FEB 2018

Maastricht Upper Area Control Centre

Predicting flight routes with a Deep Neural Network in the operational Air Traffic Flow and Capacity Management system

Trajectory prediction is an essential component of Air Traffic Management (ATM) systems but is hampered by route uncertainty because of future air traffic controller clearances. By augmenting traditional trajectory prediction logic with machine learning, a considerable improvement to accuracy can be achieved.

A deep neural network is trained on historical trajectories and a set of predictors. The neural network predicts the most likely route through the airspace, and has some ability to generalise to flights and conditions not seen before. Through iterative training on newly recorded data, the neural network can keep up with changes.

The neural network has been integrated into the operational Air Traffic Flow and Capacity Management system. Through the use of 'what-if?' trajectories, the new approach enhances existing capabilities rather than replacing them. This way, strengths are combined, paving the way for a gradual increase in the role of machine learning.

The problem to be solved

Flight trajectory prediction underpins much of the functionality of air traffic management systems, both in the tactical (air traffic control) and pre-tactical (air traffic flow & capacity management) phases of a flight. Systems in use today generally apply predefined rules and models to predict trajectories from available input data. Prediction logic is static and is grounded on domain knowledge of human experts and kinematic equations.

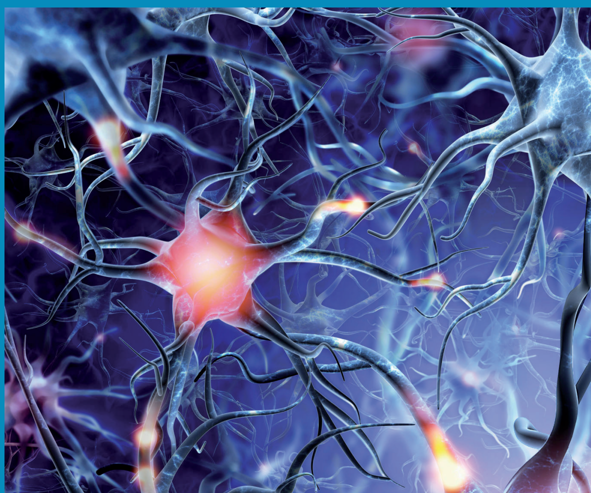
Accuracy of the predicted trajectories is far from perfect, degrading performance of the ATM system.

A problematic element at Maastricht UAC (MUAC) is route uncertainty. Flights do not conform to the route in the filed flight plan because air traffic controllers give permission to fly shorter routes. The instructions originate from controllers in the local centre or colleagues in upstream centres. They are driven by a multitude of factors and may change over time.

Machine learning is the science of giving computers the ability to learn from data, without being programmed explicitly.

Deep neural networks are inspired by the human brain, in the same way birds inspired us to build aircraft. They consist of layers of artificial neurons that are interconnected.

Deep neural networks power speech recognition systems (e.G. Apple siri), image recognition (e.G. Google images), learn to beat the world champion at the game of go by examining past games (deepmind), and will soon drive cars.



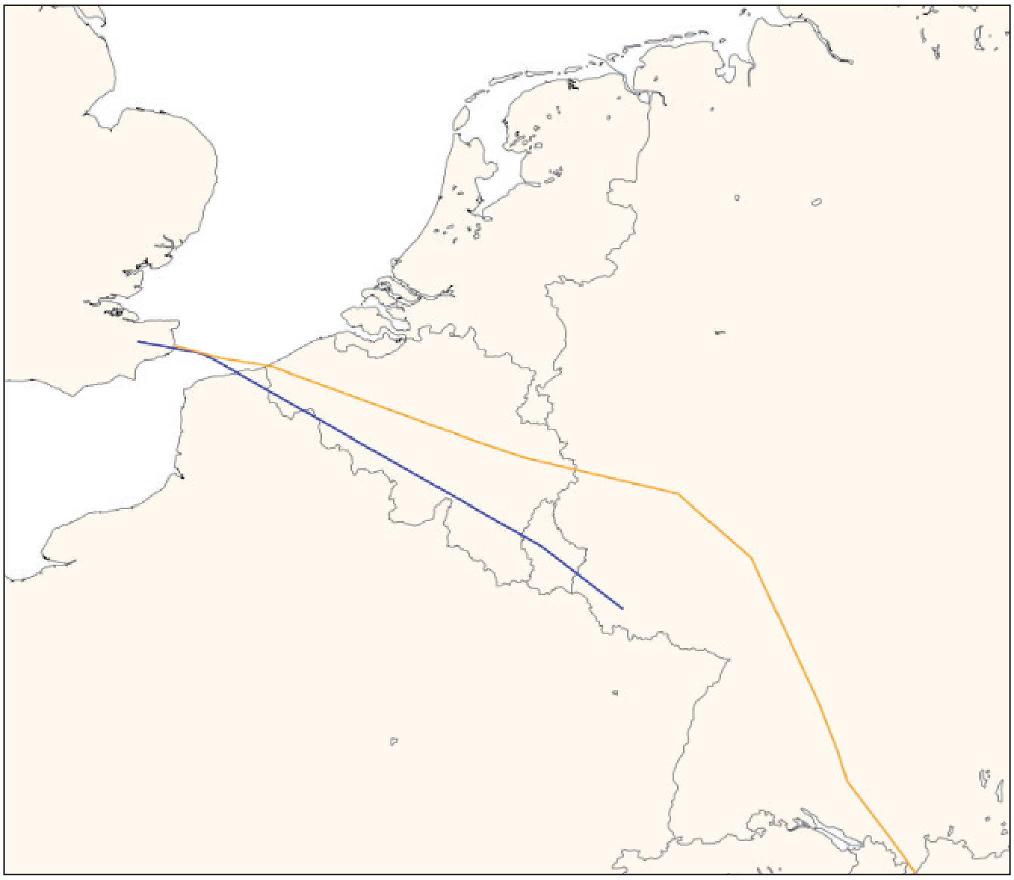


Figure 1 illustrates a typical case where the predicted trajectory deviates from the actual trajectory flown.

The line in yellowshows the initial predicted trajectory of a flight from London to Rome. The initial prediction is derived from the flight plan filed by the pilot[1] and is based entirely on fixed waypoints and routes.

The line in blue shows the route actually flown by the aircraft. Only the part relevant to MUAC is depicted. The flight enters MUAC airspace to the south of the expected entry point and is allowed to fly a direct route to a waypoint in the south of Germany.

The route deviation triggers several problems. Different sectors are crossed, which invalidates workload planning. The deviating route decreases the accuracy of predicted hotspots and medium term conflicts. And because crossing times and exit point are different, there is also an impact for the downstream control centre.

Being able to predict these deviations sufficiently in advance would bring great benefits to trajectory prediction. The factors that drive air traffic controller decisions are complex and intertwined. Moreover, they depend on working procedures and habits that may change over time.

Because it is hard to set out concise rules describing actual routes flown, other approaches have been studied that build on recent advances with machine learning. The growing amount of recorded historical data is a key enabler.

Historical data used for training the neural network

Given that the concept of machine-learned trajectories is fundamentally different from the existing approach, the new functionality is initially restricted to a subset of the traffic. This allows the supervisors and flow managers to grasp the trajectory differences and get acquainted with the concept of predicting trajectories from past data rather than from the route filed by the pilot. The selected traffic covers flights from the UK to European destinations to the south and southeast. The traffic amounts to about 10% of all MUAC traffic and has been selected on the basis that it suffers greatly from route deviations due to the presence of military airspace.

Flight and airspace data, including actual trajectories observed from radar tracks, was taken from the period 15 December 2015 – 12 December 2017. The dataset includes more than 328,600 flights.

[1] The route in yellow is extracted from the EFD messages received from the Network Manager.

Predictors

Input to the neural network is:

- Entry coordination point (NCOP), Exit Coordination point (XCOP), After-Boundary Exit point (BPXXCOP), Entry flight level (NFL), Requested flight level (RFL) and Exit flight level (XFL).
- Departure and destination airport
- Day of the week
- The time of the day interval the flight is expected to enter the AoR.
- Reservation of military areas, expressed as grid cells. The grid cell mapping ensures that, in case geographic definitions of areas change, old training data still reflects useful information. The approach also supports the future use of weather data as a predictor. Knowledge about the upper reserved flight level, in combination with NFL, RFL and XFL, allows the neural network to learn patterns about which flights overfly certain areas.

The neural network

A feed-forward neural network with 3 hidden layers containing 170 units each is used. The last layer is connected to the readout layer with 8 units, corresponding to the 8 coordinate values to be predicted.

A lot of work has been done selecting appropriate activation functions connecting the layers in the neural network, defining a good cost function, and choosing an effective optimizer.

The cost function expresses how 'good' the prediction is with respect to the actual flown route, and allows the optimizer to adapt the neural network weights and biases accordingly in the training phase.

2,600,000 iterations of 1,000 random samples are used to train the neural network. The process takes multiple hours.

Validation of the neural network output

When predicting routes for flights not seen before by the neural network, 65% of the predicted routes are within 6 NM of the actual trajectory flown at any point. This result is much better than the current prediction (tagged 'EFD'):

Max distance from flown route	EFD	Neural Network
6 NM	10%	65%
15 NM	60%	94%
30 NM	85%	99%

The deviation metric and visualisations allow the performance of the neural network to be assessed, and are very useful for hyper-parameter tuning, but do not perfectly reflect real-life performance. The time element is not considered and correct knowledge about the predictors is assumed. In real life, the activation schedule of military areas may change up until the time of actual activation, the expected flight entry time could change due to departure noise, or the flight plan could be refiled with different parameters. Depending on look-ahead time, the quality of prediction will vary.

At the end of the paper, values of the lateral deviation are presented at different look-ahead times during real-life operation. This metric reflects full integration of the neural network into the operational system.

The neural network is implemented in **TensorFlow**, a framework for deep learning open sourced by Google. TensorFlow was selected because of its powerful features and the possibility to integrate it in production systems by means of a C++ and Java API.

TensorFlow allows offloading operations to a **GPU**. Modern graphic cards are very suitable for highly parallelised computations.

The neural network is trained on a workstation with one of the fastest GPUs in the world. It has 3840 cores and 24GB of memory.

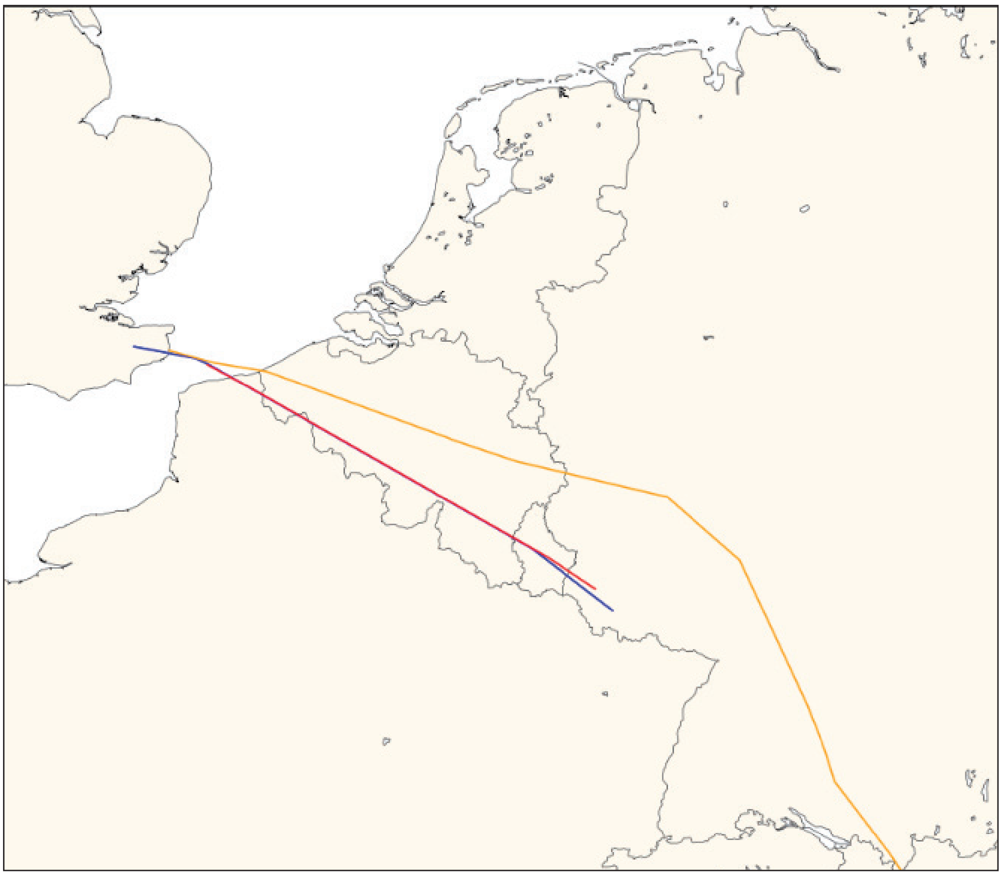


Figure 2 visualises the prediction for the flight used in Figure 1. The predicted trajectory is in red.

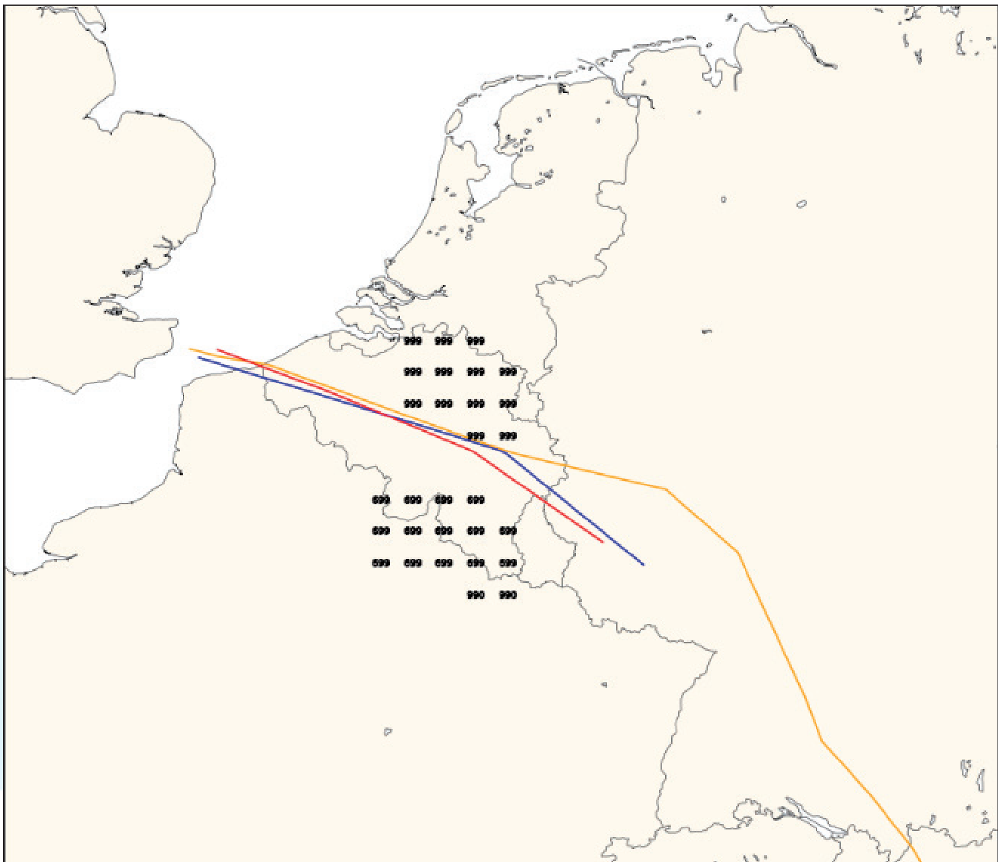


Figure 3 visualises the filed, flown and predicted routes for a sample with active military areas.

Integration of the neural network into the operational ATFCM system

After training, the network is saved as a binary representation. The resulting file is less than 1MB in size and is transferred to the operational system as adaptation data.

Because the new approach to trajectory prediction is fundamentally different from existing methods, and will bring many benefits in the pre-tactical phase when air traffic controller inputs are still unknown, the first operational implementation concerned the ATM Flow & Capacity (ATFCM) system.

The MUAC ATFCM system consists of (a.o.):

- Flight Data Processing (FDP) system, responsible for trajectory prediction and sector sequence calculation in the planning phase, synchronized with the FDP system used in tactical operations.
- Integrated Flow Management Position (iFMP) system, responsible for calculating traffic load and complexity metrics (occupancy & entry counts, weighted occupancy counts, clusters) and evaluating different airspace configurations for a given manpower schedule.

The FDP system is fed with flight plan data for flights already being controlled and flights expected to enter the airspace in the next hours by means of EFD messages from the Network Manager.

Based on the flight plan data received from the FDP system, and the consolidation of airspace reservation data (TSA) from several sources (e.g. the LARA system), the iFMP application continuously calculates the neural network predictors. If predictors change for a flight, iFMP invokes the neural network with a direct call to the TensorFlow Java API, which has been built into the iFMP application.

The predicted route is used to construct a 'what if?' request for the FDP system. The 'what if?' request triggers the FDP system to predict a 4D trajectory using its internal logic but constraining the route to the coordinates provided in the request. The 'what if?' trajectory is maintained in parallel to the original trajectory and both are provided to the iFMP system, displaying them as traffic load to users (supervisors and flow managers).

Flow and capacity management functions on iFMP use the 'what if?' trajectory, for instance to calculate more realistic sector occupancy values. If problems arise with the new prediction logic, iFMP can switch back to the original trajectories. 'What if?' trajectories can be visualised and compared to the original trajectories.

The architecture offers the following benefits:

- Novel techniques can be used in a safety-related system. It is not necessary to change the core of the FDP system and at any point in time it is possible to switch back to unmodified trajectories.
- The approach allows machine learning to be merged with legacy prediction logic, and supports a roadmap for gradually replacing other parts of the logic, e.g. predicting the climb or descent profile through machine learning, or predicting entry times.

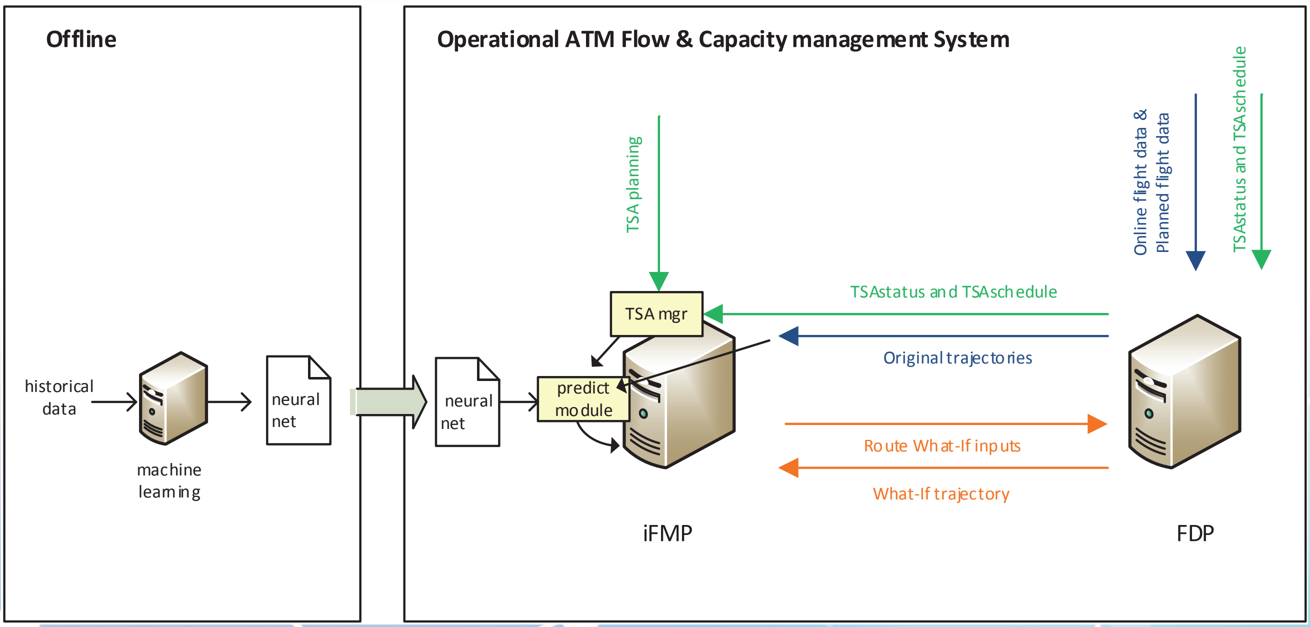


Figure 4: integration of the neural network in the production system

Assessment of real life performance

Figure 5 shows the accumulated lateral distances between the predicted trajectories and corresponding trajectories flown at different look-ahead times for the prediction. The same is done for the trajectories derived from filed flight plan data (label: 'EFD').

The Figure relates to 19 January 2018, and includes 376 flights from the UK to the east and southeast of Europe. Each box plot covers a 30min look-ahead period. Lateral deviation is measured in meters.

The box plot denotes the 25%, 50% and 75% percentiles. The dotted tails are the outliers. Note that the extreme outliers in look-ahead periods >4h are caused by refiled flight plans with different routes. They exist for both types of prediction.

Figure 6 depicts the consolidated statistics for all look-ahead periods up to 6 hours. The Figure relates to Monday 22 January 2018 and covers 432 flights.

The Figure shows that for the vast majority of flights lateral error is reduced by half, with no negative impact on outliers.

Figure 5

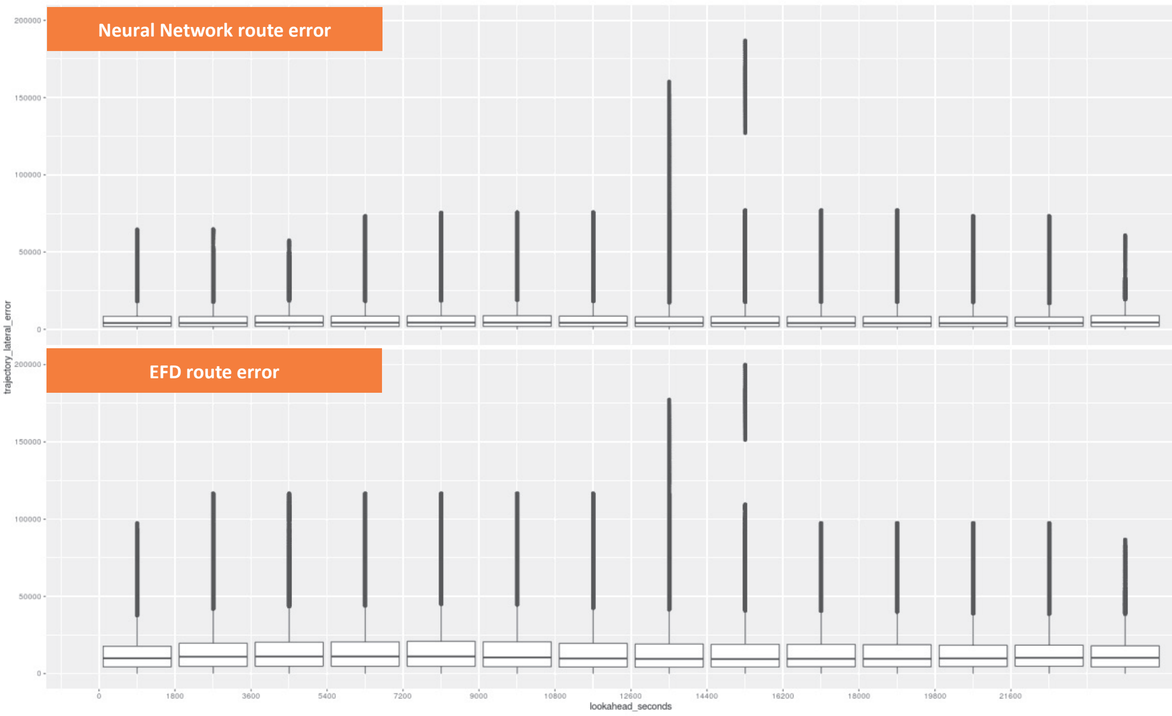
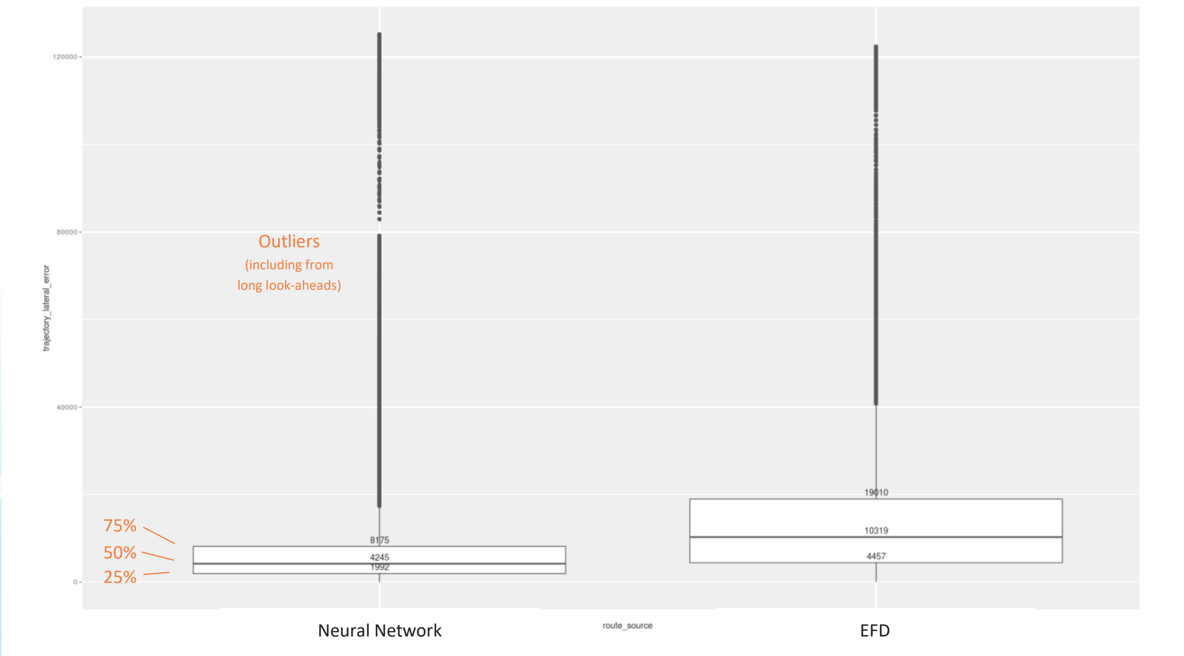
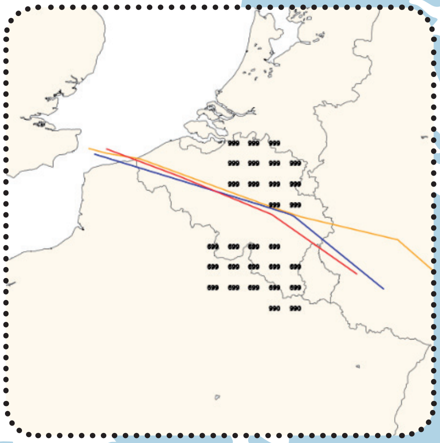


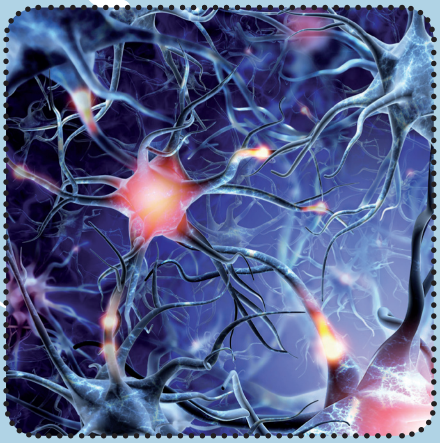
Figure 6



Better route prediction

- Machine learning of several 100.000's of historical trajectories as they were flown
- Takes into account military areas, flight levels, time, day of week, airports, ...
- Reduces lateral error by half when compared to filed route

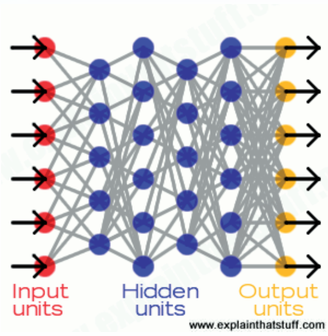




How do neural networks work?

Suppose you have a number of inputs (say, three) and one output. In its simplest form, the output is a * input1 + b * input2 + c * input3. The only thing you then need to do is determine the values of a, b and c. If you have enough observations (i.e. combinations of inputs and corresponding output), you can 'tune' a, b and c to give the best fit.

Actually, if you have ever used the Excel functionality to determine a line-of-best-fit, this is exactly what you've been doing!



Neural networks do exactly the same, except that they can link the inputs to multiple layers (the 'hidden units') which in the end link to the output. The connections between one unit and another are represented by the weighting factors (a, b and c in our example above), which can be either positive (if one unit excites another) or negative (if one unit suppresses or reverses another). The higher the weight, the more influence one unit has on another. As you can see in the picture, the number of combinations and connections rapidly rises to formidable numbers as you increase the number of hidden units and the number of layers!

In addition, each neuron is followed by a so-called 'activation function', which applies a non-linear operation to the output of the neuron. In this way any arbitrary relation between variables can be approximated by a neural network.

In the training phase, the neural network is fed with inputs (in our case flight plans) and known outputs (in our case flown trajectories). It uses this data set to find the best fit for parameters a, b and c (and many more!). For this, it uses a cost function that expresses how 'good' the prediction is with respect to the actual route flown, and that allows the optimizer to adapt the neural network weights and biases accordingly in the training phase.

Then, in the application phase (i.e. during live operations) it receives the inputs (flight plans) and applies the weighting factors that it has determined in the learning phase to produce the output – the statistically most likely trajectory.

Maastricht Upper Area Control Centre
Stakeholder Management

EUROCONTROL

Horsterweg 11
6199 AC Maastricht-Airport
The Netherlands
Tel: +31 43 366 1352 / 1234
muac.info@eurocontrol.int
www.eurocontrol.int/muac