Calculating the turbulent fluxes in the atmospheric surface layer with neural networks

Lukas Hubert Leufen[1,2] and Gerd Schädler[1]

[1]Institute of Meteorology and Climate Research - Department Troposphere Research, Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]now at Forschungszentrum Jülich GmbH, Jülich, Germany

Correspondence: Gerd Schädler (gerd.schaedler@kit.edu)

Abstract. The turbulent fluxes of momentum, heat and water vapour link the Earth's surface with the atmosphere. The correct modelling of the flux interactions between these two systems with very different time scales is therefore vital for climate and weather forecast models. Conventionally, these fluxes are modelled using Monin-Obukhov similarity theory (MOST) with stability functions derived from a small number of field experiments; this results in a range of formulations of these functions and thus also in differences in the flux calculations; furthermore, the underlying equations are nonlinear and have to be solved iteratively at each time step of the model. We tried here a different and more flexible approach, namely using an artificial neural network (ANN) to calculate the fluxes resp. the scaling quantities u_* and θ_*, thus avoiding function fitting and iteration. The network was trained and validated with multi-year datasets from seven grassland, forest and wetland sites worldwide using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton backpropagation algorithm and six-fold cross validation. Extensive sensitivity tests showed that an ANN with six input variables and one hidden layer gave results comparable to (and in some cases even slightly better than) the standard method; moreover, this ANN performed considerably better than a multivariate linear regression model. Similar satisfying results were obtained when the ANN routine was implemented in a one-dimensional stand alone land surface model (LSM), opening the way to implementation in three-dimensional climate models. In the case of the one-dimensional LSM, no CPU time was saved when using the ANN version, since the small time step of the standard version required only one iteration in most cases. This could be different in models with longer time steps, e.g. global climate models.

1 Introduction

The turbulent fluxes of momentum, heat, water vapour and trace gases link the atmosphere with the Earth's surface. The faithful representation of these fluxes is therefore essential for a proper functioning of climate and weather forecast models. In these models, the fluxes are parameterised as momentum flux $\tau = \rho u_*^2$ and heat flux $H = -\rho c_p u_* \theta_*$ (ρ is air density, c_p is air heat capacity), using a velocity scale u_* and a (potential) temperature scale θ_*. u_* and θ_* depend on near surface wind and temperature, their gradients, surface roughness and atmospheric stability. In the framework of the almost exclusively used Monin-Obukhov similarity theory (MOST; Monin and Obukhov, 1954), one has to determine stability functions for momentum and heat which depend on a single stability parameter (for details, see e.g. Arya, 2001). These stability functions must be determined empirically and were obtained by different authors from regressions on observations from a small number of field experiments. As shown in Högström (1996), the results vary considerably, especially in the very stable and the very unstable regimes, due to a lack and/or a large scatter of the observations and possibly violations of the assumptions of MOST. Furthermore, the underlying nonlinear equations must be solved iteratively at each time step of a model run which can be time consuming.

In the present study, artificial neural networks (ANN) and their ability to simulate a wide range of relationships between input and output variables as a universal approximator (Hornik et al., 1989) are used to model the stability functions. Our goals in this study are a) to see how well ANNs can approximate the stability relationships, b) possibly increase accuracy through using larger data sets, c) using the more flexible ANN approach instead of function fitting, and d) possibly speed up the calculations. With positive outcomes, we ultimately want to replace the relevant subroutines in a climate model by ANNs in order to improve overall model performance.

A good overview of various applications of ANNs in different disciplines can be found in Zhang (2008). Several studies (e.g. Gardner and Dorling, 1999; Elkamel et al., 2001; Kolehmainen et al., 2001) describe applications of ANNs to meteorological and air quality problems. In these studies, long time series of observational data are available for ANN training and only one station is involved in the training and validation process. Comrie (1997) compares ozone forecasts using ANNs with forecasts using standard linear regression models and finds that ANNs are “somewhat, but not overwhelmingly“ better than the regression models. Best performance is obtained with an ANN incorporating time lagged data. Gomez-Sanchis et al. (2006) use a multi-layer perceptron (MLP) to predict ozone concentrations near Valencia based on meteorological and traffic information. Different model architectures are tested and good agreement with observations is found. However, for different years different model architectures for optimal results are required, which they attribute to varying relative importance of the input variables. Elkamel et al. (2001) use a one hidden layer ANN and meteorological and precursor concentrations to predict ozone levels in Kuwait. They find that the ANN gives consistently better predictions than both linear and nonlinear (log output) multivariate regression models. Kolehmainen et al. (2001) compare the ability of self-organising maps and MLP to predict NO_2 concentrations when combined with different methods to preprocess the data. They find that direct application of the MLP gives best results. In all these studies just one hidden layer is sufficient and it is pointed out that careful selection of the input data is crucial.

Some papers deal with the idea of replacing whole models or model components by ANNs. For example, Knutti et al. (2003) teach a neural network to simulate certain output variables of a global climate model and use the result to establish probability density functions as well as to enlarge a global climate model ensemble considerably. Gentine et al. (2018) use an ANN to parameterise the effects of subgrid scale convection in a global climate model. The ANN learns the combined effects of turbulence, radiation and cloud microphysics from a convection resolving submodel. They find that using the ANN, many of these processes can be predicted skilfully, but spatial variability is reduced compared to the original climate model; they attribute this to chaotic dynamics accounted for in the original model, but not in the version using the ANN which is deterministic by construction. Sarghini et al. (2003) and Vollant et al. (2017) use an ANN trained with direct numerical simulation data as a subgrid scale model in a large eddy simulation model. Sarghini et al. (2003) find that the ANN is able to reproduce the nonlinear behaviour of the turbulent flows, whereas Vollant et al. (2017) find that the ANN performs well for the flow cases the ANN was trained for, but that it can fail for other flow configurations.

This paper is structured as follows: in Sect. 2, we give a short overview over Monin-Obuhkov similarity theory and artificial neural networks, introduce cross-validation, present the data used (including important quality checks) and describe our strategy to find the best network. Thereafter, trained ANNs (which are in fact MLPs, but we will stick to the generic name ANN here) are validated and results are discussed (Sect. 3). In Sect. 4, the best performing ANN is implemented in a one-dimensional land surface model (LSM) and results are compared with the ones of the standard version. A summary is given in Sect. 5.

2 Methods and data

2.1 Monin-Obukhov similarity theory (MOST)

In weather forecast and climate models, the turbulent fluxes of momentum, heat, water vapour and trace gases between the Earth's (land and water) surface and the atmosphere are usually calculated on the basis of Monin-Obukhov similarity theory (MOST, Monin and Obukhov (1954)). We give here a very brief survey over MOST, focussing on momentum and heat fluxes; details can be found in Arya (2001). The main assumptions of MOST are: horizontally homogeneous terrain (in particular, flow characteristics are independent of wind direction), stationarity, fair (i.e. dry) weather conditions and small terrain roughness, i.e. no or low vegetation; for tall vegetation, the last assumption is usually circumvented by introducing the concept of displacement height $d \approx 0.67z$. MOST postulates that turbulence in the surface (also called Prandtl or constant flux) layer depends only on four quantities: the height above ground z resp. (tall) canopy $z - d$, a velocity scale u_*, a temperature scale θ_* and a buoyancy term g/θ, where g is gravitational acceleration and θ denotes potential temperature. The velocity and temperature scales depend on velocity resp. temperature gradients and on atmospheric stability, and this dependence will be used later to build the neural networks. According to the Buckingham Pi-Theorem, these four quantities based on length, time and temperature can be combined to a single non-dimensional quantity $\zeta = z/L$, where $L = u_*^2\theta/(\kappa g\theta_*)$ is the Obukhov length and $\kappa \approx 0.40$ is the von Kármán constant; other dimensionless quantities like dimensionless wind and temperature gradients can be expressed as functions of ζ. The Obukhov length L measures the stratification of the surface layer: large (positive or negative) values (i.e. $\zeta \approx \pm 0$) indicate neutral stratification, positive values indicate stable stratification, negative values indicate unstable stratification. Since momentum flux is expressed as $\tau = \rho u_*^2$, and heat flux as $H = -\rho c_p u_* \theta_*$ (ρ is air density, c_p is air heat capacity), our goal is to determine u_* and θ_* from known quantities, which are in our case modelled or observed wind and temperature gradients in the surface layer. Non-dimensional wind shear ϕ_m and the non-dimensional gradient of the potential temperature ϕ_h (also called stability functions) can be written as

$$\phi_m(\zeta) = \frac{\kappa z}{u_*}\frac{\partial u}{\partial z}, \quad \phi_h(\zeta) = \frac{\kappa z}{\theta_*}\frac{\partial \theta}{\partial z} \tag{1}$$

respectively, where u is the mean wind speed at height z.

The 'universal' functions ϕ_m and ϕ_h can be obtained from simultaneously measured values of the wind speed and temperature gradients and the momentum and heat fluxes (providing u_* and θ_*). Conversely, u_* and θ_* can be calculated from these universal functions, given the wind speed and temperature gradients; this is how these functions are used in weather and climate models. Data from field experiments, notably the Kansas experiment in 1968, have been used to derive these universal functions by Businger et al. (1971). Generally, the stability functions thus obtained have the form

$$\phi_{m,h}(\zeta) = (\alpha_{m,h} + \beta_{m,h}\zeta)^{\gamma_{m,h}} \tag{2}$$

with the coefficients depending on $\zeta > 0$ or $\zeta \leq 0$. An overview of these functions can be found in Högström (1988); it is shown there that there is considerable scatter in the data (especially under very stable and very unstable conditions) and, as a result, also in the derived universal functions.

In applications, differences are known rather than gradients. Integrating the functions (1) between a reference height z_r and z yields

$$\kappa(u(z) - u(z_r))/u_* = ln(z/z_r) - \Psi_m(z/L), \quad \kappa(\theta(z) - \theta(z_r))/\theta_* = ln(z/z_r) - \Psi_h(z/L) \tag{3}$$

where

$$\Psi_{m,h}(z/L) = \int\limits_{z_r/L}^{z/L} (1 - \phi_{m,h}(x))dx/x \tag{4}$$

For the purpose of climate modelling, i.e. obtaining fluxes from simulated wind and temperature profiles, u_* and θ_* need to be derived from wind resp. temperature data at two heights and Eq. (1) or Eq. (3). Since ζ itself depends on u_* and θ_*, this amounts to solving a system of two nonlinear equations; we will call this traditional method the MOST method.

2.2 Neural networks

We describe here only those aspects of neural networks which are relevant to our study; for more information on neural networks, the reader is referred the literature, e.g. Rojas (2013); Kruse et al. (2016). Neural networks, or more precisely artificial neural networks (ANN), are a widely used technique to solve classification and regression problems as well as to analyse time series (Zhang, 2008). The building blocks of an ANN are the so-called neurons, arranged in different layers. An ANN has at least an input and an output layer; between these, there can be so-called hidden layers. The neurons in successive layers (but not within the same layer) are connected through weights (see Fig. 7). A neuron processes input data as follows:

$$o_j = f\left(\sum_i^N o_i \cdot w_{ij}\right), \tag{5}$$

where o_j is the output of the neuron j, N is the number of neurons in the preceding layer (including the bias neuron, see below), o_i is the output of the ith neuron in the preceding layer and w_{ij} is corresponding weight. Nonlinear behavior of the network is induced by using nonlinear activation functions f. Each neuron belongs to a unique layer in a directed graph. Here, we use so-called multi-layer perceptrons (MLP), also known as feed-forward networks due to the unidirectional information flow. Each MLP consists of an input, an output and at least one hidden layer with an arbitrary number of neurons. The input layer takes (normalised) input data and the output data returns the (also normalised) results of the MLP. Normalisation is essential for equal weighting of the input and for consistency with the domain and range of the activation functions. Input information is propagated from layer to layer while each neuron responds to the signal. Bias neurons are used to adjust the activation level. All free parameters (i.e. weights) of a MLP need to be determined by a training process. In the case of supervised learning, the MLP knows its deviation from target values at every time and an error can be calculated using this deviation (Zhang, 2008). The aim of the training is to minimise an error metric by adjusting the network's weights. Here we use the mean squared error (MSE)

$$MSE = \frac{1}{|P|}\sum_{p\in P}\frac{1}{N^{\Omega}}\sum_{j}^{N^{\Omega}}\left(t_{jp}-o_{jp}\right)^2 \tag{6}$$

P is the total number of data points, N^{Ω} is the number of neurons in the output layer, t_{jp} is the target value of data point p and o_{jp} is the output of the MLP for data point p. In the study described here, we use a MLP with tangens hyperbolicus as activation functions in the hidden layer(s) (here one or two) and linear functions in the output layer trained by the Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton backpropagation algorithm (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970).

2.3 Regression model

To compare the performance of neural networks described in Sect. 2.2 with a standard regression model, we used the multivariate linear regression (MLR) model as implemented in the mvregress MATLAB routine

$$y_j = \sum_i \beta_{ij}\cdot x_i + \epsilon_j \tag{7}$$

where the β_{ij} are the regression coefficients and the ϵ_j are the residual errors with a multivariate normal distribution. The model uses a multivariate normal maximum likelihood estimation. The resulting values for β_{ij} maximise the log-likelihood function $\log L(\beta,\epsilon|y,x)$. We used the same six element input vector and two element target vector as for the ANN (both described in Sect. 2.6), as well as the same training and independent test data (DE-Keh) sets (see Sect. 2.4 and Sect. 2.5).

2.4 Data

To train and validate the neural network, data from 20 meteorological towers in Europe, Brazil and Russia over different land use types including forest, grassland and crop fields were collected. All data were measured after year 2000 and observation periods range from a few months to several years. Figure 1 shows a map of the sites which provided data. Stations varied widely in environmental surrounding, instrumental set-up and measurement heights. The tower configuration of the sites is shown schematically in Fig. 2. For our purposes, we required temperatures and wind speed in two measurement heights as well as the momentum and sensible heat fluxes to calculate the scaling quantities u_* and θ_* (see Sect. 2.6). The fluxes at the sites used were all measured by the eddy covariance method. If not available, density was calculated from the ideal gas equation using virtual temperature in case of available humidity data, otherwise we used the temperature of dry air. For forests, all observations had to be above the canopy, and all vertical distances were reduced by the displacement height, assumed to be two-thirds of the canopy height. The original temporal resolution of the data was either 10 minutes or 30 minutes; these were aggregated to 1-hour averages.

An important step before using data as input for the ANN was to check if the data were compatible with Monin-Obuhkov theory, i.e. if an (at least approximate) functional relationship between ζ and the right-hand sides of Eq. (1) was present and if yes, how well they were represented by the universal stability functions Eq. (1). It turned out that for some sites, no relationship existed. Reasons for this could be a violation of the assumptions of the Monin-Obuhkov theory like inhomogeneous terrain around the site or wind direction dependence of the roughness length. Data from these sites were not used further, except for the DE-Tha site (see Sect. 4). The remaining stations (see Table 1) with about 113,500 hourly averaged data points in total (see Table 2) were used to train and validate the networks. For these, agreement was generally better for temperature than for wind; also, agreement was better for unstable than for stable stratification, an observation which is often mentioned in the literature.

Data were preprocessed before they were presented to the ANN. Input and output data were normalised according to their extrema to the interval $[0,1]$ (see Table 3). Furthermore, weak wind situations with wind speeds below $0.3\ \mathrm{ms}^{-1}$ were filtered out. Because of large scatter of wind and temperature gradients under atmospheric conditions with absolute heat fluxes below $10\ \mathrm{Wm}^{-2}$ or small scaling wind speeds ($u_* < 0.1\ \mathrm{ms}^{-1}$), such data were excluded. Finally, the signs of the temperature scale θ_* and of the potential temperature gradient had to be the same, thus excluding counter-gradient fluxes which can be observed over forest (Denmead and Bradley, 1985) and ice (Sodemann and Foken, 2005), but violate the assumptions of MOST (Foken, 2017a, b).

2.5 Cross-validation and generalisation

Trained networks were validated using k-fold cross-validation (Kohavi, 1995; Andersen and Martinez, 1999) to prevent overfitting (Domingos, 2012). Overfitting originates from the trade-off to minimise the error on given data and to maximise performance on new unknown data (Chicco, 2017). In a first experiment, the full data set is divided into $k = 6$ subsets by a random data split with approximately equal size first. Cyclically, one subset is kept for independent testing, the remaining $k-1$ subsets are used for training and validation. With this experiment, we can show that ANNs are able to learn from the data and to represent their characteristics. In a second experiment, we go one step further and check if the found ANNs can handle not only unknown data but also completely new stations not used previously, i.e. if they are able to generalise. For this experiment, we decided to validate trained models with the station NL-Cab and to test the best ANNs finally on the station DE-Keh which had been left out in the training and validation phases of this experiment (see details on stations in Sect. 2.4). For these two stations, the MOST method performed best; thus, they present a strong challenge for the ANNs to achieve similar quality.

2.6 ANN setup and selection of best ANN

Neural networks are very flexible in terms of number of layers, number of nodes, error metrics, training method, activation function etc.; thus, a series of sensitivity runs was performed, which always consisted of a training and a validation phase. To find an optimal network architecture, we varied the following parameters:

- The number and type of input variables
- The number of hidden layers (one or two)
- The total number of nodes in the hidden layer(s) (between 1 and 14)

To avoid an excessive number of sensitivity runs, the parameters listed in Table 3 were kept fixed based on recommendations in the literature (Zhang, 2008; Kruse et al., 2016). Training is done in batch mode, therefore the network's weights are adjusted after each epoch. Training ended at most after 1000 epochs or if the error on the validation data increased for 50 successive epochs (early stopping). In the latter case, the state of the trained network with the lowest error on the validation data (and not the early stopping state) was set as final state. We tested network architectures with six and seven element input vectors. The six element input vector consisted of the wind speed and potential temperature averages over the two heights, the vertical gradients of wind and potential temperature and their ratio and a classifier to distinguish between low $(c_{veg} = 0)$ and tall $(c_{veg} = 1)$ vegetation. For the seven element input vector, we replaced the temperature gradient by its absolute value and added an additional input node describing the sign of potential temperature gradient. The target vector remained in both cases a two element vector consisting of the wind scale u_* and the temperature scale θ_*. As mentioned above, we experimented with ANNs having one and two hidden layers. For the ANNs with one hidden layer, we varied the number of neurons in the hidden layer from one to twice the size of the input layer. For ANNs with two hidden layers, the number of neurons in each layer is increased up to the number of input neurons.

All networks were trained to minimise the overall (sum of u_* and θ_*) MSE on normalised data from Eq. (6). To compare the different ANNs, we used: the root mean squared error (RMSE) $RMSE = \sqrt{MSE}$, the mean absolute error (MAE)

$$MAE = \frac{1}{|P|} \sum_{p \in P} \frac{1}{N^{\Omega}} \sum_{j=1}^{N^{\Omega}} \left| t_{jp} - y_{jp} \right| \tag{8}$$

and Pearson's correlation coefficient r

$$r = \frac{1}{N^{\Omega}} \sum_{j=1}^{N^{\Omega}} \frac{\sum_p (y_{jp} - \bar{y}_j)(t_{jp} - \bar{t}_j)}{\sqrt{\sum_p (y_{jp} - \bar{y}_j)^2} \cdot \sqrt{\sum_p (t_{jp} - \bar{t}_j)^2}} \quad \in [-1, 1], \tag{9}$$

where $\bar{y}_j$ and $\bar{t}_j$ are the averages of the jth net output and the target value with $\bar{y}_j = \frac{1}{|P|} \sum_p y_{jp}$ and $\bar{t}_j = \frac{1}{|P|} \sum_p t_{jp}$.

When ANNs are to be used in climate models, one has to find a trade-off between two aspects: on the one hand, the model should perform well according to the quality metrics described above; on the other hand, a superior model in terms of small errors but with higher computational demands may not be the best choice to use in climate models where saving computing time is a very high priority criterion. For ANNs, computing time normally increases with complexity of a network, i.e. with its size. We therefore tested also ANNs with smaller-than-optimal numbers of neurons in view of this trade-off. To find smaller networks requiring possibly less computing time, we looked at networks that meet the requirement that the size of each hidden layer n_{h_i} is less or equal to the size of the input layer n_I minus 1.

$$n_I - 1 \geq n_{h_1} \left(\geq n_{h_2} \right) \tag{10}$$

This condition was found after some experimenting and is somewhat arbitrary, but there is no hard rule defining the simplicity of a model. We will call ANNs that satisfy this condition ’simple networks’.

3 Results

As described in Sect. 2.5, ANNs are always trained on the training data set only and validated on a disjoint validation data set. If the MSE on the validation set rises continuously, training is stopped to prevent overfitting (early stopping). After this training and validation stage, the ability of the thus found ANNs to generalise is tested on data completely new to the ANNs. All in all, more than 100,000 networks were trained and tested this way.

3.1 Effect of data splitting

The validation results from ANNs with six inputs and one single hidden layer trained under six-fold cross-validation with random data splitting are shown in the box-and-whiskers plot in Fig. 3 as a function of the number of hidden neurons. One can see that the validation MSE decreases with increasing number of hidden neurons and reaches an asymptotic value of about 0.008 already with 6 to 7 neurons. Furthermore, the scatter of MSE is quite small, meaning that the quality of the results of ANNs trained on different sets varies only slightly.

If the training data are not split randomly but stationwise, a larger MSE and a considerably larger scatter of MSE results. Comparing Fig. 4 with Fig. 3 shows that MSE is roughly doubling, whereas scatter increases by about a factor of ten, almost independent of the network architecture. On the other hand, increasing the network size doesn’t necessarily imply a lower MSE. Using two hidden layers reduces slightly the median and error minimum, but increases the MSE spread, too. Comparison of Fig. 3 with Fig. 4 also shows that the stationwise error minima are comparable to the ones obtained from random data split. In both types of validation, ANNs with one and two hidden layers are not significantly different.
All in all, comparing Fig. 3 with Fig. 4 shows that the stationwise data split reduces the ANN performance substantially. This implies that using not enough stations as well as stationwise training impairs the generalisation of learned relationships between inputs and target values. Among the reasons for this could be the tendency of the ANNs to overfit training data by memorising relationships and local effects contaminating the validity of MOST, like not ideal positioning of sites or not ideal atmospheric conditions. These findings confirm the need for independent testing with data yet unknown to the ANN in order to estimate the ANNs real ability to generalise. This will be discussed in the next section.

3.2 Generalisation to unknown data

After having shown that ANNs are able to extract u_* and θ_* from training data successfully, our next step is to assess how the ANNs found in the previous section can handle input from stations which weren't used neither for training nor for validation, i.e. data completely unknown to them; this simulates the situations where ANNs are used in climate models (where grid points play the role of stations). To test this, we choose the station NL-Cab for validation and DE-Keh as the unknown station. We selected these two stations because the MOST method performed best for these stations and it is therefore a strong challenge for the ANNs to produce equivalent results. The results of the networks performing best on the validation set are summarised in Table 4, where we compare the ANNs according to increasing complexity of their net architecture. For comparison and in view of reducing CPU time, we show in this table also the results of the best simple networks (as defined in Sect. 2.6). Table 4 shows that in terms of MSE and correlation coefficient r, all ANNs perform better than the MOST method on the validation data set (NL-Cab). Applying these ANNs to the test data set DE-Keh results in an increased MSE and lower correlation coefficient, whereas the MOST method performs better on the test data set. Among the ANNs, the 6-5-3-2 ANN reached the best test performance with an MSE of $0.68 \cdot 10^{-2}$, but the simpler 6-3-2 ANN is second best (also in terms of MSE); thus, simple networks can be almost as good as larger networks. Networks with seven inputs have in our case no substantial advantage over networks with six inputs. ANNs with two hidden layers perform slightly better on the test data than ANNs with a single hidden layer. The overall correlation between network outputs and target values is in all cases quite high $(r \geq 0.85)$.

We also did a comparison for the turbulent momentum and heat fluxes $\tau = \rho u_*^2$ and $H = -\rho c_p u_* \theta_*$, which are the quantities ultimately needed in climate simulations. Results for the momentum and heat fluxes of three well-performing networks as well as for the MOST method are shown in Fig. 5 and Fig. 6 and in Table 5 and Table 6, respectively. In the tables we also show the results of the multivariate linear regression (MLR) described in Sect. 2.3. Both ANNs, MLR and the standard method tend to underestimate larger momentum fluxes, but differences among ANNs are quite small. Best agreement is achieved with the 6-5-3-2 ANN which is almost as good as the standard method.

Regarding the heat flux, the differences between the ANNs are again relatively small, but the ANNs as well as the standard method tend to overestimate the heat fluxes, whereas MLR understimates them (not shown). Best results are obtained with the 6-3-2 ANN. For heat flux, the 7-5-2-2 ANN behaves markedly different than the other ANNs. It produces two distinct states, one around -30 Wm^{-2} and the other from 50 Wm^{-2} to 200 Wm^{-2}; as a result, r is reduced but MAE is lowest for this 7-5-2-2 ANN. Thus, the 7-5-2-2 ANN acts more like a dichotomous classifier of stability rather than the continuous regression we are looking for. As for the momentum fluxes, the ANNs perform considerably better than the regression model. These results show again that smaller networks can be as good or even better than larger ones.

All ANNs perform considerably better than the multivariate linear regression model. This is not really surprising, since the scaling quantities to be approximated are nonlinear functions of stability (Arya, 2001), so that an ANN with a nonlinear activation function can be expected to perform better than any linear model; as Table 5 and Table 6 show, this is the case even for the small 6-3-2 ANN with one hidden layer.

A comparison of the CPU time required by the different ANNs relative to the 6-3-2 ANN is shown in Table 7. The table shows that the increase of computational demand is approximately proportional to the number of weights (as could be expected), and therefore increases considerably when two layer networks are used. As the discussion above shows, these costs are not reflected in a substantially higher quality of results.

We can conclude that generalisation entails a reduced performance of the ANNs with quite small differences between the various ANNs. The performance of the ANNs is comparable to the MOST method, and the simplest 6-3-2 network has the best score in terms of accuracy and computational efficiency.

4 Implementation of an ANN in a land surface model

As already mentioned, our goal is to replace the MOST method to calculate fluxes by an ANN in the land surface component of climate models, expecting more flexibility, accuracy and possibly saving of CPU time. The results presented in the previous section indicate that from the accuracy as well as computational efficiency point of view, the 6-3-2 ANN seems to be most suitable for implementation into a land surface model (LSM). This ANN is shown in Fig. 7.

We implemented the 6-3-2 ANN with weights as obtained in the previous sections in a stand-alone version of the one-dimensional LSM Veg3d (Braun and Schädler, 2005); this replaced the implemented routine using the MOST method to calculate the scaling quantities u_* and θ_*. We will call here the LSM version with the original MOST version the reference version. Input data for the ANN and data normalisation was the same as described in Sect. 2.4 and output was analogously de-normalised. Since the LSM requires apart from momentum and heat fluxes also the moisture flux, we calculated the scaling specific humidity q_* as proportional to θ_* following the standard procedure in boundary layer meteorology (Arya, 2001). Meteorological input for the LSM was 30 min values of short- and longwave radiation, wind speed, temperature, specific humidity and air pressure at two heights; additionally soil type and land use were prescribed; in the present study, these data were only available for the DE-Fal site for the year 2011 and for the DE-Tha site for the year 1998. For comparison with observations, time series of heat and moisture fluxes as well as soil temperature and soil moisture in the upper soil layers were available, so that the effect of the ANN on the soil component could also be assessed. We performed the comparison with data from the DE-Fal (grassland, year 2011) and DE-Tha (evergreen needleleaf forest, year 1998) stations for years which had not yet been used neither for training nor for validation; thus, the data were new to the ANN in the sense that time periods were used which had not been used previously for training and validation. The DE-Tha site had not been used at all before, because the other sites selected in Sect. 2.4 were more consistent with MO than DE-Tha and because the DE-Tha time series covered only one year. We compared the RMSE and the correlation coefficient of the calculated values with the observed ones for the reference version and the ANN version. Additionally, we compared the required CPU times. The results of the comparison are shown in Table 8 and 9.

Especially for grassland, results of the reference version are very good in terms of RMSE and correlation coefficients and it is difficult for the ANN version to outperform this. However, the results show that the ANN version is able to produce results of similar quality as the reference version for the fluxes as well as for soil temperature and soil moisture. For tall vegetation, RMSEs are larger and correlation is less; but the differences between the ANN version and the reference version are even smaller than for grassland and for soil moisture the ANN version even outperforms the reference version. In terms of fluxes, the reference version is generally slightly better. Regarding CPU time, there are only minor differences, although we had expected the ANN version to be faster. However, due to the small prognostic time step used, once initialised, the reference version does in most cases not need to do more than one iteration to find a solution to the nonlinear equation and to update the scaling quantities, so that the expensive iteration is reduced considerably. In summary, as a result of this first comparison one can say that the ANN version works equally well as the reference version.

5 Summary

We have used an ANN (more precisely, a MLP) to obtain the scaling quantities u_* and θ_* as defined in MOST; these are used in weather and climate models to calculate the turbulent fluxes of heat and momentum in the atmospheric surface layer. To train, validate and test the neural network, a large set of worldwide observations was used, representing tall vegetation (forests) and low vegetation (grassland, agricultural terrain). A quality assessment of the data sets showed that not all of them were compatible with MOST, so only 7 of the initially 20 data sets could be used.

Sensitivity studies were performed with different sets of input parameters, data sampling methods and network architectures; validation was done with 6-fold cross validation. An important part of the overall network validation was to check the ability of the network to generalise, i.e. to produce acceptable output if input is data from stations completely unknown to the network. These studies showed that even a relatively small 6-3-2 network with six input parameters and one hidden layer yields satisfying results in terms of RMSE and correlation coefficient. With respect to the trade-off between quality of results and computational efficiency, this network performed best.

We could show that results of the ANN were equivalent to the standard method in all tests we performed. A final validation with the heat and momentum fluxes instead of the scaling quantities showed that the MOST method and the ANN approach were also in this case almost equal in terms of quality, with the 6-3-2 ANN performing best. Furthermore, we could show that the ANNs outperform a multivariate linear regression model with the same input and output variables and training and test data. This could be expected, since the stability functions are nonlinear functions, so that even a small ANN with one hidden layer and a nonlinear activation function can be expected to perform better than any linear model. An implementation of the 6-3-2 ANN into an existing LSM showed that the ANN version gives results equivalent to the standard implementation with sometimes even higher correlations. However, no saving of CPU time was found.

In summary, it could be shown that even in this stage, an ANN gives results comparable in quality to the MOST method. Some obvious improvements will include more and better differentiated land use classes (e.g. water, urban areas) and including more situations of strong stratification. Next steps will include more experiments with the input parameters (e.g. including a time lag) and some fine tuning to improve the computational efficiency (e.g. using different activation functions). We intend to implement and test the neural network routine in a three-dimensional regional climate model (RCM). We expect to save about five percent of CPU time, taking account of parallelisation. This may not seem much, but RCMs in particular are very expensive to run (climatologically relevant multidecadal simulations at high resolution can take several tens of weeks on a high performance system), so every saving counts. The implementation will require the ANN to learn some additional land use types like urban areas or water surfaces. If these tests are positive, this would open the possibility to replace other “uncertain” components of climate models (e.g. cloud microphysics, sea ice) by neural network subroutines, similar to the work described in Sarghini et al. (2003) and Vollant et al. (2017), thus gaining flexibility and saving CPU time. The main hindrance to do that is presently the lack of suitable training and validation data. An alternative to ’real’ data could be to use data from more detailed models like LES or urban climate models.

Code availability. A MATLAB script (run.m) running the 6-3-2-net with a sample dataset (DE-KaN.dat) can be found under http://doi.org/10.23728/b2share.36ef510c515c4a00bb963113647e44a9.

Data availability. The data for this study have been obtained from the sources mentioned in the acknowledgements.

Appendix A

Author contributions. LL was responsible for data collection, quality checks and data preprocessing. Also, he trained, validated and generalised the ANNs, performed the MLR analysis and made the comparisons with the MOST method. GS implemented the 6-3-2 network in a land surface model and carried out performance measurements in terms of result quality and CPU time. GS prepared the manuscript with contributions from LL.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. The authors would like to thank following persons and institutions for providing station data: Martin Kohler and Rainer Steinbrecher (Karlsruhe Institute for Technology), Mathias Göckede, Olaf Kolle and Fanny Kittler (Max Planck Institute for Biogeochemistry Jena), Frank Beyrich (German Weather Service), Ingo Lange (University of Hamburg), Clemens Drüe (University of Trier), Marius Schmidt (Forschungszentrum Jülich) and Thomas Grünwald (TU Dresden). In addition, data from following sources were collected and used: Integrated Carbon Observation System Sweden (ICOS), Cabauw Experimental Site for Atmospheric Research (CESAR) database, Oak Ridge National Laboratory Distributed Active Archive Center (ORNL DAAC) and University Corporation for Atmospheric Research (UCAR). We thank the two anonymous reviewers and the editor for their helpful comments and suggestions. Finally, we acknowledge support by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

References

Andersen, T. and Martinez, T.: Cross validation and MLP architecture selection, in: Neural Networks, 1999. IJCNN'99. International Joint Conference on, vol. 3, pp. 1614–1619, IEEE, 1999.

Arya, P. S.: Introduction to micrometeorology, Int. Geophys. Series, San Diego, Calif., Academic Press, 79, 2001.

Braun, F. and Schädler, G.: Comparison of Soil Hydraulic Parameterizations for Mesoscale Meteorological Models., J. Appl. Meteorol., 44, 1116–1132, 2005.

Broyden, C. G.: The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations, IMA J. Appl. Math., 6, 76–90, https://doi.org/10.1093/imamat/6.1.76, 1970.

Businger, J. A., Wyngaard, J. C., Izumi, Y., and Bradley, E. F.: Flux-Profile Relationships in the Atmospheric Surface Layer, J. Atmos. Sci., 28, 181–189, https://doi.org/10.1175/1520-0469(1971)028<0181:FPRITA>2.0.CO;2, 1971.

Chicco, D.: Ten quick tips for machine learning in computational biology, BioData Mining, 10, 35, 2017.

Comrie, A. C.: Comparing neural networks and regression models for ozone forecasting, J. Air Waste Manag. Assoc., 47, 653–663, 1997.

Denmead, O. T. and Bradley, E. F.: Flux-Gradient Relationships in a Forest Canopy, in: The Forest-Atmosphere Interaction: Proceedings of the Forest Environmental Measurements Conference held at Oak Ridge, Tennessee, October 23–28, 1983, edited by Hutchison, B. A. and Hicks, B. B., pp. 421–442, Springer Netherlands, Dordrecht, 1985.

Domingos, P.: A few useful things to know about machine learning, Commun. of the ACM, 55, 78–87, 2012.

Elkamel, A., Abdul-Wahab, S., Bouhamra, W., and Alper, E.: Measurement and prediction of ozone levels around a heavily industrialized area: a neural network approach, Adv. Environ. Res., 5, 47–59, 2001.

Fletcher, R.: A new approach to variable metric algorithms, Comput. J., 13, 317–322, 1970.

Foken, T.: Micrometeorology, SpringerLink : Bücher, Springer, Berlin, Heidelberg, 2nd ed. 2017 edn., 2017a.

Foken, T.: Energy and Matter Fluxes of a Spruce Forest Ecosystem, vol. 229, Springer, 2017b.

Gardner, M. and Dorling, S.: Neural network modelling and prediction of hourly NOx and NO2 concentrations in urban air in London, Atmos. Environ., 33, 709–719, 1999.

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., and Yacalis, G.: Could machine learning break the convection parameterization deadlock?, Geophys. Res. Lett., 2018.

Goldfarb, D.: A family of variable-metric methods derived by variational means, Math. Comput., 24, 23–26, 1970.

Gomez-Sanchis, J., Martín-Guerrero, J. D., Soria-Olivas, E., Vila-Francés, J., Carrasco, J. L., and del Valle-Tascón, S.: Neural networks for analysing the relevance of input variables in the prediction of tropospheric ozone concentration, Atmos. Environ., 40, 6173–6180, 2006.

Högström, U.: Non-dimensional wind and temperature profiles in the atmospheric surface layer: A re-evaluation, Bound.-Lay. Meteorol., 42, 55–78, https://doi.org/10.1007/BF00119875, 1988.

Högström, U.: Review of some basic characteristics of the atmospheric surface layer, Bound.-Lay. Meteorol., 78, 215–246, https://doi.org/10.1007/BF00120937, 1996.

Hornik, K., Stinchcombe, M., and White, H.: Multilayer feedforward networks are universal approximators, Neural Netw., 2, 359–366, 1989.

Knutti, R., Stocker, T., Joos, F., and Plattner, G.-K.: Probabilistic climate change projections using neural networks, Clim. Dyn., 21, 257–272, 2003.

Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Ijcai, vol. 14, pp. 1137–1145, Montreal, Canada, 1995.

Kolehmainen, M., Martikainen, H., and Ruuskanen, J.: Neural networks and periodic components used in air quality forecasting, Atmos. Environ., 35, 815–825, 2001.

Kruse, R., Borgelt, C., Braune, C., Mostaghim, S., and Steinbrecher, M.: Computational intelligence: a methodological introduction, Springer, 2016.

Monin, A. and Obukhov, A.: Basic laws of turbulent mixing in the surface layer of the atmosphere, Contrib. Geophys. Inst. Acad. Sci. USSR, 151, e187, 1954.

Rojas, R.: Neural networks: a systematic introduction, Springer Science & Business Media, 2013.

Sarghini, F., de Felice, G., and Santini, S.: Neural networks based subgrid scale modeling in large eddy simulations, Computers & Fluids, 32, 97–108, 2003.

Shanno, D. F.: Conditioning of quasi-Newton methods for function minimization, Math. Comput., 24, 647–656, 1970.

Sodemann, H. and Foken, T.: Special characteristics of the temperature structure near the surface, Theor. Appl. Climatol., 80, 81–89, 2005.

Vollant, A., Balarac, G., and Corre, C.: Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures, Journal of Turbulence, 18, 854–878, https://doi.org/10.1080/14685248.2017.1334907, 2017.

Zhang, G. P.: Neural Networks For Data Mining, in: Soft Computing for Knowledge Discovery and Data Mining, edited by Maimon, O. and Rokach, L., chap. 21, pp. 17–44, Springer US, Boston, MA, https://doi.org/10.1007/978-0-387-69935-6_2, 2008.

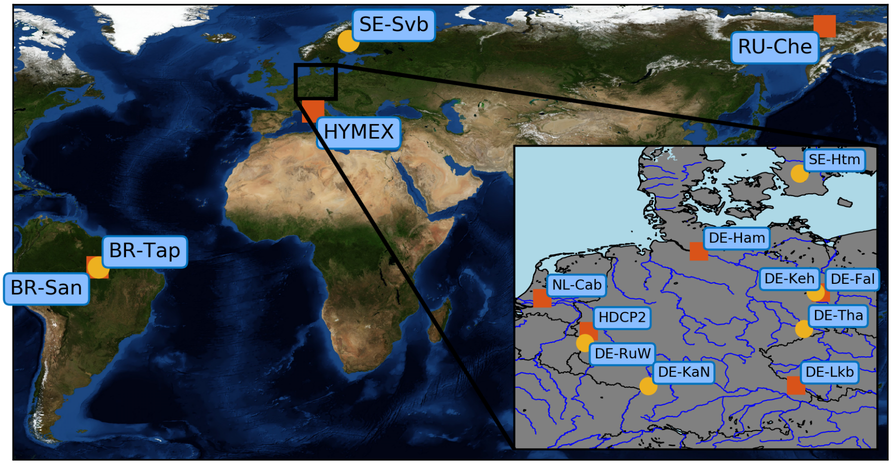

Figure 1. Location of the stations which provided data for this study. Station symbols are according to low (red square; grasslands, croplands, wetlands) and tall (yellow circle; forest) vegetation. HDCP2 includes stations DE-Nie07, DE-Nie13 and DE-Was06, HYMEX includes stations FR-CorX and FR-GiuX. Further information can be found in Table A1.

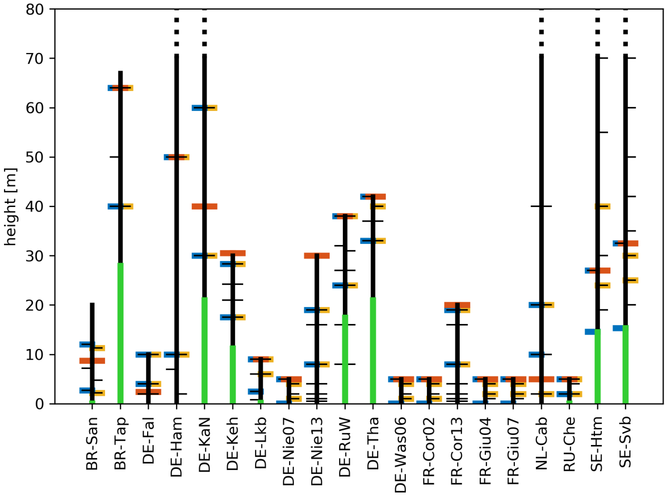

Figure 2. Schematic setup of the meteorological towers used for this study. Available measurements for wind velocity (black, left side arm) and temperature (black, right side arm) are shown as well as the finally used measurement height for wind (blue), temperature (yellow) and turbulent fluxes (red). Vegetation height is illustrated in green and towers with a total height above 80m are clipped.

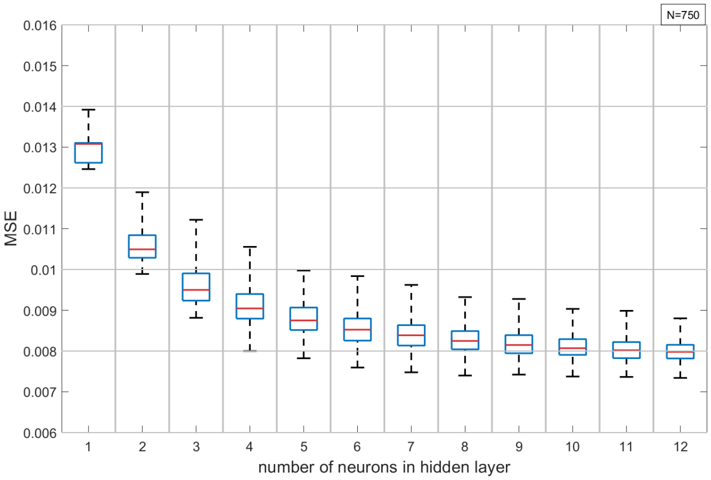

Figure 3. Network with six inputs and one hidden layer under six-fold cross-validation: MSE of the trained network on validation data set using random data split as a function of hidden layer size. Whiskers indicate interquartile range. Each box summarises results from 750 single networks.

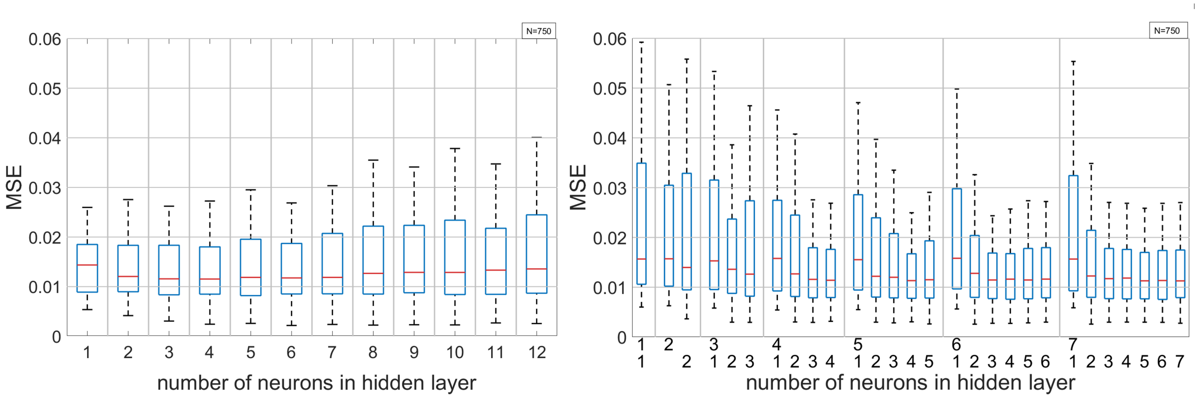

Figure 4. Validation MSE of trained networks using stationwise data split as a function of hidden layer size for a) the network with six inputs and one hidden layer (left), and b) the network with seven inputs and two hidden layers (right). Numbers at the bottom axis indicate the number of neurons in the first (top row) and second (bottom row) hidden layer. Values for the other networks considered are similar. Whiskers indicate the length of interquartile range and each box summarises results from 750 single networks.

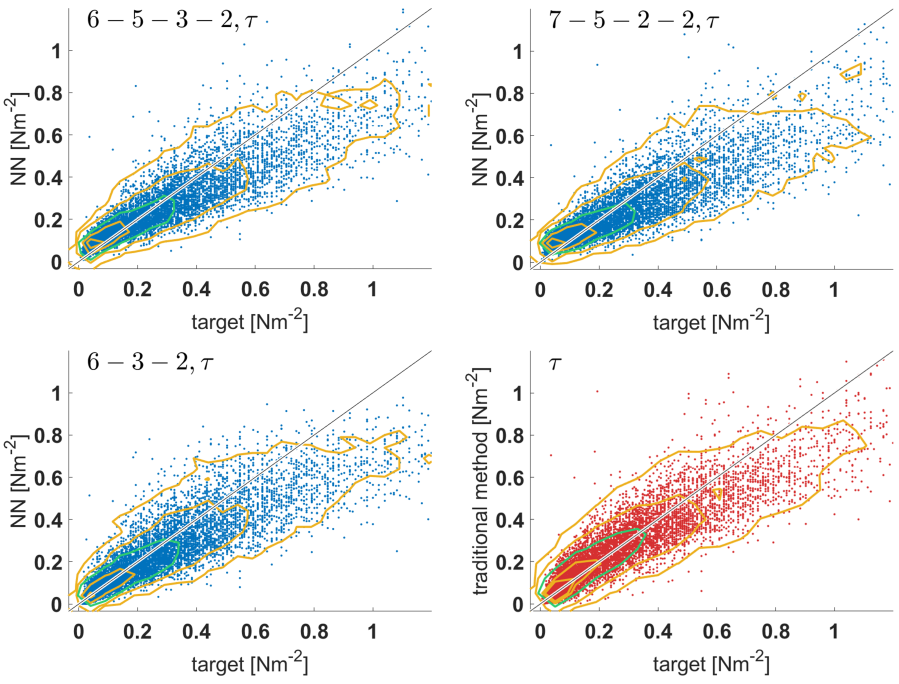

Figure 5. Plots of network output versus target values for momentum flux on unknown test data (DE-Keh). Clockwise from bottom left: 6-3-2 ANN, 6-5-3-2 ANN, 7-5-2-2 ANN, MOST method. Contoured are kernel density estimates of two-dimensional probability density distribution with the 95th, 75th, 25th and 5th percentiles (yellow line) starting outside and the 50th percentile (green).

Table 1. Station information for the meteorological towers selected for training and validation (see Sect. 2.4); a list of all stations is given in Table A1. Land usage classification follows the International Geosphere-Biosphere Programme (IGBP) standards: evergreen needleleaf forests (ENF), grasslands (GRA), permanent wetlands (WET) and croplands (CRO).

station	complete station name	lat [°]	lon [°]	height m a.s.l.	IGBP	tower height [m]
BR-San	Santarem Pasture Tower Site (Para, Brazil)	-3.02	-54.89	100	GRA/CRO	20
DE-Fal	Grenzschichtmessfeld Falkenberg	52.17	14.12	73	GRA	10
DE-KaN	KIT CN Messmast	49.09	8.43	110	ENF	200
DE-Keh	Messstation Forst Kehrigk	52.18	13.95	49	ENF	30
NL-Cab	CESAR observatory	51.97	4.93	-0.7	GRA	213
RU-Che	Cherksii Tower	68.61	161.34	6	WET	5
SE-Svb	Svartberget ICOS Sweden	64.25	19.77	270	ENF	150

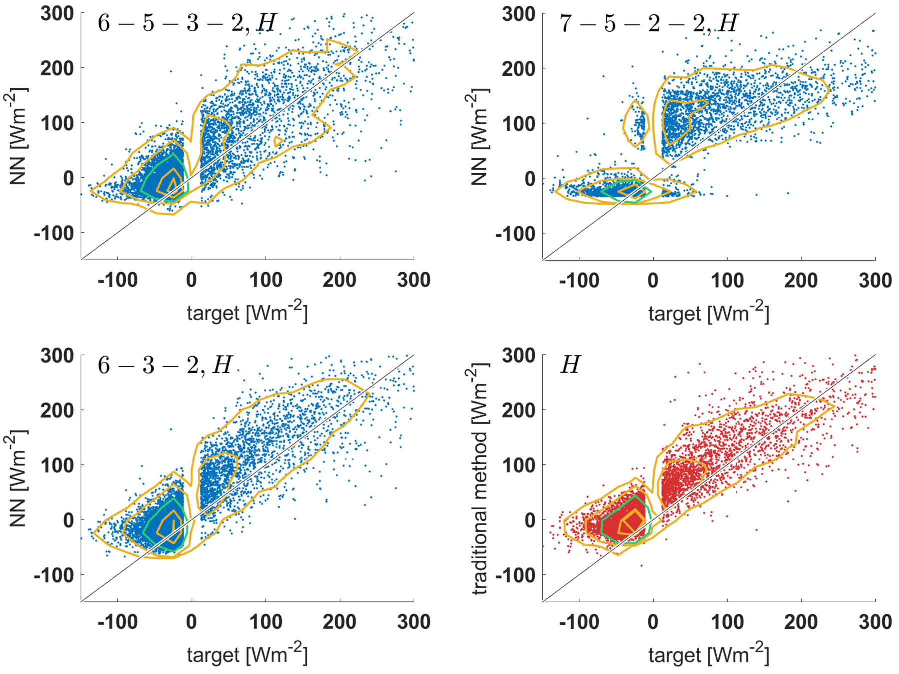

Figure 6. Plots of network output versus target values for heat flux on unknown test data (DE-Keh). Clockwise from bottom left: 6-3-2 ANN, 6-5-3-2 ANN, 7-5-2-2 ANN, MOST method. Contoured are kernel density estimates of two-dimensional probability density distribution with the 95th, 75th, 25th and 5th percentiles (yellow line) starting outside and the 50th percentile (green). The vertical gap is due to the exclusion of heat fluxes between $\pm 10\mathrm{Wm}^{-2}$.

Table 2. Time series information for the meteorological towers selected for training and validation. Count and availability are measured on an hourly interval and not on the original resolution of each time series.

station	from	to	availability	count
BR-San	2001-01-01	2005-09-22	62 %	25,503
DE-Fal	2008-01-01	2009-12-21	70 %	12,118
DE-KaN	2015-03-01	2016-12-30	78 %	12,541
DE-Keh	2008-01-01	2009-12-29	70 %	12,207
NL-Cab	2014-01-01	2017-11-30	94 %	32,337
RU-Che	2014-05-26	2016-10-14	40 %	8,283
SE-Svb	2015-01-18	2016-11-01	68 %	10,707
total				113,696

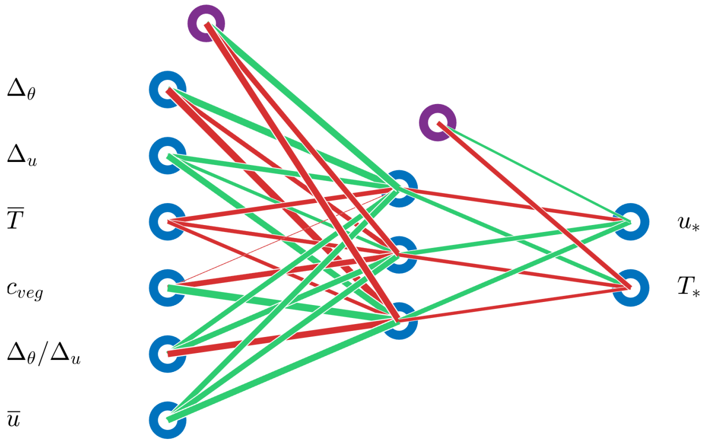

Figure 7. The architecture of the 6-3-2 ANN implemented in the land surface model. Input is described in Sect. 2.6. Purple circles are bias neurons.

Table 3. Fixed network parameters for training (after Zhang (2008); Kruse et al. (2016)).

normalisation	$\widetilde{x_i} = (x_i - \min_{Data}(x_i))/(\max_{Data}(x_i) - \min_{Data}(x_i))$
activation function	tangens hyperbolicus
activation function output	linear
training algorithm	BFGS quasi-Newton backpropagation
error metric	MSE
early stopping after ... epochs	50
maximum number of epochs	1000
training mode	batch

Table 4. Performance results of overall best and best simple networks. MSE and r are measured on normalised data and are non-dimensional. MSE_v and r_v are calculated on validation data and MSE_t and r_t on test data. Performance of MOST method (benchmark) is also shown.

condition	net structure	# weights	$\mathrm{MSE}_v[10^{-2}]$	r_v	$\mathrm{MSE}_t[10^{-2}]$	r_t
overall best net	6-5-2	47	0.17	0.94	0.90	0.89
	7-11-2	112	0.18	0.92	0.96	0.86
	6-5-3-2	61	0.20	0.93	0.68	0.88
	7-5-2-2	58	0.19	0.92	0.79	0.88
best simple net	6-3-2	29	0.38	0.92	0.74	0.87
	7-4-2	42	0.21	0.92	1.36	0.87
	6-3-3-2	41	0.27	0.91	0.84	0.85
	7-4-2-2	48	0.22	0.90	1.01	0.86
benchmark	-	-	0.92	0.85	0.58	0.92

Table 5. Performance of networks vs. multivariate linear regression (MLR) and MOST method (benchmark) for momentum flux at the DE-Keh site.

net structure	$\mathrm{MSE}[10^{-2}\mathrm{N}^2\mathrm{m}^{-4}]$	$\mathrm{RMSE}[\mathrm{Nm}^{-2}]$	$\mathrm{MAE}[\mathrm{Nm}^{-2}]$	r
6-5-3-2	2.11	0.15	0.09	0.90
7-5-2-2	2.44	0.16	0.10	0.89
6-3-2	2.56	0.16	0.09	0.87
MLR	5.81	0.24	0.17	0.89
benchmark	1.72	0.13	0.08	0.90

Table 6. Performance of networks vs. multivariate linear regression (MLR) and MOST method (benchmark) for heat flux at the DE-Keh site.

net structure	$\mathrm{MSE}[\mathrm{W}^2\mathrm{m}^{-4}]$	$\mathrm{RMSE}[\mathrm{Wm}^{-2}]$	$\mathrm{MAE}[\mathrm{Wm}^{-2}]$	r
6-5-3-2	2461	49.6	37.6	0.85
7-5-2-2	2329	48.3	31.4	0.82
6-3-2	2092	45.8	35.1	0.88
MLR	4447	66.7	53.3	0.65
benchmark	1915	43.8	34.4	0.90

Table 7. Relative computational demand of the ANNs discussed in the text.

net structure	no. of weights	CPU time (relative to 6-3-2 ANN)
6-3-2	29	1
6-5-2	47	1.6
7-11-2	112	3.7
6-5-3-2	61	2.5
7-5-2-2	58	2.4

Table 8. Comparison of the reference version with the ANN version of Veg3d for the DE-Fal grassland station. H denotes the heat flux, M is moisture flux, T_s is soil temperature, w_s is soil moisture.

	Ref	ANN
CPU time	10.83	10.65
RMSE H $[\mathrm{Wm}^{-2}]$	16.8	27.3
rH	0.87	0.81
RMSE M $[\mathrm{Wm}^{-2}]$	15.1	20.5
rM	0.91	0.86
RMSE T_s $[^\circ\mathrm{C}]$	0.8	1.3
rT_s	.99	.99
RMSE w_s $[\%]$	4.8	5.5
rw_s	0.87	0.89

Table 9. Same as above, but for forest station DE-Tha

	Ref	ANN
CPU time	95.47	97.74
RMSE H $[\mathrm{Wm}^{-2}]$	39.0	40.9
rH	0.52	0.57
RMSE M $[\mathrm{Wm}^{-2}]$	27.9	33.1
rM	0.78	0.71
RMSE T_s $[^\circ\mathrm{C}]$	2.4	2.2
rT_s	0.98	0.98
RMSE w_s $[\%]$	5.3	3.7
rw_s	0.53	0.75

Table A1. Station information for all collected meteorological towers. Land use classification follows the International Geosphere-Biosphere Programme (IGBP) standards: evergreen needleleaf forests (ENF), evergreen broadleaf forests (EBF), grasslands (GRA), permanent wetlands (WET) and croplands (CRO).

station	complete station name	lat [°]	lon [°]	height m a.s.l.	IGBP	tower height [m]
BR-San	Santarem Pasture Tower Site (Para, Brazil)	-3.02	-54.89	100	GRA/CRO	20
BR-Tap	Tapajos National Forest (Santarem, Para, Brazil)	-3.01	-54.58	100	EBF	67
DE-Fal	Grenzschichtmessfeld Falkenberg	52.17	14.12	73	GRA	10
DE-Ham	Wettermast Hamburg	53.52	10.10	0.3	GRA	300
DE-KaN	KIT CN Messmast	49.09	8.43	110	ENF	200
DE-Keh	Messstation Forst Kehrigk	52.18	13.95	49	ENF	30
DE-Lkb	Lackenberg Messstation	49.10	13.30	1300	GRA	9
DE-Nie07	HDCP2 Flux Station 07 Hambach Niederzier	50.90	6.46	110	GRA	5
DE-Nie13	HDCP2 Tower 13 Hambach Niederzier	50.90	6.46	110	GRA	30
DE-RuW	Wüstebach	50.50	6.33	621	ENF	38
DE-Tha	Anchor Station Tharandt	50.96	13.57	380	ENF	42
DE-Was06	HDCP2 Flux Station 06 Wasserwerk	50.89	6.43	96	CRO	5
FR-Cor02	HYMEX Flux Station 02 Corte	43.30	9.17	369	GRA	5
FR-Cor13	HYMEX Tower 13 Corte	43.30	9.17	369	GRA	20
FR-Giu04	HYMEX Flux Station 04 San-Giuliano	42.27	9.52	39	GRA	5
FR-Giu07	HYMEX Flux Station 07 San-Giuliano	42.27	9.52	39	GRA	5
NL-Cab	CESAR observatory	51.97	4.93	-0.7	GRA	213
RU-Che	Cherksii Tower	68.61	161.34	6	WET	5
SE-Htm	Hyltemossa ICOS Sweden	56.10	13.42	115	ENF	150
SE-Svb	Svartberget ICOS Sweden	64.25	19.77	270	ENF	150