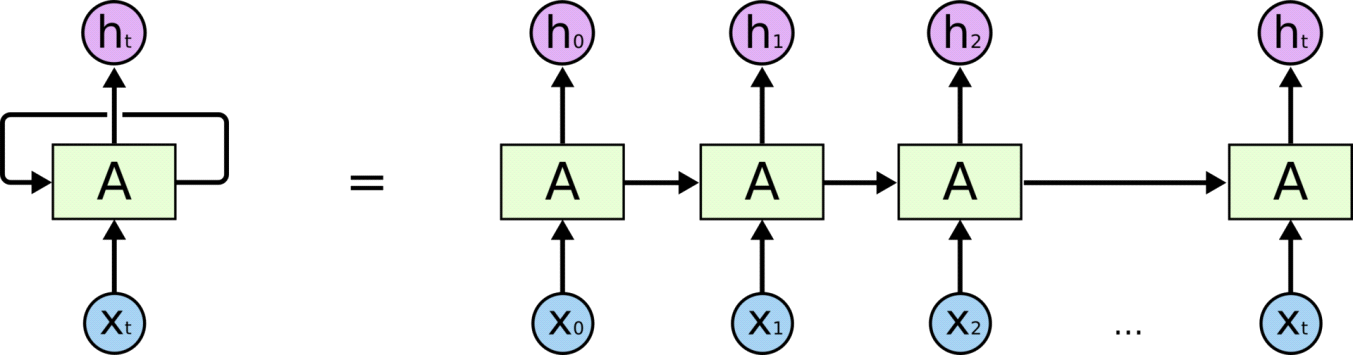
Делая прогнозы, люди не начинают думать с нуля каждую секунду. Часто для прогнозов применяется предыдущий опыт. Читая книгу, вы понимаете каждое новое слово исходя из предыдущих слов. Мы не забываем определенную информацию и не начинаем думать в каждый момент времени с нуля. Традиционные нейронные сети не обладают этим свойством и в этом их главный недостаток. Поэтому для решения данной проблемы используются рекуррентные нейронные сети - мощный тип нейронной сети, предназначенной для обработки серии событий во времени или последовательные пространственные цепочки.



**Рисунок 1**

Рекуррентную сеть можно рассматривать, как несколько копий одной и той же сети, каждая из которых передает информацию последующей копии. На рисунке 1 фрагмент нейронной сети A принимает входное значение xt и возвращает значение ht. Наличие обратной связи позволяет передавать информацию от одного шага сети к другому.

Сети с долгой краткосрочной памятью (LSTM) являются модификацией рекуррентных нейронных сетей, которые разработаны для избегания проблемы долговременной зависимости. Поэтому они хорошо подходят в ситуациях когда важные события разделены временными лагами с неопределённой продолжительностью и границами.

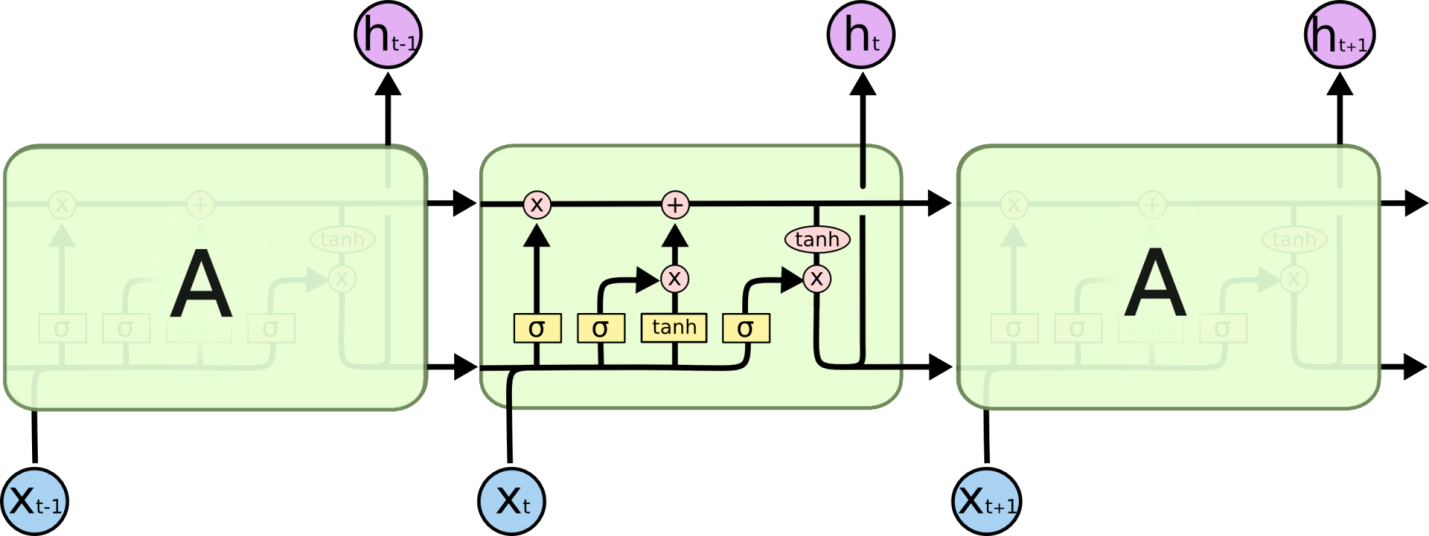
Основная концепция LSTM - это состояние ячейки и ее различные фильтры (gates). Состояние ячейки действует как транспортная магистраль, которая передает информацию по всей цепочке последовательности. Можно ассоциировать ее с понятием «памяти» сети. Таким образом, даже информация с более ранних временных шагов может пробиться к более поздним временным шагам, уменьшая действие краткосрочной памяти. Однако данная сеть может и удалять информацию из состояния ячейки с помощью специальных структур - фильтров. Фильтры позволяют решать, какая информация пропускается дальше в состояние ячейки, а какая должна быть забыта. Они состоят из слоя сигмоидальной нейронной сети и операции поточечного умножения. Сигмоидальная функция активации похожа на активацию tanh. Однако, вместо сжатия значений от -1 до 1, сжатие значений происхожит в диапазоне от 0 до 1. Таким образом можно обновить или забыть данные, потому что любое число, умноженное на 0, равно 0, в результате чего значения исчезают или «забываются». Любое число, умноженное на 1 - это то же значение, поэтому это значение остается тем же или «сохраняется». В LSTM сети три таких фильтра: фильтр забывания, входной и вы.

Первым идет слой фильтра "забывания". Этот слой решает, какую информацию следует выбросить или сохранить. Информация из предыдущего состояния ht-1 и информация из текущего ввода xt передаются через сигмоидную функцию. Она возвращает значения в диапазоне от 0 до 1. 1 означает “полностью сохранить”, а 0 – “полностью выбросить”.

Чтобы обновить состояние ячейки, в сети имеется слой входного фильтра. Сначала старое состояние ht-1 и текущий ввод xt передаются в сигмоидную функцию. Она решает, какие значения будут обновлены путем преобразования значений в диапазон 0 - 1. 0 означает, что информация не важна, а 1 - важна. Старое состояние ht-1 и текущий ввод также передаются в функцию tanh, чтобы сжать значения в диапазон -1 и 1. Затем выходное значение tanh и сигмоида перемножаются. Значения, полученные на выходе сигмоида определяют, какую информацию важно сохранить от tanh.

Теперь у сети достаточно информации для расчета состояния ячейки. Во-первых, состояние ячейки поточечно умножается на вектор забывания. Это дает возможность отбрасывать неинформативные значения в состоянии ячейки. Затем поступает выходной сигнал от входного фильтра и производится точечное сложение, которое обновляет состояние ячейки до новых значений, которые нейронная сеть считает релевантными. В результате получается новое состояние ячейки.

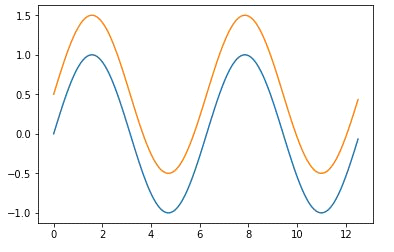
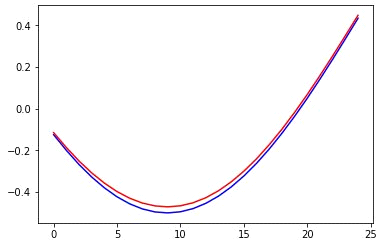
Выходные данные основаны на состоянии ячейки, к которому применяются некоторые фильтры. Сперва применяется сигмоидальный слой, который решает, какая информация из состояния ячейки будет выводиться. Затем значения состояния ячейки проходят через tanh-слой, чтобы получить на выходе значения из диапазона от -1 до 1, и перемножаются с выходными значениями сигмоидального слоя, что позволяет выводить только требуемую информацию.



**Рисунок 2. Структура LSTM сети**

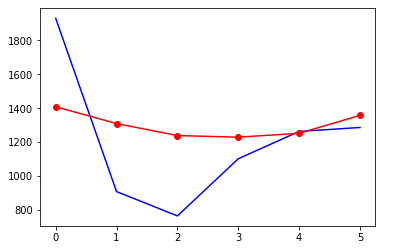
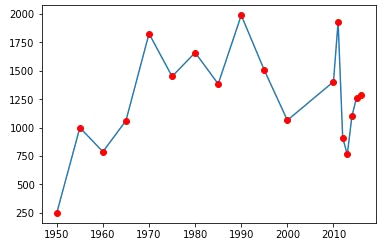
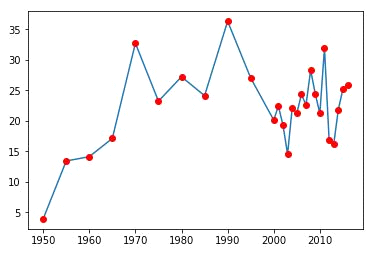
Попросту говоря, LSTM сети содержат несколько внутренних ячеек контекстуального состояния, которые действуют как ячейки долговременной или кратковременной памяти. Выходной сигнал сети модулируется состоянием этих ячеек. Это очень важное свойство, когда нам нужно, чтобы прогноз нейронной сети зависел от исторического контекста входных данных, а не только от самого последнего входного сигнала.

Исходя из описанной выше архитектуры сети, можно сделать вывод, что данная нейронная сеть больше всего подходит для задач прогнозирования. Для прогнозирования по аналогии на вход нейронной сети подавались значения первого ряда, а выходами служили значения второго, похожего ряда. Предварительно данные были нормализованы, а пропуски в выборках заполнены путем интерполяции. При настройке сети необходимо выбрать функцию потерь и алгоритм оптимизации. В качестве функции потерь была выбрана средняя квадратическая ошибка, как одна из наиболее часто используемых функций в задачах прогнозирования. Для оптимизации был выбран алгоритм Adam (adaptive moment estimation). Он сочетает в себе и идею накопления движения и идею более слабого обновления весов для типичных признаков. У данного алгоритма относительно низкие требования к памяти и он хорошо работает даже при небольшой настройке гиперпараметров. Сеть содержала 5 нейронов и обучалась в течении 1500 эпох. Результаты прогнозирования представлены на рисунках ниже.

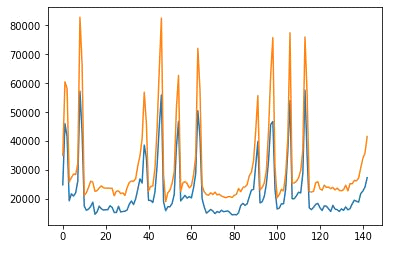
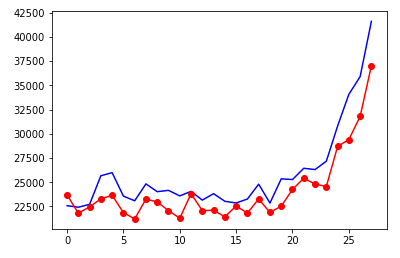
**Рисунок 3.** Красным цвет соответствует спрогнозированным значениям, а синий – реальным значениям

На искусственно смоделированных данных (синус) с большой обучающей выборкой (100 точек) сеть делает достаточно хороший прогноз. Если же данные реальные, а выборка небольшая, сети не удается найти аналогию.



**Рисунок 4.** Производство основных сельскохозяйственных культур / Урожайность основных сельскохозяйственных культур и результат прогнозирования второго ряда исходя из данных первого ряда (красный – прогноз, синий – реальный результат)

Однако, когда обучающая выборка содержит больше данных, прогноз получается достаточно точным. На рисунке ниже приведен пример прогнозирования продаж магазина сети Walmart исходя из данных продаж аналогичного магазина в другом штате.

**Линейная регрессия**

Линейная регрессия (англ. Linear regression) — используемая в статистике регрессионная модель зависимости одной (объясняемой, зависимой) переменной y от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) x с линейной функцией зависимости.

Регрессионная модель



где b — параметры модели, epsilon — случайная ошибка модели; называется линейной регрессией, если функция f(x,b) имеет вид



где  — параметры (коэффициенты) регрессии,  — регрессоры (факторы модели),  — количество факторов модели. Требуется найти .

Решим уравнение линейной регрессии при помощи МНК Метод наименьших квадратов (МНК, англ. Ordinary Least Squares, OLS) — математический метод, применяемый для решения различных задач, основанный на минимизации суммы квадратов отклонений некоторых функций от искомых переменных.

Оптимальным решением МНК является:



Получим искомые веса для линейной регрессии. Таким образом метод сможет найти зависимость между входными данными и ответами на них.

Воспользуемся данным методом для восстановления второго ряда по первому, известному (предсказанному) наперед, предполагая, что существует корреляция между рядами. Для этого в МНК подадим матрицу следующего вида:



Где вектор:  - значения второго ряда. Матрица - значения первого ряда, преобразованные в матрицу, как показано на рисунке. Вектор - искомые параметры регрессии. При реализации, можно искать не 3, а k параметром регрессии, тогда матрица  будет иметь k столбцов и заполняться аналогичным образом. Перебирая k, можно будет добиться лучших результатов.

Заметим, что модель не содержит вектора параметров X временных рядов - предполагается, что интервалы рядов, поданные линейной регрессии на обучение, совпадают. Если некоторые интервалы времени в обучающей выборке отсутствуют - их необходимо заполнить. Сделать это можно при помощи линейной интерполяции.

**Бутстрэп**

Будем использовать метод для оценки построенной линейной модели. Пусть дана конечная выборка Xℓ и ответы на ней Yℓ. Будем решать задачу линейной регрессии. Сгенерируем подвыборку с помощью бустрэппинга. Равномерно возьмем из выборки ℓ объектов с возвращением. Из-за возвращения среди них окажутся повторы; разрешив повторы, мы получим подвыборку такого же размера Xl. Повторив процедуру n раз, сгенерируем n подвыборок . Обучим по каждой из них линейную функцию регрессии, получив алгоритмы .

После посмотрим на распределение параметров полученных алгоритмов . Построим гистограмму, если вариация параметров очень большая, значит модель плохая. Решение об оптимальной вариации параметров принимается экспертом.

**Доверительные интервалы**

Мы можем построить доверительные интервалы погрешности предсказания, посмотрев на распределение ошибок. Составим ряд, состоящий из  где .

Восстановим плотность распределения ошибки. Решение об оптимальности ошибки принимается экспертом. На рисунках ниже показаны доверительное оценивание предсказания: красный - накопительная ошибка, с учетом самой худшей ошибки, которую видели. Зеленый - 80% накопительное доверительное оценивание. Оранжевый - оценивание для одного следующего предсказания.

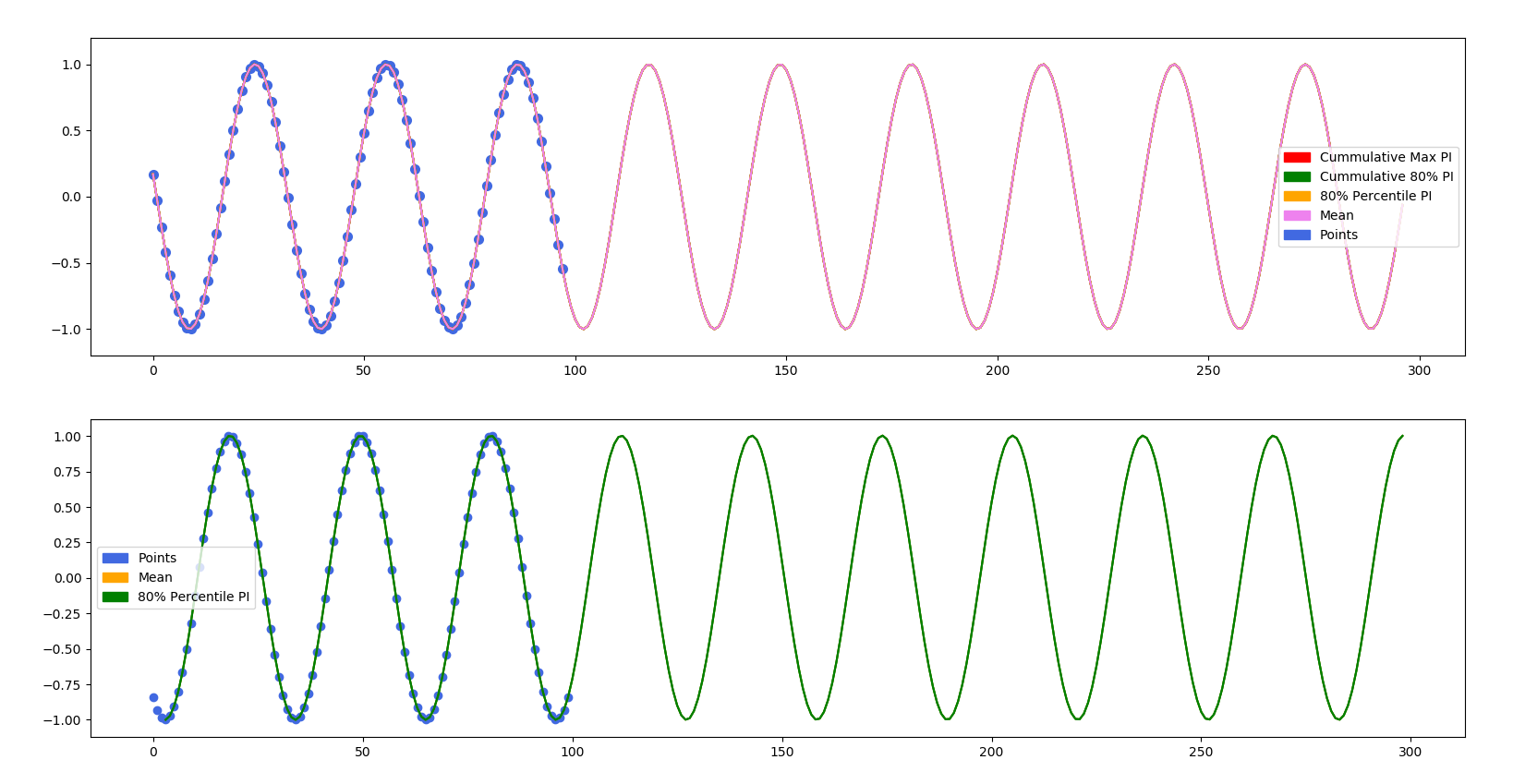
**Результаты**

Для демонстрации работы алгоритмов, сначала предскажем первый временной ряд, потом восстановим второй ряд по первому.

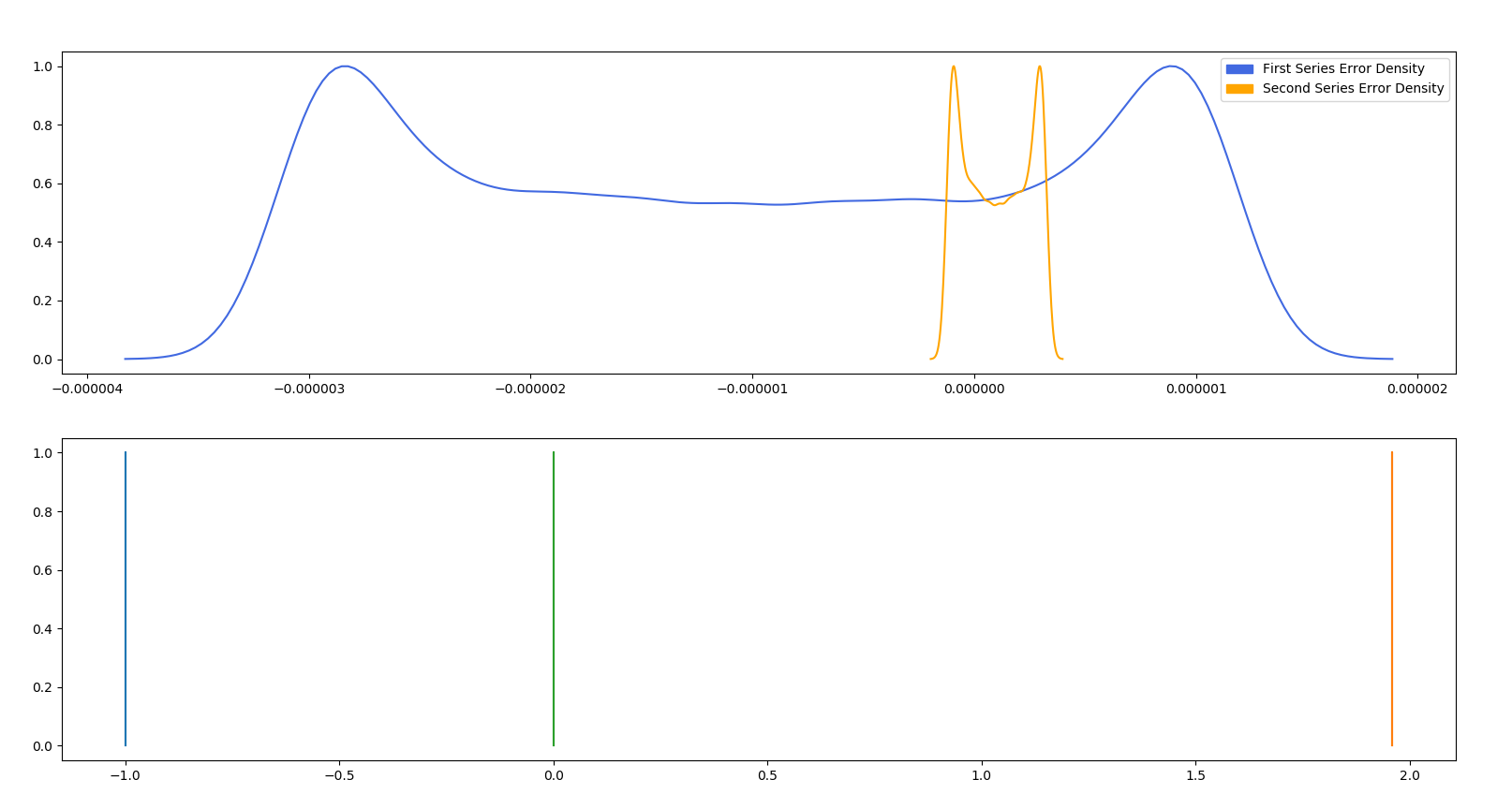
На графиках ниже: точками изображены известные значения временных рядов, эти значения подавались на вход модели.

Оранжевый, красный и зеленый – доверительные интервалы.  
На верхнем графике: розовая кривая – как обучилась выбранная модель и ее прогноз. На нижнем графике: оранжевая кривая - как обучилась выбранная модель и ее прогноз.

**Рисунок...** На рисунке изображен предсказанный первый ряд - синус и восстановленный по нему - косинус. Метод: линейная регрессия при k=2.

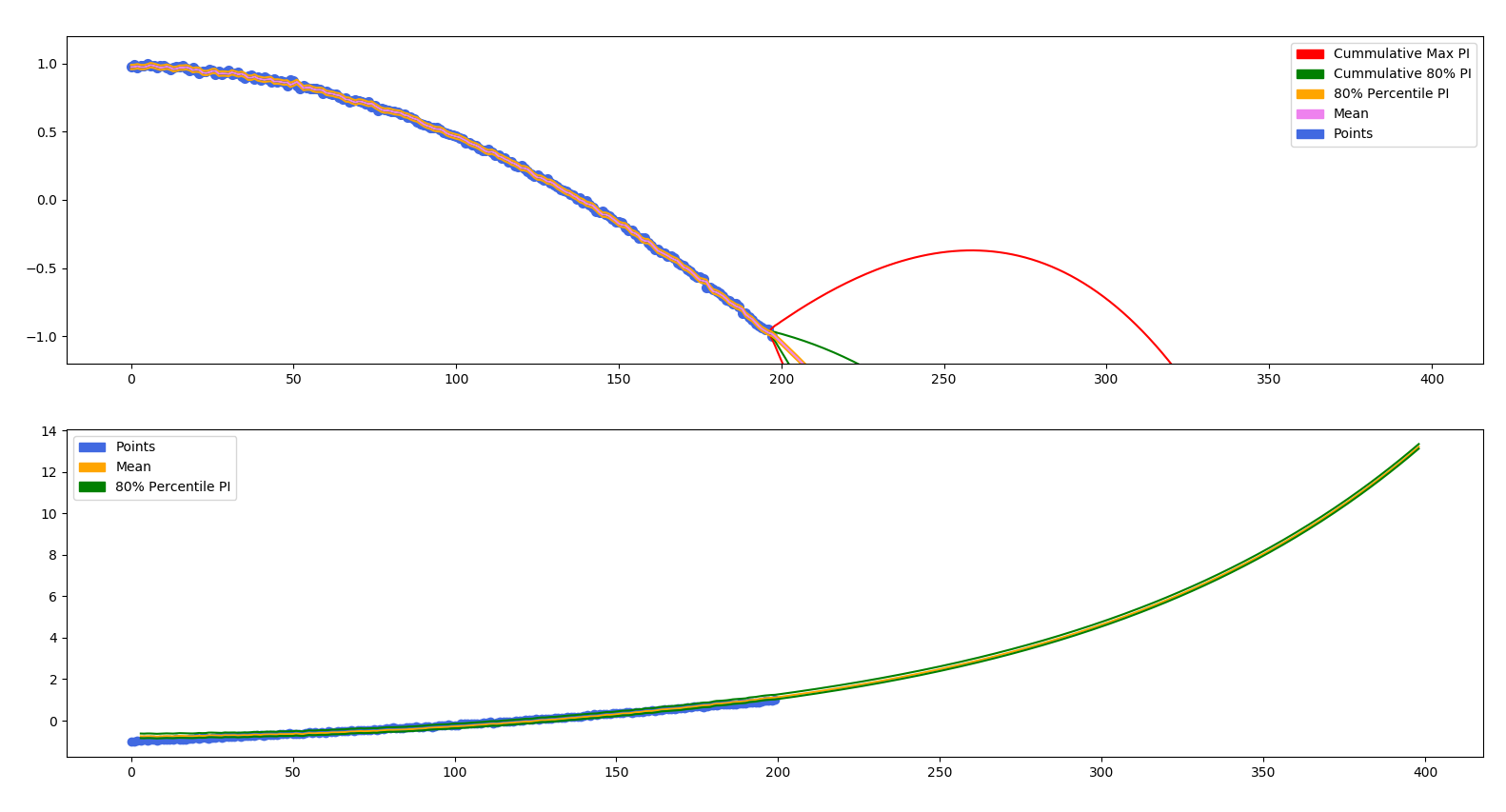


**Рисунок**... Плотность распределения ошибки (значения около 0) и вариация параметров модели линейной регрессии.

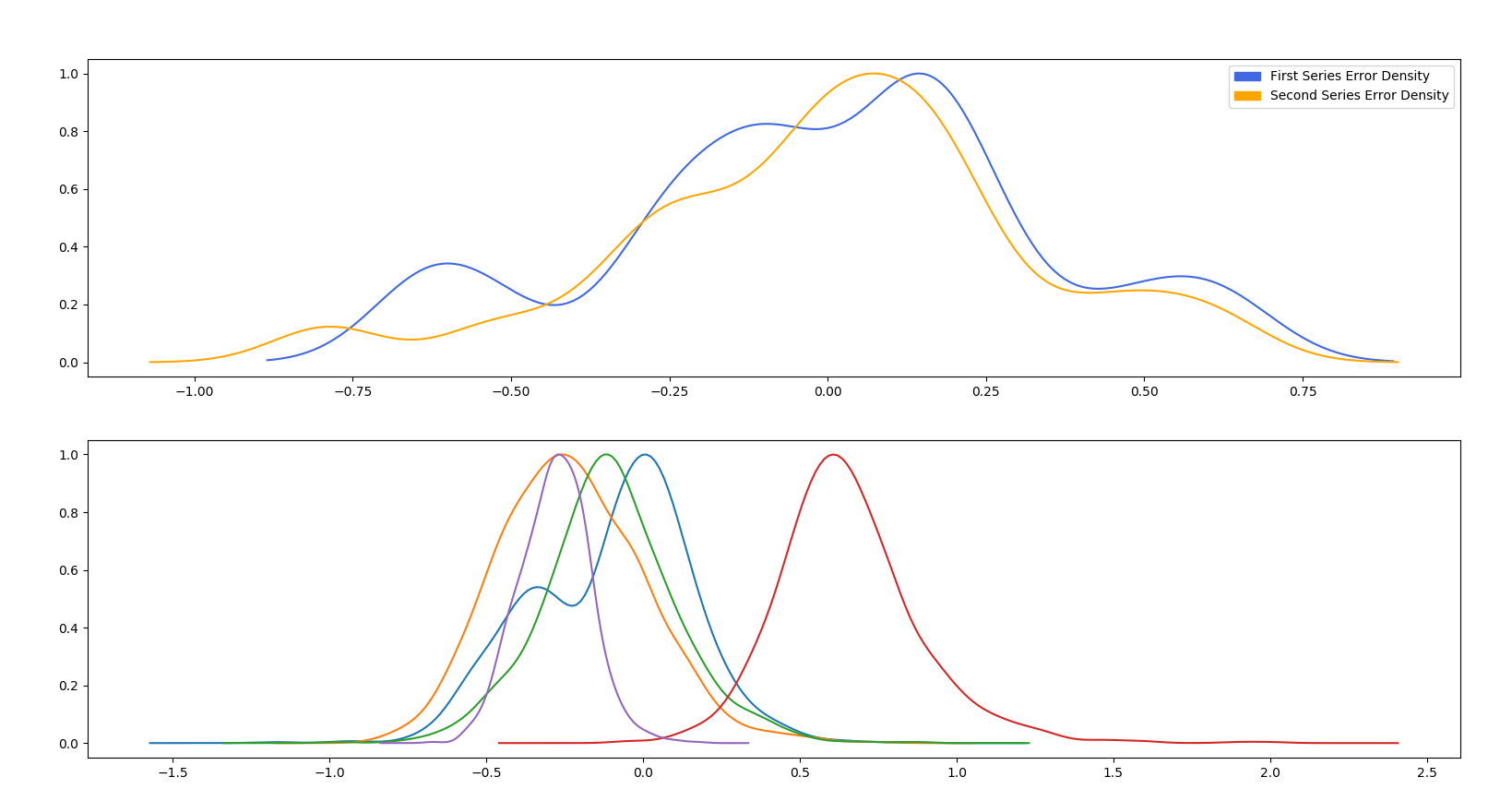


Мы видим, что результат очень хороший, алгоритм допускает маленькую ошибку. Предсказание было хорошим, как и восстановление косинуса по синусу.

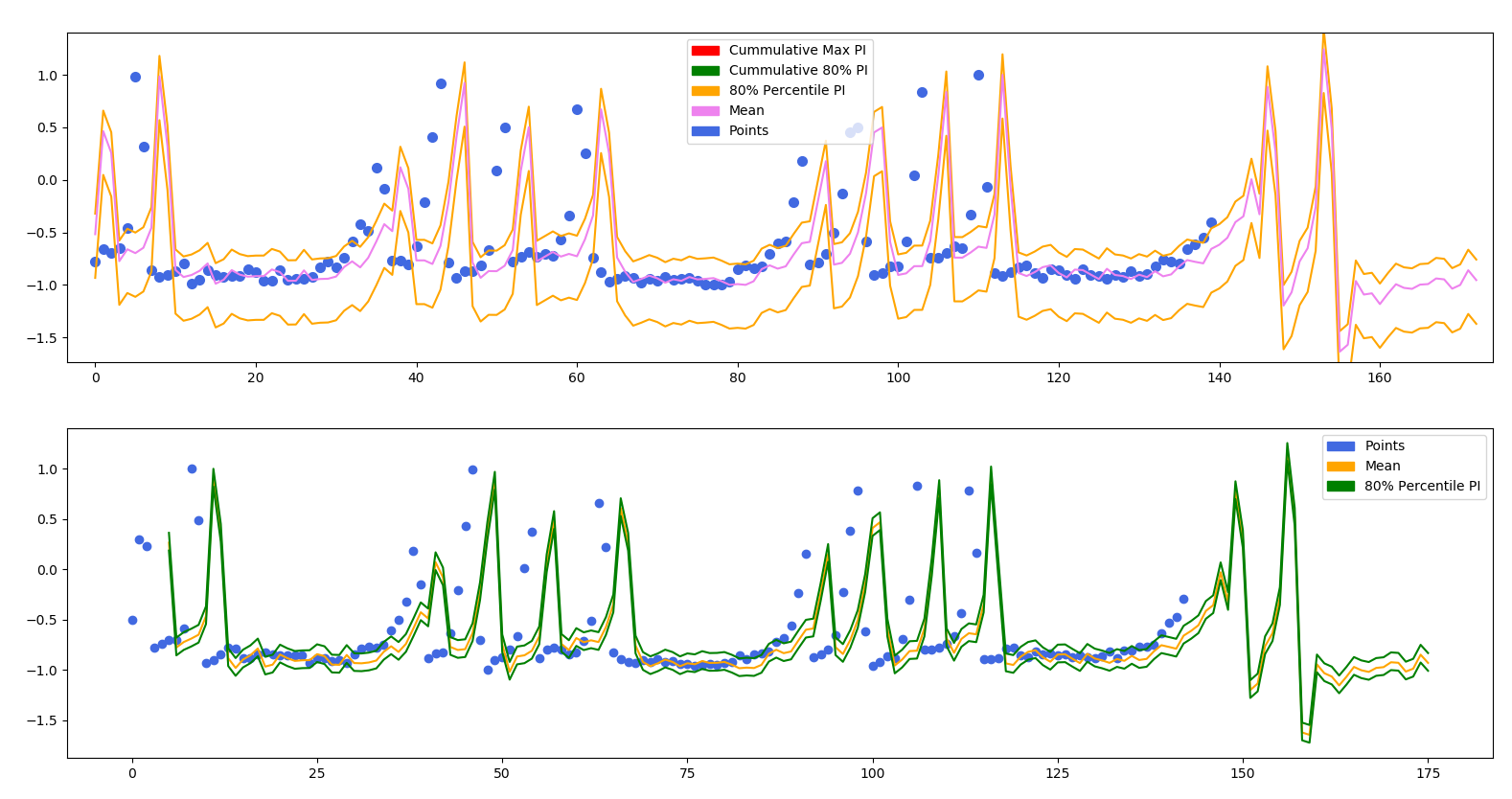
**Рисунок...** На рисунке изображен предсказанный первый ряд – квадратичный выпуклый вверх и восстановленный по нему - квадратичный выпуклый вниз. Метод: линейная регрессия.



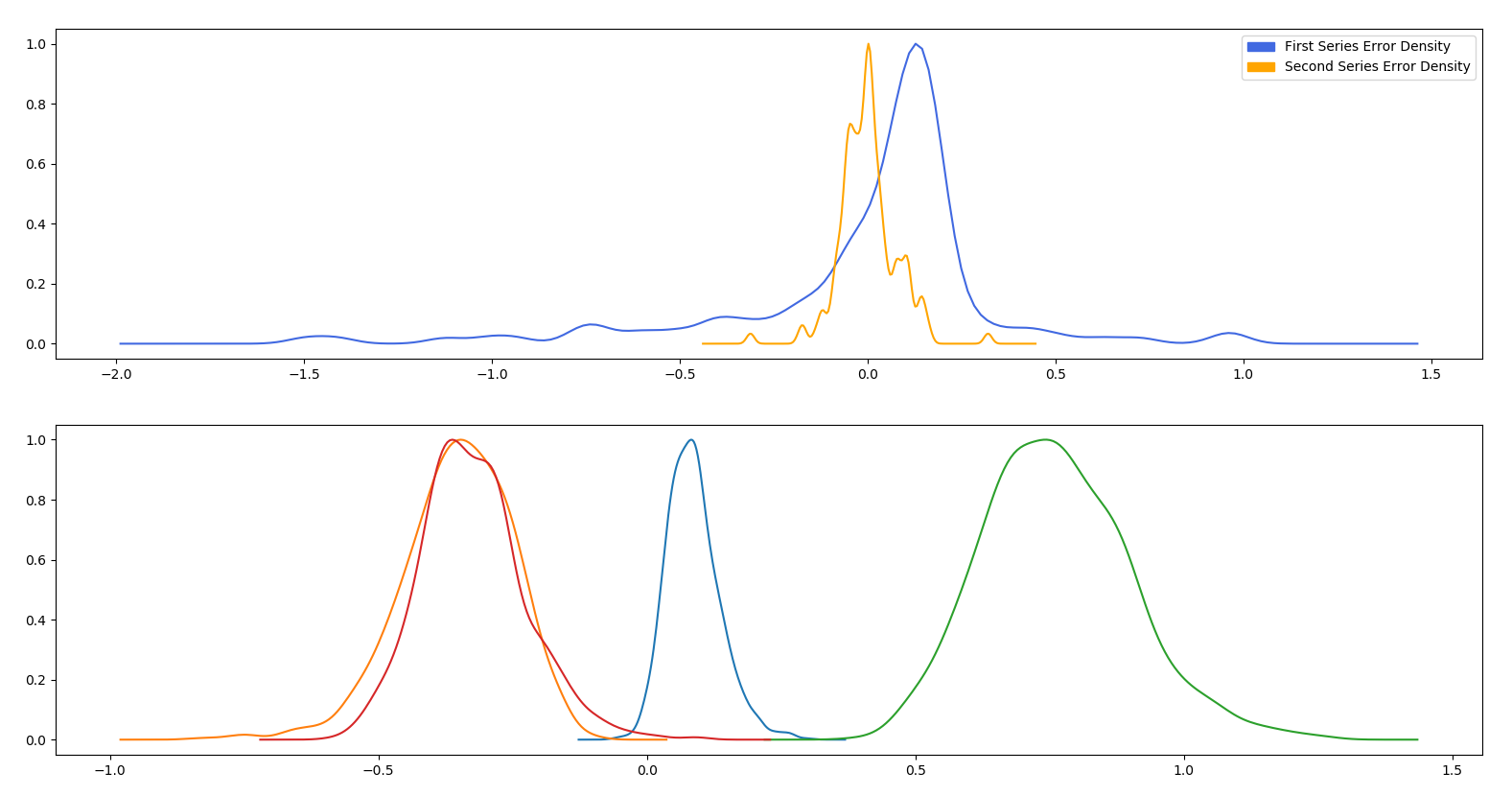
**Рисунок**... Плотность распределения ошибки (значения от -1 до 0.75) и вариация параметров модели линейной регрессии.



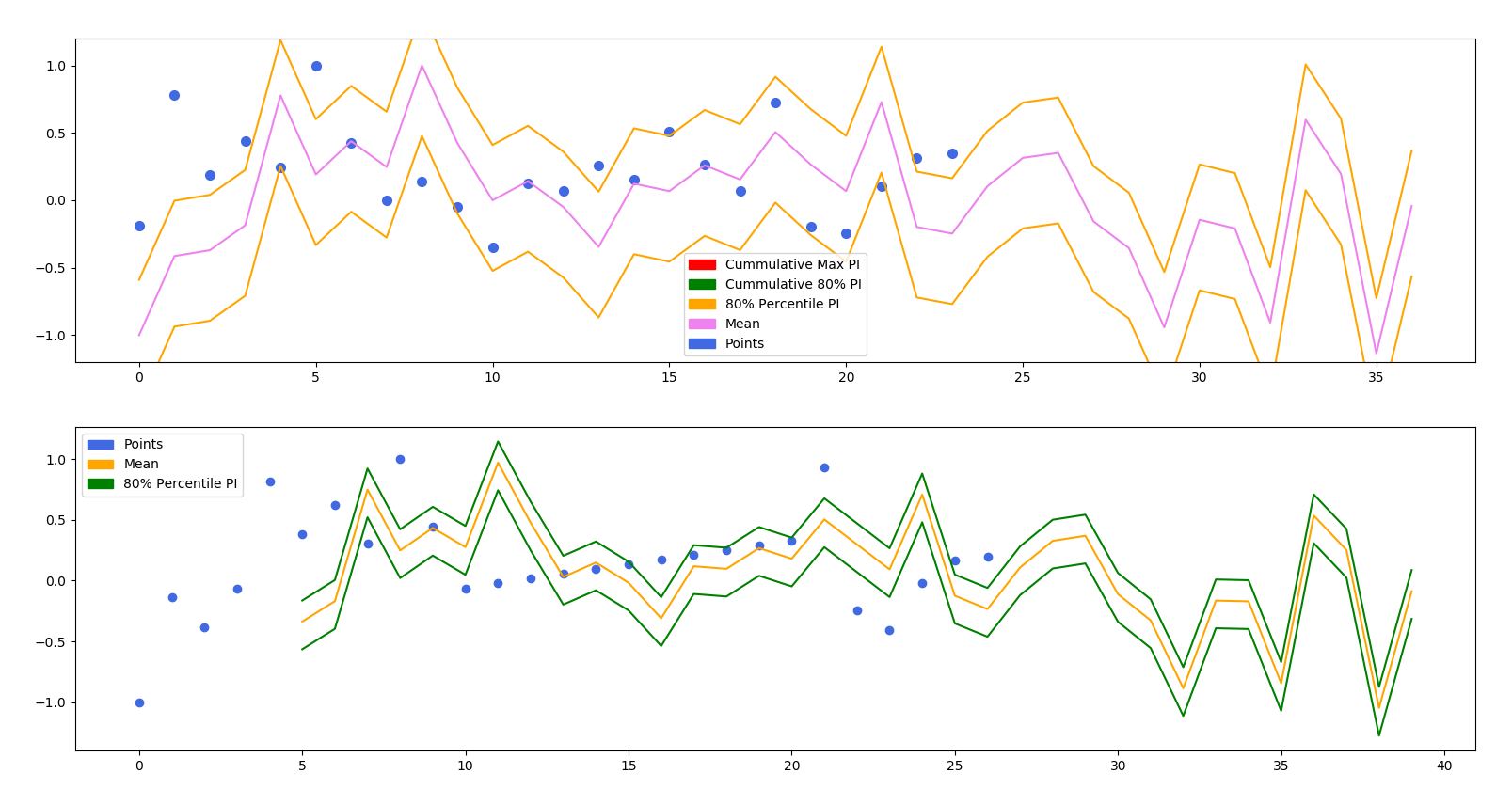
**Рисунок...** На рисунке изображен предсказанный первый ряд и восстановленный по нему – продаж магазина сети «Walmart». Метод предсказания первого ряда: нейронная сеть LSTM. Метод восстановления второго ряда: линейная регрессия.



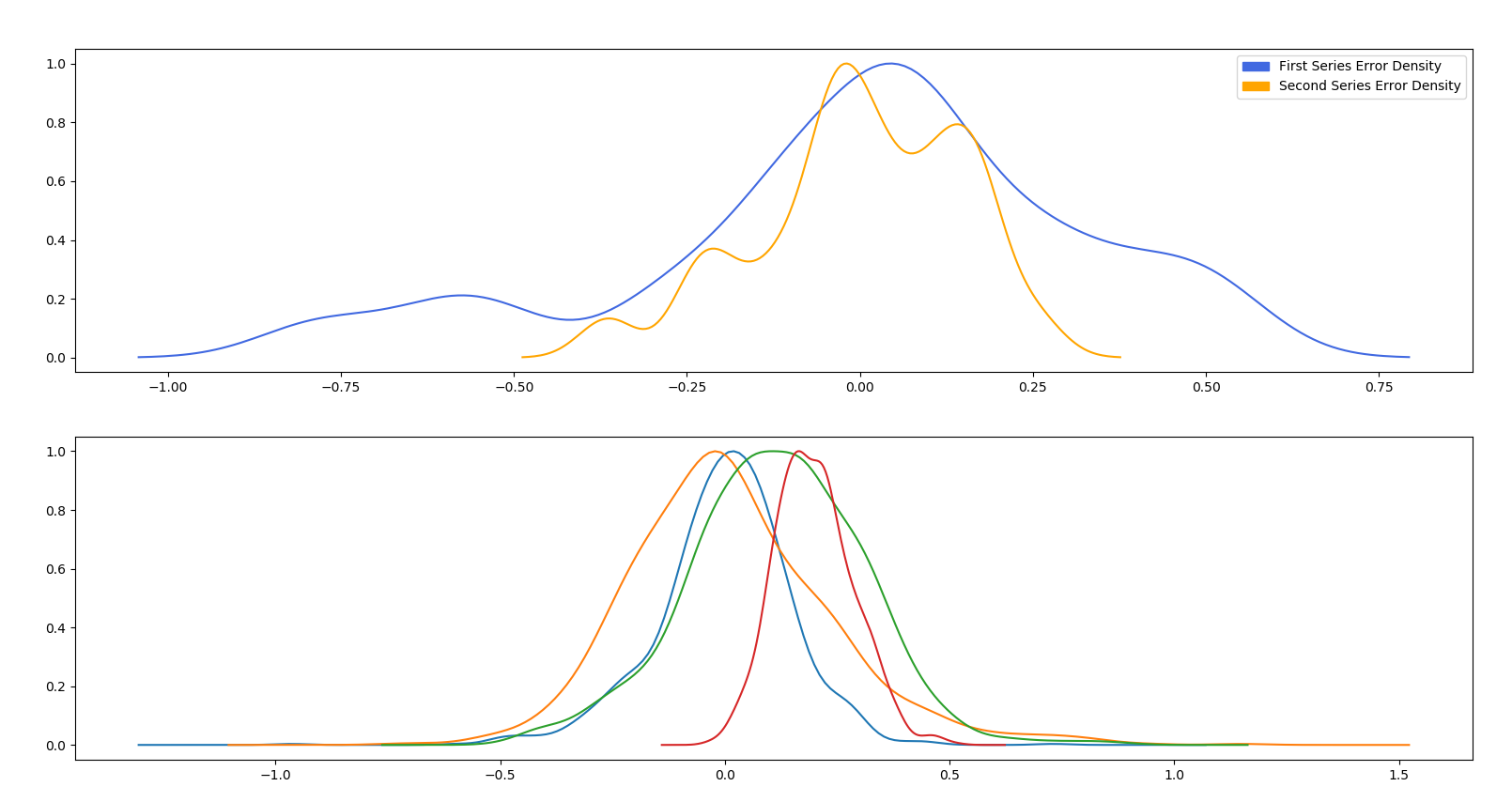
**Рисунок**... Плотность распределения ошибки (значения от -2 до 1.5) и вариация параметров модели линейной регрессии.



**Рисунок...** На рисунке изображен предсказанный первый ряд и восстановленный по нему – выборка «Производство основных сельскохозяйственных культур». Метод предсказания первого ряда: нейронная сеть LSTM. Метод восстановления второго ряда: линейная регрессия.



**Рисунок**... Плотность распределения ошибки (значения от -1 до 0.75) и вариация параметров модели линейной регрессии.



Источники

* LONG SHORT-TERM MEMORY. Authors: Sepp Hochreiter, Jurgen Schmidhuber  
  . Neural Computation 9(8):1735-1780, 1997, pages-32 link: <http://www.bioinf.jku.at/publications/older/2604.pdf>
* Recurrent Neural Networks and LSTM Author: Niklas Donges <https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5>
* Michael Nguyen <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>
* Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>
* <https://habr.com/company/wunderfund/blog/331310/>
* <https://en.wikipedia.org/wiki/Long_short-term_memory>
* ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. Authors: Diederik P. Kingma, Jimmy Lei Ba. Published as a conference paper at ICLR 2015  
  . pages - 15 link: <https://arxiv.org/pdf/1412.6980.pdf>
* <https://keras.io/optimizers/>
* <https://keras.io/layers/recurrent/>
* <https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data>
* References
* [1] C. Burrus. Least squared error design of FIR filters. Connexions Web site, 2012. http://cnx.org/content/m16892/1.3/.
* [2] C. Burrus. General solutions of simultaneous equations. Connexions Web site, 2013. http://cnx.org/content/m19561/1.5/.