Ф.Л.Быков

ФБГУ «Гидрометцентр России»

Нейронные сети в задачах прогноза погоды

2018

Нейронные сети

Нейронная сеть – вычислительный алгоритм, состоящий из блоков: линейных операторов $\mathbb{R}^n \to \mathbb{R}^m$ и нелинейных функций (***т.н. функций активации***). Процесс ***обучения*** нейронной сети – настройка коэффициентов в линейных операторах. Показано, что если функции активации – кусочно-гладкие с ограниченной производной, то процесс обучения сходится.

Так как два последовательных линейных (нелинейных) оператора – так же линейный (нелинейный) оператор, то, как правило, сеть делят на ***слои*** – блоки, состоящие из линейного и следующего за ним нелинейного оператора.

Обучение с учителем и функция потерь

Обучение с учителем – процесс обучения по наборам данных, состоящих из различных реализаций векторов входных данных и соответствующих им целевых векторов

Функция потерь – минимизируемая кусочно-гладкая метрика. Например:

$$RMS(X_{fact}, X_{pred}) = \frac{1}{2}\sum_j (X_{predj} - X_{factj})^2, \frac{dRMS(X_{fact}, X_{pred})}{dX_{predj}} = X_{predj} - X_{factj}$$

$$ABS(X_{fact}, X_{pred}) = \sum_j \left| X_{predj} - X_{factj} \right|, \frac{dABS(X_{fact}, X_{pred})}{dX_{predj}} = sign\left(X_{predj} - X_{factj} \right)$$

Прогноз вероятности

Пусть мы прогнозируем вероятность события X_{pred}. Если оно состоялось a раз и не состоялось b раз, то логарифм функции правдоподобия:

$$\log X_{pred}^{a}\left(1-X_{pred}\right)^{b}=a\log X_{pred}+b\log\left(1-X_{pred}\right)\rightarrow\max$$

Пусть $X_{fact}=0$ или 1, тогда т.н. ***бинарная*** функция потерь:

$$binloss(X_{fact},X_{pred})=-X_{fact}\log X_{pred}-\left(1-X_{fact}\right)\log\left(1-X_{pred}\right)\rightarrow\min$$

$$\frac{dbinloss(X_{fact},X_{pred})}{dX_{pred}}=\frac{1-X_{fact}}{1-X_{pred}}-\frac{X_{fact}}{X_{pred}}$$

Метод обратного распространения ошибки

Метод ***обратного распространения ошибки*** (МОРО) позволяет оценить градиент функции потерь *e* – по коэффициентам операторов и по результирующим векторам $\vec{y}$ каждого слоя сети, начиная с последнего, для которого $\vec{y} = X_{pred}$.

Полносвязный слой

Полносвязный слой с функцией активации *f*:

$$\vec{y} = f(W_{yx}\vec{x} + \vec{b}_y),$$

где $\vec{x}$ – входной вектор, $\vec{y}$ – результирующий, *f* применяется поэлементно. Здесь и далее везде буквой *W* обозначается линейный оператор, а *b* – вектор, которые необходимо оптимизировать в процессе обучения. Пусть *E* – выборочное мат. ожидание. МОРО для полносвязного слоя:

$$\frac{\partial e}{\partial \vec{x}} = W_{yx}{}^{T}\left(\frac{df}{d\vec{y}} \circ \frac{\partial e}{\partial \vec{y}}\right)$$

$$\frac{\partial e}{\partial W_{yx}} = E\left[\left(\frac{df}{d\vec{y}} \circ \frac{\partial e}{\partial \vec{y}}\right)\vec{x}^{T}\right]$$

$$\frac{\partial e}{\partial \vec{b}} = E\left(\frac{df}{d\vec{y}} \circ \frac{\partial e}{\partial \vec{y}}\right)$$

Инициализация

Перед обучением необходимо как-то инициализировать коэффициенты сети. Сеть обучается быстрее, если модуль градиента функции потерь не слишком сильно отличается для разных слоёв, то есть

$$Var\left(W_{yx}{}^{T} \frac{\partial e}{\partial \vec{y}} \right) \approx Var\left(\frac{\partial e}{\partial \vec{y}} \right)$$

Можно показать, что это будет выполнено, если коэффициенты оператора W_{yx} выбраны из распределения с нулевым средним и с дисперсией

$$Var\left(W_{yx}\right) = \frac{6}{\dim(\vec{x}) + \dim(\vec{y})}$$

- т.н. ***инициализация Ксавье.***

Стохастический градиентный спуск

На больших архивах данных каждый раз вычислять градиенты по всему архиву данных слишком дорого. Архив случайным образом разобьем на малые части minibatch **(**s**)** и подстраиваем веса по каждой из частей последовательно методом градиентного спуска:

$$W_{yx} \to W_{yx} - \eta \frac{\partial e}{\partial W_{yx}}$$

$$\vec{b} \to \vec{b} - \eta \frac{\partial e}{\partial \vec{b}}$$

где η – скорость обучения, которую необходимо уменьшать в процессе обучения.

Проход по всему архиву для обучения – ***эпоха обучения***.

Адаптивный градиентный спуск Adam

«Метод тяжелого шарика», т.е. градиентный спуск с инерцией:

$$m_{yx} \to \alpha m_{yx} + (1-\alpha)\frac{\partial e}{\partial W_{yx}}$$

$$D_{yx} \to \beta D_{yx} + (1-\beta)\left(\frac{\partial e}{\partial W_{yx}}\right)^2$$

$$W_{yx} \to W_{yx} - \eta \frac{m_{yx}}{\sqrt{D_{yx}} + \varepsilon}$$

Здесь все операции выполняются поэлементно. Стандартные параметры: $\alpha = 0.9, \beta = 0.999, \varepsilon = 10^{-6}$. В этом алгоритме скорость обучения η можно не уменьшать.

При $\alpha = 0$ - алгоритм RMSProp, при $\beta = 0$ - алгоритм Adagrad.

Embedding

Embedding – отображение из *K* категорий в *N*-мерное линейное пространство. Можно рассматривать как линейный оператор: если *x* принадлежит категории *k,* то его можно задать *M*-мерным вектором $\vec{x}$:

$$x_j = \delta_{jk},$$
$$Embedding = W_{xk}\vec{x},$$

где W_{xk} – линейный оператор $\mathbb{R}^K \to \mathbb{R}^N$. Таким образом, Embedding – специальный слой нейронной сети. Embedding принимает на вход категории, а значит должен быть первым слоем.

Примеры: отображение слов языка в 300-мерное пространство; отображение номеров синоптических станций в 2-мерное пространство.

Задача коррекции прогноза

Далее *T* – температура воздуха, *D* – температура точки росы, *P* – атмосферное давление, *S* – скорость ветра, *G* – скорость порывов ветра.

Пусть $X(t,z)$ – прогноз $X = T, D, S$ от срока t на $t+z.$ Задача коррекции: по имеющимся данным найти поправки $\delta X(t,z)$ к существующим прогнозам, такие что $X(t,z) + \delta X(t,z)$ – прогноз лучше, чем исходный.

Синоптики замечают, что некоторые модели систематически недо- или переоценивают определенные параметры при определенных условиях.

Систематическая коррекция прогнозов COSMO-ENA13

Выберем скользящий период обучения длиной $t_z = 30$ суток.

L - оператор осреднения функции $x(t,z)$ с весами:

$$L[x(t,z)] = \frac{\sum_{j=\lfloor z \rfloor}^{j=t_z} w(t-j,t,z)x(t-j,z)}{\sum_{j=\lfloor z \rfloor}^{j=t_z} w(t-j,t,z)}, \qquad w(s,t,z) = \exp\left(\lambda(t-s)\right), \quad \lambda = 0.13 сут^{-1}$$

где веса

Оценим смещение прогнозов $\mu_X(t,z) = L\left[X(t,z) - X_{fact}(t+z)\right]$

Тогда систематически поправленные прогнозы:

$$T_C(t,z) = T_0(t,z) - \mu_{T_0}(t,z)$$
$$D_C(t,z) = D_0(t,z) - \mu_{D_0}(t,z)$$
$$U_C'(t,z) = U_0(t,z) - \mu_{U0}(t,z)$$
$$V_C'(t,z) = V_0(t,z) - \mu_{V0}(t,z)$$
$$S_C'(t,z) = S_0(t,z) - \mu_{S0}(t,z)$$
$$S_C(t,z) = \max\left(0, S_C'(t,z)\right)$$
$$\left\langle U_C(t,z), V_C(t,z)\right\rangle = \frac{S_C(t,z)\left\langle U_C'(t,z), V_C'(t,z)\right\rangle}{\sqrt{U_C'(t,z)^2 + V_C'(t,z)^2}}$$

Слой коррекции

При выполнении условий (которые необходимо найти) на входные параметры вводим коррекции (которые так же необходимо определить) этих параметров.

Формально:

$$\vec{h} = \max\left(0, W_{hx}\vec{x} + \vec{b}_h\right)$$
$$\vec{r} = W_{rh}\vec{h} + \vec{b}_r$$
$$\vec{g} = W_{gh}\vec{h} + \vec{b}_g$$
$$\vec{y} = \vec{x} + \vec{r} \circ \sigma\left(\vec{g}\right)$$

$\vec{h}$ - условия введения поправок

$\vec{r}$ - величины поправок

$\sigma\left(\vec{g}\right)$ - нелинейность поправок:

$$\sigma\left(x\right) = \left(1 + \mathrm{e}^{-x}\right)^{-1}$$

При больших положительных g - линейная поправка, при больших отрицательных – без поправки.

Предикторы сети N1 для коррекции прогнозов

Всего используем 49 предикторов:

1. $Y_0(t_0, z_0)$, где $Y = P, T, D, G, S$;

2. $Y_0(t, z) - Y_0(t_0, z_0)$, где $Y = P, T, D, G, S$ в 8 моментах времени: $t = t_0, t_0 - 1, t_0 - 2,$ $z = z_0 - 3ч, z_0, z_0 + 3ч$, но $(t, z) \neq (t_0, z_0)$;

3. При $t = t_0 + z_0 - 3ч, t_0 + z_0, t_0 + z_0 + 3ч$ - высота Солнца над горизонтом;

4. Заблаговременность z_0.

У сети N2 те же предикторы, но вместо $T_0(t, z)$, $D_0(t, z)$, $S_0(t, z)$ используются поправленные прогнозы $T_C(t, z)$, $D_C(t, z)$, $S_C(t, z)$.

Номер слоя	Тип слоя, функция активации	Размерность выходного вектора	Количество обучаемых мультипликативных/аддитивных параметров
1	Полносвязный, th	16	784/16
2	Коррекция	16	768/48
3	Полносвязный, th	12	192/12
4	Коррекция	12	432/36
5	Полносвязный, th	8	96/8
6	Полносвязный	1	8/1
Всего			2280/121

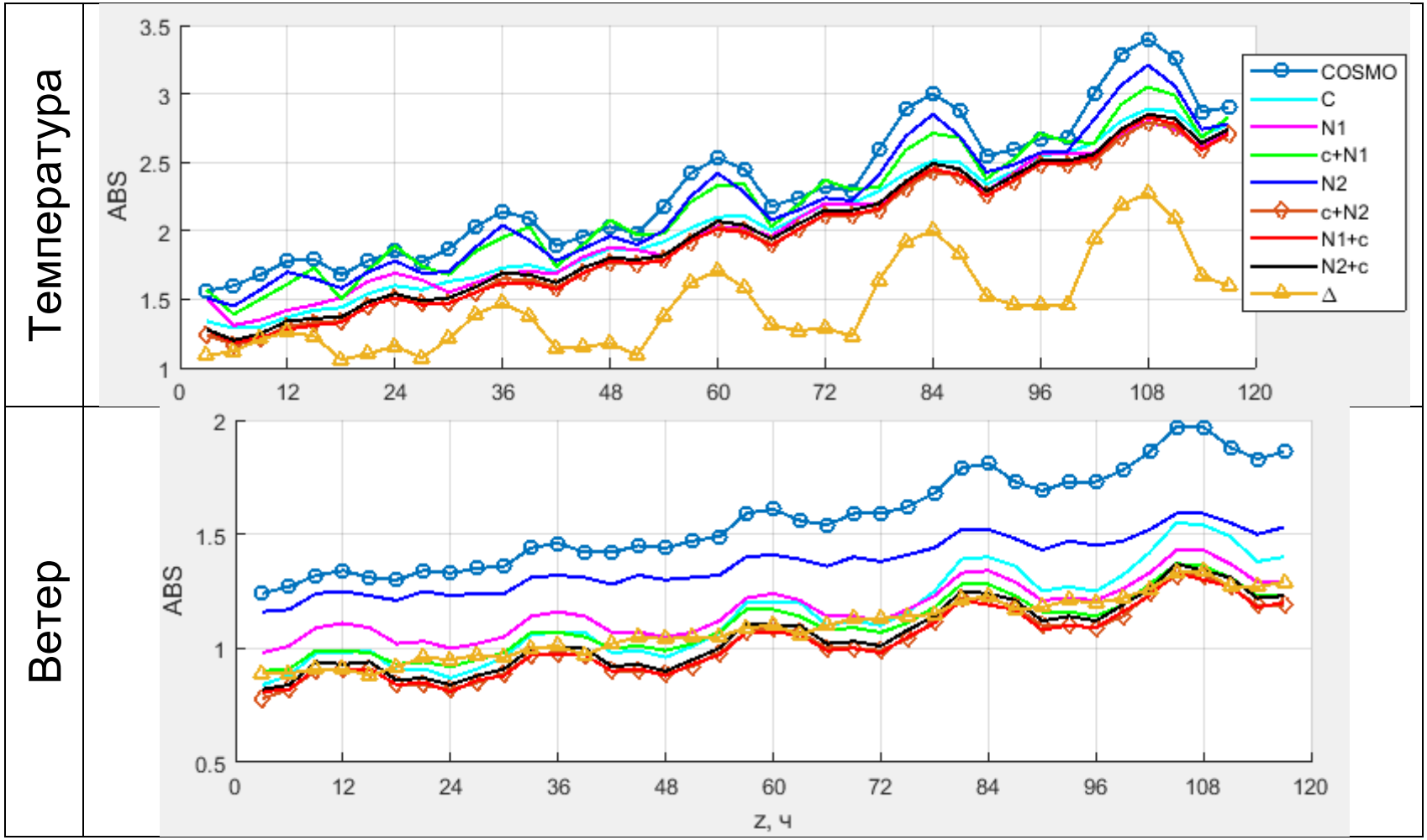
Технические подробности

Наборов данных для обучения **~39 млн** прогнозов модели COSMO-ENA13 за 2017г. 250 эпох обучения. Размер minibatch = 2048. Скорость обучения Adam $\eta = 0.01$.

Для улучшения устойчивости (робастности) алгоритма обучения, каждая из нейронных сетей обучались в 3 различных вариантах с различной инициализацией и на различных, случайным образом выбранных, 80% данных, а при тестировании выбиралось медианное значение из этих 3 вариантов.

Далее метод систематической коррекции обозначается как **C**, а, например, последовательное применение **C**, а затем **N2** как **C+N2**. Тестирование на архиве **~22 млн.** прогнозов за январь-октябрь 2018г

Оценки на ETP в зависимости от заблаговременности *z*

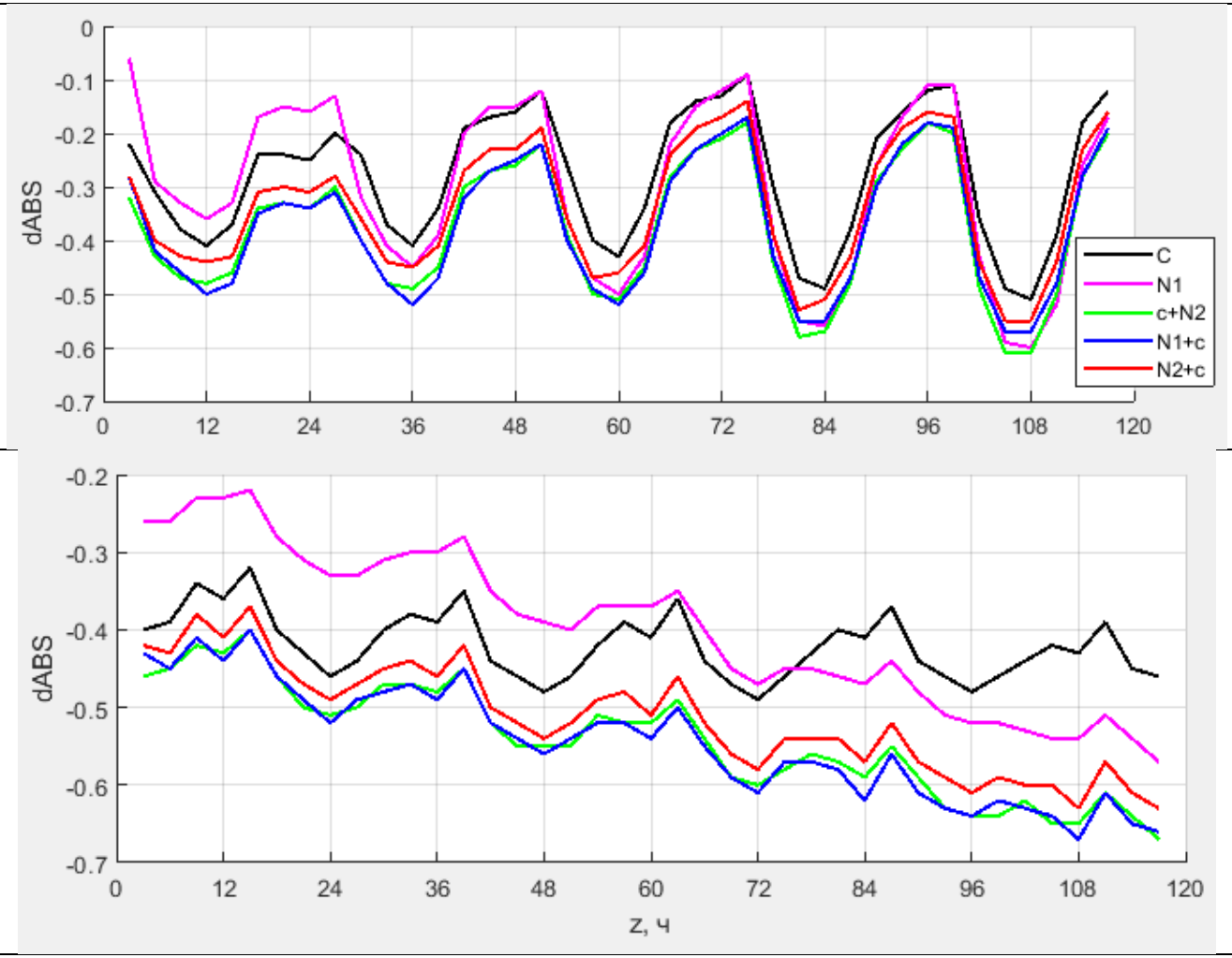


Δ - средняя абсолютная величина поправки

Изменение погрешностей на ETP

Температура

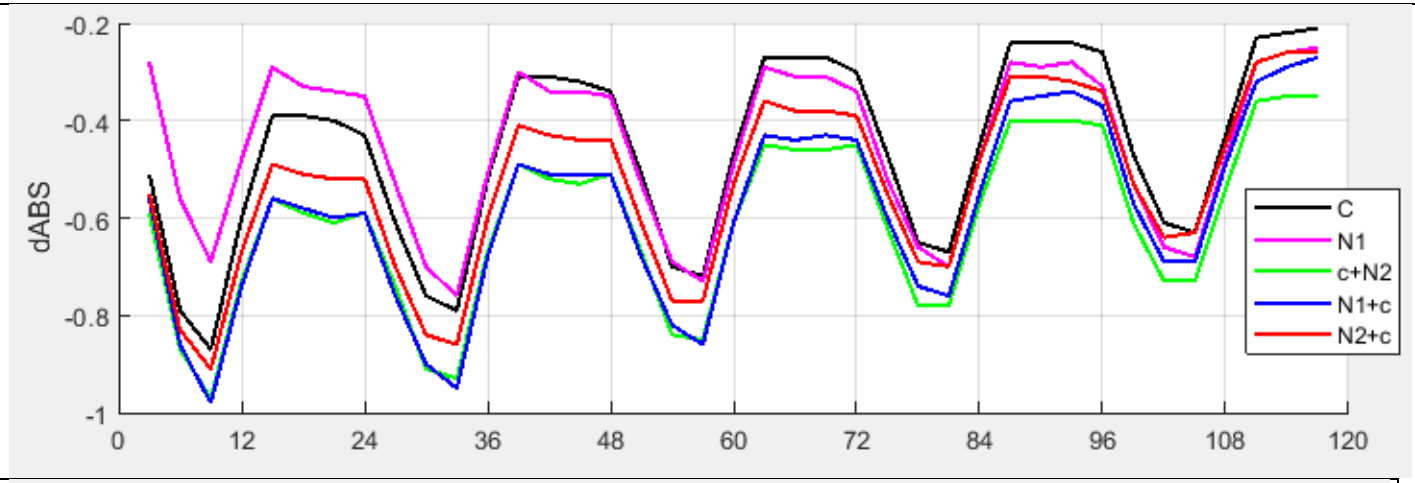
Ветер



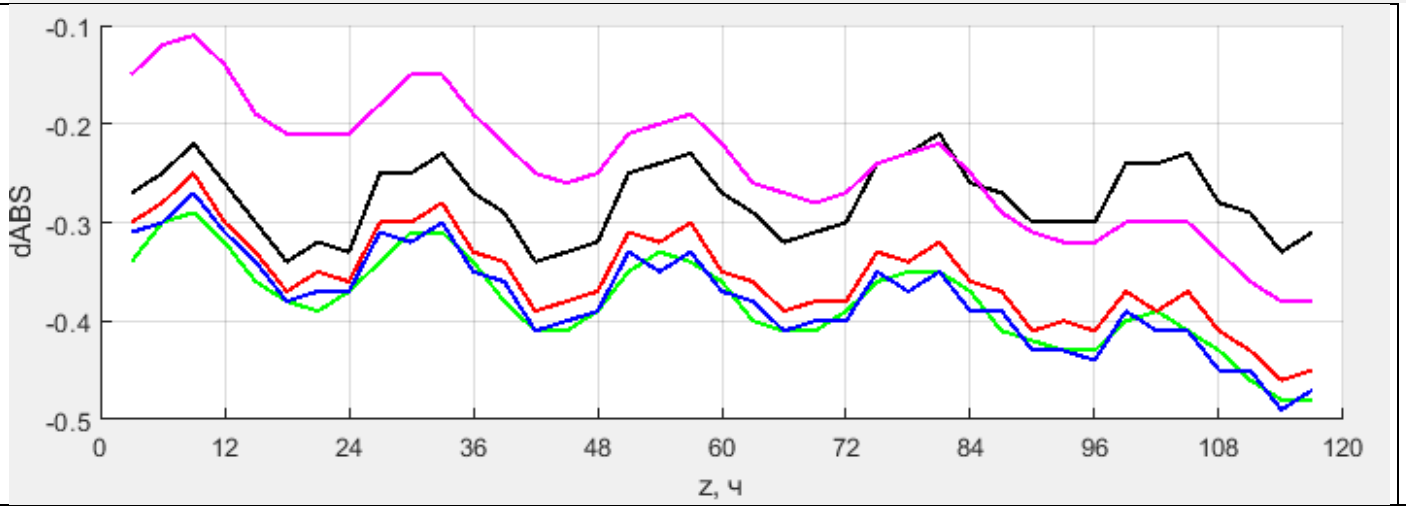
При $z > 30ч$ (для ветра при $z > 75ч$) сеть **N1** (не использующая измерения) лучше, чем систематическая коррекция **C**

Изменение погрешностей на АТР

Температура



Ветер

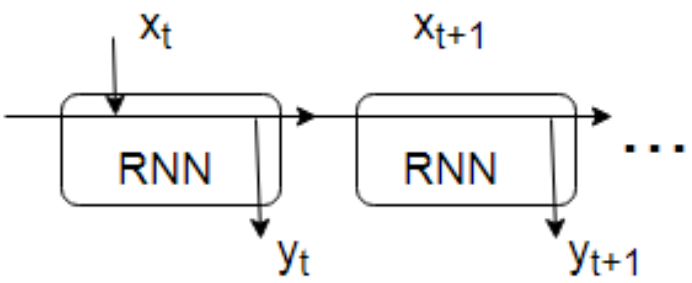


Первое место делят комбинированные методы **N1+C** и **C+N2.**

Для температуры **C+N2** имеет преимущество при $z > 60ч$

Рекуррентные нейронные сети

Рекуррентные нейронные сети – сети для обработки временных рядов, а каждый момент времени они используют, в том числе и результаты вычислений сети в предыдущий момент времени



Обучение рекуррентных сетей осуществляется методом распространения ошибки сквозь время (BTTP). В этом методе рекуррентная сеть рассматривается как большая нейронная сеть, принимающая и возвращающая данные за все моменты времени. МОРО осуществляется, начиная с последнего момента времени. Затем градиент функции потерь на каждом из коэффициентов усредняется по времени.

Vanilla RNN

$$\vec{h}_{t+1} = W_{hh} f\left(\vec{h}_t\right) + W_{hx}\vec{x}_{t+1} + \vec{b}_t$$

Распространение ошибки сквозь время: $\frac{\partial e}{\partial \vec{h}_t} = W_{hh}{}^T \left(f'\left(\vec{h}_t\right) \circ \frac{\partial e}{\partial \vec{h}_{t+1}} \right)$

Проблемы Vanilla RNN:

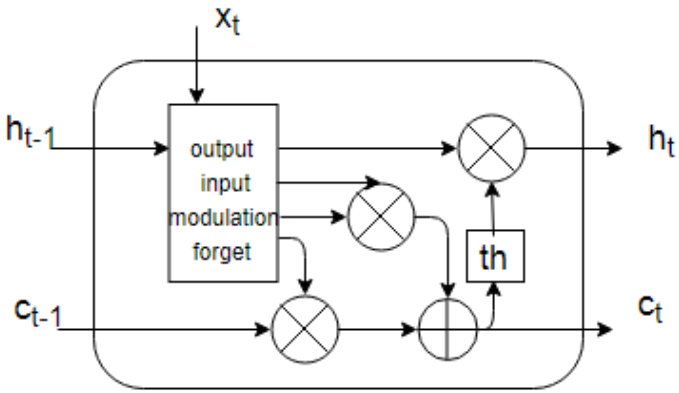
1. Нужна инициализация при t = 0.
2. Экспоненциальный рост/убывание градиентов функции потерь в зависимости от W_{hh}: $\left\| \frac{\partial h_s}{\partial h_t} \right\| = \left\| \prod_{j=t+1}^{s} \frac{\partial \vec{h}_j}{\partial \vec{h}_{j-1}} \right\| \approx \left\| W_{hh} \right\|^{s-t} \left\| \prod_{j=t+1}^{s} f'\left(\vec{h}_{j-1}\right) \right\|$

   Vanilla RNN невозможно научить искать дальние зависимости.

Сеть с памятью LSTM

Пусть $\sigma(x) = \left(1 + \mathrm{e}^{-x}\right)^{-1}$, тогда уравнения LSTM можно записать как

Забывающий вентиль	$f_t = \sigma\left(W_{fx} x_t + W_{fh} h_{t-1} + b_f\right)$
Входящий вентиль	$i_t = \sigma\left(W_{ix} x_t + W_{ih} h_{t-1} + b_i\right)$
Модуляция	$m_t = W_{mx} x_t + W_{mh} h_{t-1} + b_c$
Обновление состояния	$c_t = f_t \circ c_{t-1} + i_t \circ m_t$
Выходящий вентиль	$o_t = \sigma\left(W_{ox} x_t + W_{oh} h_{t-1} + b_o\right)$
Результат	$h_t = \tanh(c_t) \circ o_t$



Градиент функции потерь затухает в соответствии с забывающим вентилем

f_t :

$$\frac{\partial e_t}{\partial c_t} = f_{t+1} \circ \frac{\partial e_t}{\partial c_{t+1}} + o_t \circ \tanh'\left(c_t\right) \circ \frac{\partial e_t}{\partial h_t}$$

Можно научиться дальние зависимости

Инициализация b_f очень важна: временной масштаб процесса

Модификация LSTM

Предпосылки:

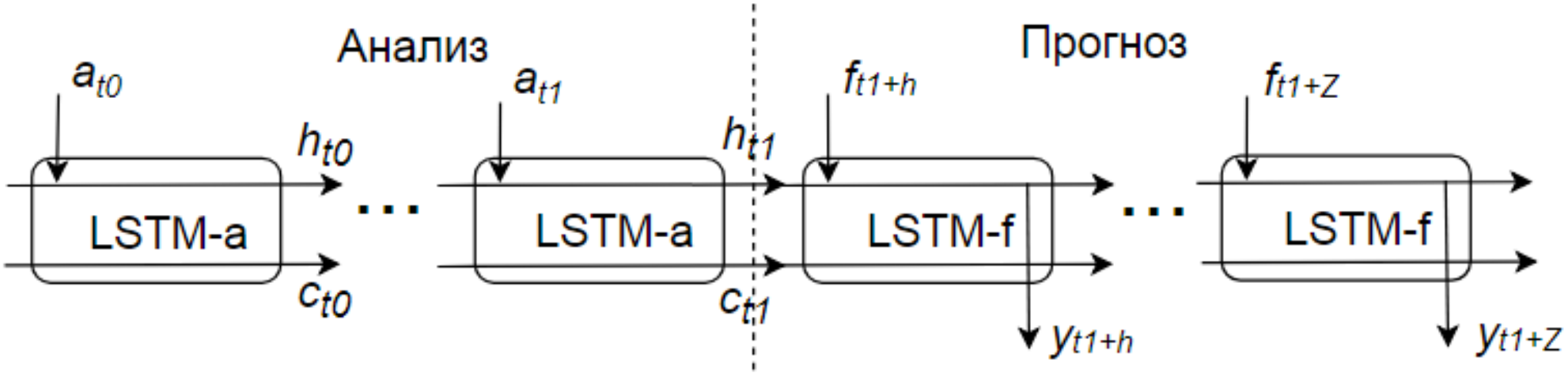
1. Не все значения состояний системы интересны
2. Размерности результирующего вектора и вектора состояний в обычном LSTM одинакового размера

Модификация: последнее уравнение $h_t = \tanh(c_t) \circ o_t$

заменяем на $h_t = \tanh\left(W_{hc} \max\left(0, c_t\right) + b_h\right) \circ o_t$

LSTM анализ-прогноз (encoder-decoder)

Учим LSTM сеть, составленную из блоков двух типов: анализа и прогноза



Данные METAR

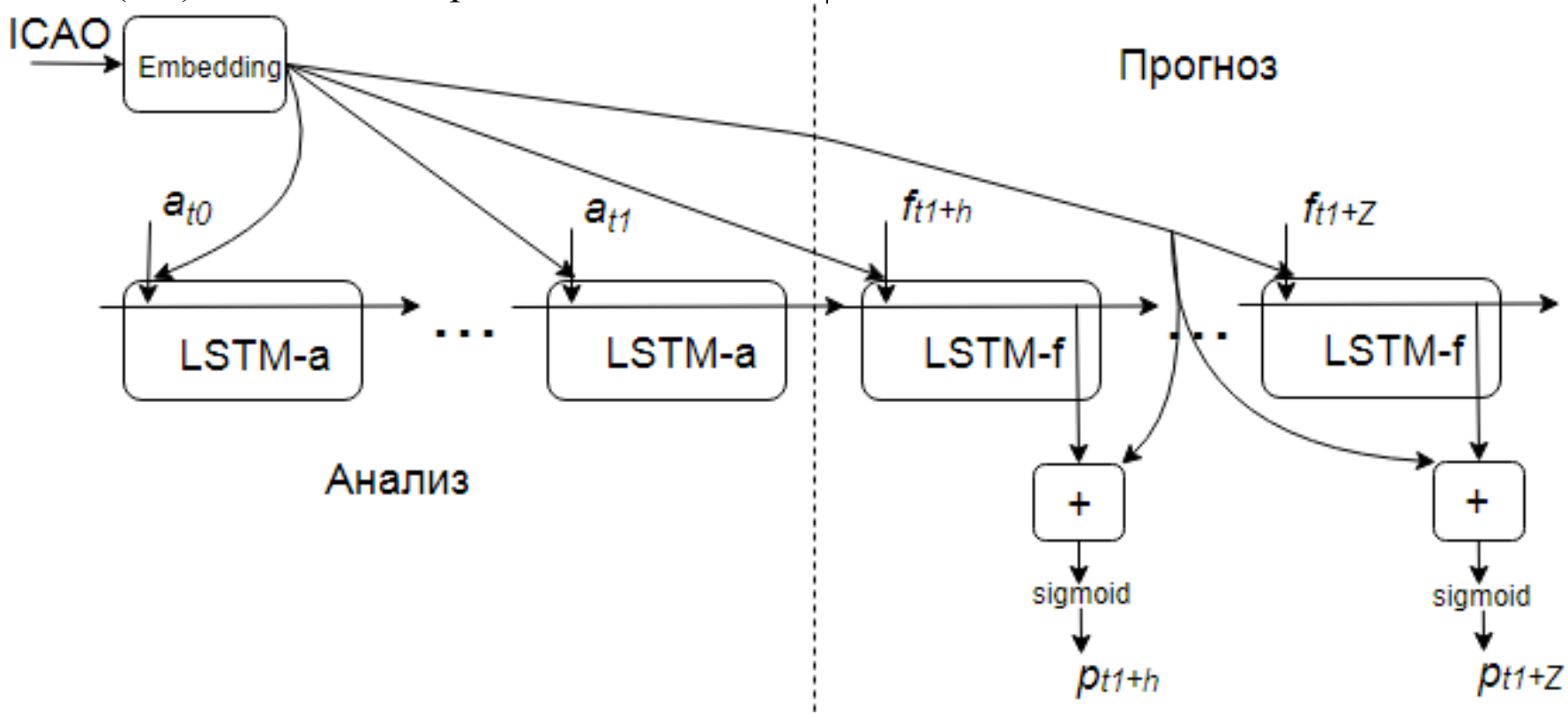
Поступают с аэропортов, как правило, раз в 30 или 60мин. Содержат данные о: температуре воздуха, температуре точки росы, атмосферном давлении, скорости ветра, **дальности видимости** *Vis*. Далее время дискретно с шагом *h=30мин.* Обучение на данных за 2015-2016гг. – всего **35088** сроков.

Всего было доступно **~69.8 млн**. сводок с **4231** аэропорта ICAO (47% возможного количества). Регулярно передавали сводки **2426** аэропорта– в сумме они передали **51,7** млн. сводок (60,7% возможного количества). Данные о температуре воздуха, температуре точки росы, атмосферном давлении были кусочно-линейно проинтерполированы по времени, но не более чем на 3ч. После интерполяции данные присутствуют в 89.3% сроков.

Цель: прогноз вероятности малой (менее 2км) видимости

Вектор предикторов анализа a_t содержит значения при *(s-t)/h = -3, -2, -1, 0:*
Температуры воздуха *T(s)* и точки росы *D(s)*
Давления на уровне моря $P_0(s)$
Высоты Солнца над горизонтом
Логарифма видимости *log(1+Vis(s)/10м)*
Флага (0/1) наличия измерения *Vis*

Вектор предикторов прогноза f_t содержит значения при *(s-t)/h = -2, -1, 0, 1, 2:*
Температуры воздуха *T(s)* и точки росы *D(s)*
Давления на уровне моря $P_0(s)$
Высоты Солнца над горизонтом



Процесс обучения

При обучении длина периода анализа t_1 - t_0 = *48ч,* максимальная заблаговременность *Z* = *12ч.* Minibatch генерировались «онлайн»:

1. Выбирались случайные 512 пар (t_0,ICAO).
2. Выбрасывались все пары: а) в интервале $t \in (t_0, t_1 + Z)$ остались пропуски данных о температур или давлении; б) в интервале $t \in (t_1 + h, t_1 + Z)$ измерений видимости *Vis* менее 40%.
3. Если осталось меньше 256 пар (t_0,ICAO), то п. 1.

Генераций minibatch 100000, скорость обучения Adam $\eta = 0.001$.

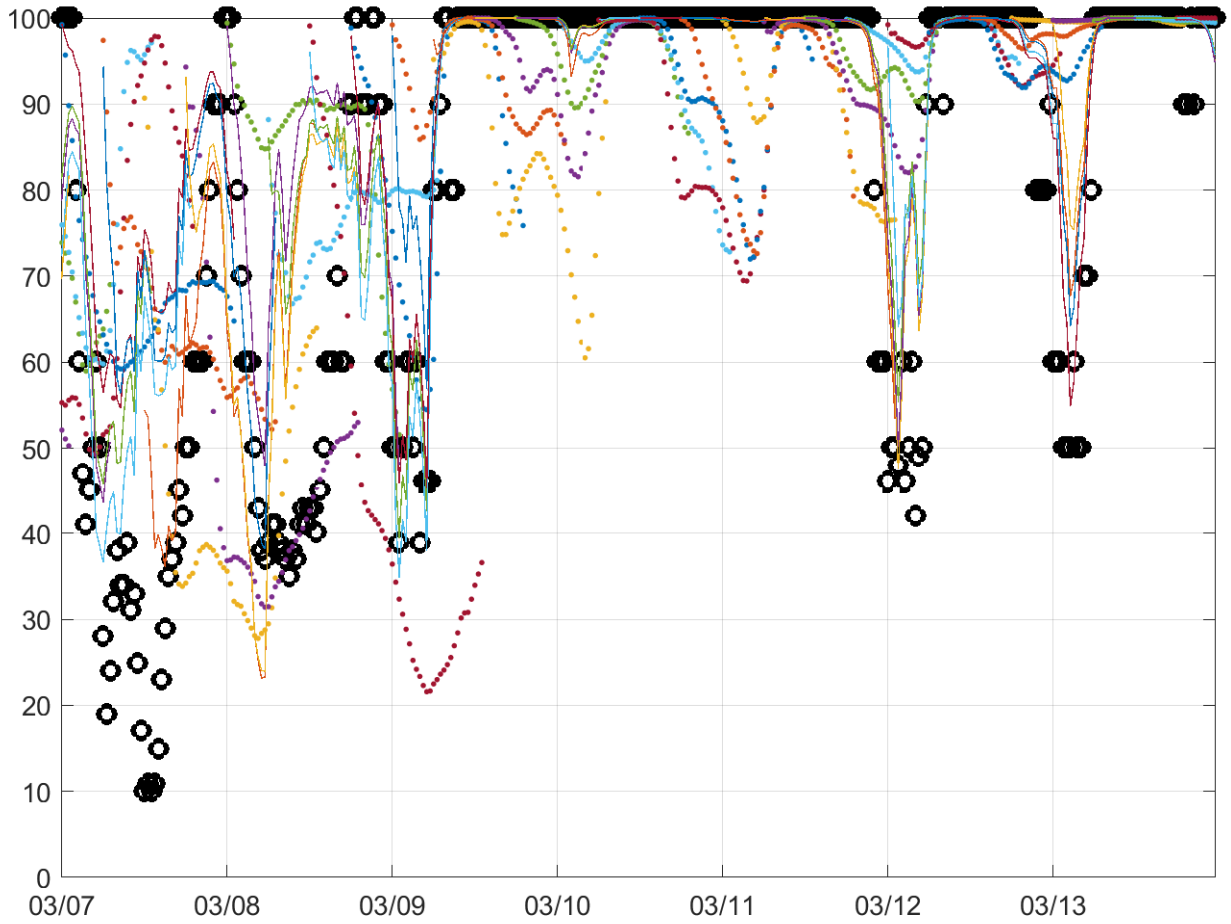
Всего обучено 15 сетей, оценивалась среднегеометрическая этих 15 прогнозов.

Оценка результатов

Обучение происходило на фактически измеренных значениях метеорологических полей (температура воздуха, точки росы и атмосферного давления). Проверка метода осуществлялась на архиве за 2017г. по 14 аэропортам ЦФО. Прогноз стартовал каждые 6ч. Всего 1448 прогнозов по каждому аэропорту. Далее прогнозы будут сравниваться при разных f_t:

1. Фактически измеренные значения
2. Прогноз модели COSMO-Ru2.2км
3. Фактически измеренные значения + случайное блуждание 0.25^{O}C/ч – приблизительный уровень точности современных моделей.

Примеры прогнозов вероятности видимости >2км



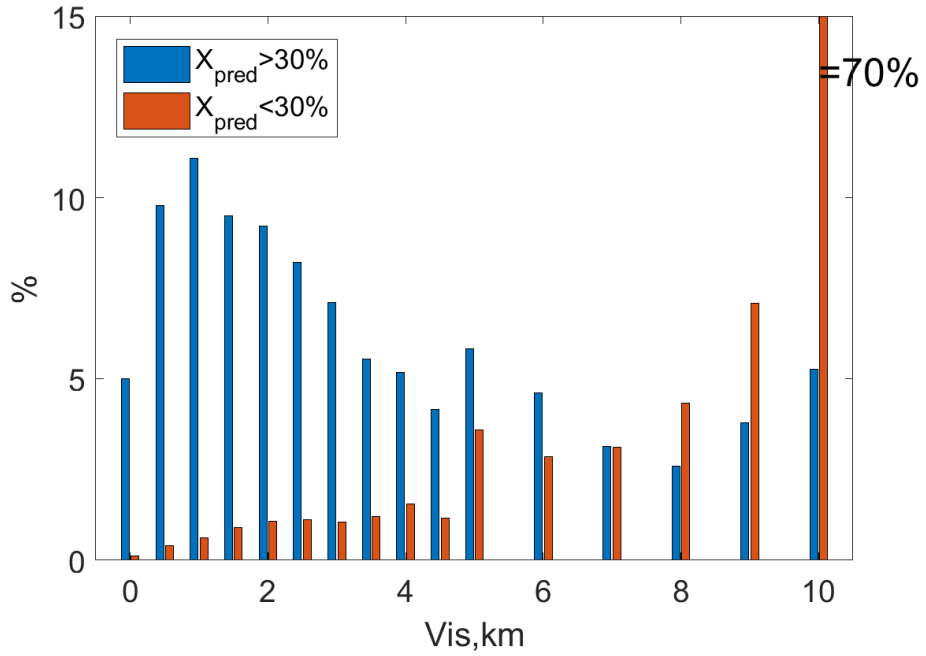
Черные кружки – видимость *100м

Линии – LSTM «прогноз» вероятности по фактической погоде

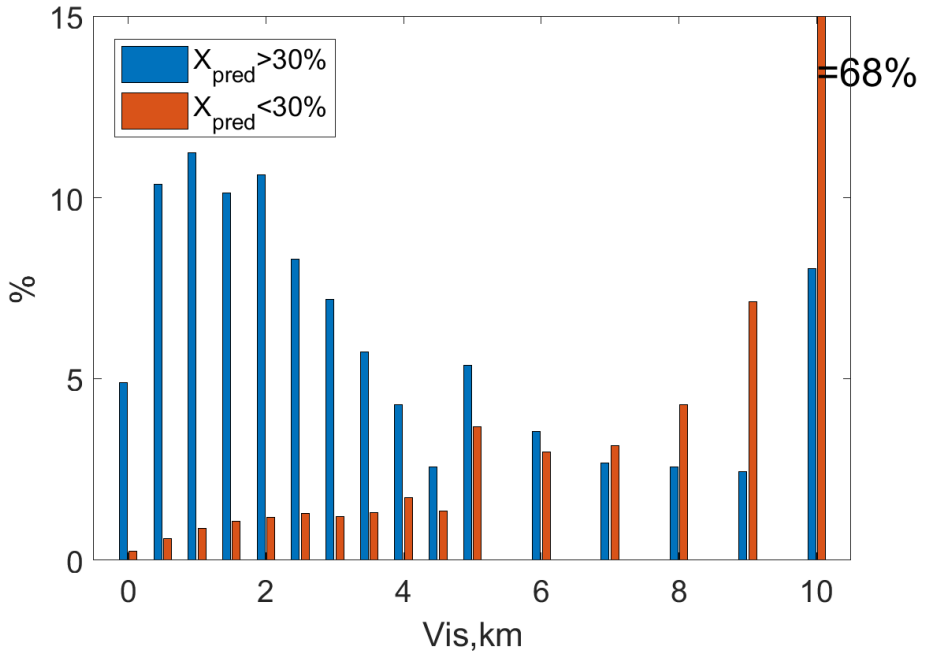
Точки – LSTM прогноз вероятности видимости более 2км по прогнозам COSMO 2.2км

Условные распределения фактической видимости

«Прогноз» на 3ч, фактическая погода

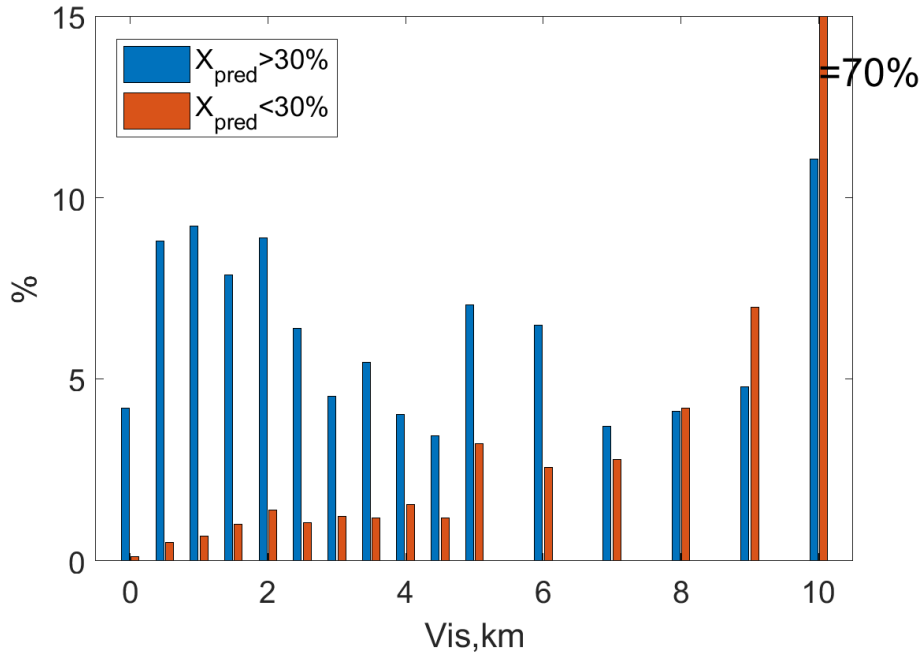


Прогноз на 3ч на базе COSMO

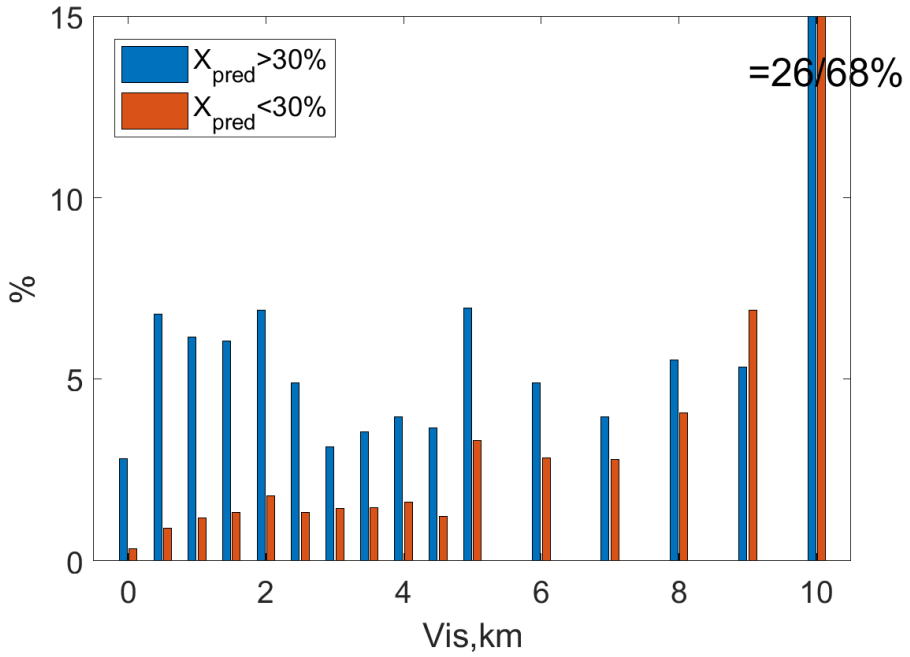


Условные распределения фактической видимости

«Прогноз» на 12ч, фактическая погода



Прогноз на 12ч на базе COSMO



Критический индекс успеха CSI

	Было	Не было
Прогнозировалось	TP	FP
Не прогнозировалось	FN	TN

Определим критический индекс успеха: $CSI = \frac{TP}{TP + FP + FN}$

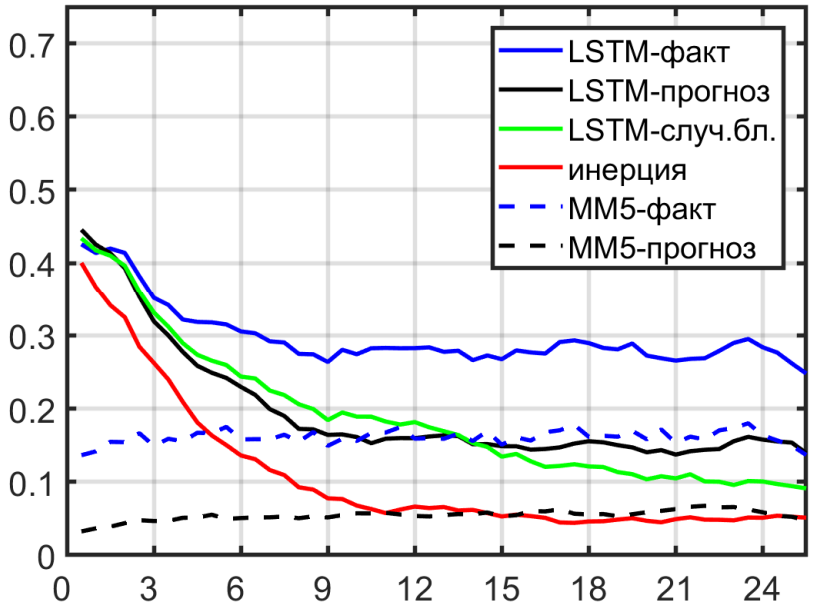
Далее приведены оценки *CSI,* если мы прогнозируем низкую видимость в случае, когда вероятностный прогноз дал более 30% вероятности видимости менее 2км. Для сравнения использован полуэмпирический метод **MM5**:

$$Vis = 9656 \frac{T - D}{R^{1.75}}$$

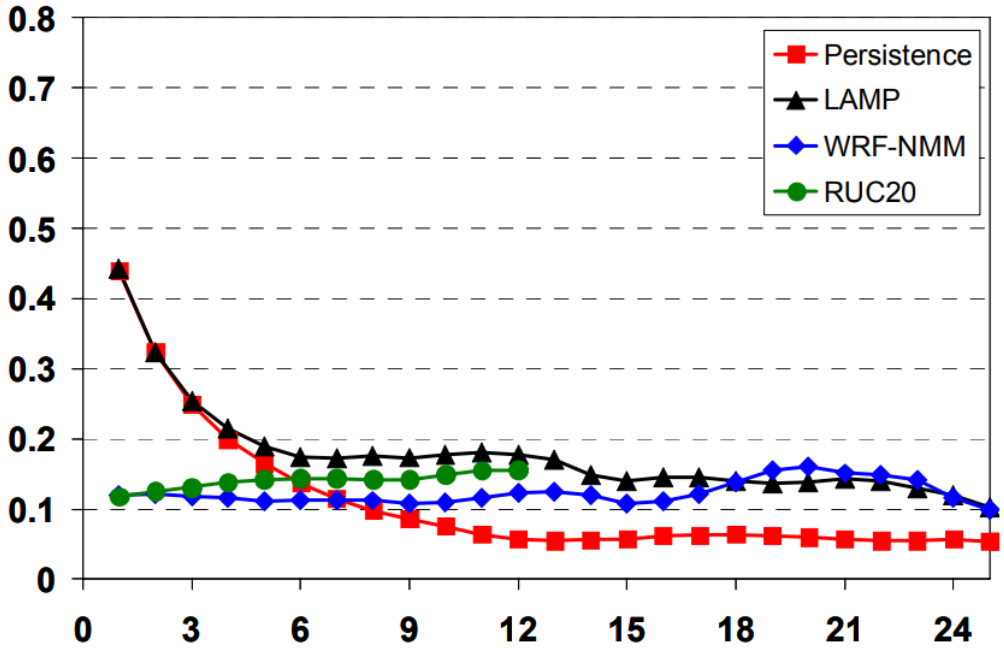
где R – относительная влажность в %.

Слева CSI при разных f_t: фактическая погода, прогноз модели COSMO-Ru2.2км, факт. погода + случайное блуждание 0.25^{O}C/ч.

14 аэропортов ЦФО, 2017г



По территории США (1462 а/п) из [Rudack-08]



Спасибо за внимание!