2 слайд.

Это достаточно мощный, наиболее ранний и широко-распространённый инструмент биоинформатики.

База приложений нейронных сетей огромна: выявление фальшивых кредитных карточек, прогнозирование изменений на фондовой бирже, составление кредитных планов, оптическое распознавание символов, профилактика и диагностика заболеваний человека, наблюдение за техническим состоянием машин и механизмов, автоматическое управление движением автомобиля, принятие решений при посадке повреждённого летательного аппарата и т.д. Дальнейшие успехи в разработке искусственных нейронных сетей будут зависеть от более глубокого понимания принципов работы мозга, но здесь имеется и обратная связь: искусственные нейронные сети являются одним из средств, с помощью которых совершенствуется наше представление о процессах, происходящих в нервной системе человека, выступая в качестве моделей соответствующих процессов.

Нейронная сеть, на сегодняшний момент, эта та область знаний, о которой должны иметь определённое представление все научные специалисты, работающие в области компьютерных технологий.

2 слайд.

Искусственные нейронные сети основаны на биологической модели нервной системы. Нервная система человека, построенная из элементов, называемых нейронами, имеет ошеломляющую сложность. Около 10^{11} нейронов участвуют в 10^{15} передающих связях, имеющих длину метр и более. Уникальной способностью нейрона является приём, обработка и передача электрохимических сигналов по нервным путям.

Интенсивность сигнала, получаемого нейроном (следовательно, и возможность его активации), сильно зависит от активности линии связи. Каждая линия связи имеет определенную протяженность, и специальные химические вещества передают сигнал вдоль неё. Предполагается, что обучение заключается в первую очередь в изменении интенсивности связей.

Мозг, построенный из очень большого числа нейронов (каждый из которых принимает взвешенную сумму входных сигналов и при определённых условиях имеет возможность передавать сигнал дальше), способен решать чрезвычайно сложные задачи. Такую же модель реализуют и искусственные нейронные сети.

Теория нейронных сетей возникла из исследований в области искусственного интеллекта, а именно из попыток воспроизвести способность нервных биологических систем обучаться и исправлять ошибки, моделируя низкоуровневую структуру мозга.

Сеть “учиться” выполнять задачу, а не программируется непосредственно.

3 слайд.

Определение. Нейронная сеть - есть ориентированный мультиграф со взвешенными взаимосвязями.

Нейронная сеть является совокупностью элементов, соединенных некоторым образом так, чтобы между ними обеспечивалось взаимодействие. Эти элементы, называемые также нейронами (по аналогии с биологическими нейронами), представляют собой простые процессы, вычислительные возможности которых обычно ограничиваются некоторым правилом комбинирования входных сигналов и правилом активации, позволяющим вычислить выходной сигнал по совокупности входных сигналов.

Нейроны первого слоя образуют слой входных нейронов, он предназначен для получения сигналов из внешней среды. Так же имеется выходной слой, для вывода во внешнюю среду результатов вычислений; слои скрытых нейронов, значения которых задействованы только во внутренней кухне “сети”, они не получают данных от внешней среды непосредственно и не посылают данные непосредственно во внешнюю среду.

Выходной сигнал нейрона посылается другим элементам по взвешенным связям. Матрица весов, приписанных соответствующим связям между слоями, называется синапсом. В зависимости от значений весовых коэффициентов передаваемый сигнал или усиливается, или подавляется. Отрицательное значение веса соответствует подавлению активности элемента, а положительное значение – усилению его активности. Весовые коэффициенты являются параметрами сети, и обучение нейронной сети заключается в изменении параметров сети. Обученная сеть, сеть с фиксированным набором параметров отражает знание об эксперименте.

4 слайд.

Несмотря на то, что отдельные элементы нейронной сети имеют очень ограниченные вычислительные возможности, вся сеть в целом, объединяя большое число таких элементов, оказывается способной выполнять довольно сложные задачи. На рисунке представлена схема применения нейронной сети для контроля технического состояния авиалайнера. Входные элементы получают информацию непосредственно от датчиков, установленных на борту самолёта. Выходной элемент является индикатором технического состояния летательного аппарата.

Структура связей отражает детали конструкции сети, а именно то, какие элементы соединены, в каком направлении работают соединения, и каков уровень значимости (т.е. вес) каждого из соединений. В одной модели (т.е. для одного типа сетей) каждый элемент может быть связан со всеми другими элементами сети, в другой модели элементы могут быть организованы в некоторой упорядоченной по уровням (слоям) иерархии, где связи допускаются только между элементами в смежных слоях, а в третьей - могут допускаться обратные связи между смежными слоями или внутри одного слоя, или же допускается посылка сигналов элементами самим себе.

5 слайд.

Нейрон - функциональная единица нейронной сети.

Нейрон устроен таким образом, что сначала на сумматор поступает вектор входных сигналов с соответствующими весами, далее к линейной комбинации применяется так называемая функция активации, значение которой является выходным сигналом нейрона. В данном случае функция активации является пороговой, т.е. она ограничивает активность нейрона значениями 1 или 0 в зависимости от значения комбинированного ввода в сравнении с некоторой пороговой величиной w_0 (называемым смещением или сдвигом). Если искусственно ввести входное значение $x_0=1$, то значение нейрона можно выразить в эквивалентной векторной форме.

6 слайд.

Основные парадигмы обучения нейронных сетей, изменения весовых коэффициентов:

Обучение с учителем (supervised learning). Программа обучается получать желаемый результат. Эта парадигма применяется для таких моделей нейронных сетей как: однопоточковые многослойные сети (алгоритм обратного распространения ошибки, Backpropagation), сети с временной задержкой, рекуррентные сети.

Обучения без учителя (unsupervised learning). Программа обучается в условиях, когда окончательный результат априори не известен. Эта парадигма применяется для самоорганизующихся карт Кохонена, анализа главных компонент.

Пошаговое обучение (reinforcement learning), когда обучение осуществляется в процессе последовательного предъявления новых объектов.

7 слайд.

Проблемы бывают линейными и нелинейными.

Данные называются линейно-отделимыми, если существует гиперплоскость, однозначно разделяющая их на два класса. Если же для решения проблемы разделения образцов на классы требуется несколько гиперплоскостей, то проблема называется нелинейной.

Широко известным примером нелинейной проблемы является проблема моделирования отношения XOR. Отношение XOR дает 1 только тогда, когда в точности одно из вводимых значений равно 1, иначе вывод оказывается равным 0. Данная проблема сводится к линейной, если к двум имеющимся вводимым признакам добавить третий, и сделать пространство данных трёхмерным (в результате два класса будут размещаться в двух противоположных вершинах куба). Однако такое решение не стоит считать нейронным, т.к. поскольку добавление третьего признака означает, что мы вмешиваемся в процесс принятия решения, тем более, что для ряда сложных проблем такое вмешательство оказывается невозможным.

8 слайд.

Обучение простейшей сети, состоящей из одного нейрона, может происходить в соответствии с правилом Видроу-Хоффа (или дельта-правилом).

Значение целевой функции для некоторого примера: t .

Выходное значение нейрона: o .

Инициализировать веса случайным образом.

Итеративно исправлять веса после каждого акта применения сети к данным:

$$\Delta w_i = \eta (t - o) x_i \quad \text{правило Видроу-Хоффа или}$$

$$w_j \leftarrow w_j + \Delta w_j \quad \text{дельта-правило}$$

η - норма обучения, параметр, который влияет на скорость сходимости правила.

Для того, чтобы убедиться, что правило работает, выберем некоторые крайние точки значений целевой функции и нейрона. Если $t=o$ для некоторого входного вектора, то веса сети не меняются, результат достигнут. Если же $t = +1$, а $o = -1$, то $\Delta w_i = \eta 2 x_i$.

Вспомним, что т.к. $o = -1$, то $\vec{w} \cdot \vec{x} \leq 0$. Предположим, что $x_i > 0$, тогда увеличение весов сети приведёт к тому, что значение сети будет стремиться к значению функции.

В предположении линейной отделимости правило Видроу-Хоффа сходится к искомому решению за конечное число итераций.

9 слайд.

Для решение нелинейных проблем требуются многослойные нейронные сети, с нелинейной функцией активности.

10 слайд.

Для описания сложной топологии решающего пространства необходимо использовать нелинейные функции активности, непрерывные, дифференцируемые, монотонно возрастающие. Этим требованиям удовлетворяет так называемая функция “сигмоид”, которая, в общем виде, определяется следующим образом:

$$\sigma(x) = \frac{1}{1 + \exp\{-ax\}}$$

где a - это параметр, отвечающий за наклон кривой, большему значению этого параметра соответствует более крутой изгиб сигмоида, при $a \rightarrow \infty$ сигмоидная функция стремится к пороговой.

Для производной функции типа сигмоид справедливо следующее соотношение:

$$\frac{d\sigma}{dx} = \sigma(x)(1 - \sigma(x)).$$

11 слайд.

Это правило преодолевает предположение о линейной разделимости обучающих примеров в том смысле, что даже если это предположение не выполнено, правило позволяет обнаружить гиперплоскость, наилучшим образом разделяющую примеры. Это правило является краеугольным камнем алгоритма обратного распространения ошибки.

Рассмотрим простейшую сеть с тождественной функцией активации:

$$o = w_0 + w_1x_1 + \dots + w_nx_n$$

Введём функционал ошибки на пространстве весов $\vec{w}$:

$$E(\vec{w}) = 1/2 \sum_{d \in D} (t_d - o_d)^2,$$

$d \in D$ - обучающие примеры;

t_d - значение целевой функции для примера d ;

o_d - выходное значение нейронной сети для примера d ;

После введения функционала ошибки целесообразно менять веса сети таким образом, чтобы ошибка уменьшалась после каждой эпохи обучения (после того, как для всех примеров вычислено значение сети).

Функционал ошибки задает параболическую поверхность, имеющую единственный глобальный минимум.

Перед началом обучения весовые коэффициенты устанавливаются равными некоторым случайным значениям. При этом точка, представляющая начальное состояние сети, может оказаться в любом месте на поверхности ошибок, но очень маловероятно, чтобы она оказалась в точке минимума этой поверхности. В процессе обучения сеть должна корректировать весовые коэффициенты в так, чтобы максимально уменьшить значение общей ошибки. Другими словами, весовые коэффициенты должны корректироваться в том направлении, в котором спуск вниз по поверхности происходит быстрее всего. Это направление задается вектором, обратным вектору градиенту. Таким образом, умножая на норму обучения, приходим к правилу:

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

12 слайд.

Выведем правило вычисления весов в явном виде:

$$\begin{aligned} \frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i} \frac{1}{2} \sum_d (t_d - o_d)^2 \\ &= \frac{1}{2} \sum_d \frac{\partial}{\partial w_i} (t_d - o_d)^2 \\ &= \frac{1}{2} \sum_d 2(t_d - o_d) \frac{\partial}{\partial w_i} (t_d - o_d) \\ &= \sum_d (t_d - o_d) \frac{\partial}{\partial w_i} (t_d - \vec{w} \cdot \vec{x}_d) \\ \frac{\partial E}{\partial w_i} &= \sum_d (t_d - o_d) (-x_{i,d}) \end{aligned}$$

Обучающий пример $x_d = (x_{1,d}, \dots, x_{i,d}, \dots, x_{n,d})$.

Окончательно, $\Delta w_i = \eta \sum_{d \in D} (t_d - o_d) x_{i,d}$.

Недостатки метода градиентного спуска:

- сходимость протекает с большими временными затратами;
- если функционал ошибки определяет поверхность, которая не исключает локальных минимумов, то есть вероятность закориться локально, и не найти глобального минимума.

13 слайд.

Эти недостатки преодолевает описанное выше дельта-правило, которое в данном случае принимает вид:

$$E_d[\vec{w}] \equiv \frac{1}{2}(t_d - o_d)^2$$

При этом изменение вектора весов происходит после каждого акта применения сети к обучающему примеру.

Правило корректировки весов сети принимает вид:

$$\Delta w_i = \eta(t - o)x_i$$

14 слайд.

Алгоритм обратного распространения определяет два потока в сети: прямой поток от входного слоя к выходному и обратный поток – от выходного слоя к входному. Прямой поток продвигает входные векторы через сеть, в результате чего в выходном слое получаются выходные значения сети. Обратный поток подобен прямому, но он продвигает назад по сети значения ошибок, в результате чего определяются величины, в соответствии с которыми следует корректировать весовые коэффициенты в процессе обучения. В обратном потоке значения проходят по взвешенным связям в направлении, обратном направлению прямого потока, т.е. в обратном потоке элементы некоторого слоя получают сигналы ошибок от каждого элемента следующего слоя.

15 слайд.

Функционал ошибки в алгоритме Backpropagation задаётся следующим образом:

$$E_d(\vec{w}) = 1/2 \sum_{k \in \text{outputs}} (t_k - o_k)^2$$

Для вывода правила корректировки весов рассмотрим сеть с одним скрытым слоем, введём необходимые обозначения:

x_{ji} – i -тый вход на нейрон j ;

w_{ji} – вес, приписанный связи между i -тым и j -тым нейронами;

$net_j = \sum_i w_{ji} x_{ji}$ – взвешенная сумма значений, поступивших на вход нейрону j ;

o_j – значение выходного нейрона с номером j ;

t_j – значение целевой функции;

$outputs$ – множество нейронов выходного слоя

16 слайд.

Напомним, что для минимизации функционала ошибки весовые коэффициенты корректируются в направлении, обратном вектору градиенту, а именно:

$$\Delta w_{ij} = -\eta \frac{\partial E_d}{\partial w_{ji}}$$

В терминах введённых обозначений, по правилу вычисления производной сложной функции:

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} x_{ji}.$$

В случае, когда j – нейрон выходного слоя:

$$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j} \frac{\partial o_j}{\partial net_j}, \quad (*)$$

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j} \frac{1}{2} \sum_{k \in outputs} (t_k - o_k)^2$$

17 слайд.

Заметим, что

$$\frac{\partial}{\partial o_j} (t_k - o_k)^2 = 0, \quad k \neq j.$$

Тогда

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j} \frac{1}{2} (t_j - o_j)^2 = \frac{1}{2} 2(t_j - o_j) \frac{\partial (t_j - o_j)}{\partial o_j} = -(t_j - o_j).$$

Осталось вычислить ещё один множитель в уравнении (*). Для этого вспомним, что

$$o_j = \sigma(net_j),$$

и в соответствии с формулой для производной сигмоидальной функции (см. слайд 10)

$$\frac{\partial o_j}{\partial net_j} = \frac{\partial \sigma(net_j)}{\partial net_j} = o_j(1 - o_j).$$

Окончательно, уравнение (*) примет вид:

$$\frac{\partial E_d}{\partial net_j} = -(t_j - o_j) o_j (1 - o_j).$$

И правило исправления весов выходного слоя:

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} (t_j - o_j) o_j (1 - o_j) x_{ji}.$$

18 слайд.

Для вывода правила корректировки весов внутреннего слоя, введём величину: $Downstream(j)$ – множество нейронов, на вход которым поступает выходное значение нейрона j .

$$\begin{aligned} \frac{\partial E_d}{\partial net_j} &= \sum_{k \in Downstream(j)} \frac{\partial E_d}{\partial net_k} \frac{\partial net_k}{\partial net_j} = \sum_{k \in Downstream(j)} (-\delta_k) \frac{\partial net_k}{\partial net_j} = \sum_{k \in Downstream(j)} (-\delta_k) \frac{\partial net_k}{\partial o_j} \frac{\partial o_j}{\partial net_j} = \\ &= \sum_{k \in Downstream(j)} (-\delta_k) w_{kj} \frac{\partial o_j}{\partial net_j} = \sum_{k \in Downstream(j)} (-\delta_k) w_{kj} o_j (1 - o_j). \end{aligned}$$

Обозначив

$$\delta_j = -\frac{\partial E_d}{\partial net_j},$$

получаем

$$\delta_j = o_j (1 - o_j) \sum_{k \in Downstream(j)} \delta_k w_{kj}.$$

Правило исправления весов внутреннего слоя:

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$

Заметим, что правило исправления весов внутреннего слоя является частным случаем правила исправления весов внутреннего слоя, когда $Downstream(j) = outputs$.

19 слайд.

Итак, алгоритм Backpropagation заключается в том, что сети предъявляется образец и вычисляется вектор ошибок, в соответствии с которым исправляются веса внутренних и выходного слоя. Процесс повторяется для каждого образца. Полный цикл рассмотрения всех имеющихся обучающих данных называется эпохой. Все образцы подаются на рассмотрение снова и снова, эпоха за эпохой, пока на протяжении одной эпохи все выходные значения сети для не попадут в допустимые рамки.

К ограничениям алгоритма относят возможность “заякориться” в локальном минимуме.

В случае, если пространство ошибки имеет несколько локальных минимумов, есть опасность не найти глобального минимума.

20 слайд.

Успех применения нейронной сети обычно требует проведения множества экспериментов. Есть целый ряд параметров, от правильного выбора которых зависит, насколько быстро будет найдено решение.

Если сеть порождает правильный вывод для большинства вводимых образцов из набора тестовых данных, говорят, что сеть обладает хорошими свойствами обобщения. Предполагается, что набор тестовых данных в процессе обучения не использовался.

Если сеть хорошо обучена строить гладкое нелинейное отображение, то её возможности можно интерполировать на новые образцы, которые не использовались в процессе обучения. Негладкое отображение возникает тогда, когда сеть перетренирована. В такой ситуации сеть скорее будет подобна памяти.

Качество обобщения сети зависит от набора учебных данных и архитектуры сети. Набор учебных данных должен быть характерным для решаемой проблемы, но важным также является и число скрытых элементов. Если скрытых элементов больше, чем требуется для обучения нужному отношению ввода-вывода, будет больше весовых коэффициентов (так называемых, степеней свободы), чем это необходимо, и, если процесс обучения продолжается слишком долго, в результате может наблюдаться слишком близкое отслеживание данных (явление переобучения сети).

На рисунке представлена зависимость ошибки предсказания (на валидационном тесте) от количества эпох обучения. Валидационный тест - это та часть данных, которая резервируется для тестирования сети после каждой эпохи обучения. Из рисунка видно, что ошибка, вычисленная по обучающим данным, меньше, чем ошибка по контрольным данным. Более того, с увеличением числа эпох обучения ошибка на контрольных данных начинает расти, а по обучающим данным продолжает падать, сеть запоминает обучающие образцы. Итак, анализ ошибки на валидационном тесте позволяет определить оптимальное время обучения сети.

Относительно числа скрытых элементов и слоёв имеются некоторые общие рекомендации. Рекомендуются выполнение следующего неравенства:

$$N > \frac{W}{\varepsilon},$$

N - размер обучающих данных;

W - число весовых коэффициентов;

ε - доля ошибок.

21 слайд.

Априорная информация об эксперименте встраивается в модель нейронной сети за счёт выбора самого типа сети (парадигмы обучения), а так же выбора топологии, т.е. количества скрытых слоёв, количества нейронов в слое, структуры связей.

До этого момента мы имело дело с полно связной сетью

Один из типов сетей – рекуррентные нейронные сети. Особенностью данной модели нейронных сетей является наличие рекуррентных связей, когда элемент посылает сигналы себе или другим элементам того же слоя или более низких слоёв. На рисунке представлена двухслойная нейронная сеть (слой входных данных не учитывается), где активность нейронов выходного слоя вновь подаётся на вход нейронов нулевого слоя. Фактически, допустимым является любой вариант обратной связи. Обратную связь, задаваемую рекуррентными связями, не следует путать с обратной связью ошибок при корректировке весов. Рекуррентная связь подаёт обратно значение активности, которое будет влиять на выход сети в течение следующих итераций.

Теорема. Рекуррентные сети “разворачиваются” до сетей с прямой связью за счёт дублирования рекуррентных связей. Т.е. Чтобы имитировать рекуррентную сеть её надо дублировать для двух шагов выполнения во времени.

Рекуррентная сеть призвана обнаруживать временные закономерности в данных.

22 слайд.

Различная конфигурация сети, её топология и алгоритмы обучения делают инструмент очень гибким, способным охватить широкий спектр задач биоинформатики:

- предсказание структуры и функции белков, и их классификация (Nakata *et al.*, 1995, Wood *et al.*, 2004)

- распознавание промоторов, кодирующих участков ДНК (Cai and Chen, 1995, Reese, 2001);

- обработка и анализ данных микроэрея: кластеризация профилей экспрессии с целью выявления биологически осмысленных групп генов (Sawa *et al.*, 2003), моделирование динамики экспрессии генов (Vohradsky, 2001), обнаружение временных закономерностей, предсказание состояния генной сети в следующий временной такт (Liang *et al.*, 1998), обнаружение классов в образцах различных тканей (Herrero *et al.*, 2001);

- нейронная сеть, в силу своей природы, служит хорошим средством для интеграции результатов работы нескольких методов.

Нейронные сети способны решать задачи с большей точностью, чем эквивалентные символьные подходы, например, решающие деревья. К преимуществам так же можно отнести устойчивость к шумам в данных.

Ограничения алгоритмов нейронных сетей: Результат обученной сети, представленный набором весов, трудно интерпретируем.

23 слайд.

В статье (JI Í VOHRADSKY, 2001) искусственные нейронные сети используются как модель динамики экспрессии генов.

В модели рассматриваются прямые и обратные связи между генами.

Весы сети отражают регуляторный эффект одного гена на экспрессию других генов.

Экспрессия генов моделируется одной сетью, а так же композицией сетей, обучаемых независимо для моделирования транскрипции и трансляции.

В статье обсуждаются методы вычисления параметров модели на основании экспериментальных данных.

Результаты, полученные при помощи модели, соотносятся с экспериментальными наблюдениями.

24 слайд.

Многие естественные процессы представляются сетью взаимодействующих элементов. Каждый узел сети символизирует один ген, выходное значение нейрона соответствует его уровню экспрессии. Наличие связи между узлами отражает факт влияния генов. На рисунке изображена полно-связная сеть, в действительности, конечно, каждый ген контролирует не все гены, а только какую-то подгруппу. Состояние сети в момент времени t_i определяется её состоянием в момент времени t_{i-1} .

25 слайд.

Эта модель может быть упрощена до модели, в которой ген имеет одно из двух состояний: ген “включен”, или ген “выключен”. Состояние каждого гена определяется набором Булевых правил. Рассмотрим простой пример, в котором продукт гена А контролирует экспрессию гена В, который, в свою очередь, инициирует экспрессию гена С. Ген С негативно контролирует экспрессию гена А, т.е. выступает в роли репрессора. Пример булевых правил, извлекаемых из сети:

$$A=B, A=\text{not } C;$$

$$B=A;$$

$$C=B$$

Когда булевы правила известны для всех членов модели, можно вычислить терминальное состояние модели, либо это некая точка покоя, когда состояние модели не меняется со временем, либо конечное число состояний.

Булевы сети не чувствительны к различному регуляторному воздействию генов, которое определяется их уровнем экспрессии, они просто фиксируют сам факт воздействия.

26 слайд.

Выход нейрона в момент времени $t + \Delta t$ определяется через уровни экспрессии в момент времени $t(y_j)$ и значениями весов w_{ij} , которые приписаны соответствующим связям. Регуляторный эффект генов, который испытывает на себе ген i :

$$g_i \approx \sum_j w_{ij} y_j.$$

Целью такого подхода является извлечение весовой матрицы из экспериментальных данных, и восстановление генной сети по весовым значениям. Сначала рассматривается, например, полносвязная сеть, и в процессе обучения некоторые связи обнуляются.

Рассмотренная в статье модель основана на предположении, что регуляторный эффект каждого гена может быть выражен нейронной сетью. Каждый слой сети отражает уровень экспрессии генов в момент времени t .

В соответствии с сигмоидной функцией нейронной сети, уровень экспрессии гена i пропорционален величине:

Регуляторный эффект:
$$g_i = \left\{ 1 + \exp \left[- \left(\sum_j w_{ij} y_j + b_i \right) \right] \right\}^{-1}$$

b_i – внешний параметр сети, параметр задержки ответа (или реакции). Большие по модулю значения b_i подавляют влияние факторов, вовлечённых в весовую матрицу.

27 слайд.

Действительный уровень экспрессии гена моделируется при помощи константы k_{Li} , которая обозначает максимальный уровень экспрессии данного гена.

Изменение уровня экспрессии гена i выражается через регуляторный вклад других генов ρ и эффект деградации:

$$dz_i/dt = \rho_i - x_i$$

Уровень экспрессии гена i :

$$\rho_i = k_{Li} g_i$$

Эффект деградации в соответствии с уравнением кинетики химической реакции первого порядка, определяется как:

$$x_i = k_{2i} z_i,$$

где k_2 есть константа деградации продукта гена с номером i .

Таким образом, модель имеет вид:

$$\frac{dz_i}{dt} = k_{1i} \frac{1}{1 + \exp[-(\sum_j w_{ij} y_j + b_i)]} - k_{2i} z_i$$

Последнее уравнение в точности описывает специальный класс нейронных сетей, рекуррентные нейронные сети. Т.е. обучение нейронной сети в соответствии с адаптированным алгоритмом обратного распространения ошибки Backpropagation эквивалентно подбору коэффициентов w_{ij} такого уравнения.

28 слайд.

Если значение параметра задержки реакции одинаково для всех членов системы, то экспрессия поддерживается на некотором постоянном уровне.

Рисунок а и с. Если реакция какой-то компоненты выше, т.е. параметр задержки реакции имеет меньшее значение, например, для гена А (на рисунке б), то после того как концентрация продукта гена С (который блокирует транскрипцию гена А) оказывается достаточно низкой, опять начинается транскрипция гена А и цикл повторяется снова.

29 слайд.

В этой модели процессы транскрипции и трансляции предлагается моделировать независимо. Тогда процесс экспрессии генов представляется двумя последовательно соединенными сетями.

В первом компартменте ген i транскрибируется в матричную РНК.

Белок i контролирует экспрессию других генов, и, в частности, самого себя.

30 слайд.

Извлечение архитектуры нейронных сетей из экспериментальных данных.

При условии достаточного количества экспериментальных данных, нейронная сеть способна качественно охарактеризовать биологическую систему. Значения весовой матрицы обученной нейронной сети отражают тип воздействия компонент, например, “подавление”, в случае отрицательного веса, и, наоборот, “активация”, в случае положительного. В частности, сеть обнулит те связи, которых в жизни нет.

Однако при задании полной системы связей нейронной сети, возникает сложная многопараметрическая система, требующая огромного количества экспериментальных данных для обучения.

Таким образом, для получения решения в виде нейронной сети, необходимо максимально использовать априорную информацию о модели, и отразить её, например, в выборе архитектуры сети, алгоритме обучения.

31 слайд.

Байесовское обучение.

- с каждым решающим правилом (гипотезой) ассоциируют вероятность, уровень доверия правила

- по сравнению с алгоритмами, которые элиминируют несостоятельную гипотезу (значения этой гипотезы не совпадают со значениями целевой функции на некотором подмножестве обучающих данных), Байесовское обучение приписывает ей маленькую вероятность, таким образом, учитывая её вклад.

- Байесовское обучение позволяет встраивать априорную вероятность о модели в виде начального распределения вероятностей

- позволяет статистически обосновать методы Machine Learning, которые явно не манипулируют вероятностями

32 слайд.

Сведения из теории вероятностей.

A, B – некоторые события

Вероятность пересечения этих событий:

$$P(A \wedge B) = P(A/B)P(B) = P(B/A)P(A)$$

Вероятность объединения:

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

Теорема полной вероятности:

$A_1, \dots, A_n$ - полная группа событий относительно события B :

$$1. \sum_{i=1}^n P(A_i) = 1$$

$$2. \bigcup_{i=1}^n A_i \supset B$$

Тогда справедливо равенство: $P(B) = \sum_{i=1}^n P(B/A_i)P(A_i)$ - формула полной вероятности.

33 слайд.

D – таблица данных

h – гипотеза относительно D

$$\text{Теорема Байеса: } P(h/D) = \frac{P(D/h)P(h)}{P(D)}$$

$P(h)$ - априорная вероятность гипотезы h ;

$P(D)$ - априорная вероятность возникновения данных D ;

$P(h/D)$ - вероятность гипотезы h при условии возникновения данных D (апостериорная вероятность гипотезы h);

$P(D/h)$ - вероятность возникновения данных D в условиях гипотезы h .

Определение. MAP-гипотезой (maximum a posteriori) называется такая гипотеза:

$$h_{MAP} = \arg \max_{h \in H} P(h/D) = \arg \max_{h \in H} \frac{P(D/h)P(h)}{P(D)} = \arg \max_{h \in H} P(D/h)P(h).$$

В условиях, когда недостаточно оснований, для того, чтобы предпочесть одну гипотезу другой, полагают:

$$P(h_i) = P(h_j),$$

всем гипотезам $h \in H$ приписывают одинаковую априорную вероятность, тогда;

$$h_{MAP} = h_{ML} = \arg \max_{h \in H} P(D/h)$$

h_{ML} - максимально правдоподобная гипотеза, maximum likelihood.

Оказывается, что уменьшение функционала ошибки при обучении нейронной сети, или поиск оптимального решающего дерева является поиском MAP-гипотезы в терминах вероятностей.

34 слайд.

Пусть имеется две гипотезы:

h_1 : *cancer* - у пациента имеется определённая форма рака;

h_2 : $\neg$ *cancer* - пациент здоров

В лаборатории при обработке анализа получают два вида результата, маркеры “+” и “-”.

Причём верный результат в лаборатории получают с некоторой вероятностью, а именно:

$P(+ / cancer) = 0.98$, тогда $P(- / cancer) = 0.02$

$P(+ / \neg cancer) = 0.03$, $P(- / \neg cancer) = 0.97$

Статистика по населению утверждает, что рак проявляется с вероятностью:

$P(cancer) = 0.008$, и, соответственно, $P(\neg cancer) = 0.992$.

Требуется установить, болен ли раком пациент, у которого результатом лабораторного теста является положительный маркер.

В данном случае априорные вероятности гипотез (о присутствии/отсутствии рака) не равны, поэтому воспользуемся формулой для обнаружения гипотезы, обладающей максимальной апостериорной вероятностью.

Вычислим необходимые произведения:

$P(+ / cancer)P(cancer) = 0.98 \cdot 0.008 = 0.0078$,

$P(+ / \neg cancer)P(\neg cancer) = 0.03 \cdot 0.992 = 0.0298$.

Итак, $h_{MAP} : \neg cancer$

35 слайд.

Наивный классификатор Байеса применяется к задачам, в которых обучающие примеры $x \in X$ представлены конъюнкцией значений признаков $\langle a_1, \dots, a_n \rangle$. Целевая функция приписывает некоторый класс $v_j \in V$ обучающему примеру $x \in X$.

Найдём класс, обладающий максимальной апостериорной вероятностью для некоторого поднабора значений признаков $\langle a_1, \dots, a_n \rangle$.

$$v_{MAP} = \arg \max_{v_j \in V} P(v_j / a_1, \dots, a_n) = \arg \max_{v_j \in V} \frac{P(a_1, \dots, a_n / v_j)P(v_j)}{P(a_1, \dots, a_n)} = \arg \max_{v_j \in V} P(a_1, \dots, a_n / v_j)P(v_j).$$

В данном произведении, множитель $P(v_j)$ легко оценивается из таблицы обучающих данных как часть примеров, принадлежащих классу v_j . Для оценки оставшегося множителя в произведении требуется большой объем обучающих данных (следует выбрать все примеры с таким сочетанием значений признаков, и оценить ту часть, которая соответствует классу v_j), поэтому наивный классификатор Байеса основан на упрощающем предположении о независимости признаков $a_1, \dots, a_n$. Это предположение позволяет переписать искомую вероятность в следующем виде:

$$P(a_1, \dots, a_n / v_j) = \prod_i P(a_i / v_j).$$

Тогда

$$v_{MAP} = v_{NB} = \arg \max_{v_j \in V} \prod_i P(v_j)P(a_i / v_j)$$

В случае, когда “наивное” предположение выполнено, этот подход сравним по точности с нейронными сетями, и решающими деревьями.

36 слайд.

Рассмотрим пример применения классификатора для таблицы обучающих данных, представленной на прошлой лекции.

В данной таблице представлены примеры концепции “дни, благоприятные для игры в теннис”. Каждый день описан четырьмя признаками: *Outlook, Temperature, Humidity, Wind*.

Для 14 наборов значений перечисленных признаков известно значение целевой функции *PlayTennis*.

37 слайд.

Допустим, требуется классифицировать новый, не содержащийся в таблице данных, пример:

$\langle Outlook = Sunny, Temperature = Cool, Humidity = High, Wind = Strong, PlayTennis = ? \rangle$

Вычисляем

$$v_{NB} = \arg \max_{v_j \in \{yes, no\}} \prod_i P(v_j)P(a_i / v_j) =$$

$$\arg \max_{v_j \in I'} P(v_j)P(Outl = Sunny / v_j)P(Temp = Cool / v_j)P(Hum = High / v_j)P(Wind = Strong / v_j)$$

В данной записи содержится 10 вероятностей, которые необходимо оценить из таблицы данных.

Для значений целевой функции:

$$P(PlayTennis = yes) = 9/14 = 0.64$$

$$P(PlayTennis = no) = 5/14 = 0.36$$

Вычислим условные вероятности для фиксированных значений признаков:

$$P(Wind = Strong / PlayTennis = yes) = 3/9 = 0.33$$

$$P(Wind = Strong / PlayTennis = no) = 3/5 = 0.60$$

Вероятности для других признаков вычисляются аналогичным образом.

Окончательно имеем;

$$P(yes) P(Sunny / yes) P(Cool / yes) P(High / yes) P(Strong / yes) = 0.0053$$

$$P(no) P(Sunny / no) P(Cool / no) P(High / no) P(Strong / no) = 0.0206.$$

Итак, классификатор Байеса склоняется к значению целевой концепции *PlayTennis = no*.

38 слайд.

Байесовские сети. Представляют собой направленный ациклический граф, который отражает распределение Байесовской вероятности, основанной на априорном знании модели.

Имеется набор признаков (*Storm, BusTourGroup, Campfire, Lightning, Thunder, ForestFire*), которые влияют друг на друга, этот факт обозначен стрелками (молния (*Lightning*) вызывает гром (*Thunder*)); признаки *Storm* и *BusTourGroup* являются предшественниками для признака *Campfire*. Каждый признак зависим от своих предшественников, и не зависим от признаков, которые непосредственно не являются его предшественниками.

В терминах вероятностей, дискретный признак *X* не зависим от признака *Y*, но зависим от *Z*, если:

$$(\forall x_i, y_j, z_k) P(X = x_i / Y = y_j, Z = z_k) = P(X = x_i / Z = z_k),$$

где x_i, y_j, z_k - значения признаков *X, Y* и *Z*.

С каждым признаком ассоциирована таблица условных вероятностей (параметры сети), как показано на рисунке. Например, значение в правом верхнем углу таблицы означает:

$$P(Campfire = true | Storm = false, BusTourGroup = false) = 0.2$$

Если все параметры сети зафиксированы, то можем найти совместное распределение значений $\langle y_1, \dots, y_n \rangle$ признаков $\langle Y_1, \dots, Y_n \rangle$:

$$P(y_1, \dots, y_n) = \prod_{i=1}^n P(y_i | Parents(Y_i)),$$

Parents(Y_i) – непосредственные предшественники признака Y_i .

Сети Байеса использовались для:

- моделирования сайтов сплайсинга последовательностей ДНК (Cai *et al.*, 2000);

- предсказания вторичной структуры белка (Schmidler *et al.*, 2000);
- симуляции генной сети и отслеживания её динамики (Husmeier, 2003).

Сети Байеса способны поддерживать данные с шумами, оперировать неполными данными, содержащими пробелы. Как и в случае решающих деревьев, и индуктивно-логических правил, такой способ интерпретации фактов как граф позволяет наиболее естественно проверить гипотезу типа “если, то”. В отличие от решающих деревьев, сети Байеса менее склонны к переобучению.

Ограничения: предполагается заданным некоторое априорное распределение Байесовской вероятности.

1. Mitchell, T. (1997). *Machine Learning*. New York: McGraw Hill.
2. Калан Р. (2003) Основные концепции нейронных сетей-М.: Изд. «Вильямс».
3. Vohradsky J. (2001) Neural network model of gene expression, *FASEB J.*, 15:3(846–854).
4. Schmidler S.C., Liu J.S., Brutlag D.L. (2000) Bayesian segmentation of protein secondary structure. *J.Comp. Biol.* 7: 233-248.
5. Husmeier D. (2003) Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks, *Bioinformatics*, 19(17), 2271–2282.
6. Cai D., Delcher A., Kao B., Kasif S. (2000) Modeling splice sites with Bayes networks. *Bioinformatics*, 16: 152-158.