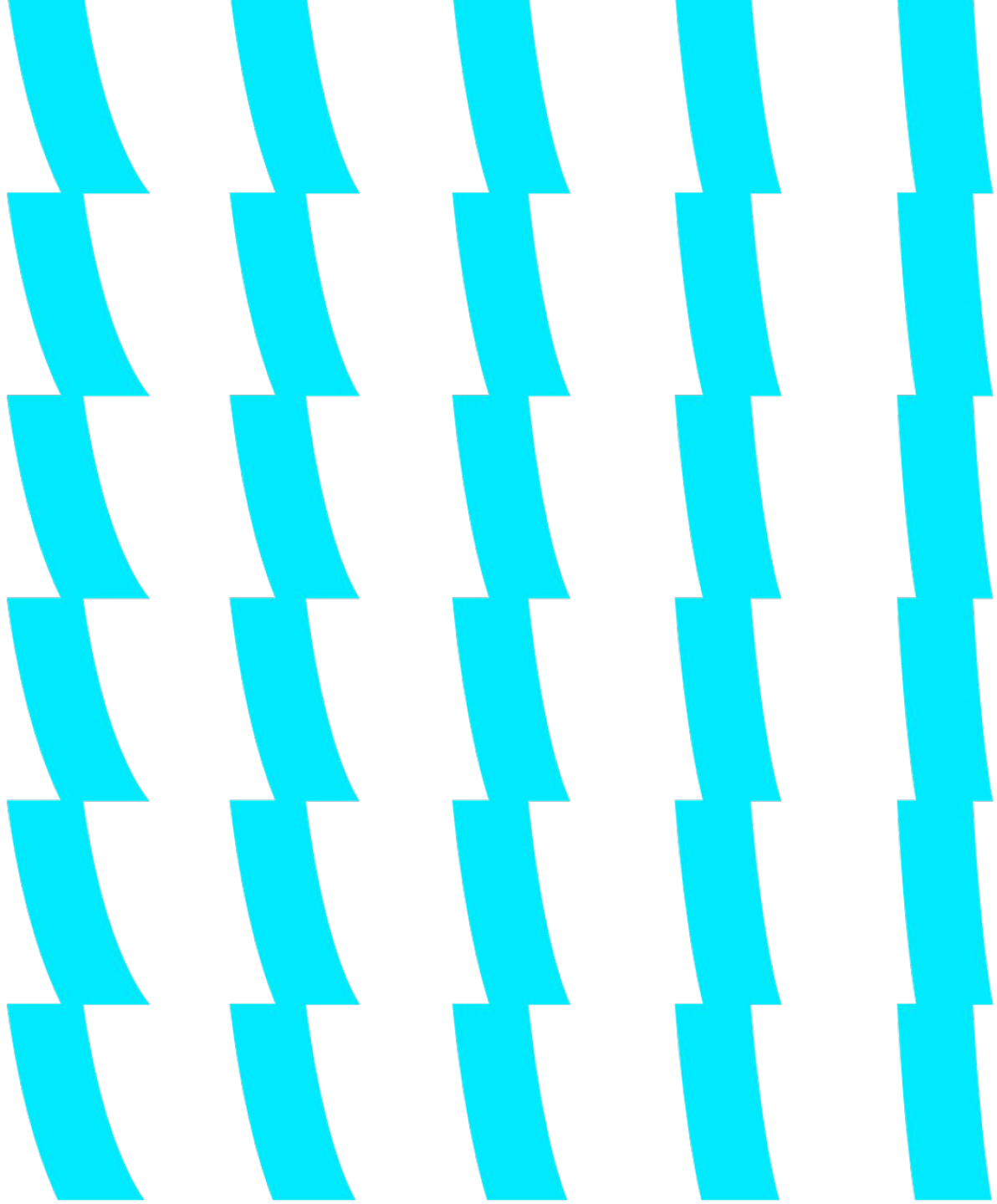
Нейронные сети

Занятие 1
Введение

Программа занятия

1. О курсе
2. О машинном обучении
3. О переобучении
4. О вероятностях
5. О нейронных сетях



Часть 1. О курсе

Дмитрий Соловьев

Денис Клюкин

Всеволод Викулин

Павел Кошкин

Дарья Савинкина

Структура курса

1. Введение
2. Автоматическое дифференцирование, полносвязный слой ❤️ **ДЗ 1**
3. Методы оптимизации
4. Сверточные сети ❤️ **ДЗ 2**
5. Детали обучение глубоких сетей
6. Рекуррентные сети ❤️ **ДЗ 3**
7. Распознавание речи ❤️ **Проект**❤️
8. Обработка естественного языка
9. Генеративно-состязательные сети
10. Синтез речи

Финальный проект

- Соревнование по распознаванию речи с **реальными данными;**
- **Командное** соревнование с **публичной** защитой;
- **Бейзлайн** от команды Маруси;
- **Море** баллов;
- **Бесценный** опыт;

Зачем это мне?

Нейронные сети – основа современного машинного обучения

Преподаватели – эксперты с многолетней практикой

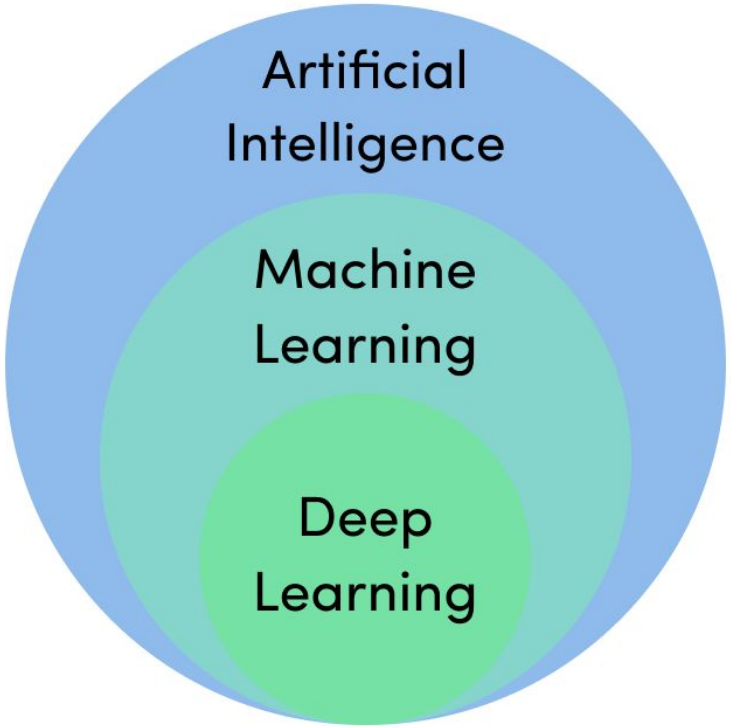
Мы активно ищем специалистов ^_^
5-7 лучших студентов пригласим на собеседование в команду

Баллы, оценки и все все все...

- Максимум 100 баллов за курс (60 домашки, 40 проект).
- Домашки имеют мягкий дедлайн (2 недели от сдачи), после которого сгорает 0.5 баллов за сутки просрочки
- И жесткий дедлайн, (4 недели от сдачи) после которого сдавать нельзя.
- 20 баллов за проект ставятся, если побили бейзлайн и дальше в зависимости от итогового места.
- Проект командный (макс 4 человека) с публичной защитой. Без защиты баллы не ставятся.
- >=50 тройка, >=70 четверка, >= 90 пятерка

Что изучаем?

ИИ – наука об интеллектуальных алгоритмов
Машинное обучение – часть ИИ о построении алгоритмов, способных обучаться
Глубокое обучение – часть машинного обучения, где в качестве алгоритмов используются **нейронные сети**



Примеры применения

- В табличных данных все еще лучше работает бустинг;
- Нейронные сети лучше работают для неструктурированных данных:
  – картинки (классификация, сегментация)
  – тексты (анализ тональности, ранжирование документов)
  – аудио (распознавание речи, синтез речи)
  – графы (предсказание друзей в соц сетях)

Суровая правда

- Не факт, что так можно построить общий ИИ;
- С головным мозгом отношение довольно опосредованное, я сознательно про него не говорю;
- Вся математика придумана в 20 веке;
- Теоретические основы глубокого обучения только только строятся;
- Хайп и бум, потому что стало много данных и технологий;
- Но все равно это очень круто ^_^

Часть 2. О машинном обучении

Постановка задачи

Функционал качества

$$Q(a, X, Y) = \frac{1}{N} \sum_{i=1}^{N} L(a, x_i, y_i), x_i \in X, y_i \in Y$$

Принцип минимизации эмпирического риска:

$a^* = \underset{A}{\operatorname{argmin}} Q(a, X_{train}, Y_{train})$, A – семейство алгоритмов.

Learning = Representation + Evaluation + Optimization

Источник: homes.cs.washington.edu/~pedrod/papers/cacm12.pdf

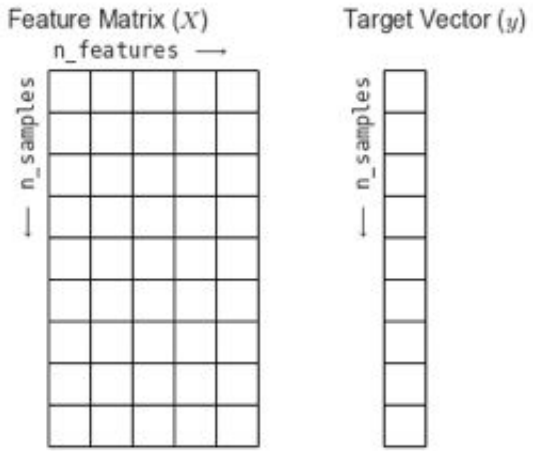
Параметризация или inductive bias

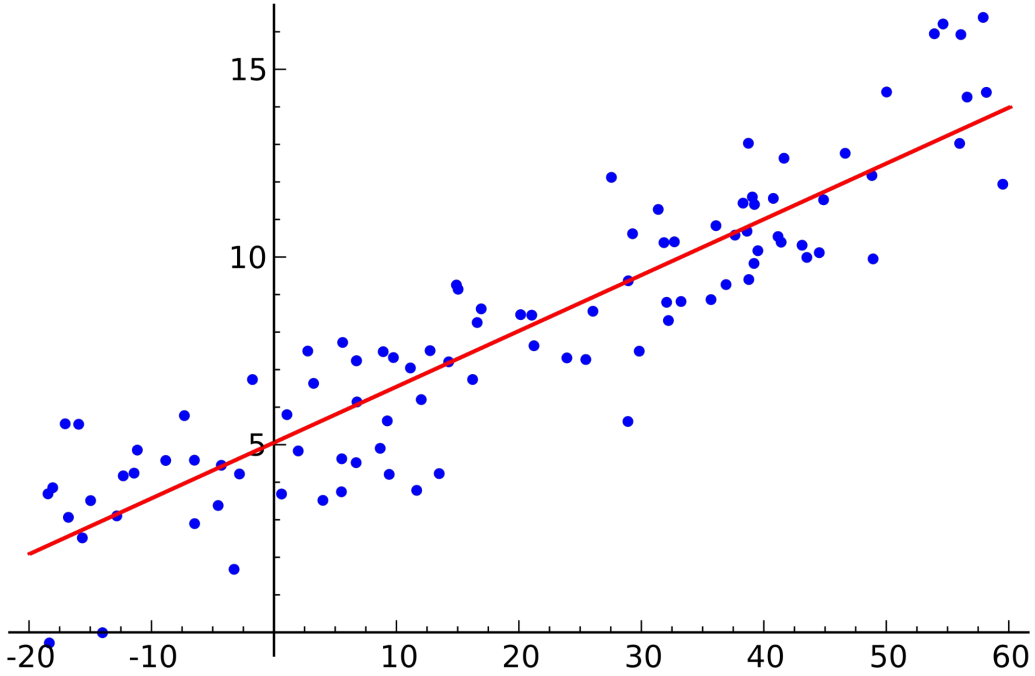
Линейная регрессия

$$Q(X, \boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x_i}\cdot\boldsymbol{w} - y_i)^2 = \frac{1}{N}||X\cdot\boldsymbol{w} - \boldsymbol{y}||^2$$

X — матрица (N, D), w — вектор весов $(D, 1)$, y — вектор ответов $(N, 1)$

$X \cdot \boldsymbol{w}$ — вектор предсказаний $(N, 1)$





NO FREE LUNCH

- a(x) имеющих нулевую ошибку на обучению бесконечно много
- Надо как-то вложить наши представления о структуре того, как может выглядеть решение иначе мы потеряемся в куче возможных решений
- Задача машинного обучения **некорректно поставлена**;

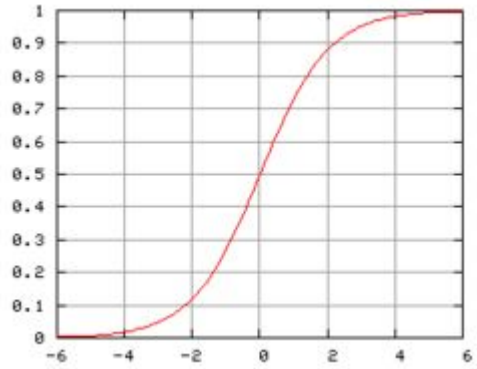
https://ru.wikipedia.org/wiki/Корректно_поставленная_задача

Логистическая регрессия

Решаем классификацию, хотим, чтобы a(x) выдавала вероятность;

$$a(x) = p(y = 1|x) = sigm(\mathbf{wx}).$$

$$sigm(z) = \frac{1}{1 + exp(-z)}$$



Для многоклассовой классификации аналог сигмоида softmax

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

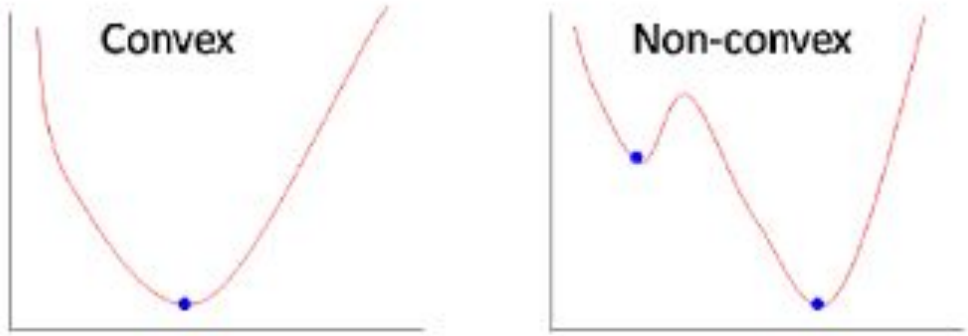
$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\mathsf{T}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\mathsf{T}\mathbf{w}_k}}$$

Итоговый функционал качества

$$Q(X, \mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}[y_i = j]\log p(y = j|x_i)$$

Optimization

- В итоге: $\mathbf{w}^* = argmin_{\mathbf{w}} Q(X, \mathbf{w})$
- Иногда можно руками посчитать
- Можно делать **градиентный** спуск (что это такое?)
- Оптимизация может давать глобальный оптимум;



Градиентный спуск

Антиградиент функции показывает направления наискорейшего убывания функции.

Ищем минимум $Q(w)$

Выбрать начальную длину шага α_0, начальное приближение w_0

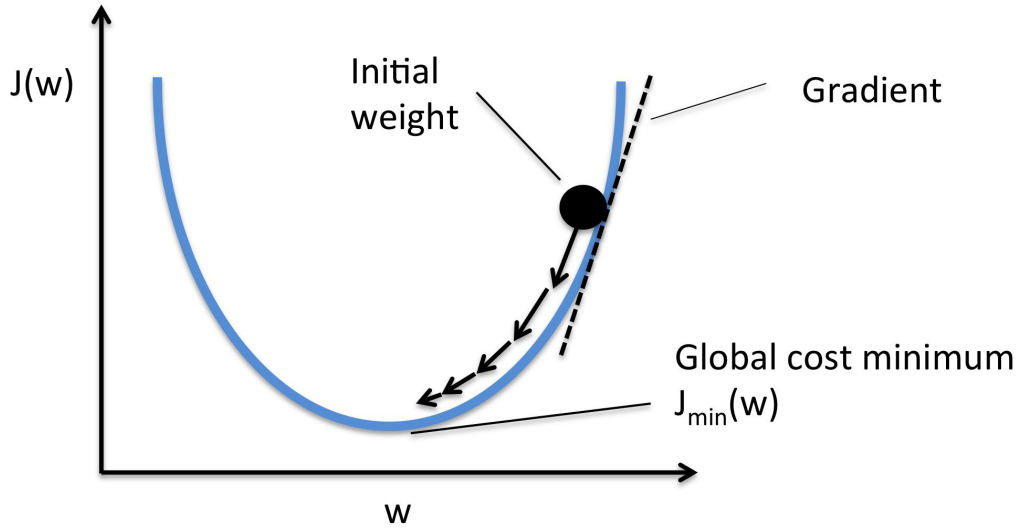
$$w_{new} = w_{old} - \alpha \nabla_w Q(w_{old})$$

$$\alpha = f(k), k = k + 1$$

Повторять (2), (3) до сходимости $Q(w)$ или w

$$f(k) = \alpha_0, f(k) = \frac{\alpha_0}{k}, f(k) = \frac{\alpha_0}{k^p}, \ldots$$

Глобальный минимум или локальный?



Стохастический градиентный спуск

В задачах машинного обучения, оптимизируемая функция имеет специальный вид:

$$Q(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{w}, \boldsymbol{x_i}, y_i)$$

$$\nabla_w Q(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} \nabla_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{x_i}, y_i)$$

Выбрать начальный шаг α_0, начальное приближение $\boldsymbol{w}_0$, размер батча n

Выбрать случайно $\{j_1, i_2, \ldots j_n\}$

Оценить градиент $\nabla_{\boldsymbol{w}} Q^*(w_{old}) = \frac{1}{n}\sum_{j=1}^{n} \nabla_{\boldsymbol{w}} L(\boldsymbol{w_{old}}, \boldsymbol{x_j}, y_j)$

$\boldsymbol{w_{new}} = \boldsymbol{w_{old}} - \alpha \nabla_{\boldsymbol{w}} Q^*(\boldsymbol{w_{old}})$

$\alpha = f(k), k = k + 1$

Повторять (2 - 5) до сходимости

Обычно перемешивают всю выборку случайно.

Когда прошли все выборку, то говорят, что прошла **одна эпоха**

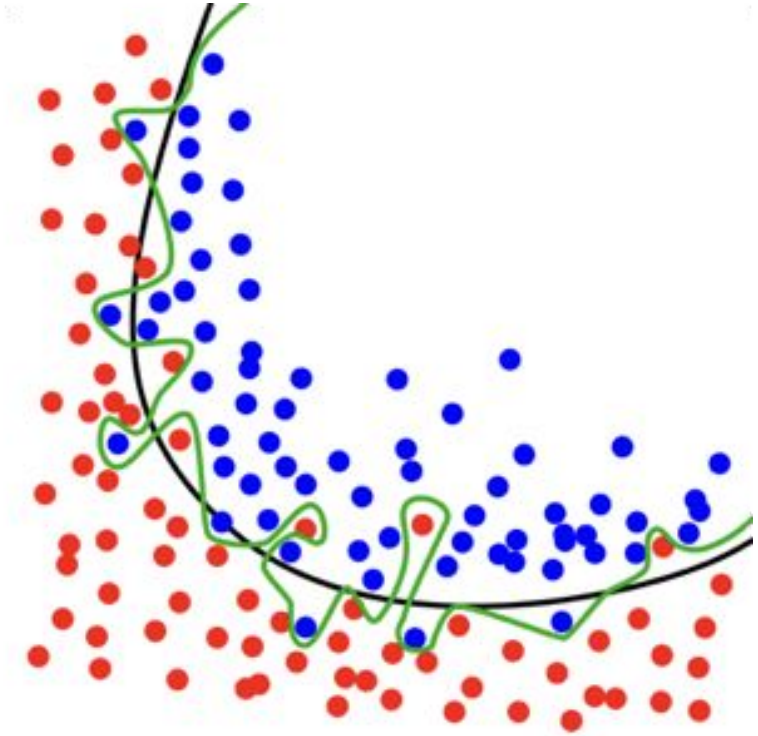
$n = N$ — градиентный спуск (Gradient Descent, GD)

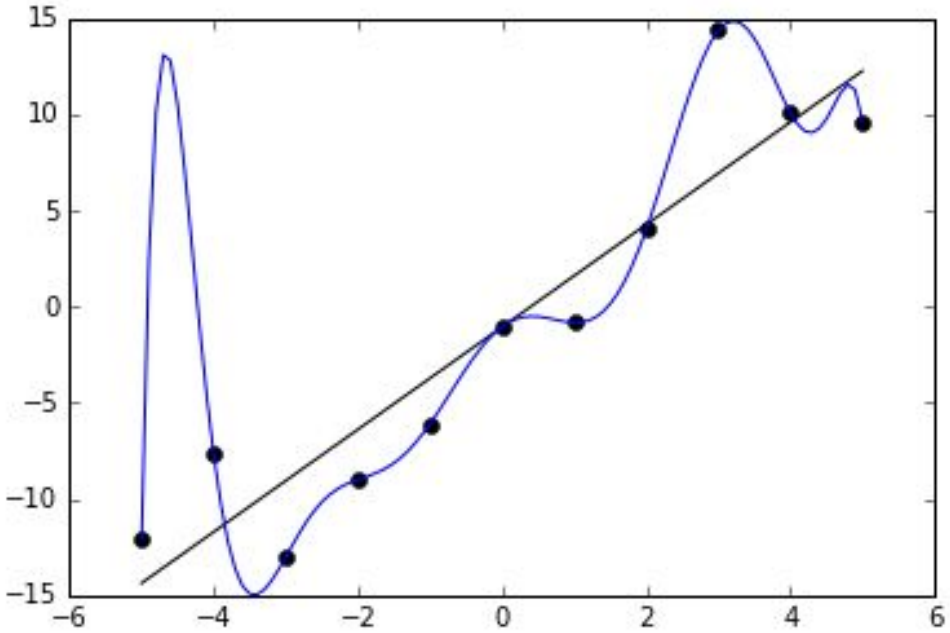
$n = 1$ — стохастический градиентный спуск (Stochastic Gradient Descent, SGD)

$n > 1, n < N$ — мини-батч градиентный спуск (Mini-Batch Gradient Descent, MBGD)

Часть 3. О переобучении

Жуткие картинки





Что же это такое?

Проблема **переобучения** - значения $Q(a, X_{train}, Y_{train})$ значительно меньше, чем значение $Q(a, X_{test}, Y_{test})$ на контрольной выборке.

Если $Q(a, X_{test}, Y_{test})$ примерно равна $Q(a, X_{train}, Y_{train})$, то говорят, что алгоритм обладает **обобщающей способностью**.

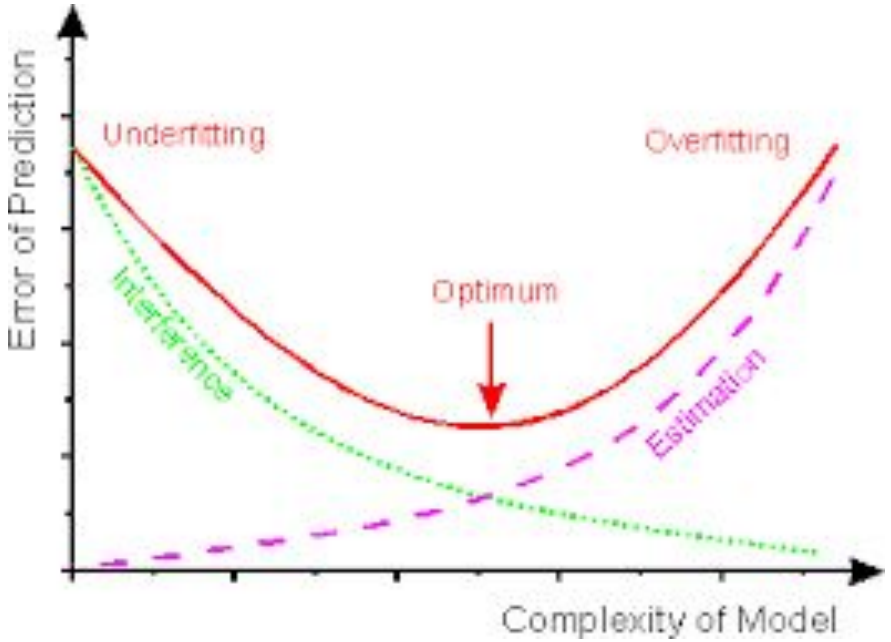
Переобучение есть всегда из-за индуктивной постановки задачи - нахождение закона природы по неполной выборке!
Но еще она может быть из-за излишней **сложности** модели.

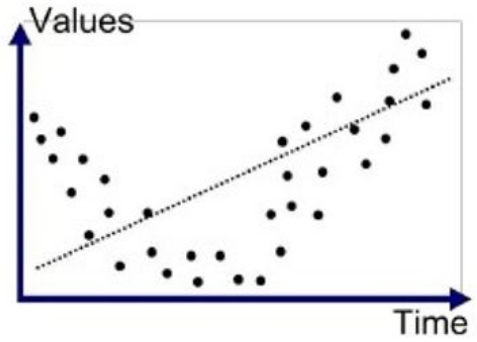
Что же это такое?

- what you see is what you get – вы минимизируете ошибку на обучении, вполне не факт, что ошибка на **тестовой** выборке будет такая же
- Какие гарантии? Очень мало: (https://arxiv.org/pdf/2105.14368.pdf)

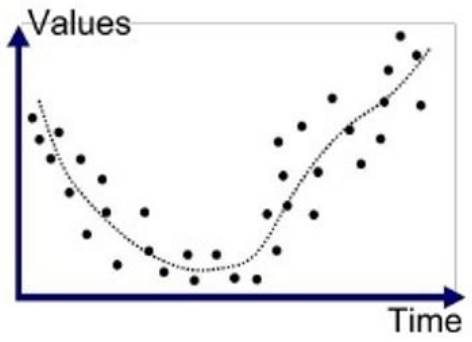
$$\underbrace{\mathcal{R}(f)}_{\text{expected risk}} - \underbrace{\mathcal{R}_{\text{emp}}(f)}_{\text{empirical risk}} < \underbrace{O^*\left(\sqrt{\frac{cap(\mathcal{H})}{n}}\right)}$$

Overfitiing underfitting

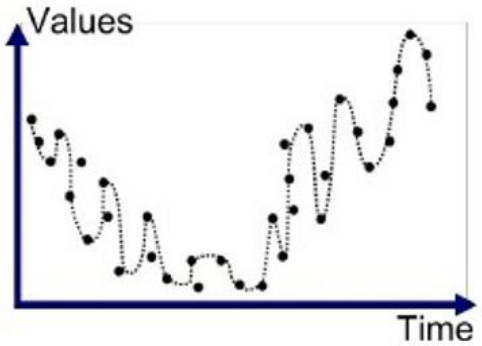




Underfitted



Good Fit/Robust



Overfitted

на самом деле все сложнее

https://arxiv.org/abs/1912.02292

Как бороться с переобучением?

Глобально всего **два варианта**

1) Уменьшать сложность модель (меньше параметров, регуляризация)
2) Увеличивать число данных (размечать новые или **аугментация**)

Часть 4. О вероятностях

Функционал среднего риска

Считаем, что наши объекты и ответы описываются распределением p(x, y), а у нас есть экземпляры из него (обучающая выборка)

Обучающая выборка взята **случайно и независимо** из этого распределения

Функция потерь $L(a, x, y)$ — неотрицательная функция, показывающая величину ошибки алгоритма a на объекте x с ответом y.

Функционал среднего риска — мат. ожидание функции потерь

$$R(a) = \int\int L(a, x, y)p(x, y)dxdy$$
$$a^* = \operatorname{argmin} R(a)$$

Было:

$$Q(a, X, Y) = \frac{1}{N}\sum_{i=1}^{N} L(a, x_i, y_i), x_i \in X, y_i \in Y$$
$$a^* = \operatorname{argmin} Q(a, X_{train}, Y_{train})$$

Стало:

$$R(a) = \int\int L(a, x, y)p(x, y)dxdy$$
$$a^* = \operatorname{argmin} R(a)$$

Суть почти всего МЛ почти в одном слайде

- Считаем, что наши объекты лежат в распределении p(x, y);
- Средний риск посчитать не можем, так как не знаем функции распределение;
- Независимо просим от оракула сэмплы из этого распределения;
- Оцениваем мат. ожидание как среднее, чем больше объектов тем точнее оценка (ЗБЧ);
- Помним про what you see is what you get

Что за люди придумывают L(a, x, y)

- Может казаться, что люди просто так придумывают функции ошибки для эмпирического риска;
- Строим функцию правдоподобия, вероятность пронаблюдать выборку;

$$L(\theta) = p(X|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$$

С суммой, как правило, удобнее работать. (Почему?)

$$\log L(\theta) = \sum_{i=1}^{N} \log p(x_i|\theta)$$

Правильные параметры те, которые максимизируют $L(\theta)$

Орел/Решка?

- Бросаем монетку 10 раз, 7 раз выпал орел, какова вероятность выпадения орла?
- Пускай вероятность выпадения орла θ

$$L(\theta) = p(Y|\theta) = \prod_{i=1}^{N} p(y_i|\theta), \text{ где}$$

$p(y_i|\theta) = \theta$, если $y_i = 1$

$p(y_i|\theta) = 1 - \theta$, если $y_i = 0$

Так как $[y_i = 1] \equiv y$ и $[y_i = 0] \equiv 1 - y$

$$L(\theta) = \prod_{i=1}^{N} \theta^{[y_i=1]} \cdot (1-\theta)^{[y_i=0]} = \prod_{i=1}^{N} \theta^{y} \cdot (1-\theta)^{1-y}$$

$$\log L(\theta) = \sum_{i=1}^{N} y_i \log\theta + (1-y_i)\log(1-\theta)$$

$$\log L(\theta) = \sum_{i=1}^{N} y_i \log\theta + (1-y_i)\log(1-\theta)$$

$$\frac{\partial \log L(\theta)}{\partial\theta} = \sum_{i=1}^{N} \frac{y_i}{\theta} - \frac{1-y_i}{1-\theta} = \sum_{i=1}^{N} \frac{y_i - \theta y_i - \theta + \theta y_i}{\theta(1-\theta)} = 0$$

$$\sum_{i=1}^{N} (y_i - \theta) = \sum_{i=1}^{N} y_i - N\theta = 0$$

$$\theta = \frac{\sum_{i=1}^{N} y_i}{N}, \text{ то есть просто процент орлов!}$$

Логистическая регрессия

Подбрасываем монетку, получаем серию из единиц и нулей.
Правдоподобие:

$$L(\theta) = \prod_{i=1}^{N} p(y_i \,|\, x, \theta)$$

Когда моделировали монетку, считали что:

$p(y_i \,|\, x, \theta) = \theta,$ если $y_i = 1$

$p(y_i \,|\, x, \theta) = 1 - \theta,$ если $y_i = 0$

То есть каждый бросок мы раньше никак не описывали, считали их все равнозначными.

Двухклассовая классификация.
Понаблюдали **обучающую выборку** из единиц и нулей, правдоподобие выборки:

$$\prod_{i=1}^{N} p(y \,|\, x, \theta)^{[y_i=1]} \cdot (1 - p(y \,|\, x, \theta))^{[y_i=0]} =$$

$$\prod_{i=1}^{N} p(y \,|\, x, \theta)^{y} \cdot (1 - p(y \,|\, x, \theta)^{1-y}$$

$$\log L(\theta) = \sum_{i=1}^{N} y_i \log p(y \,|\, x, \theta) + (1 - y_i) \log(1 - p(y \,|\, x, \theta))$$

Используем сигмоид, чтобы предки перевести в вероятность:

$$p(y \,|\, x, \theta) = \frac{1}{1 + exp(-a(x, \theta))}$$

Если $a(x, \theta) = x^T \theta,$ то получаем логистическую регрессию!
$\theta_{ML} = \operatorname{argmax} \log L(\theta)$, находим оптимум градиентным спуском

Регуляризация

Метод максимума правдоподобия “хорошо работает”, только если число объектов много больше числа параметров. А если вы 3 раза подбросили монетку?

Спасает формула Байеса!

Хотим делать не ММП, а MAP, чтобы уменьшить переобучение.

$$\theta_{MAP} = \text{argmax} L(\theta)p(\theta) = \text{argmax} \log L(\theta) + \log p(\theta)$$

Пусть каждая компонента $p(\theta)$ распределена нормально с нулевым средним и все компоненты независимы и имеют одинаковую дисперсию σ^2:

$$p(\theta) = \prod_{j=1}^{D} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{\theta_j^2}{2\sigma^2})$$

$$\theta_{MAP} = \text{argmax} \sum_{i=1}^{N} y_i \log p(y|x,\theta) + (1-y_i)\log(1-p(y|x,\theta)) - \sum_{j=1}^{D} C\frac{\theta_j^2}{\sigma^2}$$

$$\theta_{MAP} = \text{argmax} \sum_{i=1}^{N} y_i \log p(y|x,\theta) + (1-y_i)\log(1-p(y|x,\theta)) - \frac{C}{\sigma^2}.||\theta||^2$$

Берем с минусом, получаем loglocc с l2 регуляризацией!

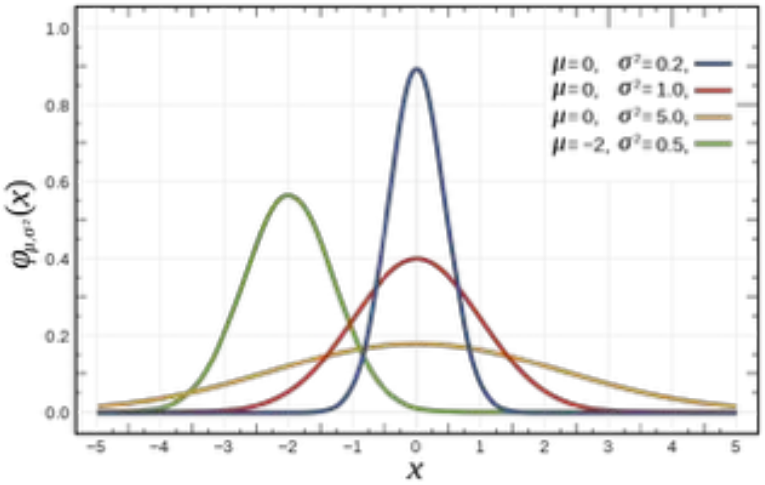
Какой смысл у дисперсии σ^2?

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)}$$

$$\theta_{MAP} = \text{argmax} p(\theta|Y) = \text{argmax} p(Y|\theta)p(\theta)$$

Регуляризация

Мы хотим, чтобы наши веса были маленькими, потому что большие веса — источник переобучения. Чем меньше сигма, тем уже нормальное распределение, то есть тем сильнее мы регуляризуем



Суть всего МЛ в одном слайде

- Все наши функционалы эмпирического риска это логарифм функции правдоподобия;
- Вся наша регуляризация это добавление априорного представления о структуре решения;
- Вся наша параметризация это то, как мы моделируем p(y|x);
- Это и есть inductive bias;
- Все наше желание найти минимум функционала это ММП;
- При стремлении выборки к бесконечности, оценка ММП сходится к истине;
- Только оптимизируем мы все равно плохо в таких больших пространствах :(Глобальный оптимум будет для выпуклых правдоподобий;
- При стремлении выборки к бесконечности, оценка эмпирического риска сходится к среднему риску по ЗБЧ;
- Да, да, в нейронках все **тоже самое,** только разный inductive bayes;
- Если Вы это поняли, Вы просто молодец;
- А если не поняли, можете задать вопрос ^_^

Часть 5. О нейронных сетях

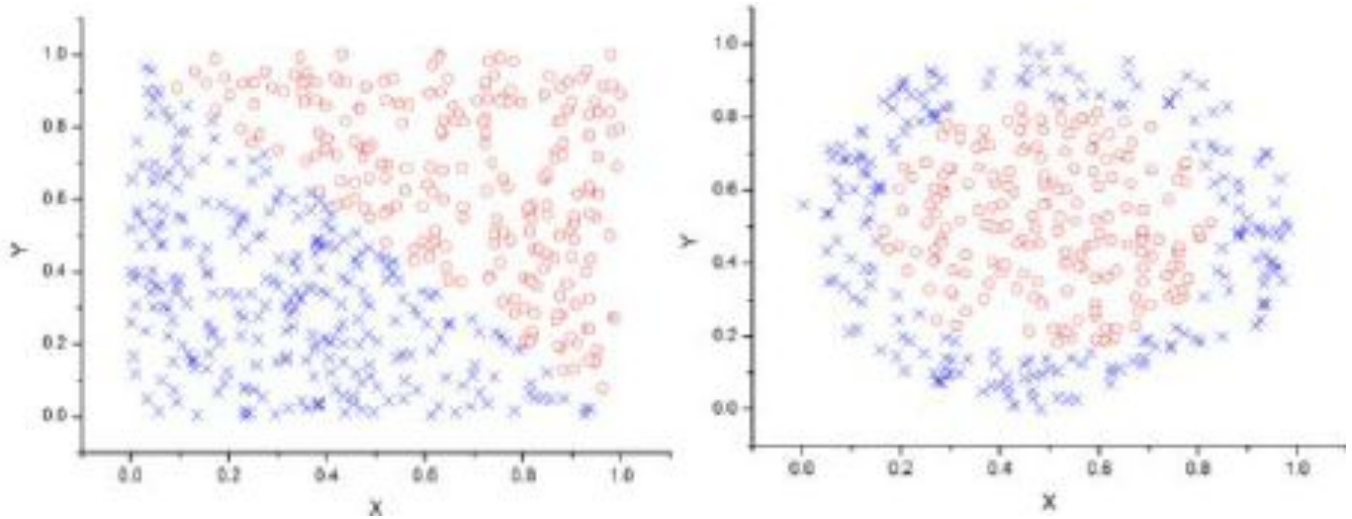
Проблемы текущего подхода

- Наш текущий inductive bias, что выход зависит линейно от входа (или разделяющая поверхность как в логистической регрессии)
- Как делать нелинейность?
- Текущие варианты решения:

<ЗДЕСЬ ДОЛЖЕН БЫТЬ ВАШ ОТВЕТ>

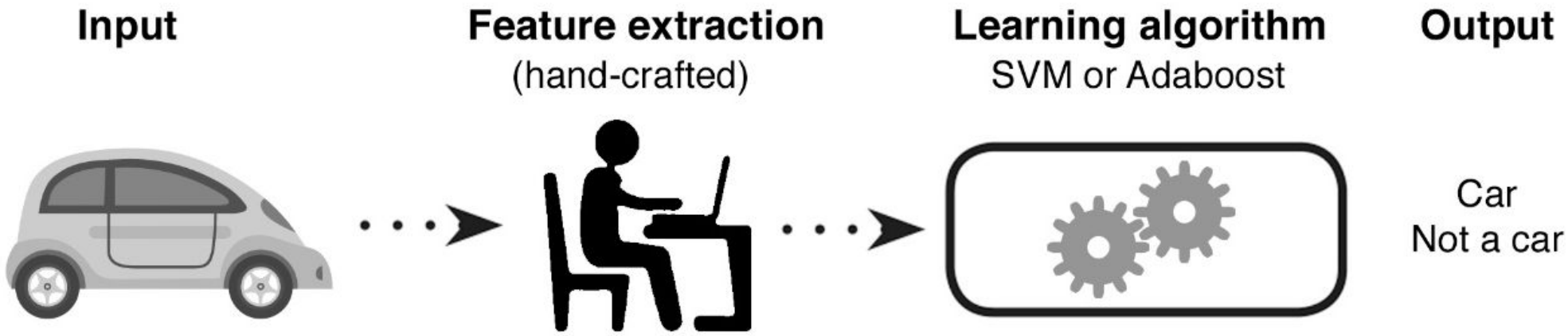
Нелинейности

- Ручные фичи в виде полиномов (а сколько фичей брать, какие полиномы делать?)
- Kernel trick (смотри лекцию про SVM)
- Другие алгоритмы (knn, деревья, наивный байес)
- Хочется иметь алгоритм, который настраивался бы точно таким же функционалам, что и линейные модели (правдоподобие, SGD), но чтобы он мог делать **нелинейные разделяющие поверхности**;

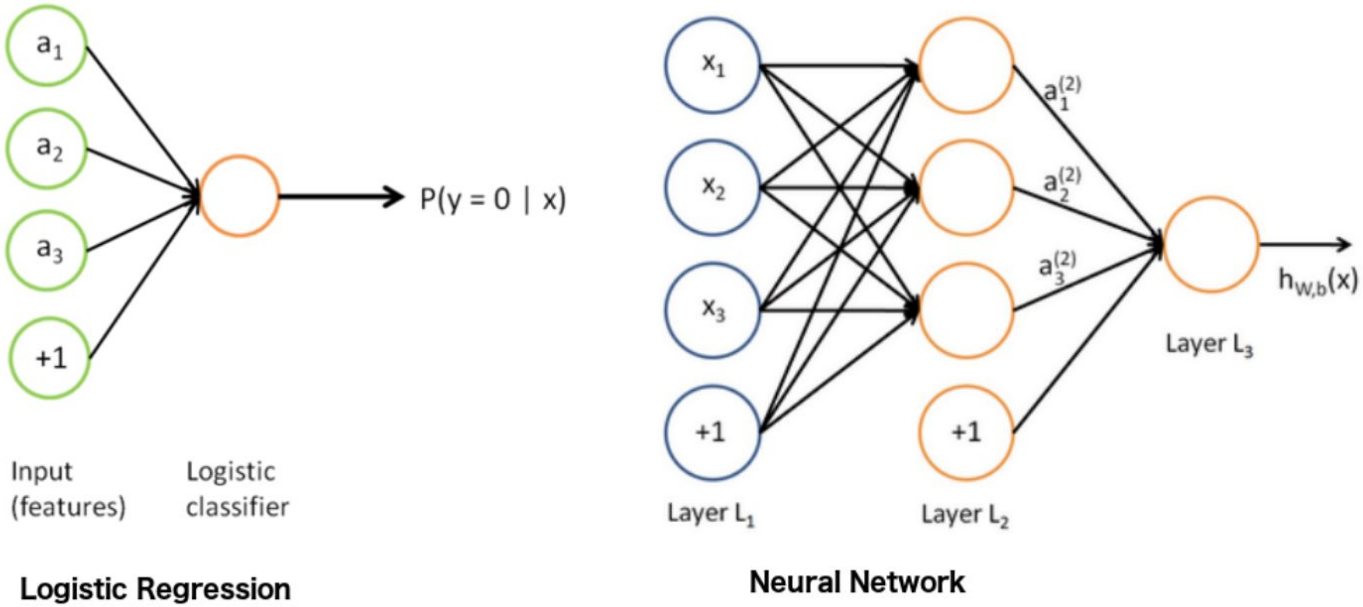


Проблемы текущего подхода

Хочется самый неприятную часть миновать, чтобы модель работала с сырыми признаками (unstructured data: картинки, тексты, аудио, временные ряды etc)



Автоматический фич инжениринг



Текущий подход:

а) делаем **хороший inductive bias** (NO FREE LUNCH)

б) майним огромный преогромный датасет

в) делаем огромное число параметров (в современных моделях миллиарды)

г) тратим 100 тысяч долларов на обучение на 1000 гпу на неделю

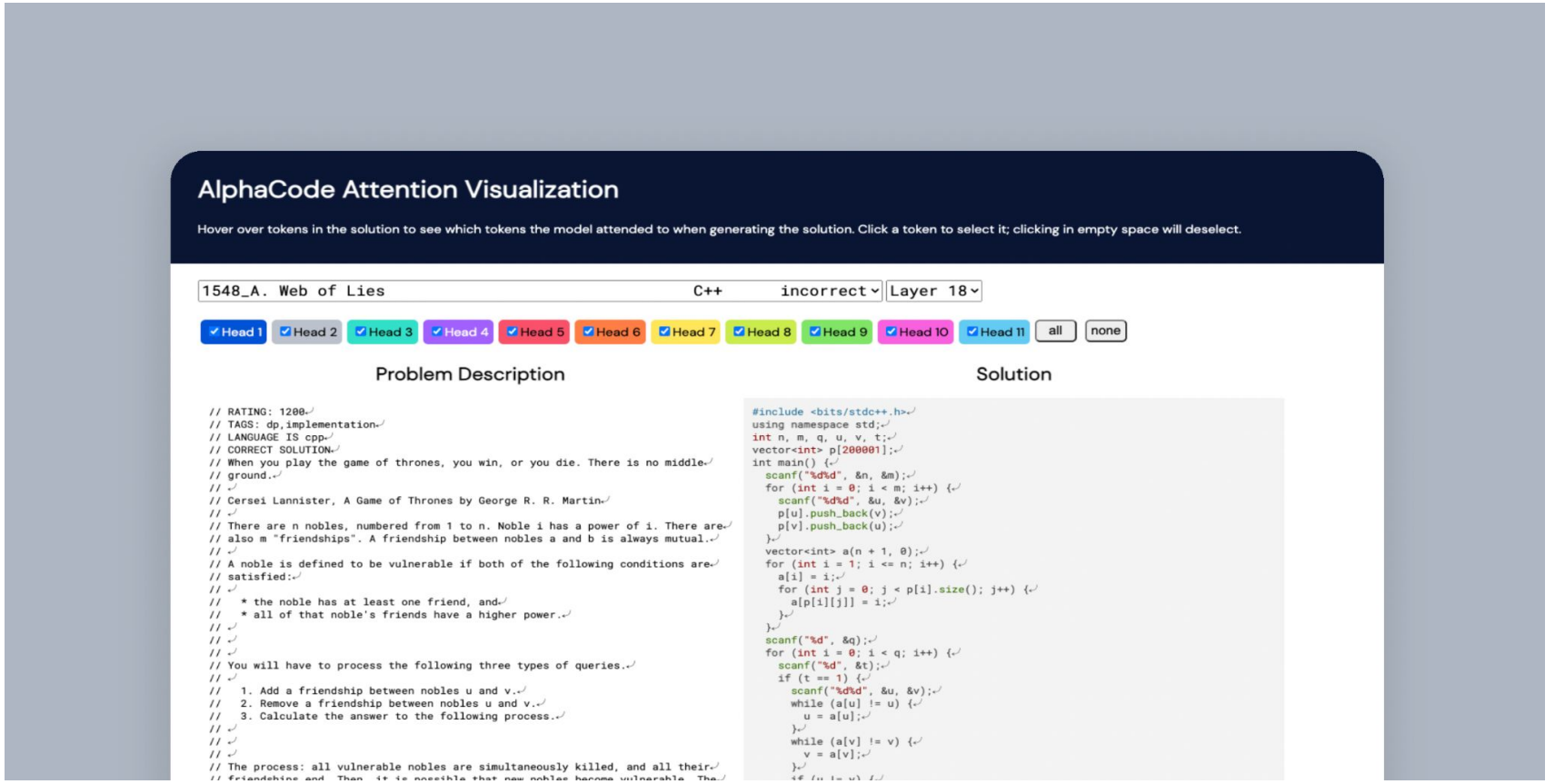
е) ….

ж) PROFIT

Примеры применения

Примеры применения

Примеры применения



Примеры применения

Итоговое резюме

Сегодня мы:

- Узнали структуру курса;
- Вспомнили, что такое inductive bias, функция ошибки и методы оптимизации;
- Обсудили проблему переобучения;
- Посмотрели на ML с вероятностной точки зрения;
- Поговорили о том, чем deep learning отличается от классического подхода;
- Просто замечательно провели время;

Вопросы?