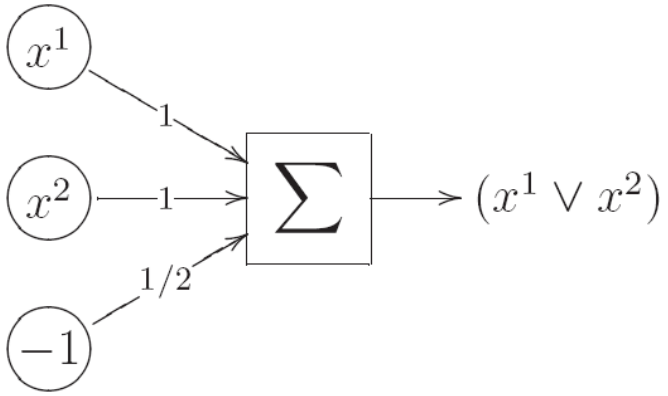
Легко построить нейроны, реализующие логические функции И, ИЛИ, НЕ от бинарных переменных x^1 и x^2.

$$x^1 \vee x^2 = \left[x^1 + x^2 - \tfrac{1}{2} > 0\right];$$
$$x^1 \wedge x^2 = \left[x^1 + x^2 - \tfrac{3}{2} > 0\right];$$
$$\neg x^1 = \left[-x^1 + \tfrac{1}{2} > 0\right];$$

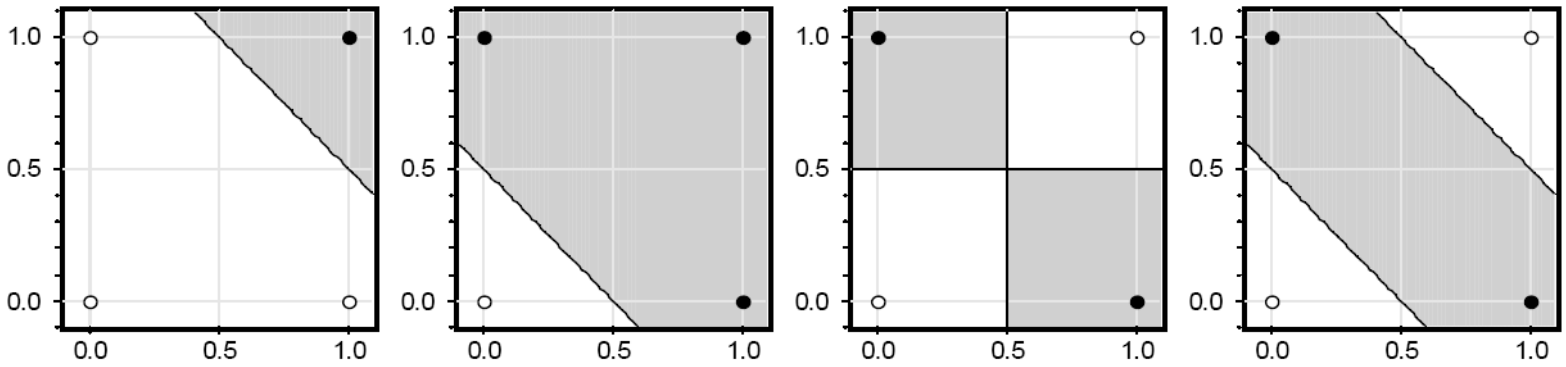
Однако функцию $x^1 \oplus x^2 = [x^1 \neq x^2]$ – *исключающее ИЛИ* (exclusive or, XOR) принципиально невозможно реализовать одним нейроном с двумя входами x^1 и x^2, поскольку множества нулей и единиц этой функции линейно неразделимы.



Возможны два пути решения этой проблемы

Первый путь — пополнить состав признаков, подавая на вход нейрона нелинейные преобразования исходных признаков. В частности, если разрешить образовывать всевозможные произведения исходных признаков, то нейрон будет строить уже не линейную, а полиномиальную разделяющую поверхность. В случае исключающего ИЛИ достаточно добавить только один вход x^1x^2, чтобы в расширенном пространстве множества нулей и единиц оказались линейно разделимыми:

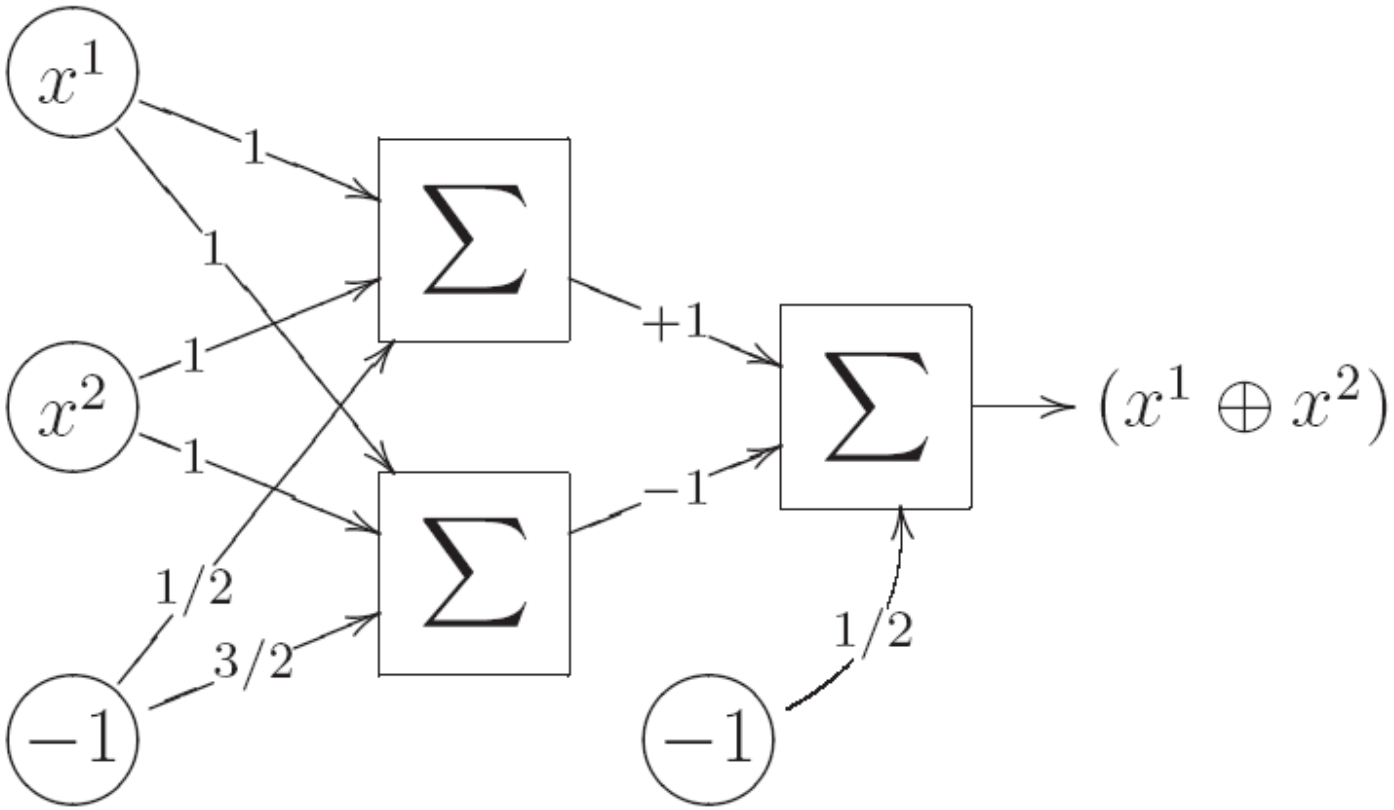
$$x^1 \oplus x^2 = \left[x^1 + x^2 - 2x^1x^2 - \frac{1}{2} > 0\right].$$



подбор нужных нелинейных преобразований является нетривиальной задачей, которая для общего случая до сих пор остаётся нерешённой.

Второй путь построить композицию из нескольких нейронов. Например, исключающее ИЛИ можно реализовать, подав выходы И-нейрона и ИЛИ-нейрона на вход ещё одному ИЛИ-нейрону, -> многослойные нейронные сети

$$x^1 \oplus x^2 = \left[(x^1 \vee x^2) - (x^1 \wedge x^2) - \tfrac{1}{2} > 0\right].$$



Вычислительные возможности нейронных сетей.

1. 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.

2. Из 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 невыпуклые и даже неодносвязные, а также аппроксимировать любые непрерывные функции.

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.

Нетрудно видеть, что записанное здесь выражение имеет структуру нейронной сети с одним скрытым слоем из $2n+1$ нейронов. Таким образом, двух слоёв уже достаточно, чтобы вычислять произвольные непрерывные функции, и не приближённо, а точно. К сожалению, представление Колмогорова не является персептроном: функции φ_{ik} не линейны, а функции h_k зависят от f, и в общем случае не являются дифференцируемыми.

Теорема 6.3 (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.

Опр. 6.1. *Набор функций F называется разделяющим точки множества X, если для любых различных $x, x' \in X$ существует функция $f \in F$ такая, что $f(x) \neq f(x')$.*

Опр. 6.2. *Набор функций $F \subseteq C(X)$ называется замкнутым относительно функции $\varphi : \mathbb{R} \to \mathbb{R}$, если для любого $f \in F$ выполнено $\varphi(f) \in F$.*

Методы обучения нейронной сети

$$E = \frac{1}{2}\sum_{l=1}^{M}\sum_{i=1}^{N}(y_i(W) - d_{l,i})^2 \to \min$$

Метод обратного Распространения ошибки

$$w_{ij}(t+1) = w_{ij}(t) - h\frac{\partial E}{\partial w_{ij}}$$

$$\delta_j^{(n)} = \frac{\partial y_j^{(n)}}{\partial S_j^{(n)}}\sum_k \delta_j^{(n+1)} w_{jk}^{(n+1)}$$

$$\delta_j^{(n)} = (y_i^{(n)} - d_i)\frac{\partial y_j^{(n)}}{\partial S_j^{(n)}}$$

$$\Delta w_{ij} = -h\delta_j^{(n)} x_i^n$$

$$w_{ij}(t+1) = w_{ij}(t) + h\Delta w_{ij}$$

Стратегии улучшения работы алгоритма обучения

Устранение возможной блокировки сети

$$\Delta w_{ij}(t) = -h\delta_j^{(n)} x_i^n + \mu\Delta w_{ij}(t-1)$$

Устранение переобучения, путем кросс-проверки точности сети на другой выборке

Обхождение локальных минимумов с помощью увеличения нейронов скрытого слоя. Случайное изменения весовых коэффициентов

Достоинства метода обратного распространения.

- Достаточно высокая эффективность. В случае двухслойной сети 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 можно применять совместно с различными градиентными методами оптимизации: методом скорейшего спуска, сопряженных градиентов, Ньютона-Рафсона и др.

Недостатки метода обратного распространения.

- Метод наследует известные недостатки градиентной настройки весов в однослойном персептроне. Здесь также возникают проблемы медленной сходимости

  или расходимости, «застревания» в локальных минимумах функционала Q, переобучения и паралича. Причём парализоваться могут отдельные связи, нейроны, или вся сеть в целом.

- 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.

Оптимизация структуры сети

Выбор структуры сети, то есть числа слоёв, числа нейронов и числа связей для каждого нейрона, является, пожалуй, наиболее сложной проблемой. Существуют различные стратегии поиска оптимальной структуры сети: постепенное наращивание, построение заведомо слишком сложной сети с последующим упрощением, поочерёдное наращивание и упрощение.

Проблема выбора структуры тесно связана с проблемами недообучения и переобучения. Слишком простые сети не способны адекватно моделировать целевые зависимости в реальных задачах. Слишком сложные сети имеют избыточное число свободных параметров, которые в процессе обучения настраиваются не только на восстановление целевой зависимости, но и на воспроизведение шума.

Выбор числа слоёв. Если в конкретной задаче гипотеза о линейной разделимости классов выглядит правдоподобно, то можно ограничиться однослойным персептроном. Двухслойные сети позволяют представлять извилистые нелинейные границы, и в большинстве случаев этого хватает. Трёхслойными сетями имеет смысл пользоваться для представления сложных многосвязных областей. Чем больше слоёв, тем более богатый класс функций реализует сеть, но тем хуже сходятся градиентные методы, и тем труднее её обучить.

Выбор числа нейронов в скрытом слое H производят различными способами, но ни один из них не является лучшим.

1. Визуальный способ. Если граница классов (или кривая регрессии) слишком сглажена, значит, сеть переупрощена, и необходимо увеличивать число нейронов в скрытом слое. Если граница классов (или кривая регрессии) испытывает слишком резкие колебания, на тестовых данных наблюдаются большие выбросы, веса сети принимают большие по модулю значения, то сеть переусложнена, и скрытый слой следует сократить. Недостаток этого способа в том, что он подходит только для задач с низкой размерностью пространства (небольшим числом признаков).

2. Оптимизация H по *внешнему критерию*, например, по критерию скользящего контроля или средней ошибки на независимой контрольной выборке $Q(X^k)$. Зависимость внешних критериев от параметра сложности, каким является H, обычно имеет характерный оптимум. Недостаток этого способа в том, что приходится много раз заново строить сеть при различных значениях параметра H, а в случае скользящего контроля — ещё и при различных разбиениях выборки на обучающую и контрольную части.

Динамическое добавление нейронов. Сначала сеть обучается при заведомо недостаточной мощности среднего слоя $H \ll \ell$. Обучение происходит до тех пор, пока ошибка не перестанет убывать. Тогда добавляется один или несколько новых нейронов. Веса новых связей инициализируются небольшими случайными числами, либо добавленные нейроны обучаются по-отдельности как однослойные персептроны. Во втором случае можно рекомендовать обучать новый персептрон на случайной подвыборке, возможно, добавив в неё те объекты, на которых текущая сеть допустила наибольшие ошибки. Веса старых связей не меняются. Затем проводится настройка сети методом обратного распространения.

После добавления новых нейронов ошибка, как правило, сначала резко возрастает, затем быстро сходится к меньшему значению. Интересно, что общее время обучения обычно оказывается лишь в 1.5–2 раза больше, чем если бы в сети сразу было нужное количество нейронов. Это означает, что информация, накопленная в сети, является полезной и не теряется при добавлении новых нейронов.

При постепенном наращивании сети целесообразно наблюдать за динамикой какого-нибудь внешнего критерия. Прохождение значения $Q(X^k)$ через минимум является надёжным критерием останова, так как свидетельствует о переобученности, вызванной чрезмерным усложнением сети.

Удаление избыточных связей. Метод *оптимального прореживания сети* (optimal brain damage, OBD) [51, 42] удаляет те связи, к изменению которых функционал Q наименее чувствителен. Уменьшение числа весов снижает склонность сети к переобучению.

Метод OBD основан на предположении, что после стабилизации функционала ошибки Q вектор весов w находится в локальном минимуме, где функционал может быть аппроксимирован квадратичной формой:

$$Q(w+\delta) = Q(w) + \tfrac{1}{2}\delta^{\mathsf{T}} H(w)\delta + o(\|\delta\|^2),$$

где $H(w) = \left(\frac{\partial^2 Q(w)}{\partial w_{jh}\partial w_{j'h'}}\right)$ — гессиан, матрица вторых производных. Как и в диагональном методе Левенберга–Марквардта, предполагается, что диагональные элементы доминируют в гессиане, а остальными частными производными можно пренебречь, положив их равными нулю. Это предположение носит эвристический характер и вводится для того, чтобы избежать трудоёмкого вычисления всего гессиана.

Если гессиан $H(w)$ диагонален, то

$$\delta^{\intercal} H(w)\delta = \sum_{j=0}^{J}\sum_{h=1}^{H} \delta_{jh}^2 \frac{\partial^2 Q(w)}{\partial w_{jh}^2}.$$

Обнуление веса w_{jh} эквивалентно выполнению условия $w_{jh} + \delta_{jh} = 0$. Введём величину *значимости* (salience) синаптической связи, равную изменению функционала $Q(w)$ при обнулении веса: $S_{jh} = w_{jh}^2 \frac{\partial^2 Q(w)}{\partial w_{jh}^2}$.

Эвристика OBD заключается в том, чтобы удалить из сети d синапсов, соответствующих наименьшим значениям S_{jh}. Здесь d — это ещё один параметр метода настройки. После удаления производится цикл итераций до следующей стабилизации функционала Q. При относительно небольших значениях d градиентный алгоритм довольно быстро находит новый локальный минимум Q. Процесс упрощения сети останавливается, когда *внутренний критерий* стабилизируется, либо когда заданный *внешний критерий* начинает возрастать.

Обнуление веса w_{jh} между входным и скрытым слоями означает, что h-й нейрон скрытого слоя не будет учитывать j-й признак. Тем самым происходит отбор информативных признаков для h-го нейрона скрытого слоя.

Метод OBD легко приспособить и для настоящего *отбора признаков*. Вводится суммарная значимость признака $S_j = \sum_{h=1}^{H} S_{jh}$, и из сети удаляется один или несколько признаков с наименьшим значением S_j.

Обнуление веса w_{hm} между скрытым и выходным слоями означает, что m-е выходное значение не зависит от h-го нейрона скрытого слоя. Если выход одномерный $(M = 1)$, то h-й нейрон можно удалить. В случае многомерного выхода для удаления нейронов скрытого слоя вычисляется суммарная значимость $S_h = \sum_{m=1}^{M} S_{hm}$.

Метод псевдообратных матриц

$$Y = F^K\left(W^K \cdot F^{K-1}\left(\ldots F^2\left(W^2 \cdot F^1\left(W^1 \cdot X\right)\right)\right)\right)$$

Однослойный персептрон

$$W = F^{-1}(Y) \cdot X^T \cdot (X \cdot X^T)^{-1}$$

Преимущества метода

Высокая скорость обучения

Многослойный персептрон

$$W^K = (F^K)^{-1}(Y) \cdot FI^{K-1}$$

$$W^{K-1} = (F^{K-1})^{-1}(WI^K \cdot (F^K)^{-1}(Y)) \cdot FI^{K-2}$$

$$W^1 = (F^1)^1(WI^2 \cdot (F^2)^{-1}(\ldots WI^K \cdot (F^K)^{-1}(Y))) \cdot X^T \cdot (X \cdot X^T)^{-1}$$

$$FI^K = FF^{T,K} \cdot (FF^K \cdot FF^{T,K})^{-1}$$

$$WI^K = ((W^K)^T \cdot W^K)^{-1} \cdot (W^K)^T$$

Сеть с комбинированным обучением