Глубинное обучение
Лекция 10: Дифференцируемое программирование

Лектор: Антон Осокин

ФКН ВШЭ, 2020

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

План лекции

- Мотивация: зачем комбинировать алгоритмы и нейросети?

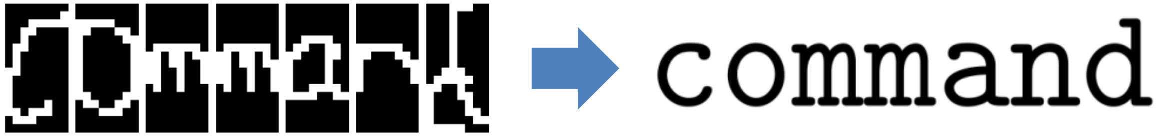
- Способы комбинирования:
  – Структурный пулинг = комбинаторная оптимизация
  – Итерации алгоритмов = слои нейросетей
  – Дифференцирование по входу алгоритма
    - Дифференцирование неявных слоев

Алгоритмы в нейросетях, зачем?

- У нейросетей очень сложные структуры
- Структуры сложно создавать для новых задач
- Часто нейросеть не может «выучить всё», надо помогать
- Комбинировать существующие решения и нейросети
- Существуют очень мощные алгоритмы для сложных задач

Задача: распознавание рукописных символы

- Модельная задача – распознавание рукописных символов

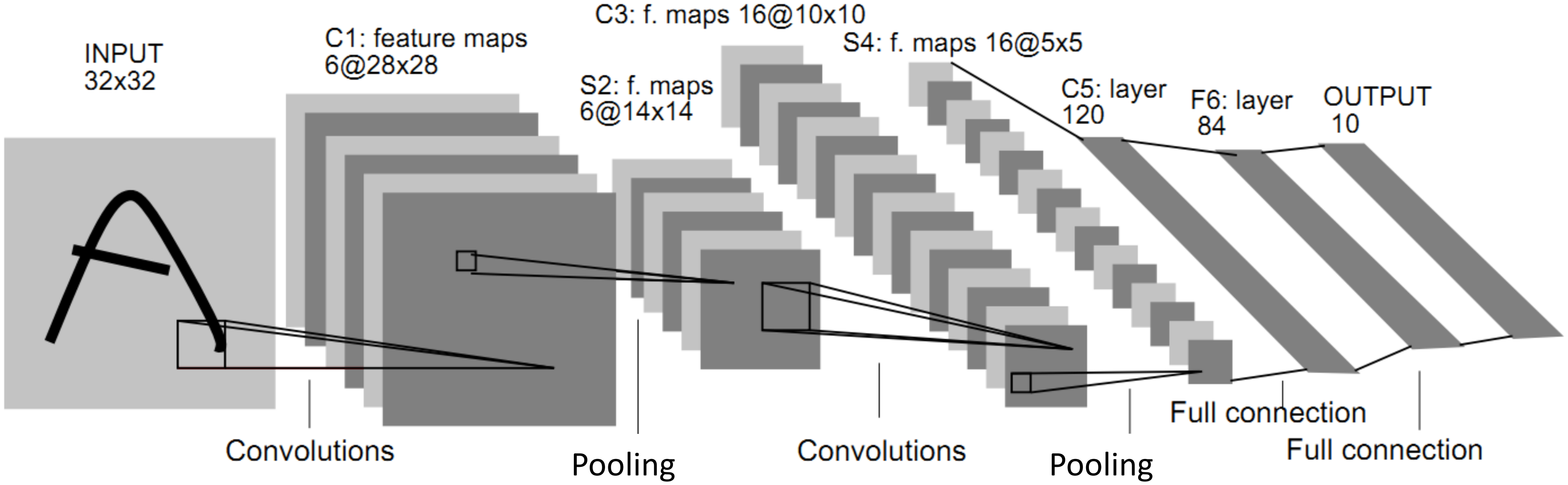


- A MNIST для structured prediction (предсказания объектов со структурой)
    - Простая задача => очень много методов применимо
    - Классификация по отдельности: 8% ошибка
    - Учёт последовательности: 1-3% ошибка

- Стандартные алгоритмы:
  динамическое программирование, СЛАУ

Структура нейросети

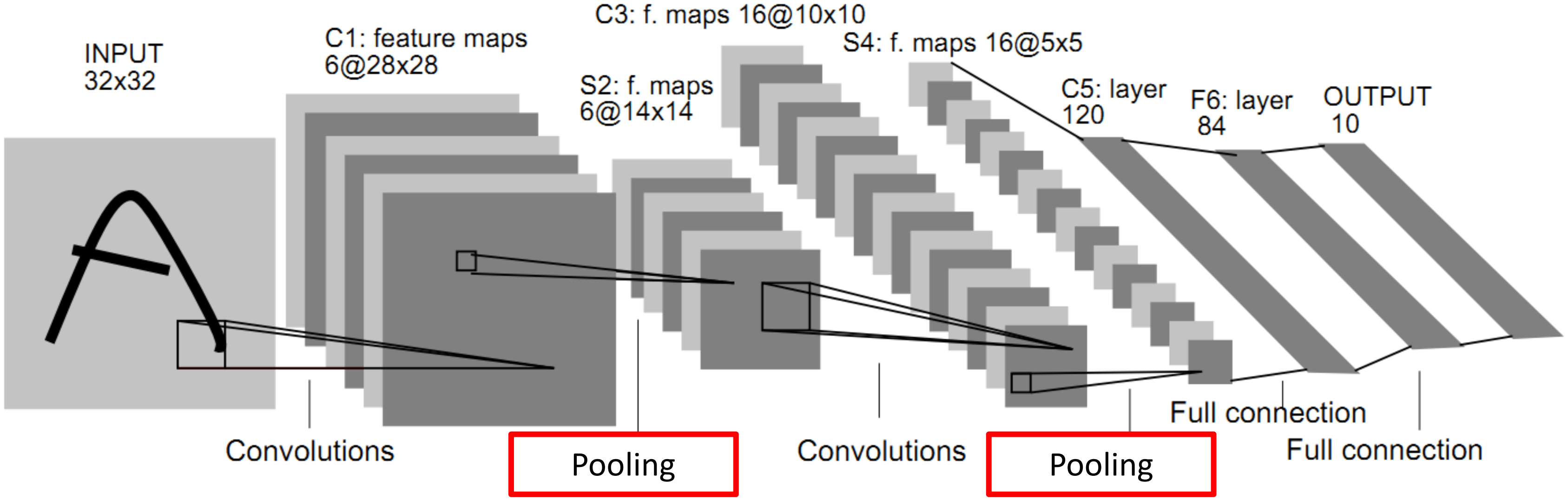
- Нейросеть для распознавания одного символа:



- Линейные операции с параметрами
  - Свёртки для изображений
- Нелинейность (sigmoid, ReLu)
- Пулинг для понижения размерности и инвариантности
- Обучение = стохастическая оптимизация

Пулинг для выбора активаций

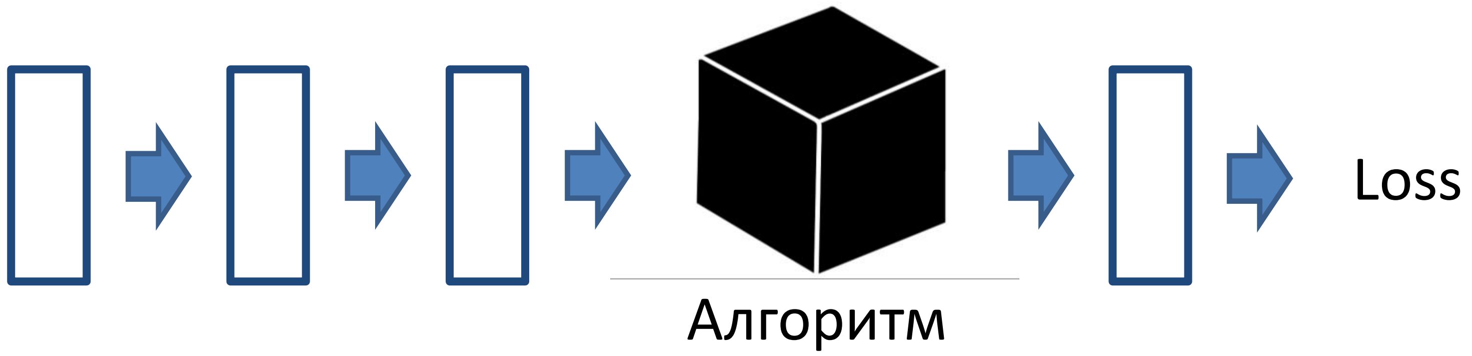
- Нейросеть для распознавания одного символа:



- Пулинг – процедура агрегирования активаций (max, sum)
- Алгоритмы для пулинга: квантиль, сортировка
- И это дифференцируемо?
  - «дифференцируемо» в смысле нейросетей

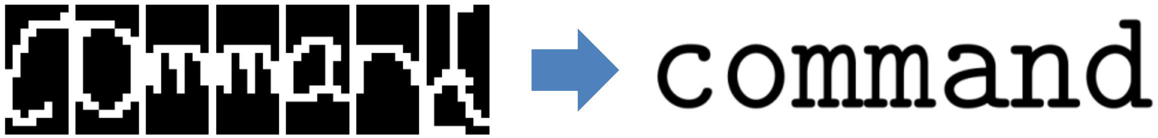
Структурный пулинг = комбинаторная оптимизация

Нейросеть: проход вперёд



Для прохода назад нужен только результат алгоритма

Conditional Random Field (CRF)

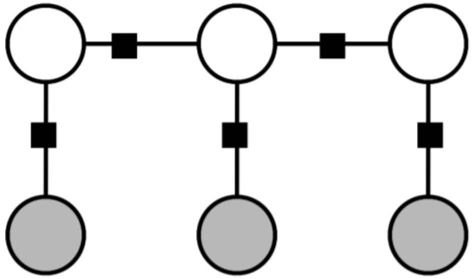


- CRF – способ учесть связи между символами
- Связи задаются функцией *F*, связывающей метки

$$F(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) = \sum_{i=1}^{T} \theta_i(y_i \mid x_i) + \sum_{i=1}^{T-1} \theta_{i,i+1}(y_i, y_{i+1})$$

- Здесь ***y*** – метки символов, ***x*** – изображения символов
  θ - потенциалы (унарные и парные)

- Графическая модель (фактор-граф)



- Унарные потенциалы вычисляются нейросетью

CRF: наилучшая конфигурация

- CRF – способ учесть связи между символами
- Связи задаются функцией меток, связывающей метки

$$F(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) = \sum_{i=1}^{T} \theta_i(y_i \mid x_i) + \sum_{i=1}^{T-1} \theta_{i,i+1}(y_i, y_{i+1})$$

- Здесь ***y*** – метки символов, ***x*** – изображения символов

  θ - потенциалы (унарные и парные)
- Чем *F* больше, тем конфигурация лучше
- Динамическое программирование – сведение к подзадачам
  - $V_i(y_i)$ – лучшее значение слагаемых *F* для слагаемых ≤ *i*
  - Проход вперёд:

$$V_1(y_1) = \theta_1(y_1 \mid x_1)$$
$$V_{i+1}(y_{i+1}) = \theta_{i+1}(y_{i+1} \mid x_{i+1}) + \max_{y_i}\Big(\theta_{i,i+1}(y_i, y_{i+1}) + V_i(y_i)\Big)$$

  - Оптимальное значение: $F^* = \max_{y_T} V_T(y_T)$
  - Проход назад для восстановления конфигурации

Обучение CRF – структурный SVM

- Обучение по размеченной выборке $\left\{ \boldsymbol{x}_n, \boldsymbol{y}_n \right\}_{n=1}^{N}$
- Обучение – задача оптимизации $\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \Psi(\boldsymbol{x}_n, \boldsymbol{y}_n \mid \boldsymbol{\theta})$
- SSVM - обобщение метода опорных векторов

$$\Psi(\boldsymbol{x}_n, \boldsymbol{y}_n \mid \boldsymbol{\theta}) = \max_{\boldsymbol{y}} \Big[F(\boldsymbol{y} \mid \boldsymbol{x}_n, \boldsymbol{\theta}) + \Delta(\boldsymbol{y}, \boldsymbol{y}_n) \Big] - F(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})$$

- Общая схема метода
  1. Вычисление потенциалов (проход вперёд через нейросеть)
  2. Вычислении функции потерь (дин. программирование)
  3. Градиент по выходу нейросети
  4. Градиент по параметрам нейросети (backprop)
  5. Шаг оптимизации

Примеры использования

- Свободный текст (Jaderberg et al., ICLR 2015)

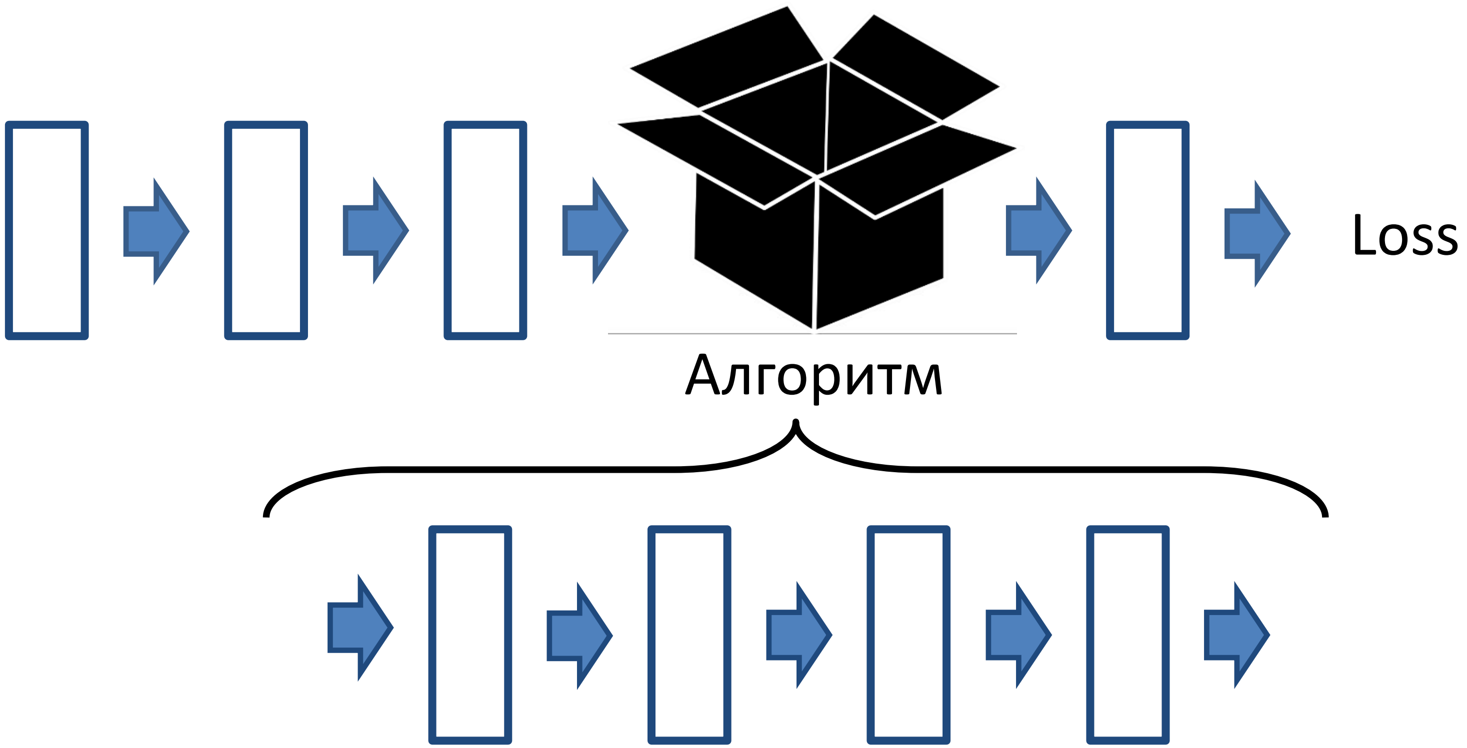
- Детектирование нескольких объектов (Vu et al., ICCV 2015)

- Тэггирование изображений (Chen et al., ICML 2015)

female/indoor/portrait sky/plant life/tree water/animals/sea animals/dog/indoor indoor/flower/plant life

Итерации алгоритма как слои сети

Нейросеть: проход вперёд



Проход назад – обычный back propagation

Обучение CRF – максимальное правдоподобие

→ command

- Обучение по размеченной выборке $\{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^{N}$
- Обучение – задача оптимизации $\min\limits_{\boldsymbol{\theta}} \frac{1}{N} \sum\limits_{n=1}^{N} \Psi(\boldsymbol{x}_n, \boldsymbol{y}_n \mid \boldsymbol{\theta})$
- Метод максимального правдоподобия

$$\Psi(\boldsymbol{x}_n, \boldsymbol{y}_n \mid \boldsymbol{\theta}) = -\log P(\boldsymbol{y} \mid \boldsymbol{x}_n, \boldsymbol{\theta}), \quad P = \frac{1}{Z(\boldsymbol{x}_n, \boldsymbol{\theta})} \exp(F(\boldsymbol{y} \mid \boldsymbol{x}_n, \boldsymbol{\theta}))$$

- Z – нормировочная константа

$$Z(\boldsymbol{x}_n, \boldsymbol{\theta}) = \sum_{\boldsymbol{y}} \exp(F(\boldsymbol{y} \mid \boldsymbol{x}_n, \boldsymbol{\theta}))$$

- Сумма экспоненциального числа слагаемых
- Используем Алгоритм!

Обучение CRF – максимальное правдоподобие

- Z – нормировочная константа

$$Z(\boldsymbol{x}_n, \boldsymbol{\theta}) = \sum_{\boldsymbol{y}} \exp(F(\boldsymbol{y} \mid \boldsymbol{x}_n, \boldsymbol{\theta}))$$

- Функция *F* обладает структурой:

$$\exp(F(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})) = \prod_{i=1}^{T} \exp(\theta_i(y_i \mid x_i)) \prod_{i=1}^{T-1} \exp(\theta_{i,i+1}(y_i, y_{i+1}))$$

- Алгоритм динамического программирования (sum-product)
  - $V_i(y_i)$ – сумма произведений *Z* для слагаемых ≤ *i*
  - Проход вперёд:

$$V_{i+1}(y_{i+1}) = \exp(\theta_{i+1}(y_{i+1} \mid x_{i+1})) \sum_{y_i} \Big(\exp(\theta_{i,i+1}(y_i, y_{i+1})) V_i(y_i) \Big)$$

  - Ответ: $Z = \sum_{y_T} V_T(y_T)$

Обучение CRF – вариационный вывод

- Вар. вывод – один из основных методов обучения байесовских вероятностных моделей
- Приближение *P* простым распределением *Q*: $Q(\boldsymbol{y}) = \prod_{i=1}^{T} q_i(y_i)$

$$\mathrm{KL}(Q \,||\, P) = -\sum_{\boldsymbol{y}} Q(\boldsymbol{y}) \log \frac{P(\boldsymbol{y})}{Q(\boldsymbol{y})} \to \min_{Q}$$

- Используем структуру модели

$$P(\boldsymbol{y}) \propto \exp(F(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})) = \prod_{i=1}^{T} \exp(\theta_i(y_i \mid x_i)) \prod_{i=1}^{T-1} \exp(\theta_{i,i+1}(y_i, y_{i+1}))$$

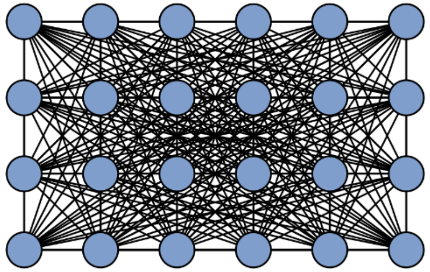
- Формулы пересчёта:

$$q_i(y_i) \propto \exp\Big(\theta_i(y_i) + \sum_{y_{i-1}} \theta_{i-1,i}(y_{i-1}, y_i) q_{i-1}(y_{i-1}) + \sum_{y_{i+1}} \theta_{i,i+1}(y_i, y_{i+1}) q_{i+1}(y_{i+1})\Big)$$

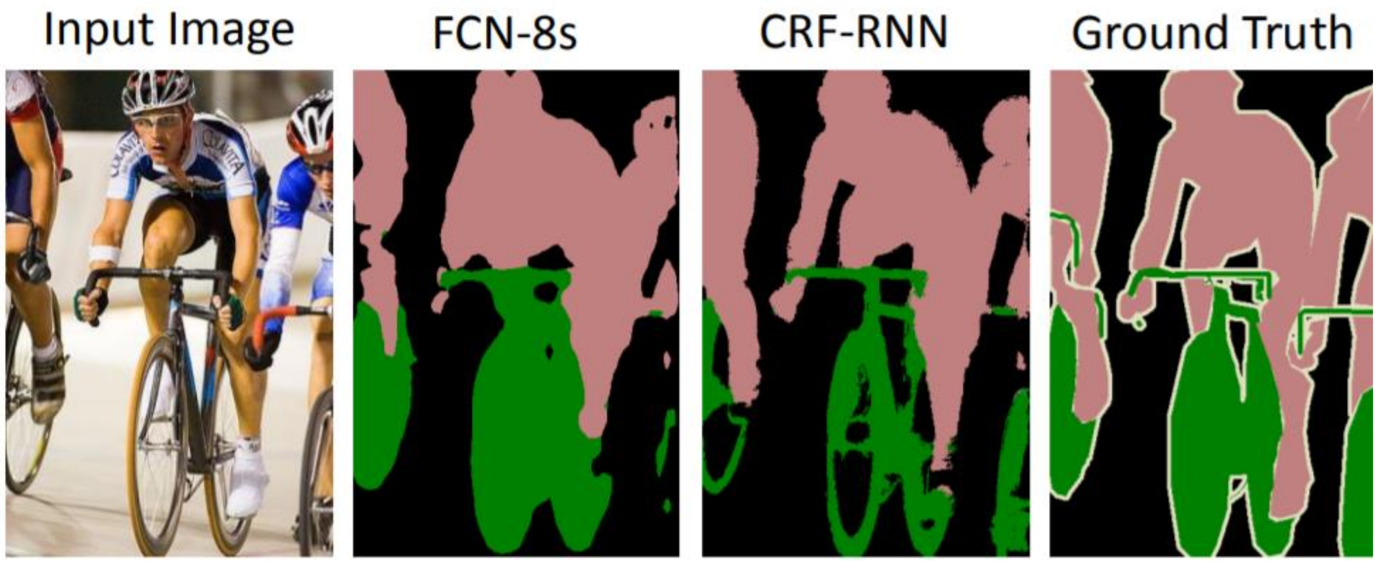
Пример: сегментация изображений

[Zheng et al., ICCV 2015]

- Вариационных вывод над полно-связной CRF



- Все операции с использованием свёрток



Пример: подавление шума

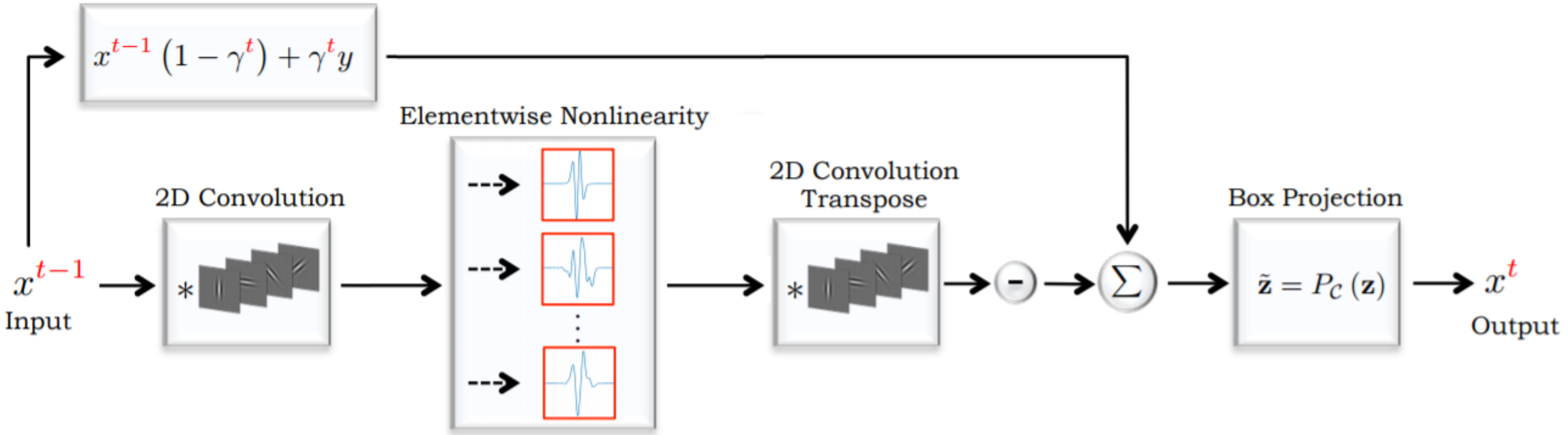
[Lefkimmiatis, CVPR 2017]

- Формулировка задачи (***x*** – ответ, ***y*** – шумное изображение):

$$\mathbf{x}^* = \underset{a \le x_n \le b}{\arg\min} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \sum_{r=1}^{R} \phi(\mathbf{L}_r \mathbf{x})$$

- Алгоритм решения – Proximal Gradient

$$\mathbf{x}^t = P_{\mathcal{C}}\left(\mathbf{x}^{t-1}(1-\gamma^t) + \gamma^t \mathbf{y} - \alpha^t \sum_{r=1}^{R} \mathbf{L}_r^{\mathsf{T}} \psi(\mathbf{L}_r \mathbf{x}^{t-1})\right)$$



Пример: подавление шума

[Lefkimmiatis, CVPR 2017]

- Формулировка задачи (***x*** – ответ, ***y*** – шумное изображение):

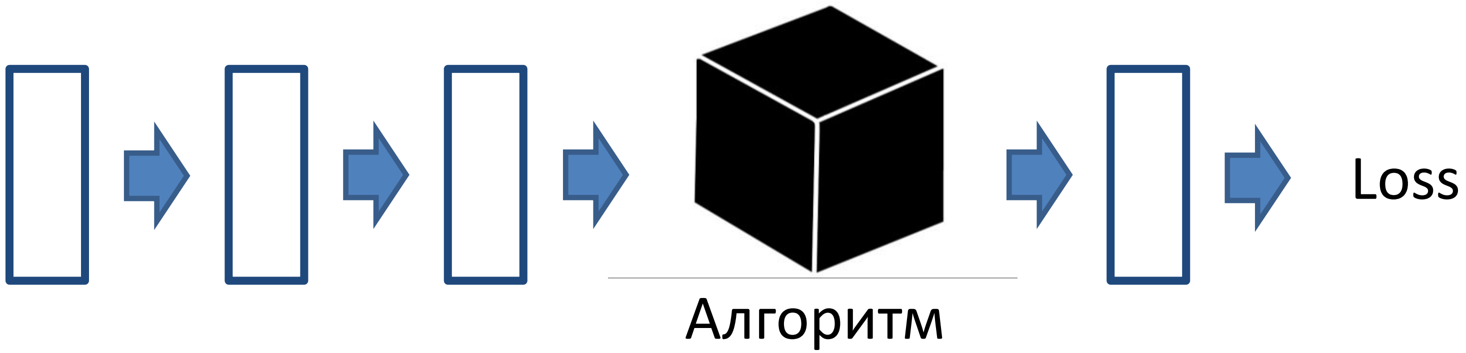
$$\mathbf{x}^* = \underset{a \le x_n \le b}{\arg\min} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \sum_{r=1}^{R} \phi\left(\mathbf{L}_r \mathbf{x}\right)$$

- Результаты:

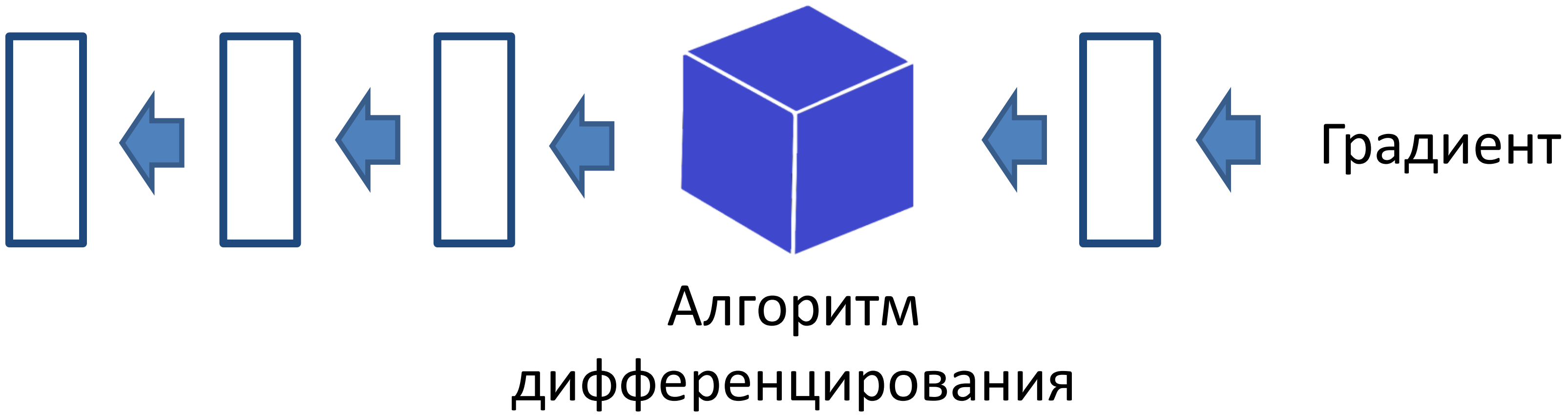
Оригинал — Оригинал + шум — Результат

Дифференцирование по входу

Нейросеть: проход вперёд



Для прохода назад нужен другой алгоритм



Пример: Gausian MRF

[Chandra&Kokkinos, ECCV 2016]

- Непрерывный аналог CRF

$$F(\boldsymbol{y}) = -\frac{1}{2}\boldsymbol{y}^T(A+\lambda I)\boldsymbol{y} + \boldsymbol{b}^T\boldsymbol{y} \to \max_{\boldsymbol{y}}$$

- ***y*** – переменные, *A*, ***b*** – параметры (выходы нейросети)
- Алгоритм предсказания – СЛАУ: $\boldsymbol{y} = (A+\lambda I)^{-1}\boldsymbol{b}$
- Задача дифференцирования
  - Вход: параметры A, ***b***, решение ***y***, градиент $\frac{d\Psi}{d\boldsymbol{y}}$
  - Найти: градиенты $\frac{d\Psi}{d\boldsymbol{b}}$ и $\frac{d\Psi}{dA}$
- Дифференцируем линейный слой (СЛАУ)

$$\frac{d\Psi}{d\boldsymbol{b}} = (A+\lambda I)^{-T}\frac{d\Psi}{d\boldsymbol{y}}$$

Пример: Gausian MRF

[Chandra&Kokkinos, ECCV 2016]

- Непрерывный аналог CRF

$$F(\boldsymbol{y}) = -\frac{1}{2}\boldsymbol{y}^T(A+\lambda I)\boldsymbol{y} + \boldsymbol{b}^T\boldsymbol{y} \rightarrow \max_{\boldsymbol{y}}$$

- ***y*** – переменные, *A*, ***b*** – параметры (выходы нейросети)
- Алгоритм предсказания – СЛАУ: $\boldsymbol{y} = (A+\lambda I)^{-1}\boldsymbol{b}$
- Задача дифференцирования
  - Вход: параметры A, ***b***, решение ***y***, градиент $\frac{d\Psi}{d\boldsymbol{y}}$
  - Найти: градиенты $\frac{d\Psi}{d\boldsymbol{b}}$ и $\frac{d\Psi}{dA}$
- Производная по *A*: $\frac{d\Psi}{dA} = -\frac{d\Psi}{d\boldsymbol{b}}\boldsymbol{y}^T$

Примеры дифференцирования

- SVD разложение $X = U\Sigma V^T$ [Ionescu et al., ICCV 2015]

$$\frac{\partial L \circ f}{\partial X} = DV^\top + U\left(\frac{\partial L}{\partial \Sigma} - U^\top D\right)_{diag} V^\top + 2U\Sigma\left(K^\top \circ \left(V^\top\left(\frac{\partial L}{\partial V} - VD^\top U\Sigma\right)\right)\right)_{sym} V^\top$$

$$K_{ij} = \begin{cases} \frac{1}{\sigma_i^2 - \sigma_j^2}, & i \neq j \\ 0, & i = j \end{cases} \qquad D = \left(\frac{\partial L}{\partial U}\right)_1 \Sigma_n^{-1} - U_2\left(\frac{\partial L}{\partial U}\right)_2^\top U_1 \Sigma_n^{-1}$$

- Дискретная! оптимизация [Djolonga&Krause, NIPS 2017]
  - Используется эквивалентность дискретной минимизации субмодулярных функций непрерывным релаксациям
- Решение дифференциальных уравнений [Chen et al., NeurIPS 2018]
  - Вычисление градиента – решение другого уравнения

Дифференцирование неявных слоев

- Что такое неявный слой?
  - Явный слой $\boldsymbol{y} := f(\boldsymbol{x})$
  - Неявный слой $h(\boldsymbol{x},\ \boldsymbol{y}(\boldsymbol{x})) = 0$ (система уравнений)
  - Алгоритм решает уравнение

- Как дифференцировать?

1. $\frac{d}{d\boldsymbol{x}} h(\boldsymbol{x},\ \boldsymbol{y}(\boldsymbol{x})) = 0$
2. $\boxed{\frac{\partial h}{\partial \boldsymbol{x}}} + \boxed{\frac{\partial h}{\partial \boldsymbol{y}}} \cdot \frac{d\boldsymbol{y}}{d\boldsymbol{x}} = 0$ Автоматическое дифференцирование
3. $\frac{d\boldsymbol{y}}{d\boldsymbol{x}} := -\left(\frac{\partial h}{\partial \boldsymbol{y}}\right)^{-1} \frac{\partial h}{\partial \boldsymbol{x}}$

A. Якобиан обратим? (+ регуляризация)
B. Неявные методы для решения

Примеры неявных слоев

- OptNet: Differentiable Optimization as a Layer [Amos&Kolter, 2017]
- Пример: квадратичное программирование (QP)

$$\hat{\boldsymbol{y}}(\boldsymbol{x}) := \arg\min_{\boldsymbol{y}} \frac{1}{2}\boldsymbol{y}^T Q(\boldsymbol{x})\boldsymbol{y} + \boldsymbol{y}^T \boldsymbol{c}(\boldsymbol{x})$$
$$\text{s.t. } A(\boldsymbol{x})\boldsymbol{y} = \boldsymbol{b}(\boldsymbol{x})$$
$$G(\boldsymbol{x})\boldsymbol{y} \le \boldsymbol{s}(\boldsymbol{x})$$

- Уравнения: условия ККТ (Каруша-Куна-Таккера)

$$Q\boldsymbol{y} + \boldsymbol{c} + A^T\boldsymbol{\nu} + G^T\boldsymbol{\lambda} = 0$$
$$A\boldsymbol{y} - \boldsymbol{b} = 0$$
$$\operatorname{diag}(\boldsymbol{\lambda})(G\boldsymbol{y} - \boldsymbol{s}) = 0$$

Еще примеры:
- SATNet [Wang et al., 2019]
  - Релаксация дискретной оптимизации
- Deep equilibrium models [Bai et al, 2019]
  - Повторение слоев ResNet/Transformer
- Движок физики в симуляторе [de Avila Belbute-Peres et al., 2018]
  - Законы физики = уравнения

Заключение

- Разные способы встраивать алгоритмы в нейросети
  - Структурный пулинг
  - Итерации алгоритма => слои нейросети
  - Прямое (аналитическое) дифференцирование по входу
- Расширение библиотеки слоев
- Использование алгоритмов позволяет встраивать знания о задаче в нейросети