep RL
- Другие способы сглаживания
  - Spatial transformer

Детектор Faster R-CNN

[Ren et al., 2015]

- Одна сеть выдаёт гипотезы объектов (proposals)
- Вторая сеть классифицирует гипотезы
- RoI pooling – недифференцируемый по координатам
- Игнорируем это!





Сглаживание через стохастичность

- Что это такое?
  - $\theta(x)$ – дифференцируемая функция
  - Распределение $p(w \mid \theta(x))$
  - Лосс $L(w)$ – (не) дифференцируемый
- Проход вперёд:
  - Вычисление $\theta(x)$ – параметры распределения
  - Сэмплирование w из $p(w \mid \theta(x))$
  - Вычисление $L(w)$
- Функция $\mathbb{E}_{w \sim p(w|\theta(x))} L(w)$ – часто дифференцируемая по θ и x
- Градиент получается из log-derivative trick

$$\nabla_\theta = \mathbb{E}_{w \sim p(w|\theta)} \nabla_\theta [\log p(w|\theta(x))] L(w) \qquad \nabla_\theta [\log p(w|\theta)] = \frac{\nabla_\theta [p(w|\theta)]}{p(w|\theta)}$$

- Большая дисперсия!

Что дают добавление стохастичности?

- Вероятностные модели относительно <u>параметров</u> сети
  - Например, прореживание сетей [Molchanov et al., 2017]

- Моделирует неопределённость

- Позволяют бороться с недифференцируемостью

Обучение: дифференцируемый лосс

- Простой случай:
  – Лосс $L(w)$ – дифференцируемый
  – Распределение $p(w \mid \theta)$ – «хорошее»

- Репараметризация (если возможна) – самое лучшее решение!
  – Разделение случайности и параметров
  – Представим распределение $p(w \mid \theta)$ как $g(\theta, \varepsilon),\ \varepsilon \sim r(\varepsilon)$

    $$z = \mu_\theta(x) + \sigma_\theta(x)\varepsilon,\ \varepsilon \sim r(\varepsilon)$$

    g – детерминированная функция
    ε – шум
  – Тогда градиент легко оценить:

$$\nabla_\theta = \nabla_\theta \int p(w|\theta)L(w)dw = \int r(\varepsilon)\nabla_\theta L(g(\theta,\varepsilon))d\varepsilon$$

  – Дисперсия градиента сильно уменьшается

Какие распределения можно репараметризовать?

$p(x\|y)$	$r(\epsilon)$	$g(\epsilon, y)$
$\mathcal{N}(x\|\mu, \sigma^2)$	$\mathcal{N}(\epsilon\|0, 1)$	$x = \sigma\epsilon + \mu$
$\mathcal{G}(x\|1, \beta)$	$\mathcal{G}(\epsilon\|1, 1)$	$x = \beta\epsilon$
$\mathcal{E}(x\|\lambda)$	$\mathcal{U}(\epsilon\|0, 1)$	$x = -\frac{\log \epsilon}{\lambda}$
$\mathcal{N}(x\|\mu, \Sigma)$	$\mathcal{N}(\epsilon\|0, I)$	$x = A\epsilon + \mu, \text{where } AA^T = \Sigma$

Slide credit: Dmitry Vetrov

Подробный обзор: [Mohamed et al., 2019; arXiv:1906.10652]

Нельзя репараметризовать дискретные распределения!

- Категориальное распределение $z \sim \mathrm{Discrete}(\alpha_1, \dots, \alpha_L)$
- Надо для argmax!

$$z = (0, 1, 0, \dots, 0)$$

- Релаксация: Gumbel-Softmax [Jang et al., 2017; Maddison et al., 2017]

$$(z_1, \dots, z_L) \sim \mathrm{RelaxedDiscrete}(\alpha_1, \dots, \alpha_L | T)$$

$$z_i = \frac{\exp((\log \alpha_i + G_i)/T)}{\sum_{j=1}^{L} \exp((\log \alpha_j + G_j)/T)}, \;\; G_k \sim \mathrm{Gumbel}$$

$$G_k = -\log(-\log u_k), \;\; u_k \sim \mathrm{Uniform}[0, 1]$$

- $\mathrm{RelaxedDiscrete}(\alpha_1, \dots, \alpha_L | T) \xrightarrow{T \to 0} \mathrm{Discrete}(\alpha_1, \dots, \alpha_L)$

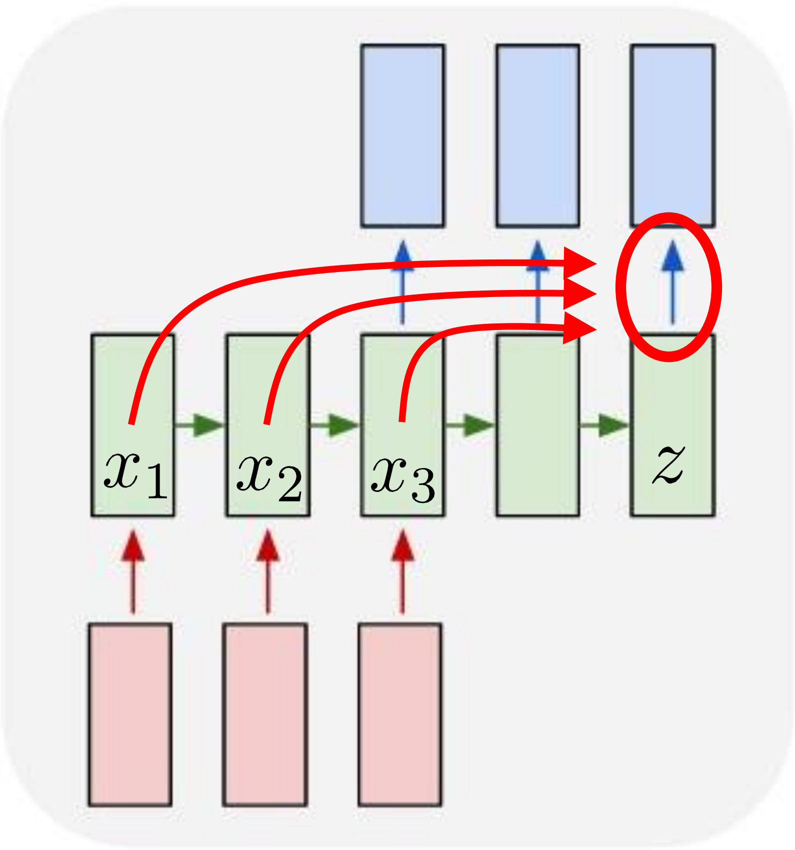
- Трюк: Straight-through estimator [Hinton et al., 2015 course] [Bengio et al. (2013)]

$$\nabla_\alpha \approx \nabla_z$$

Градиент 1 в выбранную позицию, остальные 0

Механизмы внимания: мягкий и жёсткий

Модель seq2seq с вниманием [Bahdanau et al. , 2015]



- Релевантность

$$s_i := \mathrm{score}(x_i, z) = x_i^T z$$

- Веса

$$a_1, a_2, \dots := \mathrm{softmax}(s_1, s_2, \dots)$$

Мягкое внимание

- Контекст: $c := \sum_i a_i x_i$

Жёсткое внимание

- Сл. величина: $i \sim \mathrm{Discrete}(a_1, a_2, \dots)$
- Контекст: $c := x_i$

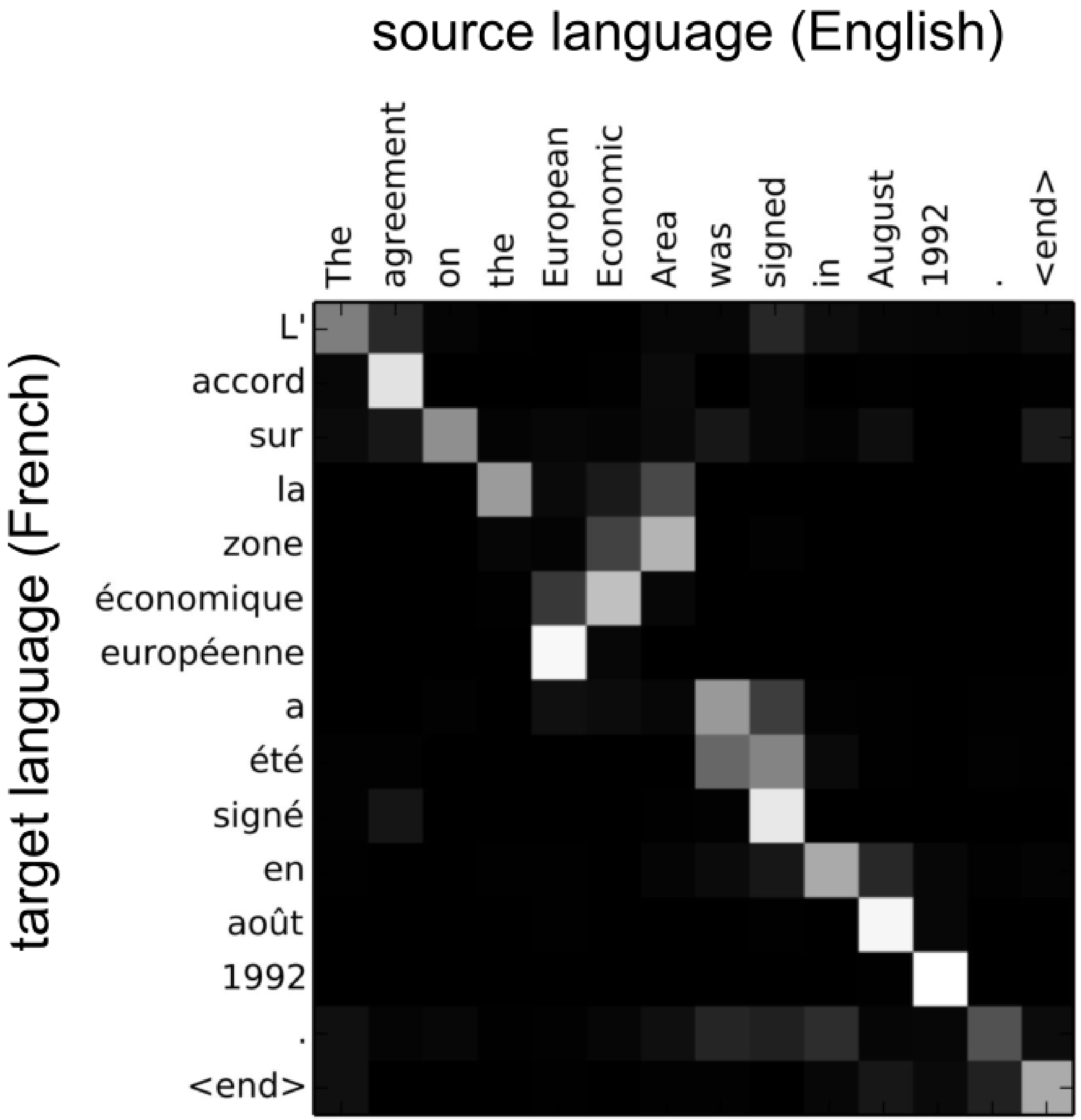
image credit: Andrej Karpathy

Мягкое или жёсткое внимание?

- Мягкое внимание легче обучать
  - Жёсткое обычно не удаётся обучить до уровня мягкого
  - В машинном переводе используют только мягкое
- Жёсткое внимание эффективнее на тесте – не надо складывать
- Жёсткое внимание позволяет выбирать из разнородных элементов
- Жёсткое внимание, например, может управлять сбором данных

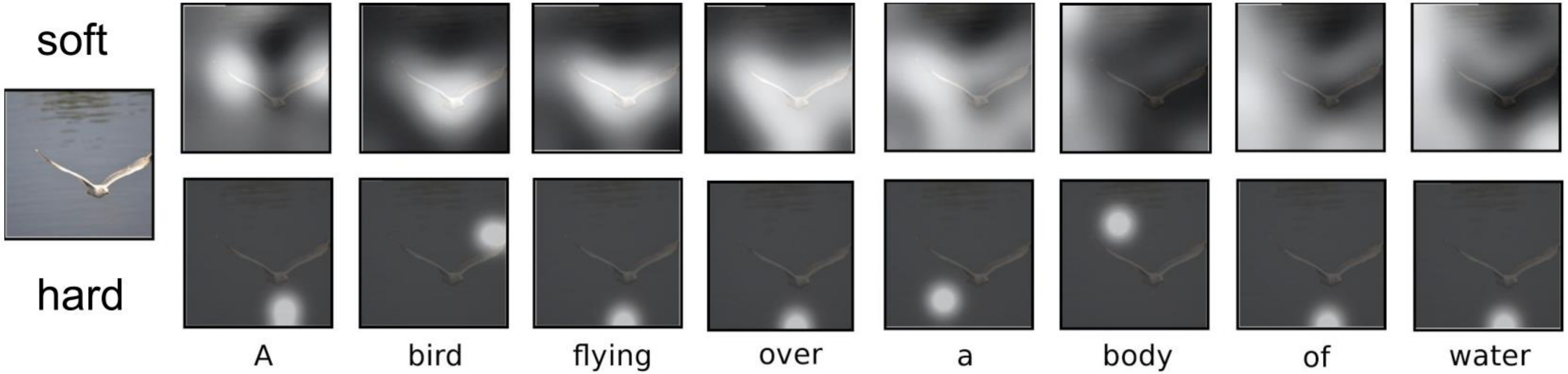
Внимание в машинном переводе

[Bahdanau et al. , 2015]



Внимание в генерации подписей

[Xu et al. , 2015]



Недифференцируемый лосс? Что делать?

- Модель:
  - $\theta(x)$ – дифференцируемая функция
  - Распределение $p(w \mid \theta(x))$
  - Лосс $L(w)$ недифференцируемый
- Обучение с подкреплением ☹
  - Политика $p(w \mid \theta(x))$
  - Награда-reward: $-L(w)$
  - Итеративное принятие решений
- Log-derivative trick лежит в основе REINFORCE, policy gradients

$$\nabla_\theta = \mathbb{E}_{w \sim p(w|\theta)} \nabla_\theta [\log p(w|\theta(x))] L(w)$$

  - Борьба с дисперсией! (всеми средствами)

Что делать с дисперсией?

- Модель:
  - θ(*x*) – дифференцируемая функция
  - Распределение *p*(*w* | θ(*x*))
  - Лосс *L*(*w*) недифференцируемый
- Градиент: $\nabla_\theta \approx \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta[\log p(w_i|\theta(x))]L(w_i)$
- Идея: baseline $\nabla_\theta \approx \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta[\log p(w_i|\theta(x))]\Big(L(w_i) - b\Big)$
  - Почему?

$$\int p(w|\theta)\nabla_\theta[\log p(w|\theta(x))]b\, dw = \int \nabla_\theta p(w|\theta) b\, dw = b\nabla_\theta \int p(w|\theta)\, dw = 0$$

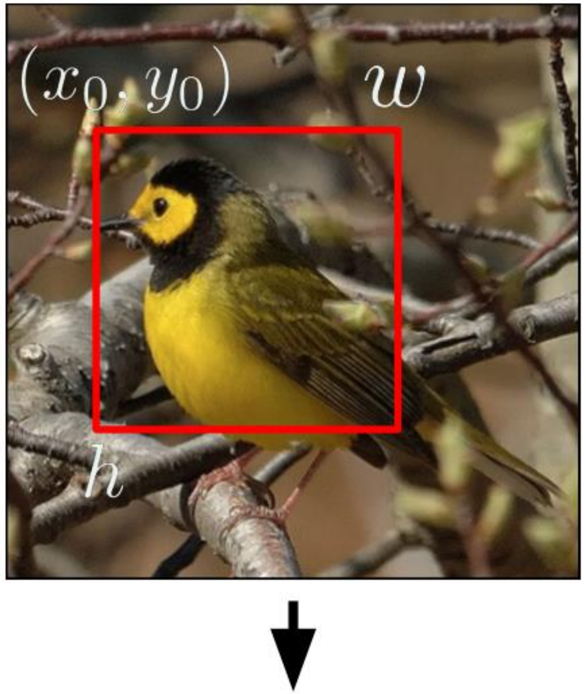
- Идея: дифференцируемый бейзлайн, зависящий от *w*
  - Компенсировать смещение репараметризацией бейзлайна

REBAR, RELAX

Другие способы сглаживания: Spatial Transformer

[Jaderberg et al., 2015]

- Параметрическая модель фрагмента (x_0, y_0, w, h)
- Сеть по картинке выдает параметры
- Нужно вырезать патч
- Можно обучать REINFORCE (но работает не очень) [Mnih et al. , 2014]



- **Spatial Transformer** – лучше
  - **Идея**: билинейная интерполяция дифференцируема
  - «Локальное внимание»
  - Расширение области определения

Slide credit: Michael Figurnov

Дифференцируемая интерполяция

[Jaderberg et al., 2015]

- Билинейная интерполяция вычисляет цвет в нецелой точке (*x*, *y*)
- *U* – картинка, *v* – значение в (*x*, *y*)
- Основное наблюдение:

$$v = \sum_{i=1}^{H} \sum_{j=1}^{W} U_{ij} \max(0, 1 - |x - i|) \max(0, 1 - |y - j|)$$

- Градиенты

$$\frac{dv}{dU_{ij}} = \max(0, 1 - |x - i|) \max(0, 1 - |y - j|)$$

$$\frac{dv}{dx} = \sum_{i=1}^{H} \sum_{j=1}^{W} U_{ij} \max(0, 1 - |y - j|) \begin{cases} 0, & \text{if } |x - i| \geq 1 \\ 1, & \text{if } x < i \\ -1, & \text{if } x \geq i \end{cases}$$

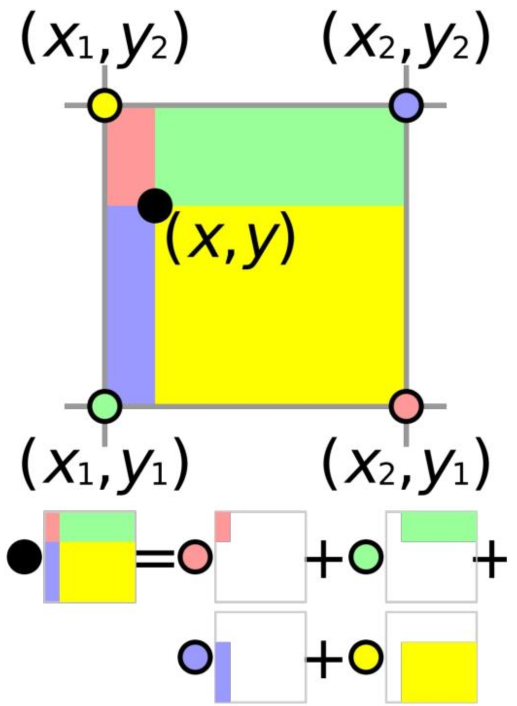
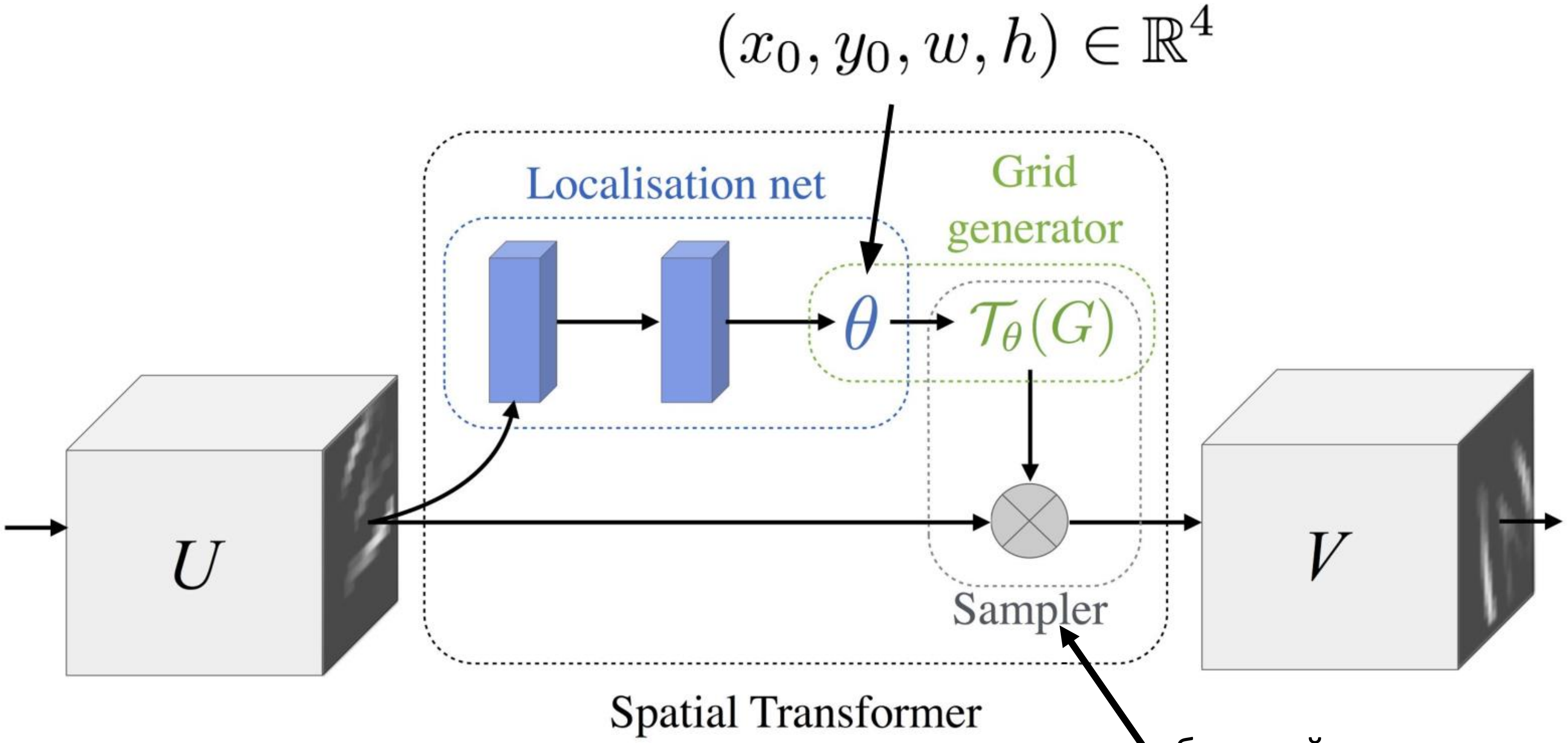


Image credit: wikipedia

Slide credit: Michael Figurnov

Другие способы сглаживания: Spatial Transformer

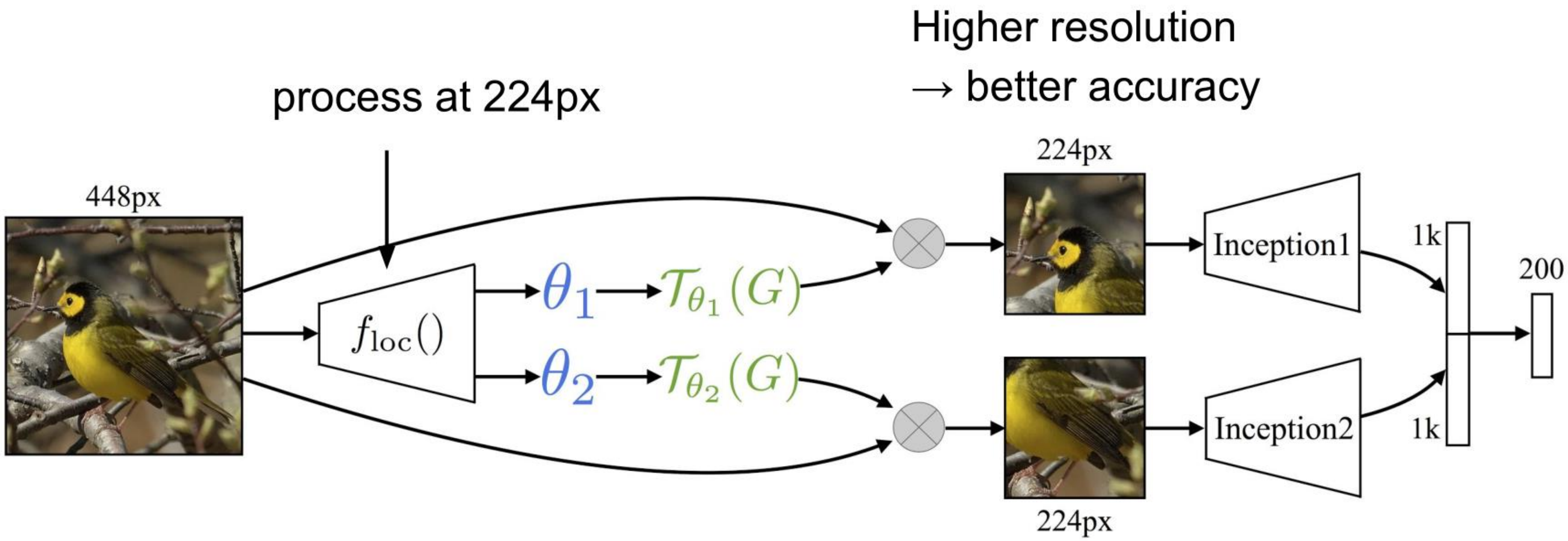
[Jaderberg et al., 2015]



билинейная
интерполяция

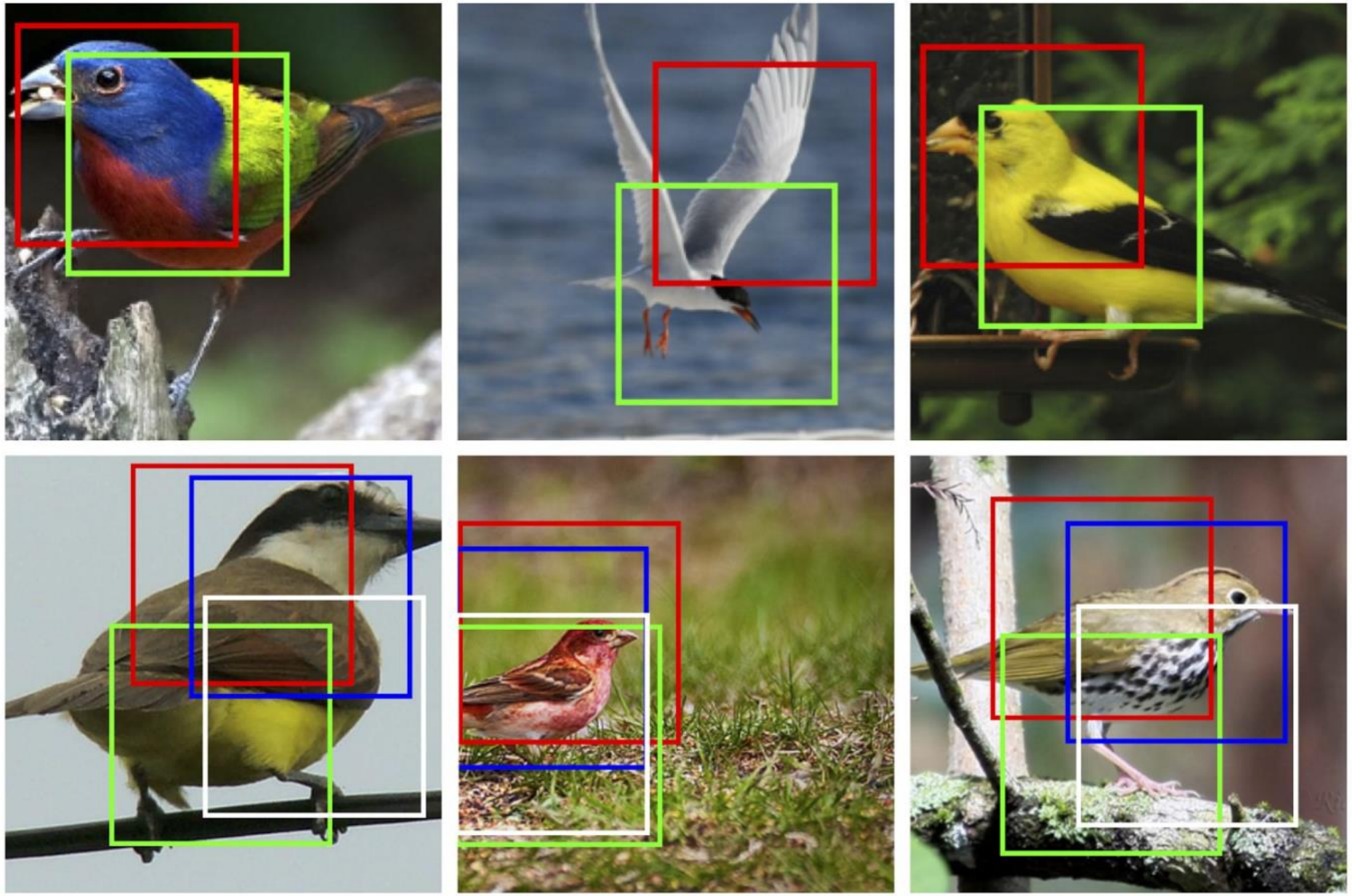
Spatial Transformer для классификации птиц

[Jaderberg et al., 2015]



Spatial Transformer для классификации птиц

[Jaderberg et al., 2015]



Заключение

- С недифференцируемыми функциями можно бороться
  - Можно попробовать проигнорировать
  - Основной способ – сведение к дифференцируемым

- Сглаживание при помощи стоастичности
  - Репараметризация – хорошо работает

- Совсем недифференцируемо – RL
  - Все сложно, нестабильно, долго, но иногда возможно

- Есть и другие способы сглаживания
- Если можно промоделировать более простой моделью, то лучше это делать