

# Стратегии оценки производительности в поиске нейронных архитектур (NAS)

На основе современных исследований

Презентация по материалам научных публикаций NAS

9 ноября 2025 г.

## Основная цель

- Цель NAS — найти архитектуру  $A$ , обеспечивающую максимум производительности (например, точности модели на тестовых данных).
- Прямая оценка: обучение каждой архитектуры с нуля и проверка на валидации.
- Проблема: огромные вычислительные затраты (до тысяч GPU-дней).
- Решение: использовать приближённые оценки — **low-fidelity метрики**.

# Low-fidelity метрики

**Идея:** оценивать производительность архитектуры на основе упрощённых условий обучения.

- Сокращённое время обучения [17, 18].
- Обучение на подмножестве данных [8].
- Снижение разрешения изображений [4].
- Уменьшение числа фильтров в слоях [14, 18].

**Преимущества:** сильное снижение вычислительных затрат.

**Недостатки:** возможная недооценка реальной производительности.

# Проблема стабильности ранжирования

- Относительное ранжирование архитектур может меняться при переходе от приближённой оценки к «полной».
- Это нарушает корректность поиска.
- Решение: постепенное повышение точности оценки по мере продвижения поиска (см. работы [6, 9]).

# Экстраполяция кривых обучения

- Использование ранних кривых обучения для прогнозирования итоговой производительности [1, 5, 7, 13, 15].
- Преждевременное прекращение обучения для слабых архитектур.
- Применение суррогатных моделей для предсказания качества на основе характеристик архитектуры [10].

**Основная сложность:** необходимость точных прогнозов при ограниченном числе обученных архитектур.

# Морфизмы сети (Network Morphisms)

- Использование весов ранее обученных архитектур.
- Модификация сети без изменения представляемой функции.
- Возможность постепенного увеличения ёмкости сети без переобучения с нуля.

**Плюсы:** сохранение знаний, ускорение поиска.

**Минусы:** риск чрезмерного усложнения, необходимость приближённых морфизмов для уменьшения архитектур.

# Однократный поиск архитектуры (One-Shot NAS)

- Все архитектуры рассматриваются как подграфы общего суперграфа (модель *one-shot*).
- Общие веса между архитектурами позволяют не обучать каждую архитектуру отдельно.
- Оценка проводится только на валидационном наборе.

## Преимущества:

- Ускорение оценки с тысяч GPU-дней до нескольких.
- Высокая корреляция оценочной и реальной производительности.

# Примеры методов One-Shot NAS

- **ENAS** [12] — контроллер RNN выбирает архитектуры, использует метод REINFORCE.
- **DARTS** [11] — непрерывная релаксация пространства поиска, оптимизация весов и архитектур одновременно.
- **Bender et al.** [2] — фиксированное распределение и стохастическое отключение путей при обучении.

**Вывод:** комбинация разделения весов и фиксированного распределения — ключевой фактор успеха One-Shot NAS.

# Гиперсети (Hypernetworks)

- Генерация весов для новых архитектур без отдельного обучения каждой.
- Общая гиперсеть предсказывает веса на основе параметров архитектуры.

**Отличие:** веса не общие напрямую, а создаются моделью-гиперсетью [3].

# Недостатки One-Shot подходов

- Пространство поиска ограничено подграфами заранее заданного суперграфа.
- Требование хранения всего суперграфа в памяти GPU.
- Возможна предвзятость при обучении — отдельные области пространства могут быть изучены лучше.

**Последствие:** риск преждевременной конвергенции поиска NAS.

# Выводы

- Методы оценки производительности — ключевой элемент эффективного NAS.
- Low-fidelity и one-shot подходы позволяют значительно ускорить поиск.
- Необходим систематический анализ ошибок и смещений, вносимых различными оценщиками.

## Перспективы

Комбинация surrogate-моделей, экстраполяции кривых обучения и weight-sharing подходов обещает улучшение точности при снижении вычислительных затрат.

# Список литературы I

-  Baker et al. (2017). *Accelerating Neural Architecture Search using Learning Curve Extrapolation.*
-  Bender et al. (2019). *Understanding and Simplifying One-Shot Architecture Search.* ICML.
-  Zhang et al. (2019). *Graph Hypernetworks for Neural Architecture Search.* ICLR.
-  Brock et al. (2017). *SMASH: One-Shot Model Architecture Search through HyperNetworks.*
-  Domhan et al. (2015). *Speeding up Automatic Hyperparameter Optimization by Learning Curves.*
-  Real et al. (2019). *Regularized Evolution for Image Classifier Architecture Search.* AAAI.

## Список литературы II

-  Klein et al. (2017). *Learning Curve Prediction with Bayesian Neural Networks.* ICLR.
-  Zoph et al. (2018). *Learning Transferable Architectures for Scalable Image Recognition.*
-  Elsken et al. (2019). *Efficient Multi-objective Neural Architecture Search via Lamarckian Evolution.* ICLR.
-  Liu et al. (2018). *Progressive Neural Architecture Search.* ECCV.
-  Liu et al. (2019). *DARTS: Differentiable Architecture Search.* ICLR.
-  Pham et al. (2018). *Efficient Neural Architecture Search via Parameter Sharing.* ICML.
-  Raval & Miikkulainen (2018). *Learning Curve Prediction and Early Stopping with Gaussian Processes.*

## Список литературы III

-  Zela et al. (2018). *Towards Automated Deep Learning: Efficient Joint Neural Architecture and Hyperparameter Search.*
-  Swersky et al. (2014). *Freeze-Thaw Bayesian Optimization.*
-  Chen et al. (2016). *Net2Net: Accelerating Learning via Knowledge Transfer.*
-  Klein et al. (2017). *Learning Curve Prediction with Bayesian Neural Networks. ICLR.*
-  Li et al. (2018). *Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. JMLR.*