Deep Learning

Остапец Андрей

22 апреля 2013 г.

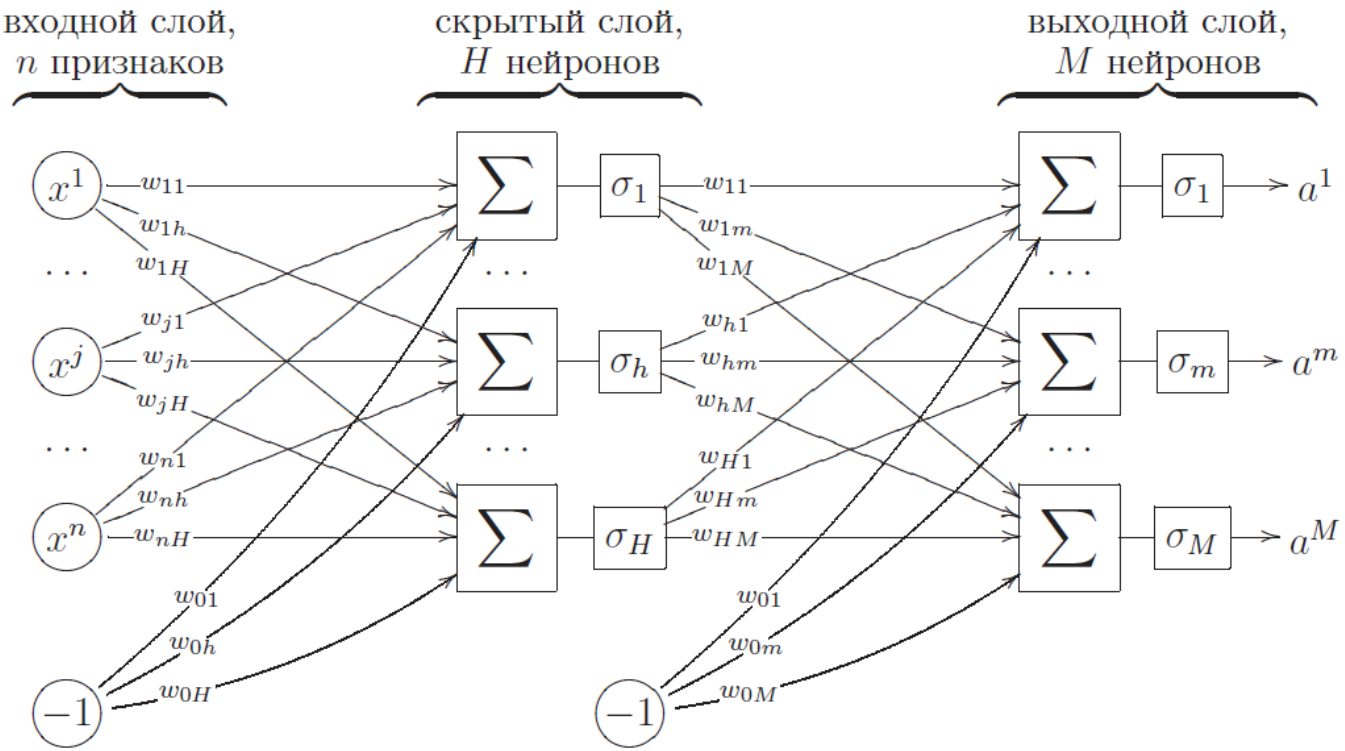
Содержание

Содержание

Содержание

Содержание

Что такое нейронные сети?



Ликбез. Back-propagation

Выходные значения $a^m(x_i), \quad m = 1 \dots M$ на объекте x_i:

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

Зафиксируем объект x_i и запишем функционал среднеквадратичной ошибки:

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

Ликбез. Back-propagation

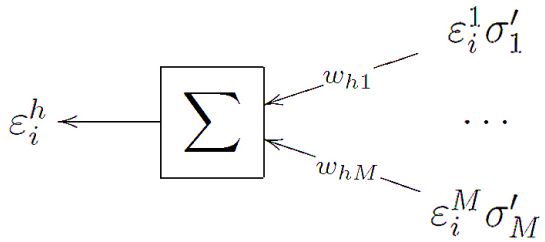
Частные производные:

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

- это ошибка на выходном слое;

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

- назовём это ошибкой на скрытом слое. Она вычисляется, если запустить сеть в „обратном направлении”:



Ликбез. Back-propagation

Теперь, имея частные производные $Q(w)$ по a^m и u^h, легко выписать градиент $Q(w)$ по весам w:

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

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

Теперь есть все необходимое для запуска алгоритма стохастического градиента.

Deep learning

Deep learning - это подраздел Machine Learning, основанный на двух идеях:

- Обучение с использованием большого количества уровней представления информации для моделирования комплексных отношений в данных
- Обучение на немаркированных данных ("без учителя") или на комбинации немаркированных и маркированных данных ("с частичным привлечением учителя").

Считаем, что существует иерархия признаков - высокоуровневые признаки и концепты определяются с помощью низкоуровневых.

Пример неудачной работы нейронных сетей

Возьмем задачу распознавания цифр из базы рукописных символов MNIST.

1. У нас есть 60 000 изображений размера 32 на 32, хотим научиться распознавать цифры.
2. Возьмем нейронную сеть, примерно 1000 входов, 10 выходов. И еще возьмем один скрытый слой, например 500 узлов.
3. Итого у нас 1000*500 + 500*10 = 505 000 весов.
4. Локальный минимум гарантирован.
5. Обобщающая способность нулевая.

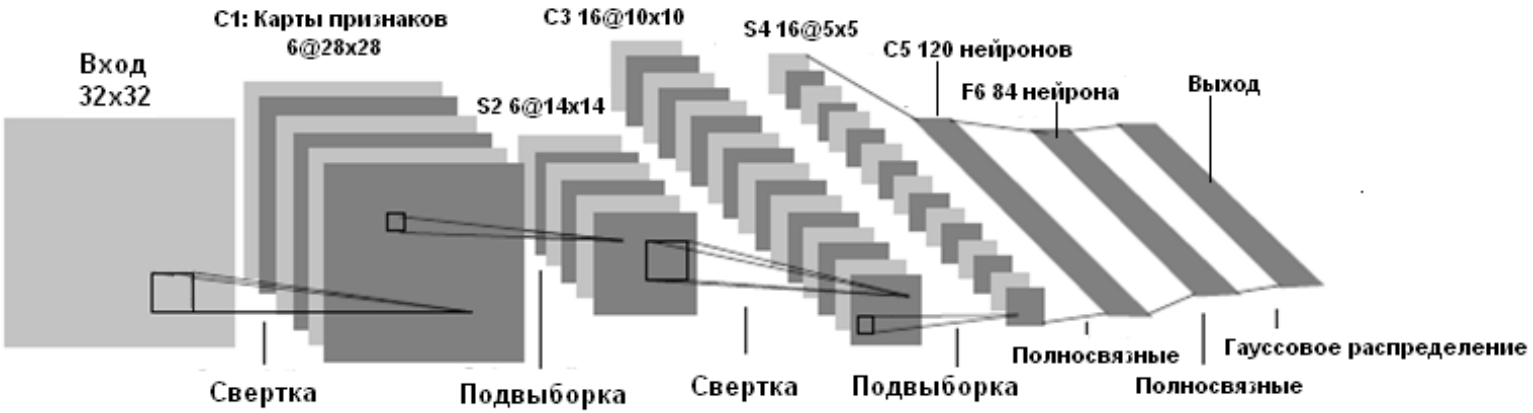
Биологические предпосылки сверточных нейронных сетей

- Мозг - это не просто куча связанных нейронов
- Cortex (кора головного мозга) пропускает информацию через несколько иерархических слоев предобработки, и каждый слой при этом выполняет свою функцию.
- Нейроны, которые выполняют похожие функции работают совместно.

Сверточные сети

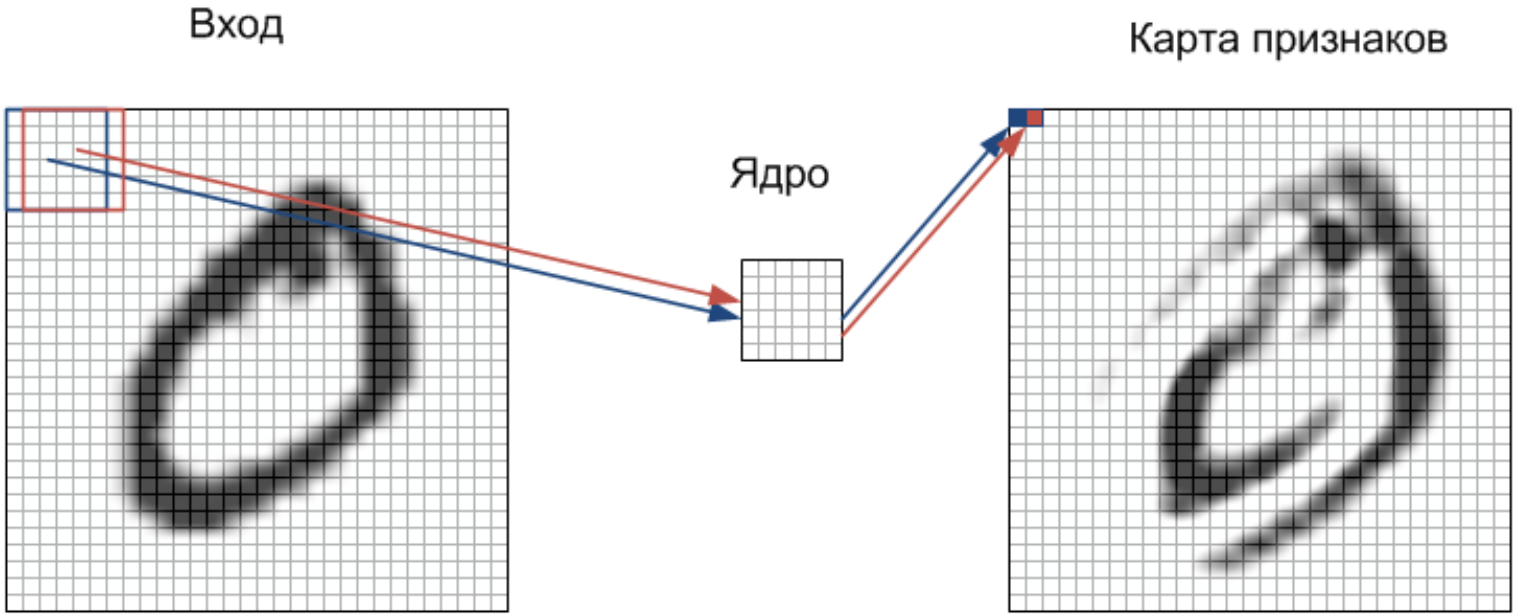
Ян ЛеКун предложил использовать так называемые сверточные нейронные сети. Идея сверточных нейронных сетей заключается в чередовании

- свёрточных слоев (C-layers),
- субдискретизирующих слоев (S-layers)
- полносвязных (F-layers) слоев на выходе.



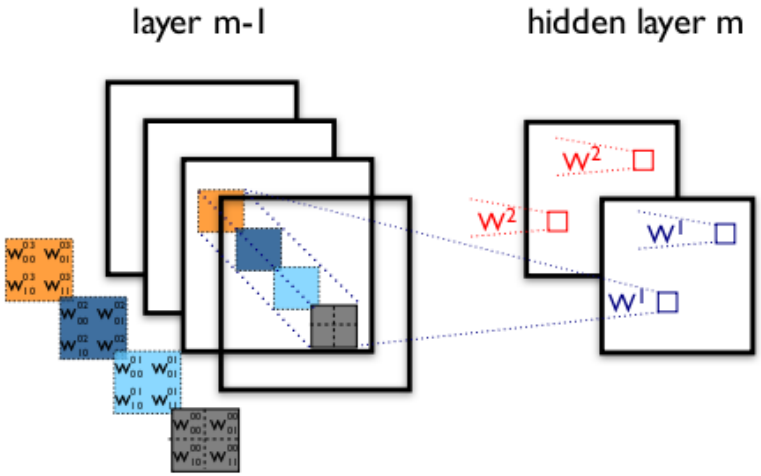
Первый сверточный слой

Каждый фрагмент изображения поэлементно умножается на небольшую матрицу весов (ядро), результат суммируется. Эта сумма является пикселом выходного изображения, которое называется картой признаков. Помимо этого, взвешенная сумма входов еще пропускается через функцию активации.



S-слой

Суть S-слоев заключается в уменьшении пространственной размерности изображения. Т.е. входное изображение грубо (усреднением) уменьшается в заданное количество раз. Чаще всего в 2 раза, хотя может быть и не равномерное изменение, например, 2 по вертикали и 3 по горизонтали.
Субдискретизация нужна для обеспечения инвариантности к масштабу.



Второй сверточный слой

- Соединение всех карт второго слоя со всеми картами третьего слоя значительно увеличило бы количество связей.
- Соединение карт „одна к одной” стало бы еще одним повторением свертки, которое уже присутствовало между первым и вторым слоями.

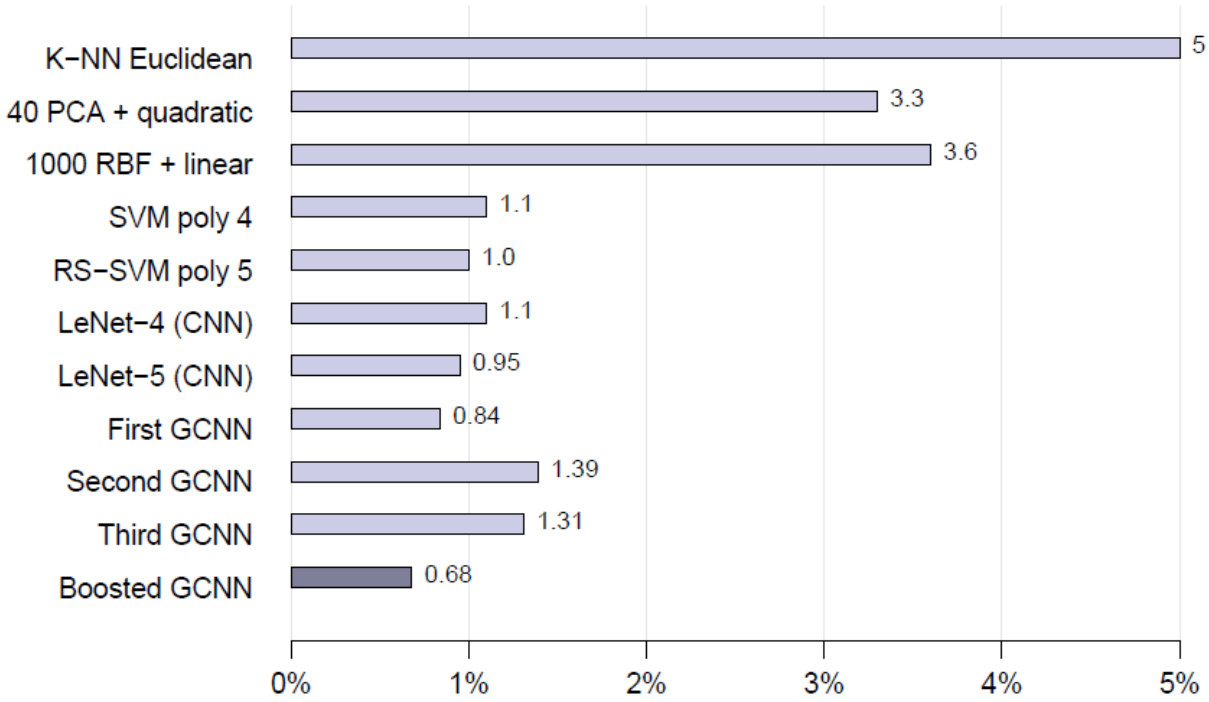
Как правило, архитектор сети сам принимает решение о том, по какому принципу организовывать соединение карт второго и третьего слоев.

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	+				+	+	+			+	+	+	+		+	+
2	+	+				+	+	+			+	+	+	+		+
3	+	+	+				+	+	+			+		+	+	+
4		+	+	+			+	+	+	+			+		+	+
5			+	+	+			+	+	+	+		+	+		+
6				+	+	+			+	+	+	+		+	+	+

Свойства CNN

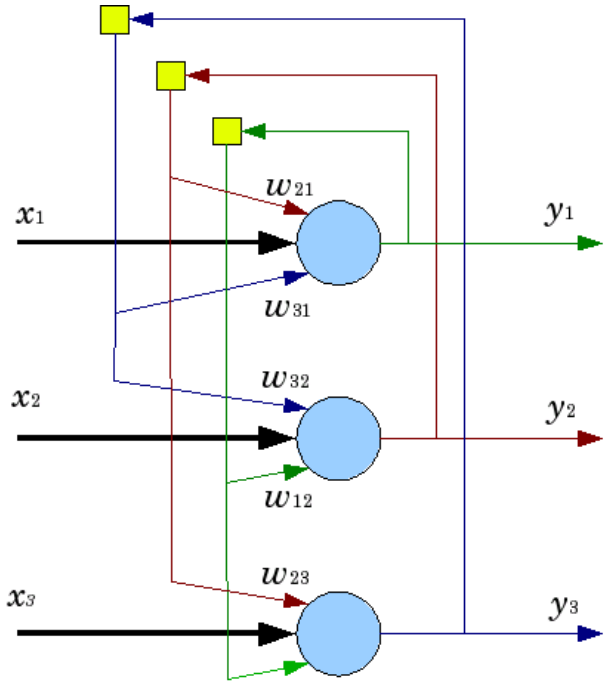
- Локальное восприятие подразумевает, что на вход одного нейрона подается не все изображение (или выходы предыдущего слоя), а лишь некоторая его область. Такой подход позволил сохранять топологию изображения от слоя к слою.
- Концепция разделяемых весов предполагает, что для большого количества связей используется очень небольшой набор весов.
- Хорошие обобщающие свойства сети, что в итоге позитивно сказывается на способности сети находить инварианты в изображении и реагировать главным образом на них, не обращая внимания на прочий шум.

Результаты работы



Нейронная сеть Хопфилда

Нейронная сеть Хопфилда - полносвязная нейронная сеть с симметричной матрицей связей. В процессе работы динамика таких сетей сходится (конвергирует) к одному из положений равновесия. Эти положения равновесия являются локальными минимумами функционала, называемого энергией сети



Нейронная сеть Хопфилда

В сети Хопфилда матрица связей является симметричной ($w_{ij} = w_{ji}$), а диагональные элементы матрицы полагаются равными нулю ($w_{ii} = 0$), что исключает эффект воздействия нейрона на самого себя и является необходимым для сети Хопфилда.

Каждый нейрон системы может принимать одно из двух состояний (что аналогично выходу нейрона с пороговой функцией активации):

$$s_i = \begin{cases} 1, & \text{if } \sum w_{ji} s_j > \theta_i, \\ -1, & \text{иначе} \end{cases}$$

Нейронная сеть Хопфилда

Энергия:

$$E = -\frac{1}{2}\sum_{i,j} w_{ij} s_i s_j + \sum_i \theta_i s_i,$$

- w_{ij} - сила связи между i-ым и j-ым нейронами,
- θ_i - индивидуальные порог для каждого нейрона
- s_i - состояние нейрона.

E - мера близости к стабильному состоянию

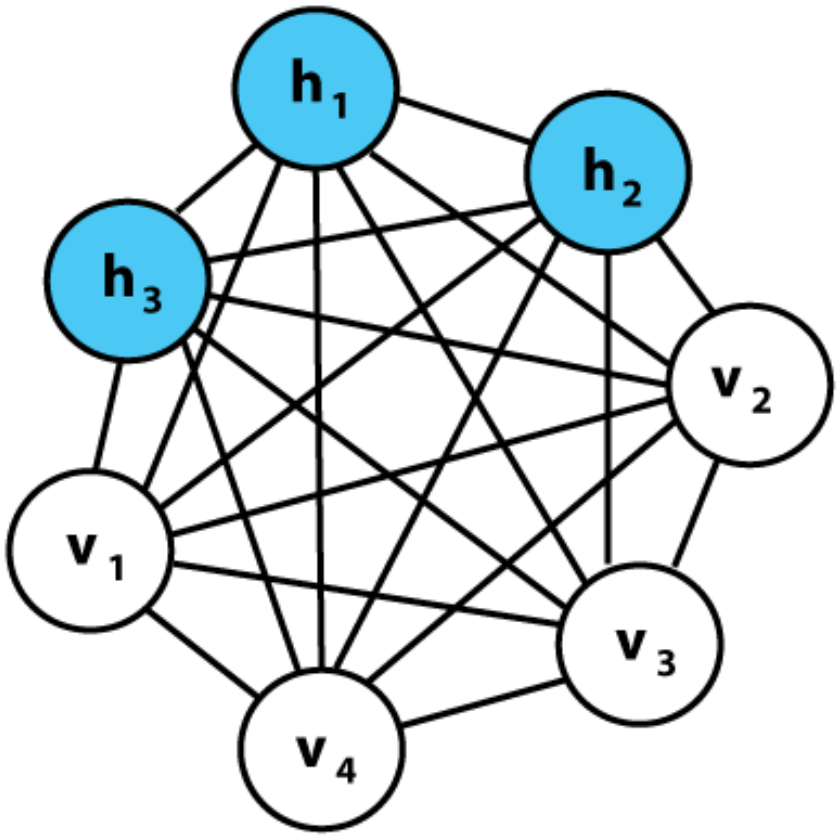
Обыкновенная машина Больцмана

Машина Больцмана представляет из себя полносвязный неориентированный граф, где нейроны поделены на две группы, описывающие обозреваемые и скрытые состояния. Таким образом, любые две вершины из одной группы зависят друг от друга.

Энергия в машинах Больцмана выражается следующим образом:

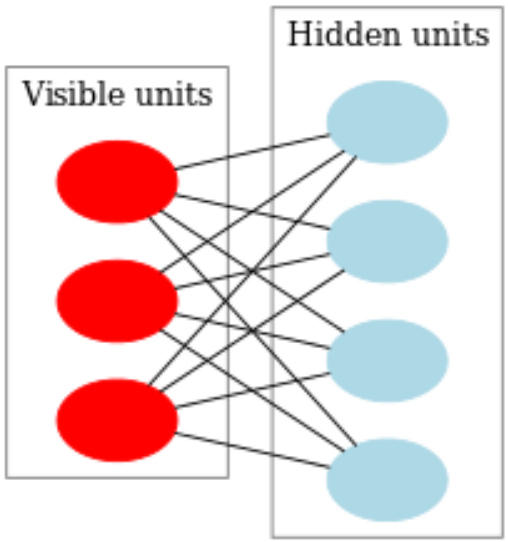
$$E = -\sum_{i<j} w_{ij} s_i s_j + \sum_i \theta_i s_i,$$

Обыкновенная машина Больцмана



Ограниченная машина Больцмана

Но если убрать связи внутри группы, чтобы получился двудольный граф, мы получим структуру модели RBM.



Особенность этой модели в том, что при данном состоянии нейронов одной группы, состояния нейронов другой группы будут независимы друг от друга.

Ограниченная машина Больцмана

RBM интерпретируются аналогично скрытым моделям Маркова.
У нас есть ряд состояний, которые мы можем наблюдать (видимые нейроны, которые предоставляют интерфейс для общения с внешней средой) и ряд состояний, которые скрыты, и мы не можем напрямую увидеть их состояние (скрытые нейроны). Но мы можем сделать вероятностный вывод относительно скрытых состояний, опираясь на состояния, которые мы можем наблюдать.

Ограниченная машина Больцмана

Введем следующие обозначения:

- w_{ij} - вес между i-ым нейроном
- a_i - смещение видимого нейрона
- b_j - смещение скрытого нейрона
- v_i - состояние видимого нейрона
- h_j - состояние скрытого нейрона

Мы будем рассматривать обучающее множество, состоящее из бинарных векторов. Предположим, что у нас n видимых нейронов и m скрытых. Введем понятие энергии для RBM:

$$E(v,h) = -\sum_{i}^{n} a_i v_i - \sum_{j}^{m} b_j h_j - \sum_{i}^{n}\sum_{j}^{m} w_{ij} v_i h_j$$

Ограниченная машина Больцмана

Нейросеть будет вычислять совместную вероятность всевозможных пар v и h следующим образом:

$$p(v,h) = \frac{1}{Z} e^{-E(v,h)},$$

где Z - это статсумма следующего вида:

$$Z = \sum_{r}^{2^n} \sum_{t}^{2^m} e^{-E(v^r,h^t)}$$

Очевидно, что полная вероятность вектора v будет вычисляться суммированием по всем h:

$$p(v) = \sum_{t}^{M} P(v,h^t) = \frac{1}{Z} \sum_{t}^{M} e^{-E(v,h^t)}$$

Ограниченная машина Больцмана

Рассмотрим вероятность того, что при данном v одно из скрытых состояний $h_k = 1$. Для этого представим один нейрон, тогда энергия системы при 1 будет E_1, а при 0 будет E_0.

$$p(h_k = 1|v) = \frac{e^{-E_1}}{e^{-E_1} + e^{-E_0}} = \frac{1}{1 + e^{E_1 - E_0}} =$$

$$= \frac{1}{1 + e^{-b - \sum_i^n v_i w_{ik}}} = \sigma\Big(- b - \sum_i^n v_i w_{ik} \Big)$$

А так как при данном v все h_k не зависят друг от друга, то:

$$P(h|v) = \prod_j^m P(h_j|v)$$

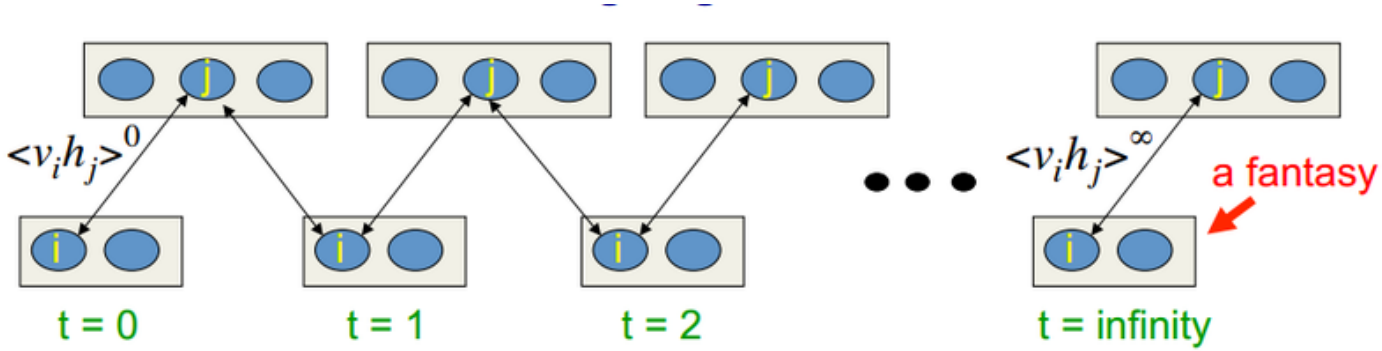
Аналогичный вывод делается и для вероятности v при данном h.

Алгоритм обучения Contrastive Divergence

Этот алгоритм придуман профессором Хинтоном в 2002 году, и он отличается своей простотой.

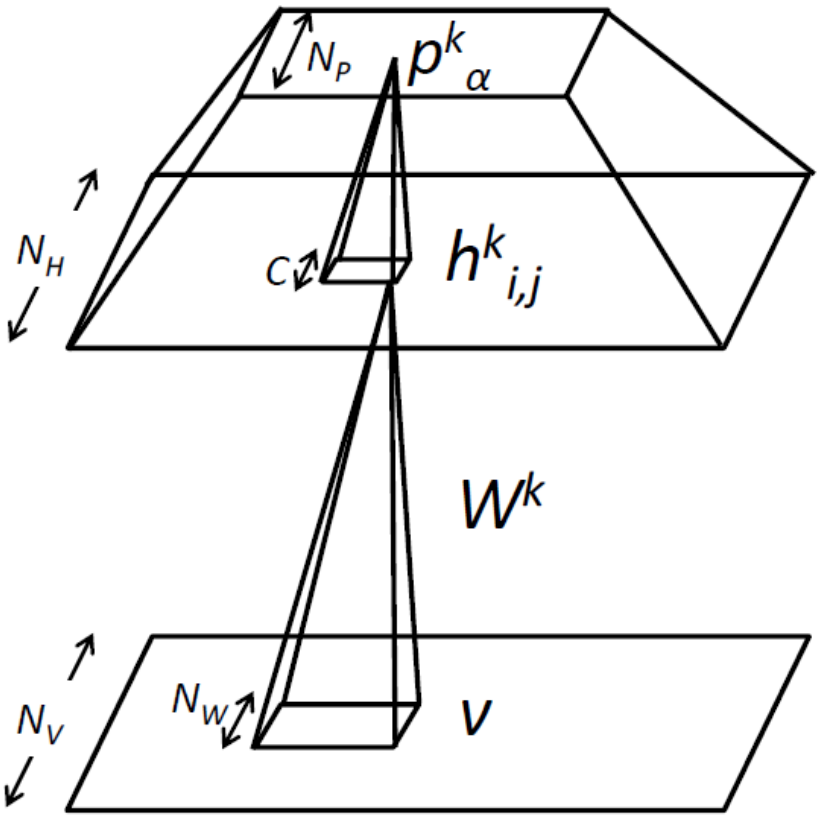
1. состояние видимых нейронов приравнивается к входному образу
2. выводятся вероятности состояний скрытого слоя
3. каждому нейрону скрытого слоя ставится в соответствие состояние «1» с вероятностью, равной его текущему состоянию
4. выводятся вероятности видимого слоя на основании скрытого
5. если текущая итерация меньше k, то возврат к шагу 2
6. выводятся вероятности состояний скрытого слоя

Алгоритм обучения Contrastive Divergence



- $\Delta w_{ij} = \eta\left(M[v_i h_j]^{(0)} - M[v_i h_j]^{(\infty)}\right)$
- $\Delta a_i = \eta\left(v_i - M[v_i]^{(\infty)}\right)$
- $\Delta b_j = \eta\left(M[h_j]^{(0)} - M[h_j]^{(\infty)}\right)$

Deep belief nets



P^k (pooling layer)

H^k (detection layer)

V (visible layer)