Flight Level Prediction with a Deep Feedforward Network

SESAR Innovation Days 2018
Dr. Matthias Poppe, Debora Fieberg, Roland Scharff, Jörg Buxbaum
04.12.2018
Background

- Controller decision support tools (DST) like Conflict detection rely on accurate 4D Trajectory Prediction (TP)
- Lateral (2D) position can be predicted quite accurately
  - After acceleration phase, aircraft operate with constant Calibrated Airspeed or Mach number; RNP enables accurate 2D navigation
- High uncertainty in vertical speed -> climb rate
  - Depends on many factors (not exhaustive): atmospheric conditions, aircraft weight, cost index, Flight Management System, operational restrictions, human intervention as controller clearance, pilot discretion
- To cater for uncertainty, DST need to apply buffer
  - Example: Controller Assistance Tools (CATO) use a static vertical buffer of ±500 feet/minute related to observed actual rate
Motivation

- For a prediction horizon of 6 minutes (typical value for short sectors with mostly vertical traffic patterns):
  - Buffer is ±3000 feet (60 Flight Level FL)

- Idea: develop and train a Deep Feedforward network, using only easily available data from Mode S plus aircraft type
  - Main objective: in the climb phase, for each departure, predict the reached flight level after n minutes (here: n=6)
  - Use this information to reduce buffer size for vertical TP and thus reduce number of false positives for conflict detection

- Once trained, each new departure flight can be used to further improve the neural network (online learning)
Feedforward Network (principle)

- Use supervised learning for regression task, i.e. predict FL to be reached in $n$ minutes (target $y$) as numerical value
- Training data: update weight $W$, bias $b$ of the neural network such that the loss function $J(\theta)$ will be minimized (e.g. mean squared error)
- Validation data will be used to evaluate the trained model

$$h^i = \sigma(b^i + W^i x)$$ (1)

$\sigma$ activation function
Raw Input Data

- Mode S Enhanced Surveillance (EHS) BDS Register (mandatory for IFR flights): 4.0 Selected Vertical Intention, 5.0 Track and Turn Report, 6.0 Heading and Speed Report (from ASTERIX CAT048)
Data Preparation

- Data cleansing -> select climbing flights
  - $\min_t \{FL(t)\} < 20$, $\max_t \{FL(t)\} > 285$ and $\arg\min_t \{FL(t)\} < \arg\max_t \{FL(t)\}$

- Extracted (raw) data for selected flights

<table>
<thead>
<tr>
<th>Index</th>
<th>Abbreviation</th>
<th>Explanation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$t$</td>
<td>Time</td>
<td>sec</td>
</tr>
<tr>
<td>1</td>
<td>FL</td>
<td>Flight Level</td>
<td>FL</td>
</tr>
<tr>
<td>2</td>
<td>AID</td>
<td>Callsign</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>TAS</td>
<td>True Airspeed</td>
<td>kt</td>
</tr>
<tr>
<td>4</td>
<td>FCU</td>
<td>Flight Control Unit Selected Altitude</td>
<td>ft</td>
</tr>
<tr>
<td>5</td>
<td>GS</td>
<td>Ground Speed</td>
<td>kt</td>
</tr>
<tr>
<td>6</td>
<td>$\alpha$</td>
<td>True Track Angle</td>
<td>deg</td>
</tr>
<tr>
<td>7</td>
<td>$h$</td>
<td>Magnetic Heading</td>
<td>deg</td>
</tr>
<tr>
<td>8</td>
<td>IAS</td>
<td>Indicated Airspeed</td>
<td>kt</td>
</tr>
<tr>
<td>9</td>
<td>Ma</td>
<td>MACH Number</td>
<td>Mach</td>
</tr>
<tr>
<td>10</td>
<td>$r_{inert}$</td>
<td>Inertial Vertical Speed (caveat: sign!)</td>
<td>ft/min</td>
</tr>
<tr>
<td>11</td>
<td>$p$</td>
<td>Barometric Pressure</td>
<td>mb</td>
</tr>
<tr>
<td>12</td>
<td>$r_{baro}$</td>
<td>Barometric Altitude Rate (caveat: sign!)</td>
<td>ft/min</td>
</tr>
<tr>
<td>13</td>
<td>$r_{calc}$</td>
<td>Calculated Rate of Climb</td>
<td>ft/min</td>
</tr>
</tbody>
</table>

04.12.2018
Derived Features

- From the raw input data, we derived several features $x_0$ to $x_{19}$ which are supposed to impact the climb behavior as input for the neural network.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_0$</td>
<td>Take Off Safety Speed $V_2$ (slide 8)</td>
</tr>
<tr>
<td>$x_1$, $x_2$</td>
<td>Average and actual Rate of Climb (slide 9)</td>
</tr>
<tr>
<td>$x_3$</td>
<td>Initial peak Rate of Climb (slide 10)</td>
</tr>
<tr>
<td>$x_4$, $x_5$</td>
<td>Wind component in flight direction</td>
</tr>
<tr>
<td>$x_6$</td>
<td>Heading Change flag (above/below threshold)</td>
</tr>
<tr>
<td>$x_7$</td>
<td>Altitude Restriction Flag (slide 11)</td>
</tr>
<tr>
<td>$x_8$</td>
<td>Time since FL100</td>
</tr>
<tr>
<td>$x_9$ to $x_{13}$</td>
<td>Flight Level 60, 45, 30, 15, 0 seconds ago</td>
</tr>
<tr>
<td>$x_{14}$ to $x_{18}$</td>
<td>IAS 60, 45, 30, 15, 0 seconds ago</td>
</tr>
<tr>
<td>$x_{19}$</td>
<td>Aircraft type (slide 12)</td>
</tr>
</tbody>
</table>
Take Off Safety Speed

- Minimal speed aircraft climbs to first safe altitude
  - With all engines operative ~ \( V_2 + 10 \) knots
  - Depends on aircraft weight, atmospheric conditions
  - \( IAS(t_x) \) with \( t_x \) such that \( FL(t_x) \leq 10 \)
  - Varies significantly due to aircraft parameters and among aircraft types

![Distribution of V2](image-url)
Available rates of climb:

- inertial vertical velocity $\text{ROC}_{\text{ins}}$
- barometric altitude rate $\text{ROC}_{\text{baro}}$
- calculation from flight level and time $\text{ROC}_{\text{calc}}$

\[ x_2 := \frac{\sum_{t=0}^{t_x} \text{ROC}(t)}{t_x} \quad \text{with} \quad FL(t_x) < 285 \]

with $\text{ROC}(t) = \begin{cases} 
\text{ROC}_{\text{baro}}(t), & \text{if } \text{prop}_{\text{ins}} \geq 0.8 \\
\text{ROC}_x(t), & \text{else}
\end{cases}$

for $x = \arg\max \{ \text{prop}_x | x \in \{\text{baro, ins}\} \}$

and $\text{prop}_x$ proportion of finite datapoints.
Initial peak Rate of Climb

- Initial peak typical for performance of aircraft (?)
- (weak) correlation of peak and average ROC from FL100 to FL285: $r = 0.40$
- might be a hint to the energy share factor
- $x_3 := \max\{ROC(t) \mid FL(t) < 28\}$
- Value of 2800 feet chosen according to data set
  - Should include typical peak
  - Caveat: depends on airport height because below transition level

![Graph showing ROC and IAS trends with annotations for BDS 4.0 FCU Sel.Alt, Mode C FL, BDS 6.0 Inert.VR, and BDS 6.0 IAS.](image)
**Altitude Restriction Flag**

- **Boolean flag** $x_7 := \begin{cases} 0, & \text{if } \exists t: FCU(t) - 100 \times FL(t) \leq \omega \\ 1, & \text{else} \end{cases}$

- We choose to set $\omega := 1400$ [feet]

- Majority of flights climb „unrestricted“ according to this criterion
Aircraft type (all airports)

April to July 2018
Training: Model Loss

- Deep feedforward network
  - Keras with tensorflow and GPU
  - N=7 layers, M~4000 neurons
  - ReLu activation function for hidden layers
  - Different optimizer tested -> uncritical if not stucked at the beginning
  - Dropout regularization with 15% to 30% drop rate

• Training after 100 epochs shows still decrease in loss for both train and test data

Adam = Adaptive moment estimation optimizer (extension of Stochastic Gradient Descent)
Learning Curve

- About 100,000 climbing flights available from all German airports
- For training, 4 samples used per flight
  - 400k features/labels
- Learning curve suggests that at least 50k samples are required
  - Even more if network becomes „deeper“, i.e. more layer or more neurons per layer

\[ R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \]

- \( SS_{res} = \sum_{i} (y_i - f_i)^2 \) (Predicted data \( \hat{y} \))
- \( SS_{tot} = \sum_{i} (y_i - \bar{y})^2 \) (Mean of observed data)

\( R^2 \) is the proportion of the variance in the dependent variable that is predictable (max = 1)
## Results

<table>
<thead>
<tr>
<th>Metric</th>
<th>Prediction Time 360 seconds</th>
<th>Prediction Time 180 seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test Data</td>
<td>Train Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R² score</td>
<td>0.922 (0.931)</td>
<td>0.931 (0.942)</td>
</tr>
<tr>
<td>Mean error</td>
<td>8.14 (7.58) FL</td>
<td>7.63 (6.78) FL</td>
</tr>
<tr>
<td>Std. Dev. error</td>
<td>7.21 (6.84) FL</td>
<td>7.04 (6.56) FL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R² score</td>
<td>0.975 (0.978)</td>
<td>0.978 (0.981)</td>
</tr>
<tr>
<td>Mean error</td>
<td>5.35 (5.13) FL</td>
<td>4.95 (4.53) FL</td>
</tr>
<tr>
<td>Std. Dev. error</td>
<td>5.32 (5.16) FL</td>
<td>5.13 (4.85) FL</td>
</tr>
</tbody>
</table>

() in brackets: only aircraft type A320 selected
1 FL = 100 feet

- Feedforward network is able to learn distribution $p(y|x)$
- Mean absolute error about 800 feet for six minutes prediction horizon with standard deviation of about 700 feet
- If only specific aircraft type (here: A320) selected, results are getting better
- The shorter the time horizon, the better the result
Conclusions

- Results confirm possibility to reduce uncertainty buffer for TP in climb phase
- Network is able to predict flight level with supervised learning and can "distinguish" different operational environments due to training data
- Most effort put in extracting features (specialist know-how)
  - Building network with Keras, tensorflow straightforward although optimization of hyperparameter needs lot of computing power
- Number of features can possibly be modified and/or reduced
  - Apply Sequential Backward Selection algorithm
  - Cluster aircraft types with similar (climb) performance
- Further improvements envisaged by
  - Bootstrap aggregating
  - Combination with Recurrent Architecture, e.g. Long Short Term Memory in order to take into account dynamics of climb phase
Future Work

- Analysis of outliers ongoing
  - First analysis suggests that biggest deviations are caused by level-off segments because of controller clearances
  - Train network only for unrestricted climbs and include clearance as a feature?

- Compare data to Full Flight Simulator
  - Controlled environment with known parameter yet realistic flight behavior
  - Derive features and predict flight level with trained feedforward network for this specific aircraft type
  - May allow to estimate parameters like Cost Index, Take Off weight

- Extend network architecture to allow online-prediction for departing flights
  - Each flight can in turn be used to further train and improve the network
Thank you very much for your attention!
## Backup: Keras model summary()

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Param #</th>
</tr>
</thead>
<tbody>
<tr>
<td>dense_1 (Dense)</td>
<td>(None, 2460)</td>
<td>415740</td>
</tr>
<tr>
<td>dense_2 (Dense)</td>
<td>(None, 1220)</td>
<td>3002420</td>
</tr>
<tr>
<td>dense_3 (Dense)</td>
<td>(None, 610)</td>
<td>744810</td>
</tr>
<tr>
<td>dense_4 (Dense)</td>
<td>(None, 100)</td>
<td>61100</td>
</tr>
<tr>
<td>dense_5 (Dense)</td>
<td>(None, 40)</td>
<td>4040</td>
</tr>
<tr>
<td>dense_6 (Dense)</td>
<td>(None, 10)</td>
<td>410</td>
</tr>
<tr>
<td>dense_7 (Dense)</td>
<td>(None, 1)</td>
<td>11</td>
</tr>
</tbody>
</table>

Total params: 4,228,531

Trainable params: 4,228,531

Non-trainable params: 0
Backup: Post results

BATCH_SIZE = 8192
EPOCHS = 200+ReduceLRonPlateau
PREDICTION_TIME: 360 seconds
ALL_AIRCRAFT_TYPES

Test set score (loss) = 0.1817
r2 score test : 0.918
r2 score train: 0.922
Test Data mean absolute error: 7.28 FL
Test Data error standard dev = 6.90 FL
Train Data mean absolute error: 6.91 FL
Train Data error standard dev = 6.80 FL