О ПРОГНОЗЕ НА ОСНОВЕ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ И РАСЧЕТНОЙ НЕЛИНЕЙНОЙ ДИНАМО МОДЕЛИ

Сафиуллин Николай Тахирович

Описание базовой расчетной динамо модели здесь не приводится, потому что ее подробное описание есть в работе [N. Safiullin, N. Kleeorin, et al., JPP Vol. 84, 2018].

Мне не известен уровень знаний в области искусственных нейронных сетей людей, потенциально читающих этот документ, поэтому заранее прошу прощения, если в первых главах для Вас все будет слишком очевидно.

Если интересует итоговое краткое описание структуры прогноза в комбинации модель плюс нейронная сеть – то прошу перейти в содержании к пункту «6. Детальная схема…».

СОДЕРЖАНИЕ

1. Структура базового нейрона

В основе большинства современных искусственных нейронных сетей лежит базовая модель **перцептрона**, в виде математической модели нейрона (рис. 1) – узлов искусственной нейронной сети (далее – просто *нейронной сети*). Каждый из этих узлов только принимает и посылает сигналы. Их конфигурация при этом может существенно отличаться (как и число слоев).

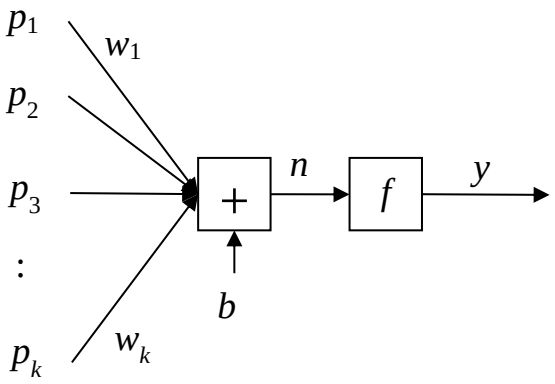

Рисунок 1 – Отдельный перцептрон и его конфигурация

Представленный на рисунке 1 нейрон имеет k входов p_k, смещение b (*bias*), весовые коэффициенты w_k, суммарный выход сети n, функцию активации/перехода f и выход y. Для такой модели нейрона выход y рассчитывается просто как:

$$y = f\left(\sum_{i=1}^{k} w_i p_i + b\right). \quad (1)$$

Если функция активации есть просто линейная функция, то выход нейрона представляет собой простую линейную взвешенную сумму его входов (плюс смещение).

Входами p_k, как мы увидим дальше, могут служить различные исходные данные, выходы других нейронов, а также и сам выход этого нейрона (обратная связь). Смещение b, чаще всего, равняется 1 (b = 1) или другому ненулевому числу, чтобы обеспечить удаленность взвешенной суммы от нулевого значения. Весовые коэффициенты w_k являются приспособляемыми и регулируемыми параметрами нейрона, то есть

именно их и калибрует каждый отдельный нейрон в ходе процесса обучения нейронной сети. Веса для одного нейрона образуют одномерный вектор значений $\{w_1, w_2, ..., w_k\}$, а для целой сети нейронов – матрицу весовых коэффициентов $\boldsymbol{W}$. При этом выражение (1) можно будет переписать в векторно-матричной форме как

$$y = f\left(\mathbf{W}\mathbf{p} + b\right). \tag{2}$$

Функция активации или функция перехода f может быть любой линейной или нелинейной функцией в зависимости от типа решаемых задач. Укажем только несколько ключевых из них. Функция ограничителя с резким порогом (hard limit transfer function) дает 0, если ее аргумент меньше нуля, и дает 1, если аргумент больше либо равен нулю (рис. 2a). Как вариант, иногда эту функцию изменяют на пределы от -1 до 1 или других констант. Линейная функция (рис. 2b) просто передает взвешенную сумму входов без изменений $y = \sum w_i p_i + b$. Сигмоидная (точнее, лог-сигмоидная) функция активации (рис. 2c) определяется формулой:

$$y = f\left(n\right) = \frac{1}{1 + e^{-n}} \ . \tag{3}$$

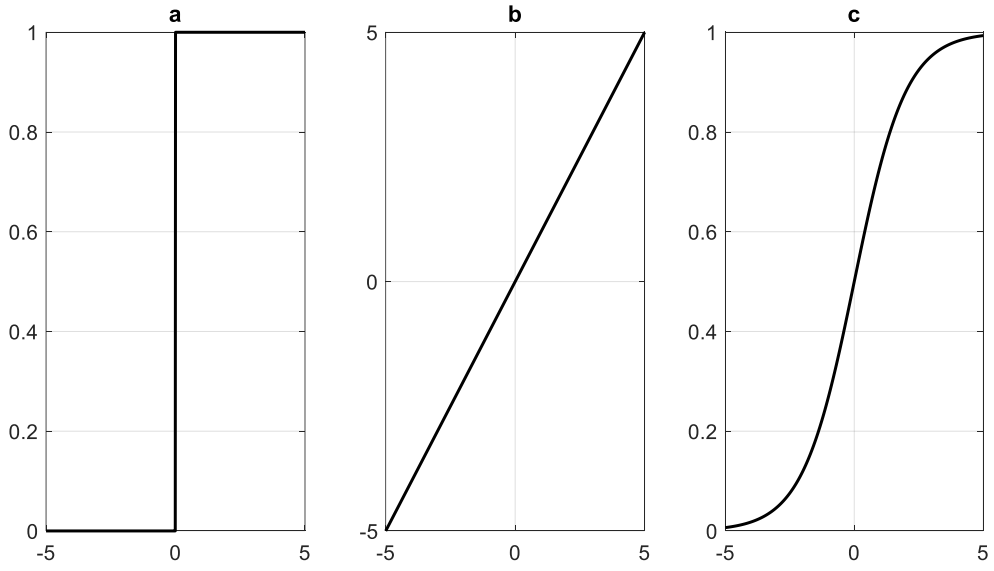

Рисунок 2 – Основные функции активации, при b=0: a – функция ограничителя с резким порогом, b – линейная, c – лог-сигмоидная функция

2. Слой нейронов

Одиночный слой нейронов, состоящий из S базовых нейронов, представлен на рисунке 3. Стоит отметить, что каждый из входов p_k подается на каждый нейрон. В самом деле, если эти входы для нейрона будут не важны, то он просто обнулит их весовые коэффициенты.

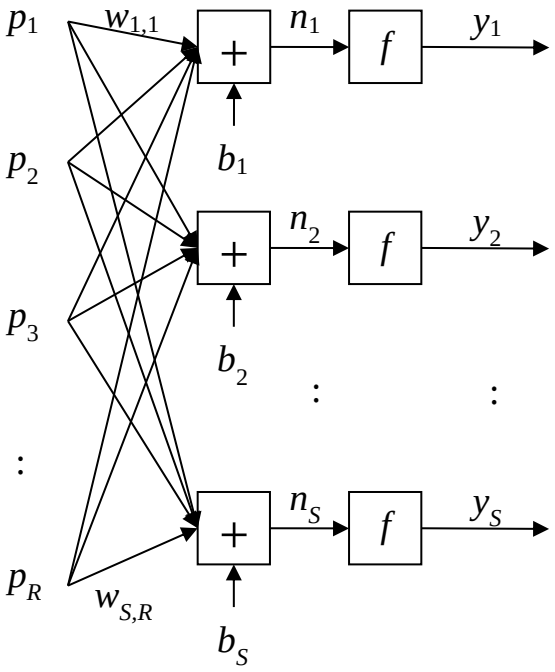

Рисунок 3 – Одиночный слой из S нейронов

В этой схеме выходы y_k могут объединяться в общий выход с помощью некоторой функции (или же объединяться на уровне n_k), а могут служить входами для следующего слоя нейронов. При этом число входов R чаще всего не совпадает с числом нейронов S в заданном слое. Для такого одиночного слоя нейронов весовые коэффициенты задаются матрицей $\mathbf{W}$:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}. \qquad (4)$$

При этом выражение (2) для выхода нейронного слоя остается в силе. Рисунок 3 громоздок, поэтому в сокращенной форме его можно свести к изображению на рисунке 4, называемым одним простым слоем нейронов.

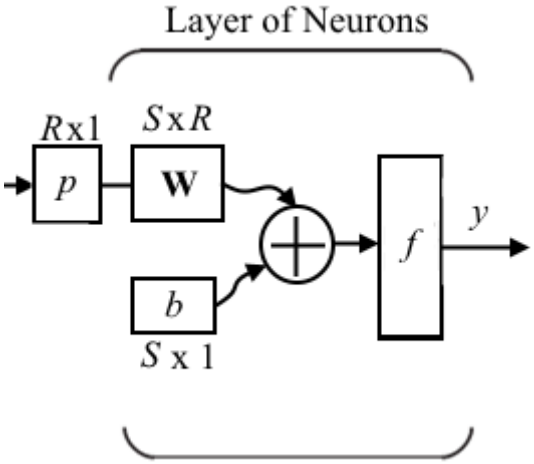

Рисунок 4 – Одиночный простой слой из *S* нейронов на *R* входов

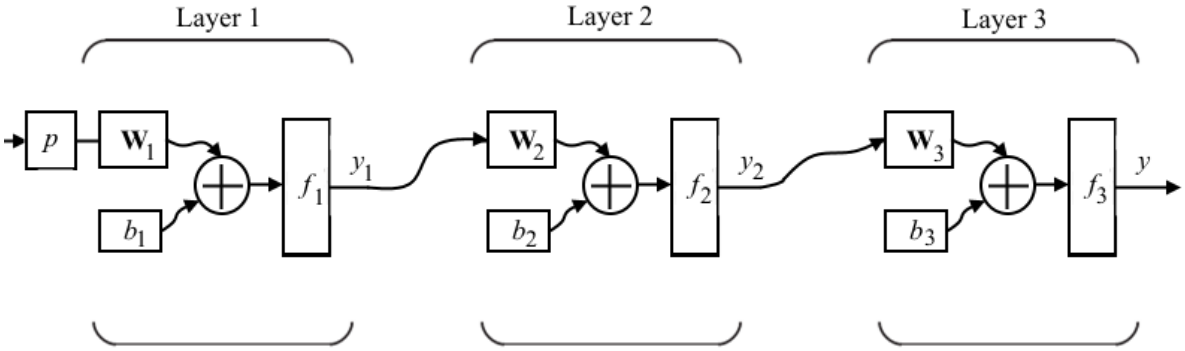

Рисунок 5 – Простая нейронная сеть из 3 слоев

Для создания многослойной нейронной сети выходы одного слоя подают на входы последующего, образуя сложную структуру связи отдельных нейронов и функций активации. Например, на рисунке 5 представлена трехслойная нейронная сеть, для которой суммарный выход определяется выражением:

$$y = f_3\left[\mathbf{W}_3 f_2\left\{\mathbf{W}_2 f_1\left(\mathbf{W}_1\mathbf{p}+\mathbf{b}_1\right)+\mathbf{b}_2\right\}+\mathbf{b}_3\right]. \qquad (5)$$

Многослойные искусственные нейронные сети обладают значительно более мощными возможностями, чем простые однослойные, поэтому на практике обычно применяют именно их, хотя рост числа слоев приводит к существенному повышению вычислительной сложности соответствующей нейронной сети. Кроме представленной структуры связи нейронных слоев, бывают еще и дополнительные – на основе рекуррентных соотношений, наличие обратной связи, наличие интеграторов и т.д.

3. Процесс обучения нейронов

Как уже было сказано ранее, весовые коэффициенты (4) слоя нейронов являются параметрами соответствующей нейронной сети, которые мы и хотим установить/откалибровать в ходе процесса обучения искусственной нейронной сети. В случае процесса **обучения с учителем**, мы хотим при заданных входах, смещении и функции активации найти такие весовые коэффициенты, которые бы наилучшим образом давали известную выходную функцию. То есть при обучении нейронной сети с учителем у нас имеется некоторое обучающее множество пар

$$\{\mathbf{p}_1,\mathbf{t}_1\},\{\mathbf{p}_2,\mathbf{t}_2\},\ldots\{\mathbf{p}_q,\mathbf{t}_q\}, \tag{6}$$

где **p** – входы нейронной сети, которым соответствуют целевые выходы **t**.

При подаче на входы нейронной сети входов **p** мы получаем выходные значения, которые затем сравниваются с целями **t**. В зависимости от близости полученных результатов к желаемым происходит коррекция весовых коэффициентов – обычно по методу обратной связи, либо с помощью других способов обратного распространения ошибки. С математической точки зрения такой процесс обучения нейронной сети полностью соответствует многопараметрической задачи нелинейной оптимизации $y = \arg\min_{y\in A} \mathbf{W}(y,t)$.

Кроме обучения с учителем, существуют и другие принципы обучения, например, обучение без учителя (самоорганизующиеся сети) и обучение с подкреплением (на основе штрафов и поощрений), и т.д.

Каким же образом калибровать сами весовые коэффициенты? Имеется множество методик (методы градиентного спуска, метод минимизации эмпирического риска и т.д.), но их общий принцип заключается в оценке значения ошибки *e* (*mse*, *sse* и любые другие) между полученным выходом нейронной сети *y* и целевым значением обучающей выборки *t*. Весовые коэффициенты, изначально заданные случайно или константой, затем изменяются пропорционально этой ошибке *e*, а весь алгоритм обучения повторяется до тех пор, пока не будет найден минимум отличия между выходом *y* и целевым значением обучающей выборки *t*.

Но именно здесь и кроется самая частая проблема и ошибка всех, реализующих нейронные сети. Если слишком хорошо минимизировать ошибку на **обучающем множестве**, то появляется проблема «**переобучения**» – явление, когда построенная модель хорошо объясняет примеры из обучающей выборки, но относительно плохо работает на примерах, не участвовавших в обучении (на примерах из **тестовой выборки**). Это связано с тем, что при построении сети в обучающей выборке обнаруживаются некоторые случайные закономерности, которые отсутствуют в генеральной совокупности. Иными словами, нейронная сеть запоминает огромное количество всех возможных примеров вместо того, чтобы научиться подмечать особенности.

Именно поэтому все реально работающие прогнозные нейронные сети обучают на средних значениях временного ряда наблюдений в обучающей выборке вместо мгновенных – чтобы избежать переобучения, чтобы прогноз был в принципе работоспособен.

4. Прогноз временного ряда только с помощью нейронной сети

Один из способов построения прогноза для заданного временного ряда требует построить некоторую формальную регрессионную модель с неизвестными коэффициентами при условии минимизации функции ошибки (например, среднеквадратичного отклонения прогнозных точек). Существует множество конфигураций нейронных сетей, отвечающих данной задаче, однако за основу была выбрана структура нейронной сети, называемая нелинейной авторегрессией **NAR** (**N**onlinear **A**uto**R**egressive), схема которой приведена на рисунке 6.

Из рисунка 6 видно, что NAR-сеть состоит из двух слоев: скрытого слоя и выходного слоя, имеет обратную связь с выхода на вход (рекуррентная кольцевая схема для возможности прогнозирования), смещение принято равным единице. Тренировка сети происходит на основе обучения с учителем, где обучающей выборкой служат отсчеты временного ряда. Функция активации скрытого слоя является лог-сигмоидой, функция активации выходного слоя является линейной.

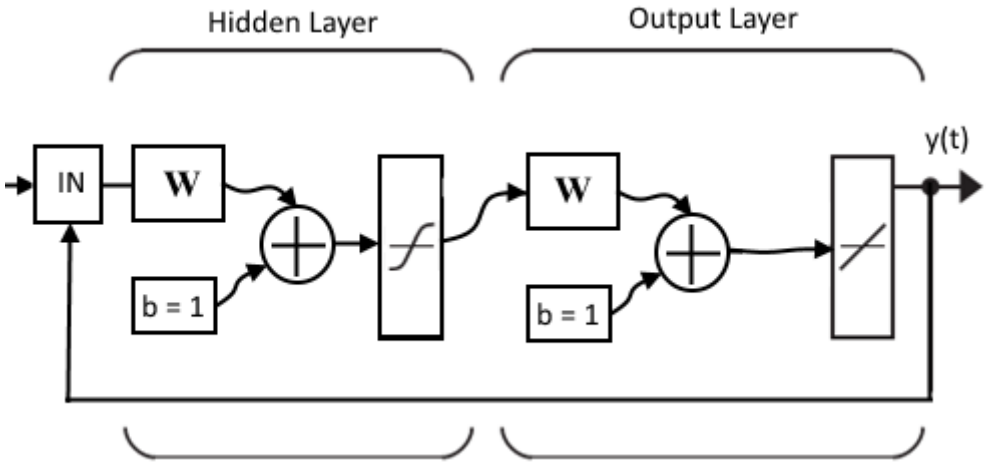

Рисунок 6 – Нейронная сеть NAR для прогнозирования временных рядов

Подобная простая схема NAR-сети в качестве входов использует сами значения исходного временного ряда, то есть нейронная сеть сопоставляет прошлые значения ряда с прогнозируемым наблюдением. Но именно поэтому такая сеть плохо приспособлена сама по себе к прогнозу реальных нестационарных временных рядов, к которым относится ряд чисел Вольфа: NAR-сеть не учитывает никаких характерных особенностей исходного временного ряда (амплитудные модуляции, циклы и т.д.), не может предсказать амплитуду, время начала и т.п. для нового цикла. С помощью чистой NAR-сети можно получить прогноз текущего цикла, для которого прошло уже несколько лет, но несколько циклов спрогнозировать ей невозможно.

5. Общая схема прогноза в комбинации модель плюс искусственная нейронная сеть

Все современные прогнозы ряда чисел Вольфа построены по принципу Ассимиляции Данных (Data Assimilation), то есть они с каждым новым наблюдением корректируют свой некоторый базовый прогноз, тем самым значительно повышая его точность. Следует взглянуть на искусственную нейронную сеть с этой точки зрения, тогда большое количество проблем удастся решить, если **сама нейронная сеть не прогнозирует значения временного ряда, а лишь занимается их коррекцией по уже имеющимся наблюдениям**.

Для этого можно доработать NAR-сеть (рис. 6) из чисто прогнозной системы в схему коррекции некоторого первичного прогноза. Пусть у нас есть базовый прогноз временного ряда (массив чисел) $x(t)$ на основе некоторой модели. Мы хотим скорректировать его по имеющимся наблюдениям и соответствующим предыдущим значениям модельного прогноза $x(t-1), x(t-2), ..., x(t-d)$. Тогда такая схема будет называться **NARX**-сетью (**N**onlinear **A**uto**R**egressive with e**X**ogenous input). Общая схема этой двухслойной нейронной сети с дополнительным входом приведена на рисунке 7.

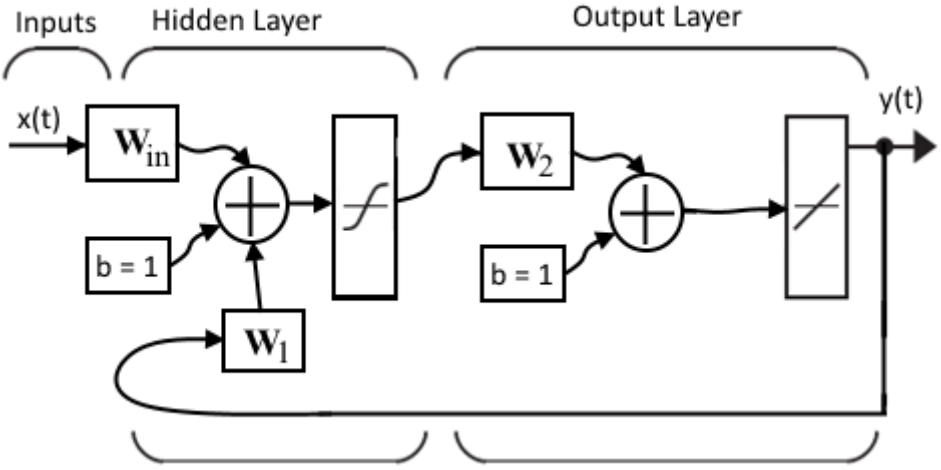

Рисунок 7 – Нейронная сеть NARX для коррекции базового прогноза $x(t)$

6. Детальная схема прогноза в комбинации модель плюс искусственная нейронная сеть

Для прогнозирования временного ряда среднемесячных чисел Вольфа используется NARX-сеть на основе коррекции базового прогноза, полученного при численном решении нелинейной динамо модели магнитного поля Солнца [N. Safiullin, N. Kleeorin, et al., JPP Vol. 84, 2018]. После обработки расчетных результатов модели можно получить временной ряд модельного числа солнечных пятен, который нельзя сказать, чтобы точно повторял все наблюдаемые солнечные циклы, но тем не менее имеет с ними очень высокую корреляцию (выше 85%), в том числе и по амплитуде/по форме этих циклов (см. рисунок/Figure 1 из [N. Safiullin, N. Kleeorin, et al., JPP Vol. 84, 2018, pp. 7]). Главное, что модель может дать нам приближенные будущие циклы. А нейронная сеть поможет нам довести эти значения до реального прогноза на основе ассимиляции текущих современных наблюдений среднемесячного ряда чисел Вольфа.

Перепишем модель из рисунка 7 в терминах прогноза конкретно среднемесячного сглаженного ряда чисел Вольфа.

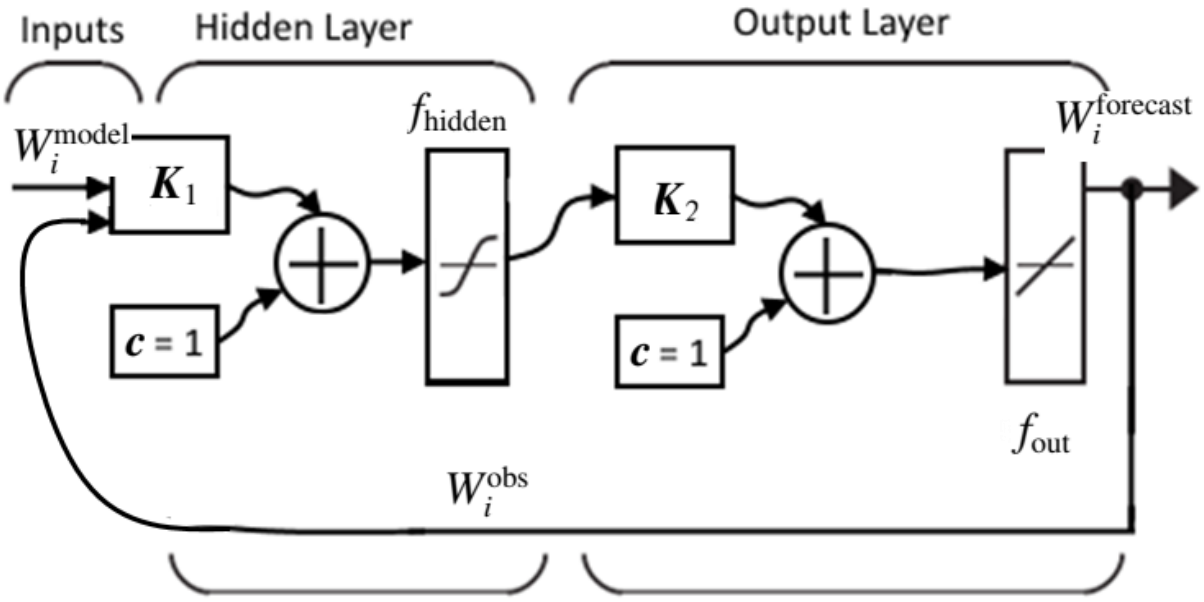

Рисунок 8 – Прогнозная нейронная сеть

Пусть базовый прогноз ряда чисел Вольфа, полученный от расчетов модели, обозначен как W_i^{model}, где i – соответствующая месячная временная метка. В «прошлом» для него всегда можно сопоставить реальное наблюдение среднемесячного сглаженного числа Вольфа W_i^{obs}. Тогда, с учетом схемы из рисунка 8 и вышеописанных выражений, общая формула прогноза описывается как:

$$W_i^{\text{forecast}} = f_{\text{out}}[\boldsymbol{K}_2 f_{\text{hidden}}(\boldsymbol{K}_1 \boldsymbol{w} + \boldsymbol{c}_1) + \boldsymbol{c}_2], \tag{7}$$

где $f_{hidden}(x) = \left[1 + \exp(-x)\right]^{-1}$, $\boldsymbol{K}_1$ – весовая матрица размерности 24 x 8 скрытого слоя нейронов, $\boldsymbol{K}_2$ – весовая матрица размерности 1 x 24 внешнего слоя нейронов, $\boldsymbol{c}_1$ & $\boldsymbol{c}_2$ – вектора смещения, $\boldsymbol{w}$ – это входной вектор 8 x 1, в котором скомбинированы 4 прошлых наблюдения $W_{i-1}^{\text{obs}}, \ldots, W_{i-4}^{\text{obs}}$ и соответствующие им по времени 4 модельных отсчета $W_{i-1}^{\text{model}}, \ldots, W_{i-4}^{\text{model}}$.

$W_i^{forecast}$ есть прогнозируемое значение на «сейчас» и, как видно из выражения (7), получается на основе модельного ряда W_i^{model} прошлых значений и с учетом их коррекции по имеющимся реальным наблюдениям в те же временные интервалы W_i^{obs}. Размерности векторов 4, 8 и другие – варьируемые, и могут быть больше/меньше для изменения топологии прогноза в целом, здесь приведены реальные значения для текущей рабочей сети.

В схеме, вроде бы, все понятно до тех пор, пока не начинается сам *чистый* прогноз (нет наблюдений для коррекции), то есть, когда текущая временная метка i принимает значения в будущем: i+1, i+2, i+3 и т.д. В этом случае для значений модели W_i^{model} ничего не меняется: расчет произведен с большим запасом на $W_{i+1}^{\text{model}}, W_{i+2}^{\text{model}}, W_{i+3}^{\text{model}}, \ldots$. А вот в качестве «наблюдений» принимается то, что прогнозная модель выдала на прошлых этапах: $W_i^{obs} = W_i^{forecast}, W_{i+1}^{obs} = W_{i+1}^{forecast}, \ldots$ Такую схему можно продолжать по одному шагу в «закрытом цикле» на любой период вперед. Ее устойчивость (форму цикла и т.п.) обеспечивают значения W_i^{model}, а краткосрочную точность – значения W_i^{obs}, которые обновляются с каждым новым появившимся наблюдением.

В качестве методики обучения сети был выбран алгоритм Левенберга-Марквардта, обучающей выборкой служил временной ряд среднемесячных чисел Вольфа, сглаженных 13-месячным окном.

Если быть совсем точным, то обучающей выборкой служили значения 19 и 20 циклов. Валидационная выборка бралась из того же ряда, как прогноз на следующий 21 цикл солнечной активности, тестовой выборкой служили прогнозы на два цикла солнечной активности 22 и 23, расчет точности проводился для всех выборок независимо.

Причина использования сглаженных значений указана на стр. 7 ранее: избежать переобучения сети, обеспечить прогнозу устойчивость и высокую точность. Прогнозировать конкретно мгновенные значения месячных чисел Вольфа не удалось еще никому – он слишком нестационарен.