электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 519.85

Способы прогнозирования биржевых индексов с использованием нейронных сетей

***Шуленко А.А.**, студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»

Научный руководитель: Кивва К.А., ассистент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»
k.kivva@bmstu.ru

Введение

Одной из основных частей любой экономической системы являются финансовые рынки. Эффективное распределение ресурсов, контроль за финансовыми потоками и активами невозможны без финансовых рынков. Ключевой задачей для финансовых рынков всегда было прогнозирование динамики фондовых индексов, котировок ценных бумаг, валютных курсов и других показателей.

Фондовые индексы отражают значения показателей фондового рынка, являются важным ориентиром для инвесторов, так как позволяют осуществлять робастную инвестиционную стратегию.

В течении длительного времени было проведено множество исследований по прогнозированию динамики фондовых индексов. Тем не менее, традиционные способы ограничены в получении точных прогнозов для динамично меняющегося рынка в связи с влиянием множества факторов, таких, как политические изменения, экономическая ситуация, военная обстановка и чрезвычайные происшествия. Поэтому проблема прогнозирования до сих пор привлекает внимание многих экспертов и исследователей. А. Лендасс, И. де Бот, В. Вертц, М. Варлейсен использовали нелинейную модель временных рядов для прогнозирования тенденции изменений ключевого фондового индекса Бельгии [6]. К. Д. Ли, А. Ю. Ши, С. Ю, Я. Я. Джин прогнозировали корейский фондовый индекс с использованием нейросетевой модели обратного распространения и модели SARIMA [7].

Анализ существующих подходов к прогнозированию

До появления компьютерных технологий люди принимали решения об операциях, совершаемых с акциями, руководствуясь интуицией. С ростом уровня инвестиционных и торговых рынков производился поиск инструментов и методов, которые повысили бы уровни доходности, минимизируя риски. Существует множество подходов к прогнозированию динамики развития индексов: статистика, технический анализ, фундаментальный анализ и линейная регрессия. Однако, эти способы представляют стандарт базового уровня. Также комбинация этих подходов позволяет минимизировать ошибки в прогнозировании. Кроме того, многие из этих методов используются для препроцессорной обработки исходных данных, и их результаты поступают на вход нейронных сетей.

Фундаментальный анализ. Фундаментальный анализ требует детального изучения производительности и рентабельности компании для определения цены акции. Анализируя экономические условия, конкурентный круг компании и другие факторы, можно определить ожидаемую доходность и риски акций. Этот тип анализа предполагает, что текущая и будущая цена акции зависит от внутренней стоимости ресурса. Преимуществами фундаментального анализа являются системный подход и возможность прогнозирования изменений графиков котировок. Компании и перспективы их роста связаны с текущей экономической и политической ситуацией. К сожалению, задача формализации внешних факторов, влияющих на компанию, в целях автоматизации не является тривиальной, а интерпретация этих знаний может быть субъективной. Тем не менее, фундаментальный анализ - метод, направленный на долгосрочную стабильность и рост.

Технический анализ. Технический анализ основывается на предположении, что история имеет циклический характер, и что динамика рынка может быть определена путем изучения закономерностей, произошедших в прошлом. Тем не менее, технический анализ считается спорным подходом, так как противоречит гипотезе эффективного рынка, и подвергается критике, поскольку данный подход является субъективным. Человеческий фактор при интерпретации графиков играет существенную роль.

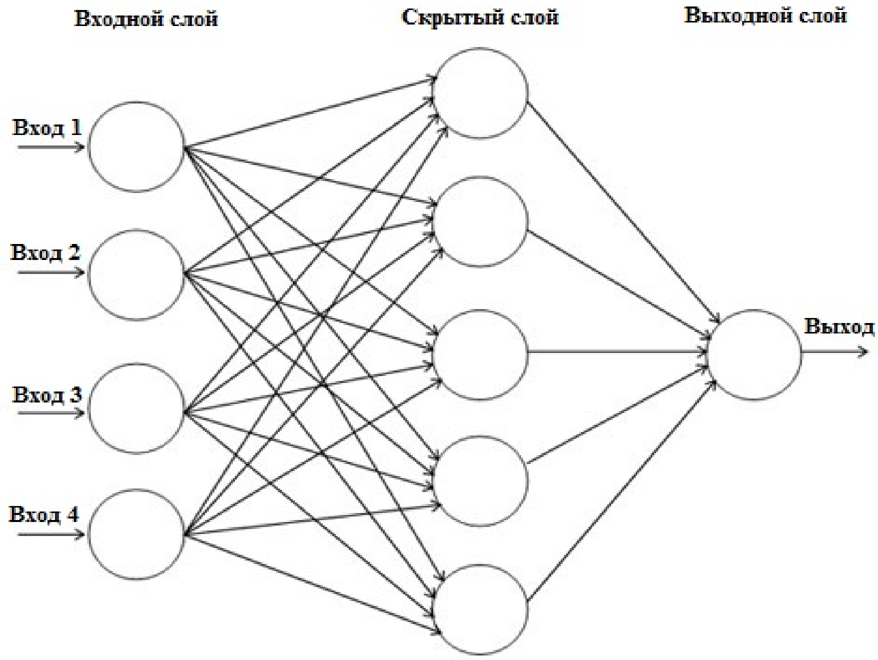
При техническом анализе графики котировок или индексов используются для выявления тенденций. Существует множество технических индикаторов, полученных из анализа графиков, которые могут быть формализованы в торговые правила или использованы в качестве входных данных нейронных сетей. В категорию технических индикаторов входят показатели фильтрации, стохастический осциллятор, японские свечи, линии Боллинджера, линии тренда, теория циклов, индексы объема, волновой анализ, а также анализ паттерна. Индикаторы могут предоставить информацию на короткий или длительный срок, помочь определить тенденции или циклы на рынке, или указать на прочность цены акции с использованием уровней поддержки и сопротивления.

Прогнозирование временных рядов. Прогнозирование временных рядов основано на анализе предыдущих данных, влияющих на оценки показателей в будущем. Данный метод моделирует нелинейную функцию с помощью рекуррентного соотношения, полученного из предыдущих значений. Рекуррентное соотношение может быть использовано для прогнозирования новых значений во временном ряду. Существует два основных типа прогнозируемых временных рядов: одномерный и многомерный. Одномерные модели Бокса-Дженкинса (ARIMA) содержат только одну переменную в уравнении повторения. Уравнения, используемые в модели, содержат предыдущие значения скользящих средних и значений котировок. Модель ARIMA показывает приемлемые результаты для краткосрочного прогнозирования, но требует большой набор данных. Многомерные модели имеют более одной переменной в уравнениях. В целом, прогнозирование временных рядов обеспечивает приемлемую точность на коротких периодах времени, но точность прогнозирования временных рядов резко уменьшается при увеличении периода прогноза.

Многослойная нейронная сеть прямого распространения

В данной статье для прогнозирования индексов будет рассмотрена многослойная нейронная сеть прямого распространения (Multi-Layer Feed Forward Neural Network) или MLP, которая является наиболее используемой моделью из-за ее робастной (нечувствительность к различным отклонениям и неоднородностям в выборке) аппроксимации поведения [1]. Как показано на рисунке, MLP содержит входной слой, выходной слой и по меньшей мере один скрытый слой. Каждый слой имеет определенное количество нейронов, которые соединяются с нейронами следующего слоя.

Входной сигнал проходит через сеть в прямом направлении, слой за слоем. На рисунке приведен MLP с 4 входами, 1 выходом, и 1 скрытым слоем.



На входы сети подаётся вектор значений признаков объекта, по которым будет производится прогнозирование. На выходах сети формируется выходной вектор значений, который, если нейронная сеть имеет подходящую структуру и правильно обучена, может быть интерпретирован, как прогноз будущих значений некоторых величин. Структура нейронной сети подбирается, исходя из решаемой задачи.

Также существует два аспекта, которые должны быть продуманы при моделировании скрытых слоев нейронной сети: количество скрытых слоев и количество нейронов в каждом скрытом слое модели MLP [2]. Задачи, требующие двух скрытых слоев и более, встречаются редко. Несмотря на то, что скрытые слои не взаимодействуют непосредственно с внешней средой, они влияют на конечный результат. Таким образом, необходимо тщательно выбирать количество нейронов в скрытом слое при разработке модели. Если количество скрытых нейронов недостаточно, то число ошибок обучения оказывается высоким из-за некорректной классификации обучающих данных. С другой стороны, если число нейронов скрытого слоя является слишком большим, может возникнуть явление оверфиттинга или переобучения, другими словами нейронная сеть перестает выполнять обобщение [3].

Несколько экспериментов показали, однако, что для различных приложений и различных примеров тестовых данных, достаточно одного скрытого слоя при условии, что общее количество нейронов скрытого слоя корректно задано[4] [5].

Также существует множество эмпирических методов определения необходимого количества нейронов скрытых слоев. Некоторые из них:

- Количество скрытых нейронов должно находиться между числом нейронов входного слоя и числом нейронов выходного слоя.
- Количество скрытых нейронов должно быть примерно равно 2/3 числа нейронов входного слоя плюс количество нейронов выходного слоя.
- Количество скрытых нейронов должно быть меньше удвоенного числа нейронов входного слоя.
- Необходимое число нейронов скрытых слоях можно определить по формуле, являющейся следствием из теоремы Колмогорова-Арнольда-Хехт-Нильсена:

$\frac{N_y \cdot N_p}{1+\log(N_p)} \le N_w \le N_y \cdot \left(\frac{N_p}{N_x}+1\right) \cdot (N_x + N_y + 1) + N_y$, где N_y - размерность выходного сигнала, N_p - число элементов обучающей выборки, N_x - размерность входного сигнала, N_w - необходимое число синаптических связей. Оценив с помощью этой формулы необходимое число синаптических связей N_w, можно рассчитать необходимое число нейронов в скрытых слоях. Например, число нейронов скрытого слоя двухслойной MLP будет равно: $N = \frac{N_w}{N_x + N_y}$

Основные элементы модели искусственного нейрона:

1. Набор сигналов на входах нейрона: $x_0, x_1, \ldots, x_n$ с соответствующими весами: $w_0, w_1, \ldots, w_n$. Веса представляют из себя "знания", которые содержит нейронная сеть по заданной учебной информации. Значения весов будут непосредственно влиять на значения выхода нейронной сети.

2. Блок вычисления взвешенной суммы значений входных сигналов: $v = \sum_{i=1}^{i=n} w_i \cdot x_i + w_0 \cdot x_0$, где (n - количество входных синапсов). Вход x_0 и соответствующий вес w_0 необходимы для инициализации нейрона [8]. Под инициализацией подразумевается смещение активационной функции нейрона по горизонтальной оси, то есть формирование порога чувствительности нейрона. То есть вектор входных значений сигнала X умножается на матрицу весов W. Результат суммирования V поступает на вход функции активации.

3. Функция активации для ограничения выходного сигнала нейрона: $o = \Phi(x)$. Функция активации генерирует выходной сигнал в соответствии с вычисленной взвешенной суммой значений входных сигналов. В общем виде, выходной сигнал каждого

нейрона может быть определен следующим образом: $o = \Phi\left(\sum_{i=1}^{i=n} w_i \cdot x_i + w_0 \cdot x_0\right)$. Важно подчеркнуть, что, если при обучении используется метод обратного распространения ошибки, то функция активации должна быть непрерывно дифференцируема на всей числовой оси. Данное требование обусловлено тем, что в методе обратного распространения ошибки требуется вычисление градиента функции активации. Как правило используется нелинейность, которая удовлетворяет этому требованию, а точнее сигмоидальная нелинейность, задающаяся логистической функцией: $\Phi(v) = \frac{1}{1 + e^{-\alpha \cdot v}}$, где α - параметр наклона сигмоидальной функции активации.

Анализ алгоритмов обучения

Обучение является средством, с помощью которого регулируются веса и пороговые значения нейронной сети. Метод обратного распространения ошибки - форма контролируемого обучения, где по алгоритму обучения получается ожидаемый выход. Метод обратного распространения ошибки может быть эффективной способом обучения для сети прямого распространения, простой рекуррентной сети и других видов нейронных сетей.

Алгоритм обратного распространения ошибки (Backpropagation). Алгоритм обратного распространения ошибки - метод обучения, включающий в себя обобщенное правило дельты.

В общем случае обучение нейронной сети состоит в том, чтобы найти некую функциональную зависимость $y = F(x)$, x – входной набор данных, а y – выходной. Подобная задача имеет бесконечное множество решений при ограниченном наборе входных данных. Поэтому при обучении возникает задача минимизации функции ошибки нейронной сети, получаемую методом наименьших квадратов: $E(w) = \frac{1}{2}\sum_{j=1}^{j=n}(o_j - t_j)^2$, где o_j - значение выхода j-го нейрона нейронной сети, t_j - целевое значение j-го выхода, n – число нейронов в выходном слое.

В методе обратного распространения ошибки обучение нейронной сети производится методом градиентного спуска, на каждой итерации изменение веса рассчитывается по формуле: $\Delta w_{ij} = -\alpha \cdot \frac{\partial E}{\partial w_{ij}}$, где $0 < \alpha < 1$ − параметр, определяющий скорость обучения, w_{ij} - вес, стоящий на ребре, соединяющем i-й и j-й узлы.

Частная производная функции ошибки по весу w_{ij} считается с использованием «цепного правила» (chain rule): $\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \cdot \frac{\partial o_j}{\partial v_j} \cdot \frac{\partial v_j}{\partial w_{ij}}$, где v_j - взвешенная сумма входных сигналов. При этом последний множитель $\frac{\partial v_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}}\left(\sum_{k=1}^{k=n} w_{kj} \cdot x_k + w_0 \cdot x_0\right) = x_i$. Первый множитель представим в виде $\frac{\partial E}{\partial o_j} = \sum_k \frac{\partial E}{\partial o_k} \cdot \frac{\partial o_k}{\partial v_k} \cdot \frac{\partial v_k}{\partial o_j} = \sum_k \frac{\partial E}{\partial o_k} \cdot \frac{\partial o_k}{\partial v_k} \cdot w_{jk}^{(n+1)}$, где k – число нейронов в n+1 слое.

Далее вводится вспомогательная переменная: $\delta_j^{(n)} = \frac{\partial E}{\partial o_j} \cdot \frac{\partial o_j}{\partial v_j}$. Тогда с использованием значений w_{ij} и δ_j (n+1)-го слоя рекурсивно определяются формулы для n-го слоя: $\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} \cdot w_{jk}\right] \cdot \frac{\partial o_j}{\partial v_j}$.

Формула выходного слоя определяется так: $\delta_j^{(N)} = \left(o_i^{(N)} - t_i\right) \cdot \frac{\partial o_i}{\partial v_i}$, где N – число слоев нейронной сети, t - целевое вектор, то есть вектор содержащий правильные ответы сети.

Формула $\Delta w_{ij} = -\alpha \cdot \frac{\partial E}{\partial w_{ij}}$ в окончательном виде представляется так: $\Delta w_{ij}^{(n)} = -\alpha \cdot \delta_j^{(n)} \cdot x_i^{(n)}$

Полный алгоритм обучения нейронной сети:

1. Инициализировать w_{ij} маленькими случайными значениями, $\Delta w_{ij} = 0$
2. Подать x_i на вход нейронной сети и определить значения выходов
3. Рассчитать для выходного слоя нейронной сети по формуле $\delta_j^{(N)} = \left(o_i^{(N)} - t_i\right) \cdot \frac{\partial o_i}{\partial v_i}$ и рассчитать изменения весов выходного слоя N по формуле $\Delta w_{ij}^{(n)} = -\alpha \cdot \delta_j^{(n)} \cdot x_i^{(n)}$. На этом этапе поступают векторы из обучающей выборки.
4. Рассчитать по формулам $\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} \cdot w_{jk}\right] \cdot \frac{\partial o_j}{\partial v_j}$ и $\Delta w_{ij}^{(n)} = -\alpha \cdot \delta_j^{(n)} \cdot x_i^{(n)}$ соответственно и для остальных слоев нейронной сети, $n = \overline{N-1,1}$
5. Скорректировать все веса по формуле $w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t)$
6. Если ошибка существенна, перейти к пункту 2.

Алгоритм эластичного распространения (Resilient propagation). Одной из проблем алгоритма обратного распределения ошибки является долгий процесс обучения. На данный момент существую алгоритмы ускоряющие процесс обучения такие, как QuickProp, метод Левенберга-Маркара, метод сопряженных градиентов. Также проблемой алгоритма обратного распространения ошибки являются недостатки метода градиентного спуска, использующегося в данном алгоритме обучения.

В алгоритме эластичного распространения в отличие от обратного распространения ошибки используются только знаки частных производных для изменения значений весовых коэффициентов. В данном алгоритме задействовано так называемое «обучение по эпохам», коррекция значений весов осуществляется уже после предъявления нейронной сети примеров обучающей выборки.

Для выявления величины коррекции используется правило:

$$\Delta_{ij}^{(t)} = \begin{matrix} \eta^{+} \cdot \Delta_{ij}^{(t)}, \dfrac{\partial E^{(t)}}{\partial w_{ij}} \cdot \dfrac{\partial E^{(t-1)}}{\partial w_{ij}} > 0 \\ \eta^{-} \cdot \Delta_{ij}^{(t)}, \dfrac{\partial E^{(t)}}{\partial w_{ij}} \cdot \dfrac{\partial E^{(t-1)}}{\partial w_{ij}} < 0 \end{matrix},$$

где $0 < \eta^{-} < 1 < \eta^{+}$, Δ_{ij} - величина коррекции.

Если на текущем шаге изменился знак частной производной по соответствующему весу w_{ij}, то последнее изменение было большим, и был пропущен локальный минимум. Следовательно, необходимо вернуть предыдущее значение весового коэффициента, а величину изменения уменьшить на η^{-} :

$$\Delta w_{ij}(t) = \Delta w_{ij}(t) - \Delta_{ij}^{(t-1)}.$$

Если на текущем шаге не изменился знак частной производной, то необходимо увеличить значение коррекции для достижения более быстрой сходимости.

Для вычисления значения коррекции весов используется следующее правило:

$$\Delta w_{ij}(t) = \begin{matrix} -\Delta_{ij}^{(t)}, \dfrac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)}, \dfrac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0, \dfrac{\partial E^{(t)}}{\partial w_{ij}} = 0 \end{matrix}.$$

Весовой коэффициент уменьшается на значение величины коррекции, если частная производная положительна. Весовой коэффициент увеличивается на значение величины коррекции, если частная производная отрицательна.

$$w_{ij}(t+1) = \Delta w_{ij}(t) + w_{ij}(t)$$

Алгоритм

1. Проинициализировать величину коррекции Δ_{ij}

2. Предъявить нейронной сети все примеры из обучающей выборки и вычислить частные производные.

3. Подсчитать новое значение Δ_{ij} по формулам $\Delta_{ij}^{(t)} = \begin{cases} \eta^{+} \cdot \Delta_{ij}^{(t)}, \dfrac{\partial E^{(t)}}{\partial w_{ij}} \cdot \dfrac{\partial E^{(t-1)}}{\partial w_{ij}} > 0 \\ \eta^{-} \cdot \Delta_{ij}^{(t)}, \dfrac{\partial E^{(t)}}{\partial w_{ij}} \cdot \dfrac{\partial E^{(t-1)}}{\partial w_{ij}} < 0 \end{cases}$ и

$$\Delta w_{ij}(t) = \begin{cases} -\Delta_{ij}^{(t)}, \dfrac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)}, \dfrac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0, \dfrac{\partial E^{(t)}}{\partial w_{ij}} = 0 \end{cases}.$$

4. Скорректировать веса по формуле $w_{ij}(t+1) = \Delta w_{ij}(t) + w_{ij}(t)$.

5. Если условие останова не выполнено, то перейти к шагу 2.

Входные данные для нейронных сетей в задачах прогнозирования биржевых индексов

На вход нейронной сети подаются исторические данные фондовых индексов, то есть значения индексов в определенные моменты времени в интервале, например, одного торгового дня.

Одним из важных эвристических условий повышения точности выходных значений нейронной сети является нормализация входных данных. Каждое из них должно быть предварительно обработано таким образом, чтобы среднее значение, усредненное по всей обучающей выборке, было близко к нулю, либо мало по сравнению с ее стандартным отклонением. Для нормализации каждого диапазона индексов на интервале [-1, 1] необходимо использовать следующую формулу: $x_i = \dfrac{(x_i - Min(X))}{(Max(X) - Min(x))}$, где x_i - элемент входной выборки X, $Min(X), Max(X)$ - минимальное и максимальное значение выборки X соответственно.

Заключение

В статье проведен анализ существующих методов прогнозирования. Приведен анализ модели многослойной нейронной сети прямого распространения, которая может быть использована для прогнозирования биржевых индексов. Проведен анализ алгоритмов обучения нейронных сетей и применимость к задаче прогнозирования биржевых индексов.

Список литературы

1. Mateo F., Carrasco J. J., Sellami A., Millán-Giraldo M., Domínguez M., Soria-Olivas E. Machine learning methods to forecast temperature in buildings // Expert Systems with Applications, 2013. Vol. 40, no. 4. P. 1061-1068. DOI: 10.1016/j.eswa.2012.08.030.
2. Nassif A. B., Ho D., Capretz L. F. Towards an early software estimation using log-linear regression and a multilayer perceptron model // Journal of Systems and Software. 2013. Vol. 86, no. 1. P. 144–160. DOI: 10.1016/j.jss.2012.07.050.
3. Benardos P. G., Vosniakos G. C. Optimizing feedforward artificial neural network architecture // Engineering Applications of Artificial Intelligence. 2007. Vol. 20, no. 3. P. 365-382. DOI: 10.1016/j.engappai.2006.06.005.
4. Safi Y., Bouroumi A. An Evolutionary Approach for Optimizing Three Layer Perceptrons Architecture // International Conference on Multimedia Computing and Systems, 2012. P. 227- 231.
5. Feng C.J., Yu Z.S., Kingit U. Threefold vs . Fivefold Cross Validation in One-Hidden Layer and Two-Hidden-Layer Predictive Neural Network Modeling of Machining Surface Roughness Data // Journal of manufacturing systems. 2005. Vol. 24, no. 2. P. 93-107. DOI: 10.1016/S0278-6125(05)80010-X.
6. Lendasse A., de Bodt E., Wertz V., Verleysen M. Non-linear financial time series forecasting-application to the Bel 20 stock market index // European Journal of Economic and Social Systems. 2000. Vol. 14, no. 1. P. 81–91. DOI: 10.1051/ejess:2000110.
7. Lee K. J., Chi A. Y., Yoo S., Jin J. J. Forecasting Korean stock price index (kospi) using back propagation neural network model, Bayesian Chiao's model, and ASRIMA model // Academy of Information and Management Sciences Journal. 2008. Vol. 11, no. 2. P. 53–62.
8. Википедия. Статья «Искусственный нейрон». Раздел «Математическая модель». Режим доступа: https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D

1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD#.D0.9C.D0.B0.D1.82.D0.B5.D0.BC.D0.B0.D1.82.D0.B8.D1.87.D0.B5.D1.81.D0.BA.D0.B0.D1.8F_.D0.BC.D0.BE.D0.B4.D0.B5.D0.BB.D1.8C (дата обращения 06.05.2015).