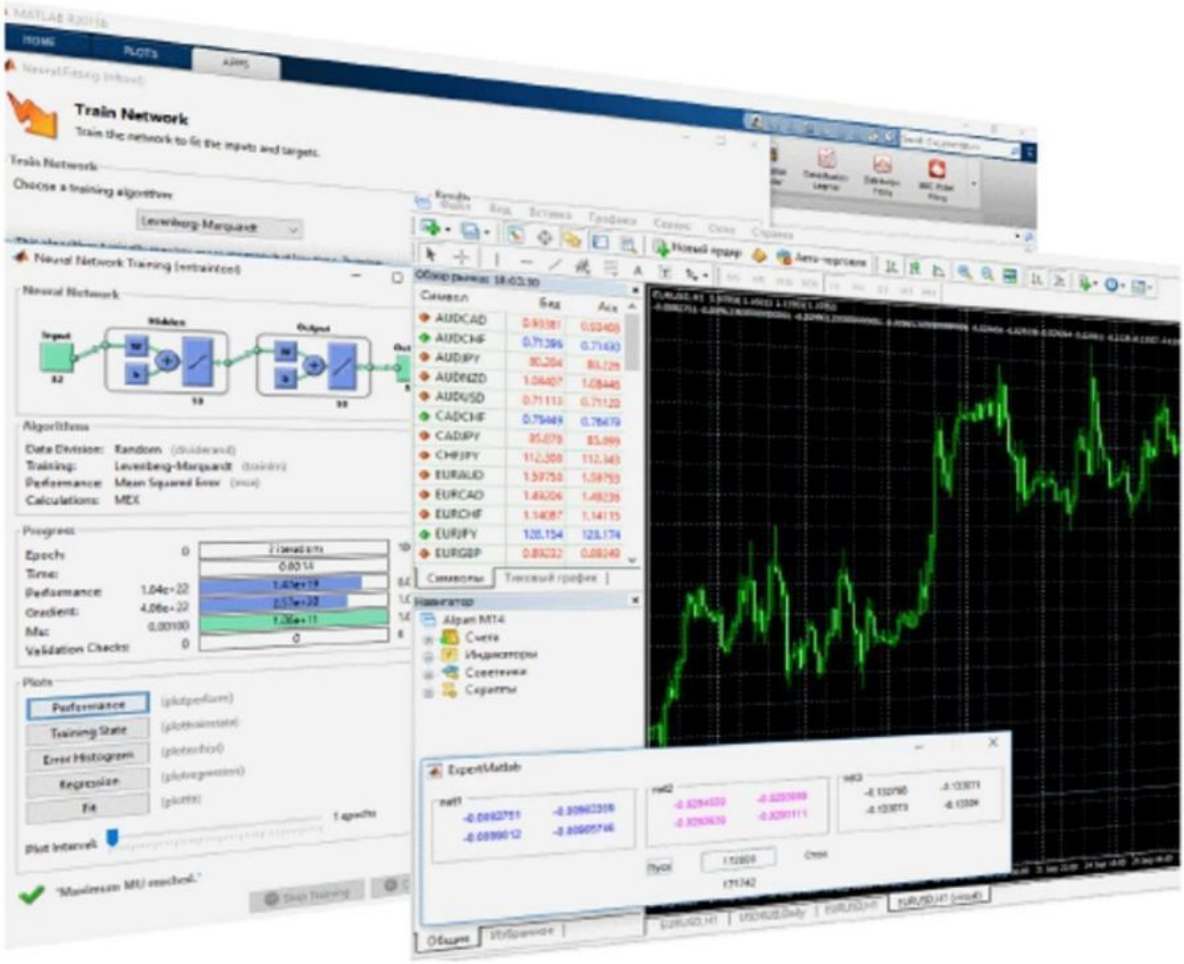
Андрей Дибров

НЕЙРОСЕТЕВАЯ ТОРГОВАЯ СИСТЕМА

ПОШАГОВАЯ РАЗРАБОТКА ДЛЯ ПЛАТФОРМЫ **Meta Trader 4** В СРЕДЕ **MATLAB**



- Гибкая торговая система
- Неограниченные возможности модернизации
- Творческий подход к разработке пользовательских систем
- Пошаговая инструкция
- Не требует глубоких знаний в прграммировании
- Комплекс новых пользовательских индикаторов

Андрей Дибров

Нейросетевая торговая система. Пошаговая разработка для платформы Meta Trader 4 в среде MATLAB. Сокращенное издание

«Издательские решения»

Дибров А.

Нейросетевая торговая система. Пошаговая разработка для платформы Meta Trader 4 в среде MATLAB. Сокращенное издание / А. Дибров — «Издательские решения»,

ISBN 978-5-44-938977-0

Книга для новичков и продвинутых трейдеров, желающих раздвинуть горизонты традиционного анализа рынка. Сокращенное описание пошаговой инструкции для разработки системы ведения автоматической торговли на финансовых рынках в терминале MT4, основанное на анализе рыночной ситуации нейронными сетями MATLAB. Неограниченные возможности модернизации, творческий подход, не требует глубоких знаний в программировании, не перегружает торговый комплекс, комплект новых пользовательских индикаторов.

ISBN 978-5-44-938977-0

© Дибров А.
© Издательские решения

Содержание

Предисловие 6
Логическое обоснование обучения нейросетей на принятие решения 7
Критические ошибки при разработке нейросетевой системы 22
Пользовательские технические индикаторы, используемые для обучения нейросети 33
StoxasticPolzMinTest 34
StoxasticTurnMin 36
WilliamsPolzMaxTest 37
WilliamsTurnMax 39
Конец ознакомительного фрагмента. 40

Нейросетевая торговая система. Пошаговая разработка для платформы Meta Trader 4 в среде MATLAB Сокращенное издание

Андрей Дибров

© Андрей Дибров, 2019

ISBN 978-5-4493-8977-0
Создано в интеллектуальной издательской системе Ridero

Предисловие

Важно! Данное издание представляет собой сокращенный вариант. Все разделы оглавления соответствую полному изданию, за исключением программных кодов основных скриптов и экспертов. Механизм работы с MATLAB+MT4 также включен в книгу. Для приобретения полного издания Вы можете связаться с автором по адресу электронной почты andreydib@yandex.ru.

Книга разрабатывалась с учетом интересов всех категорий трейдеров, а так же тех читателей, которые готовятся заняться трейдингом. В данной книге вы не найдете никакого теоретического материала по проблемам нейросетей и самого трейдинга. В любой литературе посвященной этим двум направлениям есть список трудов различных уважаемых авторов. Мне так же хотелось бы снабдить эту книгу таким же списком. Однако, увы, я этого не могу сделать, так как максимально постарался уйти от какой либо теории и психологии, которым в основном посвящена литература о трейдинге. Однако это не значит, что в свое время я не изучал подобную литературу и, что от нее нет пользы. Вот не полный список авторов, труды которых мною изучались – Чарльз Лебо и Дэвид В. Лукас, Юрий Жваколюк, Д. Ю. Пискулов, В. С. Сафонов, Шерри Де Ковни и Кристин Такки, Анна Эрлих, Александр Элдер, Джон Дж. Мэрфи. Но для понимания и практического применения материала представленного мною в принципе, достаточно теоретической информации, которая подается на сайтах дилинговых компаний и официального сайта Matlab. То есть, любой читатель имеющий представление о трейдинге может выполнить пошаговую инструкцию из моей книги и получить готовую автоматическую нейросетевую систему торговли. Причем, при кажущейся сложности системы в итоге вы придете к пониманию, что конечный результат в плане применения программных кодов поразительно легок, но в тоже время самодостачен и функционален. Ведь основная нагрузка в данной системе происходит при обучении нейросетей. Но и здесь, вникнув в процесс, вы обнаружите, что настроив систему, обучение не занимает много времени. Однако нам все равно придется немного пофилософствовать в следующем разделе на тему логического обоснования обучения нейросетей на принятие решения. От этого обоснования во многом зависит конечный результат.

Важно! Данная книга ориентирована на Matlab. Matlab не поставляется с этой книгой, прежде чем приступать к изучению и разработки автоматической нейросетевой системы торговли, вы должны приобрести ее отдельно и установить.

Также обращаю Ваше внимание на достоинствах нейросетевых систем как способов трейдинга. Мной разработано еще несколько систем, в том числе на основе платформы «NeuroSolutions» http://neurotrade.ucoz.com. Видео с примерами этой системы и системы представленной в этой книге также можно посмотреть по ссылкам https://youtu.be/5GwhRnSqT78, https://youtu.be/cIegQGJKbhY.

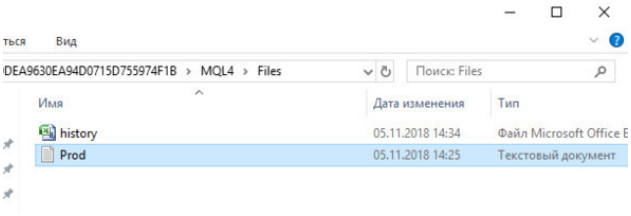
Логическое обоснование обучения нейросетей на принятие решения

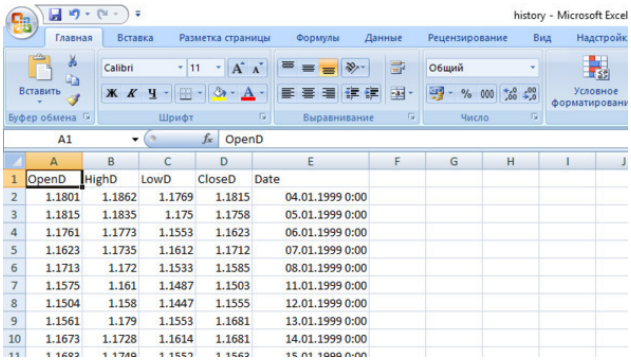
Прежде чем приступать к разработке любой торговой системы, мы задаемся вопросом – на каких принципах данная система будет функционировать? У нас есть два основополагающих принципа – использования флэтов и продолжение тенденции. Пока не будем рассматривать более узкие производные от них – внутри дневная торговля или нет, на фундаментальных данных, на новостях, на открытии рынков и т. д. Мне пришлось сталкиваться с описанием нейросетевых продуктов, где их авторы в примерах использования предлагали прогнозирование каких либо курсов – акций, валют и т. д. Приведем пример, используя платформу NeuroSolutions. Весь процесс повторять не обязательно, так как данную платформу мы в построении нашей системы использовать не будем, а я в данном случае использую ее как пример. Напишем коротенький скрипт для получения ценовых данных в MT4. Хочется обратить внимание на то, что при копировании программного кода из файла в формате PDF не сохраняется его стиль – все строки при переносе сохраняются без отступов. Так же могут быть скопированы номера страниц. Для текстовых редакторов эта проблема отсутствует.

```
//+ – — – — – — – — – — – — – — – — – — – — – — – +
//| History.mq4 |
//| Copyright © 2009, Andrey Dibrov. |
//+ – — – — – — – — – — – — – — – — – — – — – — – +
#property copyright «Copyright © 2009, Andrey Dibrov.»

int file=FileOpen («history. csv», FILE_CSV|FILE_READ|FILE_WRITE,»;»);
//+ – — – — – — – — – — – — – — – — – — – — – — – — – +
//| Script program start function |
//+ – — – — – — – — – — – — – — – — – — – — – — – — – +
void OnStart ()
{
// – —
FileWrite (file,«OpenD; HighD; LowD; CloseD; Date»);

if (file> 0)
{
Alert («Идет запись файла»);
for (int i=iBars (NULL,1440) -1; i> =0; i – )
{

FileWrite (file,

iOpen (NULL,1440,i),
iHigh (NULL,1440,i),
iLow (NULL,1440,i),
iClose (NULL,1440,i),
TimeToStr (iTime (NULL,1440,i)));
}
}
```

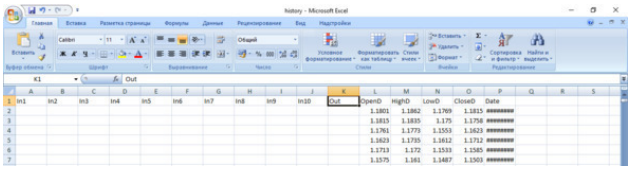
```
Alert («Файл записан»);
FileClose (file);
}
//+ – — – — – — – — – — – — – — – — – — – — – — – — – — – +
```

Запустив данный скрипт – в папке …MQL4/Filse каталога данных терминала, получим файл «history».

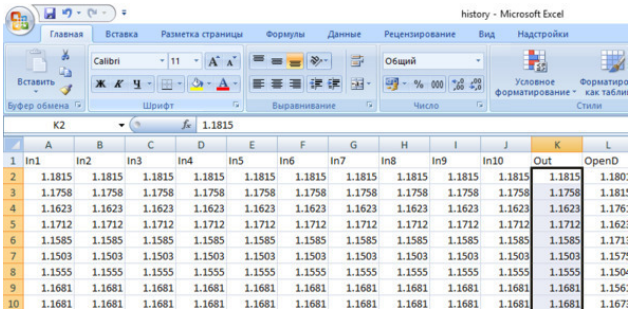




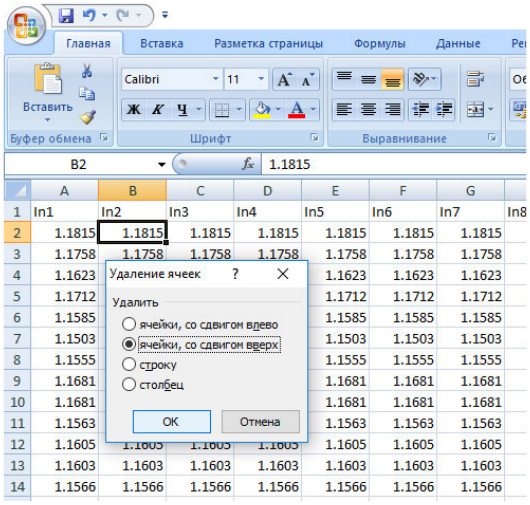
Откроем этот файл и добавим в начале десять столбцов In1—10 и один столбец Out.

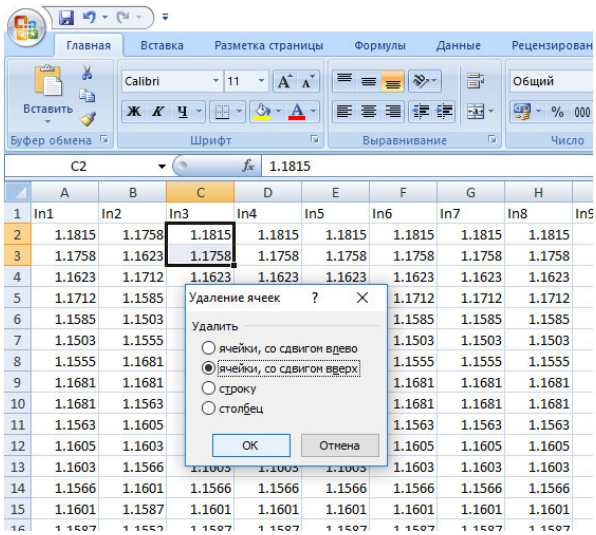


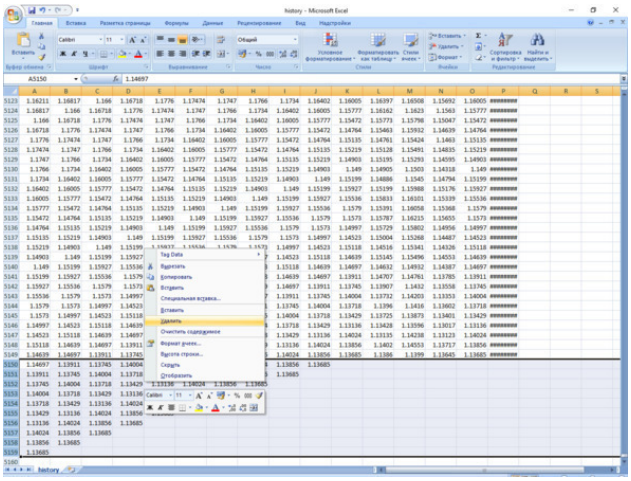
Заполним эти столбцы Данными из столбца CloseD. Как Вы уже поняли, это данные дневных закрытий.



Далее мы сдвинем эти данные в наших столбцах последовательно на одну ячейку вверх.

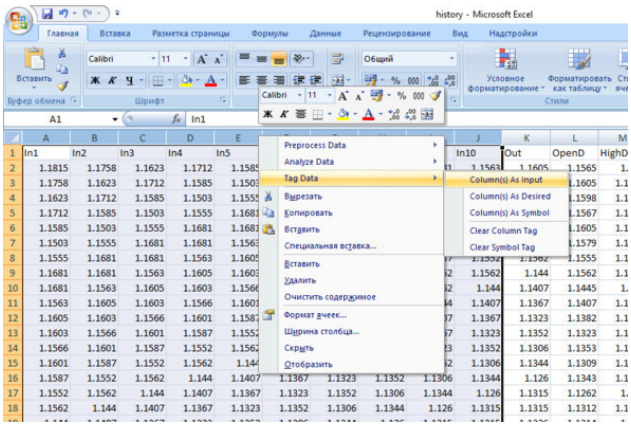




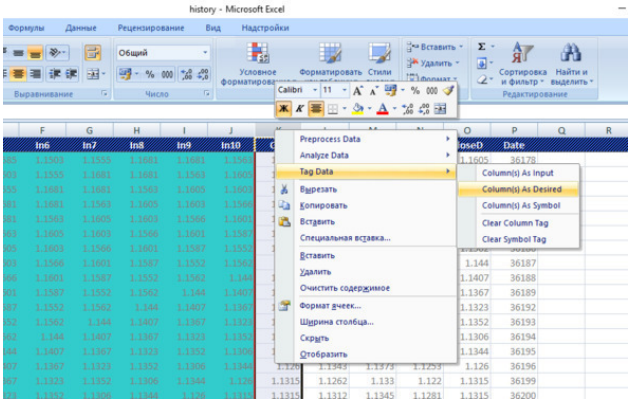


Таким образом, мы получим в каждой строке вектор из дневных цен закрытия с глубиной в десять дней – это будут входы нейросети. А в столбце Out, который также сдвинут на один день вперед по отношению к In10, будут обучающие примеры закрытия дня для нейросети.

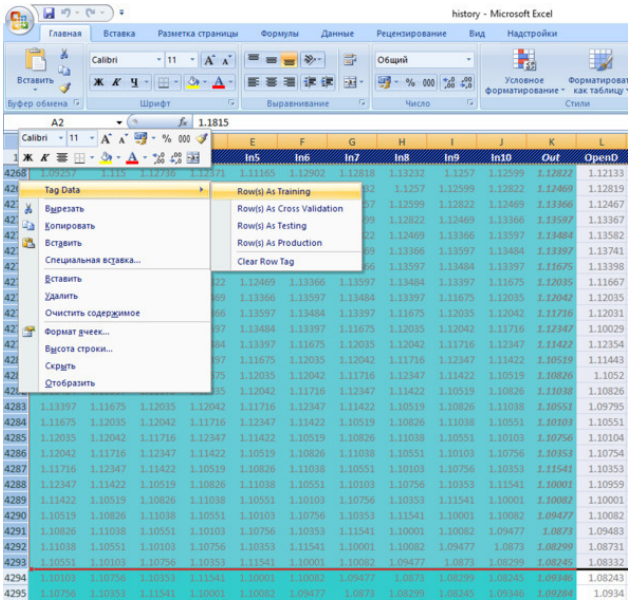
С помощью надстройки NeuroSolutions, выделив столбцы In1-In10, отформатируем их как входы.



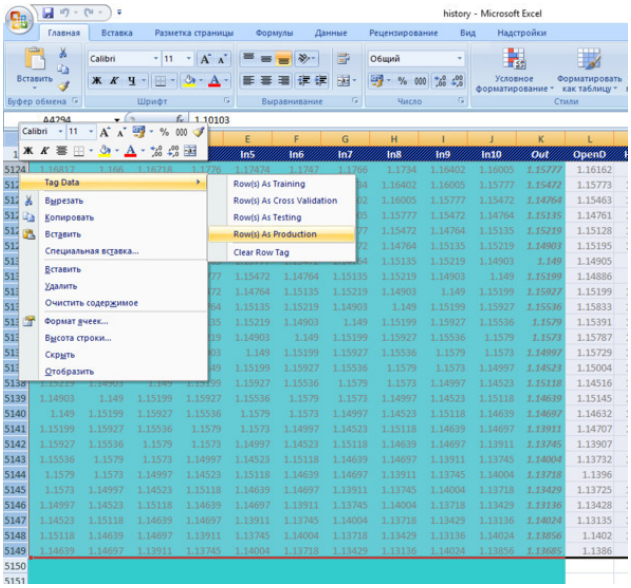
А столбец Out как выход нейросети.



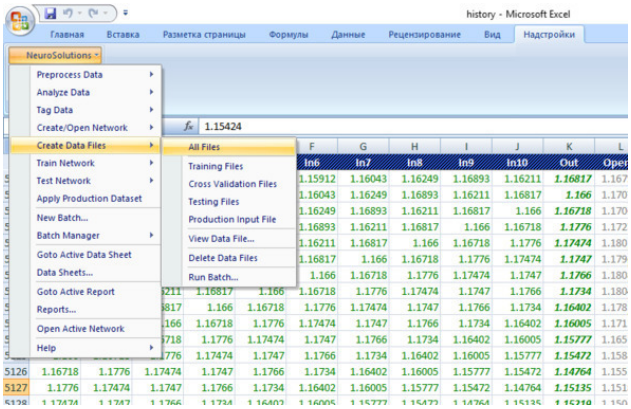
Аналогичным образом разобьем нашу матрицу построчно на обучающее множество.

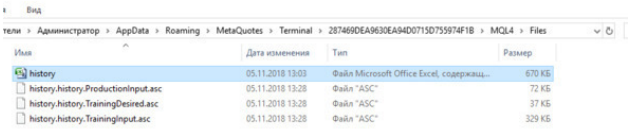


И множество, которое мы будем использовать для анализа.

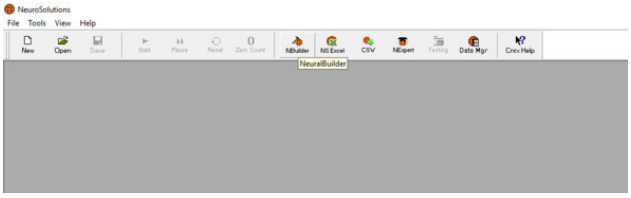


Теперь мы сформируем файлы для программы NeuroSolutions.

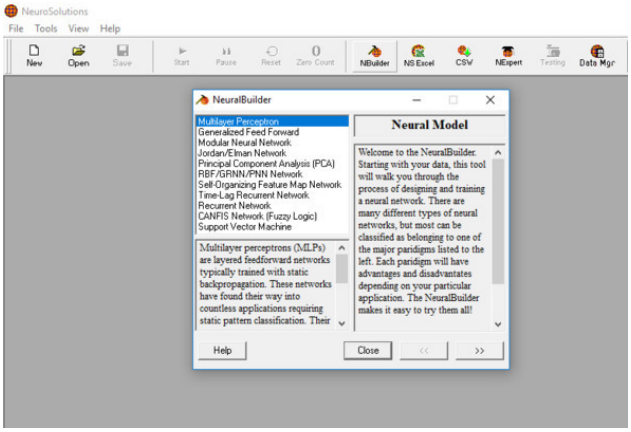




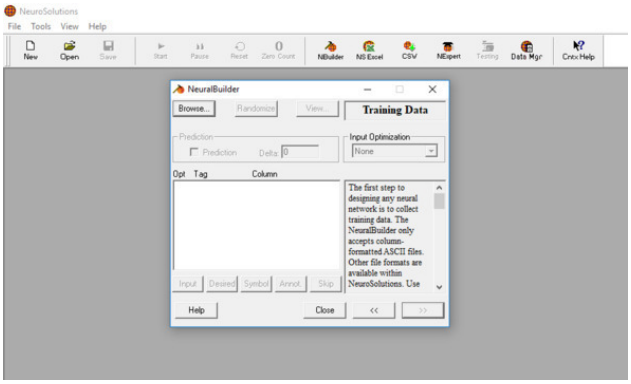
Откроем NeuroSolutions и нажмем кнопку NeuralBuilder.



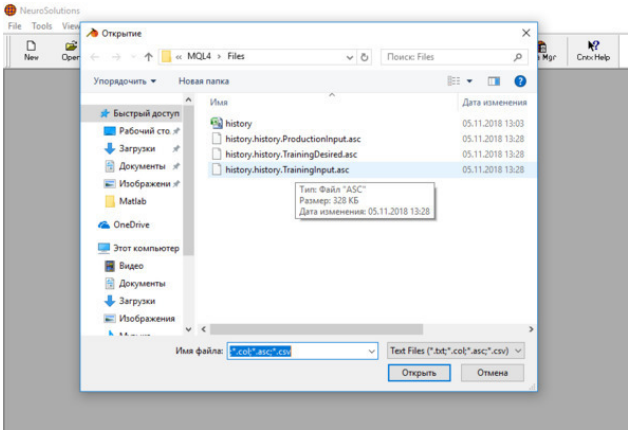
Выберем модель нейросети Multilayer Perceptron.

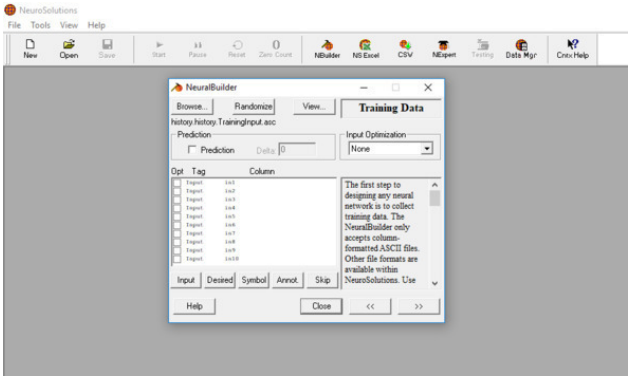


Нажмем кнопку Browse…

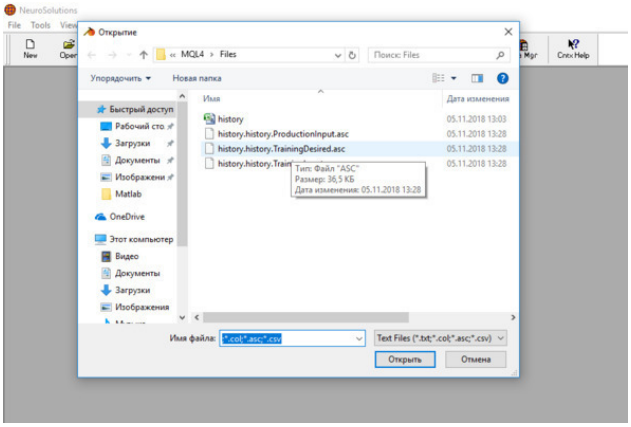


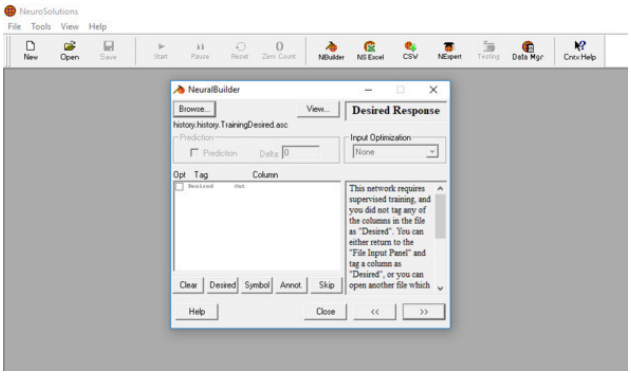
И откроем файл с обучающими входами.





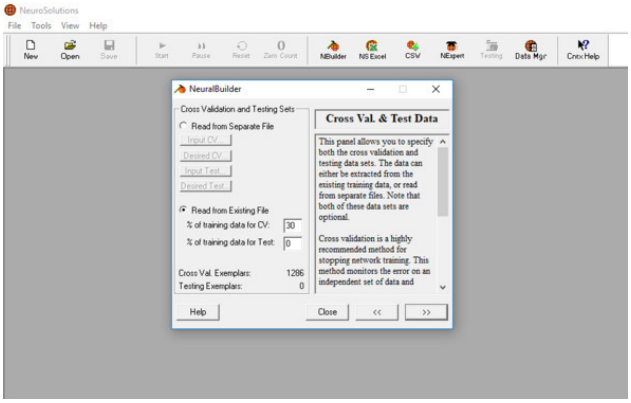
Далее откроем файл с обучающим выходом.

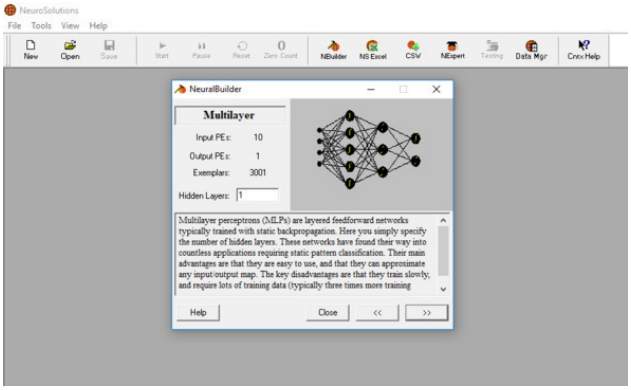


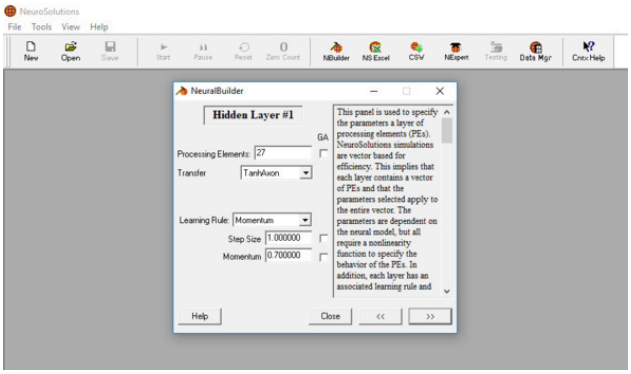


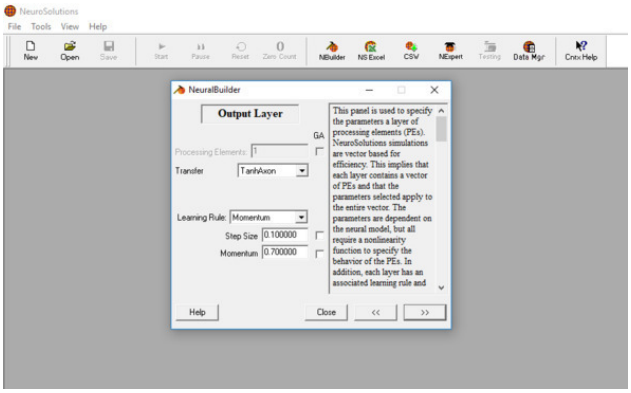
Определим 30% данных из тренировочного множества для перекрестной проверки в процессе обучения нейросети.

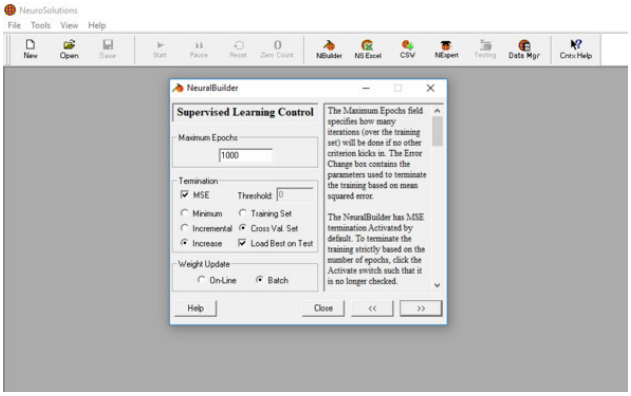
Жмем кнопку Next до тех пор, пока не сформируется нейросеть.

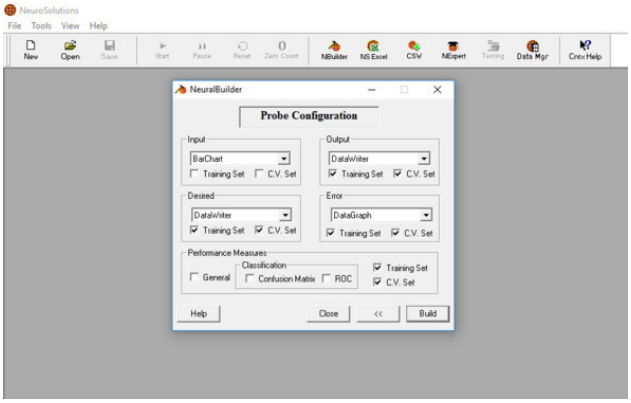

NeuroSolutions
File Tools View Help
New Open Save Start Pause Reset Zero Count NBuilder NS Excel CSV NExpert Testing Data Mgr Cntx Help
NeuralBuilder
Cross Validation and Testing Sets
Read from Separate File
Input CV
Desired CV
Input Test
Desired Test
Read from Existing File
% of training data for CV: 30
% of training data for Test: 0
Cross Val. Exemplars: 1286
Testing Exemplars: 0
Cross Val. & Test Data
This panel allows you to specify both the cross validation and testing data sets. The data can either be extracted from the existing training data, or read from separate files. Note that both of these data sets are optional.
Cross validation is a highly recommended method for stopping network training. This method monitors the error on an independent set of data and
Help
Close
<<
>>

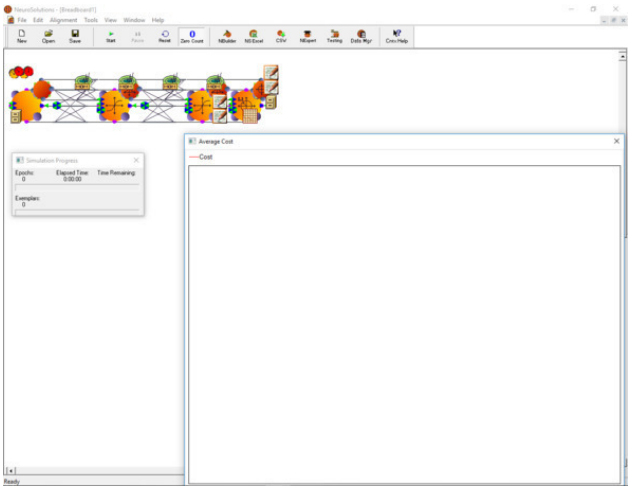

NeuroSolutions
File Tools View Help
New Open Save Start Pause Reset Zero Count NBuilder NS Excel CSV NExpert Testing Data Mgr Cntx Help
NeuralBuilder
Multilayer
Input PEs: 10
Output PEs: 1
Exemplars: 3001
Hidden Layers: 1
Multilayer perceptrons (MLPs) are layered feedforward networks typically trained with static backpropagation. Here you simply specify the number of hidden layers. These networks have found their way into countless applications requiring static pattern classification. Their main advantages are that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train slowly, and require lots of training data (typically three times more training
Help
Close
<<
>>


NeuroSolutions
File Tools View Help
New Open Save Start Pause Reset Zero Count NBuilder NS Excel CSV NExpert Testing Data Mgr Cntx Help
NeuralBuilder
Hidden Layer #1
GA
Processing Elements: 27
Transfer TanhAxon
Learning Rule: Momentum
Step Size 1.000000
Momentum 0.700000
This panel is used to specify the parameters a layer of processing elements (PEs). NeuroSolutions simulations are vector based for efficiency. This implies that each layer contains a vector of PEs and that the parameters selected apply to the entire vector. The parameters are dependent on the neural model, but all require a nonlinearity function to specify the behavior of the PEs. In addition, each layer has an associated learning rule and
Help
Close
<<
>>

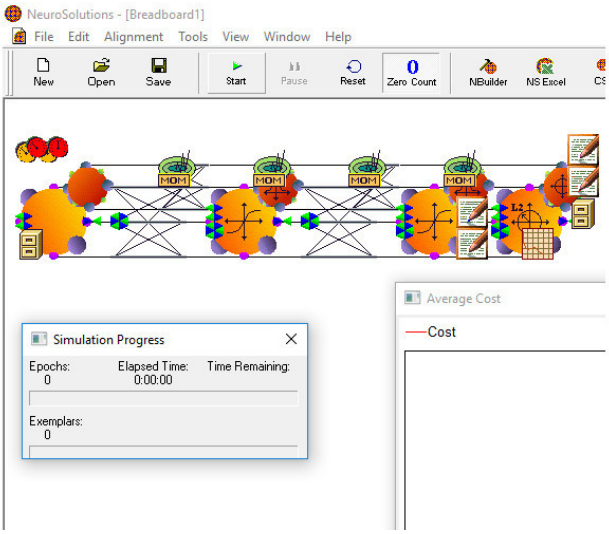

NeuroSolutions
File Tools View Help
New Open Save Start Pause Reset Zero Count NBuilder NS Excel CSV NExpert Testing Data Mgr Cntx Help
NeuralBuilder
Output Layer
GA
Processing Elements: 1
Transfer TanhAxon
Learning Rule: Momentum
Step Size 0.100000
Momentum 0.700000
This panel is used to specify the parameters a layer of processing elements (PEs). NeuroSolutions simulations are vector based for efficiency. This implies that each layer contains a vector of PEs and that the parameters selected apply to the entire vector. The parameters are dependent on the neural model, but all require a nonlinearity function to specify the behavior of the PEs. In addition, each layer has an associated learning rule and
Help
Close
<<
>>

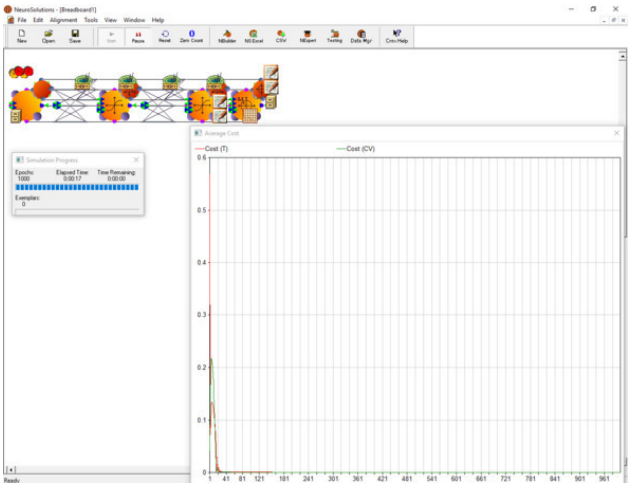




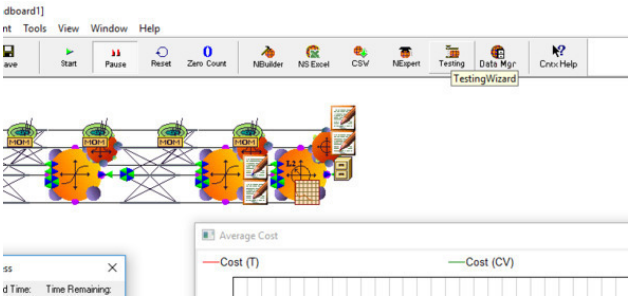


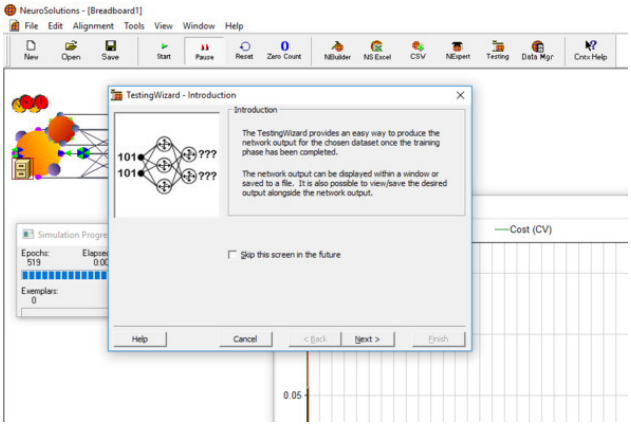
С помощью кнопки Start и запустим процесс обучения.



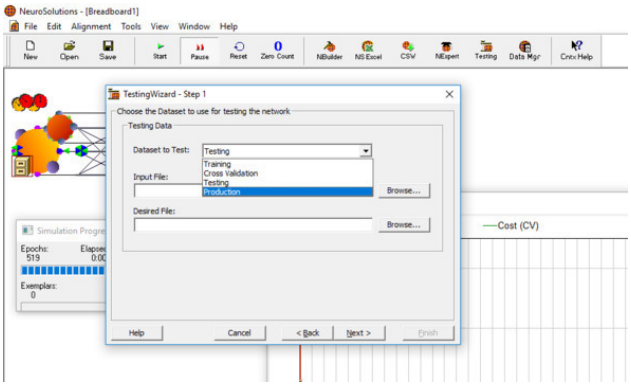


После завершения процесса обучения нажмем кнопку Testing.

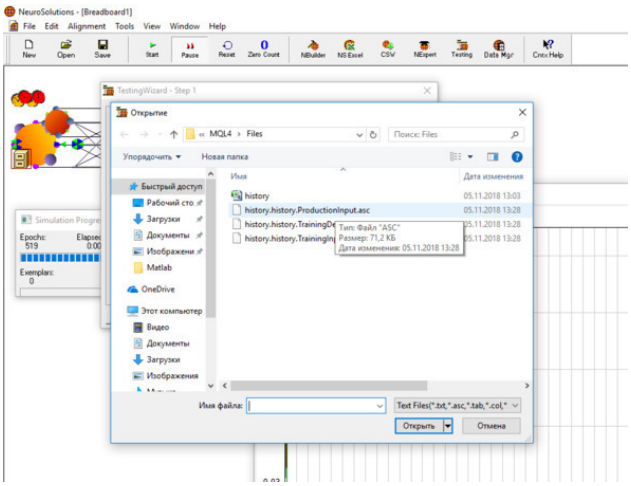


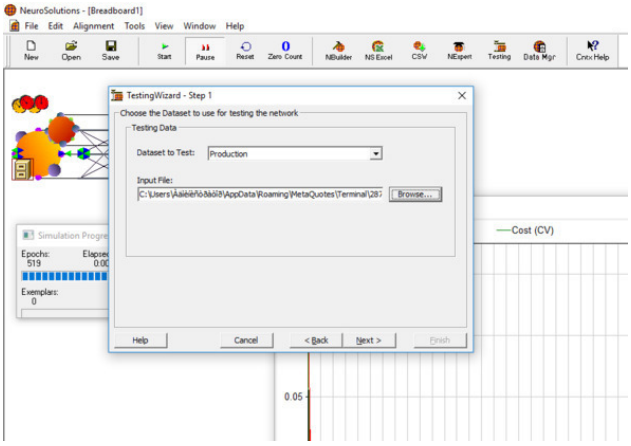


В выпадающем списке выберем Production.

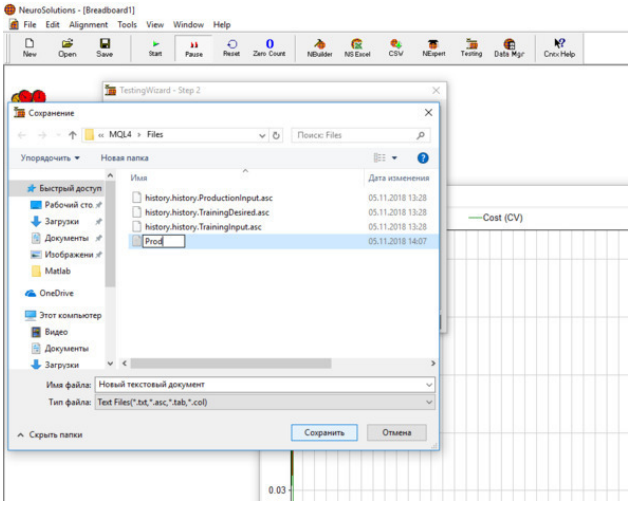


Выберем файл с данными для анализа.

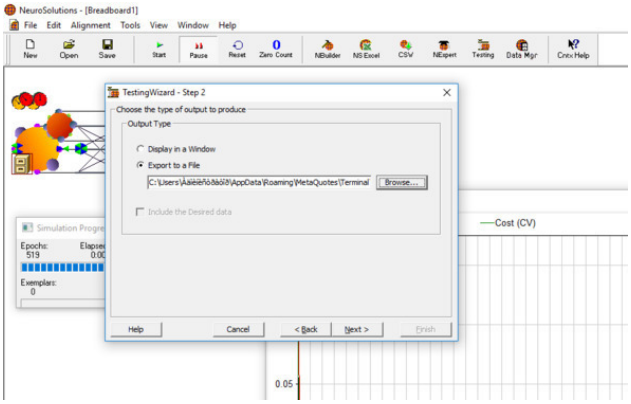


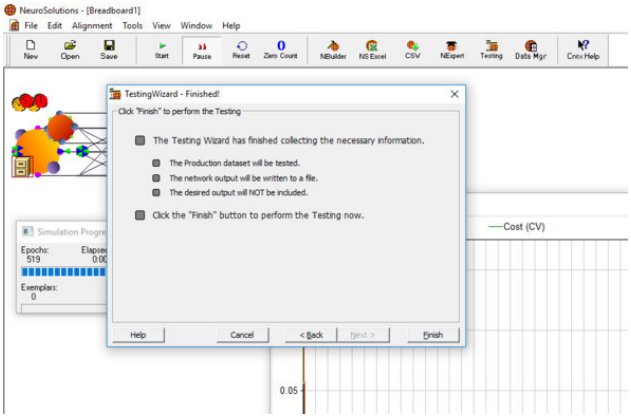


Создадим текстовой файл Prod.

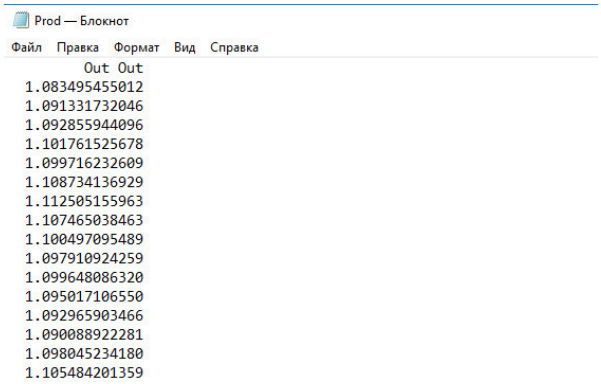


И экспортируем в него данные с результатами, полученными от нейросети.





Откроем файл Prod и скопируем из него отклики нейросети.

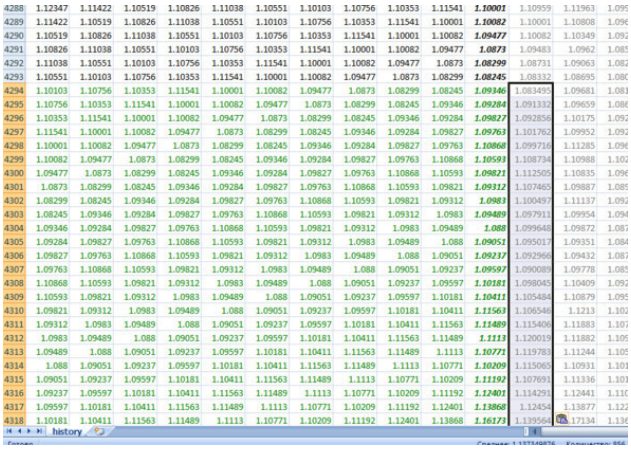


Prod — Блокнот

Файл Правка Формат Вид Справка

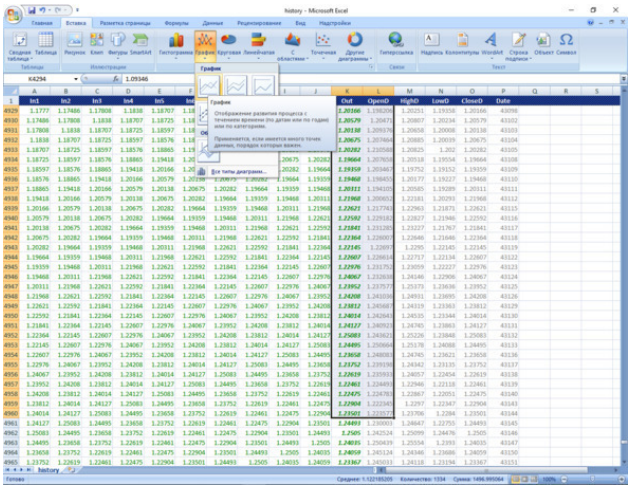
```
      Out Out
1.083495455012
1.091331732046
1.092855944096
1.101761525678
1.099716232609
1.108734136929
1.112505155963
1.107465038463
1.100497095489
1.097910924259
1.099648086320
1.095017106550
1.092965903466
1.090088922281
1.098045234180
1.105484201359
```

Вставим эти отклики рядом с реальными дневными закрытиями, которые мы хотели бы получить в результате работы нейросети.

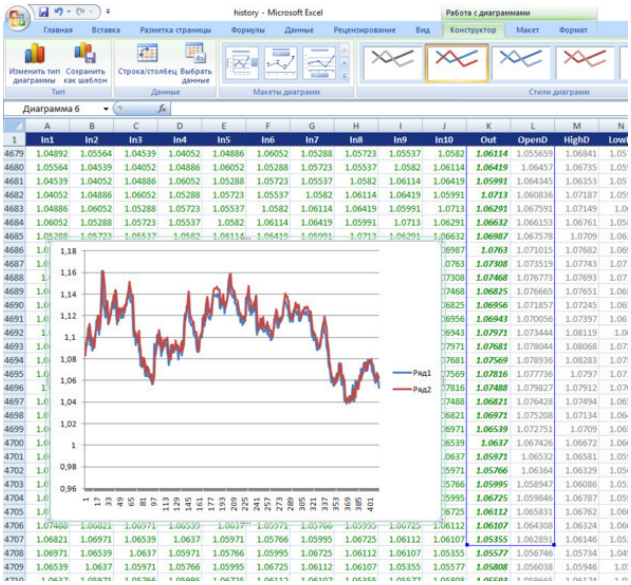


4288	1.12347	1.11422	1.10519	1.10826	1.11038	1.10551	1.10103	1.10756	1.10353	1.11541	**1.10001**	1.10959	1.11963	1.09
4289	1.11422	1.10519	1.10826	1.11038	1.10551	1.10103	1.10756	1.10353	1.11541	1.10001	**1.10082**	1.10001	1.10808	1.09
4290	1.10519	1.10826	1.11038	1.10551	1.10103	1.10756	1.10353	1.11541	1.10001	1.10082	**1.09477**	1.10082	1.10349	1.09
4291	1.10826	1.11038	1.10551	1.10103	1.10756	1.10353	1.11541	1.10001	1.10082	1.09477	**1.0873**	1.09483	1.0962	1.08
4292	1.11038	1.10551	1.10103	1.10756	1.10353	1.11541	1.10001	1.10082	1.09477	1.0873	**1.08299**	1.08731	1.09063	1.08
4293	1.10551	1.10103	1.10756	1.10353	1.11541	1.10001	1.10082	1.09477	1.0873	1.08299	**1.08245**	1.08332	1.08695	1.08
4294	1.10103	1.10756	1.10353	1.11541	1.10001	1.10082	1.09477	1.0873	1.08299	1.08245	**1.09346**	1.083493	1.09681	1.08
4295	1.10756	1.10353	1.11541	1.10001	1.10082	1.09477	1.0873	1.08299	1.08245	1.09346	**1.09284**	1.091332	1.09659	1.08
4296	1.10353	1.11541	1.10001	1.10082	1.09477	1.0873	1.08299	1.08245	1.09346	1.09284	**1.09827**	1.092856	1.10175	1.09
4297	1.11541	1.10001	1.10082	1.09477	1.0873	1.08299	1.08245	1.09346	1.09284	1.09827	**1.09763**	1.101762	1.09952	1.09
4298	1.10001	1.10082	1.09477	1.0873	1.08299	1.08245	1.09346	1.09284	1.09827	1.09763	**1.10868**	1.099716	1.11285	1.09
4299	1.10082	1.09477	1.0873	1.08299	1.08245	1.09346	1.09284	1.09827	1.09763	1.10868	**1.10593**	1.108734	1.10988	1.10
4300	1.09477	1.0873	1.08299	1.08245	1.09346	1.09284	1.09827	1.09763	1.10868	1.10593	**1.09821**	1.112505	1.10835	1.09
4301	1.0873	1.08299	1.08245	1.09346	1.09284	1.09827	1.09763	1.10868	1.10593	1.09821	**1.09312**	1.107465	1.09887	1.08
4302	1.08299	1.08245	1.09346	1.09284	1.09827	1.09763	1.10868	1.10593	1.09821	1.09312	**1.0983**	1.100497	1.11137	1.09
4303	1.08245	1.09346	1.09284	1.09827	1.09763	1.10868	1.10593	1.09821	1.09312	1.0983	**1.09489**	1.097911	1.09954	1.09
4304	1.09346	1.09284	1.09827	1.09763	1.10868	1.10593	1.09821	1.09312	1.0983	1.09489	**1.088**	1.099648	1.09872	1.08
4305	1.09284	1.09827	1.09763	1.10868	1.10593	1.09821	1.09312	1.0983	1.09489	1.088	**1.09051**	1.095017	1.09351	1.08
4306	1.09827	1.09763	1.10868	1.10593	1.09821	1.09312	1.0983	1.09489	1.088	1.09051	**1.09237**	1.092966	1.09432	1.08
4307	1.09763	1.10868	1.10593	1.09821	1.09312	1.0983	1.09489	1.088	1.09051	1.09237	**1.09597**	1.090089	1.09778	1.08
4308	1.10868	1.10593	1.09821	1.09312	1.0983	1.09489	1.088	1.09051	1.09237	1.09597	**1.10181**	1.098043	1.10409	1.09
4309	1.10593	1.09821	1.09312	1.0983	1.09489	1.088	1.09051	1.09237	1.09597	1.10181	**1.10411**	1.105484	1.10879	1.09
4310	1.09821	1.09312	1.0983	1.09489	1.088	1.09051	1.09237	1.09597	1.10181	1.10411	**1.11563**	1.106546	1.1213	1.10
4311	1.09312	1.0983	1.09489	1.088	1.09051	1.09237	1.09597	1.10181	1.10411	1.11563	**1.11489**	1.115406	1.11883	1.10
4312	1.0983	1.09489	1.088	1.09051	1.09237	1.09597	1.10181	1.10411	1.11563	1.11489	**1.1113**	1.120019	1.11882	1.10
4313	1.09489	1.088	1.09051	1.09237	1.09597	1.10181	1.10411	1.11563	1.11489	1.1113	**1.10771**	1.119783	1.11244	1.10
4314	1.088	1.09051	1.09237	1.09597	1.10181	1.10411	1.11563	1.11489	1.1113	1.10771	**1.10209**	1.115065	1.10931	1.10
4315	1.09051	1.09237	1.09597	1.10181	1.10411	1.11563	1.11489	1.1113	1.10771	1.10209	**1.11192**	1.107691	1.11336	1.10
4316	1.09237	1.09597	1.10181	1.10411	1.11563	1.11489	1.1113	1.10771	1.10209	1.11192	**1.12401**	1.114291	1.12441	1.11
4317	1.09597	1.10181	1.10411	1.11563	1.11489	1.1113	1.10771	1.10209	1.11192	1.12401	**1.13868**	1.12454	1.13877	1.12
4318	1.10181	1.10411	1.11563	1.11489	1.1113	1.10771	1.10209	1.11192	1.12401	1.13868	**1.16173**	1.139564	1.17134	1.13

history



Поместим эти данные на график.



Результат вроде бы нас должен устроить. Кажется, что полученный результат хорошо накладывается на график цен закрытия. Однако, увеличив масштаб, мы обнаружим, что —

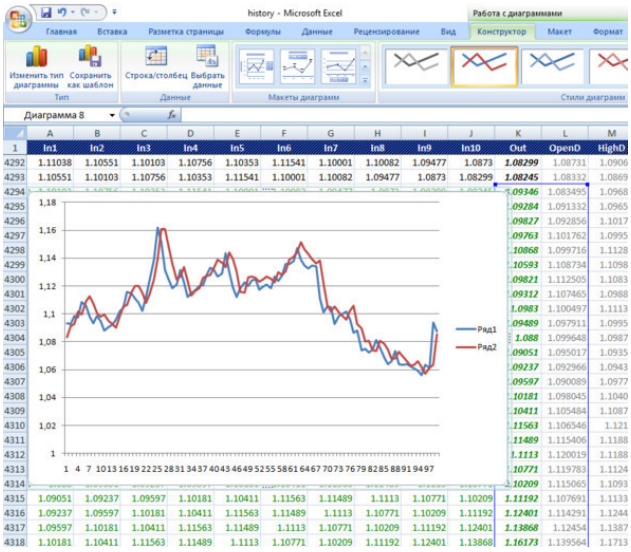


график отклика нейросети, хоть и повторяет график цен, но на один шаг от него отстает. Причем это не зависит – прогнозируем ли мы ценовые данные или производные от них. Исходя из этого, мы можем вывести какой-то постулат. Например – «То, что для нас – вчера, для нейросети – сегодня». Согласитесь, что здесь, в принципе, ни о каком прогнозе речи идти не может. Однако, забегая вперед, отмечу, что данный вариант, при определенной доработке мы так же будем использовать. Но, мы бы, конечно, хотели бы использовать постулат – «То, что для нейросети сегодня, для нас – завтра». Машина времени, какая то. Но мы с Вами ведь понимаем, что все-таки самая лучшая нейросеть – это наш мозг. И то, мы можем использовать этот постулат максимум с 50% успехом (если мы говорим о вероятности да или нет), а то и хуже. Но ведь есть еще и третий вариант постулата – «То, что для нейросети – вчера, для нас – сегодня». Разберем, что для нас означают эти постулаты в трейдинге:

первый – мы совершаем сделку и завтра получаем ответ от нейросети, что мы открылись в правильном направлении или нет. Хотя мы это уже знаем и без нейросети;

второй – мы получаем информацию от нейросети, совершаем сделку и завтра видим, правильная рекомендация была или нет;

третий – мы получаем информацию от нейросети, когда нам надо совершить ту или иную сделку.

Первый вариант, естественно мы отбрасываем сразу. А вот второй и третий для торговли подходят. Однако второй вариант – вариант как бы заглядывания в будущее. Утрировано этот вариант торговли заключается в том, что мы получаем сигнал от нейросети в определенный момент времени – например по закрытию дня с прогнозом как закроется следующий день. Реализовать его для чисто механической торговли на данном этапе сложно. Ну, а если представить, что им получит возможность воспользоваться большинство торговцев – то он сразу же потеряет свою актуальность. Смысл третьего варианта, заключается в том, что мы отслеживаем отклик нейросети на протяжении торговой сессии и покупаем либо продаем его интерпретируя. И здесь нам надо понять основное. Какой из вариантов мы сможем реализовать зависит от того как мы будем обучать нейросеть. И согласитесь, что третий вариант реализовать все-таки легче. Если во втором – мы будем использовать, какую либо информацию с прицелом на получение результата на следующий день – его закрытия (день выбран как пример, естественно может быть какой либо другой период), то в третьем варианте мы используем информацию, пришедшую за шаг до принятия решения – куда двинется цена в этот момент времени.

Критические ошибки при разработке нейросетевой системы

Рассмотрим, на примере как допускаются ошибки при тестировании нейросети. Если у вас нет программы «NeuroSolutions 6», то пропустите дальнейшее описание работы с ней, а рассмотрите результаты и сделанные выводы. В этой книге я не буду рассматривать создание системы на основе «NeuroSolutions». Хотя в принципе, автоматическая нейросетевая система реализована и на основе данного продукта.

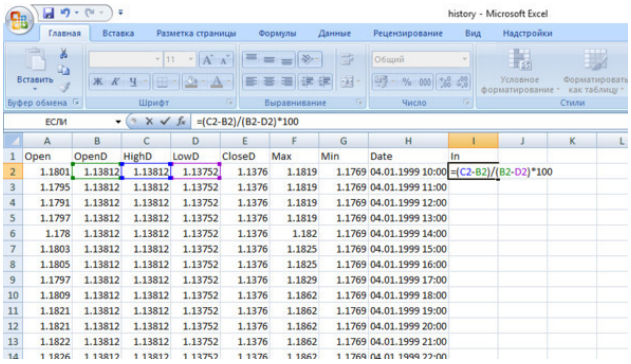
Отметим, что следует понимать разницу между обучением и тестированием. Обучать нейросеть можно на любых примерах даже, некорректных для тестового множества – ведь обучение мы проводим на событиях, которые уже произошли. Данный пример взят из реальной жизни. Данный способ работы с нейросетями продавался в интернете. Позиционировался как система, которая дает 80—90% прибыльных сделок. Причем продавец, мне кажется, искренне заблуждался в идеальной результативности данного способа подготовки нейросети. Я делаю этот вывод из того, что ошибка возникала на стадии тестирования отклика сети.

Итак, сначала модернизируем и скомпилируем скрипт для получения исторических данных.

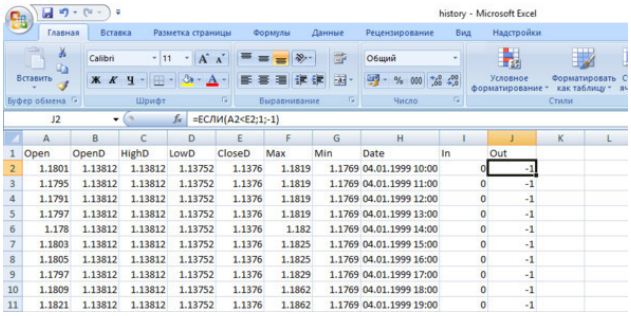
```
//+ – — – — – — – — – — – — – — – — – — – — – — – — – — – — – — – +
//| History.mq4 |
//| Copyright © 2009, Andrey Dibrov. |
//+ – — – — – — – — – — – — – — – — – — – — – — – — – — – — – — – +
#property copyright «Copyright © 2009, Andrey Dibrov.»

int file=FileOpen («history. csv», FILE_CSV|FILE_READ|FILE_WRITE,»;»);
//+ – — – — – — – — – — – — – — – — – — – — – — – — – — – — – — – +
//| Script program start function |
//+ – — – — – — – — – — – — – — – — – — – — – — – — – — – — – — – +
void OnStart ()
{
// – —
FileWrite (file,«Open; OpenD; HighD; LowD; CloseD; Max; Min; Date»);

if (file> 0)
{
Alert («Идет запись файла»);
for (int i=iBars (NULL,60) -1; i> =0; i – )
{

FileWrite (file,

iOpen (NULL,60,i),
iOpen (NULL,60,iBarShift (NULL,1440,iTime (NULL,60,i))),
iHigh (NULL,60,iBarShift (NULL,1440,iTime (NULL,60,i))),
iLow (NULL,60,iBarShift (NULL,1440,iTime (NULL,60,i))),
iClose (NULL,60,iBarShift (NULL,1440,iTime (NULL,60,i))),
iCustom (NULL,60,«Max», 0,1440,60,0,i),
iCustom (NULL,60,«Min», 0,1440,60,0,i),
```

```
TimeToStr (iTime (NULL,60,i)));
}
}
Alert («Файл записан»);
FileClose (file);
}
//+ – — – — – — – — – — – — – — – — – — – — – — – — – — – +
```

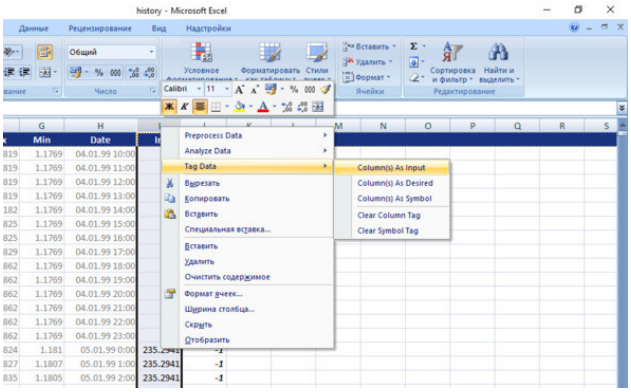
В результате работы скрипта получим файл. Создадим колонку «In» и заполним ее формулой.



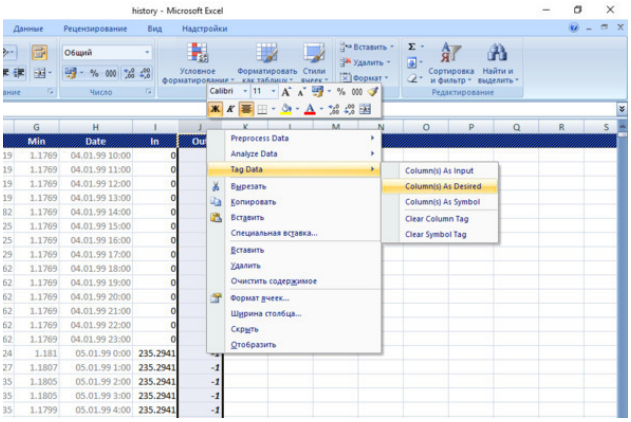
Добавим колонку «Out» и заполним ее формулой.



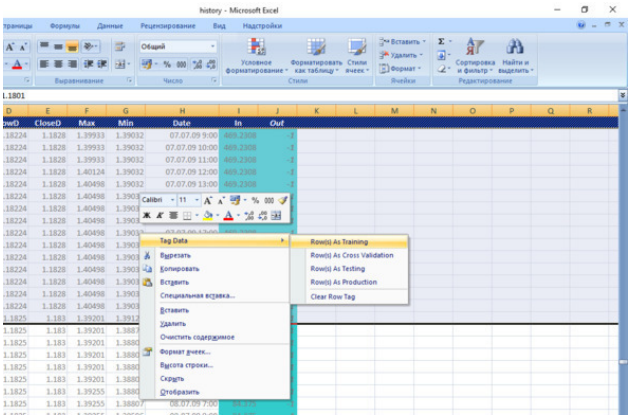
Воспользуемся программой «NeuroSolutions 6». С помощью ее надстройки колонку «In» обозначим как «Input».



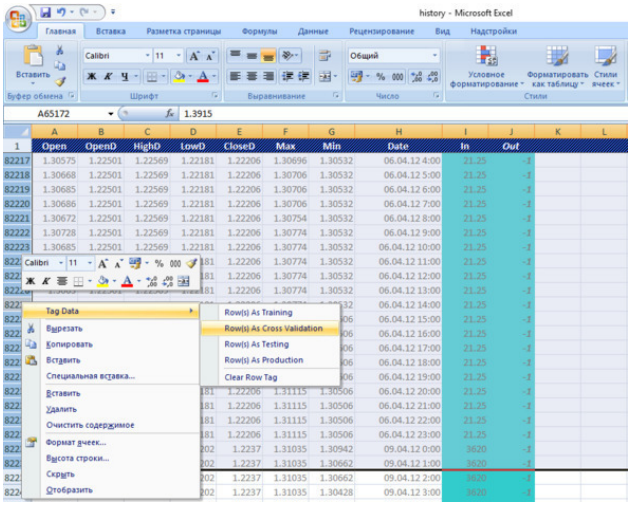
А колонку «Out» как «Desired».



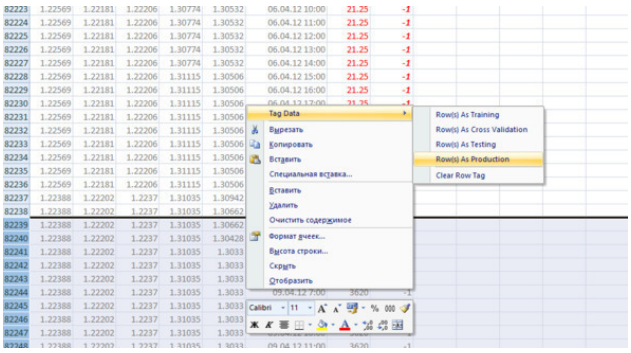
Энное количество строк обозначим, как «Training»



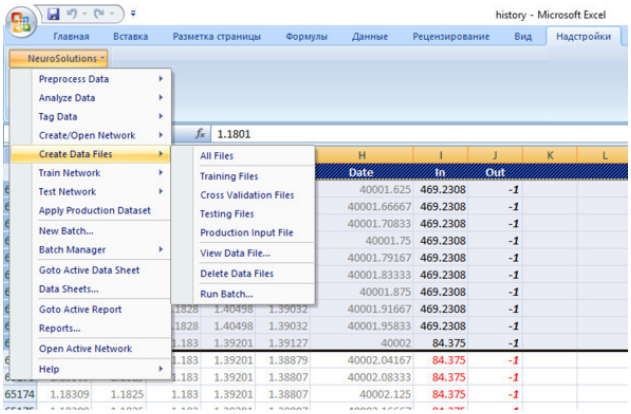
Энное количество строк обозначим, как «Cross Validation».



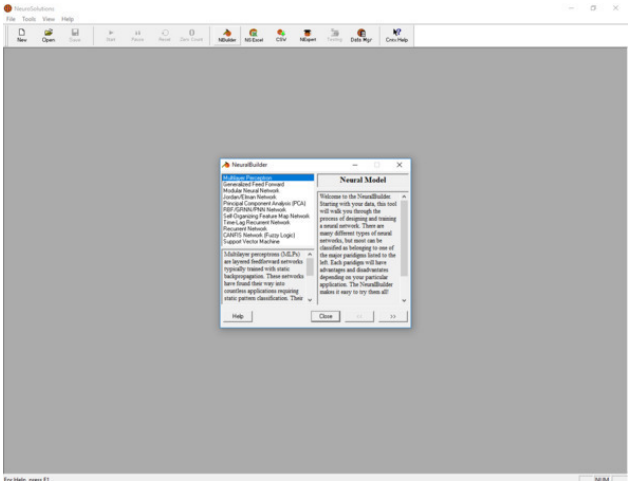
И последние строки как «Production».



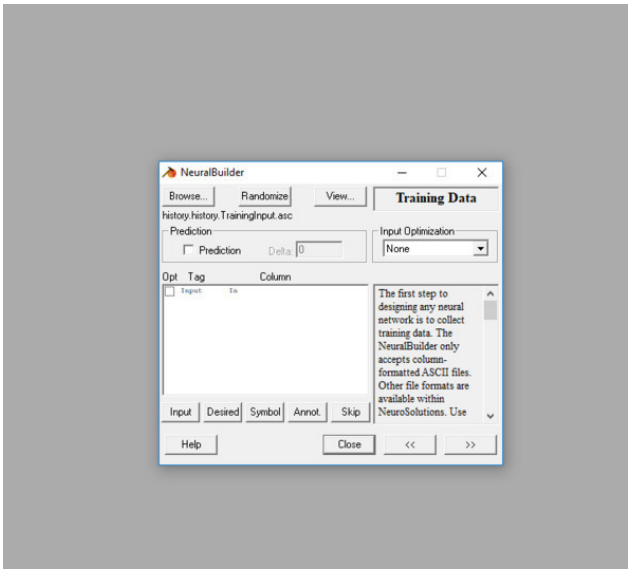
Далее создадим файлы для NS6.



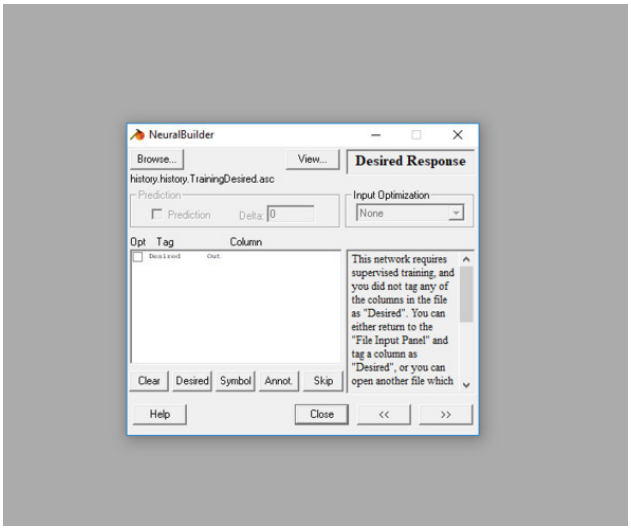
Запустим в рабочей среде NS – NeuralBilder.



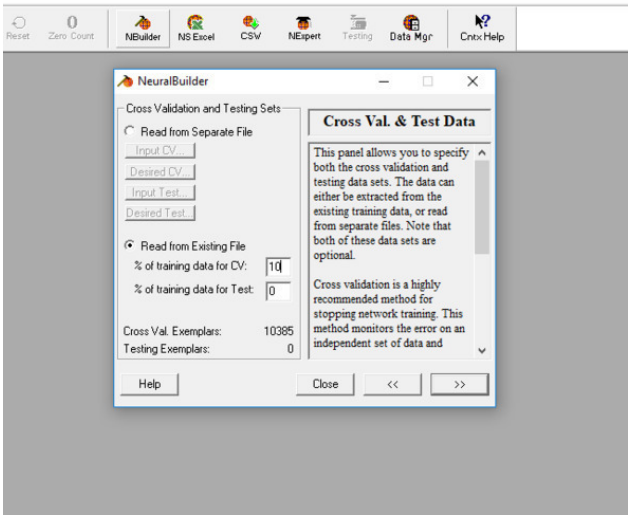
Здесь мы можем выбрать нейросеть нужной нам архитектуры. Остановимся на первой сети Multilayer perceptron. Нажимаем кнопку с правыми стрелками. Откроется окно Training Data. С помощью кнопки Browse откроем файл TrainingInput.



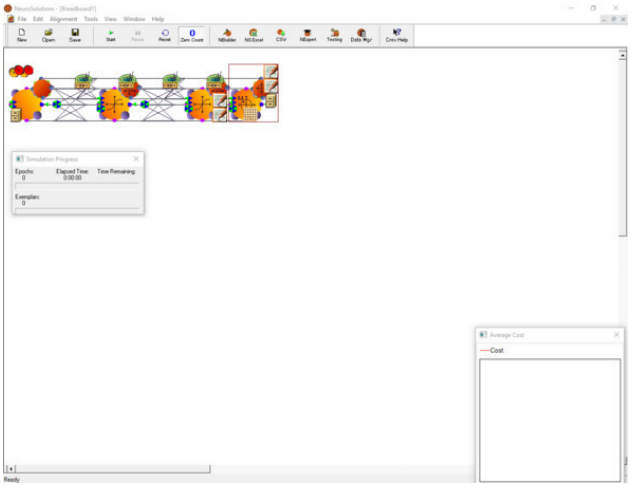
Далее перейдем в окно Desired Response и откроем файл TrainingDesire.



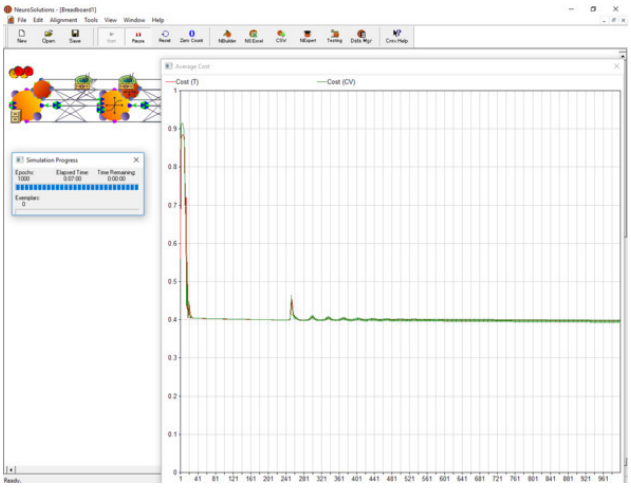
Откроем окно Cross Val. & Test Data. По умолчанию активно окошечко для ввода текста % of training data for CV. Введем 10%. Программа автоматически зарезервирует данное количество строк под CV из TD. Активировав радиокнопку Read from Separate File, мы можем выбрать файлы сохраненные, нами ранее.



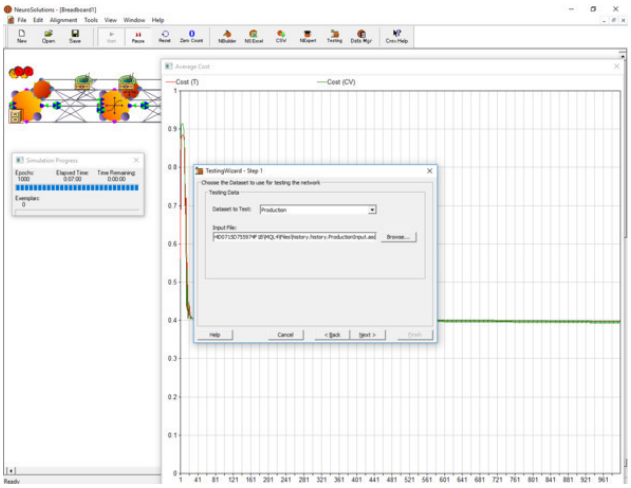
Далее продвигаемся по окнам ничего в них, не меняя, пока в рабочей среде не сформируется нейросеть.



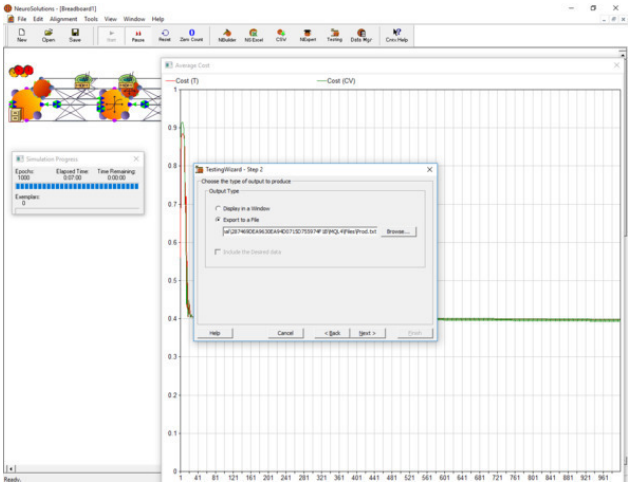
Для лучшей визуализации расширим окно Average Cost и нажмем кнопку Start. Подождем, пока закончится обучение.



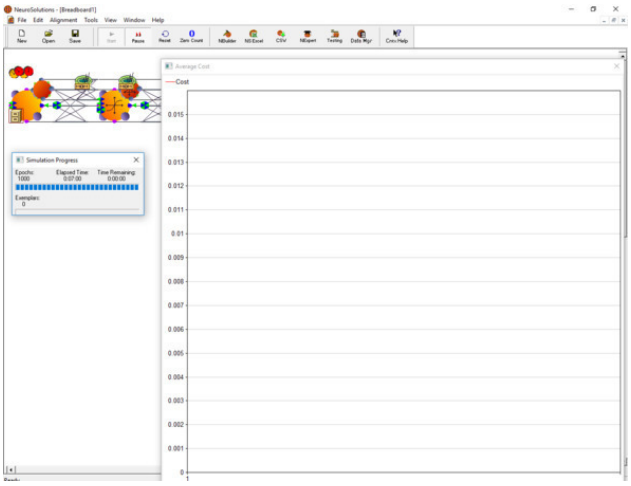
Жмем кнопку Testing, Next и в выпадающем окне выбираем Production. С помощью кнопки Browse находим файл ProductionInput.



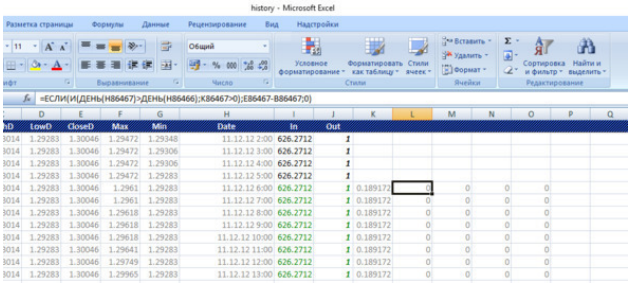
Двигаемся далее и в следующем окне активируем радиокнопку Export to a File. И находим заранее созданный нами текстовой файл с, допустим, выбранным нами именем Prod. txt.



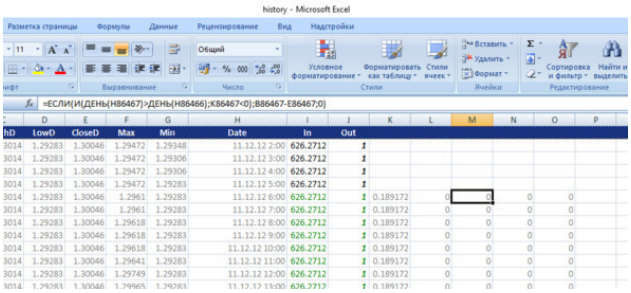
Нажимаем кнопки Next и Finish. Таким образом, мы экспортировали отклик нейросети для тестирования в файл Prod. txt. С помощью кнопки Save, сохраним нейросеть.



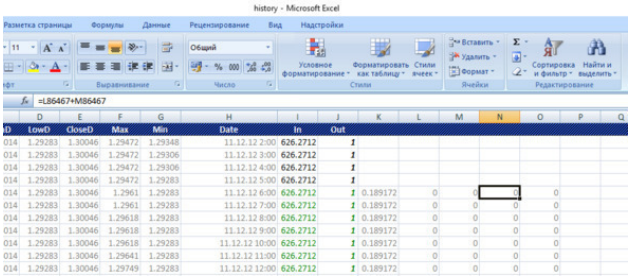
Данные из файла Prod. txt подставим в файл history. csv рядом с Tag Data «Production». В соседнюю ячейку вставим формулу условия совершения сделок на покупку.



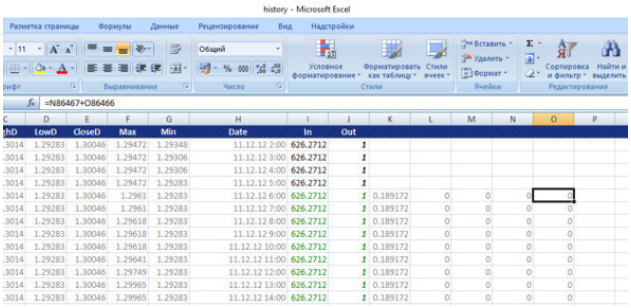
В соседнюю ячейку вставим формулу условия совершения сделок на продажу.



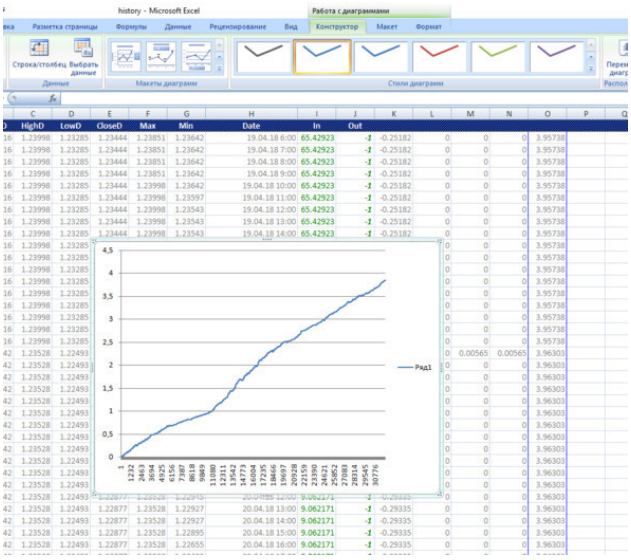
В соседнюю ячейку вставим формулу суммирования сделок.



В соседнюю ячейку вставим формулу суммирования и заполним этими формулами ячейки до конца истории.

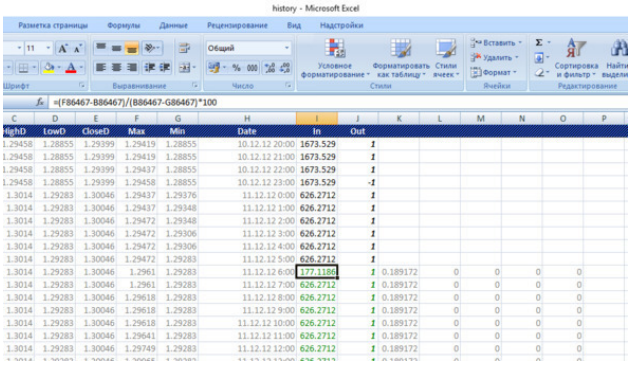


Далее вставим график нашей условной прибыли.

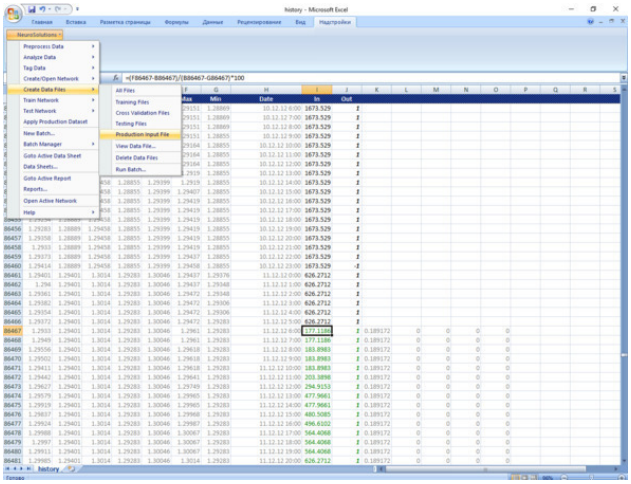


Как мы видим – график прибыльности у нас идеальный. Хотя мы в процессе обучения и тестирования допустили некоторые ошибки и обучали нейросеть всего на одном входе. Далее мы протестируем, обученную нейросеть на данных, которые будут к нам поступать реально. Т.е. в формуле ячеек колонки «In» мы будем использовать локальные максимумы и минимумы, которые мы записали в файл «history. csv» с помощью индикаторов «Max» и «Min».

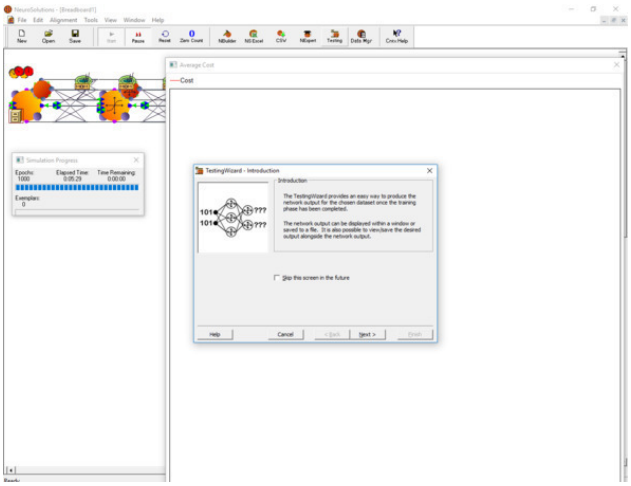
Введем формулу, подставив вместо High и Low дня, локальные максимумы и минимумы.



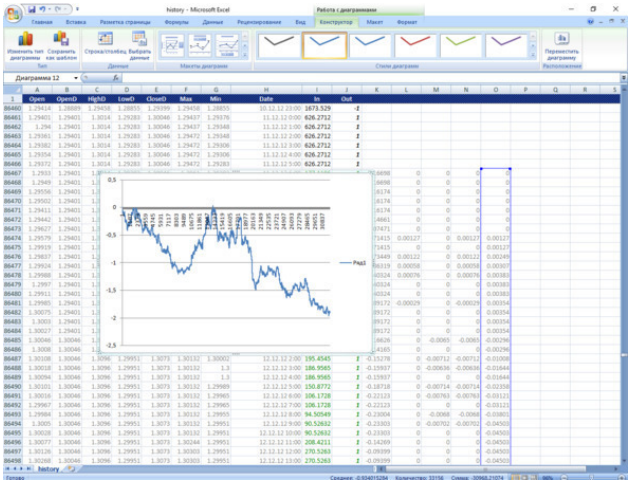
Еще раз воспользуемся надстройкой NeuroSolutions для создания Production Input File.



Запустим программу NS6 и в ее среде откроем сохраненную ранее нейросеть. Протестируем теперь измененные данные.



Изменим формулы условий совершения сделок. Сделки будем открывать по открытию следующего часа после достижения локальных максимумов и минимумов. Закрывать будем по закрытию дня.



Как мы видим, результат мы получаем совершенно другой.

Пользовательские технические индикаторы, используемые для обучения нейросети

Индикаторы, используемые для обучения НС, являются пользовательскими, т.е. написанными на языке MQL автором этой книги, но в тоже время они являются производным от классических. Это сделано исходя из логики обучения НС. Нам необходимо готовить нейросеть для принятия решения на открытие позиции внутри дня – по достижению максимума или минимума периода. Для повышения качества нам необходимо заглянуть на несколько периодов назад и математически описать уровни исследуемых торговых дней. А так же использовать движение цены внутри дня. Исходя из вышесказанного, индикаторы являются двухуровневыми, т.е. используют два периода графиков – дневной и часовой (хотя предусмотрена возможность использовать и другие периоды). Еще одной особенностью является то, что для обучения нейросети на покупку и продажу используются модифицированные индикаторы. Далее будут предоставлены описания индикаторов и их коды.

Для инициализации индикаторов в терминале должны быть загружены исторические данные обоих периодов. Инициируются на младшем таймфрейме

И так… индикаторы —

В названиях индикаторов используются названия оригинальных индикаторов, что дает возможность понимать их производность.

– StoxasticPolzMinTest;
– StoxasticTurnMin;
– WilliamsPolzMaxTest;
– WilliamsTurnMax;
– MaPolzMin;
– MaPolzMax;
– McadPolzMin (включает в себя индикатор MaPolzMin);
– McadPolzMax (включает в себя индикатор MaPolzMax);
– Max-OpenOld (включает в себя индикатор Max);
– OpenOld-Min (включает в себя индикатор Min);
– Max;
– Min;

Индикаторы Max и Min используются также для формирования выборки обучения нейросети.

Я заранее извиняюсь, если код программ на ваш взгляд выглядит не профессионально, так как сам не является профессиональным программистом. Но это является и плюсом, показывая, что с задачами, описываемыми в книге, могут с успехом справляться и трейдеры без специального программистского образования.

StoxasticPolzMinTest

Данный индикатор – перепрограммирован из классического осциллятора «Stoxastic».

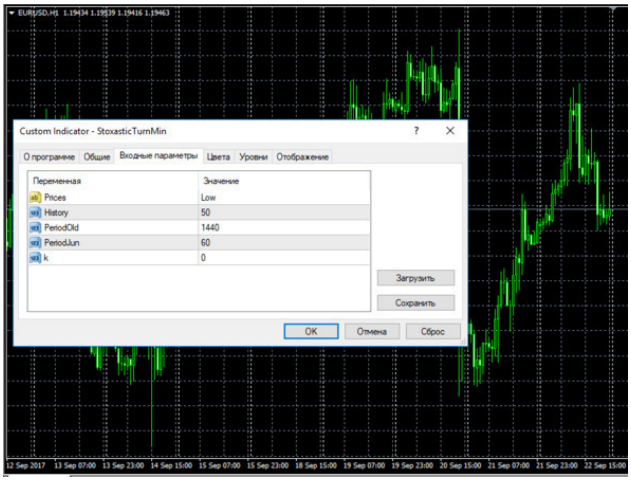
Для дальнейшего описания индикаторов я буду использовать в качестве старшего таймфрейма – дневные данные предшествующие торговой сессии. В качестве младшего таймфрейма – движение цены по часовым данным. Хотя повторюсь, что мы можем использовать и другие периоды.

Вычисляется индикатор на основании минимума и максимума количества прошедших дней, которое мы хотим использовать для обучения, и движения цены стремящейся к минимуму в течение каждого часа во время торговой сессии, что, как было сказано выше, обеспечивает динамичное принятие решения. Либо можем использовать цены открытия каждого часа – что является статичным принятием решения.

На рисунке показан данный индикатор с однодневным и десятидневным периодами.

Входные параметры индикатора:

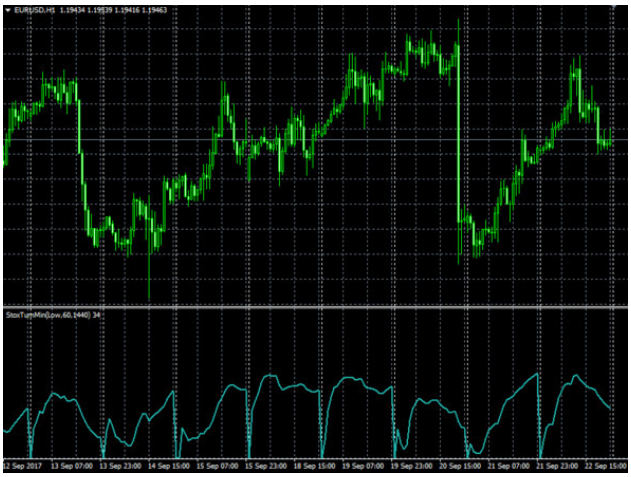
Prices=«Low» – либо «Open».
History=50 – историческая глубина. 0 – отрисовывается на всей истории.
KPeriod=1 – количество предшествующих дней.
PeriodOld=1440 – старший период в минутах.
PeriodJun=60 – младший период в минутах.
k=0 —сдвиг входного параметра Prices (не «Open») назад.




```
//+ – — – — – — – — – — – — – — – — – — – — – — – — – — – — – — – +
//| StoxasticPolzMinTest.mq4 |
//| Copyright © 2009, Andrey Dibrov. |
//+ – — – — – — – — – — – — – — – — – — – — – — – — – — – — – — – +
#property copyright «Copyright © 2009, Andrey Dibrov.»
```

StoxasticTurnMin

Данный индикатор так же является видоизмененным стохастическим осциллятором. Вычисляется индикатор на основании минимума и максимума количества прошедших баров с 00:00 часов текущих суток и движения цены стремящейся к минимуму в течение каждого часа во время торговой сессии, что, как было сказано выше, обеспечивает динамичное принятие решения. Либо можем использовать цены открытия каждого часа – что является статичным принятием решения.



Исходя из логики индикатора – он имеет однодневный период.

Входные параметры индикатора:

Prices=«Low» – либо «Open».
History=50 – историческая глубина. 0 – отрисовывается на всей истории.
PeriodOld=1440 – старший период в минутах.
PeriodJun=60 – младший период в минутах.
k=0 —сдвиг входного параметра Prices (не «Open») назад.

```
//+ – — – — – — – — – — – — – — – — – — – — – — – +
//| StoxasticTurnMin.mq4 |
//| Copyright © 2009, Andrey Dibrov. |
//+ – — – — – — – — – — – — – — – — – — – — – — – +
#property copyright «Copyright © 2009, Andrey Dibrov.»
```

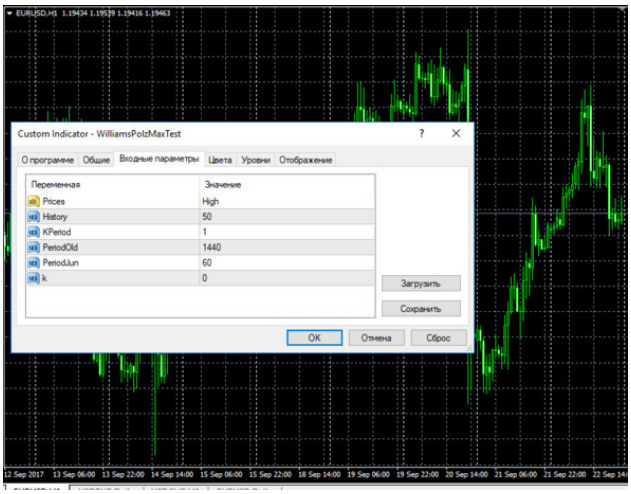
WilliamsPolzMaxTest

Данный индикатор, также, перепрограммирован из классического осциллятора «Stoxastic». Вычисляется индикатор на основании минимума и максимума количества прошедших дней, которое мы хотим использовать для обучения, и движения цены стремящейся к максимуму в течение каждого часа во время торговой сессии, что, как было сказано выше, обеспечивает динамичное принятие решения. Либо можем использовать цены открытия каждого часа – что является статичным принятием решения.

На рисунке показан данный индикатор с однодневным и десятидневным периодами.

Входные параметры индикатора:

Prices=«Low» – либо «Open».
History=50 – историческая глубина. 0 – отрисовывается на всей истории.
KPeriod=1 – количество предшествующих дней.
PeriodOld=1440 – старший период в минутах.
PeriodJun=60 – младший период в минутах.
k=0 —сдвиг входного параметра Prices (не «Open») назад.



```
//+ – — – — – — – — – — – — – — – — – — – — – — – — – — – +
//| WilliamsPolsMaxTest.mq4 |
//| Copyright © 2009, Andrey Dibrov. |
//+ – — – — – — – — – — – — – — – — – — – — – — – — – — – +
#property copyright «Copyright © 2009, Andrey Dibrov.»
```

WilliamsTurnMax

Данный индикатор так же является видоизмененным стохастическим осцилятором. Вычисляется индикатор на основании минимума и максимума количества прошедших баров с 00:00 часов текущих суток и движения цены стремящейся к максимуму в течение каждого часа во время торговой сессии, что, как было сказано выше, обеспечивает динамичное принятие решения. Либо можем использовать цены открытия каждого часа – что является статичным принятием решения.

Исходя из логики индикатора – он имеет однодневный период.

Входные параметры индикатора:

Prices=«High» – либо «Open».
History=50 – историческая глубина. 0 – отрисовывается на всей истории.
PeriodOld=1440 – старший период в минутах.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, купив полную легальную версию на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.