



**СПбГЭТУ «ЛЭТИ»**  
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

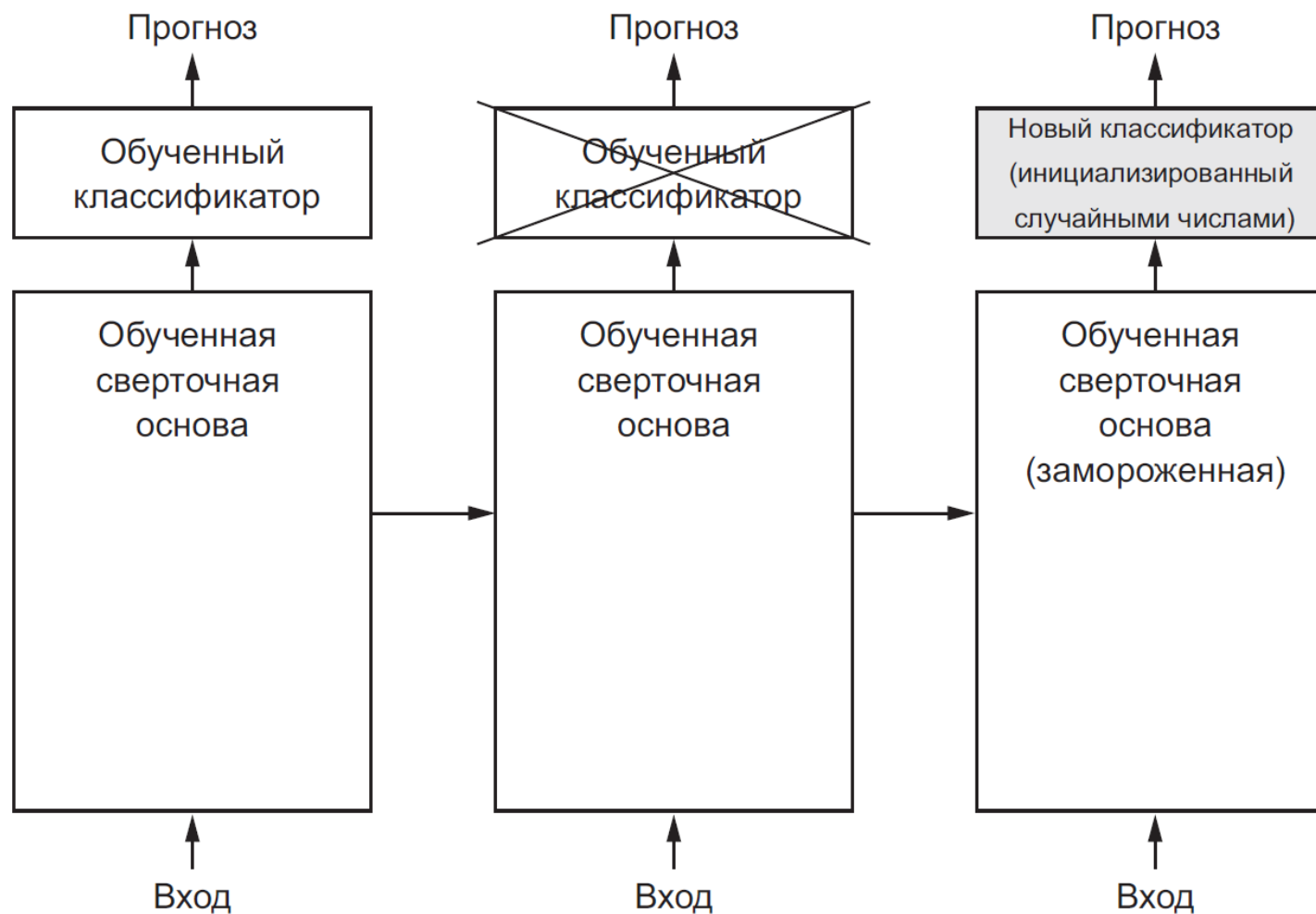
Александр Калиниченко

# **ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ**

Модуль 2. Методы искусственного интеллекта

Лекция 14. Комбинированные архитектуры ИНС

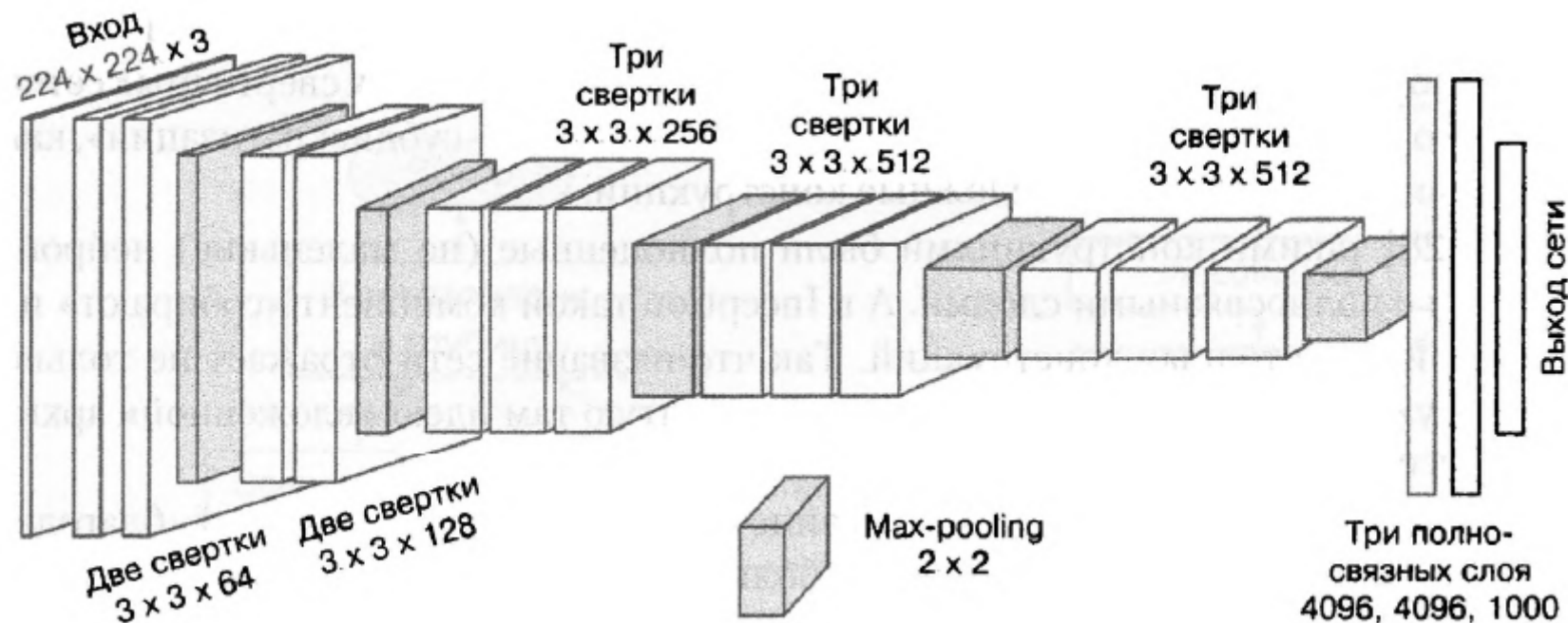
# ИСПОЛЬЗОВАНИЕ ПРЕДВАРИТЕЛЬНО ОБУЧЕННОЙ СЕТИ



Обучение классификатора



# СЕТЬ “VGG”

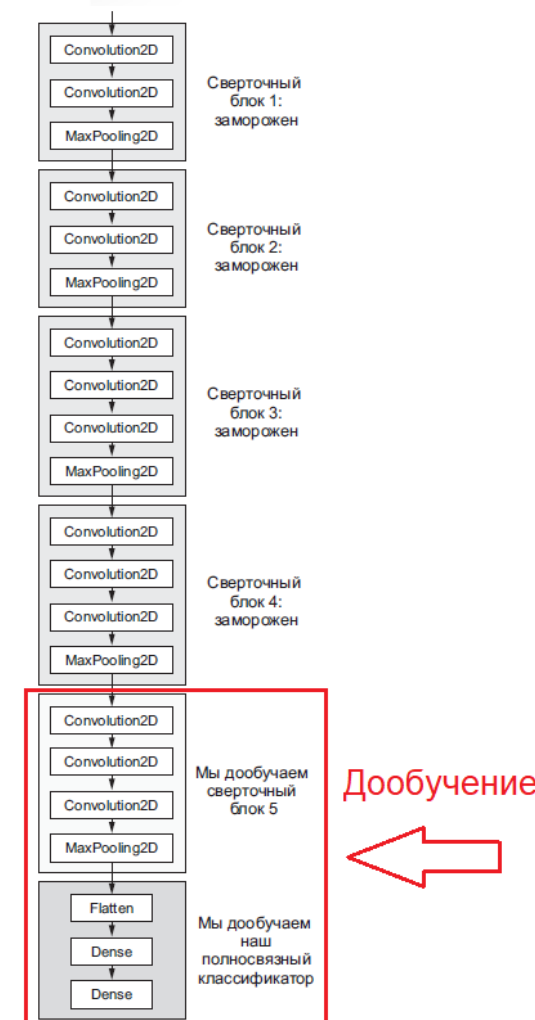


Модель **VGG** (Visual Geometry group) была разработана в Оксфордском университете в 2014 г. и очень хорошо проявила себя в ряде соревнований по распознаванию изображений

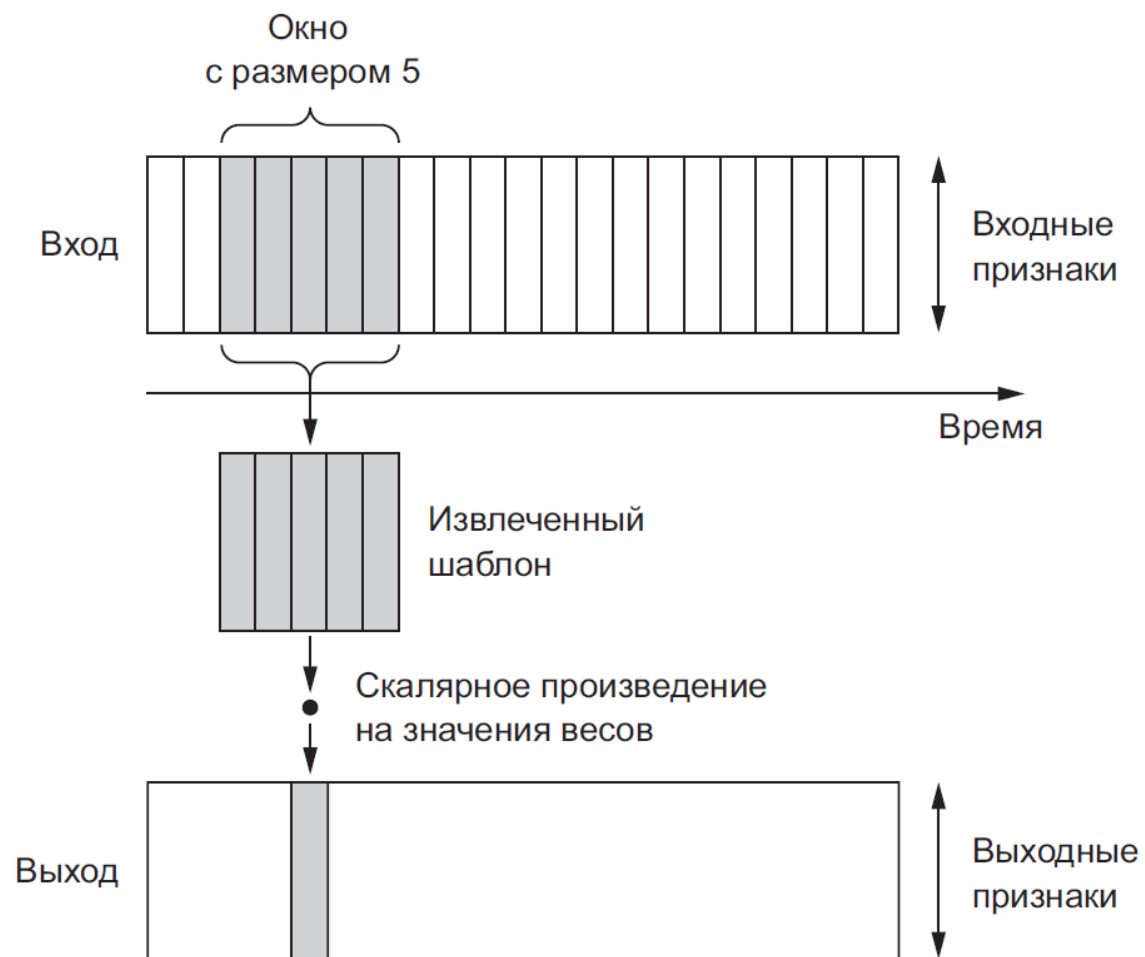
# ДООБУЧЕНИЕ СЕТИ

Для дообучения имеющейся сети требуется выполнить следующие шаги:

- Добавить свою сеть поверх обученной базовой сети
- Заморозить базовую сеть
- Обучить добавленную часть
- Разморозить несколько слоев в базовой сети
- Обучить эти слои и добавленную часть вместе



# ОБРАБОТКА ПОСЛЕДОВАТЕЛЬНОСТЕЙ С ПОМОЩЬЮ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ



**Сверточные** нейронные сети работают особенно хорошо благодаря своей способности извлекать функции из локальных входных сегментов и обеспечивать модульность представления и эффективность данных

Как работает **1D-свертка**: каждый выходной временной шаг получается из временного сегмента во входной последовательности

## 1D СВЕРТКА И СУБДИСКРЕТИЗАЦИЯ (POOLING)

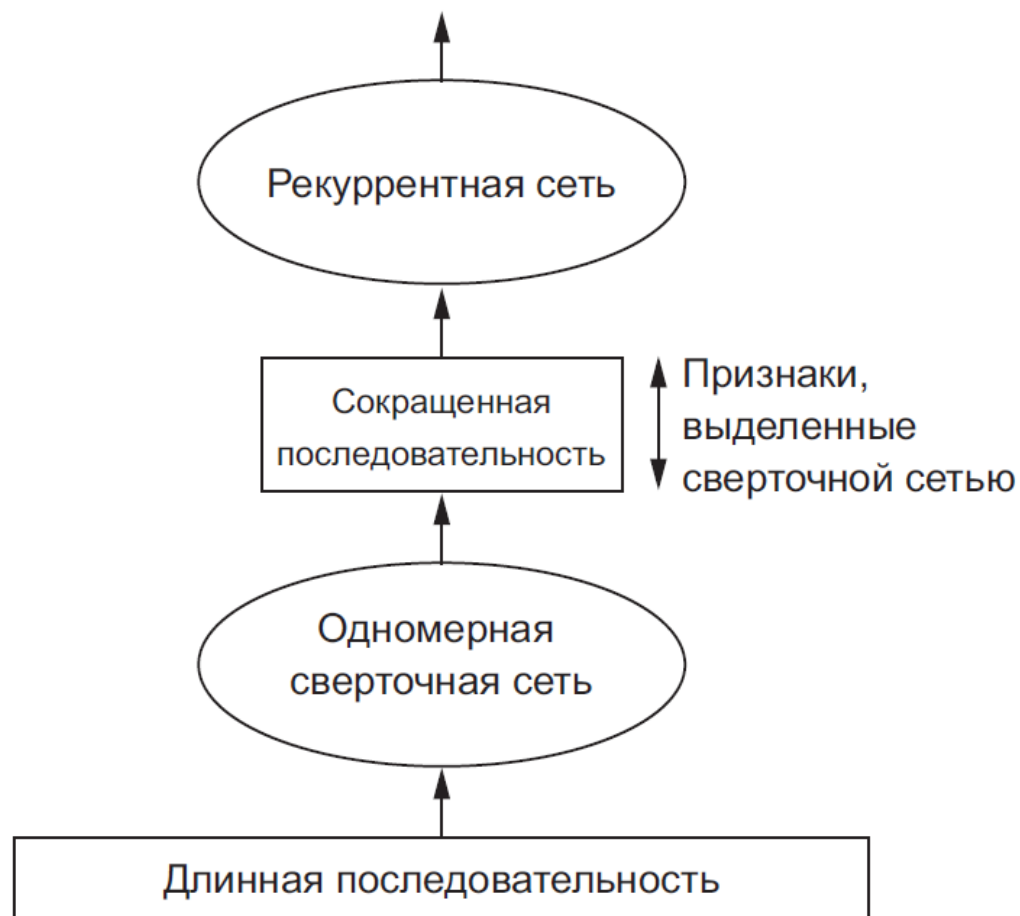
- **1D свертка** извлекает локальные 1D патчи (подпоследовательности) из последовательностей

Поскольку на каждом патче выполняется одно и то же входное преобразование, шаблон, изученный в определенном положении, может быть позже распознан в другом положении, что делает трансляцию 1D-сверточных сетей инвариантной

- **1D-пулинг**: извлечение 1D-патчей (подпоследовательностей) из входных данных и вывод максимального значения (max pooling) или среднего значения (average pooling).

Это используется для уменьшения длины 1D-входов (субдискретизации).

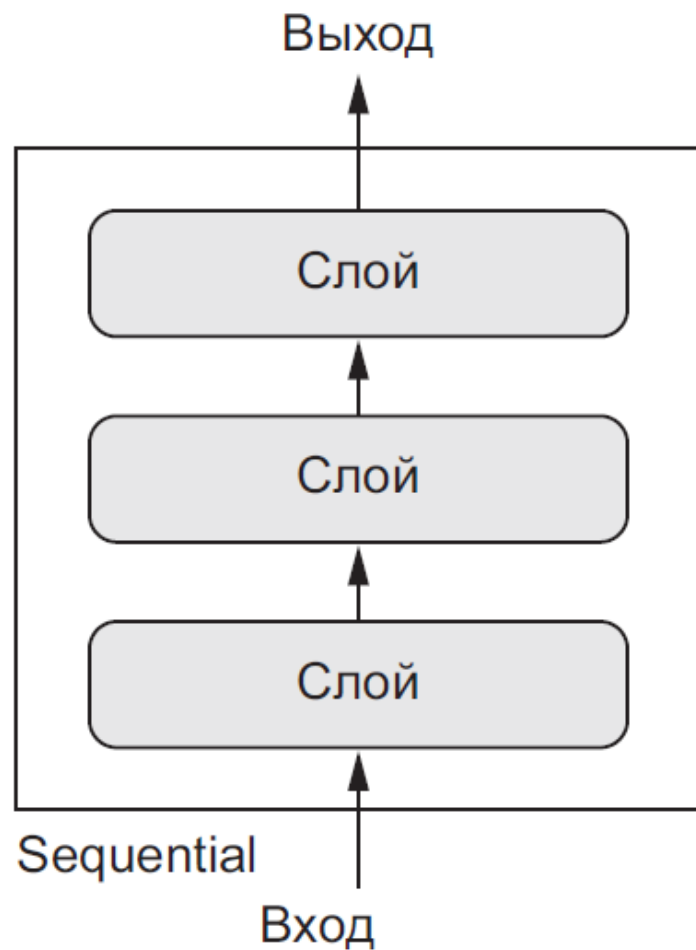
# СОЧЕТАНИЕ 1 D СВЕРТКИ И RNN



- Способ объединить скорость и легкость сверточных сетей с чувствительностью к порядку данных RNN заключается в использовании 1D-сверточной сети в качестве этапа предварительной обработки перед RNN
- Это особенно полезно при работе с последовательностями, которые настолько длинны, что они не могут быть реально обработаны с помощью RNN
- Сверточная сеть превратит длинную входную последовательность в гораздо более короткие (с пониженной частотой выборки) последовательности объектов более высокого уровня
- Эта последовательность извлеченных объектов затем становится входом в RNN-часть сети



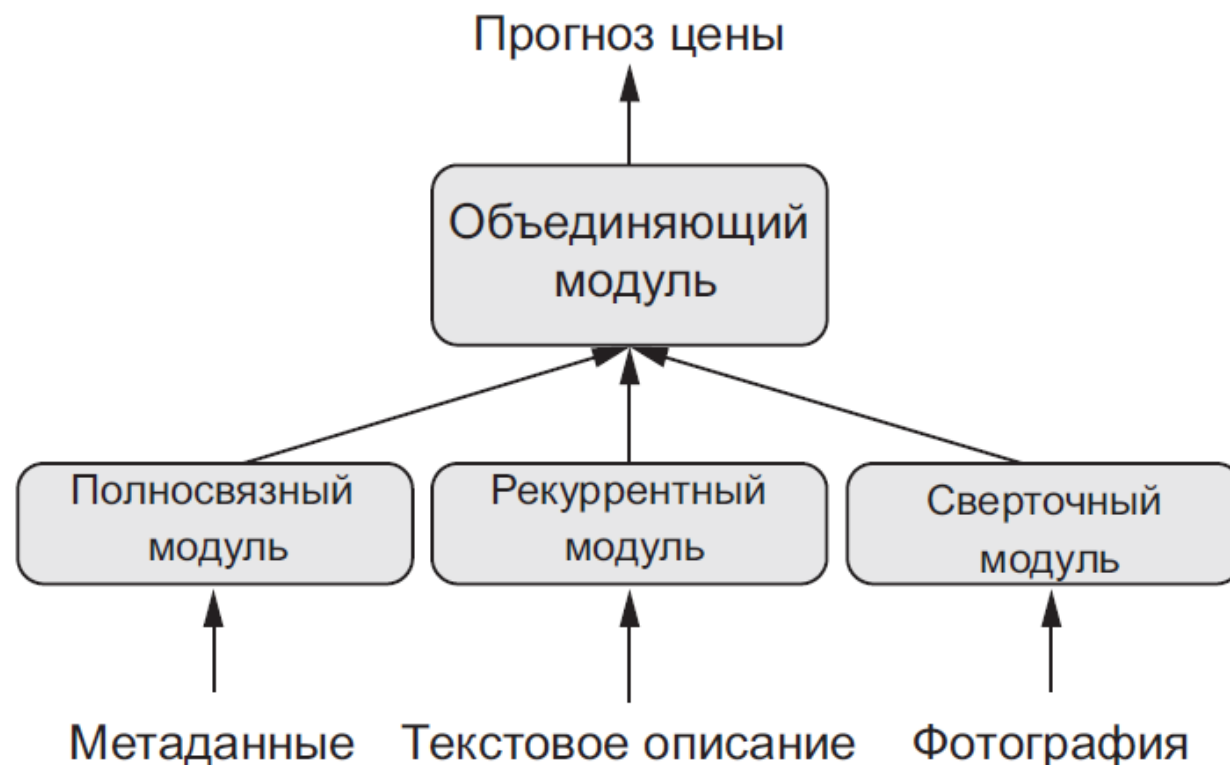
# ПОСЛЕДОВАТЕЛЬНАЯ МОДЕЛЬ



- **Последовательная** (Sequential) модель делает предположение, что сеть имеет ровно один вход и ровно один выход, и что она состоит из линейного стека слоев
- Но этот набор предположений слишком **негибок** в ряде случаев
- Некоторые сети требуют нескольких независимых входов, другие требуют нескольких выходов, а некоторые сети имеют внутреннее разветвление между уровнями

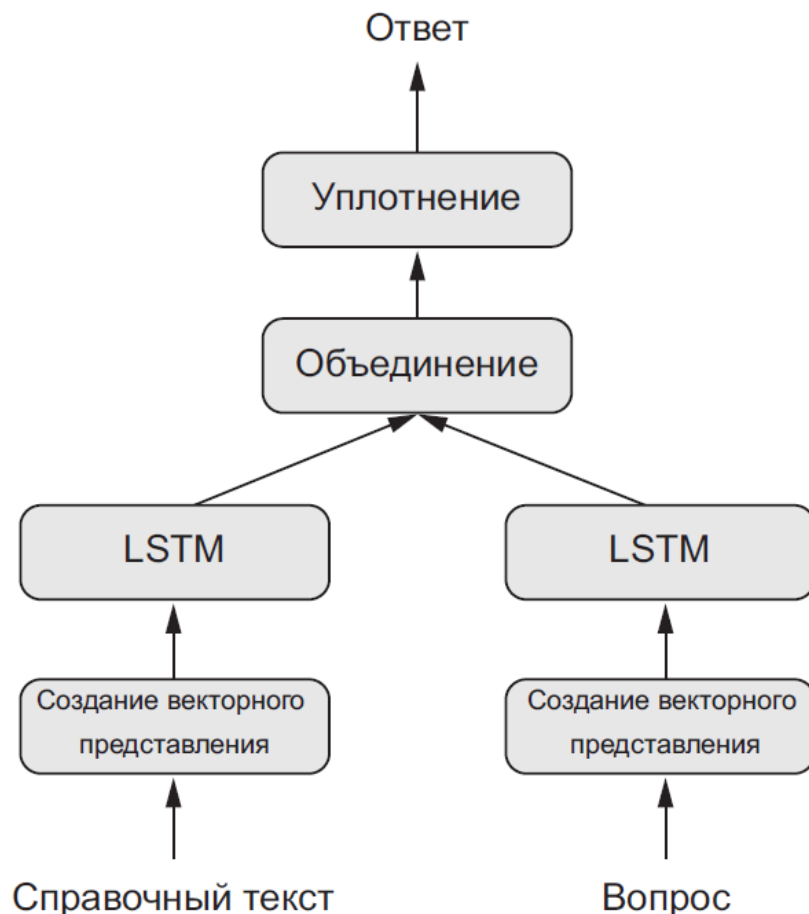


# МОДЕЛЬ С НЕСКОЛЬКИМИ ВХОДАМИ



Совместное обучение модели данных с помощью модели, которая может видеть все доступные модальности ввода одновременно: модель с несколькими входными ветвями

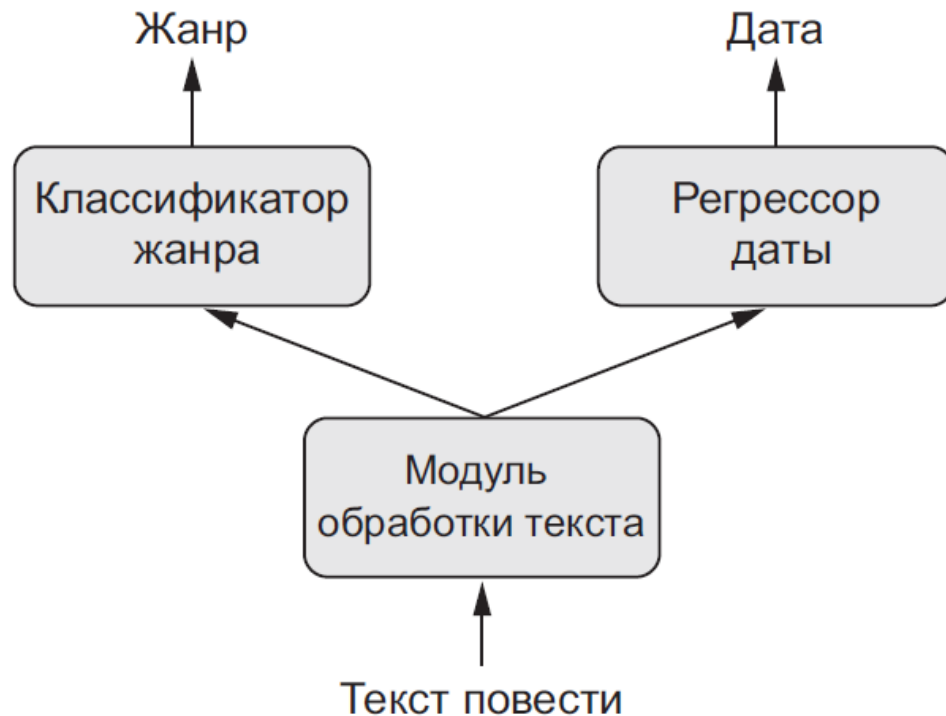
# ПРИМЕР: МОДЕЛЬ ВОПРОС/ОТВЕТ



Пример модели с несколькими входами:

- Типичная модель вопрос/ответ имеет два входа: вопрос на естественном языке и текстовый фрагмент, предоставляющий информацию, которая будет использоваться для ответа на вопрос
- Затем модель должна дать ответ: как правило, это ответ из одного слова, полученный из некоторого predetermined словаря.

# МОДЕЛЬ С НЕСКОЛЬКИМИ ВЫХОДАМИ (ГОЛОВАМИ)

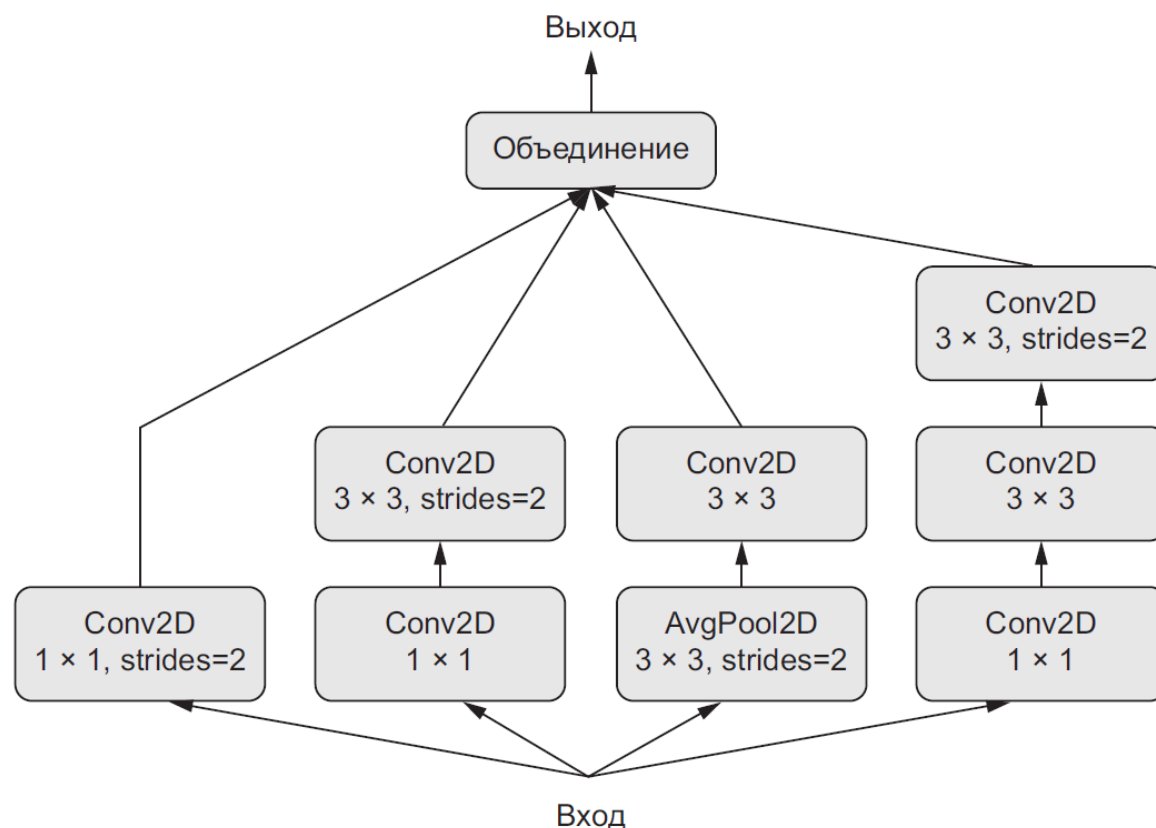


В этом примере:

- Из-за корреляции между жанром и датой книги, знание даты романа помогает модели изучать богатые, точные представления о пространстве жанров романа, и наоборот

- Некоторые задачи должны прогнозировать несколько (два в этом примере) целевых атрибутов входных данных
- Конечно, можно было бы обучить две отдельные модели. Но поскольку эти атрибуты не являются статистически независимыми, можно построить лучшую модель, научившись совместно прогнозировать оба результата одновременно
- Такая совместная модель будет иметь два выхода, или «головы»

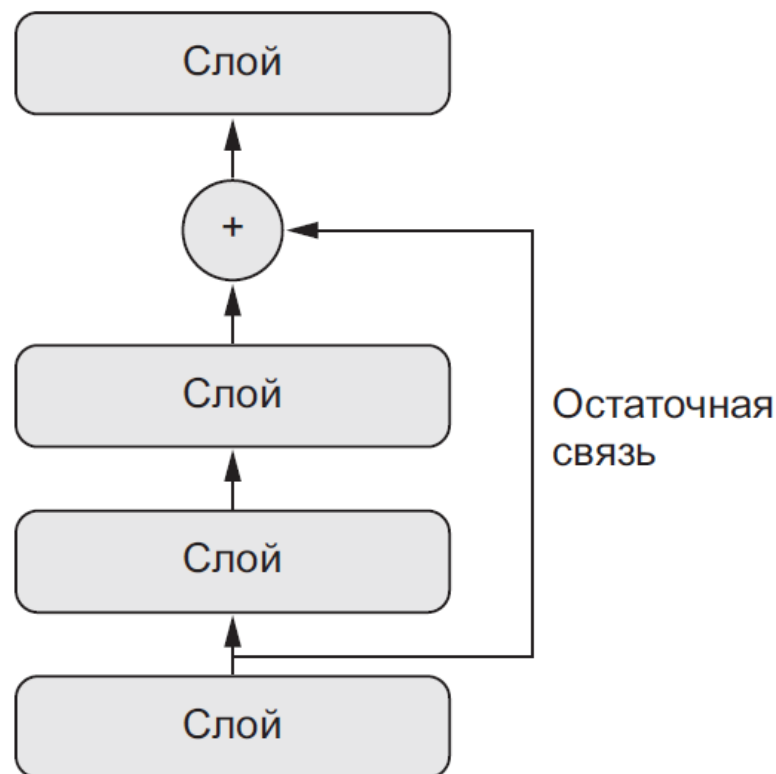
# МОДУЛЬ «INCEPTION»



Модуль Inception: подграф уровней с несколькими параллельными сверточными ветвями

- Многие недавно разработанные нейронные архитектуры требуют нелинейной топологии сети: сетей, структурированных как направленные ациклические графы.
- Это семейство сетей опирается на модули Inception, где входные данные обрабатываются несколькими параллельными сверточными ветвями, выходы которых затем объединяются обратно в один тензор

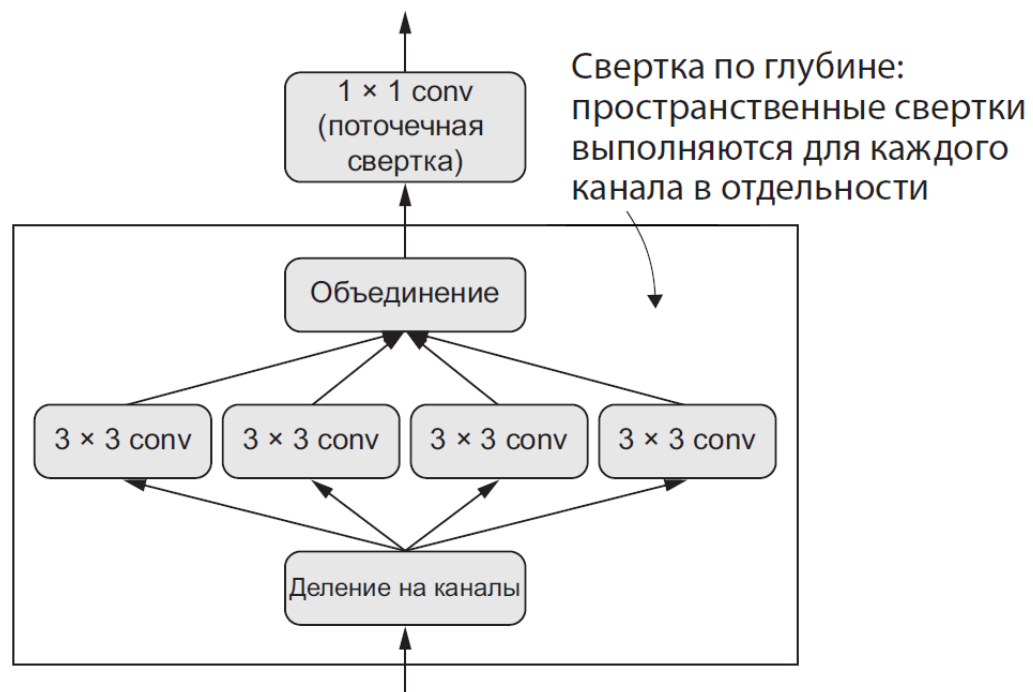
# ОСТАТОЧНЫЕ СВЯЗИ



Остаточные связи: повторное внедрение предыдущей исходящей информации добавлением в карту признаков

- Существует также недавняя тенденция добавления остаточных соединений в модель
- Остаточное соединение состоит из повторной вставки предыдущих представлений в нисходящий поток данных путем добавления тензора прошлого выхода к более позднему выходному тензору, что помогает предотвратить потерю информации в потоке обработки данных

# РАЗДЕЛЬНАЯ СВЕРТКА ПО ГЛУБИНЕ



Раздельная свертка по глубине: за сверткой по глубине следует поточечная свертка

## Слой раздельной свертки по глубине:

- Этот слой выполняет пространственную свертку на каждом канале своего входа независимо перед смешиванием выходных каналов через точечную свертку (свертка  $1 \times 1$ )
- Это эквивалентно разделению изучения пространственных объектов и изучения канальных признаков, что имеет большой смысл, если предполагается, что пространственные местоположения на входе сильно коррелируют, но разные каналы довольно независимы
- Он требует значительно меньше параметров и включает в себя меньше вычислений, что приводит к меньшим и более быстрым моделям
- И поскольку это более репрезентативный способ выполнения свертки, он имеет тенденцию лучше изучать представления, используя меньше данных, что приводит к более эффективным моделям

# ТИПИЧНЫЙ ПРОЦЕСС МАШИННОГО ОБУЧЕНИЯ

- 1. Определите **проблему**: какие данные доступны, и что вы пытаетесь предсказать?
- 2. Определите **способ надежного измерения успеха** в достижении вашей цели (точность прогнозирования или некоторые другие показатели).
- 3. Подготовьте **процесс проверки**, который будет использоваться для оценки моделей (обучающий набор, проверочный набор и тестовый набор)
- 4. **Векторизуйте данные**, превратив их в векторы и предварительно обработав их таким образом, чтобы они были более легко доступны нейронной сети (нормализация и так далее)
- 5. Разработайте **первую модель**, которая превзойдет тривиальный базовый уровень здравого смысла, тем самым демонстрируя, что машинное обучение может работать над вашей проблемой
- 6. Постепенно **совершенствуйте архитектуру** модели, настраивая гиперпараметры и добавляя регуляризацию. Остерегайтесь переобучения при настройке гиперпараметров
- 7. **Реализуйте окончательную модель** в области применения. Продолжайте следить за ее производительностью на реальных данных и используйте свои результаты для уточнения следующей итерации модели



# КЛЮЧЕВЫЕ СЕТЕВЫЕ АРХИТЕКТУРЫ

- **Векторные данные** — плотно связанные модели (плотные слои)
- **Изображения** — 2D сверточные сети
- **Последовательности** — RNN или 1D сверточные сети
- **Видеоданные** — либо 3D-сверточные сети, либо комбинация 2D-сверточных сетей на уровне кадров для извлечения объектов с последующей моделью обработки последовательностей
- **Объемные данные** — 3D сверточные сети