

Нейронные сети

Часть 7

Сеть Хопфилда

Сети с обратными связями могут работать в качестве *ассоциативной памяти*. Это означает, что по вектору, поданному на вход, сетью будет создан на выходе один из запомненных ранее векторов, наиболее "похожий" (в некотором выбранном смысле) на данный входной вектор. Такой способ выборки данных называется *адресацией по содержимому*, в отличие от адресации по номеру ячейки памяти, принятому в ЭВМ фон неймановского типа.

Память с такой адресацией является весьма перспективной для создания систем искусственного интеллекта, систем распознавания речевых сигналов и изображений.

Пусть задано множество векторов $\{x^k\} = x^1 \dots x^k$, подлежащих запоминанию в нейросети. Критерий "похожести" векторов зависит от задачи и может быть весьма сложным.

Если, к примеру, входной вектор состоит из нескольких отсчетов сигнала во времени, взятых подряд, то критерий близости векторов должен обладать инвариантностью к масштабированию отсчетов, к переносу вдоль оси времени (фазе сигнала), и к зашумленности входных данных. Так что поиск критерия близости специфичен для каждой задачи и представляет собой серьезную проблему.

Рассмотрим простейший случай, когда в качестве меры близости двух векторов используется их скалярное произведение $\sum_k x^k y^k$. Чем более "похожи" вектора, тем больше мера близости. Тогда можно ожидать, что изменение вектора x с течением времени по закону

$x^k = x_0^k e^{-\alpha k}$ в конце концов приведет к тому, что x совпадет с y на наиболее похожим эталоном, т.е. требуемая ассоциация будет найдена.

Если найти такую функцию $H(x)$, что $\frac{dx}{dt} = -\nabla H$

то задача поиска ассоциации для вектора x будет совпадать с задачей поиска минимума функции $H(x)$. Если выбрать функцию H , которая называется *энергией сети*, в виде

$$H(x) = -\frac{1}{2} \sum_k (x^k, x)^2 + \frac{1}{2} \lambda \sum_i (x_i^2 - 1)^2$$

то ее градиент

$$\nabla H = -\sum_k x^k (x^k, x) + \lambda \sum_i e^i (x_i^2 - 1) x_i$$

здесь e^i - i -й вектор базиса. Нижними индексами обозначены компоненты вектора, верхними - номер вектора в множестве. Тогда

$$\frac{dx}{dt} = \sum_k x^k (x^k, x) - \lambda \sum_i e^i (x_i^2 - 1) x_i$$

Первое слагаемое обеспечивает стремление x к ближайшим эталонам, а второе — приближение компонент вектора x к допустимым значениям $+1$ или 1 . Коэффициент λ соотносит интенсивности этих двух процессов. Обычно λ растет с течением времени от $\lambda < 1$ до $\lambda > 1$ с ростом времени обучения. Чтобы получить весовые коэффициенты и пороговые уровни, запишем последнее выражение покомпонентно:

$$\begin{aligned} \left(\frac{dx}{dt} \right) &= \sum_j \sum_k x_i^k x_j^k x_j - \lambda (x_i^2 - 1) x_i = \\ &= \sum_{j \neq i} \sum_k x_i^k x_j^k x_j - (\lambda (x_i^2 - 1) - 1) x_i \end{aligned}$$

Отсюда видно, что весовые коэффициенты должны определяться:

$$W_{ij} = \begin{cases} \sum_k x_i^k x_j^k, & i \neq j \\ 0, & i = j \end{cases}$$

Веса не зависят от направления от i к j или от j к i . Данная формула аналогична формуле Хебба для обучения перцептрона, когда связь между нейронами i и j положительна, если состояния нейронов одинаковы и отрицательна, если состояния противоположны.

Каждый i -й нейрон содержит управляемый нелинейный порог со значением

$$\theta_i = (\lambda(x_i^2 - 1) - 1)x_i$$

рассчитываемый на каждой итерации. Каждый выходной сигнал подается обратно на вход сети с весом W_{ij} и на вход того же нейрона для расчета порога. Данная модель называется *сетью Хопфилда*.

Решение задач с помощью сетей Хопфилда;

1. Построить функцию энергии таким образом, чтобы точка глобального минимума этой функции совпадала с решением задачи. При этом градиент функции энергии должен допускать вычисление с помощью НС.

2. Записать формулы для расчета параметров сети (весовых коэффициентов и пороговых уровней) для расчета градиента функции энергии.

3. Разорвать цепочку обратной связи и предъявить сети входной вектор. Рассчитать значения выходов.

4. Замкнуть обратную связь и предоставить сети возможность самостоятельно менять свое состояние (релаксация). Остановить процесс релаксации после того, как выходной вектор перестанет меняться, т.е. по достижении минимума функции энергии. Полученные выходы сети дают решение задачи.

Задача комбинаторной оптимизации

В задачах комбинаторной оптимизации требуется найти наилучшее из конечного, но обычно очень большого числа возможных решений.

Рассмотрим задачу комбинаторной оптимизации на примере задачи коммивояжера. Для решения этой задачи с помощью нейронной сети Хопфилда нужно закодировать маршрут активностью нейронов и так подобрать связи между ними, чтобы энергия сети оказалась связанной с полной длиной маршрута. Хопфилд предложил для этого следующий способ.

Рассмотрим сеть, состоящую из $N \times N$ бинарных нейронов, состояния которых мы обозначим $v_{i\alpha}$, где индекс i - кодирует город, а индекс α - номер города в маршруте.

Если обозначить через d_{ij} расстояние между i -м и j -м городами, решение задачи коммивояжера сводится к минимизации целевой функции

$$L(\mathbf{v}) = \frac{1}{2} \sum_{i,j,\alpha}^{i \neq j} d_{ij} v_{i\alpha} (v_{j\alpha-1} + v_{j\alpha+1})$$

при дополнительных условиях

$$\sum_i v_{i\alpha} = 1 \quad (\forall \alpha) \quad \sum_{\alpha} v_{i\alpha} = 1 \quad (\forall i)$$

первое из которых говорит о том, что любой город в маршруте встречается лишь однажды, а второе - что маршрут проходит через каждый город.

Общий подход к ограничениям в задачах оптимизации состоит в том, что в итоговый функционал, подлежащий минимизации, включаются штрафные члены, увеличивающие целевую функцию при отклонении от накладываемых ограничений.

В данном случае в качестве энергии состояния сети можно выбрать функционал

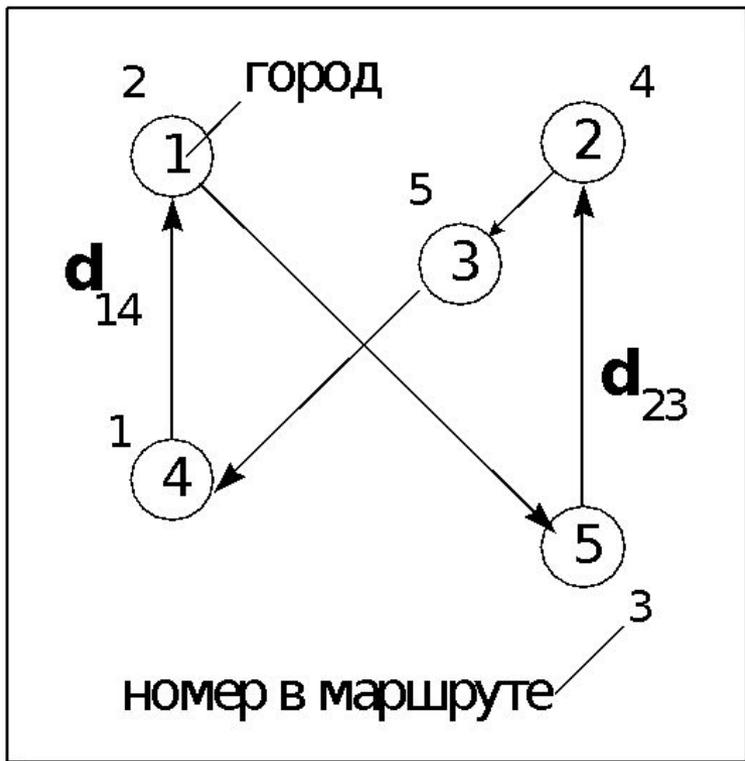
$$E = \frac{1}{2} \sum_{i,j,\alpha}^{i \neq j} d_{ij} v_{i\alpha} (v_{j\alpha-1} + v_{j\alpha+1}) +$$

$$+ \frac{\gamma}{2} \left[\sum_{\alpha} \left(\sum_i v_{i\alpha} - 1 \right)^2 + \sum_i \left(\sum_{\alpha} v_{i\alpha} - 1 \right)^2 \right]$$

где γ регулирует строгость соблюдения дополнительных условий в конечном решении.

Решению задачи будет соответствовать стационарное состояние сети, в котором лишь N нейронов сети будут активными ($v_{i\alpha} = 1$) и в каждом столбце и в каждой строке матрицы $\|v_{i\alpha}\|$ будет находиться один и только один единичный элемент.

Порядок прохождения городов



Нейронная кодировка маршрута

Diagram illustrating the neural encoding of the route. The matrix shows the presence (1) or absence (0) of a city at a specific step in the route.

1	0	1	0	0	0
2	0	0	0	1	0
3	0	0	0	0	1
4	1	0	0	0	0
5	0	0	1	0	0
	1	2	3	4	5

Labels: "город" (city) and "номер в маршруте" (number in route).

Величина γ регулирует “торг” между поиском маршрута минимальной протяженности и осмысленностью вида самого маршрута. Частное решение, соответствующее локальному минимуму функционала E , может быть осмысленным (второе слагаемое обращается на нем в ноль), но первое слагаемое (длина маршрута) для него, возможно будет слишком велико. Наоборот, длина маршрута может быть достаточно мала, но одно из оставшихся слагаемых будет ненулевым и маршрут окажется не интерпретируемым или недостаточен (например, проходит не через все города).

После того, как минимизируемая целевая функция для задачи коммивояжера построена, можно определить, какие связи в нейронной сети Хопфилда следует выбрать, так чтобы функционал энергии состояния в ней совпал с этой функцией. Для этого достаточно приравнять выражение для $E(v)$ к энергии рекуррентной сети:

Таким образом находятся значения синаптических связей в сети:

$$w_{i\alpha j\beta} = -d_{ij}(\delta_{\alpha-1\beta} + \delta_{\alpha+1\beta}) - \gamma\delta_{\alpha\beta} - \gamma\delta_{ij}$$

и значений порогов нейронов $\vartheta_{i\alpha} = -\gamma$. Общее число весов в сети - порядка N^3 .