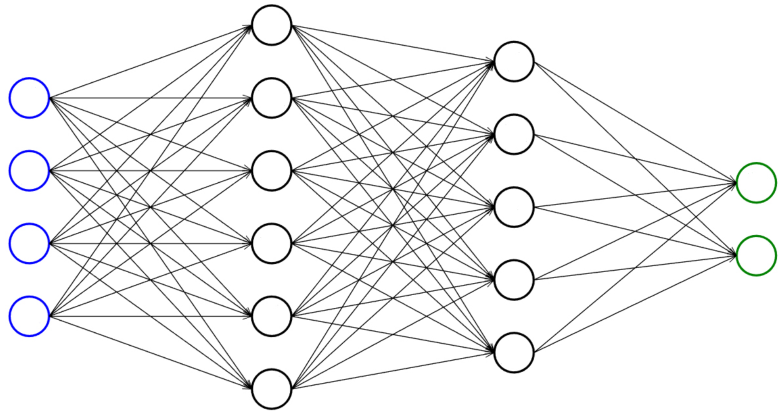
Математика для Data Science. Линейная алгебра. Шпаргалка

Содержание

Вторая неделя. Пространства, базисы и ранг

Нейронные сети



Кружки на картинках — это *нейроны*. Слой нейронов, который выделен синим на картинке, называют *входным, распределительным* или *нулевым*. Числа, принимаемые на вход, будут называться *состояниями* нейронов нулевого слоя.

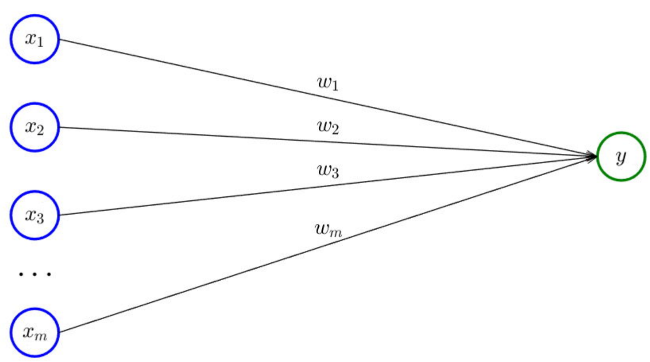
Каждой *связи*, то есть каждому ребру между нейронами, соответствует некоторое преобразование (функция) — из числа в число. Слои в сети нумеруются слева направо и по состоянию нейронов предыдущего слоя с помощью связей мы вычисляем состояние нейронов следующего слоя. В результате, каждому нейрону будет присвоено *состояние* (число). Состояние нейронов последнего слоя — это *выход* нейронной сети.

Каждая нейросеть — это пара:

1. **Архитектура** — количество слоев, число нейронов на каждом слое, описание типов связей между нейронами. На картинке сверху нарисована именно архитектура нейронной сети (правда, без описания типов связей)

2. **Веса.** Каждой связи соответствует функция из числа в число. Архитектура задает только тип таких функций. То есть некоторый класс функций, которые могут быть использованы: например, линейные функции. У этих функций есть параметры, также называемые весами.

Рассмотрим простейшую нейросеть: входной слой $x_1, x_2, \ldots, x_m$ и один нейрон y на выходном слое.

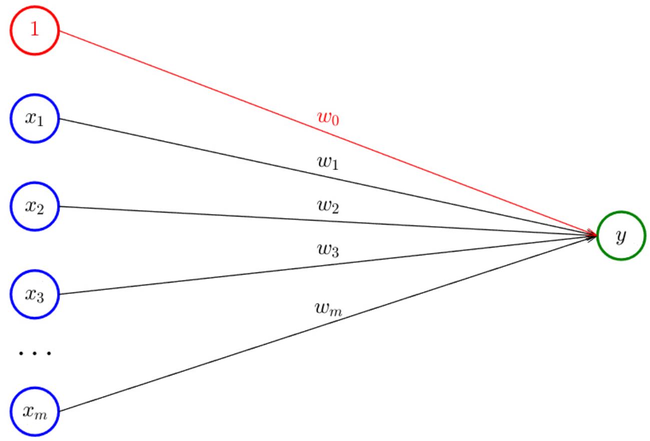


Нейросеть с *линейной связью* работает следующим образом: $y = w_1x_1 + w_2x_2 + \cdots + w_mx_m = \sum_{i=1}^{m} w_ix_i$. Или, то же самое через умножение матриц:

$$y = \begin{pmatrix} w_1 & w_2 & \ldots & w_m \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}$$

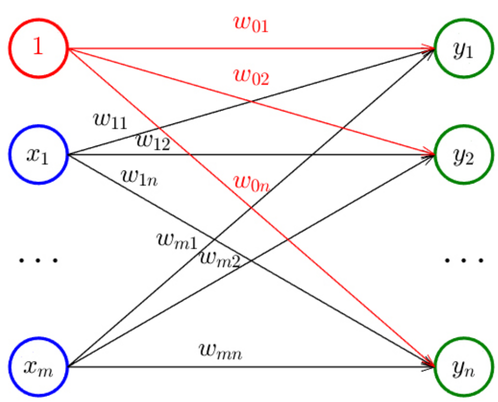
Веса в данном случае — это значения коэффициентов $w_1, w_2, \ldots, w_m \in \mathbb{R}$.

Для более точных предсказаний при вычислении состояния каждого нейрона мы будем добавлять ещё *сдвиг* (*bias*) — нейрон с постоянным единичным состоянием.



Получим $y = \begin{pmatrix} w_0 & w_1 & w_2 & \ldots & w_m \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = w_0 + \sum\limits_{i=1}^{m} w_i x_i$

Однослойная нейронная сеть



Пусть $x_1, x_2, \ldots, x_m$ — вход нейронной $y_1, y_2, \ldots, y_n$ — состояния первого слоя (линейного слоя), он же выход, w_{ij} — вес на связи между i-ым нейроном входа и j-ым нейроном выхода. Тогда $\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} w_{01} + \sum\limits_{i=1}^{m} w_{i1} x_i \\ w_{02} + \sum\limits_{i=1}^{m} w_{i2} x_i \\ \vdots \\ w_{0n} + \sum\limits_{i=1}^{m} w_{in} x_i \end{pmatrix}$

Пусть A — матрица $n \times m$. *Транспонированной* матрицей для A будет матрица B размеров $m \times n$ такая, что $A_{ij} = B_{ji}$ для всех $1 \leqslant i \leqslant n$ и $1 \leqslant j \leqslant m$. Транспонированная матрица обозначается A^T.

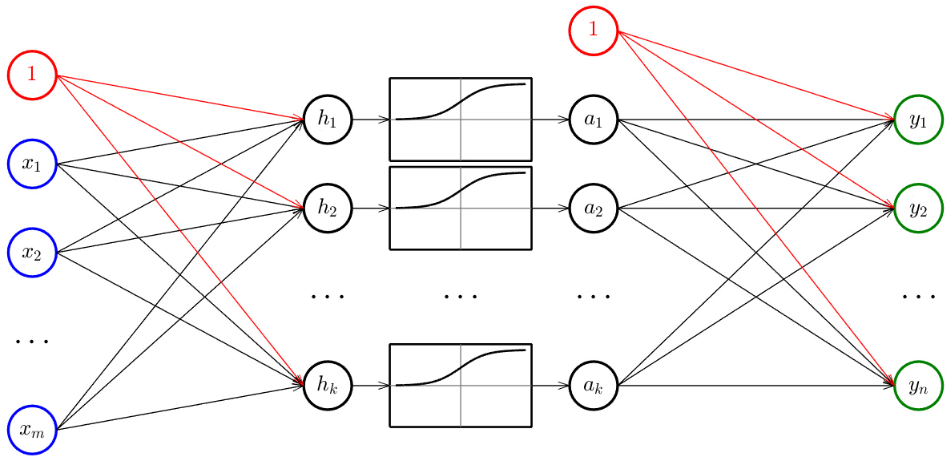
Таким образом, обозначив $W := \begin{pmatrix} w_{01} & w_{02} & \ldots & w_{0n} \\ w_{11} & w_{12} & \ldots & w_{1n} \\ w_{21} & w_{22} & \ldots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & \ldots & w_{mn} \end{pmatrix}$, получаем формулу $\vec{y} = W^T \vec{x}$, где $\vec{x} = (x_1, x_2, \ldots, x_m)^T$

и $\vec{y} = (y_1, y_2, \ldots, y_m)^T$.

Нейросети называются *эквивалентными* в случае, если для любого входного вектора их выход одинаков.

Утверждение. N подряд идущих линейных слоев эквивалентны одному линейному слою.

Поэтому между линейными слоями применяют нелинейную функцию к состояниям нейронов — *функцию активации*.



Обозначим $\vec{x}' = (1, x_1, x_2, \ldots, x_m)^T$ — входной вектор расширенный нейроном сдвига,

$\vec{h} = (h_1, h_2, \ldots, h_k)^T$ — вектор состояний скрытого слоя

$\vec{a} = (a_1, a_2, \ldots, a_k)^T$ — вектор состояний скрытого слоя после применения активации (способ нахождения a_i описан ниже)

$\vec{a}' = (1, a_1, a_2, \ldots, a_k)^T$ — вектор $\vec{a}$ расширенный нейроном сдвига

$\vec{y} = (y_1, y_2, \ldots, y_n)^T$ — выход нейросети

W — матрица весов первого линейного слоя, ее размер $k \times (m+1)$

U — матрица весов второго линейного слоя, её размер $n \times (k+1)$

f — функция активации, применяемая между первым и вторым линейным слоем

Нейросеть с одним скрытым слоем и активацией работает следующим образом:

- На основе входа расширенного нейроном сдвига вычисляются состояния скрытого слоя: $\vec{h} = W^T \vec{x}'$
- К каждой компоненте вектора $\vec{h}$ применяется функция активации f — получаем вектор $\vec{a}$: $a_i = f(h_i)$
- По активированным состояниям скрытого слоя с нейроном сдвига вычисляется выход нейросети: $\vec{y} = U^T \vec{a}'$

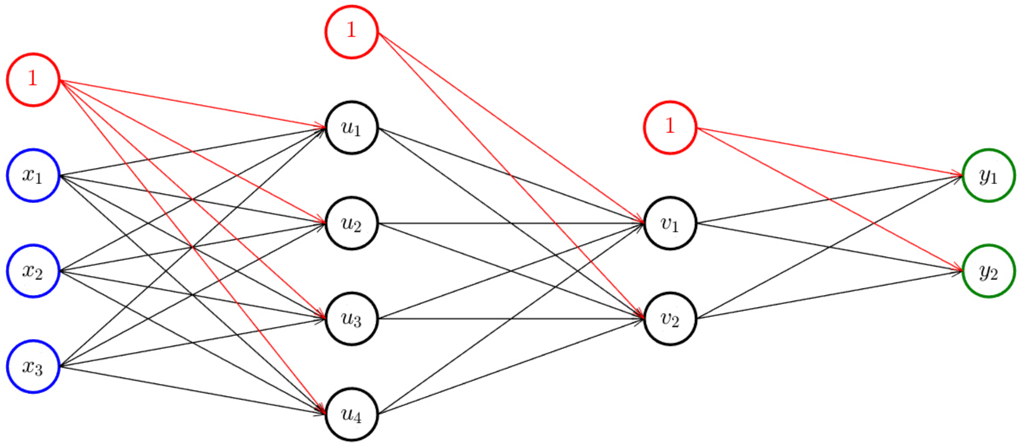
Перцептрон — это нейросеть с одним скрытым слоем, первый и второй слой линейные, между ними функция активации, выход один.

Теорема. Любая непрерывная функция многих переменных может быть приближена перцептроном с сигмоидальной функцией активации с любой наперед заданной точностью. Пример сигмоидальной функции это $f(x) = \frac{1}{1+e^{-x}}$.

Примеры функций активации

1. $\mathrm{ReLU} = \max(0, x)$
2. $\text{Leaky ReLU} = \begin{cases} x, & \text{если } x > 0 \\ 0.01x, & \text{иначе} \end{cases}$
3. $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ — гиперболический тангенс
4. $\sigma(x) = \frac{1}{1+e^{-x}}$ сигмоидальная (логистическая) функция активации
5. $\mathrm{Softmax}(z_1, z_2, \ldots, z_n) = \left(\frac{e^{z_1}}{\sum\limits_{i=1}^{n} e^{z_i}}, \frac{e^{z_2}}{\sum\limits_{i=1}^{n} e^{z_i}}, \ldots, \frac{e^{z_n}}{\sum\limits_{i=1}^{n} e^{z_i}} \right)$

Многослойные нейронные сети строятся по тем же принципам, что мы уже разобрали: последовательное умножение на матрицы и применение функций активации. Все слои, кроме нулевого и последнего называются скрытыми слоями.



Подпространства и линейные комбинации

Линейной комбинацией векторов $\vec{a}_1, \ldots, \vec{a}_k$ называется любой вектор вида $\lambda_1\vec{a}_1 + \lambda_2\vec{a}_2 + \cdots + \lambda_k\vec{a}_k$, где $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$.

Множество V всех линейных комбинаций векторов $\vec{a}_1, \ldots, \vec{a}_k \in \mathbb{R}^n$ называется линейным подпространством, *порождённым* векторами $\vec{a}_1, \ldots, \vec{a}_k$. Набор векторов $\vec{a}_1, \ldots, \vec{a}_k$ называется *порождающим*.

Комментарий. Слово "линейное" часто опускается, так что иногда мы будем писать просто "подпространство".

Нестрогое определение. *Векторное пространство* это множество V с определёнными на нём операциями сложения и умножения на число, такими что

1. Если $\vec{x} \in V$, $\vec{y} \in V$, то и $\vec{x} + \vec{y} \in V$.
2. Если $\vec{x} \in V$ и $c \in \mathbb{R}$, то $c\vec{x} \in V$.

Если V — линейное подпространство в $\mathbb{R}^n$, порождённое векторами $\vec{a}_1, \ldots, \vec{a}_k$, то оно является векторным пространством.

Линейная зависимость

Линейные комбинации, в которых все коэффициенты равны нулю, называются *тривиальными*.

Набор векторов $\vec{a}_1, \ldots, \vec{a}_k$ называется *линейно зависимым*, если $\vec{0}$ можно представить в виде нетривиальной линейной комбинации векторов из этого набора.

Другими словами, набор векторов $\vec{a}_1, \ldots, \vec{a}_k$ называется линейно зависимым, если существуют такие числа $\lambda_1, \ldots, \lambda_k$, что

- $\vec{0} = \lambda_1\vec{a}_1 + \cdots + \lambda_k\vec{a}_k$
- хотя бы одно из чисел $\lambda_1, \ldots, \lambda_k$ не равно нулю

Наборы, которые не являются линейно зависимыми, называют *линейно независимыми*.

Утверждение. Если набор линейно зависим, то хотя бы один из его векторов представляется в виде линейной комбинации оставшихся векторов. Другими словами, хотя бы один из векторов набора можно выразить через другие.

Обратное утверждение тоже верно: если один из векторов набора выражается через другие, то набор линейно зависим.

Утверждение "V порождено векторами $\vec{a}_1, \ldots, \vec{a}_k$" записывают так: $V = \mathrm{Span}\{\vec{a}_1, \ldots, \vec{a}_k\}$. То же самое можно сказать так: V — *линейная оболочка* векторов $\vec{a}_1, \ldots, \vec{a}_k$.

Свойства линейной оболочки векторов

1. $\mathrm{Span}\{\vec{a}_1\} = \mathrm{Span}\{\vec{a}_1, c\vec{a}_1\}$, где c – любое число

2. $\mathrm{Span}\{\vec{a}_1, \vec{a}_2, \dots, \vec{a}_k\} = \mathrm{Span}\{\vec{a}_1, \vec{a}_2, \dots, \vec{a}_k, c_1\vec{a}_1 + c_2\vec{a}_2 + \dots + c_k\vec{a}_k\}$, где $c_1, \dots, c_k$ – любые числа

3. $\mathrm{Span}\{\vec{a}_1, \vec{a}_2, \dots, \vec{a}_k\} = \mathrm{Span}\{\vec{a}_1 + c\vec{a}_2, \vec{a}_2, \dots, \vec{a}_k\}$, где c – любое число

Базис, размерность, ранг

Базис векторного пространства это упорядоченный набор векторов, такой что любой вектор этого пространства единственным способом представляется в виде линейной комбинации векторов из этого набора.

Пусть дан вектор $\vec{x} \in V$ и базис $\vec{a}_1, \dots, \vec{a}_k$ пространства V. Тогда $\vec{x} = \lambda_1\vec{a}_1 + \dots + \lambda_k\vec{a}_k$ для некоторых $\lambda_1, \dots, \lambda_k$. Упорядоченный набор чисел $(\lambda_1, \dots, \lambda_k)$ называется *координатами* вектора $\vec{x}$ в базисе $\vec{a}_1, \dots, \vec{a}_k$.

Набор векторов $\vec{e}_1, \vec{e}_2, \dots, \vec{e}_n$ называется *стандартным* базисом $\mathbb{R}^n$. Напомним, что $\vec{e}_i$ — это вектор, у которого стоит единица на i-ом месте, а все остальные координаты равны нулю.

Следующие три утверждения эквивалентны:

1. набор является базисом V

2. набор линейно независим и порождает V

3. набор порождает V и при удалении любого элемента из набора оставшиеся векторы не порождают V

Теорема. Все базисы векторного пространства содержат одно и то же число элементов.

Размерностью векторного пространства V называется число элементов в базисе V.

Ранг матрицы это максимальное число линейно независимых столбцов матрицы.

Выражение "ранг матрицы A равен k" записывают так: $\mathrm{rank}(A) = k$. Иногда можно встретить обозначение $\mathrm{rk}(A) = k$.

Утверждение. Ранг матрицы A равен размерности пространства, порождённого столбцами A.

Метод Гаусса

Пусть необходимо найти ранг данной матрицы. Будем построчно приводить её к виду, в котором легко найти ранг.

Введём два *элементарных преобразования* матрицы, которые не меняют ранг:

1. поменять местами два столбца матрицы

2. прибавить к одному из столбцов другой столбец, умноженный на число

Преобраование последней строки

Пусть в последней строке есть хотя бы одно ненулевое число. Применив 1-ое элементарное преобразование, переставим столбцы так, чтобы это ненулевое число стояло в последнем столбце. Тем самым, наша матрица сейчас выглядит так:

$$\begin{pmatrix} * & * & \dots & * & * \\ * & * & \dots & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ * & * & \dots & * & * \\ a_1 & a_2 & \dots & a_{n-1} & a_n \end{pmatrix}$$

где $a_n \neq 0$, а звёздочками обозначены числа, которые нам сейчас не интересны.

Для каждого $i < n$ вычтем из i-ого столбца последний столбец, умноженный на $\frac{a_i}{a_n}$. Получим такую матрицу:

$$\begin{pmatrix} * & * & \dots & * & * \\ * & * & \dots & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ * & * & \dots & * & * \\ (a_1 - \frac{a_1}{a_n}a_n) & (a_2 - \frac{a_2}{a_n}a_n) & \dots & (a_{n-1} - \frac{a_{n-1}}{a_n}a_n) & a_n \end{pmatrix} = \begin{pmatrix} * & * & \dots & * & * \\ * & * & \dots & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ * & * & \dots & * & * \\ 0 & 0 & \dots & 0 & a_n \end{pmatrix}$$

Комментарий. Если у изначальной матрицы последняя строка состояла только из нулей, то найдём первую снизу строку, в которой есть хотя бы одно ненулевое число и проведём с ней описанную процедуру.

Преобраование предпоследней строки

Далее, посмотрим на часть матрицы, составленную из всех столбцов, кроме последнего. Проведём с ней такую же процедуру, как и на прошлом шаге. А именно:

- переставим столбцы так, чтобы b_{n-1} был не равен нулю: $\begin{pmatrix} * & * & \dots & * & * & * \\ * & * & \dots & * & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ * & * & \dots & * & * & * \\ b_1 & b_2 & \dots & b_{n-2} & b_{n-1} & * \\ 0 & 0 & \dots & 0 & 0 & a_n \end{pmatrix}$
- вычитая предпоследний столбец из предыдущих, занулим все числа левее b_{n-1}: $\begin{pmatrix} * & * & \dots & * & * & * \\ * & * & \dots & * & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ * & * & \dots & * & * & * \\ 0 & 0 & \dots & 0 & b_{n-1} & * \\ 0 & 0 & \dots & 0 & 0 & a_n \end{pmatrix}$

Если встречались нулевые строки, то мы их пропускаем.

Преобраование остальных строк

Зафиксируем последние два столбца, и действуем, как и на предыдущих двух шагах. Продолжаем повторять, пока не закончатся строки или столбцы.

Утверждение. Ранг матрицы, полученной методом Гаусса, равен количеству ненулевых столбцов.

Кроме того, метод Гаусса позволяет, например, исследовать линейную зависимость, находить базис и размерность пространства.

Теорема. Максимальное число линейно независимых <u>столбцов</u> матрицы равно максимальному числу линейно независимых <u>строк</u> матрицы.

Следствие из теоремы. Элементарные преобразования в методе Гаусса можно применять не только к столбцам, но и к строкам.