

## ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.8.032.26; 620.22

А.В. Лемзиков, С.П. Кундас, В.В. Куцыло, Д.Г. Иванов

ПРИМЕНЕНИЕ НЕЙРОСЕТЕЙ ДЛЯ ОПРЕДЕЛЕНИЯ  
ХАРАКТЕРНЫХ ТОЧЕК ФАЗОВЫХ ПРЕВРАЩЕНИЙ В СТАЛЯХ  
С РАЗЛИЧНЫМ ХИМИЧЕСКИМ СОСТАВОМ

*Рассматриваются принципы построения нейронных сетей для решения задач материаловедения, в частности для прогнозирования характерных точек на диаграмме фазовых превращений в сталях в зависимости от химического состава и параметров технологического процесса. Представляется архитектура используемых нейронных сетей, обсуждаются особенности их программной реализации в составе разработанной авторами среды моделирования KMTPFramework.*

**Введение**

Использование нейронных сетей в материаловедении приобретает все большую популярность в мире в связи с тем, что обработка материалов является сложным процессом, зависящим от множества факторов [1]. Например, вычисление температуры, при которой аустенит превращается в мартенсит, представляет значительный практический интерес, однако эта температура зависит от большого числа факторов, таких как механические напряжения, исходная структура стали, содержание примесей и др. Процесс образования аустенита при нагреве значительно отличается от его распада при охлаждении. Если при распаде кинетика превращения зависит от химического состава стали, размера аустенитного зерна, то в случае его формирования добавляются такие параметры, как распределение химических элементов и различных фаз в объеме, наличие неметаллических добавок и степень однородности исходного аустенита [2].

В настоящее время в Республике Беларусь нейросетевой подход активно используется в задачах классификации [3] распознавания объектов на изображениях [4]. Применительно к материаловедению нейронные сети использовались в работе [5], где была сделана попытка прогнозирования механических свойств сплавов (прочности, ударной вязкости и др.) в зависимости от их химического состава и режимов термической обработки (закалки). К недостаткам указанной работы можно отнести то, что обучение нейросети производилось на достаточно малом множестве данных (82 набора, 10 из которых использовались в качестве тестовых). Результирующая погрешность прогнозирования при этом достигала 10 %, что может составить 80–100 °С при определении температур начала аустенитного превращения и 20–30 °С при мартенситном превращении. В то же время авторами показана эффективность применения нейросетей при решении подобных задач.

Также можно выделить работу [6], в которой обосновывается эффективность применения нейросетей для анализа различных тепловых, гидродинамических и физико-химических процессов. Несмотря на то, что практическое применение нейронных сетей касалось только исследований термодинамических свойств воды, сама работа представляет интерес, так как автор значительное внимание уделил оценке ошибки обучения и погрешности прогнозирования, рассмотрел такие критерии, как процентная ошибка, среднеквадратичная погрешность, нормализованная среднеквадратичная погрешность, коэффициент корреляции и алгоритмы обучения с использованием этих критериев.

Наибольший вклад в использование нейросетей в материаловедении внесли сотрудники Кембриджского университета. В частности, Бхадешия обосновал возможность использования нейронных сетей при прогнозировании свойств материалов, показал следующие недостатки регрессионных методов по сравнению с нейронными сетями [1].

1. Существует тенденция к тому, что зависимости, полученные методом регрессии, являются линейными. Для получения псевдолинейных зависимостей необходимо добавлять нели-

нейные элементы уравнения. В большинстве случаев для упрощения последующих вычислений нужно стремиться к получению линейных зависимостей, что, однако, далеко не всегда позволяет достичь приемлемой точности.

2. Вид зависимости должен быть выбран до анализа. Общий вид линейного регрессионного уравнения может быть представлен в виде

$$y = \sum_j w_j x_j + \theta, \quad (1)$$

где  $w_j$  – вес, соответствующий входу  $x_j$ ;  $\theta$  – константа сдвига.

В случае когда необходимо учитывать взаимное влияние входов, к уравнению (1) добавляются члены  $\sum_i \sum_k w_{ij} x_i x_k$ , описывающие это влияние. Следующим этапом является определение весовых коэффициентов  $w$ , приводящих к наименьшей погрешности представления исходных данных. На этом этапе сам вид уравнения уже не изменяется.

3. Полученные регрессионные уравнения применяются на всем пространстве входов, что может привести к увеличению погрешности.

Для внесения нелинейностей при использовании нейронных сетей Бхадешия предлагает использовать гиперболический тангенс в качестве активационной функции (2), (3), который обладает рядом преимуществ по сравнению с другими (значительное изменение формы в зависимости от коэффициента перед  $x$ ):

$$y = \sum_i w_i^{(2)} h_i + \theta^{(2)}; \quad (2)$$

$$h_i = \left( \tanh \sum_j w_{ij}^{(1)} x_j + \theta^{(1)} \right). \quad (3)$$

В работах [2, 7] предложен подход, позволяющий избежать «переподгонки» (overfitting) весов нейросети в случае, когда в обучающих наборах присутствуют области либо с редкими значениями, либо наоборот – локальным шумом (области  $A$  и  $B$  на рис. 1). Метод основывается на использовании Байесовых нейронных сетей, т. е. на распознавании ситуации, при которой существует несколько наборов весов, удовлетворяющих обучающему множеству. Вместо формирования уникального набора весов вычисляется распределение вероятности использования различных наборов весов, что обеспечивает решение проблемы неопределенности.

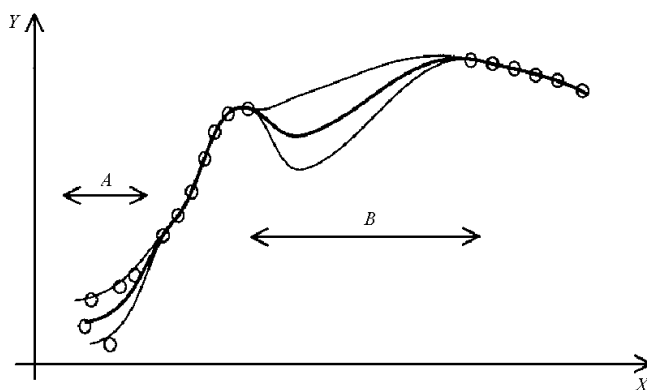


Рис. 1. Пример неопределенности при «шумном» (область  $A$ ) и редком (область  $B$ ) распределении данных в обучающем множестве [1]

## 1. Особенности построения нейронной сети и выбора данных для ее обучения

Как упоминалось выше, нейронные сети уже используются для определения характерных точек на диаграмме фазовых превращений. Однако применяемый в настоящее время подход может быть улучшен. В частности, авторы работы [2] предложили нейронную сеть, прогнози-

рующую точки Ac1 и Ac3 в зависимости от химического состава и скорости нагрева детали. На наш взгляд, такой подход обладает рядом недостатков:

1. Используется нейронная сеть с двумя выходами. В этом случае возможна ситуация, когда погрешность значения одного из выходов приводит к изменению весов нейронов, влияющих и на другой выход. Таким образом, нейросеть начинает уменьшать ошибку одного входа за счет увеличения ошибки другого, что не позволяет ей достичь минимально возможной погрешности.

2. Применяется гиперболический тангенс (2) в качестве активационной функции. Для ряда задач это позволяет значительно уменьшить ошибку прогнозирования. Однако, как показали наши исследования, применение сигмоидальной активационной функции для решения задачи прогнозирования точек Ac1 и Ac3 позволяет увеличить скорость обучения (за счет меньшей вычислительной сложности, более простой производной) при незначительном увеличении погрешности.

Исключить указанные выше недостатки можно путем создания отдельных нейросетей, прогнозирующих точки Ac1, Ac3 и Ms. При этом обучение каждой нейросети не влияет на остальные и позволит уменьшить погрешность прогнозирования. Конечно, при использовании такого подхода придется производить обучение трех нейросетей вместо одной, однако применение сигмоидальной функции активации позволяет уменьшить время, требуемое на обучение.

Наибольшей сложностью при обучении нейронных сетей является формирование достаточно подробных обучающих множеств, т. е. результатов экспериментальных исследований процессов термообработки. В настоящей работе в качестве обучающих множеств использовались открытые источники сети Интернет. В частности, авторы работы [8] предоставляют используемую ими базу данных, содержащую значения температуры начала мартенситного превращения в зависимости от химического состава. Значительную работу по корректировке указанной базы данных произвели Соурмэйл и Гарсиа-Матео [9], обнаружившие множество неточностей, связанных с некорректным переводом различных единиц измерения. Сама база содержит 1092 набора данных с различными комбинациями химических элементов и значениями точки Ms. Для прогнозирования точек Ac1 и Ac3 использовалась база данных, разработанная Бэйлер-Джонсом и Бхадешия для программы MAP\_STEEL\_ACPROS [10]. База содержит 781 набор различных комбинаций химического состава, скорости нагрева и значений точек Ac1 и Ac3. С учетом информации, содержащейся в базах данных, были выбраны входы и выходы для каждой из нейросетей (табл. 1 и 2). Каждый вход нормировался таким образом, чтобы на нейросеть подавались значения в диапазоне от 0 до 1. Соответственно, полученные выходы денормировались для перевода в градусы Цельсия.

Таблица 1

Минимальные (min) и максимальные (max) значения входов и выходов нейросетей для прогнозирования точек Ac1 и Ac3

Название	Min	Max	Название	Min	Max
C (%)	0,001	0,959	Mo (%)	0	4,8
Si (%)	0	2,13	Nb (%)	0	0,17
Mn (%)	0	3,059	V (%)	0	2,45
Cu (%)	0	2,009	W (%)	0	8,59
Ni (%)	0	9,119	Co (%)	0	4,07
Cr (%)	0	17,979	HR (°C/c) (скорость нагрева)	0,029	50
Ac1 (°C)	530	909	Ac3 (°C)	651	1060

Таблица 2

Минимальные (min) и максимальные (max) значения входов и выходов нейросети для прогнозирования точки Ms

Название	Min	Max	Название	Min	Max
C (%)	0	2,25	V (%)	0	4,55
Si (%)	0	3,8	W (%)	0	18,59
Mn (%)	0	10,24	Co (%)	0	16,08
Cu (%)	0	3,037	Al (%)	0	3,00
Ni (%)	0	31,5	Ti (%)	0	2,52

Cr (%)	0	17,98	V (%)	0	0,006
Mo (%)	0	8	N (%)	0	2,65
Nb (%)	0	1,98	Ms (°C)	-119,85	665

## 2. Обучение нейронных сетей

Обучение каждой сети производилось в несколько этапов. На первом этапе эта операция осуществлялась на 1000 итерациях с параметром  $\alpha = 0,1$ . Параметр  $\alpha$  влияет на степень изменения весов в каждом шаге корректировки и, соответственно, в значительной степени определяет скорость обучения. На втором этапе сеть дообучалась с параметром  $\alpha = 0,01$ . В случае если по результатам второго этапа обучения не достигался установившийся режим нейросети (т. е. за последние 100 итераций обучения погрешность прогнозирования продолжала уменьшаться), производилось дополнительное обучение с параметром  $\alpha = 0,01$  на 10 000 итерациях. Таким образом, производилась имитация адаптации скорости обучения, что позволило решить проблему локальных минимумов нейронной сети.

Для обучения был выбран стандартный алгоритм «обучение с учителем». Наборы обучающего множества перед обучением перемешивались в случайном порядке и подавались на входы нейросети в цикле указанное количество раз. Разница между полученным и желаемым ответом умножалась на коэффициент  $\alpha$  и корректировала веса нейронов. Во всех случаях использовались многослойные нейронные сети с заранее заданной структурой слоев, которая включала:

- входной слой – в зависимости от количества входов нейросети;
- скрытые слои – три слоя по пять нейронов в каждом;
- выходной слой – один нейрон.

Структура скрытых слоев выбиралась исходя из вычислительных возможностей имеющегося в наличии оборудования с целью уменьшения времени обучения. Количество скрытых слоев выбиралось экспериментально и в то же время учитывалось, что увеличение количества слоев не всегда уменьшает погрешность прогнозирования [1]. Существует вероятность того, что нейросеть, наиболее точно аппроксимируя обучающее множество, дает большие погрешности на тестовых входах, чем нейросеть с более простой структурой (рис. 2).

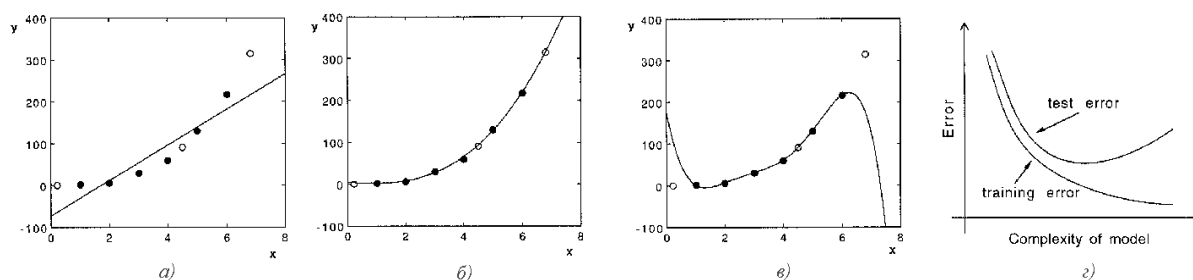


Рис. 2. Изменение ошибок обучения и прогнозирования в зависимости от сложности нейросети. Заполненные точки представляют собой обучающее множество, пустые – тестовые значения [1]

## 3. Программная реализация

Реализация нейронной сети осуществлена в среде KMTPFramework [11], что позволило использовать стандартный интерфейс пользователя, предоставляемый средой, интегрировать нейросетевой модуль в общее пространство с модулем ThermoSim, обеспечивающим моделирование процессов термообработки методом конечных элементов, а также с модулем ThermoHw, предназначенным для управления технологическим оборудованием закалки. До-

полнительные модули к среде моделирования KMTPFramework реализованы в виде динамически подключаемых библиотек (DLL) и содержат классы, реализующие определенные интерфейсы. Так, при разработке модуля нейронных сетей было решено сделать его «расширением» модуля ThermoSim, а не в виде самостоятельной библиотеки, так как в этом случае значительно упрощается процесс обмена информацией с остальными «расширениями».

С точки зрения архитектуры программного обеспечения были выделены и реализованы два класса:

**NeuroNet** – представляет собой непосредственно нейронную сеть и реализует все алгоритмы ее обучения, прогнозирования, сохранения и восстановления состояния и т. п.;

**NeuroProject** – осуществляет объединение нескольких сетей в единое целое, предоставляет пользовательский интерфейс для прогнозирования, анализа полученных результатов, построения графиков.

Оба класса реализуют определенный в пространстве имен **ThermoSimInterfaces** интерфейс **ITSPProject**, который содержит следующие методы:

`void InitModeller(ITSHostApp tsHostApp, string name, string filename)` – производит инициализацию экземпляра класса. В качестве параметров передаются указатель на интерфейс родительского объекта (в нашем случае – на модуль ThermoSim), имя проекта и имя файла для последующего сохранения;

`System.Drawing.Image GetLargeImage()` – возвращает пиктограмму размером 32×32 пикселя для размещения в галерее доступных проектов;

`System.Drawing.Image GetSmallImage()` – возвращает пиктограмму размером 16×16 пикселей для размещения в галерее доступных проектов в режиме мелких значков;

`bool BeforeClose()` – вызывается перед закрытием проекта. Возвращает true в случае, если проект готов к закрытию, и false в противном случае;

`void RemoveFromTree()` – производит удаление из дерева проектов всех элементов, связанных с текущим проектом;

`void Save()` – осуществляет сохранение в файл по умолчанию;

`void SaveAs(string newFileName)` – производит сохранение в файл, переданный в качестве параметра;

`string SaveToDataSet(System.Data.DataSet d)` – производит сохранение в набор данных для последующей записи в файл проекта;

`void Open(string fileName)` – производит восстановление состояния из файла, переданного в качестве параметра;

`void Open(System.Data.DataSet data)` – производит восстановление состояния из набора данных, переданного в качестве параметра;

`KmtpTreeManagement.WindowsHandler GetWindowsHandler()` – возвращает экземпляр класса `WindowsHandler`, осуществляющего обработку дочерних окон проекта. В случае если проект не обладает дочерними окнами или не производит их централизованную обработку, возвращается null;

`string ProjectName` – возвращает имя проекта;

`string ProjectFileName` – возвращает имя файла проекта;

`string ProjectPath` – возвращает путь к файлу проекта;

`Guid GUID` – возвращает глобально-уникальный идентификатор (GUID) проекта;

`ITSHostApp TSHostApp` – возвращает указатель на интерфейс родительского объекта.

Кроме интерфейса **ITSPProject** необходимо реализовать интерфейс **ITreeElement** пространства имен **KmtpTreeManagement** для организации взаимодействия с пользователем. Это дает возможность добавлять элементы проекта в дерево проектов приложения, реагировать на действия пользователя, производить обмен данными с другими проектами посредством технологии drag&drop.

Для расширения функциональности нейросетей было принято решение вынести реализацию активационной функции в отдельный класс и реализовать его в качестве дополнительной библиотеки. Это позволит в будущем добавлять новые активационные функции без необходи-

мости перекомпиляции всего приложения. Методы интерфейса, реализуемого классом активационной функции, выглядят следующим образом:

`double ActivFn(double net)` – возвращает результат применения активационной функции к значению `net`;

`double ActivFnDeriv(double net)` – возвращает результат применения производной активационной функции к значению `net`;

`string Description` – возвращает описание текущей активационной функции;

`double Normalize(double min, double max, double value)` – возвращает нормализованное значение параметра `value`, изменяемого в пределах от `min` до `max`;

`double Denormalize(double min, double max, double value)` – возвращает денормализованное значение параметра `value`, изменяемого в пределах от `min` до `max`.

Внешний вид приложения KMTPFramework с используемыми модулями нейросети и нейропроекта показан на рис. 3.

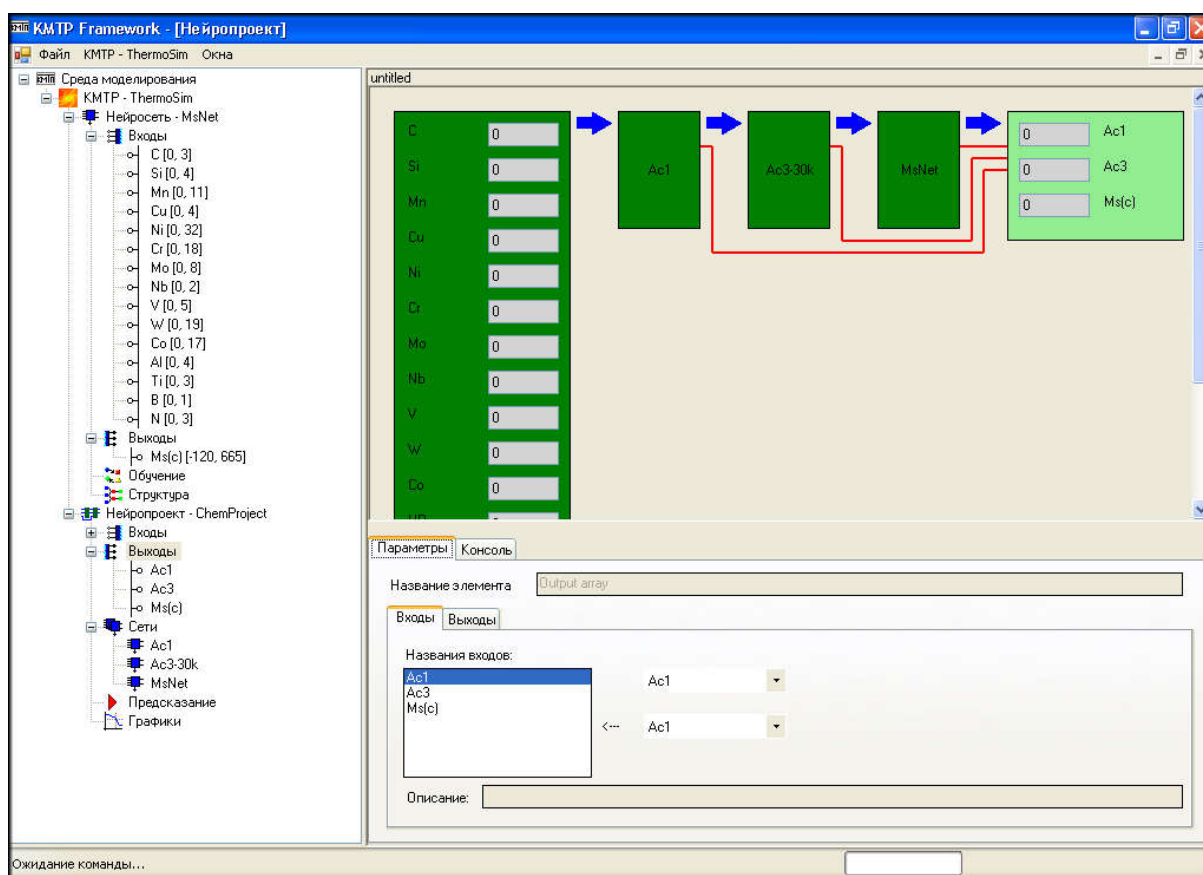


Рис. 3. Среда моделирования KMTPFramework с модулями нейросети и нейропроекта, реализующими прогнозирование точек Ac1, Ac3 и Ms

#### 4. Результаты обучения и применения нейронных сетей

Результаты обучения анализировались с применением стандартных средств KMTPFramework. На рис. 4 показан результат верификации нейросети, полученный путем прохода по набору тестовых значений входов. Учитывая, что невозможно наглядно отобразить многомерную зависимость выходных значений нейросетей от 13 (для Ac1 и Ac3) и 15 (для Ms) входов, по оси X откладывается порядковый номер тестовой строки в файле, а по оси Y – эталонный и полученный ответы. Наборы сортируются в порядке уменьшения эталонного ответа для большей наглядности результатов. На графике выделен диапазон 10 %-й погрешности. Видно, что нейросеть в подавляющем большинстве случаев дает погрешность прогнозирования на уровне 5–7 %. Аналогичные результаты могут быть получены и для остальных выходов (Ac3, Ms).

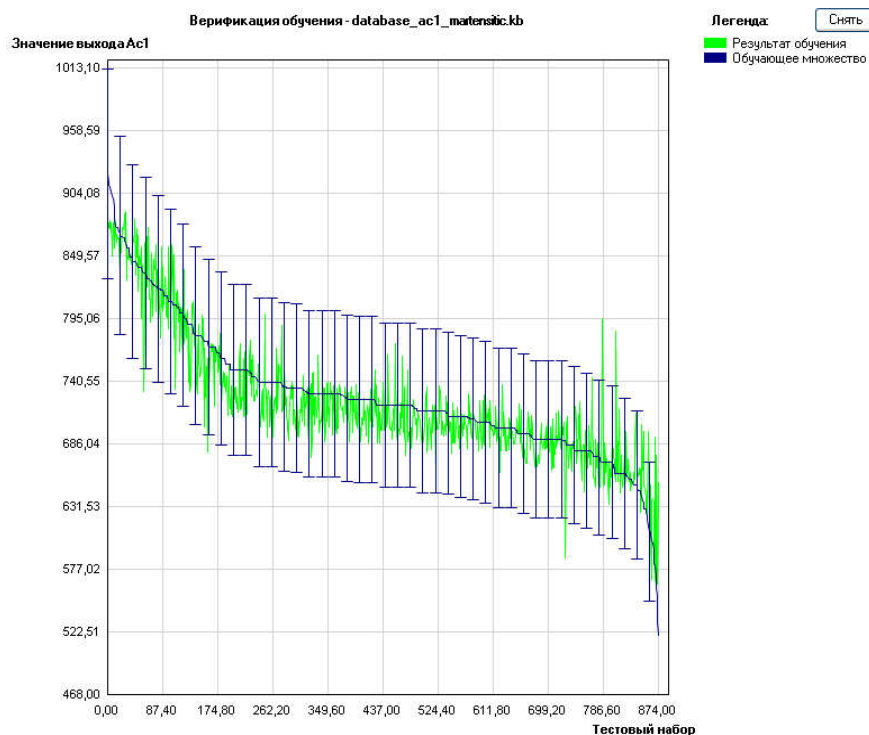


Рис. 4. Результаты верификации нейронной сети на примере прогнозирования значений точки Ас1

Нейросеть предоставляет возможность анализа влияния различных элементов на изменение выходных величин. На рис. 5 показана зависимость значения точки Ас3 от содержания углерода (откладывается по оси X) и магния (отображается цветом). Видно, что с увеличением содержания магния значение точки Ас3 уменьшается, при этом сохраняется, но ослабляется обратная зависимость от содержания углерода.

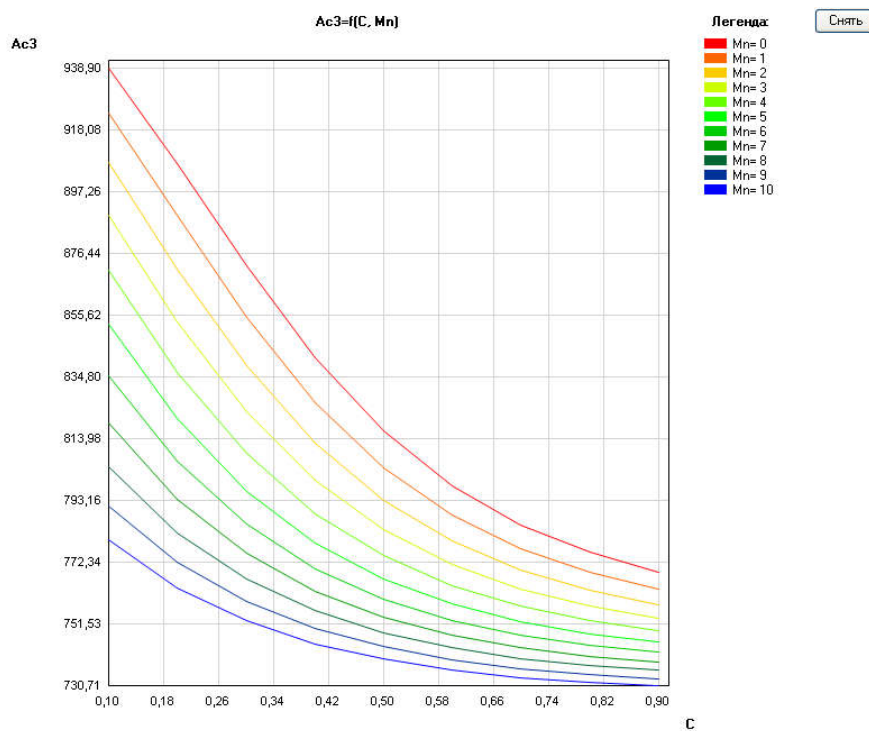


Рис. 5. Зависимость значения точки Ас3 от содержания углерода и магния

### Заключение

Разработанная нейронная сеть позволяет с приемлемой для практического применения погрешностью (5–7 %) осуществлять прогнозирование критических точек на диаграмме фазовых переходов в сталях при их термической обработке в зависимости от химического состава сплава и параметров технологического процесса. Применение нейросетей при разработке и оптимизации процессов термообработки позволит существенно сократить время и расход материальных средств на техническую подготовку производства.

Следует отметить, что использование большого количества данных в качестве обучающих множеств позволяет в значительной степени уменьшить погрешность последующего прогнозирования характерных точек диаграммы фазовых превращений в сталях. В то же время существует ряд направлений дальнейшего совершенствования применяемого подхода, что позволит, с одной стороны, повысить точность обучения, а с другой – сохранить производительность на этапе обучения. Одной из таких мер должно стать внедрение алгоритма адаптации скорости обучения (параметра  $\alpha$ ) в зависимости от изменения ошибки обучения. В результате нейросеть будет более гибко реагировать на попадание в локальный минимум и увеличит скорость обучения в случае линейных или близких к ним зависимостей.

Вторым направлением усовершенствований может стать применение Байесовых нейронных сетей. Это будет более актуальным в случае наличия недостаточно подробных обучающих множеств, когда несколько функций способны с достаточной точностью аппроксимировать заданные зависимости.

Следующий подход, который способен значительно повысить точность прогнозирования, представляет собой применение алгоритма послойного наращивания нейронов в нейросети, т. е. на этапе обучения происходит адаптация самой структуры нейросети, изменяется количество нейронов в слоях, количество слоев (как в сторону увеличения, так и в сторону уменьшения), что обеспечит в конечном итоге достижение минимально возможной погрешности прогнозирования. Недостатком этого подхода является значительное увеличение времени обучения, однако, учитывая повышение эффективности последующей работы нейросети, применение указанного алгоритма является целесообразным.

В качестве еще одного актуального метода оптимизации алгоритмов обучения и работы нейросетей можно отметить применение разных активационных функций у разных нейронов либо в пределах одного слоя, либо для соседних слоев. В результате нейронная сеть приобретет способность адаптации к поставленной задаче, подбора наиболее «выгодных» активационных функций путем увеличения либо уменьшения значений весов у соответствующих нейронов.

Стоит заметить, что уже разработанная нейронная сеть позволяет с приемлемой погрешностью осуществлять прогнозирование значений критических точек фазовых переходов в сталях при их термической обработке в зависимости от химического состава сплава и параметров технологического процесса. Ее практическое использование даст возможность повышать эффективность применения информационных технологий в термическом производстве, поэтому дальнейшее совершенствование этой разработки представляет как научный, так и практический интерес.

### Список литературы

1. Bhadeshia, H.K.D.H. Neural Networks in Materials Science / H.K.D.H. Bhadeshia // ISIJ International. – Vol. 39. – 1999. – P. 966–979.
2. Gavard, L. Bayesian neural network model for austenite formation in steels / L. Gavard, H.K.D.H. Bhadeshia, D.J.C. MacKay // Materials Science and Technology. – Vol. 12. – June 1996. – P. 453–462.
3. Новоселова, Н.А. Построение нечеткой нейросетевой модели для решения задач классификации / Н.А. Новоселова // Информатика. – 2006. – № 3(11). – С. 5–14.
4. Ваткин, М.Е. Многослойная нечеткая нейронная сеть для классификации объектов на полутоновых изображениях / М.Е. Ваткин, А.А. Дудкин // Информатика. – 2005. – № 3(7). – С. 60–70.



5. Тухватулин, И.Х. Прогноз свойств металлических сплавов с помощью нейронных сетей / И.Х. Тухватулин, Ю.П. Ланкин [Электронный ресурс]. – Режим доступа: [http://library.mephi.ru/data/scientific-sessions/2000/Neuro\\_2/106.html](http://library.mephi.ru/data/scientific-sessions/2000/Neuro_2/106.html).
6. Катковский, Е.А. Методы нейронных сетей для расчетов на ЭВМ тепловых, гидродинамических и физико-химических процессов / Е.А. Катковский [Электронный ресурс]. – Режим доступа: [http://www.neuroset.ru/index.php?option=com\\_docman&task=doc\\_details&gid=71&Itemid=70](http://www.neuroset.ru/index.php?option=com_docman&task=doc_details&gid=71&Itemid=70).
7. MacKay, D.J.C. Neural Computation / D.J.C. MacKay. – England, 1992. – Vol. 4. – 415 p.
8. Capdevilla, C. ISIJ International / C. Capdevilla, F.G. Caballero, C.G. de Andres. – England, 2002. – Vol. 42. – P. 894–902.
9. Thomas Sourmail: Home page [Electronic resource]. – Mode of access: [www.thomas-sourmail.org](http://www.thomas-sourmail.org).
10. Gibbs, M. Materials Algorithms Project Program Library / M. Gibbs, D. MacKay [Electronic resource]. – Mode of access: <http://www.msm.cam.ac.uk/map/steel/programs/acpros-e.html>.
11. Архитектура программных комплексов для моделирования технологических систем на основе динамически заменяемых модулей / С.П. Кундас [и др.] // Вестник компьютерных и информационных технологий. – 2005. – № 7. – С. 14–18.

Поступила 23.01.07

*Белорусский государственный университет  
информатики и радиоэлектроники,  
Минск, П. Бровки, 6  
e-mail: [adminset@bsuir.unibel.by](mailto:adminset@bsuir.unibel.by)*

**A.V. Lemzиков, S.P. Kundas, V.V. Kutsylo, D.G. Ivanov**

**ARTIFICIAL NEURAL NETWORKS IMPLEMENTATION  
FOR PREDICTION OF CHARACTERISTIC POINTS OF PHASE  
TRANSFORMATIONS IN STEELS WITH A WIDE RANGE  
OF CHEMICAL COMPOSITION**

Principles of the development of neural networks for application in the material science, in particular for prediction of characteristic points of phase transformations in steels according to their chemical composition and parameters of technological process were considered. Architecture and software implementation of the neural networks as a part of author's software KMTPFramework were presented and discussed.