ЧЕБЫШЕВСКИЙ СБОРНИК
Том 19. Выпуск 1

УДК 004.032.26, 004.424.62 DOI 10.22405/2226-8383-2018-19-1-187-199

Классификация последовательностей на основе коротких мотивов

Офицеров Евгений Петрович — кафедра прикладной математики и информатики, Тульский государственный университет.
e-mail: eofitserov@gmail.com

Аннотация

Задачи, связанные с классификацией последовательностей символов некоторого алфавита, часто возникают в таких областях, как биоинформатика и обработка естественного языка. Методы глубокого обучения, в особенности модели на основе рекуррентных нейронных сетей, в последние несколько лет зарекомендовали себя как наиболее эффективный способ решения подобных задач. Однако существующие подходы имеют серьезный недостаток — низкую интерпретируемость получаемых результатов. Крайне сложно установить какие именно свойства входной последовательности ответственны за её принадлежность к тому или иному классу. Упрощение же таких моделей с целью повышения их интерпретируемости, в свою очередь, приводит к снижению качества классификации. Такие недостатки ограничивают применение современных методов машинного обучения во многих предметных областях. В настоящей работе мы представляем принципиально новую, интерпретируемую архитектуру нейронных сетей, основанную на поиске набора коротких подпоследовательностей — мотивов, наличие которых влияет на принадлежность последовательности к определенному классу. Ключевой составляющей предлагаемого решения является разработанный нами алгоритм дифференцируемого выравнивания, являющийся дифференцируемым аналогом таких классических способов сравнения строк, как редакционное расстояние Левенштейна и алгоритм Смита–Ватермана. В отличие от предыдущих работ, посвященных классификации последовательностей на основе мотивов, новый метод позволяет не только выполнять поиск в произвольной части строки, но и учитывать возможные вставки.

Ключевые слова: классификация последовательностей, машинное обучение, нейронные сети, поиск мотивов.

Библиография: 15 названий.

Для цитирования:
Е. П. Офицеров. Классификация последовательностей на основе коротких мотивов // Чебышевский сборник, 2018, т. 19, вып. 1, с. 187–199.

CHEBYSHEVSKII SBORNIK
Vol. 19. No. 1

UDC 004.032.26, 004.424.62 DOI 10.22405/2226-8383-2018-19-1-187-199

Motif based sequence classification

Ofitserov Evgeny Petrovich — department of applied mathematics and computer science, Tula State University,
e-mail: eofitserov@gmail.com

Abstract

Sequence classification problems often arise in such areas as bioinformatics and natural language processing. In the last few year best results in this field were achieved by the deep learning methods, especially by architectures based on recurrent neural networks (RNN). However, the common problem of such models is a lack of interpretability, i.e., extraction of key features from data that affect the most the model's decision. Meanwhile, using of less complicated neural network leads to decreasing predictive performance thus limiting usage of state-of-art machine learning methods in many subject areas. In this work we propose a novel interpretable deep learning architecture based on extraction of principal sets of short substrings — sequence motifs. The presence of extracted motif in the input sequence is a marker for a certain class. The key component of proposed solution is differential alignment algorithm developed by us, which provides a smooth analog of classical string comparison methods such as Levenshtein edit distance, and Smith–Waterman local alignment. Unlike previous works devoted to the motif based classification, which used CNN for shift-invariant searching, ours model provide a way to shift and gap invariant extraction of motifs.

Keywords: sequence classification, machine learning, neural network, motif extraction.

Bibliography: 15 titles.

For citation:
E. P. Ofitserov, 2018, "Motif based sequence classification" , *Chebyshevskii sbornik*, vol. 19, no. 1, pp. 187–199.

1. Введение

Методы глубокого обучения показали свою эффективность в задачах, связанных с классификацией последовательностей. При этом наилучших результатов достигают архитектуры, в основе которых лежат рекуррентные нейронные сети такие LSTM (Long short–term memory) [1] и GRU (Gated recurrent unit) [2], [3]. Однако, не смотря на все достоинства, важный недостаток подобных архитектур — плохая интерпретируемость получаемой модели. Существующие способы визуализации процесса принятия решения [4], [5] не позволяют однозначно ответить на вопрос, какие особенности входной последовательности ответственны за принадлежность к определенному классу. Этот недостаток ограничивает использование глубокого обучения в задачах классификации биологических последовательностей, где важна не только точность получаемой модели, но и возможность проанализировать какие особенности входной строки влияют на решение классификатора.

Другим подходом, активно применяемым в биоинформатике, является классификация на основе мотивов. При построении классификаторов такого типа, предполагается, что ключевым признаком, влияющим на принадлежность последовательности к определенному классу, является наличие в ней некоторой короткой подстроки — мотива. При этом, сами мотивы являются неизвестными параметрами модели, определяемыми в процессе обучения. Данное предположение является оправданным для многих задач, связанных с классификацией биологических последовательностей, а также при обработке естественного языка.

Примером использования классификатора на основе мотивов является работа [6], в которой авторы используют сверточную нейронную сеть для предсказания ДНК-связывающих белков. В основе предложенного ими решения лежит использование одномерного сверточного слоя. Такой слой состоит из набора ядер — скользящих окон, которые «просматривают» исходную последовательность. Каждое ядро представляет собой матрицу из $4 \times K$ коэффициентов, которые могут быть интерпретированы как позиционно-весовые матрицы соответствующих мотивов. После применения такого слоя образуется карта признаков, содержащая информацию о наличии мотивов в различных участках входной последовательности. В свою очередь, полученная карта признаков классифицируется с помощью одного или нескольких полносвязных слоев.

Использование сверток позволяет сети выделять интересующие подстроки, независимого от того, в какой части входной последовательности они находятся. Однако, важным недостатком такого решения является неспособность предложенной архитектуры учитывать возможные разрывы в мотиве (таблица 1).

a.	A	C	**T**	**G**	**A**	C
b.	**T**	**G**	**A**	A	G	A
c.	G	**T**	C	**G**	**A**	T

Таблица 1: Полужирным шрифтом выделены возможные положения мотива (TGA) в последовательности. Сверточная сеть естественным образом инвариантна к сдвигу и может правильно классифицировать варианты a и b, но не учитывает возможные вставки — пример c

Одним из возможных решений проблемы разрывов является использование более глубоких архитектур, комбинирующих сверточные и рекуррентные слои [7]–[9]. Однако при таком подходе теряется возможность явно визуализировать найденные мотивы и однозначно установить их связь с конкретными классами.

Целью данной работы является разработка нейронной сети с принципиально новой архитектурой, позволяющей выполнять поиск с учетом возможных разрывов не прибегая к усложнению модели. Так же, как и в сверточной сети, мотивы кодируются в виде коэффициентов —

параметров модели, организованных в матрицы размера $G \times K$, где G — мощность алфавита, а K — длина мотива. Однако в отличие от предыдущих работ, поиск мотива в последовательности и формирование карты признаков выполняется не с помощью операции свертки, а с использованием алгоритма дифференцируемым выравнивания мотива, что позволяет естественным образом учитывать возможные разрывы.

2. Дифференцируемое выравнивание

Пусть $x = x_1 x_2 \dots x_L$, $x_i \in G$, $L > 1$, — входная последовательность символов из алфавита G, $|G| = N$. Мотив $m = m_1 m_2 \dots m_K$ — короткая последовательность символов того же алфавита длины $K \leq L$.

Классическими критерием, позволяющим сказать содержит ли последовательность x мотив m, является расстояние Левенштейна [10], которое определяет схожесть двух строк, как минимальное количество вставок, замен и удалений необходимых чтобы перевести одну строку в другую. Используя такую метрику, можно ввести меру сходства мотива m и последовательности x как $f_{\text{Levenstein}}(x, m) = L - \text{Edit}(x, m)$, где $\text{Edit}(x, m)$ — редакционное расстояние, L — длина последовательности. Однако метрика Левенштейна является дискретной функцией, а получаемый критерий соответствия — не дифференцируемым, что не позволяет использовать градиентные методы для обучения модели и поиска неизвестных параметров мотива.

В данном разделе предлагается способ построить гладкую меру соответствия мотива и последовательности $f(x, m)$, которая также позволяет сделать вывод о том, содержится ли m в x, но при этом может быть продифференцирована по параметрам мотива. В дальнейшем, такая функция может быть использована для построения нейронной сети, в которой символы мотивов являются неизвестными, обучаемыми параметрами, а значения $f(x, m)$ — признаками, используемыми для принятия решения о принадлежности x к определенному классу.

В основе предлагаемого решения лежит понятие выравнивания последовательностей. Выравниванием мотива по последовательности называется отображение символов мотива m на подпоследовательность x' последовательности x. Такое соответствие может быть записано с помощью бинарной матрицы: $T_{K \times L}$, где $T_{i,j} = 1$, если i-й элемент мотива соответствует j-му символу последовательности x.

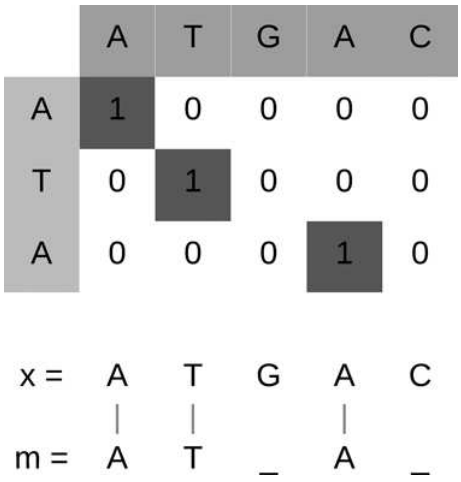

Рис. 1: Пример выравнивания мотива m по последовательности x и соответствующая ему матрица

Далее рассматриваются только те выравнивания, при которых мотив целиком содержится в последовательности x, то есть каждый элемент мотива соответствует какому-либо символу

последовательности. При таком предположении, в каждой строке матрицы T должна быть ровно одна единица: $\sum_{j=1}^{L} T_{i,j} = 1, \forall i$. Также, для элементов $T_{i,j}$ выполняется следующее условие: $T_{i,j} = 0,\ j < i \vee j \geq i + K$.

Для каждого T, отвечающего этим условиям, можно ввести следующую оценку, показывающую насколько хорошо мотив m соответствует $x' \subset x$:

$$S(X, \Theta, c_g, T) = n_g c_g + \sum_{i=1}^{K} \sum_{j=1}^{L} T_{i,j} C_{i,j}, \quad C = \Theta^T X. \tag{1}$$

В этой формуле T — матрица выравнивания, $\Theta_{N \times K}$ — позиционно весовая матрица мотива, i, j-й элемент которой равняется «штрафу» за замену j-го символа мотива на i-й символ алфавита, X — бинарная матрица входной последовательности, столбцы которой представляют собой унитарные коды символов исходной последовательности, n_g —- количество разрывов в выравнивании, $c_g \leq 0$ — штраф за разрыв. При таких обозначениях, значения коэффициентов $C_{i,j}$ показывают насколько хорошо j-й символ входной последовательности x соответствует i-му символу мотива.

Используя формулу (1), можно определить не зависящую от конкретного выравнивания меру сходства последовательности и мотива, как максимум $S(X, \Theta, c_g, T)$ по всем возможным T:

$$f\left(X, \Theta, c_g\right) = \max_{T} S\left(X, \Theta, c_g, T\right). \tag{2}$$

Рассчитанная таким образом мера сходства $f\left(X, \Theta, c_g\right)$ является аналогом расстояния Левенштейна, в котором штрафы за замены и вставки не являются постоянными, а задаются параметрами Θ и c_g. Классические алгоритмы выравнивания последовательностей, такие как алгоритм Смита–Ватермана [11]–[13], так же используют схожую меру соответствия последовательностей. При этом, для поиска максимальной оценки выравнивания в алгоритме Смита–Ватермана используется эффективный метод динамического программирования, позволяющий решить задачу оптимизации (2) за полиномиальное время.

Недостатком такого подхода является недифференцируемость получаемой функции f по параметрам Θ и c_g. Для того, чтобы добиться гладкости критерия соответствия мотива и последовательности, в работе предлагается заменить в выражении (2) поиск максимума $S\left(X, \Theta, c_g, T\right)$ на взвешенную сумму по всем возможным выравниваниям:

$$f\left(X, \Theta, c_g, \right) = \sum_{T \in D} S\left(X, \Theta, c_g, T\right) p\left(T \mid X, \Theta, c_g, \right). \tag{3}$$

В этой формуле D — множество всех возможных выравниваний, $p\left(T \mid X, \Theta, c_g\right)$ — вероятности соответствующих выравниваний, при условии входной последовательности X и мотива с параметрами Θ и c_g, для которых выполняется нормировочное условие $\sum_{T \in D} p\left(T \mid X, \Theta, c_g, \right) = 1$.

Используя теорему Байеса для $p\left(T \mid X, \Theta\right)$, можно переписать сумму в виде:

$$p\left(T \mid X, \Theta, c_g\right) = \frac{p\left(X \mid T, \Theta, c_g\right)}{\sum\limits_{u \in D} p(X \mid u, \Theta, c_g)},$$

$$f\left(X, \Theta, c_g, \right) = \frac{\sum\limits_{T \in D} S\left(X, \Theta, c_g, T\right) p\left(X \mid T, \Theta, c_g\right)}{\sum\limits_{T \in D} p(X \mid T, \Theta, c_g)},$$

где $p(X \mid T, \Theta, c_g)$ — представляет собой вероятность входной последовательности X, при условии того, что в ней содержится заданный мотив с заданным выравниванием. Эта вероятность может быть рассчитана по формуле:

$$p(X \mid T, \Theta) = \left(\prod_{i=1}^{K} \prod_{j=1}^{L} W_{i,j}^{T_{i,j}} \right) p_g^{n_g}, \quad W = P^T X. \tag{4}$$

В этом выражении P — позиционно-вероятностная матрица мотива, i,j-й элемент которой показывает вероятность встретить i-й символ алфавита на j-й позиции мотива, p_g — вероятность разрыва. По аналогии с формулой (1), коэффициенты $W_{i,j}$ выражают вероятность того, что j-й символ входной последовательности соответствует i-му элементу мотива. Позиционно-вероятностная матрица мотива P, а также вероятность разрыва p_g могут быть получены из позиционно-весовой матрицы мотива Θ и коэффициента c_g соответственно:

$$P = \begin{pmatrix} \dfrac{e^{\Theta_{1,1}}}{\sum\limits_{i=1}^{N} e^{\Theta_{i,1}}} & \dfrac{e^{\Theta_{1,2}}}{\sum\limits_{i=1}^{N} e^{\Theta_{i,2}}} & \cdots \\ \vdots & \ddots & \\ \dfrac{e^{\Theta_{N,1}}}{\sum\limits_{i=1}^{N} e^{\Theta_{i,1}}} & \cdots & \dfrac{e^{\Theta_{N,K}}}{\sum\limits_{i=1}^{N} e^{\Theta_{i,K}}} \end{pmatrix}, \quad p_g = e^{c_g}.$$

3. Прямой и обратный ход

Для расчета значений значений $f\,(X, \Theta, c_g,)$ по формуле (3) требуется выполнить суммирование по всем возможным выравниваниям мотива по последовательности. Классическим подходом для решения задачи такого перебора является использования методов динамического программирования, основанных на префиксном кодировании. В работе предлагается алгоритм, основанный на аналогичном подходе, который позволяет за полиномиальное время вычислить сумму из формулы (3), не выполняя явный перебор множества D.

Пусть

$$\alpha_{i,j} = \sum_{T \in D} p\left(X_{1:j} \mid T, \Theta_{1:i}\right),$$

$$\beta_{i,j} = \sum_{T \in D} S\left(X_{1:j}, \Theta_{1:i}, T\right) p\left(X_{1:j} \mid T, \Theta_{1:i}\right),$$

$$i = \overline{1, K},\ j = \overline{1, L}.$$

Здесь $p\left(X_{1:j} \mid T, \Theta_{1:i}\right) = p_g^{n_g(i,j)} \prod\limits_{k=1}^{i} \prod\limits_{l=1}^{j} W_{k,l}^{T_{k,l}}$, — вероятность первых j символов входной последовательности, при условии того, что они содержат первые i элементов мотива, $S\left(X_{1:j}, \Theta_{1:i}, T\right) = p_g n_g(i,j) + \sum\limits_{k=1}^{i} \sum\limits_{l=1}^{j} C_{k,l} T_{k,l}$ — оценка соответствующего выравнивания первых i элементов мотива по первым j символам последовательности. При таких обозначениях формула (3) может быть переписана в виде:

$$f\,(X, \Theta) = \frac{\beta_{K,L}}{\alpha_{K,L}}.$$

В свою очередь, с учетом предположения о том, что каждый символ мотива соответствует какому–либо символу входной последовательности, можно показать, что для $\alpha_{i,j}$ и $\beta_{i,j}$ справедливы следующие рекуррентные соотношения:

$$\alpha_{i,j} = \begin{cases} \alpha_{i-1,j-1} W_{i,j} + \alpha_{i,j-1} p_g, & i = \overline{2, K-1},\ j \geq i, \\ W_{i,j} + \alpha_{i,j-1} p_g, & i = 1, \\ \alpha_{i-1,j-1} W_{i,j} + \alpha_{i,j-1}, & i = K,\ j \geq i, \\ 0, & j < i, \end{cases} \tag{5}$$

$$\beta_{i,j} = \begin{cases} W_{i,j} \left(\beta_{i-1,j-1} + C_{i,j} \alpha_{i-1,j-1}\right) + p_g \left(\beta_{i,j-1} + c_g \alpha_{i,j-1}\right), & i = \overline{2, K-1},\ j \geq i, \\ W_{i,j} C_{i,j} + p_g \left(\beta_{i,j-1} + c_g \alpha_{i,j-1}\right), & i = 1, \\ W_{i,j} \left(\beta_{i-1,j-1} + C_{i,j} \alpha_{i-1,j-1}\right) + \beta_{i,j-1}, & i = K,\ j \geq i, \\ 0, & j < i. \end{cases} \tag{6}$$

Приведенные выше формулы предоставляют эффективный алгоритм для расчета всех коэффициентов $\alpha_{i,j}$ и $\beta_{i,j}$, что позволяет за $O(LK)$ операций вычислить меру соответствия мотива входной последовательности. При этом, получаемая функция $f\left(X, \Theta, c_g\right)$ является гладкой и может быть продифференцирована по параметрам мотива:

$$\frac{\partial f\left(X, \Theta\right)}{\partial \Theta} = \frac{\alpha_{K,L} \dfrac{\partial \beta_{K,L}}{\partial \Theta} - \beta_{K,L} \dfrac{\partial \alpha_{K,L}}{\partial \Theta}}{\alpha_{K,L}^2},$$

$$\frac{\partial \alpha_{K,L}}{\partial \Theta} = \sum_{i,j} \frac{\partial \alpha_{K,L}}{\partial C_{i,j}} \frac{\partial C_{i,j}}{\partial \Theta} + \frac{\partial \alpha_{K,L}}{\partial W_{i,j}} \frac{\partial W_{i,j}}{\partial \Theta},$$

$$\frac{\partial \beta_{K,L}}{\partial \Theta} = \sum_{i,j} \frac{\partial \beta_{K,L}}{\partial C_{i,j}} \frac{\partial C_{i,j}}{\partial \Theta} + \frac{\partial \beta_{K,L}}{\partial W_{i,j}} \frac{\partial W_{i,j}}{\partial \Theta}.$$

Для расчета производных $\frac{\partial \alpha_{K,L}}{\partial C_{i,j}}$, $\frac{\partial \alpha_{K,L}}{\partial W_{i,j}}$, $\frac{\partial \beta_{K,L}}{\partial C_{i,j}}$, $\frac{\partial \beta_{K,L}}{\partial W_{i,j}}$ также могут быть использованы рекуррентные соотношения (5),(6). Для этого требуется продифференцировать их по соответствующим параметрам:

$$\frac{\partial \alpha_{i,j}}{\partial W_{k,l}} = \begin{cases} \dfrac{\partial \alpha_{i-1,j-1}}{\partial W_{k,l}} W_{i,j} + \dfrac{\partial \alpha_{i,j-1}}{\partial W_{k,l}} p_g + \delta_{i,j=k,l} \alpha_{i-1,j-1}, & i = \overline{2, K-1},\ j \geq i, \\ \delta_{i,j=k,l} + \dfrac{\partial \alpha_{i,j-1}}{\partial W_{k,l}} p_g, & i = 1, \\ \dfrac{\partial \alpha_{i-1,j-1}}{\partial W_{k,l}} W_{i,j} + \dfrac{\partial \alpha_{i,j-1}}{\partial W_{k,l}} + \delta_{i,j=k,l} \alpha_{i-1,j-1}, & i = K,\ j \geq i, \\ 0, & j < i, \end{cases}$$

$$\frac{\partial \alpha_{i,j}}{\partial C_{k,l}} = 0,$$

$$\frac{\partial\beta_{i,j}}{\partial W_{k,l}} = \begin{cases} W_{i,j}\left(\dfrac{\partial\beta_{i-1,j-1}}{\partial W_{k,l}} + C_{i,j}\dfrac{\partial\alpha_{i-1,j-1}}{\partial W_{k,l}}\right) + \\ + p_g\left(\dfrac{\partial\beta_{i,j-1}}{\partial W_{k,l}} + c_g\dfrac{\partial\alpha_{i,j-1}}{\partial W_{k,l}}\right) + \\ + \delta_{i,j=k,l}\left(\beta_{i-1,j-1} + C_{i,j}\alpha_{i-1,j-1}\right), & i = \overline{2,K-1},\ j \geq i, \\ \delta_{i,j=k,l}C_{i,j} + p_g\left(\dfrac{\partial\beta_{i,j-1}}{\partial W_{k,l}} + c_g\dfrac{\partial\alpha_{i,j-1}}{\partial W_{k,l}}\right), & i = 1, \\ W_{i,j}\left(\dfrac{\partial\beta_{i-1,j-1}}{\partial W_{k,l}} + C_{i,j}\dfrac{\partial\alpha_{i-1,j-1}}{\partial W_{k,l}}\right) + \dfrac{\partial\beta_{i,j-1}}{\partial W_{k,l}} + \\ + \delta_{i,j=k,l}\left(\beta_{i-1,j-1} + C_{i,j}\alpha_{i-1,j-1}\right), & i = K,\ j \geq i, \\ 0, & j < i, \end{cases}$$

$$\frac{\partial\beta_{i,j}}{\partial C_{k,l}} = \begin{cases} W_{i,j}\dfrac{\partial\beta_{i-1,j-1}}{\partial C_{k,l}} + p_g\dfrac{\partial\beta_{i,j-1}}{\partial C_{k,l}} + \delta_{i,j=k,l}C_{i,j}\alpha_{i-1,j-1}, & i = \overline{2,K-1},\ j \geq i, \\ W_{i,j} + p_g\dfrac{\partial\beta_{i,j-1}}{\partial C_{k,l}}, & i = 1, \\ W_{i,j}\dfrac{\partial\beta_{i-1,j-1}}{\partial C_{k,l}} + \dfrac{\partial\beta_{i,j-1}}{\partial C_{k,l}} + \delta_{i,j=k,l}C_{i,j}\alpha_{i-1,j-1}, & i = K,\ j \geq i, \\ 0, & j < i. \end{cases}$$

Здесь $\delta_{i,j=k,l} = \begin{cases} 1, & i = k \wedge j = l, \\ 0, & i \neq k \vee j \neq l \end{cases}$ — символ Кронекера.

Аналогичным образом можно получить выражение для производных по p_g и c_g:

$$\frac{\partial\alpha_{i,j}}{\partial p_g} = \begin{cases} \dfrac{\partial\alpha_{i-1,j-1}}{\partial p_g}W_{i,j} + \dfrac{\partial\alpha_{i,j-1}}{\partial p_g}p_g + \alpha_{i,j-1}, & i = \overline{2,K-1},\ j \geq i, \\ \dfrac{\partial\alpha_{i,j-1}}{\partial p_g}p_g + \alpha_{i,j-1}, & i = 1, \\ \dfrac{\partial\alpha_{i-1,j-1}}{\partial p_g}W_{i,j} + \dfrac{\partial\alpha_{i,j-1}}{\partial p_g}, & i = K,\ j \geq i, \\ 0, & j < i, \end{cases}$$

$$\frac{\partial\alpha_{i,j}}{\partial c_g} = 0,$$

$$\frac{\partial\beta_{i,j}}{\partial p_g} = \begin{cases} W_{i,j}\left(\dfrac{\partial\beta_{i-1,j-1}}{\partial p_g} + C_{i,j}\dfrac{\partial\alpha_{i-1,j-1}}{\partial p_g}\right) + \\ + p_g\left(\dfrac{\partial\beta_{i,j-1}}{\partial p_g} + c_g\dfrac{\partial\alpha_{i,j-1}}{\partial p_g}\right) + \beta_{i,j-1} + c_g\alpha_{i,j-1}, & i = \overline{2,K-1},\ j \geq i, \\ p_g\left(\dfrac{\partial\beta_{i,j-1}}{\partial p_g} + c_g\dfrac{\partial\alpha_{i,j-1}}{\partial p_g}\right) + \beta_{i,j-1} + c_g\alpha_{i,j-1}, & i = 1, \\ W_{i,j}\left(\dfrac{\partial\beta_{i-1,j-1}}{\partial p_g} + C_{i,j}\dfrac{\partial\alpha_{i-1,j-1}}{\partial p_g}\right) + \dfrac{\partial\beta_{i,j-1}}{\partial p_g}, & i = K,\ j \geq i, \\ 0, & j < i. \end{cases}$$

$$\frac{\partial\beta_{i,j}}{\partial c_g} = \begin{cases} W_{i,j}\dfrac{\partial\beta_{i-1,j-1}}{\partial c_g} + p_g\left(\dfrac{\partial\beta_{i,j-1}}{\partial c_g} + c_g\alpha_{i,j-1}\right), & i = \overline{2,K-1},\ j \geq i, \\ p_g\left(\dfrac{\partial\beta_{i,j-1}}{\partial c_g} + c_g\alpha_{i,j-1}\right), & i = 1, \\ W_{i,j}\dfrac{\partial\beta_{i-1,j-1}}{\partial c_g} + \dfrac{\partial\beta_{i,j-1}}{\partial c_g}, & i = K,\ j \geq i, \\ 0, & j < i. \end{cases}$$

4. Архитектура классификатора

Приведенные выше формулы позволяют эффективно рассчитывать, насколько хорошо мотив с параметрами Θ и c_g соответствует последовательности x, а также дифференцировать его по соответствующим параметрам. Это позволяет использовать описанный алгоритм в задачах связанных с классификацией последовательностей. В работе предлагается архитектура нейронной сети, в основе которой лежит слой поиска мотивов, принимающий на вход последовательность символов некоторого алфавита и возвращающий n-мерный вектор признаков. Обучаемыми параметрами такого слоя являются n позиционно-весовых матриц мотивов Θ^i и соответствующие им c_g^i. Формируемый таким слоем вектор признаков содержит информацию о том, насколько i-ый мотив соответствует входной последовательности. Дифференцируемость алгоритма выравнивания позволяет оптимизировать параметры такого слоя с помощью стандартных градиентных методов.

Для построения итогового классификатора, такой слой используется в сочетании с обычной полносвязной нейронной сетью. При такой архитектуре к векторам признаков, полученным с помощью выравнивания мотивов, применяется активационная функция ReLU (Rectifier Linear Unit) [15] в сочетании с бетч–нормализацией [14]. После этого полносвязная нейронная сеть формирует финальные вероятности классов.

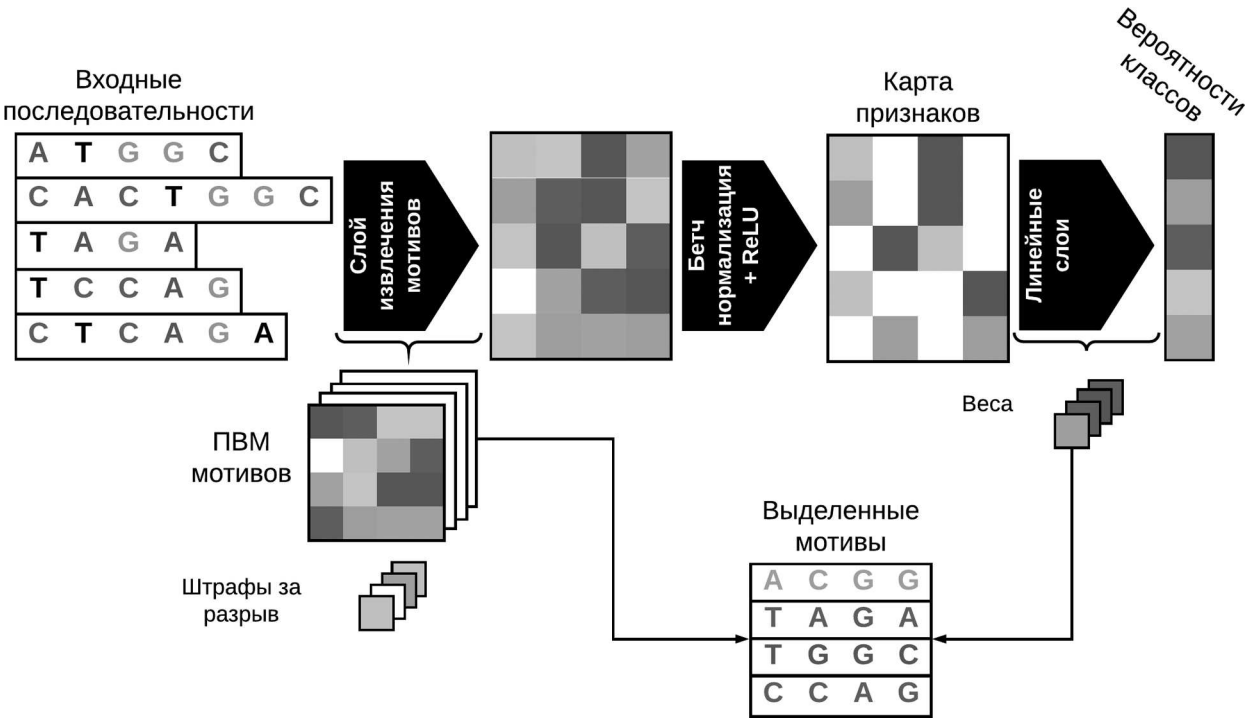

Рис. 2: Предлагаемая архитектура классификатора последовательностей

Преимуществом подобной архитектуры, по сравнению с рекуррентными нейронными сетями, является возможность визуализировать процесс принятия решения с помощью позиционно-вероятностных матриц мотивов.

Если в качестве финального классификатора используется один линейный слой, то его веса так же могут быть использованы для получения дополнительной информации о влиянии мотивов на принадлежность последовательности к определенному классу.

5. Классификация синтетических данных

Для тестирования описанного алгоритма была сгенерирована синтетическая обучающая выборка D_{train}, состоящая из 50000 последовательностей над алфавитом из 4 букв: A, T, G и C, длиной от 5 до 20 символов, разбитых на два класса. Последовательности первого класса были получена из мотива m_1 путем вставок случайных символов алфавита в различных местах мотивах. Количество вставляемых символов n выбиралось случайно в зависимости от позиции вставки — $n \in \overline{0,12}$, для вставок внутри мотива и $n \in \overline{0,15}$ вначале и в конце. По такому же принципу последовательности второго класса были получены из мотива m_2.

Аналогичным образом, используя те же мотивы m_1 и m_2, была сгенерирована тестовая выборка D_{test}.

Последовательности 1-го класса	Последовательности 2-го класса
AGCTcTtcaata	CcgATGC
AGCTTt	cCAggTcatGC
AttccGCTT	atcCATGaggCgc
gccAGCgTcT	CAcggaTGC

Таблица 2: Пример синтетических данных для $m_1 = \text{AGCTT}$, $m_2 = \text{CATGC}$

Выборка D_{train} была использована для обучения нейронной сети, предсказывающей к какому классу принадлежит последовательность. Эксперименты были проведены для различных мотивов m_1 и m_2. Для оценки качества мотивов, найденных нейронной сетью использовалось следующие соотношения:

$$\Delta_1 = \text{Edit}(\tilde{m}_1, m_1), \quad \tilde{m}_1 = \arg\max_i P^1_{i,j},$$

$$\Delta_2 = \text{Edit}(\tilde{m}_2, m_1), \quad \tilde{m}_2 = \arg\max_i P^2_{i,j}.$$

В этих соотношениях $\tilde{m}_1$ и $\tilde{m}_2$ — найденные мотивы, $P^1_{i,j}$ и $P^2_{i,j}$ — соответствующие им позиционно-вероятностные матрицы, Edit — редакционное расстояние.

m_1	m_2	точность на тесте	Δ_1	Δ_2
AGCTT	CATGC	0.98	0	0
ATACT	CTGAC	0.97	0	0
GTTCCA	CACGTG	0.99	0	0

Таблица 3: Результаты классификации для различных мотивов m_1 и m_2

6. Заключение

В работе разработан принципиально новая архитектура нейронная сети, основанная на извлечении мотивов, с использованием алгоритма дифференцируемого выравнивания. Численные эксперименты на синтетических данных подтвердили, что предложенная модель способна правильно находить мотивы с учетом вставок случайных символов и классифицировать последовательности на их основе. Описанные алгоритмы могут быть эффективно распараллелены для вычисления на графических процессорах.

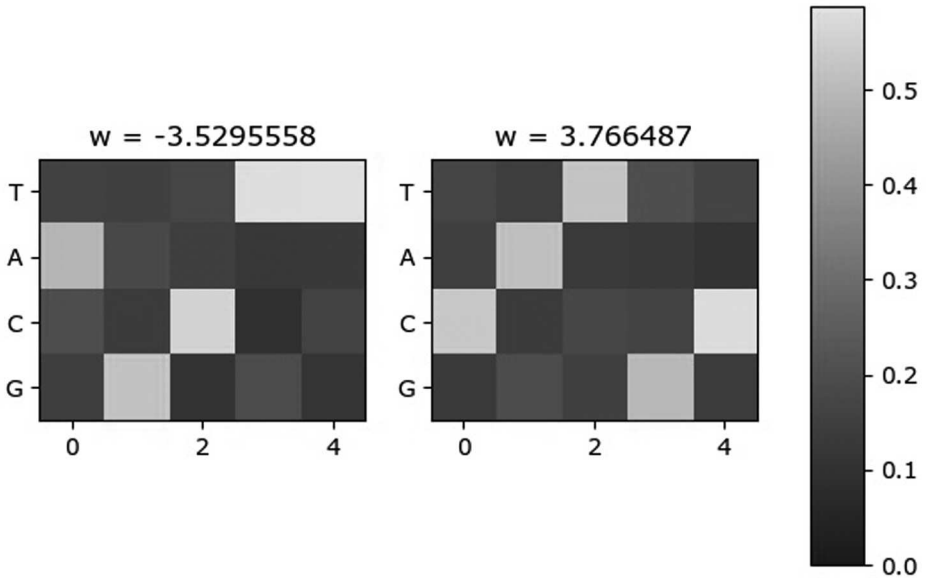

Рис. 3: Визуализация полученных позиционно-вероятностных матриц для m_1 = AGCTT, m_2 = CATGC, w — коэффициент финального линейного классификатора, соответствующий матрице

СПИСОК ЦИТИРОВАННОЙ ЛИТЕРАТУРЫ

1. Hochreiter S., Schmidhuber J. Long short-term memory // Neural computation. 1997. Vol. 9, № 8. P. 1735–1780.

2. Learning phrase representations using RNN encoder-decoder for statistical machine translation / K. Cho [et al.]. // arXiv:1406.1078. 2014.

3. Empirical evaluation of gated recurrent neural networks on sequence modeling / J. Chung [et al.]. // arXiv:1412.3555. 2014.

4. Karpathy A., Johnson J., Fei-Fei L. Visualizing and understanding recurrent networks // arXiv:1506.02078. 2015.

5. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks / H. Strobelt [et al.]. // IEEE transactions on visualization and computer graphics. 2018. Vol. 24, № 1. P. 667–676.

6. Convolutional neural network architectures for predicting DNA–protein binding / H. Zeng et al. // Bioinformatics. 2016. Vol. 32, № 12. P. i121–i127.

7. Zhou J., Troyanskaya O. G. Predicting effects of noncoding variants with deep learning–based sequence model // Nature methods. 2015. Vol. 12, № 10. P. 931.

8. Deep motif: Visualizing genomic sequence classifications / J. Lanchantin [et al.]. // arXiv:1605.01133. 2016.

9. Quang D., Xie X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences // Nucleic acids research. 2016. Vol. 44, № 11. P. e107–e107.

10. Левенштейн В. И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Докл. АН СССР. 1965. Т. 163, № 4. С. 845–848.

11. Smith T. F., Waterman M. S. Comparison of biosequences // Advances in applied mathematics. 1981. Vol. 2, № 4. P. 482–489.

12. Gotoh O. An improved algorithm for matching biological sequences // Journal of molecular biology. 1982. Vol. 162, № 3. P. 705–708.

13. Manavski S. A., Valle G. CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment // BMC bioinformatics. 2008. Vol. 9, № 2. P. S10.

14. Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift // arXiv:1502.03167. 2015.

15. Hahnloser R. H. R. et al. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit // Nature. 2000. Vol. 405, № 6789. P. 947.

REFERENCES

1. Hochreiter, S. & Schmidhuber, J. 1997, “Long short-term memory“, *Neural computation*, vol. 9, no. 8, pp. 1735–1780.

2. Cho K. et al. 2014, “Learning phrase representations using RNN encoder-decoder for statistical machine translation“, *arXiv:1406.1078*.

3. Chung J. et al. 2014, “Empirical evaluation of gated recurrent neural networks on sequence modeling“, *arXiv:1412.3555*.

4. Karpathy, A., Johnson, J. & Fei-Fei, L. 2015, “Visualizing and understanding recurrent networks“, *arXiv:1506.02078*.

5. Strobelt H. et al. 2018, “Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks“, *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 667–676.

6. Zeng H. et al. 2016, “Convolutional neural network architectures for predicting DNA–protein binding“, *Bioinformatics*, vol. 32, no. 12, pp. i121–i127.

7. Zhou, J. & Troyanskaya, O. G. 2015, “Predicting effects of noncoding variants with deep learning–based sequence model“, *Nature methods*, vol. 12, no. 10, pp. 931.

8. Lanchantin J. et al. 2016, “Deep motif: Visualizing genomic sequence classifications“, *arXiv:1605.01133*.

9. Quang D. & Xie X. 2016, “DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences“, *Nucleic acids research*, vol. 44, no. 11, pp. e107–e107.

10. Levenshtein, V. I. 1965, “Binary codes with correction of fallouts, inserts and notes“, *Reports of the Academy of Sciences* (in Russian), vol. 163, no. 4, pp. 845–848. [in Russian]

11. Smith T. F. & Waterman M. S. 1981, “Comparison of biosequences“, *Advances in applied mathematics*, vol. 2, no. 4, pp. 482–489.

12. Gotoh, O. 1982, "An improved algorithm for matching biological sequences", *Journal of molecular biology*, vol. 162, no. 3, pp. 705–708.

13. Manavski S. A., Valle G. 2008, "CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment", *BMC bioinformatics*, vol. 9, no. 2, pp. S10.

14. Ioffe S. & Szegedy C. 2015, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *arXiv:1502.03167*.

15. Hahnloser R. H. R. et al. 2000, Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit", *Nature*, vol. 405, no. 6789, pp. 947.