$c_1 = (0,0), c_2 = (1,0), c_3 = (0,1), c_4 = (1,1)$, обеспечивающим равномерную "загрузку" кодирующих нейронов.

Отличие между входными и выходными переменными

В заключении данного раздела отметим одно существенное отличие способов кодирования входных и выходных переменных, вытекающее из определения градиента ошибки: $\frac{\partial E}{\partial w_{ij}^{[n]}} = \delta_i^{[n]} x_j^{[n]}$. А именно, входы участвуют в обучении непосредственно, тогда как выходы - лишь *опосредованно* - через ошибку верхнего слоя. Поэтому при кодировании категорий в качестве выходных нейронов можно использовать как логистическую функцию активации $f(a) = 1/(e^{-a} + 1)$, определенную на отрезке $[0,1]$, так и ее антисимметричный аналог для отрезка $[-1,1]$, например: $f(a) = \tanh(a)$. При этом кодировка выходных переменных из обучающей выборки будет либо $\{0,1\}$, либо $\{-1,1\}$. Выбор того или иного варианта никак не скажется на обучении.

В случае со входными переменными дело обстоит по-другому: обучение весов нижнего слоя сети определяется *непосредственно* значениями входов: на них умножаются невязки, зависящие от выходов. Между тем, если с точки зрения операции умножения значения ± 1 равноправны, между 0 и 1 имеется существенная асимметрия: нулевые значения не дают никакого вклада в градиент ошибки. Таким образом, выбор схемы кодирования входов влияет на процесс обучения. В силу логической равноправности обоих значений входов, более предпочтительной выглядит симметричная кодировка: $\{-1,1\}$, сохраняющая это равноправие в процессе обучения.

Нормировка и предобработка данных

Как входами, так и выходами нейросети могут быть совершенно разнородные величины. Очевидно, что результаты нейросетевого моделирования не должны зависеть от единиц измерения этих величин. А именно, чтобы сеть трактовала их значения единообразно, все входные и выходные величины должны быть приведены к единому - единичному - масштабу. Кроме того, для повышения скорости и качества обучения полезно провести дополнительную предобработку данных, выравнивающую распределение значений еще до этапа обучения.

Индивидуальная нормировка данных

Приведение данных к единичному масштабу обеспечивается нормировкой каждой переменной на диапазон разброса ее значений. В простейшем варианте это - линейное преобразование:

$$\tilde{x}_i = \frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}}$$

в единичный отрезок: $\tilde{x}_i \in [0,1]$. Обобщение для отображения данных в интервал $[-1,1]$, рекомендуемого для входных данных тривиально.

Линейная нормировка оптимальна, когда значения переменной x_i плотно заполняют определенный интервал. Но подобный "прямолинейный" подход применим далеко не всегда. Так, если в данных имеются относительно редкие выбросы, намного превышающие типичный разброс, именно эти выбросы определят согласно предыдущей формуле масштаб нормировки. Это приведет к тому, что основная масса значений нормированной переменной $\widetilde{x}_i$ сосредоточится вблизи нуля: $|\widetilde{x}_i| << 1$.

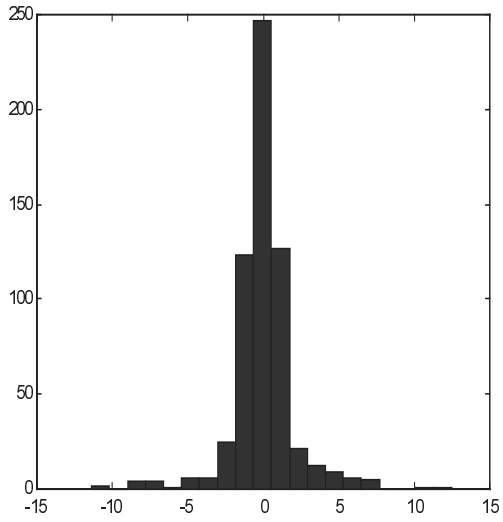

Рисунок 2. Гистограмма значений переменной при наличии редких, но больших по амплитуде отклонений от среднего

Гораздо надежнее, поэтому, ориентироваться при нормировке не на экстремальные значения, а на типичные, т.е. статистические характеристики данных, такие как среднее и дисперсия[3]:

$$\widetilde{x}_i = \frac{x_i - \bar{x}_i}{\sigma_i}, \qquad \bar{x}_i \equiv \frac{1}{P}\sum_{\alpha=1}^{P} x_i^{\alpha}, \quad \sigma_i^2 \equiv \frac{1}{P-1}\sum_{\alpha=1}^{P}\left(x_i^{\alpha} - \bar{x}_i\right)^2 .$$

В этом случае основная масса данных будет иметь единичный масштаб, т.е. типичные значения всех переменных будут сравнимы (см. Рисунок 2).

Однако, теперь нормированные величины не принадлежат гарантированно единичному интервалу, более того, максимальный разброс значений $\widetilde{x}_i$ заранее не известен. Для входных данных это может быть и не важно, но выходные переменные будут использоваться в качестве эталонов для выходных нейронов. В случае, если выходные нейроны - сигмоидные, они могут принимать значения лишь в единичном диапазоне. Чтобы установить соответствие между обучающей выборкой и нейросетью в этом случае необходимо ограничить диапазон изменения переменных.

Линейное преобразование, как мы убедились, неспособно отнормировать основную массу данных и одновременно ограничить диапазон возможных значений этих данных. Естественный

[3] Â âûðàæåíèè äëÿ äèñïåðñèè çíàìåíàòåëü ðàâåí $(P\text{-}1)$, à íå P èç-çà ôëóêòóàöèé *îöåíêè* ñðåäíåãî $\bar{x}_i$

выход из этой ситуации - использовать для предобработки данных функцию активации тех же нейронов. Например, нелинейное преобразование

$$\widetilde{x}_i = f\left(\frac{x_i - \bar{x}_i}{\sigma_i}\right), \qquad f(a) = \frac{1}{1+e^{-a}}$$

нормирует основную массу данных одновременно гарантируя, что $\widetilde{x}_i \in [0, 1]$ (см. Рисунок 3).

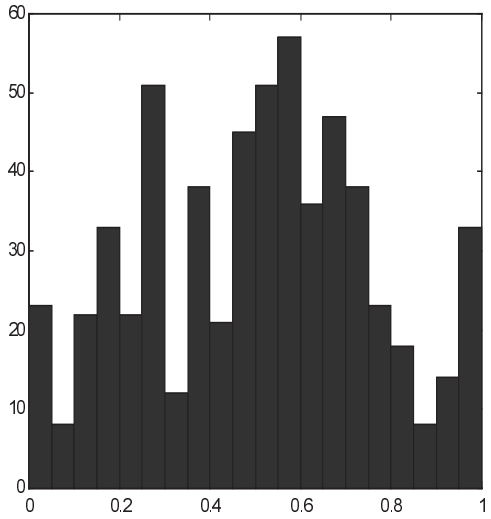

Рисунок 3. Нелинейная нормировка, использующая логистическую функцию активации $f(a) = \left(1+e^{-a}\right)^{-1}$

Как видно из приведенного выше рисунка, распределение значений после такого нелинейного преобразования гораздо ближе к равномерному.

До сих пор мы старались максимизировать энтропию каждого входа (выхода) по отдельности. Но, вообще говоря, можно добиться гораздо большего максимизируя их *совместную* энтропию. Рассмотрим эту технику на примере совместной нормировки входов, подразумевая, что с таким же успехом ее можно применять и для выходов а также для всей совокупности входов-выходов.

Совместная нормировка: *выбеливание* входов

Если два входа статистически не независимы, то их совместная энтропия меньше суммы индивидуальных энтропий: $H\left(\widetilde{x}_i \widetilde{x}_j\right) \le H\left(\widetilde{x}_i\right) + H\left(\widetilde{x}_i\right)$. Поэтому добившись статистической независимости входов мы, тем самым, повысим информационную насыщенность входной информации. Это, однако, потребует более сложной процедуры совместной нормировки входов.

Вместо того, чтобы использовать для нормировки индивидуальные дисперсии, будем рассматривать входные данные в совокупности. Мы хотим найти такое линейное преобразование, которое максимизировало бы их совместную энтропию. Для упрощения

задачи вместо более сложного условия статистической независимости потребуем, чтобы новые входы после такого преобразования были декоррелированы[4]. Для этого рассчитаем средний вектор и ковариационную матрицу данных по формулам:

$$\bar{\mathbf{x}} \equiv \frac{1}{P}\sum_{\alpha=1}^{P}\mathbf{x}^{\alpha}, \quad \Sigma_{ij}^{X} \equiv \frac{1}{P-1}\sum_{\alpha=1}^{P}\left(x_i^{\alpha}-\bar{x}_i\right)\left(x_j^{\alpha}-\bar{x}_j\right)$$

Затем найдем линейное преобразование, диагонализующее ковариационную матрицу. Соответствующая матрица составлена из столбцов - собственных векторов ковариационной матрицы:

$$\sum_j \Sigma_{ij}^{X} U_{jk} = \lambda_k U_{ik}$$

Легко убедиться, что линейное преобразование, называемое *выбеливанием* (whitening)

$$\tilde{x}_i = \left(x_k - \bar{x}_k\right) U_{ki} \Big/ \sqrt{\lambda_i}$$

превратит все входы в некоррелированные величины с нулевым средним и единичной дисперсией.

Если входные данные представляют собой многомерный эллипсоид, то графически выбеливание выглядит как растяжение этого эллипсоида по его главным осям (Рисунок 4).

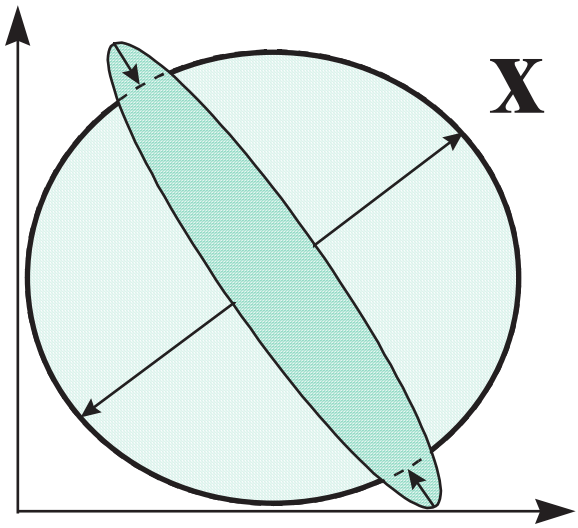

Рисунок 4. Выбеливание входной информации: повышение информативности входов за счет выравнивания функции распределения

Очевидно, такое преобразование увеличивает совместную энтропию входов, т.к. оно выравнивает распределение данных в обучающей выборке.

---

[4] Äåêîððåëèðîâàííûå âåëè÷èíû x_i èìåþò íóëåâûå êðîññ-êîððåëÿöèè: $\overline{\left(x_i-\bar{x}_i\right)\left(x_j-\bar{x}_j\right)} = 0, \quad \forall i \neq j$