

H² P

Компьютерный дизайн новых материалов

Лекция: Модели свойств. Машинное обучение.

Автор модуля: Ефим Мажник, аспирант Сколковского института науки и технологий (Лаборатория компьютерного дизайна новых материалов)

График реализации модуля

10:00-11:30. Лекция по применению машинного обучения в материаловедении

14:00-15:30. Основы построения и обучения нейронных сетей (практика)

16:00-17:30. Графовые сверточные нейронные сети и их применение для поиска новых материалов (практика)

Устанавливать ничего не нужно, но проверьте, что у вас есть доступ к google документам. Практическая работа будет осуществляться через браузер.

Поиск новых материалов



Экспериментальный	Компьютерный
<ol style="list-style-type: none">1. Попытка угадать правильный материал на основании накопленного опыта2. Синтез материала3. Измерения	<ol style="list-style-type: none">1. Создание выборки из материалов (например, при помощи генетического алгоритма)2. Расчет свойств и оптимизация3. Экспериментальное подтверждение



Если мы умеем считать определенные свойства, то компьютерный дизайн материалов может сократить время на порядок (и больше)

Машинное обучение



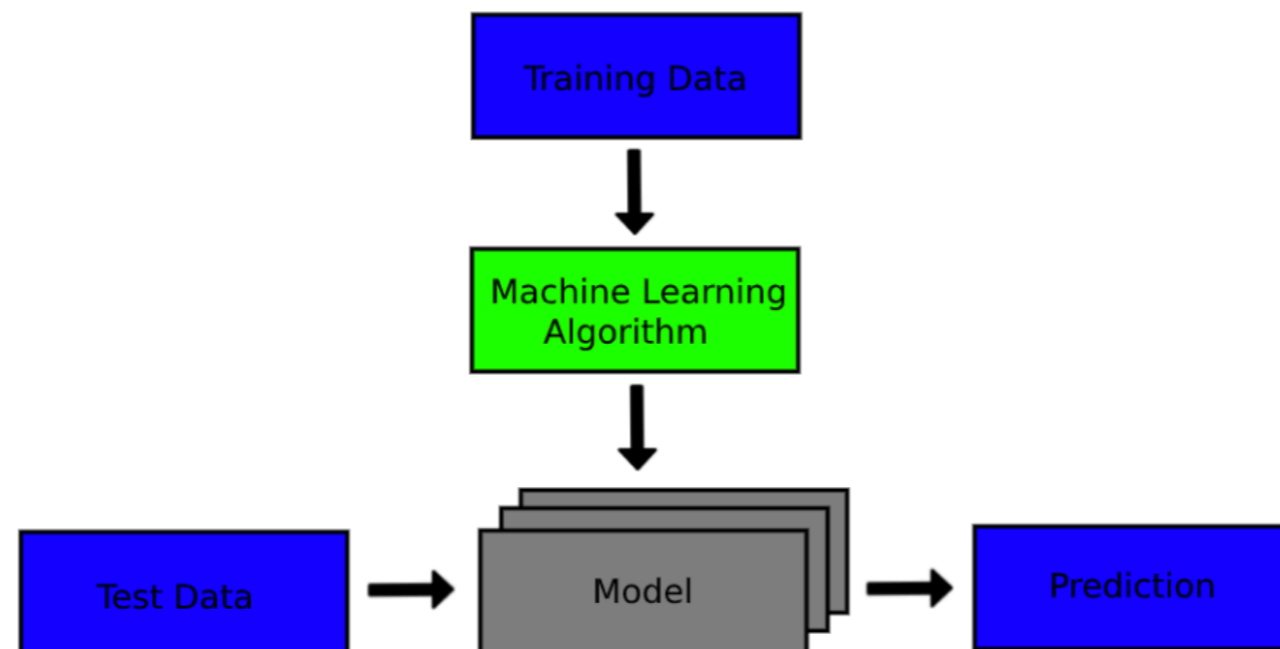
Машинное обучение	Из первых принципов
<ol style="list-style-type: none">1. Теоретическая модель не нужна2. Нужны данные для обучения3. Быстрые расчеты4. Не точные расчеты	<ol style="list-style-type: none">1. Теоретическая модель нужна2. Данные не нужны (по крайней мере, чтобы построить модель)3. Медленные расчеты4. Точные расчеты



Часто нам не нужна высокая точность при поиске новых материалов, поэтому методы машинное обучения находят широкое применение, даже если не дают точных результатов.

Основные принципы

- Модель машинного обучения по своей сути является сложной математической функцией (которая содержит большое количество коэффициентов)
- Мы можем использовать данные, которые у нас есть (экспериментальные или полученные теоретическим путем) и обучить модель получать ответы для других данных



- Лучшие результаты модели могут быть затем перепроверены более точными методами

Взрывное развитие

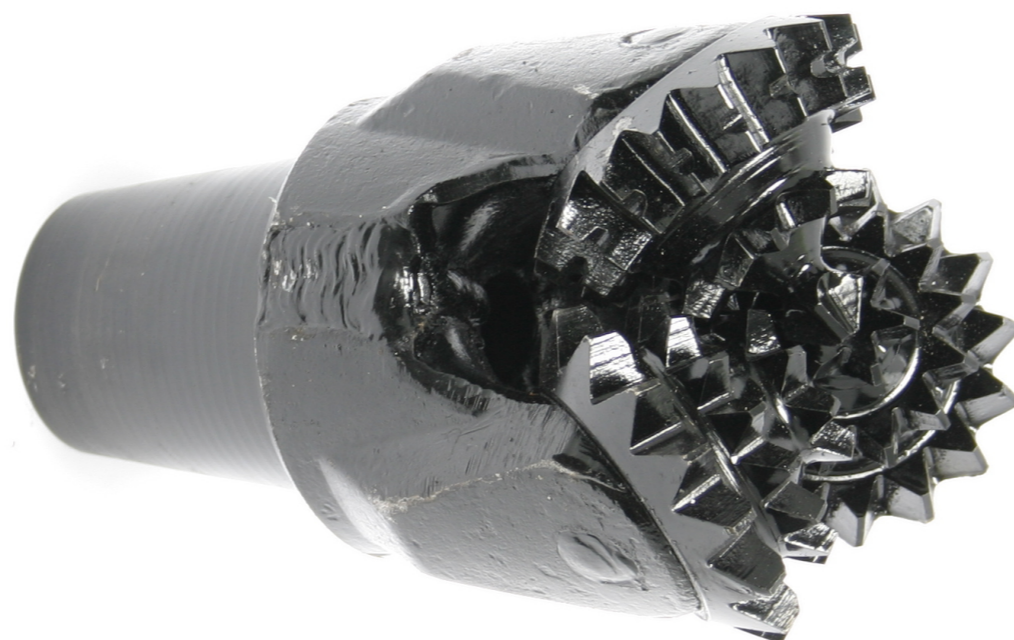
В последние годы модели машинного обучения в науках о материалах развивались особенно быстро. Причины:

- Появляются качественные базы данных с расчетами свойств для большого количества структур (Materials Project, OQMD, Crystallography Open Database, ICSD и т. д.)
- Развитие алгоритмов машинного обучения, способов их обучения и доступности вычислительных мощностей (вслед за остальными индустриями)
- Развитие понимания многих физических свойств материалов и их составляющих
- Нарастающая потребность индустрии в оптимизации свойств используемых материалов

Поиск сверхтвердых материалов

Сверхтвердые материалы необходимы для многих практических применений, поэтому их поиск является важной задачей.

Применимость материала часто определяется по значению двух свойств: твердости и трещиностойкости. Во многих отраслях необходимы материалы, которые обладают высокими значениями каждого из этих параметров.



Поиск сверхтвердых материалов

- Данных для обучения моделей для каждого из этих свойств не так много. Поэтому мы начали с физической модели:

Твердость (по Викерсу)

$$H_v = \gamma_0 \chi(\nu) E$$

Трещиностойкость

$$K_{IC} = \alpha_0^{-1/2} V_0^{1/6} [\zeta(\nu) E]^{3/2}$$

- Используя связь с упругими модулями (данных для которых гораздо больше), мы можем предсказать и эти свойства

Диаграмма Эшби

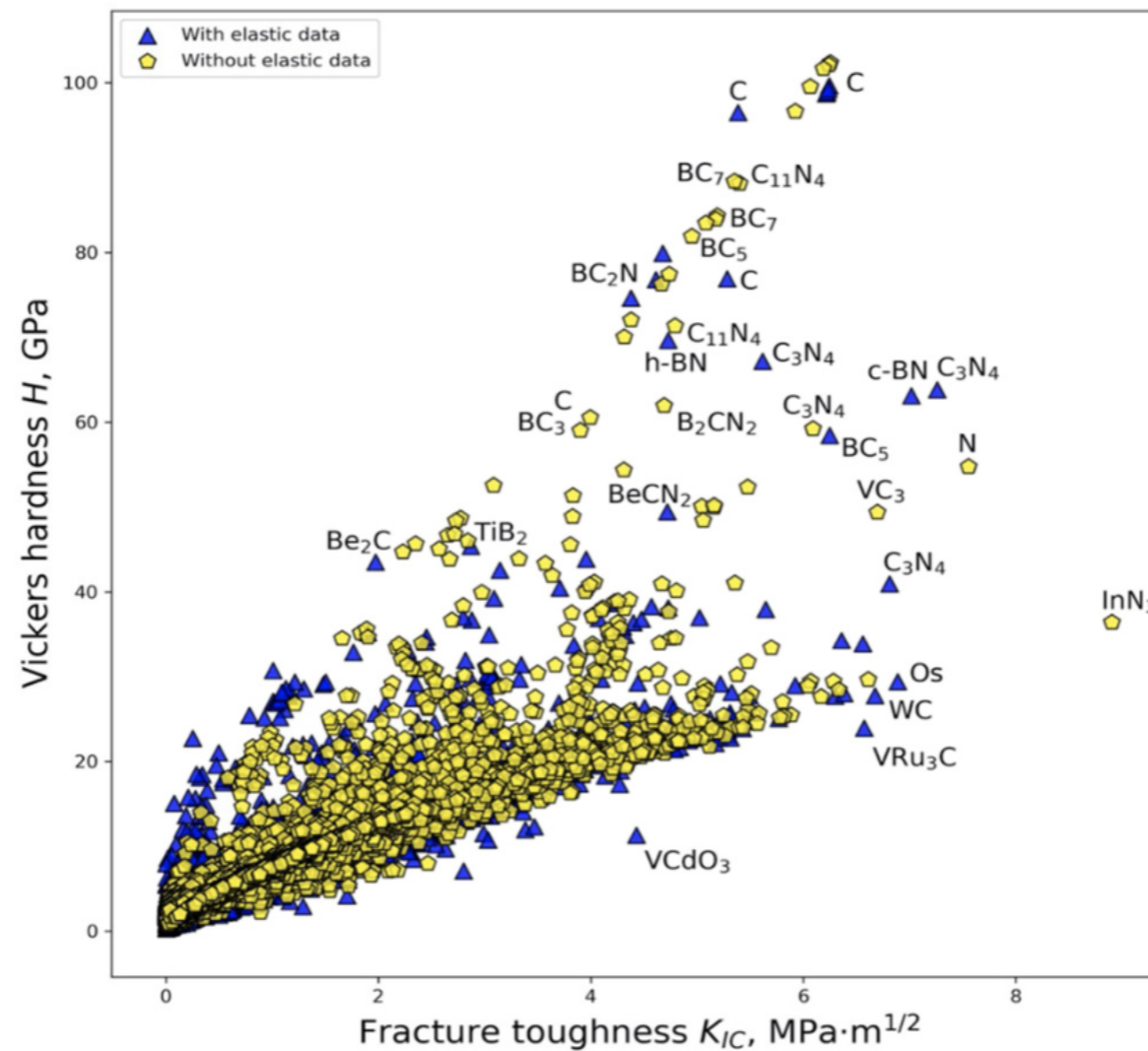




График ~100 000 кристаллических структур с разными значениями твердости и трещиностойкости

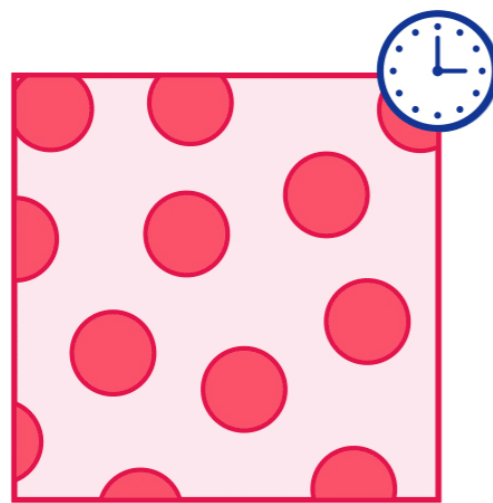


Unveiling the predictive power of static structure in glassy systems

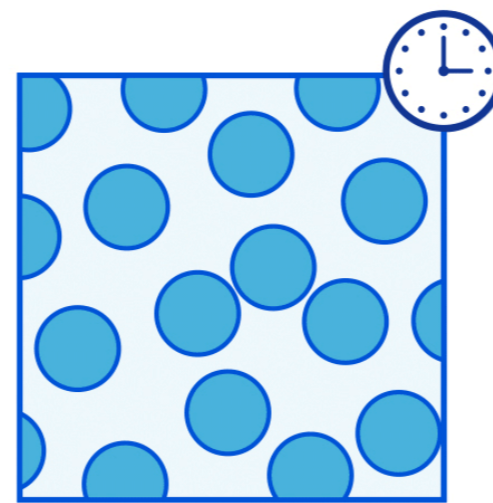
V. Bapst^{1,3}  , T. Keck^{1,3}, A. Grabska-Barwińska¹, C. Donner¹, E. D. Cubuk², S. S. Schoenholz², A. Obika¹, A. W. R. Nelson¹, T. Back¹, D. Hassabis¹ and P. Kohli¹

Despite decades of theoretical studies, the nature of the glass transition remains elusive and debated, while the existence of structural predictors of its dynamics is a major open question. Recent approaches propose inferring predictors from a variety of human-defined features using machine learning. Here we determine the long-time evolution of a glassy system solely from the initial particle positions and without any handcrafted features, using graph neural networks as a powerful model. We show that this method outperforms current state-of-the-art methods, generalizing over a wide range of temperatures, pressures and densities. In shear experiments, it predicts the locations of rearranging particles. The structural predictors learned by our network exhibit a correlation length that increases with larger timescales to reach the size of our system. Beyond glasses, our method could apply to many other physical systems that map to a graph of local interaction.

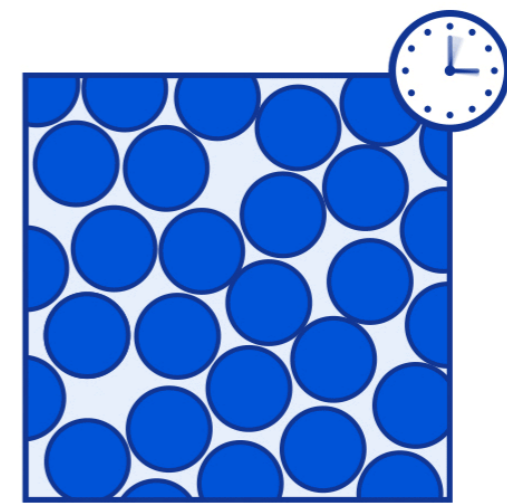
Paper by Google DeepMind (AlphaGo, AlphaZero, AlphaFold)



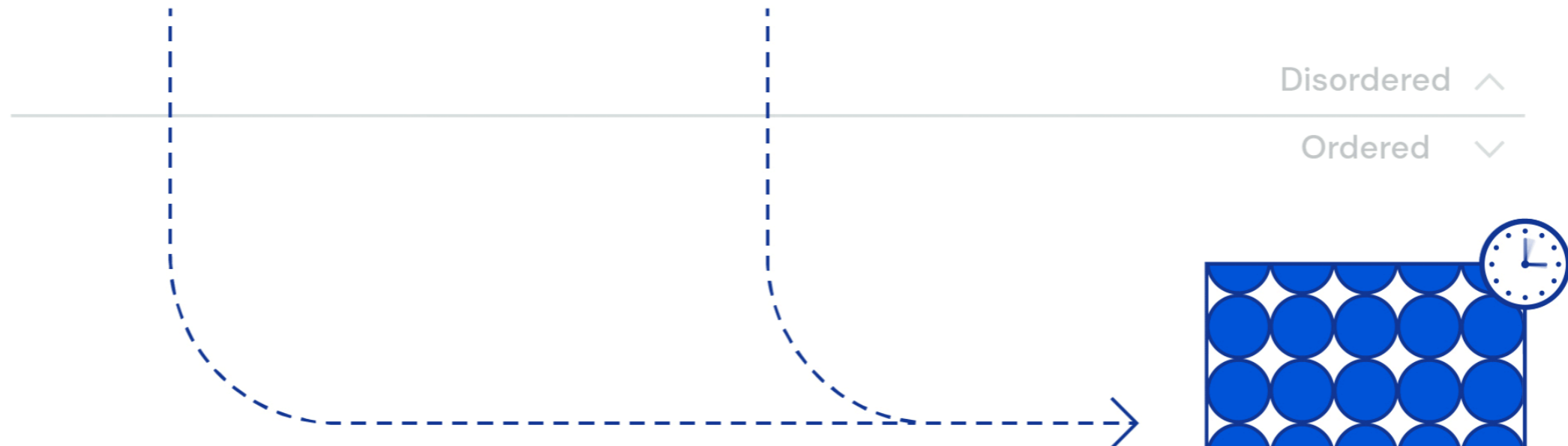
Liquid



Supercooled liquid

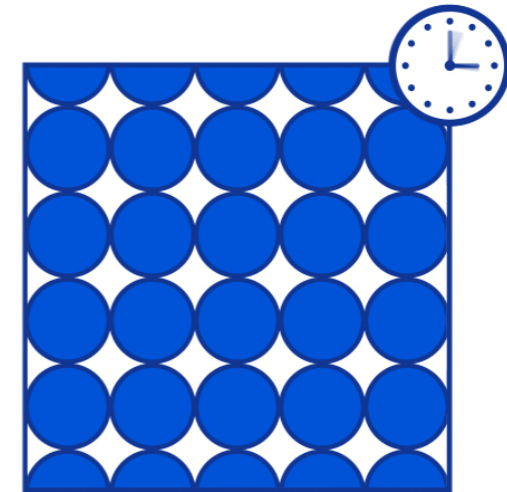


Glass



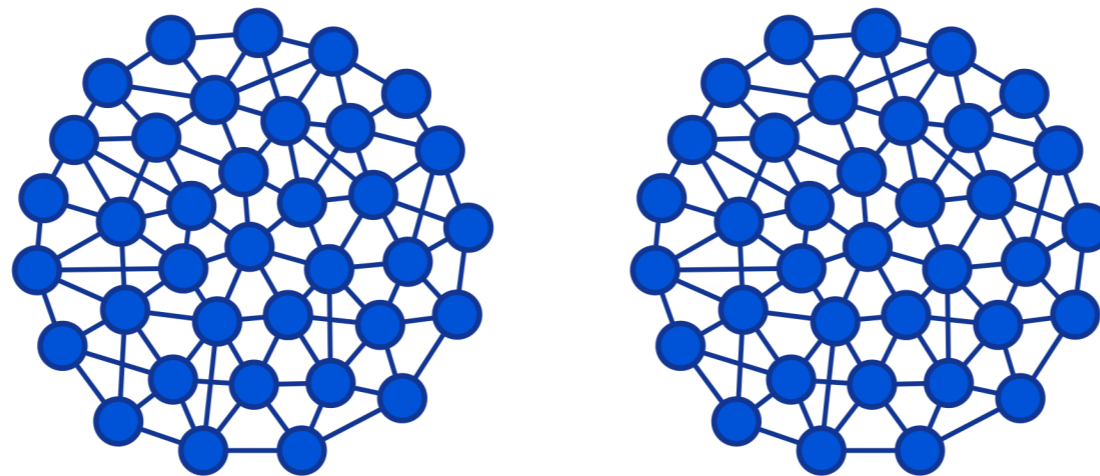
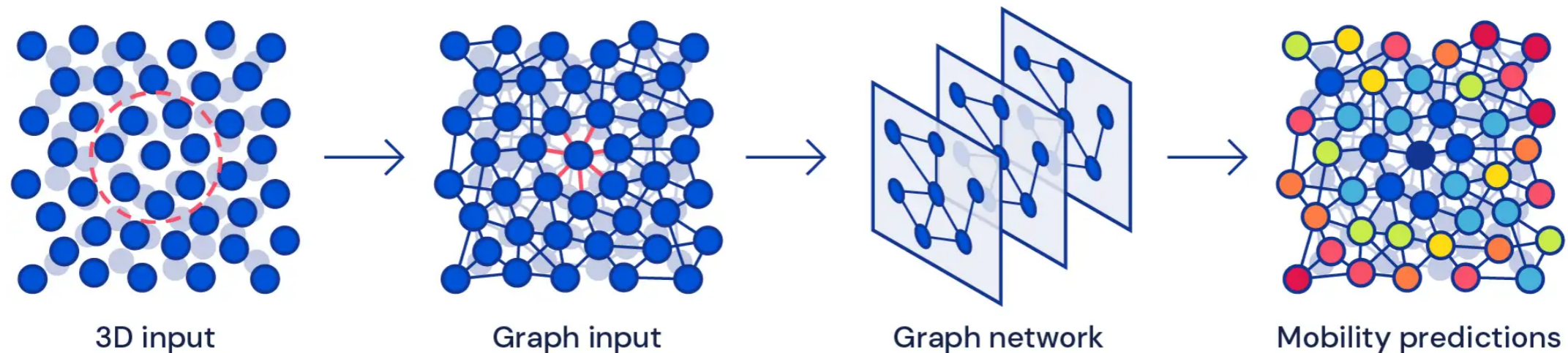
Disordered ^

Ordered v



Crystal

Используя нейронные сети мы можем предсказывать подвижность для каждого атома структуры в зависимости от внешних условия



Так же мы можем оценить насколько на те или иные процессы влияют эффекты дальнего действия

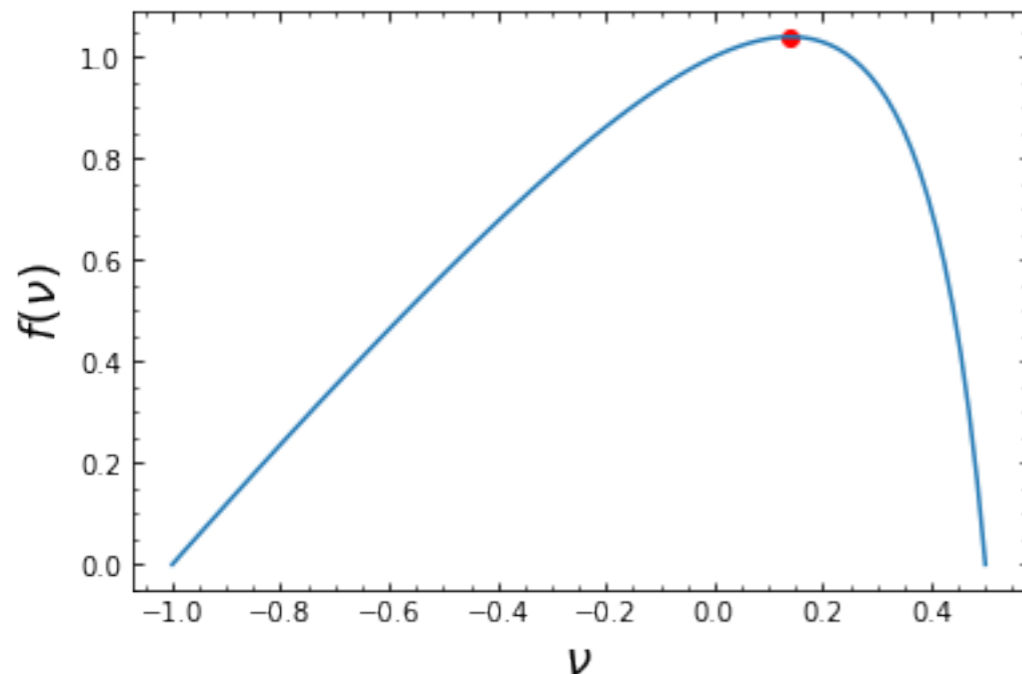
Потенциалы

- С помощью различных методов (moment tensor potentials, bispectrum) удается хорошо описывать энергию системы в зависимости от положений ядер
- В таких подходах энергия системы представляется как сумма энергий от атомных окружений (что хорошо выполняется для короткодействующих сил)
- Такой подход сложно распространить на другие свойства. Так же имеется зависимость сложности обучения от числа атомов
- Хорошо работает подход active learning (предсказываем только там, где неуверены)

Предел упругих свойств и температурного расширения

- Результаты теоретической оценки:

$$\alpha_L E = \gamma c_V \rho (1 - 2\nu) \leq \frac{2(1 + \nu)(1 - 2\nu)c_V \rho}{2 - 3\nu} = f(\nu)c_V \rho$$



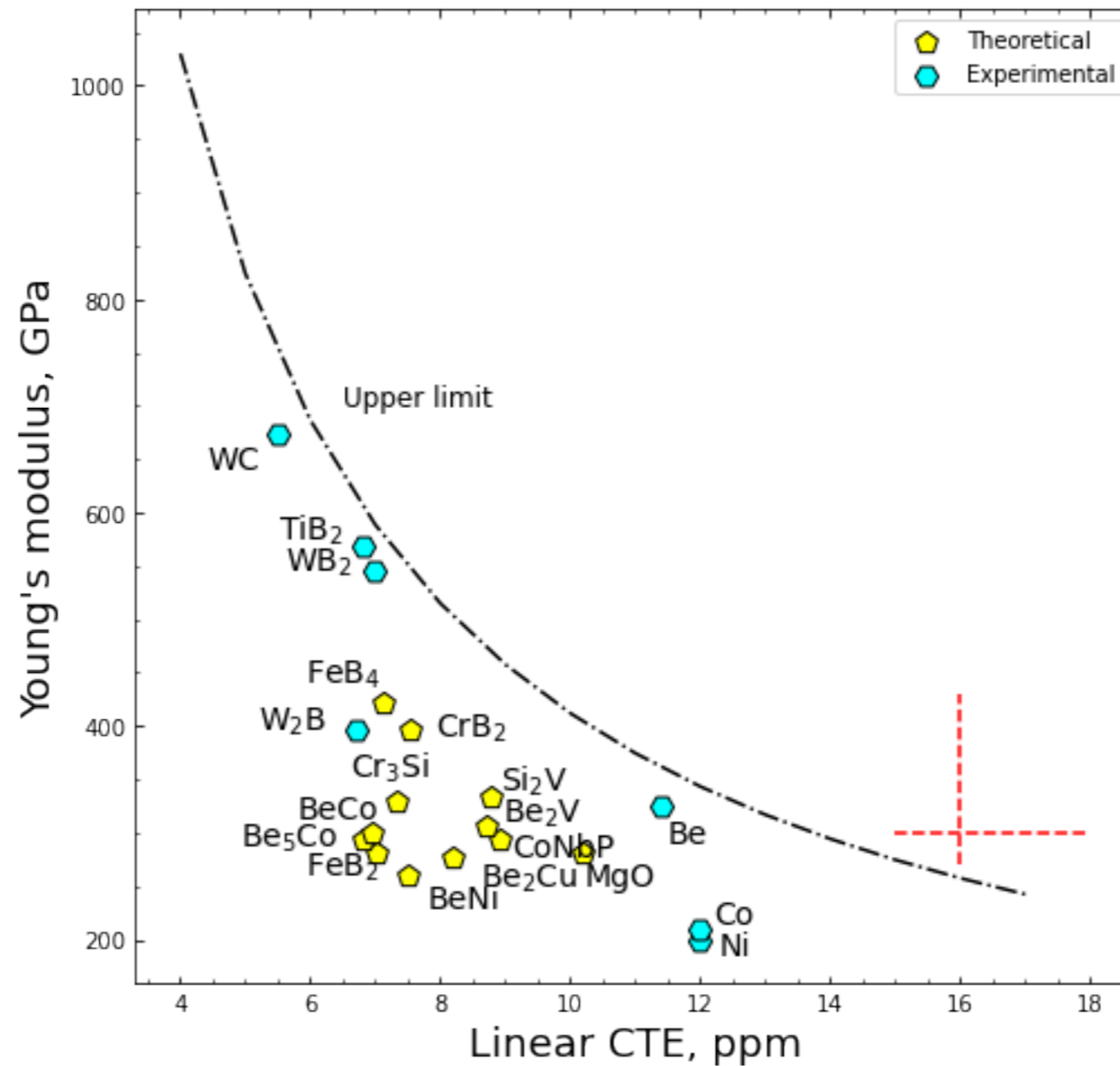
$$\max f(\nu) = \frac{2(11 - 2\sqrt{10})}{9}$$

при $\nu \approx 0.139$

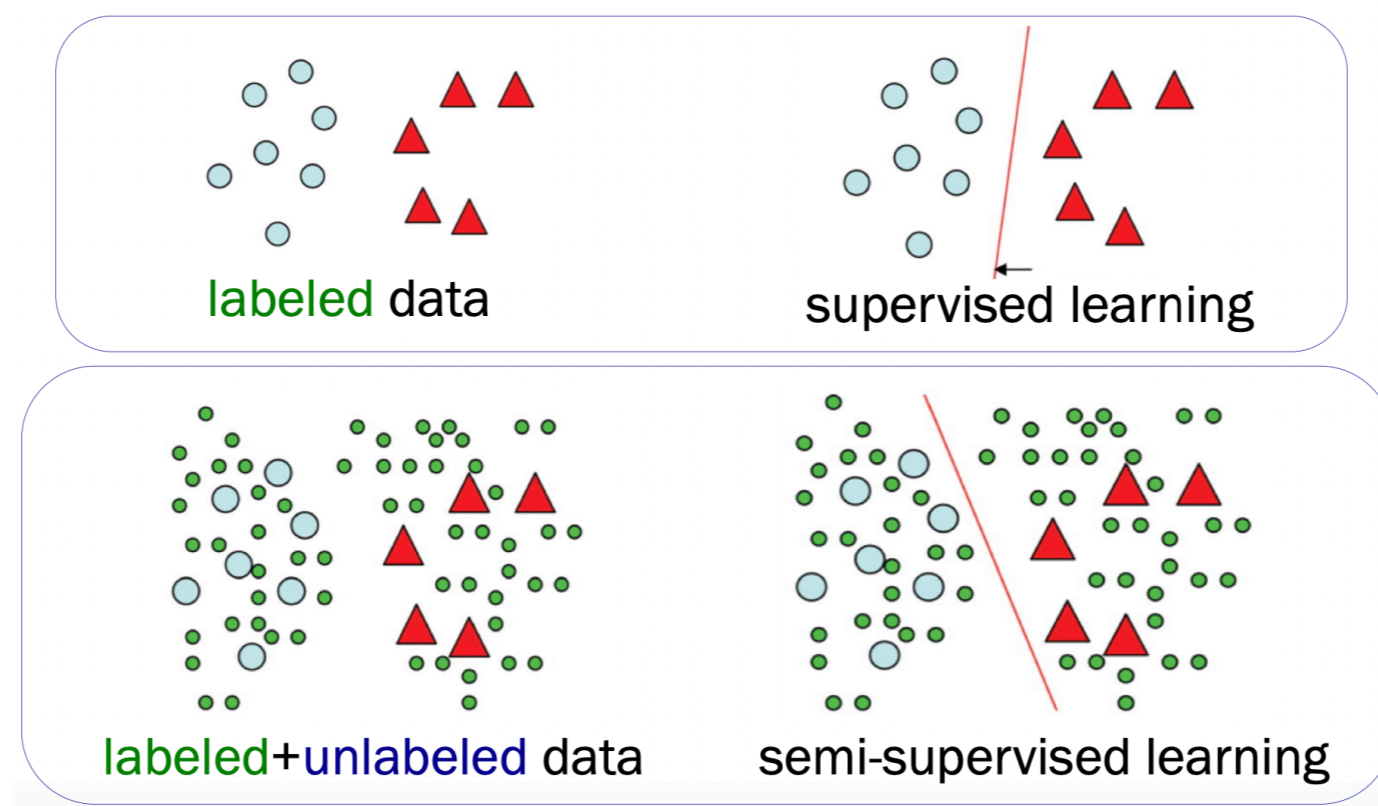
Иерархическое повышение точности

1. Предсказание свойств с использованием машинного обучения для большого числа структур (грубая оценка) (~200000 структур)
2. Более точная оценка с использованием модели Дебая (~100 структур)
3. Более точные вычисления при помощи DFT & квази-гармонического приближения (~20 структур)

Упругие свойства и температурное расширение



Semi-supervised & transfer learning



Мы можем использовать semi-supervised learning когда неизвестны свойства для каких-то структур или отдельных атомов в структуре

Transfer learning: предобучение

PU learning



pubs.acs.org/JACS

Article

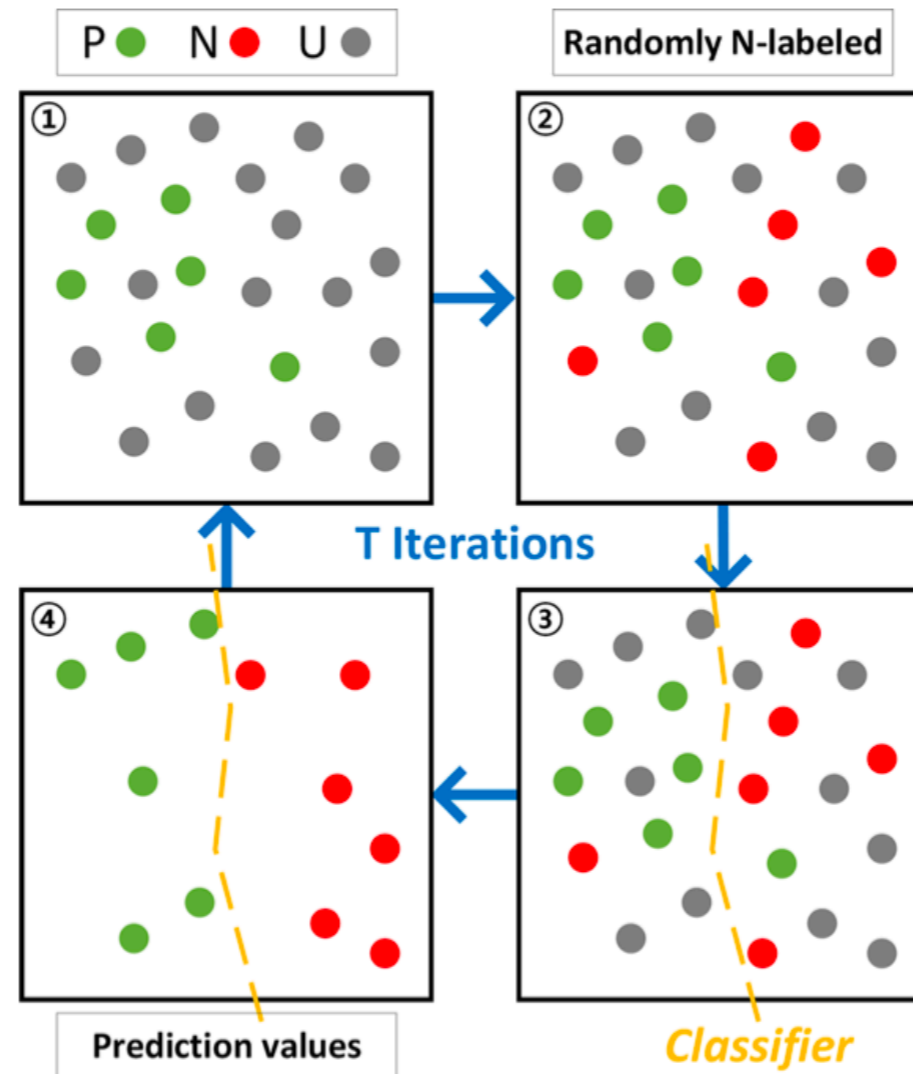
Structure-Based Synthesizability Prediction of Crystals Using Partially Supervised Learning

Jidon Jang,[†] Geun Ho Gu,[†] Juhwan Noh, Juhwan Kim, and Yousung Jung*

- Materials project - 124 515 структур.
- 46 781 структур так же имеют запись в базе данных ICSD, что означает, что их можно синтезировать
- 77 734 структур про которые неизвестно, можно ли их синтезировать

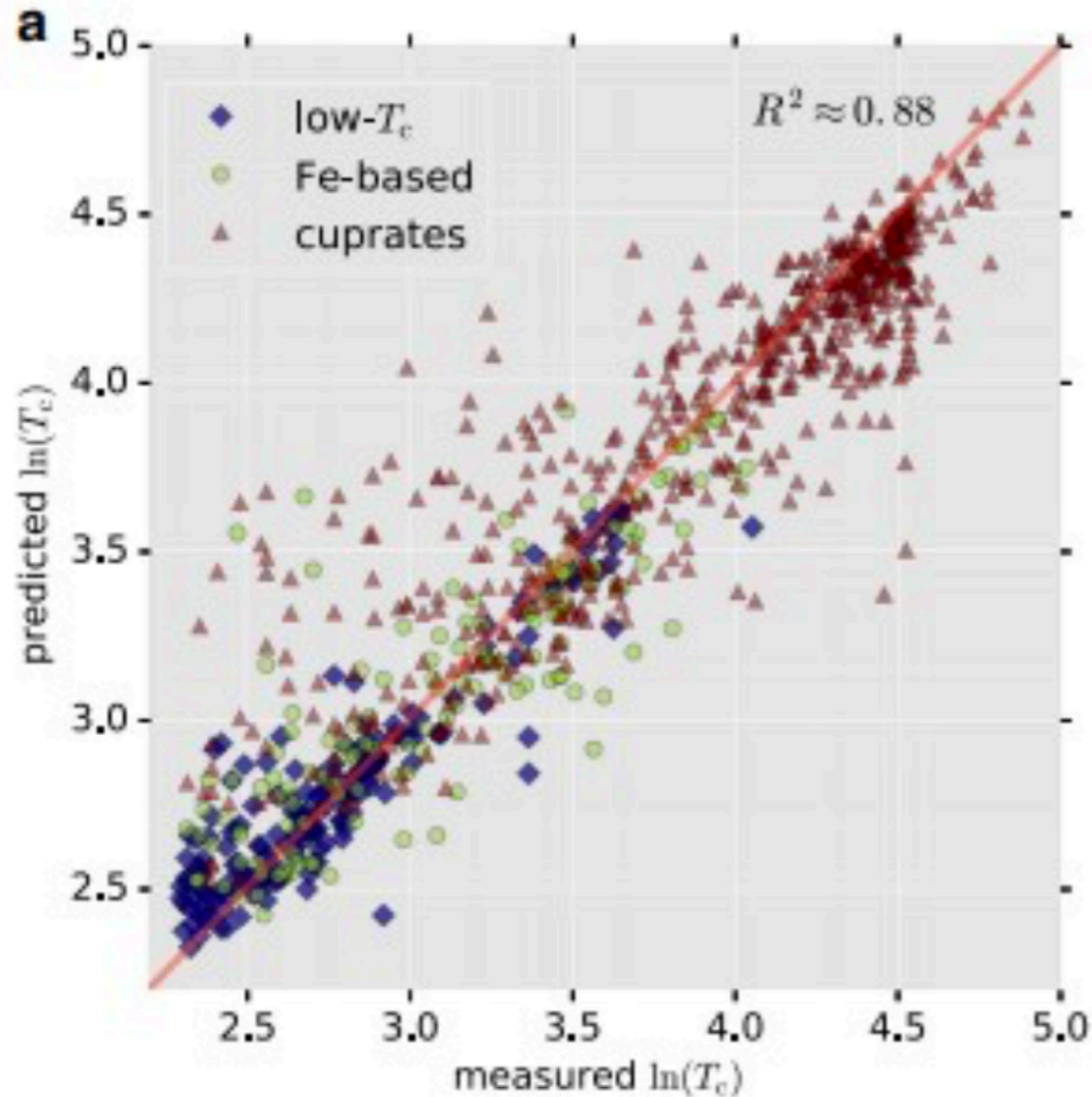
Итого у нас есть данные одного класса (positive) и неразмеченные (unlabeled) данные (PU)

Transductive bootstrap aggregating



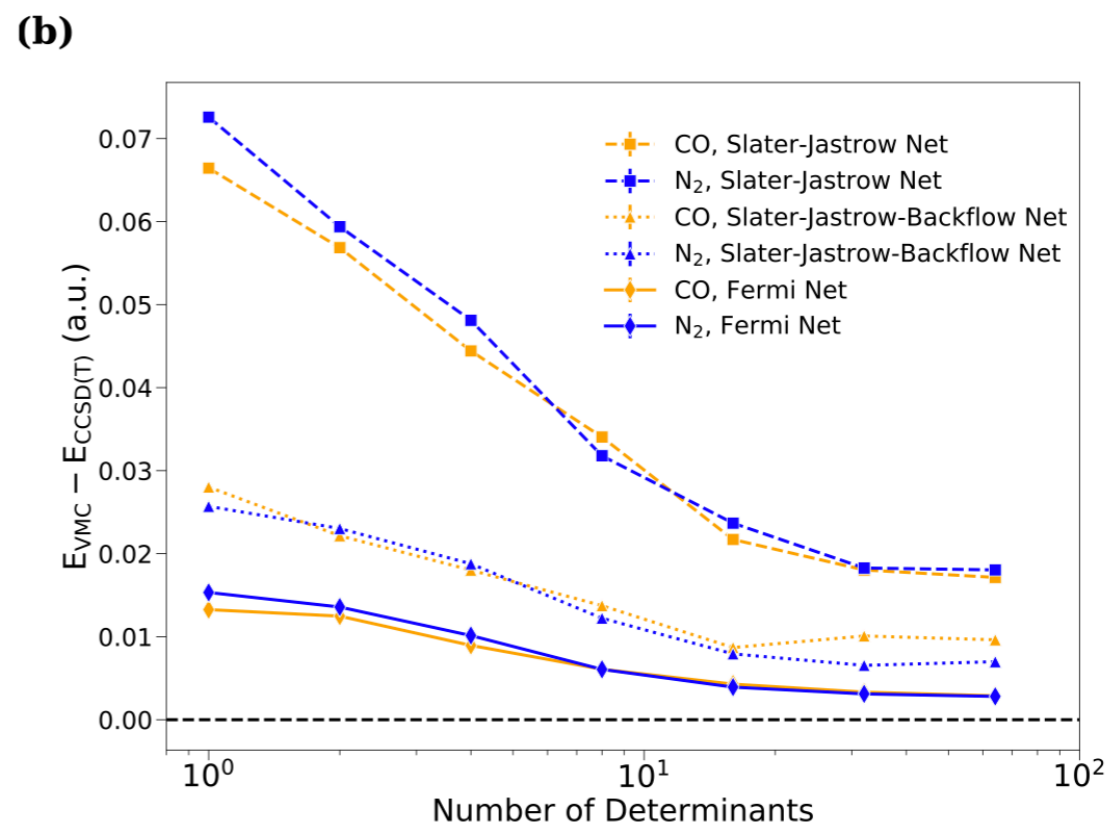
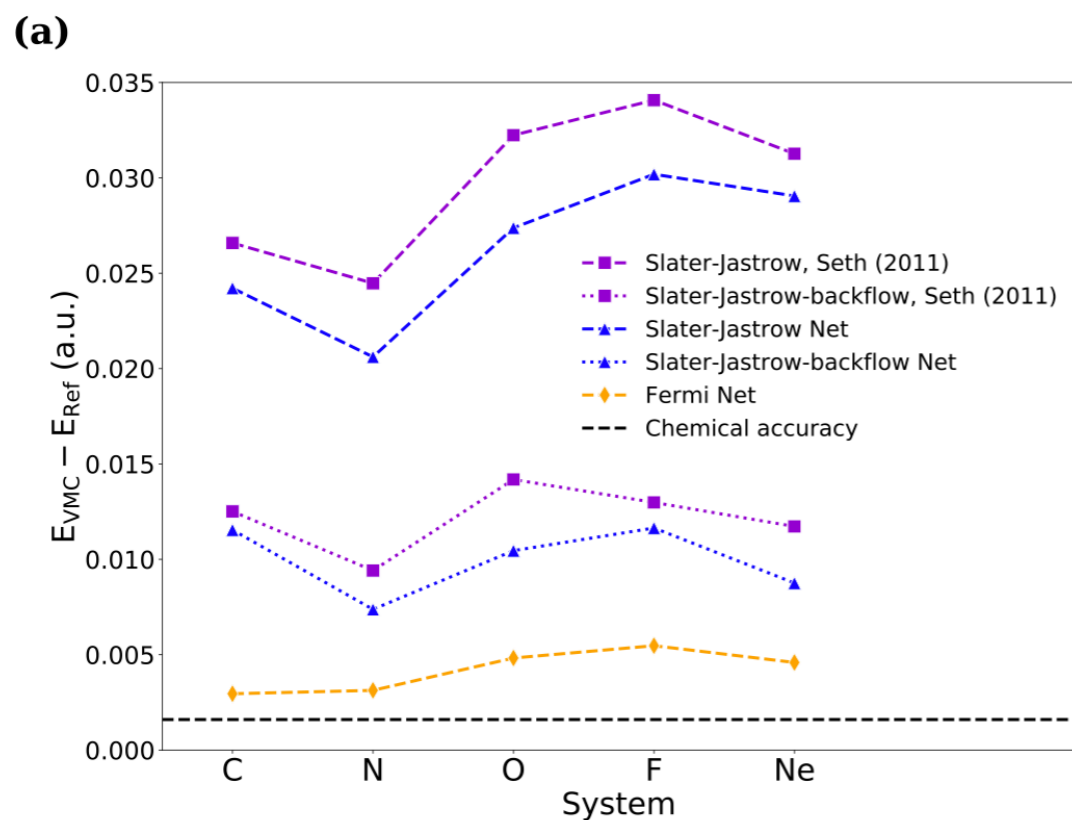
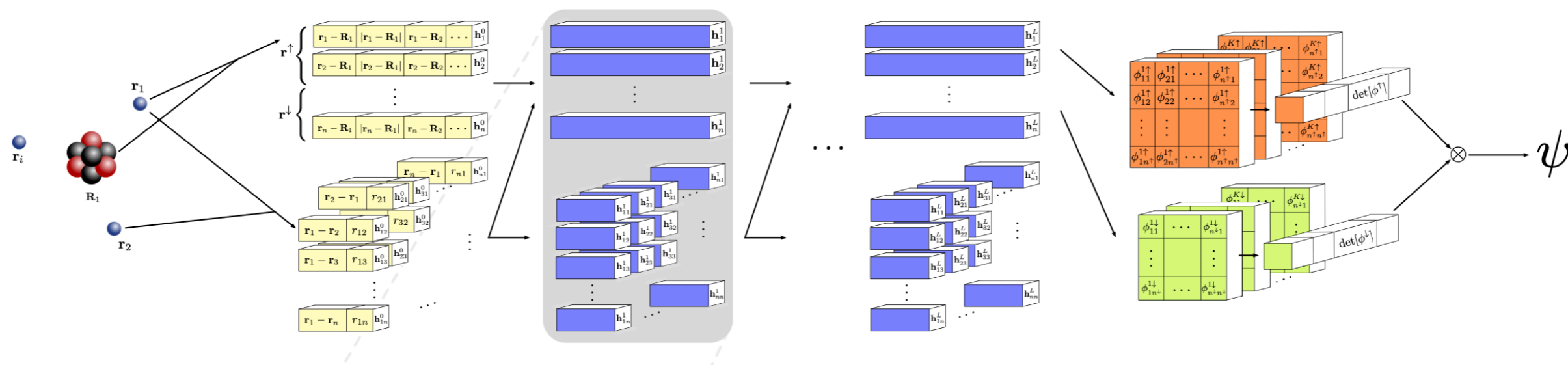
В самом конце - усреднение всех предсказаний шага 4!

Сверхпроводники



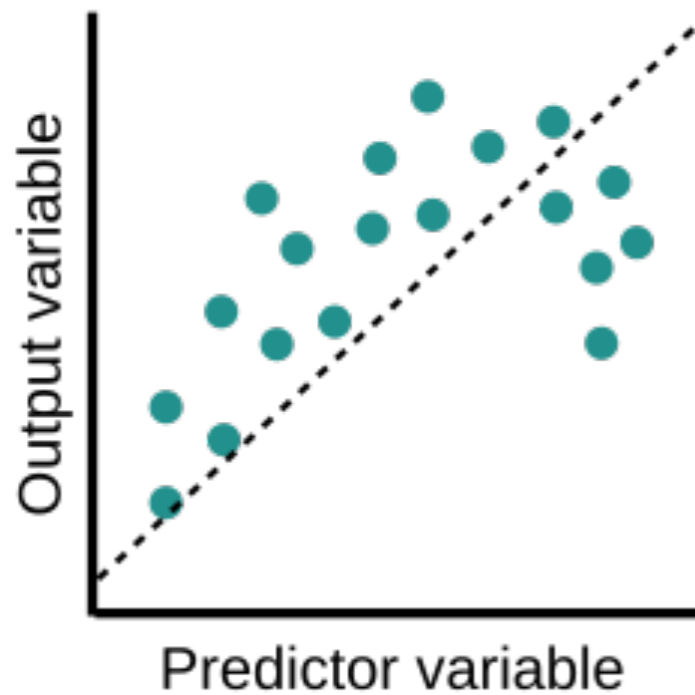
~14000 структур из базы данных SuperCon. Предсказания воспроизводили некоторые из известных физических законов (изотопный эффект, связь с плотностью состояний). Кроме того, не было прямой связи между различными типами сверхпроводимости.

FermiNet

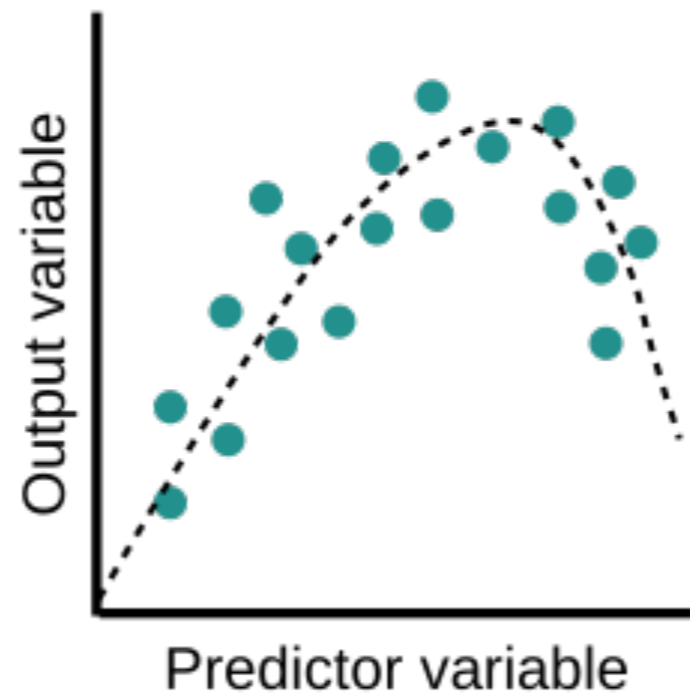


Overfitting & underfitting

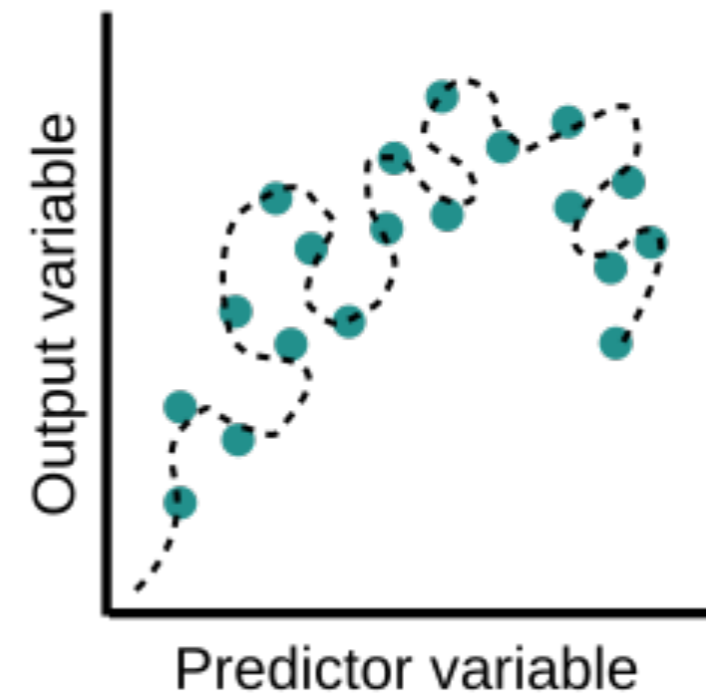
Underfit



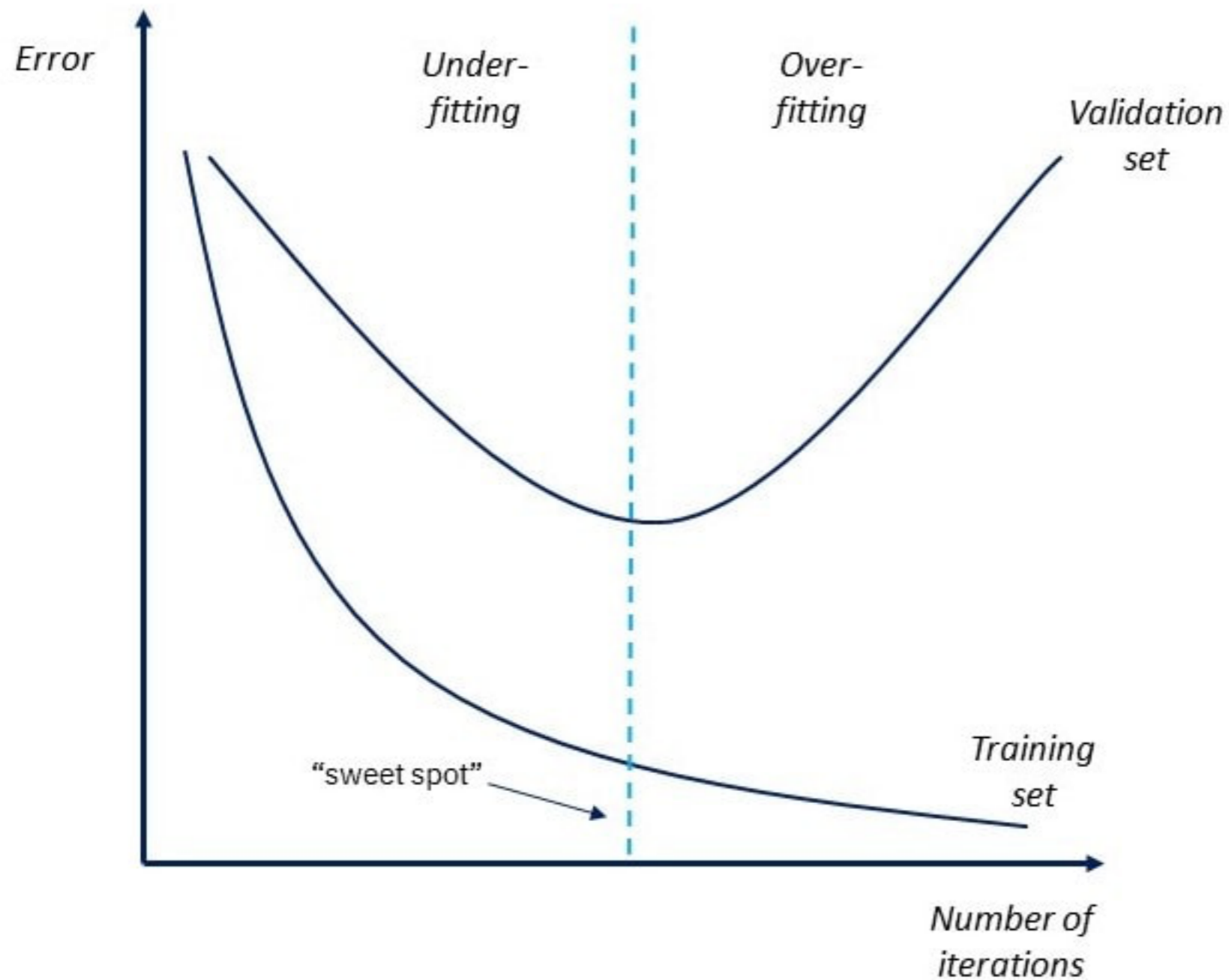
Optimal



Overfit



Деление на train и test



Деление на train и test

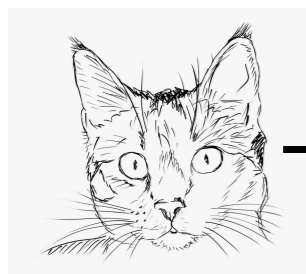
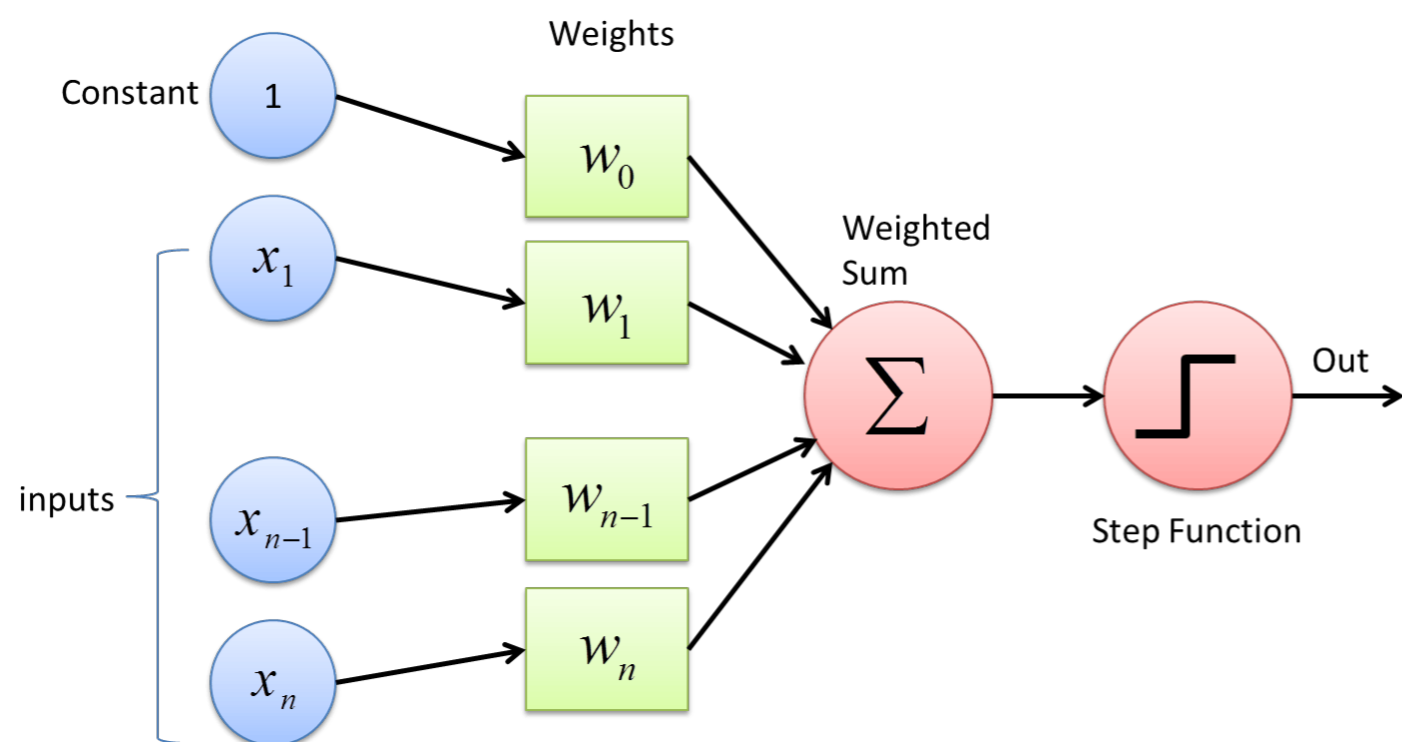
4-fold validation (k=4)



1. Кросс-валидация является более точным подходом к оценке точности модели, но на практике часто затратна.
2. Иногда данные делятся на 3 части: train/validation/test

Персептрон

Персептрон: самая простая нейронная сеть



----- это кошка?

В виде формулы:

$$f(x) = \begin{cases} 1 & \text{если } w^T x + w_0 > 0 \\ 0 & \text{иначе} \end{cases}$$

Как найти w и w_0 ?

$$\text{Loss} = (y - w^T x - w_0)^2$$

$$\frac{\partial \text{Loss}}{\partial w} = 2x(y - w^T x - w_0)$$

$$\frac{\partial \text{Loss}}{\partial w_0} = 2(y - w^T x - w_0)$$

Градиентный спуск: $w_{t+1} = w_t - \alpha \frac{\partial \text{Loss}}{\partial w}$

Бинарная кросс-энтропия

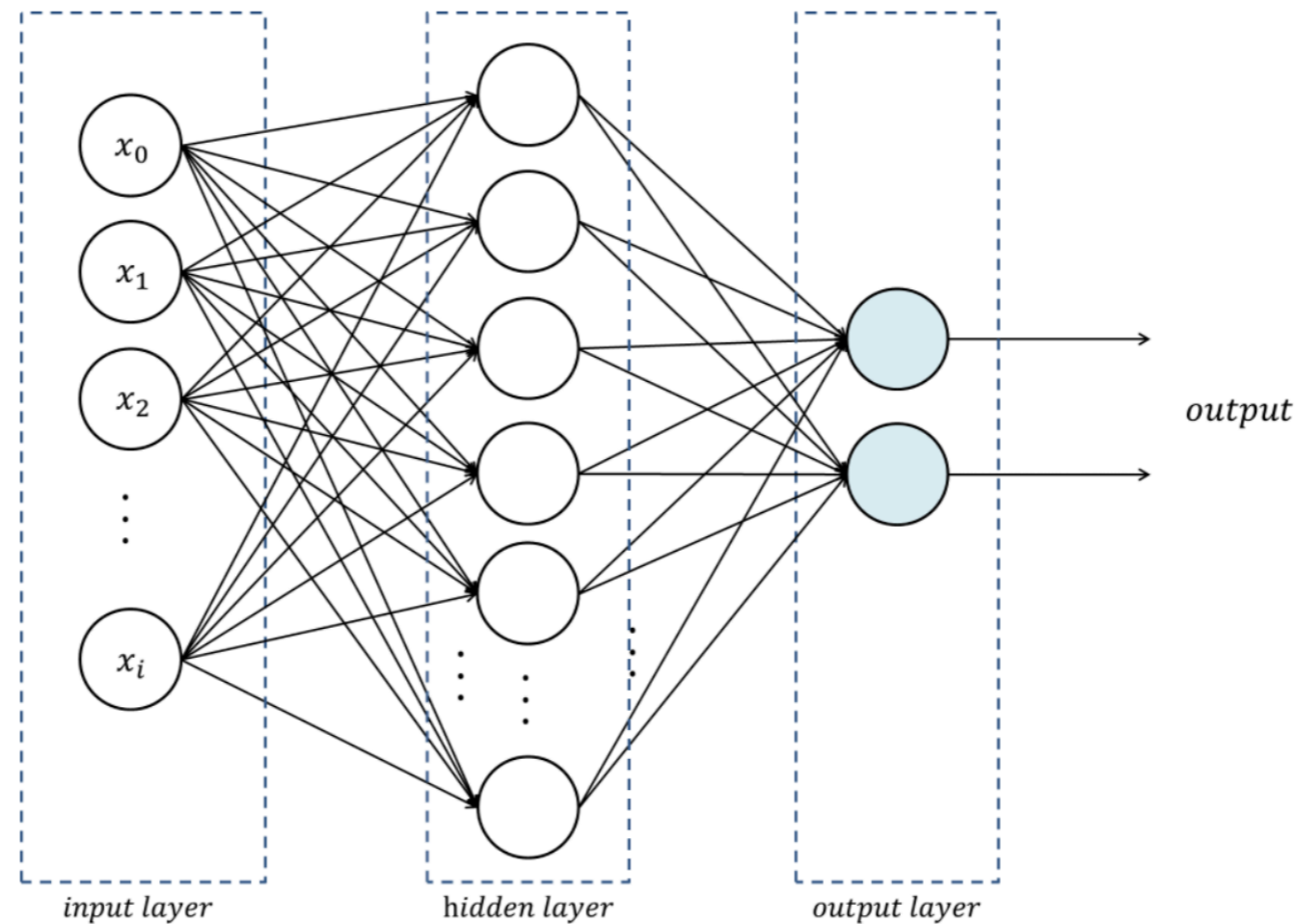
Точность бинарной классификации можно вычислять по следующей формуле:

$$H = -\frac{1}{N} \sum_{i=1}^N y_i \log \tilde{y}_i + (1 - y_i) \log(1 - \tilde{y}_i)$$

Чем меньше значение кросс-энтропии, тем лучше предсказания (тем ближе распределение к правильному)

Много персептронов

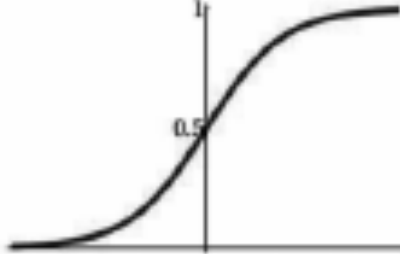
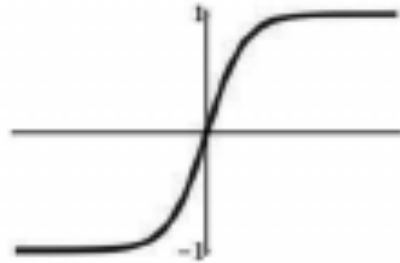
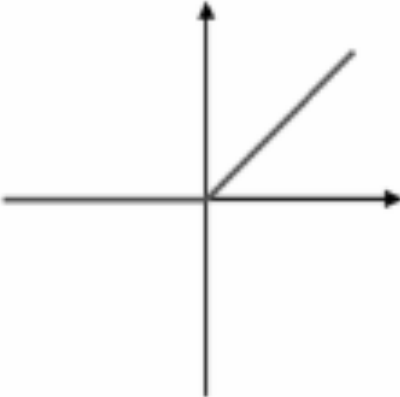
Терминология:
малослойная
(shallow)
нейронная сеть
имеет один
скрытый слой,
глубокая (deep)
имеет несколько
скрытых слоев



Может улавливать более сложные паттерны в данных!

**Теорема Цыбенко (универсальная теорема
аппроксимации)**

Виды нелинейностей

Name	Function	Derivative	Figure
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))^2$	
tanh	$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$f'(x) = 1 - f(x)^2$	
ReLU	$f(x) \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0. \end{cases}$	$f(x) \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0. \end{cases}$	
Softmax	$f(x) = \frac{e^x}{\sum_i e^x}$	$f'(x) = \frac{e^x}{\sum_i e^x} - \frac{(e^x)^2}{(\sum_i e^x)^2}$	

Обратное распространение ошибки

Градиенты для всех весов сети можно считать по цепному правилу

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial C} \frac{\partial C}{\partial w}$$

На практике часто используются адаптивные методы градиентного спуска (Adam, Adadelata, Adagrad и так далее)

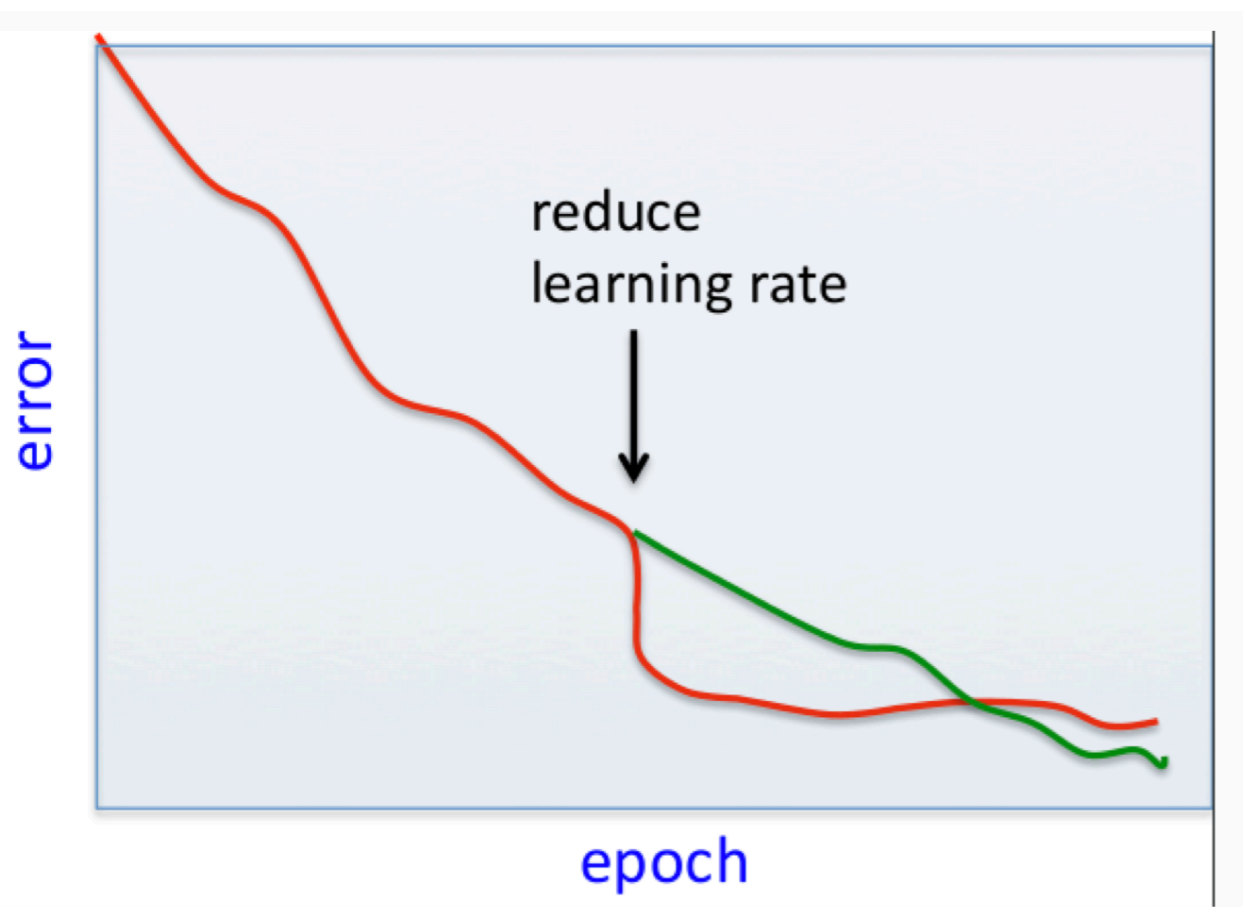
Гарантирует сходимость только к локальному минимуму!

Может наступить паралич сети (vanishing gradients)

Подсчет градиентов и обновление весов гораздо быстрее происходит на GPU/TPU, чем на CPU.

Адаптивные методы

- Что если уменьшать шаг градиентного спуска?



$$\theta_t = \theta_{t-1} - \eta \nabla E(\theta)$$

$$\eta = \eta_0(1 - T/T_0), \quad \eta = \eta_0 \exp(-T/T_0)$$

Адаптивные методы

- Метод моментов (сохраняем часть скорости спуска)

$$u_t = \gamma u_{t-1} + \eta \nabla E(\theta), \quad \theta_t = \theta_{t-1} - u_t$$

- Методы второго порядка? Проблема: гессиан.
- Можно отслеживать изменения отдельных параметров - получаем Adam, Adadelta, Adagrad, что на практике обычно работает лучше всего

Обучение с использованием минибатчей

На практике весь датасет делится на части - минибатчи, содержащие определенное количество данных (например, 100 картинок). Это:

- Уменьшает требования к памяти для подсчета градиентов
- Добавляет стохастичность (поэтому - стохастический градиентный спуск)

Когда все минибатчи проходят через модель - прошла одна эпоха

Сверточные сети



0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

Kernel

0	-1	0
-1	5	-1
0	-1	0

114				

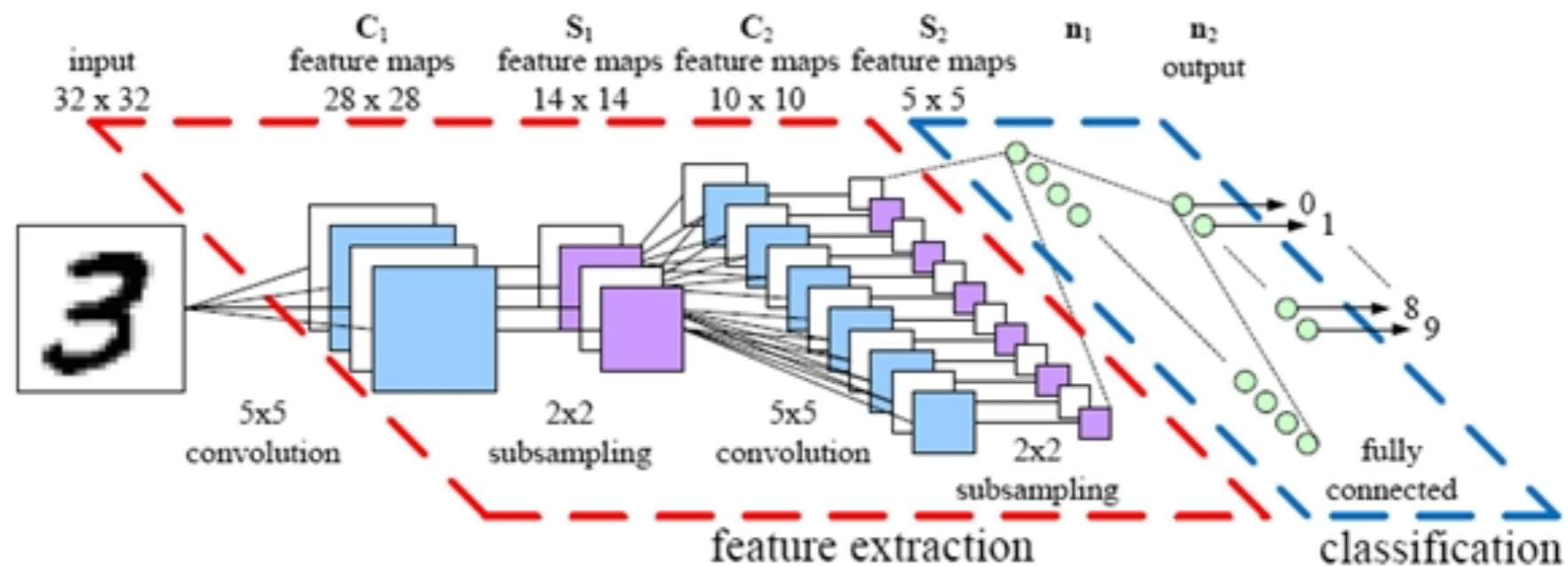
Pooling

От картинки надо как-то прийти к вектору. Можно уменьшать размер постепенно. Разные способы:

- MaxPool
- AvgPool
- MinPool

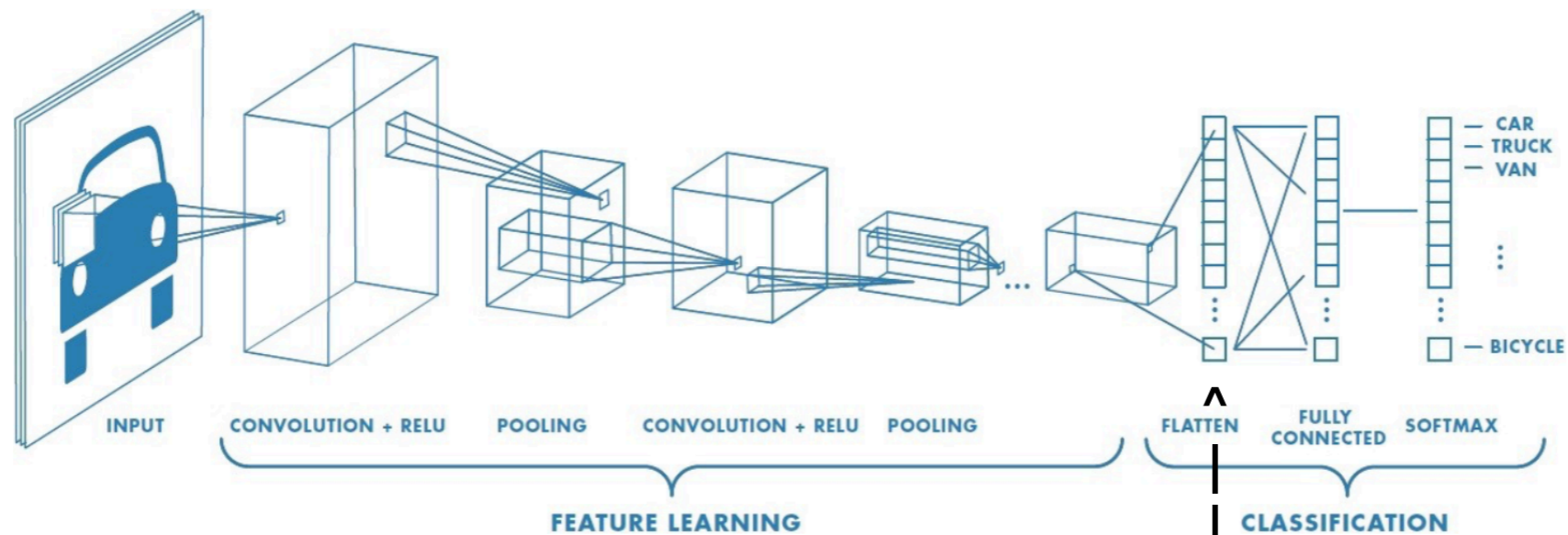
Но есть альтернатива: свертки со сдвигом, это обычно работает лучше

Каналы (channels)



Нейронная сеть может состоять из нескольких параллельных слоев, которые называются каналами (channels)

Сверточные сети



Использует свойства иерархической структуры данных

Использует локальные связи

Дескриптор (будет далее)

BatchNorm

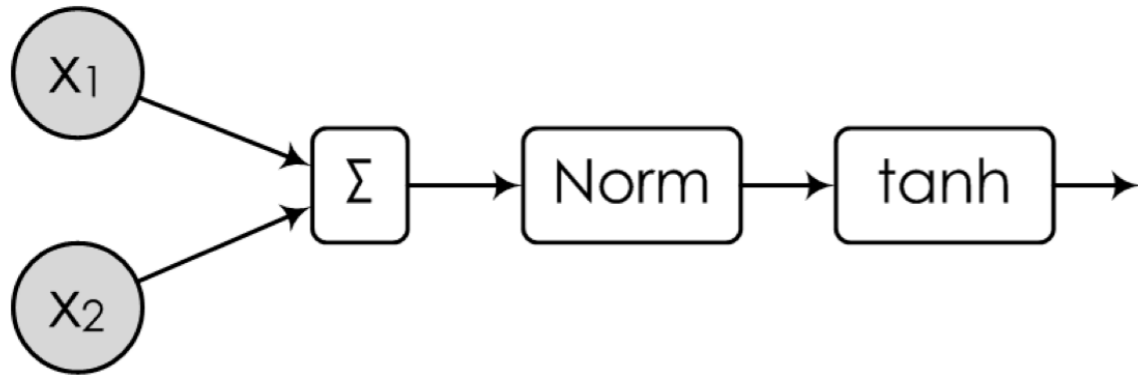
Убирает внутренний сдвиг переменных (internal covariate shift) (но это не точно)

$$y_k = \gamma_k + \frac{x_k - \mathbb{E}[x_k]}{\sqrt{\text{Var}[x_k]}} + \beta_k$$

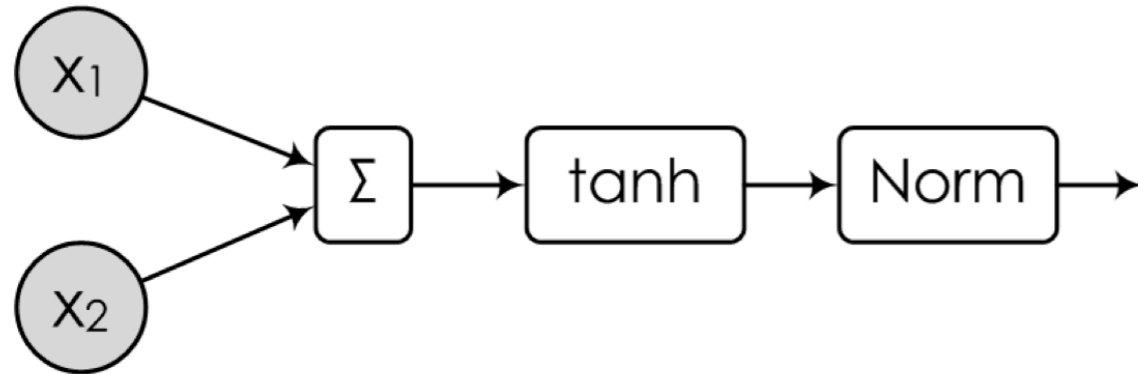
Дополнительные переменные нужны, чтобы не пропали нелинейности

BatchNorm

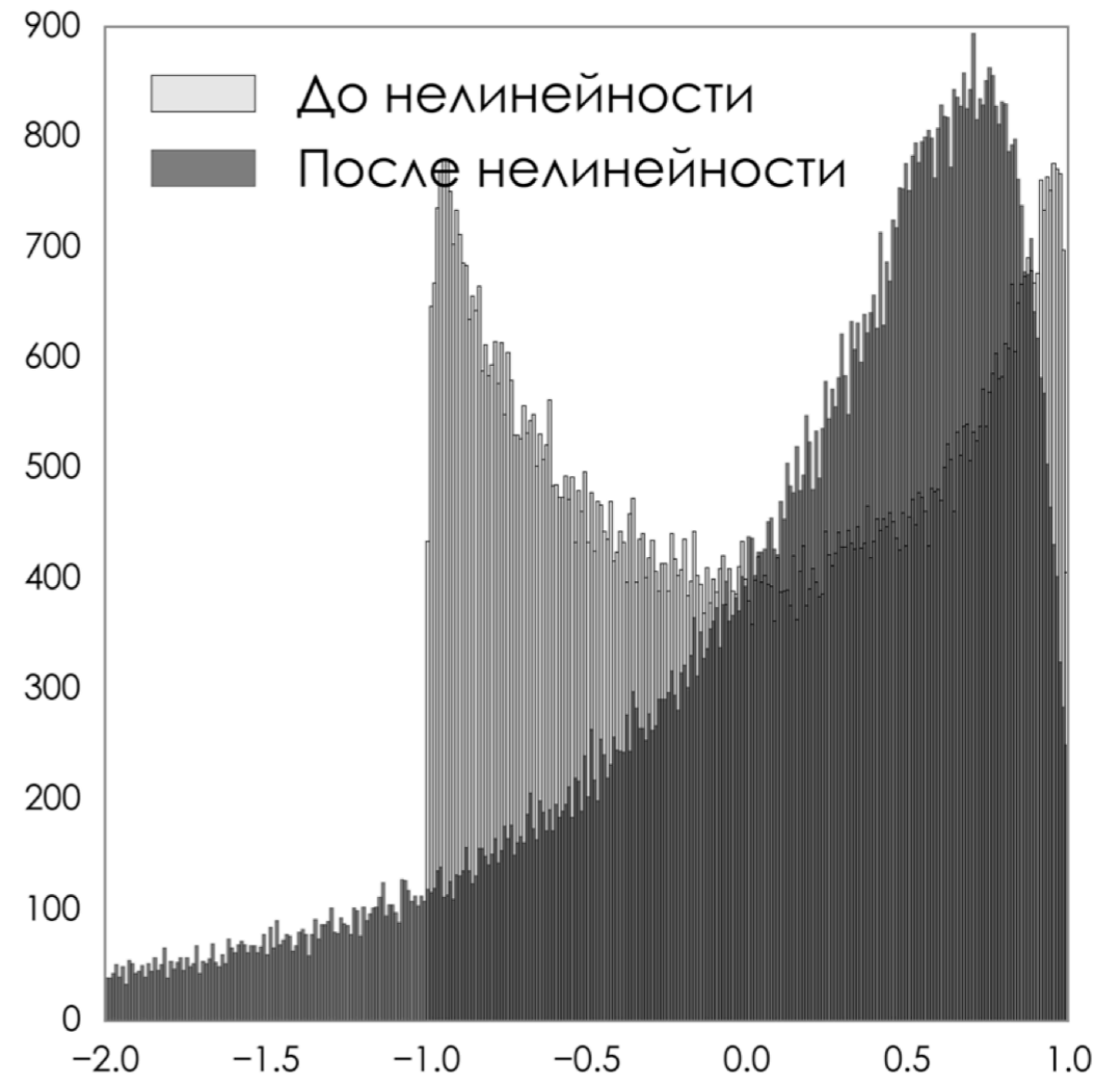
Нормализация до нелинейности



Нормализация после нелинейности

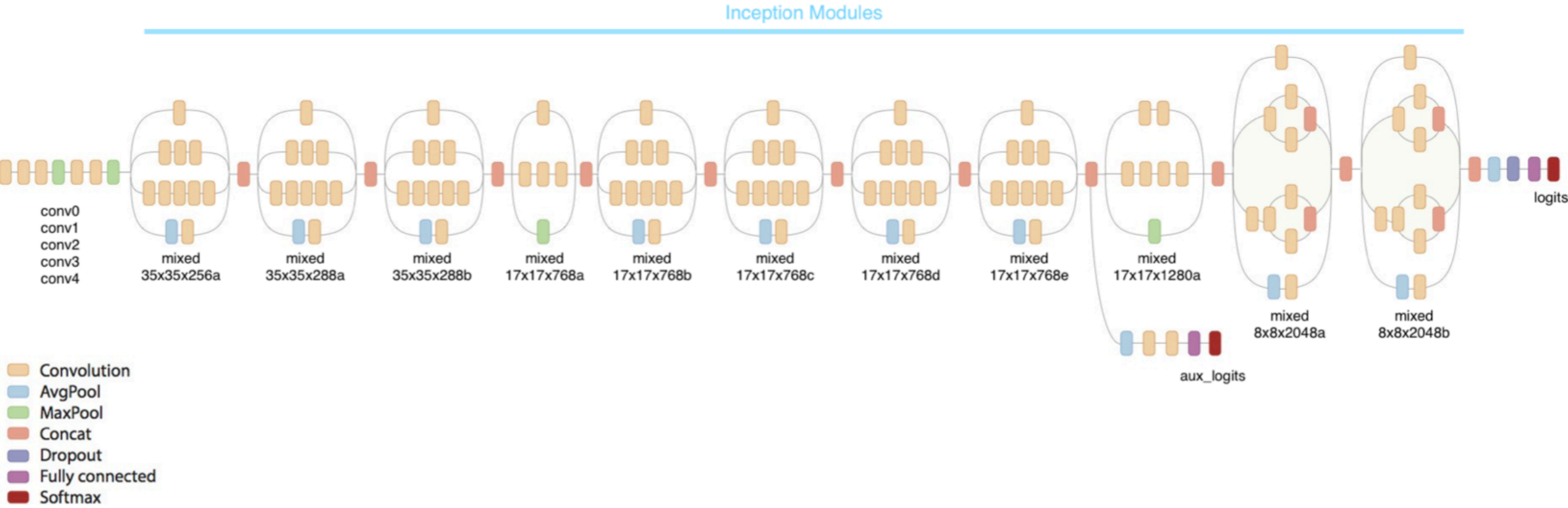


(a)



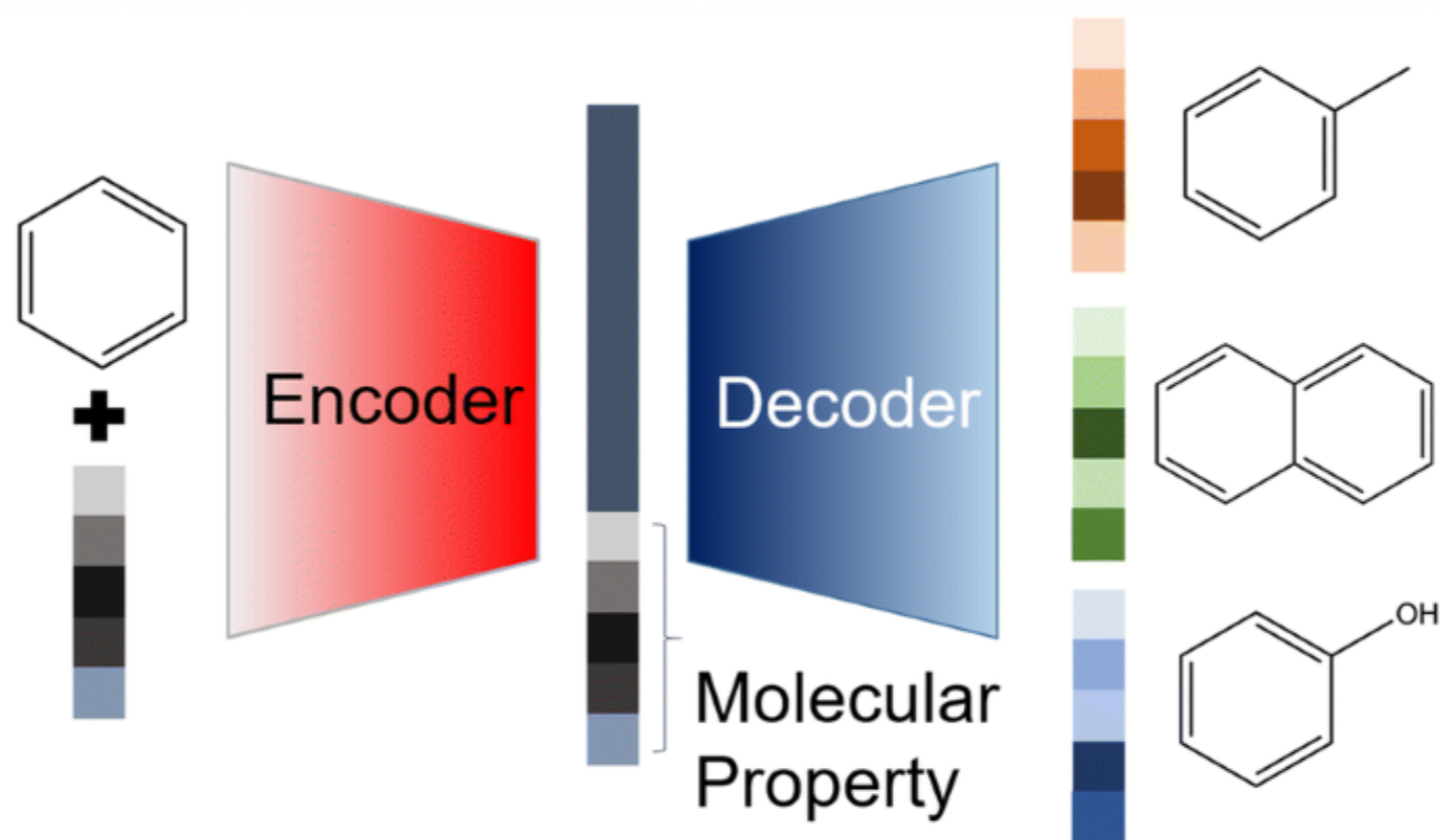
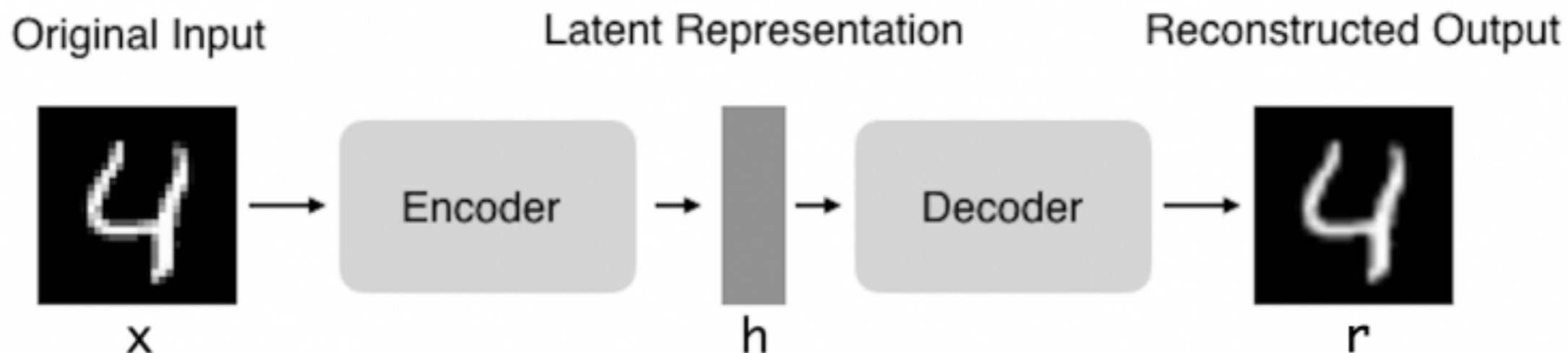
(б)

Сверточные сети

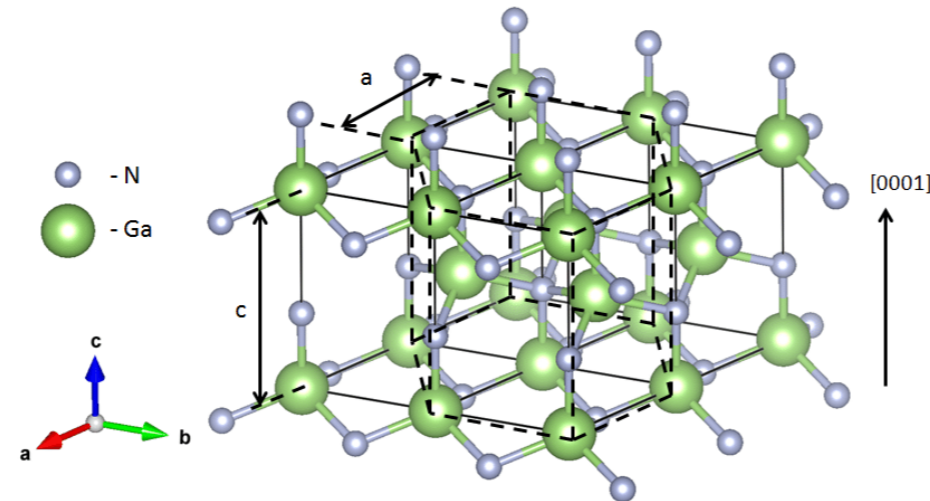


Пример сверточной сети, используемый для классификации изображений с хорошей точностью

Сверточные сети (автоэнкодеры)


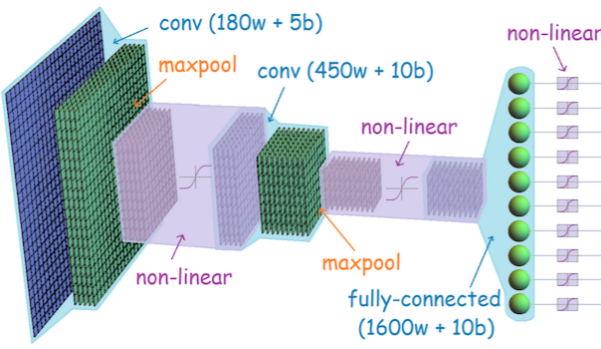
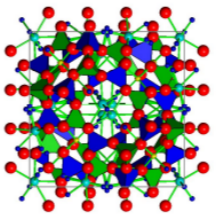
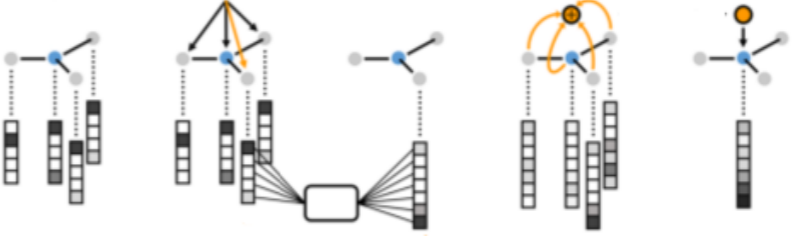


Представление структуры



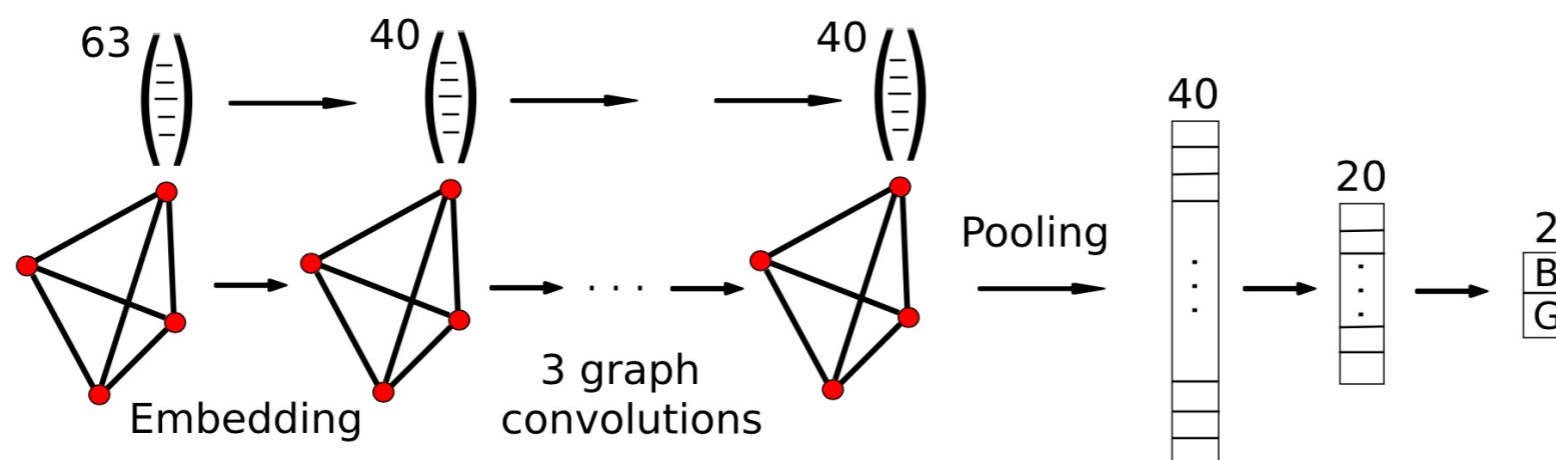
1. Описание по свойствам. Например, массы элементов, период и номер группы, заряд ядер, разные физические свойства
2. Дескриптор. Декартовы координаты не подходят. Должны быть инвариантны относительно вращений, перестановок, перемещения
3. Графовые нейронные сети

Сверточные сети на графах

Обычная сверточная сеть	Графовая сверточная сеть
 <p>→ Предсказание</p>  <p>The diagram illustrates a standard CNN architecture. It starts with an input image (represented by a grid of blue and green pixels). The first layer is a convolutional layer labeled "conv (180w + 5b)". This is followed by a max pooling layer labeled "maxpool". The second convolutional layer is labeled "conv (450w + 10b)". This is followed by another max pooling layer labeled "maxpool". The final layer is a fully-connected layer labeled "fully-connected (1600w + 10b)". Non-linear activation functions, represented by the symbol f, are applied between the convolutional and pooling layers.</p>	 <p>→ Предсказание</p>  <p>The diagram illustrates a graph convolutional network (GCN) architecture. It starts with a graph structure (represented by a cluster of red, blue, and green nodes connected by edges). The graph is processed through a series of layers, each represented by a smaller graph structure. The final output is a prediction, represented by a single node with a yellow circle.</p>

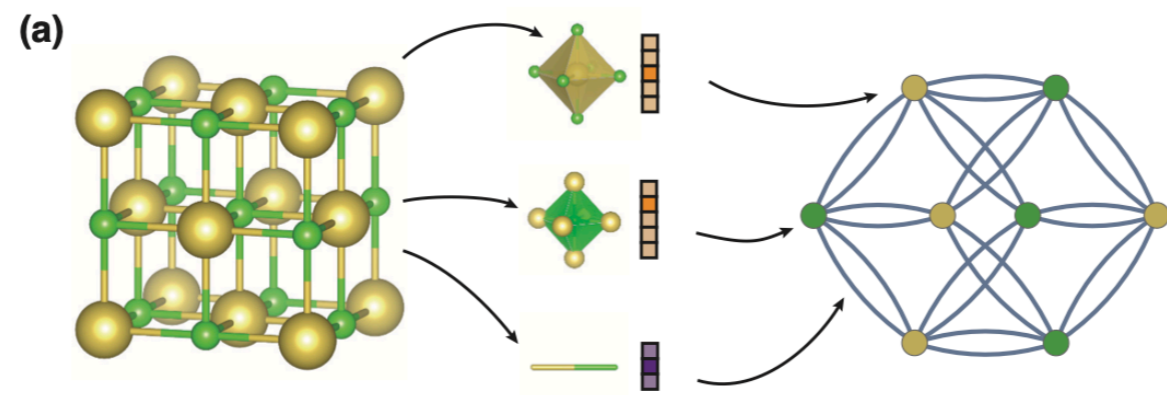
Сверточные сети на графах

- Сверточные сети на графах позволяют обучать модели напрямую на кристаллических структурах

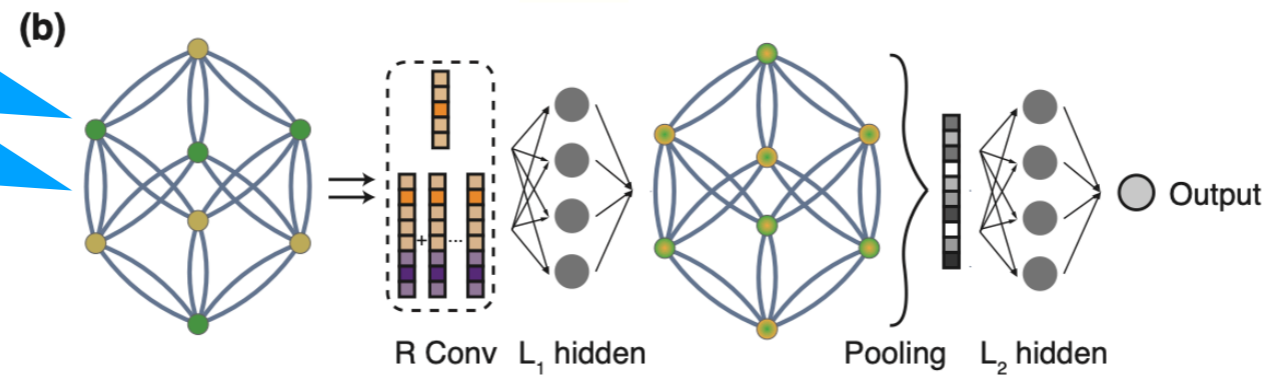


- Предсказанием является определенное свойство кристалла или атома (или много свойств)

Сверточные сети на графах



Atom feature vectors
Bond feature vectors



Атомные и связевые признаки

- Атомный признак (пример):

$$v_i = [0, 0, 0, 1, \dots]$$

- Связевый признак (пример)

$$u_{ij} = \left[\exp\left(-\frac{(d-0 \cdot s)^2}{s^2}\right), \exp\left(-\frac{(d-1 \cdot s)^2}{s^2}\right), \dots, \exp\left(-\frac{(d-R_{cut})^2}{s^2}\right) \right]$$

Слои в сверточная графовой сети

- Линейный слой:

$$v_i^{t+1} = Wv_i^t + b$$

- Сверточный слой:

$$v_i^{t+1} = v_i^t + \sum_j \sigma(z_{ij}W_1 + b_1) \odot \sigma(z_{ij}W_2 + b_2)$$

$$z_{ij} = v_i \oplus v_j \oplus u_{ij}$$

- Pooling:

$$v = \frac{1}{N} \sum_i v_i \quad \text{(Из графа в 1D вектор)}$$

Больше слоев - лучше результаты?

Нет. Реальность: часто после 4-5 слоев предсказания больше не улучшаются

Почему?

- Больше параметров -> overfitting
- Больше слоев -> vanishing gradient -> сложнее обучать
- Oversmoothing (атомные признаки становятся слишком похожи) -> сложнее предсказывать

Графовые сети: плюсы и минусы

Плюсы:

- Сложность не зависит от числа атомов. Более того, предсказания могут распространяться на типы атомов, которые сеть ранее не встречала
- Легко применить подход transfer learning
- Способны предсказывать очень сложные зависимости

Минусы:

- Предсказания могут быть нестабильны к небольшому изменению параметров
- Бывают сложны в обучении