

Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра математических методов прогнозирования

Кузнецов Максим Дмитриевич

Применение свёрточных нейронных сетей для анализа временных рядов

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:

д.ф.-м.н., профессор

академик РАН

В.В.Рязанов

Москва, 2017

Содержание

1	Введение	3
2	Постановка задачи	3
3	Традиционные методы классификации	5
3.1	DTW-KNN	5
3.2	SAX-BOP	6
3.3	SFA-BOSS	8
3.4	Shapelets	9
4	Нейронные сети	11
4.1	Что такое нейронная сеть?	11
4.2	Свёрточная нейронная сеть	12
5	Применение свёрточных нейронных сетей для классификации временных рядов	14
5.1	Multiscale CNN	14
5.2	GAF-MTF-CNN	15
5.3	MTF-MCNN	18
6	Эксперименты.	20
6.1	Цели	20
6.2	Сравнение нейросетевых методов с традиционными	20
6.3	Сравнение MCNN и MTF-MCNN	22
7	Выводы	23
8	Заключение	23
	Список литературы	25

Аннотация

В последние несколько лет появилось множество работ в области нейронных сетей, которые показали потрясающие результаты во многих областях науки анализа данных - в том числе и области классификации временных рядов.

Целью данной работы является исследование методов классификации временных рядов свёрточными нейронными сетями, их сравнение с традиционными методами, разработанными для данной области за долгое время.

1 Введение

Очень часто каждому из нас приходится иметь дело со значениями какой-то величины, изменяющимися во времени. Прогноз погоды, курсы валют, кардиограмма в больнице – продолжать этот список можно очень долго. В математике такие объекты называются временными рядами, и их анализу и предсказанию посвящён целый раздел науки анализа данных; для решения прикладных задач, связанных с ними, разработаны десятки методов. Одной из задач, которая может возникнуть перед исследователем временных рядов, является задача их классификации. Примером этого может служить кардиограмма, анализируя которую доктор решает, имеет ли человек какие-то сердечные патологии. Не менее интересным примером являются временные ряды из замеров гироскопа и акселерометра, по которым можно с высокой степенью достоверности понять, какое действие человек совершал в определённый промежуток времени. Специалисты по компьютерной безопасности часто исследуют временные ряды из параметров исполняемой программы, чтобы понять, не является ли она вредоносной. Лишь эти несколько примеров уже дают понять, что задача классификации временных рядов – одна из важнейших задач в науке анализа данных. Целью данной работы является исследование классических методов классификации временных рядов, их преимуществ и недостатков, а также более пристальное рассмотрение новых методов, основанных на использовании свёрточных нейронных сетей.

2 Постановка задачи

Пусть исследователь работает с некоторой величиной $x(t)$, которая меняется во времени. Он замеряет эту величину в моменты времени $\{t_1, t_2, \dots, t_n\}$ (обычно предполагается, что моменты времени идут через равные периоды) и получает последовательность значений величины $X = \{x_1, x_2, \dots, x_n\}$. Именно эта последовательность замеров и называется временным рядом.

Задача классификации временных рядов состоит в поиске как можно более точного отображения $f : X \rightarrow \{1, 2, \dots, N\}$, которое принимает на вход временной ряд и ставит ему в соответствие один из N классов. Процесс поиска

этого отображения называется обучением. Обучение осуществляется по обучающей выборке - набору $[(X_1, y_1), (X_2, y_2), \dots, (X_m, y_m)]$, для объектов которого известен ответ искомой функции.

Результат обучения - функция, которая максимизирует качество классификации на отложенной выборке. В качестве меры качества обычно рассматривают точность - процент правильно классифицированных объектов: $Acc = \frac{1}{K} \sum_i^K [f(X_i) = y_i]$, где объекты (X_i, y_i) берутся из отложенной выборки размера K .

3 Традиционные методы классификации

За долгие годы исследований в этой области классификации временных рядов было изобретено множество алгоритмов и их модификаций. Познакомиться с частью этих методов необходимо, чтобы произвести тщательное сравнение подходов, их преимуществ и недостатков по сравнению с нейросетевым подходом к классификации временных рядов.

3.1 DTW-KNN

Пожалуй, одним из самых простых для понимания и реализации методов машинного обучения является метод K ближайших соседей: [1]. Несмотря на свою простоту, он используется при решении многих практических задач. Не обошёл стороной он и задачу классификации временных рядов. Метод находит в обучающей выборке K наиболее похожих на классифицируемый ряд объектов и в качестве ответа берёт самый часто встречающийся среди них класс.

Вот только как находить похожие на классифицируемый ряд объекты из обучающей выборки? Понятие близости формулируется через метрику. Метрика – положительная функция, которая тем меньше, чем ближе объекты, – аналог всем нам привычного расстояния.

Самой простой используемой метрикой является евклидова, которая задаётся так:

$$\rho(r, p) = \sqrt{\sum_{i=1}^n (r_i - p_i)^2}$$

Необходимым условием применимости этой метрики является одинаковая длина всех временных рядов. Кроме того, она накладывает достаточно жёсткие ограничения на похожие временные ряды : они должны быть почти копиями друг друга, без сдвигов и перестановок подрядов. На самом деле это очень сильные ограничения на временные ряды, которые в реальной жизни почти никогда не выполняются.

В попытках смягчить требования к похожим временным рядам была придумана метрика DTW-Euclidean, которая стала почти нечувствительной к сдвигу и изменению масштаба. DTW(сокращённо от Dynamic Time Wrapping) – методика, основанная на динамическом программировании, которая находит для индексов

точек одного ряда неубывающую последовательность индексов второго ряда так, чтобы евклидова метрика оказалась наименьшей. Изначале DTW применялся для обработки речи. Интуиция была такой – в произнесённой фразе паттерны звуков могут только растягиваться/сжиматься и сдвигаться, не меняя свой порядок - именно к этим трансформациям и устойчив DTW.

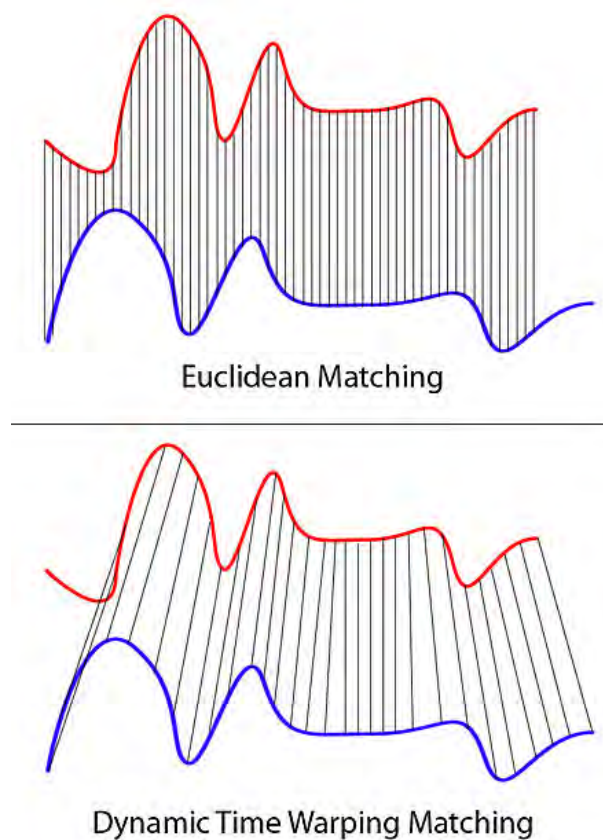


Рис. 1: Отличие DTW-евклидовой метрики от евклидовой

Именно метод K ближайших соседей с метрикой DTW и является одним из самых часто используемых методов классификации временных рядов.

3.2 SAX-BOP

Метод очень похож на популярный в разделе Natural Language Processing метод Bag Of Words [2]. Сначала множество возможных значений измеряемой переменной разбивается на несколько равновероятных интервалов (далее они будут именоваться бинами) и каждому бину ставится в соответствие некоторая метка (для простоты будем считать, что это буква). В дальнейшем, каждое значение во временном ряде

мы заменяем на метку, которая соответствует тому бину, в котором лежит значение. В итоге мы преобразуем временной ряд в последовательность меток. Такой метод преобразования временного ряда в последовательность символов называется Symbolic Aggregation approXimation (сокращенно SAX).

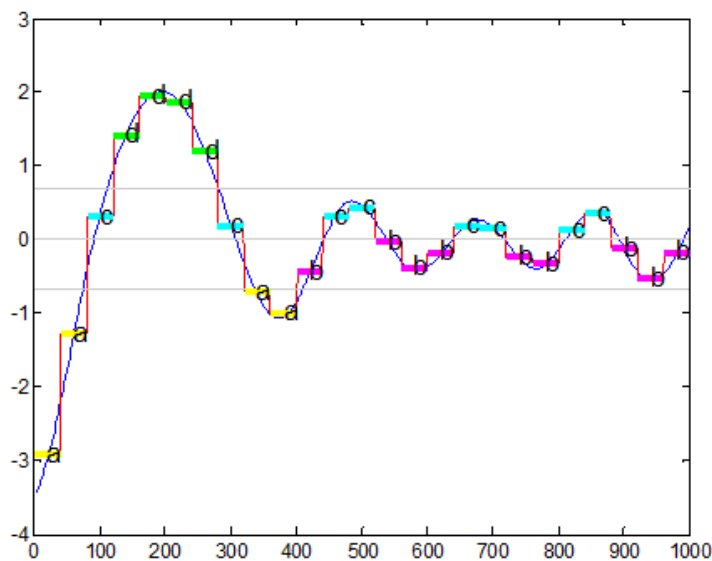


Рис. 2: Разбиение на бины и замена значений на метки, соответствующие этим бинам

Далее из получившейся последовательности мы скользящим окном постоянной ширины выбираем подпоследовательности из меток. Эти подпоследовательности называются паттернами. В итоге формируется вектор, в котором в i -ой позиции стоит количество раз, которые встретился i -ый паттерн. Такой метод дальнейшего преобразования последовательности меток в вектор(как правило, разреженный) называется Bag of Patterns(сокращенно BOP).

Далее над векторами обучают обычные модели машинного обучения(например линейную регрессию или случайный лес).

Достоинства и недостатки

К достоинствам метода относится простота его реализации и малое количество настраиваемых параметров(количество бинов и длины паттернов), в то же время высокие результаты, которые он показывает на популярных датасетах – по этой причине его очень часто берут в качестве baseline-решения. К тому же метод имеет великолепную интерпретируемость : если обучить на векторах BOP модель, которая

позволяет оценить важность компонент (линейную регрессию с регуляризатором L1 или Random Forest например), то можно оценить, наблюдение каких паттернов влияет на выбор того или иного класса.

Недостаток – неиспользование информации о далёких корреляциях наблюдаемой переменной во времени; неспособность метода находить паттерны, у которых может изменять масштаб.

Попытки избавления данного популярного метода от этих недостатков породили несколько модификаций метода. Многие из них жертвуют простотой реализации и настройки, что иногда сводит на нет преимущества алгоритма.

Однако есть способ частично побороть неспособность метода учитывать удалённые корреляции, не пожертвовав при этом простотой метода. Для этого используется Time Warping : мы “прореживаем” исходный временной ряд, оставляя только подпоследовательность наблюдений. Над прореженным временным рядом мы так же делаем SAX-BOP и конкатенируем получившийся вектор к SAX-BOP исходного ряда. Такое прореживание мы можем сделать несколько. Чем сильнее прореживаем ряд – тем более дальние корреляции метод начинает учитывать.

3.3 SFA-BOSS

Метод SFA-BOSS очень похож на SAX-BOP. Идея опять в использовании Bag Of Words над паттернами, однако теперь описание паттернов более хитрое:

1. Сначала над каждым рядом производим дискретное преобразование Фурье и рассматриваем коэффициенты в этом разложении.
2. Для ряда длины n у нас получается n коэффициентов : первые коэффициенты отвечают за низкие частоты, последние – за высокие. Так как шум часто имеет высокочастотную природу, то он будет оседать на последних коэффициентах, поэтому часть последних коэффициентов обычно выкидывают.
3. Для каждого коэффициента мы проделываем то же самое преобразование, что и в SAX – разделяем множество значений коэффициента на равновероятные бины и каждому бину ставим в соответствие какую-то метку. Далее, если значение коэффициента попадает в i -тый бин, то мы заменяем его на

i -тую метку. Очевидно, разбиение на бины будет индивидуальным для каждого коэффициента. Таким образом, мы преобразуем s коэффициентов в последовательность меток длины s .

Это суть SFA(Symbolic Fourier Approximation).

Далее, как и в BOP, мы опять идём скользящим окном по классифицируемому временному ряду и преобразуем подряд в этом окне в символьный паттерн через SFA. В отличие от SAX-BOP, здесь при движении окна паттерны не будут резко меняться, поэтому нам надо будет игнорировать паттерн в скользящем окне, если он совпал с паттерном в предыдущем положении скользящего окна. В итоге из оставшихся паттернов мы формируем вектор методом Bag of Words (который в данном методе назвали Bag Of SFA Symbols, что по сути просто переименование). Как и в SAX-BOP, далее над векторами строятся модели машинного обучения.

Преимущества и недостатки

По сравнению с SAX-BOP, этот метод становится более сложным в реализации, к тому же частично теряет свою интерпретируемость.

К недостаткам метода мы опять отнесём рассмотрение только паттернов, масштаб которых сопоставим со скользящим окном, неспособность выделить паттерны и корреляции, которые сильно разбросаны во времени.

Как и в SAX-BOP, частично этот недостаток можно убрать, используя Time-Wrapping.

К преимуществам стоит отнести его в среднем лучшие, по сравнению с SAX-BOP результаты.

3.4 Shapelets

Шэйплеты – набор паттернов, наличие которых во временном ряде позволяет судить о принадлежности ряда к каким-то классам [3].

Расстояние от временного ряда до шэйплета вычисляется так: идём скользящим окном такого же размера, как и шэйплет, по ряду и считаем евклидово расстояние от шэйплета до подряда в скользящем окне. Тогда расстояние от временного ряда до шэйплета будет минимальное евклидово расстояние до подряда среди всех положений скользящего окна. Идея шэйплетного метода классификации – в выборе

таких шейплетов, чтобы по расстояниям от них до временных рядов в обучающей выборке классы стали хорошо разделимы.

В оригинальной статье выбор таких шейплетов происходит так: добавление шейплетов происходит последовательно – каждый новый шейплет должен наибольшим способом увеличивать Information Gain разделения на классы. Более формально : перебираются все возможные непрерывные подпоследовательности временных рядов различных длин, в предположении, что текущая подпоследовательность – шейплет, ищется расстояние до всех временных рядов, далее подбирается такой порог по расстоянию, разделение классов относительно которого будет наилучшим, считается information gain для этого разбиения. Далее в качестве нового шейплета мы берём такой из перебранных подрядов, который наибольшим способом увеличит Information Gain. Этот метод очень похож на построение дерева решений, за исключением того, на каждом уровне дерева выбирается один и тот же шейплет и один и тот же порог для дальнейшего подразбиения на листы.

Преимущества и недостатки

К преимуществам метода относятся крайне высокая интерпретируемость полученных результатов, малое количество параметров для настройки (количество шейплетов).

К недостаткам – крайне высокая алгоритмическая сложность обучения из-за перебора на каждом шаге всех возможных подрядов – $O(k * n * m^3)$, где k – количество шейплетов, n – размер обучающей выборки, m – максимальная длина временного ряда.

В попытках избавиться от такой высокой алгоритмической сложности было получено много модификаций алгоритма [4, 5], отличие которых – в схемах перебора кандидатов в шейплеты и эвристиках. Увеличения качества метода и его возможностей эти вариации не приносят.

4 Нейронные сети

Несколько прекрасно зарекомендовавших себя методов классификации временных рядов были придуманы в классе нейронных сетей, особенно свёрточных.

4.1 Что такое нейронная сеть?

Учёных давно интересовало, как устроен человеческий мозг. Оказалось, что наш мозг состоит из миллиардов клеток, которые называли нейронами. Отличие нейронов от других клеток состоит в том, что они соединяются между собой и являются накопителями и передатчиками сигналов. Нейрон собирает от соседей, к которым он присоединён дендритами сигнал, преобразовывает его и передаёт к соседям, с которыми он соединён аксоном.

По сути, нейронная сеть в математическом смысле делает то же самое: формально она является графом вычислений, где каждая вершина собирает от соседей, для которых она является непосредственным потомком, значения, преобразует их как-то и присваивает себе это преобразованное значение. Далее её потомки будут использовать это значение, чтобы вычислить свои значения.

Нейроны делятся на три типа:

- **Входные**

Нейроны, значения которых задаются извне. Не имеют предков.

- **Выходные**

Нейроны, значения которых используются извне. Как правило, не имеют потомков.

- **Внутренние**

Нейроны, не относящиеся ни к первому, ни ко второму классу.

Хоть и написано “как-то преобразует” значения, на самом деле почти всегда используется модель нейрона как на рисунке ниже:

При такой структуре нейрон собирает с некоторыми весами значения предков, суммирует их, а дальше к сумме применяет некоторую нелинейную функцию.

В качестве нелинейных функций обычно выступают эти три:

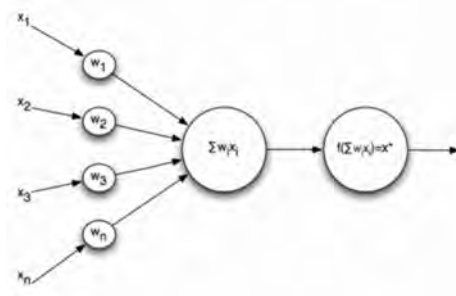


Рис. 3: Модель нейрона

- **Sigmoid**

$$S(x) = \frac{1}{1+e^{-x}}$$

- **Tanh**

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- **ReLU**

$$ReLU(x) = \max(0, x)$$

Кроме того, во многих архитектурах современных нейронных сетей нейроны образуют слои, которые упорядочены. Внутри одного слоя нейроны не связаны между собой, связи имеются только с соседними слоями – входы берутся с предыдущего слоя, вычисленное значение передаётся на следующий по порядку. Первый слой состоит из входных нейронов, поэтому он называется входной. Последний- из выходных, поэтому его название - выходной. Аналогично все остальные слои называются внутренними.

4.2 Свёрточная нейронная сеть

Описание нейронной сети в предыдущем пункте достаточно общее – существуют давно зарекомендовавшие себя архитектуры нейронных сетей, которые используют для решения тех или иных задач. Одной из самых известных архитектур является свёрточная. Её особенность заключается в использовании свёрточных слоёв и пулинговых внутри себя.

Само понятие свёртки появилось задолго до повсеместного использования нейронных сетей. Представьте, что у вас есть чёрно-белое изображение – по сути

матрица некоторого размера из вещественных элементов – яркостей пикселей. И есть матрица из весов. Выберем в исходном изображении подматрицу такого же размера, как и матрица из весов, поэлементарно умножим их, а после этого сложим результаты умножения. Получится некоторое значение – значение свёртки на подматрице исходного изображения. Выбирая разные подматрицы и применяя к ним свёртку, мы получим множество вещественных чисел – результат применения свёртки к изображению.

На самом деле свёртки - более общий аппарат, двумерное изображение было выбрано как пример, на котором можно легко объяснить суть. Тензор может иметь любое количество размерностей, кроме того у тензора выделяют одну размерность как размерность каналов. В случае, если тензор имеет d каналов, то и свёртка тоже будет иметь d каналов.

Операция пулинга на результатах свертки – преобразование этого множества результатов в меньшее количество элементов. Например, max-pooling выдаёт наибольшее из результатов свёртки на каком-то подмножестве результатов, mean-pooling находит среднее арифметическое результатов.

Мотивация этих слоёв такая. Каждая свёртка кодирует некоторый паттерн, и если мы его встретим в исходной матрице, то в том месте, где он есть, результат применения свёртки будет большим. Max-pooling позволит нам обнаружить факт наличия паттерна, независимо от его местоположения. Mean-pooling позволит оценить среднюю выраженность паттерна в матрице, к которой применяем свёртку.

К исходной матрице применяются несколько свёрток, тем самым проверяя наличие в матрице сразу нескольких паттернов.

Весы свёрток и соответственно паттерны, соответствующие им, определяются в результате обучения нейронной сети так, чтобы как можно сильнее повысить качество классификации.

Типичная свёрточная нейронная сеть имеет несколько свёрточных и пулинговых слоёв, расположенных последовательно. В конце расположены полносвязные слои и softmax-слой, который выдаёт вероятности классов.

Если заменить размерность изображения и свёрток с двойки на один, то получится описание свёрточной нейронной сети для временных рядов.

5 Применение свёрточных нейронных сетей для классификации временных рядов

5.1 Multiscale CNN

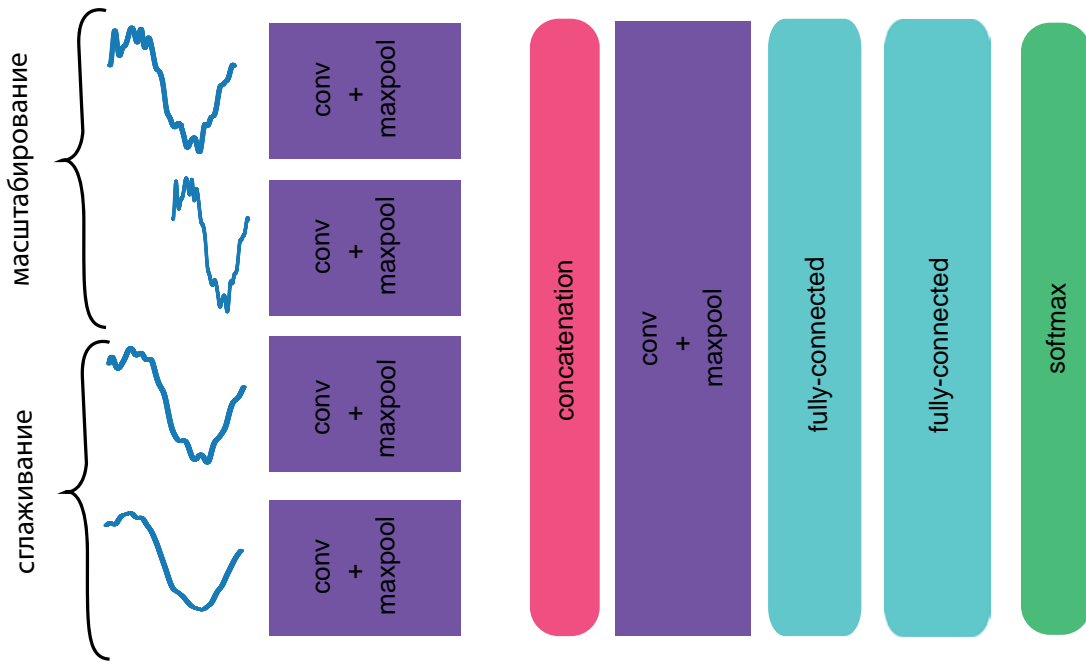


Рис. 4: Структура MCNN

Метод, предложенный в [6], отличается от обычной CNN тем, что применяет свёртки и пулинг к разным преобразованиям временного ряда – для каждого преобразования свои свёртки. В качестве преобразования подразумеваются изменение масштаба, сглаживание. Далее результаты применения конкатенируются в одно целое, и к ним ещё раз применяется свёртки+пулинг, в конце стоит полносвязный слой и softmax-слой.

Преобразование к произвольному масштабу можно сделать с помощью дискретного преобразования Фурье:

- Сначала найти амплитуды синусоидальных сигналов в исходном временном ряде $\{x_1, x_2, \dots, x_n\}$:

$$X_k = \sum_{s=1}^n x_s e^{-\frac{2\pi i}{n}(k-1)s}, k = 1, \dots, n$$

- Получить временной ряд нужной длины m по следующей формуле:

$$\hat{x}_j = \frac{1}{n} \sum_{s=1}^n X_s e^{\frac{2\pi i}{m}(s-1)j}, j = 1, \dots, m$$

Данная архитектура показывает значимое улучшение качества классификации по сравнению с обычной CNN на исходном ряде без преобразований, и оно ничуть не удивительно – во временном ряде могут присутствовать паттерны разного масштаба, наличие которых и позволяет отнести временной ряд к тому или иному классу. Обычной одномасштабной CNN будет трудно углядеть паттерны намного больших размеров, чем размеры свёрток, приведённой же выше архитектуре преодолеть эту проблему не составляет особого труда.

5.2 GAF-MTF-CNN

В работе [7] был предложен ещё один метод использования свёрточных нейронных сетей для классификации временных рядов. В этом методе временной ряд преобразуется в изображения двумя способами, а дальше применяется обычная свёрточная нейронная сеть для изображения.

GAF

Матрица GAF (Gramian Angular Field) - это матрица, построенная по ряду $X = \{x_1, x_2, \dots, x_n\}$ следующим образом:

1. Сначала ряд нормируется в отрезок $[-1, 1]$ так: $\hat{x}_i = \frac{(x_i - \max(X)) + (x_i - \min(X))}{\max(X) - \min(X)}$
2. Далее полученные значения переводятся в полярную систему координат следующим образом:

$$\begin{cases} \phi_i = \arccos(x_i) \\ r_i = \frac{t_i}{N} \end{cases}$$

3. Сама матрица GAF вычисляется по такой формуле:

$$G = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \cdots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \cdots & \cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \cdots & \cos(\phi_n + \phi_n) \end{bmatrix}$$

Или же проще - через прежде посчитанный вектор значений \hat{X} и единичный вектор $I = [1, 1, \dots, 1]$:

$$G = \hat{X}^T * \hat{X} - (\sqrt{I - \hat{X}^2})^T * (\sqrt{I - \hat{X}^2})$$

Итоговая матрица сохраняет всю информацию о ряде, кроме исходных границ значений x_i , которые мы теряем в шаге (1) после масштабирования - то есть мы можем по полученной матрице восстановить исходный ряд но только отмасштабированный в отрезок $[-1, 1]$.

МТФ

А вот исходные границы ряда и распределение значений ряда, которые мы теряем при применении свёрток+пулинга к GAF, сохраняются в матрице МТФ (Markov Transition Field):

1. Разбиваем всё множество значений наблюдений на m квантильных бинов - отрезки с одинаковой вероятностью попадания значения наблюдения в каждый из них. Это можно просто сделать с помощью обучающей выборки - объединить все значения наблюдений переменной в одно множество, отсортировать его, а дальше так расставить в нём $m - 1$ границ, чтобы между двумя соседними границами было примерно одинаковое количество значений. Пространство между двумя соседними границами и будет квантильным бином.
2. Пусть w_{ij} - это количество пар соседних наблюдений во временном ряде, который мы рассматриваем - таких, что левое наблюдение лежит в i -том бине, а правое в j -том бине. \hat{w}_{ij} - это отнормированные w_{ij} так, что $\sum_i \hat{w}_{ij} = 1, \forall j$. Тогда матрица МТФ для временного ряда - это матрица вероятностей переходов между бинами:

$$M = \begin{bmatrix} \hat{w}_{11} & \cdots & \hat{w}_{1m} \\ \hat{w}_{21} & \cdots & \hat{w}_{2m} \\ \vdots & \ddots & \vdots \\ \hat{w}_{m1} & \cdots & \hat{w}_{mm} \end{bmatrix}$$

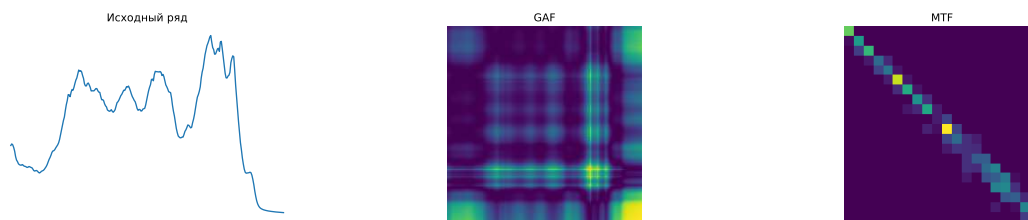


Рис. 5: Преобразование исходного ряда в MTF и GAF

Итоговая архитектура

Несмотря на то что полученные матрицы не являются изображениями, к ним вполне применимы методы классификации с помощью свёрточных нейронных сетей. В оригинальной статье над GAF/MTF матрицами применяют Tiled CNN.

Применение к многомерным временным рядам Можно очень просто модифицировать данный метод так, чтобы он классифицировал многомерный временной ряд. Достаточно делать MTF и GAF отдельно для каждого канала многомерного временного ряда - соответственно и свёртки в CNN будут многомерными.

Достоинства и недостатки

Пожалуй, самым главным недостатком данного метода является квадратичная сложность хранения временного ряда (т.к. происходит преобразование временного ряда размера n в матрицы размера $n \times n$ и $m \times m$) и, соответственно, квадратичная сложность применения свёрток. Данный недостаток решаются следующими двумя способами:

- Разбиением длинного временного ряда на несколько маленьких рядов длины n так, чтобы $n \times n$ было приемлемым числом, а далее вычислением класса как наиболее часто встречающегося предсказанного класса для этих подрядов. Однако для такого метода возникает проблема - обучаться придётся на подрядах. Было бы логично присвоить каждому подряду метку класса исходного ряда, однако не всегда это будет правильным решением : мы можем взять подряд, не обладающий ярко выраженными признаками класса, и тогда нейронная сеть будет пытаться обучиться под эту "пустышку". Примером может послужить временной ряд сигнала и выбранный подряд, состоящий

из шума или тишины. В такой ситуации нужна ручная или эвристическая фильтрация таких "пустышек".

- Понижением размера ряда. Это можно сделать либо путём выкидывания промежуточных значений, либо путём интерполирования значений в нужные моменты времени через дискретное преобразование Фурье.

Кроме того, к недостатками также следует отнести почти нулевую интерпретируемость модели - малую информативность полученных из Tiled CNN свёрток, невозможность выяснить, какие части ряда влияли на выбор того или иного класса.

К плюсам стоит отнести высокую точность классификации на популярных датасетах.

5.3 MTF-MCNN

MCNN, несмотря на применение свёрток на разных преобразованиях исходного ряда, всё равно теряет некоторую глобальную информацию о ряде - о распределении значений в нём, о их взаимосвязи. Это связано с локальностью свёрток и применением пулинга.

В то же время, метод GAF-MTF-CNN менее подвержен потери глобальной информации о ряде из-за использования матрицы MTF при классификации.

Так почему бы не объединить сильные стороны этих методов?

Давайте возьмём MCNN, однако добавим к ней ещё часть, которая будет обрабатывать MTF. Эта часть будет состоять из двух полносвязных слоёв и объединяться с MCNN только на последнем полносвязанном слое.

Комбинированная архитектура показана ниже.

Добавление матрицы MTF и части нейронной сети, обрабатывающей её, значительно увеличивает количество параметров для настройки. При взгляде на MTF становится очевидным, что ненулевыми являются лишь ячейки около диагонали. Так почему бы не оставить лишь k околodiагональных строк, а все остальные выбросить? Тогда одновременно уменьшится размер входных данных - вместо матрицы MTF размера $m \times m$ можем передавать лишь её часть размера $m \times k$, соответственно уменьшится количество параметров сети. Параметр k должен оцениваться аналитиком. В случае

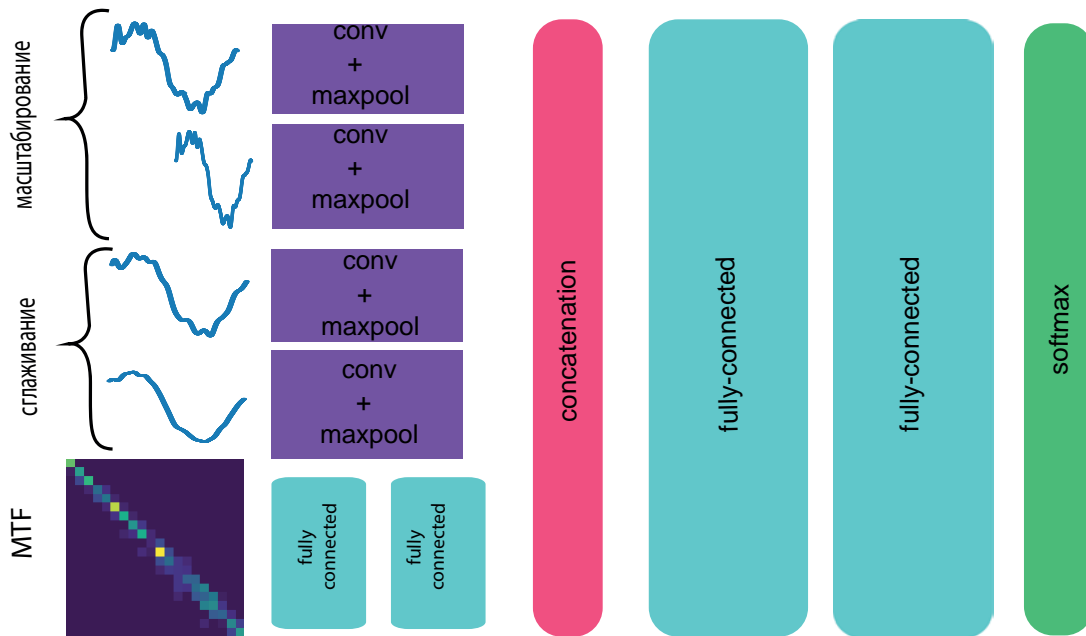


Рис. 6: Структура предложенной модификации MCNN-MTF

многомерного временного ряда, параметр k должен подбираться индивидуально для каждого канала.

6 Эксперименты.

6.1 Цели

Было проведено несколько экспериментов, целью которых было:

- Сравнить нейросетевые подходы с традиционными по качеству классификации.
- Сравнить методы MCNN и MTF-MCNN с целью понять, улучшает ли качество классификации дополнительная информация из матрицы MTF.

6.2 Сравнение нейросетевых методов с традиционными

Датасеты

Набор датасетов и результаты классификации на них для традиционных методов были взяты с [8] - сайта, который создали и поддерживают в актуальном состоянии пять ученых University of East Anglia. Мера качества алгоритмов классификации - точность. Особенностью этого набора датасетов является его разнообразность. Там есть датасеты разной природы - они помечены метками IMAGE, SPECTRO, SENSOR, SIMULATED, ECG, MOTION, DEVICE. Количество классов в датасетах варьируется от 2 до 60. Сильно варьируется и длина временных рядов в датасетах.

Результаты

Ниже приведена таблица с результатами на самых показательных датасетах.

Видно, что нейросетевые подходы показали себя прекрасно - имели качество либо наравне с традиционными методами, либо обгоняли их.

Достойную конкуренцию нейросетевым подходам составляет Shapelet-метод классификации временных рядов. Это и не удивительно - ведь он, по сути, делает то же, что и нейросетевые подходы, - ищет паттерны, по которым классы начинают быть хорошо разделимыми.

Кроме того, стало очевидно, что из нейросетевых подходов почти всегда наихудшее качество у GAF-MTF-CNN, среднее у MCNN и с достаточно серьёзным отрывом лидирует MTF-MCNN.

Название	Тип	Классов	Длина	Об.выборка	Тест.выбока
CBF	SIMULATED	3	128	30	900
Mallat	SIMULATED	8	1024	55	2345
Coffee	SPECTRO	2	286	28	28
Strawberry	SPECTRO	2	235	613	370
ECG5000	ECG	5	140	500	4500
NonInvasiveFatalECGThorax2	ECG	42	750	1800	1965
StarlightCurves	SENSOR	3	1024	1000	8236
FordA	SENSOR	2	500	3601	1320
UWaveGestureLibraryAll	MOTION	8	945	896	3582
InlineSkate	MOTION	7	1882	100	550

Таблица 1: Описание датасетов

Name	DTW-1NN	SAX-BOP	SFA-BOSS	ST	MCNN	GAF-MTF	MTF-MCNN
CBF	0.9933	0.9628	0.9981	0.9856	0.9951	0.9705	0.9979
Mallat	0.9422	0.8522	0.9486	0.9723	0.9527	0.9289	0.9613
Coffee	0.9889	0.9436	0.9886	0.9950	1.0	0.9305	1.0
Strawberry	0.9554	0.9645	0.9703	0.9684	0.92	0.9235	0.9510
ECG5000	0.9256	0.9098	0.9405	0.9434	0.9404	0.9381	0.9413
ECGThorax2	0.8606	0.7452	0.9036	0.9539	0.8791	0.82	0.8914
StarlightCurves	0.9143	0.9430	0.9776	0.9774	0.9542	0.9241	0.9621
FordA	0.5769	0.7942	0.9195	0.9654	0.9314	0.9355	0.9589
UWaveGestures	0.8955	0.8138	0.9445	0.9421	0.8523	0.8264	0.9301
InlineSkate	0.3951	0.3883	0.5027	0.3930	0.4115	0.3887	0.4391

Таблица 2: Результаты методов на датасетах

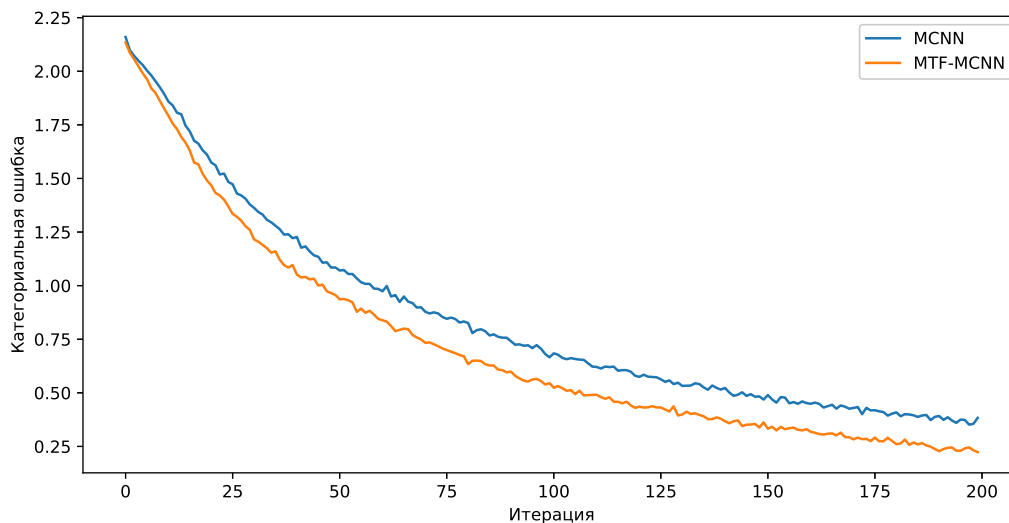


Рис. 7: Зависимость ошибки от итерации

6.3 Сравнение MCNN и MTF-MCNN

Из предыдущей таблицы видно, что предложенный метод MTF-MCNN почти всюду показывает себя лучше, чем оригинальный MCNN. Эта ситуация требует более детального исследования.

Во-первых, добавление информации о MTF сильно ускоряет сходимость нейронной сети. Это можно заметить в графике ниже, где показано изменение категориальной ошибки для MCNN и MTF-MCNN. MTF-MCNN имеет приблизительно ту же конфигурацию свёрточных частей, однако добавлена ещё полносвязная часть на матрице MTF размера 20×20 . Измерения производились на датасете UWaveGestureLibraryAll.

Кроме того, было исследовано качество MCNN и MTF-MCNN на отложенной выборке в зависимости от количества параметров в сети. В графике ниже видно, что при одинаковом количестве параметров MTF-MCNN показывает результаты лучше, т. к. использует уже готовую информацию из матрицы MTF, а не подстраивает свои параметры, чтобы извлекать эту информацию.

Исходя из проведённых экспериментов, можно сделать вывод, что добавление матрицы MTF в нейронную сеть значительно повышает качество классификации.

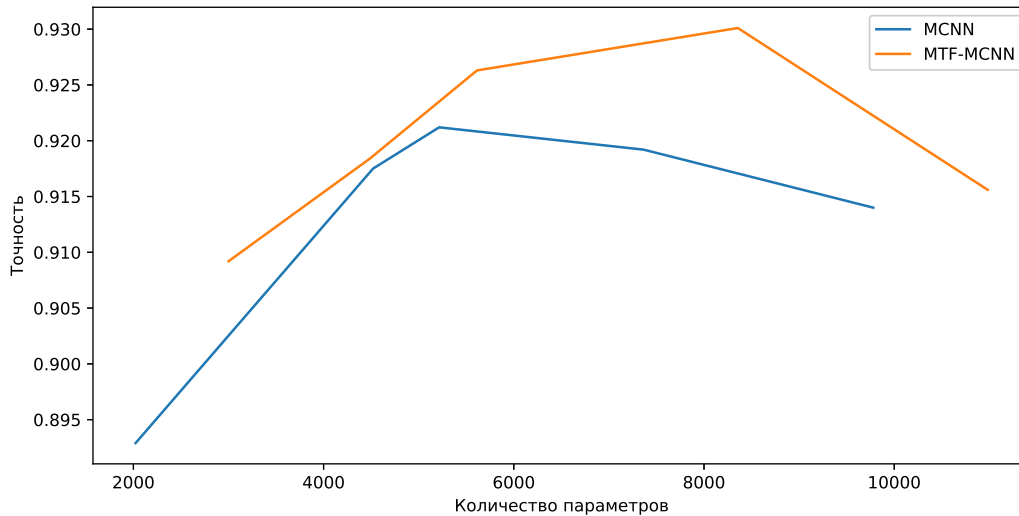


Рис. 8: Зависимость максимально достигаемого уровня качества от количества параметров

7 Выводы

На основании проведённых экспериментов можно сделать следующие выводы:

- Свёрточные нейронные сети - мощный инструмент для классификации временных рядов. Они очень редко проигрывают в качестве традиционным методам, а иногда даже и превосходят их результаты.
- В среднем метод GAF-MTF-CNN хуже, чем MCNN. Это происходит из-за того, что исходный, как правило длинный ряд, приходится перемасштабировать к короткому, чтобы матрицы MTF и GAF не занимали много места в памяти, что приводит к потере информации об исходном временном ряде.
- Добавление в метод MCNN матрицы MTF значительно улучшает метод в плане качества результирующей классификации и в плане сходимости нейронной сети.

8 Заключение

В данной работе было проведено исследование и сравнение свёрточных нейронных сетей и традиционных методов в задаче классификации временных рядов. Кроме

двух разных нейросетевых подходов был исследован и реализован их гибрид, который показал себя значимо лучше, чем эти два подхода.

Список литературы

- [1] Van The Huy и Duong Tuan Anh. “An efficient implementation of anytime k-medoids clustering for time series under dynamic time warping”. В: *Proceedings of the Seventh Symposium on Information and Communication Technology, SoICT 2016, Ho Chi Minh City, Vietnam, December 8-9, 2016*. 2016, с. 22—29. DOI: 10.1145/3011077.3011128. URL: <http://doi.acm.org/10.1145/3011077.3011128>.
- [2] Zhiguang Wang и Tim Oates. “Time Warping Symbolic Aggregation Approximation with Bag-of-Patterns Representation for Time Series Classification”. В: *13th International Conference on Machine Learning and Applications, ICMLA 2014, Detroit, MI, USA, December 3-6, 2014*. 2014, с. 270—275. DOI: 10.1109/ICMLA.2014.49. URL: <http://dx.doi.org/10.1109/ICMLA.2014.49>.
- [3] L. Ye и E. Keogh. “Time series shapelets: a new primitive for data mining”. В: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2009.
- [4] Jason Lines и др. “A shapelet transform for time series classification”. В: *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*. 2012, с. 289—297. DOI: 10.1145/2339530.2339579. URL: <http://doi.acm.org/10.1145/2339530.2339579>.
- [5] Martin Wistuba, Josif Grabocka и Lars Schmidt-Thieme. “Ultra-Fast Shapelets for Time Series Classification”. В: *CoRR* abs/1503.05018 (2015). URL: <http://arxiv.org/abs/1503.05018>.
- [6] Zhicheng Cui, Wenlin Chen и Yixin Chen. “Multi-Scale Convolutional Neural Networks for Time Series Classification”. В: *CoRR* abs/1603.06995 (2016). URL: <http://arxiv.org/abs/1603.06995>.
- [7] Zhiguang Wang и Tim Oates. “Spatially Encoding Temporal Correlations to Classify Temporal Data Using Convolutional Neural Networks”. В: *CoRR* abs/1509.07481 (2015). URL: <http://arxiv.org/abs/1509.07481>.
- [8] *Time Series Classification Web-Site*. <http://timeseriesclassification.com>.