АНАЛИТИКА БОЛЬШИХ ДАННЫХ, ОТ ПОИСКА МИНИМУМА ДО НЕЙРОННЫХ СЕТЕЙ

Ольга Развенская, инженер про разработке ПО, Intel

Немного о себе

- Магистр прикладной математики и информатики, университет Лобачевского, выпуск 2015 года;
- В Intel с 2013 года;
- Участвую в разработке Intel Data Analytics Acceleration Library (Intel DAAL).

План лекции

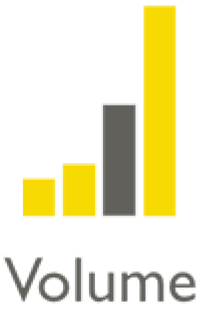
- Аналитика больших данных
  - Что такое большие данные
  - От 3V до 7V
- От поиска минимума до нейронных сетей
  - Задача поиска минимума
  - Стохастический градиентный спуск
  - Нейронные сети

АНАЛИТИКА БОЛЬШИХ ДАННЫХ

Что такое большие данные?

Большие данные — это совокупность технологий, которые призваны совершать три операции:

1. обрабатывать бóльшие по сравнению со «стандартными» сценариями объемы данных
2. уметь работать с быстро поступающими данными в очень больших объемах
3. уметь работать со структурированными и плохо структурированными данными параллельно в разных аспектах



pictref ref

Что такое большие данные?

- 3V = Volume + Velocity + Variety
- 4V = … + Value
- 5V = … + Veracity
- 7V = … + Variability + Visualization

picture ref

Примеры задач анализа больших данных

Характеристики задач

- Данные существенно многомерные (много переменных)
- Система слишком сложна, чтобы описать ее системой уравнений (машинное обучение)

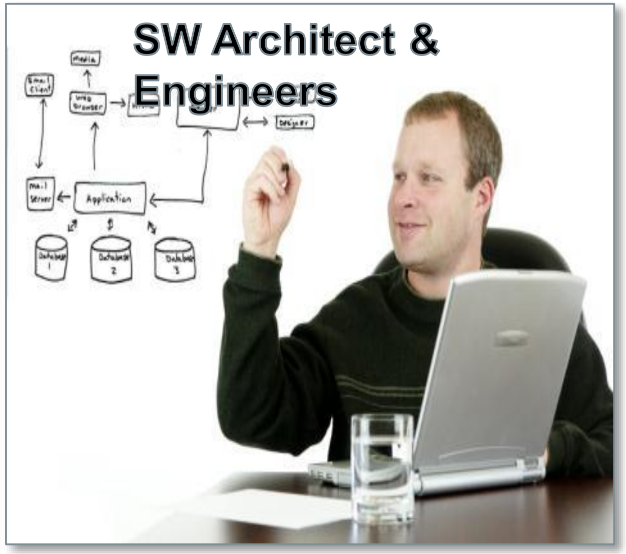
Примеры

- Распознавание «шаблонов» (геномика, распознавание образов и речи)
- Распознавание аномалий (кредитные рейтинги, аномальные транзакции)
- Моделирование финансовых инструментов, потребления энергии, молекулярная биология

Кто решает задачи анализа «больших данных»

Domain expert. Formulates problem statement. Works in conjunction with Data Scientist and SW Architect as a consultant on extracting business value from the domain-specific data

Expert in math and statistics. Works in conjunction with Domain Expert to create math model for domain specific problem. Works in conjunction with SW Architect on numerical method for knowledge discovery. Typically uses modeling/programming tools, e.g. R, Matlab, Python



Professional programmer (but not necessarily the mathematician). Creates complex SW for knowledge discovery. Usually leads the team of SW engineers. Skilled in development tools, databases, Big Data platforms, etc. May be skilled in general SW optimization techniques

Геофизика, гидро- и аэродинамика

Обработка сигналов

Финансовая аналитика

Инженерия

Создание цифрового контента

Наука и исследования

ОТ ПОИСКА МИНИМУМА ДО НЕЙРОННЫХ СЕТЕЙ

Минимум функции, функция «звездочка» (*)

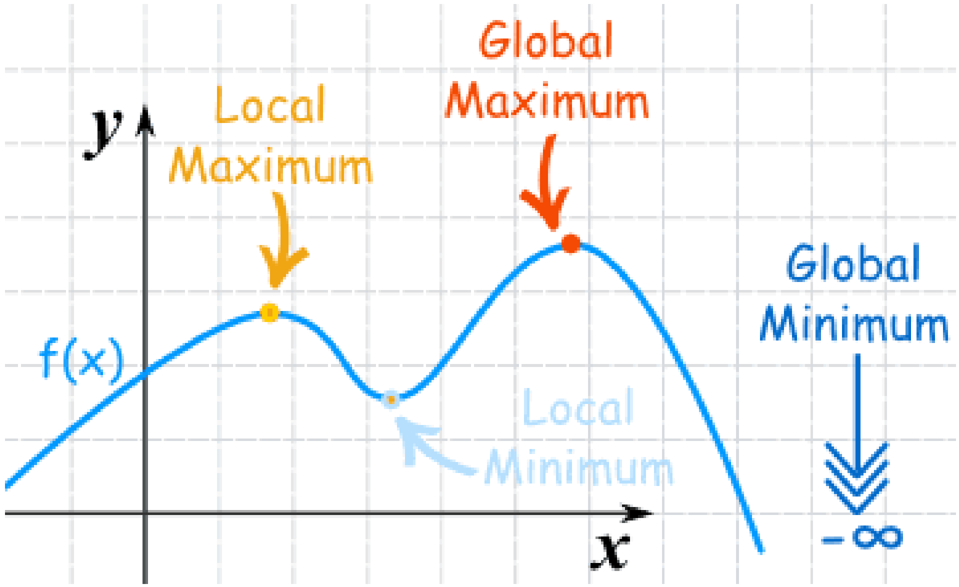
- Найти $w \in W$, при котором функция $F(w)$ имеет локальный минимум:
$$F(w) \rightarrow min$$

- Пусть функция задана в виде суммы:
$$F(w) = \frac{1}{n}\left(F_1(w) + \cdots + F_n(w)\right) = \frac{1}{n}\sum_{i=1}^{n} F_i(w)$$

- Пусть каждое слагаемое зависит от x_i
$$F(w) = \frac{1}{n}\sum_{i=1}^{n} F_i(w, x_i)$$

- Переименуем: $E(w) = F(w)$

- Пусть все слагаемы – одна и та же функция $F_i(w, x_i) = loss(w, x_i)$



$$\boxed{E(w) = \frac{1}{n}\sum_{i=1}^{n} loss(w, x_i) \quad (*)}$$

Picture

Методы поиска локального минимума

Некоторые примеры методов оптимизации:

- Методы основанные на свойствах минимизируемой функции (симплекс метод в линейном программировании)
- Итерационные методы
  - Метод половинного деления
  - Метод золотого сечения
  - …
- Итерационные методы для дифференцируемой функции
  - Градиентный спуск
  - …

Методы поиска локального минимума

Некоторые примеры методов оптимизации:

- Методы основанные на свойствах минимизируемой функции (симплекс метод в линейном программировании)
- Итерационные методы
  - Метод половинного деления
  - Метод золотого сечения
  - …
- Итерационные методы для дифференцируемой функции
  - **Градиентный спуск**
  - …

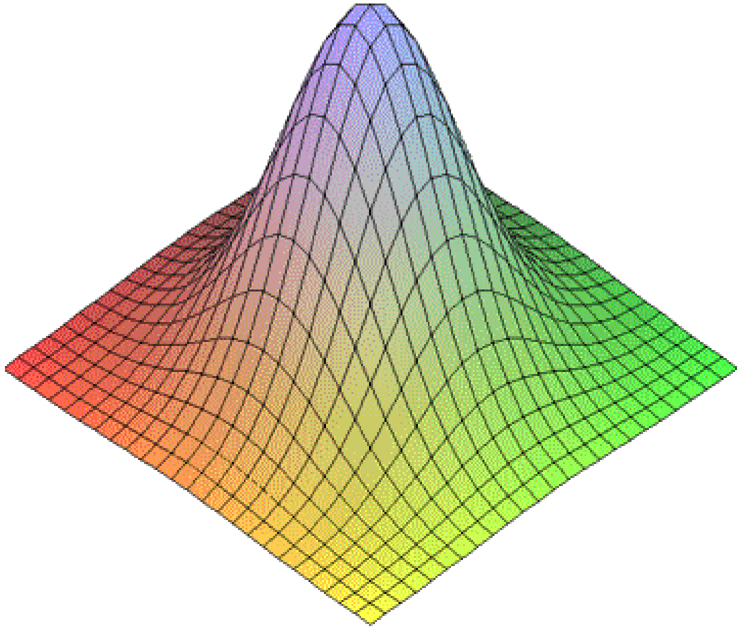
Что такое градиент?

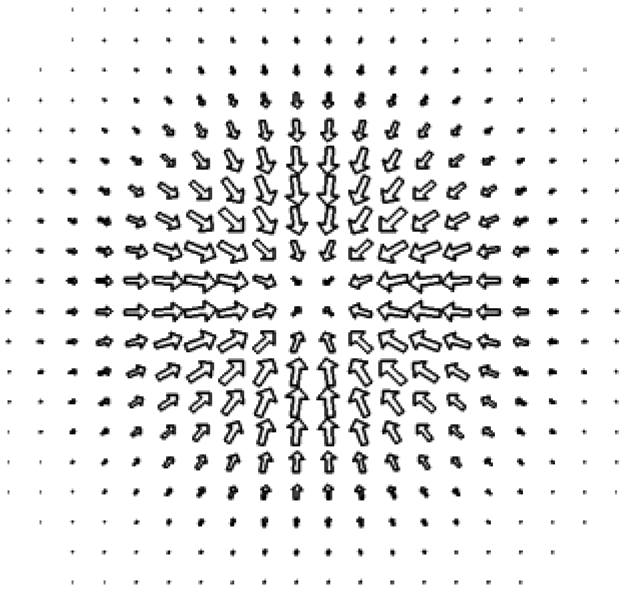
- Градиент - вектор, своим направлением указывающий направление наибольшего возрастания.
    - Градиент со знаком «минус» - направление наибольшего убывания
- Понятие градиента тесно связано с понятием частной производной

$$grad\ F(x_1, \dots, x_n) = \left\{\frac{\partial F}{\partial x_1}, \dots, \frac{\partial F}{\partial x_n}\right\}$$

- Может быть обозначен через оператор «набла»: ∇

$$\nabla F \equiv grad\ F$$





Метод градиентного спуска для нахождения локального минимума

Если градиент со знаком «минус» указывает на направление убывания функции, то можно выбрать некоторую стартовую точку, и двигаться от нее в направлении убывания функции. В конце концов придем в локальный минимум или «уйдем» на $-\infty$.

Запишем это математически:

$$F(w) \to min$$
$$w \coloneqq w - \alpha \cdot \nabla F(w)$$

Скорость спуска к минимуму определяется длиной вектора градиента и шагом α.

Градиентный спуск → Стохастический градиентный спуск

$$F(w) = \frac{1}{n}\sum_{i=1}^{n} F_i(w)$$

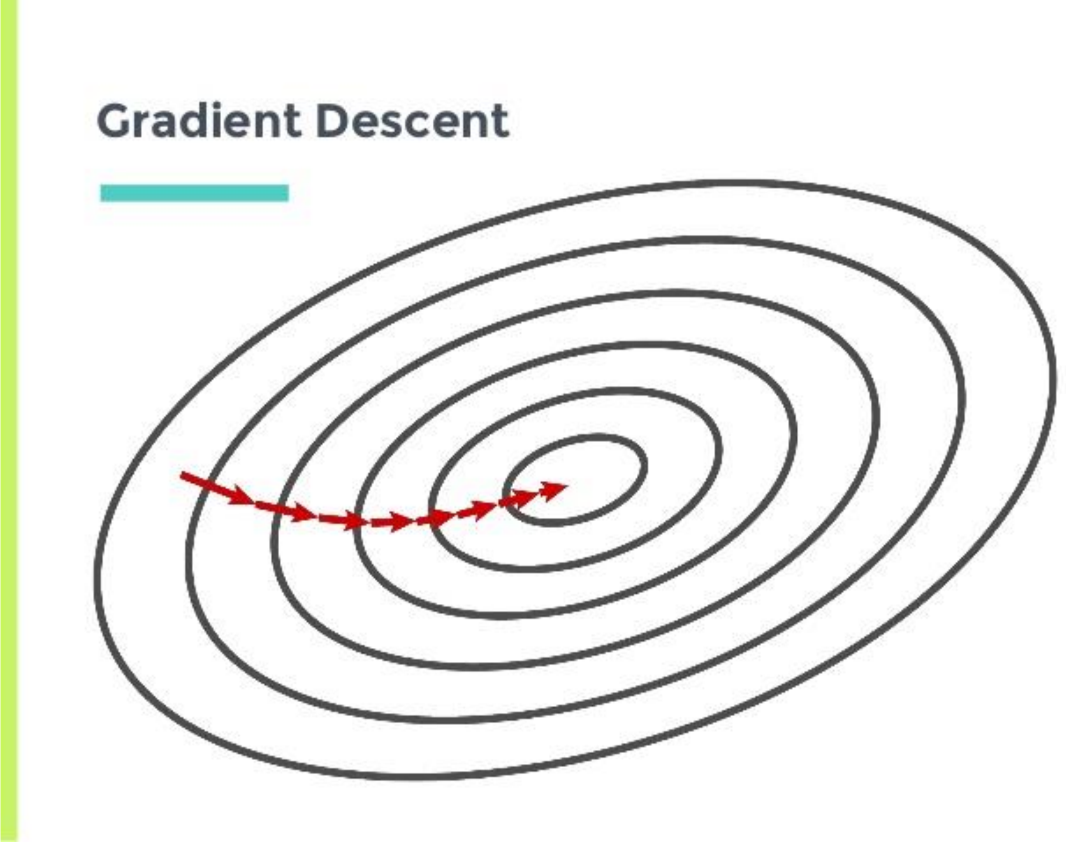
$$\nabla F(w) = \frac{1}{n}\cdot\sum_{i=1}^{n} \nabla F_i(w)$$

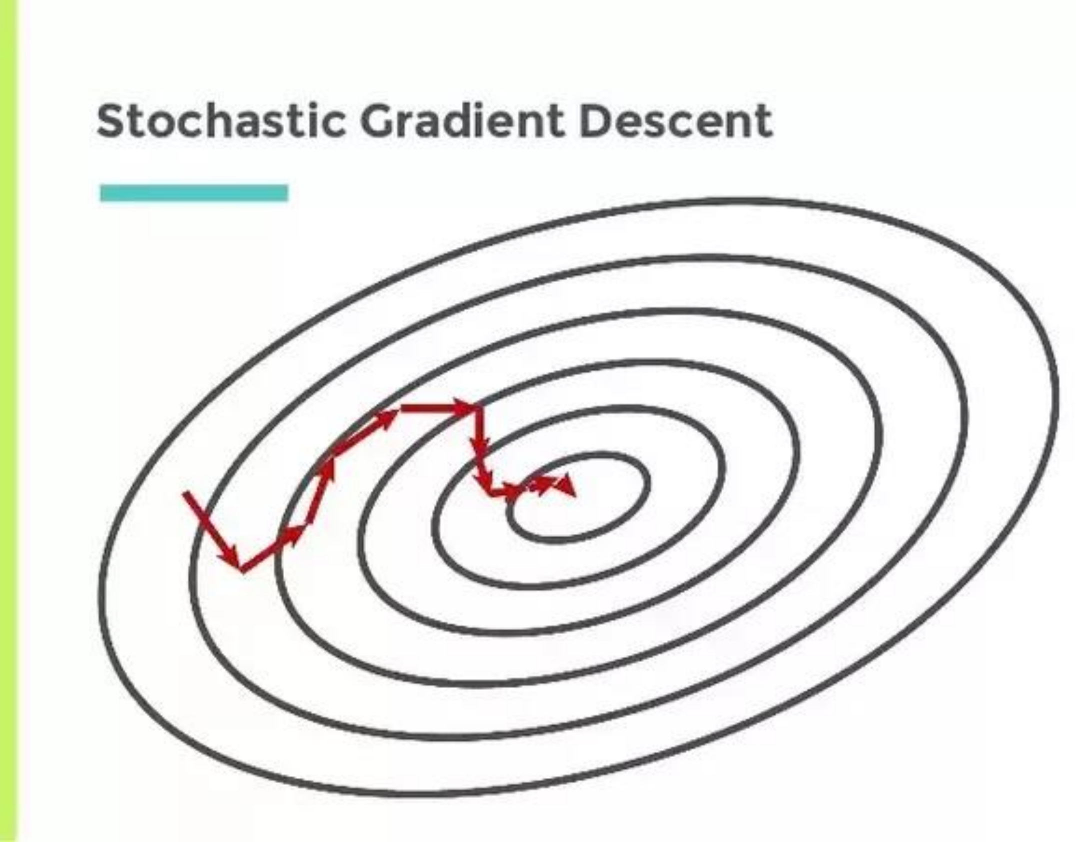
- Градиент аппроксимируется стохастическим градиентом

$$\nabla F(w) \approx \nabla F_i(w)$$

$$w := w - \alpha \nabla F_i(w)$$

Стохастический градиентный спуск





Picture #1, picture #2

Стохастический градиентный спуск для $(*)$

$$w \coloneqq w - \alpha \nabla F_i(w)$$

Для функции $(*)$ $E(w) = \frac{1}{n}\sum_{i=1}^{n} loss(w, x_i)$

$$\boxed{w := w - \alpha \cdot \nabla loss(w, x_i)}$$

Виды нейронных сетей

- Feedforward neural networks – сети прямого распространения
- Рекуррентные нейронные сети
- Сеть радиально-базисных функций
- Самоорганизующиеся карты
- …

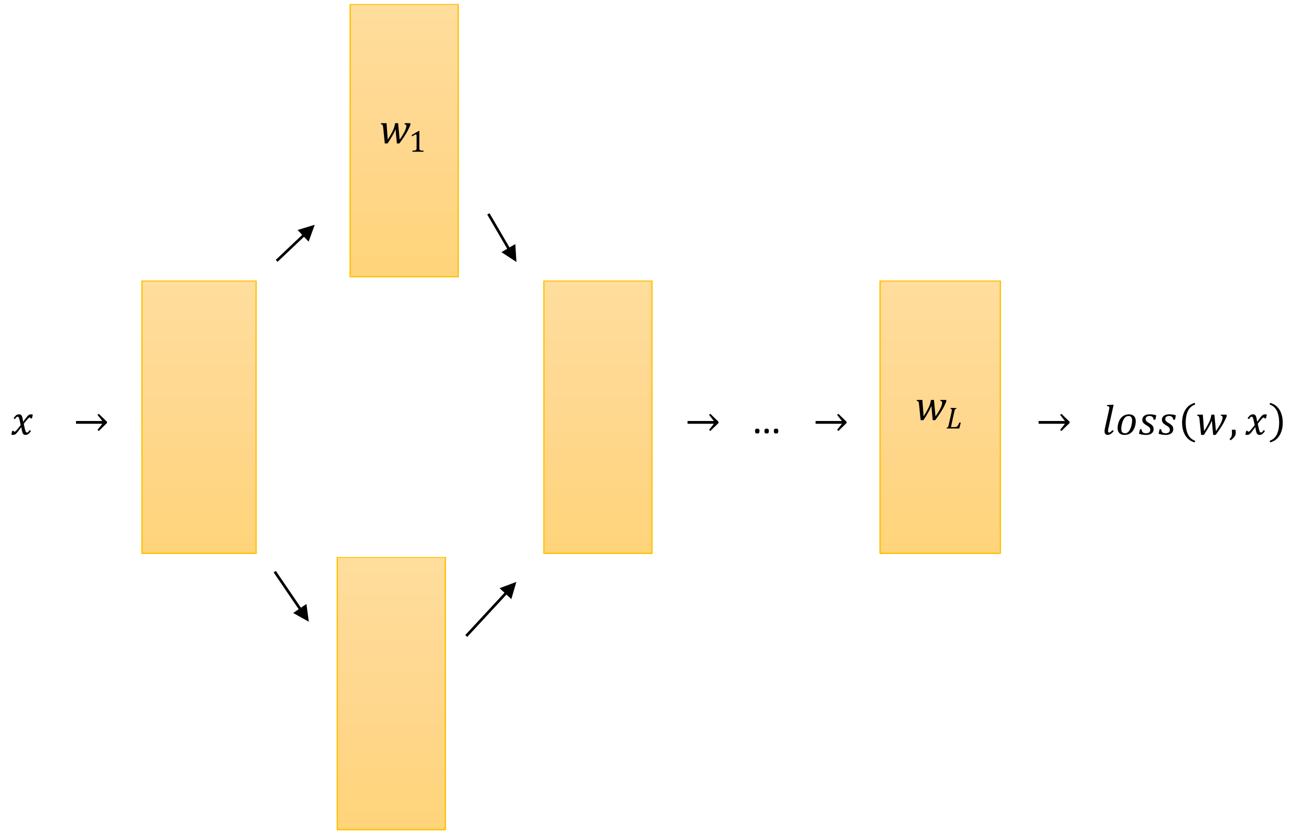
Виды нейронных сетей

- **Feedforward neural networks – сети прямого распространения**
- Рекуррентные нейронные сети
- Сеть радиально-базисных функций
- Самоорганизующиеся карты
- …

Далее под нейронной сетью понимаем сеть прямого распространения

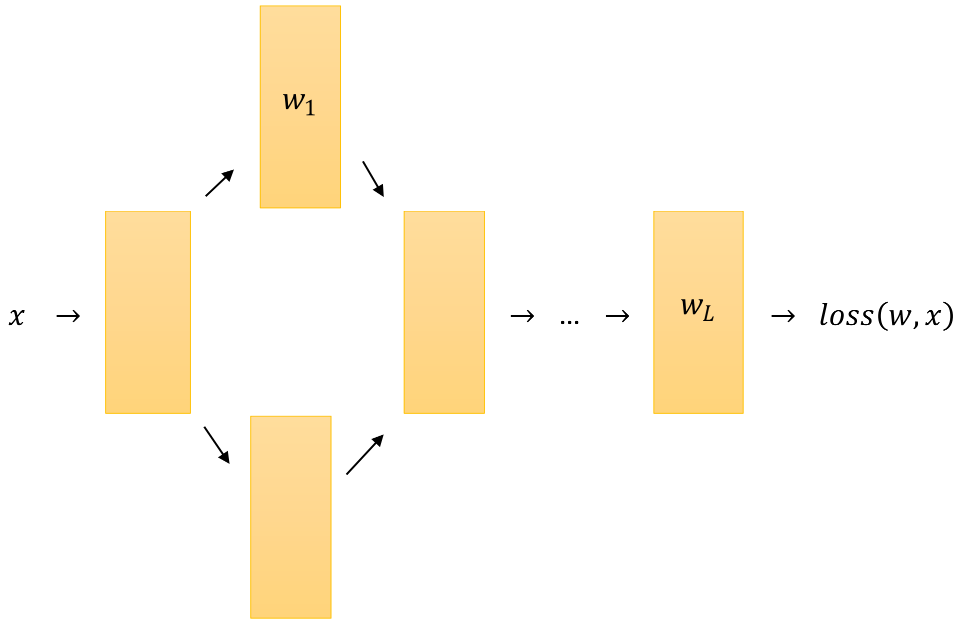
Из чего состоит нейронная сеть?

- Слои
  - С обучаемыми параметрами
  - Без обучаемых параметров
- Связи между слоями
- Функция ошибки
- Входные данные



Зачем нужна нейронная сеть?

- Минимизировать функцию ошибок на всем датасете;
- Функция ошибок – среднее значение ошибки на элементах датасета



$$E(w) = \frac{1}{n}\sum_{i=1}^{n} loss(w, x_i) \quad (*)$$

Нейронная сеть: детали

$$x_i \to \underset{[w_1]}{f_1} \to \underset{[w_2]}{f_2} \to \dots \to \underset{[w_{L-2}]}{f_{L-2}} \to \underset{[w_{L-1}]}{f_{L-1}} \to loss(w, x_i)$$

- Всего L слоев: $(L-1)$ на картинке + loss layer
- Все веса целевой функции $w = \{w_1, \dots, w_L\}$

Попробуем собрать все вместе

- Целевая функция $$\boxed{E(w) = \frac{1}{n}\sum_{i=1}^{n} loss(w, x_i) \ (*)}$$
- Веса
- Алгоритм поиска локального минимума $$\boxed{w := w - \alpha \cdot \nabla loss(w, x_i)}$$
  - Шаг спуска
  - Градиент

$$x_i \to f_1 \to f_2 \to \dots \to f_{L-2} \to f_{L-1} \to loss(w, x_i)$$

$$[w_1] \quad [w_2] \quad [w_{L-2}] \quad [w_{L-1}]$$

Попробуем собрать все вместе

- Целевая функция $\boxed{E(w) = \frac{1}{n}\sum_{i=1}^{n} loss(w, x_i) \ (*)}$
- Веса
- Алгоритм поиска локального минимума $\boxed{w := w - \alpha \cdot \nabla loss(w, x_i)}$
  - Шаг спуска
  - Градиент

$$x_i \to \underset{[w_1]}{\boxed{f_1}} \to \underset{[w_2]}{\boxed{f_2}} \to \dots \to \underset{[w_{L-2}]}{\boxed{f_{L-2}}} \to \underset{[w_{L-1}]}{\boxed{f_{L-1}}} \to loss(w, x_i)$$

Как получить градиент в нейронной сети

- Найти по определению, численно

$$x_i \rightarrow \boxed{f_1} \rightarrow \boxed{f_2} \rightarrow \ldots \rightarrow \boxed{f_{L-2}} \rightarrow \boxed{f_{L-1}} \rightarrow loss(w, x_i)$$

$$[w_1] \quad [w_2] \quad [w_{L-2}] \quad [w_{L-1}]$$

$$x_i \rightarrow \boxed{F} \rightarrow loss(w, x_i)$$

$$[w] = \{w_1, \ldots, w_n\}$$

$$F_i(w) = loss(w, x_i) =$$

$$= f_{L-1}\Big(w_{L-1}, f_{L-2}\Big(w_{L-2,} \ldots, f_2\big(w_2, f_1(w_1, x_1)\big)\Big)\Big)$$

$$\nabla F_i(w) = \left\{\frac{\partial F_i}{\partial w_1}, \ldots \frac{\partial F_i}{\partial w_n}\right\}$$

$$\frac{\partial F_i}{\partial w_1} = \lim_{\Delta w_1 \to 0} \frac{F_i(w_1 + \Delta w_1, w_2, \ldots, w_n) - F_i(w_1, w_2, \ldots, w_n)}{\Delta w_1}$$

Итого, получается, что для того, чтобы получить значение градиента, нужно прогнать сеть для $(n+1)$ значений аргумента:

- $F_i(w_1, w_2, \ldots, w_n)$
- $F_i(w_1 + \Delta w_1, w_2, \ldots, w_n)$
- $F_i(w_1, w_2 + \Delta w_2, \ldots, w_n)$
- $\ldots$
- $F_i(w_1, w_2, \ldots, w_n + \Delta w_n)$

Это слишком дорого.

Как получить градиент в нейронной сети

- Найти по определению, численно
- Формула обратного распространения ошибки (backpropagation)

Формула обратного распространения ошибки

- Сеть как композиция функций
- Разделяй и властвуй

Формула обратного распространения ошибки

- Сеть как композиция функций

$$x_i \to \underset{[w_1]}{f_1} \xrightarrow{y_1} \underset{[w_2]}{f_2} \xrightarrow{y_2} \dots \to \underset{[w_{L-2}]}{f_{L-2}} \xrightarrow{y_{L-2}} \underset{[w_{L-1}]}{f_{L-1}} \xrightarrow{y_{L-1}} loss(w, x_i)$$

$$F_i(w) = loss(w, x_i) =$$
$$= loss(y_{L-1}) = loss\big(f_{L-1}(w_{L-1}, y_{L-2})\big) =$$
$$= loss\Big(f_{L-1}\Big(w_{L-1}, f_{L-2}\Big(w_{L-2}, \dots, f_2\big(w_2, f_1(w_1, x_i)\big)\Big)\Big)\Big)$$

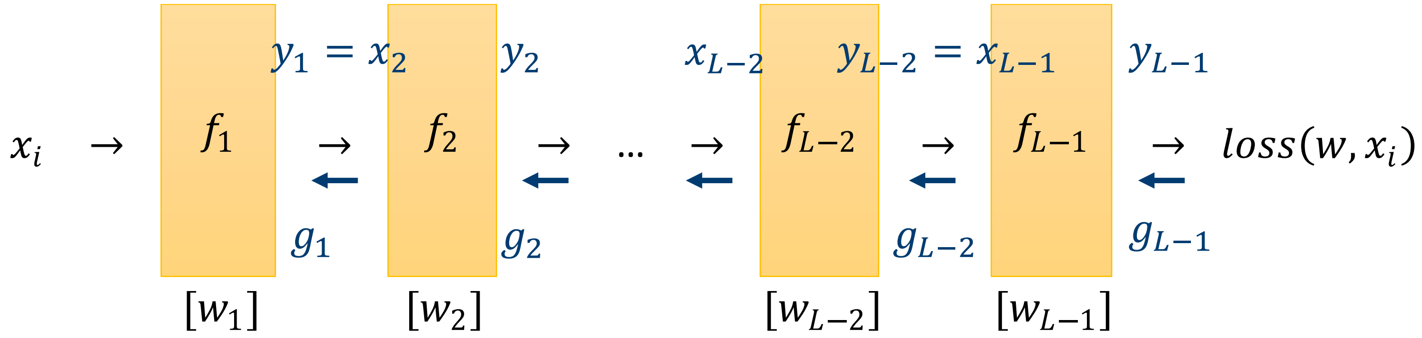
$$x_i \to \underset{[w] = \{w_1, \dots, w_n\}}{F} \to loss(w, x_i)$$

$$\nabla F_i(w) = \left\{\frac{\partial F_i}{\partial w_1}, \dots \frac{\partial F_i}{\partial w_n}\right\}$$

Формула обратного распространения ошибки

- Разделяй и властвуй

$$\frac{\partial loss}{\partial w_1} = ?, \ldots, \frac{\partial loss}{\partial w_{L-1}} = ?$$



$$\frac{\partial loss}{\partial w_{L-1}} = \sum \frac{\partial loss}{\partial y_{L-1}} \cdot \frac{\partial y_{L-1}}{\partial w_{L-1}} = \sum \frac{\partial y_{L-1}}{\partial w_{L-1}} \cdot g_{L-1}$$

$$\frac{\partial loss}{\partial w_{L-2}} = \sum \frac{\partial loss}{\partial y_{L-2}} \cdot \frac{\partial y_{L-2}}{\partial w_{L-2}} = \sum \frac{\partial y_{L-2}}{\partial w_{L-2}} \cdot \sum \frac{\partial loss}{\partial y_{L-1}} \cdot \frac{\partial y_{L-1}}{\partial x_{L-1}} =$$

$$= \sum \frac{\partial y_{L-2}}{\partial w_{L-2}} \cdot \sum \frac{\partial y_{L-1}}{\partial x_{L-1}} \cdot \frac{\partial loss}{\partial y_{L-1}} = \sum \frac{\partial y_{L-2}}{\partial w_{L-2}} \cdot \sum \frac{\partial y_{L-1}}{\partial x_{L-1}} \cdot g_{L-1} = \sum \frac{\partial y_{L-2}}{\partial w_{L-2}} \cdot g_{L-2}$$

…

$$\frac{\partial loss}{\partial w_1} = \cdots = \sum \frac{\partial y_1}{\partial w_1} \cdot g_1$$

- Все необходимые производные получены за один(!) проход в обратном направлении
- Англоязычное название алгоритма – backpropagation algorithm

Собираем все вместе

- Целевая функция

$$\boxed{E(w) = \frac{1}{n}\sum_{i=1}^{n} loss(w, x_i) \ \ (*)}$$

- Веса
- Алгоритм поиска локального минимума

$$\boxed{w := w - \alpha \cdot \nabla loss(w, x_i)}$$

  - Шаг спуска
  - Градиент

Backpropagation алгоритм

$$x_i \rightarrow \underset{[w_1]}{f_1} \rightarrow \underset{[w_2]}{f_2} \rightarrow \ldots \rightarrow \underset{[w_{L-2}]}{f_{L-2}} \rightarrow \underset{[w_{L-1}]}{f_{L-1}} \rightarrow loss(w, x_i)$$

Заключение

- Понятие больших данных
  - Большие данные – комплексное понятие, включающее в себя не только объем данных, но и их структуру, скорость поступления. Это также новый взгляд на решение проблем, без ответа на вопрос «Почему?», а с ответом на вопрос «Что?».
- Переход от минимума к нейронным сетям
  - Задачи, в которых применяются нейронные сети - это задачи на поиск локально минимума функции.