

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ (государственный университет)  
ФАКУЛЬТЕТ УПРАВЛЕНИЯ И ПРИКЛАДНОЙ МАТЕМАТИКИ  
КАФЕДРА «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ»  
ПРИ ВЫЧИСЛИТЕЛЬНОМ ЦЕНТРЕ ИМ. А. А. ДОРОДНИЦЫНА РАН

Черных Владимир Юрьевич

**Оптимизация сложности моделей глубокого  
обучения в условиях нехватки данных**

010900 — Прикладные математика и физика

БАКАЛАВРСКАЯ ДИССЕРТАЦИЯ

**Научный руководитель:**  
к.ф.-м.н  
Рябенко Евгений Алексеевич

Москва

2016 г.

# Содержание

<b>1</b>	<b>Введение</b>	<b>4</b>
<b>2</b>	<b>Постановка задачи и используемые обозначения</b>	<b>5</b>
2.1	Обучение модели . . . . .	5
2.2	Выбор модели . . . . .	6
2.2.1	SRM . . . . .	7
2.2.2	Кросс-валидация . . . . .	7
<b>3</b>	<b>Сверточные нейронные сети</b>	<b>8</b>
<b>4</b>	<b>Методы анализа ЭЭГ</b>	<b>12</b>
4.1	Традиционные техники . . . . .	12
4.2	Нейросетевой подход . . . . .	13
<b>5</b>	<b>Описание данных</b>	<b>15</b>
<b>6</b>	<b>Эксперимент</b>	<b>18</b>
6.1	Обзор оригинальных решений . . . . .	19
6.2	Выбор оптимальной модели . . . . .	20
6.3	Перенос архитектуры модели . . . . .	24
<b>7</b>	<b>Заключение</b>	<b>27</b>

### **Аннотация**

В данной работе рассматривается задача оптимизации архитектуры и сложности моделей глубокого обучения по данным. Описывается исходная структура сверточной нейронные сети и методы ее модификации в зависимости от количества данных и их особенностей. Предложенные методы и алгоритмы применяются к задачам построения интерфейса мозг–компьютер (BCI), а именно рассматривается распознавание и классификация сигналов электроэнцефалографии (ЭЭГ). Приводятся данные вычислительного эксперимента, которые показывают непереносимость в общем случае выбранной архитектуры между данными.

# 1 Введение

Глубокие нейронные сети в последнее время хорошо зарекомендовали себя в решении задач распознавания и классификации во множестве практических приложений, включая распознавание изображений, видео, речи и текста [1, 2, 3, 4]. Сверточные нейронные сети лежат в основе лучших структур работы с изображениями и видео в силу их возможности извлекать и строить представления данных, которые инвариантны к локальным трансляциям и деформациям в исходном сигнале.

Несмотря на множество успешных приложений глубоких нейронных сетей в описанных областях, они пока остаются относительно незамеченными в сфере задач построения интерфейса мозг–компьютер (BCI) [5], где до сих пор используются классические подходы, описанные в разделе 4.1. Суть задач BCI состоит в распознавании действий и/или намерений человека по данным его мозговой деятельности. Эта задача в последнее время актуальна в связи с, например, развитием области искусственного протезирования человеческих органов. Если мы научимся распознавать намерение человека поднять левую руку по данным, снятым с его мозга, то сможем делать протезы, которые полностью смогут заменять руку. Подробно различные типы датасетов описаны в разделе 5.

Возможная причина непопулярности моделей глубокого обучения в задачах BCI кроется в том, что число прецедентов в большинстве BCI датасетов ограничено. Преимущества же глубоких сетей над обычными моделями машинного обучения проявляются, только когда объемы данных становятся очень большими [6]. Несмотря на это, в последнее время стали появляться работы, в которых глубокие сети используются для обучения представлений [7] в задачах классификации данных электроэнцефалографии. В этом исследовании авторы на основе спектрального преобразования генерируют серию изображений из временного ряда ЭЭГ и используют сверточную рекуррентную нейронную сеть для распознавания полученных изображений. Однако, в описанной работе отсутствует выбор оптимальной структуры сети и основной упор сделан на хорошую генерацию представления исходных данных. В одной из статей [8] авторы подбирают оптимальную архитектуру нейронной сети, пытаясь тем самым максимизировать совместное правдоподобие данных и модели. Но они используют обычную полносвязную нейронную сеть и генетический алгоритм, что имеет свои недостатки, которые описаны в разделе 6.1. Однако ни одна из этих работ не исследует вопрос выбора оптимальной структуры глубокой сверточной нейронной

сети и ее переносимости на другие данные.

В настоящей работе предлагается использовать глубокую сверточную нейронную сеть как единый фреймворк, объединяющий все стадии традиционного много-ступенчатого анализа ЭЭГ. Исследуется вопрос поиска оптимальной архитектуры такой сети в зависимости от датасета. Также предлагается способ переноса структуры модели с данных большого объема на данные меньшего объема. Описанные методы позволяют уйти не только от ручного анализа данных и генерации признаков, но и от экспертного выбора модели, что является одной из самых актуальных проблем машинного обучения.

## 2 Постановка задачи и используемые обозначения

Задачу оптимизации сложности моделей глубокого обучения условно можно разбить на две подзадачи. Первая — это правильно выбрать структуру нейронной сети. Вторая — при выбранной и зафиксированной структуре обучить модель. Здесь будут описаны оба этапа.

В качестве входных данных есть конечная выборка  $\mathcal{D} = \{(X_i, \mathbf{y}_i)\}_{i=1}^N$ , где  $X_i \in \mathcal{X} = \mathbb{R}^n \times \mathbb{R}^c$  — пространство  $c$ -канальных временных рядов длины  $n$ ,  $\mathbf{y}_i \in \mathcal{Y}$  — пространство меток классов. Таким образом, признаковое описание одного объекта есть матрица размера  $n \times c$ . Классы представляются с помощью унитарной (one-hot) кодировки — рассматривается  $k$ -мерное (где  $k$  - число классов) пространство  $\mathcal{Y} = \{\mathbf{y} \in [0; 1]^k\}$ . Каждая компонента  $y_i$  вектора  $\mathbf{y}$  дает вероятность принадлежности объекта к  $i$ -ому классу. На выборке  $\mathcal{D}$  метки известны достоверно, поэтому в ней все векторы ответов будут иметь бинарный вид  $\{0; 1\}^k$ . Заметим, что мы не делаем ограничение на количество классов, которым принадлежит объект, т.е. в зависимости от специфики выборки могут быть рассмотрены как случаи мультиклассовой (multi-class) классификации, так и классификации с многими метками (multi-label). Произведем разбиение выборки на обучающую и валидационную  $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_v$  размерами  $N_l$  и  $N_v$  соответственно.

### 2.1 Обучение модели

Введем априорно множество классификаторов  $\mathcal{F} = \{f : \mathcal{X} \mapsto \mathcal{Y}\}$ , с которыми мы работаем. Отметим, что  $\mathcal{F}$  можно отождествить с пространством  $\mathcal{W}$  параметров

модели. Для нейронных сетей при зафиксированной структуре модели  $\mathcal{W}$  будет пространством весов сети. Далее обозначения  $f$  и  $\mathbf{w}$  будут взаимозаменяемыми. Предположим, что есть идеальная целевая зависимость  $f^* : \mathcal{X} \mapsto \mathcal{Y}$ . Требуется выбрать классификатор  $f_{\mathcal{F}}^* \in \mathcal{F}$ , наилучшим образом приближающий  $f^*$  [9].

Для формализации понятия наилучшего приближения введем функцию ошибки  $\mathcal{L}_e(\mathbf{y}, \hat{\mathbf{y}})$ . Величина, которую мы всегда хотим минимизировать — точность классификации (ассигасу). Но реально это сделать не получается, т.к. функция потерь не является выпуклой. Поэтому на практике используется ее приближение, которое далее будет обозначено как  $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ . Задача записывается в виде:

$$f_{\mathcal{F}}^* = \arg \min_{f \in \mathcal{F}} \mathbb{E}_{(X, \mathbf{y})} [\mathcal{L}_e(\mathbf{y}, f(X))] = \arg \min_{f \in \mathcal{F}} L(f).$$

Этот подход называется минимизацией риска. Но так как реальное распределение данных в пространстве  $\mathcal{X} \times \mathcal{Y}$  нам неизвестно, то будем минимизировать эмпирический риск:

$$\hat{f}_e = \arg \min_{f \in \mathcal{F}} \frac{1}{N_l} \sum_{\mathfrak{D}_l} \mathcal{L}_e(\mathbf{y}_i, f(X_i)) = \arg \min_{f \in \mathcal{F}} L_n(f).$$

Теперь сделаем финальное уточнение. Так как оптимизировать точность классификации напрямую мы не можем в силу отсутствия выпуклости, то будем оптимизировать ее оценку сверху  $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ , выбор которой зависит от конкретных условий. Итоговая формулировка задачи, которая будет решаться при выбранной структуре модели:

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{N_l} \sum_{\mathfrak{D}_l} \mathcal{L}(\mathbf{y}_i, f(X_i)) = \arg \min_{\mathbf{w} \in \mathcal{W}} Q(\mathbf{w}, \mathfrak{D}_l, \mathcal{L}).$$

## 2.2 Выбор модели

Решая сформулированную выше в разделе 2.1 задачу, мы находим с некоторой погрешностью лучший классификатор  $f_{\mathcal{F}}^*$  из класса  $\mathcal{F}$ . Погрешность дает потерю в качестве  $L(\hat{f}) - L(f_{\mathcal{F}}^*)$ . Это ошибка оценивания. Вспомним, что  $\mathcal{F}$  был задан априорно. Будем выбирать еще и оптимальный класс. Иными словами, теперь структура нейронной сети не фиксирована, т.е. можно варьировать различные гиперпараметры модели, как то: количество слоев и их тип, число нейронов в каждом слое, активационные функции и многие другие. Рассмотрим пространство гиперпараметров  $\mathcal{H}$ . Каждая точка  $\mathbf{h} \in \mathcal{H}$  определяет структуру модели. Зададим параметрическое множество семейств классификаторов  $\{\mathcal{F}_{\mathbf{h}}\}_{\mathbf{h} \in \mathcal{H}}$ . Для выбора надо уметь сравнивать два

семейства  $\mathcal{F}_i$  и  $\mathcal{F}_j$ . Логично выбирать то семейство, у которого функция наилучшего приближения имеет меньший риск, т.е.  $\mathcal{F}_i$  лучше  $\mathcal{F}_j$  если  $L(\hat{f}_{\mathcal{F}_i}) < L(\hat{f}_{\mathcal{F}_j})$ .

Опишем два подхода к решению проблемы неизвестности распределения в пространстве  $\mathcal{X} \times \mathcal{Y}$ .

### 2.2.1 SRM

Первый называется структурная минимизация эмпирического риска (SRM) [10, 11, 12]. В силу невозможности сравнить чистые риски, будем сравнивать их эмпирические аналоги, из-за чего вводится ненулевой порог сравнения:  $\mathcal{F}_i$  предпочтительнее  $\mathcal{F}_j$  при  $Q(\hat{f}_{\mathcal{F}_i}) - Q(\hat{f}_{\mathcal{F}_j}) < \alpha(\mathbf{i}, \mathbf{j}, N)$ . Вводя порог как разность штрафов каждой модели за сложность  $\alpha(\mathbf{i}, \mathbf{j}, N) = p(\mathbf{j}, N) - p(\mathbf{i}, N)$ , получаем следующую формулировку задачи выбора модели с регуляризацией:

$$\hat{\mathbf{h}} = \arg \min_{\mathbf{h} \in \mathcal{H}} \left( Q(\hat{f}_{\mathcal{F}_h}) + p(\mathbf{h}, N) \right).$$

Далее, решение и получаемая ошибка аппроксимации  $L(f_{\hat{\mathcal{F}_h}^*}) - L(f^*)$  зависят от выбора штрафной функции  $p$ , но обычно итоговое минимизируемое выражение имеет единственный минимум.

### 2.2.2 Кросс-валидация

Другим методом выбора модели, чаще применяемом на практике, является кросс-валидация [13]. Отличие от SRM состоит в том, что от сравнений рисков  $L(\hat{f}_{\mathcal{F}_i})$  и  $L(\hat{f}_{\mathcal{F}_j})$  мы переходим не к сравнению эмпирических рисков, но к сравнению других оценок риска. При этом порог сравнения, т.е. функцию  $\alpha$ , мы полагаем равной нулю — поэтому не появляется штрафной член. Ниже описан сам метод оценивания риска. Обучающая выборка дробится на  $q$  непересекающихся блоков одинаковой длины  $\mathcal{D}_l = \bigcup_{i=1}^q \mathcal{D}_l^i$ . Каждый блок по очереди становится контрольной подвыборкой, при этом обучение производится по остальным  $q - 1$  блокам. Тогда итоговое семейство классификаторов выбирается согласно следующему правилу:

$$\hat{\mathbf{h}} = \arg \min_{\mathbf{h} \in \mathcal{H}} \frac{1}{q} \sum_{i=1}^q Q(f_{\text{CV}}, \mathcal{D}_l^i, \mathcal{L}),$$

$$f_{\text{CV}} = \arg \min_{f \in \mathcal{F}_h} Q(f, \mathcal{D}_l \setminus \mathcal{D}_l^i, \mathcal{L}).$$

Это наиболее часто используемый на практике метод, поскольку он напрямую оценивает обобщающую способность и довольно прост в реализации. SRM сложен тем, что требует вложенности классов по сложности, вычисляемой через VC-размерность, друг в друга [10]. VC-размерность, в свою очередь, может быть просто вычислена только для очень простых классов функций  $\mathcal{F}$ , например линейных. В случае же нейронных сетей на настоящий момент существуют только очень неточные верхние оценки [14, 15]. Поэтому в дальнейшем в работе будет применяться кросс-валидация.

### 3 Сверточные нейронные сети

Нейронные сети являются универсальным методом машинного обучения и хорошо зарекомендовали себя во множестве задач. Рассмотрим один из типов моделей, входящий в этот широкий класс, а именно сверточные нейросети. В отличие от обычных нейронных сетей [16], в сверточных сетях [17, 18] различают несколько типов слоев в зависимости от их свойств и назначения. Приведем краткое описание каждого них.

- Сверточные

Название слоя произошло от названия математической операции свертки. Пусть есть сигнал  $x(t)$  и функция взвешивания  $w(t)$ , которую еще называют ядро или фильтр. Тогда операция свертки запишется как  $s(t) = (x * w)(t) = \int x(a)w(t - a)da$ . Переходя к терминам сверточных нейронных сетей,  $x$  — вход слоя, обозначаемый далее  $X$ ,  $w$  — фильтр, обозначаемый за  $K$ . Рассматривая дискретный случай и ядро с ограниченным носителем, получим  $S(i) = \sum_m X(i)K(i - m)$ . В большинстве библиотек машинного обучения реализуется чуть другая операция — кросс-корреляция:

$$S(i) = \sum_m X(i + m)K(m),$$

или в двумерном случае:

$$S(i, j) = \sum_m \sum_n X(i + m, j + n)K(m, n).$$

Таким образом, основное отличие сверточных слоев от полносвязных состоит в том, что каждый нейрон в нем соединен только с ограниченным числом нейронов из предыдущего слоя, а не со всеми. Обычно такой слой рассматривают



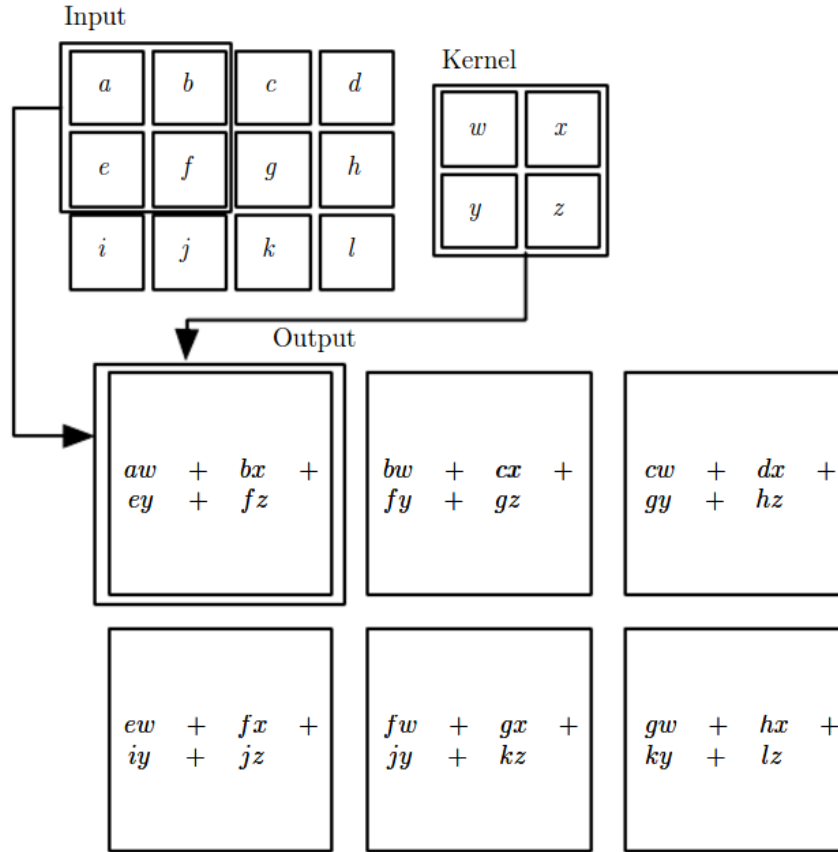


Рис. 1: Свертка одним фильтром

как совокупность фильтров. Каждому фильтру ставят в соответствие несколько параметров (рис. 1):

1. Размер (filter size)
2. Шаг (stride)
3. Набор весов (weights and bias)

Каждый фильтр в сети реализуется набором нейронов, каждый из которых подсоединен только к своей области видимости — это свойство носит название локальности зоны восприимчивости нейрона (local receptive field). Еще одной важной парадигмой в этой теории является разделение параметров (parameters sharing) — идея о том, что нейроны, реализующие один фильтр, имеют одинаковые веса. Объединяя два этих концепта в одно, можно сказать, что сверточный слой обладает свойством разреженности связей (sparse connectivity). В силу того, что размер фильтра обычно на порядки меньше размера входа, требуется гораздо меньше параметров и машинного времени для того, чтобы вычислить выход каждого нейрона.

Главным назначением одного конкретного сверточного слоя является выделение простых паттернов во входе с помощью обучаемых фильтров. Располагая сверточные слои один за другим и в комбинации с другими типами слоев, получаем, что с ростом глубины сети растет абстрактность и сложность выделяемых признаков. Так если фильтры настроены на выделение прямых линий на картинке, то после первого слоя будут распознаны только прямые линии, после второго — их комбинаций, то есть, например, выпуклые фигуры и так далее.

- Активационные

Выход сверточного слоя есть линейной преобразование над его входом. Композиция линейных преобразований дает линейное преобразование. У нас же есть желание аппроксимировать функции высокой сложности с помощью нейронной сети. Поэтому необходимо добавлять нелинейности. Именно это и делает активационный слой. Обычно в качестве нелинейных функций в промежуточных активационных слоях используют ReLU [19] и ее параметрические аналоги такие как LeakyReLU или PReLU [20]. Использование традиционных функций активации, таких как сигмоида или гиперболический тангенс, несет за собой проблемы размывающихся (vanishing) или взрывающихся (exploding) градиентов на этапе обучения, а также некоторые другие [19].

- Пулинг

В самом общем виде пулинг — замена элементов входа на некоторую статистику его близлежащих элементов. Главная функция таких слоев — снижение размера представления данных после очередного слоя, чтобы уменьшить число параметров, количество вычислений в сети и избежать переобучения. Наиболее часто используемый вид статистики — максимум (рис. 2), когда оператор возвращает максимум из области, поданной на вход. Также реже используются другие функции — среднее по области; L2-норма; взвешенное среднее с весами, пропорциональными расстоянию до центральной точки и другие. Выбор конкретной функции зависит от специфики задачи.

Наряду со снижением размера данных, пулинг-слои выполняет еще одну очень важную функцию. Они делают представление данных приближенно инвариантным относительно малых трансляций признака во входных данных. Это легко объяснить — при маленьком сдвиге во входных данных значения на выходе

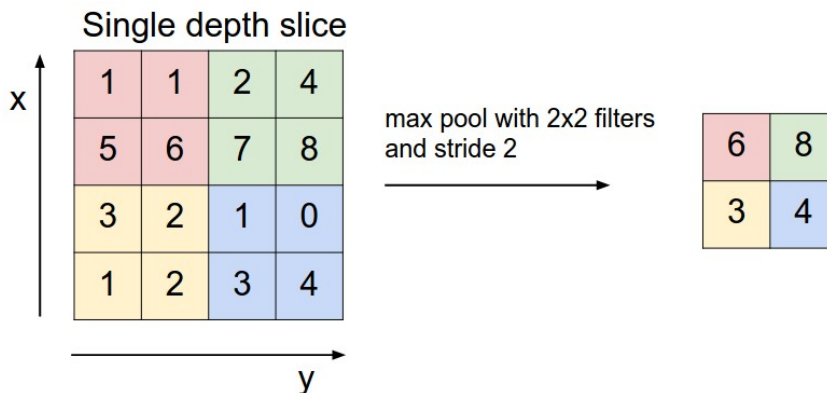


Рис. 2: max-пулинг на квадратной области

пулинг-слоя почти не будут меняться, т.к. происходит агрегация с помощью статистики по определенной окрестности. Это свойство может быть полезно, если мы больше интересуемся наличием признака во входе, нежели точным его положением. Чтобы сделать итоговое представление инвариантным не только к трансляциям, но и к другим преобразованиям, например вращениям, можно агрегировать в пулинг-слое выходы из сверточных слоев параметризованных по-разному, т.е. имеющих различный формы и веса фильтров.

- Полносвязные

Все предыдущие слои, скомбинированные в некоторую структуру, по сути своей являются генератором нового представления входных данных. По полученным признакам, используя обыкновенную нейронную сеть из нескольких полносвязных слоев, производится классификация.

- Дополнительные

Архитектуры почти всех сверточных нейронных сетей содержат в себе слои, описанные выше. Но глубокие нейронные сети — это модели с очень большим числом параметров, как правило порядка  $10^6$  и более. Поэтому в целях уменьшения переобучения используются некоторые дополнительные слои, например Dropout, DropConnection, нормализационные, шумовые и множество других [21]

Отметим также, что наиболее часто используемая архитектура сверточных нейронных сетей выглядит следующим образом: стекинг нескольких сверточных слоев друг за другом, затем, как правило, нелинейная функция активации и пулинг-слой.

Такая последовательность может повторяться несколько раз. Более подробное описание современных архитектур можно найти в [1, 22, 23]

## 4 Методы анализа ЭЭГ

### 4.1 Традиционные техники

Классический подход к решению задачи анализа данных ЭЭГ является собой многоступенчатую систему [5, 24], в которой можно выделить следующие основные этапы:

1. Частотная фильтрация.

В сигнале ЭЭГ существует множество помех, которые условно можно разделить на две категории — физиологические и технические. Частотная фильтрация, например полосовая, FIR, IIR и другие, может быть использована [25] для того, чтобы исключить из сигнала разного рода технические помехи, как то: наводки от линии электропитания, моментные изменения сопротивления электродов в датчиках и т.д.

2. Удаление артефактов.

Этот этап связан с фильтрацией данных от физиологических артефактов, то есть сигналов, не относящихся непосредственно к деятельности мозга — мышечная активность, движение и моргание глазами и т.д. Эти помехи могут быть исключены несколькими путями, например ручным выбором признаков и отбросом ненужных каналов на основе визуального анализа; использованием метода независимых компонент ICA [26] с разложением исходного сигнала в виде  $X = SA$ , где  $S$  — независимые компоненты, нахождением артефактных компонент в  $S$ , обнулением соответствующих коэффициентов в  $A \rightarrow \tilde{A}$  и последующем восстановлении сигнала  $\tilde{X} = S\tilde{A} = XA^{-1}\tilde{A}$  или же любым другим методом.

3. Уменьшение размерности.

Показания ЭЭГ с разных, но близких каналов сильно коррелированы. Чтобы избавиться от мультиколлинеарности можно воспользоваться одним из многих методов понижения размерности, например методом главных компонент PCA

или методом общих пространственных структур CSP [27], который обычно используют в задачах ЭЭГ.

#### 4. Устранение зависимости от субъекта.

Показания разных людей и даже показания одного и того же человека в разные сессии имеют большую вариабельность, следовательно, необходимо искать стационарные подпространства и признаки, которые будут устойчивы относительно этой вариабельности. Одним из возможных способов является усовершенствованный алгоритм CSP, использующий дивергенцию Кульбака-Лейблера [28].

#### 5. Разбиение сигнала на отрезки и последующая классификация.

После предобработки сигнала производится классификация по разбитым отрезкам. Можно использовать любой алгоритм, но чаще используются линейные модели — SVM, логистическая регрессия, или простые нелинейные алгоритмы — kNN, полносвязные нейронные сети [5].

Таким образом, традиционный метод анализа требует человеческого участия и состоит из нескольких шагов, каждый из которых использует свои алгоритмы и, как следствие, свои критерии качества, что также является недостатком.

## 4.2 Нейросетевой подход

Стоит отметить, что первые три шага классического анализа есть линейные преобразования исходного сигнала, поэтому можно попробовать выполнять их с помощью одного и того же фреймворка — сверточной нейронной сети. Сверточные слои, как обсуждалось ранее в разделе 3, в силу локальной зоны видимости нейронов и линейной функции активации, по сути своей являются инструментом автоматического подбора фильтров из классического анализа сигналов. Пулинг-слои, в свою очередь, обеспечивают трансляционную инвариантность отыскиваемых паттернов. Таким образом, учитывая тот факт, что нейронные сети хорошо себя зарекомендовали в решении задач анализа сигналов и изображений [1], логичным и перспективным выглядит их применение к анализу ЭЭГ.

Одна из главных проблем в области глубоких нейронных сетей — правильный подбор архитектуры сети, т.е. класса  $\mathcal{F}$ . Сейчас это делается либо на основе нефор-

мализуемых экспертных знаний в области, либо по опыту предыдущих задач, либо методом проб и ошибок, т.е. по сути наугад.

В данной работе предлагается использовать кросс-валидацию (см. раздел 2.2.2) для выбора локально оптимальной модели по данным. Пространство структур моделей  $\mathcal{H}$  зададим с помощью описанных ниже параметров сверточной нейросети.

- Количество нейронов в полносвязных классификационных слоях

Самое большое число параметров сети сосредоточено именно в полносвязных слоях по понятным причинам. Именно поэтому переобучение наиболее вероятно в этих слоях и поэтому там имеет смысл варьировать число нейронов.

- Вероятность Dropout в слоях классификации

Dropout позволяет избежать переобучения путем выкидывания каждого нейрона в слое с вероятностью  $p$  на этапе обучения [29]. Это не дает весам сильно уходить в разброс, что было бы явным признаком переобученности.

- Длина временного ряда  $n$  в каждом из объектов

- Параметры фильтров в первом сверточном слое в зависимости от длины временного ряда  $n$

Это необходимо для того, чтобы размер представления данных после первого слоя не сильно изменялся от длины временного ряда  $n$ .

Также в работе рассматривается проблема переноса архитектуры сети с одних больших данных ЭЭГ на другие маленькие. Предлагается использовать следующую стратегию:

- Берем хорошую, но сложную модель на больших данных
- Постепенно уменьшаем объем больших данных, пока он примерно не сравняется с объемом маленькой выборки
- На каждом шаге выбираем лучшую архитектуру из пространства  $\mathcal{H}$

В идеальном случае должен наблюдаться следующий вариант — начальная модель слишком сложна, переобучается и дает плохое качество; наивная модель слишком проста, чтобы выделить все закономерности из данных; максимум качества находится где-то посередине.

В данной работе проверяется несколько гипотез о переносимости архитектуры сети.

**Гипотеза 4.1.** *Модель без изменений можно перенести на другие данные без значимой потери в качестве.*

Забегая вперед, отметим, что эта гипотеза не подтверждается, что будет показано в разделе 6, и поэтому рассматриваются еще две гипотезы о природе плохой переносимости моделей.

**Гипотеза 4.2.** *Модель плохо переносима из-за различия в объеме данных.*

**Гипотеза 4.3.** *Плохая переносимость обусловлена излишней сложностью модели.*

Проверка этих гипотез и вычислительный эксперимент приведены в разделе 6, но прежде опишем данные, которые используются в работе.

## 5 Описание данных

В работе исследуется 4 различных набора данных. Краткое описание каждого датасета приведено в таблице 1.

Структура		Непрерывная		Попытками	
Название		Kaggle EEG	BBCI IV-I	Ewan Nurse	BBCI II-IV
Год		2014	2007	2015	2002
Размер	Попыток	~ 250	~ 400	~ 1350	316
	Точек	~ $1.5 \cdot 10^6$	~ $2 \cdot 10^6$	~ $1350 \cdot 100$	$316 \cdot 50$
Частота		500	1000	250	100
Каналов		32	59	10 – 62	28
Классов		6	3	3	2

Таблица 1: Краткое описание данных

В настоящее время различают две основных структуры ЭЭГ-датасетов — непрерывная и попытками (trials). В непрерывном случае снятие показаний с датчиков ведется потоково, в то время как субъект выполняет заданный набор команд. Этот вариант наиболее приближен к реальным задачам BCI, в которых в большинстве

своем требуется онлайн распознавание действий или намерений субъекта по данным ЭЭГ. Подход с попытками подразумевает изначальную экспертную (или любую другую, но уже заданную) нарезку временного ряда на отрезки, внутри которых мозг находится в определенном ментальном состоянии, например человек выполняет конкретное моторное действие. Такие отрезки имеют, как правило, равную длину. Их наличие упрощает решение задачи, но отдаляет на один шаг от реальных задач, где такое разбиение не дается и надо встраивать его в сам алгоритм.

Проблема нарезки непрерывных данных решается в этой работе следующим образом — используется скользящее окно определенного размера  $n$ . Для каждой точки во времени обучающим объектом является значение  $n$  точек ряда перед текущей точкой по всем  $s$  каналам, а обучающей меткой — метка действия в этой временной точке.

Отметим еще одну особенность в работе с непрерывными данными. В ходе решения задачи переноса модели требуется уменьшать объем данных. Если в случае датасетами, которые имеют структуру попыток все ясно — просто убираем из выборки попытки таким образом, чтобы баланс классов примерно сохранялся в исходном состоянии, то с непрерывными данными не все так очевидно. Заметим, что в обоих используемых непрерывных датасетах есть ярко выраженная структура — период выполнения действия сменяется таким же или бóльшим по промежутку периодом бездействия. Поэтому по размеченным данным удастся с очень хорошей точностью нарезать данные по попыткам. В классическом анализе временных рядов данные всегда отсекаются с одного из концов ряда — это обусловлено тем, что вырезая данные из середины можно нарушить их структуру и тем самым потерять информацию. Во время уменьшения объема данных будем удалять их с начала ряда, отрезая размеченные до этого попытки. Это связано с тем, что с конца ряда данные берутся для валидации и выкидывая данные с конца обучающей выборки мы бы столкнулись с только что описанной проблемой.

Теперь рассмотрим каждый из наборов данных подробнее.

- Kaggle EEG [30]

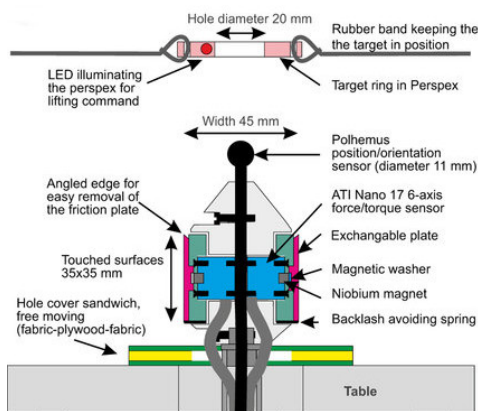
Данные имеют непрерывную структуру и содержат записи ЭЭГ людей, которые выполняли одну и ту же последовательность из 6 действий [31]:

1. Начало движения

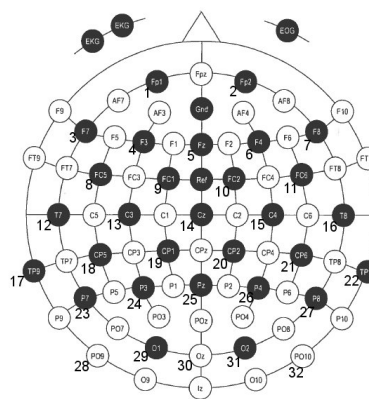


2. Первое касание объекта
3. Захват с помощью большого и указательного пальцев
4. Поднятие и удержание в течение нескольких секунд
5. Помещение объекта в исходное положение
6. Разжатие пальцев

Этот датасет предназначен для определения ощущений, намерений и действий основываясь на данных ЭЭГ. 12 участников выполняли описанную последовательность действий с предметами, различающимися по весам (165, 330 или 660 грамм) и типу поверхности (наждачная бумага, замша или шёлк), причем субъект не был уведомлен о данных изменениях, что заставляло регулировать силу сжатия по ходу попытки. Стимулом к началу попытки служил световой сигнал LED-дисплея. Данные снимались одновременно с помощью 32 датчиков на специально собранной для эксперимента установке (см. рисунки 3a, 3b). Для каждого из 12 субъектов было проведено 10 серий измерений, в каждой из которых было примерно по 30 попыток. В качестве выборки берутся первые 8 серий для каждого субъекта, а 9 и 10 серии оставляются для тестирования.



(a) Схема установки



(b) Расположение электродов

Рис. 3: Описание установки Kaggle EEG

- Berlin VCI IV-I [32]

Выборка также имеет непрерывную структуру, но она содержит данные исследований задачи “воображаемого движения” (motor imagery). Отличие от реальных движений состоит в том, что человек лишь представляет, что он делает

движение определенной конечностью, но по факту его не совершает. Такие задачи актуальны в связи с разработкой методов протезирования конечностей и управления ими с помощью ВСИ [33]. Конкретно в этом эксперименте каждому человеку было предложено выбрать два варианта движений из трех — левая рука, правая рука, любая нога. Таким образом получается три конечных класса — два действия и состояние покоя. Сигналом к началу попытки на обучающей выборке служил визуальный стимул в виде стрелки в нужную сторону. На тестовой же выборке это был звуковой сигнал, что осложняет задачу.

- Ewan Nurse [8]

Этот набор данных, в отличие от двух предыдущих, имеет структуру попыток. Дизайн эксперимента был следующий — человеку давалось по специальному резиновому мячику в каждую руку. От него требовалось сжимать эти мячики произвольно выбранной рукой в произвольные моменты времени, то есть без стимулов — тем самым авторы намеревались уменьшить влияние сигналов, поступающих от глаз или других органов чувств. Нарезка сигнала производилась с помощью показаний дополнительных датчиков электромиографии (EMG), которые фиксируют электрические сигналы непосредственно от сокращения мышц.

- Berlin VCI II-IV [34]

Датасет также имеет структуру попыток и является как самым маленьким, так и самым старым среди представленных. Подопытные садились в обычный стул и клали руки в удобную позицию на клавиатуру. Их задачей было нажимать клавиши указательным пальцем и мизинцем в произвольном порядке и темпе. Эксперимент состоял из 3 сессий по 6 минут каждая с небольшими перерывами между ними (порядка нескольких минут). Данные, как и в случае с датасетом Ewan Nurse, были нарезаны с помощью сигнала EMG.

## 6 Эксперимент

Целью эксперимента был выбор локально оптимальной модели с помощью кросс-валидации и проверка гипотез 4.1, 4.2, 4.3.

## 6.1 Обзор оригинальных решений

Отметим, что к каждому из датасетов прилагается авторское решение либо же решения, победившие в конкурсах. В случае двух датасетов Berlin VCI это были традиционные методы, поэтому в этой работе они не рассматриваются.

Решение [8], продемонстрированное авторами набора данных Ewan Nurse, использует обычный многослойный перцептрон. Для отбора лучшей структуры сети, т.е. нахождения  $\hat{\mathbf{h}}$ , авторы применяют генетический алгоритм. Число слоев сети ограничено 3, а число нейронов в слое — 100. Таким образом, описание одной сети состоит из трех чисел и пространство  $\mathcal{H} = [0; 100]^3$ . Затем случайным образом генерируются 30 описаний сетей и помещаются в нулевое поколение. На каждом шаге генетического алгоритма, чтобы отобрать лучшего особи для создания потомства, необходимо обучить все сети в поколении. Обучение производится с помощью обычного метода обратного распространения ошибки (BackPropagation) с модификацией алгоритмом имитации отжига (Simulated Annealing) [35] — после обучения сети ко всем весам добавляются случайные составляющие и сеть обучается еще, но уже начиная с этого приближения весов. Такой цикл повторяется 10 раз. Это позволяет снизить вероятность застопоривания в локальном минимуме.

В настоящей работе автор попробовал воспроизвести предложенный алгоритм с генетическим способом отбора моделей, а также с помощью алгоритма Tree Parzen Estimator (TPE) [36]. Недостаток генетического метода в том, что приходится обучать очень много сетей, а учитывая, что используется связка BP+SA, это занимает очень много времени. Этот минус устраняется вторым алгоритмом. Но главный недостаток, который принципиально не могут устранить алгоритмы выбора модели — нестабильность обучения самой модели, т.е. очень высокая ошибка оценивания и низкая точность нахождения  $f_{\mathcal{F}}^*$ .

Чтобы продемонстрировать это, проведем следующий эксперимент — будем обучать одну и ту же сеть несколько раз и смотреть на итоговое качество, которое получается. Изначально отсечем случайным сбалансированным по классам образом 20% данных на тест. Алгоритм имитации отжига для своих внутренних шагов требует валидационной выборки, ее отсечем таким же образом в размере 20% от оставшихся данных. Обучая сеть с помощью BP+SA будем получать некоторое качество на тестовой и на валидационной выборке, по последнему из которых и надо отбирать наилучшую модель. Для достоверности будем обучать одну и ту же сеть 20 раз. По-

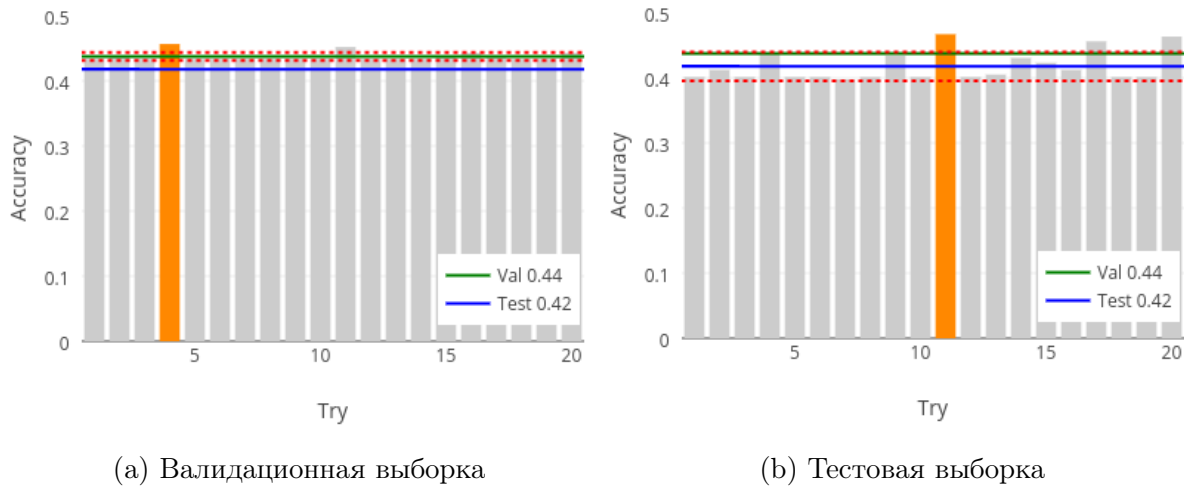


Рис. 4: Качество работы сети Ewan Nurse

лученные результаты для одной из архитектур можно увидеть на рисунках 4a, 4b. Как видно из этих рисунков, различия в валидационном качестве конкретно для этой сети не велики, но в других экспериментах составляли до 5–7%. Более существенно здесь то, что качество на тестовой выборке скачет очень сильно и в итоге, выбирая лучшую модель по валидации, мы очень часто будем ошибаться и иногда эта ошибка будет существенной — в районе 5%. В этом случае любой выбор модели оказывается довольно бессмысленным. Поэтому данную модель также стоит отбросить.

Остается единственная модель, которая изначально и предполагалось исследовать в деталях — с соревнование Kaggle EEG.

## 6.2 Выбор оптимальной модели

Рассмотрим в подробностях модель, архитектура которой в общих чертах похожа на решение участников, занявших 3 место в соревновании Kaggle EEG [37].

Полную структуру сети можно увидеть на рисунке 5. Опишем по порядку, чем отличаются слои и зачем нужен каждый из них.

- На вход сети подается сигнал почти в исходной форме. Он отличается от сигнала с аппаратуры только применением полосового фильтра для отсекающей той части спектра, в которой происходит основная мозговая деятельность. Первый сверточный слой осуществляет понижение размерности в канальной области. То есть, если изначально был  $s$ -канальный сигнал из  $n$  точек по времени, то первая свертка будет производиться именно по каналам и на выходе слоя будет меньшее число каналов, конкретно в этой модели 6. Как уже упоминалось

в разделе 4.1, показания с различных но близких каналов сильно коррелированы, поэтому осуществление понижения размерности разумный шаг, который осуществляется и в традиционной схеме, но здесь реализуется с помощью одного сверточного слоя. Также этот этап дает возможность обучиться фильтрам таким образом, чтобы снизить зашумленность данных.

- После понижения размерности в канальной области, производится свертка по времени с целью также понизить размерность и агрегировать информацию в меньшем числе точек. По сути своей, это даунсемплинг по времени, т.е. понижение частоты временного ряда, но более сложным способом, чем просто выкидывание точек.
- Далее следует стандартный стекинг из сверточных и пулинг слоев для получения нового представления данных. Цель такого участка сети — выделить сложные паттерны из сигнала. Также отметим, что на схеме не изображен промежуточный этап с расширением признаков до 64 и последующего сжатия обратно до 32 путем пулинга по каналному измерению. Это сделано для того, чтобы еще больше усилить акцент именно на наличии какого-то паттерна, а не на его точном положении и, также, для того, чтобы снизить переобучение.

- Последний участок сети — полносвязные классификационные слои. Между ними стоят слои Dropout для снижения вероятности переобучения, ведь именно здесь сосредоточено наибольшее число параметров. Отметим, что в качестве

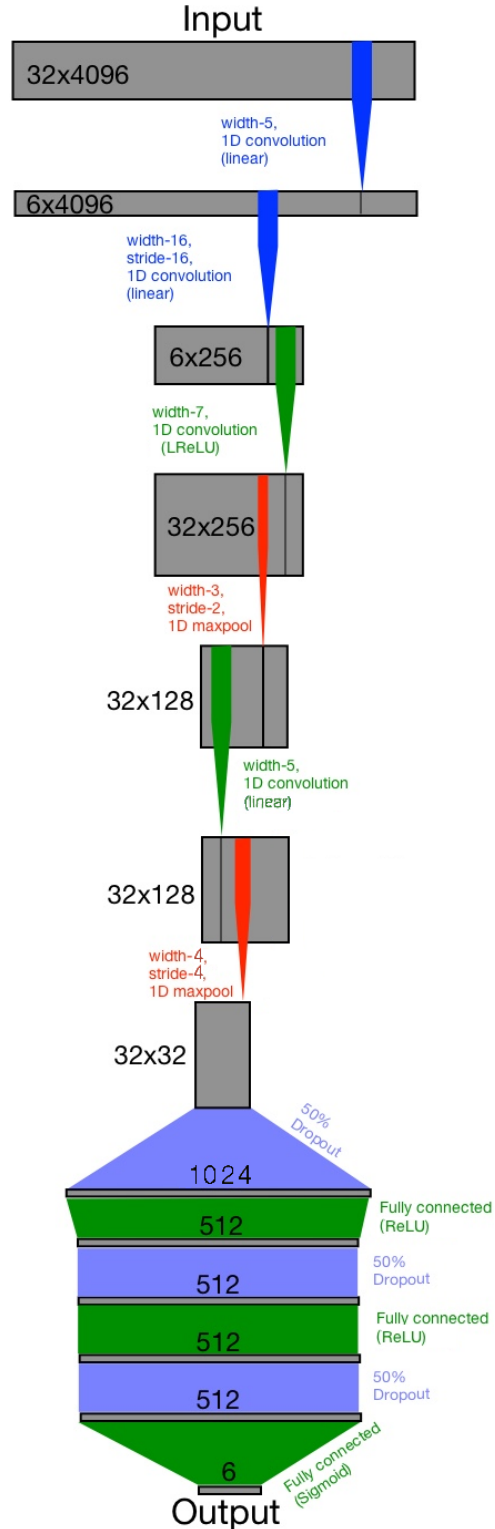
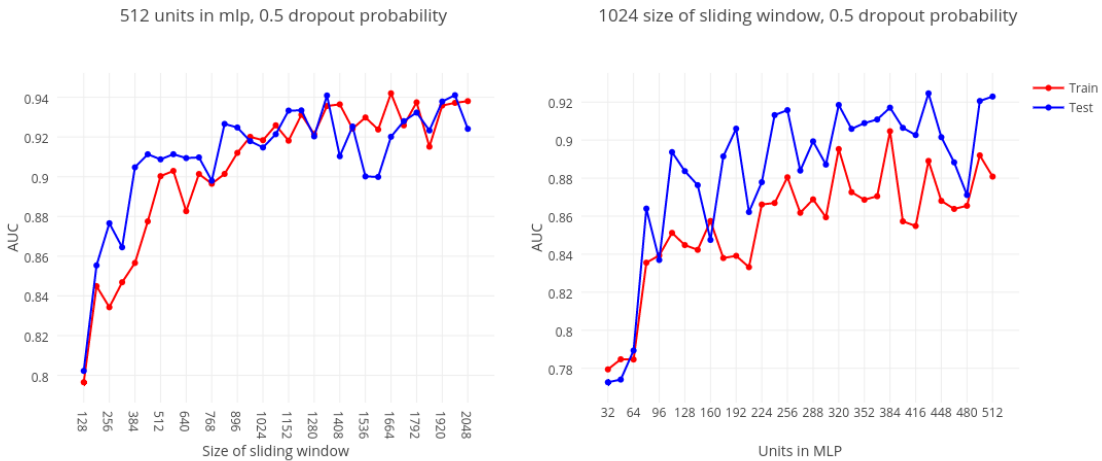


Рис. 5: Архитектура исходной сети

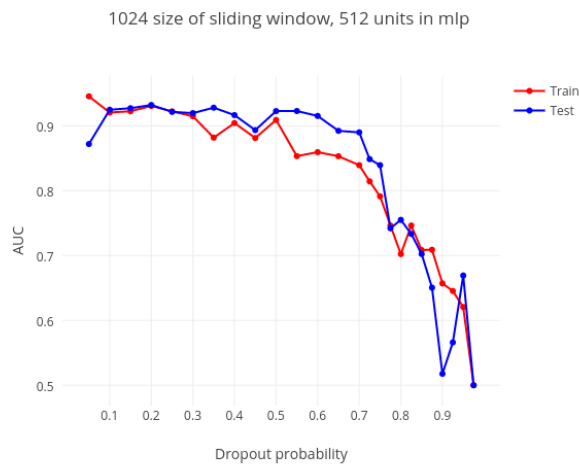
активационной функции выбирается ReLU, т.к. помимо других плюсов, обучение с ней происходит заметно быстрее, ибо она очень легко вычислима [19]. В последнем слое используется сигмоида, т.к. необходима вероятностная интерпретация результата.

Будем искать локально оптимальную архитектуру в окрестности описанной только что модели. Выбирать будем из моделей, которые получаются из начальной изменением описанных в разделе 4.2 параметров. Способ выбора наилучшей архитектуры  $\hat{\mathbf{h}}$  — кросс-валидация (см. раздел 2.2.2), метод построения лучшей модели  $f_{\mathcal{F}_{\mathbf{h}}}^*$  в выбранном классе  $\mathbf{h}$  — метод обратного распространения ошибки. Полученные результаты можно увидеть на рисунках 6, 7



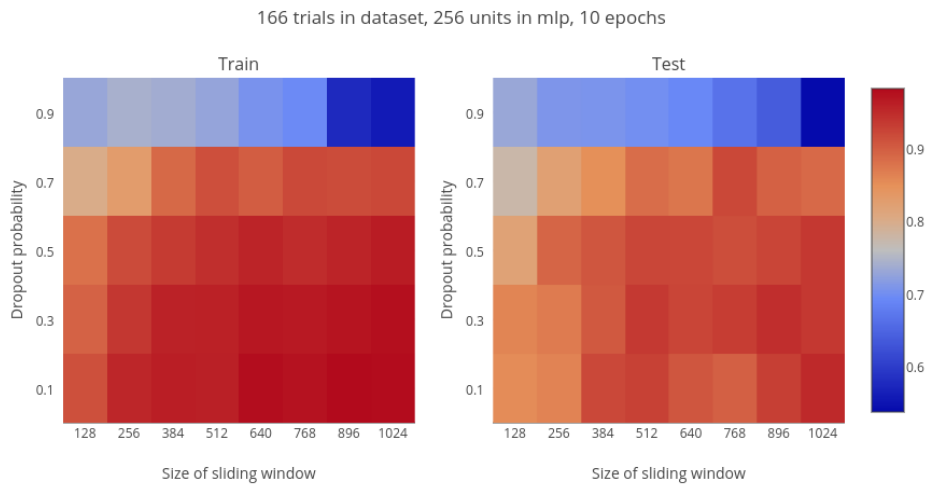
(a) Размер окна

(b) Нейронов в полносвязных слоях

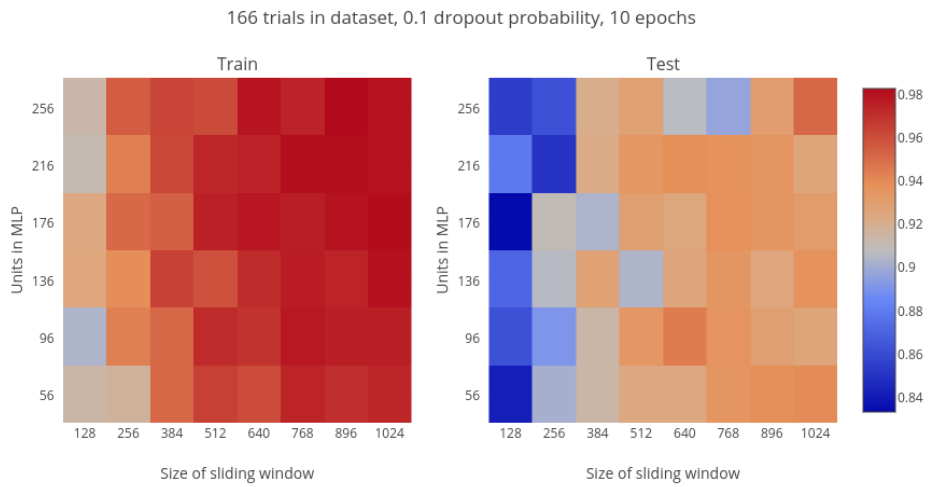


(c) Вероятность Dropout

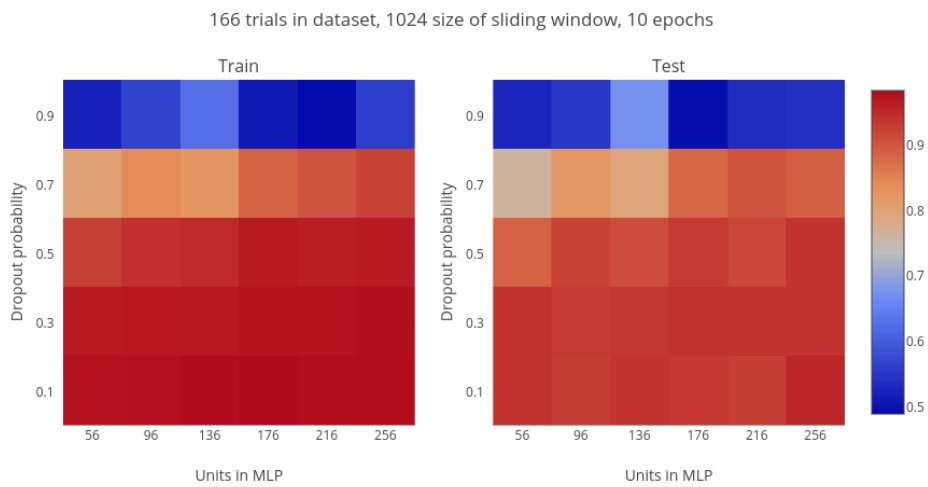
Рис. 6: Зависимость качества AUC от параметров в 1D-представлении



(a) Размер окна/Dropout



(b) Размер окна/Число нейронов



(c) Число нейронов/Dropout

Рис. 7: Зависимость качества AUC от параметров в 2D-представлении

Данные картинки показывают лишь срезы трехмерного распределения качества по параметрам. По результатам анализа всех совместных вариантов параметров была выбрана наилучшая модель со следующими параметрами: размер окна = 2048, число нейронов в полносвязных слоях = 432, вероятность dropout = 0.35.

Но главное в этих графиках — отметить то, что качество сначала примерно постоянно, а затем начинает монотонно падать с упрощением модели. Это значит, что почти без потери качества можно упростить исходную модель. А при одном и том же качестве в приоритете всегда более простая модель.

### 6.3 Перенос архитектуры модели

Теперь экспериментально исследуем возможность переноса архитектуры модели, построенной в разделе 6.2, на другие данные.

В таблице 2 показана краткая сводка качества работы разных моделей на каждом из датасетов. Здесь RMSE — средняя квадратичная ошибка, Acc — точность классификации.

Структура		Непрерывная		Попытками	
Название		Kaggle EEG	BBCI IV-I	Ewan Nurse	BBCI II-IV
AUC	Начальная	0.93	0.69	0.59	0.72
	Простая	0.89	0.66	0.58	0.72
Метрика		AUC	RMSE	Acc	Acc
	Начальная	0.93	5.04	0.45	0.64
	Простая	0.89	4.54	0.44	0.65
	Авторская	<b>0.98</b>	<b>0.42</b>	<b>0.58</b>	<b>0.84</b>

Таблица 2: Сравнение качества моделей

Качество приведено, во-первых, в терминах функционала качества AUC — площади под ROC-кривой. Но AUC в чистом виде применим только к бинарной классификации. В нашем случае каждая компонента выхода по сути своей дает ответ о принадлежности или непринадлежности к конкретному классу, что есть бинарная классификация. Поэтому в случае многих классов берется средний по колонкам AUC. Этот функционал качества выбран в качестве общего для сравнения разных датасетов и моделей, т.к. именно он был функционалом качества в задаче Kaggle EEG, откуда



взята исследуемая модель. Во-вторых, приводится сравнение качества в терминах метрики, которая была основной для каждого датасета.

В гипотезе 4.1 выдвигается предположение о том, что можно перенести архитектуру модели без изменений на другие данные. Как видно из таблицы 2, сделать это не получается.

Следующие две гипотезы 4.2 и 4.3 выдвигают предположения о том, почему сделать этого не получается.

Гипотеза 4.2 гласит, что плохая переносимость обусловлена нехваткой данных. На рисунке 8 показана зависимость качества классификации данных ЭЭГ моделью с начальной архитектурой от количества данных в обучающей выборке. Вертикальными линиями отмечены размеры других датасетов с качеством, которое дает эта структура на них.

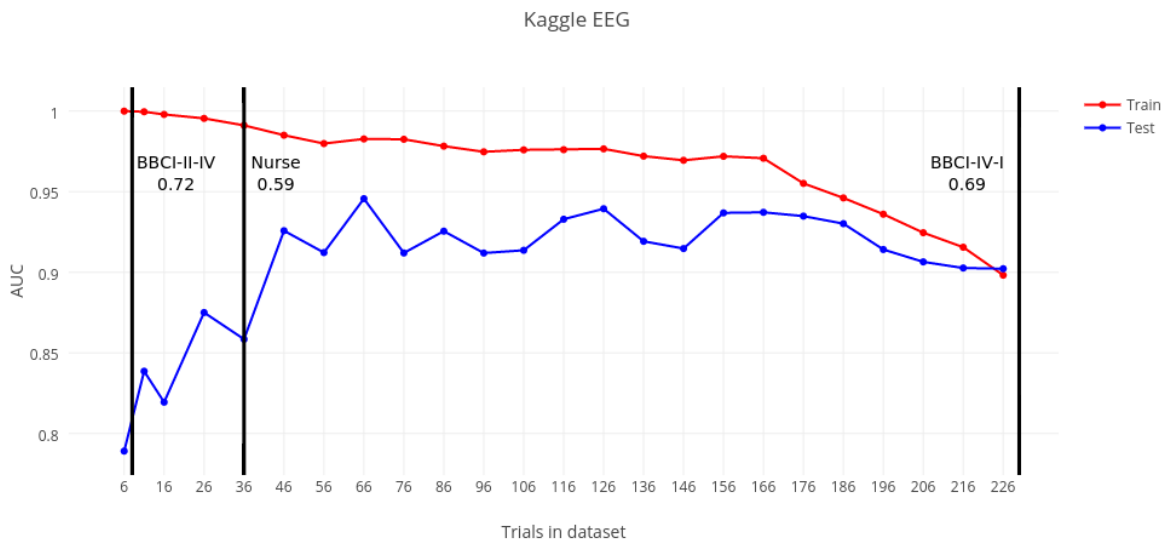


Рис. 8: Зависимость качества от объема обучающей выборки

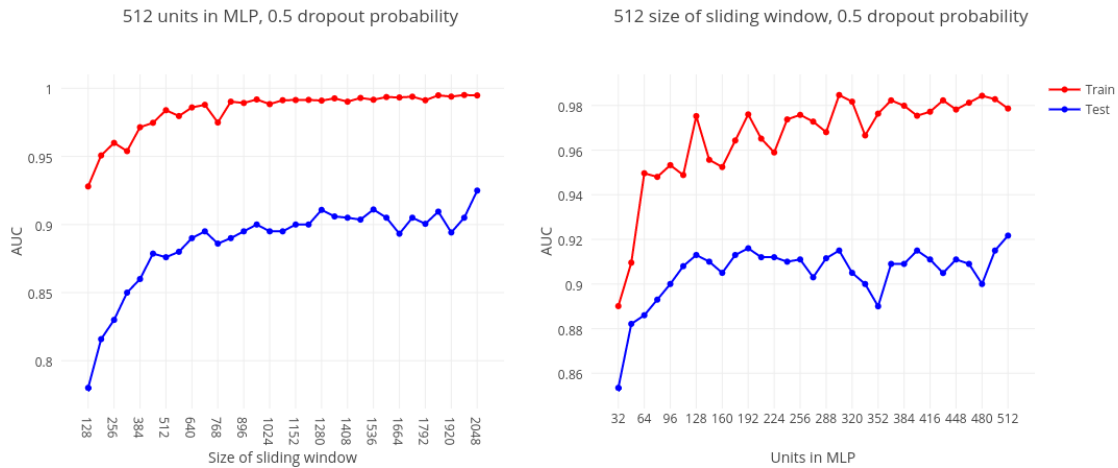
Из графика на рисунке 8 видно, что качество, которое реально получается на других данных и которое должно было бы получиться, если бы модель хорошо переносилась, сильно отличаются. Таким образом, гипотеза 4.2 неверна.

Проверим оставшуюся гипотезу 4.3 о том, что исходная модель излишне сложна. Для каждого датасета возьмем размер обучающей выборки в задаче Kaggle EEG соответствующий объему этого датасета. На выбранном объеме данных посмотрим как изменяется качество в зависимости от архитектуры сети на выборке Kaggle EEG. Как описано в разделе 4.2, в идеале должен наблюдаться максимум качество на

модели средней сложности. Эту архитектуру и стоило бы переносить.

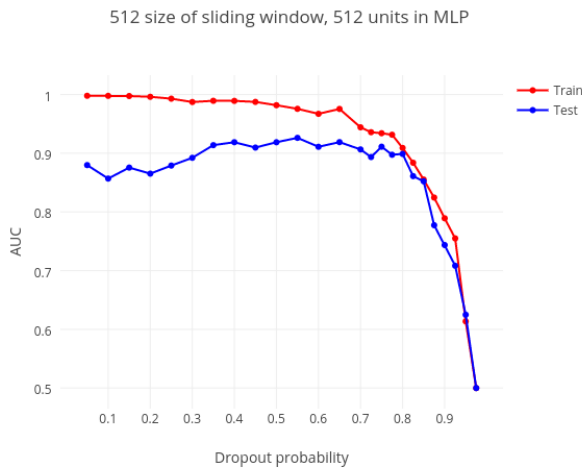
Размер данных Berlin VCI IV-I практически такой же, как и Kaggle EEG, поэтому чтобы увидеть описанные только что графики достаточно посмотреть на рисунок 6. Ничего близкого к качеству показанному реально на BSCI IV-I не наблюдается. Более того, качество монотонно падает с упрощением модели, поэтому по нашему предположению надо переносить самую сложную модель. В таблице 2 приведены результаты работы самой сложной и самой простой сети.

Результаты для объема данных, соответствующих датасету Ewan Nurse, приведены на рисунке 9.



(a) Размер окна

(b) Нейронов в полносвязных слоях



(c) Вероятность Dropout

Рис. 9: Зависимость качества AUC от параметров для уменьшенного объема данных

Точно также наблюдается лишь монотонное падение качества и никакого локального максимума. Лишь в графике зависимости AUC от вероятности Dropout можно увидеть очень слабый максимум в районе 0.5. Это в целом может подтверждать правильность идеи метода переноса, но неприменимость их к конкретным данным.

К датасету VBCI II-IV применим результаты с тем же размером данных, что и на Ewan Nurse, так как данных в VBCI II-IV слишком мало.

Из полученных результатов можно заключить, что гипотеза об излишней сложности модели также не подтверждается.

Главный вывод, который можно сделать по итогам проведенных экспериментов — не существует универсальной модели, которая могла бы решать разные задачи VCI лишь с незначительными модификациями структуры.

## 7 Заключение

Задача анализа электроэнцефалографии особенно актуальна в последнее время в связи с развивающейся областью построения интерфейсов мозг-компьютер. В работе предпринимается попытка найти универсальную методологию для классификации намерений и/или движений человека по данным многоканальных временных рядов ЭЭГ. Отличие от других работ в этой области заключается в использовании глубоких сверточных нейронных сетей и поиске локально оптимальной архитектуры сети в окрестности исходной на основе конкретных данных. Также предлагается метод переноса архитектуры сети с наборов данных большого объема на маленькие датасеты. Вместо использования сложных преобразований исходного сигнала, как это принято делать в традиционном анализе ЭЭГ с помощью фильтрации и понижения размерности, используются первые несколько сверточных слоев нейронной сети, которые обучаются на выделение наиболее информативного представления данных. Это позволяет избежать этапа предобработки данных и позволяет использовать "сырой" сигнал, что является плюсом.

В ходе работы было показано, что данные ЭЭГ, хоть и кажутся на первый взгляд схожими, на деле являются достаточно разными. Различия в используемом оборудовании, методах снятия сигнала и типе выполняемых субъектом действий играют важную роль. В рамках исследования было показано, что архитектура, которая хорошо себя показывает на одних датасетах, является непригодной для анализа других

и причиной этому служит именно фундаментальные различия в данных.

В качестве дальнейших исследований автор может выделить следующие направления:

- Более полное описание пространства архитектур сверточных нейронных сетей. В этой работе параметризуется лишь небольшое число параметров сети. На практике же в сверточных нейронных сетях этих параметров гораздо больше и построение пространства, которое описывает все или хотя бы бóльшую часть возможных архитектур, представляется очень сложной, но перспективной задачей для исследования.
- При наличии пространства, которое параметризует архитектуры сети, можно использовать более продвинутые методы для поиска оптимальной структуры в этом пространстве, нежели чем обычный полный перебор с помощью кросс-валидации. В качестве возможных претендентов выступают алгоритмы суррогатного моделирования, например MSRBF [38] или TPE [36], и алгоритмы стохастической оптимизации, например генетические или имитации отжига [35].
- Наиболее интересным направлением развития работы автор считает поиск методов формализации понятия сложности для глубоких нейронных сетей. В настоящее время существует по сути одна хорошая развитая теория сложности моделей, их обобщающей способности и степени переобученности, начало которой положила теория Вапника и Червоненкиса. Но эта теория плохо применима к таким сложным моделям как нейронные сети. В последнее время начали появляться работы, которые формализуют понятие сложности для глубоких нейронных сетей и анализируют роль глубины сети с помощью радемахеровского анализа [39] и проведения аналогий с топологией [40]. Но тем не менее в этой области еще есть большой простор для исследований.

## Список литературы

- [1] *Krizhevsky A., Sutskever I., Hinton G. E.* Imagenet classification with deep convolutional neural networks // *Advances in Neural Information Processing Systems* 25. — 2012. — Pp. 1097–1105.
- [2] *Karpathy A., Toderici G.* Large-scale video classification with convolutional neural networks // *Computer Vision and Pattern Recognition (CVPR)*. — 2014. — Pp. 1725–1732.
- [3] *Graves A., Mohamed A.-R., Hinton G. E.* Speech recognition with deep recurrent neural networks. — 2013. — Pp. 6645–6649.
- [4] *Hermann K. M., al.* Teaching machines to read and comprehend // *CoRR*. — 2015. — Vol. abs/1506.03340.
- [5] *Nicolas-Alonso L. F., Gomez-Gil J.* Brain computer interfaces, a review // *Sensors*. — 2012. — Vol. 12, no. 2. — Pp. 1211–1279.
- [6] *LeCun Y., Bengio Y., Hinton G.* Deep learning // *Nature*. — 2015. — Vol. 521. — Pp. 436–444.
- [7] *Bashivan P., al.* Learning representations from eeg with deep recurrent-convolutional neural networks. — 2016.
- [8] *Nurse E. S., Karoly P. J., al.* A generalizable brain-computer interface (bci) using machine learning for feature discovery // *PLoS ONE*. — 2015. — Vol. 10, no. 6.
- [9] *Mohri M., Rostamizadeh A., Talwalkar A.* *Foundations of Machine Learning*. — The MIT Press, 2012. — 432 pp.
- [10] *Vapnik V. N.* *Statistical Learning Theory*. — Wiley-Interscience, 1998. — 768 pp.
- [11] *Lugosi G., Wegkamp M.* Complexity regularization via localized random penalties // *The Annals of Statistics*. — 2004. — Vol. 32, no. 4.
- [12] *Koltchinskii V.* Rademacher penalties and structural risk minimization // *IEEE Transactions on Information Theory*. — 2001. — Vol. 47, no. 5.
- [13] *Hastie T., Tibshirani R., Friedman J.* *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. — Springer, 2009. — 745 pp.

- [14] *Bartlett P. L., Maïorov V., Meir R.* Almost linear vc dimension bounds for piecewise polynomial networks // *Advances in Neural Information Processing Systems 11.* — The MIT Press, 1999. — Pp. 190–196.
- [15] *Bartlett P. L., Maass W.* Vapnik-chervonenkis dimension of neural nets // *The Handbook of Brain Theory and Neural Networks*, 2nd ed. — The MIT Press, 2003. — Pp. 1188–1192.
- [16] *Anthony M., Bartlett P. L.* *Neural Network Learning: Theoretical Foundations.* — Cambridge University Press, 1999. — 404 pp.
- [17] *Goodfellow I., Bengio Y., Courville A.* *Deep Learning.* — Book in preparation for MIT Press.
- [18] *Karpathy A.* Cs231n convolutional neural networks for visual recognition. <http://cs231n.github.io/convolutional-networks/>.
- [19] *Glorot X., Bordes A., Bengio Y.* Deep sparse rectifier neural networks // *Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011).* — 2011. — Pp. 315–323.
- [20] *He K., al.* Delving deep into rectifiers: Surpassing human-level performance on imagenet classification // *2015 IEEE International Conference on Computer Vision (ICCV).* — 2015. — Pp. 1026–1034.
- [21] Keras layers. <http://keras.io/layers/core/>.
- [22] *Simonyan K., Zisserman A.* Very deep convolutional networks for large-scale image recognition // *CoRR.* — 2014. — Vol. abs/1409.1556.
- [23] *He K., al.* Deep residual learning for image recognition // *CoRR.* — 2015. — Vol. abs/1512.03385.
- [24] *Tangermann M., al.* Review of the bci competition iv // *Frontiers in neuroscience.* — 2012.
- [25] *Fatourechi M., al.* Emg and eog artifacts in brain computer interface systems: A survey // *Clinical Neurophysiology.* — 2007. — Vol. 118, no. 3. — Pp. 480–494.

- [26] *Hyvarinen A., Oja E.* Independent component analysis: algorithms and applications // *Neural Networks*. — 2000. — Vol. 13, no. 4–5. — Pp. 411–430.
- [27] *Ang K. K., al.* Filter bank common spatial pattern (fbcsp) in brain-computer interface // 2008 IEEE International Joint Conference on Neural Networks. — 2008. — Pp. 2390–2397.
- [28] *Arvaneh M., al.* Optimizing spatial filters by minimizing within-class dissimilarities in electroencephalogram-based brain-computer interface // *IEEE Transactions on Neural Networks and Learning Systems*. — 2013. — Vol. 24, no. 4. — Pp. 610–619.
- [29] *Srivastava N., Hinton G., al.* Dropout: A simple way to prevent neural networks from overfitting // *Journal of Machine Learning Research*. — 2014. — Vol. 15. — Pp. 1929–1958.
- [30] *Luciw M. D., Jarocka E., Edin B. B.* Multi-channel eeg recordings during 3,936 grasp and lift trials with varying weight and friction // *Scientific Data*. — 2014. — no. 140047.
- [31] *Cukierski W.* Grasp and lift trial. [https://www.youtube.com/watch?v=y3\\_Izuop2gY](https://www.youtube.com/watch?v=y3_Izuop2gY).
- [32] *Blankertz B., al.* The non-invasive berlin brain-computer interface: Fast acquisition of effective performance in untrained subjects // *NeuroImage*. — 2007. — Vol. 37, no. 2. — Pp. 539 – 550.
- [33] *Li Y. N., Zhang X. D., Huang Z. X.* A practical method for motor imagery based real-time prosthesis control // 2012 IEEE International Conference on Automation Science and Engineering (CASE). — 2012. — Pp. 1052–1056.
- [34] *Blankertz B., Curio G., Muller K.-R.* Classifying single trial eeg: Towards brain computer interfacing // *Advances in Neural Information Processing Systems 14*. — MIT Press, 2002. — Pp. 157–164.
- [35] *Schneider J., Kirkpatrick S.* Stochastic Optimization. — Springer, 2006. — 568 pp.
- [36] *Bergstra J. S., Bengio Y., al.* Algorithms for hyper-parameter optimization // *Advances in Neural Information Processing Systems 24*. — 2011. — Pp. 2546–2554.

- [37] Kaggle grasp-and-lift eeg detection. <https://www.kaggle.com/c/grasp-and-lift-eeg-detection>.
- [38] *Regis R. G., Shoemaker C. A.* A stochastic radial basis function method for the global optimization of expensive functions // *INFORMS Journal on Computing*. — 2007. — Vol. 19, no. 4. — Pp. 497–509.
- [39] *Sun S., al.* On the depth of deep neural networks: A theoretical view // Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16). — 2016.
- [40] *Bianchini M., Scarselli F.* On the complexity of neural network classifiers: A comparison between shallow and deep architectures // *IEEE Transactions on Neural Networks and Learning Systems*. — 2014. — Vol. 25, no. 8. — Pp. 1553–1565.