Московский физико-технический институт

(Государственный университет)

Факультет управления и прикладной математики

Кафедра «Интеллектуальные системы»

ДИПЛОМНАЯ РАБОТА СТУДЕНТА 074 ГРУППЫ

«Анализ структурной и статистической сложности суперпозиции нейронных сетей»

Выполнил:

студент 4 курса 074 группы

Перекрестенко Дмитрий Олегович

Научный руководитель:

к.ф-м.н., н.с. ВЦ РАН

Стрижов Вадим Викторович

Москва, 2014

Содержание

Аннотация

Исследуется проблема определения оптимальной струкруры нейронной сети — числа нейронов и скрытых слоев. Предлагается определить оптимальную структуру нейронной сети без использования переборных методов и процедур обучения. Предлагается оценивать структурную сложность нейронной сети по структурной сложности другой модели, при этом считается что получение структурной сложности второй модели требует меньшего объема вычислений. Вводятся критерии геометрической сложности выборки и структурной сложности сети. Предлагается по сложности выборки определять субоптимальную сложность сети и искать оптимум в окрестности этой сложности. Предложен алгоритм построения сети субоптимальной сложности для заданной задачи классификации. Предложен способ прогнозирования сложности сети по сложности выборки. Качество алгоритма проверяется на задачах классификации разнородных выборок реальных данных.

Ключевые слова: структурная сложность, автокодировщик, обучение без учителя, глубокое обучение, нейронные сети

1 Введение

В данной работе решается задача многоклассовой классификации временных рядов акселерометра по классам физической активности человека. В работе рассматривается задача классификации по четырем классам активности – бег, ходьба, сидение и стояние. Ранее [1] задача решалась путем экспертного выбора признаков, таких как среднее значение ускорения для каждой координаты, время между пиками ускорения и т.д. Классифицирующая нейрононная сеть обучалась на признаковых описаниях входных данных. В данной работе задача решается методами глубокого обучения — методами, которые дополняют алгоритм обратного распространения ошибки фазой предобучения без учителя. Тем самым позволяя более точно оптимизировать нейронные сети с большим количеством скрытых слоев [2]. Впервые идея предобучения сети была упомянута в 2006 году в работе [2], где был представлен новый класс моделей, называемых *deep belief networks* (DBN). Они состоят из стека ограниченных Больцмановских машин (RBM). Ключевая особенность сетей DBN — жадный послойный алгоритм обучения, который оптимизирует веса DBN за линейное время по отношению к размеру и глубине сети. В работе [2] было выяснено, что инициализация весов нейронной сети с помощью соответсвующей ей DBN дает лучший результат по сравнению с произвольной инициализацией параметров, которая применялась ранее. Также в 2006 году в работе [3] был предложен вариант предобучения нейронных сетей с помощью автокодировщиков. Данных подход полностью аналогичен описанному в [2], но вместо стека из ограниченных Больцмановских машин применяется стек из автокодировщиков.

Проблема нахождения архитектуры нейронной сети, которая имеет хорошую обобщающую способность поднималась в литературе не раз. В частности в работе [4] было выведено эмпирическое правило Уидроу, которое задает минимально необходимое количество параметров-весов необходимое для обобщения выборки мощности N нейронной сетью:

$$N = O\left(\frac{W}{\varepsilon}\right),$$

где W — количество параметров сети, а ε — допустимая часть неправильно классифицированных объектов. Кроме того, на вопрос о количестве скрытых слоев необходимых для аппрокимации выборки нейронной сетью отвечает теорема об универсальной аппроксимации:

Теорема об универсальной аппроксимации

Обозначим $\phi(\cdot)$ — нетривиальную, ограниченную и монотонно-возрастающую непрерывную функцию. Обозначим I_{m_0} — m_0-мерный гиперкуб $[0,1]^{m_0}$. Пространство непрерывных функций на I_{m_0} обозначим $C(I_{m_0})$. Тогда, для любой функции $f \in C(I_{m_0})$ и любого $\varepsilon > 0$, существует целое m_1 и действительные α_i, b_i и ω_{ij}, где $i = 1, \ldots, m_1$ и $j = 1, \ldots, m_0$, такие, что функция:

$$F(x_1, \ldots, x_{m_0}) = \sum_{i=1}^{m_1} \alpha_i \phi(\sum_{j=1}^{m_0} \omega_{ij} x_j + b_i),$$

есть аппроксимация реализации функции $f(\cdot)$, такая что

$$|F(x_1, \ldots, x_{m_0}) - f(x_1, \ldots, x_{m_0})| < \varepsilon,$$

для всех $x_1, x_2, \ldots, x_{m_0}$ из любой конечной выборки $D = [x_1, x_2, \ldots, x_{m_0}, f(x_1, \ldots, x_{m_0})]$.

Согласно этой теореме нейронной сети с одним скрытым слоем расмера m_0 и выходным слоем размера m_1 и сигмоидными функциями активации достаточно для аппроксимации произвольной выборки с любой точностью. Однако эта теорема говорит только об аппроксимации и ничего не говорит об конкретном размере скрытого слоя m_0. Вопрос о размере скрытого слоя решает в [5] Баррон, где приводится данная теорема:

Теорема Баррон (1993)

Для каждой непрерывной функции $f(\mathbf{x})$ с конечным первым моментом

$$C_f = \int_{\mathbb{R}^{m_0}} |\tilde{f}(\boldsymbol{\xi})| \times ||\boldsymbol{\xi}||^{\frac{1}{2}} d\boldsymbol{\xi}$$

и для каждого $m_1 \geq 1$, существует линейная комбинация сигмоидальных функций $F(\mathbf{x})$ такая, что если для функции $f(\mathbf{x})$ заданы ее значения на множестве $\{\mathbf{x}\}_{i=1}^N$ значений входного вектора $\mathbf{x}$, которые лежат в шаре $B_r = \{\mathbf{x} : ||\mathbf{x}|| \leq r\}$, то существует оценка на эмпирический риск:

$$Q_{av}(N) = \frac{1}{N} \sum_{i=1}^{N} \Big(f(\mathbf{x}_i) - F(\mathbf{x}_i) \Big)^2 \leq \frac{(2rC_f)^2}{m_1}.$$

Этот результат используется для оценки эмпирического риска для двуслойной нейронной сети с входным слоем размера m_0 и скрытым размера m_1. Оценка эмпирического риска:

$$Q_{\text{av}}(N) \leq O\left(\frac{C_f^2}{m_1}\right) + O\left(\frac{m_0 m_1}{N} \log N\right).$$

Но данный результат ничего не говорит об обобщающих способностях построенной таким образом нейронной сети. Проблема построения сети с оптимальной обобщающей способностью рассматривается в работах [6],[7],[8]. В этих работах считается, что обобщающая способность ухудшается при усложнении модели, поэтому в этих работах наряду с эмпирическим риском предлагается минимизировать также некий функционал сложности сети Q_C. В [6] минимизируется функция общего риска:

$$R(\mathbf{w}) = Q_{\text{av}}(\mathbf{w}) + \lambda Q_C(\mathbf{w}),$$

где λ это параметр регуляризации. В работе [6] функционал сложности задается как:

$$Q_C(\mathbf{w}) = ||\mathbf{w}||^2,$$

где $\mathbf{w}$ — вектор параметров модели. Этот функционал минимизирует все параметры, тем самым деля множество параметров на две группы — значимые и незначимые. В итоге, незначимые будут ≈ 0, а значимые будут ограничены в росте. Тем самым уменьшается эффект переобучения и улучшается обобщающая способность. Другой подход, называемый оптимальным прореживнием реализован в работе [7], где для проверки значимости весов используется матрица Гэссе функции эмпирического риска. Суть метода заключается в минимизации функции ΔQ_{av}:

$$\Delta Q_{\text{av}} = Q_{\text{av}}(\mathbf{w} + \Delta\mathbf{w}) - Q_{\text{av}}(\mathbf{w}) = \frac{1}{2}\Delta\mathbf{w}^\top \mathbf{H} \Delta\mathbf{w},$$

где $\mathbf{H}$ — матрица Гессе функции $Q_{\text{av}}(\mathbf{w})$.

Алгебраический подход к измерению сложности классификатора был предложен в работе [8], где было введено понятие размерности Вапника-Червоненкиса (VC-размерности).

Определение Пусть задано множество X и некоторое семейство индикаторных функций (алгоритмов классификации, решающих правил) $\mathfrak{F} = \{f(\mathbf{x}, \mathbf{w})\}$, где $\mathbf{x} \in X$ — аргумент функций, $\mathbf{w}$ — вектор параметров, задающий функцию. Каждая такая функция $f(\mathbf{x}, \mathbf{w})$ сопоставляет каждому элементу множества X один из двух заданных классов. VC-размерностью семейства $\mathfrak{F}$ называется наибольшее число s, такое, что существует подмножество из s элементов множества X, которые функции из $\mathfrak{F}$ могут разбить на два класса всеми возможными способами. Если же такие подмножества существуют для сколь угодно большого s, то VC-размерность полагается равной бесконечности.

Концепция VC-размерности напрямую связана с обобщающей способностью сети-классифиатора. В работе [9] приведены верхние и нижние оценки размерности нейронных сетей прямого распространения с сигмоидальными функциями активации. Верхняя оценка сети, обозначим ее $\mathbf{N}$, с количеством параметров W равна:

$$\mathrm{VCdim}(\mathbf{N}) = O(W^4),$$

нижняя оценка:

$$\mathrm{VCdim}(\mathbf{N}) = \Omega(W^2).$$

В данной работе, в отличие от работ [6-9], оценки сложности сети имеют графовое происхождение, например сложность задается как сумма весов всех ребер по всем возможным подграфам графа-нейронной сети. Кроме того, сложность не используется в качестве компоненты функции общего риска, которая является целевой функцией минимизации. Вместо этого предлагается оценивать суб-оптимальную структурную сложность сети по геометрической сложности выборки. Геометрическая сложность выборки задается таким образом, чтобы ее нахождение требовало значительно меньшего объема вычислений чем получение оптимальной структурной сложности напрямую. Для нахождения оптимума используются введеные в работе графовые сложности сети. Для этого по геометрической сложности выборки оценивается графовая сложность сети и после оптимум ищется с помощью перебора моделей имеющих сложность в ε-окрестности полученной сложности. В работе не рассматривается процесс нахождения моделей в ε-окрестности сложности, эта проблема остается предметом для дальнейших исследований.

2 Постановка задачи

Поставим задачу классификации. Задана выборка D из генеральной совокупности D_{gen}:

$$D = \{\mathbf{x}_i, y_i\}_{i=1}^{m}, \tag{1}$$

где $\mathbf{x}_i \in \mathbb{R}^n$ — вектор, набор n-точечных временных рядов, а $y_i \in \{1, 2, \ldots, k\}$ — метка класса из номинальной шкалы. Требуется найти модель:

$$\mathbf{f} : (\mathbf{x}, \mathbf{w}) \to \mathrm{c},$$

$$\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^w \to \{1, \ldots, k\}$$

из множества $\mathbf{f} \in \mathfrak{F}$ нейронных сетей, которая классифицирует генеральную совокупность D_{gen}.

2.1 Мультиномиальная логистическая регрессия

Для решения задачи мультиклассовой классификации нейронной сетью, необходимо чтобы последний слой сети был классификатором. В силу того, что каждый нейрон сети реализует классификацию на два класса методом логистической регрессии, то логистическая регрессия обобщается на случай многоклассовой классификации, но для этого используется целый слой нейронов вместо одного.

Мультиномиальная логистическая регрессия — это метод классификации обобщающий метод логистической регрессии для задач многоклассовой классификации. Обозначим y зависимую переменную, принимающую значения из $\{1, 2, \ldots, k\}$. Обозначим вектор регрессоров $\mathbf{x}$. Предположим, что вероятность наступления события $y = \mathrm{c} \in \{1, \ldots, k\}$ равна:

$$p(y = \mathrm{c}|\mathbf{x}; \boldsymbol{\theta}) = \frac{e^{\boldsymbol{\theta}_\mathrm{c}^\top \mathbf{x}}}{\sum_{j=1}^{k} e^{\boldsymbol{\theta}_j^\top \mathbf{x}}}, \tag{2}$$

где $\boldsymbol{\theta}_j$ — параметры регрессии. Итоговая модель регрессии:

$$\mathbf{f}(\mathbf{x}) = \arg\max_l \left(\begin{bmatrix} p(y=1|\mathbf{x};\boldsymbol{\theta}) \\ p(y=2|\mathbf{x};\boldsymbol{\theta}) \\ \vdots \\ p(y=k|\mathbf{x};\boldsymbol{\theta}) \end{bmatrix}^\top \cdot \mathbf{e}_l \right) = \arg\max_l \left(\frac{1}{\sum_{j=1}^{k} e^{\boldsymbol{\theta}_j^\top \mathbf{x}}} \begin{bmatrix} e^{\boldsymbol{\theta}_1^\top \mathbf{x}} \\ e^{\boldsymbol{\theta}_2^\top \mathbf{x}} \\ \vdots \\ e^{\boldsymbol{\theta}_k^\top \mathbf{x}} \end{bmatrix}^\top \cdot \mathbf{e}_l \right), \tag{3}$$

где $\mathbf{e}_l$ — l-й столбец единичной матрицы $\mathbf{E}_k$. Для нахождения параметров используется метод максимума апостериорной вероятности:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} \prod_{\mathrm{c}=1}^{k} [y_i = \mathrm{c}] \cdot p(y = y_i|\mathbf{x}_i; \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta}). \tag{4}$$

В качестве априорного распределения $\boldsymbol{\theta}$ выступает многомерное нормальное распределение $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ с нулевым средним и матрицей ковариации $\sigma^2\mathbf{I}$, соответствующее априорному убеждению о том, что все коэффициенты регрессии $\boldsymbol{\theta}$ должны быть небольшими числами. Задача (4) нахождения оптимальных параметров $\boldsymbol{\theta}$ эквивалентна минимизации функции ошибки:

$$S(\boldsymbol{\theta}) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{\mathrm{c}=1}^{k} [y_i = \mathrm{c}] \log p(y_i = \mathrm{c}|\mathbf{x}_i; \boldsymbol{\theta}) + \alpha ||\boldsymbol{\theta}||_F,$$

где m — количество элементов в обучающей выборке, k — количество классов, α — параметр регуляризации, $||\boldsymbol{\theta}||_F$ — норма Фробениуса матрицы $[\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k]$.

2.2 Автокодировщик

Метод обратного распространения ошибки не применяется для оптимизации параметров нейронных сетей с большим количеством скрытых слоев из-за очень низкой скорости сходимости метода, а также сходимости к незначительным локальным минимумам. Однако если его применять к сети с параметрами находящимися в окрестности своих оптимальных значений, то он показывает себя хорошо. Для приближения к оптимальным значениям параметров нейронная сеть разбивается на слои, называемые автокодировщиками и оптимизирует параметры, минимизируя функцию ошибки автокодировщика.

Автокодировщик $\mathbf{h}$ это монотонное нелинейное отображение входного вектора свободных переменных $\mathbf{x} \in \mathbb{R}^n$ в скрытое представление $\mathbf{h} \in \mathbb{R}^\nu$ следующего вида:

$$\mathbf{h}(\mathbf{x}) = \boldsymbol{\sigma}(\underset{\nu \times n}{\mathbf{W}}\mathbf{x} + \mathbf{b}). \tag{5}$$

Скрытое представление $\mathbf{h}$ создает линейную реконструкцию вектора $\mathbf{x}$:

$$\mathbf{r}(\mathbf{x}) = \underset{n \times \nu}{\mathbf{W}'}\mathbf{h} + \mathbf{b}'. \tag{6}$$

Параметры автокодировщика

$$\boldsymbol{\lambda} = \{\mathbf{W}', \mathbf{W}, \mathbf{b}', \mathbf{b}\} \tag{7}$$

оптимизированы таким образом, чтобы сделать реконструкцию $\mathbf{r}(\mathbf{x})$ максимально близкой к $\mathbf{x}$. Реконструкция $\mathbf{r}(\mathbf{x})$ получится значимо отличной от $\mathbf{x}$ если размерность скрытого представления ν слишком мала и если компоненты x_j вектора $\mathbf{x}$ независимы друг от друга, например если все x_j независимые одинаково распределенные нормальные величины. Однако если входной вектор $\mathbf{x}$ имеет структуру, например, если некоторые из его компонент между собой корреллируют, то автокодировщик может обнаружить эти корреляции. Рассуждения выше основываются на том, что размерность скрытого представления ν меньше чем число элементов n вектора $\mathbf{x}$, однако их можно обобщить на произвольный размер скрытого отображения $\mathbf{h}$, наложив ограничение разреженности на процедуру реконструкции. Разреженность состоит в том, чтобы большинство компонент скрытого представления $\mathbf{h}$ были ≈ 0. Другими словами, разреженность равносильна малости для каждого j среднего значения j-ой

компоненты вектора $\mathbf{h}$ по всей выборке D:

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m} h_j(\mathbf{x}_i). \tag{8}$$

Следовательно требуется выполнения равенства $\hat{\rho}_j = \rho$, где ρ — параметр разреженности (малая величина, например 0.05). Для реализации этого ограничения к функции ошибки добавляется дополнительное штрафное слагаемое, например:

$$\sum_{j=1}^{m} \mathrm{KL}(P_\rho || P_{\hat{\rho}_j}) = \sum_{j=1}^{m} \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}, \tag{9}$$

где $\mathrm{KL}(P_\rho || P_{\hat{\rho}_j})$ — дивергенция Кульбака-Лейблера между распределением бернулевской случайной величины со средним ρ и бернулевской случайной величины со средним $\hat{\rho}_j$. Здесь разумно применять дивергенцию Кульбака-Лейблера, т.к. это стандартная мера схожести двух распределений. Итоговая функция ошибки автокодировщика:

$$S(\boldsymbol{\lambda}) = \frac{1}{2m}\sum_{i=1}^{m} \|\mathbf{r}(\mathbf{x}_i) - \mathbf{x}_i\|^2 + \|\mathbf{W}\|_F^2 + \beta \sum_{j=1}^{m} \Big[\rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}\Big], \tag{10}$$

где первое слагаемое отвечает за среднеквадратичную ошибку реконструкции (6), второе слагаемое есть регуляризация, а третье слагаемое разреживает значения скрытого представления (9). Тут m — количество элементов в обучающей выборке, β — вес разреживающего слагаемого, ρ — параметр разреженности, желаемое среднее значение каждой компоненты скрытого представления $\mathbf{h}$, а $\hat{\rho}_j$ — среднее значение j-ой компоненты вектора $\mathbf{h}$ (8).

2.3 Структура модели

В данной работе модель $\mathbf{f}$ представлена в виде суперпозиции блоков:

$$\mathbf{f} = \mathbf{a}(\mathbf{h}_N(\dots \mathbf{h}_1(\mathbf{x}))),$$

где $\mathbf{h}_k$ — блоки-автокодировщики (5), вида

$$\mathbf{h}_k(\mathbf{x}) = \boldsymbol{\sigma}(\mathbf{W}_k\mathbf{x} + \mathbf{b}_k),$$

а блок $\mathbf{a}$ — классификатор мультиномиальной логистической регрессии вида

$$\mathbf{a}(\mathbf{x}) = \arg\max_l \left(\frac{1}{\sum_{j=1}^{k} e^{\boldsymbol{\theta}_j^\top \mathbf{x}}} \begin{bmatrix} e^{\boldsymbol{\theta}_1^\top \mathbf{x}} \\ e^{\boldsymbol{\theta}_2^\top \mathbf{x}} \\ \vdots \\ e^{\boldsymbol{\theta}_k^\top \mathbf{x}} \end{bmatrix}^\top \cdot \mathbf{e}_l \right),$$

где $\mathbf{e}_l$ — l-й столбец единичной матрицы $\mathbf{E}_k$. Функция ошибки модели $\mathbf{f}$:

$$S(\boldsymbol{\alpha}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}[y_i = j]\log p(\hat{y}_i = j|\mathbf{x}_i;\boldsymbol{\alpha}),$$

где $\boldsymbol{\alpha} = \{\mathbf{W}_1, \ldots, \mathbf{W}_N, \mathbf{b}_1, \ldots, \mathbf{b}_N, \boldsymbol{\theta}\}$ — вектор состоящий из параметров всех блоков модели $\mathbf{f}$, а

$$p(\hat{y}_i = j|\mathbf{x}_i;\boldsymbol{\alpha}) = \frac{e^{\boldsymbol{\theta}_i^T \mathbf{h}_N(\ldots\mathbf{h}_1(\mathbf{x}_i))}}{\sum_{j=1}^{k} e^{\boldsymbol{\theta}_j^T \mathbf{h}_N(\ldots\mathbf{h}_1(\mathbf{x}_i))}}.$$

Требуется найти вектор параметров $\boldsymbol{\alpha}_{\text{opt}} \in \mathbb{W}$, который минимизирует функцию ошибки на заданной выборке D:

$$\boldsymbol{\alpha}_{\text{opt}} = \arg\min_{\boldsymbol{\alpha}\in\mathbb{W}} S(\boldsymbol{\alpha}|D). \tag{11}$$

3 Выбор модели

3.1 Функции структурной сложности модели

Будем вводить критерий структурной сложности так, чтобы чем выше была структурная сложность, тем выше был объем вычислений требуемый для оптимизации параметров. Таким образом, при равном качестве классификации двух моделей предпочительно выбирать модель с меньшей структурной сложностью. Кроме того, к преимуществам выбора модели с меньшим количеством параметров относится удовлетворение принципа бритвы Оккама. Уменьшая количество параметров при этом не ухудшая качество классификации, мы тем самым уменьшаем вероятность переобучения.

Определение 1. Назовем функцию $g : \mathbb{W} \to \mathbb{R}$ *структурной сложностью* модели $A = \{\mathbf{f}(\mathbf{x}, \mathbf{w})|\mathbf{w} \in \mathbb{W}\}$, если $g(\mathbf{w})$ возрастает с ростом N, где N — математическое ожидание числа элементарных шагов алгоритма настройки модели к параметрам $\mathbf{w}$ из случайно заданного начального приближения $\mathbf{w}_0$.

Все четыре (12),(13),(14),(15) нижеописанные сложности сильно зависят от количества параметров. Сложность (12) зависит только от количества параметров, тогда как сложность (14) дополнительно учитывает глубину сети. Сложности (13) и (15) есть взвешенные модификации сложностей (12) и (14) соответственно. Взвешивание параметров имеет смысл, так как значение параметра прямо пропорционально его влиянию на результат классификации.

1. Количественная сложность:

$$\mathrm{Comp}_1(\mathbf{f}) = k\nu_N + \sum_{i=1}^{N} \nu_i(\nu_{i-1} + 1), \tag{12}$$

где k — количество классов, а ν_i — размер i-го блока(скрытого слоя) модели $\mathbf{f}$. Размерная сложность это количество параметров модели $\mathbf{f}$.

2. Взвешенная количественная сложность:

$$\mathrm{Comp}_2(\mathbf{f}) = \|\hat{\boldsymbol{\theta}}\|_F^2 + \sum_{i=1}^{N} \Big(\|\hat{\mathbf{W}}_i\|_F^2 + \|\hat{\mathbf{b}}_i\|_F^2 \Big), \tag{13}$$

где $\hat{\boldsymbol{\theta}}$ — матрица настроенных параметров классификатора см. (11), а $\hat{\mathbf{W}}_i, \hat{\mathbf{b}}_i$ — настроенные параметры i-го блока-автоэнкодера см. (7). Взвешенная размерная сложность это сумма значений всех параметров модели.

3. Графовая сложность: представим слоевую нейронную сеть в виде взвешенного $N+2$-дольного графа с ориентированными к нейронам большей доли ребрами(входной слой имеет номер доли - 1, выходной - $N+2$; вес ребра равен квадрату значения соответствующего параметра). Назовем i-ым подграфом графа V граф V^i, который состоит из всех вершин в V в которые существует маршрут из i-ой вершины, а также из всех ребер V, которые соединяют вершины V^i. Назовем сложностью слоевой нейронной сети-графа V количество ребер во всех его подграфах. Итого, сложность модели-нейросети $\mathbf{f}$:

$$\mathrm{Comp}_3(\mathbf{f}) = \sum_{i=1}^{M} \sum_{(k,j)\in V^i} \omega_{kj}, \tag{14}$$

где V^i — i-й подграф, ω_{ij} — индикатор существования ребра (k,j), M — число вершин графа.

4. Взвешенная графовая сложность:

$$\mathrm{Comp}_4(\mathbf{f}) = \sum_{i=1}^{M} \sum_{(k,j)\in V^i} w_{kj}, \tag{15}$$

где V^i — i-й подграф, w_{kj} — вес ребра (k,j), M — число вершин графа.

Утверждение 1. Количественная Comp_1 и графовая Comp_3 сложности являются *структурными сложностями* для слоевых нейронных сетей настраиваемых алгоритмом глубокого обучения.

3.2 Функции сложности выборки

Геометрическая сложность вводится таким образом, чтобы она монотонно возрастала вместе с минимальной структурной сложностью модели, которая ее хорошо классифицирует. Кроме того задача вычисления геометрической сложности должна быть значительно менее трудоемкой, чем задача вычисления структурной сложности. Поэтому, логично вводить геометрическую сложность как сложность либо некой простой непараметрической модели либо как сложность некой сложной модели, параметры какой вычисляются более эффективной процедурой чем полный перебор. В данной работе для оценки сложности выборки используется оба подхода. В первом - сложность измеряется как сумма размерностей собственного подпространтсва всех производных выборки. Размерность собственного пространства определяется с помощью автокодировщика, поэтому, одновременно с вычислением сложности выборки мы получаем в жадном смысле оптимально классифицирующую архитектуру нейронной сети. Во втором подходе геометрическая сложность выборки оценивается по структурной сложности приближающей ее непараметрической модели такой как сеть радиальных базисных функций.

3.2.1 Размерная сложность

Определение 2. Назовем k-ой производной выборки D выборку $D_k = \{\mathbf{x}_{ik}, y_i\}_{i=1}^m$, где

$$\mathbf{x}_{ik} = \mathbf{h}_k(\mathbf{h}_{k-1}(\dots \mathbf{h}_1(\mathbf{x}_i))), \tag{16}$$

$\mathbf{x}_{ik}$ — вектор значений нейронов k-го скрытого слоя нейронной сети $\mathbf{f}$ при аргументе $\mathbf{x}_i$. Заметим, что $D_0 = D$.

Определение 3. Назовем *размерной сложностью* выборки D число нейронов в скрытом слое автокодировщика, задающее минимум его функции ошибки.

Определение 4. Назовем производную выборки D *значимой*, если если ее номер меньше $N + 1$, где N определяется так:

$$N = \arg\min_k [\mathrm{Comp}(D_k) = \mathrm{Comp}(D_{k+1})].$$

$$\mathrm{Comp}(D) = \arg \min_{\mathrm{size}(\hat{\mathbf{W}},1)} S(\hat{\boldsymbol{\lambda}}|D),$$

где $\hat{\boldsymbol{\lambda}}$ — оптимальные согласно функции (10) параметры (7) автокодировщика.

Определение 5. Назовем *суммарной размерной сложностью* выборки D сумму размерных сложностей всех значимых производных выборки. Таким образом, *суммарная размерная сложность* определяется как сумма *размерных сложностей* всех значимых производных выборки D:

$$\mathrm{DimComp}(D) = \sum_{k=0}^{N} \mathrm{Comp}(D_k).$$

Стратегия построения модели Здесь приводится процедура построения нейросети у которой количественная структурная сложность совпадает с размерной сложностью выборки. В процедуре построения нейронной сети используется жадный алгоритм. Суть процедуры в том, что размеры блоков $\mathbf{h}_i$ оптимизируются не в совокупности, а последовательно. Тем самым задача N-мерной оптимизации нейронной сети сводится к N одномерным задачам. Сначала мы вычисляем число скрытых слоев (блоков-автоэнкодеров), которое определяется как минимальное число слоев — N, при достижении которого сложность выборки D перестает уменьшаться. Т.е.

$$N = \arg\min_k \Big[\mathrm{Comp}(D_k) = \mathrm{Comp}(D_{k+1})\Big].$$

Потом для каждого блока $\mathbf{h}_k$ (k-й скрытый слой сети $\mathbf{f}$) задаем размер таким же как $\mathrm{Comp}(D_{k-1})$. Итоговая сеть представится в таком виде:

$$\mathbf{f} = \mathbf{a}(\mathbf{h}_N(\dots \mathbf{h}_1(\mathbf{x}))),$$

где $\mathbf{h}_j$ — блоки-автоэнкодеры, каждый со скрытым слоем размера $\mathrm{Comp}(D_j)$.

3.2.2 Геометрическая сложность

Определение 6. Назовем γ-*геометрической сложностью* выборки D минимальное число радиальных базисных функций $\phi_k(\mathbf{x}) = \exp(-\frac{\rho(\mathbf{c}_k - \mathbf{x})}{a_k^2})$ необходимых для классификации выборки D с точностью γ.

Метод классификации:

$$a(\mathbf{x}) = \arg\max_{y \in Y} \lambda_y P_y \sum_{i=1}^{k_y} \omega_{yj} \phi_{yj}(\mathbf{x}'_i).$$

Функция ошибки:

$$J(D) = -\frac{1}{l} \sum_{i=1}^{l} [y'_i = a(\mathbf{x}_i)],$$

где l — мощность выборки. Настройка параметров базисных функций проводится с помощью EM-алгоритма. Геометрическая сложность это минимальное число радиальных базисных функций задающее ошибку не более чем $-\gamma$ на скользящем контроле. Зададим множество K так, чтобы выполнялось:

$$\Big[k \in K\Big] \Leftrightarrow \Big[\frac{1}{N}\sum_{i=1}^{N} J(D/X_i, k) + \gamma < 0\Big],$$

тогда

$$\mathrm{GeomComp}_{\gamma}(D) = \min_{k}\Big[k \in K\Big],$$

где $k = \sum_{y \in Y} k_y$ — общее число радиальных базисных функций, X_i — случайная выборка без повторений из D.

4 Прогнозирование оптимальной структурной сложности модели

Гипотеза *Предполагается что между сложностью выборки и структурной сложностью нейросети есть зависимость.*

Пусть задано M выборок $\{D_1, \ldots, D_M\}$, где $D_i = \{\mathbf{x}_m^i, y_m^i\}_{m=1}^n$, таких что для каждой из них известна оптимальная структурная сложность классифицирующей их нейронной сети. Будем восстанавливать по этим выборкам регрессию структурной сложности по геометрической:

$$\hat{\boldsymbol{\chi}} = \arg\min_{\boldsymbol{\chi}} \sum_{i=1}^{M} \Big(\boldsymbol{f}(\mathrm{GeomComp}_i, \boldsymbol{\chi}) - \mathrm{StrComp}_i\Big)^2,$$

получив модель регрессии мы можем получать суб-оптимальные значения структурной сложности сети для заданной геометрической сложности выборки.

$$\mathrm{StrComp}_{\mathrm{subopt}}(\mathrm{GeomComp}) = \boldsymbol{f}(\mathrm{GeomComp}, \hat{\boldsymbol{\chi}}),$$

где $\boldsymbol{\chi}$ — вектор параметров. Теперь можно найти оптимальную структурную сложность, т.к. считается, что она лежит в ε-окрестности суб-оптимальной сложности.

5 Вычислительный эксперимент

5.1 Визуализация работы автокодировщика

Работа автокодировщика проиллюстрирована на задаче понижения размерности четырехмерного множества точек до трех и двухмерной размерности. На рис. 1 изображено обрабатываемое множество, которое произвольным преобразованием поворота было переведено в четырехмерное пространство. На рис. 2 показаны продукты отображения обрабатываемого множества в трехмерную и двумерную области с помощью автокодировщика. Проекции полученные автокодировщиком сравнивается с методом главных компонент(PCA) для тех же обрабатываемых данных рис. 3.

На рис. 4 визуализирована процедура оптимизации параметров автокодировщика. Так как количество параметров автокодировщика существенно больше трех, то на рисунке изображена процедура оптимизации проекции вектора параметров $\boldsymbol{\lambda}$ на трехмерную плоскость с помощью произвольной матрицы поворота $\mathbf{Z}$:

$$\mathbf{t}(\mathbf{x}) = \underset{3\times W}{\mathbf{Z}} \cdot \underset{W\times 1}{\boldsymbol{\lambda}},$$

тут W — количество параметров. Из-за того что процесс оптимизации представляет собой задачу поиска минимума многоэкстремальной функции, то результат сильно зависит от начального приближения. На графике крестом отмечено начальное приближение, окружностью — проекцию вектора параметров к которой сошлась процедура оптимизации. Радиус окружности обратно пропорционален ошибке $S(\boldsymbol{\lambda})$.

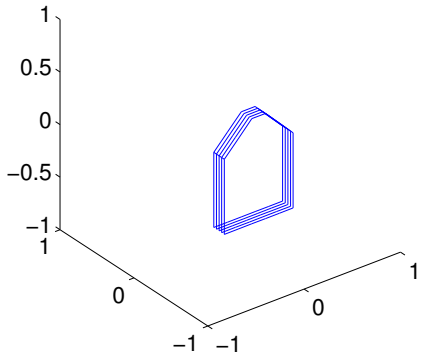


Рис. 1: Оригинал(3d)

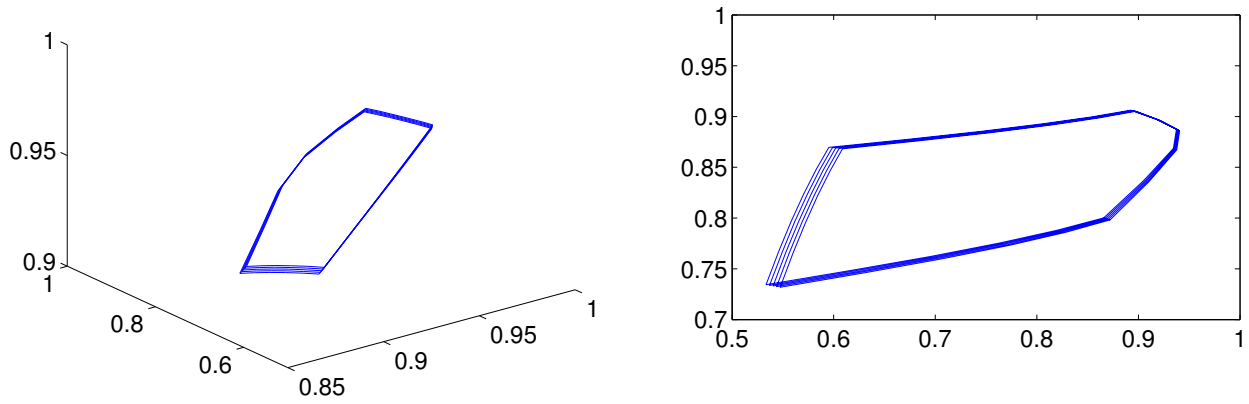


Рис. 2: Восстановленное автокодировщиком из 4d(слева) и из 3d(справа)

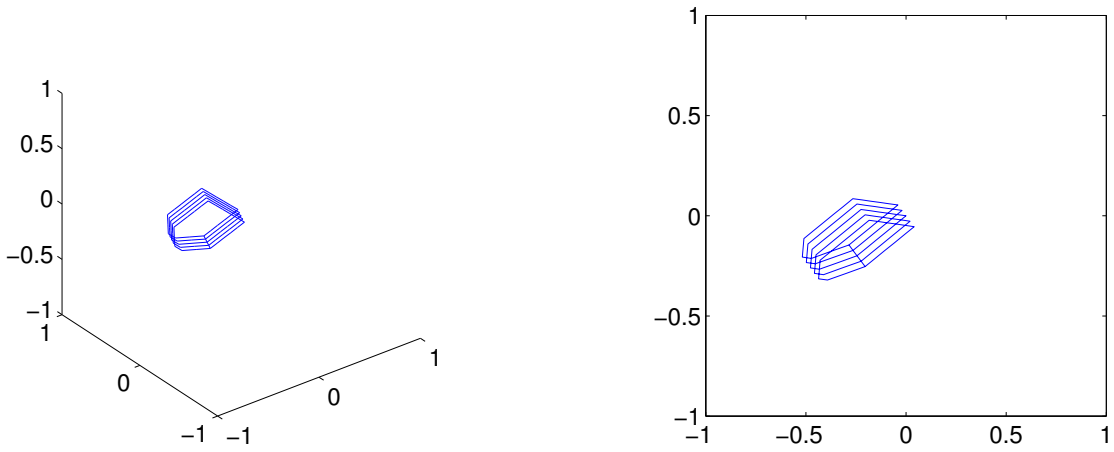


Рис. 3: Понижение размерности с помощью PCA 3d(слева) и 2d(справа)

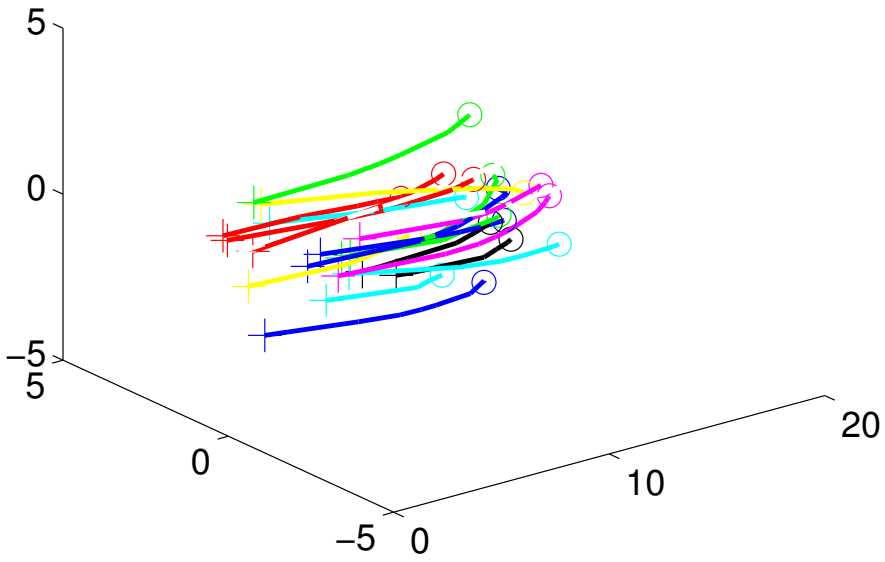


Рис. 4: Обучение параметров автокодировщика при различных начальных приближениях

5.2 Практическая задача исследования зависимости между структурной сложностью модели и сложностью выборки

Вычислительный эксперимент состоит из двух этапов, сначала исследуется зависимость на выборках с малой геометрической сложностью(<20), для классификации которых хватает нейронной сети с 1-2-мя скрытыми слоями. На выборках с малой сложностью отбираются пары сложностей хорошо согласующиеся с моделью линейной регрессии. После чего отобранные пары исследуются на выборках с высокой геометрической сложностью (>20), для классификации которых требуются нейронные сети с количеством скрытых слоев слоев более одного. Параметр геометрической сложности γ был принят равным 0.95.

5.2.1 Случай простых выборок

Для вычислительного эксперимента использовалось 5 выборок:

1. Временные ряды акселерометра. Количество признаков — 600, классов — 4.

2. Синтетически сгенерированная выборка. Количество признаков — 2, классов — 2.

3. Распознавание сортов вин. Количество признаков — 13, классов — 3.

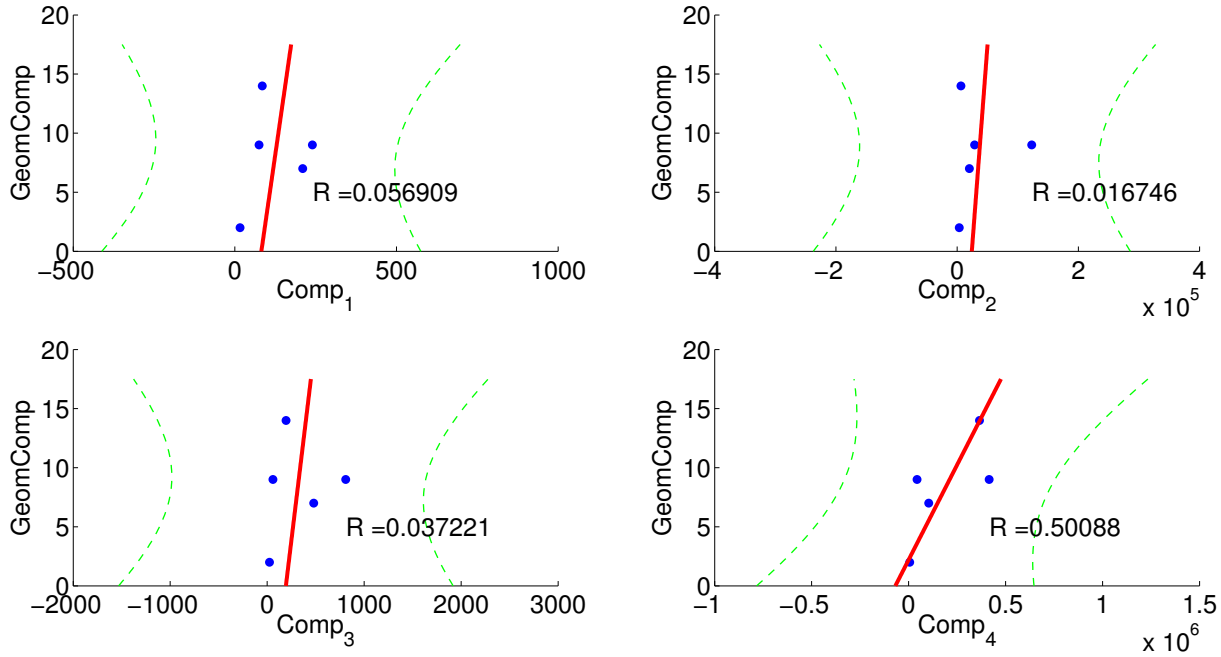
4. Распознавание ирисов. Количество признаков — 4, классов — 3.

5. Распознавание патологий кожного покрова груди. Количество признаков — 9, классов — 6.

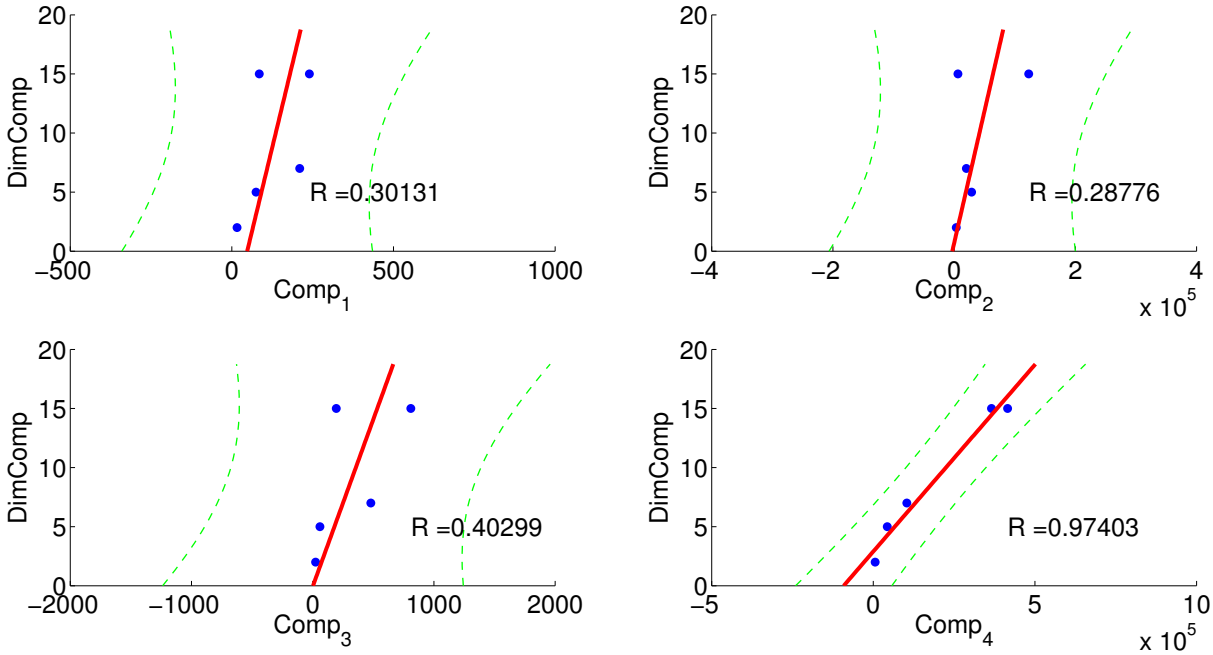
Ниже в таблице приведены вычисленные значения сложностей выборки и структурных сложностей нейронных сетей достаточных для классификации заданной выборки.

DataSet	$Comp_1$	$Comp_2$	$Comp_3$	$Comp_4$	$GeomComp_{0.95}$	DimComp
№1	210	2.0243e+04	4800	1.0360e+05	7	7
№2	75	2.8950e+04	60	4.3208e+04	9	5
№3	85	6.2355e+03	195	3.6551e+04	14	20
№4	16	3.6485e+03	24	5.6853e+03	2	2
№5	240	1.2314e+05	810	5.1547e+05	6	15

Прогнозирование структурной сложности модели по геометрической сложности выборки



Прогнозирование структурной сложности модели по размерной сложности выборки

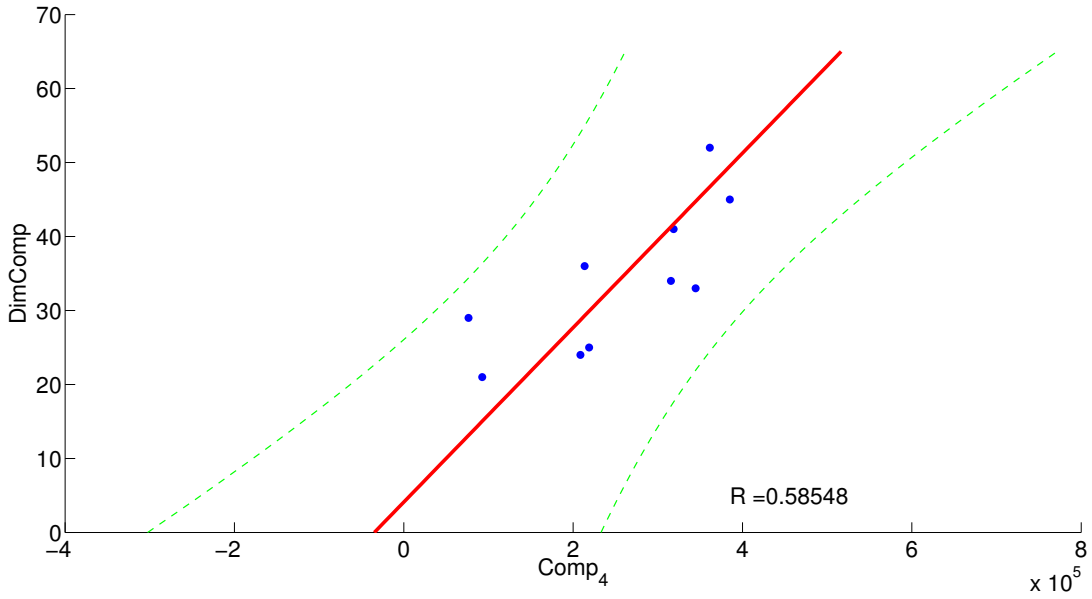


Как видно из графиков, пары (DimComp, Comp_4) и (GeomComp, Comp_4) оказались наиболее отвечающими линейной модели. Остальные пары имеют коэффициент детерминации < 0.5, что маловероятно при линейной зависимости данных. Будем рассматривать случай сложных выборок только для пар сложностей согласующихся с линейной моделью.

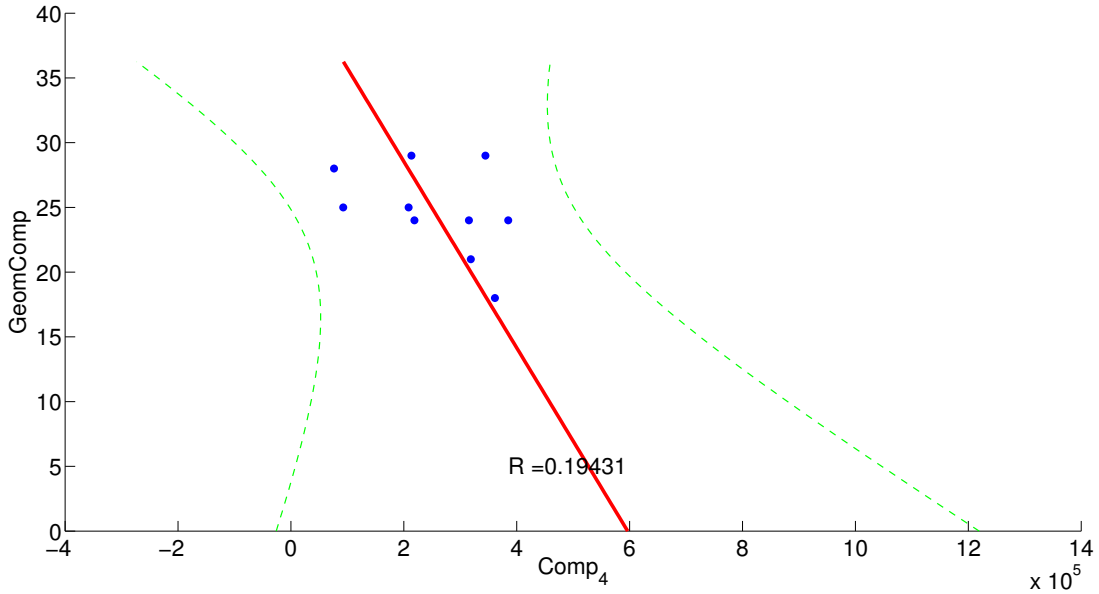
5.2.2 Случай сложных выборок

Выборки с высокой геометрической сложностью порождались из выборки рукописных цифр MNIST. Так как задача классификации MNIST имеет 10 классов, то можно породить C_{10}^d выборок d-классовой классификации. Для данной задачи, чтобы сохранить сложность максимальной d было принято равным 9-ти.

Прогнозирование структурной сложности модели по геометрической сложности выборки



Прогнозирование структурной сложности модели по размерной сложности выборки



Как видно пара ($Comp_4$, DimComp) по прежнему хорошо согласуется с линейной моделью. Эту пару рекомендуется использовать для практических задач.

6 Заключение

- Реализован и исследован алгоритм прогнозирования структурной сложности нейронной сети по сложности выборки.

- Предложены четыре критерия структурной сложности универсальной модели нейросети.
- Предложены критерии геометрической и размерной сложности выборки.
- Проведена серия численных экспериментов на модельных данных. В результате получено, что пара Comp_4 и DimComp являются хорошо корреллирующими между собой. Полученная зависимость позволяет определять структурную сложность по быстровычислимой размерной сложности и тем самым позволяет значительно сократить перебор гиперпараметров нейросетей.

Список литературы

[1] *Kwapisz J.R.* Activity Recognition using Cell Phone Accelerometers// ACM SIGKDD Explorations Newsletter , 2010, 12, Pp. 74–82.

[2] *Hinton G. E., Salakhutdinov R. R* Reducing the dimensionality of data with neural networks// Science, Vol. 313. No. 5786, Pp. 504–507.

[3] *Bengio Y., Lamblin, P., Popovici D., Larochelle H.* Greedy Layer-Wise Training of Deep Networks// Advances in Neural Information Processing Systems, Vol. 19, 2006, Pp. 153–160.

[4] *Widrow B., Stearns, S.D.* Adaptive Signal Processing// Prentice-Hall, 1985.

[5] *Barron A.R.* Approximation and Estimation Bounds for Artificial Neural Networks// Machine Learning, 1994, Vol. 14, Issue 1, Pp. 115–133.

[6] *Hinton G.E.* Connectionist learning procedures// Artificial Intelligence, Vol. 40, Pp. 185–234.

[7] *Hassibi B., Stork D.G.* Second Order Derivatives for Network Pruning: Optimal Brain Surgeon// Advances in Neural Information Processing Systems, Vol. 5, Pp. 164–171.

[8] *Vapnik V., Chervonenkis A.* On the uniform convergence of relative frequencies of events to their probabilities// Theoretical Probabibility and its Applicactions, Pp. 264–280.

[9] *Koiran P., Sontag E.D.* Neural Networks with Quadratic VC Dimension// Journal of Computer and System Sciences, Vol. 54, Issue 1, Pp. 190–198.