# Обучение нейронной сети: эксперимент

*Карта закрашена почти идеально: каждая известная точка на своём фоне. Можно ли теперь отправлять робота в неизведанные места?*

Допустим, после очередного обучения карта выглядит почти идеально: синие точки окружены синим фоном, оранжевые — оранжевым. Можно ли теперь отправлять робота в неизведанные места?

Красивой границы для ответа мало. Мы хотели получить правило для новых мест, но подбирали веса по уже посещённым. Сеть могла уловить общую закономерность, а могла слишком старательно повторить случайные особенности этих посещений. Различить эти случаи нам поможет часть журнала, которую мы до сих пор почти не обсуждали.

**Ранее**

Веса меняются во время обучения. Скрытые слои с нелинейной активацией позволяют строить более сложные границы между классами. В первых опытах мы уже обучали такие сети. Теперь сравним архитектуры и входные признаки, а результат оценим на отложенных точках.

← Полносвязные слои и многослойные сети

← Градиентный спуск

## 1. Проверка на новых примерах

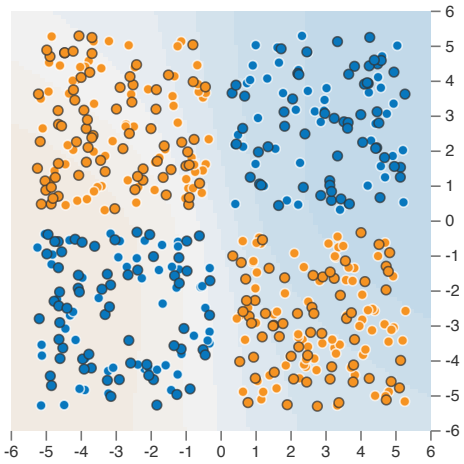
С самого первого опыта лаборатория делила точки пополам. По одной половине алгоритм менял веса. Метки второй половины оставались известны нам, но в эти изменения не вмешивались. Это **проверочные примеры**: для сети они такие же незнакомые места, ради которых мы и начали работу. Мы можем сопоставить её ответ с известным результатом, не отправляя туда робота вслепую.

Теперь становятся важны оба числа справа от сети. **Ошибка на обучении** показывает, насколько хорошо сеть подстроилась под доступные ей примеры. **Ошибка на проверке** — насколько хорошо найденное правило сработало на отложенных точках. Это значения функции потерь, а не процент неверных ответов. Маленькая учебная ошибка сама по себе ничего не обещает второй половине карты.

На новой карте «Чередование» безопасные участки расположены по диагонали. Одной прямой опять недостаточно: она не сможет оставить оба диагональных участка одного класса по одну сторону, а оба участка другого — по другую. Поэтому в лаборатории подготовлены два скрытых слоя по четыре нейрона. Сейчас нас интересует другое: будет ли улучшение на знакомых местах сопровождаться улучшением на незнакомых?

**Попробуйте.** Предскажите, как будут меняться две ошибки, запустите обучение и наблюдайте за обеими. После 500 проходов нажмите «Записать результат» и сопоставьте наблюдение с прогнозом.

# Самостоятельная лаборатория




Чередование. Синий класс — безопасно, оранжевый — опасно.

Состояние опыта: 0 проходов. Нейронов в скрытых слоях: 4 → 4.

Ошибка на обучении: 0,554; на проверке: 0,518.

Опыт 4. «Чередование», два скрытых слоя по четыре нейрона. Сначала наблюдаем две ошибки; затем здесь же сравним устройство сети и входные признаки.

*Интерактивный эксперимент — в электронной версии урока.*

Если обе ошибки уменьшаются, найденное правило помогает и в местах, на которых веса не подбирались. Если падает только учебная, этого вывода уже сделать нельзя. Именно для такого различия мы и сохранили часть наблюдений в стороне. Но пока перед нами лишь одна карта и одно обучение. Проверим, что изменится, если наблюдения станут менее однозначными, а сеть — более гибкой.

> Ошибка на обучении уменьшается, но на проверочных точках остаётся высокой. Что можно заключить?
>
> ○ Сеть уже надёжна во всех местах карты
>
> ○ Проверочные метки нужно передать в обучение, чтобы оценка стала честной
>
> ○ Проверочные точки не имеют отношения к нашей задаче
>
> ○ Сеть лучше подстроилась под знакомые примеры; улучшение на новых пока не подтверждено

**Заметка**

Меняйте сеть кнопками «+» и «−» на её схеме, карту — миниатюрами слева, признаки — непосредственно на их рисунках. «Записать результат» останавливает обучение и сохраняет условия вместе с обеими ошибками. Семь признаков и остальные настройки оригинальной лаборатории доступны.

## 2. Шум и переобучение

До сих пор отметки довольно ясно подсказывали общую форму участков. Но представим, что координаты в журнале измерены неточно: рядом могут оказаться записи с разными исходами. В лаборатории такую неоднозначность добавляет настройка **«Шум»**. Теперь граница, старающаяся угодить каждому учебному примеру, рискует превратиться в набор мелких изгибов.

Большая сеть умеет строить более сложные границы. Для спиралей такая гибкость нужна: граница там много раз меняет направление. Но здесь та же способность может сыграть против нас. Если мелкие изгибы повторяют случайные особенности учебных отметок, они не помогут предсказывать новые. Как это заметить? Снова посмотреть на отложенную часть журнала.

**Попробуйте.** На «Чередовании» оставьте $x_1$ и $x_2$, установите шум 20 и долю обучения 30%. Сравните один скрытый слой из четырёх нейронов с двумя слоями по восемь: по 500 проходов, одинаковые точки и остальные настройки. Запишите обе ошибки для каждого варианта.

Может оказаться, что большая сеть точнее повторяет учебные точки, но хуже оценивает проверочные. Такой разрыв между подстройкой под обучение и качеством на новых примерах — признак **переобучения**. Дополнительные изгибы не принесли полезного знания о карте. Но это не обязательный исход: если обе ошибки снизились, большая сеть помогла в этом опыте. Запишите именно то, что произошло, и рассмотрите построенную границу.

Повторение нужно, чтобы не принять удачный старт за свойство всей архитектуры. Кнопка ↺ начинает обучение с других весов на прежних точках. «Новые точки» меняет сами данные: после неё нужно снова испытать оба варианта на общем новом наборе. Сохранилось ли преимущество или результаты поменялись местами?

> На одних проверочных точках у сети А ошибка 0,20, у сети Б — 0,06. Какой вывод допустим?
>
> ○ Сеть А совсем не обучалась
>
> ○ Проверочные примеры теперь можно удалить
>
> ○ Сеть Б лучше прошла эту конкретную проверку
>
> ○ Сеть Б будет лучше на любой будущей карте

**Заметка**

Близость двух ошибок ещё не означает хорошую модель: обе могут быть большими. Нас интересует, насколько мала ошибка на новых примерах и насколько устойчив результат. Для настоящего робота нужно было бы отдельно учитывать цену опасной ошибки; здесь мы исследуем учебный классификатор.

## 3. Представление данных

Мы пробовали менять количество вычислений внутри сети. Но есть ещё один путь: иначе описать место, которое ей предстоит оценить. Вернёмся к **кольцу**. Для круглого острова важно удаление от центра; сами по себе направления «вправо» и «влево» не делают место безопаснее. А выходной нейрон по-прежнему получает две координаты со знаками.

Попробуем заранее преобразовать их так, чтобы это сходство стало заметнее. Если возвести число в квадрат, −3 и 3 превратятся в 9. Знак потеряется, зато одинаковое удаление от центра по каждой оси будет описываться одинаково. Более того, сумма квадратов двух координат одинакова во всех точках одной окружности. Сравнение этой суммы с порогом уже позволяет отделить внутреннюю область от внешней — даже без скрытых нейронов.

**Попробуйте.** Выберите кольцо без шума, верните долю обучения 50%, уберите скрытые слои и поставьте шаг 0.003. На одних точках сравните два варианта: сначала только $x_1$ и $x_2$, затем только $x_1^2$ и $x_2^2$. Дайте обоим по 1000 проходов и запишите результаты.

Получившаяся окружность не опровергает ограничение одного нейрона. Прямая граница относилась к исходным координатам. Сейчас нейрон применяет простое правило к **преобразованным признакам**, а на исходной карте это правило очерчивает круг. Мы сами выбрали представление, в котором задача стала проще. Скрытые слои тоже создают новые представления, но их параметры подбираются по примерам.

Вернёмся к вопросу о новых местах: посмотрите в двух записях не только на форму границы, но и на ошибку на проверке. Удачное представление позволяет решить эту задачу более простым правилом, без добавления скрытых нейронов. Это ещё один вариант, который стоит испытать прежде, чем усложнять сеть. Сам по себе выбор квадратов не доказывает надёжность: её по-прежнему оцениваем на отложенных точках.

**Заметка**

После возведения в квадрат входные числа крупнее, поэтому здесь выбран меньший шаг обучения. Если фон застыл почти одноцветным, начните заново кнопкой ↻. Возможность построить окружность и удачное нахождение весов — разные вещи. Сравнивайте варианты при одинаковом шаге и числе проходов.

> Нейрон без скрытых слоёв разделил кольцо после включения квадратов координат. Почему это возможно?
>
> - ○ Он тайно добавил себе скрытый слой
> - ○ Число проходов отменило ограничения любого нейрона
> - ○ Цвета точек поменяли правильные ответы
> - ○ Изменилось представление входных данных

Так можно ли доверить роботу новые места? Теперь у нас есть способ обосновать ответ: оценить ошибку на отложенных точках, посмотреть на форму границы и проверить устойчивость результата в повторных запусках. Хорошие результаты поддерживают ожидание, что правило пригодится на похожих местах этой карты. Они не гарантируют каждое предсказание и ничего не доказывают о другой карте. Вместо уверенности по красивой картинке у нас появилась проверка, на которую можно опереться.

### *Выбор модели*

Выберите вариант для одной карты и обоснуйте его результатами на отложенных точках. Объясните, что помогло: другая архитектура, представление входов или обучение. Если опыт не позволяет их различить, укажите, какое сравнение нужно провести следующим.

Классификатор оценивает новое место по его входным признакам.

Архитектура, представление данных и длительность обучения — разные условия опыта.

Качество на учебных примерах не заменяет проверку на новых.

**Домашняя работа**

1. Вернитесь к спиралям. Сравните один слой из восьми нейронов и два слоя по четыре при одинаковой длительности. В обоих вариантах восемь скрытых нейронов, но связи устроены по-разному. Сделайте по два запуска и напишите, какие выводы подтверждают наблюдения, а какие остались предположениями.

2. После добавления нейронов ошибка на обучении снизилась, а на тех же проверочных точках выросла. Что следует из наблюдения?

   Сеть лучше подстроилась под обучение, но хуже прошла эту проверку

   Нужно оценивать только обучение

   Добавление нейронов улучшило оба результата

   Проверочные точки участвовали в обновлении весов

### Источники

- Оригинальный тренажёр взят из TensorFlow Playground (https://playground.tensorflow.org/). Авторы: Daniel Smilkov и Shan Carter.
- Открытые исходники и лицензия Apache 2.0 (https://github.com/tensorflow/playground)

## Повторить эту тему