Научная статья
Original article
УДК 002.304

СОВРЕМЕННЫЕ ПРАКТИКИ ПРИМЕНЕНИЯ МАШИННОГО ОБУЧЕНИЯ В ЗАДАЧЕ ПРОГНОЗА ПОГОДЫ

MODERN APPLICATION PRACTICES OF MACHINE LEARNING IN WEATHER FORECASTING

Быков Филипп Леонидович, научный сотрудник Гидрометцентра России Россия, г. Москва

Царалов Никита Дмитриевич, Магистрант 2 курс, Институт информационных технологий кафедра «Прикладной математики», Российский технологический университет, Россия, г. Москва

Bykov Philipp Leonidovich, researcher at the Hydrometeorological Center of Russia, Russia, Moscow

Tsaralov Nikita Dmitrievich, 2nd year master student, Institute of Information Technologies, Department of Applied Mathematics, Russian Technological University, Russia, Moscow

Аннотация: В статье рассматриваются современные практики применения различных архитектур нейронных сетей, таких как CNN (сверточные нейронные сети), TCN (темпоральные сверточные сети), RNN (рекуррентные нейронные сети) в задаче прогноза погоды. В статье приводятся описания подходов и практик для прогноза погоды на данных в нерегулярном наборе точек. Задача прогноза сводится к прогнозированию погодных временных рядов для статистических данных, image-to-image

translation и Seq2Seq (последовательность-к-последовательности) для картографических данных. Также была рассмотрена технология трансформеров для работы с изображениями, их классификации, сегментации и прогнозирования. Статья была создана для определения «строительных блоков» нейронных сетей прогноза погоды, и изучения практик их применения.

Annotation: The article is dedicated to modern application practices of neural networks architectures such as CNN (convolutional neural network), TCN (temporal convolutional network), RNN (recurrent neural network) for weather forecasting task. The article provides approaches and practices for weather forecast on data in irregular set of point. Forecasting task is reduced to weather time series forecasting for statistical data, image-to-image translation or sequence-to-sequence for cartographic data. The technology of transformers for working with images, their classification, segmentation and forecasting was also considered. The article was created to identify «building blocks» of neural networks for weather forecasting and to study the practices of their application.

Ключевые слова: прогноз погоды, погодные временные ряды, нейронные сети, машинное обучение, CNN (сверточные нейронные сети), TCN (темпоральные сверточные сети), RNN (рекуррентные нейронные сети), трансформеры

Keywords: weather forecasting, weather time series, neural networks, CNN (convolutional neural network), TCN (temporal convolutional network), RNN (recurrent neural network), Transformers

Введение

Целью данной статьи является анализ архитектур нейронных сетей применимо к задаче прогноза погоды, описания подходов и практик для прогноза погоды на данных в нерегулярном наборе точек.

Основными типами архитектур нейронных сетей, являются архитектуры, разработанные для решения задач следующих типов:

– задача прогнозирования временных рядов

– задача Seq2Seq (последовательность-к-последовательности)

– задача image-to-image

Рассмотрим применение этих архитектур в прогнозе погоды.

Полносвязные архитектуры

Для начала стоит отметить, что задачу прогноза погоды (weather forecast) можно трактовать как задачу прогнозирования временных рядов (time series forecasting).

Нейронные сети уже долгое время применялись для задачи прогноза погоды. Архитектура представляла собой определенное количество полносвязных слоев, в качестве функции активации использовался гиперболический тангенс или линейная функция, функцией ошибок являлась среднеквадратическая ошибка (mean squared error) (Kumar Abhishek, 2012). Подобная архитектура отражала тенденции сведения задачи прогноза погоды к задаче прогнозирования временных рядов.

Рекуррентные архитектуры

Часто используемым методом прогнозирования временных рядов с использованием нейронных сетей являются рекуррентные нейронные сети (RNN). Рекуррентные нейронные сети — вид нейронных сетей, где связи между элементами образуют упорядоченную (по времени) последовательность (рис. 1).

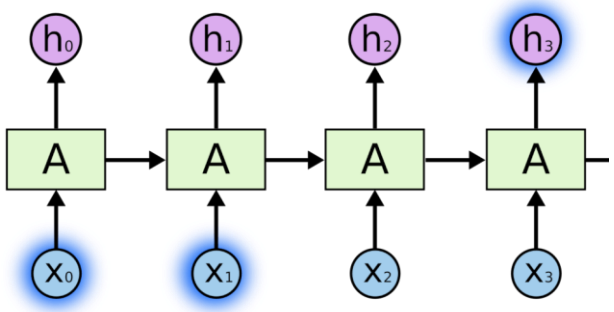


Рисунок 1. Общий вид рекуррентной нейронной сети.

Преимуществами использования рекуррентной нейронной сети в данной задаче является:

– Моделирование последовательности так, что каждое следующее состояние зависит от предыдущих;

– Рекуррентные нейронные сети комбинируются со сверточными слоями для работы с изображениями (видеорядом).

Недостатками же являются:

– Проблема затухания и взрыва градиента.

В процессе обратного распространения ошибки при прохождении через слои нейронной сети в элементах градиента могут накапливаться большие значения, что будет приводить к сильным изменениям весов. Это в свою очередь может сделать нестабильным алгоритм обучения нейронной сети. В таком случае элементы градиента могут переполнить тип данных, в котором они хранятся. Такое явление называется взрывающимся градиентом.

Существует аналогичная обратная проблема, когда в процессе обучения при обратном распространении ошибки через слои нейронной сети градиент становится все меньше. Это приводит к тому, что веса при обновлении изменяются на слишком малые значения, и обучение проходит неэффективно или останавливается, то есть алгоритм обучения не сходится. Это явление называется затухающим градиентом (Pykes, 2020).

– Невозможность прогнозирования слишком длинных последовательностей при использовании гиперболического тангенса или ReLU-функции ($f(x)=max(0,x)$) в качестве активации;

– Трудность обучения подобных сетей, которая заключается в вычислительной сложности и невозможности параллелизации вычислений в зависимых RNN-блоках.

Рассмотрим наиболее популярные решения последних лет. В последнее время можно выделить два лидирующих типа подобных сетей: LSTM (Long Sort-Term Memory) и GRU (Gated Recurrent Unit). Отличительной особенностью ячеек LSTM и GRU является различие в количестве внутренних операций. На иллюстрации изображены ячейки сетей LSTM и GRU соответственно (рис. 2):

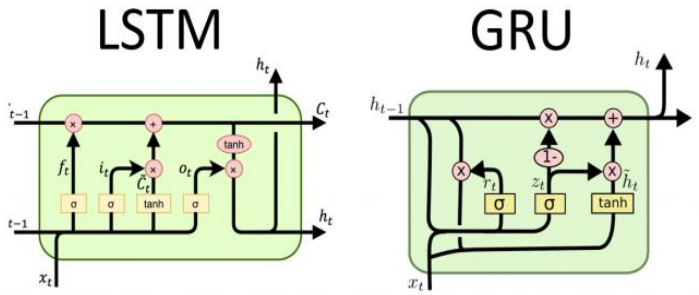


Рисунок. 2. Внутренняя архитектура блоков LSTM и GRU.

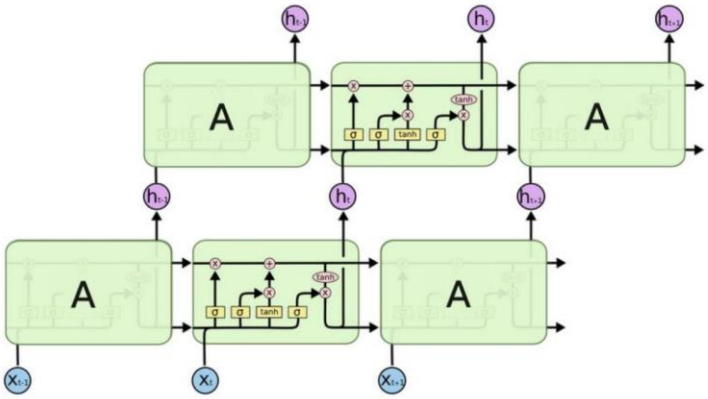


Рисунок 3. Архитектура нейронной сети на основе блоков LSTM.

Данные блоки объединяются по горизонтали и вертикали (рис. 3), формируя архитектуру нейронной сети. Как видно, для вычисления промежуточных состояний и выходов GRU блока требуется меньше операций, чем в LSTM, поэтому данная архитектура обучается значительно быстрее (Roberto Cahuantzi, 2021), хотя и обладает меньшим количеством обучаемых параметров и иногда хуже подходит для обнаружения сложных закономерностей. Применение блоков зависит от задачи. Возможно также их совместное использование в задачах Seq2Seq, где блок-кодировщик (Encoder) и блок-дешифровщик (Decoder) могут быть реализованы с использованием и LSTM и GRU.

Сверточные архитектуры

Еще одним интересным подходом к прогнозированию временных рядов являются временные сверточные сети (рис. 4) (Temporal Convolutional Networks). Механизм применения временных сверточных сетей подробно описан в статье «Temporal Convolutional Networks for Action Segmentation and

Detection» (Colin Lea, 2016). Базовая идея состояла в том, что на основе сверточных слоев с прореживанием (dilation) построена архитектура, состоящая из блока-кодировщика (Encoder) и блока-дешифратора (Decoder), на основе которых можно было выполнять задачу сегментации видео потока. Суммирование выходов блоков с различным прореживанием решает проблему затухания градиентов аналогично ResNet (Sun, 2015).

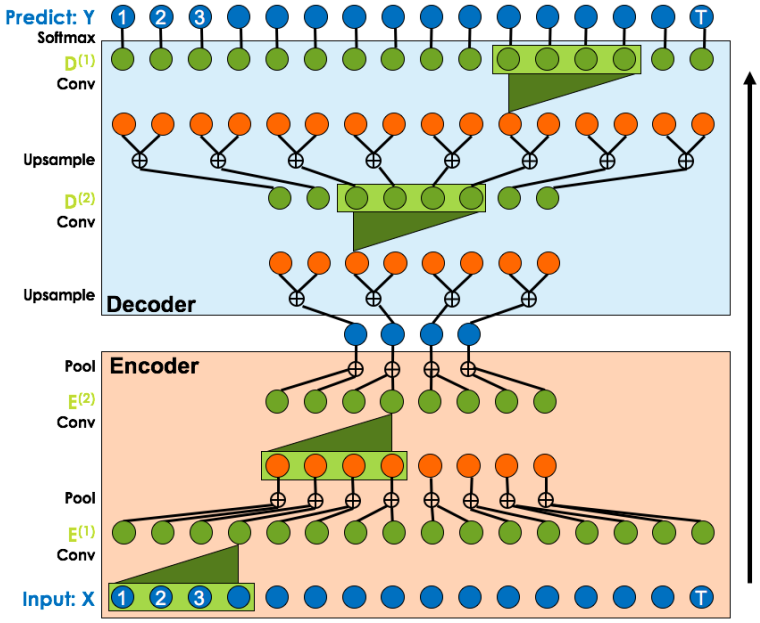


Рисунок 4. Архитектура Encoder-Decoder Temporal Convolutional сети

Сама концепция решения задачи image-to-image близка к задаче прогнозирования кадров в видеопотоке. Модель на основе сверточных слоев была использована в модели вероятностного прогнозирования в статье «Probabilistic Forecasting with Temporal Convolutional Neural Network» (Yitian Chen, 2019) где рассматривалось использование сверток с прореживанием (dilation) для увеличения контекста (окна восприятия) прогнозируемых значений.

$$s(t) = (x *_d w)(t) = \sum_{k=0}^{K-1} w(k)x(t - d \cdot k),$$

где s - вектор признаков в момент t разреженных сверток с ядром w, d - коэффициент прореживания, а K размер ядра.

Последовательное применение сверток с прореживанием (рис. 5) позволяет значительно увеличить окно восприятия при той же глубине сети и улавливать долгосрочные зависимости.

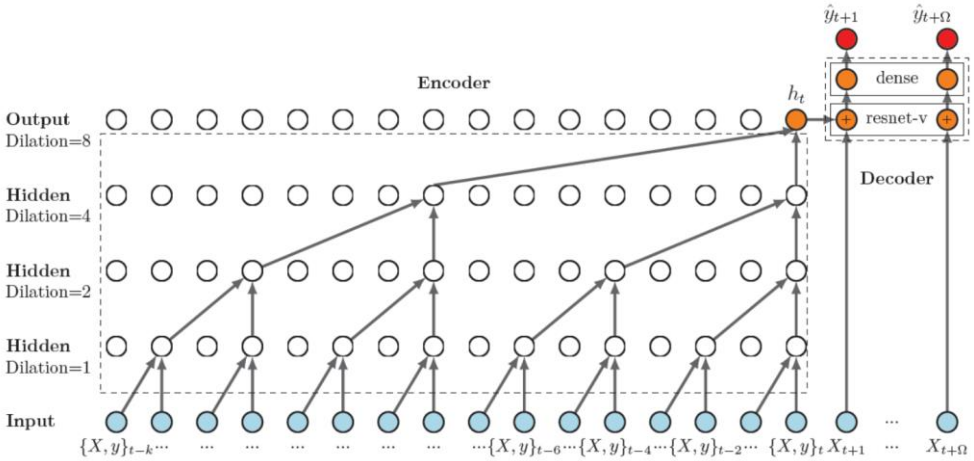


Рисунок 5. Архитектура глубокой временной сверточной сети (DeepTCN)

В статье «Probabilistic Forecasting with Temporal Convolutional Neural Network» (Yitian Chen, 2019) было предложено использовать подобные блоки в качестве блока кодировщика и дешифратора для построения прогнозов. Подобная архитектура показала результат лучший, чем достигнутый с помощью рекуррентных нейронных сетей (MASE = 0.926 для DeepTCN и 0.937 для RNN на задаче 24-часового прогнозирования на основе недельных данных потребления электричества и 0.438 против 0.449 для задачи прогнозирования трафика).

Механизм внимания

Механизмы внимания — это подход в машинном обучении, заключающийся в выделении части входных данных (регионов изображений, фрагментов текста) для более детальной обработки (рис. 6).

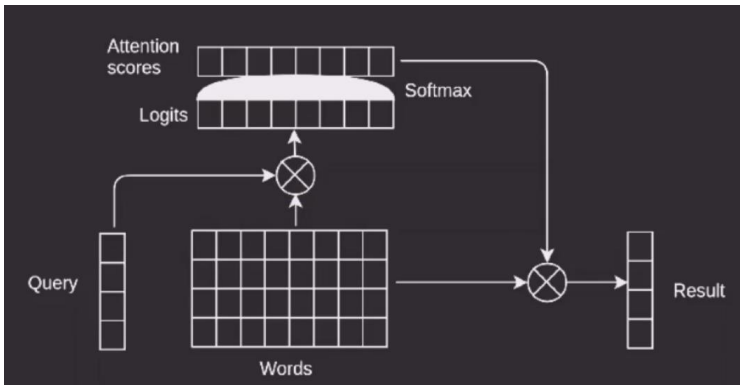


Рисунок 6. Механизм внимания по заданному запросу (query) в задаче NLP (обработка естественных языков)

Механизм внимания состоит в получении вектора значимости признаков и применении вектора значимости к входной матрице (рис.6).

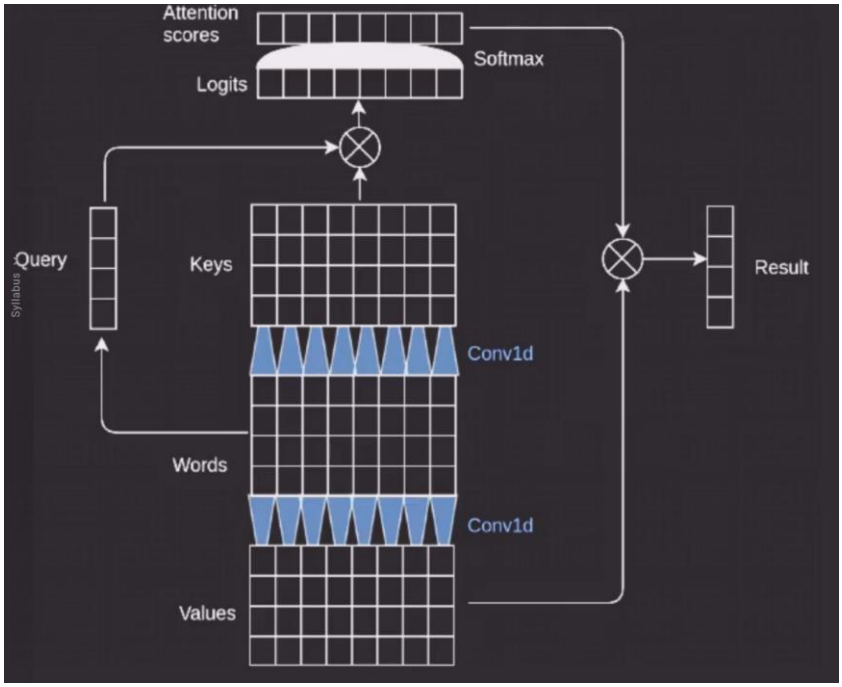


Рисунок 7. Механизм собственного внимания (self-attention)

Механизм собственного внимания (рис. 7) отличается от механизма внимания с ключем-запросом (query) тем, что в механизме собственного внимания этот ключ-запрос вычисляется из исходной матрицы с помощью слоев нейронной сети. Однако механизм собственного внимания обладает существенными недостатками:

- Потеря информации;
- Формирование весов только по одному аспекту.

Эти недостатки были устранены в механизме Multi-Head Attention, о котором будет рассказано далее.

Трансформеры

Возрастающий рост популярности задач обработки естественного языка (natural language processing) привел к распространению такой архитектуры нейронной сети как трансформер. Данная архитектура была создана преимущественно для задач моделирования языка, генерации текста и перевода. Подобные применения модели трансформера являются частными случаями задачи Seq2Seq. До открытия и популяризации трансформеров для задач Seq2Seq чаще всего применялись рекуррентные нейронные сети, о которых говорилось выше. Однако с возникновением трансформеров (в 2017 году), основанных на матричных умножениях и 1D свертках, которые позволяют значительно ускорить скорость обучения за счет параллельных вычислений и повысить точность за счёт увеличения количества обучаемых параметров, рекуррентные сети были вытеснены из таблиц лидеров в большинстве соревнований в сферах NLP и Seq2Seq задач.

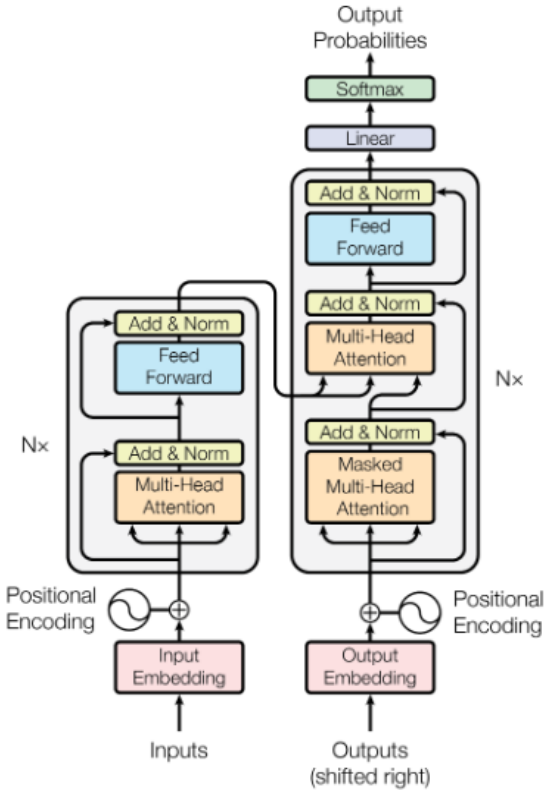


Рисунок 8. Архитектура базового трансформера

Архитектура базового трансформера (рис. 8), описанного в статье «Attention is all you need» (Ashish Vaswani, 2017), представляет собой Encoder (левая часть) и Decoder. В основе трансформера лежит концепция Multi-head Attention.

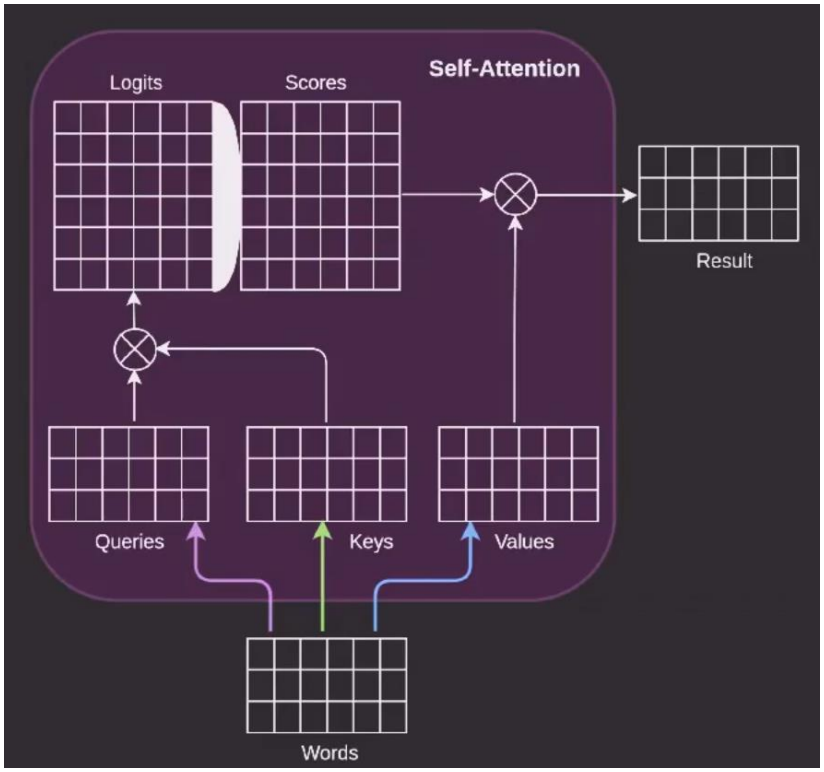


Рисунок 9. Внутреннее устройство механизма Multi-Head Attention и Scaled Dot-Product Attention

Из исходной последовательности вычисляются несколько наборов проекций запросов (Queries), ключей (Keys) и значений (Values) подобным способом:

$Q = Proj_Q * W$

$K = Proj_K * W$

$V = Proj_V * W$

Несколько наборов подобных проекций создаются для устранения недостатков механизма собственного внимания

Таким образом, именно для решения данных проблем и используется несколько наборов матриц. Каждый набор «ключ-значение-запрос» формируется для расчета взвешенного скалярного внимания (Scaled Dot-product Attention), рассчитываемого по следующей формуле (рис. 6):

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

После чего происходит конкатенация результатов выхода с предыдущего шага для каждого набора матриц ключей, значений и запросов полученных ранее.

В случае маскированного мультиголового внимания механизм Scaled Dot-product Attention отличается от обычного добавлением поэлементного умножения «маски» из нулей и единиц на результат перемножения ключей и запросов. Это используется в декодере для управления зависимостями. Например, «маска», представленная на рисунке 10, служит для учета слов только слева при решении задачи генерации текста. Исходная последовательность текста с закодированными токенами (слова, знаки препинания, пробельные символы и т. д.) разбивается на блоки (куски закодированного текста определённой длины), в которых маска зануляет ненужные токены.

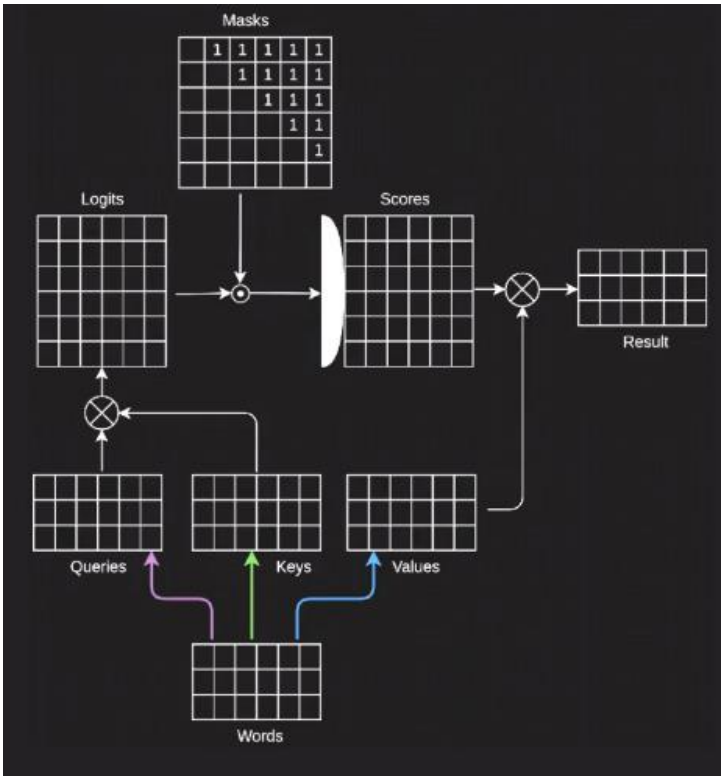


Рисунок 10. Механизм маскированного внимания (Masked Self-Attention)

Механизм многоголового внимания позволяет определить для различных элементов входной матрицы свой вес относительно других элементов (свое влияние) в различных аспектах. В случае задачи NLP на вход механизму внимания приходит матрица размерности количество токенов на количество признаков, а выходом слоя внимания является матрица весов токена в контексте других токенов.

Хотя механизм внимания и позволяет учитывать контекст (влияние других слов на конкретное слово), но в текущей редакции трансформер не способен никак учитывать взаимное расположение исходной последовательности. Для того, чтобы учитывать позицию элементов, в трансформер было добавлено позиционное кодирование, которое применяется непосредственно к входной последовательности элементов. То есть к входной матрице размера количество элементов на количество признаков прибавляется еще вектор позиционного кодирования размерности количество признаков (рисунок 6). Механизм формирования позиционного вектора может быть различным. Авторы изначальной архитектуры использовали кодирование синусоидальным сигналом по следующей формуле:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

В настоящее время архитектура трансформера пользуется популярностью в связи с тем, что из-за большого количества обучаемых параметров модели, трансформеры обладают возможностью улавливать сложные зависимости при параллельности процессов обучения гораздо лучше, чем у рекуррентных нейронных сетей. Поэтому трансформеры используются даже для задач прогнозирования временных рядов (Neo Wu, 2020). В статье «Deep Transformer Models for Time Series Forecasting:The Influenza Prevalence Case» используется архитектура классического трансформера, описанного выше, и сравнивается с решениями на основе рекуррентных моделей, авторегрессионных моделей и Seq2Seq моделей на основе GRU. Для применения трансформера к задаче прогнозирования временных рядов

необходимо только изменить механизм формирования позиционных кодов. Например, в статье «Transformers for Time-series Forecasting» предлагается представить время в виде трех составляющих: час, день и месяц (рис. 11), - и разложить каждый компонент по синусам и косинусам (Klingenbrunn, 2021).

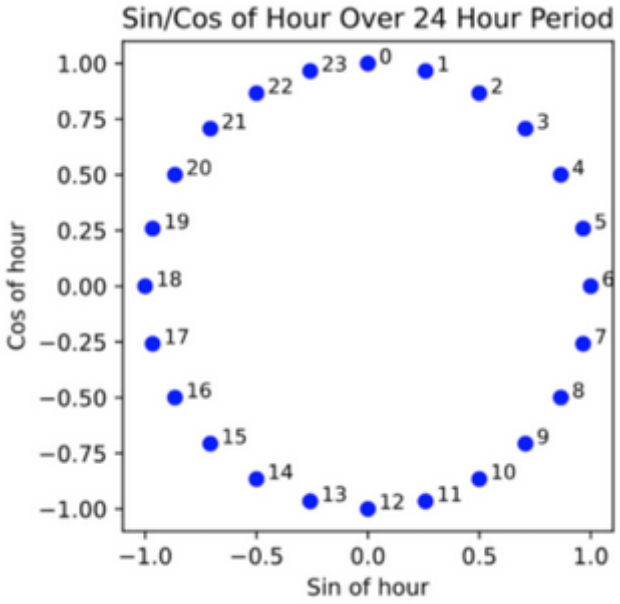


Рисунок 11. Пример позиционного кодирования по часам.

После внедрения механизма позиционного кодирования можно использовать обычный трансформер для задач генерации текста. Класс, описывающий датасет, должен возвращать последовательности с времени [Xt-k1 … Xt-1], [Xt … Xt+k2], где k1 - количество предшествующих измерений, а k2 - количество прогнозируемых значений.

Также хотелось бы акцентировать внимание на развитии использования трансформеров в задачах компьютерного зрения и их превосходство в задачах классификации и сегментации (René Ranftl, 2021). Механизм работы Vision трансформера для задач классификации (Alexey Dosovitskiy, 2020) связан с разбиением исходного изображения на части, получением их линейных представлений (embeddings) и добавлением кодов позиций каждой части изображения. Затем полученные данные подаются в классический блок-кодировщик трансформера и по его прохождении в классификатор (рис. 12).

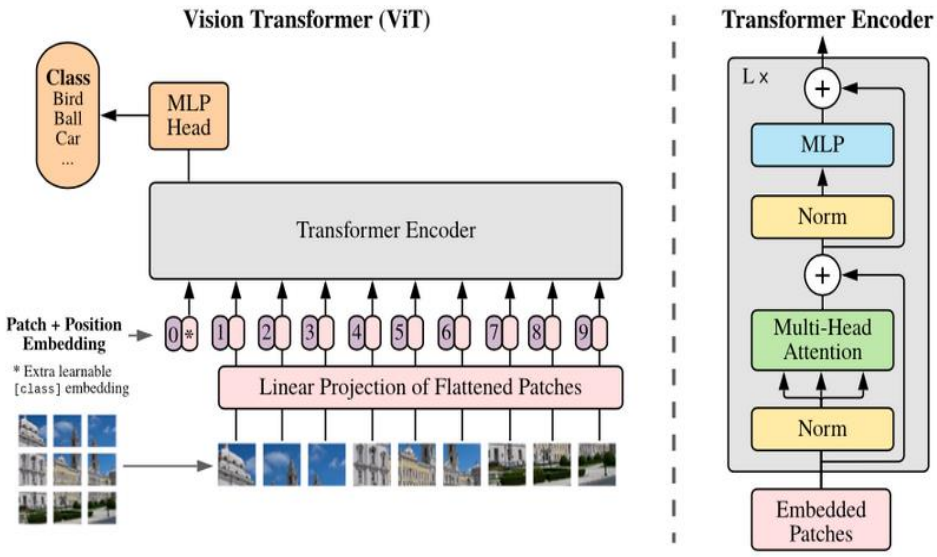


Рисунок 12. Архитектура Vision Transformer для задачи классификации

Статья «Vision Transformers for Dense Prediction» является развитием идеи сегментации изображений. Подготовка изображений аналогична описанной выше, то есть изображения разбиваются и приводятся к плоским представлениям (векторам, полученным с помощью линейного слоя или путем применения ResNet-50), к которым добавляется позиционное кодирование каждой части изображения. Полученные токены пропускаются через ряд блоков-трансформеров. Результаты отработки блоков снова собираются в изображение различных размеров (reassemble блоки). Далее Fusion-блоки постепенно объединяют и повышают дискретизацию представлений для генерации детального прогноза (рис. 13).

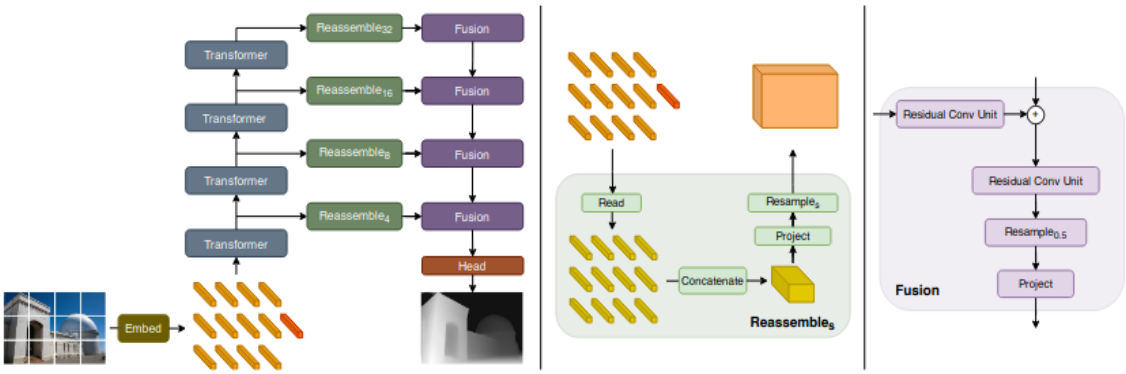


Рисунок 13. Архитектура Vision Transformer для задачи определения глубины

Хотя в настоящий момент подобная архитектура не применялась для задач прогнозирования погоды, несложно представить ее применение в

качестве image-to-image решения для прогноза следующих кадров в видео потоке (движения воздушных масс).

Недостатками трансформеров является…

Нейронные сети в задачах прогноза погоды

Далее обратимся к реальным современным практикам использования описанных выше архитектур для решения задач прогнозирования погоды нейронными сетями, учитывающими данные географии точек измерения и орографии.

Интересной практикой применения глубоких нейронных сетей в задаче прогноза погоды стала статья «Machine Learning for Precipitation Nowcasting from Radar Images» (Shreya Agrawal, 2019) опубликованная в Google AI Blog, посвященная прогнозированию выпадения осадков, базируясь на изображениях со спутников. Подобное решение является применением глубокого обучения для решения задачи image-to-image translation в высоком разрешении (1 км на 1 км), использующим в качестве основы архитектуру сверточной нейронной сети U-Net (сверточную нейронную сеть, разработанную для решении задачи сегментации (Olaf Ronneberger, 2015)). Сеть содержит сверточную (слева) и разверточную части (справа), поэтому архитектура похожа на букву U, что и отражено в названии (рис. 14). На каждом шаге мы удваиваем количество каналов признаков.

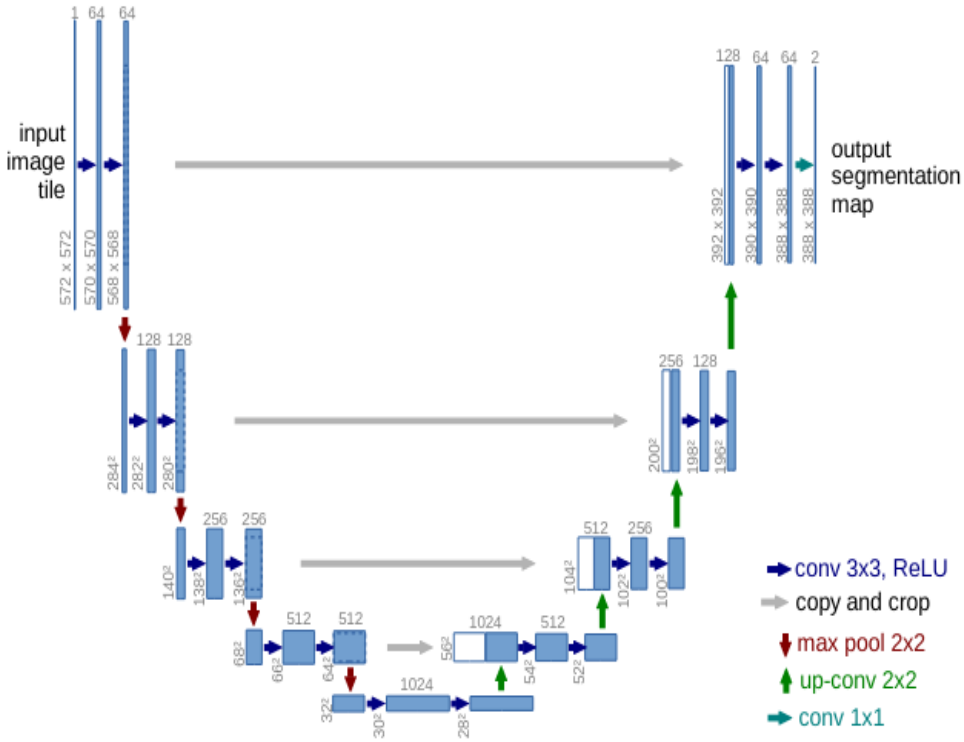


Рисунок 14. Архитектура U-Net.

В архитектуру сети для прогнозирования осадков изменения размера изображения, так исходное изображение представлено размером 256 на 256 км. Также стоит упомянуть препроцессинг и исходные данные. Модель обучалась на данных с 2017 по 2019 годы с мультирадарной мультисенсорной системы (MRMS), дающей показания с разрешением 1 км на 1 км. Данные с каждой измерительной точки представляют собой карты отражательной способности (volumetric reflectivity map). После получения данные из аналоговых преобразуются в дискретные по следующим интервалам [0,0.1), [0.1,1.0), [1.0,2.5) и [2.5,∞) соответственно. Затем площадь покрытия делится на участки 256 на 256 км, которые подаются на вход сверточной нейронной сети U-Net. Результат работы данной архитектуры при сегментации оказался лучше, чем классические подходы, такие как численное моделирование и решение задачи оптического потока (рис. 15).

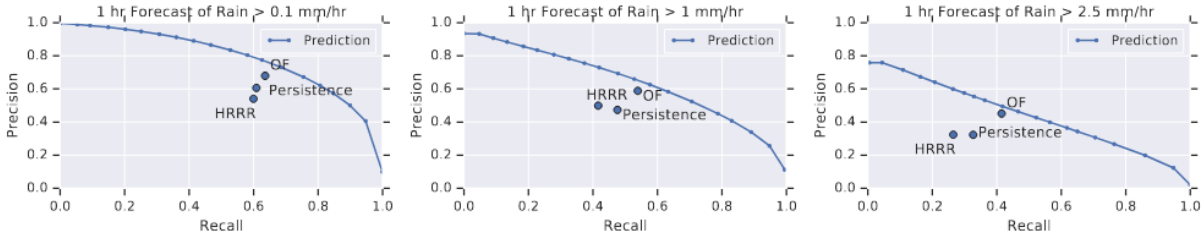


Рисунок 15. График точность-полнота для построенной модели (HRRR - численное моделирование, OF - оптический поток).

MetNet

Больший интерес представляет статья «MetNet: A Natural Weather Model for Precipitation Forecasting» (Casper Kaae Sønderby, 2020). MetNet является объединением рекуррентных нейронных сетей LSTM (Temporal Encoder), механизмов собственного внимания (Spatial Aggregator) и сверточных нейронных сетей понижения размерности (Spatial Downsampler) (рис. 16). Целью разработки являлось создание глубокой нейронной сети для прогнозирования осадков на 8 часов с разрешением 1 км на 1 км.

Далее поясним принципы работы этих блоков

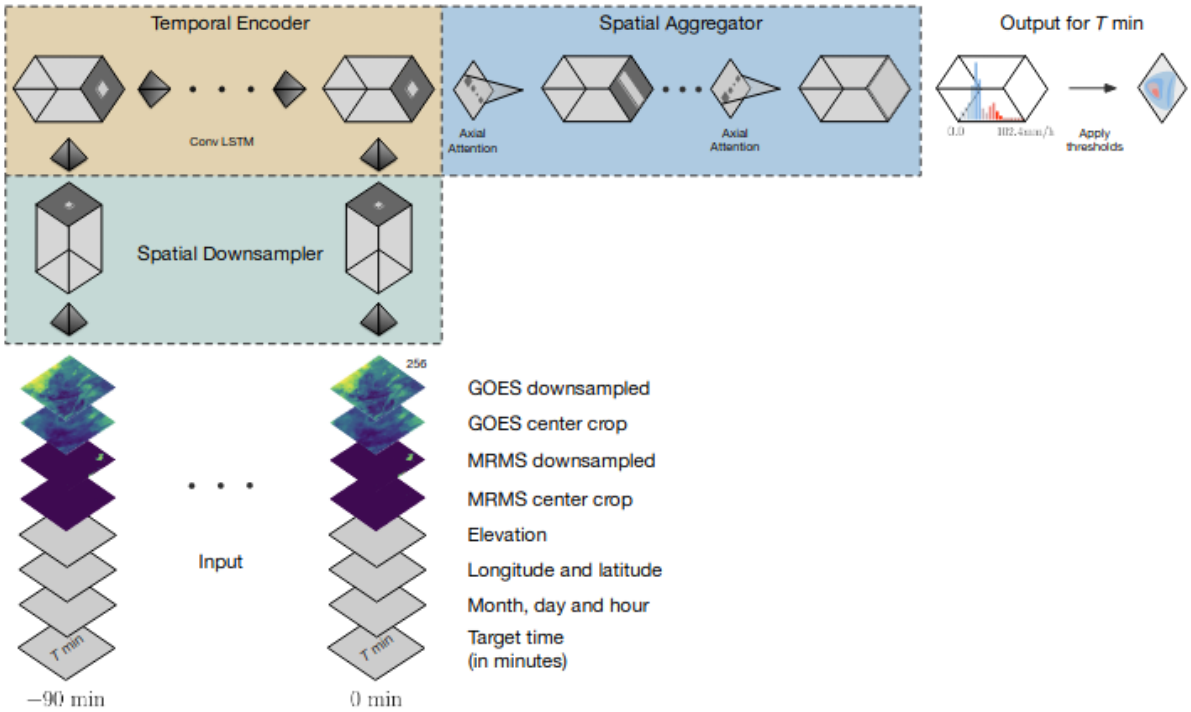


Рисунок 16. Архитектура MetNet

Главным звеном сети MetNet является ConvLSTM рекуррентная нейронная сеть для работы с изображениями. Примерами задач ConvNet могут служить обработка и прогнозирование:

- Следующих кадров видео;
- Изменения на снимках из космоса (используется MetNet);
- Действий на камерах безопасности (видеонаблюдения?).

Архитектура блока ConvLSTM отличается от LSTM тем, что внутренние матричные произведения заменяются на операции свертки (Xingjian Shi, 2015). Так, что внутренние операции над тензорами можно описать следующим образом:

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o) \\
\mathcal{H}_t &= o_t \circ \tanh(\mathcal{C}_t)
\end{aligned}
$$

Изначально архитектура была предложена для решения задачи прогноза, где блоки ConvLSTM используются в качестве блока-кодировщика (Encoder) и блока-дешифратора (Decoder), образуя структуру для решения image-to-image translation задач, которые распространены в задачах гидрометеорологии (рис. 17).

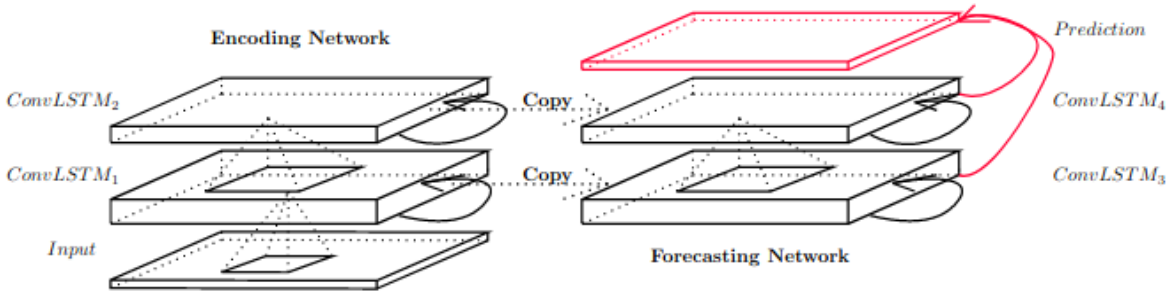


Рисунок 17. Архитектура ConvLSTM Encoder-Decoder для краткосрочного (nowcasting) прогнозирования погоды

Помимо подхода ConvLSTM ранее использовался так называемый Convolutional-LSTM или FC-LSTM, хотя он и выполняет ту же функцию, что и ConvLSTM, подход к реализации совершенно отличается. Исходное

изображение каждого кадра подается на вход сверточной нейронной сети (CNN), целью которой является извлечение вектора признаков, который в свою очередь поступает на вход в ячейку LSTM. В результате FC-LSTM работает с векторами признаков, а не с тензорами изображений, и выходом нейронной сети является набор векторов признаков (Qingqing Wang, 2020). Архитектура ConvLSTM была представлена для решения проблемы избыточности преобразований в FC-LSTM, так как она позволяет отказаться от промежуточных извлечений векторов признаков с помощью сверточных сетей и работать с изображениями непосредственно (рис. 18).

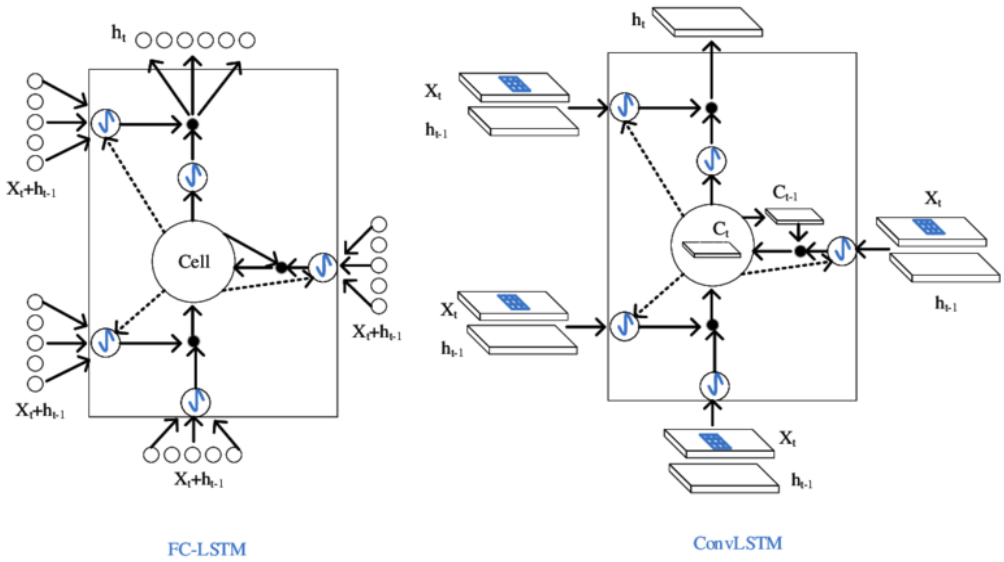


Рисунок 18. Сравнение внутренней архитектуры ячейки FC-LSTM и ConvLSTM.

Temporal Encoder сети MetNet имеет архитектуру ConvLSTM. Рассмотрим оставшиеся части MetNet. Spatial Downsampler входной тензор размерности [t,w,h,c], где t - время, h - высота, w - ширина, а c - количество каналов. Ширина и высота заданы площадью получаемых измерений 1024 км на 1024 км, время - срезы по 15 минут в интервале 90 минут, предшествующего Tx, где Tx - момент времени, в который модель делает прогноз. Входные характеристики (каналы) представлены MRMS радарными данными, данными со спутника GOES-16 и параметрами широты, долготы, времени и орографии

(высоты поверхности Земли). Затем тензор поступает на вход сверточной нейронной сети, целью которой является снижение размерности входных данных. Архитектура сверточной нейронной сети понижения размерности представлена следующими слоями: сверточный слой с ядром размера 3 и количеством каналов 160, слой max-pooling с ядром 2 и шагом 2, три сверточных слоя с ядром 3 и 256 выходными каналами и последний max-pooling с ядром 2 и шагом 2. По прохождению данной нейронной сети на выходе имеем тензор размерности 64 на 64 на 256, который передается на следующий блок архитектуры — ConvLSTM.

В данной редакции используется ConvLSTM с ядром 3 и количеством каналов 384. Выходом ConvLSTM является тензор 64 на 64 на 384. Последним блоком является блок, состоящий из нескольких восьми осевых блоков собственного внимания (self-attention). Особенностью механизма внимания применимого к изображениям является квадратичная зависимость количества параметров от разрешения входной последовательности. Из-за данной особенности сети, использующие классические механизмы внимания, требуют огромных вычислительных ресурсов. Для решения данной проблемы авторы архитектуры MetNet используют механизм осевого внимания (Jonathan Ho, 2019).

Механизм осевого внимания базируется на принципе работы свертки. В классическом случае при применении слоя внимания к изображению мы получаем влияние каждого пикселя изображения на конкретный пиксель, что является аналогом полносвязного слоя. Такой подход требует значительных вычислительных мощностей. Автор статьи предлагает использование осевого внимания, где в отличие от предыдущего случая, после применения слоя внимания мы получаем влияние ряда и колонки из пикселей на пиксель их пересечения (рис.19).

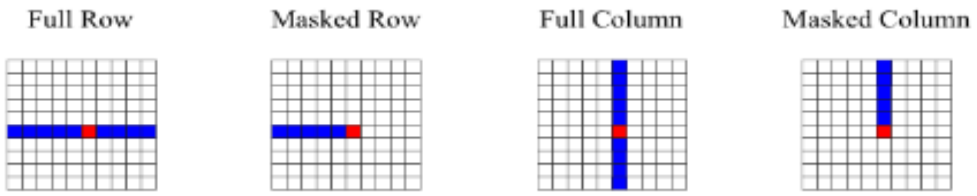


Рисунок 19. Типы слоев осевого внимания.

Пиксели голубого цвета обозначают рецептивное поле (receptive field) пикселя красного цвета. Последовательное применение слоев осевого внимания позволяет установить влияние пикселей изображения друг на друга, функционируя подобно обычным сверточным слоям. Как описано в статье MetNet, архитектура использует 8 осевое внимание, то есть 4 блока осевого внимания типа Full Column и 4 слоя типа Full Row. Подобный механизм внимания способен учитывать такие параметры как орография местности и движение воздушных масс.

В результате прохождения блока Spatial Aggregator получается 512-позиционное категориальное распределение. Каждая позиция из 512 отвечает за интервал 0.2 мм/час прогнозируемой нормы осадков (от 0 мм/час до 102.4 мм/час). Все прогнозы свыше 102.4 мм/час помещаются в последнюю категорию. После чего происходит классификация результатов с применением пороговых значений, результаты которого наносятся на карту размером 64 на 64 км. Подобный метод прогнозирования можно назвать одним из самых продвинутых на сегодняшний день, поскольку он является комбинацией улучшенных механизмов внимания для работы с изображениями, рекуррентных нейронных сетей, работающих с объемными данными и сверточных нейронных сетей понижения и повышения размерности (рис. 20).

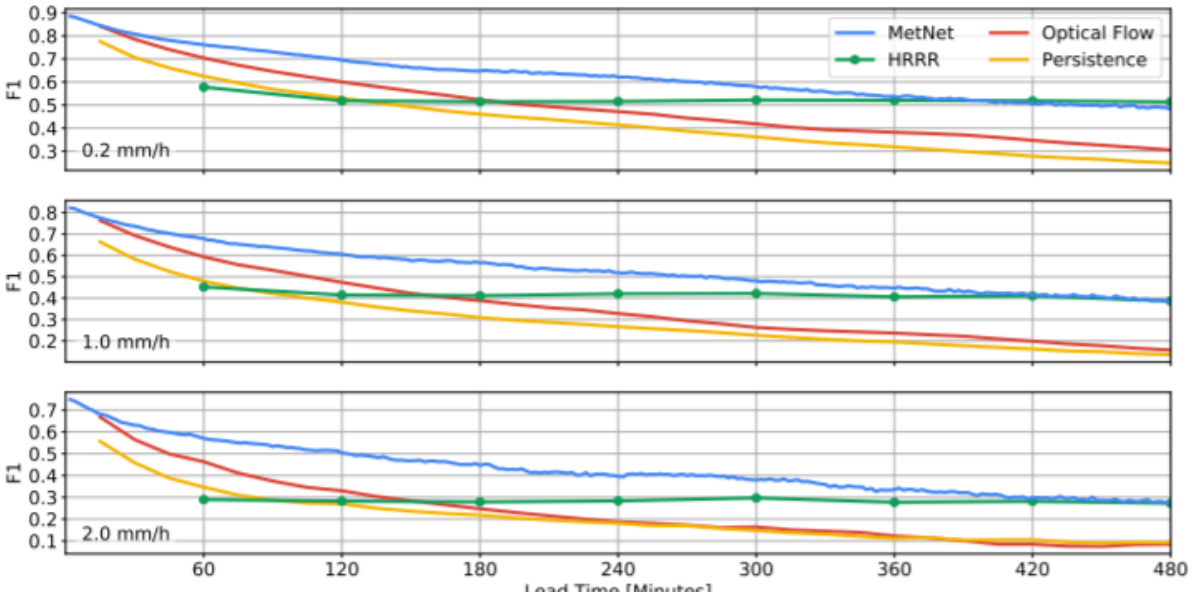


Рисунок 20. F1 метрика для MetNet, HRRR, Optical Flow от 2x до 480 минут

Заключение

В данной статье были рассмотрены основные современные архитектуры, на которых базируются нейросетевые методы прогноза погоды:

– Рекуррентные нейронные сети (RNN);

– Временные сверточные сети (TCN);

– Сверточные сети (CNN);

– Трансформеры.

Для каждой из архитектур были предложены их применения для решения задачи прогнозирования временных рядов. Далее было рассмотрено применение некоторых из данных архитектур для решения задачи image-to-image translation, которая впоследствии оказалась случаем задачи прогноза погоды (примером служит MetNet).

Далее были рассмотрены механизмы для улучшения результатов и оптимизации обучения:

– Механизмы понижения размерности на основе сверточных нейронных сетей;

– Механизмы собственного внимания для учета глобального контекста для каждой точки пространства;

– Механизм осевого внимания - композицию слоев внимания по вертикали и горизонтали и подобным образом оптимизирует вычисления.

Также была рассмотрена технология трансформеров для работы с изображениями, их классификации, сегментации и прогнозирования.

Статья была создана для определения «строительных блоков» нейронных сетей прогноза погоды, и изучения практик применения.

Литература

1. Alexey Dosovitskiy Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [Online]. - Oct 22, 2020. - Sep 7, 2021. - https://arxiv.org/abs/2010.11929.
2. Ashish Vaswani Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin Attention Is All You Need [Online]. - Jun 12, 2017. - Jun 20, 2021. - https://arxiv.org/abs/1706.03762.
3. Casper Kaae Sønderby Lasse Espeholt, Jonathan Heek, Mostafa Dehghani, Avital Oliver, Tim Salimans, Shreya Agrawal, Jason Hickey, Nal Kalchbrenner MetNet: A Neural Weather Model for Precipitation Forecasting [Online]. - Mar 24, 2020. - Sep 20, 2021. - https://arxiv.org/abs/2003.12140.
4. Colin Lea Michael D. Flynn, Rene Vidal, Austin Reiter, Gregory D. Hager Temporal Convolutional Networks for Action Segmentation and Detection [Online]. - Nov 16, 2016. - Aug 18, 2021. - https://arxiv.org/abs/1611.05267.
5. Jonathan Ho Nal Kalchbrenner, Dirk Weissenborn, Tim Salimans Axial Attention in Multidimensional Transformers [Online]. - Dec 20, 2019. - Oct 20, 2021. - https://arxiv.org/abs/1912.12180.
6. Klingenbrunn Natasha Transformers for Time-series Forecasting [Online]. - Feb 19, 2021. - Sep 7, 2021. - https://medium.com/mlearning-ai/transformer-implementation-for-time-series-forecasting-a9db2db5c820.

7. Kumar Abhishek M.P. Singh, Saswata Ghosh, Abhishek Anand Weather Forecasting Model using Artificial Neural Network [Journal] // Procedia Technology. - 2012. - Vol. 4. - pp. 311-318. - 2212-0173.
8. Neo Wu Bradley Green, Xue Ben, Shawn O'Banion Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case [Online]. - Jan 23, 2020. - Sep 3, 2021. - https://arxiv.org/abs/2001.08317.
9. Olaf Ronneberger Philipp Fischer, Thomas Brox U-Net: Convolutional Networks for Biomedical Image Segmentation [Online]. - May 18, 2015. - Sep 10, 2021. - https://arxiv.org/abs/1505.04597.
10. Pykes Kurtis The Vanishing/Exploding Gradient Problem in Deep Neural Networks [Online]. - 5 17, 2020. - Август 15, 2021. - https://towardsdatascience.com/the-vanishing-exploding-gradient-problem-in-deep-neural-networks-191358470c11.
11. Qingqing Wang Ye Huang, Wenjing Jia, Xiangjian He, Michael Blumenstein, Shujing Lyu, Yue Lu FACLSTM: ConvLSTM with Focused Attention for Scene Text Recognition [Online]. - Jan 5, 2020. - Oct 3, 2021. - https://www.researchgate.net/publication/332590561_FACLSTM_ConvLSTM_with_Focused_Attention_for_Scene_Text_Recognition/link/603632e4299bf1cc26e84d12/download.
12. René Ranftl Alexey Bochkovskiy, Vladlen Koltun Vision Transformers for Dense Prediction [Online]. - Mar 24, 2021. - Sep 3, 2021. - https://arxiv.org/abs/2103.13413.
13. Roberto Cahuantzi Xinye Chen, Stefan Güttel A comparison of LSTM and GRU networks for learning symbolic sequences [Online]. - Jul 5, 2021. - Aug 17, 2021. - https://arxiv.org/abs/2107.02248.
14. Shreya Agrawal Luke Barrington, Carla Bromberg, John Burge, Cenk Gazen, Jason Hickey Machine Learning for Precipitation Nowcasting from Radar Image [Online]. - Dec 11, 2019. - Sep 8, 2021. - https://arxiv.org/abs/1912.12132.

15. Sun Kaiming He Xiangyu Zhang Shaoqing Ren Jian Deep Residual Learning for Image Recognition [Online]. - Dec 10, 2015. - Aug 16, 2021. - https://arxiv.org/abs/1512.03385.
16. Xingjian Shi Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, Wang-chun Woo Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting [Online]. - Jun 13, 2015. - Oct 10, 2021. - https://arxiv.org/abs/1506.04214.
17. Yitian Chen Yanfei Kang, Yixiong Chen, Zizhuo Wang Probabilistic Forecasting with Temporal Convolutional Neural Network [Online]. - Jun 11, 2019. - Aug 17, 2021. - https://arxiv.org/abs/1906.04397.

© Быков Ф.Л., Царалов Н.Д., 2022 Научно-образовательный журнал для студентов и преподавателей «StudNet» №4/2022.

Для цитирования: Быков Ф.Л., Царалов Н.Д. СОВРЕМЕННЫЕ ПРАКТИКИ ПРИМЕНЕНИЯ МАШИННОГО ОБУЧЕНИЯ В ЗАДАЧЕ ПРОГНОЗА ПОГОДЫ// Научно-образовательный журнал для студентов и преподавателей «StudNet» №4/2022.