

Построение высокопроизводительного сервиса интеллектуальной обработки медицинской информации для оценки электромиографических исследований поперечно-полосатых мышц

А.В. Богинич

*Уральская государственная медицинская академия
Boginich_A.V@usma.ru*

М.Л. Морнев, В.Ю. Попов

*Уральский государственный университет им. А.М.Горького
max.mornev@gmail.com, Vladimir.Popov@usu.ru*

Электромиографические исследования поперечно-полосатых мышц позволяют получить информацию о динамике изменения во времени поверхностного потенциала левой и правой нижних конечностей. Для этой информации можно получить экспертные оценки патологий, сделанные специалистами. Кроме того, существуют математические модели, позволяющие получать аналогичные оценки. Эти оценки можно использовать для обучения интеллектуальных систем.

На настоящем этапе развития электромиографических исследований поперечно-полосатых мышц ставится задача разделения входных данных только на два класса: данные здоровых людей и данные людей с патологией соответствующих мышц.

Исходные медицинские данные представляют собой пары наборов действительных чисел. Каждый набор получен в результате дискретизации сигнала, снятого с мышцы. Шаг дискретизации равен 0,01 с. Первый набор в паре отвечает за показатели правой мышцы, а второй - левой. Числа изменяются в диапазоне от -1000 до 1000 и указываются с точностью двух знаков после запятой. Наборы данных, полученные от разных испытуемых, как правило, имеют разный размер. Во время испытания все тестируемые проходили одинаковый маршрут. Поскольку нормальная скорость передвижения здорового человека не может быть ниже определенного порогового значения, время, затрачиваемое здоровым испытуемым на преодоление дистанции ограничено сверху некоторой константой. Если испытуемый не справился с дистанцией за это время, можно определенно считать, что он болен. Соответственно, можно считать, что все тестовые выборки, включая еще не полученные, имеют максимальное ограничение на размер.

Для анализа подобного сорта данных естественно использовать нейронные сети. Представляет интерес рассмотрение как рекуррентных нейронных сетей, так и односторонних. В данном случае для исследования были выбраны модифицированная сеть Хемминга (см. [1]) и многослойный персептрон.

Учитывая то, что рассматриваемые типы нейронных сетей имеют фиксированное количество входов, а медицинские данные - переменную длину, необходимо специальное соглашение о кодировании данных на входе. Существует несколько способов классификации сигналов, зависящих от времени, с помощью нейронных сетей. Простейший способ состоит в дополнении всех тестовых наборов нулями до одинакового размера, и подаче их на вход многослойного персептрона. В силу того, что тестовые наборы имеют естественное верхнее ограничение на размер, описанное выше, такой подход позволит создать сеть, пригодную, как минимум, для одного используемого способа обследования. Преимущество этого способа в простоте реализации и скорости обучения. Недостаток - необходимость сбора большой базы примеров для обучения. Причина последнего явления в том, что для достаточно большого размера входа многослойная сеть будет иметь большое количество весов. Соответственно, велика вероятность такого

разделения пространства примеров, которое корректно на обучающей выборке, но при этом практически не экстраполируется. Создание большой базы для обучения наталкивается также на еще одну проблему, специфичную для решаемой задачи. А именно, количество здоровых испытуемых, которых можно привлечь к эксперименту, по понятным причинам, весьма ограничено. Соответственно, при возрастании размера базы тестовых выборок, диспропорция между объемом данных здоровых и больных пациентов будет увеличиваться.

Также, имеет смысл рассмотрение рекуррентной сети Хэмминга. Поскольку сеть Хэмминга работает с биполярными сигналами, а данные в обучающей выборке представлены векторами чисел с фиксированной точностью, требуется соглашение о перекодировании таких векторов в биполярные векторы. Подходящим вариантом является представление каждого числа в векторе как двоичного числа с фиксированной точкой. В нашем случае будет достаточно десяти разрядов до точки и четырех разрядов после.

Чтобы упростить обучение сети Хэмминга, нам следует постараться увеличить число ее выходов. Мы можем поступить следующим образом. Дополним биполярные векторы для обучения сети Хэмминга еще одним разрядом. Этот разряд будет хранить информацию о диагнозе: болен или здоров. На вход сети Хэмминга будет подаваться вектор, значение дополненного разряда которого будет фиксированным. Например, единица. Выход сети будет соответствовать наиболее похожему вектору из обучающей выборки, но уже с диагнозом в дополнительном разряде. Эталонный вектор на выходе мы будем отбрасывать, оставляя только разряд диагноза.

Теперь обратим внимание на возможность распараллеливания описанных типов сетей. В принципе, алгоритм работы многослойного персептрона хорошо поддается распараллеливанию вручную. Но методы автоматического распараллеливания с применением интеллектуальных алгоритмов также заслуживают внимания. Опишем некоторые подобные методы.

Рассмотрим произвольный многослойный персептрон. В большинстве случаев, персептроны являются полносвязными, т. е. все выходы нейронов n -го слоя служат входами для всех нейронов слоя $n+1$. Соответственно, даже после распараллеливания, каждый последующий слой должен дожидаться результатов вычисления предыдущего. Поэтому, для получения высоких коэффициентов ускорения, существует препятствие, связанное с необходимостью обмена данными между вычислительными узлами. Мы можем попытаться избавиться от этой необходимости следующим образом. Подберем некоторое множество межнейронных соединений и удалим его из нейронной сети. Так мы добьемся некоторого уменьшения количества обменов на итерациях вычислений. В качестве исходной конфигурации сети для подобного метода можно рассмотреть результат применения алгоритма Optimal Brain Surgery к исходной сети. Дальнейшее исключение межнейронных соединений может быть проведено генетическим алгоритмом. Значение функции приспособленности может быть рассчитано на основании получившегося коэффициента ускорения и величины ошибки, по сравнению с исходной нейронной сетью. Поскольку число возможных вариантов удаления межнейронных соединений экспоненциально относительно числа нейронов, имеет смысл рассмотрение способа исключения соединений, дающего меньшее количество вариантов для перебора. Например, мы можем разделить исходную сеть на несколько не ориентированно связанных компонент, получив, таким образом, коэффициент ускорения, равный числу компонент.

Развивая этот метод дальше, мы можем попробовать дообучить получившуюся нейронную сеть. Следует отметить, что в силу самой структуры алгоритма обратного распространения ошибки, он будет распараллеливаться точно также, как и алгоритм вычисления выхода нейронной сети. Полученная после дообучения ошибка будет ис-

пользоваться для расчета значения функции приспособленности. Конечно, такой метод будет работать существенно дольше, чем описанный абзацем выше. Но он, потенциально, способен дать гораздо большее количество вариантов распараллеливания исходной сети, уже с другими весами. В частности, не будут исключены некоторые варианты, работающие с большим коэффициентом ускорения, но дающие слишком высокую ошибку с весами, взятыми из исходной сети. Чтобы несколько сократить время распараллеливания по этому методу, можно действовать в два этапа. На первом этапе подбираются сети с не очень большой ошибкой и достаточно большим коэффициентом ускорения. На втором этапе эти сети подвергаются дообучению и дальнейшему исключению межнейронных соединений.

Аналогичный подход применим и в случае использования рекуррентных нейронных сетей.

Описанные методы разделения данных на данные больных и здоровых пациентов можно эффективно получить на случай постановки более точного диагноза. При этом, можно ограничиться созданием одной нейронной сети для каждого типа болезни, в тандеме с полученной на предыдущем этапе сетью, отличающей здоровых пациентов от больных. Когда ни одна из сетей, отвечающих за отдельные типы болезней, не распознает некоторый вектор как свой, но в то же время, сеть классификации векторов больных и здоровых пациентов опознает этот вектор, как вектор больного пациента, у нас открывается две возможности. Либо рассматриваемый вектор принадлежит одному из уже имеющихся типов болезней. Тогда нам необходимо дообучить с его помощью соответствующую нейронную сеть. Либо же рассматриваемый вектор относится к новому, пока еще неизвестному типу болезней. Тогда мы должны создать новую нейронную сеть, отвечающую за этот тип болезней. После этого мы должны осуществить ее постепенное обучение. Для принятия соответствующего решения нам необходим дополнительный интеллектуальный модуль.

Рассмотрим теперь эксперименты по анализу способности сетей распознавать больных и здоровых пациентов по данным электромиографического исследования. Нами был проведен ряд экспериментов с многослойным персептроном на небольшой базе обучающих выборок. Нейронные сети создавались с помощью пакета GANS [2], написанного на C# и работающего на платформе .NET 2.0. Выбор платформы .NET и языка C# для реализации нейронных сетей связан с тем, что C# и .NET позволяют, благодаря автоматической сборке мусора и наличию богатой и удобной библиотеки времени исполнения, серьезно сократить время разработки. В то же время, как показывает исследование [3], на тестах целочисленной и плавающей арифметики, виртуальная машина Microsoft .NET проигрывает коду, написанному на C++ и скомпилированному оптимизирующим компилятором gcc, не более чем в 3.9 раз. Более того, в [3] указывается, что в некоторых тестах, а именно, Division-intensive loop и Polynomial evaluation, виртуальная машина .NET превосходит по производительности код, сгенерированный gcc, на 2%--20%. Соответственно, применение .NET не приведет к катастрофическим потерям в производительности. Кроме того, GANS позволяет достаточно просто выбрать оптимальную архитектуру нейронной сети из некоторых возможных, а также, подобрать алгоритмы обучения и их параметры.

Во всех экспериментах с многослойным персептроном в качестве алгоритма обучения применялось обратное распространение ошибки. Сети состояли из одного скрытого слоя и одного выходного. Размер скрытого слоя варьировался, а выходной слой всегда содержал два нейрона. Сети обучались так, чтобы на выходных нейронах получались оценки, в интервале от 0 до 1, принадлежности обучающего вектора к классу показаний больных испытуемых, или здоровых, на первом и втором нейронах соответственно. Обучающая выборка была столь мала, что даже при скромном размере скры-

того слоя, порядка 50 нейронов, сеть была в состоянии запомнить все обучающие вектора. Соответственно, было принято решение попытаться искусственно нарастить размер обучающей выборки. А именно, обучающие примеры разрезались на куски одинакового размера, и в таком виде подавались на вход нейронной сети. Такой эксперимент позволяет установить размер характерного участка набора данных испытуемого, по которому уже можно судить, болен он или здоров. Было установлено, что уже при разрезании обучающего вектора на векторы, соответствующие отрезкам экспериментальных данных длиной в 0,6 секунды, нейронная сеть с одним нейроном в скрытом слое и двумя нейронами в выходном может распознать всю обучающую выборку с ошибкой меньше 5%. Другой вариант увеличения обучающей выборки - это применение "плавающего окна". А именно, по каждому обучающему вектору, от одного его конца до другого, движется с фиксированным шагом окно фиксированного размера. Координаты, попадающие в это окно, становятся новым обучающим вектором. Например, при длине исходного обучающего вектора 1200, размере окна 60 и шаге 1, мы получаем 1141 новый обучающий вектор. Эксперименты с плавающим окном показали, что даже для окна длины 1 с. и с шагом 0,1 с. требуется увеличение размера скрытого слоя как минимум в 10 раз, по сравнению со случаем разрезания исходных векторов на непересекающиеся куски.

Все эксперименты по исследованию поведения нейронных сетей при различных параметрах архитектуры и алгоритма обучения проводились на кластере, путем простого распределения вариантов сетей на вычислительные узлы. Подбор параметров занимал 8-12 часов даже для скромных размеров сетей, содержащих меньше 100 нейронов в скрытом слое.

Выше было отмечено, что данные испытаний здоровых людей обладают особенной ценностью, в силу сложности их сбора. Поэтому, необходим централизованный сбор данных электромиографических исследований поперечно-полосатых мышц. Центр обработки таких данных позволит даже лабораториям с ограниченным ресурсом проводить качественную диагностику, в то же время, пополняя глобальное хранилище результатов исследований.

Опишем возможную архитектуру сервиса для постановки диагноза. Результат диагностики - весьма важные данные, как для лечащего врача, так и для пациента. Их искажение недопустимо. Соответственно, облачные вычислители являются сомнительным, с точки зрения безопасности, вариантом архитектуры, поскольку, узлы таких вычислителей, как правило, доступны посторонним пользователям. GRID-системы не могут обеспечить гарантированного времени на решение задачи, что также не вполне удобно: пациент и лечащий врач, как правило, не могут ждать неопределенное время. Соответственно, классический многопроцессорный вычислитель является наиболее привлекательным вариантом архитектуры. Вычислительную мощность этой архитектуры можно подбирать и увеличивать на основе прогноза, составленного, скажем, специализированной нейронной сетью, анализирующей статистику использования.

Литература

1. В.Ю. Попов. Сети Хемминга. Научная сессия МИФИ-2008. Т.10 Интеллектуальные системы и технологии. С.152.
2. Д.О. Конончук, Ю.С. Окуловский. Универсальный пакет поддержки интеллектуальных вычислений GANS. VI Всероссийская межвузовская конференция молодых ученых. Труды конференции. Выпуск 4. Математическое моделирование и программное обеспечение. Санкт-Петербург: 2009. С. 151-157.
3. Numeric performance in C, C# and Java. Peter Sestoft. 2009. www.itu.dk/~sestoft/papers/numericperformance.pdf.