# Содержание:

1. **История развития науки о искусственном интеллекте.**
2. **Описание нейронных сетей.**
	1. Цель классификации
	2. Использование нейронных сетей в качестве классификатора.
	3. Подготовка исходных данных.
	4. Кодирование выходных значений.
	5. Выбор объема сети.
	6. Выбор архитектуры сети.
	7. Алгоритм построения классификатора на основе нейронных сетей.
3. **Прогнозирование объёма продаж кондитерских изделий с помощью нейронных сетей.**
	1. Постановка задачи.
	2. Метод решения.
	3. Результат.
4. **Вывод.**

## История развития науки о искусственном интеллекте.

Искусственный интеллект - одна из новейших наук, появившихся во второй половине 20-го века на базе вычислительной техники, математической логики, программирования, психологии, лингвистики, нейрофизиологии и других отраслей знаний. Искусственный интеллект - это образец междисциплинарных исследований, где соединяются профессиональные интересы специалистов разного профиля. Само название новой науки возникло в конце 60-х годах, а в 1969 г. в Вашингтоне (США) состоялась первая Всемирная конференция по искусственному интеллекту. Известно, что совокупность научных исследований обретает права науки, если выполнены два необходимых условия. У этих исследований должен быть объект изучения, не совпадающий с теми, которые изучают другие науки. И должны существовать специфические методы исследования этого объекта, отличные от методов других, уже сложившихся наук. Исследования, которые объединяются сейчас термином "искусственный интеллект", имеют свой специфический объект изучения и свои специфические методы. В этой статье мы обоснуем это утверждение. Когда в конце 40-х - начале 50-х годов появились ЭВМ, стало ясно, что инженеры и математики создали не просто быстро работающее устройство для вычислений, а нечто более значительное. Оказалось, что с помощью ЭВМ можно решать различные головоломки, логические задачи, играть в шахматы, создавать игровые программы. ЭВМ стали принимать участие в творческих процессах: сочинять музыкальные мелодии, стихотворения и даже сказки. Появились программы для перевода с одного языка на другой, для распознавания образов, доказательства теорем. Это свидетельствовало о том, что с помощью ЭВМ и соответствующих программ можно автоматизировать такие виды человеческой деятельности, которые называются интеллектуальными и считаются доступными лишь человеку. Несмотря на большое разнообразие невычислительных программ, созданных к началу 60-х годов, программирование в сфере интеллектуальной деятельности находилось в гораздо худшем положении, чем решение расчетных задач. Причина очевидна. Программирование для задач расчетного характера опиралось на соответствующую теорию - вычислительную математику. На основе этой теории было разработано много методов решения задач. Эти методы стали основой для соответствующих программ. Ничего подобного для невычислительных задач не было. Любая программа была здесь уникальной, как произведение искусства. Опыт создания таких программ никак не обобщался, умение их создавать не формализовалось. Никто не станет отрицать, что, в отличие от искусства, у науки должны быть методы решения задач. С помощью этих методов все однотипные задачи должны решаться единообразным способом. И "набив руку" на решении задач определенного типа, легко решать новые задачи, относящиеся к тому же типу. Но именно таких методов и не смогли придумать те, кто создавал первые программы невычислительного характера. Когда программист создавал программу дл игры в шахматы, то он использовал собственны знания о процессе игры. Он вкладывал их в программу, а компьютер лишь механически выполняли эту программу. Можно сказать, что компьютер "не отличал" вычислительные программы от невычислительных. Он одинаковым образом находил корни квадратного уравнения или писал стихи. В памяти компьютера не было знаний о том, что он на самом деле делает. Об интеллекте компьютера можно было бы говорить, если бы он сам, на основании собственных знаний о том, как протекает игра в шахматы и как играют в эту игру люди, сумел составить шахматную программу или синтезировал программу для написания несложных вальсов и маршей. Не сами процедуры, с помощью которых выполняется та или иная интеллектуальная деятельность, а понимание того, как их создать, как научиться новому виду интеллектуальной деятельности, - вот где скрыто то, что можно назвать интеллектом. Специальные метапроцедуры обучения новым видам интеллектуальной деятельности отличают человека от компьютера. Следовательно, в создании искусственного интеллекта основной задачей становится реализация машинными средствами тех метапроцедур, которые используются в интеллектуальной деятельности человека. Что же это за процедуры? В психологии мышления есть несколько моделей творческой деятельности. Одна из них называется лабиринтной. Суть лабиринтной гипотезы, на которой основана лабиринтная модель, состоит в следующем: переход от исходных данных задачи к решению лежит через лабиринт возможных альтернативных путей. Не все пути ведут к желаемой цели, многие из них заводят в тупик, надо уметь возвращаться к тому месту, где потеряно правильное направление. Это напоминает попытки не слишком умелого школьника решить задачу об упрощении алгебраических выражений. Для этой цели на каждом шагу можно применять некоторые стандартные преобразования или придумывать искусственные приемы. Но весьма часто вместо упрощения выражения происходит его усложнение, и возникают тупики, из которых нет выхода. По мнению сторонников лабиринтной модели мышления, решение всякой творческой задачи сводится к целенаправленному поиску в лабиринте альтернативных путей с оценкой успеха после каждого шага. С лабиринтной моделью связана первая из метапроцедур - целенаправленный поиск в лабиринте возможностей. Программированию этой метапроцедуры соответствуют многочисленные процедуры поиска, основанные на соображениях здравого смысла (человеческого опыта решения аналогичных задач). В 60-х годах было создано немало программ на основе лабиринтной модели, в основном игровых и доказывающих теоремы "в лоб", без привлечения искусственных приемов. Соответствующее направление в программировании получило название эвристического программирования. Высказывались даже предположения, что целенаправленный поиск в лабиринте возможностей - универсальная процедура, пригодная для решения любых интеллектуальных задач. Но исследователи отказались от этой идеи, когда столкнулись с задачами, в которых лабиринта возможностей либо не существовало, либо он был слишком велик для метапроцедуры поиска, как, например, при игре в шахматы. Конечно, в этой игре лабиринт возможностей - это все мыслимые партии игры. Но как в этом астрономически большом лабиринте найти те партии, которые ведут к выигрышу? Лабиринт столь велик, что никакие мыслимые скорости вычислений не позволят целенаправленно перебрать пути в нем. И все попытки использовать для этого человеческие эвристики (в данном случае профессиональный опыт шахматистов) не дают пути решения задачи. Поэтому созданные шахматные программы уже давно используют не только метапроцедуру целенаправленного поиска, но и другие метапроцедуры, связанные с другими моделями мышления. Долгие годы в психологии изучалась ассоциативная модель мышления. Основной метапроцедурой модели является ассоциативный поиск и ассоциативное рассуждение. Предполагается, что решение неизвестной задачи так или иначе основывается на уже решенных задачах, чем-то похожих на ту, которую надо решить. Новая задача рассматривается как уже известная, хотя и несколько отличающаяся от известной. Поэтому способ ее решения должен быть близок к тому, который когда-то помог решить подобную задачу. Для этого надо обратиться к памяти и попытаться найти нечто похожее, что ранее уже встречалось. Это и есть ассоциативный поиск. Когда, увидев незнакомого человека, вы стараетесь вспомнить, на кого он похож, реализуется метапроцедура ассоциативного поиска. Но понятие ассоциации в психологии шире, чем просто "похожесть". Ассоциативные связи могут возникнуть и по контрасту, как противопоставление одного другому, и по смежности, т. е. в силу того, что некоторые явления возникали в рамках одной и той же ситуации или происходили одновременно (или с небольшим сдвигом по времени). Ассоциативное рассуждение позволяет переносить приемы, использованные ранее, на текущую ситуацию. К сожалению, несмотря на многолетнее изучение ассоциативной модели, не удалось создать стройную теорию ассоциативного поиска и ассоциативного рассуждения. Исключение составляет важный, но частный класс ассоциаций, называемых условными рефлексами. И все же метапроцедура ассоциативного поиска и рассуждения сыграла важную роль: она помогла создать эффективные программы в распознавании образов, в классификационных задачах и в обучении ЭВМ.

 Но одновременно эта метапроцедура привела к мысли о том, что для ее эффективного использования надо привлечь результаты, полученные в другой модели мышления, опирающейся на идею внутреннего представления проблемной области, на знания о ее особенностях, закономерностях и процедурах действия в ней. Это представление о мыслительной деятельности человека обычно называют модельной гипотезой. Согласно ей, мозг человека содержит модель проблемной ситуации, в которой ему надо принять решение. Для решения используются метапроцедуры, оперирующие с совокупностью знаний из той проблемной области, к которой принадлежит данная проблемная ситуация. Например, если проблемная ситуация- переход через улицу с интенсивным движением, то знания, которые могут помочь ее разрешить, касаются способов организации движения транспорта, сигналов светофоров, наличия дорожек для перехода и т. п. В модельной гипотезе основными метапроцедурами становятся представление знаний, рассуждения, поиск релевантной (связанной с данной проблемной ситуацией) информации в совокупности имеющихся знаний, их пополнение и корректировка.

Эти метапроцедуры составляют ядро интеллектуальных возможностей современных программ и программных систем, ориентированных на решение творческих задач. В совокупности с метапроцедурами целенаправленного поиска в лабиринте возможностей, ассоциативного поиска и рассуждения они образуют арсенал интеллектуальных средств, которым располагают современные интеллектуальные системы, часто называемые системами, основанными на знаниях. Можно сформулировать основные цели и задачи искусственного интеллекта. Объектом изучения искусственного интеллекта являются метапроцедуры, используемые при решении человеком задач, традиционно называемых интеллектуальными, или творческими. Но если психология мышления изучает эти метапроцедуры применительно к человеку, то искусственный интеллект создает программные (а сейчас уже и программно-аппаратные) модели таких метапроцедур. Цель исследований в области искусственного интеллекта - создание арсенала метапроцедур, достаточного для того, чтобы ЭВМ (или другие технические системы, например роботы) могли находить по постановкам задач их решения. Иными словами, стали автономными программистами, способными выполнять работу профессиональных программистов- прикладников (создающих программы для решения задач в определенной предметной области). Разумеется, сформулированная цель не исчерпывает всех задач, которые ставит перед собой искусственный интеллект. Это цель ближайшая. Последующие цели связаны с попыткой проникнуть в области мышления человека, которые лежат вне сферы рационального и выразимого словесно (вербально) мышления. Ибо в поиске решения многих задач, особенно сильно отличающихся от ранее решенных, большую роль играет та сфера мышления, которую называют подсознательной, бессознательной, или интуитивной. Основными методами, используемыми в искусственном интеллекте, являются разного рода программные модели и средства, эксперимент на ЭВМ и теоретические модели. Однако современные ЭВМ уже мало удовлетворяют специалистов по искусственному интеллекту. Они не имеют ничего общего с тем, как устроен человеческий мозг. Поэтому идет интенсивный поиск новых технических структур, способных лучше решать задачи, связанные с интеллектуальными процессами. Сюда относятся исследования по нейроподобным искусственным сетям, попытки построить молекулярные машины, работы в области голографических систем и многое другое. Более подробно здесь рассматриваются нейронные искусственные сети.

## Нейронные сети

Решение задачи классификации является одним из важнейших применений нейронных сетей.

Задача классификации представляет собой задачу отнесения образца к одному из нескольких попарно не пересекающихся множеств. Примером таких задач может быть, например, задача определения кредитоспособности клиента банка, медицинские задачи, в которых необходимо определить, например, исход заболевания, решение задач управления портфелем ценных бумаг (продать купить или "придержать" акции в зависимости от ситуации на рынке), задача определения жизнеспособных и склонных к банкротству фирм.

## 1. Цель классификации

При решении задач классификации необходимо отнести имеющиеся *статические образцы* (характеристики ситуации на рынке, данные медосмотра, информация о клиенте) к *определенным классам.* Возможно несколько способов представления данных. Наиболее распространенным является способ, при котором образец представляется вектором. Компоненты этого вектора представляют собой различные характеристики образца, которые влияют на принятие решения о том, к какому классу можно отнести данный образец. Например, для медицинских задач в качестве компонентов этого вектора могут быть данные из медицинской карты больного. Таким образом, на основании некоторой информации о примере, необходимо определить, к какому классу его можно отнести. Классификатор таким образом относит объект к одному из классов в соответствии с определенным разбиением N-мерного пространства, которое называется *пространством входов,* и размерность этого пространства является количеством компонент вектора.

Прежде всего, нужно определить уровень сложности системы. В реальных задачах часто возникает ситуация, когда количество образцов ограничено, что осложняет определение сложности задачи. Возможно выделить три основных уровня сложности. Первый (самый простой) – когда классы можно разделить прямыми линиями (или гиперплоскостями, если пространство входов имеет размерность больше двух) – так называемая *линейная разделимость*. Во втором случае классы невозможно разделить линиями (плоскостями), но их возможно отделить с помощью более сложного деления – *нелинейная разделимость*. В третьем случае классы пересекаются и можно говорить только о *вероятностной разделимости*.

В идеальном варианте после предварительной обработки мы должны получить линейно разделимую задачу, так как после этого значительно упрощается построение классификатора. К сожалению, при решении реальных задач мы имеем ограниченное количество образцов, на основании которых и производится построение классификатора. При этом мы не можем провести такую предобработку данных, при которой будет достигнута линейная разделимость образцов.

## 2. Использование нейронных сетей в качестве классификатора.

Сети с прямой связью являются универсальным средством аппроксимации функций, что позволяет их использовать в решении задач классификации. Как правило, нейронные сети оказываются наиболее эффективным способом классификации, потому что генерируют фактически большое число регрессионных моделей (которые используются в решении задач классификации статистическими методами).

К сожалению, в применении нейронных сетей в практических задачах возникает ряд проблем. Во-первых, заранее не известно, какой сложности (размера) может потребоваться сеть для достаточно точной реализации отображения. Эта сложность может оказаться чрезмерно высокой, что потребует сложной архитектуры сетей. Так Минский в своей работе "Персептроны" доказал, что простейшие однослойные нейронные сети способны решать только линейно разделимые задачи. Это ограничение преодолимо при использовании многослойных нейронных сетей. В общем виде можно сказать, что в сети с одним скрытым слоем вектор, соответствующий входному образцу, преобразуется скрытым слоем в некоторое новое пространство, которое может иметь другую размерность, а затем гиперплоскости, соответствующие нейронам выходного слоя, разделяют его на классы. Таким образом сеть распознает не только характеристики исходных данных, но и "характеристики характеристик", сформированные скрытым слоем.

## 3. Подготовка исходных данных

Для построения классификатора необходимо определить, какие параметры влияют на принятие решения о том, к какому классу принадлежит образец. При этом могут возникнуть две проблемы. Во-первых, если количество параметров мало, то может возникнуть ситуация, при которой один и тот же набор исходных данных соответствует примерам, находящимся в разных классах. Тогда невозможно обучить нейронную сеть, и система не будет корректно работать (невозможно найти минимум, который соответствует такому набору исходных данных). *Исходные данные обязательно должны быть непротиворечивы*. Для решения этой проблемы необходимо увеличить размерность пространства признаков (количество компонент входного вектора, соответствующего образцу). Но при увеличении размерности пространства признаков может возникнуть ситуация, когда число примеров может стать недостаточным для обучения сети, и она вместо обобщения просто запомнит примеры из обучающей выборки и не сможет корректно функционировать. Таким образом, при определении признаков необходимо найти компромисс с их количеством.

Далее необходимо определить способ представления входных данных для нейронной сети, т.е. определить способ нормирования. Нормировка необходима, поскольку нейронные сети работают с данными, представленными числами в диапазоне 0..1, а исходные данные могут иметь произвольный диапазон или вообще быть нечисловыми данными. При этом возможны различные способы, начиная от простого линейного преобразования в требуемый диапазон и заканчивая многомерным анализом параметров и нелинейной нормировкой в зависимости от влияния параметров друг на друга.

## 4. Кодирование выходных значений.

Задача классификации при наличии двух классов может быть решена на сети с одним нейроном в выходном слое, который может принимать одно из двух значений 0 или 1, в зависимости от того, к какому классу принадлежит образец. При наличии нескольких классов возникает проблема, связанная с представлением этих данных для выхода сети. Наиболее простым способом представления выходных данных в таком случае является вектор, компоненты которого соответствуют различным номерам классов. При этом i-я компонента вектора соответствует i-му классу. Все остальные компоненты при этом устанавливаются в 0. Тогда, например, второму классу будет соответствовать 1 на 2 выходе сети и 0 на остальных. При интерпретации результата обычно считается, что номер класса определяется номером выхода сети, на котором появилось максимальное значение. Например, если в сети с тремя выходами мы имеем вектор выходных значений (0.2,0.6,0.4), то мы видим, что максимальное значение имеет вторая компонента вектора, значит класс, к которому относится этот пример, – 2. При таком способе кодирования иногда вводится также понятие *уверенности* сети в том, что пример относится к этому классу. Наиболее простой способ определения уверенности заключается в определении разности между максимальным значением выхода и значением другого выхода, которое является ближайшим к максимальному. Например, для рассмотренного выше примера уверенность сети в том, что пример относится ко второму классу, определится как разность между второй и третьей компонентой вектора и равна 0.6-0.4=0.2. Соответственно чем выше уверенность, тем больше вероятность того, что сеть дала правильный ответ. Этот метод кодирования является самым простым, но не всегда самым оптимальным способом представления данных.

Известны и другие способы. Например, выходной вектор представляет собой номер кластера, записанный в двоичной форме. Тогда при наличии 8 классов нам потребуется вектор из 3 элементов, и, скажем, 3 классу будет соответствовать вектор 011. Но при этом в случае получения неверного значения на одном из выходов мы можем получить неверную классификацию (неверный номер кластера), поэтому имеет смысл увеличить расстояние между двумя кластерами за счет использования кодирования выхода по коду Хемминга, который повысит надежность классификации.

Другой подход состоит в разбиении задачи с k классами на k\*(k-1)/2 подзадач с двумя классами (2 на 2 кодирование) каждая. Под подзадачей в данном случае понимается то, что сеть определяет наличие одной из компонент вектора. Т.е. исходный вектор разбивается на группы по два компонента в каждой таким образом, чтобы в них вошли все возможные комбинации компонент выходного вектора. Число этих групп можно определить как количество неупорядоченных выборок по два из исходных компонент. Из комбинаторики

Тогда, например, для задачи с четырьмя классами мы имеем 6 выходов (подзадач) распределенных следующим образом:

|  |  |
| --- | --- |
| № подзадачи(выхода) | КомпонентыВыхода |
| 1 | 1-2 |
| 2 | 1-3 |
| 3 | 1-4 |
| 4 | 2-3 |
| 5 | 2-4 |
| 6 | 3-4 |

Где 1 на выходе говорит о наличии одной из компонент. Тогда мы можем перейти к номеру класса по результату расчета сетью следующим образом: определяем, какие комбинации получили единичное (точнее близкое к единице) значение выхода (т.е. какие подзадачи у нас активировались), и считаем, что номер класса будет тот, который вошел в наибольшее количество активированных подзадач (см. таблицу).

|  |  |
| --- | --- |
| № класса | Акт. Выходы |
| 1 | 1,2,3 |
| 2 | 1,4,5 |
| 3 | 2,4,6 |
| 4 | 3,5,6 |

Это кодирование во многих задачах дает лучший результат, чем классический способ кодирование.

## 5. Выбор объема сети.

Правильный выбор объема сети имеет большое значение. Построить небольшую и качественную модель часто бывает просто невозможно, а большая модель будет просто запоминать примеры из обучающей выборки и не производить аппроксимацию, что, естественно, приведет к некорректной работе классификатора. Существуют два основных подхода к построению сети – конструктивный и деструктивный. При первом из них вначале берется сеть минимального размера, и постепенно увеличивают ее до достижения требуемой точности. При этом на каждом шаге ее заново обучают. Также существует так называемый метод каскадной корреляции, при котором после окончания эпохи происходит корректировка архитектуры сети с целью минимизации ошибки. При деструктивном подходе вначале берется сеть завышенного объема, и затем из нее удаляются узлы и связи, мало влияющие на решение. При этом полезно помнить следующее правило: *число примеров в обучающем множестве должно быть больше числа настраиваемых весов*. Иначе вместо обобщения сеть просто запомнит данные и утратит способность к классификации – результат будет неопределен для примеров, которые не вошли в обучающую выборку.

## 6. Выбор архитектуры сети.

При выборе архитектуры сети обычно опробуется несколько конфигураций с различным количеством элементов. При этом основным показателем является объем обучающего множества и обобщающая способность сети. Обычно используется алгоритм обучения Back Propagation (обратного распространения) с подтверждающим множеством.

## 7. Алгоритм построения классификатора на основе нейронных сетей.

1. Работа с данными
	1. Составить базу данных из примеров, характерных для данной задачи
	2. Разбить всю совокупность данных на два множества: обучающее и тестовое (возможно разбиение на 3 множества: обучающее, тестовое и подтверждающее).
2. Предварительная обработка
	1. Выбрать систему признаков, характерных для данной задачи, и преобразовать данные соответствующим образом для подачи на вход сети (нормировка, стандартизация и т.д.). В результате желательно получить линейно отделяемое пространство множества образцов.
	2. Выбрать систему кодирования выходных значений (классическое кодирование, 2 на 2 кодирование и т.д.)
3. Конструирование, обучение и оценка качества сети:
	1. Выбрать топологию сети: количество слоев, число нейронов в слоях и т.д.
	2. Выбрать функцию активации нейронов (например "сигмоида")
	3. Выбрать алгоритм обучения сети
	4. Оценить качество работы сети на основе подтверждающего множества или другому критерию, оптимизировать архитектуру (уменьшение весов, прореживание пространства признаков)
	5. Остановится на варианте сети, который обеспечивает наилучшую способность к обобщению и оценить качество работы по тестовому множеству.
4. Использование и диагностика
	1. Выяснить степень влияния различных факторов на принимаемое решение (эвристический подход).
	2. Убедится, что сеть дает требуемую точность классификации (число неправильно распознанных примеров мало)
5. При необходимости вернутся на этап 2, изменив способ представления образцов или изменив базу данных.
6. Практически использовать сеть для решения задачи.

## Прогнозирование объёма продаж кондитерских изделий с помощью нейронных сетей.

1. Постановка задачи

Объем продаж – один из ключевых показателей, характеризующих деятельность коммерческой фирмы. Поэтому задача прогнозирования объема продаж представляет собой большой интерес, например, для компаний, которые занимаются оптовой торговлей. Товароведам необходимо знать примерное количество продукции, которое они смогут реализовать в ближайшее время, для того, чтобы, с одной стороны, иметь достаточное количество товаров на складе, а с другой – не перегрузить склады продукцией, что особенно актуально, если продукция имеет небольшой срок хранения.

В большинстве случаев объем продаж того или иного товара поддается прогнозу. Например, многие товары продаются в соответствие с ярко выраженной сезонной составляющей, что легко определяется при помощи аналитических технологий. С их помощью можно прогнозировать объемы продаж по всем товарным позициям, что особенно актуально в случае их большого количества. При необходимости можно также учитывать и дополнительные факторы, например, рекламную компанию, конъюнктуру рынка, действия конкурентов и т.д. Комплексный учет всех факторов может значительно повысить качество прогноза.

2. Метод решения

Проиллюстрировать решение данной задачи мы сможем на примере прогнозирования объема продаж мармелада 'Лимонные дольки' на основе реальных данных компании, занимающейся оптовыми продажами кондитерских изделий. Прогнозирование объема продаж построим только на основе истории продаж по данной товарной позиции за определенный период. Эта информация собирается в базу данных, состоящую из двух колонок: дата и продажи в количественном выражении. В нашем случае история продаж разбита по неделям, соответственно, прогнозировать мы также будем на одну или несколько недель (исходные данные [здесь](http://www.basegroup.ru/images/practice/forecast/lemon_before.txt)).

Для получения качественного прогноза нам необходимо провести предварительную обработку данных при помощи программы [RawData Analyzer](http://www.basegroup.ru/deductor/rawdata.htm), входящей в состав пакета [Deductor](http://www.basegroup.ru/deductor/). Во-первых, данные по истории продаж следует сгладить, т.к. по зашумленным данным достаточно сложно установить зависимость изменения объема продаж. После сглаживания данных при помощи вейвлетов динамика изменений определяется и прогнозируется гораздо качественнее.

Пояснение к рисунку: тёмным цветом отображены реальные данные, светлым – сглаженные.

Во-вторых, для проведения прогнозирования структуру входных данных необходимо преобразовать по специальной схеме. Для этого выбирается глубина погружения, т.е. количество временных интервалов, по которым мы будем прогнозировать следующий. Возьмем глубину погружения равной 4, т.е. прогнозирование объема продаж на следующую неделю будет осуществляться по результатам четырех предыдущих недель (исходные данные [здесь](http://www.basegroup.ru/images/practice/forecast/lemon.txt)). Разумеется, и глубина погружения, и горизонт прогнозирования, т.е. количество прогнозируемых показателей, подбираются отдельно в каждой конкретной задаче. Далее следует преобразовать данные по продажам к следующему виду:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **smoothB3** | **smoothB2** | **smoothB1** | **smoothB0** | **smoothF1** |
| m-4 | m-3 | m-2 | m-1 | m |
| m-3 | m-2 | m-1 | m | m+1 |
| m-2 | m-1 | m | m+1 | m+2 |
| … | … | … | … | … |

Мы получаем так называемое 'скользящее окно', в котором представлены данные только за 5 недель. Первые 4 колонки – это данные за недели, на основе которых будем строить прогноз. Последняя колонка – показатель, который мы будем прогнозировать. Так как данных у нас больше, чем за 5 недель, мы можем сдвигать это окно по временной оси. Таким образом готовится обучающая выборка, и именно в таком виде представляются данные для последующего анализа.

Для решения поставленной задачи воспользуемся программой [Neural Analyzer](http://www.basegroup.ru/deductor/neural.htm), также входящей в состав пакета Deductor. Нейронная сеть не только способна установить зависимость изменения целевой переменной, которой в данном случае является количество проданного мармелада, но и позволит прогнозировать объем продаж на несколько недель вперед. После окончания процесса обучения на графике выходов сети можно заметить, что сеть достаточно точно моделирует поведение кривой. Однако, на последних неделях ошибка заметно увеличивается, причем тенденция в конце временного отрезка – место, которое нас больше всего интересует, была неверно угадана нейросетью.

Пояснение к рисунку: зелеными точками отображаются реальные показатели, красными – выход сети.

Получить хороший прогноз при помощи такой модели не удастся. Однако, это не означает, что нейронные сети не могут успешно решить поставленную задачу. Неудовлетворительное качество результата можно объяснить недостаточностью данных: для обучения сети мы использовали данные по продажам чуть более, чем за полгода. Получается, что нейросеть просто ничего не может знать обо всех сезонных зависимостях, например, о падении продаж в период летних отпусков.

Для повышения качества прогноза необходимо дополнить анализируемую информацию данными за аналогичный период прошлого года, также разбитыми на недели. Таким образом, внесем для обучения информацию о сезонности, и структура данных будет следующей:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Prev\_year** | **smoothB3** | **smoothB2** | **smoothB1** | **smoothB0** | **smoothF1** |
| Pm | m-4 | m-3 | m-2 | m-1 | m |
| Pm+1 | m-3 | m-2 | m-1 | m | m+1 |
| Pm+2 | m-2 | m-1 | m | m+1 | m+2 |
| … | … | … | … | … | … |

Pm, Pm+1, Pm+2 и т.д. – количество проданного товара за соответствующую неделю прошлого года. При таком подходе качество прогнозирования заметно улучшается.

Пояснение к рисунку: зелеными точками отображаются реальные показатели, красными – выход сети.

На рисунке видно, что прогноз на последние три недели весьма точен. В данном случае при прогнозировании учитываются оба фактора: прошлогодние продажи, служащие шаблоном для прогнозирования, и тренд, т.е. тенденция, которая сложилась в этом году. Именно это и обусловило качественный прогноз.

3. Результат

При помощи аналитических технологий мы решили задачу прогнозирования оптовых продаж мармелада. При помощи построенной нами системы на основе нейронных сетей мы имеем возможность строить краткосрочные и среднесрочные прогнозы. Кроме того, положительно на качестве прогноза может сказаться информация о внешней среде (курс доллара, рекламная поддержка и т.д.), а также категориальная переменная, обозначающая квартал или время года. Благодаря использованию нейронных сетей при прогнозировании, такого рода изменения в модели сводятся, фактически, к добавлению новых колонок в обучающую выборку и переобучению сети.

Главная проблема для качественного прогноза – наличие истории продаж за достаточно длительный срок и грамотная предобработка данных. Привлечение эксперта в этой области поможет дать ответ на вопрос, учитываются ли при анализе все факторы, влияющие на результат.

## Вывод.

 Существуют несколько основных проблем, изучаемых в искусственном интеллекте. Представление знаний - разработка методов и приемов для формализации и последующего ввода в память интеллектуальной системы знаний из различных проблемных областей, обобщение и классификация накопленных знаний, использование знаний при решении задач. Моделирование рассуждений - изучение и формализация различных схем человеческих умозаключений, используемых в процессе решения разнообразных задач, создание эффективных программ для реализации этих схем в вычислительных машинах. Диалоговые процедуры общения на естественном языке, обеспечивающие контакт между интеллектуальной системой и человеком- специалистом в процессе решения задач. Планирование целесообразной деятельности - разработка методов построения программ сложной деятельности на основании тех знаний о проблемной области, которые хранятся в интеллектуальной системе. Обучение интеллектуальных систем в процессе их деятельности, создание комплекса средств для накопления и обобщения умений и навыков, накапливаемых в таких системах. Кроме этих проблем исследуются многие другие, составляющие тот задел, на который будут опираться специалисты на следующем витке развития теории искусственного интеллекта. В практику человеческой деятельности интеллектуальные системы уже внедряются. Это и наиболее известные широкому кругу специалистов экспертные системы, передающие опыт более подготовленных специалистов менее подготовленным и интеллектуальные информационные системы (например, системы машинного перевода) и интеллектуальные роботы, другие системы, имеющие полное право называться интеллектуальными. Без таких систем современный научно- технический прогресс уже невозможен.