Statistical Post-processing of wind speed forecasts using convolutional neural networks.

Simon Veldkamp[1,2], Maurice Schmeits[1] and Kirien Whan[1].

[1]Royal Netherlands Meteorological Institute (KNMI), the Netherlands
[2]Mathematical Institute, Utrecht University, the Netherlands

May 1, 2020

Contact: s.veldkamp@uu.nl,maurice.schmeits@knmi.nl,kirien.whan@knmi.nl

Introduction

The model output of numerical weather prediction models is spatial in nature. The aim of this study is to use convolutional neural networks, which are capable of analyzing this spatial information, for statistical postprocessing of windspeed forecasts in the Netherlands. We compare convolutional neural networks to quantile regression forests and fully connected neural networks.

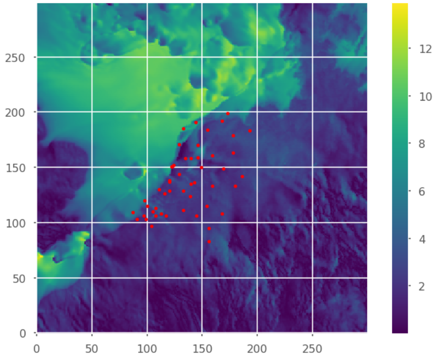
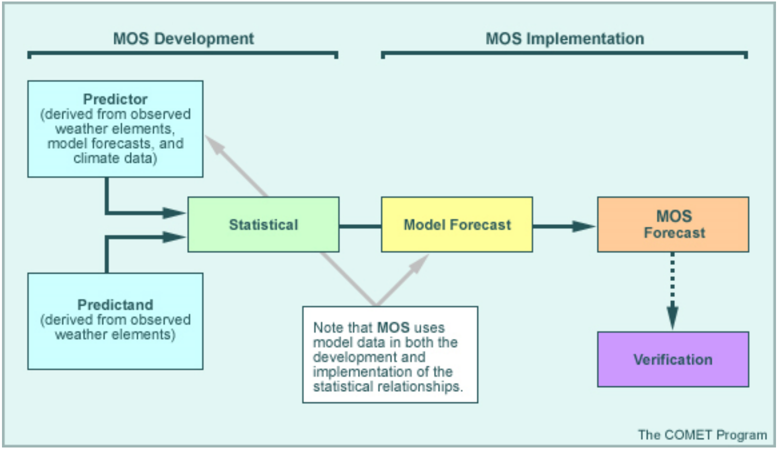


Figure: Windspeed forecasts above the Netherlands. The red dots give the locations of weatherstations in the Netherlands.

Statistical postprocessing

In model output statistics we try to find statistical relationships between the forecast made by a numerical weather prediction model and corresponding measurements to correct the bias and estimate uncertainty of forecasts.



Decision Trees

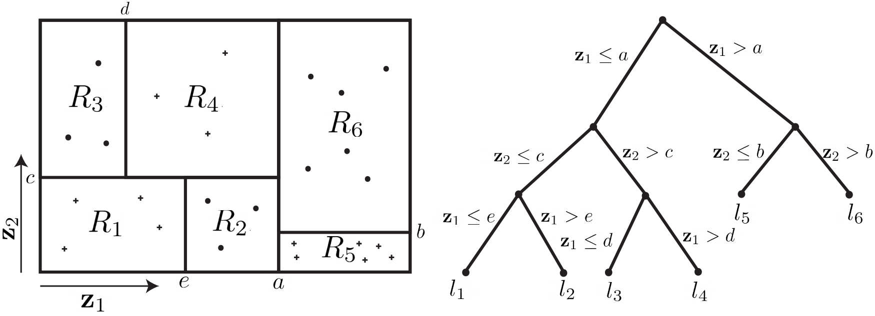


Figure: Example of a decision tree.

A random forest is an ensemble of independent decision trees. This independence between the trees is obtained by:

- Every tree is trained on a bootstrapped subset of the trainingdata.
- Every decision is based on a random subset of the predictors.

A *Quantile regression forest*(QRF) uses random forests to estimate a cumulative density function.

Neural Networks

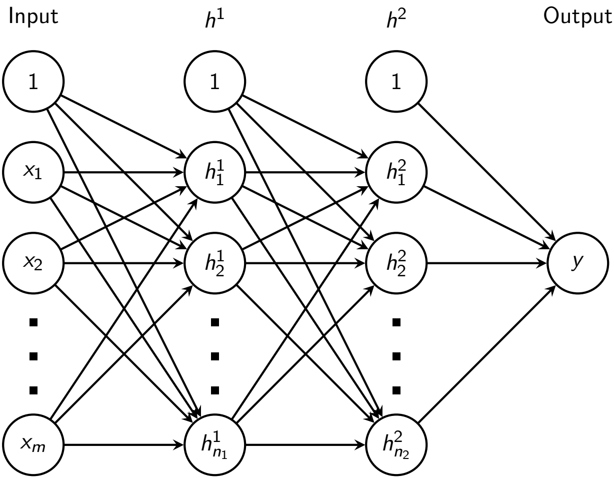


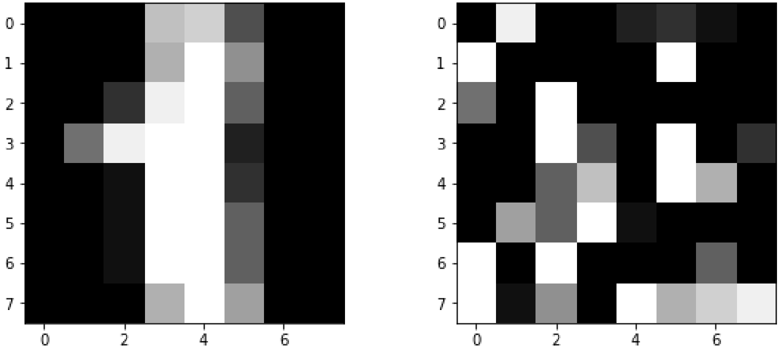
Figure: Example of a simple neural network with two hidden layers.

Neural Networks

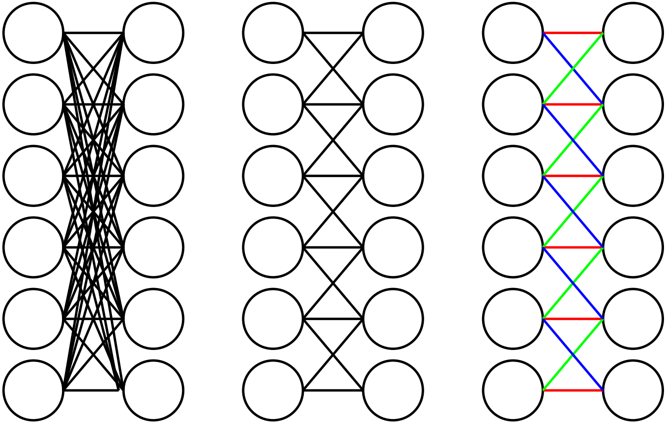
Some problems for image recognition: a 100 by 100 pixel image has 10000 input variables. So a single hidden layer of for example 100 nodes will need 1000000 different parameters.

Solutions:

- Limit the number of connections
- Use parameter sharing



Neural Networks



Convolutions

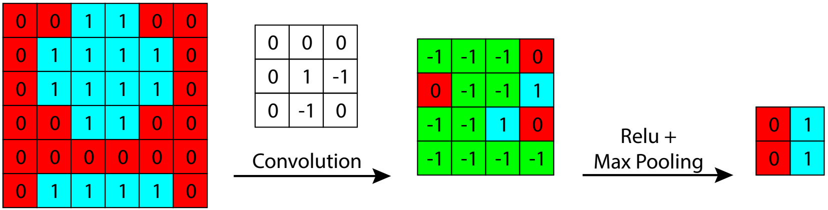


Figure: Example of a convolution, followed by a Relu activation function and Max pooling layer. A convolutional neural network generally has a few of these layers.

- The first layers of a convolutional neural network detects simple structure such as corners and edges.
- The subsequent layers combine these simple structures into more complex structures.

Conditional Density Estimation

We used three different methods to obtain a probability density estimate with neural networks:

Quantized Softmax: In Quantized softmax the model output is a histogram with pre-defined bins.

Kernel Mixture Network: In a Kernel Mixture Network the model output are the weights and standard deviations for a sum of Gaussian with pre-defined means.

Truncated normal: In the final method the output layer of the neural network has only two neurons representing the mean and standard deviation of a truncated normal.

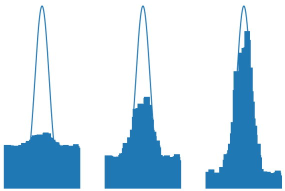


Figure: Quantized Softmax

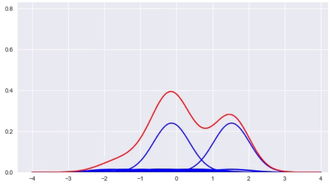


Figure: Kernel mixture network

Data

Training data

- Data from October to March of 2015, 2016 and 2017
  - +48 h Harmonie-Arome (HA) reforecasts of
    - wind speed at 10 m (ff)
    - (co)sine of wind direction at 10 m and 925 hPa
    - surface roughness
    - surface pressure
  - observations of 10-m wind speed (ff) at stations in the Netherlands
  - all stations are pooled
  - valid at 00 UTC
- For CNNs +48 h HA 10-m wind speed fields (100x100 grid points) around the stations are also used.

Test data

- Data from November 2018 to March of 2019, October 2019 to November 2019

Forecast verification

Forecasts are generally evaluated using the CRPS and the Brier score. The Brier score for a threshold value c is defined as:

$$BS_c(\hat{F}, y) = ((1 - \hat{F}(c)) - \mathbb{1}_{[c,\infty)}(y))^2.$$

$$CRPS(\hat{F}, y) = \int_{\mathbb{R}} (F(c) - \mathbb{1}_{[y,\infty)}(c))^2 dc,$$

Models are generally compared to each other through skill scores:

$$BSS_c = 1 - \frac{BS_c(a)}{BS_c(b)}$$

$$CRPSS = 1 - \frac{CRPS(a)}{CRPS(b)}.$$

Finally to asses whether a model is well calibrated we use the cumulative rank histogram:

$$\hat{F}_{F(Y)}(z) = \frac{1}{N} \sum_{1=1}^{N} \mathbb{1}_{(\hat{F}(y_i|\mathbf{x_i}),\infty)}(z).$$

Results

	rmse			mae			CRPS		
	CV1	CV2	CV3	CV1	CV2	CV3	CV1	CV2	CV3
NN	2.457	2.331	2.391	1.176	1.142	1.158	0.820	0.799	0.809
NN_LR:	2.204	2.126	2.176	1.109	1.090	1.099	0.793	0.779	0.786
QRF	2.244	2.220	2.245	1.116,	1.113	1.115	0.782	0.776	0.779
QRF_LR:	2.157	2.151	2.154	1.094	1.096	1.091	0.780	0.781	0.780
CNN_LR_KMN:	1.968	1.886	1.922	1.045	1.032	1.039	0.752	0.744	0.748
CNN_LR_N0:	**1.818**	1.861	2.117	**1.008**	1.021	1.076	**0.722**	0.732	0.770
CNN_LR:	1.851	**1.814**	**1.889**	1.011	**1.003**	**1.026**	0.724	**0.718**	**0.733**

Table: Results on the independent test set. Here CV1,CV2 and CV3 describe the training data used for the model(CV1 is 2015 and 2016, CV2 is 2016 and 2017, and CV3 is 2015 and 2017). The standard deviation in the CRPS was estimated by bootstrapping the test data a 1000 times and was found to be around 0.007 in all cases.

Brier Skill Scores

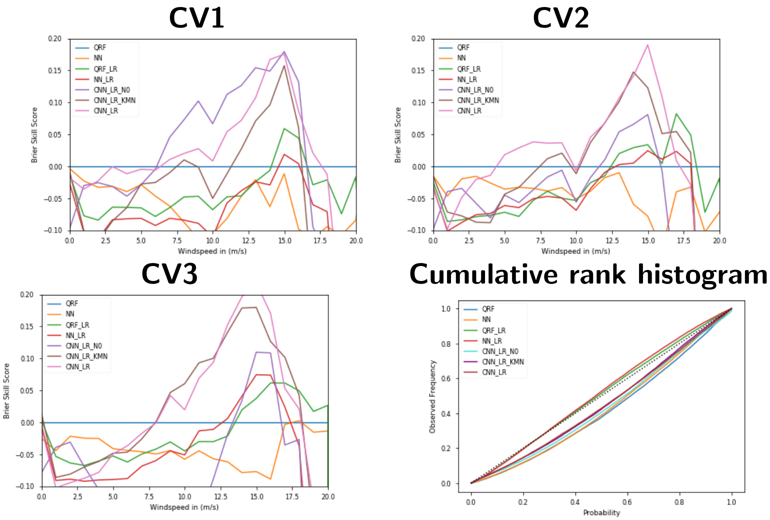


Figure: Brier skill scores relative to the Brier score of QRF, for predictions trained on three different trainingssets.

Brier Skill Scores

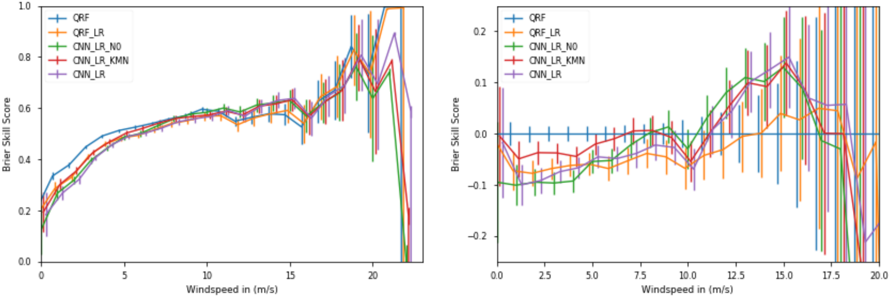


Figure: Brier skill scores relative to the station climatology(left) and QRF(right), for models trained on the full data set. Here error bars give the standard deviation which is obtained by bootstrapping the test data a 1000 times.

Results by station

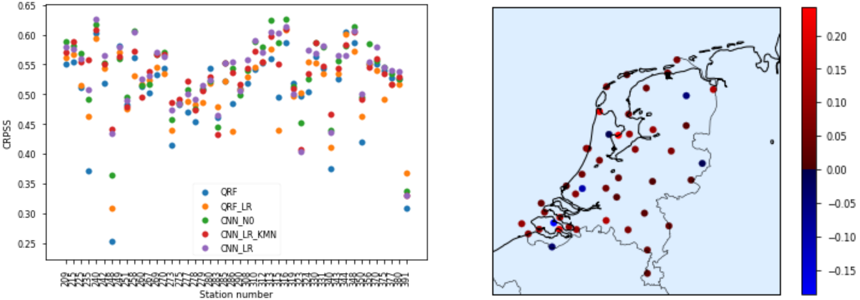


Figure: CRPSS with respect to the climatology in the left panel. The right panel shows the CRPSS of the convolutional neural network with respect to quantile regression forests.

Conclusions and future work

Conclusions:

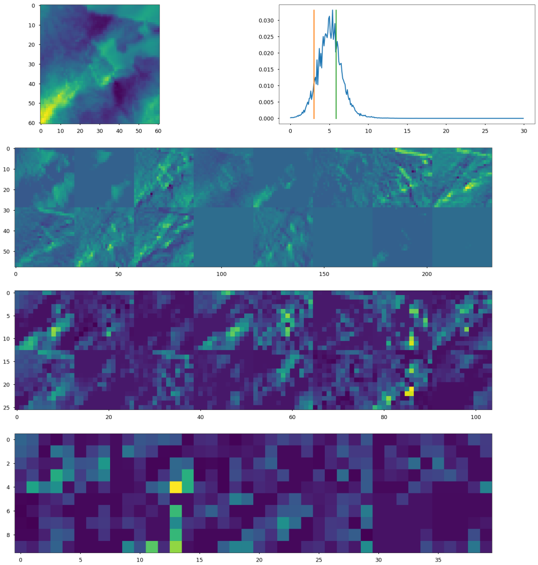
CNNs, when trained on the residual errors of linear regression, are more skillful than QRF and neural networks without convolutions:

- lowest RMSE, MAE and CRPS
- highest BSS for medium to higher wind speed thresholds

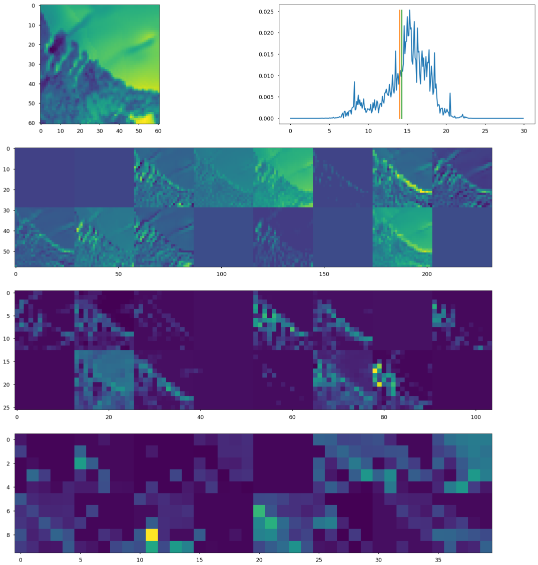
Future work:

- Try to explain what the convolutional neural networks are basing the forecast on.
- Add spatial information of other variables
- Try to use convolutional neural networks for other variables, such as, for example, precipitation.
- Use convolutional neural networks on ensemble forecasts.

Ex. 1



Ex. 2



0 10 20 30 40 50 60
0 10 20 30 40 50 60
0.025 0.020 0.015 0.010 0.005 0.000
0 5 10 15 20 25 30
0 10 20 30 40 50
0 50 100 150 200
0 5 10 15 20 25
0 20 40 60 80 100
0 2 4 6 8
0 5 10 15 20 25 30 35