Понижение размерности входов

Сильной стороной нейроанализа является возможность получения предсказаний при минимуме априорных знаний. Поскольку заранее обычно неизвестно насколько полезны те или иные входные переменные для предсказания значений выходов, возникает соблазн увеличивать число входных параметров, в надежде на то, что сеть сама определит какие из них наиболее значимы. Однако, как это уже обсуждалось в Главе 3, сложность обучения персептронов быстро возрастает с ростом числа входов (а именно - как куб размерности входных данных $C \propto d^3$). Еще важнее, что с увеличением числа входов страдает и точность предсказаний, т.к. увеличение числа весов в сети снижает предсказательную способность последней (согласно предыдущим оценкам: $\varepsilon \geq \sqrt{d/P}$).

Таким образом, количество входов приходится довольно жестко лимитировать, и выбор наиболее информативных входных переменных представляет важный этап подготовки данных для обучения нейросетей. Глава 4 специально посвящена использованию для этой цели самих нейросетей, обучаемых без учителя. Не стоит, однако, пренебрегать и традиционными, более простыми и зачастую весьма эффективными методами линейной алгебры.

Один из наиболее простых и распространенных методов понижения размерности - использование главных компонент входных векторов. Этот метод позволяет не отбрасывая конкретные входы учитывать лишь наиболее значимые комбинации их значений.

Понижение размерности входов методом главных компонент

Собственные числа матрицы ковариаций λ_i, фигурировавшие в предыдущем разделе, являются квадратами дисперсий вдоль ее главных осей. Если между входами существует линейная зависимость, некоторые из этих собственных чисел стремятся к нулю. Таким образом, наличие малых λ_i свидетельствует о том, что реальная размерность входных данных объективно ниже, чем число входов. Можно задаться некоторым пороговым значением ε и ограничиться лишь теми главными компонентами, которые имеют $\lambda \geq \varepsilon \lambda_{\max}$. Тем самым, достигается понижение размерности входов, при минимальных потерях точности представления входной информации.

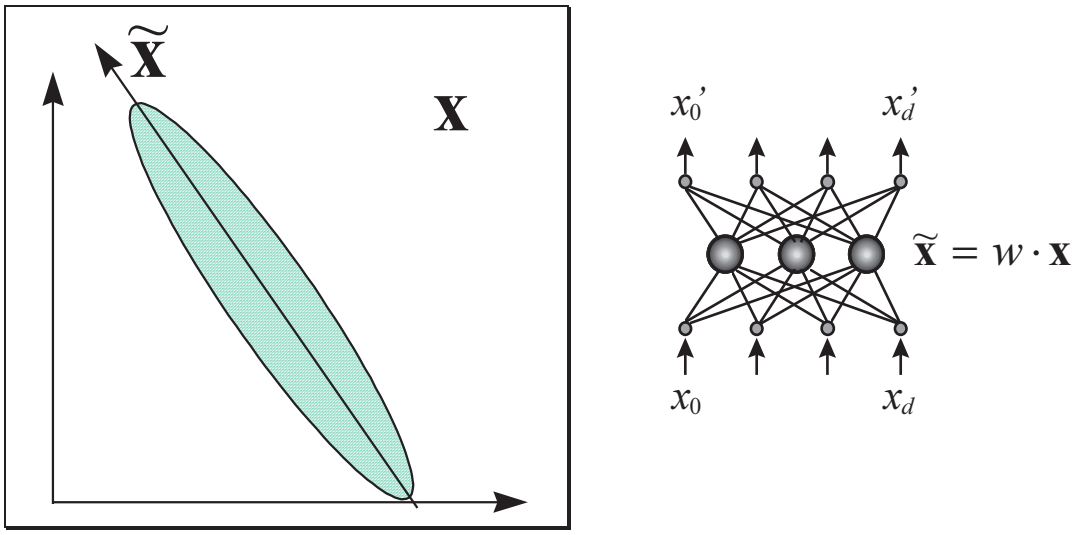

Рисунок 5. Понижение размерности входов методом главных компонент.

Восстановление пропущенных компонент данных

Главные компоненты оказываются удобным инструментом и для восстановления пропусков во входных данных. Действительно, метод главных компонент дает наилучшее линейное приближение входных данных меньшим числом компонент: $\tilde{\mathbf{x}} = w\mathbf{x}$ (Здесь мы, как и прежде, для учета постоянного члена включаем фиктивную нулевую компоненту входов, всегда равную единице - см. Рисунок 5, где справа показана нейросетевая интерпретация метода главных компонент. Таким образом, w - это матрица размерности $N \times (d+1)$). Восстановленные по N главным компонентам данные из обучающей выборки $\mathbf{x}'^{\alpha} = w^T \tilde{\mathbf{x}} = w^T w \mathbf{x}^{\alpha}$ имеют наименьшее среднеквадратичное отклонение от своих прототипов $\mathbf{x}^{\alpha}$. Иными словами, при отсутствии у входного вектора k компонент, наиболее вероятное положение этого вектора - на гиперплоскости первых $N = (d-k)$ главных компонент. Таким образом, для восстановленного вектора имеем: $\mathbf{x}'^{\alpha} = w^T \tilde{\mathbf{x}} = w^T w \mathbf{x}'^{\alpha}$, причем для известных компонент $\mathbf{x}'^{\alpha} = \mathbf{x}^{\alpha}$.

Пусть, например, у вектора $\mathbf{x}^{\alpha}$ неизвестна всего одна, k-я координата. Ее значение находится из оставшихся по формуле:

$$x'^{\alpha}_{k} = \left[w^T w\right]_{ki} x_i^{\alpha} \Big/ \left(1 - \left[w^T w\right]_{kk}\right),$$

где в числителе учитываются лишь известные компоненты входного вектора $\mathbf{x}^{\alpha}$.

В общем случае восстановить неизвестные компоненты (с индексами из множества K) можно с помощью следующей итеративной процедуры (см. Рисунок 6):

$$x_i^{n+1} = x_i^{\alpha}, \qquad i \notin K$$
$$x_i^{n+1} \to w_{ki} w_{kj} x_j^{n}, \quad i \in K$$

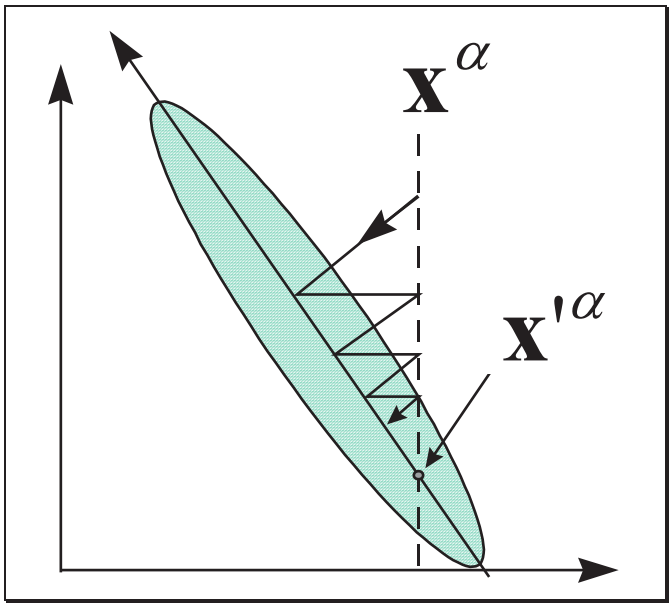

Рисунок 6. Восстановление пропущенных значения с помощью главных компонент. Пунктир - возможные значения исходного вектора с $k = 1$ неизвестными координатами. Наиболее вероятное его значение - на пересечении с $N = (d - k) = 2 - 1 = 1$ первыми главными компонентами

Понижение размерности входов с помощью нейросетей

Для более глубокой предобработки входов можно использовать все богатство алгоритмов самообучающихся нейросетей, о которых шла речь ранее. В частности, для оптимального понижения размерности входов можно воспользоваться методом *нелинейных главных компонент* (см. Рисунок 7).

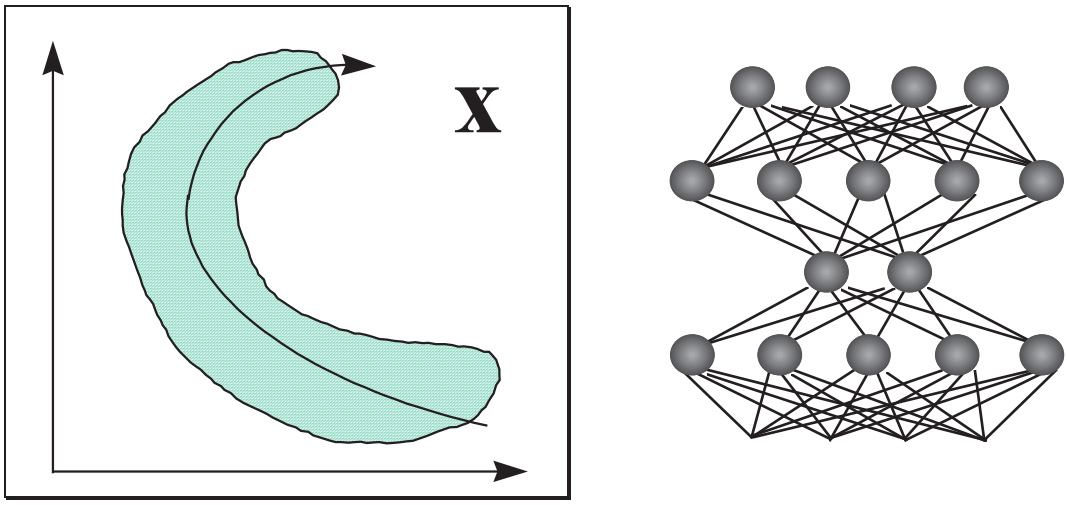

Рисунок 7. Понижение размерности входов методом нелинейных главных компонент

Такие сети с узким горлом также можно использовать для восстановления пропущенных значений - с помощью итерационной процедуры, обобщающей линейный вариант метода главных компонент (см. Рисунок 8).

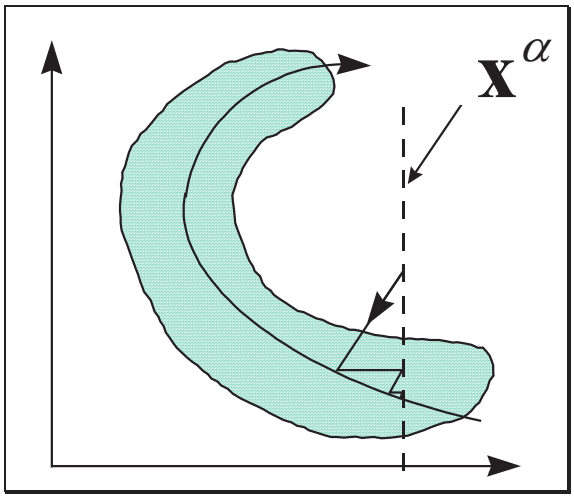

Рисунок 8. Восстановление пропущенных компонент данных с помощью нелинейных главных компонент

Однако, такую глубокую "предобработку" уже можно считать самостоятельной нейросетевой задачей. И мы не будем дале углубляться в этот вопрос.

Квантование входов

Более распространенный вид нейросетевой предобработки данных - *квантование входов*, использующее слой соревновательных нейронов (см. Рисунок 9).

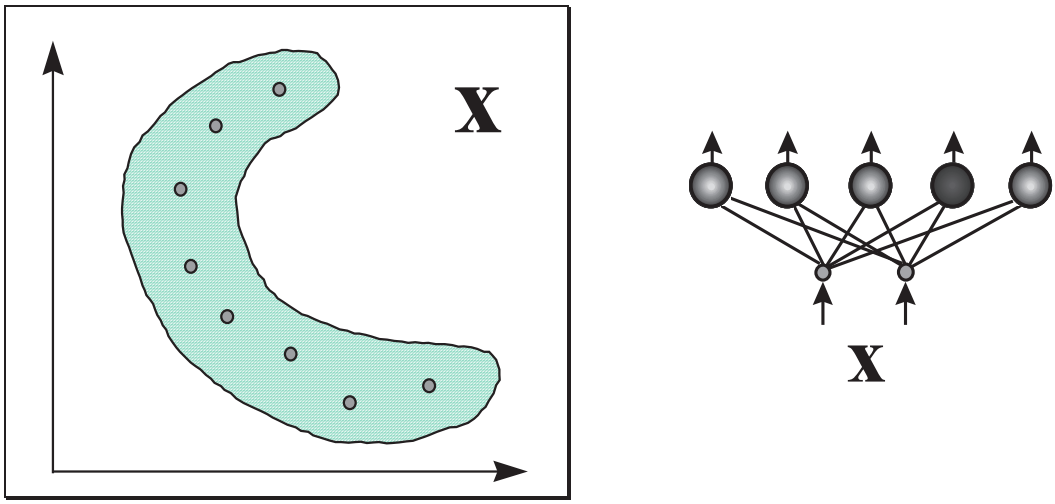

Рисунок 9. Понижение разнообразия входов методом квантования (кластеризации)

Нейрон-победитель является прототипом ближайших к нему входных векторов. Квантование входов обычно не сокращает, а наоборот, существенно увеличивает число входных переменных. Поэтому его используют в сочетании с простейшим линейным дискриминатором - однослойным персептороном. Получающаяся в итоге гибридная нейросеть, предложенная Нехт-Нильсеном в 1987 году, обучается послойно: сначала соревновательный слой кластеризует входы, затем выходным весам присваиваются значения выходной функции, соответствующие данному кластеру. Такие сети позволяют относительно быстро получать грубое - кусочно-постоянное - приближение аппроксимируемой функции (см. Рисунок 10).

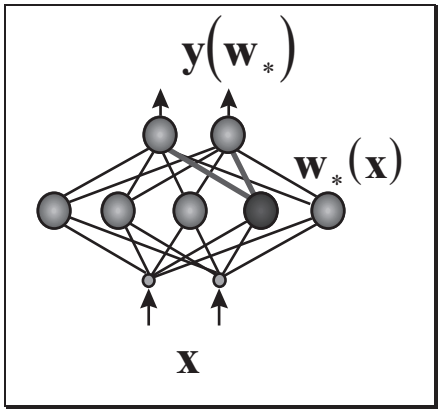

Рисунок 10. Гибридная сеть с соревновательным слоем, дающая кусочно-постоянное приближение функций

Особенно полезны кластеризующие сети для восстановления пропусков в массиве обучающих данных. Поскольку работа соревновательного слоя основана на сравнении расстояний между данными и прототипами, осутствие у входного вектора $\mathbf{x}^{\alpha}$ некоторых компонент не препятствует нахождению прототипа-победителя: сравнение ведется по оставшимся компонентам $i \notin K$:

$$\left|\mathbf{x}^{\alpha} - \mathbf{w}_k\right| = \sqrt{\sum_{i \notin K} \left(x_i^{\alpha} - w_{ki}\right)^2} .$$

При этом все прототипы $\mathbf{w}_k$ находятся в одинаковом положении. Рисунок 11 иллюстрирует эту ситуацию.

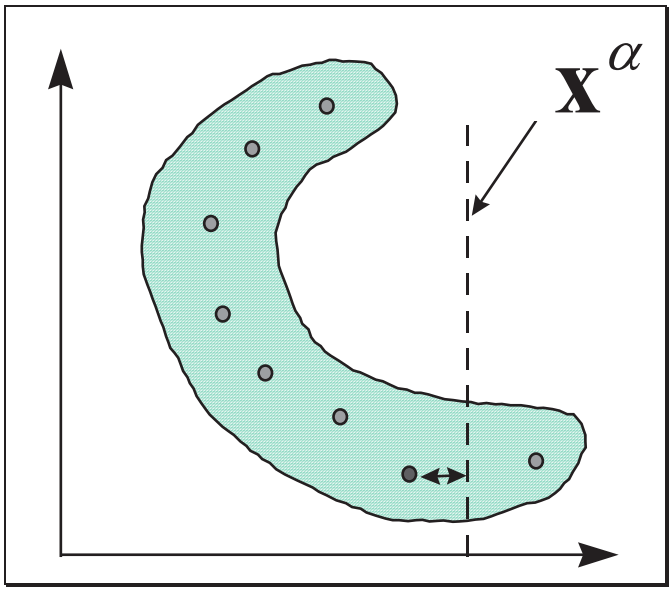

Рисунок 11. Наличие пропущенных компонент не препятствует нахождению ближайшего прототипа по оставшимся компонентам входного вектора $\mathbf{x}^{\alpha}$

Таким образом, слой квантующих входные данные нейронов нечувствителен к пропущенным компонентам, и может служить “защитным экраном” для минимизации последствий от наличия пропусков в обучающей базе данных.

Отбор наиболее значимых входов

До сих пор мы старались лишь представить имеющуюся входную информацию наилучшим - наиболее информативным - образом. Однако, рассмотренные выше методы предобработки входов никак не учитывали зависимость выходов от этих входов. Между тем, наша задача как раз и состоит в выборе входных переменных, наиболее значимых для предсказаний. Для такого более содержательного отбора входов нам потребуются методы, позволяющие оценивать *значимость* входов.

Линейная значимость входов

Легче всего оценить значимость входов в линейной модели, предполагающей линейную зависимость выходов от входов:

$$\left(y_j^{\alpha} - \bar{y}_j\right) = \sum_k w_{jk}\left(x_k^{\alpha} - \bar{x}_k\right)$$