Липецкий государственный технический университет

Кафедра прикладной математики

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ МАТЕМАТИЧЕСКИХ ИССЛЕДОВАНИЙ

Лекция 3

Искусственные нейронные сети

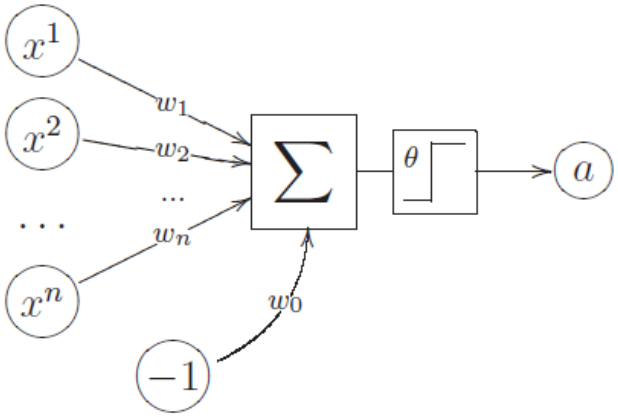
Составитель - Сысоев А.С., к.т.н., доцент

Липецк - 2021

Outline

3.1. Естественный нейрон и его формальная модель

- Пусть X — пространство объектов; Y — множество допустимых ответов; $y^*: X \to Y$ — целевая зависимость, известная только на объектах обучающей выборки $X^\ell = (x_i, y_i)$, $y_i = y^*(x^i)$. Требуется построить алгоритм $a: X \to Y$, аппроксимирующий целевую зависимость y^* на всём множестве X. Будем предполагать, что объекты описываются n числовыми признаками $f_j: X \to R$, $j = 1,\ldots,n$. Вектор $f_1(x),\ldots,f_n(x) \in R^n$ называется признаковым описанием объекта x
- В 1943 году МакКаллок и Питт



- признаки бинарные
- поступающие в нейрон импульсы складываются с весами $w_1,\ldots,w_n$
- Если суммарный импульс превышает заданный порог активации w_0, то нейрон возбуждается и выдаёт на выходе 1, иначе выдаётся 0

$$a(x) = \varphi\left(\sum_{j=1}^{n} w_j x^j - w_0\right),$$

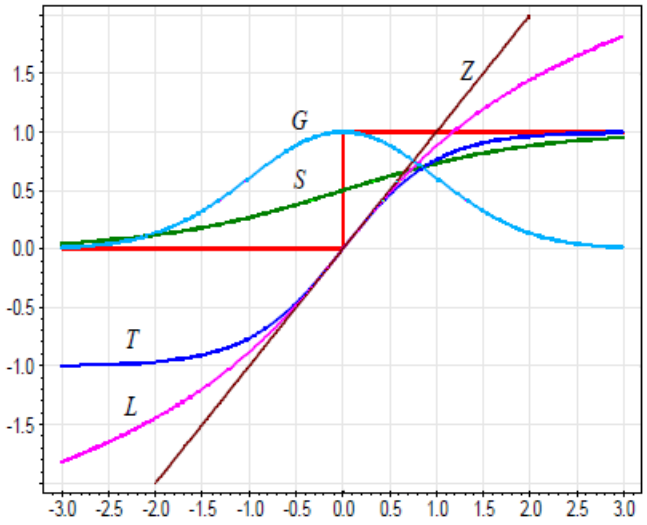
$$\varphi(z) = [z \geqslant 0]$$

3.1. Естественный нейрон и его формальная модель

- Введём дополнительный константный признак $x_0 \equiv -1$

$$a(x) = \varphi\left(\sum_{j=0}^{n} w_j x^j\right) = \varphi(\langle w, x\rangle).$$

- Модель МакКаллока–Питтса была обобщена на случай произвольных вещественных входов и выходов, и произвольных функций активации.



$\theta(z) = [z \geqslant 0]$	ступенчатая функция Хэвисайда;
$\sigma(z) = (1 + e^{-z})^{-1}$	сигмоидная функция (S);
$\mathrm{th}(z) = 2\sigma(2z) - 1$	гиперболический тангенс (T);
$\ln(z + \sqrt{z^2 + 1})$	логарифмическая функция (L);
$\exp(-z^2/2)$	гауссовская функция (G);
z	линейная функция (Z);

3.2. Методы обучения синаптических весов нейрона

- В 1957 году Розенблатт предложил эвристический алгоритм обучения нейрона, основанный на принципах, «подсмотренных» в нейрофизиологии. Экспериментально было обнаружено, что при синхронном возбуждении двух связанных нервных клеток синаптическая связь между ними усиливается.

Алгоритм 1.1. Обучение персептрона Розенблатта

Вход:
X^ℓ — обучающая выборка;
η — темп обучения;

Выход:
синаптические веса $w_0, w_1, \ldots, w_n$;

1: инициализировать веса w_j;
2: **повторять**
3: **для всех** $i = 1, \ldots, \ell$
4: $w := w - \eta\big(a(x_i) - y_i\big)x_i$;
5: **пока** веса w изменяются;

3.2. Методы обучения синаптических весов нейрона

- Правило Хэбба. Иногда удобнее полагать, что классы помечены числами −1 и 1, а нейрон выдаёт знак скалярного произведения:

$$a(x) = \text{sign}(\langle w, x \rangle).$$

- Тогда несовпадение знаков <*w*, x_i> и y_i означает, что нейрон ошибается на объекте x_i.
- Правило модификации весов:

$$\textbf{если } \langle w, x_i \rangle y_i < 0 \textbf{ то } w := w + \eta x_i y_i$$

- Исходя из принципа минимизации эмпирического риска задача настройки синаптических весов может быть сведена к поиску вектора w, доставляющего минимум функционалу качества:

$$Q(w) = \sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i), y_i\big) \to \min_w$$

$\mathscr{L}(a, y)$ - заданная функция потерь, характеризующая величину ошибки ответа *a* при правильном ответе *y*.

метод стохастического градиента

$$w := w - \eta \frac{\partial Q}{\partial w},$$

$$w := w - \eta \sum_{i=1}^{\ell} \mathscr{L}'_a\big(a(x_i), y_i\big) \varphi'(\langle w, x_i \rangle) x_i.$$

3.2. Методы обучения синаптических весов нейрона

Алгоритм 1.2. Обучение персептрона методом стохастического градиента.

Вход:

X^ℓ — обучающая выборка;

η — темп обучения;

Выход:

Синаптические веса $w_0, w_1, \ldots, w_n$;

1: инициализировать веса:
$w_j := \text{random}\left(-\frac{1}{2n}, \frac{1}{2n}\right)$;

2: инициализировать текущую оценку функционала:
$Q := \sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i), y_i\big)$;

3: **повторять**

4: выбрать объект x_i из X^ℓ случайным образом;

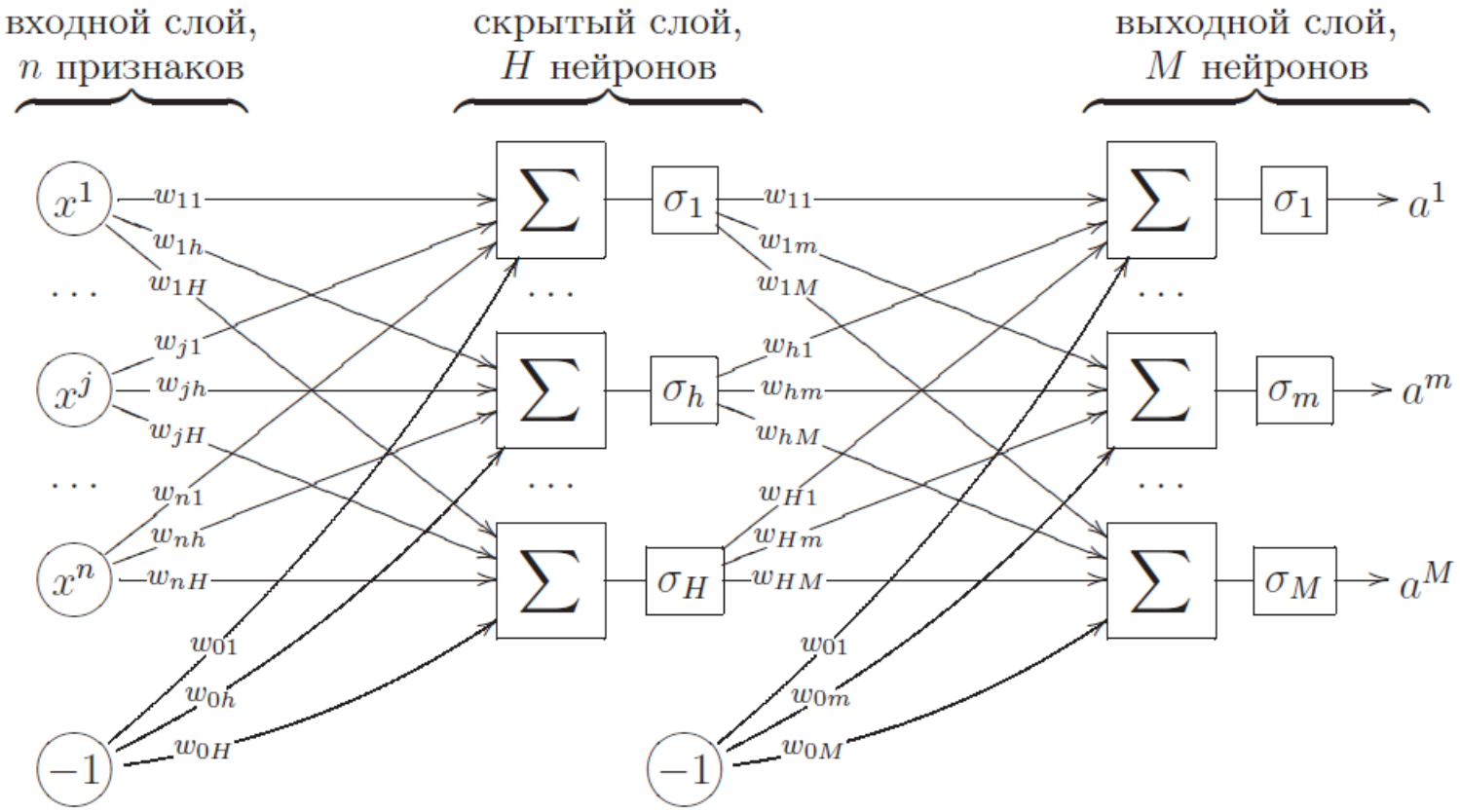
5: вычислить выходное значение алгоритма $a(x_i)$ и ошибку:
$\varepsilon_i := \mathscr{L}\big(a(x_i), y_i\big)$;

6: сделать шаг градиентного спуска:
$w := w - \eta \mathscr{L}'_a\big(a(x_i), y_i\big)\varphi'(\langle w, x_i\rangle)x_i$;

7: оценить значение функционала:
$Q := \frac{\ell-1}{\ell} Q + \frac{1}{\ell}\varepsilon_i^2$;

8: **пока** значение Q не стабилизируется;

3.3. Многослойные нейронные сети



3.3. Многослойные нейронные сети

- Любая булева функция представима в виде двухслойной сети. Это тривиальное следствие нейронной представимости функций И, ИЛИ, НЕ и представимости произвольной булевой функции в виде дизъюнктивной нормальной формы
- Из простых геометрических соображений вытекает, что двухслойная сеть с пороговыми функциями активации позволяет выделить произвольный выпуклый многогранник в *n*-мерном пространстве признаков. Трёхслойная сеть позволяет вычислить любую конечную линейную комбинацию характеристических функций выпуклых многогранников, следовательно, аппроксимировать любые области с непрерывной границей, включая неодносвязные, а также аппроксимировать любые непрерывные функции.
- В 1900 году Гильберт предложил список из 23 нерешённых задач, которые, по его мнению, должны были стать вызовом для математиков XX века. Тринадцатая проблема заключалась в следующем: возможно ли произвольную непрерывную функцию *n* аргументов представить в виде суперпозиции функций меньшего числа аргументов. Ответ был дан А.Н. Колмогоровым

3.3. Многослойные нейронные сети

Теорема (Колмогоров, 1957). *Любая непрерывная функция n аргументов на единичном кубе $[0,1]^n$ представима в виде суперпозиции непрерывных функций одного аргумента и операции сложения:*

$$f(x^1, x^2, \ldots, x^n) = \sum_{k=1}^{2n+1} h_k \left(\sum_{i=1}^{n} \varphi_{ik}(x^i) \right),$$

где h_k, φ_{ik} — непрерывные функции, причём φ_{ik} не зависят от выбора f.

- Известна классическая теорема Вейерштрасса о том, что любую непрерывную функцию *n* переменных можно равномерно приблизить полиномом с любой степенью точности. Более общая теорема Стоуна утверждает, что любую непрерывную функцию на произвольном компакте *X* можно приблизить не только многочленом от исходных переменных, но и многочленом от любого конечного набора функций *F*, разделяющих точки

Опр. *Набор функций F называется разделяющим точки множества X, если для любых различных $x, x' \in X$ существует функция $f \in F$ такая, что $f(x) \neq f(x')$.*

Теорема (Стоун, 1948). *Пусть X — компактное пространство, $C(X)$ — алгебра непрерывных на X вещественных функций, F — кольцо в $C(X)$, содержащее константу $(1 \in F)$ и разделяющее точки множества X. Тогда F плотно в $C(X)$.*

3.3. Многослойные нейронные сети

Опр. *Набор функций $F \subseteq C(X)$ называется замкнутым относительно функции $\varphi : \mathbb{R} \to \mathbb{R}$, если для любого $f \in F$ выполнено $\varphi(f) \in F$.*

Теорема (Горбань, 1998). *Пусть X — компактное пространство, $C(X)$ — алгебра непрерывных на X вещественных функций, F — линейное подпространство в $C(X)$, замкнутое относительно нелинейной непрерывной функции φ, содержащее константу $(1 \in F)$ и разделяющее точки множества X. Тогда F плотно в $C(X)$.*

- Это интерпретируется как утверждение об универсальных аппроксимационных возможностях произвольной нелинейности: с помощью линейных операций и единственного нелинейного элемента ϕ можно получить устройство, вычисляющее любую непрерывную функцию с любой желаемой точностью. Однако данная теорема ничего не говорит о количестве слоёв нейронной сети (уровней вложенности суперпозиции) и о количестве нейронов, необходимых для аппроксимации произвольной функции.

3.4. Многослойные нейронные сети

АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

Рассмотрим многослойную сеть, в который каждый нейрон предыдущего слоя связан со всеми нейронами последующего слоя. Для большей общности положим $X = \mathbb{R}^n$, $Y = \mathbb{R}^M$.

Введём следующие обозначения. Пусть выходной слой состоит из M нейронов с функциями активации σ_m и выходами a^m, $m = 1, \ldots, M$. Перед ним находится скрытый слой из H нейронов с функциями активации σ_h и выходами u^h, $h = 1, \ldots, H$.

Веса синаптических связей между h-м нейроном скрытого слоя и m-м нейроном выходного слоя будем обозначать через w_{hm}. Перед этим слоем может находиться либо распределительный слой, либо ещё один скрытый слой с выходами v^j, $j = 1, \ldots, J$ и синаптическими весами w_{jh}. В общем случае число слоёв может быть произвольным. Если сеть двухслойная, то v^j есть просто j-й признак: $v^j(x) \equiv f_j(x) \equiv x^j$, и $J = n$. Обозначим через w вектор всех синаптических весов сети.

Выходные значения сети на объекте x_i вычисляются как суперпозиция:

$$a^m(x_i) = \sigma_m\left(\sum_{h=0}^{H} w_{hm} u^h(x_i)\right); \qquad u^h(x_i) = \sigma_h\left(\sum_{j=0}^{J} w_{jh} v^j(x_i)\right).$$

3.4. Многослойные нейронные сети

АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

Запишем функционал среднеквадратичной ошибки для отдельного объекта x_i:

$$Q(w) = \frac{1}{2}\sum_{m=1}^{M}\left(a^m(x_i) - y_i^m\right)^2.$$

В дальнейшем нам понадобятся частные производные Q по выходам нейронов. Выпишем их сначала для выходного слоя:

$$\frac{\partial Q(w)}{\partial a^m} = a^m(x_i) - y_i^m = \varepsilon_i^m.$$

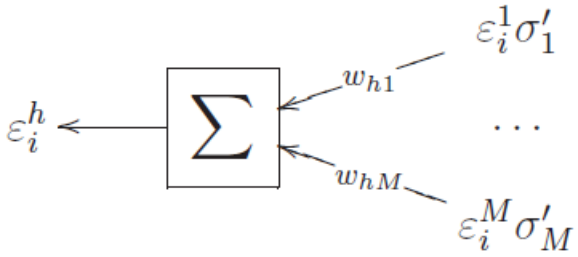
Оказывается, частная производная Q по a^m равна величине ошибки ε_i^m на объекте x_i. Теперь выпишем частные производные по выходам скрытого слоя:

$$\frac{\partial Q(w)}{\partial u^h} = \sum_{m=1}^{M}\left(a^m(x_i) - y_i^m\right)\sigma'_m w_{hm} = \sum_{m=1}^{M}\varepsilon_i^m\sigma'_m w_{hm} = \varepsilon_i^h.$$

3.4. Многослойные нейронные сети

АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

Заметим, что ε_i^h вычисляется по ε_i^m, если запустить сеть «задом наперёд», подав на выходы нейронов скрытого слоя значения $\varepsilon_i^m \sigma'_m$, а результат ε^h получив на входе. При этом входной вектор скалярно умножается на вектор весов w_{hm}, находящихся справа от нейрона, а не слева, как при прямом вычислении (отсюда и название алгоритма — *обратное распространение ошибок*):



Имея частные производные по a^m и u^h, легко выписать градиент Q по весам:

$$\frac{\partial Q(w)}{\partial w_{hm}} = \frac{\partial Q(w)}{\partial a^m}\frac{\partial a^m}{\partial w_{hm}} = \varepsilon_i^m \sigma'_m u^h, \qquad m = 1,\ldots,M, \quad h = 0,\ldots,H;$$

$$\frac{\partial Q(w)}{\partial w_{jh}} = \frac{\partial Q(w)}{\partial u^h}\frac{\partial u^h}{\partial w_{jh}} = \varepsilon_i^h \sigma'_h v^j, \qquad h = 1,\ldots,H, \quad j = 0,\ldots,J;$$

и так далее для каждого слоя. Если слоёв больше двух, то остальные частные производные вычисляются аналогично – обратным ходом по слоям сети справа налево.

3.4. Многослойные нейронные сети

АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ. ДОСТОИНСТВА

- Достаточно высокая эффективность. Прямой ход, обратный ход и вычисления градиента требуют порядка $O(Hn + HM)$ операций.
- Через каждый нейрон проходит информация только о связных с ним нейронах. Поэтому back-propagation легко реализуется на вычислительных устройствах с параллельной архитектурой.
- Высокая степень общности. Алгоритм легко записать для произвольного числа слоёв, произвольной размерности выходов и выходов, произвольной функции потерь и произвольных функций активации, возможно, различных у разных нейронов. Кроме того, back-propagation не накладывает никаких ограничений на используемый метод оптимизации. Его можно применять вместе с методом скорейшего спуска, сопряженных градиентов, Ньютона-Рафсона и др.

3.4. Многослойные нейронные сети

АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ. НЕДОСТАТКИ

- Метод не всегда сходится. Для улучшения сходимости приходится применять большое количество различных эвристических ухищрений.
- Процесс градиентного спуска склонен застревать в многочисленных локальных минимумах функционала Q.
- Приходится заранее фиксировать число нейронов скрытого слоя H. В то же время, это критичный параметр сложности сети, от которого может существенно зависеть качество обучения и скорость сходимости.
- При чрезмерном увеличении числа весов сеть склонна к переобучению.
- Если применяются функции активации с горизонтальными асимптотами, типа сигмоидной или th, то сеть может попадать в состояние «паралича». Чем больше значения синаптических весов на входе нейрона, тем ближе значение производной σ' к нулю, тем меньше изменение синаптических весов
  Если нейрон один раз попадает в такую «мёртвую
  зону», то у него практически не остаётся шансов из неё выбраться. Парализоваться могут отдельные связи, нейроны, или вся сеть в целом.

3.4. Многослойные нейронные сети

УЛУЧШЕНИЕ СХОДИМОСТИ И КАЧЕСТВА ГРАДИЕНТНОГО ОБУЧЕНИЯ

- Нормализация данных
- Выбор функций активации
- Выбор начального приближения
- Порядок предъявления объектов
- Сокращение весов
- Выбор величины шага
- Выбивание сети из локальных минимумов
- Выбор критерия останова
- Ранний останов
- Выбор градиентного метода оптимизации
- Оптимизация структуры сети

Литература

Воронцов, К. В. (2007). Лекции по искусственным нейронным сетям.