Приложение № 2
к распоряжению РТУ МИРЭА
от__ _____№_____

Титульный лист материалов по дисциплине
(заполняется по каждому виду учебного материала)

ДИСЦИЛИНА Проектирование и обучение нейронных сетей
(полное наименование дисциплины без сокращений)

ИНСТИТУТ Информационые технологии

КАФЕДРА Вычислительная техника
полное наименование кафедры

ВИД УЧЕБНОГО МАТЕРИАЛА Лекция
(в соответствии с пп.1-11)

ПРЕПОДАВАТЕЛЬ Сорокин Алексей Борисович
(фамилия, имя, отчество)

СЕМЕСТР 7 семестр, 2023/2024
(указать семестр обучения, учебный год)

6. ЛЕКЦИЯ. Стохастические методы обучения нейронных сетей

Стохастические методы полезны как для обучения искусственных нейронных сетей, так и для получения выхода от уже обученной сети. Стохастические методы обучения приносят большую пользу, позволяя исключать локальные минимумы в процессе обучения. Но с ними также связан ряд проблем.

Использование обучения

Искусственная нейронная сеть обучается с помощью некоторого процесса, модифицирующего ее веса. Если обучение успешно, то предъявление сети множества входных сигналов приводит к появлению желаемого множества выходных сигналов. Имеется два класса обучающих методов: детерминистский и стохастический.

Детерминистский метод обучения шаг за шагом осуществляет процедуру коррекции весов сети, основанную на использовании их текущих значений, а также величин входов, фактических выходов и желаемых выходов. Обучение персептрона является примером подобного детерминистского метода.

Стохастические методы обучения выполняют псевдослучайные изменения величин весов, сохраняя те изменения, которые ведут к улучшениям. Чтобы показать это наглядно, рассмотрим рис. 1, на котором изображена типичная сеть, где нейроны соединены с помощью весов.

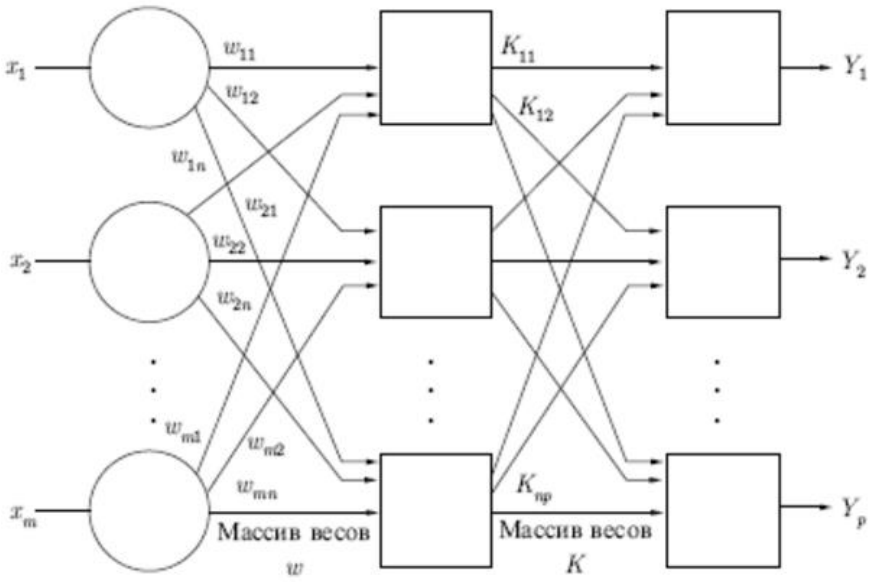

Рис. 1. Нейросеть

Выход нейрона является здесь взвешенной суммой его входов, которая преобразована с помощью нелинейной функции. Для обучения сети могут быть использованы следующие процедуры:

1. Выбрать вес случайным образом и подкорректировать его на небольшое случайное число. Предъявить множество входов и вычислить получающиеся выходы.

2. Сравнить эти выходы с желаемыми выходами и вычислить величину разности между ними. Общепринятый метод состоит в нахождении разности между фактическим и желаемым выходами для каждого элемента обучаемой пары, возведение разностей в квадрат и нахождение суммы этих квадратов. Целью обучения является минимизация этой разности, часто называемой целевой функцией.

3. Выбрать вес случайным образом и подкорректировать его на небольшое случайное значение. Если коррекция помогает (уменьшает целевую функцию), то сохранить ее, в противном случае вернуться к первоначальному значению веса.

Повторять шаги с 1 по 3 до тех пор, пока сеть не будет обучена в достаточной степени.

Этот процесс стремится минимизировать целевую функцию, но может попасть, как в ловушку, в неудачное решение. На рис. 2 показано, как это может происходить в системе с единственным весом.

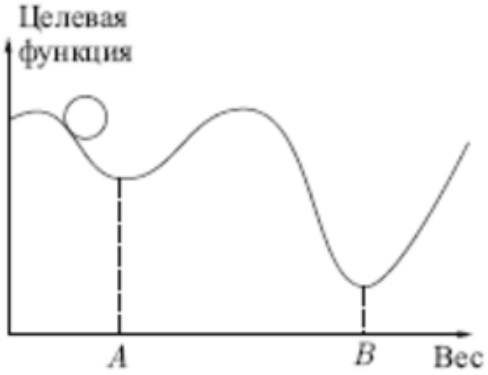

Рис. 2. Нейросеть

Допустим, что первоначально вес взят равным значению в точке A. Если случайные шаги по весу малы, то любые отклонения от точки A увеличивают целевую функцию и будут отвергнуты. Лучшее значение веса, принимаемое в точке B, никогда не будет найдено, и система будет поймана в ловушку локальным минимумом вместо глобального минимума в точке B. Если же случайные коррекции веса очень велики, то как точка A, так и точка B будут часто посещаться, но то же самое будет верно и для каждой другой точки. Вес будет меняться так резко, что он никогда не установится в желаемом минимуме.

Полезная стратегия для избежания подобных проблем состоит в больших начальных шагах и постепенном уменьшении размера среднего случайного

шага. Это позволяет сети вырываться из локальных минимумов и в то же время гарантирует окончательную стабилизацию сети.

Ловушки локальных минимумов досаждают всем алгоритмам обучения, основанным на поиске минимума (включая персептрон и сети обратного распространения), и представляют серьезную и широко распространенную трудность, которую почему-то часто игнорируют. Стохастические методы позволяют решить эту проблему. Стратегия коррекции весов, вынуждающая веса принимать значение глобального оптимума в точке B, вполне возможна.

В качестве объясняющей аналогии предположим, что на рис. 2 изображен шарик на поверхности внутри коробки. Если коробку сильно потрясти в горизонтальном направлении, то шарик будет быстро перекатываться от одного края к другому. Нигде не задерживаясь, в каждый момент времени шарик будет с равной вероятностью находиться в любой точке поверхности.

Если постепенно уменьшать силу встряхивания, то будет достигнуто условие, при котором шарик будет на короткое время «застревать» в точке B. При еще более слабом встряхивании шарик будет на короткое время останавливаться как в точке A, так и в точке B. При непрерывном уменьшении силы встряхивания будет достигнута критическая точка, когда сила встряхивания достаточна для перемещения шарика из точки A в точку B, но недостаточна для того, чтобы шарик мог «вскарабкаться» из B в A. Таким образом, окончательно шарик остановится в точке глобального минимума, когда амплитуда встряхивания уменьшится до нуля.

Искусственные нейронные сети могут обучаться, по существу, тем же способом при помощи случайной коррекции весов. Вначале делаются большие случайные коррекции с сохранением только тех изменений весов, которые уменьшают целевую функцию. Затем средний размер шага постепенно уменьшается, и глобальный минимум в конце концов достигается.

Эта процедура весьма напоминает отжиг металла, поэтому для ее описания часто используют термин «имитация отжига». В металле, который нагрет до температуры, превышающей его точку плавления, атомы находятся в сильном беспорядочном движении. Как и во всех физических системах, атомы стремятся к состоянию минимума энергии (единому кристаллу, в данном случае), но при высоких температурах энергия атомных движений препятствует этому. В процессе постепенного охлаждения металла возникают все более низкоэнергетические состояния, пока, в конце концов, не будет достигнуто самое малое из возможных состояний, глобальный минимум. В процессе отжига распределение энергетических уровней описывается следующим соотношением:

$$P(e) = exp(-e/kT), \quad (1)$$

где $P(e)$ — вероятность того, что система находится в состоянии с энергией e; k — постоянная Больцмана; T — температура по шкале Кельвина.

При высоких температурах $P(e)$ приближается к единице для всех энергетических состояний. Таким образом, высокоэнергетическое состояние почти столь же вероятно, как и низкоэнергетическое. По мере уменьшения температуры вероятность высокоэнергетических состояний уменьшается по отношению к низкоэнергетическим. При приближении температуры к нулю становится весьма маловероятным, чтобы система находилась в высокоэнергетическом состоянии.

Больцмановское обучение

Этот стохастический метод непосредственно применим к обучению искусственных нейронных сетей:

1. Определить переменную T, представляющую искусственную температуру. Придать T большое начальное значение.
2. Предъявить сети множество входов и вычислить выходы и целевую функцию.
3. Дать случайное изменение весу и пересчитать выход сети и изменение целевой функции в соответствии со сделанным изменением веса.
4. Если целевая функция уменьшилась (улучшилась), то сохранить изменение веса.

Если изменение веса приводит к увеличению целевой функции, то вероятность сохранения этого изменения вычисляется с помощью распределения Больцмана (1).

Выбирается случайное число r из равномерного распределения от нуля до единицы. Если $P(c)$ больше, чем r, то изменение сохраняется, в противном случае величина веса возвращается к предыдущему значению. Это позволяет системе делать случайный шаг в направлении, портящем целевую функцию, и дает ей тем самым возможность вырываться из локальных минимумов, где любой малый шаг увеличивает целевую функцию.

Для завершения больцмановского обучения повторяют шаги 3 и 4 для каждого из весов сети, постепенно уменьшая температуру T, пока не будет достигнуто допустимо низкое значение целевой функции. В этот момент предъявляется другой входной вектор, и процесс обучения повторяется. Сеть обучается на всех векторах обучающего множества, с возможным повторением, пока целевая функция не станет допустимой для всех них.

Величина случайного изменения веса на шаге 3 может определяться различными способами. Например, подобно тепловой системе, весовое изменение w может выбираться в соответствии с гауссовским распределением:

$$P(w) = exp(-w^2/T^2),$$

где $P(w)$ – вероятность изменения веса на величину w, T — искусственная температура.

Так как требуется величина изменения веса Δw, а не вероятность изменения веса, имеющего величину w, то метод Монте-Карло может быть использован следующим образом:

1. Найти кумулятивную вероятность, соответствующую $P(w)$. Это есть интеграл от $P(w)$ в пределах от 0 до w. Поскольку в данном случае $P(w)$ не может быть проинтегрирована аналитически, она должна интегрироваться численно, а результат необходимо затабулировать.
2. Выбрать случайное число из равномерного распределения на интервале $(0,1)$. Используя эту величину в качестве значения $P(w)$, найти в таблице соответствующее значение для величины изменения веса.

Свойства машины Больцмана широко изучены. Машина Больцмана представляет собой стохастическую машину, компонентами которой являются стохастические нейроны. Стохастический нейрон находится в одном из двух возможных вероятностных состояний. Этим двум состояниям формально можно присвоить значения $+1$ (соответствующее включенному состоянию) И -1 (соответствующее выключенному состоянию). Аналогично, можно принять значениями этих состояний $+1$ и 0 соответственно. Примем первое допущение. Еще одним отличительным свойством машины Больцмана является использование симметричных синоптических связей между нейронами. Использование этой формы синоптической связи обусловлено соглашениями статистической физики.

Стохастические нейроны машины Больцмана разбиваются на две функциональные группы: видимые и скрытые (рис. 3). Видимые нейроны предоставляют интерфейс между сетью и средой, в которой она работает. Во время этапа обучения сети, видимые нейроны фиксируются в своих специфичных состояниях, определяемых средой. С другой стороны, скрытые нейроны всегда работают свободно – они используются для выражения ограничений, содержащихся во входных векторах. Скрытые нейроны выполняют эту задачу с помощью извлечения статистических корреляций высокого порядка в ограничивающих векторах. Сеть, описанная выше, является частным случаем машины Больцмана. Ее можно рассматривать как процедуру

обучения без учителя моделированию распределения вероятности, которое применяется к видимым нейронам с соответствующими вероятностями. Таким образом, сеть может осуществлять дополнение образов. В частности, если вектор с неполной информацией поступает в подмножество видимых нейронов, сеть (в предположении правильности процедуры обучения) дополняет эту информацию в оставшихся видимых нейронах.

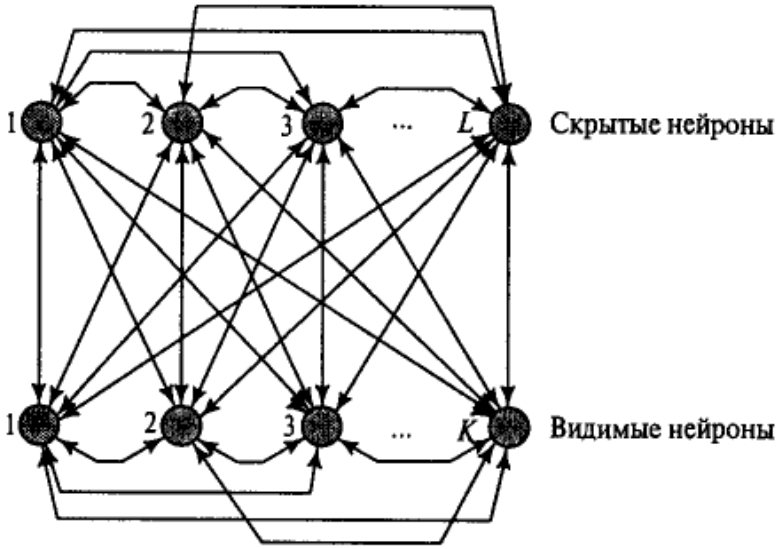

Рис. 3. Архитектурный граф машины Больцмана (где К – количество видимых, а L – количество скрытых нейронов)

Главной целью обучения Больцмана является создание нейронной сети, которая правильно моделирует входные образы в соответствии с распределением Больцмана. При использовании этой формы обучения делаются два предположения.

- Каждый входной вектор внешней среды подается на вход сети достаточно долго, чтобы система достигла температурного равновесия.
- Не существует определенной последовательности подачи векторов среды в видимые элементы сети.

Скорость уменьшения температуры должна быть обратно пропорциональна логарифму времени, чтобы была достигнута сходимость к глобальному минимуму. Скорость охлаждения в такой системе выражается следующим образом:

$$T(t) = \frac{T_0}{\log(1+t)}$$

где $T(t)$ – искусственная температура как функция времени; T_0 – начальная искусственная температура; t — искусственное время.

Этот разочаровывающий результат предсказывает очень медленную скорость охлаждения (и вычислений). Вывод подтвержден и экспериментально. Машины Больцмана часто требуют для обучения очень большого ресурса времени. Считается, что множество синоптических весов реализует совершенную модель структуры среды, если она приводит к точно такому же распределению вероятности состояний видимых элементов (при свободной работе сети), к какому приводит подача входных векторов среды. В общем случае, если количество скрытых нейронов не является экспоненциально большим по сравнению с количеством видимых элементов, такой совершенной модели достичь невозможно. Если же среда имеет упорядоченную структуру, а сеть использует скрытые элементы для извлечения этих закономерностей, можно достичь хорошего соответствия при достаточном количестве скрытых элементов.

Сигмоидальные сети доверия

Сигмоидальные сети доверия или логистические сети доверия (были разработаны в попытке найти стохастическую машину, которая была бы способна обучаться произвольному распределению вероятности на множестве двоичных векторов, но не имела бы, в отличие от машины Больцмана, потребности в отрицательной фазе. Эта цель была достигнута заменой симметричных связей в машине Больцмана прямыми соединениями, формирующими ацикличный граф. говоря более точно, сигмоидальные сети доверия имеют многослойную архитектуру, состоящую из двоичных стохастических нейронов (рис. 4).

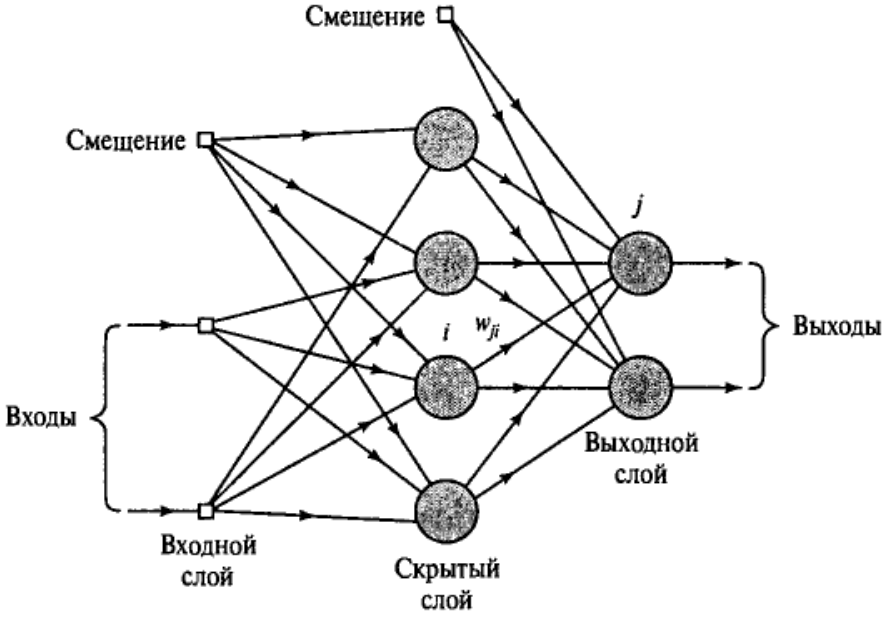

Рис. 4. Архитектурный граф сигмоидальной сети доверия

Ацикличная природа этих машин облегчает осуществление вероятностных вычислений. В частности, эта сеть использует сигмоидальную функцию, по аналогии с машиной Больцмана, для вычисления условной вероятности того, что нейрон будет активирован в ответ на свое собственное индуцированное локальное поле.

Обучение в сигмоидальных сетях доверия

Обозначим символом T множество примеров обучения, отобранных из интересующее нас распределения вероятности. Предполагается, что каждый из примеров является двоичным и представляет некоторый атрибут. При обучении допускаются повторения примеров с частотой, пропорциональной частоте встречи на практике подобной комбинации атрибутов. Для моделирования распределения, из которого отобрано множество T, выполним следующие действия.

1. Для сети определим размер вектора состояний x.

2. Выберем подмножество этого вектора (обозначенное x_α), представляющее атрибуты примеров обучения. Это значит, что x_α представляет собой вектор состояний видимых нейронов.

3. Оставшаяся часть вектора состояний (обозначенная x_β) определяет вектор состояний скрытых нейронов (т.е. расчетных узлов, для которых значения не устанавливаются).

Архитектура сигмоидальной сети доверия в значительной мере зависит от организации состояний видимых и скрытых элементов в векторе x. Таким образом, разные композиции состояний скрытых и видимых нейронов могут привести к разным конфигурациям.

Процедура обучения сигмоидальной сети доверия в сжатом виде представлена следующим образом;

Инициализация. Сеть инициализируется путем присвоения весам w_{ij} сети случайных значений, равномерно распределенных в диапазоне $[-a, a]$. Обычно значением a является число $0{,}5$.

1. Для данного множества примеров обучения T видимые нейроны сети фиксируются в состояниях x_αa, где $x_\alpha \in T$.

2. Для каждого x_α выполняется отдельное квантование Гиббса при не которой рабочей температуре T, после чего наблюдается полученный вектор состояний x всей сети. Предполагая, что моделирование проводится достаточно долго, значения x для разных классов, содержащихся в T, должны принять условное распределение соответствующего случайного вектора X, соответствующего данному множеству примеров.

3. Вычисляется среднее по множеству:

$$P_{ji} = \sum_{x_\alpha \in T} \sum_{x_\beta} P(X = x | X_\alpha = x_\alpha) x_j x_i \varphi(-x_j \sum_{i<j} w_{ji}\, x_i)$$

где случайный вектор X_α является подмножеством вектора X и $x = (x_\alpha, x_\beta)$. Векторы x_α и x_β соответствуют состояниям видимых и скрытых нейронов, x_j является j-м элементом вектора состояний x (т.е. состоянием нейрона j), а w_{ji} – синоптическим весом, направленным от нейрона i к нейрону j. Сигмоидальная функция $\varphi(v)$ определяется следующим образом:

$$\varphi(v) = \frac{1}{1 + \exp(-v)}$$

4. Каждый из синоптических весов подвергается коррекции на величину

$$\Delta w_{ji} = \eta \rho_{ji}$$

где η – параметр скорости обучения. Эта коррекция должна перемещать синаптические веса сети в направлении градиента в сторону локального максимума функции логарифмического правдоподобия $L(w)$ в соответствии с принципом максимального правдоподобия.

В алгоритме не учтено использование модели отжига. Именно поэтому температура T устанавливается в значение единицы. Тем не менее, как и в машине Больцмана, моделирование отжига в случае необходимости может быть внедрено в процедуру обучения сигмоидальной сети доверия для ускорения достижения точки термального равновесия.

В отличие от машины Больцмана для обучения сигмоидальной сети доверия требуется всего одна фаза. Причиной такого упрощения является то, что нормализация распределений вероятности по векторам состояния выполняется на локальном уровне каждого из нейронов с помощью сигмоидальной функции , $\varphi(v)$ а не глобально посредством сложного вычисления функции разбиения Z, при котором учитываются все возможные конфигурации состояний. Как только условное распределение вектора X для данных значений x_α из множества примеров обучения T было корректно промоделировано с помощью квантования Гиббса, роль отрицательной фазы процедуры обучения машины Больцмана выполняет весовой множитель $\varphi(-\frac{x_j}{T}\sum_{i<j} w_{ji}\, x_i)$ участвующий в вычислении усредненной по множеству корреляции $\mathrm{p_{ji}}$ между нейронами i и j. Когда достигается локальный минимум функции логарифмического правдоподобия, этот весовой множитель становится равным нулю, если сеть обучалась детерминированному отображению; в противном случае его усредняющий эффект сводится к нулю. Эти сети способны обучаться быстрее машин

Больцмана; такие преимущества сигмоидальных сетей доверия перед машиной Больцмана появились вследствие устранения из процедуры обучения отрицательной фазы.

Обучение Коши

В этом методе при вычислении величины шага распределение Больцмана заменяется на распределение Коши. Распределение Коши имеет, как показано на рис. 5, более длинные «хвосты», увеличивая тем самым вероятность больших шагов.

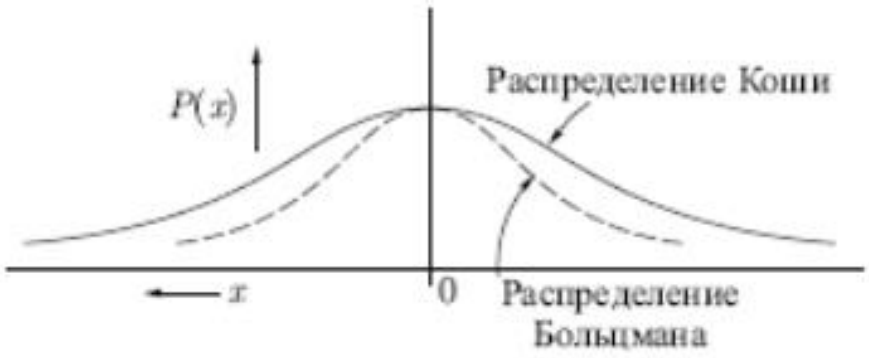

Рис. 5. Распределение Коши

В действительности, распределение Коши имеет бесконечную (неопределенную) дисперсию. С помощью такого простого изменения максимальная скорость уменьшения температуры становится обратно пропорциональной линейной величине, а не логарифму, как для алгоритма обучения Больцмана. Это резко уменьшает время обучения. Зависимость может быть выражена следующим образом:

$$T(t) = \frac{T_0}{1+t}$$

Распределение Коши имеет вид

$$P(x) = \frac{T(t)}{T(t)^2 + x^2}$$

где $P(x)$ есть вероятность шага величины x.

В данном уравнении $P(x)$ может быть проинтегрирована стандартными методами. Решая относительно x, получаем

$$x_c = \rho T(t) tg(P(x))$$

где ρ – коэффициент скорости обучения; x_c — изменение веса.

Теперь применение метода Монте-Карло становится очень простым. Для нахождения x в этом случае выбирается случайное число из равномерного распределения на открытом интервале $(-\pi/2, \pi/2)$ (необходимо ограничить функцию тангенса). Оно подставляется в формулу выше в качестве $P(x)$, и с помощью текущей температуры вычисляется величина шага.

Метод искусственной теплоемкости

Несмотря на улучшение, достигаемое с помощью метода Коши, время обучения может оказаться все еще слишком большим. Для дальнейшего ускорения этого процесса может быть использован способ, уходящий своими корнями в термодинамику. В этом методе скорость уменьшения температуры изменяется в соответствии с искусственной «теплоемкостью», вычисляемой в процессе обучения.

Во время отжига металла происходят фазовые переходы, связанные с дискретными изменениями уровней энергии. При каждом фазовом переходе может происходить резкое изменение величины, называемой теплоемкостью. Теплоемкость определяется как скорость изменения температуры в зависимости от изменения энергии. Изменения теплоемкости происходят из-за попадания системы в локальные энергетические минимумы.

Искусственные нейронные сети проходят аналогичные фазы в процессе обучения. На границе фазового перехода искусственная теплоемкость может скачкообразно измениться. Эта псевдотеплоемкость определяется как средняя скорость изменения температуры с целевой функцией. В примере шарика в коробке, приведенном выше, сильная начальная встряска делает среднюю величину целевой функции фактически не зависящей от малых изменений температуры, т. е. теплоемкость близка к константе. Аналогично, при очень низких температурах система замерзает в точке минимума, так что теплоемкость снова близка к константе. Ясно, что в каждой из этих областей допустимы сильные изменения температуры, так как не происходит улучшения целевой функции.

При критической температуре небольшое уменьшение ее значения приводит к большому изменению средней величины целевой функции. Возвращаясь к аналогии с шариком, при «температуре», когда шарик обладает достаточной средней энергией, чтобы перейти из A в B, но не достаточной для перехода из B в A, средняя величина целевой функции испытывает скачкообразное изменение. В этих критических точках алгоритм должен изменять температуру очень медленно, чтобы гарантировать, что система не "замерзнет" случайно в точке A, оказавшись пойманной в локальный минимум. Критическая температура может быть обнаружена по резкому уменьшению искусственной теплоемкости, т.е. средней скорости изменения температуры с целевой функцией. При достижении критической температуры скорость изменения температуры должна замедляться, чтобы гарантировать сходимость к глобальному минимуму. При всех остальных температурах может без риска

использоваться более высокая скорость снижения температуры, что приводит к значительному снижению времени обучения.

Обратное распространение и обучение Коши

Обратное распространение обладает преимуществом прямого поиска, т.е. веса всегда корректируются в направлении, минимизирующем функцию ошибки. Хотя время обучения и велико, оно существенно меньше, чем при случайном поиске, выполняемом машиной Коши, когда отыскивается глобальный минимум, но многие шаги выполняются в неверном направлении и "съедают" много времени.

Соединение этих двух методов дало хорошие результаты. Коррекция весов, равная сумме, вычисленной алгоритмом обратного распространения, и случайный шаг, задаваемый алгоритмом Коши, приводят к системе, которая сходится и находит глобальный минимум быстрее, чем система, обучаемая каждым из методов в отдельности. Простая эвристика используется для избежания паралича сети, который может возникнуть как при обратном распространении, так и при обучении по методу Коши.

Трудности, связанные с обратным распространением

Несмотря на богатые возможности, продемонстрированные методом обратного распространения, при его применении возникает ряд трудностей, часть из которых, однако, облегчается благодаря использованию нового алгоритма.

Сходимость. Д.Е.Румельхарт доказал сходимость на языке дифференциальных уравнений в частных производных. Таким образом, доказательство справедливо лишь в том случае, когда коррекция весов выполняется с помощью бесконечно малых шагов. Это условие ведет к бесконечному времени сходимости, и тем самым метод теряет силу в практических применениях. В действительности нет доказательства, что обратное распространение будет сходиться при конечном размере шага. Эксперименты показывают, что сети обычно обучаются, но время обучения велико и непредсказуемо.

Локальные минимумы. В обратном распространении для коррекции весов сети используется градиентный спуск, продвигающийся к минимуму в соответствии с локальным наклоном поверхности ошибки. Он хорошо работает в случае сильно изрезанных невыпуклых поверхностей, которые встречаются в практических задачах. В одних случаях локальный минимум является приемлемым решением, в других случаях он неприемлем.

Даже после того как сеть обучена, невозможно сказать, найден ли с помощью обратного распространения глобальный минимум. Если решение неудовлетворительно, приходится давать весам новые начальные случайные значения и повторно обучать сеть без гарантии, что обучение закончится на этой попытке или что глобальный минимум вообще будет когда-либо найден.

Паралич. При некоторых условиях сеть может при обучении попасть в такое состояние, когда модификация весов не ведет к действительным изменениям сети. Такой "паралич сети" является серьезной проблемой: один раз возникнув, он может увеличить время обучения на несколько порядков.

Паралич возникает, когда значительная часть нейронов получает веса достаточно большие, чтобы дать большие значения NET. В результате величина OUT приближается к своему предельному значению, а производная от сжимающей функции приближается к нулю. Как мы видели, алгоритм обратного распространения при вычислении величины изменения веса использует эту производную в формуле в качестве коэффициента. Для пораженных параличом нейронов близость производной к нулю приводит к тому, что изменение веса становится близким к нулю.

Если подобные условия возникают во многих нейронах сети, то обучение может замедлиться до почти полной остановки.

Нет теории, способной предсказывать, будет ли сеть парализована во время обучения или нет. Экспериментально установлено, что малые размеры шага реже приводят к параличу, но шаг, малый для одной задачи, может оказаться большим для другой. Цена же паралича может быть высокой. При моделировании многие часы машинного времени могут уйти на то, чтобы выйти из паралича.

Трудности с алгоритмом обучения Коши

Несмотря на улучшение скорости обучения, даваемое машиной Коши по сравнению с машиной Больцмана, время сходимости все еще может в 100 раз превышать время для алгоритма обратного распространения. Отметим, что сетевой паралич особенно опасен для алгоритма обучения Коши, в особенности для сети с нелинейностью типа логистической функции. Бесконечная дисперсия распределения Коши приводит к изменениям весов до неограниченных величин. Далее, большие изменения весов будут иногда приниматься даже в тех случаях, когда они неблагоприятны, часто приводя к сильному насыщению сетевых нейронов с вытекающим отсюда риском паралича.

Комбинирование обратного распространения с shape (формой) обучением Коши. Коррекция весов в комбинированном алгоритме, использующем обратное

распространение и обучение Коши, состоит из двух компонент: (1) направленной компоненты, вычисляемой с использованием алгоритма обратного распространения, и (2) случайной компоненты, определяемой распределением Коши. Эти компоненты вычисляются для каждого веса, и их сумма является величиной, на которую изменяется вес. Как и в алгоритме Коши, после вычисления изменения веса вычисляется целевая функция. Если происходит улучшение, изменение сохраняется безусловно. В противном случае, оно сохраняется с вероятностью, определяемой распределением Больцмана. Коррекция веса вычисляется с использованием представленных ранее уравнений для каждого из алгоритмов:

$$w_{mn,k}(n+1) = w_{mn,k}(n) + \eta\left[\alpha \Delta w_{mn,k}(n) + (1-\alpha)\delta_{n,k} OUT_{m,j}\right] + (1-\eta)x_c$$

где η — коэффициент, управляющий относительными величинами Коши и обратного распространения в компонентах весового шага.

Если η приравнивается нулю, система становится полностью машиной Коши. Если η приравнивается единице, система становится машиной обратного распространения. Изменение лишь одного весового коэффициента между вычислениями весовой функции неэффективно. Оказалось, что лучше сразу изменять все веса целого слоя, хотя для некоторых задач может стать выгоднее иная стратегия. Преодоление сетевого паралича комбинированным методом обучения. Как и в машине Коши, если изменение веса ухудшает целевую функцию, — с помощью распределения Больцмана решается, сохранить ли новое значение веса или восстановить предыдущее значение. Таким образом, имеется конечная вероятность того, что ухудшающее множество приращений весов будет сохранено. Так как распределение Коши имеет бесконечную дисперсию (диапазон изменения тангенса простирается от $-\infty$ до $+\infty$ на области определения), то весьма вероятно возникновение больших приращений весов, часто приводящих к сетевому параличу.

Очевидное решение, состоящее в ограничении диапазона изменения весовых шагов, ставит вопрос о математической корректности полученного таким образом алгоритма. На сегодняшний день доказана сходимость системы к глобальному минимуму лишь для исходного алгоритма. Подобного доказательства при искусственном ограничении размера шага не существует. В действительности экспериментально выявлены случаи, когда для реализации некоторой функции требуются большие веса и два больших веса, вычитаясь, дают малую разность.

Другое решение состоит в рандомизации весов тех нейронов, которые оказались в состоянии насыщения. Его недостаток в том, что оно может серьезно нарушить обучающий процесс, иногда затягивая его до бесконечности.

Для решения проблемы паралича был найден метод, не нарушающий достигнутого обучения. Насыщенные нейроны выявляются с помощью измерения их сигналов OUT. Когда величина OUT приближается к своему предельному значению, положительному или отрицательному, на веса, питающие этот нейрон, действует сжимающая функция. Она подобна используемой для получения нейронного сигнала OUT, за исключением того, что диапазоном ее изменения является интервал $(+5, -5)$ или другое подходящее множество. Тогда модифицированные весовые значения равны

$$w_{mn} = -5 + \frac{10}{1 + \exp(-w_{mn}/5)}$$

Эта функция заметно уменьшает величину очень больших весов, воздействие на малые веса значительно более слабое. Далее, она поддерживает симметрию, сохраняя небольшие различия между большими весами. Экспериментально было показано, что эта функция выводит нейроны из состояния насыщения без нарушения достигнутого в сети обучения. Не было затрачено серьезных усилий для оптимизации используемой функции, и другие значения констант могут оказаться лучшими.