Информационные процессы, Том 15, № 3, 2015, стр. 279–297

МАТЕМАТИЧЕСКИЕ МОДЕЛИ, ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ

Влияние инициализации параметров на время обучения и точность нелинейной регрессионной модели[1]

Е.В. Бурнаев*, П.Д. Ерофеев*

Институт проблем передачи информации, Российская академия наук, Москва, Россия
Поступила в редколлегию 26.06.2015

Аннотация—Одной из задач, возникающих при построении нелинейной регрессионной модели, является правильная (в определенном смысле) инициализации её параметров. В данной работе проводится сравнение некоторых широко распространенных методов и нескольких новых подходов к инициализации параметров регрессионной модели, представляющей из себя разложение по словарю параметрических функций специального вида (сигмоидов). В работе предложен общий детерминированный подход к инициализации, позволивший добиться повторяемости результатов, сокращения времени обучения и в некоторых случаях повышения точности регрессии; разработаны два новых алгоритма (на основе кусочно-линейной аппроксимации и на основе локальных особенностей аппроксимируемой зависимости) в рамках предложенного подхода; разработан рандомизированный алгоритм инициализации для эффективной аппроксимации зависимостей высокой размерности (сферическая инициализация); усовершенствован классический метод SCAWI (за счет расположения центров сигмоидов в точках исходной выборки), что позволило повысить точность регрессии на специфических классах зависимостей (гладкие функции и разрывные функции со множеством особенностей в области определения) при использования для обучения алгоритма RProp; проведено сравнение классических и новых предложенных методов иницализации, позволившее выявить наиболее эффективные алгоритмы.

Ключевые слова: нелинейная регрессия, аппроксимация, нейронные сети, инициализация параметров, алгоритм SCAWI, алгоритм RProp, алгоритм обратного распространения ошибки.

1. ВВЕДЕНИЕ

Построение суррогатных моделей является одной из актуальнейших задач современного анализа данных [1]. Возможность быстрой и точной аппроксимации сложных нелинейных зависимостей может заменить или существенно сократить объемы сложных и/или дорогостоящих инженерных экспериментов и востребована в разнообразных областях современной прикладной науки: машиностроение, биоинформатика, анализ экономических связей и т.д. В последнее время было разработано множество алгоритмов нелинейной регрессии [2], включая методы на основе нейронных сетей (Artifiacial Neural Networks), кригинга (Kriging), машины опорных векторов (Support Vector Machine), k-ближайших соседей (k-Nearest Neighbours) и др. Эти алгоритмы оказываются достаточно эффективными на некоторых задачах, однако, часть вопросов, связанных с построением подобных нелинейных моделей, остается открытой. В первую очередь большинство нелинейных моделей зависят от некоторых параметров, значения которых подстраиваются в процессе так называемого “обучения”. Как показывает практика и время построения и точность аппроксимации конечной модели существенно зависят от

[1] Исследование выполнено в ИППИ РАН исключительно за счет гранта Российского научного фонда (проект № 14-50-00150)

выбора начальных значений этих параметров [3]. Вторым открытым вопросом несомненно является сам алгоритм подстройки параметров, то есть "обучения" модели. Однако, этот вопрос значительно лучше освещён в литературе.

Большинство работ по инициализации посвящены рандомизированным алгоритмам [4–8]. Такие алгоритмы, несмотря на явные преимущества (простота реализации и небольшие вычислительные затраты), обладают существенным недостатком: качество конечной аппроксимации и время обучения на одних и тех же данных могут значительно отличаться при разных запусках алгоритма инициализации. В некоторых задачах подобное поведение алгоритма аппроксимации является недопустимым и требуется исключить возможность случайного появления аппроксимации заведомо низкого качества или ограничить время обучения модели. В ряде работ делается попытка при инициализации использовать свойства исходной выборки данных с целью сокращения времени обучения или уменьшения вариативности конечной аппроксимации. Однако, предложенные в этих работах алгоритмы либо не подходят для данных высокой размерностей в силу больших вычислительных затрат [9], либо используют некоторые априорные свойства [10], которые не известны в общем случае.

В данной работе рассматривается аппроксимация на основе разложения по словарю параметрических функций специального вида (сигмоидов) [11–13]. Предложены подходы к построению детерминированных алгоритмов, максимально использующих информацию, содержащуюся в исходной выборке, а также приведены основные алгоритмы рандомизированной инициализации. Проведено сравнение эффективности всех описанных алгоритмов по качеству конечной аппроксимации и времени подстройки параметров для двух вариантов алгоритма обучения. Основной целью исследования являлось определение влияния инициализации параметров на качество конечной модели и время её обучения, и определение наиболее эффективных алгоритмов выбора начальных значений параметров.

Работа состоит из трех частей. В первой части описана общая постановка задачи построения нелинейной регрессии, а также приведено краткое описание метода аппроксимации на основе разложения по словарю параметрических функций. Также в этой части описана задача инициализации и приведены два основных алгоритма подстройки параметров регрессионной модели. Во второй части приведена классификация существующих алгоритмов инициализации с кратким их описанием и предложены три новых алгоритма:

1. Сферическая инициализация параметров;
2. Инициализация на основе кусочно-линейного приближения;
3. Инициализация на основе локальных особенностей аппроксимируемой зависимости.

Третья часть посвящена описанию и анализу результатов численных экспериментов с различными алгоритмами инициализации. Там же приведены выводы о влиянии инициализации на качество конечной аппроксимации и время обучения модели.

2. ПОСТАНОВКА ЗАДАЧИ АППРОКСИМАЦИИ НЕЛИНЕЙНОЙ ЗАВИСИМОСТИ

Пусть задана некоторая выборка данных[1]

$$S = \{(\mathbf{X}_i, y_i), i = 1, \ldots, N\},\ \mathbf{X}_i \in \mathbb{R}^n,\ y_i \in \mathbb{R}^1,$$

порожденная неизвестной функцией $y = f(\mathbf{X})$. Необходимо построить функцию $\hat{f}(\mathbf{X})$, которая будет близка к исходной функции $f(\mathbf{X})$ в смысле некоторой нормы (обычно это среднеквадратичная ошибка). В общем случае функция $f(\mathbf{X})$ нелинейная.

[1] Предполагается, что значения функции заданы только в конечном наборе точек из неизвестного распределения и получение значения аппроксимируемой зависимости в других точках невозможно по причине большой стоимости проведения соответствующих экспериментов или значительных временных затрат.

Существует множество подходов к решению поставленной задачи. Наиболее известные из них используют [2] нейронные сети, кригинг, машины опорных векторов, метод на основе k-ближайших соседей и др. В данной работе рассмотрен алгоритм аппроксимации нелинейных зависимостей на основе разложения по словарю параметрических функций специального вида (сигмоидов), описанный в следующем подразделе.

2.1. Аппроксимация нелинейной зависимости на основе разложения по словарю параметрических функций

Мы будем рассматривать аппроксимирующие функции вида

$$\hat{f}(\mathbf{X}, \mathbf{W}) = \sum_{j=1}^{p} V_j \sigma(\mathbf{X} \times \mathbf{W}_j^T + b_j) + V_0,\ \mathbf{W}_j \in \mathbb{R}^n, \tag{1}$$

представляющие собой разложение по словарю параметрических функций [11] (нейронная сеть с двумя слоями). Здесь в качестве функции $\sigma(\cdot)$ выступают функции специального вида — сигмоиды (гиперболический тангенс). Настраиваемые параметры модели: p, V_0, V_j, $\mathbf{W}_j$ и b_j, $(j = 1, \ldots, p)$. Веса $V_j\,(j = 0, \ldots, p)$ могут быть однозначно определены по остальным параметрам модели с помощью решения линейной регрессионной задачи [14].

В дальнейшем будем считать, что выборка S поделена на два множества

$$S = S_{train} \cup S_{val}, \tag{2}$$

где выборка S_{train} мощностью N_{train} используется непосредственно для обучения, а S_{val} мощностью N_{val} — для валидации (оценки среднеквадратичной ошибки).

Инициализация модели

Таким образом, в контексте рассматриваемой проблемы инициализации модели необходимо решать сразу две задачи:

1. Подбор количества p функций для словаря;
2. Инициализация параметров этих функций. При этом порядок решения этих задач может быть разным.

К эффективному алгоритму инициализации предъявляются следующие требования:

– Качество построенной модели. Конечная модель (после обучения) должна быть приемлемой в смысле величины средней ошибки аппроксимации и величины 95% квантили абсолютной ошибки аппроксимации[2];
– Время обучения должно быть минимальным, то есть начальные значения параметров должны быть максимально близкими к оптимальным, что сократит время, требуемое на обучение модели (подстройку параметров);
– Повторяемость результатов. Разброс ошибки аппроксимации на данных одного и того же происхождения должен быть минимальным.

Таким образом, необходимо построить алгоритмы инициализации, отвечающие поставленным требованиям, и провести сравнение эффективности этих алгоритмов с уже существующими подходами.

[2] Величина максимальной ошибки не является репрезентативной из-за случайной природы, поэтому более правильно рассматривать, например, 95% квантиль ошибки, которая является более стабильной характеристикой.

Подстройка параметров

Для подстройки параметров модели (1) обычно используются два алгоритма обучения [15]: алгоритм эластичного обратного распространения ошибки (Resilient backpropagation, RProp) и алгоритм Левенберга-Марквардта (Levenberg-Marquardt, LM).

Алгоритм RProp в отличии от стандартного алгоритма обучения моделей типа (1) — алгоритма обратного распространения ошибки, использует только знаки частных производных для подстройки параметров (весовых коэффициентов), а само обучение происходит "по эпохам", то есть коррекция весов проводится после предъявления всех примеров из обучающей выборки. Для определения величины коррекции используется следующее правило

$$\widetilde{\Delta}_i^{(t)} = \begin{cases} \eta^+ \Delta_i^{(t)}, & \frac{\partial E^{(t)}}{\partial w_i} \frac{\partial E^{(t-1)}}{\partial w_i} > 0, \\ \eta^- \Delta_i^{(t)}, & \frac{\partial E^{(t)}}{\partial w_i} \frac{\partial E^{(t-1)}}{\partial w_i} < 0, \end{cases}$$

где $0 < \eta^- < 1 < \eta^+$, $E = \frac{1}{2} \sum_{k=1}^{N} \left(y_k - \hat{f}\left(\mathbf{X}_k, \mathbf{W}\right) \right)^2$ — среднеквадратичная ошибка аппроксимации.

Для вычисления значения коррекции весов используется правило

$$\Delta w_i(t) = \begin{cases} -\widetilde{\Delta}_i, & \frac{\partial E^{(t)}}{\partial w_i} > 0, \\ +\widetilde{\Delta}_i, & \frac{\partial E^{(t)}}{\partial w_i} < 0, \\ 0, & \frac{\partial E^{(t)}}{\partial w_i} = 0. \end{cases}$$

Затем веса подстраиваются согласно формуле $w_i(t+1) = w_i(t) + \Delta_i(t)$.

Алгоритм LM основан на следующей идее: чтобы минимизировать ошибку аппроксимации E, нужно найти такие параметры модели, при которых производные $\frac{\partial E}{\partial w_i}$ были бы минимальными. Рассмотрим линейную аппроксимацию функции $f(\mathbf{X}, \mathbf{W})$ вблизи точки $\mathbf{W}(t)$, соответствующей значениям параметров модели на t-ой итерации алгоритма LM

$$\tilde{f}(\mathbf{X}, \mathbf{W}) = f(\mathbf{X}, \mathbf{W}(t)) + \sum_i (w_i - w_i(t)) \frac{\partial f(\mathbf{X}, \mathbf{W}(t))}{\partial w_i}. \tag{3}$$

Обозначим $J = \left\{ \frac{\partial f(\mathbf{X}, \mathbf{W}(t))}{\partial \mathbf{W}} \right\} \Big|_{\mathbf{W}=\mathbf{W}(t)}$ – якобиан функции $f(\mathbf{X}, \mathbf{W})$, $\Delta w_i = w_i - w_i(t)$. Подставляя выражение (3) в формулу для подсчета ошибки аппроксимации и дифференцируя по $\mathbf{\Delta w}$, получаем систему

$$J^T J \mathbf{\Delta w} = J^T (\mathbf{Y} - f(\mathbf{X}, \mathbf{W})).$$

Данная система линейных уравнений может быть разрешена относительно величины $\mathbf{\Delta w}$, которая затем используется для адаптации вектора параметров в ходе процесса обучения регрессионной модели.

3. ПОДХОДЫ К ИНИЦИАЛИЗАЦИИ НЕЛИНЕЙНЫХ РЕГРЕССИОННЫХ МОДЕЛЕЙ

В данной работе в рамках решения поставленной задачи рассмотрены два принципиально разных подхода:

- Рандомизированные методы инициализации, широко используемые в подобных задачах [4, 16–18], предполагают случайную инициализацию параметров модели из некоторого распределения;
- Детерминированные методы инициализации, учитывающие характерные особенности аппроксимируемой выборки, являются более предпочтительными в смысле повторяемости результатов.

Рандомизированные алгоритмы имеют следующие преимущества: простота реализации и незначительные вычислительные затраты. Благодаря этому они получили широкое распространение [16]. Однако, если при использовании рандомизированной инициализации качество конечной модели и время, затраченное на обучение, оказываются приемлемыми, то получение повторяемости результатов даже на одних и тех же данных не представляется возможным[3].

3.1. Рандомизированная инициализация параметров

Широкое распространение для инициализации моделей типа (1) получили алгоритмы рандомизированной инициализации. В этом разделе будут рассмотрены некоторые наиболее известные из них.

Инициализация Нгуена-Видроу

Самым распространенным способом инициализации нелинейных моделей типа (1) является рандомизированный алгоритм NW, предложенный Нгуеном и Видроу [17]. Параметрам $\mathbf{W}_j$ и b_j присваиваются начальные значения так, чтобы активные области соответствующих сигмоидов были распределены примерно равномерно в пространстве регрессоров [18]. Таким образом каждый элемент матрицы параметров $\mathbf{W}$ инициализируется числом из равномерного распределения[4]:

$$\mathbf{W}_j \sim U[-I,\, I]^n, \tag{4}$$

где $I = p^{\frac{1}{N_{train}}}$. Компоненты вектора $\mathbf{b}$ также выбираются из равномерного распределения $U[-I, I]$.

Инициализация Драго и Риделла

Подход к инициализации параметров модели, используемый Драго и Риделла[5] [4], напоминает алгоритм NW, но с другой границей значений:

$$\mathbf{W}_j \sim U[-I,\, I]^n, \tag{5}$$

где $I = \frac{1.3}{\sqrt{1+N\nu^2}}$, ν^2 — среднее значение квадратов входных переменных:

$$\nu^2 = \frac{1}{n\,N}\sum_{i=1}^{N}\sum_{j=1}^{n} x_{ij}^2.$$

Такая инициализация позволяет гарантировать, что значения аргументов сигмоидов будут находится в области ненасыщения сигмоида, при этом оказываясь значительно отличными от нуля. Компоненты вектора $\mathbf{b}$ выбираются аналогично предыдущему методу из равномерного распределения.

Сферическая инициализация

Для многомерных пространств покомпонентная случайная генерация векторов параметров $\mathbf{W}_j$ приводит к их кластеризации, тем самым порождая кластеризацию направлений сигмоидов. Существуют теоретические результаты [19], согласно которым наилучшая аппроксимация получается в случае равномерного распределения направлений по сфере.

[3] Как показали численные эксперименты, средняя ошибка аппроксимации на одних и тех же данных может отличаться на несколько порядков для одинаковых моделей с разными начальными параметрами.

[4] Здесь и далее будем считать, что в ходе предварительной обработки данных пространство регрессоров ограничено гиперкубом $[-1, 1]^n$.

[5] Алгоритм также известен под названием SCAWI (Statistically Controlled Activation Weight Initialization).

Представим параметры модели (1) в виде $\mathbf{W}_j = R_j\mathbf{S}_j$, где $\mathbf{S}_j$ — случайный вектор, расположенный на единичной сфере, а R_j — некоторый радиус. Предлагается использовать следующую схему сферической инициализации параметров модели. На первом этапе получаем значения углов $\phi_k\,(k = 1, \ldots, n-1)$ из случайного равномерного распределения $U[-\pi, \pi]^{n-1}$ (используя для этого равномерное заполнение пространства с помощью оптимизированных латинских гиперкубов), а затем переходим в декартовы координаты:

$$\begin{gathered} w_{j,1} = R\cos(\phi_1), \\ w_{j,2} = R\sin(\phi_1)\cos(\phi_2), \\ \ldots \\ w_{j,n-1} = R\sin(\phi_1)\cdot\ldots\cdot\sin(\phi_{n-2})\cos(\phi_{n-1}), \\ w_{j,n} = R\sin(\phi_1)\cdot\ldots\cdot\sin(\phi_{n-2})\sin(\phi_{n-1}). \end{gathered}$$

Радиус по аналогии с алгоритмом SCAWI предлагается выбирать равным $R = \frac{1.3}{\sqrt{1+N\nu^2}}$. Вектора, полученные таким образом, оказываются распределенными более равномерно в многомерных пространствах. Компоненты вектора $\mathbf{b}$ выбираются аналогично методу Нгуена-Видроу из равномерного распределения.

Подбор числа сигмоидов

Приведенные алгоритмы не позволяют ответить на вопрос, сколько сигмоидов p необходимо для построения аппроксимации. Предлагается два варианта решения этой проблемы:

- подбор числа сигмоидов по сетке (по минимальной ошибке на валидационном множестве (2));
- жадный отбор сигмоидов [20] по критерию минимальной ошибки (до тех пор пока ошибка аппроксимации на валидационном множестве не начнет возрастать) или по критерию наибольшей корреляции.

Один из возможных методов подобного рода описан в общем виде в алгоритме 1.

Алгоритм 1. Жадный отбор регрессоров

Цель: Аппроксимировать решение задачи $\min_{\mathbf{x}} \|\mathbf{x}\|_0$ при условии $\mathbf{A}\mathbf{x} = \mathbf{d}$.
Параметры: Заданы матрица $\mathbf{A}$, вектор $\mathbf{d}$ и начальный порог ошибки ϵ_0.
Инициализация: Присвоить $k = 0$ и задать
- начальное решение $\mathbf{x}^0 = \mathbf{0}$,
- начальные остатки $\mathbf{r}^0 = \mathbf{d} - \mathbf{A}\mathbf{x}^0 = \mathbf{d}$,
- начальный носитель решения $S_0 = \mathrm{supp}\{\mathbf{x}_0\} = \emptyset$.

Главная итерация: Увеличить k на 1 и выполнить следующие шаги:
- Вычислить ошибки $\epsilon(j) = \min_{z_j} \|\mathbf{a}_j z_j - \mathbf{r}^{k-1}\|_2^2$ для всех j, используя оптимальный выбор параметра $z_j^* = \mathbf{a}_j^T\mathbf{r}^{k-1}/\|\mathbf{a}_j\|_2^2$.
- Найти j_0: $\forall j \notin S^{k-1}, \epsilon(j_0) \le \epsilon(j)$ и обновить $S^k = S^{k-1} \cup \{j_0\}$.
- Вычислить $\mathbf{x}^k = \arg\min \|\mathbf{A}\mathbf{x} - \mathbf{d}\|_2^2$ при условии $supp\{\mathbf{x}\} = S^k$.
- Вычислить $\mathbf{r}^k = \mathbf{d} - \mathbf{A}\mathbf{x}^k$.
- Если $\|\mathbf{r}^k\|_2^2 < \epsilon_0$, остановить алгоритм. В противном случае перейти к следующей итерации.

Выход: Решение $\mathbf{x}^k$, полученное после k-ой итерации.

При использовании алгоритма 1 роль вектора $\mathbf{d}$ играют значения y_i, роль вектора $\mathbf{x}$ – значения V_j, а матрица $\mathbf{A}$ формируется из значений сигмоидов на точках обучающей выборки.

Выбор начального положения центров сигмоидов

Во всех описанных ранее алгоритмах центры сигмоидов выбираются случайно (из-за случайности выбора вектора $\mathbf{b}$). Однако в силу того, что аппроксимируемая функция задана в ограниченном числе точек, разумно предположить, что мы сможем построить эффективное приближение функции только вблизи этих точек. Ничего неизвестно о поведении искомой функции в областях, где не заданы точки выборки, поэтому правильно было бы располагать активные области сигмоидов рядом с точками выборки. Таким образом, предлагается ставить начальные центры сигмоидов так, чтобы линия нулевого значения проходила через точки выборки. Это легко сделать, если переписать функцию активации в следующем виде:

$$\sigma(\mathbf{X}\mathbf{W}_j^T + b_j) = \sigma((\mathbf{X} - \mathbf{X}_0)\mathbf{W}_j^T), \tag{6}$$

где $b_j = \mathbf{X}_0\mathbf{W}_j^T$, а $\mathbf{X}_0$ – некоторая точка выборки.

3.2. Детерминированная инициализация параметров

Рандомизированные алгоритмы инициализации имеют существенный недостаток: качество и время обучения могут существенно отличаться для двух разных запусков на одних и тех же данных. В этом разделе приведено описание разработанных детерминированных алгоритмов инициализации, позволяющих решить эту проблему.

Детерминированная инициализация на основе латинских гиперкубов

Самый простой способ перехода к детерминированный инициализации — замена случайного равномерного распределения в рандомизированных алгоритмах на некоторое фиксированное равномерное разбиение пространства, например, латинские гиперкубы [21]. Такой подход рождает сразу три детерминированных алгоритма: для инициализации Нгуена-Видроу, Драго-Риделла и сферической инициализации. Однако единственное отличие этих методов от рандомизированных состоит в том, что они дают одинаковые результаты на одних и тех же данных. Поэтому будем считать их псевдодетерминированными в отличии от описанных ниже алгоритмов, которые при инициализации используют информацию, заложенную в исходных данных.

Инициализация на основе кусочно-линейной аппроксимации

Более сложный подход предполагает построение начальной аппроксимации заданной выборки с помощью кусочно-линейных сигмоидов вида[6]:

$$\sigma_h(\widetilde{\mathbf{X}}, W_+, W_-, \mathbf{W}_{lin}) = \begin{cases} W_+, & \widetilde{\mathbf{X}}\mathbf{W}_{lin}^T > W_+, \\ \widetilde{\mathbf{X}}\mathbf{W}_{lin}^T, & \widetilde{\mathbf{X}}\mathbf{W}_{lin}^T \in [W_-, W_+], \\ W_-, & \widetilde{\mathbf{X}}\mathbf{W}_{lin}^T < W_-. \end{cases} \tag{7}$$

Для этого минимизируется ошибка аппроксимации [22]

$$E = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \sigma_h\left(\widetilde{\mathbf{X}}_i, W_+, W_-, \mathbf{W}_{lin}\right)\right)^2.$$

[6] $\widetilde{\mathbf{X}} = [1\ \mathbf{X}]$ – расширенный вектор признаков

Выражение для ошибки может быть разбито на три части за счет разбиения обучающего множества на три подмножества:

$$E = \frac{1}{2}\sum_{i\in S_-}(y_i - W_-)^2 + \frac{1}{2}\sum_{i\in S_+}(y_i - W_+)^2 + \frac{1}{2}\sum_{i\in S_{lin}}\left(y_i - \widetilde{\mathbf{X}}\mathbf{W}_{lin}^T\right)^2,$$

где $S_- = \{(\mathbf{X}_i, y_i) \in S : \widetilde{\mathbf{X}}\mathbf{W}_{lin}^T < W_-\}$, аналогичным образом определяются S_+ и S_{lin}.

При дальнейшем упрощении полученного выражения получаем задачу квадратичного программирования:

$$\min \frac{1}{2}\mathbf{W}^T R_\lambda \mathbf{W} - \mathbf{W}^T\mathbf{r} \qquad \text{при условии} \qquad A\mathbf{W} \le \mathbf{0}, \tag{8}$$

$$R_\lambda = \begin{bmatrix} \sum_{i\in S_{lin}} \widetilde{\mathbf{X}}_i\widetilde{\mathbf{X}}_i^T & 0 & 0 \\ 0 & N_+ & 0 \\ 0 & 0 & N_- \end{bmatrix}; \mathbf{r} = \begin{bmatrix} \sum_{i\in S_{lin}} \widetilde{\mathbf{X}}_i y_i \\ s_y^+ \\ s_y^- \end{bmatrix}; \mathbf{W} = \begin{bmatrix} \mathbf{W}_{lin} \\ W_+ \\ W_- \end{bmatrix},$$

где N_+ – число точек выборки в области S_+, N_- – число точек в области S_-, $s_y^+ = \sum_{S_+} y_i$, $s_y^- = \sum_{S_-} y_i$, матрица

$$A = \begin{bmatrix} A_+ \\ A_{l+} \\ A_{l-} \\ A_- \end{bmatrix}$$

задает ограничения, а ее строки A_+, A_{l+}, A_{l-}, A_- выглядят соответственно следующим образом:

$$A_+^T = [-\widetilde{\mathbf{X}}_i^T\ 1\ 0],\ \mathbf{X}_i \in S_+,$$
$$A_{l+}^T = [\widetilde{\mathbf{X}}_i^T\ -1\ 0], \mathbf{X}_i \in S_{lin},$$
$$A_{l-}^T = [-\widetilde{\mathbf{X}}_i^T\ 0\ 1], \mathbf{X}_i \in S_{lin},$$
$$A_-^T = [\widetilde{\mathbf{X}}_i^T\ 0\ -1],\ \mathbf{X}_i \in S_-.$$

Решая задачу (8) (см. [22]) мы получаем один кусочно-линейный сигмоид вида (7), приближающий заданную выборку. Повторяя последовательно эту операцию применительно к остаткам, полученным на каждом предыдущем шаге, мы получаем кусочно-линейную аппроксимацию начальной выборки. Затем необходимо выполнить переход к непрерывным сигмоидам. Предлагается осуществлять этот переход так, чтобы производные кусочно-линейного и непрерывного сигмоидов в нуле совпадали. Формальное описание метода приведено в алгоритме 2.

Алгоритм 2. Инициализация на основе кусочно-линейного приближения

Вход: выборка $S = \{(\mathbf{X}_i, y_i)\}_{i=1}^N$, число сигмоидов p.
Выход: матрица параметров $\mathbf{W}$ и вектор $\mathbf{b}$.
Шаг 1. Кусочно-линейная аппроксимация:

- инициализируем остатки $\mathbf{R} = [y_1, \ldots, y_N]^T$;
- выполняем k-ю итерацию
  - решая задачу (8) для приближения текущих остатков $\mathbf{R}$, получаем значения параметров $\mathbf{W}_{lin}^{(j)}, W_+^{(j)}$ и $W_-^{(j)}$;
  - вычисляем новые остатки

$$\mathbf{R} = \mathbf{R} - [\sigma_h(\mathbf{X}_1, W_+^{(j)}, W_-^{(j)}, \mathbf{W}_{lin}^{(j)}), \ldots, \sigma_h(\mathbf{X}_N, W_+^{(j)}, W_-^{(j)}, \mathbf{W}_{lin}^{(j)})]^T;$$

- вычисляем норму полученных остатков $R_k = \|\mathbf{R}\|^2$

– итерируем пока (норма R_k не станет увеличиваться 5 раз подряд) и $(k < p)$

Шаг 2. Переход к непрерывным сгмоидам:

– для каждого кусочно линейного сигмоида определяем: $\mathbf{W}_j = [\mathbf{W}^{(j)}_{2,lin}, \dots \mathbf{W}^{(j)}_{n+1,lin}]$, $\alpha = \frac{1}{2}(W_+ - W_-)$, $\beta = \frac{1}{2}(W_+ - W_-)$, $b_j = \left(\mathbf{W}^{(j)}_{1,lin} - \beta\right)/\alpha$;

– составляем матрицу параметров $\mathbf{W} = [\mathbf{W}_1^T, \dots, \mathbf{W}_k^T]^T$.

Инициализация на основе локальных особенностей исходных данных

В основе этого алгоритма лежит идея о том, что исходно сигмоиды необходимо располагать в тех областях, где аппроксимируемая функция имеет локальные особенности.

Алгоритм состоит из двух основных частей.

– Локальная аппроксимация. На этом шаге для каждой точки $\mathbf{X}_i \in S_{train}$ исходной обучающей выборки (2) строится локальная аппроксимация с помощью одного сигмоида. Для этого формируется матрица весов $\mathbf{P} = [p_{jk}]_{j,k=1}^N$

$$p_{jk} = \begin{cases} \dfrac{\exp\left(-\sum_{m=1}^n \frac{(x_{km}-x_{jm})^2}{h_m^2}\right)}{\sum_{l=1}^N \exp\left(-\sum_{m=1}^n \frac{(x_{km}-x_{lm})^2}{h_m^2}\right)}, & j \neq k \\ 0, & j = k, \end{cases} \tag{9}$$

где h_m — ширина ядра, которая может быть оценена по классической формуле Боумана-Аззалини [23]:

$$h_m = s_m \left\{\frac{4}{(n+2)\cdot N}\right\}^{\frac{1}{n+4}}, \tag{10}$$

в которой s_m — оценка стандартного отклонения по m-ой компоненте входных векторов обучающей выборки. Затем решается задача линейной аппроксимации:

$$\min_{\mathbf{W}_i} \sum_{j=1}^N p_{ji}^2 \left\| \sigma^{-1}\left(\frac{y_j}{V_i}\right) - (\mathbf{X}_j - \mathbf{X}_i)\mathbf{W}_i^T \right\|_2^2, \tag{11}$$

где значение V_i подбирается по равномерной сетке. Таким образом, в каждую точку обучающей выборки ставится свой сигмоид, описывающий локальные особенности аппроксимируемой функции вблизи этой точки.

– Отбор сигмоидов. После того, как построены все сигмоиды, из них необходимо выбрать наиболее коррелированные с заданными целевыми переменными. Если задан параметр p, то путем жадного набора, предложенного в разделе 3.1, формируется словарь из p сигмоидов. Если этот параметр не задан, то подбирается оптимальное (по ошибке на валидационном множестве) число сигмоидов для начальной аппроксимации.

Формальное описание метода приведено в алгоритме 3. Предложенный алгоритм имеет недостатки. При построении очередного сигмоида не учитываются уже существующие в словаре. Также, как показали опыты, качество конечной аппроксимации существенно зависит от ширины ядра h_m (см. рис. 1): небольшие изменения ширины ядра h_m могут привести к значительному изменению качества аппроксимации.

Алгоритм 3. Инициализация на основе локальных особенностей аппроксимируемой зависимости

Вход: выборка $S = \{(\mathbf{X}_i, y_i)\}_{i=1}^N$, число сигмоидов p (опционально).
Выход: матрица параметров $\mathbf{W}$ и вектор $\mathbf{b}$.
Шаг 1. Локальная аппроксимация:

- задается ширина ядра h_m в соответствии с (10);
- строится матрица весов $\mathbf{P}$ (9) и для каждой точки выборки $\mathbf{X}_i$ решается задача (11);
- в итоге получаем словарь из N сигмоидов с параметрами $\mathbf{W}_i$ и $b_i = -\mathbf{X}_i\mathbf{W}_i^T$, $i = 1, \ldots, N$;

Шаг 2. Отбор сигмоидов:

- осуществляется жадный набор сигмоидов из множества сигмоидов, полученных на первом шаге в соответствии с алгоритмом, описанным в разделе 3.1;
- из отобранных сигмоидов составляется матрица параметров $\mathbf{W}$ и вектор $\mathbf{b}$.

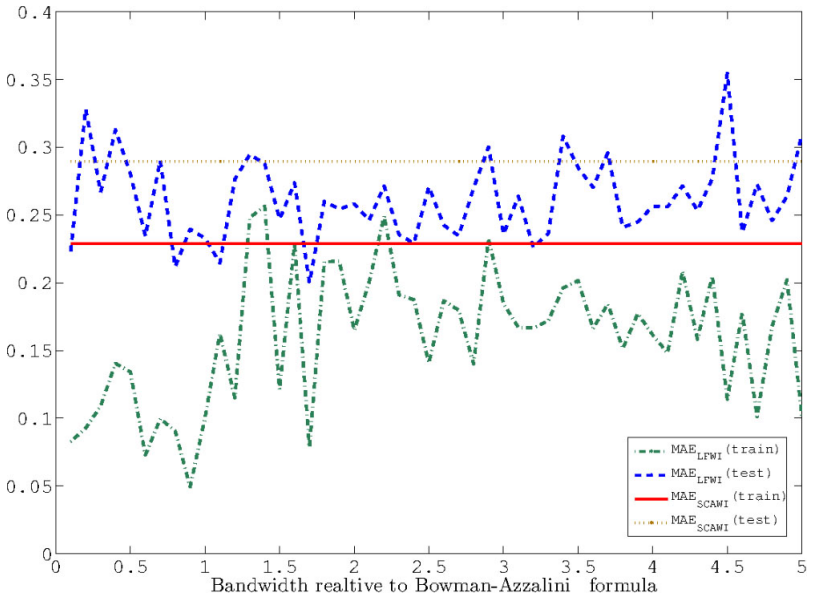

Рис. 1. Зависимость качества конечной аппроксимации от ширины ядра для алгоритма инициализации на основе локальных особенностей функции (алгоритм обучения RProp).

4. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

Для сравнения эффективности методов, описанных в предыдущих разделах, были проведены численные эксперименты на искусственных данных. Алгоритмы инициализации обозначены следующим образом:

- NWWI – алгоритм Нгуена-Видроу;
- SCAWI(1) – модифицированный алгоритм Драго-Риделла с расстановкой центров сигмоидов в точках выборки (см. раздел 3.1);
- SCAWI(2) – оригинальный алгоритм Драго Риделла;
- SWI – сферическая инициализация весов (см. раздел 3.1);
- PLWI – инициализация на основе приближения кусочно-линейными сигмоидами (см. раздел 3.2);
- LFWI – инициализация на основе локальных особенностей аппроксимируемой зависимости (см. раздел 3.2).

4.1. Постановка экспериментов

Для сравнения выбранных моделей были использованы данные, сгенерированные с помощью стандартных искусственных функций. В двумерном случае ($x_i \in [-1,1], i = 1,2$) использовались следующие функции:

$$f_1(x_1,x_2) = \frac{\sin^2\left(\sum_{i=1}^2 (x_i+0.6)^2 - 0.3\right)}{\tanh\left[\sum_{i=1}^2 \left((x_i+0.6)^2 - 0.3\right)^2 + 0.4\right]};$$

$$f_2(x_1,x_2) = \frac{x_1+x_2}{1+4\left(x_1^2+x_2^2\right)};$$

$$f_3(x_1,x_2) = \sum_{i=1}^2 x_i + 1\cdot\left(\sum_{i=1}^2 x_i^2 < 0.25\right) - 2\cdot\left(\sum_{i=1}^2 (x_i-0.7)^2 < 1\right);$$

$$f_4(x_1,x_2) = ((6x_1)^2 + 6x_2 - 11)^2 + (6x_1 + (6x_2)^2 - 7)^2;$$

$$f_5(x_1,x_2) = \exp\left[-10\left(\frac{x_1}{2}+\frac{1}{4}\right)^2 + \left(\frac{x_2}{2}+\frac{1}{4}\right)^2\right] + 2\exp\left[-20\left(\frac{x_1}{2}-\frac{1}{4}\right)^2 + \left(\frac{x_2}{2}\right)\right];$$

$$f_6(x_1,x_2) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1-\frac{1}{8\pi}\right)\cos(x_1) + 10;$$

$$f_7(x_1,x_2) = 2 + 0.01\left(x_2 - x_1^2\right)^2 + (1-x_1)^2 + 2(2-x_2)^2 + 7\sin(0.5x_1)\sin(0.7x_1x_2);$$

$$f_8(x_1,x_2) = \sin(x_1)\sin\left(x_1^2/\pi\right) + \sin(x_2)\sin\left(2x_2^2/\pi\right);$$

$$f_9(x_1,x_2) = \left(\sin(x_1)\sin\left(x_1^2/\pi\right) + \sin(x_2)\sin\left(2x_2^2/\pi\right)\right)^2;$$

$$f_{10}(x_1,x_2) = 100\left(x_2 - x_1^2\right)^2 + (1-x_1)^2;$$

$$f_{11}(x_1,x_2) = x_1^2 + 2x_2^2;$$

$$f_{12}(x_1,x_2) = 20 + (x_1 - 10\cos(2\pi x_1)) + (x_2 - 10\cos(2\pi x_2));$$

$$f_{13}(x_1,x_2) = x_1\sin\left(\sqrt{|x_1|}\right) - x_2\sin\left(\sqrt{|x_2|}\right);$$

$$f_{14}(x_1,x_2) = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2;$$

$$f_{15}(x_1,x_2) = \min(x_1^2 + x_2^2, 2x_1^2, 2).$$

Эксперименты проводились следующим образом. Для каждой функции генерировалось 10 выборок, состоящих из 300 случайных точек в диапазоне $[-1,1]\times[-1,1]$. Каждый рандомизированный алгоритм инициализации запускался 10 раз на каждой выборке.

Для тестирования алгоритмов инициализации для аппроксимации зависимостей высокой размерности $n \gg 1$ были использованы следующие функции:

$$\phi_1(x_1,\ldots,x_n) = \sum_{i,j=1}^n \frac{\left(100\left(x_i^2 - x_j\right)^2 + (1-x_j)^2\right)^2}{4000} - \cos\left(100\left(x_i^2 - x_j\right)^2 + (1-x_j)^2\right) + 1;$$

$$\phi_2(x_1,\ldots,x_n) = \sum_{i=1}^{n-1}\left(e^{0.2}\sqrt{x_i^2 + x_{i+1}^2} + 3\left(\cos(2x_i) + \sin(2x_{i+1})\right)\right);$$

$$\phi_3(x_1,\ldots,x_n) = \sum_{i=1}^n i x_i^2.$$

В этом случае каждый рандомизированный алгоритм инициализации запускался 20 раз на каждой выборке.

4.2. Результаты

В качестве индикаторов эффективности конечной аппроксимации были выбраны четыре ошибки: средняя абсолютная ошибка, среднеквадратичная ошибка, 95% квантиль и 99% квантиль абсолютной ошибки. В качестве показателя скорости обучения было выбрано число итераций в алгоритме обучения после инициализации.

Нас интересует сравнение работы алгоритмов в среднем. Поэтому для каждого алгоритма были взяты медианы всех четырех ошибок по всем запускам для каждой функции. Конечным индикатором эффективности алгоритма будем считать десятичный логарифм отношения соответствующих медиан ошибок к медианам ошибок эталонного алгоритма[7]. Аналогично с показателями скорости обучения. Эталонным будем считать алгоритм, предложенный Нгуеном и Видроу (NWWI) в силу того, что этот алгоритм наиболее часто используется при построении нелинейных моделей вида (1).

Результаты для двумерных функций приведены в таблицах 1-8, для многомерных — в таблице 9.

В первую очередь нужно отметить, что алгоритмы псевдодетерминированной инициализации по точности конечной аппроксимации и по времени обучения модели практически не отличаются от соответствующих рандомизированных алгоритмов.

Рассмотрим результаты, полученные после обучения алгоритмом RProp (см. раздел 2.1). Алгоритмы детерминированной инициализации оказались неэффективными в этом случае (см. таблицы 1-4). При этом время обучения при использовании детерминированных алгоритмов существенно сократилось по сравнению со временем обучения при использовании рандомизированных алгоритмов. Также важно отметить, что усовершенствованный алгоритм SCAWI с расстановкой центров сигмоидов в точках выборки позволил получить небольшое, но стабильное улучшение качества аппроксимации по сравнению с оригинальным алгоритмом.

Теперь обратимся к результатам, полученным при обучении модели с помощью алгоритма LM (см. раздел 2.1). Качество модели при использовании детерминированных алгоритмов инициализации и обучения с помощью алгоритма LM во всех тестах оказалось не хуже, чем при рандомизированном подходе, а в подавляющем большинстве существенно лучше. Важно отметить, что квантили абсолютных ошибок и разброс значений ошибок заметно уменьшился при использовании детерминированных алгоритмов инициализации по сравнению с рандомизированными. При этом при использовании детерминированных алгоритмов значительно сокращается число итераций, необходимое для подстройки параметров.

Второй блок результатов относится к данным высокой размерности $n \gg 1$ (см. таблицу 9). Здесь следует отметить, что на рассмотренных многомерных искуственных функциях ошибки аппроксимации в среднем и максимальные ошибки при использовании предложенной в работе сферической инициализации оказались существенно ниже, чем при стандартных методах инициализации Нгуена Видроу (NWWI) и Драго-Риделла (SCAWI). Этот результат полностью соответствует изначальной идее сферической инициализации.

[7] Таким образом, если это значение близко к 0, то данная инициализация не отличается от эталонной, по данной характеристике; если значение имеет порядок 1, то данная инициализация имеет ошибку аппроксимации (или время обучения), которая на порядок больше, чем при эталонной инициализации; если значение близко к -1, то инициализация имеет ошибку аппроксимации (или время обучения), которая на порядок меньше, чем при эталонной инициализации.

4.3. Анализ полученных результатов

Алгоритм обучения Левенберга-Маквардта (LM) является более робастным, чем алгоритм RProp – небольшое изменение в начальных параметрах не вызывает сильных отклонений в конечной аппроксимации. Это объясняет полученные результаты. И, хотя для рандомизированных алгоритмов инициализации конечная аппроксимация при обучении с помощью LM оказалась хуже, чем при обучении с помощью RProp, результат работы LM при использовании детерминированных алгоритмов по качеству аппроксимации не уступает результатам работы RProp при использовании рандомизированных алгоритмов. Этот результат позволяет рекомендовать использование на небольших размерностях данных комбинацию детерминированной инициализации и алгоритма обучения модели LM. Такой подход приведет к повторяемости результатов на одних и тех же данных, сокращению времени обучения модели и меньшему разбросу в результатах для данных одной природы. Однако методы детерминированной инициализации не подходят для случая данных высокой размерности в силу быстрого роста вычислительных затрат при увеличении размерности.

Алгоритм сферической инициализации оказался эффективным на больших размерностях, что подтверждает теоретические выводы, сделанные в [19]. Поэтому на данных больших размерностей рекомендуется использовать именно этот алгоритм инициализации.

Таблица 1. Логарифмы отношения медиан абсолютных ошибок аппроксимации к соотвествующим ошибкам эталонной аппроксимации Нгуена-Видроу (алгоритм обучения RProp).

Функция	Алгоритмы инициализации									
	SCAWI(1)		SCAWI(2)		SWI		LFWI		PLWI	
	median	std	median	std	median	std	median	std	median	std
f1	-0.0208	-0.1081	0.02256	0.0086	-0.0111	-0.0391	0.02068	-0.177	0.08118	-0.1551
f2	0.116	0.21441	0.06193	0.21388	0.1267	0.04705	0.13861	-0.0147	0.06202	-0.1094
f3	0.09109	0.39633	-0.0658	-0.0184	0.04753	0.18232	0.27497	-0.1199	0.09572	-0.0553
f4	-0.0175	-0.1769	-0.2128	-0.4083	-0.0169	-0.222	0.52162	0.02036	-0.3823	-0.8365
f5	-0.0007	0.02255	0.01905	0.10235	0.03382	0.14909	-0.0167	0.05495	0.04865	0.02292
f6	-0.6445	-0.8995	-0.4652	-0.8479	-0.6526	-0.9331	1.50143	1.27925	1.13927	1.00636
f7	-0.1347	-0.2808	-0.0474	-0.5346	-0.0853	-0.192	0.17444	-0.2799	0.01428	-0.1054
f8	-0.3987	-0.7663	-0.547	-1.0854	-0.4998	-0.6335	1.51098	0.92349	1.05148	0.43837
f9	-0.1056	-0.0276	-0.1551	-0.1702	-0.0996	-0.125	0.40102	-0.094	0.16068	0.04361
f10	-0.3804	-0.6887	-0.2939	-0.8421	-0.483	-0.579	2.12644	2.31479	1.83525	1.81137
f11	-0.7921	-0.9875	-0.7716	-0.9557	-0.7693	-0.9978	2.51941	2.45233	1.46916	1.54023
f12	-0.0007	0.0633	0.00097	0.01697	-0.0041	-0.1346	-0.0136	-0.1548	0.01156	0.10263
f13	0.17185	0.46035	0.39619	0.28343	-0.0924	-0.0221	0.42204	-0.3316	0.22854	-0.1253
f14	-0.5147	-0.8817	-0.2396	-0.3082	-0.5885	-0.5187	1.68542	1.2602	1.45987	1.13887
f15	-0.0968	-0.0486	-0.0194	0.17395	-0.1022	0.11601	0.30249	-0.022	-0.1017	-0.1825

5. ЗАКЛЮЧЕНИЕ

В данной работе был проведен анализ влияния выбора алгоритма инициализации параметров на время обучения и качество конечной аппроксимации нелинейной регрессионной модели на основе разложения по словарю параметрических функций специального вида.

В первую очередь было выявлено заметное влияние инициализации на качество конечной аппроксимации. Было определено, что на рассмотренных классах функций (гладких и разрывных) выбор алгоритма инициализации зависит от выбора алгоритма последующей подстройки параметров (обучения модели). В случае обучения модели с помощью алгоритма RProp предпочтительным является алгоритм инциализации SCAWI с начальными центрами сигмоидов, находящимися в точках выборки. В случае обучения с помощью LM — детерминированные алгоритмы, основанные на кусочно-линейной аппроксимации или локальных особенностях исследуемой зависимости.

Таблица 2. Логарифмы отношения медиан среднеквадртичных ошибок аппроксимации к соотвествующим ошибкам эталонной аппроксимации Нгуена-Видроу (алгоритм обучения RProp).

Функция	Алгоритмы инициализации									
	SCAWI(1)		SCAWI(2)		SWI		LFWI		PLWI	
	median	std	median	std	median	std	median	std	median	std
f1	-0.028	-0.0781	0.02382	-0.0523	0.00407	-0.12	0.00176	-0.1703	0.06019	-0.1217
f2	0.14566	0.13716	0.06557	0.1256	0.13388	-0.0992	0.15215	0.04167	0.08281	-0.1006
f3	0.06749	0.41513	0.05415	-0.0463	-0.0033	0.0621	0.29013	0.12254	0.05966	0.08361
f4	-0.0179	-0.1449	-0.1937	-0.3447	-0.0245	-0.1968	0.51873	0.05224	-0.398	-0.8326
f5	0.00736	0.09639	0.00778	0.11322	0.01344	0.08356	-0.0332	0.11851	0.02239	-0.0044
f6	-0.5989	-0.7946	-0.3834	-0.7876	-0.655	-0.8638	1.59207	1.52968	1.23442	1.16905
f7	-0.1705	-0.4245	-0.0608	-0.9824	-0.1502	-0.3884	0.09734	-0.6319	-0.0447	-0.2807
f8	-0.4311	-0.7265	-0.4183	-1.1112	-0.4687	-0.6238	1.41161	0.82946	0.98601	0.38568
f9	-0.047	-0.0432	-0.0407	-0.1674	-0.058	-0.1747	0.41417	-0.0826	0.17852	-0.0434
f10	-0.4533	-0.4462	-0.413	-0.5889	-0.5687	-0.6378	2.30614	2.46974	1.83546	2.11121
f11	-0.7851	-1.0487	-0.7726	-0.788	-0.7969	-1.0836	2.62774	2.61034	1.5	1.43152
f12	-0.003	0.07606	-0.0024	0.05746	-0.0069	-0.0935	-0.0111	-0.1152	0.00528	0.11435
f13	0.1699	0.43842	0.37203	0.26853	-0.1109	-0.0135	0.3921	-0.3315	0.23209	-0.1333
f14	-0.5957	-0.8786	-0.1578	-0.2889	-0.6386	-0.5498	1.79431	1.35561	1.51197	1.18442
f15	-0.2053	-0.1867	-0.0019	-0.0308	-0.1448	0.03803	0.25436	0.22327	-0.2455	-0.3211

Таблица 3. Логарифмы отношения медиан 95% квантилей абсолютных ошибок аппроксимации к соответствующим ошибкам эталонной аппроксимации Нгуена-Видроу (алгоритм обучения RProp).

Функция	Алгоритмы инициализации									
	SCAWI(1)		SCAWI(2)		SWI		LFWI		PLWI	
	median	std	median	std	median	std	median	std	median	std
f1	-0.028	-0.0781	0.02382	-0.0523	0.00407	-0.12	0.00176	-0.1703	0.06019	-0.1217
f2	0.14566	0.13716	0.06557	0.1256	0.13388	-0.0992	0.15215	0.04167	0.08281	-0.1006
f3	0.06749	0.41513	0.05415	-0.0463	-0.0033	0.0621	0.29013	0.12254	0.05966	0.08361
f4	-0.0179	-0.1449	-0.1937	-0.3447	-0.0245	-0.1968	0.51873	0.05224	-0.398	-0.8326
f5	0.00736	0.09639	0.00778	0.11322	0.01344	0.08356	-0.0332	0.11851	0.02239	-0.0044
f6	-0.5989	-0.7946	-0.3834	-0.7876	-0.655	-0.8638	1.59207	1.52968	1.23442	1.16905
f7	-0.1705	-0.4245	-0.0608	-0.9824	-0.1502	-0.3884	0.09734	-0.6319	-0.0447	-0.2807
f8	-0.4311	-0.7265	-0.4183	-1.1112	-0.4687	-0.6238	1.41161	0.82946	0.98601	0.38568
f9	-0.047	-0.0432	-0.0407	-0.1674	-0.058	-0.1747	0.41417	-0.0826	0.17852	-0.0434
f10	-0.4533	-0.4462	-0.413	-0.5889	-0.5687	-0.6378	2.30614	2.46974	1.83546	2.11121
f11	-0.7851	-1.0487	-0.7726	-0.788	-0.7969	-1.0836	2.62774	2.61034	1.5	1.43152
f12	-0.003	0.07606	-0.0024	0.05746	-0.0069	-0.0935	-0.0111	-0.1152	0.00528	0.11435
f13	0.1699	0.43842	0.37203	0.26853	-0.1109	-0.0135	0.3921	-0.3315	0.23209	-0.1333
f14	-0.5957	-0.8786	-0.1578	-0.2889	-0.6386	-0.5498	1.79431	1.35561	1.51197	1.18442
f15	-0.2053	-0.1867	-0.0019	-0.0308	-0.1448	0.03803	0.25436	0.22327	-0.2455	-0.3211

Таблица 4. Логарифмы отношения медиан 99% квантилей абсолютных ошибок аппроксимации к соответствующим ошибкам эталонной аппроксимации Нгуена-Видроу (алгоритм обучения RProp).

Функция	Алгоритмы инициализации									
	SCAWI(1)		SCAWI(2)		SWI		LFWI		PLWI	
	median	std	median	std	median	std	median	std	median	std
f1	0.00143	-0.1215	0.03365	-0.0439	0.00694	-0.0556	0.01192	-0.0995	0.07104	-0.1942
f2	0.11702	0.19982	0.09137	0.29239	0.15682	0.03329	0.09749	0.03919	0.05784	-0.1979
f3	0.06611	0.37589	-0.1011	-0.0885	0.05493	0.19989	0.32021	0.04927	0.10956	0.05395
f4	-0.0285	-0.1197	-0.1769	-0.3483	0.00159	-0.2148	0.51602	0.07588	-0.3914	-0.8015
f5	-0.002	0.08152	0.02466	0.14214	0.00192	0.12769	-0.0332	0.06066	0.00907	0.10318
f6	-0.6773	-0.9252	-0.4478	-0.832	-0.714	-0.9459	1.50244	1.25033	1.09032	0.98517
f7	-0.1169	-0.2502	-0.0387	-0.4292	-0.0896	-0.1361	0.1879	-0.2208	0.04233	-0.0448
f8	-0.4185	-0.7927	-0.5826	-1.1004	-0.446	-0.6888	1.50654	0.86976	1.05718	0.34239
f9	-0.0685	-0.0576	-0.1637	-0.1634	-0.0545	-0.0566	0.44076	0.00706	0.20288	-0.0415
f10	-0.4375	-0.6828	-0.285	-0.8528	-0.5131	-0.5728	2.00109	2.30441	1.83249	1.8046
f11	-0.8334	-1.0593	-0.8347	-0.997	-0.7683	-1.0395	2.64417	2.4171	1.52734	1.54451
f12	-0.0026	0.22524	-0.0004	0.03793	-0.0128	0.27982	-0.0016	-0.0501	-0.0031	0.14585
f13	0.17198	0.41262	0.36133	0.25003	-0.1015	-0.0367	0.37273	-0.3512	0.19874	-0.1614
f14	-0.5563	-0.8902	-0.3371	-0.3307	-0.5763	-0.5521	1.74475	1.37398	1.48583	1.19062
f15	-0.0923	-0.0682	-0.0135	0.16917	-0.1056	0.14622	0.2064	-0.1311	-0.1479	-0.3589

Таблица 5. Логарифмы отношения медиан абсолютных ошибок аппроксимации к соотвествующим ошибкам эталонной аппроксимации Нгуена-Видроу (алгоритм обучения LM).

Функция	Алгоритмы инициализации									
	SCAWI(1)		SCAWI(2)		SWI		LFWI		PLWI	
	median	std	median	std	median	std	median	std	median	std
f1	-0.0219	0.27179	0.19124	0.48363	0.17252	0.52238	-0.2111	0.49054	-0.0682	0.21034
f2	0.10501	-0.2081	0.14949	-0.3698	0.15221	-0.1161	-0.1279	0.03388	-0.2925	-0.0633
f3	0.67087	0.03355	0.63867	1.01664	0.86904	0.81907	-0.0816	0.09923	-0.3794	-0.2862
f4	0.95373	0.69338	1.01766	1.08851	1.08815	1.31961	-0.7407	-0.4694	-1.7216	-1.3943
f5	0.10042	0.31298	0.21553	0.20553	0.21675	0.37971	-0.0214	0.21212	-0.0115	0.20087
f6	3.02598	2.89488	3.12277	2.88195	3.17327	2.28147	0.29105	-0.0461	0.09075	-0.3522
f7	0.47294	1.17987	0.85518	1.2282	1.11759	1.27004	-0.0221	0.02541	-0.1705	0.24627
f8	2.2349	2.22828	0.57579	1.98802	2.28367	1.09704	-1.1132	-0.5773	-1.5007	-1.1141
f9	1.03919	-0.2422	0.67234	0.85609	1.06724	0.30665	-0.3469	-0.5783	-0.7156	-0.5749
f10	1.22474	0.40139	1.14904	3.28902	0.61305	0.50546	-0.1626	0.7427	-0.3628	0.11495
f11	-0.0981	-0.0252	0.21381	1.16828	-0.6462	-0.0161	-0.0443	0.43197	0.44243	0.70029
f12	-0.0246	-0.2003	-0.0418	-0.7629	-0.0347	-0.2413	-0.0426	-0.2469	-0.0139	0.17383
f13	1.56304	-0.2025	1.66585	0.16682	1.57102	-0.0547	0.41793	0.06791	0.02341	0.04083
f14	0.96672	0.62916	1.13284	3.19859	0.36588	0.71113	-0.2814	0.13526	-0.5908	-0.4571
f15	1.27413	1.06155	1.23834	1.18634	1.46007	0.92935	0.33605	0.19234	-0.318	-0.3291

Таблица 6. Логарифмы отношения медиан среднеквадртичных ошибок аппроксимации к соотвествующим ошибкам эталонной аппроксимации Нгуена-Видроу (алгоритм обучения LM).

Функция	Алгоритмы инициализации									
	SCAWI(1)		SCAWI(2)		SWI		LFWI		PLWI	
	median	std	median	std	median	std	median	std	median	std
f1	-0.0446	0.09738	0.16153	0.16373	0.14898	0.32834	-0.1966	0.37268	-0.0889	0.15234
f2	0.10555	-0.9674	0.16761	-0.9393	0.13076	-0.827	-0.2227	-0.0725	-0.3013	-0.4143
f3	0.67578	-0.3532	0.67769	0.87229	0.86073	0.72587	-0.0155	-0.0118	-0.0996	-0.1895
f4	0.94072	1.00883	0.96055	1.088	1.03632	1.39451	-0.7848	-0.2543	-1.722	-1.2005
f5	0.08714	0.04672	0.14669	-0.0638	0.15869	0.02708	-0.0012	0.20605	0.03311	0.13543
f6	2.83092	2.47369	2.94035	2.47782	2.96966	1.83531	0.2095	-0.1039	0.3468	-0.0826
f7	0.32174	1.0647	0.6829	1.09236	0.88266	1.134	-0.0993	0.10308	-0.1892	0.06502
f8	2.18867	2.19998	0.65477	1.98432	2.21695	1.06279	-0.9904	-0.5002	-1.2932	-1.0939
f9	1.01626	0.04863	0.7283	0.75642	1.03714	0.21829	-0.3085	0.12963	-0.4688	-0.63
f10	1.03019	0.17879	0.94433	2.93455	0.41882	0.15178	-0.2558	0.70774	-0.0498	-0.1942
f11	-0.226	-0.3113	0.02703	0.72037	-0.8236	-0.4522	-0.2066	0.70763	0.26276	0.61552
f12	-0.0291	-0.2411	-0.0495	-0.7882	-0.0411	-0.2551	-0.0492	-0.2094	-0.0226	0.20942
f13	1.50908	-0.2602	1.59831	0.16322	1.50272	-0.0825	0.38457	0.09963	0.00525	0.05269
f14	0.78199	0.19539	0.89744	2.64489	0.14134	0.1707	-0.4336	-0.1383	-0.5923	-0.1247
f15	1.1694	0.92207	1.19446	1.06382	1.34139	0.79226	0.41709	0.32571	-0.2148	-0.2043

Таблица 7. Логарифмы отношения медиан 95% квантилей абсолютных ошибок аппроксимации к соотвествующим ошибкам эталонной аппроксимации Нгуена-Видроу (алгоритм обучения LM).

Функция	Алгоритмы инициализации									
	SCAWI(1)		SCAWI(2)		SWI		LFWI		PLWI	
	median	std	median	std	median	std	median	std	median	std
f1	-0.0394	-0.0194	0.19213	0.30702	0.17545	0.32611	-0.1982	0.10371	-0.0803	0.00826
f2	0.06064	-0.3014	0.09265	-0.4539	0.09934	-0.4556	-0.1264	-0.0625	-0.2311	-0.2108
f3	0.67705	0.25452	0.65806	0.96049	0.86453	0.82185	-0.056	0.07079	-0.2652	-0.0865
f4	0.99457	1.00847	1.03951	1.11224	1.12533	1.53727	-0.7043	-0.3592	-1.6518	-1.1439
f5	0.06957	-0.0724	0.09348	-0.5198	0.10189	-0.1952	-0.0304	0.06423	0.00446	0.06589
f6	2.95369	2.76207	3.06348	2.80204	3.07672	2.24741	0.30808	-0.0039	0.08553	-0.1585
f7	0.50005	1.22887	0.87865	1.24671	1.04604	1.23823	-0.0262	0.21063	-0.1709	0.28516
f8	2.14989	2.17588	0.53357	1.97164	2.17092	1.39878	-1.1186	-0.5922	-1.5507	-1.0473
f9	0.99459	0.40407	0.59666	0.80788	1.02738	0.40837	-0.3356	-0.4444	-0.6418	-0.4129
f10	1.16506	0.375	1.03871	3.16077	0.52082	0.34527	-0.1607	0.61471	-0.3308	-0.0376
f11	-0.2059	0.06333	0.16091	1.23983	-0.754	0.08124	-0.1145	0.3341	0.33585	0.89719
f12	-0.0368	-0.2507	-0.0616	-0.7411	-0.0429	-0.3198	-0.0578	-0.216	-0.0324	0.15411
f13	1.44511	-0.2909	1.5314	0.1317	1.4457	-0.107	0.37208	0.08796	-0.0094	0.03005
f14	0.9813	0.60184	1.14009	3.08731	0.38825	0.62167	-0.2143	0.06568	-0.5731	-0.5377
f15	1.29636	0.9769	1.24223	1.1562	1.4965	0.89878	0.40511	0.27836	-0.2359	-0.4771

Таблица 8. Логарифмы отношения медиан 99% квантилей абсолютных ошибок аппроксимации к соответствующим ошибкам эталонной аппроксимации Нгуена-Видроу (алгоритм обучения LM).

Функция	Алгоритмы инициализации									
	SCAWI(1)		SCAWI(2)		SWI		LFWI		PLWI	
	median	std	median	std	median	std	median	std	median	std
f1	-0.0553	-0.1617	0.11668	-0.4996	0.1052	-0.224	-0.1606	0.07671	-0.0795	-0.0669
f2	0.3346	-0.7173	0.40516	-0.7561	0.33624	-0.8926	-0.1896	-0.2946	-0.2546	-0.6427
f3	0.49776	0.5002	0.52285	0.66097	0.70062	0.45673	-0.0497	-0.0246	-0.0988	-0.1772
f4	0.95548	0.8278	0.9245	0.8816	1.04636	1.24129	-0.7215	-0.1887	-1.6161	-1.159
f5	0.0095	-0.3033	0.01983	-0.7042	0.05094	-0.176	0.00096	0.15361	0.04079	0.1809
f6	2.67599	2.65529	2.81734	2.75591	2.74812	2.06191	0.22185	0.31902	0.26038	0.27562
f7	0.32591	0.56263	0.62911	0.60311	0.67242	0.5584	-0.0656	0.01038	-0.1291	-0.0842
f8	2.11446	2.02532	0.8097	1.8961	2.18035	1.71232	-0.8537	-0.4995	-1.3427	-1.1537
f9	0.98165	0.26766	0.76992	0.40675	0.98747	0.06038	-0.2491	-0.3713	-0.3342	-0.8463
f10	1.02609	0.54366	1.08678	2.93977	0.47773	0.17812	-0.1605	0.63656	-0.0572	-0.1964
f11	-0.1043	0.2995	0.12816	0.99792	-0.8243	-0.157	-0.1455	1.00136	0.34421	1.30483
f12	-0.0807	-0.1892	-0.1166	-0.6666	-0.07	-0.2059	-0.0908	-0.2318	-0.0722	0.24307
f13	1.38154	-0.2473	1.45534	0.13262	1.36479	-0.0593	0.35942	0.18274	0.02097	0.05881
f14	0.6886	0.22727	0.7261	2.70465	-0.0145	0.18055	-0.4886	-0.2036	-0.6673	-0.2892
f15	1.05449	0.84972	1.04175	0.94398	1.18963	0.73769	0.35474	0.20897	-0.3732	-0.1281

Таблица 9. Логарифмы ошибок аппроксимации по отношению к эталонному алгоритму Нгуена-Видроу (NWWI) для многомерных функций.

Функция	ϕ_1	ϕ_2	ϕ_3
Размерность	6	5	6
Мощность выборки	1000	160	160
Логарифмы отношения абсолютных ошибок			
SCAWI(2)	0,096	-0,693	-0,098
SWI	-0,285	-0,994	-0,146
Логарифмы отношения среднеквадратичных ошибок			
SCAWI(2)	0,087	-0,660	-0,104
SWI	-0,284	-0,989	-0,154
Логарифмы отношения 95% квантилей абсолютных ошибок			
SCAWI(2)	0,077	-0,652	-0,103
SWI	-0,288	-1,012	-0,159
Логарифмы отношения 99% квантилей абсолютных ошибок			
SCAWI(2)	0,051	-0,600	-0,108
SWI	-0,291	-0,965	-0,154

Основные результаты работы:

1. Предложен общий детерминированный подход к инициализации моделей вида (1), позволяющий добиваться повторяемости результатов, сокращения времени обучения и в некоторых случаях повышения качества конечной аппроксимации;
2. Разработаны два новых алгоритма (на основе кусочно-линейной аппроксимации и на основе локальных особенностей аппроксимируемой зависимости) в рамках предложенного подхода;
3. Разработан рандомизированный алгоритм инициализации для эффективной аппроксимации зависимостей высокой размерности (сферическая инициализация);
4. Разработано усовершенствование классического метода SCAWI (за счет расположения центров сигмоидов в точках исходной выборки), позволившее повысить качество конечной апроксимации специфических классов зависимостей (гладкие функции и разрывные функции со множеством особенностей в области определения) при использования для обучения модели алгоритма RProp;
5. Проведено сравнение классических и новых предложенных методов инициализации, позволившее выявить наиболее эффективные алгоритмы.

СПИСОК ЛИТЕРАТУРЫ

1. A. Bernstein, E. Burnaev, and A. Kuleshov. Adaptive models of complex systems based on data handling. In Proceedings of the 3rd International Conference on Inductive Modelling ICIM-2010, May 16-22, 2010, Kyiv, Ukraine, pages 64–71, May 2010.
2. D. Banks. Statistical data mining. Wiley Interdisciplinary Reviews: Computational Statistics, 2:9–25, 2010.
3. J. Kolen and J. Pollack. Back propagation is sensitive to initial conditions. In Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3, NIPS-3, pages 860–867, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.
4. G. Drago and S. Ridella. Possibility and Necessity Pattern Classification using an Interval Arithmetic Perceptron. Neural Computing & Applications, 8(1):40–52, 1999.
5. M. Hyder, M. Shahid, M. Kashem, and M. Islam. Initial Weight Determination of a MLP for Faster Convergence. Journal of Electronics and Computer Science, 10, 2009.
6. E. Oja. Robust fitting by nonlinear neural units. Neural Networks, 9(3):435–444, 1996.
7. G. Thimm and E. Fiesler. Optimal setting of weights, learning rate, and gain. Idiap-RR Idiap-RR-04-1997, IDIAP, 1997.
8. R. Sutton, C. Szepesvári, A. Geramifard, and M. Bowling. Dyna-style planning with linear function approximation and prioritized sweeping. In D. McAllester and P. Myllymäki, editors, Proceedings of the Twenty-fourth Conference in Uncertainty in Artificial Intelligence (UAI 2008), pages 528–536. AAAI Press, 2008.
9. Xi Min Zhang, Yan Qiu Chen, Nirwan Ansari, and Yun Q Shi. Mini-max initialization for function approximation. Neurocomputing, 57:389–409, 2004.
10. Z. Houkes. Incorporating a priori knowledge into initialized weights for neural classifier. In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, volume 2, pages 291–296. IEEE, 2000.
11. M. Belyaev and E. Burnaev. Approximation of a multidimensional dependency based on a linear expansion in a dictionary of parametric functions. Informatics and Applications, 7:114–125, 2013.
12. S. Grihon, E. Burnaev, M. Belyaev, and P. Prikhodko. Surrogate Modeling of Stability Constraints for Optimization of Composite Structures, pages 359–391. Springer Series in Computational Science & Engineering. Springer, New York, 2013.

13. E. Burnaev and P. Prikhodko. On a method for constructing ensembles of regression models. Automation and Remote Control, 74:1630–1644, 2013.

14. O. Fontenla-Romero, D. Erdogmus, J. Príncipe, A. Alonso-Betanzos, and E. Castillo. Accelerating the convergence speed of neural networks learning methods using least squares. In ESANN 2003, 11th European Symposium on Artificial Neural Networks, Bruges, Belgium, April 23-25, 2003, Proceedings, pages 255–260, 2003.

15. C. Igel, M. Toussaint, and W. Weishui. Rprop using the natural gradient. In D. Mache, J. Szabados, and M. de Bruin, editors, Trends and Applications in Constructive Approximation, volume 151 of ISNM International Series of Numerical Mathematics, pages 259–272. Birkhauser Basel, 2005.

16. M. Fernández-Redondo and C. Hernandez-Espinosa. Weight initialization methods for multilayer feedforward. In Proceedings of the 9th European Symposium on Artificial Neural Networks ESANN-2001, pages 119–124, April 2001.

17. D. Nguyen and B. Widrow. Improving the Learning Speed of 2-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), volume 3, pages 21–26. IEEE, June 1990.

18. A. Pavelka and A. Procházka. Algorithms for initialization of neural network weights. In Sborník príspevku 12 rocníku konference MATLAB 2004, Prague, pages 453–459, 2004.

19. V. Maiorov, K. Oskolkov, and V. Temlyakov. Gridge approximation and Radon compass, pages 284–309. DARBA, Sofia, 2002.

20. A. Bruckstein, D. Donoho, and M. Elad. From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images. SIAM Review, 51(1):34–81, 2009.

21. V. Joseph and H. Ying. Orthogonal-Maximin Latin Hypercube Designs. Statistica Sinica, 18(1):171–186, 2008.

22. D. Hush and B. Horne. Efficient algorithms for function approximation with piecewise linear sigmoidal networks. IEEE Transactions on Neural Networks, 9(6):1129–1141, 1998.

23. A. Bowman and A. Azzalini. Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations (Oxford Statistical Science Series). Oxford University Press, USA, November 1997.

INFLUENCE OF INITIALIZATION ON LEARNING TIME AND ACCURACY OF NONLINEAR REGRESSION MODEL

Burnaev E.V., Erofeev P.D.

Institute for Information Transmission Problems,
Russian Academy of Sciences, Moscow, Russia

In order to construct a nonlinear regression model we have to accurately (in some sense) initialize parameters of the model. In this work we performed comparison of several widely used methods and several newly developed approached for initialization of parameters of a regression model, represented as a decomposition in a linear dictionary of some parametric functions (sigmoids). We proposed a general deterministic approach for initialization, providing repeatability of results, reduction of a learning time and in some cases increase of a regression model accuracy; we developed two new algorithms (based on a piecewise-linear approximation and based on local properties of approximable dependency) in the framework of the proposed approach; we developed randomized initialization algorithm (spherical initialization) for effective approximation of high-dimensional dependencies; we improved the classical initialization method SCAWI (by locating centers of sigmoids in sample points), providing a regression model accuracy improvement on specific classes of dependencies (smooth functions and discontinuous functions with a number of local peculiarities in an

input domain) when using RProp algorithm for learning; we performed comparison of classical and newly proposed initialization methods and highlighted the most efficient ones.

KEYWORDS: nonlinear regression, neural networks, parameters initialization, SCAWI algorithm, RProp algorithm, Levenberg-Marquardt algorithm, backpropagation algorithm.