
СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

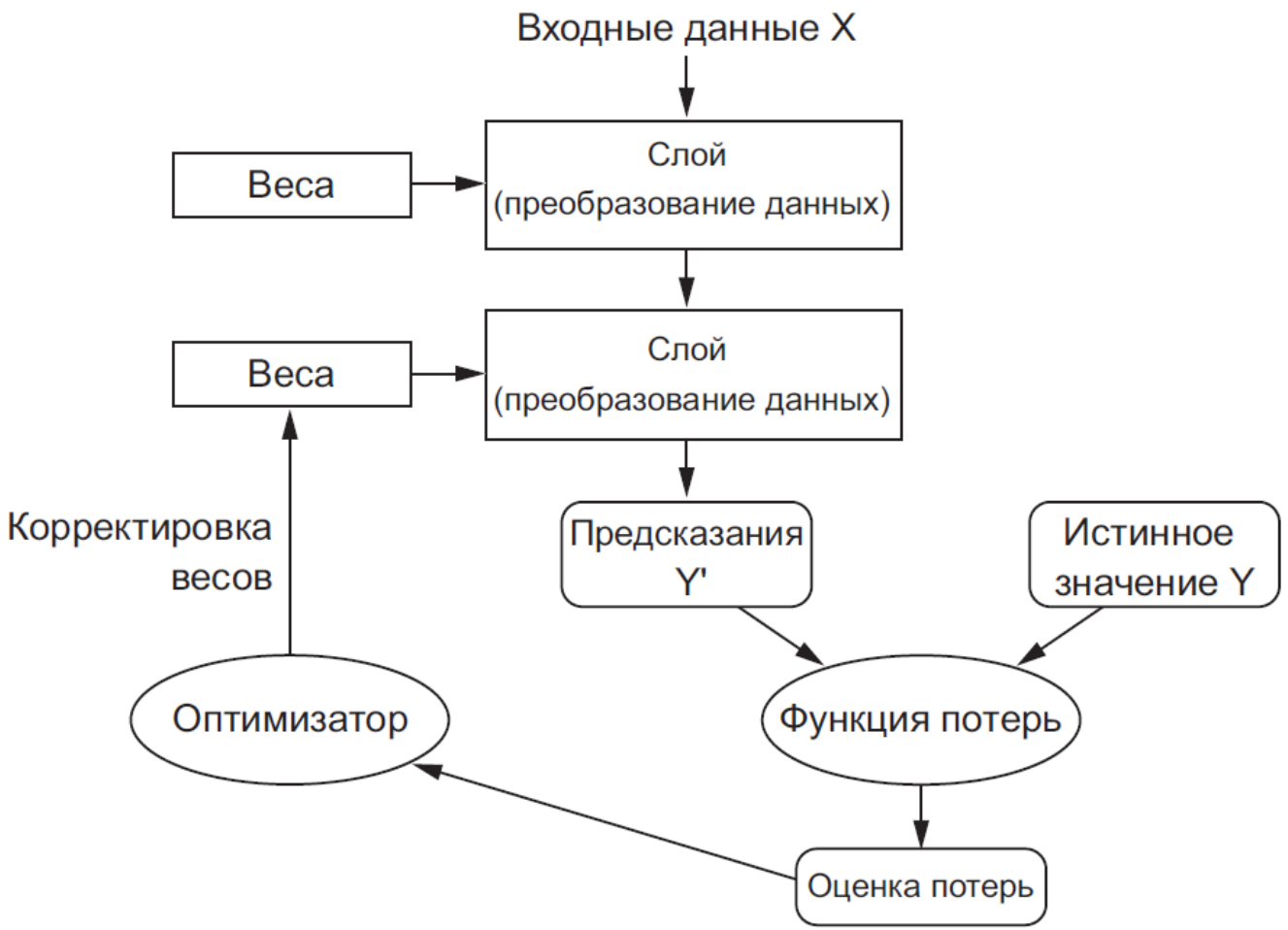
Александр Калиниченко

# ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ

Модуль 2. Методы искусственного интеллекта

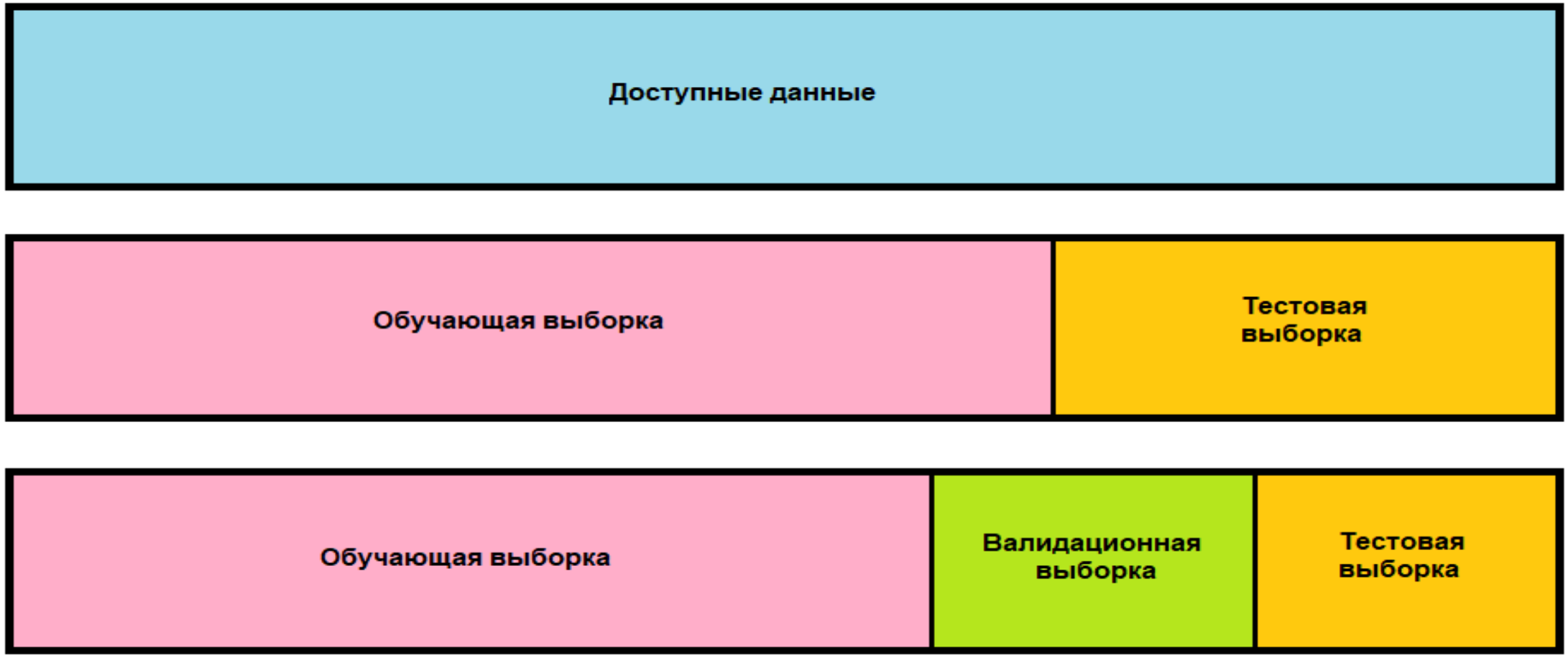
Лекция 11. Процесс разработки ИНС

# СТРУКТУРА ПРОЦЕССА ОБУЧЕНИЯ СЕТИ




- ***Слои***, параметризуемые весами и объединяемые в сеть (модель)
- ***Исходные данные*** и соответствующие им ***цели***
- ***Функция потерь***, которая определяет сигнал обратной связи, используемый для обучения
- ***Оптимизатор***, определяющий, как происходит обучение

# ВЫБОРКИ ДАННЫХ

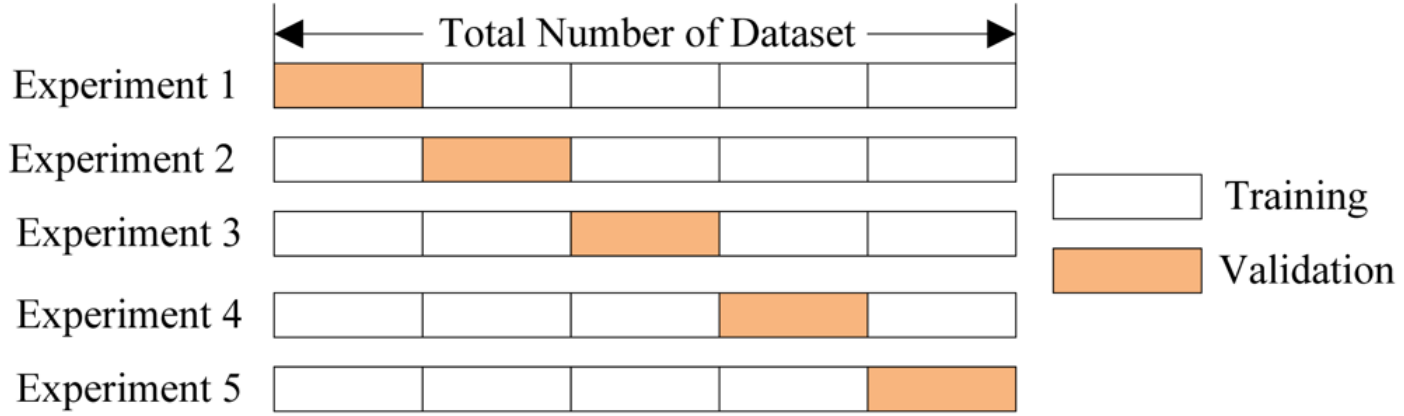


# ПОЧЕМУ ТРЕБУЮТСЯ ТРИ НАБОРА ДАННЫХ

- Процесс конструирования модели связан с настройкой ее **гиперпараметров** (тип и количество слоев, их размерности). Для настройки в качестве сигнала **обратной связи** используются **проверочные** (валидационные) данные. Фактически настройка сама является разновидностью обучения: поиск более удачной конфигурации в некотором пространстве параметров. Как результат, настройка конфигурации модели по качеству прогнозирования на проверочных данных может привести к **переобучению** на этих данных, хоть модель напрямую и не обучается на них
- Главной причиной этого является так называемая **утечка информации**. Каждый раз, настраивая гиперпараметры модели и опираясь на качество прогноза по проверочным данным, мы допускаем просачивание в модель некоторой информации из этих данных. Если повторить настройку много раз, в модель будет просачиваться все больший объем информации о проверочном наборе данных
- В результате получится модель, искусственно настроенная на достижение высокого качества прогнозирования по проверочным данным, потому что именно на этих данных она была оптимизирована
- Для корректного тестирования модель не должна иметь доступа ни к какой информации из контрольного набора, даже косвенно

# ДЕЛЕНИЕ ДАННЫХ НА ТРЕНИРОВОЧНЫЙ И ПРОВЕРОЧНЫЙ НАБОРЫ

- **Простое расщепление** выборки
- **Перекрестная проверка** по К блокам
- Итерационная **перекрестная проверка** по К блокам **с перемешиванием** (данные случайным образом перемешиваются перед каждым расщеплением)




Использование перекрестной проверки с перемешиванием позволяет снизить требования к объему обучающих данных. Однако эта процедура приводит к существенному росту вычислительных затрат

# ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ДАННЫХ

- **Векторизация**

  Представление входных данных в виде матриц специального формата (тензоров)

- **Нормализация значений**

  Приведение данных к одному масштабу (обычно в диапазоне от 0 до 1, либо нормализованные по СКО и с нулевым средним)

- **Обработка недостающих значений**

  Замена отсутствующих значений признаков нулями (при этом в обучающей выборке обязательно должны содержаться экземпляры с ненулевыми значениями соответствующих признаков)

# КОНСТРУИРОВАНИЕ ПРИЗНАКОВ

Варианты конструирования признаков для определения времени по циферблату часов

При удачном выборе признаков:

- Задача решается моделью меньшей емкости (с меньшим числом нейронов)
- Требуется значительно меньший объем обучающей выборки

| | | |
|---|---|---|
| Исходные данные: сетка с пикселами |  |  |
| Более удачные признаки: координаты стрелок | {x1: 0.7, y1: 0.7} {x2: 0.5, y2: 0.0} | {x1: 0.0, y2: 1.0} {x2: -0.38, 2: 0.32} |
| Еще более удачные признаки: углы отклонения стрелок | theta1: 45 theta2: 0 | theta1: 90 theta2: 140 |

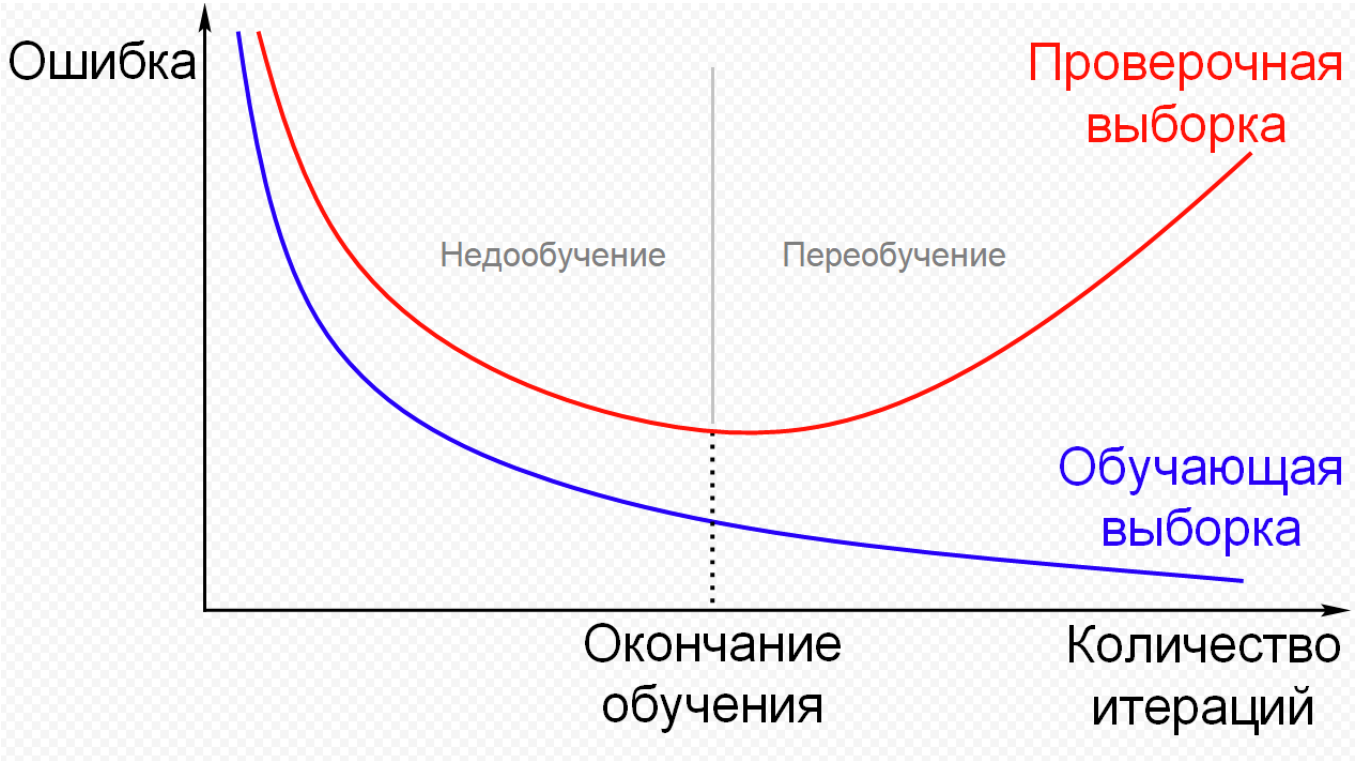
# ПЕРЕОБУЧЕНИЕ И НЕДООБУЧЕНИЕ

Противоречие между **оптимизацией** и **общностью:**

- **Оптимизация** - процесс настройки модели для получения максимального качества на тренировочных данных
- **Общность** - качество обученной модели на данных, которые она прежде не видела

**Недообучение** – на первых итерациях обучения, чем ниже потери (ошибки) на тренировочных данных, тем они ниже на контрольных данных

**Переобучение** - после некоторого числа итераций на тренировочных данных общность перестает улучшаться, проверочные метрики начинают ухудшаться



# БОРЬБА С ПЕРЕОБУЧЕНИЕМ И НЕДООБУЧЕНИЕМ

Лучший способ предотвращения переобучения – **увеличение объема тренировочных данных**

Если такой возможности нет, то следующим лучшим способом является повышение качества информации или добавление ограничений на информацию, которую модели будет позволено сохранить. Такие процедуры называются **регуляризацией**:

- **Уменьшение размера сети**
- **Регуляризация весов**
- **Прореживание признаков**

# УМЕНЬШЕНИЕ РАЗМЕРА СЕТИ

- Количество изучаемых параметров в модели называют **емкостью** модели. Емкость определяется количеством слоев и количеством нейронов (размерностью) в каждом слое
- Чем больше емкость модели, тем больше образцов она способна просто "запомнить", что может привести к **переобучению** (к подгонке модели под данные)
- Самый простой способ предотвратить переобучение — **уменьшить размер модели**. При этом повышается способность к обобщению. Но при слишком низком размере модели она теряет способность к запоминанию существенных свойств объектов

**В общем случае процесс поиска подходящего размера модели должен начинаться с относительно небольшого количества слоев и параметров, а затем размеры слоев и их количество должны постепенно увеличиваться, пока не произойдет увеличение потерь на проверочных данных**

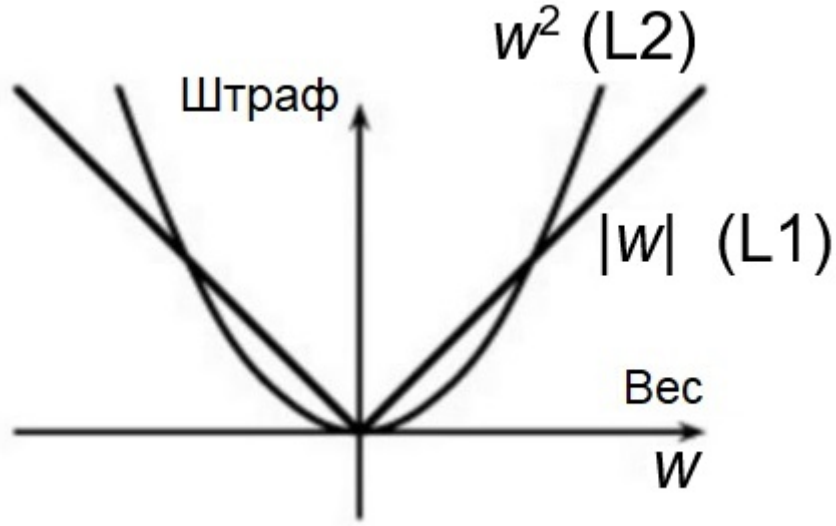
# РЕГУЛЯРИЗАЦИЯ ВЕСОВ

**Более простые модели менее склонны к переобучению, чем сложные**

Простая модель — это модель, в которой распределение значений параметров имеет меньшую энтропию (или модель с меньшим числом параметров)

Упрощение модели может быть достигнуто путем ограничения значений ее весовых коэффициентов, что делает их распределение более равномерным. Этот прием называется **регуляризацией весов**, он реализуется **добавлением в функцию потерь сети штрафа за увеличение весов** и имеет две разновидности:

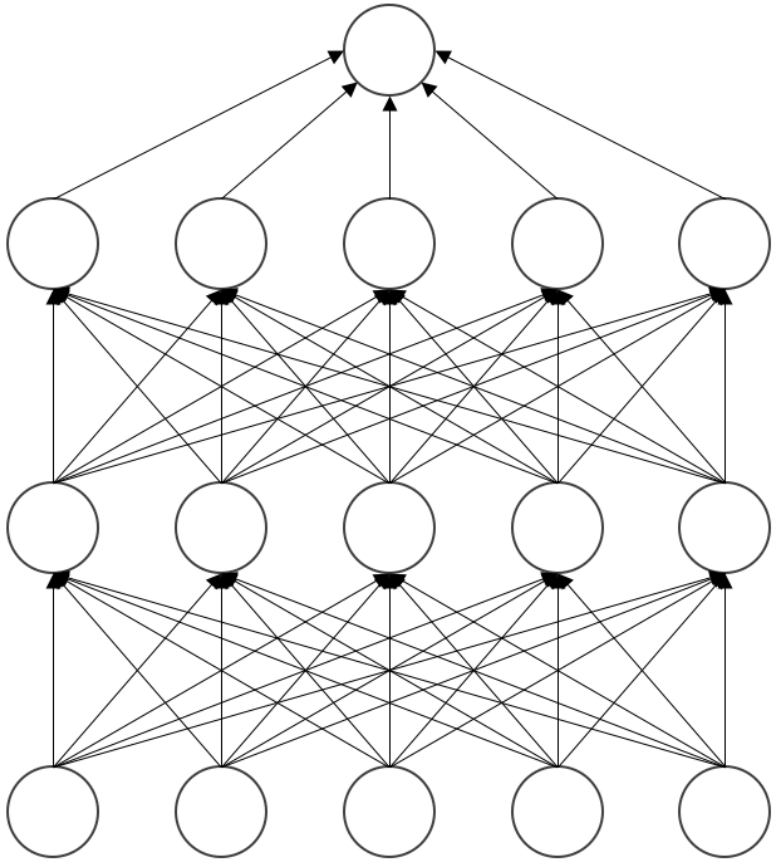
- **L1-регуляризация** — штраф прямо пропорционален абсолютным значениям весовых коэффициентов (L1-норма весов)
- **L2-регуляризация** — штраф пропорционален квадратам значений весовых коэффициентов (L2-норма весов)
  В контексте нейронных сетей L2-регуляризация также называется **сокращением весов**



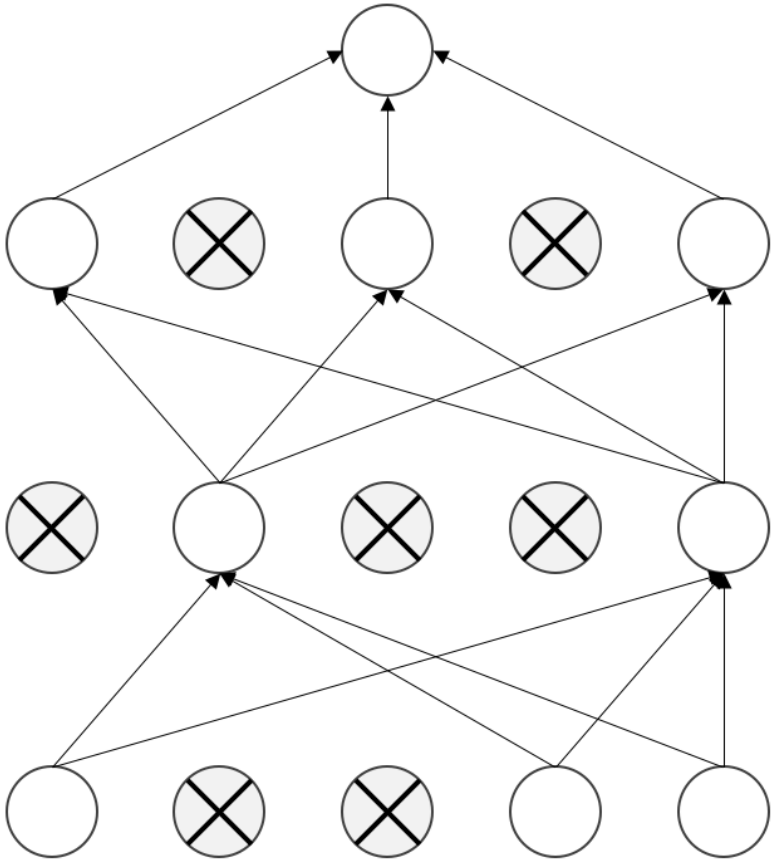
# ПРОРЕЖИВАНИЕ ПРИЗНАКОВ

- **Прореживание** (**dropout**) — один из наиболее эффективных и распространенных приемов регуляризации для нейронных сетей, разработанный Джеффом Хинтоном в Университете Торонто
- Прореживание, которое применяется к слою, заключается в **присваивании нуля** случайно выбираемым признакам на этапе обучения (это равносильно удалению этих признаков)
- Коэффициент прореживания — это доля обнуляемых признаков. Обычно он выбирается в диапазоне от 0,2 до 0,5
- На этапе тестирования прореживание не выполняется. Вместо этого выходные значения уровня уменьшаются на коэффициент, равный коэффициенту прореживания, чтобы компенсировать разницу в активности признаков на этапах тестирования и обучения
- Основная идея прореживания заключается в том, что введение шума в выходные значения нейронов может разбить **случайно складывающиеся шаблоны**, не имеющие большого значения, которые модель может запоминать в отсутствие шума (что ведет к переобучению)

# ПРОРЕЖИВАНИЕ (DROPOUT)



Исходная нейронная сеть



Результат прореживания

# ОБОБЩЕННЫЙ ПРОЦЕСС РЕШЕНИЯ ЗАДАЧ МАШИННОГО ОБУЧЕНИЯ

Далее представлена универсальная схема, которая может быть использована для решения любых задач машинного обучения. Эта схема связывает воедино основные идеи:

- определение задачи
- оценка качества решения задачи
- конструирование признаков
- ослабление проблемы переобучения

# ОПРЕДЕЛЕНИЕ ЗАДАЧИ

Прежде всего, необходимо **определить задачу**:

- Какой вид будут иметь входные данные?
- Что требуется предсказать?

**Гипотезы**, выдвигаемые на этой стадии:

- Гипотеза о том, что выходные данные можно предсказать по входным данным
- Гипотеза о том, что доступные данные достаточно информативны для изучения отношений между входными и выходными данными

На данном этапе ограничивающим фактором является **доступность данных** для обучения

# ОПРЕДЕЛЕНИЕ ТИПА ЗАДАЧИ

К какому **типу** относится поставленная задача:

- Бинарная классификация?
- Многоклассовая классификация?
- Скалярная регрессия?
- Векторная регрессия?
- Многозначная (нечеткая) классификация?
- Что-то иное?

Идентификация типа задачи определит выбор **архитектуры** модели, **функции потерь** и т. д.

# ВЫБОР МЕРЫ УСПЕХА

Сначала необходимо определить, что понимается под **успехом**

Показатель успеха будет определять выбор **функции потерь** — что именно должна оптимизировать модель

Наиболее типичные варианты меры успеха:

- Для задач симметричной классификации, когда каждый класс одинаково вероятен, используется площадь под ROC-кривой (AUC)
- Для задач несимметричной классификации используются точность и полнота
- Для задач регрессии используется среднеквадратическая ошибка
- Для задач ранжирования или многозначной классификации используется среднее математическое ожидание

Иногда необходимо определять специальную меру успеха

# ВЫБОР ПРОТОКОЛА ОЦЕНКИ

Три наиболее распространенных **протокола оценки:**

- **Выделение из общей выборки отдельного проверочного набора данных** — этот способ хорошо подходит при наличии большого объема данных
- **Перекрестная проверка по К блокам** — правильный выбор при небольшом количестве исходных образцов, из которых нельзя выделить представительную выборку для проверки
- **Итерационная проверка по К блокам с перемешиванием** — позволяет с высокой точностью оценить модель, когда имеется ограниченный объем данных

# ПРЕДВАРИТЕЛЬНАЯ ПОДГОТОВКА ДАННЫХ

Преобразование исходных данных в формат, в котором их можно передать в модель машинного обучения — в данном случае в глубокую нейронную сеть:

- Данные должны быть помещены в тензоры
- Значения, помещаемые в тензоры, обычно требуют масштабирования и приведения к меньшим величинам: например, в диапазоне [–1, 1] или [0, 1]
- Если значения разных признаков находятся в разных диапазонах (разнородные данные), их следует нормализовать
- Возможно, также понадобится выполнить конструирование признаков, особенно при небольшом объеме исходных данных

# РАЗРАБОТКА МОДЕЛИ, БОЛЕЕ СОВЕРШЕННОЙ, ЧЕМ БАЗОВЫЙ СЛУЧАЙ

**Базовый случай** - простой случайный выбор

Цель на этом этапе — достичь **статистической мощности**, то есть разработать небольшую модель, способную выдать более качественный результат по сравнению с базовым случаем

Три ключевых выбора:

- **Функция активации** для последнего уровня — устанавливает эффективные ограничения на результат сети
- **Функция потерь** — должна соответствовать типу решаемой задачи
- **Алгоритм оптимизации**

# ФУНКЦИЯ АКТИВАЦИИ И ФУНКЦИЯ ПОТЕРЬ

Функция активации для последнего уровня и функция потерь выбираются в зависимости от решаемой задачи

| Тип задачи | Функция активации для последнего уровня | Функция потерь |
|---|---|---|
| Бинарная классификация | Сигмоид | Бинарная кросс-энтропия |
| Многоклассовая, однозначная классификация | Многомерная логистическая функция | Категориальная кросс-энтропия |
| Многоклассовая, многозначная классификация | Сигмоид | Бинарная кросс-энтропия |
| Регрессия по произвольным значениям | Нет | Среднеквадратическая ошибка |
| Регрессия по значениям между 0 и 1 | Сигмоид | Среднеквадратическая ошибка или бинарная кросс-энтропия |

# РАЗРАБОТКА МОДЕЛИ С ПЕРЕОБУЧЕНИЕМ

Первоначально конструируется избыточная модель:

- Добавление слоев
- Большое число параметров в слоях
- Обучение на большом количестве эпох

Ухудшение качества на проверочных (валидационных) данных сигнализирует о достижения переобучения

Дальнейшие усилия будут направлены на оптимизацию полученной модели

# РЕГУЛЯРИЗАЦИЯ И НАСТРОЙКА ГИПЕРПАРАМЕТРОВ

Возможные пути улучшения модели:

- Прореживание (dropout)
- Разные варианты архитектуры: добавление и удаление слоев
- L1- и/или L2-регуляризация
- Подбор гиперпараметров (число нейронов на слой или шаг обучения оптимизатора)
- Дополнительный цикл конструирования признаков: добавление новых или удаление тех, которые оказываются неинформативными

При изменении гиперпараметров необходимо помнить об опасности просачивания информации из проверочной (валидационной) выборки (не должно быть слишком много циклов коррекции гиперпараметров)

# ОКОНЧАТЕЛЬНОЕ ОБУЧЕНИЕ И ПРОВЕРКА

- Получив удовлетворительную конфигурацию, можно обучить окончательный вариант модели на всех доступных данных (тренировочных и проверочных) и оценить ее качество на контрольном наборе
- Если качество модели на контрольных данных окажется значительно хуже, чем на проверочных, это может означать, что процедура проверки была ненадежной или в процессе настройки параметров модели проявился эффект переобучения на проверочных данных
- В этом случае можно попробовать переключиться на использование другого, более надежного протокола оценки (такого, как итерационная проверка по К блокам с перемешиванием)