



ТЕХНОСФЕРА

Лекция n1 Основы нейронных сетей

Нестеров Павел

4 декабря 2014 г.

План лекции

Предпосылки

Краткая история теории нейронных сетей

Многослойная нейронная сеть прямого распространения

Алгоритм обратного распространения ошибки

Что дальше?

Принципиальное отличие

Теория статистического обучения

Линейная и логистическая регрессии, expectation-maximization, naive bayes classifier, random forest, support vector machine, gradient boosting trees и т.д.

Имитация работы мозга человека

Perceptron, cognitron, self-organizing maps, multi-layer feedforward network, convolution network, Boltzmann machine, deep neural network и т.д.

С другой стороны

Искусственная нейронная сеть

- ▶ алгоритмическая композиция (ансамбль) слабых моделей
- ▶ байесова или марковская сеть
- ▶ ???
- ▶ или любое другое *обобщение*, важны идеи лежащие в основе теории

Нервная система до нейробиологии



Рис.: 17 век, Рене Декарт о нервной системе: «Раздражение ступни передаётся по нервам в мозг, взаимодействует там с духом и таким образом порождает ощущение боли».

Нервная система в современном понимании

- ▶ В 1906 году врач и гистолог Сантьяго Рамон-и-Кахаль совместно с врачом Камилло Гольджи получают нобелевскую премию за "за работы по структуре нервной системы"; их работы заложили основы нейронной теории нервной системы и современной нейробиологии.

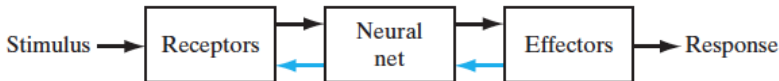
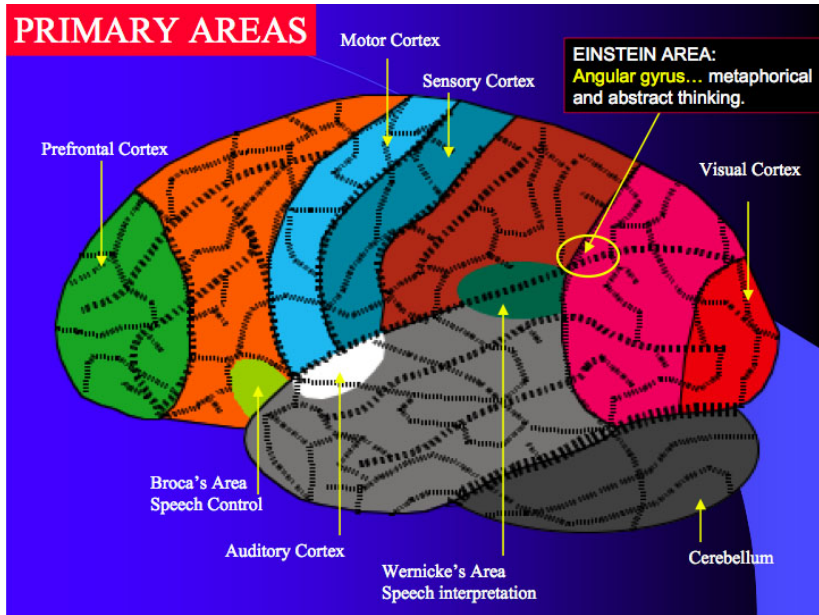


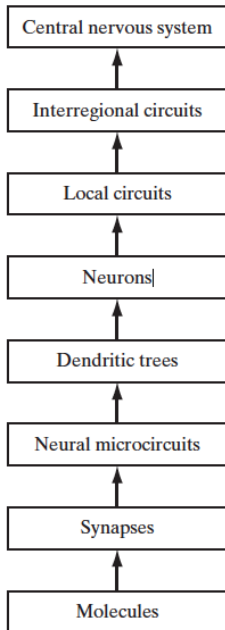
Рис.: Блок-схема нервной системы¹

¹Neural Networks and Learning Machines (3rd Edition), Simon O. Haykin

Мозг человека, #1

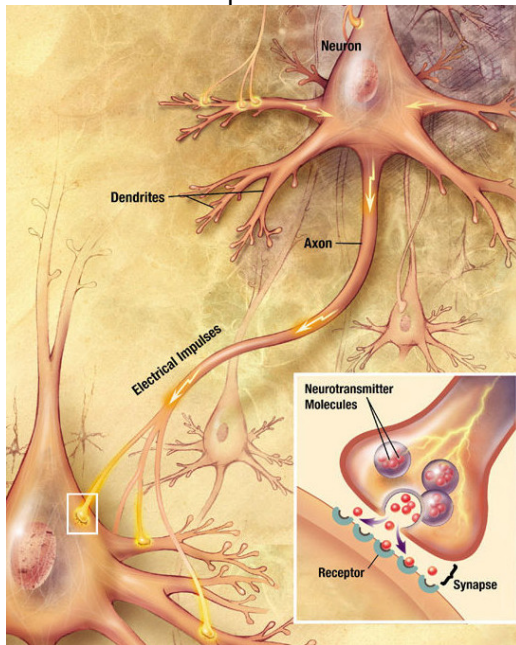


Мозг человека, #2



- ▶ $86 \cdot 10^9$ нейронов в мозге, из них $16.3 \cdot 10^9$ в коре
 - ▶ CPU: *15-core Xeon IvyBridge-EX* - $4.3 \cdot 10^9$ транзисторов
 - ▶ GPU: *Nvidia Tesla GK110 Kepler* - $7.08 \cdot 10^9$ транзисторов
- ▶ нейроны (и мозг в целом) обладают **нейропластичностью** - способностью изменяться под действием опыта;
- ▶ мозг - комплексная, нелинейная система параллельной обработки данных, способная изменять структуру своих составных частей;
- ▶ мозг решает определенный тип задач значительно быстрее чем любой современный компьютер, несмотря на то, что нейрон крайне *медленная* вычислительная единица.

Схема биологического нейрона



Нейронные ансамбли

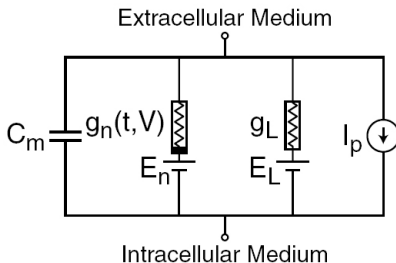
- ▶ Физиолог и нейропсихолог Дональд Хебб разработал теорию взаимосвязи головного мозга и мыслительных процессов в книге "The Organization of Behavior"(1949).
- ▶ Нейронный ансамбль - совокупность нейронов, составляющих функциональную группу в высших отделах мозга.
- ▶ Нейроансамбль - распределенный способ кодирования информации.
- ▶ Нейрон сам по себе генерирует по мимо сигнала еще и шум, но ансамбль в среднем генерирует чистый сигнал (*bagging?*).

Нейронная модель Ходжкина-Хаксли, #1

- ▶ Модель Ходжкина—Хаксли (1952 год) — математическая модель, описывающая генерацию и распространение потенциалов действия в нейронах².
- ▶ Потенциал действия — волна возбуждения, перемещающаяся по мембране живой клетки в процессе передачи нервного сигнала.
- ▶ Нобелевская премия по физиологии и медицине "за открытия, касающиеся ионных механизмов возбуждения и торможения в периферических и центральных участках нервных клеток"(1963 год).

²см. приложение Cellmembranion.gif

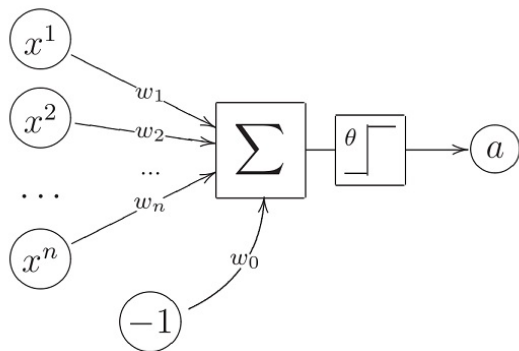
Нейронная модель Ходжкина-Хаксли, #2



Каждому элементу схемы соответствует свой биохимический аналог:

- ▶ C_m - емкость внутреннего липидного слоя клеточной мембраны;
- ▶ g_n - потенциал-зависимые ионные каналы отвечают за нелинейную электрическую проводимость;
- ▶ g_L - каналы мембранных пор отвечают за пассивную проводимость;
- ▶ E_x - источники напряжения побуждают ионы к движению через мембранные каналы.

Модель МакКаллока-Питтса (1943 год)



- ▶ $a(x) = \theta \left(\sum_{j=1}^n w_j \cdot x_j - w_0 \right);$
- ▶ $\theta(z) = [z \geq 0] = \begin{cases} 0, & z < 0 \\ 1, & z \geq 0 \end{cases}$ - функция Хевисайда;
- ▶ эквивалентно линейному классификатору.

Данная модель, с незначительными изменениями, актуальна и по сей день.

Правила Хебба (1949)

В своей книге Дональд Хебб описал процесс адаптирования нейронов в мозге в процессе обучения, и сформулировал базовые механизмы нейропластичности:

1. если два нейрона по разные стороны от синапсов активируются синхронно, то "вес" синапса слегка возрастает;
2. если два нейрона по разные стороны от синапсов активируются асинхронно, то "вес" синапса слегка ослабевает или синапс удаляется³.

Эти правила легли в основу зарождающейся теории нейросетей, сегодня в мы можем увидеть этот мета-алгоритм в основных методах обучения нейронных сетей.

³это расширенное правило, в оригинале второй части не было

Правила Хебба, математическая формулировка, #1

Допустим у нас имеется набор бинарных векторов размерности n , каждому из которых соответствует бинарный выход y :

- ▶ $X = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_m\}, \forall \vec{x}_i \in \{0, 1\}^n$
- ▶ $Y = \{y_1, y_2, \dots, y_m\}, \forall y_i \in \{0, 1\}$
- ▶ $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$

Тогда нейрон может совершить два типа ошибок:

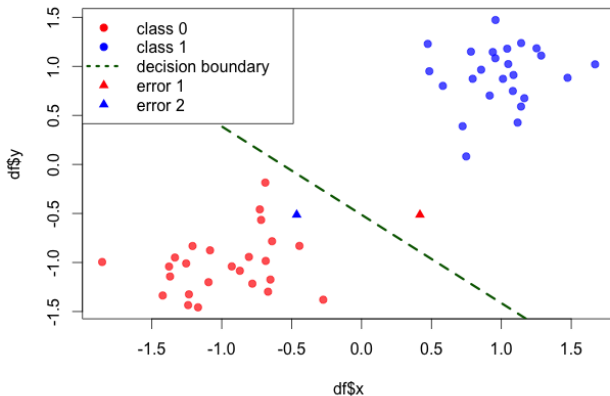
1. $\hat{y}_i = 0 \wedge y_i = 1 \Rightarrow$ *увеличить* веса при тех входах равных 1
 - ▶ *зачем?*
2. $\hat{y}_i = 1 \wedge y_i = 0 \Rightarrow$ *уменьшить* веса при тех входах равных 1
 - ▶ *зачем?*

Правила Хебба, математическая формулировка, #2

Тогда нейрон может совершить два типа ошибок:

1. $\hat{y}_i = 0 \wedge y_i = 1 \Rightarrow$ *увеличить* веса при тех входах равных 1
 - ▶ не преодолели порог \Rightarrow увеличить скалярное произведение
2. $\hat{y}_i = 1 \wedge y_i = 0 \Rightarrow$ *уменьшить* веса при тех входах равных 1
 - ▶ перешагнули порог \Rightarrow уменьшить скалярное произведение

Two types of error

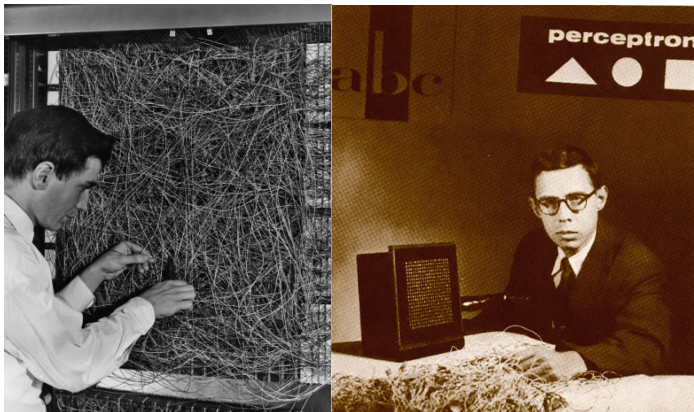


Однослойный персептрон Розенблатта (1958 год)

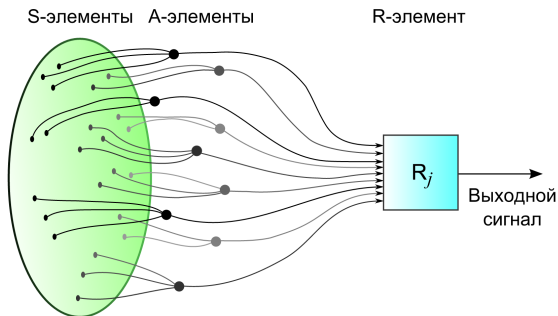
Нейрофизиолог Френк Розенблатт предложил схему устройства, моделирующего процесс человеческого восприятия, и назвал его "персептроном". Помимо этого:

- ▶ показал, что персептрон может выполнять базовые логические операции;
- ▶ разработал алгоритм обучения такой модели - метод коррекции ошибки;
- ▶ доказал сходимость алгоритма (теорема сходимости персептрона), но только для линейно разделимых классов;
- ▶ реализовал физический прототип такой модели;
- ▶ реализовал первый в мире нейрокомпьютер "MARK-1".

Нейрокомпьютер MARK-1



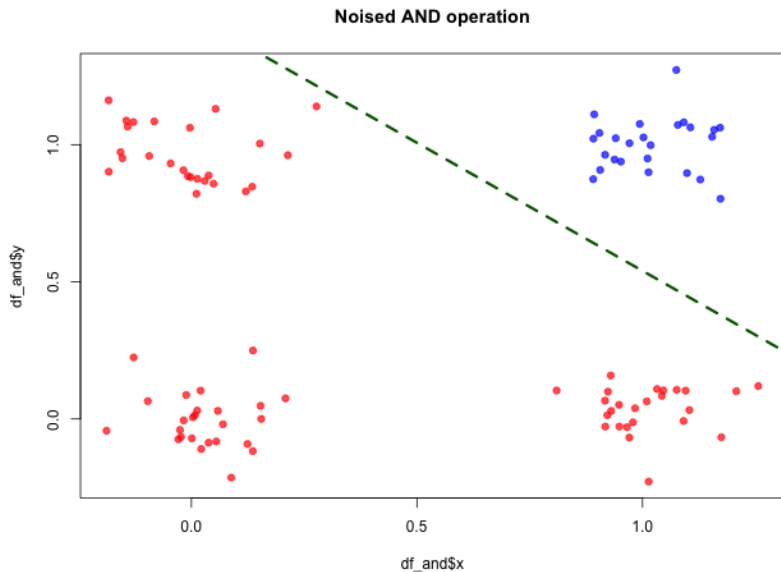
Элементарный персептрон



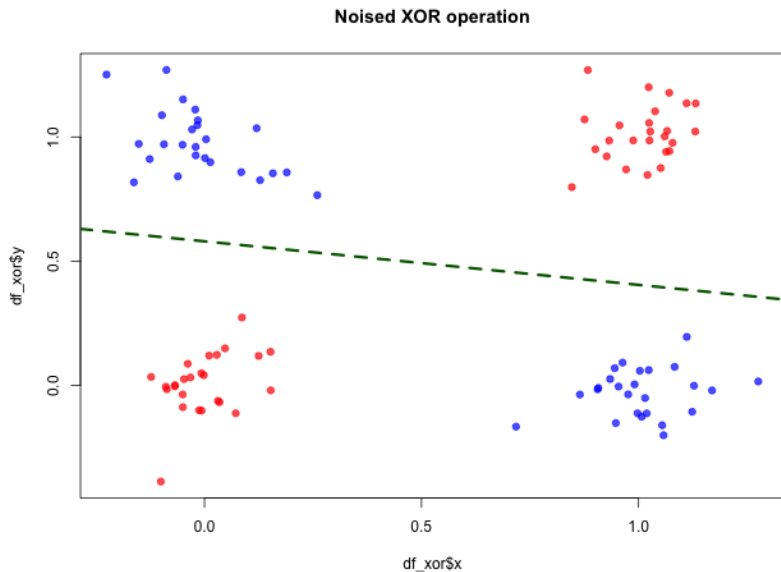
Метод коррекции ошибки

- ▶ $\hat{y}_i = 0 \wedge y_i = 1 \Rightarrow \Delta w = \eta(n) \cdot x(n)$
- ▶ $\hat{y}_i = 1 \wedge y_i = 0 \Rightarrow \Delta w = -\eta(n) \cdot x(n)$
 - ▶ $\eta(n)$ - скорость обучения, зависящая от итерации
 - ▶ $x(n)$ - входной образ на итерации n

Недостатки элементарного персептрона, AND



Недостатки элементарного перцептрона, XOR



Анимация сходимости для операций AND и XOR

- ▶ операция OR - 1layer-net-or.gif
- ▶ операция AND - 1layer-net-and1.gif
- ▶ операция XOR - 1layer-net-xor.gif ⁴

⁴<http://theclevermachine.wordpress.com/2014/09/11/a-gentle-introduction-to-artificial-neural-networks/>

Доказательства неэффективности нейронных сетей

- ▶ В 1969 году математик и исследователь ИИ Марвин Минский провел детальный математический анализ персептрона и опубликовал формальное доказательство ограниченности этой модели.
- ▶ *"**There is no reason to suppose** that any of these virtues carry over to the many-layered version. Nevertheless, we consider it to be an important research problem to elucidate (or reject) our intuitive judgement that the extension to multilayer systems is sterile."*⁵
- ▶ Отсутствие преимуществ + ограничения модели в итоге поубавили интерес научного сообщества к нейронным сетям, требовалось, что то принципиально новое

⁵Персептроны, Марвин Минский в соавторстве с Сеймуром Папертом, MIT Press, 1969

Период "забвения"

Исследования *искусственных* нейросетей не спеша продолжают, но в режиме поиска чего-то нового:

- ▶ 1972: Т. Кохонен разрабатывает новый тип нейросетей, способных функционировать в качестве памяти;
- ▶ 1975-1980: К. Фукусима разрабатывает когнитрон и неокогнитрон, совершенно новый тип многослойной сверточной сети, созданной по аналогии со строением зрительной системы;
- ▶ 1982: Дж. Хопфилд разрабатывает новый тип нейросети с обратными связями, выполняющей функции ассоциативной памяти;
- ▶ 1986: Дэвид Румельхарт, **Дж. Хинтон** и Рональд Вильямс разрабатывают *вычислительно эффективный* многослойных нейросетей - метод обратного распространения ошибки (именно работа этих авторов возрождает интерес к нейронным сетям в мире).

Теорема универсальной аппроксимации⁶

Введем следующие обозначения:

- ▶ $\phi(x)$ - не тождественная, ограниченная и монотонно возрастающая функция
- ▶ I_n - n -мерный единичный гиперкуб
- ▶ $C(I_n)$ - множество непрерывных функций на I_n

$\Rightarrow \forall f \wedge \forall \epsilon > 0 \exists$

- ▶ $m \in \mathbb{N}$
- ▶ $\{\beta_i\}_{i=1\dots m}, \forall \beta_i \in \mathbb{R}$
- ▶ $\{\alpha_i\}_{i=1\dots m}, \forall \alpha_i \in \mathbb{R}$
- ▶ $\{w_{ij}\}_{j=1\dots n, i=1\dots m}, \forall w_{ij} \in \mathbb{R}$

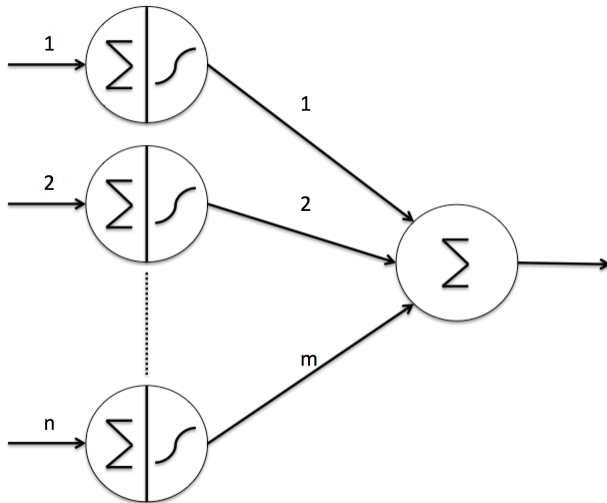
$\wedge \exists F(x_1, \dots, x_n) = \sum_{i=1}^m \alpha_i \phi\left(\sum_{j=1}^n w_{ij} \cdot x_j + \beta_i\right) :$

- ▶ $|F(x_1, \dots, x_n) - f(x_1, \dots, x_n)| < \epsilon$

Какая архитектура нейросети удовлетворяет такой формулировке?

⁶Neural Networks and Learning Machines (3rd Edition), Simon O. Haykin

Универсальный аппроксиматор



В чем проблема универсального аппроксиматора, исходя из условий теоремы?

Демонстрация сходимости нейросети с одним скрытым слоем

- ▶ операция XOR - 2layer-net-xor.gif
- ▶ бинарная классификация - 2layer-net-ring.gif
- ▶ аппроксимация функции \sin - 2layer-net-regression-sine.gif
- ▶ аппроксимация функции abs - 2layer-net-regression-abs.gif⁷

⁷<http://theclevermachine.wordpress.com/2014/09/11/a-gentle-introduction-to-artificial-neural-networks/>

Многослойная нейронная сеть прямого распространения

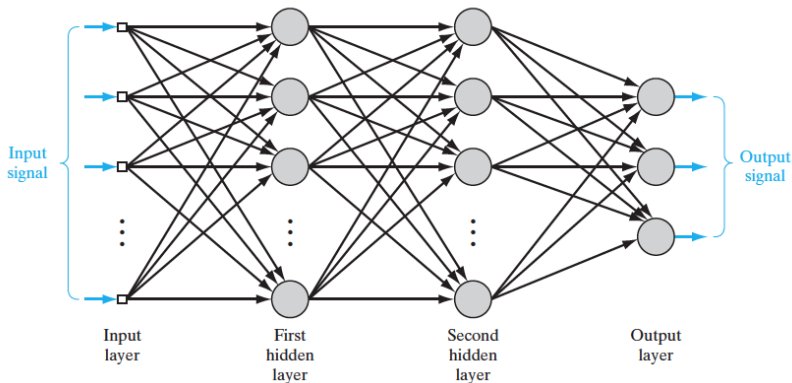


Рис.: Архитектура сети с двумя скрытыми слоями⁸

⁸Neural Networks and Learning Machines (3rd Edition), Simon O. Haykin

Отличие персептрона Румельхарта от персептрона Розенблатта

- ▶ Нелинейная функция активации;
- ▶ один и более скрытых слоев (до работ Хинтона по ограниченной машине Больцмана, на практике не использовали более двух скрытых слоев, а чаще всего один);
- ▶ сигналы на входе и на выходе не обязательно бинарные;
- ▶ произвольная архитектура сети (в рамках многослойности);
- ▶ ошибка сети интерпретирует в смысле некоторой меры, а не как число неправильных образов в режиме обучения.

Модифицированная модель нейрона МакКаллока-Питтса

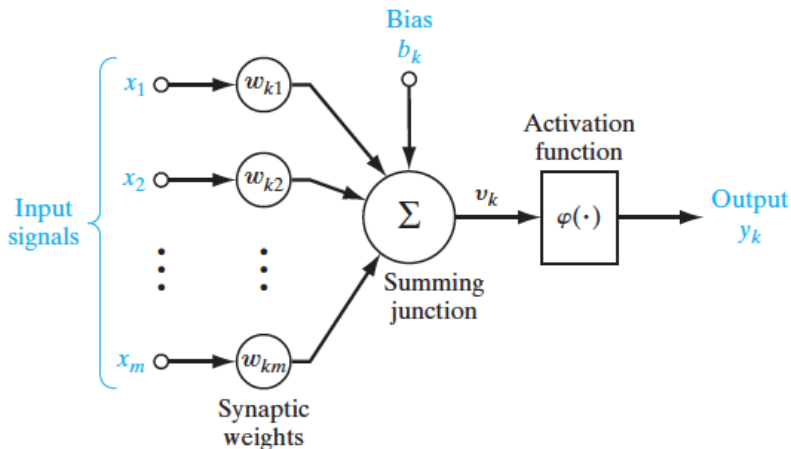


Рис.: Схема искусственного нейрона⁹

⁹Neural Networks and Learning Machines (3rd Edition), Simon O. Haykin

Функция активации

Задача функции активации - ограничить амплитуду выходного значения нейрона; чаще всего для этого используется одна из сигмоидальных (S-образных) функций:

- ▶ логистическая функция: $f(z) = \frac{1}{1 + e^{-a \cdot z}}, \forall a \in \mathbb{R}$
- ▶ гиперболический тангенс: $f(z) = \frac{e^{a \cdot x} - e^{-a \cdot x}}{e^{a \cdot x} + e^{-a \cdot x}}, \forall a \in \mathbb{R}$

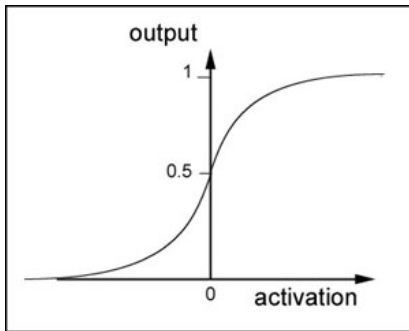
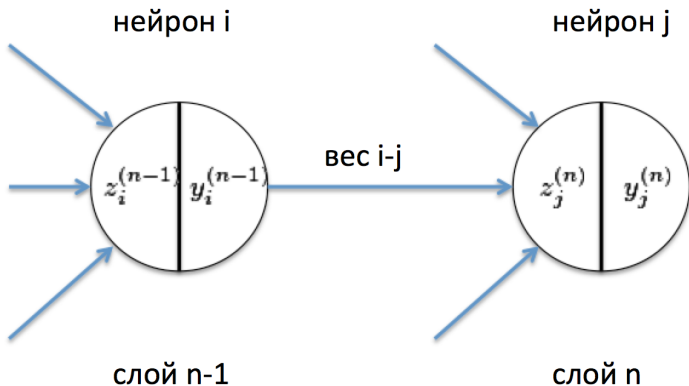


Рис.: Логистический сигмоид

Васкпроп, обозначения #1

$$z_j^{(n)} = b_j^{(n)} + \sum_{i=1}^{N_{n-1}} w_{ij}^{(n)} x_i^{(n)} = \sum_{i=0}^{N_{n-1}} w_{ij}^{(n)} x_i^{(n)} \quad (1)$$

$$y_k^{(n)} = f_k^{(n)} \left(z_k^{(n)} \right) \quad (2)$$



Backprop, обозначения #2

Обычное обучение с учителем:

- ▶ дан набор данных

$$D = \{(x_1, t_1), (x_2, t_2), \dots, (x_{|D|}, t_{|D|})\}, x_i \in \mathbb{R}^{N_{\text{INPUT}}}, y_i \in \mathbb{R}^{N_{\text{OUTPUT}}}$$

- ▶ необходимо построить такое отображение (нейросеть)

$f_{\text{NETWORK}} : X \rightarrow Y$, которое минимизирует некоторый функционал ошибки $E : \mathbb{R}^{N_{\text{OUTPUT}}} \times \mathbb{R}^{N_{\text{OUTPUT}}} \rightarrow \mathbb{R}$, например

- ▶ Евклидово расстояние для задачи регрессии
- ▶ логарифм функции правдоподобия распределения Бернулли для задачи классификации среди пересекающихся классов
- ▶ кросс-энтропия для задачи классификации среди непересекающихся классов

Градиентный спуск, #1

Алгоритм backprop - это модификация классического градиентного спуска. Параметрами модели являются только веса всех нейронов сети:

$$\delta_{ij}^{(n)} = -\eta \frac{\partial E(\vec{y}^{(n)}, \vec{t})}{\partial w_{ij}^{(n)}} \quad (3)$$

- ▶ η - скорость обучения (спуска, learning rate)
- ▶ $\vec{y}^{(n)}$ - вектор выходов нейросети (выходы последнего слоя)
- ▶ \vec{t} - ожидаемые выходы нейросети для текущего примера

Есть идеи?

Градиентный спуск, #2

- ▶ $\frac{\partial E}{\partial w_{ij}^{(n)}} = \frac{\partial E}{\partial z_j^{(n)}} \frac{\partial z_j^{(n)}}{\partial w_{ij}^{(n)}}$

- ▶ ???

Градиентный спуск, #3

$$\begin{aligned} \blacktriangleright \frac{\partial E}{\partial w_{ij}^{(n)}} &= \frac{\partial E}{\partial z_j^{(n)}} \frac{\partial z_j^{(n)}}{\partial w_{ij}^{(n)}} \\ \blacktriangleright \frac{\partial z_j^{(n)}}{\partial w_{ij}^{(n)}} &= \sum_i \frac{\partial w_{ij}^{(n)} x_i^{(n-1)}}{\partial w_{ij}^{(n)}} = x_i^{(n-1)} \end{aligned}$$

В итоге получим:

$$\frac{\partial E}{\partial w_{ij}^{(n)}} = x_i^{(n-1)} \frac{\partial E}{\partial z_j^{(n)}} \quad (4)$$

Градиентный спуск, выходной слой, #1

- ▶ $\frac{\partial E}{\partial w_{ij}^{(n)}} = x_i^{(n-1)} \frac{\partial E}{\partial z_j^{(n)}}$
- ▶ $E(y(z), t) ???$

Градиентный спуск, выходной слой, #2

$$\frac{\partial E}{\partial w_{ij}^{(n)}} = x_i^{(n-1)} \frac{\partial E}{\partial z_j^{(n)}} = x_i^{(n-1)} \frac{\partial E}{\partial y_j^{(n)}} \frac{\partial y_j^{(n)}}{\partial z_j^{(n)}} \quad (5)$$

Таким образом при условии дифференцируемости целевой функции и функции активации, вычисление градиента любого из весов выходного слоя становится легко решаемой задачей.

Градиентный спуск, любой скрытый слой, #1

$$\begin{aligned}\frac{\partial E}{\partial w_{ij}^{(n)}} &= x_i^{(n-1)} \frac{\partial E}{\partial z_j^{(n)}} \\ &= x_i^{(n-1)} \sum_k \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial z_k^{(n+1)}}{\partial z_j^{(n)}} \\ &= x_i^{(n-1)} \sum_k \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial z_k^{(n+1)}}{\partial y_j^n} \frac{\partial y_j^n}{\partial z_j^n}\end{aligned}$$

► $\frac{\partial z_k^{(n+1)}}{\partial y_j^n} ???$

Градиентный спуск, любой скрытый слой, #2

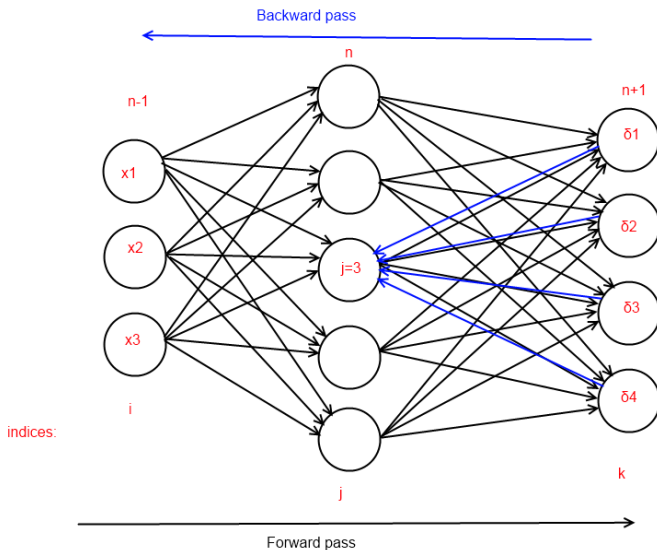


Рис.: Схема прямого (нелинейного) и обратного (линейного) распространения сигнала в сети

Градиентный спуск, любой скрытый слой, #3

$$\begin{aligned}\frac{\partial E}{\partial w_{ij}^{(n)}} &= x_i^{(n-1)} \frac{\partial E}{\partial z_j^{(n)}} \\&= x_i^{(n-1)} \sum_k \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial z_k^{(n+1)}}{\partial z_j^{(n)}} \\&= x_i^{(n-1)} \sum_k \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial z_k^{(n+1)}}{\partial y_j^n} \frac{\partial y_j^n}{\partial z_j^n}\end{aligned}$$

$$\blacktriangleright \frac{\partial z_k^{(n+1)}}{\partial y_j^n} = \sum_i \frac{\partial w^{(n+1)}_{ik} y_i^n}{\partial y_j^n} = w_{ik}^{(n+1)}$$

Градиентный спуск, любой скрытый слой, #4

$$\begin{aligned}\frac{\partial E}{\partial w_{ij}^{(n)}} &= x_i^{(n-1)} \frac{\partial E}{\partial z_j^{(n)}} \\&= x_i^{(n-1)} \sum_k \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial z_k^{(n+1)}}{\partial z_j^{(n)}} \\&= x_i^{(n-1)} \sum_k \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial z_k^{(n+1)}}{\partial y_j^n} \frac{\partial y_j^n}{z_j^n} \\&= x_i^{(n-1)} \sum_k w_{ik}^{(n+1)} \frac{\partial E}{\partial z_k^{(n+1)}} \frac{\partial y_j^n}{\partial z_j^n}\end{aligned}$$

Некоторые функции стоимости, #1

Среднеквадратичная ошибка:

- ▶ $E = \frac{1}{2} \sum_{i \in \text{OUTPUT}} (t_i - y_i)^2$
- ▶ $\frac{\partial E}{\partial y_i}$???

Логарифм правдоподобия Бернулли:

- ▶ $E = - \sum_{i \in \text{OUTPUT}} (t_i \log y_i + (1 - t_i) \log (1 - y_i))$
- ▶ $\frac{\partial E}{\partial y_i}$???

Некоторые функции стоимости, #2

Среднеквадратичная ошибка:

$$\blacktriangleright E = \frac{1}{2} \sum_{i \in \text{OUTPUT}} (t_i - y_i)^2$$

$$\blacktriangleright \frac{\partial E}{\partial y_i} = y_i - t_i$$

Логарифм правдоподобия Бернулли:

$$\blacktriangleright E = - \sum_{i \in \text{OUTPUT}} (t_i \log y_i + (1 - t_i) \log (1 - y_i))$$

$$\blacktriangleright \frac{\partial E}{\partial y_i} = \frac{t_i}{y_i} - \frac{1 - t_i}{1 - y_i}$$

Некоторые функции активации, #1

Логистическая функция:

- ▶ $f(z) = \frac{1}{1 + e^{-a \cdot z}}$

- ▶ $\frac{\partial f}{\partial z} ???$

Гиперболический тангенс:

- ▶ $f(z) = \frac{e^{a \cdot z} - e^{-a \cdot z}}{e^{a \cdot z} + e^{-a \cdot z}}$

- ▶ $\frac{\partial f}{\partial z} ???$

Некоторые функции активации, #2

Логистическая функция:

- ▶ $f(z) = \frac{1}{1 + e^{-a \cdot z}}$
- ▶ $\frac{\partial f}{\partial z} = a \cdot f(z) \cdot (1 - f(x))$

Гиперболический тангенс:

- ▶ $f(z) = \frac{e^{a \cdot z} - e^{-a \cdot z}}{e^{a \cdot z} + e^{-a \cdot z}}$
- ▶ $\frac{\partial f}{\partial z} = a \cdot (1 - f^2(z))$

Режимы обучения

- ▶ online learning
- ▶ batch learning
- ▶ full-batch learning

Регуляризация в нейронной сети, #1

Что это и зачем?

- ▶ $E_R = E(\vec{y}, \vec{t}) + R(W)$

Примеры L1 и L2 регуляризации:

- ▶ $R_{L1}(W) = \sum_{ijn} |w_{ij}^{(n)}|$

- ▶ $\frac{\partial R_{L1}(W)}{\partial w_{ij}^{(n)}} ???$

- ▶ $R_{L2}(W) = \frac{1}{2} \sum_{ijn} \left(w_{ij}^{(n)}\right)^2$

- ▶ $\frac{\partial R_{L2}(W)}{\partial w_{ij}^{(n)}} ???$

Регуляризация в нейронной сети, #2

- ▶ $E_R = E(\vec{y}, \vec{t}) + \lambda \cdot R(W)$

Примеры L1 и L2 регуляризации:

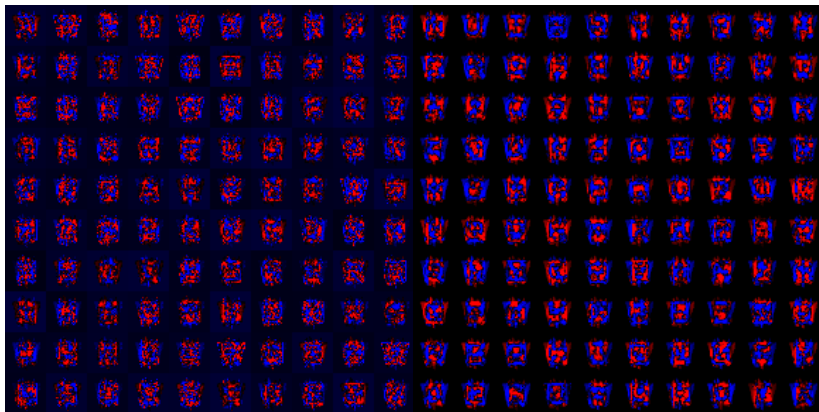
- ▶ $R_{L1}(W) = \sum_{ijn} |w_{ij}^{(n)}|$

- ▶ $\frac{\partial R_{L1}(W)}{\partial w_{ij}^{(n)}} = \text{SIGN}(w_{ij}^{(n)})$

- ▶ $R_{L2}(W) = \frac{1}{2} \sum_{ijn} (w_{ij}^{(n)})^2$

- ▶ $\frac{\partial R_{L2}(W)}{\partial w_{ij}^{(n)}} = w_{ij}^{(n)}$

Регуляризация в нейронной сети, #2

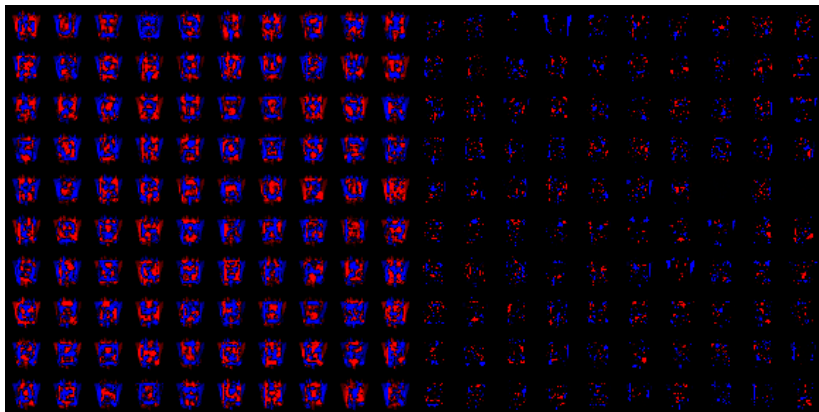


(a) RBM, no reg

(b) RBM, L2 reg

Рис.: Иллюстрация эффекта регуляризации

Регуляризация в нейронной сети, #3



(a) RBM, L2 reg

(b) RBM, L1 reg

Рис.: Иллюстрация эффекта регуляризации

Критерий остановки

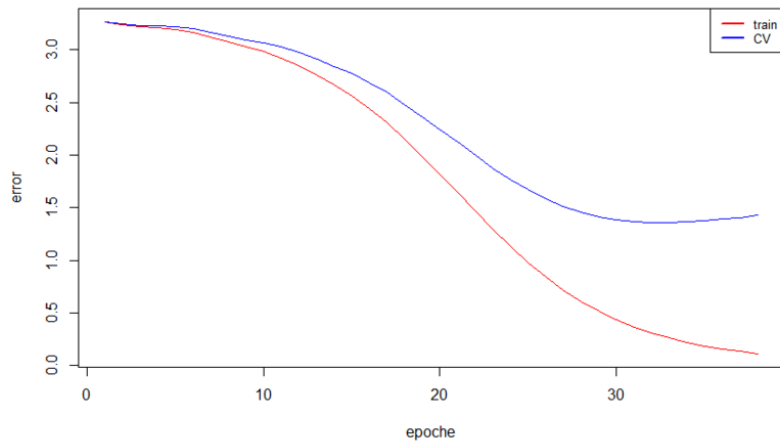


Рис.: Кроссвалидация

Ускорение сходимости

Добавление момента обучения:

$$\Delta w_{ij}(\tau) = \eta (\mu \Delta w_{ij}(\tau - 1) + \nabla w_{ij}) \quad (6)$$

Локальная скорость обучения:

$$\delta_{ij}^{(n)} = -\eta \cdot r_{ij}^{(n)} \cdot (\dots) \quad (7)$$

$$r_{ij}^{(n)} = \begin{cases} r_{ij}^{(n)} = b + r_{ij}^{(n)}, \nabla w_{ij}^{(n)}(\tau - 1) \cdot \nabla w_{ij}^{(n)}(\tau) > 0 \\ r_{ij}^{(n)} = p \cdot r_{ij}^{(n)} \end{cases} \quad (8)$$

где

- ▶ b - аддитивный бонус
- ▶ p - мультипликативный штраф
- ▶ $b + p = 1$
- ▶ есть смысл добавить верхнюю и нижнюю границы для значения $r_{ij}^{(n)}$

Планы

- ▶ softmax слой в сети прямого распространения
- ▶ обучение без учителя;
- ▶ стохастический нейрон и стохастическая нейросеть;
- ▶ ограниченная машина Больцмана.
- ▶ глубокие сети

Вопросы