

Лекция 3

Машинное обучение в вычислительной физике

Лю Шисян

Кафедра теплофизики (Э6)

https://sxliu.site/teaching/

План лекции

1. Персептрон и классификация
2. Нейронная сеть и прямой проход
3. Обучение и проверка модели
4. Прогноз теплопроводности по данным
5. Температурное поле из уравнения теплопроводности

Репозиторий кода

`https://github.com/lyushisyan/computational-methods-course`

1. **Персептрон и классификация**
2. Нейронная сеть и прямой проход
3. Обучение и проверка модели
4. Прогноз теплопроводности по данным
5. Температурное поле из уравнения теплопроводности

Одна строка наблюдений

Два выключателя управляют лампой. Она горит, когда включены оба. Единица означает «включён» или «горит», ноль означает «выключен» или «погашена».

Выключатель x_1	Выключатель x_2	Лампа y
0	0	0
0	1	0
1	0	0
1	1	1

Одна строка даёт один **пример**. Состояния выключателей служат входными **признаками**, а состояние лампы задаёт **метку класса** y. Ответ модели обозначим $\hat{y}$.

Как нейрон складывает входные сигналы

Персептрон умножает каждый признак на его **вес**, складывает произведения и прибавляет **смещение**:

$$z = w_1 x_1 + w_2 x_2 + b.$$

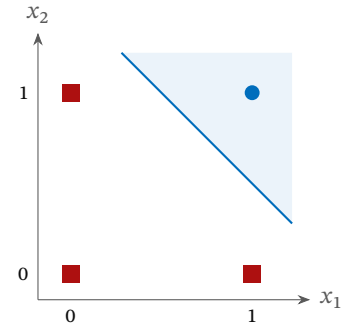
Возьмём $w_1 = 2,\ w_2 = 1,\ b = -1{,}5$.

Что включено	Вход	Значение z
Только первый выключатель	$(1, 0)$	$2 - 1{,}5 = 0{,}5$
Только второй выключатель	$(0, 1)$	$1 - 1{,}5 = -0{,}5$

Вклад первого сигнала вдвое больше. Если увеличить только b на единицу, значение z вырастет на единицу при любом входе.

Как знак суммы разделяет два класса

Положим $w_1 = w_2 = 1$ и $b = -1{,}5$. При $z > 0$ ответ равен 1, при $z \le 0$ он равен 0.



На **границе классов** взвешенная сумма со смещением равна нулю:

$$x_1 + x_2 = 1{,}5.$$

Для $(1, 0)$ получаем $z = -0{,}5$, лампа погашена.
Для $(1, 1)$ имеем $z = +0{,}5$, лампа горит.

Выше прямой лежит только синий круг. Так один нейрон реализует логическую операцию И.

Поправка после ошибки

Метка и ответ принимают значения 0 и 1. После прогноза вычислим разность $e = y - \hat{y}$.

Результат	e	Как изменить z для этого входа
Лампа должна гореть, ответ 0	$+1$	Увеличить
Лампа должна быть погашена, ответ 1	-1	Уменьшить
Ответ верен	0	Сохранить параметры

Правило обучения персептрона со скоростью обучения $\eta > 0$:

$$w_j \leftarrow w_j + \eta e x_j, \qquad b \leftarrow b + \eta e.$$

Один шаг обучения персептрона

Вернёмся к операции И. Начнём с $w_1 = w_2 = 1$, $b = -0{,}5$. Для входа $(1, 0)$ правильный ответ $y = 0$.

$$z = 1 + 0 - 0{,}5 = 0{,}5, \qquad \hat{y} = 1, \qquad e = 0 - 1 = -1.$$

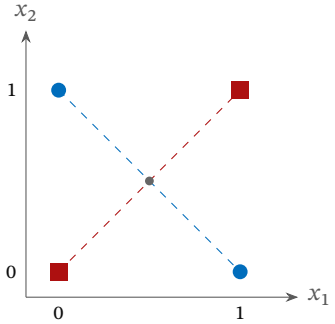
При $\eta = 0{,}5$ веса и смещение станут равны

$$w_1' = 1 - 0{,}5 \cdot 1 = 0{,}5, \quad w_2' = 1 - 0{,}5 \cdot 0 = 1, \quad b' = -0{,}5 - 0{,}5 = -1.$$

Подставим тот же вход. Теперь $z' = 0{,}5 + 0 - 1 = -0{,}5$ и ответ равен 0. Остальные строки таблицы тоже нужно проверить с новыми весами.

Почему одной прямой не хватает для XOR

Изменим схему. Пусть лампа горит, когда включён **ровно один** выключатель. Это операция «исключающее ИЛИ» (XOR).



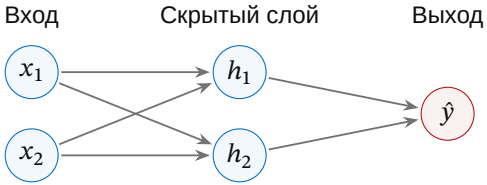
Синие круги означают, что лампа горит, красные квадраты —что погашена. Точки одного класса лежат на концах диагонали.

Чтобы прямая разделила классы, две диагонали должны целиком лежать по разные стороны от неё. Но у них общая середина.

1. Персептрон и классификация
2. **Нейронная сеть и прямой проход**
3. Обучение и проверка модели
4. Прогноз теплопроводности по данным
5. Температурное поле из уравнения теплопроводности

Два нейрона в скрытом слое

Включён ровно один выключатель, если включён хотя бы один и вместе с тем включено не больше одного.



Нейрон h_1 проверяет первое условие, h_2 —второе. Выходной нейрон получает оба ответа и зажигает лампу, только если оба равны 1.

Как получить ответ по состояниям выключателей

Пусть $s = x_1 + x_2$. Тогда $h_1 = 1$ при $s > 0{,}5$, а $h_2 = 1$ при $s < 1{,}5$. В остальных случаях соответствующий нейрон выдаёт 0.

Вход	Число замкнутых s	h_1	h_2	Ответ $\hat{y}$
$(0, 0)$	0	0	1	0
$(1, 0)$ или $(0, 1)$	1	1	1	1
$(1, 1)$	2	1	0	0

Выходной слой выполняет знакомую операцию И. Для этого он проверяет, положительна ли величина $h_1 + h_2 - 1{,}5$.

Послойный расчёт в NumPy

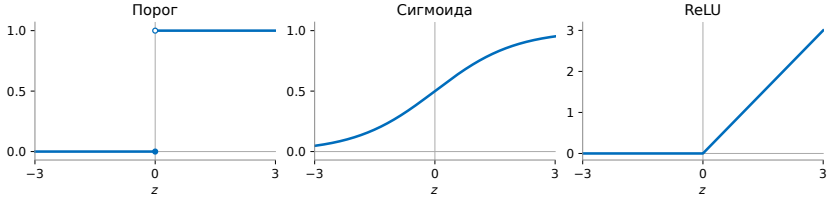
Запишем четыре входные пары по строкам массива X. Каждый столбец матрицы весов соответствует одному скрытому нейрону.

```
import numpy as np
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
W = np.array([[1, -1], [1, -1]])
b = np.array([-0.5, 1.5])
H = (X @ W + b > 0).astype(int)
y_hat = (H @ [1, 1] - 1.5 > 0).astype(int)
print(y_hat)
```

В первом столбце сравниваем с нулём $s - 0{,}5$, во втором $-1{,}5 - s$. Получаем четыре ответа `[0 1 1 0]`.

Что меняет функция активации

Сначала нейрон вычисляет сумму z, а в следующий слой передаёт $h = f(z)$. Функцию f называют **функцией активации**.



Пороговая функция даёт 0 или 1. ReLU сохраняет положительные значения и обнуляет отрицательные: $\mathrm{ReLU}(-2) = 0$, $\mathrm{ReLU}(2) = 2$.

Сколько ни повторяй взвешенную сумму со смещением, получится одна такая же операция. Нелинейная активация позволяет скрытому слою строить более сложные зависимости.

Небольшая сеть для прогноза числа

Возьмём один вход x, один скрытый нейрон и один выходной. В скрытом слое используем ReLU, а на выходе получим вещественное число.

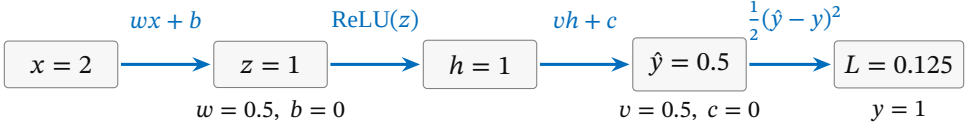
$$\underbrace{z = wx + b, \quad h = \max(0, z)}_{\text{скрытый слой}} \qquad \underbrace{\hat{y} = vh + c}_{\text{выходной слой}} .$$

Здесь w и v —веса двух слоёв, b и c —их смещения. Пока зафиксируем параметры:

$$w = 0{,}5, \qquad b = 0, \qquad v = 0{,}5, \qquad c = 0.$$

Прямой проход от входа до потерь

Пусть $x = 2$ и известен правильный ответ $y = 1$. Идём по синим стрелкам слева направо. В каждом прямоугольнике записан результат одного вычисления.



Прогноз $\hat{y} = 0{,}5$ ниже цели на $0{,}5$. Оценим отклонение с помощью **квадратичной функции потерь**:

$$L = \frac{1}{2}(\hat{y} - y)^2 = \frac{1}{2}(0{,}5 - 1)^2 = 0{,}125.$$

Как получить прогноз для нового входа

Подадим три значения $x = 1, 2, 4$, сохранив прежние веса и смещения. NumPy выполняет для каждого входа тот же прямой проход.

```
import numpy as np
x = np.array([1.0, 2.0, 4.0])
z = 0.5*x
h = np.maximum(0, z)
y_hat = 0.5*h
print(y_hat)                          # [0.25 0.5  1.  ]
```

Для второго входа известен ответ. Прогноз снова занижен. Как изменить веса и смещения обоих слоёв, чтобы уменьшить потери?

1. Персептрон и классификация
2. Нейронная сеть и прямой проход
3. **Обучение и проверка модели**
4. Прогноз теплопроводности по данным
5. Температурное поле из уравнения теплопроводности

Как измерить ошибку одного прогноза

Вернёмся к сети из предыдущего раздела. При $x = 2$ скрытый нейрон дал $h = 1$, а сеть выдала $\hat{y} = 0{,}5$. Для этого примера известен правильный ответ $y = 1$.

Отклонение прогноза равно $\hat{y} - y = -0{,}5$. Измерим его величину с помощью **квадратичной функции потерь**:

$$L = \frac{1}{2}(\hat{y} - y)^2 = \frac{1}{2}(0{,}5 - 1)^2 = 0{,}125.$$

Для прогнозов $0{,}8$ и $1{,}2$ потери одинаковы и равны $0{,}02$. Первый прогноз нужно повысить, второй понизить. Как определить направление изменения по функции потерь?

Что показывает производная потерь

Представим, что выход сети $\hat{y}$ можно немного изменить. Целевое значение по-прежнему равно единице. При $\hat{y} = 0{,}5$ наклон кривой потерь равен

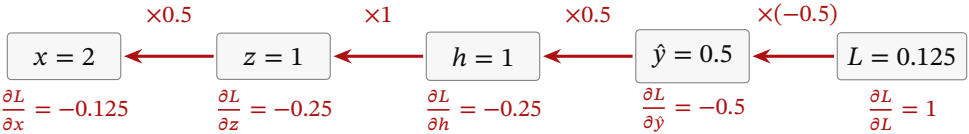
$$\frac{\partial L}{\partial \hat{y}} = \hat{y} - y = -0{,}5.$$

Увеличим прогноз на $0{,}01$. По производной ожидаем изменение потерь примерно на $(-0{,}5) \cdot 0{,}01 = -0{,}005$.

Проверим подстановкой. При прогнозе $0{,}51$ потери равны $0{,}12005$. Они изменились на $-0{,}00495$. Значит, небольшой рост прогноза здесь уменьшает потери.

Обратный проход по тому же графу

Начнём с потерь. Проходя через каждую операцию, умножаем уже найденную производную на локальную производную этой операции. Это **цепное правило**.



На выходе скрытого слоя получаем $\partial L/\partial h = (-0{,}5)\upsilon = -0{,}25$. Здесь $z = 1 > 0$, наклон ReLU равен единице, поэтому $\partial L/\partial z = -0{,}25$.

Производные по весам и смещениям

На выходе $\hat{y} = vh + c$, в скрытом слое $z = wx + b$. Из прямого прохода известны значения $h = 1$ и $x = 2$.

Параметр	Множитель	Производная потерь
Вес выхода v	$h = 1$	$\partial L/\partial v = (-0{,}5) \cdot 1 = -0{,}5$
Смещение c	1	$\partial L/\partial c = (-0{,}5) \cdot 1 = -0{,}5$
Вес скрытого слоя w	$x = 2$	$\partial L/\partial w = (-0{,}25) \cdot 2 = -0{,}5$
Смещение b	1	$\partial L/\partial b = (-0{,}25) \cdot 1 = -0{,}25$

Соберём четыре частные производные в порядке параметров. Полученный вектор называется **градиентом**.

Что даст один шаг обучения

Возьмём **скорость обучения** $\eta = 0{,}1$. Из каждого параметра вычтем его производную, умноженную на η. Например, $w = 0{,}5 - 0{,}1(-0{,}5) = 0{,}55$.

$$(w, b, v, c): \quad (0{,}5,\ 0,\ 0{,}5,\ 0) \longrightarrow (0{,}55,\ 0{,}025,\ 0{,}55,\ 0{,}05).$$

Повторим прямой проход для того же входа:

$$h = \mathrm{ReLU}(0{,}55 \cdot 2 + 0{,}025) = 1{,}125, \qquad \hat{y} = 0{,}55 \cdot 1{,}125 + 0{,}05 = 0{,}66875.$$

Прогноз приблизился к единице. Потери уменьшились с $0{,}125$ до $0{,}05486$.

Проверка четырёх производных в PyTorch

Запишем те же параметры в одном тензоре и повторим расчёт:

```
import torch
p = torch.tensor([0.5, 0.0, 0.5, 0.0], requires_grad=True)
w, b, v, c = p
h = torch.relu(w*2 + b)
prediction = v*h + c
loss = (prediction - 1)**2 / 2
loss.backward()
print(p.grad.tolist())
```

Вывод `[-0.5, -0.25, -0.5, -0.5]` в порядке w, b, v, c.

От одного примера к выборке наблюдений

Теперь подберём прямую $\hat{y} = ax + d$ по данным с шумом. Равномерно выберем 160 значений x в $[-2, 2]$ и вычислим $y = 2x - 1 + \varepsilon$. Стандартное отклонение шума равно $0{,}3$.

До обучения случайно разделим 160 независимых наблюдений:

Выборка	Точек	Для чего её используем
Обучающая	96	Обновляем наклон a и свободный член d.
Валидационная	32	Сравниваем три скорости обучения.
Тестовая	32	Оцениваем ошибку выбранной модели на новых точках.

Одно обновление и одна эпоха

Соберём B обучающих точек в одну группу, или **мини-батч**. Каждая точка даёт своё отклонение. Возьмём средние потери в группе:

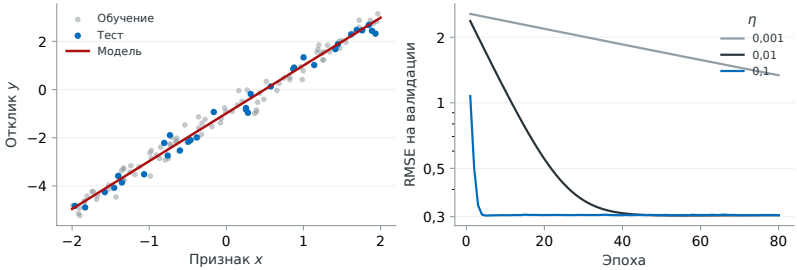
$$L_B = \frac{1}{2B} \sum_{i=1}^{B} (\hat{y}_i - y_i)^2.$$

Рассчитаем прогнозы, средние потери и их градиент. Обновим параметры один раз. К следующей группе переходим уже с новыми параметрами.

В нашем опыте в группе 16 точек. Полный проход по 96 обучающим точкам называют **эпохой**. За эпоху получаем $96/16 = 6$ обновлений, за 80 эпох их будет 480.

Проверка на отложенных точках

На правом графике сравним ошибки валидации в конце 80-й эпохи. Из трёх скоростей обучения выбираем $\eta = 0{,}1$. Слева показана полученная прямая.



Для прямой $\hat{y} = 1{,}984x - 0{,}986$ среднеквадратичная ошибка на 32 тестовых точках равна $0{,}275$. Она сравнима со стандартным отклонением шума $0{,}3$.

1. Персептрон и классификация
2. Нейронная сеть и прямой проход
3. Обучение и проверка модели
4. **Прогноз теплопроводности по данным**
5. Температурное поле из уравнения теплопроводности

Теплопроводность по температуре и толщине

В прошлой лекции мы рассчитали теплопроводность плёнки при заданных условиях. Если нужны сотни пар (T, L), по части расчётов можно обучить **суррогатную модель**, которая быстро даст прогноз для остальных пар.

В учебном примере данные зададим формулой:

$$\kappa_{\text{ref}}(T, L) = \frac{200\, e^{-L/200}}{1 + 0{,}0005\,(T - 300)^2}.$$

Подставляем T в К, L в нм. При $(300, 100)$ получаем $121{,}3$ Вт/(м·К). Здесь зависимость от толщины задана формулой. Для кремния в прошлой лекции мы рассчитывали перенос тепла.

Какие точки оставим для проверки?

Выберем 1000 случайных пар (T, L) равномерно в пределах $100–600$ К и $10–500$ нм. До обучения разделим их на три группы.

Выборка	Точек	Для чего они нужны
Обучающая	700	Подбираем веса сети.
Валидационная	150	Выбираем лучшую эпоху.
Тестовая	150	Оцениваем ошибку на новых парах (T, L).

На вход подаём $T' = (T - 100)/500$ и $L' = (L - 10)/490$. Целевое значение $y = \kappa_{\text{ref}}/100$. Чтобы получить теплопроводность, выход сети умножаем на 100.

Один шаг обучения в PyTorch

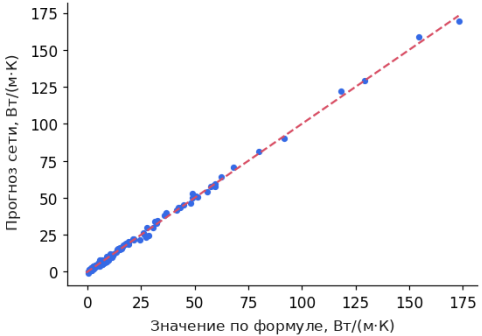
Два входа, два скрытых слоя по 32 нейрона с активацией ReLU и один выход $\hat{y}$. Возьмём очередную группу обучающих точек.

```
prediction = model(X[batch])
loss = (prediction - y[batch]).square().mean()

optimizer.zero_grad()
loss.backward()
optimizer.step()
```

За один шаг используем до 64 точек, за эпоху проходим все 700. После каждой эпохи считаем ошибку на валидационной выборке. Из 2000 эпох наименьшую ошибку дала 747-я. Её веса и возьмём.

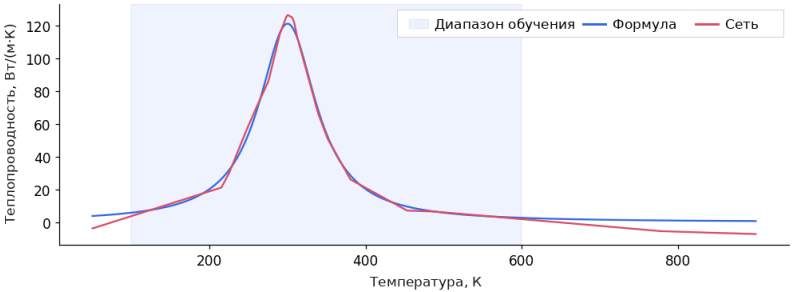
Ошибка прогноза на новых парах параметров



Точка соответствует одной тестовой паре. По горизонтали отложено значение по формуле, по вертикали — прогноз. Выше диагонали прогноз завышен.

На 150 точках средняя абсолютная ошибка равна $0{,}921$, а среднеквадратическая (RMSE) — $1{,}296$ Вт/(м·К).

Прогноз за пределами обучающего диапазона

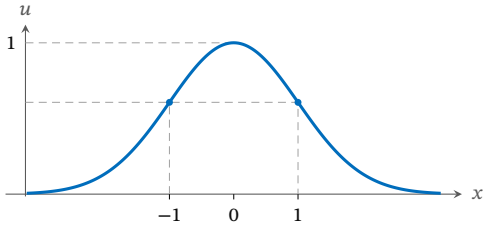


Зафиксируем $L = 100$ нм. В диапазоне обучения кривые близки. При дальнейшем нагреве сеть предсказывает отрицательную теплопроводность, хотя заданная функция остаётся положительной.

1. Персептрон и классификация
2. Нейронная сеть и прямой проход
3. Обучение и проверка модели
4. Прогноз теплопроводности по данным
5. **Температурное поле из уравнения теплопроводности**

Локальный нагрев в центре стержня

Длинный однородный стержень сначала имел одинаковую температуру по всей длине. За $t = 0$ примем момент выключения нагрева, а за $x = 0$ — центр нагретого участка.



Начальный гауссов профиль

$$u(x, 0) = e^{-x^2/2}.$$

Здесь u — безразмерный избыток температуры. В центре он равен 1, при $x = \pm 1$ — примерно $0{,}607$.

Как быстро остывает центр и как далеко успевает распространиться тепло?

Как уравнение ограничивает прогноз сети?

В единицах времени тепловой диффузии уравнение принимает вид $u_t = u_{xx}$. В вершине начального пика $u_{xx} < 0$, поэтому температура там начинает снижаться.

Физически информированная нейронная сеть (PINN) по координате и времени (x, t) выдаёт температуру $u_\theta(x, t)$. Для точки внутри стержня вычислим **невязку уравнения**

$$r_\theta = \frac{\partial u_\theta}{\partial t} - \frac{\partial^2 u_\theta}{\partial x^2}.$$

Проверим $u_\theta \equiv 0$. Невязка всюду равна нулю, а начальный максимум равен 0 вместо 1. Для решения нужны ещё начальное и граничные условия.

Три группы точек для обучения

Возьмём область $-5 \le x \le 5$, $0 \le t \le 1$. Для каждой группы точек возведём отклонения в квадрат и усредним.

Где заданы точки	Что проверяем
Внутри области	Невязка r_θ близка к нулю.
При $t = 0$	Прогноз $u_\theta(x, 0)$ близок к $e^{-x^2/2}$.
На границах $x = \pm 5$	Прогноз $u_\theta(\pm 5, t)$ близок к заданной температуре $g(t)$.

Например, если сеть даёт $u_\theta(0, 0) = 0{,}8$, квадрат ошибки в этой начальной точке равен $(0{,}8 - 1)^2 = 0{,}04$.

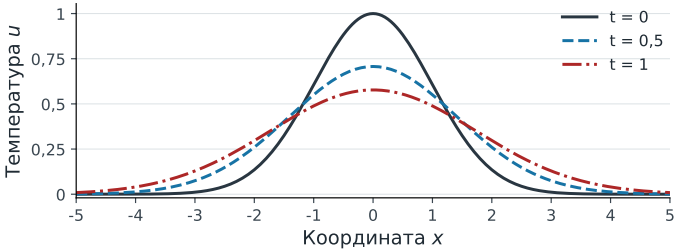
По чему берём производные при обучении PINN?

Сначала берём производные по **входным координатам**. Два столбца массива `xt` содержат координату x и время t.

```
xt = xt.detach().requires_grad_(True)
u = model(xt)
def grad(a):
    return torch.autograd.grad(a.sum(), xt, create_graph=True)[0]
du = grad(u)
ux, ut = du[:, :1], du[:, 1:]
r = ut - grad(ux)[:, :1]
```

Из невязки и отклонений от заданных температур составляем потери. Затем `loss.backward()` вычисляет градиенты по **весам и смещениям**, чтобы оптимизатор мог их обновить.

Высота и ширина теплового пика



При диффузии в длинном стержне точный профиль сохраняет гауссову форму. Его высота и ширина равны

$$u(0,t) = \frac{1}{\sqrt{1+2t}},$$

$$\sigma(t) = \sqrt{1+2t}.$$

При $t = 1$ высота равна $0{,}577$, а ширина в $1{,}73$ раза больше начальной.

Итоги лекции

Прямой проход даёт прогноз. При обратном проходе вычисляют градиент потерь по весам. Оптимизатор использует этот градиент, чтобы обновить веса.

	Теплопроводность	**Температура стержня**
Обучение	Известные значения при заданных параметрах	Уравнение, начальное и граничные условия
Проверка	Сравнение с опорными значениями в отложенных точках	Сравнение с точным решением в тех же точках пространства и времени

В тестовой выборке ошибка прогноза теплопроводности мала. За пределами изученных температур та же сеть дала отрицательное значение. Поэтому важно, где именно измерена ошибка.