Основы обучения нейронных сетей

Также как и человеческий мозг, нейронная сеть (НС) способна к обучению. Аналогия с мозгом не заканчивается на структуре нейрона и их сети. Идея обучения НС также позаимствована из природы. Известно, что человеческий мозг способен к самообучению, причем достигает успехов зачастую, не зная природы процессов, лежащих в основе выполняемых действий. Однако непременным атрибутом обучения является многократное повторение и возможность немедленной оценки полученного результата. Обучение НС может проводиться в трех режимах: с учителем, без учителя и с подкреплением. Рассмотрим далее задачу в режиме обучения НС с учителем.

Цель обучения НС с учителем состоит в поиске (подборе) таких значений весов дуг $w(u)$ в структуре орграфа $G(X,U)$, чтобы при заданных входных сигналах *input(x)* для нейронов входного слоя $x \in X_{IN}$ получить выходные сигналы *output(x)* для нейронов выходного слоя $x \in X_{OUT}$, которые будут близки к ожидаемым значениям *output*(x)*.

Набор входных сигналов и правильных (ожидаемых) выходных сигналов сети называется **обучающим примером**. Для обучения НС может быть использован не один, а несколько обучающих примеров, совокупность которых принято называть **выборкой** обучающих примеров. Рассмотрим далее особенности решения задачи обучения НС на некотором обучающем примере.

Для поиска весов дуг НС в пространстве [-1,1] требуется **критерий**, который определяет, как правило, расстояние в той или иной метрике между полученным решением и ожидаемым (правильным) ответом:

$$E(w) \to \min, \tag{1}$$

где $E(w)$ – непрерывная и дифференцируемая функция многих переменных, характеризующая ошибку полученного решения.

Решение этой задачи сводится к построению убывающей последовательности $E^0(w), E^1(w),..., E^k(w),...$ где k – шаг обучения. При построении этой последовательности используются различные **правила остановки** процесса поиска. Например, можно задать фиксированное количество шагов обучения $k_{\max}$ или условие $\left|E^k(w) - E^{k-1}(w)\right| \le \varepsilon$, где ε – допустимая точность решения.

Для решения этой задачи используются **градиентные методы** оптимизации и их различные модификации. Суть этих методов сводится к уточнению (коррекции) весов дуг (x_i, x_j) на k – шаге обучения по следующим формулам:

$$w_{ij}{}^k = w_{ij}{}^{k-1} + \Delta w_{ij}{}^k, \tag{2}$$

$$\Delta w_{ij}{}^k = h^k p^k(w_{ij}), \tag{3}$$

где $p^k(w_{ij})$ – направление движения (вектор) в многомерном пространстве поиска, а h^k – шаг, который надо сделать в этом направлении. Для начала обучения следует задать веса дуг $w^0(u)$. Эти веса выбираются случайным образом на отрезке [-1,1] как ненулевые значения близкие к 0 или берутся из предыдущего сеанса обучения НС. Полагаем, что в начале обучения для каждой дуги сети $\Delta w^0(u) = 0$.

Из курса математики известно, что направление наискорейшего увеличения функции определяется вектором градиента $\frac{dE}{dw}$, отсюда и название этих методов. Т.к. в задаче требуется минимизация целевой функции (1), то двигаться надо в обратном направлении вектора градиента, т.е. в направлении антиградиента.

Обучение НС с применением градиентных методов требует определения антиградиента относительно всех слоев сети, что необходимо для правильного выбора направления $p^k(w_{ij})$. Эта задача имеет очевидное решение только для весов выходного слоя, где у нейронов нет исходящих дуг. Для нейронов других слоев используется метод обратного распространения ошибки, суть которого состоит в движении поиска от нейронов выходного слоя к нейронам других слоев в обратном порядке.

Метод обратного распространения ошибки в НС

Рассмотрим далее этот метод при условии, что функция активации нейронов будет сигмовидная $f(a)=\frac{1}{1+e^{-a}}$, для которой известно, что:

$$f'(a)=f(a)(1-f(a)). \tag{4}$$

Здесь a – входной сигнал нейрона x_j, который определяется по формуле:

$$a=\sum_{\forall x_i\in\Gamma^{-1}(x_j)} output(x_i)w(x_i,x_j), \tag{5}$$

В этом методе для критерия (1) используется функция следующего вида:

$$E=\frac{1}{2}\sum_{\forall x_j\in X_{OUT}}(output(x_j)-output^*(x_j))^2. \tag{6}$$

Функция (6) характеризует ошибку, полученную на нейронах выходного слоя, и определяется полусуммой расстояния между некоторым полученным решением и правильным ответом задачи в евклидовой метрике.

После дифференцирования функции (6) получим следующую формулу для определения градиента:

$$\frac{dE}{dw}=(output(x_j)-output^*(x_j))\frac{df}{da}\frac{da}{dw}, \tag{7}$$

а после подстановки формулы (4) и замены направления на противоположное получим следующую формулу для определения антиградиента:

$$(output^*(x_j)-output(x_j))output(x_j)(1-output(x_j))\frac{da}{dw}. \tag{8}$$

Для расчета вектора антиградиента **у нейронов выходного слоя**, у которых нет исходящих дуг (не зависит от w), введем следующую поправку (ошибка в выходном сигнале у данного нейрона):

$$\delta(x_j)=(output^*(x_j)-output(x_j))output(x_j)(1-output(x_j)). \tag{9}$$

У нейронов скрытого слоя, у которых есть исходящие дуги, учтем поправку на его выходной сигнал по ошибкам на нейронах следующего слоя:

$$(output^*(x_j)-output(x_j))=\sum_{\forall x_s\in\Gamma(x_j)} w_{js}\delta(x_s).$$

Тогда для расчета вектора антиградиента **у нейронов скрытого слоя** получим следующую формулу поправки:

$$\delta(x_j) = output(x_j)(1 - output(x_j)) \sum_{\forall x_s \in \Gamma(x_j)} w_{js} \delta(x_s) . \qquad (10)$$

Таким образом, вектор антиградиента (8), задающего направление поиска, определяется для каждого нейрона сети по следующей формуле:

$$\delta(x_j) \frac{da}{dw} , \qquad (11)$$

а после дифференцирования функции (5) получим следующую формулу для каждого компонента антиградиента на k –ом шаге обучения:

$$p^k(w_{ij}) = \delta(x_j) output(x_i) . \qquad (12)$$

Расчет приращение веса дуги (x_i, x_j) в этом методе использует модифицированную следующим образом формулу (3):

$$\Delta w_{ij}{}^k = \eta \times p^k(w_{ij}) + \alpha \times \Delta w_{ij}{}^{k-1}, \qquad (13)$$

где η и α – параметры алгоритма, которые задаются учителем (η – скорость обучения, α – момент обучения). Второе слагаемое в формуле (13) введено для сглаживания функции (6). Определение нового значения веса дуги $w_{ij}{}^k$ в результате обучения на k –ом шаге производится по формуле (2). Рассмотрим далее алгоритмическую реализацию этого метода.

Алгоритм обучения НС на основе метода обратного распространения ошибки

Пусть имеется НС, структура которой задана орграфом $G(X,U)$ с установленными начальными значениями весов дуг $w^0(u)$. Пусть имеется некоторый обучающий пример, включающий в себя:

- входные сигналы нейронов входного слоя $\forall x \in X_{IN}$: *input(x),*
- выходные сигналы нейронов выходного слоя $\forall x \in X_{OUT}$: *output*(x).*

Пусть учителем заданы следующие параметры алгоритма: η, α, $k_{\max}$. Тогда данный алгоритм будет сводиться к следующим действиям.

1. Положить: $k = 0$, $\forall u \in U : \Delta w^0(u) = 0$
2. Рассчитать $\forall x \in X \ : output^0(x)$ с помощью алгоритма последовательного распространения сигналов в НС
3. Найти оценку решения $E^0(w)$ по формуле (6)
4. **ЦИКЛ** по шагам обучения $k = 1, k_{\max}$

   4.1. **ЦИКЛ** по всем дугам $(x_i, x_j) \in U$ в обратном порядке

      4.1.1. **ЕСЛИ** $x_j \in X_{OUT}$, **ТО** рассчитать $\delta^k(x_j)$ по формуле (9), **ИНАЧЕ** рассчитать $\delta^k(x_j)$ по формуле (10)

4.1.2. Рассчитать $p^k(w_{ij})$ по формуле (12)

4.1.3. Рассчитать $\Delta w_{ij}^{\ k}$ по формуле (13)

4.1.4. Рассчитать $w_{ij}^{\ k}$ по формуле (2)

4.2. Рассчитать $\forall x \in X\ : output^k(x)$ с помощью алгоритма последовательного распространения сигналов в НС

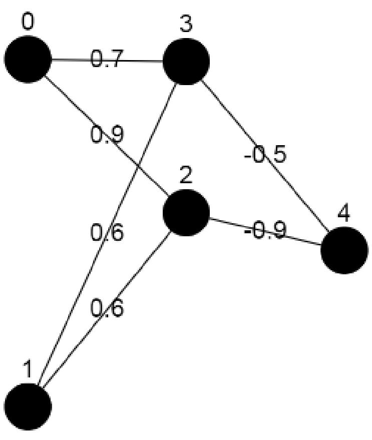
4.3. Найти оценку решения $E^k(w)$ по формуле (6)

Примечание:

1. Для проведения расчетов *output(x)* рекомендуется использовать встроенный калькулятор https://www.google.ru/.
2. Округление сигналов нейрона следует проводить до второго знака после запятой по следующему правилу. Если значение после третьего знака после запятой больше, чем 0.005, то округлить значение второго знака после запятой в большую сторону, если меньше или равно 0.005, то оставить второй знак после запятой без изменения. Например: $0.475 \approx 0.47$, $0.4785 \approx 0.48$, $0.4705 \approx 0.47$.
3. Округление оценок $E^0(w), E^1(w),\ldots$ следует проводить аналогично, но до четвертого знака после запятой.

ПРИМЕР

Обучить заданную нейронную сеть с сигмовидной функцией активации нейронов с помощью алгоритма обратного распространения ошибки при условии, что input(0)=0.9, input(1)=0.4, output*(4)=1, η=0.7, α=0.3, k_{max} =1.



РЕШЕНИЕ

1. Положим: $k = 0$, $\forall u \in U : \Delta w^0(u) = 0$
2. Рассчитаем $\forall x \in X\ : output^0(x)$ с помощью алгоритма последовательного распространения сигналов в НС.

2.1. Определим выходные сигналы для нейронов входного слоя:

$$output(x_0) = input(x_0) = 0.9$$

$$output(x_1) = input(x_1) = 0.4$$

2.2. Рассчитаем входной и выходной сигналы нейрона x_2 с прообразами $\Gamma^{-1}(x_2) = \{x_0, x_1\}$:

$$input(x_2) = 0.9 \times 0.9 + 0.4 \times 0.6 = 1.05$$

$$output(x_2) = f(input(x_2)) = 0.74077 \approx 0.74$$

2.3. Рассчитаем входной и выходной сигналы нейрона x_3 с прообразами $\Gamma^{-1}(x_3) = \{x_0, x_1\}$:

$$input(x_3) = 0.9 \times 0.7 + 0.4 \times 0.6 = 0.87$$

$$output(x_3) = f(input(x_3)) = 0.704745 \approx 0.7$$

2.4.Рассчитаем входной и выходной сигналы нейрона x_4 с прообразами $\Gamma^{-1}(x_4) = \{x_2, x_3\}$:

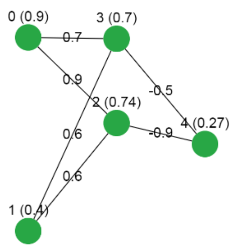
$$input(x_4) = 0.74 \times (-0.9) + 0.7 \times (-0.5) = -1.016 \approx -1.02$$

$$output(x_4) = f(input(x_4)) = 0.2658 \approx 0.27$$

3. Оценим полученное решение по формуле (6):

$$E^0(w) = \frac{1}{2} \sum_{\forall x_j \in X_{OUT}} (output(x_j) - output^*(x_j))^2 = 0.269512 \approx 0.2695$$

Результаты расчетов приведены ниже.



k = 0

+ -

X	Прообразы X	input(X)	output(X)
0	-	0.9	0.9
1	-	0.4	0.4
2	0,1	1.05	0.74
3	0,1	0.87	0.7
4	2,3	-1.02	0.27

$E^0(w)$: 0,2695

4. Положим шаг обучения $k = 1$

4.1. Определим для нейрона x_4 значение $\delta^k(x_j)$ по формуле (9):

$$\delta^1(x_4) = (1 - 0.27) \times 0.27 \times (1 - 0.27) = 0.143883 \approx 0.14$$

Рассчитаем для дуги (x_3, x_4):

$$p^1(w_{34}) = 0.14 \times 0.7 = 0.098 \approx 0.1$$

$\Delta w_{34}{}^{1} = 0.7 \times 0.1 + 0.3 \times 0 = 0.07$

$w_{34}{}^{1} = -0.5 + 0.07 = -0.43$

Рассчитаем для дуги (x_2, x_4):

$p^1(w_{24}) = 0.14 \times 0.74 = 0.1036 \approx 0.1$

$\Delta w_{24}{}^{1} = 0.7 \times 0.1 + 0.3 \times 0 = 0.07$

$w_{34}{}^{1} = -0.9 + 0.07 = -0.83$

4.2. Определим для нейрона x_3 значение $\delta^k(x_j)$ по формуле (10):

$\delta^1(x_3) = 0.7 \times (1 - 0.7) \times 0.14 \times (-0.5) = -0.0147 \approx -0.01$

Рассчитаем для дуги (x_1, x_3):

$p^1(w_{13}) = -0.01 \times 0.4 = -0.004 \approx 0$

$\Delta w_{13}{}^{1} = 0.7 \times 0 + 0.3 \times 0 = 0$

$w_{13}{}^{1} = 0.6 + 0 = 0.6$

Рассчитаем для дуги (x_0, x_3):

$p^1(w_{03}) = -0.01 \times 0.9 = -0.009 \approx -0.01$

$\Delta w_{03}{}^{1} = 0.7 \times (-0.01) + 0.3 \times 0 = -0.01$

$w_{03}{}^{1} = 0.7 - 0.01 = 0.69$

4.3.Определим для нейрона x_2 значение $\delta^k(x_j)$ по формуле (10):

$\delta^1(x_2) = 0.74 \times (1 - 0.74) \times 0.14 \times (-0.9) = -0.02424 \approx -0.02$

Рассчитаем для дуги (x_1, x_2):

$p^1(w_{12}) = -0.02 \times 0.4 = -0.008 \approx -0.01$

$\Delta w_{12}{}^{1} = 0.7 \times (-0.01) + 0.3 \times 0 = -0.007 \approx -0.01$

$w_{12}{}^{1} = 0.6 - 0.01 = 0.59$

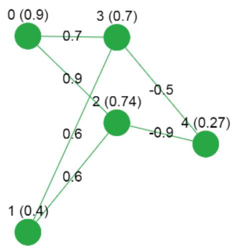
Рассчитаем для дуги (x_0, x_2):

$p^1(w_{02}) = -0.02 \times 0.9 = -0.018 \approx -0.02$

$\Delta w_{02}{}^{1} = 0.7 \times (-0.02) + 0.3 \times 0 = -0.014 \approx -0.01$

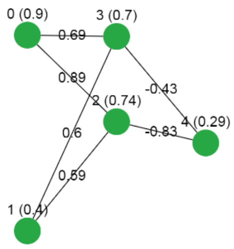
$w_{02}{}^{1} = 0.9 - 0.01 = 0.89$

Результаты обучения нейронной сети на шаге $k=1$ приведены ниже.



(x_i, x_j)	w_{ij}^0	Δw_{ij}^0	$\delta(x_j)$	$p^1(w_{ij})$	Δw_{ij}^1	w_{ij}^1
(3, 4)	-0.5	0	0.14	0.1	0.07	-0.43
(2, 4)	-0.9	0	0.14	0.1	0.07	-0.83
(1, 3)	0.6	0	-0.01	0	0	0.6
(0, 3)	0.7	0	-0.01	-0.01	-0.01	0.69
(1, 2)	0.6	0	-0.02	-0.01	-0.01	0.59
(0, 2)	0.9	0	-0.02	-0.02	-0.01	0.89

5. Рассчитаем $\forall x \in X : output^0(x)$ с помощью алгоритма последовательного распространения сигналов в НС и оценим полученное решение (результаты расчетов приведены ниже).



k = 1

X	Прообразы X	input(X)	output(X)
0	-	0.9	0.9
1	-	0.4	0.4
2	0,1	1.04	0.74
3	0,1	0.86	0.7
4	2,3	-0.92	0.29

$E^1(w)$: 0,255

Вывод: За один шаг обучения нейронной сети с помощью данного алгоритма мы снизили ошибку с 0,2695 до 0.255.