

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Математико-Механический факультет
Кафедра системного программирования

Extreme Learning Machine: алгоритмическая оптимизация и подбор параметров

Курсовая работа студента 344 группы
Луценко Александра Николаевича

Научный руководитель:
аспирант Кафедры системного программирования
Невострюев К. Н.

Санкт-Петербург, 2014

Оглавление

Введение.....	3
1. Постановка задачи.....	4
2. Обзор существующих алгоритмов.....	5
3. Принцип работы ELM.....	6
4. Улучшения ELM.....	8
4.1. Точность: RBF	8
4.2. Скорость: ECODLS.....	8
5. Подбор числа нейронов.....	10
5.1. Модификация Grid Search.....	10
5.2. Constructive Selection.....	12
6. Результаты тестов.....	13
Заключение.....	15
Список литературы.....	16

Введение

В последние годы наблюдается лавинообразный рост количества доступной человечеству информации. Всю эту информацию нужно обрабатывать, а из-за огромного её объёма сделать это вручную зачастую просто невозможно. На помощь приходит машинное обучение — раздел искусственного интеллекта, изучающий подходы к построению самообучающихся алгоритмов. К сожалению, большинство традиционных методов плохо проявляют себя на больших объёмах данных: они имеют либо неудовлетворительную точность, либо неприемлемо высокую вычислительную сложность. Поэтому данная область нуждается в исследованиях. Если конкретнее, нужны эффективные алгоритмы, способные справляться с очень большими объёмами данных в условиях ограниченных ресурсов (времени и памяти). Одним из таких алгоритмов является ELM (Extreme Learning Machine) — нейронная сеть, предназначенная для решения задач регрессии и классификации. Собственно, эффективная реализация ELM, а также поиск методов автоматического подбора количества нейронов для неё и является темой данной работы.

1. Постановка задачи

В рамках курсовой работы были поставлены следующие задачи:

- Реализовать нейронную сеть ELM
- Произвести возможные модификации алгоритма с целью повышения его точности и скорости
- Попытаться найти и реализовать алгоритм автоматического нахождения оптимального числа нейронов для ELM.

2. Обзор существующих алгоритмов

Машины опорных векторов (Support Vector Machines, SVMs)

Изначально предназначались для решения задач классификации. Принцип работы — в нахождении гиперплоскости в пространстве признаков, которая бы разделяла точки из этого пространства, принадлежащие разным классам. Причём из всех возможных вариантов выбирается такая гиперплоскость, которая максимально отдалена ото всех точек.

Существуют модификации машин опорных векторов (например, SVR — Support Vector Regression), которые вместо классификации решают задачи регрессионного анализа.

Искусственные нейронные сети (Artificial Neural Networks, ANNs)

Вычислительные модели, построенные в некотором смысле по принципу организации и функционирования центральной нервной системы живого организма. По своей природе нейронные сети регрессоры, однако легко могут быть адаптированы к задачам классификации. Состоят из отдельных узлов, называемых искусственными нейронами, взаимодействующих между собой. Каждый нейрон принимает вектор входных сигналов от других нейронов или извне, в соответствии со своими коэффициентами связей преобразует в одиночный выходной сигнал и передаёт его далее по цепочке. Задача обучения нейронной сети состоит в определении коэффициентов связей между нейронами.

Деревья принятия решений (Decision Trees)

Представляют собой древовидную структуру. На ней есть метки. В узлах, не являющихся листьями — атрибуты, по которым различаются случаи. В листьях — значения целевой функции. На рёбрах — значения атрибута, из которого исходит ребро. Чтобы составить предсказание для новых входных данных, нужно спуститься по рёбрам до листа и выдать соответствующее значение.

Одиночные деревья не могут достичь точности, сопоставимой с ранее упомянутыми алгоритмами машинного обучения. Однако так называемые леса принятия решений хорошо зарекомендовали себя на практике. Суть в том, чтобы построить много отличающихся друг от друга деревьев, а результат предсказания определять через всеобщее «голосование». Существуют разновидности деревьев/лесов принятия решений, специализирующихся на задачах как классификации, так и регрессии.

3. Принцип работы ELM

Extreme Learning Machine [1] — нейронная сеть с прямой связью и одним скрытым слоем (SLFN - Single Layer Feedforward Network), веса и смещения нейронов на котором инициализируются случайными значениями.

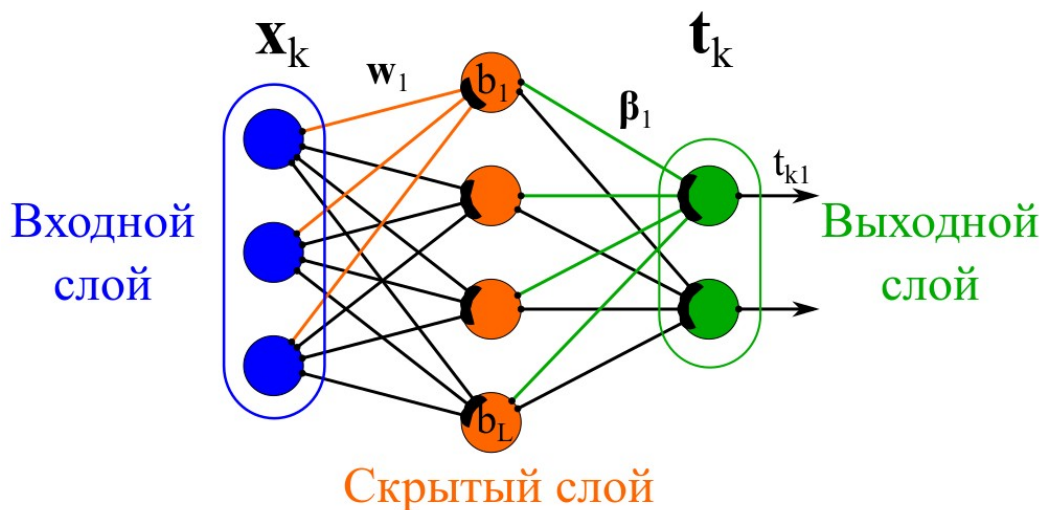


Рис.1: Схема ELM

Предположим, имеется обучающая выборка из N произвольных примеров $(\mathbf{x}_k, \mathbf{t}_k)$, где $\mathbf{x}_k \in \mathbb{R}^n$ - вектор признаков, а $\mathbf{t}_k \in \mathbb{R}^m$ - целевой вектор k -го примера. Выберем число нейронов на скрытом слое L и функцию активации G .

Математическая модель SLFN следующая:

$$\mathbf{t}_{k,s} = \sum_{i=1}^L \beta_{s,i} G(\mathbf{w}_i \mathbf{x}_k + b_i), \quad s = 1, \dots, m, \quad k = 1, \dots, N,$$

где

$\mathbf{t}_{k,s}$ - s -я компонента целевого вектора \mathbf{t}_k , соответствующего вектору признаков \mathbf{x}_k ,

$\mathbf{w}_i \in \mathbb{R}^n$ - веса (weights) i -го нейрона на скрытом слое,

$b_i \in \mathbb{R}$ - смещение (bias) i -го нейрона на скрытом слое,

$\beta_s \in \mathbb{R}^L$ - веса s -го нейрона на выходном слое.

Суть обучения ELM заключается в нахождении весов и смещений для нейронов на скрытом и выходном слое.

Веса \mathbf{w}_i и смещения b_i для каждого нейрона на скрытом слое инициализируются случайными значениями (из равномерного распределения на $[0, 1]$). Веса β_s для скрытых нейронов неизвестны, их необходимо найти.

Указанные выше $N \times m$ уравнений могут быть переписаны в матричной форме:

$$\mathbf{H} \boldsymbol{\beta} = \mathbf{T},$$

$$\text{где } \mathbf{H} = \begin{pmatrix} G(\mathbf{w}_1 \mathbf{x}_1 + b_1) & \dots & G(\mathbf{w}_L \mathbf{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ G(\mathbf{w}_1 \mathbf{x}_N + b_1) & \dots & G(\mathbf{w}_L \mathbf{x}_N + b_L) \end{pmatrix}_{N \times L}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_m \end{pmatrix}_{L \times m}, \quad \mathbf{T} = \begin{pmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_N \end{pmatrix}_{N \times m}.$$

Тогда матрица весов выходного слоя $\boldsymbol{\beta}$ вычисляется как

$$\boldsymbol{\beta} = \mathbf{H}^\dagger \mathbf{T},$$

где \mathbf{H}^\dagger - псевдообращение Мура-Пенроуза матрицы \mathbf{H} .

Таким образом, получаем следующий алгоритм обучения ELM:

Дан обучающий набор $(\mathbf{x}_i, \mathbf{t}_i)$, $i = 1, \dots, N$, функция активации G и количество нейронов на скрытом слое L

1. Инициализировать случайными значениями \mathbf{w}_i и b_i
2. Построить матрицу \mathbf{H}
3. Вычислить веса на выходном слое: $\boldsymbol{\beta} = \mathbf{H}^\dagger \mathbf{T}$

4. Улучшения ELM

4.1. Точность: RBF

В классических персептронных нейронных сетях (которой является стандартная ELM) функция активации вычисляется, как было описано выше, от скалярного произведения между входным вектором признаков \mathbf{x}_k и вектором весов \mathbf{w}_i . В случае же с RBF-сетями (Radial Basis Function) функция активации есть функция от расстояния между \mathbf{x}_k и некоторым эталонным элементом $\boldsymbol{\mu}_i$ из пространства признаков, называемым центром i -го ядра. На многих обучающих выборках сети с RBF достигают большей точности предсказания, нежели классические сети.

В работе [2] показано, что стандартный ELM может быть модифицирован для работы с RBF. Математическая модель RBF-ELM:

$$t_{k,s} = \sum_{i=1}^L \beta_{s,i} \phi(\|\mathbf{x}_k - \boldsymbol{\mu}_i\|, \sigma_i), \quad s = 1, \dots, m, \quad k = 1, \dots, N,$$

где

ϕ - функция активации

$t_{k,s}$ - s -я компонента целевого вектора \mathbf{t}_k , соответствующего вектору признаков \mathbf{x}_k ,
 $\boldsymbol{\mu}_i \in \mathbb{R}^n$ - центр (center) i -го ядра,

$\sigma_i \in \mathbb{R}$ - радиус воздействия (impact width) i -го ядра,

$\beta_s \in \mathbb{R}^L$ - веса s -го нейрона на выходном слое.

В качестве ядра $\boldsymbol{\mu}_i$ можно брать случайный вектор признаков \mathbf{x}_r из обучающей выборки, а σ_i генерировать в диапазоне $[0, 1]$. Дальнейший процесс обучения такой же, как и в случае классического ELM.

4.2. Скорость: ECODLS

Самая затратная операция в процессе обучения ELM — нахождение псевдообратной матрицы \mathbf{H}^\dagger . В оригинальной статье её предложено производить при помощи сингулярного разложения (singular value decomposition, SVD):

$$\mathbf{H}^\dagger = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{T}$$

Вычислительная сложность такого метода составляет $O(4NL^2 + 8L^3)$. Из данной оценки видно, что время обучения ELM линейно зависит от количества обучающей выборки (что не может не радовать), однако кубически — от числа нейронов. Последний факт создаёт серьёзные трудности при обучении сложных моделей (когда требуется много нейронов). Для решения этой проблемы исходный алгоритм нахождения \mathbf{H}^\dagger был заменён алгоритмом ECODLS, описанным в работе [3]. В нём используется расширенное полное ортогональное

разложение (extended complete orthogonal decomposition, ECOD), сложность которого $O(2NL^2 - 2L^3/3 + L^2)$.

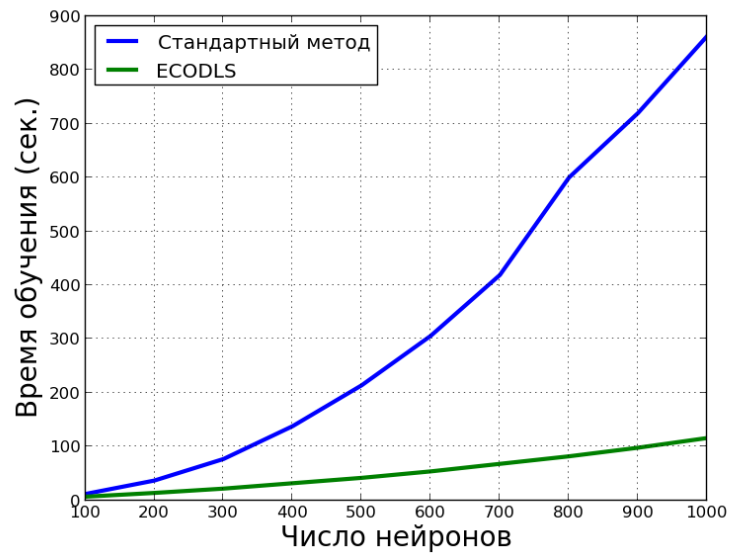


рис. 2: Время обучения ELM, осуществляемого стандартным методом и ECODLS. Набор данных — MNIST (60 000 обучающих примеров).

По графику видно, что внедрение ECODLS в ELM позволило весьма существенно сократить время обучения в случаях, когда число нейронов L приближается к размеру обучающей выборки N .

5. Подбор числа нейронов

Основной параметр, который необходимо определить для успешного обучения нейронной сети — это её топология, т. е. количество нейронов на скрытом слое.

5.1. Модификация *Grid Search*

В области машинного обучения, пожалуй, самым распространённым методом подбора параметров обучающего алгоритма является решётчатый поиск (*Grid Search*). Пространство допустимых параметров алгоритма разбивается на сетку (чаще всего равномерную). Координаты узлов сетки — это какие-то конкретные значения параметров. Алгоритм обучается на всех узлах сетки, окрестность того узла, на котором алгоритм показывает наилучшие результаты, дробится на более мелкую сетку, и весь процесс повторяется.

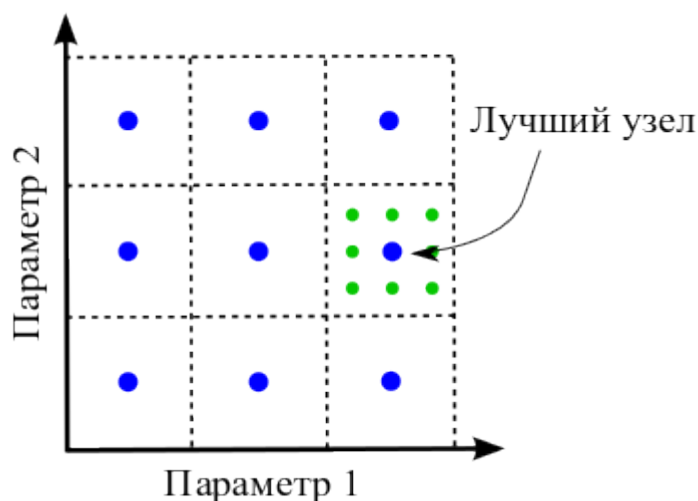


рис.2: Схема работы *Grid Search*

В нашем случае параметр всего один — число нейронов, стало быть, сетка одномерная.

Решётчатый поиск имеет существенные достоинства: простота понимания, хорошая скорость работы, если число параметров невелико ($O(\log(m))$, где m — количество узлов сетки). Из недостатков — чувствительность к значительным неровностям поверхности поиска, незащищённость от попадания в локальные минимумы.

Логичной представляется мысль попробовать искать оптимальное число нейронов для ELM при помощи *Grid Search*. Однако стохастическая природа данной нейронной сети создаёт значительные трудности в применении решётчатого поиска. Дело в том, что на получающейся поверхности поиска часто наблюдаются сильные аномальные всплески (рис.3). Следовательно, для успешного поиска минимума требуется очень мелкая сетка, что практически сводит на нет преимущество в скорости относительно обычного перебора.

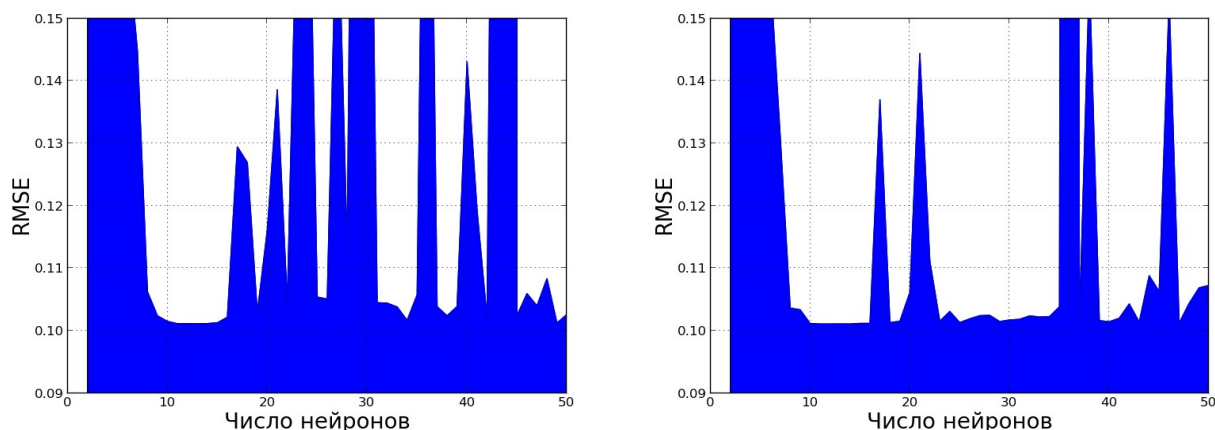


рис.3: Зависимость погрешности предсказания (RMSE) от количества нейронов в ELM на примере набора данных Sinc function (регрессия)

Проблему всплесков на поверхности можно решить, если параллельно обучать несколько ELM, а в качестве итоговой ошибки брать медианную ошибку среди них (рис.4).

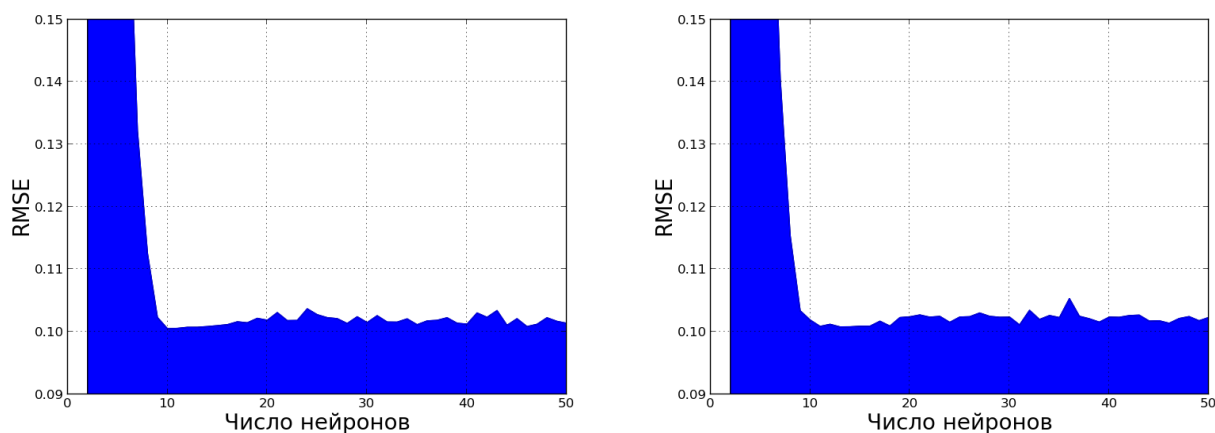


рис.4: Зависимость погрешности предсказания (RMSE) от количества нейронов, медиана по 10 независимо обученным ELM. Набор данных — Sinc function

Ясно, что такой подход в разы увеличивает время подбора параметров, однако он может быть оправдан, если диапазон количества нейронов велик (порядка нескольких сотен штук).

Отдельный вопрос в том, как вычислять погрешность предсказания на конкретном наборе параметров. Решено было применять стандартный подход: измерять RMSE (Root Mean Square Error) для регрессии и процент точности для классификации при помощи кросс валидации (Cross Validation).

Время поиска было существенно сокращено за счёт того, что в каждом последующем узле решетки ELM не обучалась заново, а пересчитывалась из уже обученной нейросети на

предыдущем узле. Данный подход был предложен в работе [6].

5.2. Constructive Selection

Предложенный ранее метод теряет своё преимущество в скорости, если диапазон возможного количества нейронов невелик. Недостаток же — существенный разброс результатов — остаётся. Задаче точного автоматического поиска оптимального числа нейронов для ELM посвящено довольно много работ. Одна из них — метод Constructive Selection[4]. Вкратце, его суть такова:

1. Генерируется L_{max} нейронов, которые затем сортируются по «полезности», т. е. своему вкладу в уменьшение погрешности предсказания.
 2. Выбирается такое L , что погрешность предсказания обученной нейросети с L нейронами минимальна.
- Предложенный в работе метод оценки погрешности более точен, нежели кросс валидация (рис. 5), однако требует, чтобы нейроны были отсортированы.
 - Как и в предложенном выше методе, нейронная сеть с $L+1$ нейронами не обучается заново, а пересчитывается из предыдущей сети (с L нейронами).

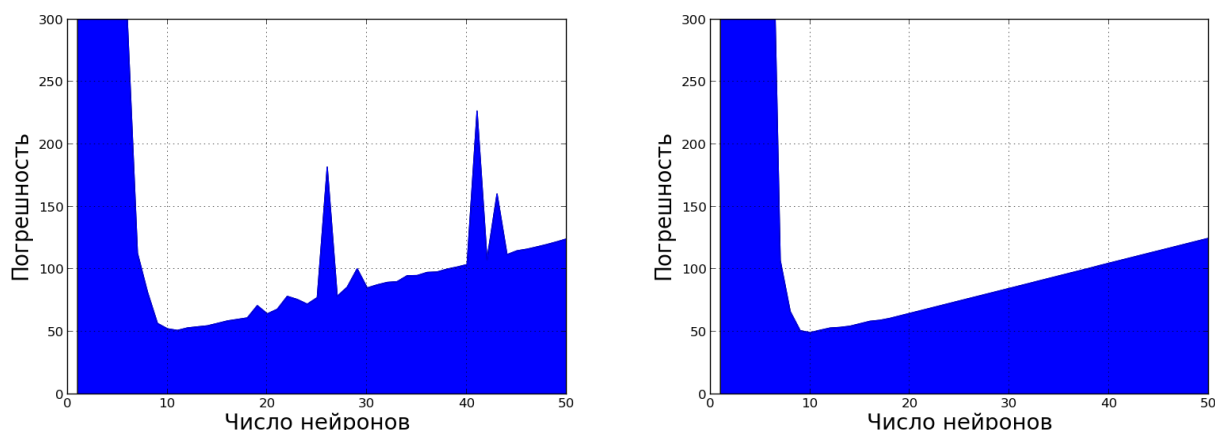


рис. 5: Зависимость погрешности (по методу Constructive Selection) от количества нейронов в ELM. Оптимальное число нейронов находится более точно за счёт наличия ярко выраженного минимума (в сравнении с рис. 4)

У оригинального Constructive Selection есть весомый изъян: используемый в работе метод пересчёта нейронной сети предполагает создание матрицы размера $N \times N$, где N - размер обучающей выборки. А это значит, для больших данных метод неприменим: уже при $N=100\,000$ требуется более 38 ГБ памяти. Альтернативный подход [6], реализованный в описанной ранее модификации Grid Search, лишён этого недостатка. Поэтому пересчитывать ELM было решено при помощи него.

6. Результаты тестов

Обучение, классификация

Набор данных	Размер обучающей выборки	Число нейронов	Время обучения	Время на 1 предсказание	Точность (\pm Std)	F-score
MNIST	60000	1500	216 сек	1.3 мс	$94.8 \pm 0.6\%$	0.948
		3000	809 сек	1.3 мс	$96.1 \pm 0.4\%$	0.961
Adults	32561	800	39 сек	0.12 мс	$85.0 \pm 0.1\%$	0.648
Poker	25010	2500	232 сек	1.47 мс	$72.5 \pm 0.3\%$	0.693
Titanic	1526	35	0.02 сек	0.03 мс	$77.3 \pm 0.0\%$	0.521
Diabetes	576	31	0.009 сек	0.04 мс	$77.0 \pm 1.4\%$	0.618

Подбор числа нейронов, регрессия:

Набор данных	Алгоритм	Макс. число нейронов	Подобранное число нейронов (\pm Std)	Время обучения	RMSE (\pm Std)
Sinc function	ELM	—	16	0.02 сек	0.0044 ± 0.0002
	Constructive Selection	30	10.3 ± 0.8	0.7 сек	0.0043 ± 0.0012
		50	9.7 ± 0.8	1.9 сек	0.0041 ± 0.0016
		100	9.9 ± 0.7	8.9 сек	0.0041 ± 0.0015
	Модифиц. Grid Search (10 ELMs)	30	23.3 ± 12.0	0.9 сек	0.0043 ± 0.0053
		50	31.3 ± 7.8	1.4 сек	0.0043 ± 0.0068
	100	29.9 ± 19.8	3.0 сек	0.0043 ± 0.0055	

Подбор числа нейронов, классификация:

Набор данных	Алгоритм	Макс. число нейронов	Подобранное число нейронов (\pm Std)	Время обучения	Точность (\pm Std)	F-score
Diabetes	ELM	—	35	0.01 сек	$77.0 \pm 1.4\%$	0.618
	Constructive Selection	50	22.5 ± 8.2	0.13 сек	$77.1 \pm 1.3\%$	0.629
		100	38.9 ± 13.3	2 сек	$77.2 \pm 1.6\%$	0.635
	Модифиц. Grid Search (10 ELMs)	50	30.6 ± 7.4	0.3 сек	$77.1 \pm 1.4\%$	0.630
		100	42.9 ± 15.6	0.6 сек	$77.1 \pm 1.5\%$	0.629

В таблицах приведены усреднённые результаты по нескольким запускам алгоритма (100 для малых и 10 для больших наборов данных). В качестве меры разброса от среднего значения взято стандартное отклонение (Standard deviation, Std).

По двум последним таблицам можно сделать выводы о достоинствах и недостатках реализованных методов нахождения оптимального числа нейронов, а также обозначить область их применения.

Constructive Selection:

- Имеет хорошую среднюю точность с малым разбросом (Std)
- Стремится к выбору минимально необходимого количества нейронов, что впоследствии благотворно сказывается на времени предсказания обученной ELM
- Достаточно быстро работает, когда максимально заданное число нейронов L_{max} невелико

Предложенная модификация Grid Search:

- Также обладает хорошей средней точностью, но с довольно сильным её разбросом
- Выгодно использовать при большом L_{max}

Заключение

В рамках данной работы были достигнуты следующие результаты:

- Реализован алгоритм ELM, а также его модификация с RBF.
- Произведена алгоритмическая оптимизация, позволившая весьма существенно ускорить процесс обучения.
- Реализован и улучшен метод автоматического подбора числа нейронов Constructive Selection, хорошо справляющийся с поставленной задачей, если предполагаемое количество нейронов относительно невелико.

Для случаев, когда число нейронов требуется выбрать из широкого диапазона, предложен модифицированный Grid Search.

Список литературы

- [1] Huang G-B, Zhu Q-Y, Siew C-K. Extreme learning machine: a new learning scheme of feedforward neural networks // Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on (Volume:2)
- [2] Huang G-B, Siew C-K. Extreme learning machine: RBF network case // Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th (Volume:2)
- [3] Horata P, Chiewchanwattana S, Sunat K. Robust extreme learning machine // Neurocomputing, Volume 102, 15 February 2013, Pages 31–44
- [4] Lan Y, Yeng Chai Soh, Huang G-B. Constructive hidden nodes selection of extreme learning machine for regression // Neurocomputing Volume 73 Issue 16-18, October, 2010
- [5] Cambria E, et al. Extreme Learning Machines [Trends & Controversies] // Intelligent Systems, IEEE (Volume:28 , Issue: 6)
- [6] Romero E, Alquézar R. Comparing error minimized extreme learning machines and support vector sequential feed-forward neural networks // Neural Networks, Volume 25, January, 2012, Pages 122-129