Алгоритм взаимного обучения удаленных нейронных сетей

Виталий ГРИБАЧЕВ,
к. т. н.
reshebnik@rambler.ru

Приходилось ли вам когда-нибудь слышать, как два человека в вашем присутствии ведут беседу, предмет которой для вас совершенно непонятен, несмотря на то, что вы внимательно вслушиваетесь в разговор? Вы видите, как двигаются губы, слышите произносимые слова и даже понимаете их смысл, но вникнуть в суть беседы все равно не можете, так как одного понимания смысла слов в этом случае недостаточно. Еще бы! Чтобы понять смысл иносказательной беседы необходимо знать ее контекст.

Разумеется, нейросетевым технологиям в их нынешнем состоянии не под силу создать что-либо, хоть отдаленно напоминающее по сложности человеческий мозг, однако уже очень многие его функции вполне поддаются моделированию, хотя и в весьма упрощенном варианте. В том числе и прямая передача информации от одной нейронной сети другой в процессе взаимного обучения. Одна из первых статей, посвященных взаимному обучению нейронных сетей, была написана Вольфгангом Кинзелом и Идо Кантером в 2002 году [1].

Нейросетевая модель, которую они использовали, чтобы исследовать этот процесс, состояла из двух однослойных персептронов, включенных встречным образом, как показано на рис. 1.

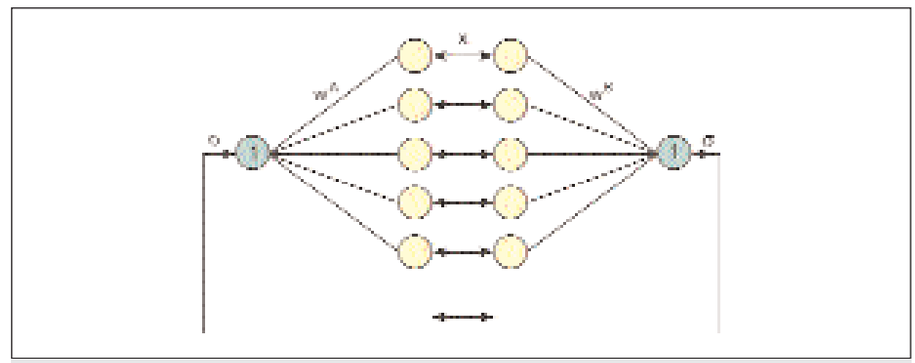

Рис. 1. Встречное включение персептронов

Рассуждения строились примерно таким образом. При классическом варианте — «с учителем» — нейронная сеть обучается на примерах, в качестве которых служит набор пар входных и выходных данных. Массив обучающих примеров в этом случае и является этим самым «учителем». Однако в качестве источника обучающих примеров можно взять что угодно, в том числе какой-нибудь регулятор управления или даже другую нейронную сеть. В данном эксперименте в качестве входного использовался случайный вектор x, подаваемый на обе сети одновременно, а выходные значения σ сети вычисляли самостоятельно. Выходные значения представляют собой единичные биты 0 или 1, получаемые с помощью активационной функции выходных нейронов.

$$\sigma = \mathrm{sign}(w \times x). \qquad (1)$$

В каждом цикле обучения сети обменивались значениями σ, и каждая из них настраивала свою матрицу коэффициентов по выходному значению противоположной сети. Коэффициенты однослойного персептрона настраиваются за один шаг и в данном случае модифицировались в соответствии с формулой:

$$w(t+1) = w(t)+(\eta/N)\times x\times\sigma\times\theta(\sigma), \qquad (2)$$

где $\theta(\sigma)$ — некоторая выбранная заранее функция; η — коэффициент, определяющий шаг обучения; N — общее количество весовых коэффициентов [2].

После некоторого числа циклов обучения коэффициенты сетей уравновешивались и становились одинаковыми по модулю, но противоположными по знаку. В данном эксперименте для весовых коэффициентов допускались отрицательные значения, и при этом на каждом шаге после настройки коэффициенты подвергались нормализации для приведения их к промежутку [0, 1]. Таким образом, в конце обучения весовые коэффициенты персептронов представляли собой зеркальное отражение друг друга.

Эксперименты показали принципиальную возможность взаимообучения нейронных сетей и высокую скорость такого процесса (в эксперименте однослойные сети с небольшим количеством весов уравновешивались приблизительно за 100 циклов, время синхронизации увеличивалось логарифмически с увеличением количества весовых коэффициентов N).

В данных экспериментах исследовалось взаимное обучение однослойных персептронов, выходные значения представляли собой единичные биты, а настройка весовых коэффициентов производилась за один шаг с помощью формулы (2). Однако использовать принципы взаимного обучения можно и для многослойных персептронов и вообще — для любых видов нейронных сетей, допускающих использование алгоритма обратного распространения ошибки. Для примера рассмотрим несколько шагов алгоритма для настройки пары выбранных заранее коэффициентов. Воспользуемся простой моделью взаимодействующих нейронных сетей, показанной на рис. 2.

Обязательные начальные условия:

1. Структура обеих сетей должна быть идентична.
2. Перед началом работы алгоритма взаимного обучения сети должны быть синхронизированы, то есть иметь одинаковые наборы весовых коэффициентов.
3. В процессе обучения все коэффициенты сетей, кроме пары симметричных, должны быть зафиксированы, то есть в процессе обучения меняется только пара коэффициентов, выбранная для настройки.

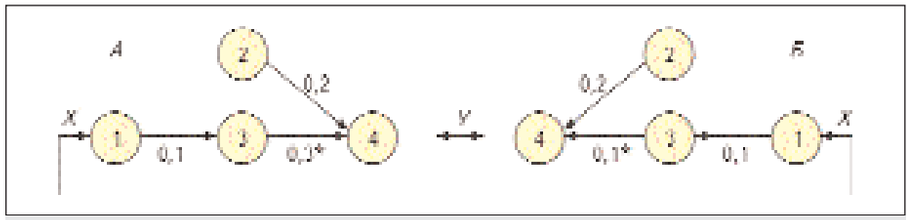

Рис. 2. Модель взаимообучения нейронных сетей

Коэффициенты, отмеченные звездочкой, подлежат настройке, остальные коэффициенты фиксированные. На входы сетей будут подаваться одинаковые случайные значения, а выходные значения сети будут пересылать друг другу с целью взаимообучения. В качестве обучающего используется классический алгоритм обратного распространения ошибки (ВРМ) [5]. Все вычисления округляются с точностью до третьего знака. Норма обучения принимается равной 1. Входные и выходные значения нейронов будем обозначать X и Y соответственно.

Рассмотрим несколько шагов данного алгоритма.

1. На вход сетей подается случайное значение 0,5.

Сеть *А* вычисляет:

$$X1 = 0{,}5;\ Y1 = 0{,}5;$$
$$X3 = 0{,}5\times0{,}1 = 0{,}05;$$
$$Y3 = 1/(1+\exp(-0{,}05)) = 0{,}513;$$
$$X4 = 0{,}2\times1+0{,}3\times0{,}513 = 0{,}354;$$
$$Y4 = 1/(1+\exp(-0{,}354)) = 0{,}588.$$

Сеть *Б* вычисляет:

$$X1 = 0{,}5;\ Y1 = 0{,}5;$$
$$X3 = 0{,}1\times0{,}5 = 0{,}05;\ Y3 = 0{,}513;$$
$$X4 = 0{,}513\times0{,}1+0{,}2\times1 = 0{,}251;$$
$$Y4 = 1/(1+\exp(-0{,}251)) = 0{,}563.$$

После обмена выходными значениями сети вычисляют свои ошибки обучения и корректируют выбранные коэффициенты:

$$\sigma_A = (0{,}563-0{,}588)\times$$
$$\times0{,}588\times(1-0{,}588) = -0{,}00061;$$
$$\Delta w_{43} = 1\times(-0{,}0061)\times0{,}513 = -0{,}0031;$$
$$w_{43} = 0{,}3-0{,}0031 = 0{,}297.$$

$$\sigma_Б = (0{,}588-0{,}563)\times$$
$$\times0{,}563\times(1-0{,}563) = 0{,}006;$$
$$\Delta w_{43} = 1\times0{,}006\times0{,}513 = 0{,}0031;$$
$$w_{43} = 0{,}1+0{,}0031 = 0{,}103.$$

2. На вход сетей подается случайное значение 0,25.

Сеть *А* вычисляет:

$$X1 = 0{,}25;\ Y1 = 0{,}25;$$
$$X3 = 0{,}25\times0{,}1 = 0{,}025;$$
$$Y3 = 1/(1+\exp(-0{,}025)) = 0{,}506;$$
$$X4 = 0{,}2\times1+0{,}506\times0{,}297 = 0{,}35;$$
$$Y4 = 1/(1+\exp(-0{,}35)) = 0{,}587.$$

Сеть *Б* вычисляет:

$$X1 = 0{,}25;\ Y1 = 0{,}25;$$
$$X3 = 0{,}1\times0{,}25 = 0{,}025;\ Y3 = 0{,}506;$$
$$X4 = 0{,}2\times1+0{,}103\times0{,}506 = 0{,}252;$$
$$Y4 = 1/(1+\exp(-0{,}252)) = 0{,}563.$$

Ошибки обучения сетей и коррекция выбранных коэффициентов:

$$\sigma_A = (0{,}563-0{,}587)\times$$
$$\times0{,}587\times(1-0{,}587) = -0{,}0006;$$
$$\Delta w_{43} = 1\times(-0{,}006)\times0{,}506 = -0{,}003;$$
$$w_{43} = 0{,}297-0{,}003 = 0{,}294.$$

$$\sigma_Б = (0{,}587-0{,}563)\times$$
$$\times0{,}563\times(1-0{,}563) = 0{,}006;$$
$$\Delta w_{43} = 1\times0{,}006\times0{,}506=0{,}003;$$
$$w_{43} = 0{,}103+0{,}003 = 0{,}106.$$

3. На вход сетей подается случайное значение 0,15

Сеть *А* вычисляет:

$$X1 = 0{,}15;\ Y1 = 0{,}15;$$
$$X3 = 0{,}15\times0{,}1 = 0{,}015;$$
$$Y3 = 1/\ (1+\exp(-0{,}015)) = 0{,}504;$$
$$X4 = 0{,}2\times1+0{,}294\times0{,}504 = 0{,}348;$$
$$Y4 = 1/(1+\exp(-0{,}348)) = 0{,}586.$$

Сеть *Б* вычисляет:

$$X1 = 0{,}15;\ Y1 = 0{,}15;$$
$$X3 = 0{,}1\times0{,}15 = 0{,}015;$$
$$Y3 = 1/(1+\exp(-0{,}015)) = 0{,}504;$$
$$X4 = 0{,}2\times1+0{,}106\times0{,}504 = 0{,}253;$$
$$Y4 = 1/(1+\exp(-0{,}253)) = 0{,}563.$$

Ошибки обучения сетей и коррекция коэффициентов:

$$\sigma_A = (0{,}563-0{,}586)\times$$
$$\times0{,}586\times(1-0{,}586) = -0{,}006;$$
$$\Delta w_{43} = 1\times(-0{,}006)\times0{,}504 = -0{,}003;$$
$$w_{43} = 0{,}294-0{,}003 = 0{,}291.$$

$$\sigma_Б = (0{,}586-0{,}563)\times$$
$$\times0{,}563\times(1-0{,}563) = 0{,}006;$$
$$\Delta w_{43} = 1\times0{,}006\times0{,}504 = 0{,}003;$$
$$w_{43} = 0{,}106+0{,}003 = 0{,}109.$$

Таким образом, можно заметить, что в процессе взаимного обучения сетей коэффициент сети *А* монотонно уменьшается, а коэффициент сети *Б* увеличивается. Коэффициенты в процессе настройки как бы движутся друг навстречу другу и уравниваются на некотором, заранее неизвестном значении, как показано на рис. 3.

Если повторить данный алгоритм настройки для всех коэффициентов сети, то в итоге сети будут иметь совершенно идентичные новые матрицы коэффициентов.

Применение алгоритма взаимного обучения

Во-первых, данную технологию можно применить при удаленной настройке нейронных сетей и передаче матрицы коэффициентов от одной нейронной сети к другой. Это будет эффективно в том случае, если открытая передача коэффициентов по каналу связи по каким-то причинам затруднена или нежелательна. Возможно также использование данного режима для гибкой удаленной подстройки отдельных коэффициентов в процессе работы нейронной сети.

Вторым перспективным направлением использования технологии, очевидно, является выработка или распределение секретных ключей. Действительно, в процессе выполнения алгоритма сами значения настраиваемых коэффициентов по каналу связи не передаются. Нейронные сети обмениваются только значениями ошибок обучения и случайными входными значениями, используемыми для вычисления выходов сетей. Между тем, по окончанию процесса настройки обе сети будут обладать одинаковыми наборами коэффициентов. Это означает, что выбранная пара коэффициентов или вся матрица целиком

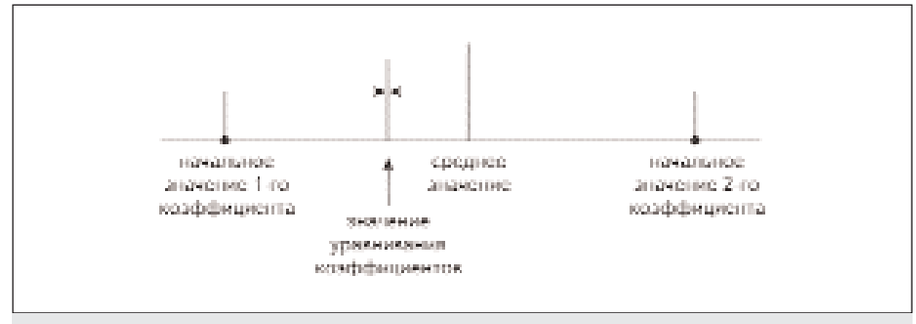

Рис. 3. Процесс настройки выбранной пары коэффициентов

могут быть использованы как материал для выработки секретного ключа, либо как сам секретный ключ. При этом активность злоумышленника, осуществляющего атаку типа «человек посередине», можно обнаружить по характеру процесса настройки коэффициентов. В случае активных действий злоумышленника процесс настройки коэффициентов перестает быть линейным и однонаправленным. На графике настройки появятся «всплески» и беспорядочные колебания величин коэффициентов. Особенно выраженными будут эти явления, когда параметры начальной синхронизации нейронных сетей для злоумышленника неизвестны. В этом случае злоумышленник не может осуществить начальную синхронизацию своей нейронной сети с нейронными сетями абонентов, выполняющих алгоритм взаимного обучения, и, следовательно, не может осуществить настройку матрицы коэффициентов своей сети и получить в итоге верный набор коэффициентов.

Сегодня единственным протоколом, позволяющим достоверно обнаружить активность злоумышленника, осуществляющего атаку типа «человек посередине», является квантовый протокол распределения ключей. Но для своей реализации этот протокол требует достаточно сложного оборудования в виде оптических преобразователей, интерферометров и высококачественной волоконно-оптической линии связи. Таким образом, схема выработки общего секретного ключа на основе алгоритма взаимного обучения нейронных сетей может стать аналогом квантового протокола для традиционных проводных или беспроводных линий связи.

Недостатки и ограничения

При всех очевидных достоинствах и простоте алгоритм взаимного обучения имеет также ряд весьма существенных недостатков. Прежде всего, это связано с особенностями технологической реализации самих нейронных сетей. Известно, что преимущества использования нейросетевых технологий наиболее значительно проявляются при их аппаратной реализации [3]. Однако в настоящее время аппаратные средства реализации все еще относительно дороги и имеют недостаточно широкое распространение.

Другим существенным недостатком данного алгоритма является медленная скорость взаимной настройки и значительное число шагов алгоритма, требуемое для настройки одной пары коэффициентов. Так, в эксперименте для настройки одной пары коэффициентов двух нейронных сетей, содержащих по 10 весовых коэффициентов и один скрытый слой, требовалось в среднем от 10 000 до 15 000 эпох [4]. Разумеется, с учетом быстродействия современных компьютеров в абсолютном временном выражении это совсем немного — считанные секунды, но, как мы видели выше, с увеличением количества весовых коэффициентов, этот промежуток времени быстро увеличивается (по логарифмическому закону).

Однако в данном случае имеются достаточно простые способы преодоления этого недостатка. Значительно ускорить процесс настройки коэффициентов может соглашение о способе коррекции коэффициентов путем выбора нового значения не из сплошного промежутка [0; 1], а из дискретного, например, с использованием таблицы. Еще лучше для настройки коэффициентов подойдет циклическое поле Галуа. Если заранее посчитать поле Галуа и в качестве настраиваемых коэффициентов на каждом шаге выбирать одну из координат точек поля Галуа, наиболее близкую к вычисленному с учетом ошибки обучения значению, то процесс настройки выбранной пары коэффициентов примет дискретный характер, и общее число шагов алгоритма взаимного обучения при этом способе настройки не превысит половины от числа точек применяемого поля Галуа. А число точек в используемом поле Галуа мы можем задать заранее.

Другим способом повышения скорости настройки коэффициентов является введение ограничения на разрядность используемых чисел, так как основные временные затраты в данном конкретном случае идут на осуществление вычислений с плавающей точкой.

Литература

1. Kanter, Kinzel W. Interacting neural networks and cryptography. **ht tp:/ /arxiv.o rg/abs/cond-mat/0203011 v1 (2002)**
2. Biehl M., Caticha N. Statistical Mechanics of On-line Learning and Generalization. The Handbook of Brain Theory and Neural Networks, ed. by M. A. Arbib. Berlin: MIT Press. 2001.
3. Грибачев В. П. Элементная база аппаратных реализаций нейронных сетей // Компоненты и технологии. 2006. № 8.
4. Грибачев В. П. Протокол ключевого обмена в криптосистемах, основанных на нейронных сетях. Сб. докладов международной конференции по мягким вычислениям и измерениям. Изд. СПбГТУ ЛЭТИ. 2004. Том 1.
5. **ht tp:/ /w ww.reshebnik.n et.r u/nn/BPM.ht ml** — описание алгоритма обратного распространения ошибки.