



Физический факультет
Кафедра вычислительной физики

Выбор гиперпараметров нейронной сети для решения задач математической физики

Выполнил: Поляков Даниил Николаевич
Научный руководитель: Степанова Маргарита Михайловна

Введение



- Дифференциальные уравнения:
 - Метод конечных разностей (МКР)
 - Метод конечных элементов (МКЭ)
- Physics-Informed Neural Network (PINN)
- Hyper-Parameter Optimization (HPO)
- Построение и тренировка модели: **PyTorch**
- Автоматическая оптимизация гиперпараметров: **Ray Tune**

Цели работы



Цели, поставленные в текущей работе:

1. изучить влияние гиперпараметров на сходимость нейронной сети к решению;
2. сравнить методы автоматического подбора гиперпараметров;
3. исследовать возможность решения уравнения Гельмгольца в пространстве большой размерности.

Решение уравнений с помощью нейронных сетей



$$f\left(\mathbf{x}; u; \frac{\partial u}{\partial x_1}, \dots, \frac{\partial u}{\partial x_n}; \frac{\partial^2 u}{\partial x_1 \partial x_1}, \dots, \frac{\partial^2 u}{\partial x_n \partial x_n}\right) = 0, \quad \mathbf{x} \in D, \quad D \subset \mathbb{R}^n$$
$$b(\mathbf{x}, u)|_{\Gamma} = 0$$

В качестве меры ошибки и целевой функции можно использовать взвешенную сумму L^2 норм отклонений (невязок) по уравнению и по границе:

$$\text{loss} = \omega_f \text{loss}_f + \omega_b \text{loss}_b$$
$$\text{loss}_f = \frac{1}{|N_f|} \sum_{\mathbf{x} \in N_f} \left\| f\left(\mathbf{x}; \hat{u}; \frac{\partial \hat{u}}{\partial x_1}, \dots, \frac{\partial \hat{u}}{\partial x_n}; \frac{\partial^2 \hat{u}}{\partial x_1 \partial x_1}, \dots, \frac{\partial^2 \hat{u}}{\partial x_n \partial x_n}\right) \right\|^2$$
$$\text{loss}_b = \frac{1}{|N_b|} \sum_{\mathbf{x} \in N_b} \|b(\mathbf{x}, \hat{u})\|^2$$

где ω_f, ω_b — веса; $N_f = \{x_1, \dots, x_{|N_f|}\}$, $N_b = \{x_1, \dots, x_{|N_b|}\}$ — множества точек коллокации (тренировочная выборка) по уравнению и по границе соответственно; $|N_f|, |N_b|$ — количества точек; \hat{u} — решение нейронной сети.

Описание задачи

Рассматриваем уравнение Гельмгольца размерности n на единичном кубе:

$$-\Delta u(\mathbf{x}) - u(\mathbf{x}) = (4\pi^2 n - 1) \cdot \prod_i^n \sin(2\pi x_i)$$

$$u|_r = 0$$

$$D = \{\vec{x}: 0 \leq x_i \leq 1, i = 1, \dots, n\}$$

Правая часть уравнения выбрана таким образом, чтобы решение рассчитывалось аналитически для любой размерности:

$$u(\mathbf{x}) = \prod_i^n \sin(2\pi x_i)$$

В решаемой задаче возможно жёсткое форсирование граничного условия (см. след. слайд), поэтому функция ошибки будет состоять только из невязки по уравнению:

$$\text{loss} = \text{MSE}_f = \left\langle \left(\Delta u(\mathbf{x}) + u(\mathbf{x}) + (4\pi^2 n - 1) \cdot \prod_i^n \sin(2\pi x_i) \right)^2 \right\rangle$$

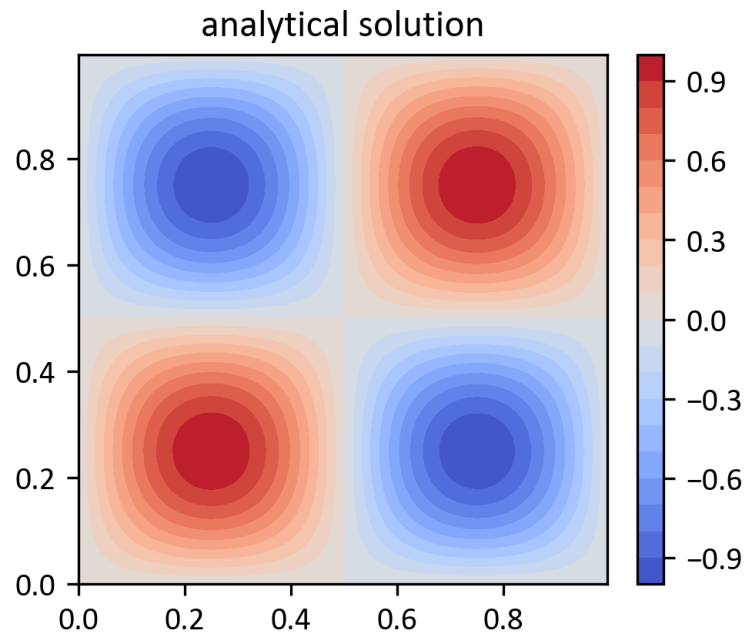


График 1. Аналитическое решение в двумерном пространстве

Структура сети

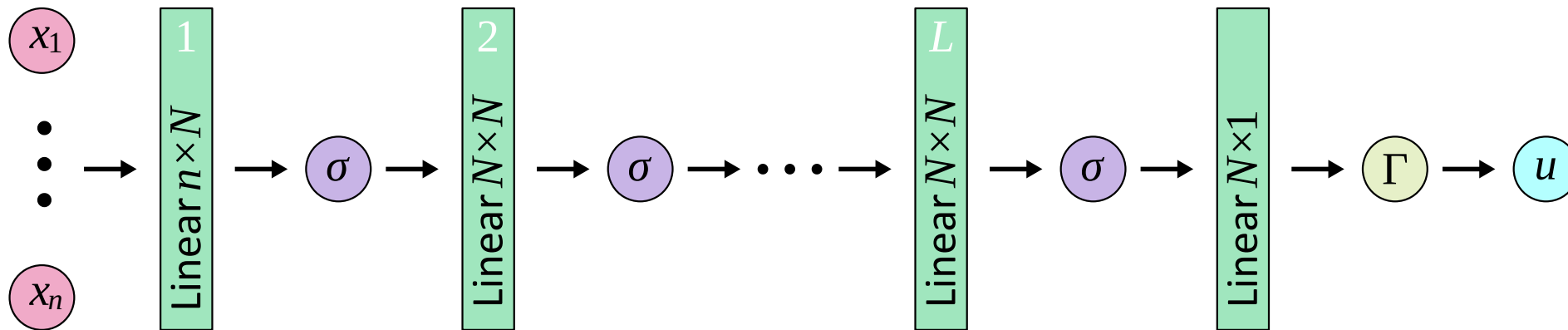


Рисунок 2. Схема используемой FNN-сети

Количество слоёв L , ширина слоя N , и функция активации σ подлежат подбору. На выходе сети производится форсирование граничного условия. Оно реализуется домножением выходного значения на гладкую функцию, удовлетворяющую граничным условиям:

$$\tilde{u} = u \cdot \prod_i^n (x_i - 0)(1 - x_i)$$

Генерация точек

Обучающую выборку генерируем двумя способами:

- классическим генератором равномерно распределённых случайных чисел;
- квазислучайным генератором из последовательности Соболя.

При квазислучайной генерации точек гарантируется равномерное заполнение области определения уравнения и каждого батча.

Тестовые точки будем всегда выбирать из классического равномерного распределения. Их количество всегда выбираем равным количеству точек в обучающей выборке.

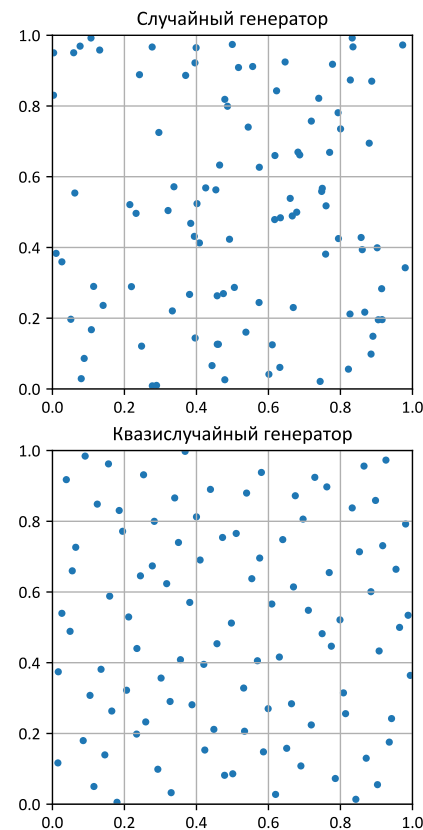


Рисунок 3. Примеры выборок, сгенерированных двумя генераторами

Оптимизация гиперпараметров



Гиперпараметры, связанные с процессом обучения:

- $|N_f|$ — количество точек в обучающей выборке;
- RNG — способ генерации точек в обучающей выборке;
- $|N_{\text{batch}}|$ — размер батча;
- lr — начальная скорость обучения;
- lr scheduler — планировщик скорости обучения:
 - None;
 - ExponentialLR-0.95;
 - ReduceLROnPlateau-0.1-10;
 - ReduceLROnPlateau-0.5-2.

Гиперпараметры, связанные с архитектурой сети:

- N — ширина скрытого слоя (количество нейронов в скрытом слое);
- L — количество скрытых слоёв;
- σ — функция активации:
 - $\text{ELU}(x) = \begin{cases} x, & x > 0 \\ e^x - 1, & x \leq 0 \end{cases}$
 - $\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$
 - $\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
 - $\sin(x)$
 - $\frac{\tan^{-1}(x)}{\pi/2}$

Вычислительные ресурсы



Обучение нейронных сетей проводилось на узлах РЦ ВЦ СПбГУ:

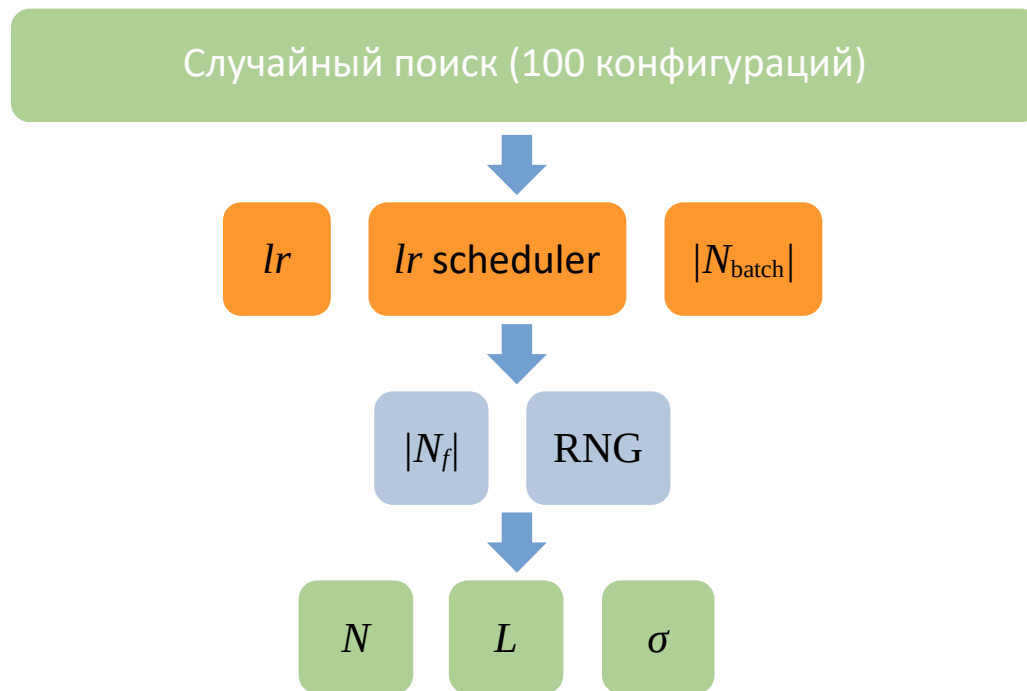
- ручная оптимизация — **NVIDIA Tesla P100**, 16 GB;
- автоматическая оптимизация и финальное обучение — **NVIDIA RTX A6000**, 48 GB.

RTX A6000 превосходит Tesla P100 по скорости обучения моделей примерно в 1.8 раз.

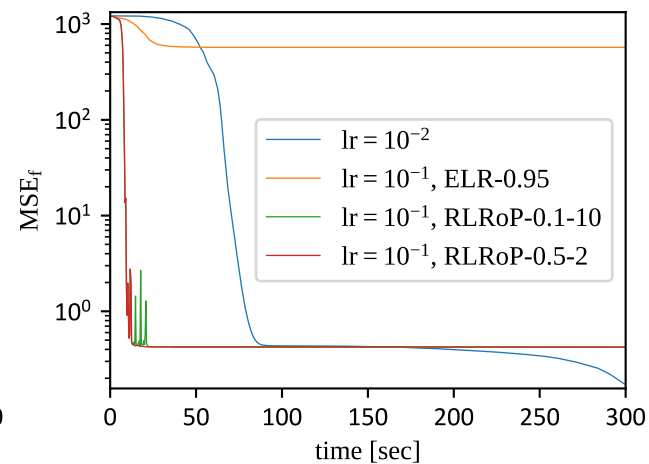
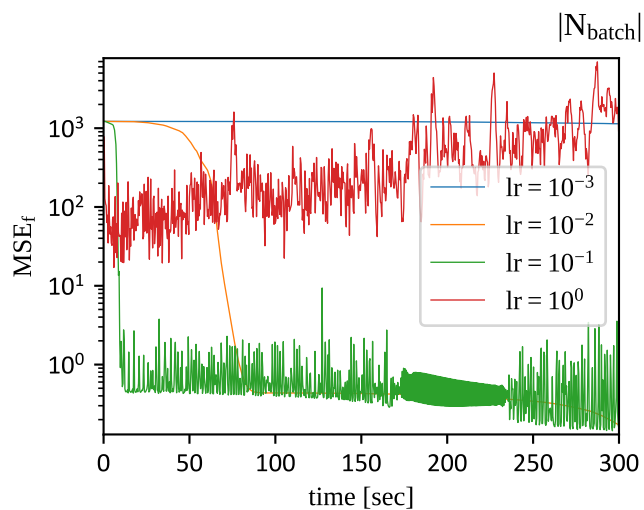
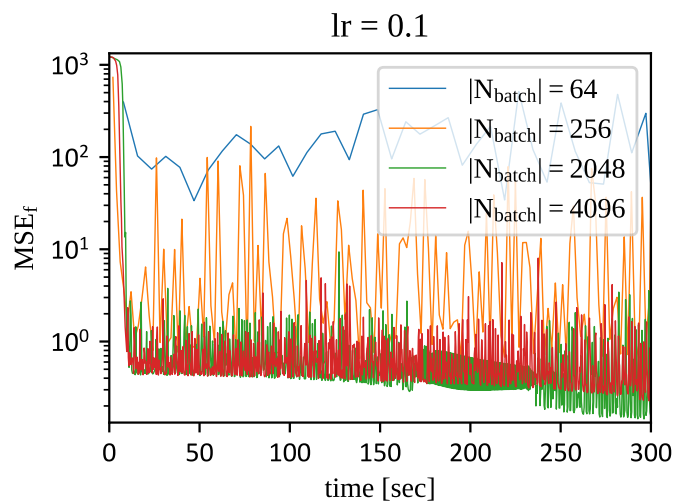
Ручная оптимизация гиперпараметров — Методика



Сначала проводим исследование влияния гиперпараметров на сходимость сети в случаях 2-мерной и 5-мерной задач.



Ручная оптимизация гиперпараметров — Результаты



Ручная оптимизация гиперпараметров — Результаты

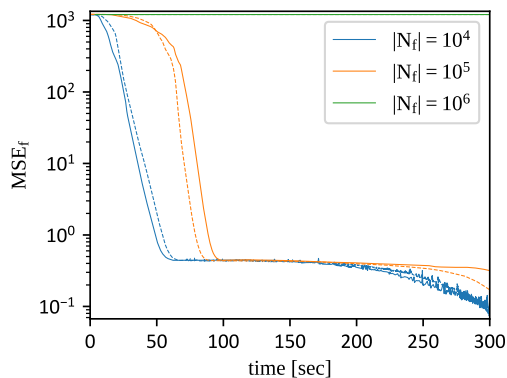
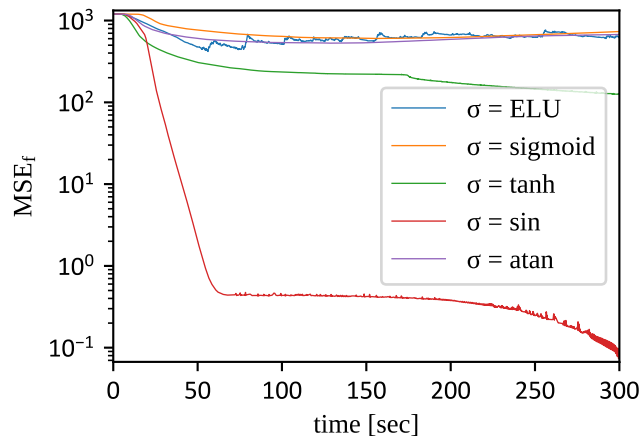


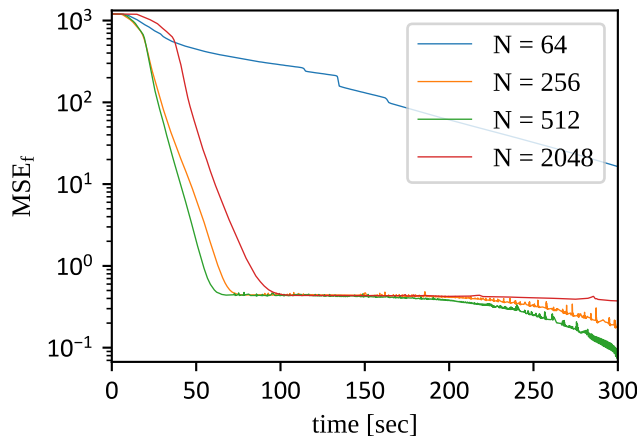
Таблица 1. Лучшая конфигурация модели для решения 5-мерной задачи, полученная вручную

t , сек	MSE_f	$RMSE_u$	N	L	σ	$ N_f $	RNG	$ N_{batch} $	lr	lr scheduler
300	8.92E-02	1.52E-03	512	1	sin	10^4	квазисл.	2048	10^{-2}	None

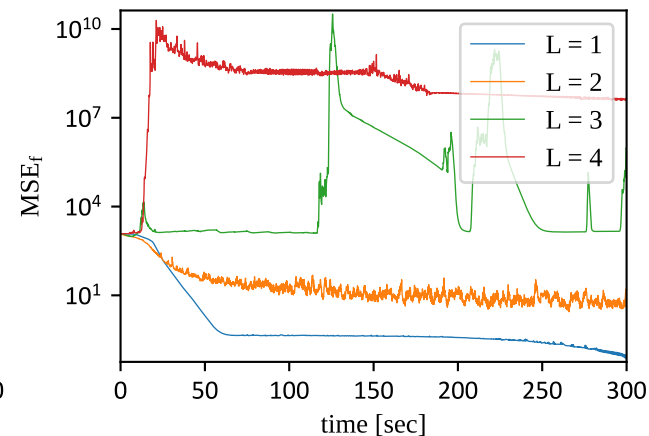
$N = 512, L = 1$



$\sigma = \sin, L = 1$



$\sigma = \sin, N = 512$



Ручная оптимизация гиперпараметров — Выводы



- Размер батча имеет относительно низкое влияние на сходимость сети по сравнению с другими параметрами. Однако не следует выбирать размер батча слишком малым.
- При больших значениях lr целевая функция быстро понижается на первых итерациях, а затем приобретает осциллирующий вид. При малых lr сходимость медленная, но более плавная, а абсолютная точность сети, как правило, выше.
- Использование планировщика lr в большинстве случаев предпочтительнее, чем постоянное значение lr , и даёт выигрыш по скорости и абсолютной точности. Однако, неправильно подобранные параметры планировщика могут ухудшить результат.
- Увеличение числа точек коллокации не всегда улучшает скорость сходимости, а иногда даже значительно ухудшает её.
- Квазислучайный метод генерации точек превосходит случайный метод, но выигрыш, как правило, заметен при небольшом количестве точек.
- Для рассмотренной задачи лучшую аппроксимацию дают неглубокие сети с 1–2 скрытыми слоями.
- Из рассмотренных функций активации \sin превосходит \tanh и сильно превосходит все остальные функции.

Суммарное процессорное время, ушедшее на ручную оптимизацию в случае 5-мерной задачи — 24 часа.

Автоматическая оптимизация гиперпараметров



Автоматическая оптимизация гиперпараметров (**НРО**) состоит из двух основных компонентов:

- **алгоритм поиска** — выбирает точку из области гиперпараметров;
- **планировщик** — заранее останавливает процесс обучения сети в случае плохой сходимости относительно предыдущих результатов.

Алгоритмы поиска

Самым базовым методом поиска является **поиск по сетке**. Такой поиск применим только для небольшого набора гиперпараметров.

Улучшение по сравнению с поиском по сетке даёт **случайный поиск**, который статистически превосходит поиск по сетке, особенно при больших размерностях области поиска.

Байесовская оптимизация — последовательный итерационный алгоритм, основанный на двух компонентах: вероятностной суррогатной модели и функции сбора. В качестве суррогатных моделей традиционно применяются гауссовы процессы и древовидный оценщик Парзена (TPE). Методы байесовской оптимизации показывают себя лучше случайного поиска и предназначены для оптимизации достаточно ресурсоёмких функций.

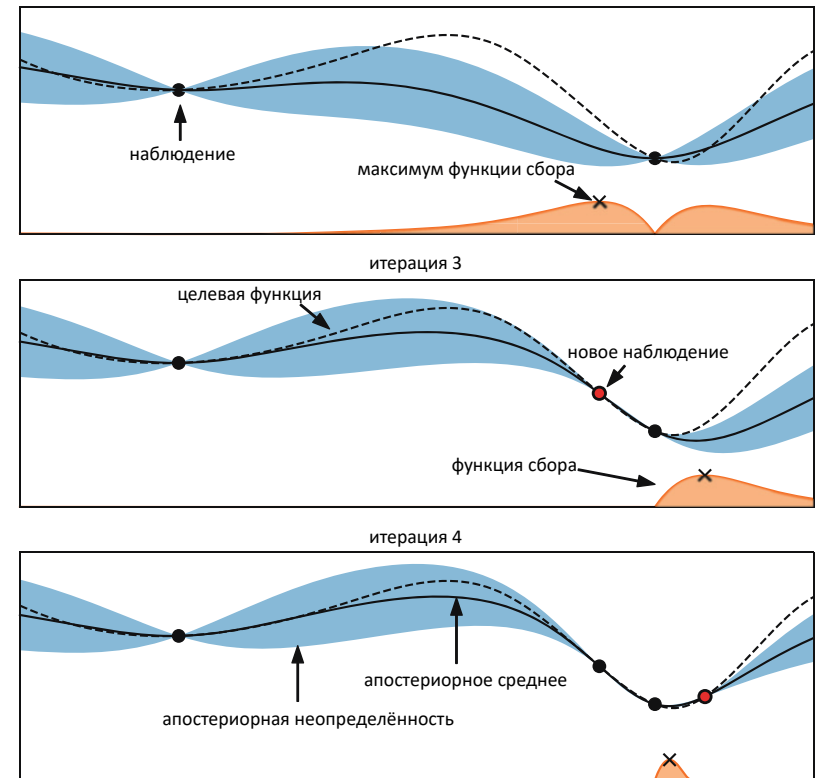


Рисунок 4. Иллюстрация байесовской оптимизации одномерной функции

Планировщики обучения



Медианная остановка является простейшим методом ранней остановки.

В алгоритме **последовательного деления (Successive Halving, SHA)** со временем прекращается обучение половины наихудших конфигураций, затем процесс повторяется для оставшихся конфигураций.

HyperBand и **Asynchronous Successive Halving (ASHA)** являются модификациями метода SHA.

Алгоритмы поиска и планировщики обучения могут работать в комбинации независимо друг от друга. **BOHB** — гибрид байесовского поиска и HyperBand.

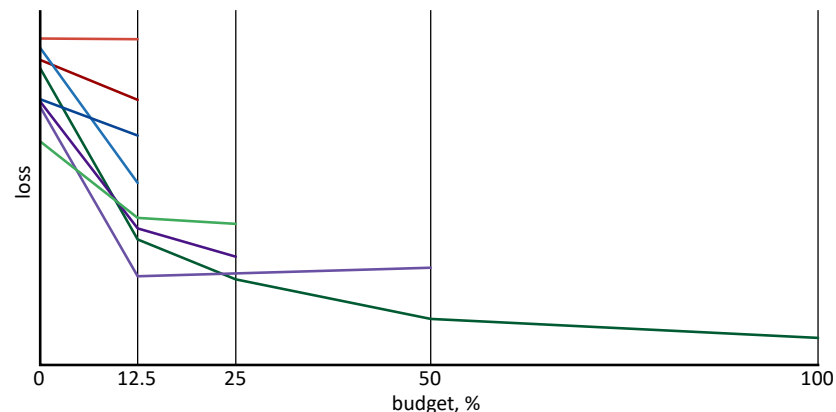


Рисунок 5. Иллюстрация метода последовательного деления

Программные пакеты



Для автоматической оптимизации гиперпараметров используем фреймворк **Ray Tune**:

- **алгоритмы поиска:** поиск по сетке, **случайный поиск**, Ax, BayesOptSearch, **BOHB**, BlendSearch, CFO, Dragonfly, HEBO, **HyperOpt**, Optuna, SigOpt, Scikit-Optimize, ZOOpt;
- **планировщики:** **ASHA**, HyperBand, Median Stopping Rule, Population Based Training (PBT), Population Based Bandits (PB2), **BOHB**.

Сравнение алгоритмов оптимизации гиперпараметров на примере 5-мерной задачи



Таблица 2. Область поиска и ограничение по времени для всех алгоритмов НПО

N	qlograndint(64, 2048, 32)
L	randint(1, 5)
σ	ELU, sigmoid, tanh, sin, atan
$ N_f $	$10^4, 10^5, 10^6$
$ N_{\text{batch}} $	qlograndint(64, 32768, 32)
lr	qloguniform(1e-4, 1, 1e-4)
lr scheduler	None, ExponentialLR-0.95, ReduceLROnPlateau-0.1-10, ReduceLROnPlateau-0.5-2
t	300 сек
T	8 часов

Сравнение алгоритмов оптимизации гиперпараметров на примере 5-мерной задачи



Таблица 3. Лучшие конфигурации модели, полученные алгоритмами НПО

$ N_{cfg} $	№	i	t , сек	MSE_f	$RMSE_u$	N	L	σ	$ N_f $	$ N_{batch} $	lr	lr scheduler
Случайный поиск												
96	5	107	300.1	5.0E-02	6.3E-04	1728	1	sin	10^5	160	3.6E-01	ReduceLRonPlateau-0.5-2
	10	697	300.0	4.3E-01	1.3E-03	96	1	sin	10^5	1056	2.2E-01	ReduceLRonPlateau-0.1-10
	21	6	304.9	3.2E+00	1.3E-02	1088	3	sin	10^6	128	8.0E-04	None
Случайный поиск + ASHAScheduler												
1593	364	550	300.1	1.2E-02	2.6E-04	1216	1	sin	10^5	1760	7.9E-01	ReduceLRonPlateau-0.5-2
	581	1039	300.1	2.8E-02	2.7E-04	704	1	sin	10^5	4608	9.5E-01	ReduceLRonPlateau-0.1-10
	5	106	300.1	4.7E-02	4.4E-04	1728	1	sin	10^5	160	3.6E-01	ReduceLRonPlateau-0.5-2
HyperOptSearch												
96	90	277	300.2	3.3E-01	1.7E-03	768	2	sin	10^5	928	8.1E-03	ReduceLRonPlateau-0.5-2
	76	1130	300.1	4.3E-01	1.3E-03	416	1	sin	10^5	2816	2.1E-01	ReduceLRonPlateau-0.5-2
	81	250	300.2	5.3E-01	2.1E-03	928	2	sin	10^5	1408	2.0E-03	ReduceLRonPlateau-0.5-2
HyperOptSearch + ASHAScheduler												
1053	817	624	300.1	1.1E-02	1.7E-04	928	1	sin	10^5	2016	1.0E+00	ReduceLRonPlateau-0.5-2
	548	696	300.1	1.1E-02	2.3E-04	736	1	sin	10^5	1824	8.6E-01	ReduceLRonPlateau-0.5-2
	607	690	300.1	1.1E-02	1.7E-04	704	1	sin	10^5	1728	9.9E-01	ReduceLRonPlateau-0.5-2
BOHB												
1108	820	1695	200.0	4.1E-01	1.4E-03	544	2	sin	10^4	736	1.4E-02	ReduceLRonPlateau-0.1-10
	223	326	219.5	4.2E-01	1.2E-03	128	1	sin	10^5	736	4.3E-01	ReduceLRonPlateau-0.1-10
	1059	2561	219.0	4.3E-01	1.2E-03	160	1	sin	10^4	736	7.9E-01	ReduceLRonPlateau-0.1-10

Автоматическая оптимизация гиперпараметров — 2-мерная задача



Таблица 4. Область поиска и ограничение по времени для 2-мерной задачи

N	qlograndint(64, 2048, 32)
L	randint(1, 5)
σ	ELU, sigmoid, tanh, sin, atan
$ N_f $	$10^2, 10^3, 10^4$
$ N_{\text{batch}} $	qlograndint(64, 32768, 32)
lr	qloguniform(1e-4, 1, 1e-4)
lr scheduler	None, ExponentialLR-0.95, ReduceLROnPlateau-0.1-10, ReduceLROnPlateau-0.5-2
t	30 сек
T	1 час

Таблица 5. Лучшие конфигурации модели, полученные для 2-мерной задачи

$ N_{\text{cfg}} $	№	i	t , сек	MSE_f	RMSE_u	N	L	σ	$ N_f $	$ N_{\text{batch}} $	lr	lr scheduler
665	457	1306	30.0	5.6E-05	1.4E-05	416	2	sin	1000	640	2.8E-02	ReduceLROnPlateau-0.5-2
	594	1286	30.0	7.8E-05	1.1E-05	672	2	sin	1000	384	1.5E-02	ReduceLROnPlateau-0.5-2
	596	1720	30.0	8.8E-05	1.2E-05	640	2	sin	1000	544	2.0E-02	ReduceLROnPlateau-0.5-2

Автоматическая оптимизация гиперпараметров — 2-мерная задача

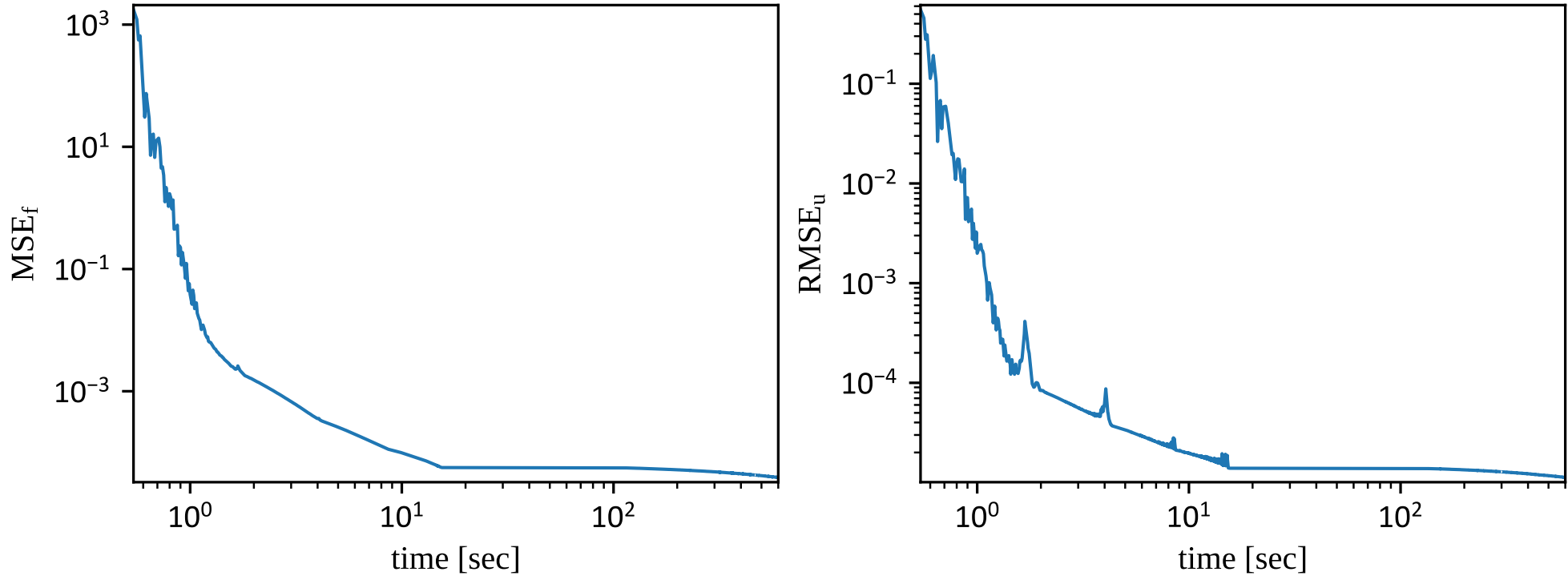


График 2.

Зависимость невязки по уравнению и невязки по решению от времени лучшей конфигурации для 2-мерной задачи

Автоматическая оптимизация гиперпараметров — 2-мерная задача



Конечные значения
невязок:

$$\text{MSE}_f = 3.885\text{E}-05$$

$$\text{RMSE}_u = 1.125\text{E}-05$$

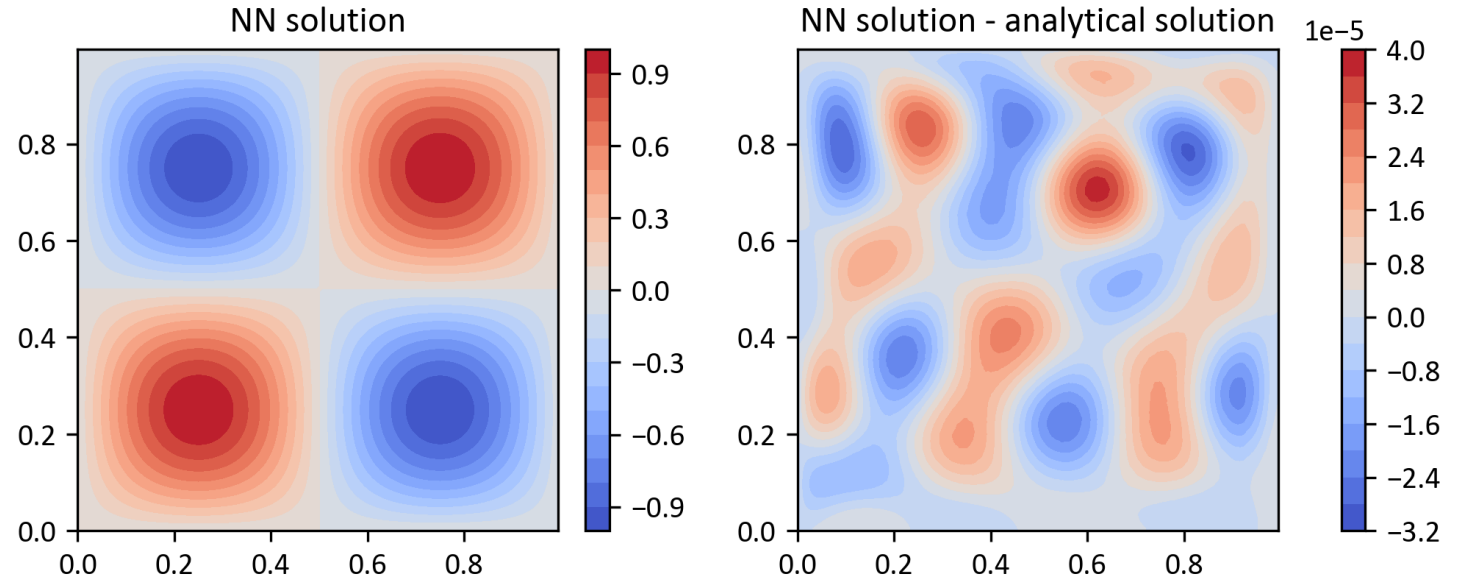


График 3.

Лучшее решение 2-мерной задачи и его отклонение от аналитического решения

Автоматическая оптимизация гиперпараметров — 2-мерная задача на области [0; 2]



Таблица 6. Область поиска и ограничение по времени для 2-мерной задачи на области [0; 2]

N	qlograndint(64, 2048, 32)
L	randint(1, 5)
σ	ELU, sigmoid, tanh, sin, atan
$ N_f $	$10^2, 10^3, 10^4$
$ N_{\text{batch}} $	qlograndint(64, 32768, 32)
lr	qloguniform(1e-4, 1, 1e-4)
lr scheduler	None, ExponentialLR-0.95, ReduceLROnPlateau-0.1-10, ReduceLROnPlateau-0.5-2
t	30 сек
T	1 час

Таблица 7. Лучшие конфигурации модели, полученные для 2-мерной задачи на области [0; 2]

$ N_{\text{cfg}} $	№	i	t , сек	MSE_f	RMSE_u	N	L	σ	$ N_f $	$ N_{\text{batch}} $	lr	lr scheduler
660	492	196	30.0	1.2E-03	2.4E-04	1504	2	sin	10^5	640	7.1E-03	ReduceLROnPlateau-0.5-2
	629	77	30.0	1.3E-03	2.0E-04	1568	2	sin	10^5	160	8.1E-03	ReduceLROnPlateau-0.5-2
	404	267	30.0	1.3E-03	1.6E-04	896	2	sin	10^5	608	1.2E-02	ReduceLROnPlateau-0.5-2

Автоматическая оптимизация гиперпараметров — 2-мерная задача на области $[0; 2]$

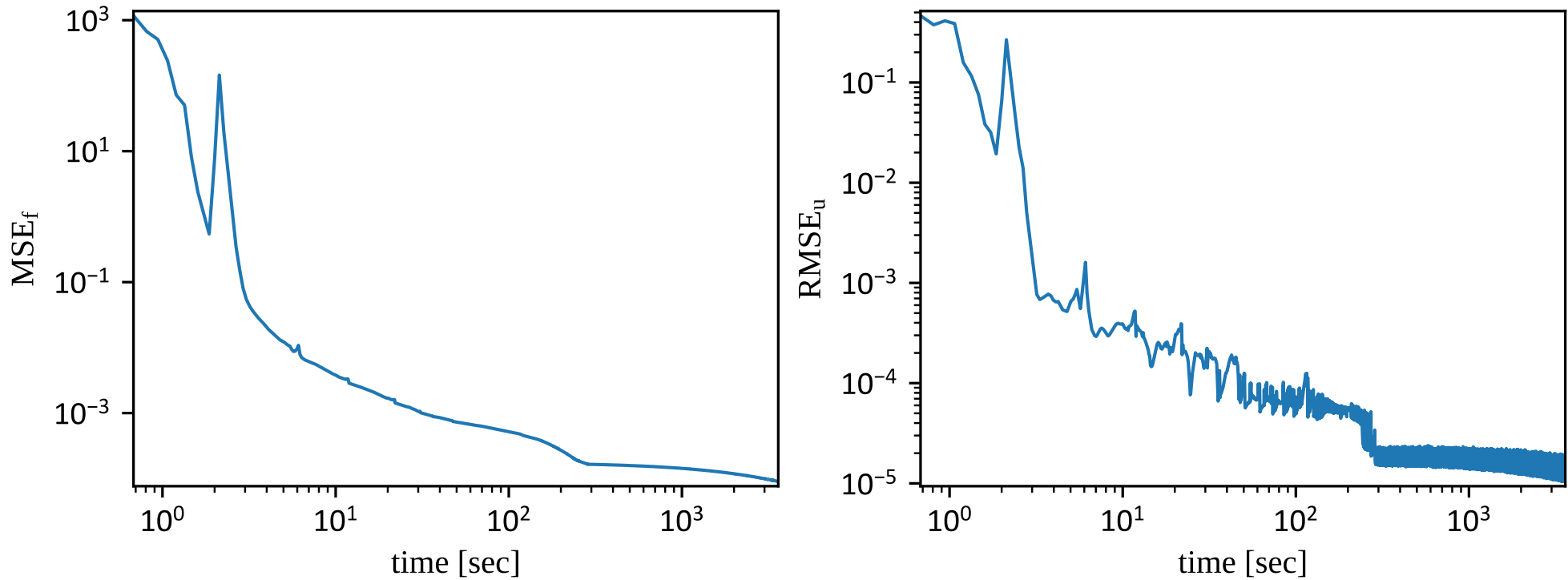


График 4. Зависимость невязки по уравнению и невязки по решению от времени лучшей конфигурации для 2-мерной задачи на области $[0; 2]$

Автоматическая оптимизация гиперпараметров — 2-мерная задача на области [0; 2]



Конечные значения
невязок:

$$\text{MSE}_f = 9.017\text{E}-05$$

$$\text{RMSE}_u = 1.153\text{E}-05$$

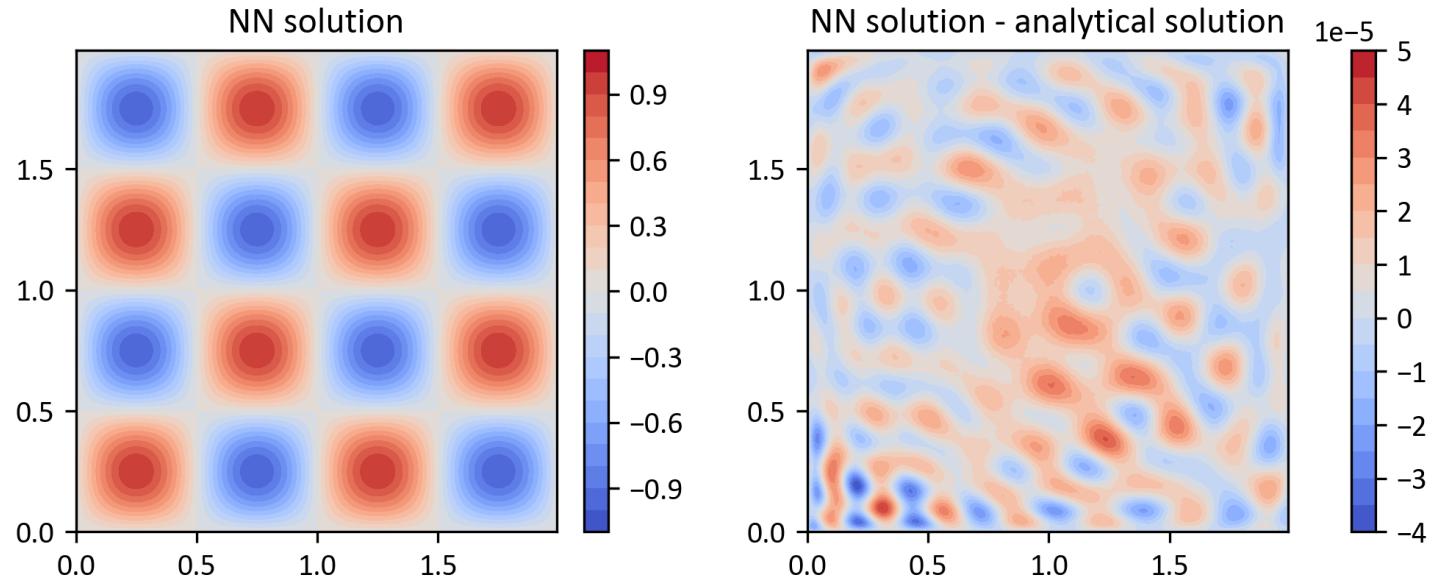


График 5.

Лучшее решение 2-мерной задачи и его отклонение от аналитического решения

Автоматическая оптимизация гиперпараметров — 5-мерная задача



Таблица 8. Область поиска и ограничение по времени для 5-мерной задачи

N	qlograndint(64, 2048, 32)
L	randint(1, 5)
σ	ELU, sigmoid, tanh, sin, atan
$ N_f $	$10^4, 10^5, 10^6$
$ N_{\text{batch}} $	qlograndint(64, 32768, 32)
lr	qloguniform(1e-4, 1, 1e-4)
lr scheduler	None, ExponentialLR-0.95, ReduceLROnPlateau-0.1-10, ReduceLROnPlateau-0.5-2
t	300 сек
T	8 часов

Таблица 9. Лучшие конфигурации модели, полученные для 5-мерной задачи

$ N_{\text{cfg}} $	№	i	t , сек	MSE_f	RMSE_u	N	L	σ	$ N_f $	$ N_{\text{batch}} $	lr	lr scheduler
1053	817	624	300.1	1.1E-02	1.7E-04	928	1	sin	10^5	2016	1.0E+00	ReduceLROnPlateau-0.5-2
	548	696	300.1	1.1E-02	2.3E-04	736	1	sin	10^5	1824	8.6E-01	ReduceLROnPlateau-0.5-2
	607	690	300.1	1.1E-02	1.7E-04	704	1	sin	10^5	1728	9.9E-01	ReduceLROnPlateau-0.5-2

Автоматическая оптимизация гиперпараметров — 5-мерная задача

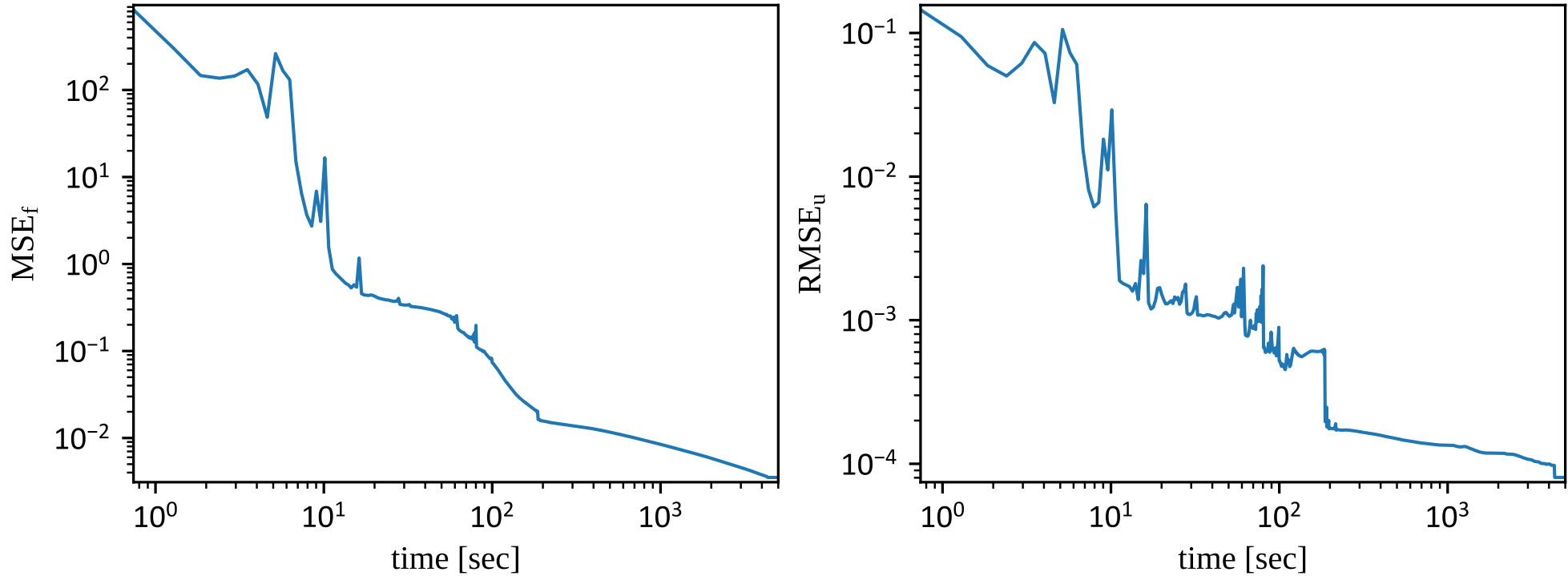


График 6.

Зависимость невязки по уравнению и невязки по решению от времени лучшей конфигурации для 5-мерной задачи

Автоматическая оптимизация гиперпараметров — 5-мерная задача



Конечные значения
невязок:

$$\text{MSE}_f = 3.518\text{E}-03$$

$$\text{RMSE}_u = 8.036\text{E}-05$$

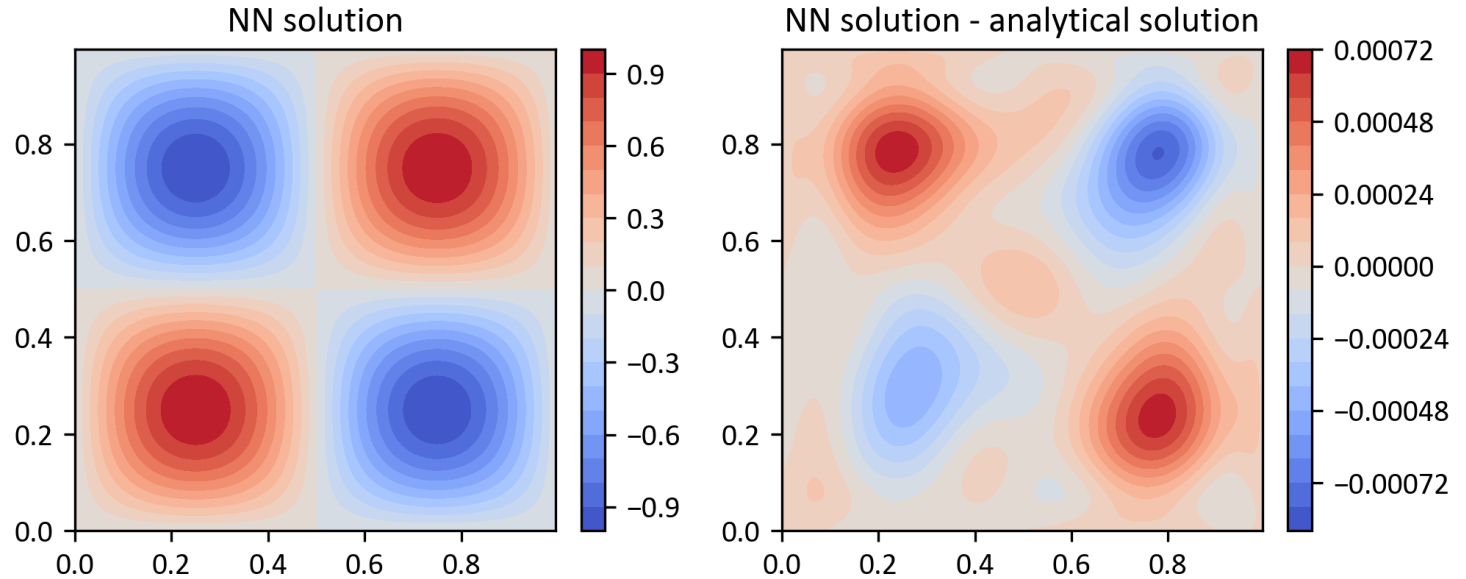


График 7.

Лучшее решение 5-мерной задачи и его отклонение от аналитического решения

Автоматическая оптимизация гиперпараметров — 8-мерная задача



Таблица 10. Область поиска и ограничение по времени для 8-мерной задачи

N	qlograndint(64, 2048, 32)
L	randint(1, 5)
σ	ELU, sigmoid, tanh, sin, atan
$ N_f $	$10^6, 10^7, 10^8$
$ N_{\text{batch}} $	qlograndint(64, 32768, 32)
lr	qloguniform(1e-4, 1, 1e-4)
lr scheduler	None, ExponentialLR-0.95, ReduceLROnPlateau-0.1-10, ReduceLROnPlateau-0.5-2
t	1800 сек
T	48 часов

Таблица 11. Лучшие конфигурации модели, полученные для 8-мерной задачи

$ N_{\text{cfg}} $	№	i	t , сек	MSE_f	RMSE_u	N	L	σ	$ N_f $	$ N_{\text{batch}} $	lr	lr scheduler
313	260	60.9	1802.1	3.3E+01	1.5E-02	2016	1	sin	10^6	160	6.8E-02	ExponentialLR-0.95
	262	60.2	1802.2	3.3E+01	1.5E-02	2016	1	sin	10^6	160	7.1E-02	ExponentialLR-0.95
	273	61.9	1803.0	3.9E+01	1.7E-02	1632	1	sin	10^6	160	7.7E-02	ExponentialLR-0.95

Автоматическая оптимизация гиперпараметров — 8-мерная задача

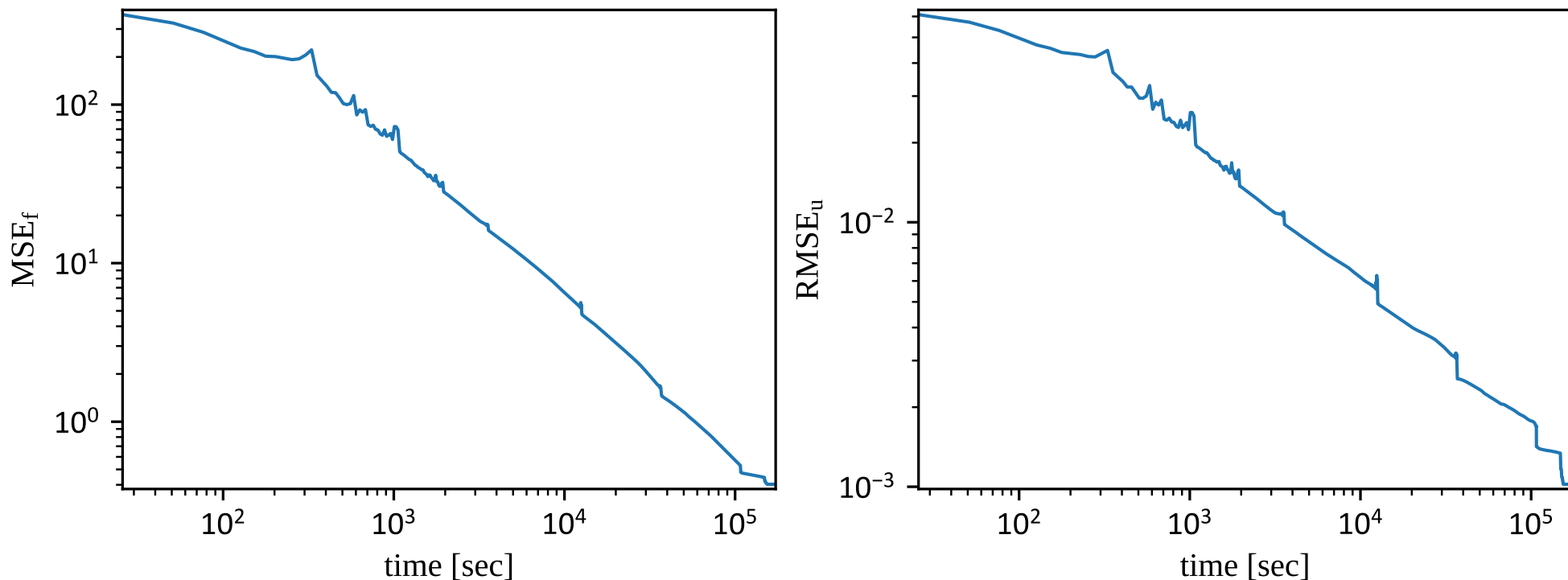


График 8.

Зависимость невязки по уравнению и невязки по решению от времени лучшей конфигурации для 8-мерной задачи

Автоматическая оптимизация гиперпараметров — 8-мерная задача



Конечные значения
невязок:

$$\text{MSE}_f = 4.030\text{E}-01$$

$$\text{RMSE}_u = 1.022\text{E}-03$$

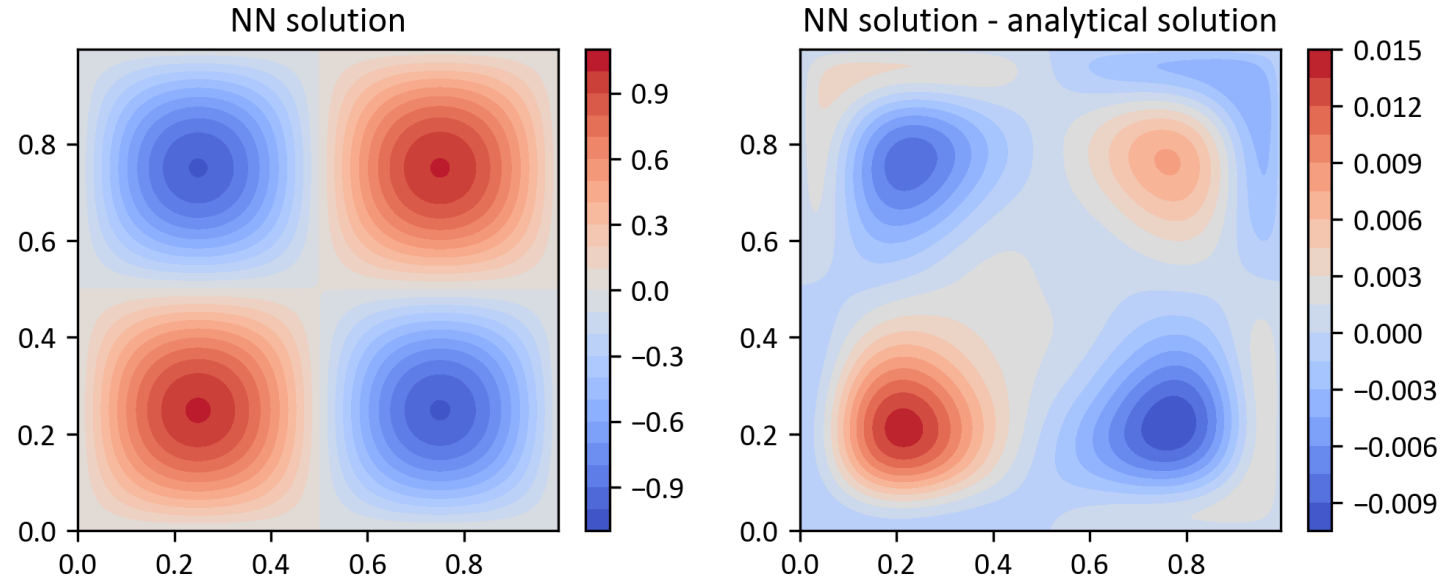


График 9.

Лучшее решение 8-мерной задачи и его отклонение от аналитического решения

Выводы



- В ходе работы удалось эффективно использовать средства автоматической оптимизации гиперпараметров (HPO) для подбора конфигурации модели, аппроксимирующей уравнение Гельмгольца. Использование алгоритма ранней остановки позволяет рассмотреть гораздо больше конфигураций, и целесообразно в первую очередь применять его, а для большей точности можно подключить и байесовский алгоритм поиска. Из рассмотренных алгоритмов HPO наилучший результат был получен при комбинации HyperOptSearch + ASHAScheduler.
- Посредством автоматической оптимизации гиперпараметров было получено более точное решение с меньшими ресурсными затратами, чем при ручной оптимизации. Таким образом, средства HPO можно эффективно применять при решении прикладных задач.
- В работе продемонстрирована возможность решения уравнения Гельмгольца посредством нейронных сетей в пространствах больших размерностей (до 8). Однако, при увеличении размерности задачи быстро растут потребности в вычислительных ресурсах, что смягчается возможностью автоматизации процесса.