Лекция по дисциплине «Нейронные сети»

Лекция 1.14 **Парадигмы обучения нейронных сетей – 1 час**

Цель: - Рассмотреть основные парадигмы обучения нейронных сетей;

План: - Обучение нейронной сети методом обратного распространения ошибки;

Метод обратного распространения ошибки (*Error back propagation*) разработан Румельхартом в 1986 году и предназначен для обучения с учителем многослойной ИНС с прямыми связями.

Идея метода состоит в:

1) определении множества обучающих пар, состоящих из векторов значений входа и желаемого выхода;

2) вычислении ошибки, равной величине разности между значением желаемого выхода и значением реального выхода;

3) распространении этой ошибки в обратном направлении – навстречу потоку данных и перераспределяется путем изменения весов синаптических связей между нейронами так, чтобы минимизировать суммарную ошибку (см. рисунок III..1).

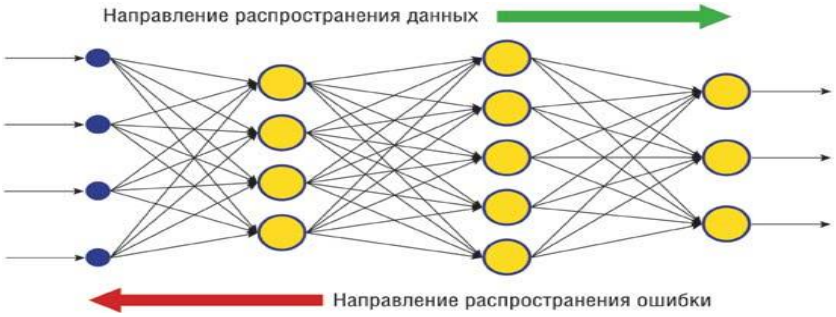

Рисунок III.1. Направление распространения ошибки.

Метод обратного распространения ошибки также известен как метод обобщения дельта-правил.

Таким образом, распространение ошибки представляет собой итеративный алгоритм обучения:

1. Выбрать случайным образом очередную обучающую пару (x_i, y_j), x_i – компоненты входного вектора, y_j – компоненты выходного вектора, $i=1,2,\ldots,K$; $j=1,2,\ldots,N$; η – коэффициент скорости обучения, $0<\eta<1$.

2. Вычислить выход ИНС в режиме обычного функционирования ИНС;

3. Вычислить разность между реальным выходом $y_{ij}^{(Q)}$ и требуемым выходом (целевым вектором) $d_{ij}^{(Q)}$ ИНС и расчитать дельту ошибки для выходного слоя по формуле:

$$\delta_{ij}^{(Q)} = (y_{ij}^{(Q)} - d_{ij}^{(Q)}) \cdot \frac{dy_{ij}}{ds_{ij}},$$

здесь $\frac{dy}{ds} = 1 - s$ в случае гиперболического тангенса и $\frac{dy}{ds} = s(1-s)$ в случае сигмоиды.

4. Вычислить изменение весов синаптических связей для выходного слоя Q ИНС, чтобы минимизировать ошибку:

$$\Delta w_{ij}^{(Q)} = -\eta \cdot \delta_{ij}^{(Q)} \cdot y_{ij}^{(Q-1)}.$$

5. Рассчитать ошибки для всех остальных слоев по формуле:

$$\delta_j^{(q)} = \left[\sum_k \delta_k^{(q+1)} \cdot w_{jk}^{(q+1)} \right] \cdot \frac{dy_j^{(q)}}{ds_j^{(q)}}.$$

6. Скорректировать все веса синаптических связей ИНС:

$$w_{ij}^{(q)}(t) = w_{ij}^{(q)}(t-1) + \Delta w_{ij}^{(q)}(t),$$

$$w_{ij}^{(q)}(t) = -\eta \cdot (\mu \cdot \Delta w_{ij}^{(q)}(t-1) + (1-\mu) \cdot \delta_j^{(q)} \cdot \delta_i^{(q-1)}),$$

где μ – коэффициент инерционности. Эта формула «гладко» изменяетть веса, что приводит к понижению перепадов ошибки. Последнюю формулу целесообразно применять в конце обучения, когда ошибка близка к заданной.

7. Повторять шаги с 1 по 6 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

8. Операции, выполняемые шагами 1, 2 и 3, сходны с теми, которые выполняются при функционировании уже обученной ИНС, т. е. подается входной вектор и вычисляется результирующий выход. Вычисления

выполняются послойно. В качестве критерия ошибки (целевой функции) используется средняя квадратичная ошибка, в которой суммирование ведется по всем нейронам выходного слоя и по всем образам, обрабатываемым ИНС: Минимизировав такой функционал, мы получим решение по методу наименьших квадратов.

В итоге каждый нейрон способен определить вклад каждого своего веса в суммарную ошибку сети.

Простейшее правило обучения соответствует методу наискорейшего спуска, в котором изменение весов синаптических связей пропорционально их вкладу в общую ошибку, то есть вычисляется чувствительность ошибки ИНС к изменению весов синаптических связей. Для этого нужно вычислить частные производные от ошибки по весам и для нахождения минимума использовать метод градиентного спуска, который обеспечивает на каждом шаге обучения изменение этих весов по формуле:

$$\Delta w_{ij}^{(q)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}^{(q)}}, \quad \text{(III.2.1.2)}$$

де $\Delta w_{ij}^{(q)}$ – коррекция весов синаптических связей, $w_{ij}^{(q)}$ – вес синапса j-го нейрона q-го слоя с i-м нейроном (q-1)-го слоя,

η – коэффициент скорости обучения, $0<\eta<1$; $q=Q$-1,Q-2,…, 2.

Таким образом, требуется определить частные производные целевой функции E по всем синаптическим весам ИНС. Согласно правилам дифференцирования сложной функции представляется:

$$\frac{\partial E}{\partial w_{ij}^{(q)}} = \frac{\partial E}{\partial y_j^{(q)}} \cdot \frac{dy_j^{(q)}}{ds_j^{(q)}} \cdot \frac{\partial s_j^{(q)}}{\partial w_{ij}^{(q)}}, \quad \text{(III.2.1.3)}$$

где $y_j^{(q)}$ – выход, $s_j^{(q)}$ – взвешенная сума входов j-го нейрона q-го слоя ИНС.

Поскольку производная функции активации должна быть определена на всей оси абсцисс, то функция единичного скачка и прочие функции активации с неоднородностями не подходят для рассматриваемых ИНС. В них

применяются такие гладкие функции, как гиперболический тангенс или классический сигмоид. Зная функцию активации можно вычислить ее производную $\frac{dy_j^{(q)}}{ds_{ij}^{(q)}}$, являющуюся вторым множителем выражения (III.2.1.3). Например, если функцией активации будет:

1) классический сигмоид – логистическая функция $f(s)=\frac{1}{1+e^{-as}}$, то ее производная равна $f'(s)=a\cdot f(s)\cdot(1-f(s))$;

2) гиперболический тангенс $f(s)=\frac{e^{ks}-e^{-ks}}{e^{ks}+e^{-ks}}$, то ее производная будет равна $f'(s)=1-s^2$.

Значение второго множителя выражения (III.2.1.3) для логистической функции вычисляется по формуле:

$$\frac{dy_j^{(q)}}{ds_j^{(q)}}=a\cdot y_j^{(q)}\cdot(1-y_j^{(q)}) \qquad \text{(III.2.1.4)}$$

Третий множитель выражения (III.2.1.3) есть ни что иное, как выход *i*-го нейрона (*q*-1)-го слоя, т.е.

$$\frac{\partial s_j^{(q)}}{\partial w_{ij}^{(q)}}=y_i^{(q-1)} \quad \text{(III.2.1.5)}$$

Теперь можно легко вычислить частные производные целевой функции по весам синаптических связей нейронов выходного слоя. Для первого множителя выражения (III.2.1.3) производя дифференцирование функции ошибки по $y_j^{(Q)}$ получим

$$\frac{\partial E}{\partial y_j^{(Q)}}=(y_j^{(Q)}-d_j^{(Q)}) \quad \text{(III.2.1.6)}$$

Далее учитывая (III.2.1.5) и (III.2.1.6) на основание выражения (III.2.1.3) можно получить:

$$\frac{\partial E}{\partial w_{ij}^{(Q)}} = (d_j^{(Q)} - y_j^{(Q)}) \cdot \frac{dy_j^{(Q)}}{ds_j^{(Q)}} \cdot y_i^{(Q-1)} \qquad \text{(III.2.1.7)}$$

Для весов синаптических связей нейронов внутренних слоев мы не можем сразу записать, чему равен первый множитель из выражения (III.2.1.3), однако его можно раскладывать следующим образом:

$$\frac{\partial E}{\partial y_j^{(q)}} = \sum_k \frac{\partial E}{\partial y_k^{(q+1)}} \cdot \frac{dy_k^{(q+1)}}{ds_k^{(q+1)}} \cdot \frac{\partial s_k^{(q+1)}}{\partial y_j^{(q)}} = \sum_k \frac{\partial E}{\partial y_k^{(q+1)}} \cdot \frac{dy_k^{(q+1)}}{ds_k^{(q+1)}} \cdot w_{jk}^{(q+1)} \quad \text{(III.2.1.8)}$$

Здесь суммирование по *k* выполняется среди нейронов слоя (q+1).

Введем новую переменную

$$\delta_j^{(q)} = \frac{\partial E}{\partial y_j^{(q)}} \cdot \frac{dy_j^{(q)}}{ds_j^{(q)}} \quad \text{(III.2.1.9)}$$

Тогда для нейронов выходного слоя учитывая (III.2.1.6) и (III.2.1.9) получим

$$\delta_j^{(Q)} = (y_j^{(Q)} - d_j^{(Q)}) \cdot \frac{dy_j^{(q)}}{ds_j^{(q)}} \quad \text{(III.2.1.10)}$$

Заметим, что в (III.2.1.8) первые два множителя есть не что иное, как $\delta_k^{(q+1)}$. Таким образом, с помощью (III.2.1.8) можно выразить величины $\delta_k^{(q)}$ для нейронов q-го слоя через $\delta_k^{(q+1)}$ для нейронов (q+1)-го слоя. Поскольку для последнего слоя $\delta_j^{(Q)}$ легко вычисляется по (III.2.1.10), то значение $\delta_j^{(q)}$ для всех нейронов всех слоев можно получить с помощью рекурсивной формулы:

$$\delta_j^{(q)} = \left[\sum_k \delta_k^{(q+1)} \cdot w_{jk}^{(q+1)} \right] \cdot \frac{dy_j^{(q)}}{ds_j^{(q)}} \quad \text{(III.2.1.11)}$$

Теперь формулу (III.2.1.2) для модификации весов синаптических связей учитывая формулы (III.2.1.5) и (III.2.1.9) можно окончательно записать в виде:

$$\Delta w_{ij}^{(q)} = -\eta \cdot \delta_j^{(q)} \cdot y_i^{(q-1)} \quad \text{(III.2.1.12)}$$

Формула (III.2.1.12) задает не–эффективное вычисление в случае, когда производные по различным весам сильно отличаются. Это соответствует ситуации, когда значение функций *s* для некоторых нейронов близки по модулю к 1 или, когда модуль некоторых весов много больше 1. В этом случае для придания процессу коррекции весов некоторой инерционности, сглаживающей резкие скачки при перемещении по поверхности целевой функции, в формулу (III.2.1.12) вводится некоторый коэффициент инерционности λ, позволяющий корректировать прираращение веса на предыдущей итерации и преодолевать локальные минимумы. Имеется два варианта модификации формулы (III.2.1.12):

1) $\Delta w_{ij}^{(q)}(t) = -\eta \cdot \delta_j^{(q)} \cdot y_i^{(q-1)} + \lambda \cdot \Delta w_{ij}^{(q)}(t-1)$, (III.2.1.13)

2) $\Delta w_{ij}^{(q)}(t) = -\eta \cdot [(1-\lambda) \cdot \delta_j^{(q)} \cdot y_i^{(q-1)} + \lambda \cdot \Delta w_{ij}^{(q)}(t-1)]$, (III.2.1.14)

где λ – коэффициент инерционности, t – номер текущей итерации, η – коэффициент скорости обучения.

Таким образом, полный алгоритм обучения ИНС с помощью алгоритма обратного распространения ошибки строиться следующими шагами:

Шаг 1. Присваиваем всем весам синаптических связей ИНС случайные начальные значения. При этом ИНС осуществляет какое-то случайное преобразование входных сигналов и значения функции ошибки (III.2.1.1) могут быть велики.

Шаг 2. Подаем на вход ИНС один из входных векторов из обучающего множества и вычисляем выходные значения ИНС, запоминая при этом выходные значения каждого из нейронов.

$$y_j^{(0)} = x_r ,$$

$I_j^{(q)}$ – число входов нейрона *j* слоя *q*; *j* =1, 2, …, J_q; $_q$ – число нейронов в слое (*q-1*) с учетом нейрона с постоянным выходным состоянием +1,

задающего смещения; = Q, Q-1,…,2,1; Q – число слоев в ИНС; – i-й вход нейрона j, слоя q; – сигмоидальная функция активации; x_r – r-я компонента входного вектора.

Шаг 3. Вычисляем сначала для выходного слоя Q значение $\delta_j^{(Q)}$ по формуле (III.2.1.10) и изменение весов синаптических связей $\Delta w^{(Q)}$ с помощью (III.2.1.12), (III.2.1.13) или (III.2.1.14), затем для всех остальных слоев q значения $\delta_j^{(q)}$ с помощью рекуррентной формулы (III.2.1.11) и изменения весов $\Delta w^{(q)}$ с помощью (III.2.1.12), (III.2.1.13) или (III.2.1.14).

Шаг 4. Скорректируем все веса синаптических связей ИНС для текущей итерации t по следующей формуле $\Delta w_{ij}^{(q)}(t) = w_{ij}^{(q)}(t-1) + \Delta w_{ij}^{(q)}(t)$.

Шаг 5. Вычисляем значение ошибки – целевой функции (III.2.1.1): если она существенна, то возвращаемся на шаг 2, иначе считаем ИНС успешно обучившейся и завершаем работу.

ИНС на шаге 1 попеременно в случайном порядке предъявляются все обучающие примеры, чтобы ИНС не забывала одни по мере запоминания других.

Из выражения $\Delta w_{ij}^{(q)} = -\eta \cdot \delta_j^{(q)} \cdot y_i^{(q-1)}$ следует, что когда выходное значение $y_i^{(q-1)}$ стремится к нулю, эффективность обучения заметно снижается. При двоичных входных векторах в среднем половина весов синаптических связей не будет корректироваться, поэтому область возможных значений выходов нейронов (0,1) нужно сдвинуть в пределы (0.5, 0.5), что достигается простыми модификациями логистических функций.

Например, классический сигмоид $f(s) = \frac{1}{1+e^{-as}}$ преобразуется к виду:

$$f(s) = -0.5 + \frac{1}{1+e^{-as}}.$$

Схема обратного распространения ошибки представлена на рисунке III.2.1.

Несмотря на многочисленные успешные применения метода обратного распространения ошибки, он очень долго обучает ИНС. Иногда ИНС может не обучиться. Длительное время обучения может быть результатом неоптимального выбора значения шага. Неудачи в обучении обычно возникают по следующим причинам: *паралича ИНС, попадания в локальный минимум и переобучение*.

Вопросы:

1. В чем заключается идея метода обратного распространения ошибки?

2. Что должно быть задано перед началом работы алгоритма обратного распространения ошибки?

3. Что используется в качестве критерия ошибки (целевой функции)?