

# Основы практического использования нейронных сетей.

## Лекция 6. Разработка эффективных НС.

Дмитрий Буряк.  
к.ф.-м.н.  
dyb04@yandex.ru

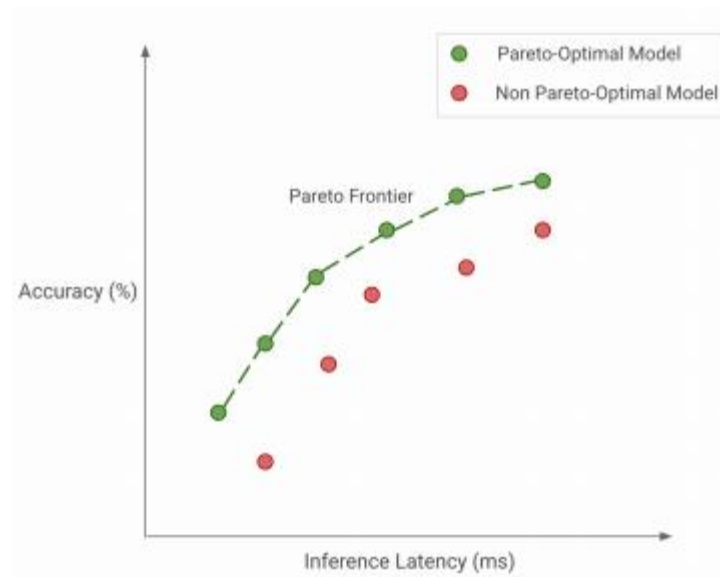
# Эффективность НС

## □ Эффективность обученной НС

- Размер сети
- Латентность
- Число MAC/FLOP

## □ Эффективность проведения обучения

- Размер сети
- Точность



# Технологии повышения эффективности НС

## Areas

Compression  
Techniques

Learning  
Techniques

Automation

Efficient  
Architectures

## Description

- уменьшение размера НС
- удаление связей, нейронов, ...
- уменьшение разрядности представления НС

- эффективные алгоритмы обучения
- устойчивые функции потерь
- дистилляция

- автоматиз-ный поиск значений гиперпараметров
- подбор архитектуры

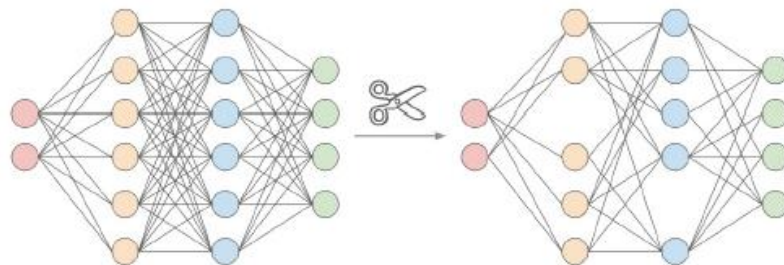
- разработка и применение эффективных архитектур

- библиотеки разработки и реализации НС
- высокопроизводительные вычислительные платформы

# Прореживание НС

# Прореживание (pruning) НС

- Удаление элементов архитектуры НС
  - Связи, нейроны, фильтры, ...



---

## Algorithm 1: Standard Network Pruning with Fine-Tuning

---

**Data:** Pre-trained dense network with weights  $W$ , inputs  $X$ , number of pruning rounds  $N$ , fraction of parameters to prune per round  $p$ .

**Result:** Pruned network with weights  $W'$ .

```
1  $W' \leftarrow W$ ;  
2 for  $i \leftarrow 1$  to  $N$  do  
3    $S \leftarrow \text{compute\_saliency\_scores}(W')$ ;  
4    $W' \leftarrow W' - \text{select\_min\_k}(S, \frac{|W'|}{p})$ ;  
5    $W' \leftarrow \text{fine\_tune}(X, W')$   
6 end  
7 return  $W'$ 
```

---

*Стандартный алгоритм прореживания обученной НС*

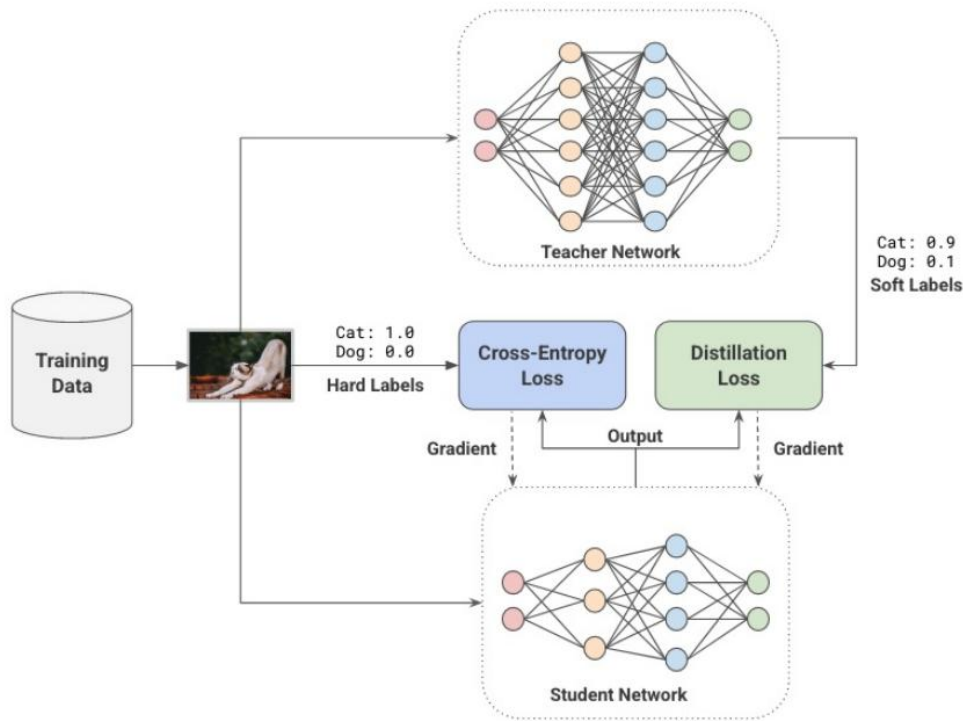
# Особенности алгоритмов прореживания НС

- ☐ Критерий выбора элементов для прореживания
  - оценка влияния на функцию потерь
  - абсолютная величина, вторая производная ...
- ☐ Элементы НС для удаления
- ☐ Распределение доли удаляемых элементов по НС
- ☐ Расписание проведения прореживания
- ☐ Возможность восстановления связей
- ☐ «Гипотеза о лотерейном билете»
  - В любой большой НС существует подсеть, которая может быть обучена с такой же эффективностью

# Дистилляция НС

# Дистилляция НС. Основная идея

- ❑ Стратегия обучения на выходы сети «учителя».
- ❑ Сеть «учитель» – «большая» НС/ансамбль НС
- ❑ Сеть «ученик» - целевая сеть, которая должна удовлетворять требованиям на число параметров и вычислительную сложность





# Дистилляция НС. Функция потерь

$$L = \lambda_1 \cdot L_{\text{ground-truth}} + \lambda_2 \cdot L_{\text{distillation}}$$

- $L_{\text{ground-truth}}$  – базовая функция потерь на основной обучающей выборке .
- $L_{\text{distillation}}$  – функция потерь относительно выходов сети «учитель».
- Задание меток относительно результатов сети «учитель»

$$Y_i^{(t)} = \frac{\exp(Z_i^{(t)}/T)}{\sum_{j=1}^n \exp(Z_j^{(t)}/T)}$$

T - температура

$$\begin{aligned} L &= \lambda_1 \cdot L_{\text{ground-truth}} + \lambda_2 \cdot L_{\text{distillation}} \\ &= \lambda_1 \cdot \text{CrossEntropy}(Y, Y^{(s)}; \theta) + \lambda_2 \cdot \text{CrossEntropy}(Y^{(t)}, Y^{(s)}; \theta) \end{aligned}$$

# Автоматизированная настройка гиперпараметров НС

# Гиперпараметры НС

- ❑ Параметры НС, которые не настраиваются в процессе обучения.
- ❑ Влияют на емкость (способность к обобщению), скорость обучения и объем требуемой памяти НС.
- ❑ Параметры, определяющие архитектуру НС (число и размеры слоев)
- ❑ Параметры алгоритма обучения

# Определение значений гиперпараметров

## ☐ «Ручной»

- требует понимания влияния гиперпараметров на свойства и характеристики сети;
- использование подтверждающей выборки, кроссвалидация.

## ☐ Автоматический

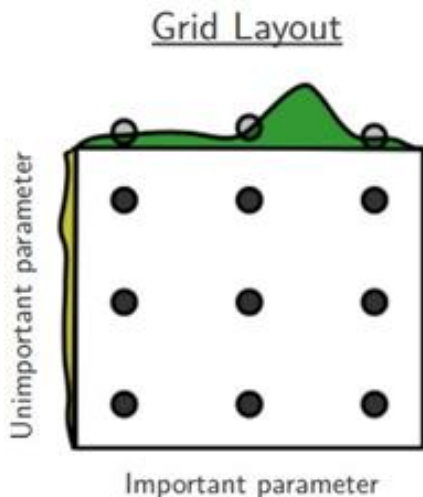
- требует большой объем дополнительных вычислений.

# Автоматический поиск значений гиперпараметров

- ❑ Оптимизация гиперпараметров (Hyper-Parameter Optimization – HPO )
- ❑ Создание «обертки» для тестирования различных наборов значений гиперпараметров и выбора оптимального.
- ❑ Функция оптимизации – значение ошибки на подтверждающей выборке.
- ❑ «Обертка» имеет собственные гиперпараметры.
  
- ❑ Подходы к формированию наборов значений гиперпараметров:
  - регулярный
  - случайный

# Регулярное формирование наборов значений гиперпараметров

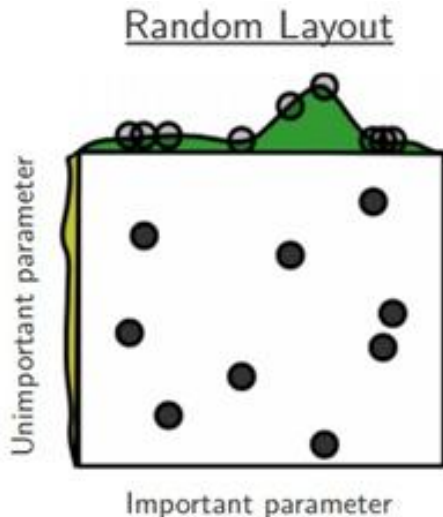
- ❑ Выбор фиксированного числа значений каждого гиперпараметра.
- ❑ Формирование комбинаций выбранных значений.



- ❑ Повторный поиск на границах интервала и на более частой сетке в выбранных областях.
- ❑ Низкая эффективная размерность пространства гиперпараметров.
- ❑ Различная значимость гиперпараметров на емкость НС.
- ❑ Экспоненциальный рост числа вычислений.
- ❑ Эффективен для малого числа гиперпараметров.
- ❑ Хорошо распараллеливается.

# Случайное формирование наборов значений гиперпараметров

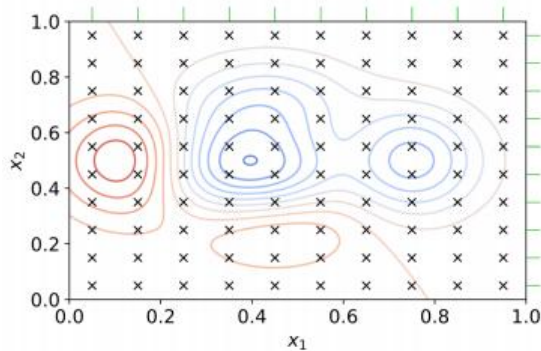
- ❑ Случайный выбор вектора значений гиперпараметров в соответствии с заданным распределением.



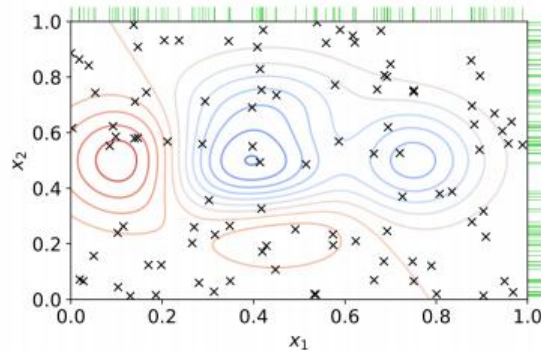
- ❑ Повторный поиск в выбранных областях.
- ❑ Устойчив к пространствам низкой эффективной размерности.
- ❑ Более эффективный метод относительно числа требуемых вычислений.
- ❑ Хорошо распараллеливается.

# Байесовская оптимизация

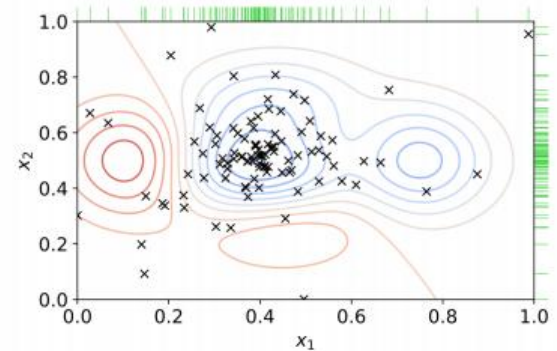
- ❑ Выбор значений для проверки на основе результатов тестирования предыдущих значений гиперпараметров.
- ❑ Построение отдельной функции для выбора значений для проверки.



(a) Grid Search



(b) Random Search



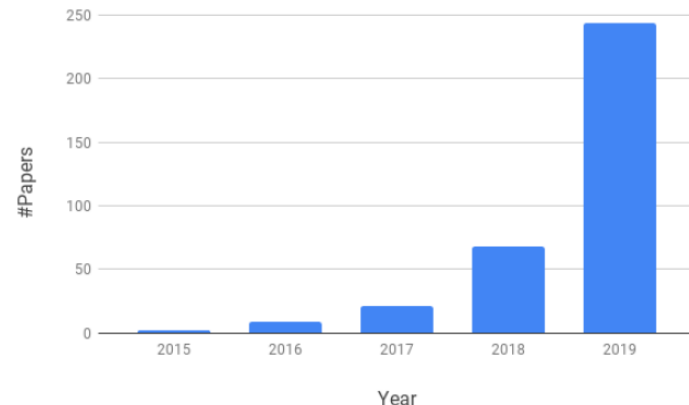
(c) Bayesian Optimization



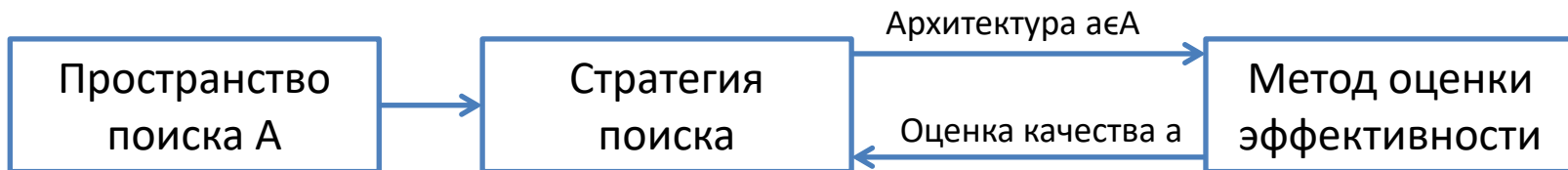
# Автоматизированный поиск архитектуры НС (1)

## □ Автоматизированный поиск архитектуры (NAS):

- Пространство поиска – среди каких архитектур искать.
- Стратегия поиска – метод поиска (оптимизации).
- Оценка эффективности – как оценивать качество архитектуры.



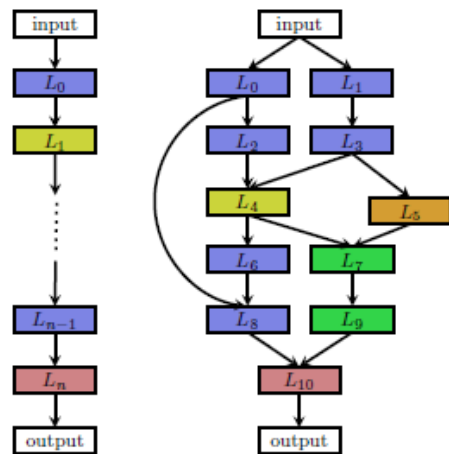
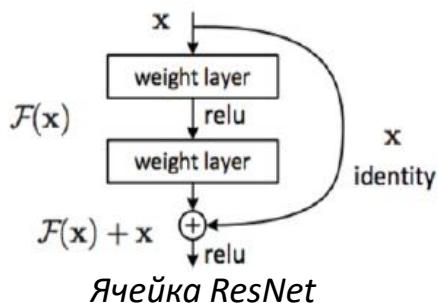
Число статей, посвященных NAS в год  
(M. Lindauer et al. Best Practices for Scientific Research on Neural Architecture Search, 2020)



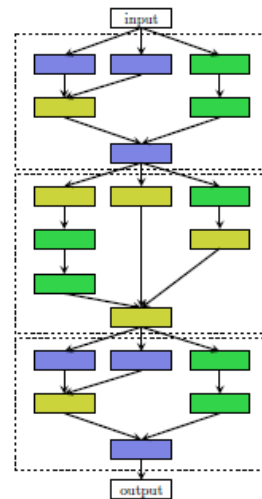
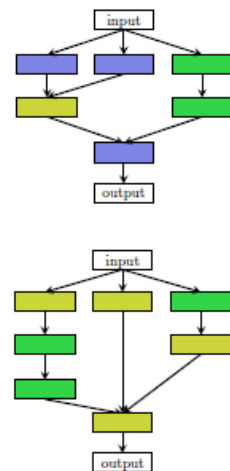
# Автоматизированный поиск архитектуры НС (2)

## □ Пространство поиска

- Пространство сетей со структурой цепочки.
- Оптимизация структуры ячеек (сеть строится из одинаковых ячеек): с сохранением и уменьшением размерности (например, ResNet).



Сети со структурой цепочки



Примеры ячеек и результирующей сети

# Автоматизированный поиск архитектуры НС (3)

## ❑ Стратегия поиска

- Случайный поиск, Байесовская оптимизация, эволюционные методы, обучение с подкреплением, градиентные методы.

## ❑ Оценка эффективности

- Обучить – протестировать.
- Уменьшить число вычислений: ограничить время обучения, размерность и т.п.
- Экстраполяция кривой обучения.
- Инициализация весов на основе протестированных архитектур.
- One-Short Architecture Search – все архитектуры – подграфы суперграфа.

# Примеры алгоритмов

## ❑ NEAT – Neuroevolution of Augmenting Topologies (2002)

- Для полносвязанных НС.
- Применение генетических алгоритмов.
- Кодирования архитектуры НС, реализация операций скрещивания и мутации.
- Расширения для глубоких НС: DeepNeat, CoDeepNeat.

## ❑ DARTS - Differentiable Neural Architecture Search (2019)

- Поиск «оптимальной» ячейки, НС строится из ячеек.
- Использование весов уже обученных сетей.
- Параметризация архитектуры → применение метода градиентного спуска.
- Вычислительная эффективность ~100х.

# Вопросы

- ❑ Для чего используют алгоритмы прореживания?
- ❑ В чем состоит основная идея дистилляции?
- ❑ Почему при оптимизации гиперпараметров случайное формирование наборов значений часто эффективнее чем регулярное (по решетке)?
- ❑ Основные элементы NAS?