

Topologia Botanica



28 мая 2020 г.

Содержание

I	Основные понятия	4
1	Внутренние законы композиции; коммутативность; ассоциативность	4
1.1	Внутренние законы композиций	4
1.2	Коммутативность законов композиции . .	5
1.3	Ассоциативность законов композиции . .	6
2	Цветы и бутоны; композиции бутонов; упрощения выражений	7
2.1	Цветы и бутоны	7
2.2	Композиции бутонов	10
2.3	Упрощения выражений	15
II	Техника дифференцирования цветов	17
3	Производная бутона; производная композиций	17
3.1	Производная бутона	17
3.2	Производная композиций	18
4	Производная сложной функции; полная производная; градиентный спуск	21
4.1	Производная сложной функции	21
4.2	Полная производная	22
4.3	Градиентный спуск	22

III	Альбом с гербариями	25
IV	Программная реализация	28
5	Введение	29
5.1	Постановка задачи	29
5.2	Распределения входных и выходных значений данных	30
6	Архитектура программы	33
6.1	Внешний модуль <i>app</i>	33
6.2	Расчётный модуль <i>settings</i>	34
6.3	Расчётный модуль <i>graph</i>	37
6.4	Расчётный модуль <i>machine</i>	38

Часть I

ОСНОВНЫЕ ПОНЯТИЯ

1 Внутренние законы композиции; коммутативность; ассоциативность

1.1 Внутренние законы композиций

Определение 1.1. Внутренним законом композиции элементов множества Σ называется отображение f некоторого подмножества A произведения $\Sigma \times \Sigma$ в Σ . Значение $f(x, y)$ отображения f при $(x, y) \in A$ называется композицией пары элементов x и y относительно этого закона. Если отображение возможно установить, то говорят, что внутренняя композиция пар элементов (x, y) определена на множестве Σ и $(x, y) \in \Sigma \times \Sigma$.

Для записи внутренних композиции x и y используется определенный порядок записи, обозначающий конкретный закон. Для обозначения характеристического знака произвольных законов композиций используется символ \top . При записи композиции, пара элементов x и y , называемых также операндами, окружают характеристический знак закона композиции с левой и правой сторон: $(x, y) \rightarrow x \top y$. В таблице 1 представлены основные используемые характеристические знаки с названием соответствующих законов.

Вид нотации первых четырех законов, представленных в таблице 1, выбран неслучайно: таким образом, подчеркивается их формальная принадлежность множе-

ству пространства графов Σ .

Таблица 1: Таблица соответствий используемых характеристических знаков законам композиции пар элементов

Характеристический знак	Наименование закона
\otimes	Произведение
\oslash	Деление
\oplus	Сложение
\ominus	Вычитание
\odot	Аргументация

Внутренней закон композиции $(x, y) \rightarrow x \top y$, где $\top \rightarrow \odot$ называется аргументацией. Данный закон композиции определяет действие операнда вида функционального оператора x , стоящего по левую сторону характеристического знака композиции \odot , на операнд y , стоящий по правую сторону характеристического знака композиции. Примерами закона композиции аргументации являются $\sin(x) = \sin \odot x$, либо $\exp(x^2) = \exp \odot x^2$ и так далее.

1.2 Коммутативность законов композиции

Определение 1.2. Закон композиции \top множества Σ называется коммутативным, если для произвольных элементов x, y из множества Σ определена, а сами элементы x и y являются перестановочными.

Определение 1.3. Элементы x и y из множества Σ называются перестановочными относительно композиции \top , если выполняется равенство

$$x \top y = y \top x, \quad (1)$$

а законы $x \top y$, $y \top x$ определены и принадлежат множеству Σ .

Утверждение 1.1. Законы композиции $(x, y) \rightarrow x \top y$, где $\top \in (\otimes, \oplus)$ являются всюду коммутативными.

Утверждение 1.2. Законы композиции $(x, y) \rightarrow x \top y$, где $\top \in (\oslash, \ominus, \odot)$ некоммутативные.

1.3 Ассоциативность законов композиции

Определение 1.4. Закон композиции $(x, y) \rightarrow x \top y$ называется ассоциативным, если для произвольных элементов x, y, z , выполняется равенство

$$(x \top y) \top z = x \top (y \top z). \quad (2)$$

Утверждение 1.3. Законы композиции $(x, y) \rightarrow x \top y$, где $\top \in (\otimes, \oplus)$ ассоциативные.

Утверждение 1.4. Законы композиции $(x, y) \rightarrow x \top y$, где $\top \in (\oslash, \ominus, \odot)$ неассоциативные.

2 Цветы и бутоны; композиции бутонов; упрощения выражений

2.1 Цветы и бутоны

Определение 2.1. Цветком \mathfrak{F}^s называется топология множества Σ , представляющая собой структуру, образованную заданием устойчивого множества \mathfrak{M} из подмножеств Σ и обладающую основными аксиомами топологических структур:

1. Объединение множеств из \mathfrak{M} есть множество \mathfrak{M} .
2. Пересечение конечного семейства множеств из \mathfrak{M} есть множество из \mathfrak{M} .

Определение 2.2. Топологическое пространство цветов \mathfrak{F} называется *лугом цветов* или просто *лугом* (*meadow*).

Определение 2.3. Бутоном \mathfrak{B} цветка \mathfrak{F}^s называется топология в множестве Σ , образующая подмножество луга \mathfrak{M} .

Определение 2.4. Количество бутонов s , образующих цветок \mathfrak{F}^s называется *пышностью* (*splendor*).

Бутоны формируются на луг \mathfrak{M} биекцией функциональной зависимости $f(x, y)$ из пространства Σ , состоящей из пары операндов x, y и закона композиции \top

между ними.

$$f(x, y) \xRightarrow{\mathfrak{M}} \mathfrak{B}_{f(x, y)}^{\top} = \left[\begin{array}{c} f(x, y) \\ \uparrow \\ \text{Т} \\ \swarrow \quad \searrow \\ x \quad y \end{array} \right]. \quad (3)$$

Графически **желтым** цветом выделяются операнды, называемые *лепестками*, а **зеленым** — композиция бутона $\mathfrak{B}_{f(x, y)}^{\top}$, также именуемая *цветоложем*. Стрелочка $[\rightarrow]$ определяет зависимость между элементами бутона и называется *стеблем*, стрелочка вида $[\Rightarrow]$ восходит к итоговой функциональной зависимости $f(x, y)$ и будет называться *корнем* цветка.

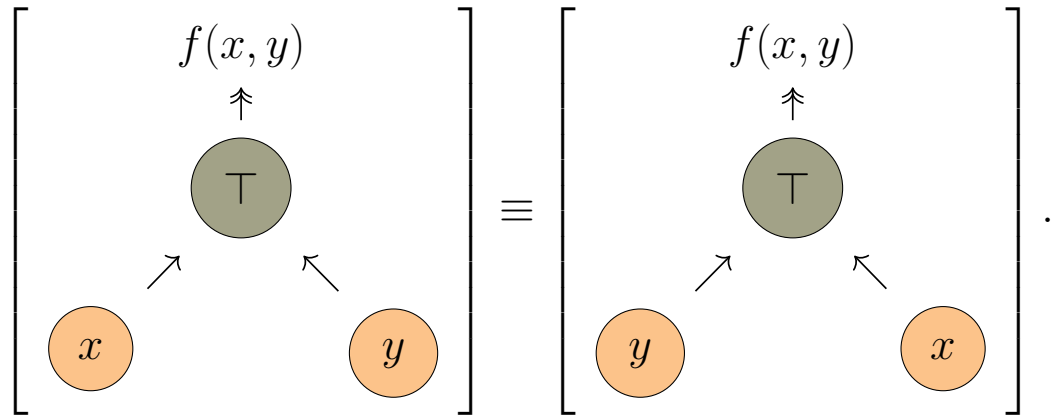
К примеру, бутон $\mathfrak{B}_{f(x, y)}^{\oplus}$ функции $f(x, y)$, состоящий из композиции закона сложения пары операндов x и y , будет иметь следующий вид

$$f(x, y) = x + y \xRightarrow{\mathfrak{M}} \mathfrak{B}_{f(x, y)}^{\oplus} = \left[\begin{array}{c} f(x, y) \\ \uparrow \\ \oplus \\ \swarrow \quad \searrow \\ x \quad y \end{array} \right].$$

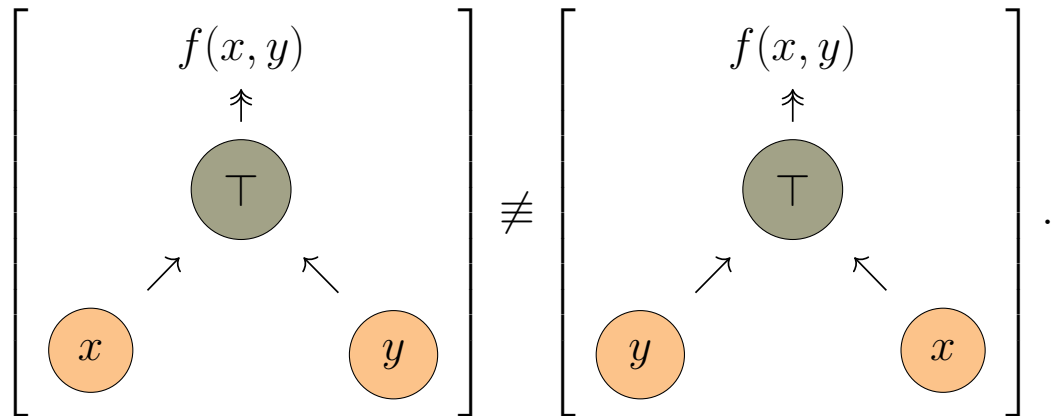
Следствие 2.1. В силу определения 1.2 и утверждения

1.1, коммутация закона композиции \top позволяет установить изоморфизм топологии бутона $\mathfrak{B}_{f(x,y)}^\top$.

Определение 2.5. Бутоны $\mathfrak{B}_{f(x,y)}^\top$ функциональной зависимости $f(x, y)$, отображающие коммутативный закон композиции \top для пар элементов x и y , называются изоморфными. Для изоморфных бутонов $\mathfrak{B}_{f(x,y)}^\top$ справедливо зеркальное отражение



Определение 2.6. Бутоны $\mathfrak{B}_{f(x,y)}^\top$ функциональной зависимости $f(x, y)$, отображающие некоммутативный закон композиции \top для пар элементов x и y , называются неизоморфными. Для неизоморфных бутонов $\mathfrak{B}_{f(x,y)}^\top$ зеркальное отражение невозможно



Определение 2.7. Цветок $\mathfrak{F}_{\top_{i=1}^{n-1} \vec{f}_i(x)}^{n-1}$, образованный групповой композицией $\top n$ вектор-функций $\vec{f}_1(x), \dots, \vec{f}_n(x)$ можно представить в виде

$$\mathfrak{F}_{\top_{i=1}^n \vec{f}_i(x)}^{n-1} = \mathfrak{B}_{(\mathfrak{F}_{\dots \mathfrak{B}_{\top}^{n-2}}(\mathfrak{F}_{\top_{i=1}^{n+k-1}(\vec{f}_i(x))}^{n+k-1}, \mathfrak{F}_{\vec{f}_{n+k+1}(x)}^1), \mathfrak{F}_{\vec{f}_n(x)}^1)}.$$

2.2 Композиции бутонов

Определение 2.8. Множество $s = n-1$ бутонов $\mathfrak{B}_{f_n(x_n, y_n)}^{\top_{f_n(x_n, y_n)}}$, где $n \in \mathbb{N}$, образованное от множества функциональных зависимостей $f_n(x_n, y_n)$ и композиций \top_{n-1} между ними, образует цветок \mathfrak{F}_f^s функциональной зависимости $f(x, y)$

$$f \xRightarrow{\mathfrak{M}} \mathfrak{F}_f^s$$

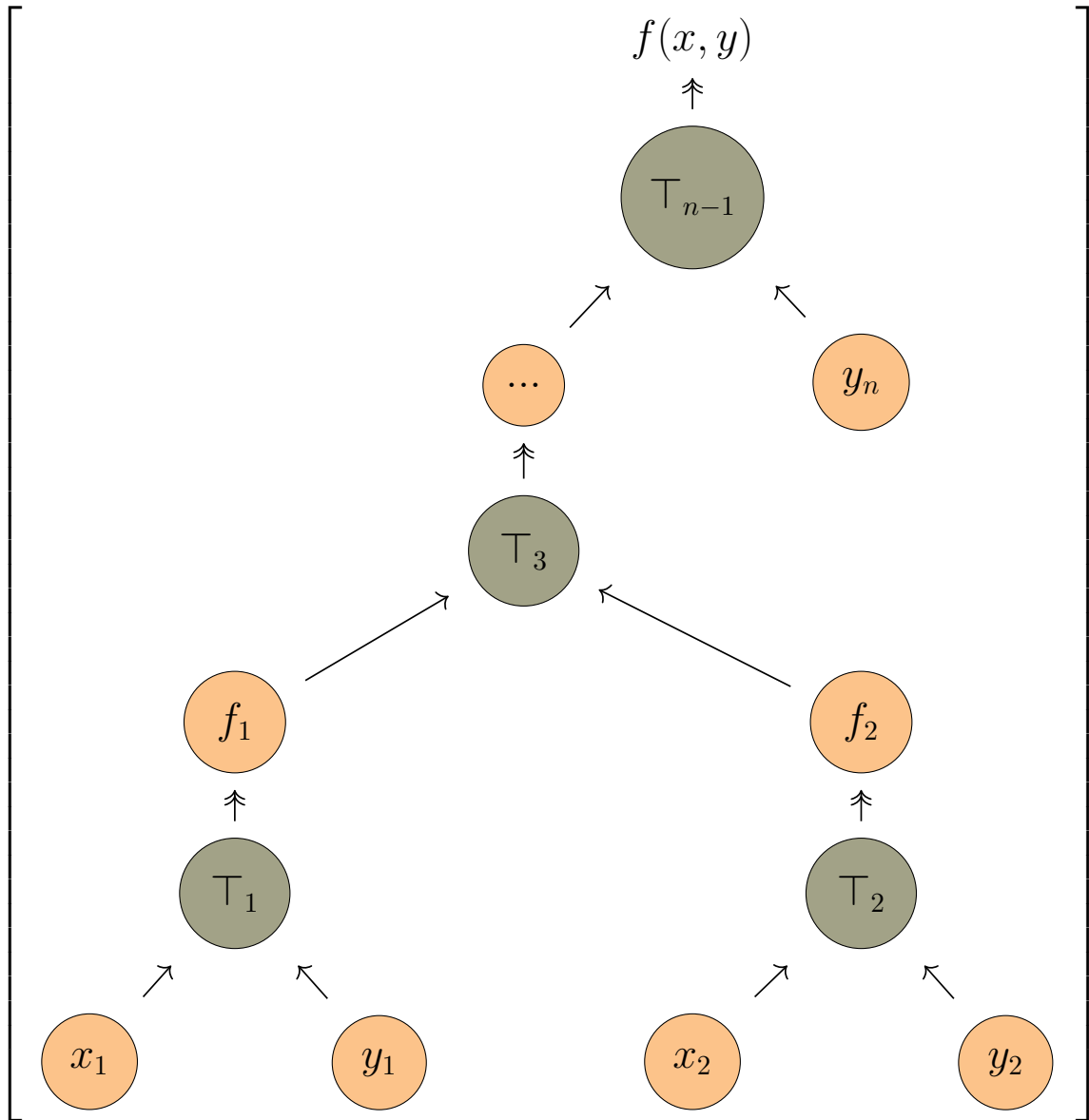
и, расписывая представления $f(x, y)$, получаем общий вид закона композиции цветка \mathfrak{F}_f^s

$$\begin{array}{c}
f_1(x_1, y_1) \top_1 \dots \top_{n-1} f_n(x_n, y_n) \\
\Longrightarrow \\
\mathfrak{M} \\
\mathfrak{B}^{\top_{n-1}} \\
(\mathfrak{B}^{\top_{n-2}} \\
(\mathfrak{B}^{\top_{n-3}} \\
(\mathfrak{B}^{\top_{n-4}} \\
(\dots \\
(\mathfrak{B}^{\top_2} \\
(\mathfrak{B}^{\top_1} \\
(f_1, f_2), f_3), f_4), f_{\dots}), f_{n-2}), f_{n-1}), f_n).
\end{array}
\tag{4}$$

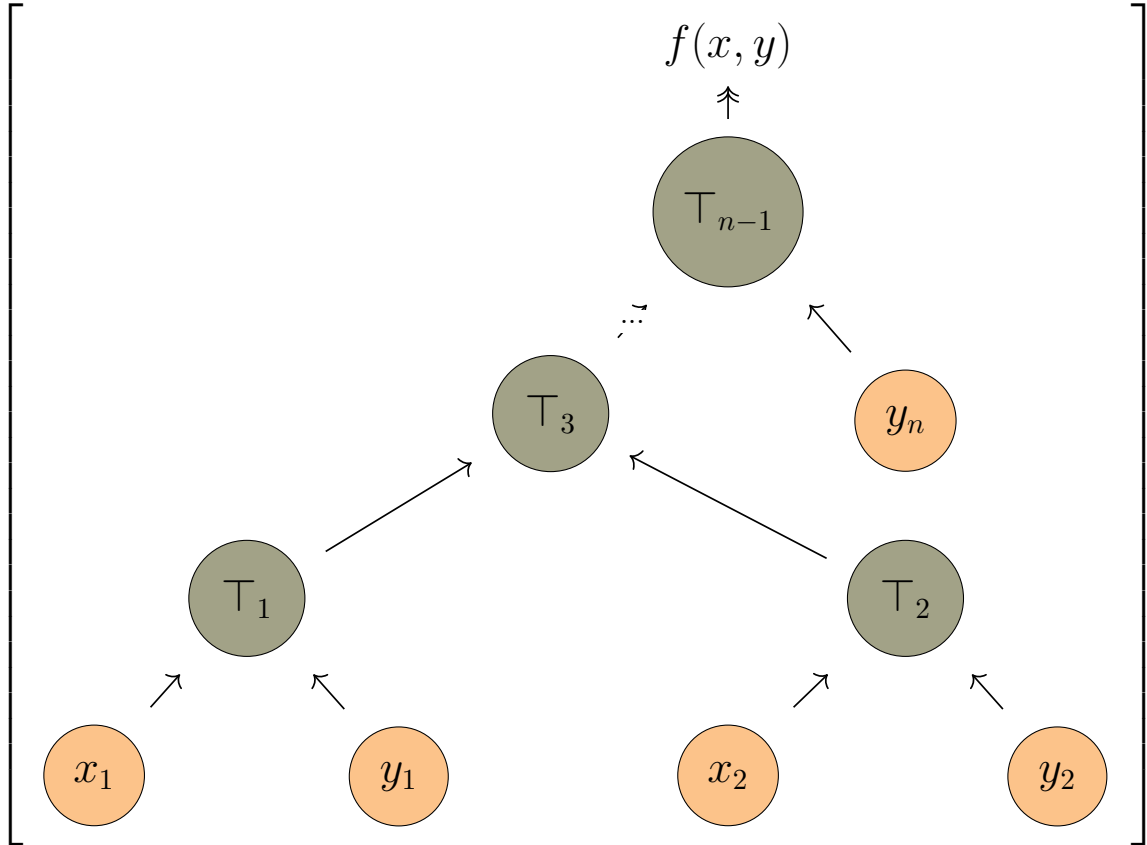
В уравнении (4), при описании цветка \mathfrak{F}_f^s , для экономии места, у функциональных зависимостей f_i были опущены пары операндов x_i и y_i . Также стоит отметить ре-

курсивный характер вычисления бутонов новых композиций: каждый предыдущий бутон цветка в новой композиции следующего бутона выступает в качестве операнда и размещается в лепестке.

Графическую иллюстрацию топологии цветка \mathfrak{F}_f^s можно представить следующим образом

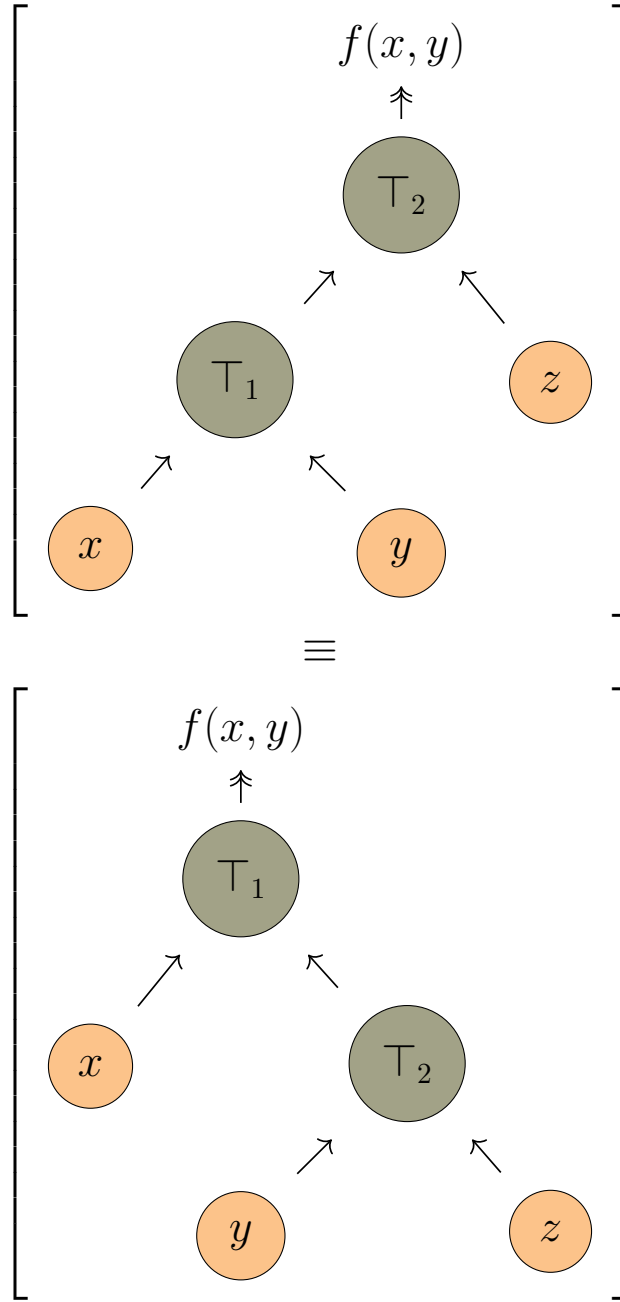


Корнем произвольного бутона $\mathfrak{B}_{f_i(x_i, y_i)}^{\top_i}$ является новый лепесток, представляющий функциональную зависимость $f_i(x_i, y_i)$. Поэтому, в целях экономии места, топологию цветка \mathfrak{F}_f , являющегося результатом композиции n бутонов, можно также представить в виде



Следствие 2.2. В силу определения 1.4 и утверждения 1.3, ассоциативность закона композиции \top позволяет установить изоморфизм топологии бутона $\mathfrak{B}_{f(x, y)}^{\top}$.

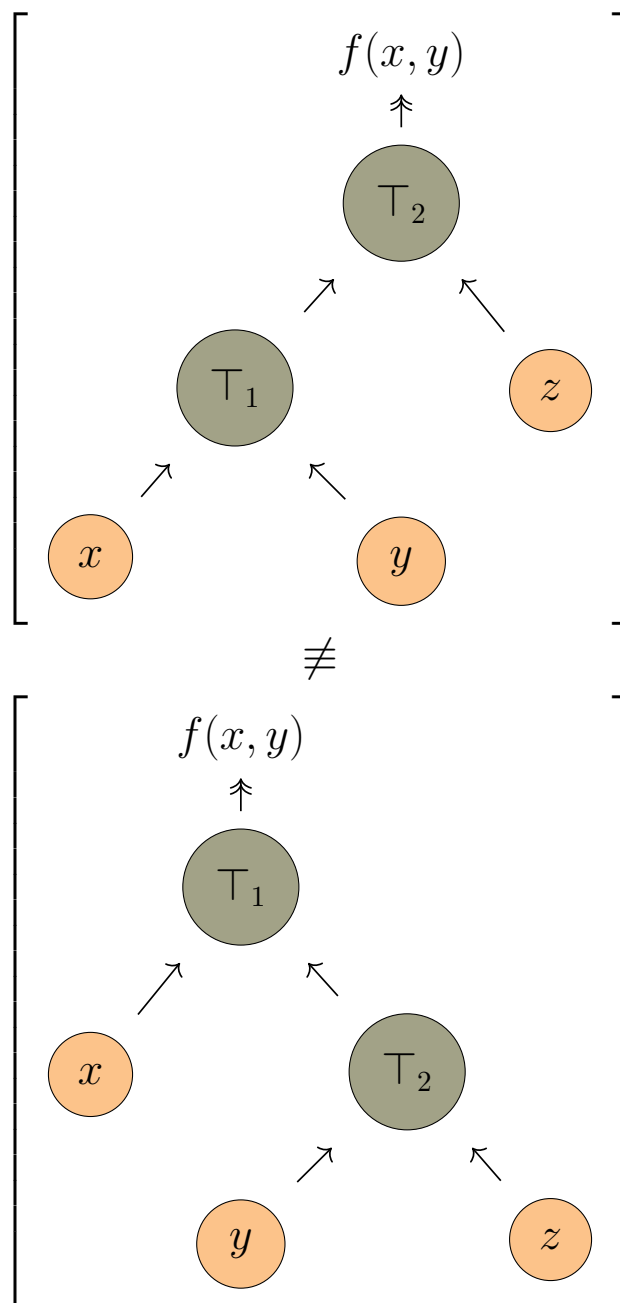
Определение 2.9. Цветок \mathfrak{F}_f^s , где $s \geq 2$, цветолага бутонов $\mathfrak{B}_{f_n(x_n, y_n)}^{\top_n}$ которого хранят ассоциативные законы композиций \top_n , является ассоциативным, а его топология изоморфна



Свойство ассоциативности можно продемонстрировать на примере цветка \mathfrak{F}_f^2 в виде формулы

$$\mathfrak{F}_f^2 = \mathfrak{B}_{(\mathfrak{B}_{(x,y)}^{\top_1}, z)}^{\top_2} \equiv \mathfrak{B}_{(x, \mathfrak{B}_{(y,z)}^{\top_2})}^{\top_1}. \quad (5)$$

Обратно определению 2.9, неассоциативные композиции, хранимые цветоложами бутонов, порождают неассоциативные цветы.

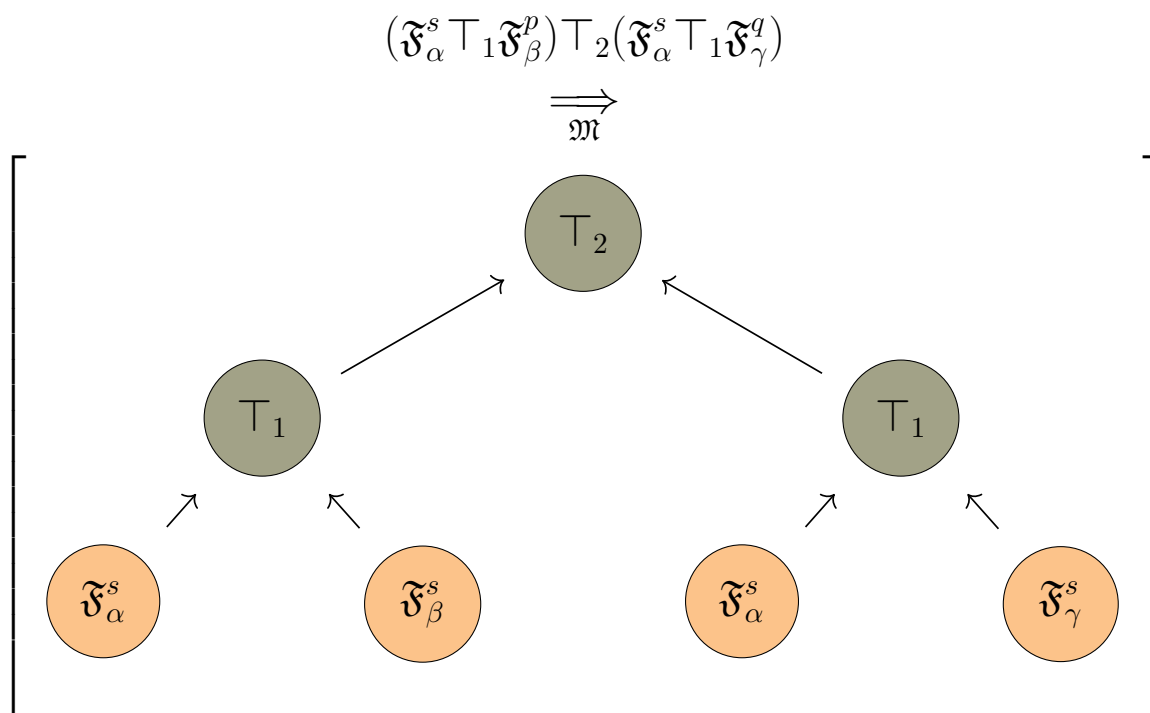


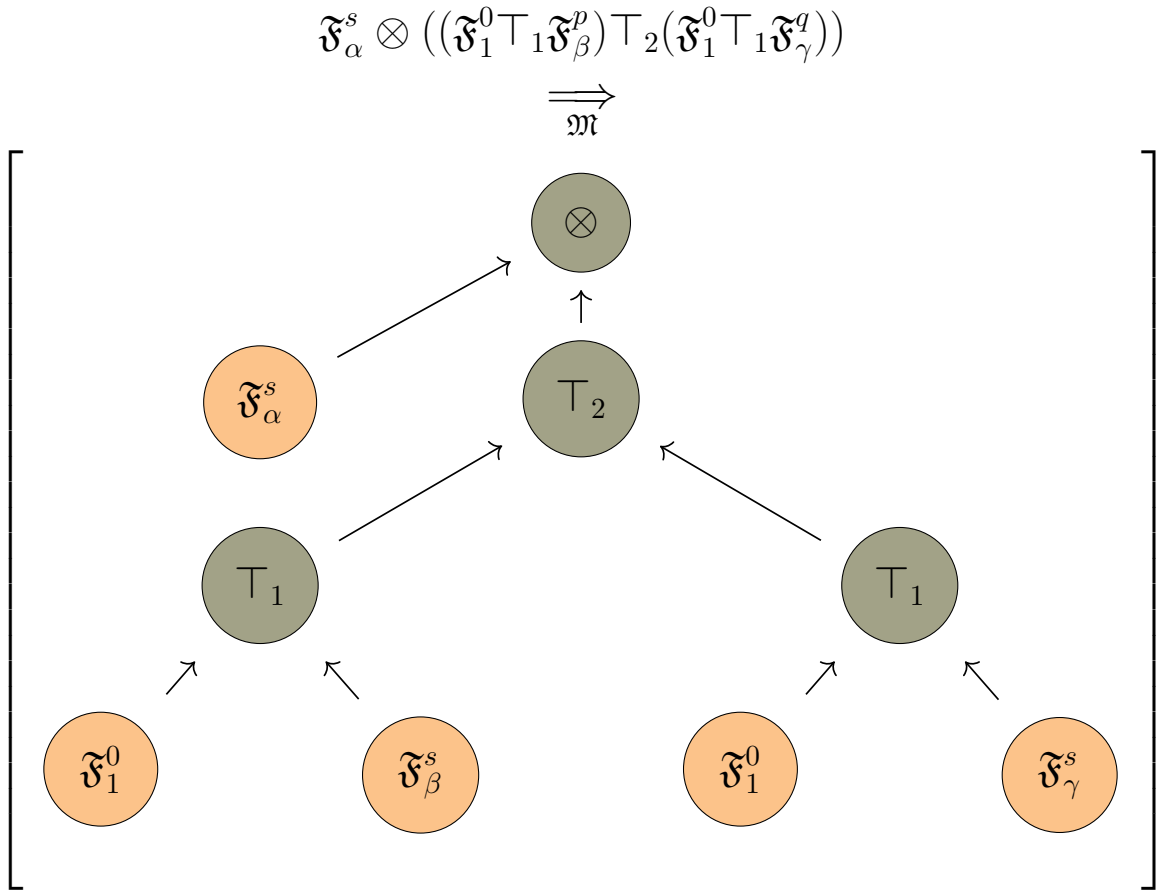
2.3 Упрощения выражений

Определение 2.10. Для законов композиции вида $\top_1 \in (\otimes, \oslash)$ и $\top_2 \in (\oplus, \ominus)$, упрощение композиции цветов имеет вид

$$(\mathfrak{F}_\alpha^s \top_1 \mathfrak{F}_\beta^p) \top_2 (\mathfrak{F}_\alpha^s \top_1 \mathfrak{F}_\gamma^q) \equiv \mathfrak{F}_\alpha^s \otimes ((\mathfrak{F}_1^0 \top_1 \mathfrak{F}_\beta^p) \top_2 (\mathfrak{F}_1^0 \top_1 \mathfrak{F}_\gamma^q)). \quad (6)$$

Топологии цветов левой и правой частей уравнения (6) различны. Высота цветка в левой части (6) больше высоты цветка из правой части уравнения на единицу, за счет дополнительной композиции со скалярным лепестком \mathfrak{F}_1^0 . Тем не менее, топологическое упрощение рационально когда высота цветка $h_{\mathfrak{F}_\alpha^s}$, выносимого за скобку $h_{\mathfrak{F}_\alpha^s} > 3$, так как, как высота скалярного цветка $h_{\mathfrak{F}_1^0} = 1$.





Часть II

Техника дифференцирования цветов

3 Производная бутона; производная композиций

3.1 Производная бутона

Исследуем дифференциальные свойства лугов \mathfrak{M} , образованных композициями вектор-функций одного действительного переменного, определенных на части тела \mathbf{R} действительных чисел и принимающих значения в некотором векторном пространстве Σ над телом \mathbf{R} .

Определение 3.1. Пусть пара вектор-функций \vec{f} и \vec{g} , определенных на не сходящемся к точке интервале $I \subset \mathbf{R}$, дифференцируемы в точке $x_0 \in I$. Тогда дифференциал цветка \mathfrak{F}^s , состоящего из одного бутона $\mathfrak{F}^1 = \mathfrak{B}_{(\vec{f}, \vec{g})}^\top$, имеет вид дифференциала закона композиции \top вектор-функций \vec{f} и \vec{g}

$$D\mathfrak{B}_{(\vec{f}, \vec{g})}^\top = D(\vec{f} \top \vec{g}) \quad (7)$$

и называется первой производной (или просто производной) бутона $\mathfrak{B}_{(\vec{f}, \vec{g})}^\top$ в точке x_0 .

3.2 Производная композиций

Определение 3.2. Пусть $(\vec{x}, \vec{y}) \longrightarrow [\vec{x} + \vec{y}]$ – непрерывное билинейное отображение суммы $\Sigma_x + \Sigma_y$ в Σ , где $\Sigma_x, \Sigma_y, \Sigma$ – топологические векторные пространства над телом \mathbf{R} , а \vec{x} и \vec{y} – вектор-функции, дифференцируемые в точке x_0 и принимающие свои значения соответственно в Σ_x и Σ_y , то вектор-функция $h(x) = [\vec{f}(x) + \vec{g}(x)]$ имеет производную, равную

$$D(h(x_0)) = D(\vec{f}(x_0)) + D(\vec{g}(x_0)).$$

Тогда дифференциал цветка суммы вектор-функций \vec{f} и \vec{g} , дифференцируемых на интервале $I \subset \mathbf{R}$, равен

$$D\mathfrak{F}_{\vec{f}(x)+\vec{g}(x)}^1 = \mathfrak{F}_{D(\vec{f}(x)+\vec{g}(x))}^1, \quad (8)$$

где дифференциал цветка суммы

$$D\mathfrak{F}_{\vec{f}(x)+\vec{g}(x)}^1 = D\mathfrak{B}_{(\vec{f}(x), \vec{g}(x))}^\oplus,$$

а цветок дифференциала суммы

$$\mathfrak{F}_{D(\vec{f}(x)+\vec{g}(x))}^1 = \mathfrak{B}_{(D(\vec{f}(x)), D(\vec{g}(x)))}^\oplus.$$

Утверждение 3.1. Учитывая коммутативность операции \oplus , можно составить соответствующее общее правило дифференцирования суммы n вектор-функций, дифференцируемых на интервале $I \subset \mathbf{R}$

$$D\mathfrak{F}_{\sum_{i=1}^n \vec{f}_i(x)}^{n-1} = \mathfrak{F}_{D(\sum_{i=1}^n \vec{f}_i(x))}^{n-1} \quad (9)$$

где

$$D\mathfrak{F}_{\sum_{i=1}^n \vec{f}_i(x)}^{n-1} = \mathfrak{B}_{(\mathfrak{F}_{\sum_{i=1}^{n+k-1} \vec{f}_i(x)}^{n-2}, D(\mathfrak{F}_{\vec{f}_n(x)}^1))}^{\oplus}, D(\mathfrak{F}_{\sum_{i=1}^{n+k} \vec{f}_i(x)}^{n+k-1}, D(\mathfrak{F}_{\vec{f}_{n+k+1}(x)}^1))$$

Определение 3.3. Учитывая некоммутативность операции \ominus , можно составить соответствующее правило дифференцирования разности вектор-функций \vec{f} и \vec{g} , дифференцируемых на интервале $I \subset \mathbf{R}$

$$D\mathfrak{F}_{\vec{f}(x)-\vec{g}(x)}^1 = \mathfrak{F}_{D(\vec{f}(x)-\vec{g}(x))}^1, \quad (10)$$

где дифференциал цветка разности

$$D\mathfrak{F}_{\vec{f}(x)-\vec{g}(x)}^1 = D\mathfrak{B}_{(\vec{f}(x), \vec{g}(x))}^{\ominus},$$

а цветок дифференциала разности

$$\mathfrak{F}_{D(\vec{f}(x)-\vec{g}(x))}^1 = \mathfrak{B}_{(D(\vec{f}(x)), D(\vec{g}(x)))}^{\ominus}.$$

Определение 3.4. Пусть $(\vec{x}, \vec{y}) \longrightarrow [\vec{x} \times \vec{y}]$ – непрерывное билинейное отображение произведения $\Sigma_x \times \Sigma_y$ в Σ , где Σ_x , Σ_y , Σ – топологические векторные пространства над телом \mathbf{R} , а \vec{x} и \vec{y} – вектор-функции, дифференцируемые в точке x_0 и принимающие свои значения соответственно в Σ_x и Σ_y , то вектор-функция $h(x) = [\vec{f}(x) \cdot \vec{g}(x)]$ имеет производную, равную

$$D(h(x_0)) = [D(\vec{f}(x_0))\vec{g}(x)] + [\vec{f}(x)D(\vec{g}(x_0))].$$

Тогда дифференциал цветка произведения вектор-функций

\vec{f} и \vec{g} , дифференцируемых на интервале $I \subset \mathbf{R}$, равен

$$D\mathfrak{F}_{\vec{f}(x) \cdot \vec{g}(x)}^1 = \mathfrak{F}_{D(\vec{f}(x) \cdot \vec{g}(x))}^3, \quad (11)$$

где дифференциал цветка произведения

$$D\mathfrak{F}_{\vec{f}(x) \cdot \vec{g}(x)}^1 = D\mathfrak{B}_{(\vec{f}(x), \vec{g}(x))}^{\otimes},$$

а цветок дифференциала произведения

$$\mathfrak{F}_{D(\vec{f}(x) \cdot \vec{g}(x))}^3 = \mathfrak{B}^{\oplus} \left(\mathfrak{B}_{(D(\vec{f}(x_0)), \vec{g}(x))}^{\otimes}, \mathfrak{B}_{(\vec{f}(x), D(\vec{g}(x_0)))}^{\otimes} \right).$$

Утверждение 3.2. Рассматривая дифференциал произведения n вектор-функций, можно установить зависимость между количеством n вектор-функций, образующих дифференциал цветка произведения $D\mathfrak{F}_{\prod_{i=1}^n \vec{f}_i(x)}^{n-1}$ и пышностью s цветка дифференциала произведения $\mathfrak{F}_{D(\prod_{i=1}^n \vec{f}_i(x))}^s$

$$s = n^2 - 1$$

и

$$D\mathfrak{F}_{\prod_{i=1}^n \vec{f}_i(x)}^{n-1} = \mathfrak{F}_{D(\prod_{i=1}^n \vec{f}_i(x))}^{n^2-1}. \quad (12)$$

Определение 3.5. Пусть f является дифференцируемой числовой функцией на интервале $I \subset \mathbf{R}$ в точке x_0 , что $f(x_0) \neq 0$, то $\frac{1}{f(x)}$ будет иметь в точке x_0 производную, равную $-\frac{D(f(x_0))}{(f(x_0))^2}$, с соответствующим отображением на луг \mathfrak{M} в виде производной цветка

$$D\mathfrak{F}_{\frac{1}{f(x)}}^1 = \mathfrak{F}_{D(\frac{1}{f(x)})}^3, \quad (13)$$

где дифференциал цветка отношения

$$D\mathfrak{F}_{\frac{1}{f(x)}}^1 = D\mathfrak{B}_{(1, f(x))}^{\odot},$$

а цветок дифференциала отношения

$$\mathfrak{F}_{D(\frac{1}{f(x_0)})}^3 = \mathfrak{B}_{(\mathfrak{B}_{(-1, D(f(x_0)))}^{\otimes}, \mathfrak{B}_{(f(x_0), f(x_0))}^{\otimes})}^{\odot}.$$

4 Производная сложной функции; полная производная; градиентный спуск

4.1 Производная сложной функции

Определение 4.1. Пусть числовая функция g определена на интервале $I \subset \mathbf{R}$, а вектор-функция \vec{f} определена на интервале из \mathbf{R} , содержащем $g(I)$. Тогда отображение на луг \mathfrak{M} сложной производной $\vec{f} \circ g$ имеет вид

$$D\mathfrak{F}_{\vec{f}(g(x))}^{p+q} = D\mathfrak{F}_{g(x)}^p \otimes \mathfrak{F}_{D_u \vec{f}(u)}^q \quad (14)$$

или это же выражение возможно представить относительно бутонов

$$D\mathfrak{B}_{(\vec{f}, g(x))}^{\odot} = D\mathfrak{B}_{(D\mathfrak{F}_{g(x)}^p, \mathfrak{F}_{D_u \vec{f}(u)}^q)}^{\otimes}.$$

Здесь $u \in \mathbf{R}$ – некоторая обобщенная переменная вектор-функции \vec{f} , D_u – частная производная по обобщенной переменной u , p – пышность цветка $\mathfrak{F}_{g(x)}^p$, отображающем числовую функцию $g(x)$ на луг \mathfrak{M} , q – пышность цвет-

ка $\mathfrak{F}_{D_u \vec{f}(u)}^q$, отображающем частную производную вектор-функции $\vec{f}(u)$ по обобщенной переменной u на луг \mathfrak{M} .

4.2 Полная производная

Определение 4.2. Пусть вектор функция \vec{f} , определенная на топологических векторных пространствах Σ_x, Σ_y , где $\Sigma_x, \Sigma_y \in \Sigma$, а Σ топология над телом \mathbf{R} , дифференцируема в точке $x_0 \in \Sigma_x$ и $y_0 \in \Sigma_y$. Тогда отображение на луг \mathfrak{M} полного дифференциала вектор-функции \vec{f} имеет вид

$$D\mathfrak{F}_{\vec{f}(x,y)}^s = \left(D_x \mathfrak{F}_{\vec{f}(x,y)}^s \otimes \mathfrak{F}_{dx}^0 \right) \oplus \left(D_y \mathfrak{F}_{\vec{f}(x,y)}^s \otimes \mathfrak{F}_{dy}^0 \right) \quad (15)$$

или относительно бутонов

$$D\mathfrak{F}_{\vec{f}(x,y)}^s = \mathfrak{B}^{\oplus}_{(\mathfrak{B}^{\otimes}_{(D_x \mathfrak{F}_{\vec{f}(x,y)}^s, dx)}, \mathfrak{B}^{\otimes}_{(D_y \mathfrak{F}_{\vec{f}(x,y)}^s, dy)})}.$$

4.3 Градиентный спуск

В теории нейронных сетей [1] для определения изменения потенциала, представляемого n -мерной поверхностью $E(\vec{\theta})$, образованной вектор-функцией $\vec{\theta} = \vec{\theta}(\theta_1, \theta_2, \dots, \theta_n)$, предлагается использовать градиент спуска, который формирует некоторый граф вычислений, представляющий собой структуру данных, в узлах которой располагаются функции, а ребра связывают функции с аргументами. Применение метода градиента спуска сводится к двум этапам:

- (1) Формирование графа вычислений на основе определенной и фиксированной грамматики слов;
- (2) Формирование дифференциала графа вычислений, посредством дифференцирования выражений в узлах структуры, образованной на предыдущем шаге.

Определение 4.3. Используя формулу (14) определения полного дифференциала 4.2, возможно переписать представить обобщение полной производной для n -мерной поверхности $E(\vec{\theta})$

$$D\mathfrak{F}_{\vec{f}(x,y)}^s = \sum_{i=1}^n \left(D_{\theta_i} \mathfrak{F}_{E(\vec{\theta})}^s \otimes \mathfrak{F}_{d\theta_i}^0 \right) \equiv \left(\nabla \mathfrak{F}_{E(\vec{\theta})}^s \otimes \mathfrak{F}_{d\vec{\theta}}^0 \right), \quad (16)$$

где $\nabla \mathfrak{F}_{E(\vec{\theta})}^s$ – отображение на луг \mathfrak{M} градиента от потенциала $E(\vec{\theta})$

$$\nabla \mathfrak{F}_{E(\vec{\theta})}^s = D_{\vec{\theta}} \mathfrak{F}_{E(\vec{\theta})}^s. \quad (17)$$

В силу определений 4.1, 4.3 и выражения (17), и так как

$$D_{\vec{\theta}} \mathfrak{F}_{\vec{\theta}}^0 = \mathfrak{F}_{\vec{e}}^0,$$

можно установить взаимосвязь между градиентом цветка и цветком градиента, имеющую вид

$$\nabla \mathfrak{F}_{E(\vec{\theta})}^s = \left(D_{\vec{\theta}} \mathfrak{F}_{\vec{\theta}}^0 \otimes \mathfrak{F}_{D_{\vec{\theta}}E(\vec{\theta})}^t \right), \quad (18)$$

а также представить уравнение (18) в матричном виде

$$\nabla \mathfrak{F}_{E(\vec{\theta})}^s = \begin{pmatrix} \mathfrak{F}_{\vec{e}_{\theta_1}}^0 \\ \dots \\ \mathfrak{F}_{\vec{e}_{\theta_n}}^0 \end{pmatrix} \otimes \begin{pmatrix} \mathfrak{F}_{D_{\theta_1} E(\vec{\theta})}^t \\ \dots \\ \mathfrak{F}_{D_{\theta_n} E(\vec{\theta})}^t \end{pmatrix}.$$

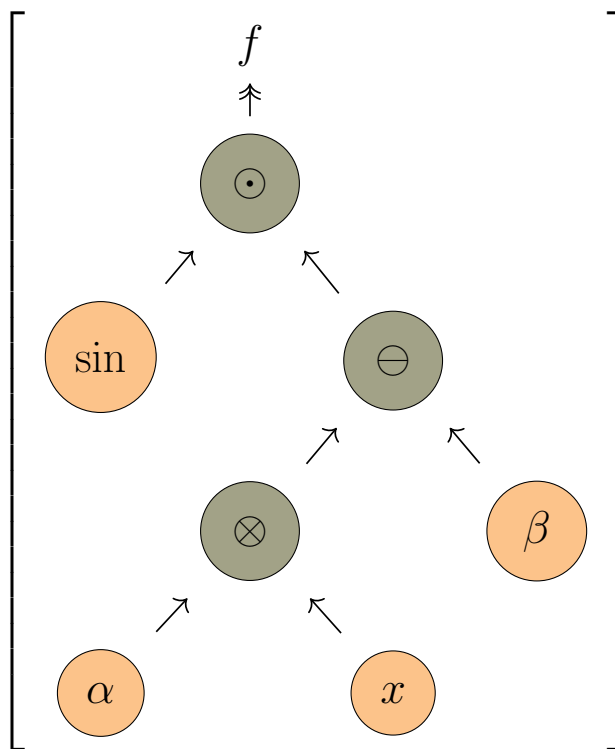
Однозначно определенный градиент цветка $\nabla \mathfrak{F}_{E(\vec{\theta})}^s$ посредством уравнения (17), используется в методе градиентного спуска при дифференцировании перцептонной ошибки $E(\vec{\theta})$. Как мы видим, уравнение (18) позволяет выразить градиент цветка $\nabla \mathfrak{F}_{E(\vec{\theta})}^s$ пышности s скалярного поля $E(\vec{\theta})$ через цветок $\mathfrak{F}_{D_{\vec{\theta}} E(\vec{\theta})}^t$ пышности t частных производных скалярного поля $E(\vec{\theta})$.

Часть III

Альбом с гербариями

Гербарий 1. Представим топологию цветка $\mathfrak{F}_{f'_x}^4$ уравнения

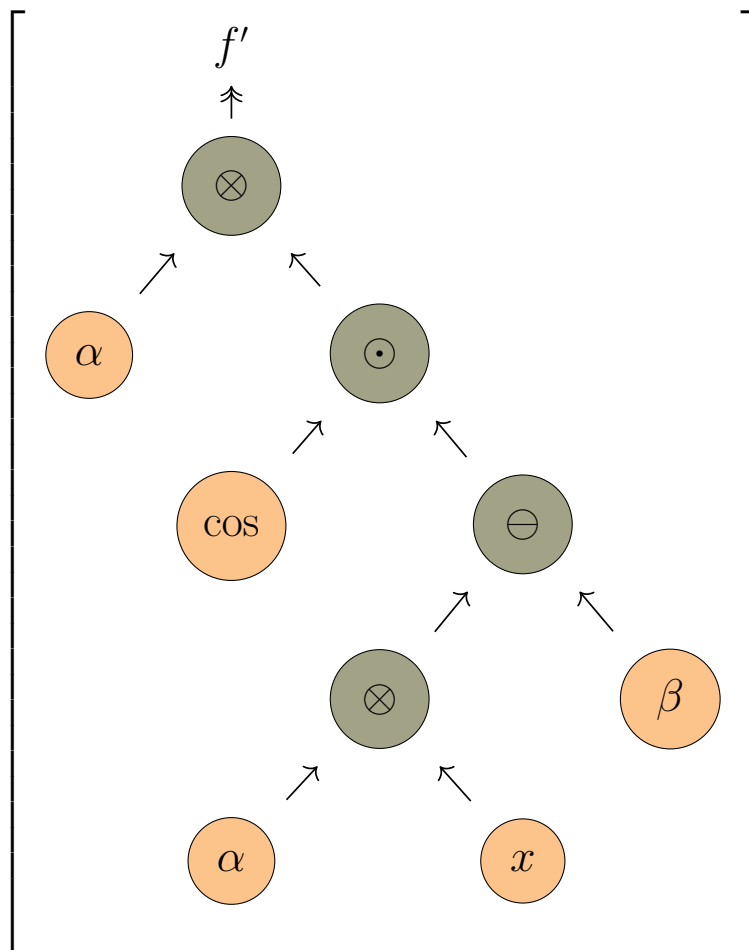
$$f(x) = \sin(\alpha x - \beta)$$

$$\xRightarrow{\mathfrak{M}}$$


дифференциал которого равен

$$f'(x) = \alpha \cos(\alpha x - \beta)$$

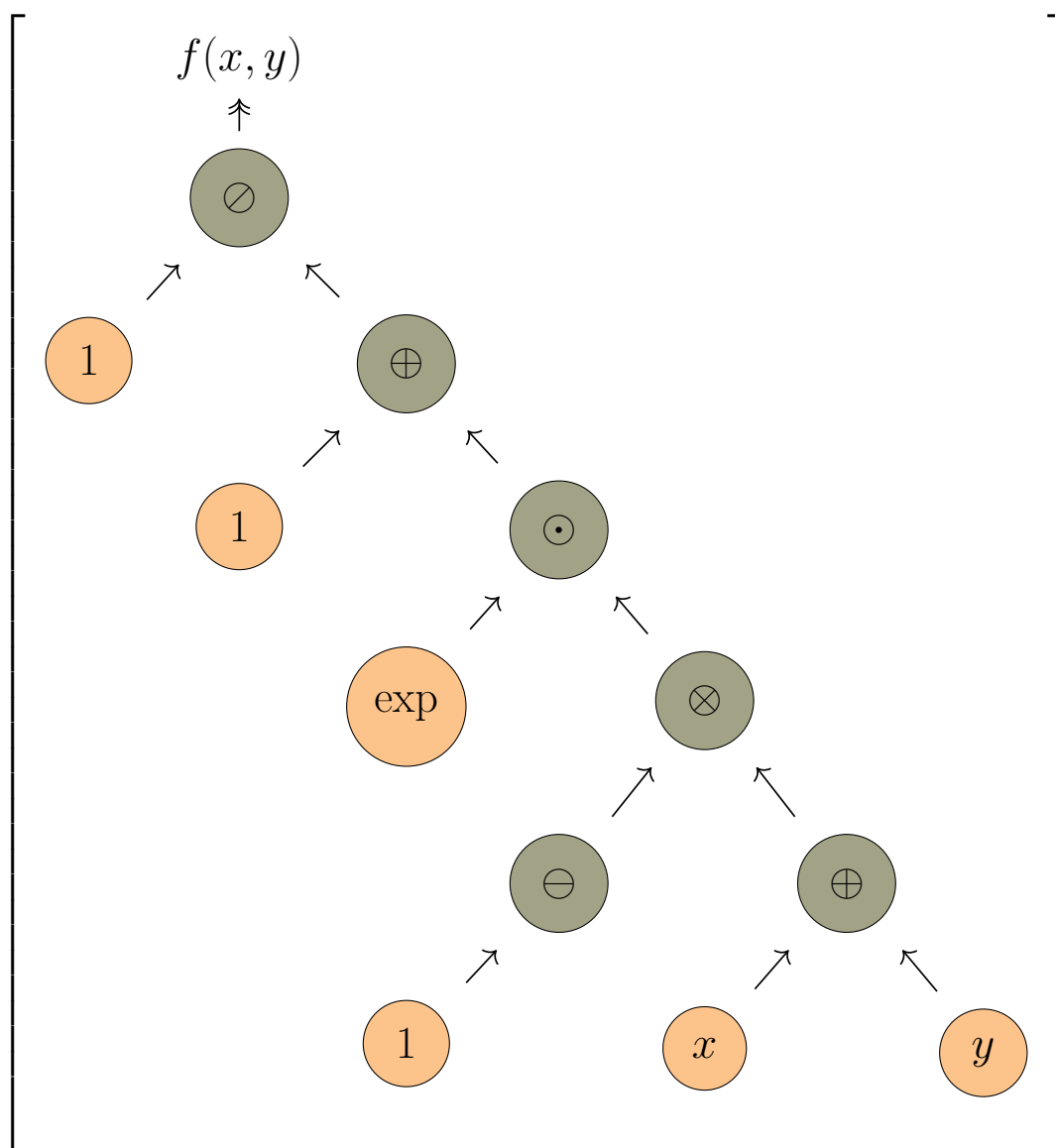
$\xRightarrow{\mathfrak{M}}$



Гербарий 2. Представим топологию цветка \mathfrak{F}_f^6 уравнения

$$f(x, y) = \frac{1}{1 + \exp(-(x + y))}$$

$$\xRightarrow{\mathfrak{M}}$$



Гербарий 3. Рассмотрим пример функции $f(x, y)$

$$f(x, y) = \cos(xy),$$

полная производная которой равна

$$Df(x, y) = -(y\delta x + x\delta y) \sin(xy).$$

Воспользовавшись уравнением (15), получим цветок полного дифференциала функции $f(x, y)$.

$$\begin{aligned} D\mathfrak{F}_f^s &= D_x \mathfrak{B}_{(\cos, \mathfrak{B}_{(x,y)}^\otimes)}^\odot \otimes \delta x \oplus D_y \mathfrak{B}_{(\cos, \mathfrak{B}_{(x,y)}^\otimes)}^\odot \otimes \delta y = \\ &= \mathfrak{F}_{D_u(\cos(u))} \otimes \left[D_x \mathfrak{B}_{(x,y)}^\otimes \otimes \delta x \oplus D_y \mathfrak{B}_{(x,y)}^\otimes \otimes \delta y \right] = \\ &= \mathfrak{F}_{-\sin(u)} \otimes \left[\mathfrak{B}_{(0,y)}^\oplus \otimes \delta x \oplus \mathfrak{B}_{(x,0)}^\oplus \otimes \delta y \right] = \\ &= \mathfrak{F}_{-\sin(u)} \otimes \mathfrak{B}_{(\mathfrak{B}_{(y,\delta x)}^\otimes, \mathfrak{B}_{(x,\delta y)}^\otimes)}^\oplus = \\ &= \mathfrak{B}_{\left(\mathfrak{F}_{-\sin(u)}, \mathfrak{B}_{(\mathfrak{B}_{(y,\delta x)}^\otimes, \mathfrak{B}_{(x,\delta y)}^\otimes)}^\oplus \right)}^\otimes = \\ &= \mathfrak{F}_{-(y\delta x + x\delta y) \sin(xy)}. \end{aligned}$$

Часть IV

Программная реализация

5 Введение

5.1 Постановка задачи

Чтобы протестировать модель цветков, мы предлагаем численную реализацию, демонстрирующую работу с основными изложенными выше идеями на классическом примере обучения линейной регрессии. Линейная регрессия моделирует работу одного нейрона, получающего на вход вектор значений \vec{x}_i и выдающего значение y_i , где $i \in (0, n)$ – индекс теста, а n – количество тестов или *количество эпох обучения*. Набор пар данных $\{\vec{x}_i, y_i\}$ подлежит обработке таким образом, чтобы минимизировать сумму квадратов отклонений на \tilde{y}_i от истинных значений y_i

$$\lim_{N \rightarrow \infty} \sum_{i=1}^N (\tilde{y}_i - y_i)^2 \rightarrow 0. \quad (19)$$

Мы будем рассматривать линейную регрессию функции вида

$$f = kx + b, \quad (20)$$

где k и b обучаемые параметры.

5.2 Распределения входных и выходных значений данных

С целью сохранения случайности процесса вектор входных параметров \vec{x}_i будет иметь равномерное случайное распределение от 0 до 10, а набор значений на выходе \vec{y}_i будет определяться по закону

$$\vec{y}_i = k_{res}\vec{x}_i + b_{res} + \vec{\xi}_i, \quad (21)$$

где $\vec{\xi}_i$ – нормально распределенный шум с ожидаемым значением $\mu = 0$, и стандартным отклонением σ , коэффициент наклона $k_{res} = 1$, коэффициент сдвига $b_{res} = -0.5$. При таком выборе значений на выходе \vec{y}_i , подходящих под распределение (21), коэффициенты k_{res} и b_{res} называются результирующими или итоговыми обучения нейрона по переменным k и b соответственно.

Для наглядности было реализовано два теста с $\sigma_1 = 0.1$ и $\sigma_2 = 0.15$. Число точек, участвующих в двух тестах, равно $N = 100$.

На рисунке 1 отмечены плотности вероятностей нормального распределения случайной величины $\vec{\xi}_i$ для пары входных наборов параметров $\{\vec{x}_i, \sigma_i, \mu\}$, полученных с помощью преобразования Бокса-Мюллера [2].

На рисунке 2 отмечены линейная зависимость (21) между входным \vec{x}_i и выходным \vec{y}_i векторами параметров при нормально распределенном шуме $\vec{\xi}_i$.

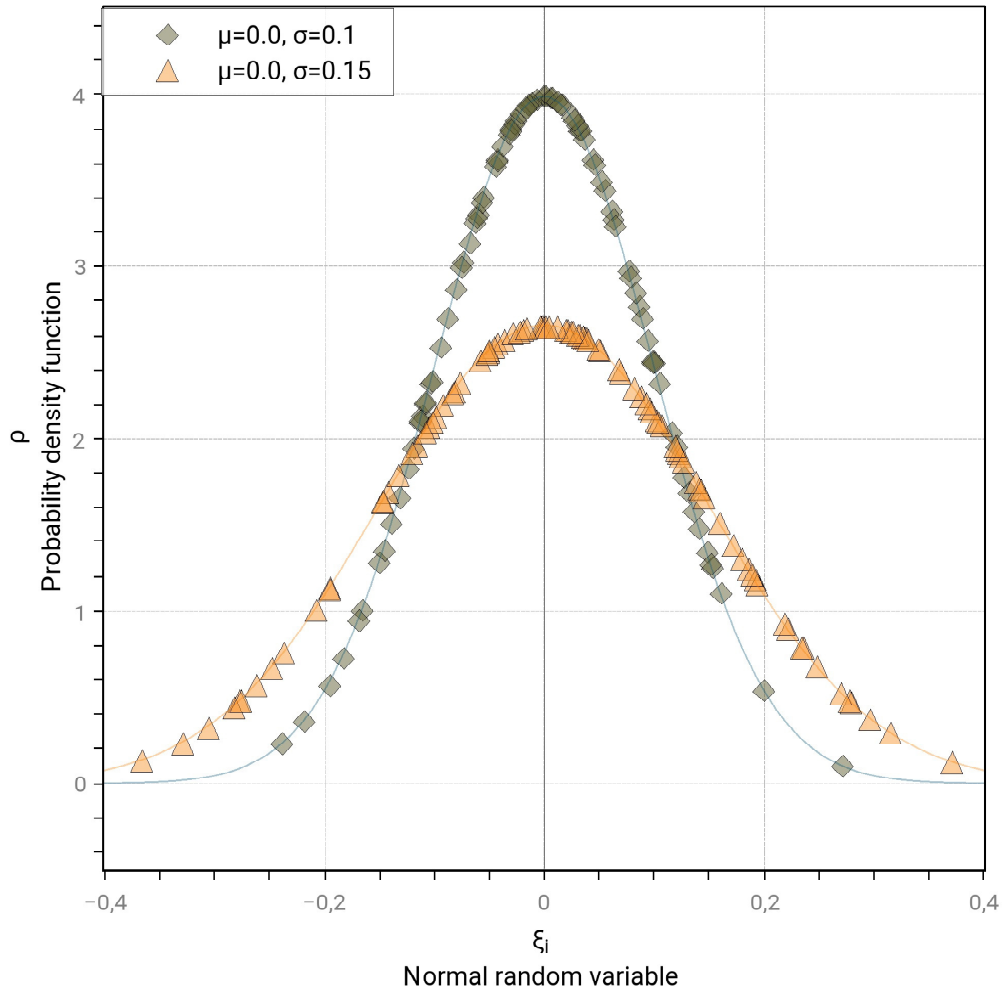


Рис. 1: Распределения плотности вероятности ρ по нормальному закону случайной величины ξ_i . Зелёными ромбиками отмечено распределение при математическом ожидании $\mu = 0$ и стандартном отклонении $\sigma = 0.1$, а желтыми треугольниками распределение при $\mu = 0$ и $\sigma = 0.15$. Наборы данных хранятся в файлах `./data/examples/test 1/density(xi)` и `./data/examples/test 2/density(xi)` соответственно.

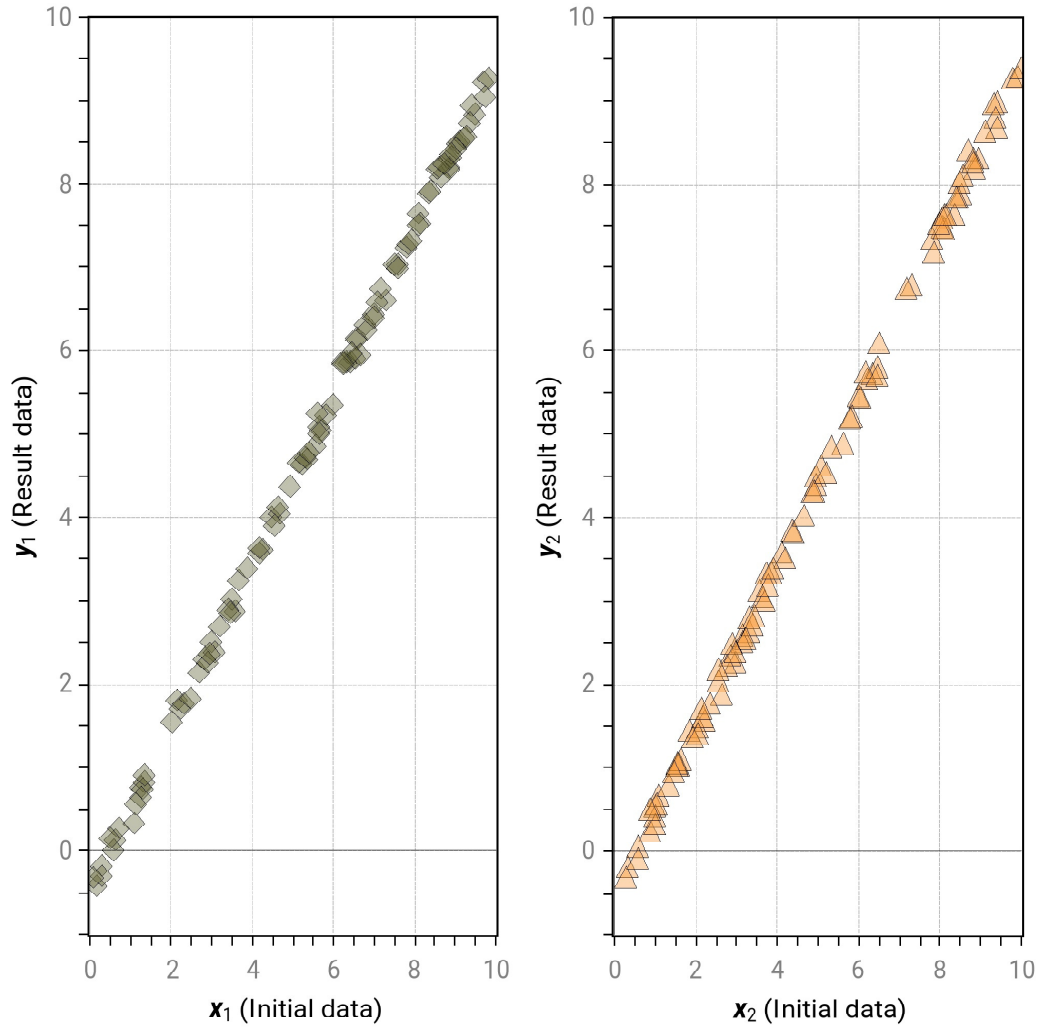


Рис. 2: Зависимости между входным \vec{x}_i и выходным \vec{y}_i векторами параметров. Зелёными ромбиками отмечено распределение при математическом ожидании $\mu = 0$ и стандартном отклонении $\sigma = 0.1$, а желтыми треугольничками распределение при $\mu = 0$ и $\sigma = 0.15$. Наборы данных хранятся в файлах `./data/examples/test 1/data_x`, `./data/examples/test 1/data_y` и `./data/examples/test 2/data_x`, `./data/examples/test 2/data_y` соответственно тестам.

6 Архитектура программы

Программа *Artificial-neural-network* (*ANN*) включает в себя три основных расчётных модуля

- *settings*;
- *graph*;
- *machine*;

и один внешний модуль (*ann*), отвечающий за интерфейс программы и запуск команды сборки *make* из файла *Makefile*.

6.1 Внешний модуль *ann*

Внешний модуль *ann* представляет собой совокупность исполняемого файла *ann*, разделяемой библиотеки *libann.so*, а также исходного кода в файле *ann.c*.

Файл *Makefile* связывает объектный файл сборки исходного кода *ANN_mkprocess.o*, который является результатом выполнения кода файла *ANN_mkprocess.c*, отвечающего за запуск модуля *settings* с помощью безопасного запуска системного вызова *execl*.

Модуль *ann* запускает каскад выполнения процессов расчётных модулей, которые последовательно отработывают друг за другом, а затем, завершаясь, также последовательно возвращаются в исходный процесс *ann*.

После аварийного, либо успешного возвращения управления в исходный модуль *ann* из расчётных модулей, процесс *ann* выполняет, с помощью безопасного запуска

системного вызова *exec*, очистку, генерируемых в процессе работы программы, промежуточных файлов:

- *./libs/settings.h*;
- *./libs/functions.c*;
- *./libs/graph_parameters.h*;

а затем выполняет *make*-команду, удаляющую разделяемые библиотеки *libsettings.so*, *libgraph.so*, *libmachine.so* и файлы запуска процессов расчетных модулей *mksettings*, *mkgraph*, *mkmachine*.

6.2 Расчётный модуль *settings*

Расчётный модуль *settings* представляет собой совокупность исполняемого файла *mksettings*, разделяемой библиотеки *libsettings.so*, исходного кода в файле *getsettings.c*, а также файлов настроек

- *./libs/map.cfg*;
- *./settings.cfg*.

В процессе работы модуль *settings* создает библиотеку *./libs/settings.h*, необходимую для работы следующего расчётного модуля *graph*.

Файл *./libs/map.cfg* содержит информацию о константах настроек и не подлежит корректировке пользователем.

Файл *./settings.cfg* содержит настройки, которые будут использоваться в процессе обучения нейронной сети. Список настроек представлен в таблице 2.

Таблица 2: Таблица доступных настроек пользователю в файле `./settings.cfg`

Настройка	Определение
EQUATION	<i>Строка оптимизируемого закона.</i> Допускается ввод операций умножения $*$, деления $/$, сложения $+$, вычитания $-$, возведения в степень $^$, круглых скобок $()$ и функций, перечисленных в приложении. Не допускается использование знака $\#$.
LEARNINGVARIABLES	<i>Обучаемые переменные.</i> Допускается перечисление через запятую переменных, которые являются обучаемыми.
PATHTOINITIALDATA	<i>Пути к начальным данным.</i> Допускается перечисление через запятую строк, содержащих псевдонимы переменных и пути к их начальным распределениям. Псевдоним от пути к данным отделяется двоеточием. Например, $x \text{ } ./\text{data}/\text{data}_x$. Данные начальных распределений должны быть помещены в каталог <code>./data</code> . Количество данных на каждую переменную должно быть одинаковым, либо, обозначаться в файле одной скаляр-строчкой. В противном случае, пользователю будет выдано сообщение о несоответствии размерностей входных векторов.

RESULTDATA	<i>Путь к итоговым данным.</i> Допускается указание строки, содержащей псевдоним результирующей переменной и пути к ее распределению. Псевдоним от пути к данным отделяется двоеточием. Например, <i>x :./data/data_x</i> . Данные итогового распределения должны быть помещены в каталог <i>./data</i> .
BATCHSIZE	<i>Размер мини-батча.</i> Допускается ввод целого положительного числа. Мини-батч является подмножеством множества исходных данных, на котором будет осуществляться обучение нейронной сети.
LEARNINGEPOCHS	<i>Количество эпох обучения.</i> Допускается ввод целого положительного числа. Количество эпох определяет количество тестов, которое будет проведено над нейронной сетью в процессе обучения.
LEARNINGRATE	<i>Скорость градиентного спуска.</i> Допускается ввод вещественного положительного числа. Начальная скорость, с которой параметры начинают двигаться вдоль профиля потенциала.

DIFFERENTIATIONACCURACY	<p>Скорость градиентного спуска. Допускается ввод вещественного положительного числа. Начальная скорость, с которой состояние нейронной сети начинает изменяться по профилю потенциала. Если задать скорость большой, то возможна ситуация “вылетания” точки состояния за пределы профиля. Такое состояние характеризуется большими значениями, сильно отличающимися от ожидаемых. Если задать скорость обучения слишком маленькой, то точка состояния рискует не успеть достигнуть минимума потенциальной ямы.</p>
-------------------------	---

После завершения обработки файла настроек и формирования библиотеки *./libs/settings.h*, процесс *mksettings* предложит запустить расчётный модуль формирования графа-цветка *graph*. В случае, если пользователь откажется от продолжения работы, то сформированный файл библиотеки очищается через процесс *ann*, а затем выполняется выход из программы в исходный терминал *ssh*.

6.3 Расчётный модуль *graph*

Расчётный модуль *graph* представляет собой совокупность исполняемого файла *mkgraph*, разделяемой библиотеки *libgraph.so*, исходного кода в файле *getgraph.c*,

а также файлов

- *./libs/functions.c*;
- *./libs/graph_parameters.h*.

Процесс *mkgraph* выполняет обработку строки уравнения, формируя по ее составляющим узлы графа-цветка. Затем, выполняет обход и записывает в определенном порядке обхода содержащиеся в бутонах выражения в специальный файл *./libs/functions.c*. Также в файл помещаются пути к начальным данным переменных, индексы обучаемых переменных.

Файл *./libs/graph_parameters.h* содержит информацию о графе цветке, а именно, количестве обучаемых переменных, бутонных и общем количестве переменных, образующих уравнение.

После завершения обработки файла настроек и формирования вспомогательных файлов, процесс *mkgraph* предложит запустить расчётный модуль формирования графа-цветка *machine*. В случае, если пользователь откажется от продолжения работы, то сформированные файлы очищаются через процесс *anp*, а затем выполняется выход из программы в исходный терминал *ssh*.

6.4 Расчётный модуль *machine*

Расчётный модуль *machine* представляет собой совокупность исполняемого файла *tkmachine*, разделяемой библиотеки *libmachine.so*, исходного кода в файле *getmachine.c*.

Процесс *tkmachine* выполняет n раз дифференцирование функций, содержащихся в бутонных графа-цветка,

по переменным, указанным в файле настроек. В ходе дифференцирования происходит слоевая передача данных по обучаемым переменным и, таким образом, вычисленный результат на шаге i становится входными данными для $i + 1$ шага.

Дифференцирование ведется по данным, которые содержатся в каталоге `./data`. Однако, обучение проводится не на всем кластере данных, объем которых может быть значительным, а на некоем выделенном случайном подмножестве, называемом, мини-батчем. Если пользователь желает отключить обработку данных на мини-батче, то достаточно установить в файле настроек значение 0 напротив строки-опции `BATCHSIZE`.

После выполнения дифференцирования графа, процесс *tkmachine* сообщает результат по обучаемым переменным и выходит в процесс *anp*, который очищает технические файлы и возвращается в *ssh*.

Список литературы

- [1] Николенко С., Архангельская Е., Кадушин А., “Глубокое обучение. Погружение в мир нейронных сетей”, изд. “Питер”, 2018
- [2] Box G.E.P., Muller Mervin E., “A Note on the Generation of Random Normal Deviates”, The Annals of Mathematical Statistics, 29 (2): 610–611, 1958