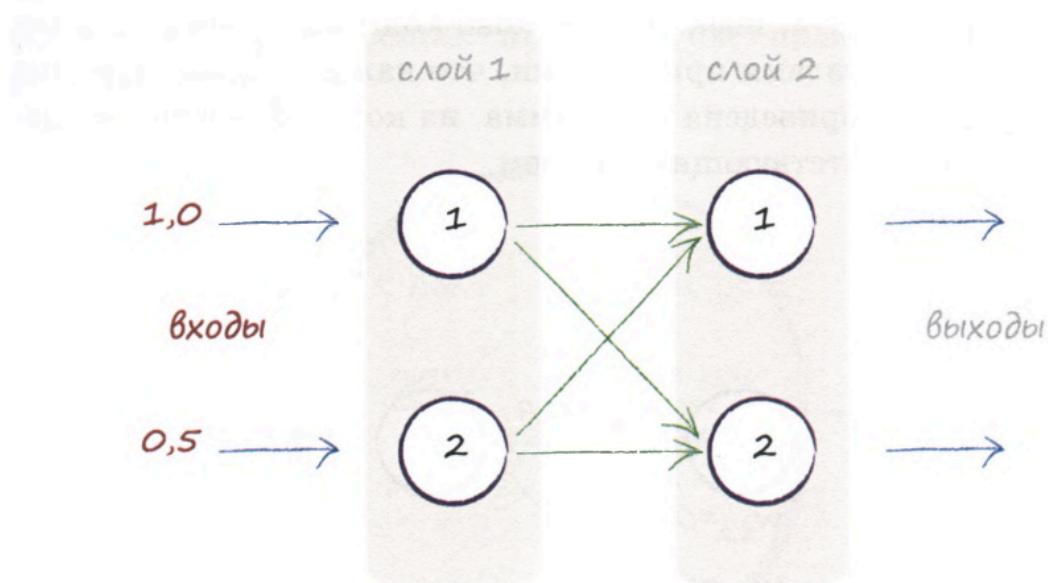


Предположим, что сигналам на входе соответствуют значения 1,0 и 0,5.



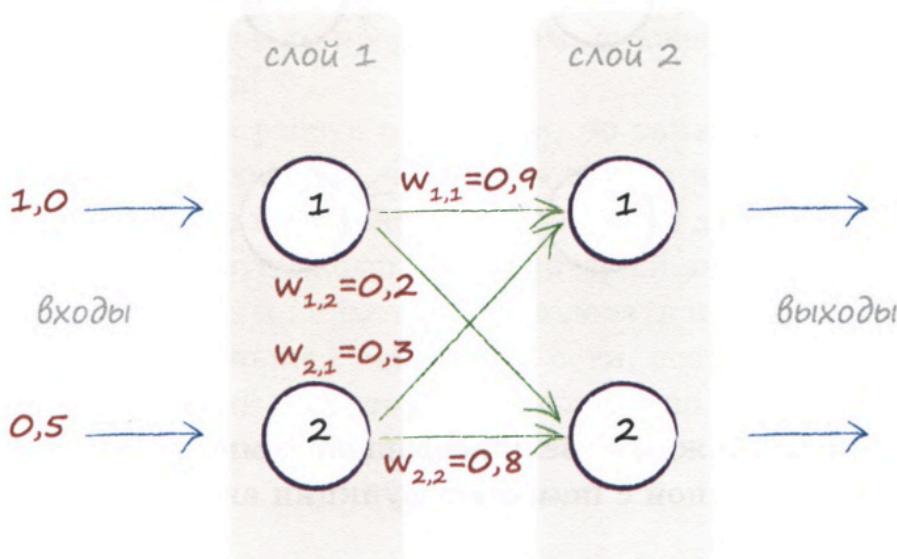
Как и раньше, каждый узел превращает сумму двух входных сигналов в один выходной с помощью функции активации. Мы также будем использовать сигмоиду  $y = \frac{1}{1 + e^{-x}}$ , с которой вы до этого познакомились, где  $x$  — это сумма сигналов, поступающих в нейрон, а  $y$  — выходной сигнал этого нейрона.

А что насчет весовых коэффициентов? Это очень хороший вопрос: с какого значения следует начать? Давайте начнем со случайных весов:

- $W_{1,1} = 0,9$
- $W_{1,2} = 0,2$
- $W_{2,1} = 0,3$
- $W_{2,2} = 0,9$

Выбор случайных начальных значений — не такая уж плохая идея, и именно так мы и поступали, когда ранее выбирали случайное начальное значение наклона прямой для простого линейного классификатора. Случайное значение улучшалось с каждым очередным тренировочным примером, используемым для обучения классификатора. То же самое должно быть справедливым и для весовых коэффициентов связей в нейронных сетях.

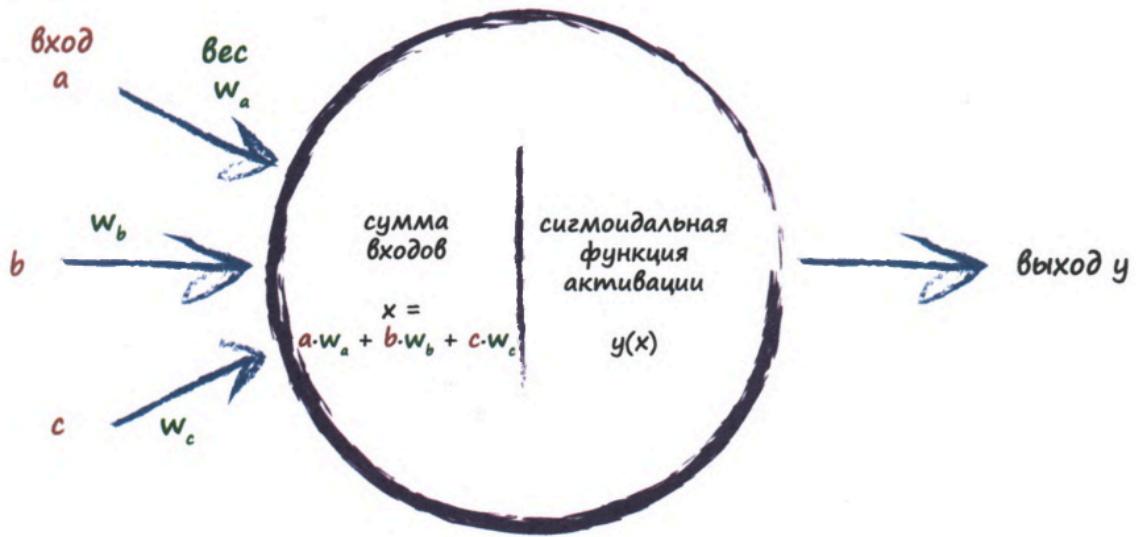
В данном случае, когда сеть небольшая, мы имеем всего четыре весовых коэффициента, поскольку таково количество всех возможных связей между узлами при условии, что каждый слой содержит по два узла. Ниже приведена диаграмма, на которой все связи про-маркированы соответствующим образом.



Первый слой узлов — входной, и его единственное назначение — представлять входные сигналы. Таким образом, во входных узлах функция активации к входным сигналам не применяется. Мы не выдвигаем в отношении этого никаких разумных доводов и просто принимаем как данность, что первый слой нейронных сетей является всего лишь входным слоем, представляющим входные сигналы. Вот и все.

С первым слоем все просто — никаких вычислений.

Далее мы должны заняться вторым слоем, в котором потребуется выполнить некоторые вычисления. Нам предстоит определить входной сигнал для каждого узла в этом слое. Помните сигмоиду  $y = \frac{1}{1 + e^{-x}}$ ? В этой функции  $x$  — комбинированный сигнал на входе узла. Данная комбинация образуется из необработанных выходных сигналов связанных узлов предыдущего слоя, сглаженных весовыми коэффициентами связей. Приведенная ниже диаграмма аналогична тем, с которыми вы уже сталкивались, но теперь на ней указано сглаживание поступающих сигналов за счет применения весовых коэффициентов связей.



Для начала сосредоточим внимание на узле 1 слоя 2. С ним связаны оба узла первого, входного слоя. Исходные значения на этих входных узлах равны 1,0 и 0,5. Связи первого узла назначен весовой коэффициент 0,9, связи второго — 0,3. Поэтому сглаженный входной сигнал вычисляется с помощью следующего выражения:

$$\begin{aligned}
 x &= (\text{выход первого узла} * \text{вес связи}) + \\
 &\quad + (\text{выход второго узла} * \text{вес связи}) \\
 &= (1,0 * 0,9) + (0,5 * 0,3) \\
 &= 0,9 + 0,15 \\
 &= 1,05
 \end{aligned}$$

Без сглаживания сигналов мы просто получили бы их сумму  $1,0 + 0,5$ , но мы этого не хотим. Именно с весовыми коэффициентами будет связан процесс обучения нейронной сети по мере того, как они будут итеративно уточняться для получения все лучшего и лучшего результата.

Итак, мы уже имеем значение  $x=1,05$  для комбинированного сглаженного входного сигнала первого узла второго слоя и теперь располагаем всеми необходимыми данными, чтобы рассчитать для этого узла выходной сигнал с помощью функции активации  $y = \frac{1}{1 + e^{-x}}$ .

Попробуйте справиться с этим самостоятельно, используя калькулятор. Вот правильный ответ:  $y = 1 / (1 + 0,3499) = 1 / 1,3499$ . Таким образом,  $y=0,7408$ .

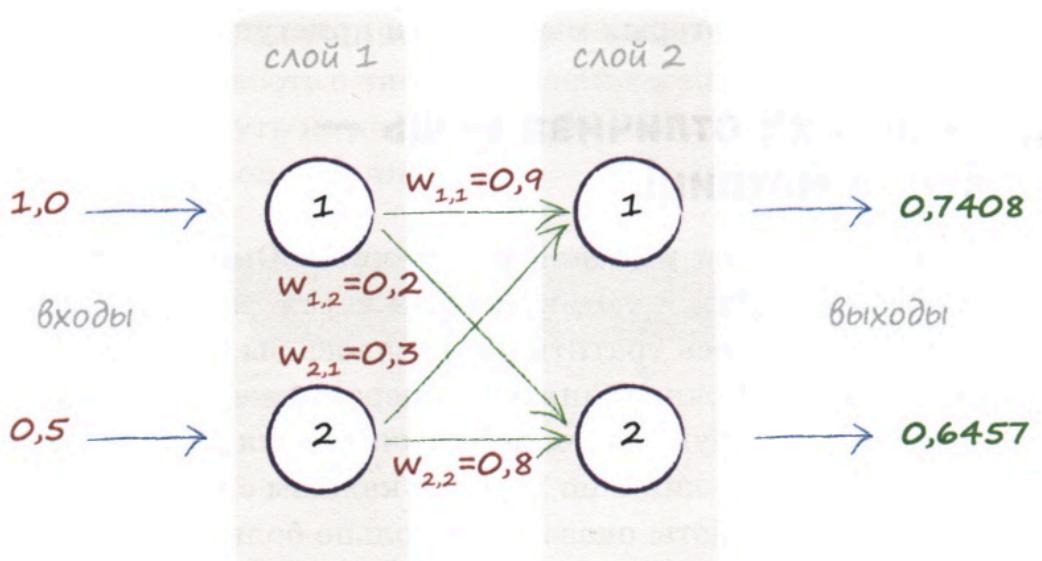
Отличная работа! Мы рассчитали фактический выходной сигнал для одного из двух выходных узлов сети.

Повторим те же вычисления для оставшегося узла — узла 2 второго слоя, т.е. вновь вычислим сглаженный входной сигнал с помощью следующего выражения:

$$\begin{aligned}
 x &= (\text{выход первого узла} * \text{вес связи}) + \\
 &\quad + (\text{выход второго узла} * \text{вес связи}) \\
 x &= (1,0 * 0,2) + (0,5 * 0,8) \\
 x &= 0,2 + 0,4 \\
 x &= 0,6
 \end{aligned}$$

Располагая значением  $x$ , можно рассчитать выходной сигнал узла с помощью функции активации:  $y = 1 / (1 + 0,5488) = 1 / 1,5488$ . Таким образом,  $y=0,6457$ .

Рассчитанные нами выходные сигналы сети представлены на приведенной ниже диаграмме.



Чем полезны матрицы, вам станет понятно, когда мы рассмотрим, как они умножаются. Возможно, вы это помните еще из курса высшей математики, а если нет, то освежите свою память.

$$\left( \begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right) \left( \begin{array}{cc} 5 & 6 \\ 7 & 8 \end{array} \right) = \left( \begin{array}{cc} (1*5) + (2*7) & (1*6) + (2*8) \\ (3*5) + (4*7) & (3*6) + (4*8) \end{array} \right)$$

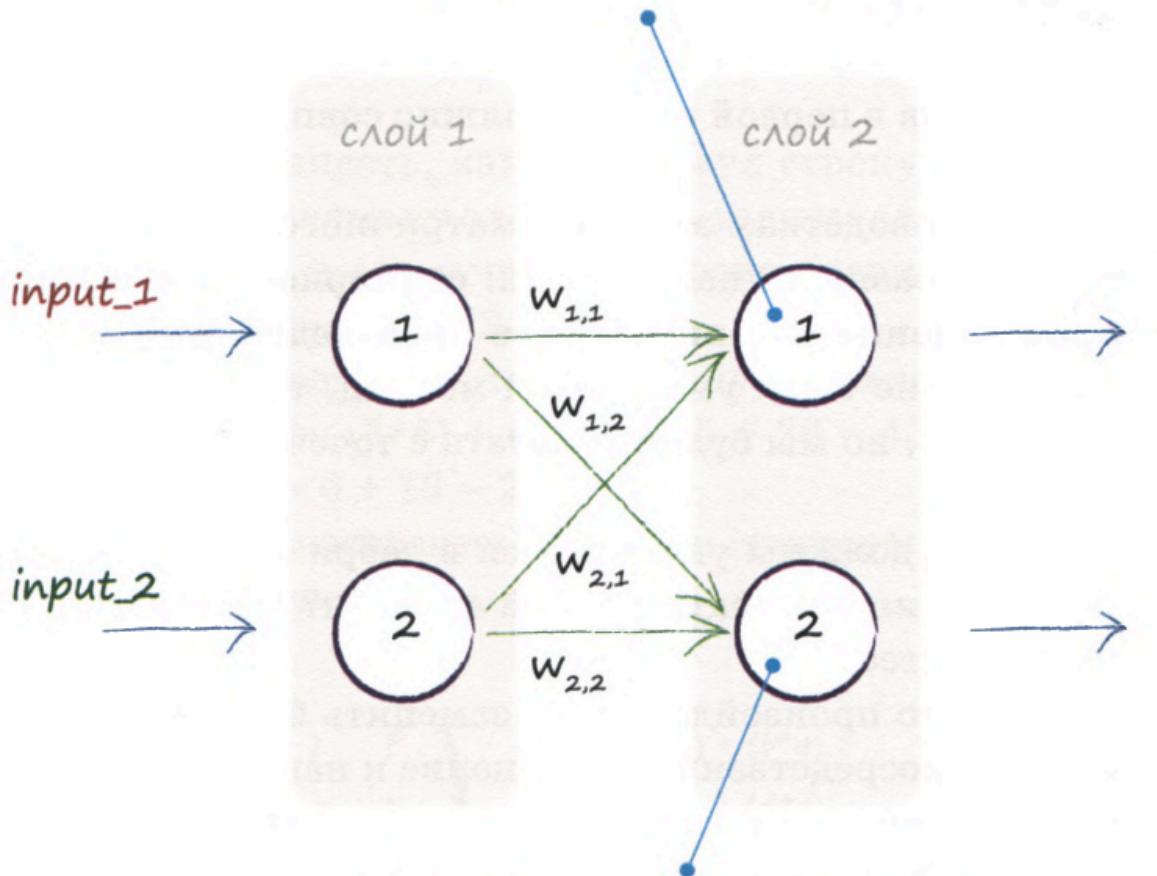
$$= \left( \begin{array}{cc} 19 & 22 \\ 43 & 50 \end{array} \right)$$

$$\left( \begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right) \left( \begin{array}{cc} 5 & 6 \\ 7 & 8 \end{array} \right) = \left( \begin{array}{cc} (1*5) + (2*7) & (1*6) + (2*8) \\ (3*5) + (4*7) & (3*6) + (4*8) \end{array} \right)$$

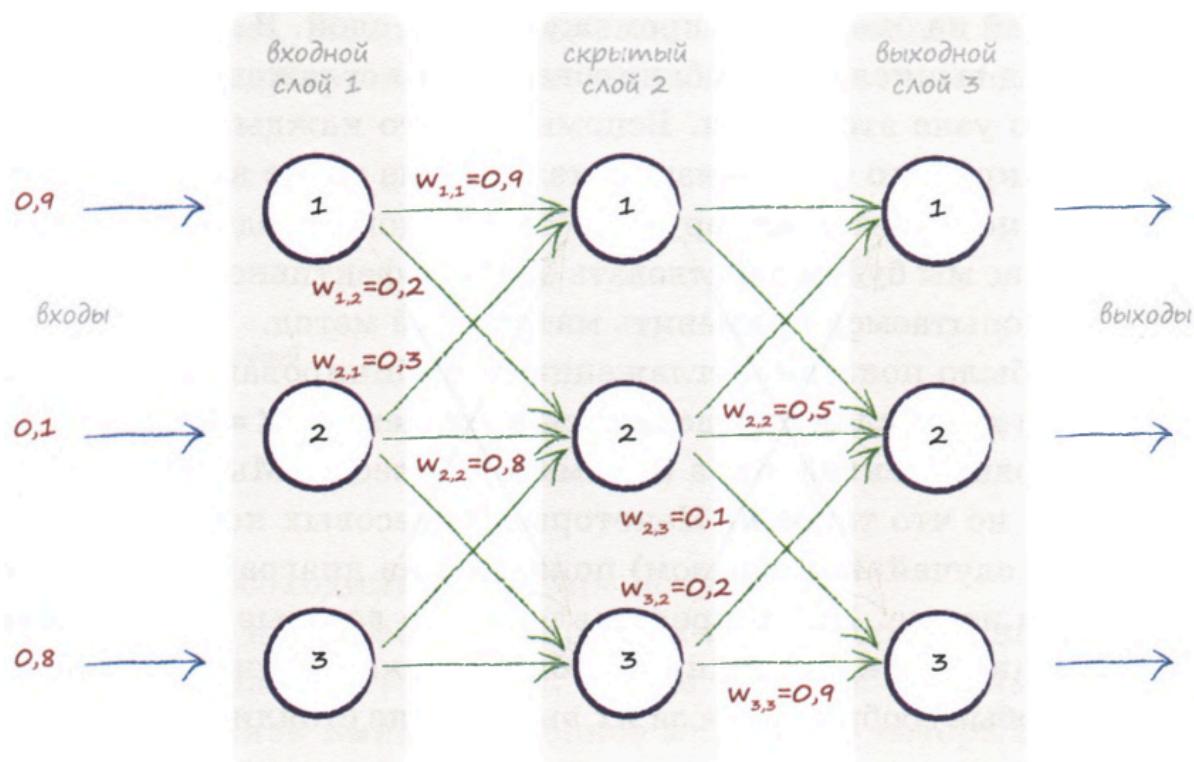
$$= \left( \begin{array}{cc} 19 & 22 \\ 43 & 50 \end{array} \right)$$

$$\begin{pmatrix} w_{1,1} & w_{2,1} \\ w_{1,2} & w_{2,2} \end{pmatrix} \begin{pmatrix} \text{input\_1} \\ \text{input\_2} \end{pmatrix} = \begin{pmatrix} (\text{input\_1} * w_{1,1}) + (\text{input\_2} * w_{2,1}) \\ (\text{input\_1} * w_{1,2}) + (\text{input\_2} * w_{2,2}) \end{pmatrix}$$

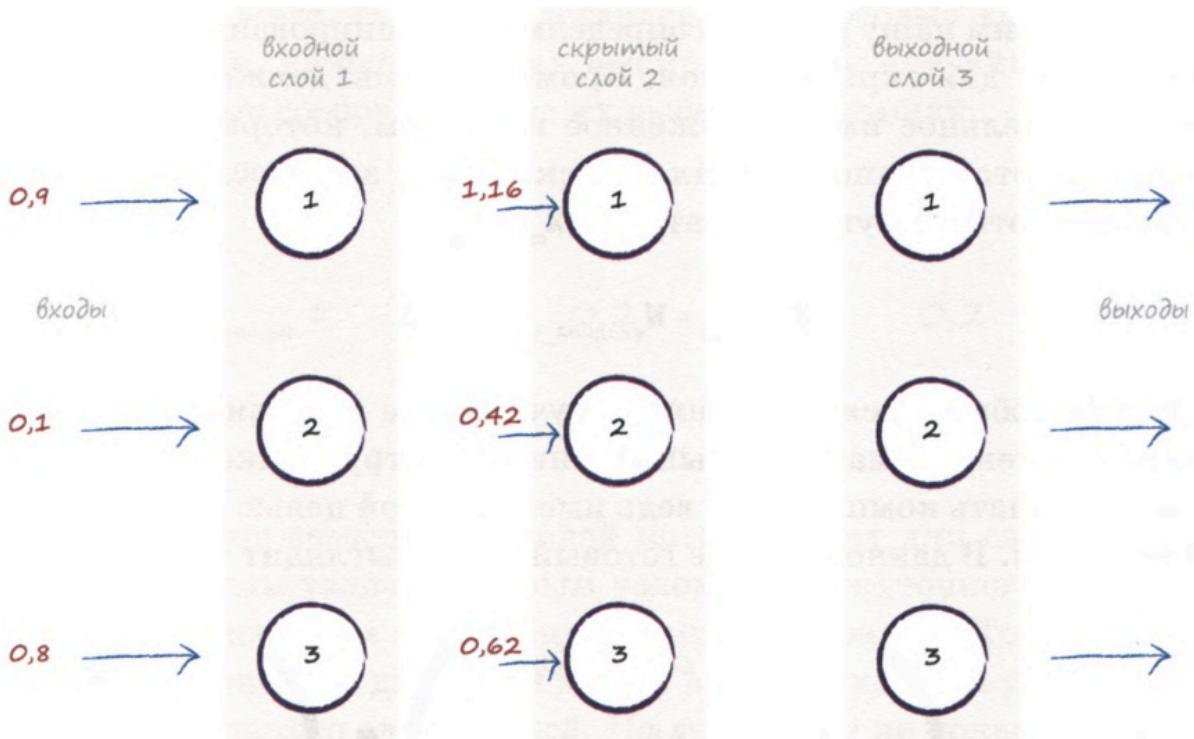
$$x = (\text{input\_1} * w_{1,1}) + (\text{input\_2} * w_{2,1})$$



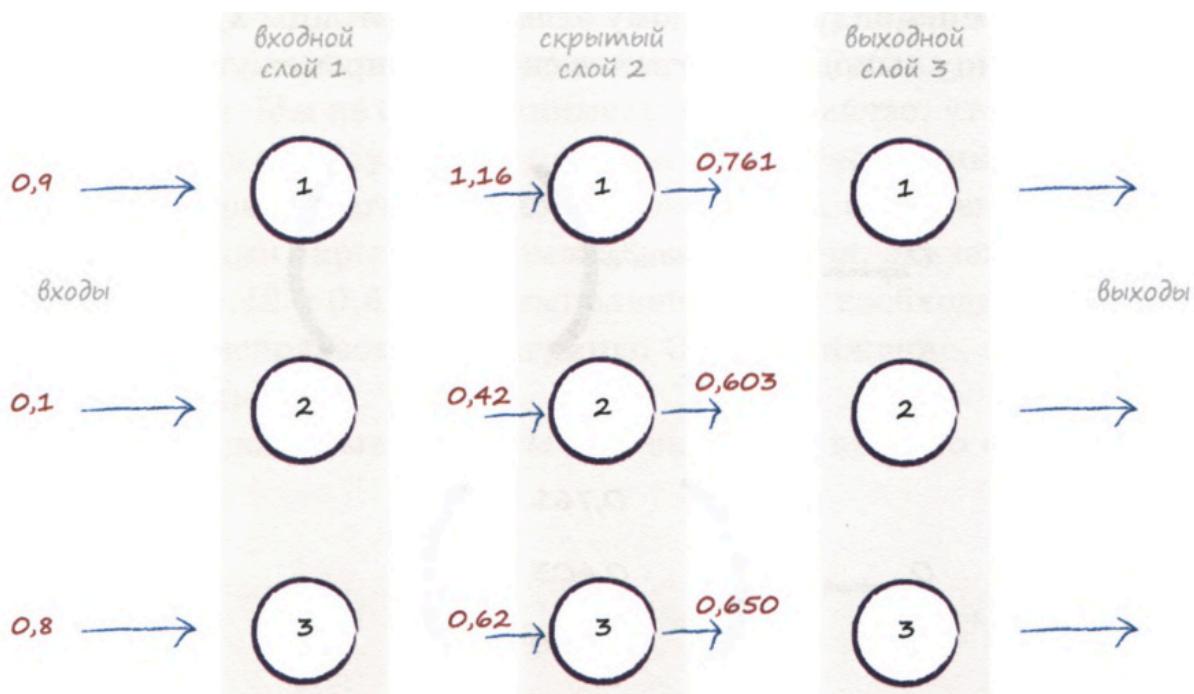
$$x = (\text{input\_1} * w_{1,2}) + (\text{input\_2} * w_{2,2})$$



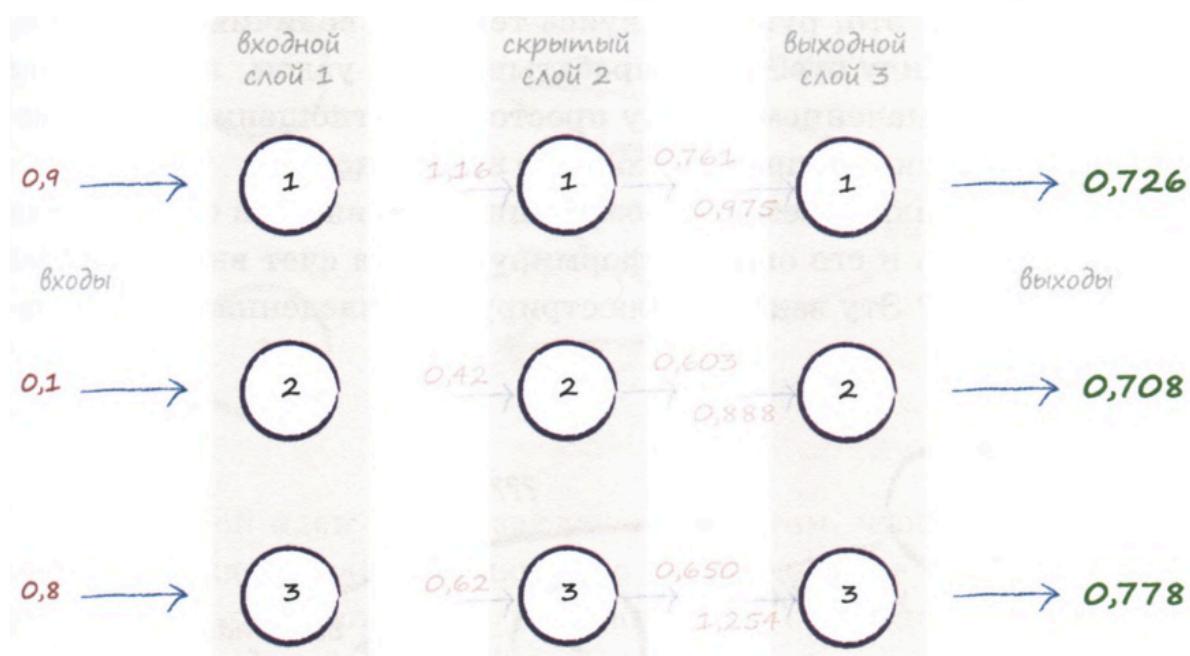
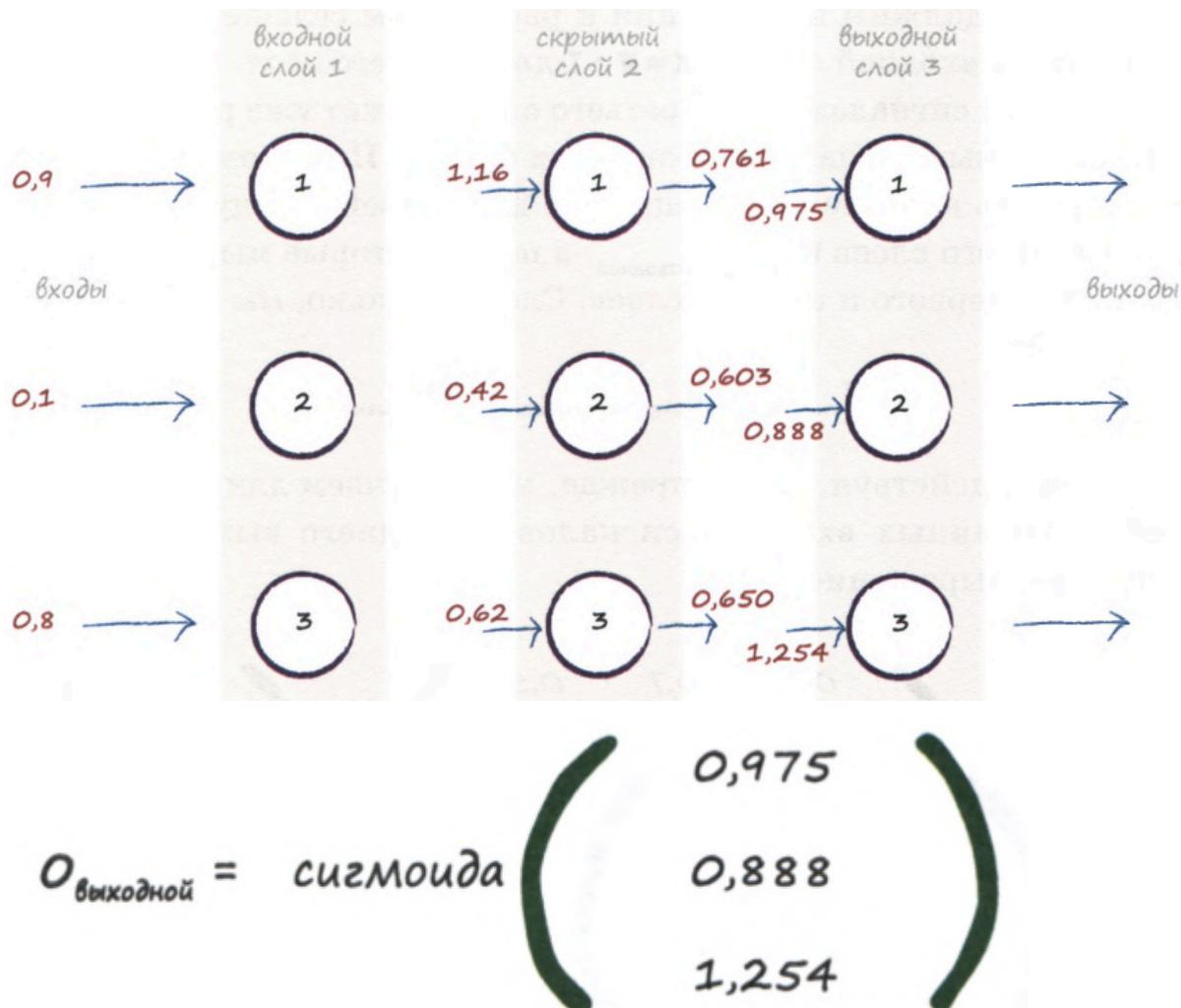
$$X_{\text{скрытый}} = \begin{pmatrix} 0,9 & 0,3 & 0,4 \\ 0,2 & 0,8 & 0,2 \\ 0,1 & 0,5 & 0,6 \end{pmatrix} \cdot \begin{pmatrix} 0,9 \\ 0,1 \\ 0,8 \end{pmatrix}$$



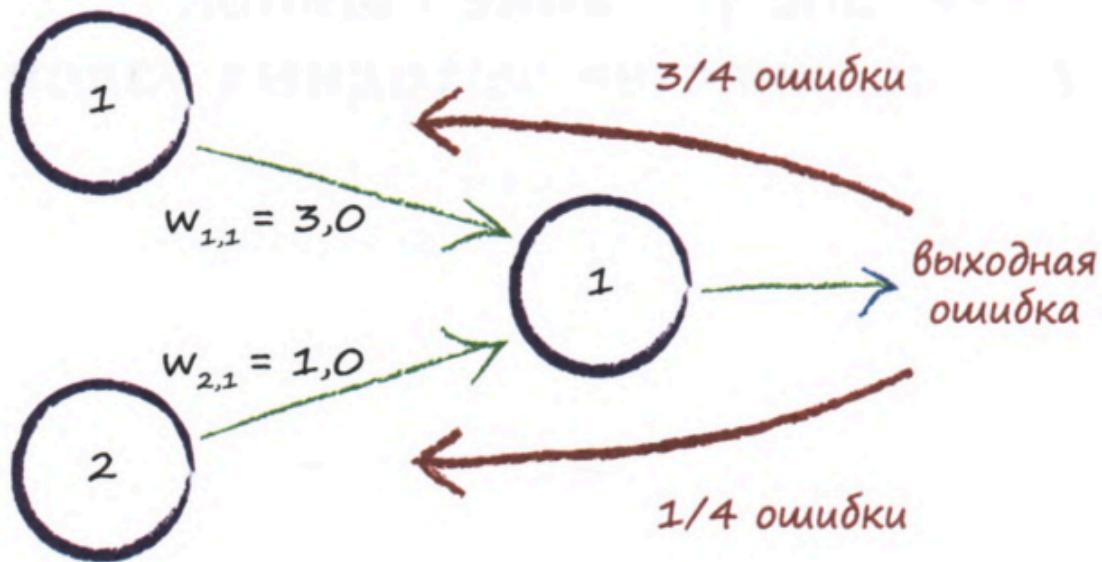
$$O_{\text{скрытый}} = \text{сигмоида} \begin{pmatrix} 1,16 \\ 0,42 \\ 0,62 \end{pmatrix}$$



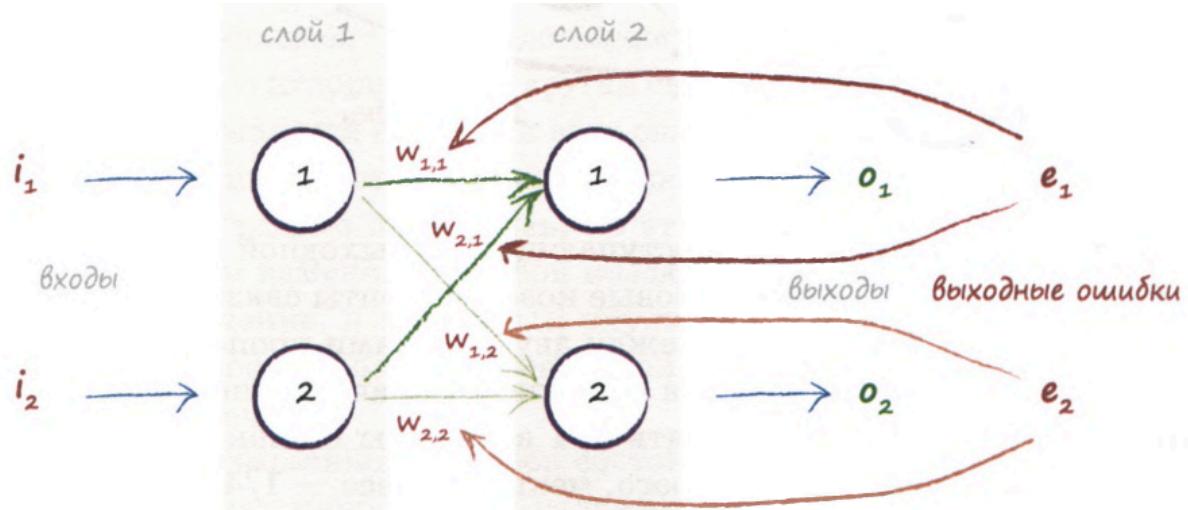
$$X_{\text{выходной}} = \begin{pmatrix} 0,3 & 0,7 & 0,5 \\ 0,6 & 0,5 & 0,2 \\ 0,8 & 0,1 & 0,9 \end{pmatrix} \cdot \begin{pmatrix} 0,761 \\ 0,603 \\ 0,650 \end{pmatrix}$$



Как нам обновлять весовые коэффициенты связей в случае, если выходной сигнал и его ошибка формируются за счет вкладов более чем одного узла? Эту задачу иллюстрирует приведенная ниже диаграмма.



Как вы могли заметить, мы используем весовые коэффициенты в двух целях. Во-первых, они учитываются при расчете распространения сигналов по нейронной сети от входного слоя до выходного. Мы интенсивно использовали их ранее именно в таком качестве. Во-вторых, мы используем веса для распространения ошибки в обратном направлении — от выходного слоя вглубь сети. Думаю, вы не будете удивлены, узнав, что этот метод называется **обратным распространением ошибки** (обратной связью) в процессе обучения нейронной сети.



Ошибка может формироваться на обоих узлах — фактически эта ситуация очень похожа на ту, которая возникает, когда сеть еще не обучалась. Вы видите, что для коррекции весов внутренних связей нужна информация об ошибках в обоих узлах. Мы можем использо-

вать прежний подход и распределять ошибку выходного узла между связанными с ним узлами пропорционально весовым коэффициентам соответствующих связей.

Доля  $e_1$ , используемая для обновления  $w_{21}$ , определяется аналогичным выражением:

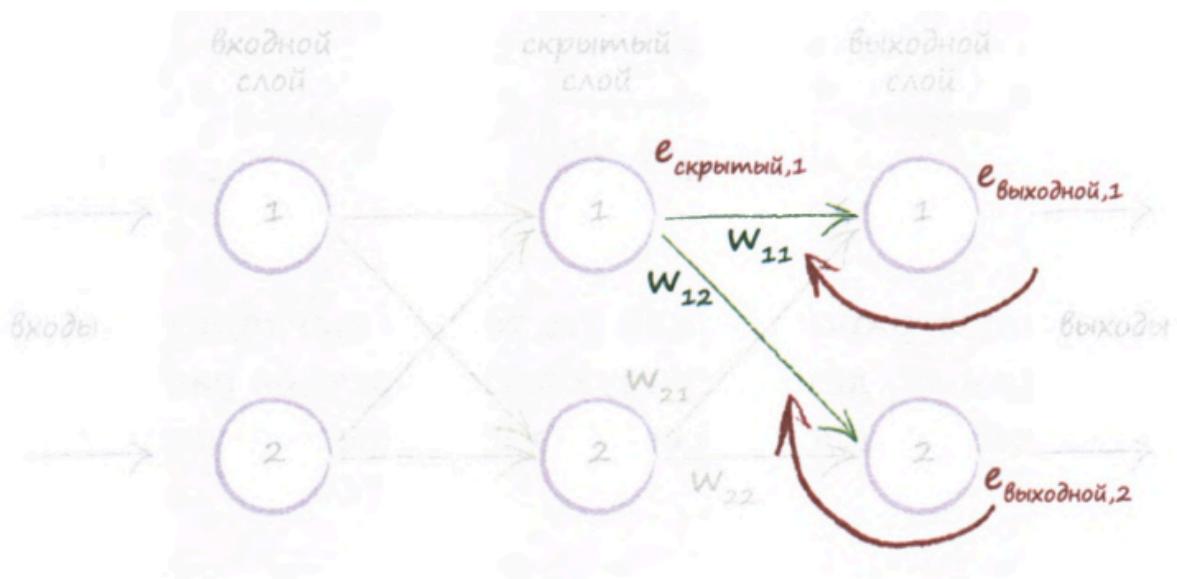
$$\frac{w_{21}}{w_{11} + w_{21}}$$

Возможно, эти выражения несколько смущают вас, поэтому рассмотрим их более подробно. За всеми этими символами стоит очень простая идея, которая заключается в том, что узлы, сделавшие больший вклад в ошибочный ответ, получают больший сигнал об ошибке, тогда как узлы, сделавшие меньший вклад, получают меньший сигнал.

Если  $w_{11}$  в два раза превышает  $w_{21}$  (скажем,  $w_{11}=6$ , а  $w_{21}=3$ ), то доля  $e_1$ , используемая для обновления  $w_{11}$ , составляет  $6/(6+3) = 6/9 = 2/3$ . Тогда для другого, меньшего веса  $w_{21}$  должно остаться  $1/3 e_1$ , что можно подтвердить с помощью выражения  $3/(6+3) = 3/9$ , результат которого действительно равен  $1/3$ .

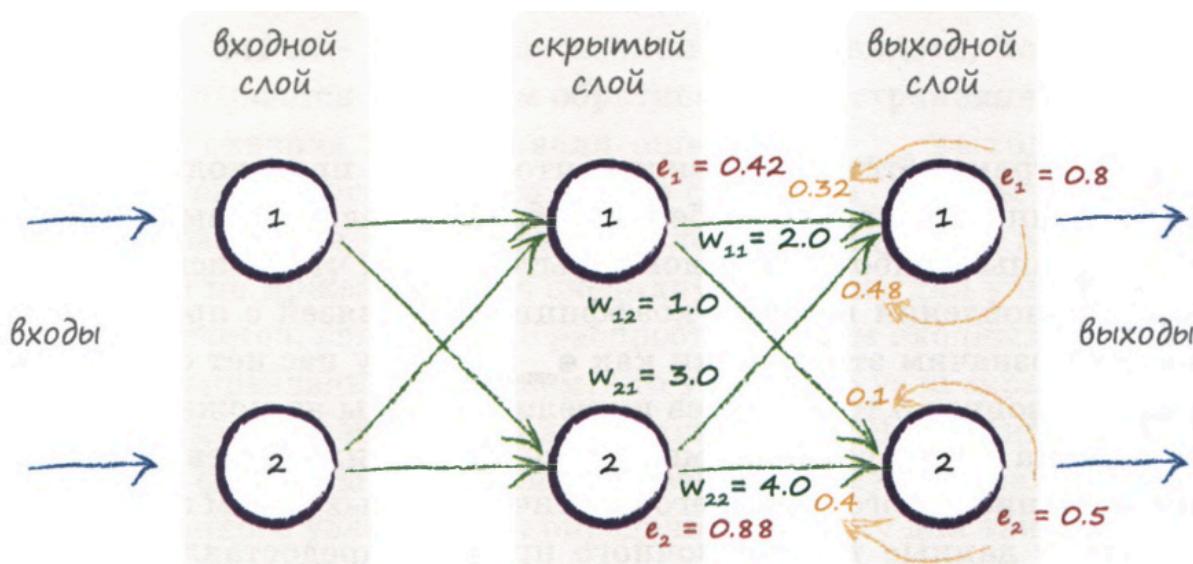


Это означает, что с каждой из двух связей, исходящих из узла промежуточного слоя, ассоциируется некоторая ошибка. Мы могли бы воссоединить ошибки этих двух связей, чтобы получить ошибку для этого узла в качестве второго наилучшего подхода, поскольку мы не располагаем фактическим целевым значением для узла промежуточного слоя. Следующая диаграмма иллюстрирует эту идею:



$e_{\text{скрытый},1}$  = сумма ошибок, распределенных по связям  $w_{11}$  и  $w_{12}$

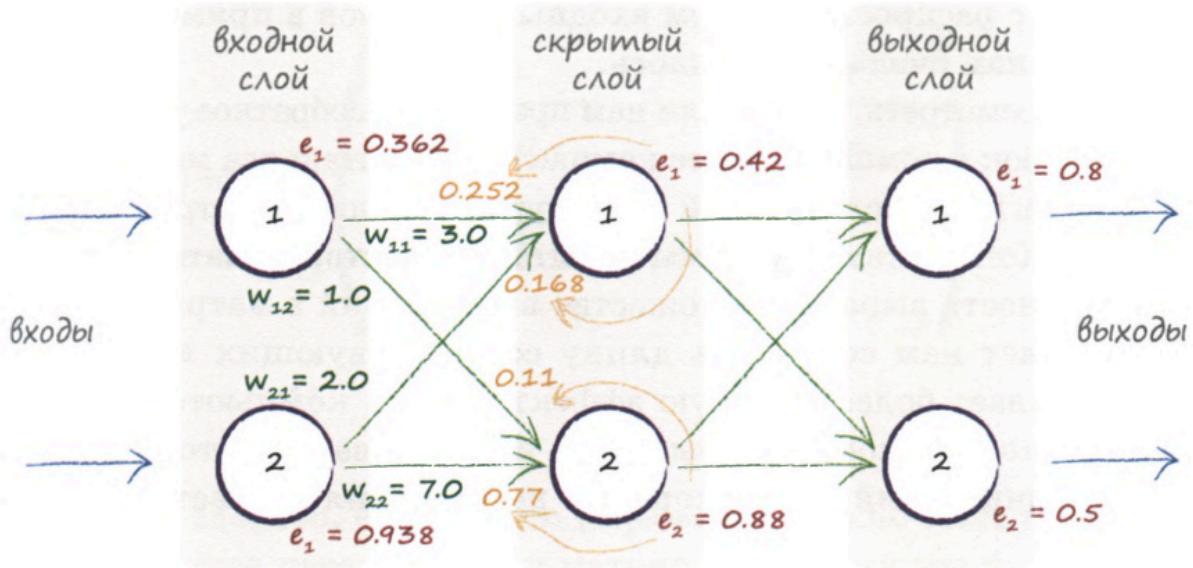
$$= e_{\text{выходной},1} * \frac{w_{11}}{w_{11} + w_{21}} + e_{\text{выходной},2} * \frac{w_{12}}{w_{12} + w_{22}}$$



Проследим за обратным распространением одной из ошибок. Вы видите, что после распределения ошибки 0,5 на втором узле выходного слоя между двумя связями с весами 1,0 и 4,0 мы получаем доли, равные 0,1 и 0,4 соответственно. Также можно видеть, что объединенная ошибка на втором узле скрытого слоя представляет собой сумму распределенных ошибок, в данном случае равных 0,48 и 0,4, сложение которых дает 0,88.

На следующей диаграмме демонстрируется применение той же методики к слою, который предшествует скрытому.

Нейронные сети обучаются посредством уточнения весовых коэффициентов своих связей. Этот процесс управляет ошибкой — разностью между правильным ответом, предоставляемым тренировочными данными, и фактическим выходным значением.



$$\text{ошибка}_{\text{скрытый}} = \left( \begin{array}{c} \frac{w_{11}}{w_{11} + w_{21}} \\ \frac{w_{21}}{w_{21} + w_{11}} \end{array} \right) \cdot \left( \begin{array}{c} e_1 \\ e_2 \end{array} \right)$$

Было бы здорово, если бы это выражение можно было переписать в виде простого перемножения матриц, которыми мы уже располагаем. Это матрицы весовых коэффициентов, прямого сигнала и выходных ошибок. Преимущества, которые можем при этом получить, огромны.

К сожалению, легкого способа превратить это выражение в сверхпростое перемножение матриц, как в случае распространения сигналов в прямом направлении, не существует. Распутать все эти доли, из которых образованы элементы большой матрицы, непросто. Было бы замечательно, если бы мы смогли представить эту матрицу в виде комбинации имеющихся матриц.

Что можно сделать? Нам позарез нужен способ, обеспечивающий возможность использования матричного умножения, чтобы повысить эффективность вычислений.

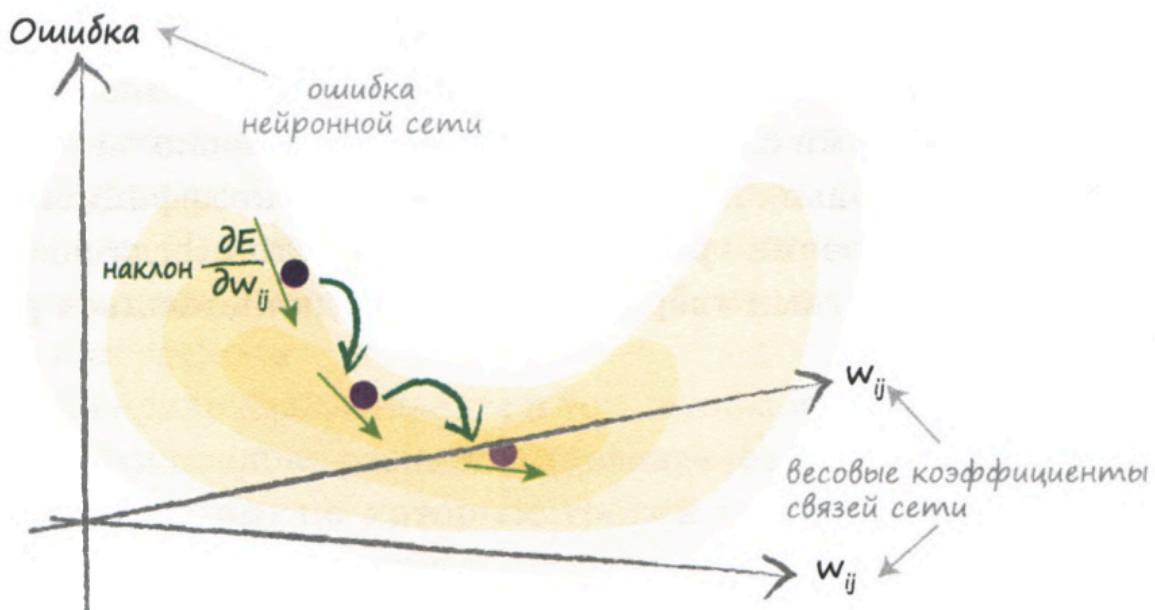
Взгляните еще раз на приведенное выше выражение. Вы видите, что наиболее важная для нас вещь — это умножение выходных ошибок  $e_n$  на связанные с ними веса  $w_{ij}$ . Чем больше вес, тем большая доля ошибки передается обратно в скрытый слой. Это важный момент. В дробях, являющихся элементами матрицы, нижняя часть играет роль нормирующего множителя. Если пренебречь этим фактором, можно потерять лишь масштабирование ошибок, передаваемых по механизму обратной связи. Таким образом, выражение  $e_1 * w_{11} / (w_{11} + w_{21})$  упростится до  $e_1 * w_{11}$ .

Сделав это, мы получим следующее уравнение.

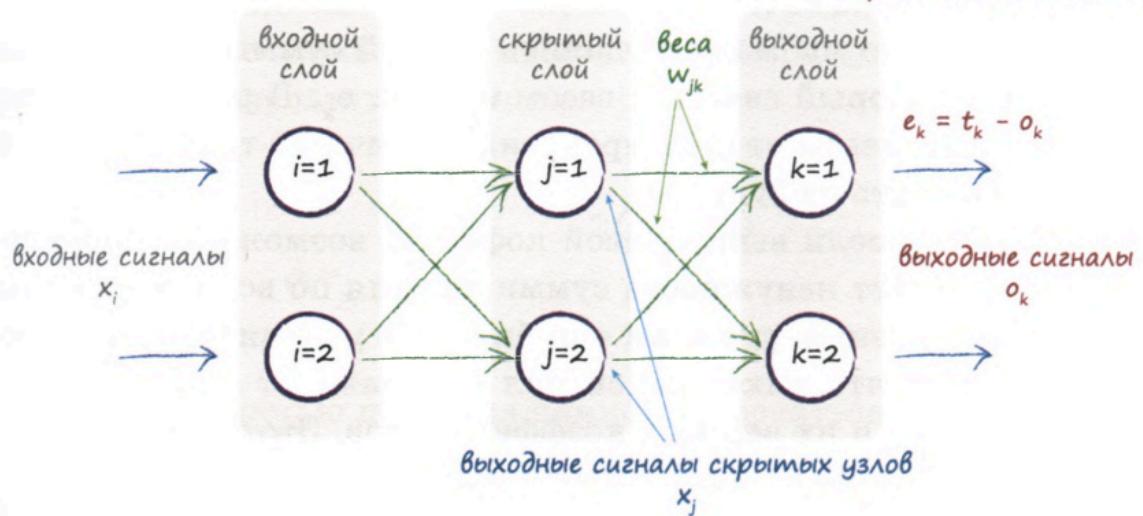
$$\text{ошибка}_{\text{скрытый}} = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} \cdot \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}$$

Эта матрица весов напоминает ту, которую мы строили ранее, но она повернута вокруг диагонали, так что правый верхний элемент теперь стал левым нижним, а левый нижний — правым верхним. Такая матрица называется транспонированной и обозначается как  $w^T$ .

$$\text{ошибка}_{\text{скрытый}} = w^T_{\text{скрытый\_выходной}} \cdot \text{ошибка}_{\text{выходной}}$$



ошибка узла = целевое значение - фактическое значение



$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}} \sum_n (t_n - o_n)^2$$

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial o_k} \cdot \frac{\partial o_k}{\partial w_{jk}}$$

$$\frac{\partial E}{\partial w_{jk}} = -2(t_k - o_k) \cdot \frac{\partial o_k}{\partial w_{jk}}$$

$$\frac{\partial E}{\partial w_{jk}} = -2(t_k - o_k) \cdot \frac{\partial}{\partial w_{jk}} \text{ сигмоида} (\sum_j w_{jk} \cdot o_j)$$

$$\frac{\partial}{\partial x} \text{ сигмоида}(x) = \text{сигмоида}(x)(1 - \text{сигмоида}(x))$$

$$\begin{aligned} \frac{\partial E}{\partial w_{jk}} &= -2(t_k - o_k) \cdot \text{сигмоида}(\sum_j w_{jk} \cdot o_j) (1 - \text{сигмоида}(\sum_j w_{jk} \cdot o_j)) \cdot \frac{\partial}{\partial w_{jk}} (\sum_j w_{jk} \cdot o_j) \\ &= -2(t_k - o_k) \cdot \text{сигмоида}(\sum_j w_{jk} \cdot o_j) (1 - \text{сигмоида}(\sum_j w_{jk} \cdot o_j)) \cdot o_j \end{aligned}$$

$$\frac{\partial E}{\partial w_{jk}} = -(t_k - o_k) \cdot \text{сигмоида}(\sum_j w_{jk} \cdot o_j) (1 - \text{сигмоида}(\sum_j w_{jk} \cdot o_j)) \cdot o_j$$

$$\frac{\partial E}{\partial w_{ij}} = -(e_j) \cdot \text{сигмоида}(\sum_i w_{ij} \cdot o_i) (1 - \text{сигмоида}(\sum_i w_{ij} \cdot o_i)) \cdot o_i$$

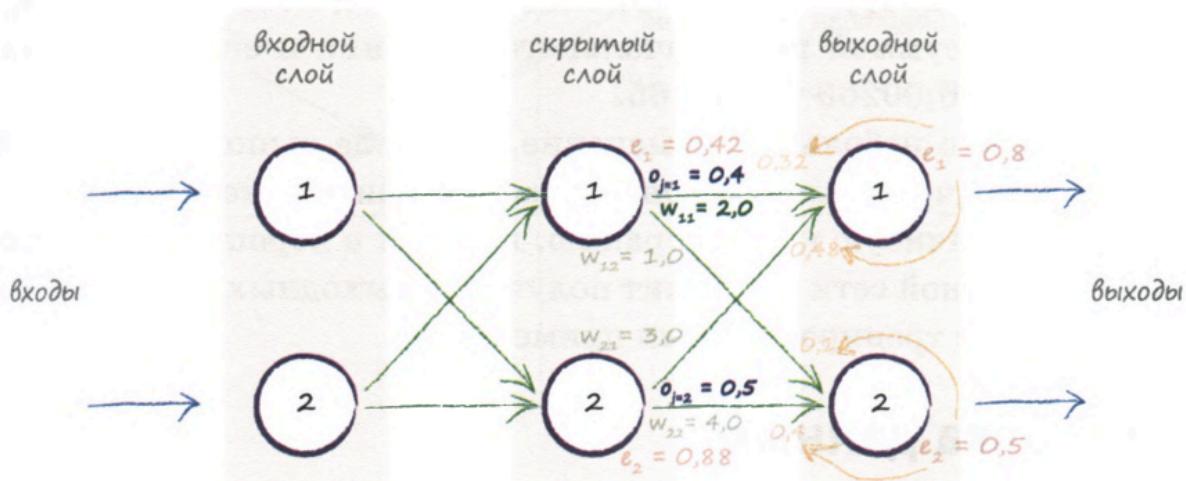
$$\text{новый } w_{jk} = \text{старый } w_{jk} - \alpha \cdot \frac{\partial E}{\partial w_{jk}}$$

$$\left( \begin{array}{cccc} \Delta w_{1,1} & \Delta w_{2,1} & \Delta w_{3,1} & \dots \\ \Delta w_{1,2} & \Delta w_{2,2} & \Delta w_{3,2} & \dots \\ \Delta w_{1,3} & \Delta w_{2,3} & \Delta w_{3,k} & \dots \\ \dots & \dots & \dots & \dots \end{array} \right) = \left( \begin{array}{c} E_1 * S_1 (1-S_1) \\ E_2 * S_2 (1-S_2) \\ E_k * S_k (1-S_k) \\ \dots \end{array} \right) \cdot \left( \begin{array}{c} o_1 \\ o_2 \\ o_j \\ \dots \end{array} \right)$$

↑  
значения из следующего слоя

↑  
значения из предыдущего слоя

$$\Delta W_{jk} = \alpha \cdot E_k \cdot o_k (1 - o_k) \cdot o_j^T$$



$$\frac{\partial E}{\partial w_{jk}} = -(t_k - o_k) \cdot \text{сигмоида}(\sum_j w_{jk} \cdot o_j) (1 - \text{сигмоида}(\sum_j w_{jk} \cdot o_j)) \cdot o_j$$

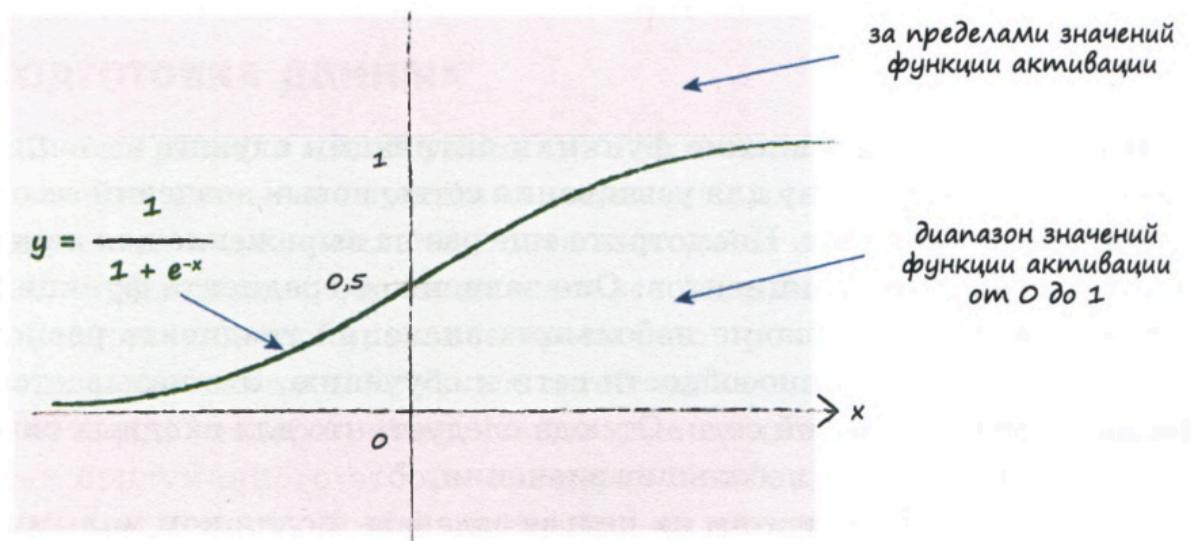
Разберем это выражение по частям.

- Первая часть ( $t_k - o_k$ ) — это уже известная вам по предыдущим диаграммам ошибка  $e_1=0,8$ .
- Сумма  $\sum_j w_{jk} o_j$ , передаваемая сигмоидам, равна  $(2,0 * 0,4) + (3,0 * 0,5) = 2,3$ .
- Тогда сигмоида  $1/(1 + e^{-2,3})$  равна 0,909. Следовательно, промежуточное выражение равно  $0,909 * (1 - 0,909) = 0,083$ .
- Последняя часть — это просто сигнал  $o_j$ , которым в данном случае является сигнал  $o_{j-1}$ , так как нас интересует вес  $w_{11}$ , где  $j=1$ . Поэтому данная часть просто равна 0,4.

Перемножив все три части этого выражения и не забыв при этом о начальном знаке “минус”, получаем значение  $-0,0265$ .

При коэффициенте обучения, равном 0,1, изменение веса составит  $-0,1 * (-0,0265) = +0,002650$ . Следовательно, новое значение  $w_{11}$ , определяемое суммой первоначального значения и его изменения, составит  $2,0 + 0,00265 = 2,00265$ .

Это довольно небольшое изменение, но после выполнения сотен или даже тысяч итераций весовые коэффициенты в конечном счете образуют устойчивую конфигурацию, которая в хорошо натренированной нейронной сети обеспечит получение выходных сигналов, согласующихся с тренировочными примерами.



Значительное спрямление функции активации служит источником проблем, поскольку для усваивания сетью новых значений весов используется градиент. Посмотрите еще раз на выражение для изменений весовых коэффициентов. Оно зависит от градиента функции активации. Использование небольших значений градиента равносильно ограничению способности сети к обучению. Это называется **насыщением** нейронной сети. Отсюда следует, что для входных сигналов лучше задавать небольшие значения.

Любопытно, что при этом их нельзя задавать и слишком малыми, поскольку они тоже входят в указанное выражение для изменений весовых коэффициентов. Слишком малые значения входных сигналов могут быть проблематичными еще и потому, что при обработке очень больших или очень малых значений точность компьютерных вычислений значительно снижается.

Неплохим решением этой проблемы является масштабирование входных сигналов до значений в интервале от 0,0 до 1,0. Иногда для входных сигналов вводят небольшое смещение, скажем, 0,01, с целью недопущения нулевых входных сигналов, которые неприятны тем, что при  $o_j=0$  выражение для поправки к весам обнуляется, тем самым лишая сеть способности к обучению.

Выходные значения нейронной сети — это сигналы, появляющиеся на узлах последнего слоя. Если мы используем функцию активации, которая не обеспечивает получение значений выше 1,0, то было бы глупо пытаться устанавливать значения большей величины в качестве целевых. Вспомните о том, что логистическая функция не дотягивает даже до значения 1,0 — она только приближается к нему. В математике это называется **асимптотическим стремлением** к 1,0.

Если мы все же установим целевые значения в этих недостижимых, запрещенных диапазонах, то тренировка сети приведет к еще большим весовым коэффициентам в попытке добиться все больших и больших значений выходных сигналов, которые фактически никогда не могут быть достигнуты вследствие использования

функции активации. Мы понимаем, что это так же плохо, как и насыщение сети.

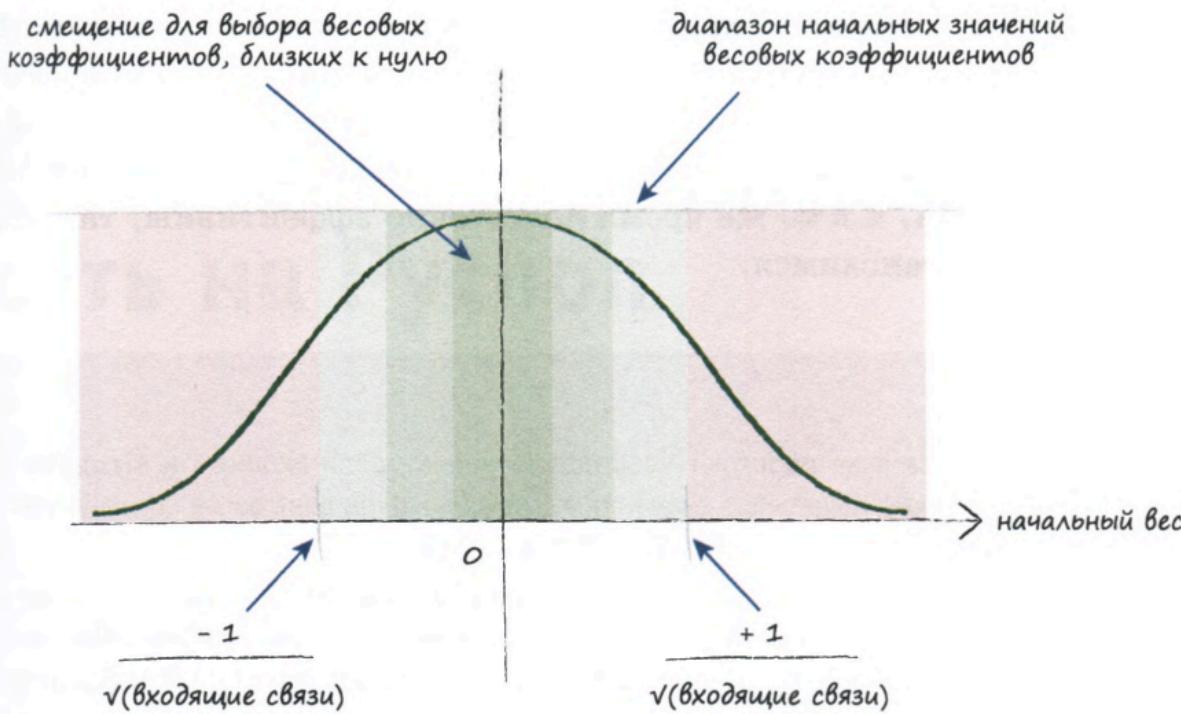
Поэтому мы должны масштабировать наши целевые выходные значения таким образом, чтобы они были допустимыми при данной функции активации, одновременно заботясь о том, чтобы избежать значений, которые в действительности никогда не могут быть достигнуты.

Общепринято использовать диапазон значений от 0,0 до 1,0, но некоторые разработчики используют диапазон от 0,01 до 0,99, поскольку значения 0,0 и 1,0, с одной стороны, не являются подходящими целевыми значениями, а с другой — могут приводить к чрезмерно большим значениям весовых коэффициентов.

Математики и ученые-компьютерщики разработали подходы, позволяющие определять эмпирические правила для задания случайных начальных значений весовых коэффициентов в зависимости от конкретной конфигурации сети и используемой функции активации. Соответствующие рецепты в высшей степени специфичны, но, невзирая на это, мы рискнем подступиться к ним!

Мы не будем вдаваться в детали математических выкладок, но центральная идея заключается в том, что если на узел нейронной сети поступает множество сигналов, причем поведение этих сигналов хорошо определено, они не достигают слишком больших значений и не распределены каким-то невероятным образом, то весовые коэффициенты не должны нарушать такое состояние сигналов в процессе их объединения и обработки функцией активации. Иными словами, мы не должны использовать весовые коэффициенты, разрушающие результаты наших попыток тщательно масштабировать входные сигналы. Суть эмпирического правила, к которому пришли математики, заключается в том, что весовые коэффициенты инициализируются числами, случайно выбираемыми из диапазона, грубая оценка которого определяется обратной величиной квадратного корня из количества связей, ведущих к узлу. Таким образом, если к каждому узлу ведут три связи, то начальные значения весов не должны превышать значение  $1/(\sqrt{3}) = 0,577$ . Если же каждый узел имеет 100 входящих связей, то веса должны находиться в диапазоне, ограниченном значением  $1/(\sqrt{100}) = 0,1$ .

Интуитивно это понятно. Некоторые слишком большие веса сместили бы функцию активации в область больших значений, что привело бы к ее **насыщению**. И чем больше связей приходится на узел, тем больше складывается весовых коэффициентов. Поэтому эмпирическое правило, которое уменьшает диапазон значений весовых коэффициентов с увеличением количества связей на узел, находит логическое объяснение.



В любом случае никогда не задавайте для начальных весов равные значения, особенно нулевые. Это был бы крайне неудачный вариант!

Этот вариант был бы неудачным по той причине, что в таком случае все узлы сети получили бы одинаковые сигналы, и сигналы на выходе каждого узла были бы одинаковыми. Если затем приступить к обновлению весов с использованием механизма обратного распространения ошибки, то ошибка распределится равномерно. Вы ведь не забыли, что ошибка распределяется между узлами пропорционально весам. Это приведет к одинаковым поправкам для всех весовых коэффициентов, что, в свою очередь, вновь приведет к весам, имеющим одинаковые значения. Подобная симметрия играет крайне отрицательную роль, ведь если правильно натренированная сеть должна иметь неодинаковые значения весовых коэффициентов (что характерно для большинства задач), то вы никогда не достигнете этого состояния.

Еще худший выбор — нулевые значения, поскольку они полностью “убивают” входной сигнал. В этом случае функция обновления весов, которая зависит от входных сигналов, обнуляется, тем самым полностью исключая саму возможность обновления весов.