DATA-DRIVEN MASS ESTIMATION OF HEAVY-DUTY VEHICLES

Abdurrahman İşbitirici

University of Bologna

Automotive Engineering for Intelligent Mobility

IPG

APPLY & INNOVATE

THE EXPERT FORUM FOR VIRTUAL TEST DRIVING

September 11–12, 2024 in Karlsruhe

Motivation

- Less expensive solution than using axle load sensors
- Availability and reliability of axle load sensors
- Fault detection in case sensor fails

- Enable advanced powertrain and chassis control, rollover avoidance
- Reduce energy consumption
- Tear on road infrastructure (overload)
- Driving style (braking and acceleration time)

- Fleet order
- Estimation of number of passenger in the buses
  - ➢ Resource planning
  - ➢ Air conditioning

Image created by OpenArt.ai

Problem Definition

- **Regression** or Classification
- Sequence-to-sequence or **sequence-to-one**
- Model-based or **AI-based**
- **Longitudinal Dynamics** or Suspension Dynamics

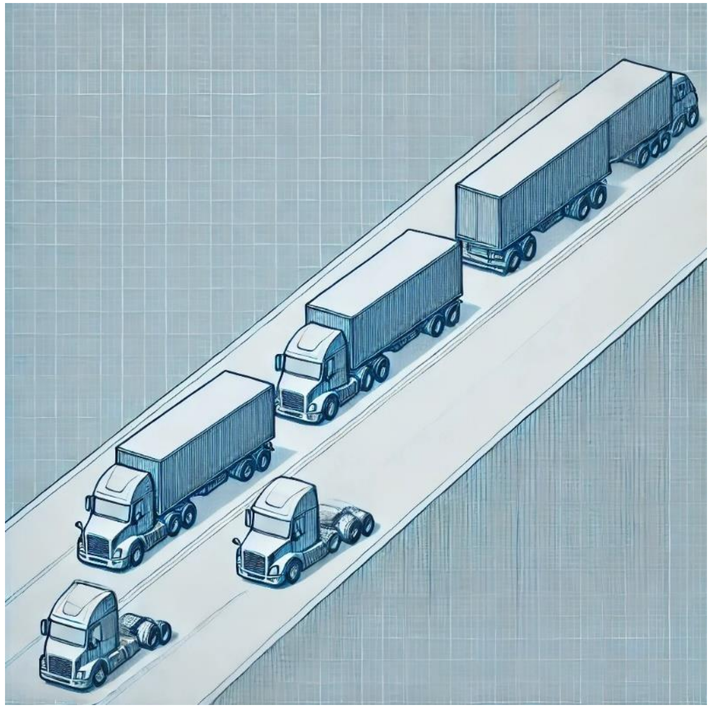


Image created by Dall-E

Methods

Model-Based

- Vehicle model is necessary.
- Vehicle parameters such as inertia, friction coefficient, efficiencies are needed.
- Vehicle parameters need to be updated vehicle to vehicle.
- Most common methods
  - Recursive Least Square (RLS)
  - Extended Kalman Filter (EKF)

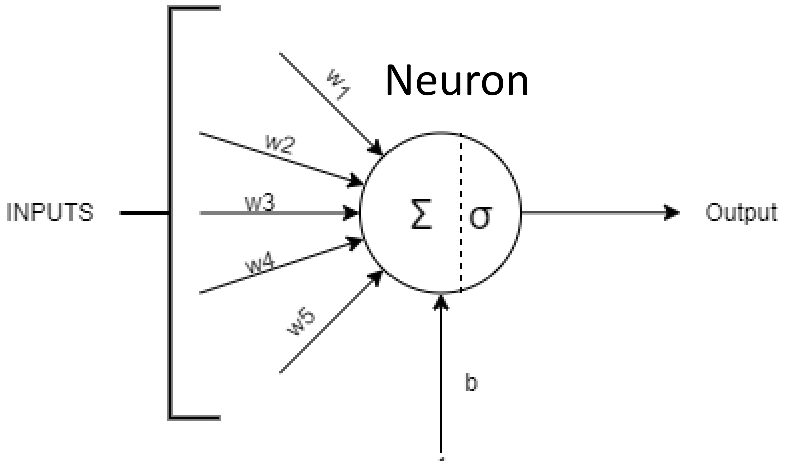
AI-Based

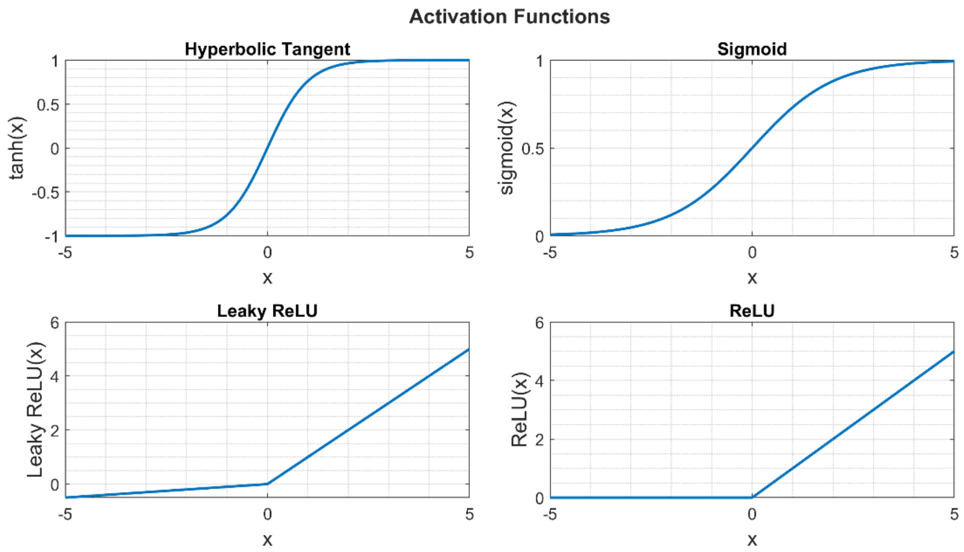
- Feed Forward Neural Networks (FFNN): Shallow or Deep Neural Network (DNN)
- Sensor measurements are necessary
- Various mass data is necessary for training
- More generic approach

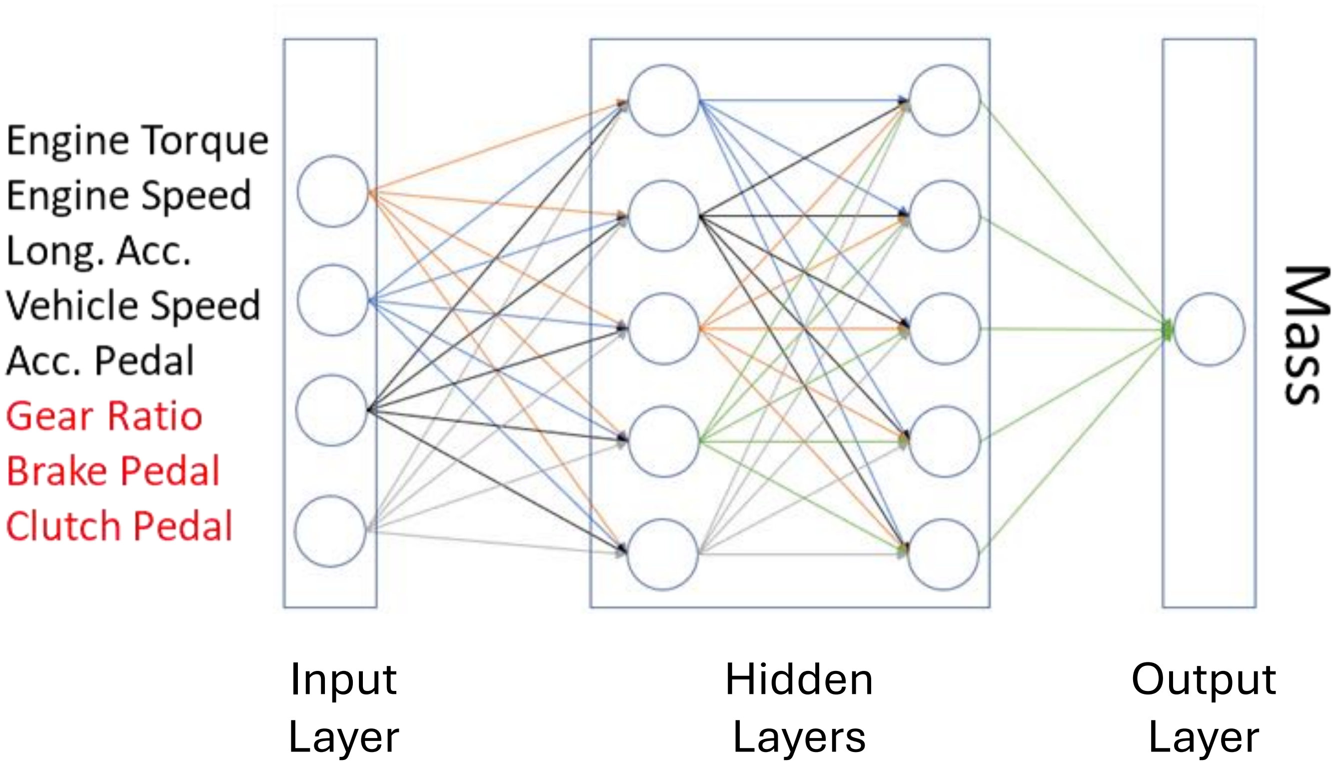
Combination

- RLS + DNN by fuzzy logic

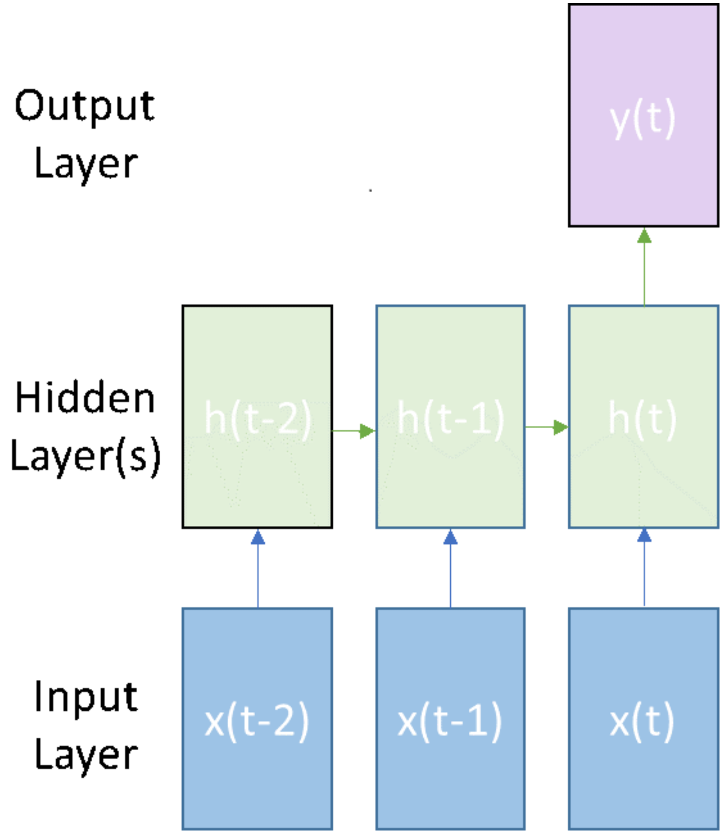
FFNN-Based Mass Estimation







Recurrent Neural Networks



sequence-to-one

- DNN has no memory, separate inputs are necessary.

$$h_t = \sigma(W_{hx}x_t + W_{hh}h_{t-1} + b_h)$$

$$\hat{y}_t = \sigma(W_{yh}h_t + b_y)$$

- RNN = FFNN + cyclic connections

- Each time step of a feature is not defined as a separate input for RNNs, therefore different time steps of same feature share the same weight.

$$h_t = g_t(x_t, x_{t-1}, \dots, x_2, x_1)$$

RNN Problems

Loss Function

$$J(\boldsymbol{\theta}) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(\hat{y}-y)^2}$$

Weight Update

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} - \alpha \nabla J(\boldsymbol{\theta}^{(i)})$$

Weights and Biases

Learning Rate

due to Backpropagation by applying chain rule

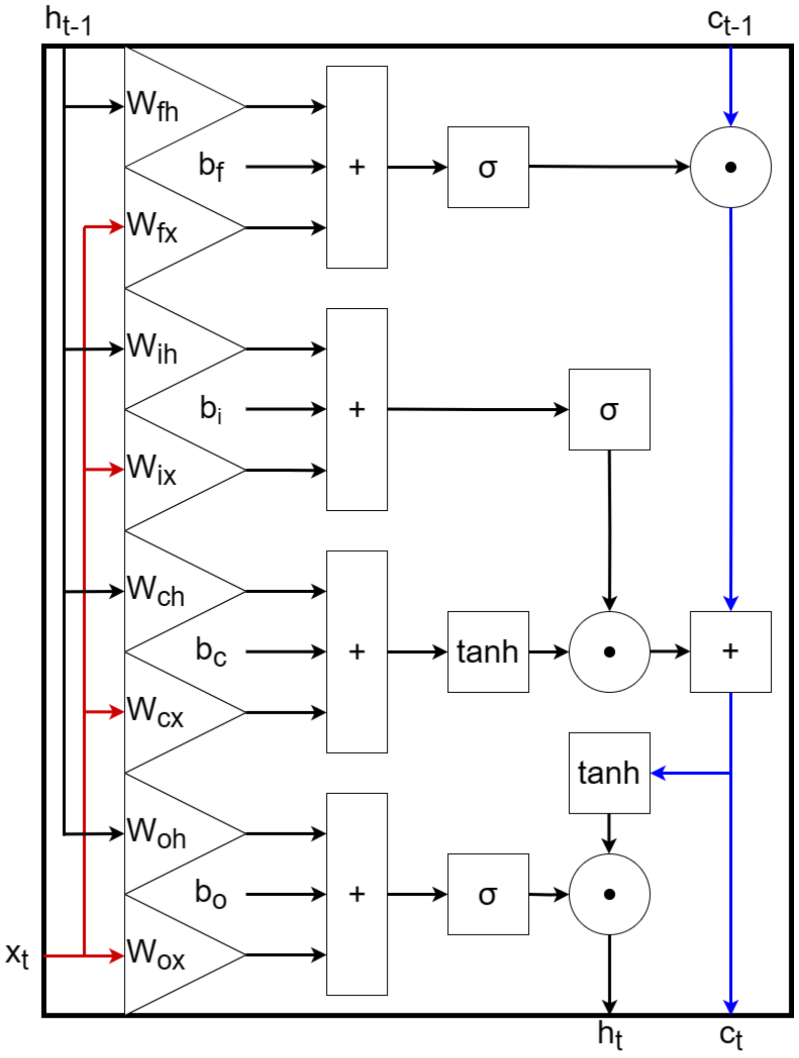
- exploding gradient

(solution: gradient clipping)

- vanishing gradient

(solution: LSTM)

Long Short-Term Memory (LSTM)



Gates

- Forget f
- Input i
- Output o

States

- Cell c
- Hidden h

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f)$$

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i)$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o)$$

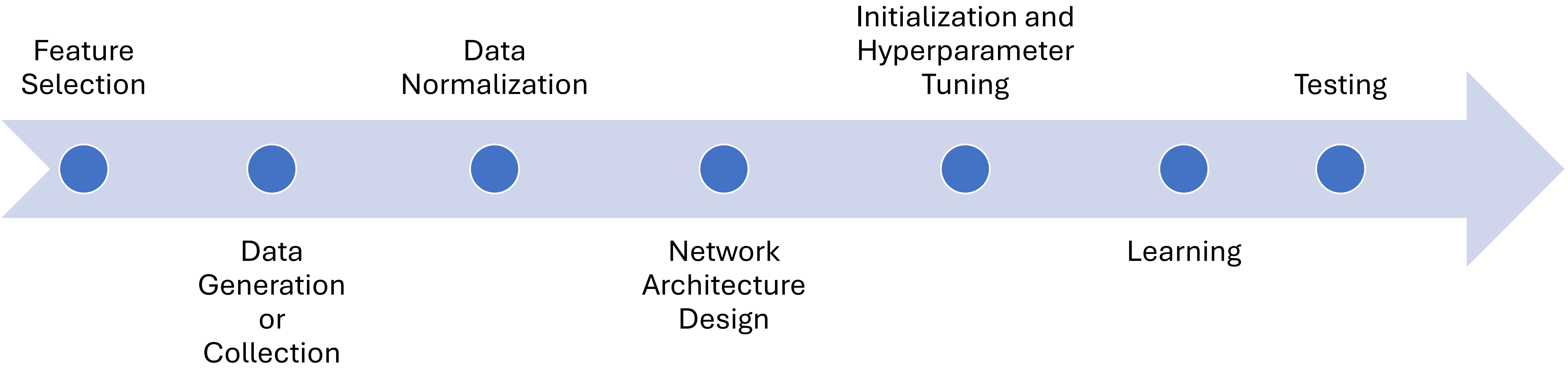
$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c)$$

$$h_t = o_t \odot tanh(c_t)$$

Activation function

- Sigmoid for gates
- Hyperbolic tangent for states

Data-Driven Mass Estimation Steps



Feature Selection

Vehicle Longitudinal Dynamics

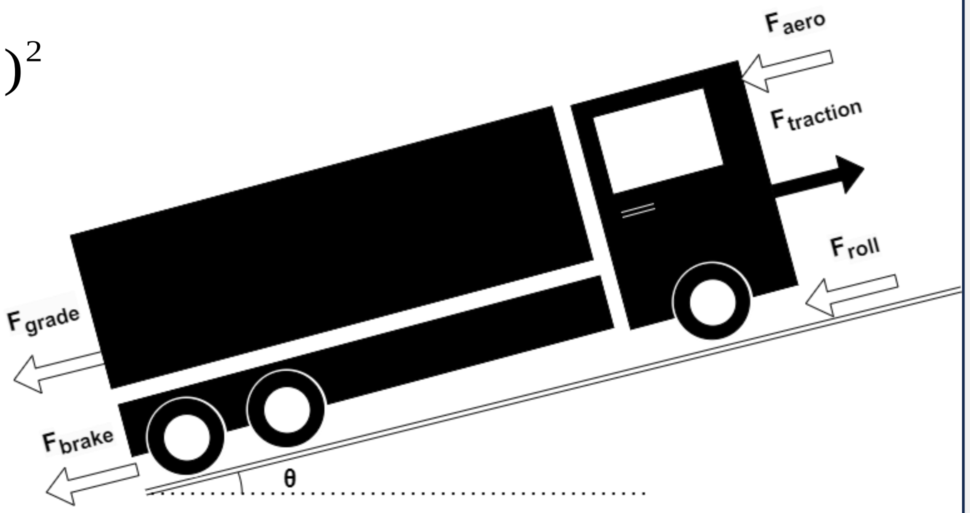
$$F_{aero} = 0.5 \rho C_d A_f (v + v_{wind})^2$$

$$F_{roll} = \mu m g \cos(\theta)$$

$$F_{grade} = m g \sin(\theta)$$

$$F_{traction} = \frac{T_{engine} - J_{eq}\dot{\omega}}{\frac{r_{wheel}}{g_r g_{final}}}$$

$$m\dot{v} = F_{traction} - F_{brake} - F_{aero} - F_{roll} - F_{grade}$$

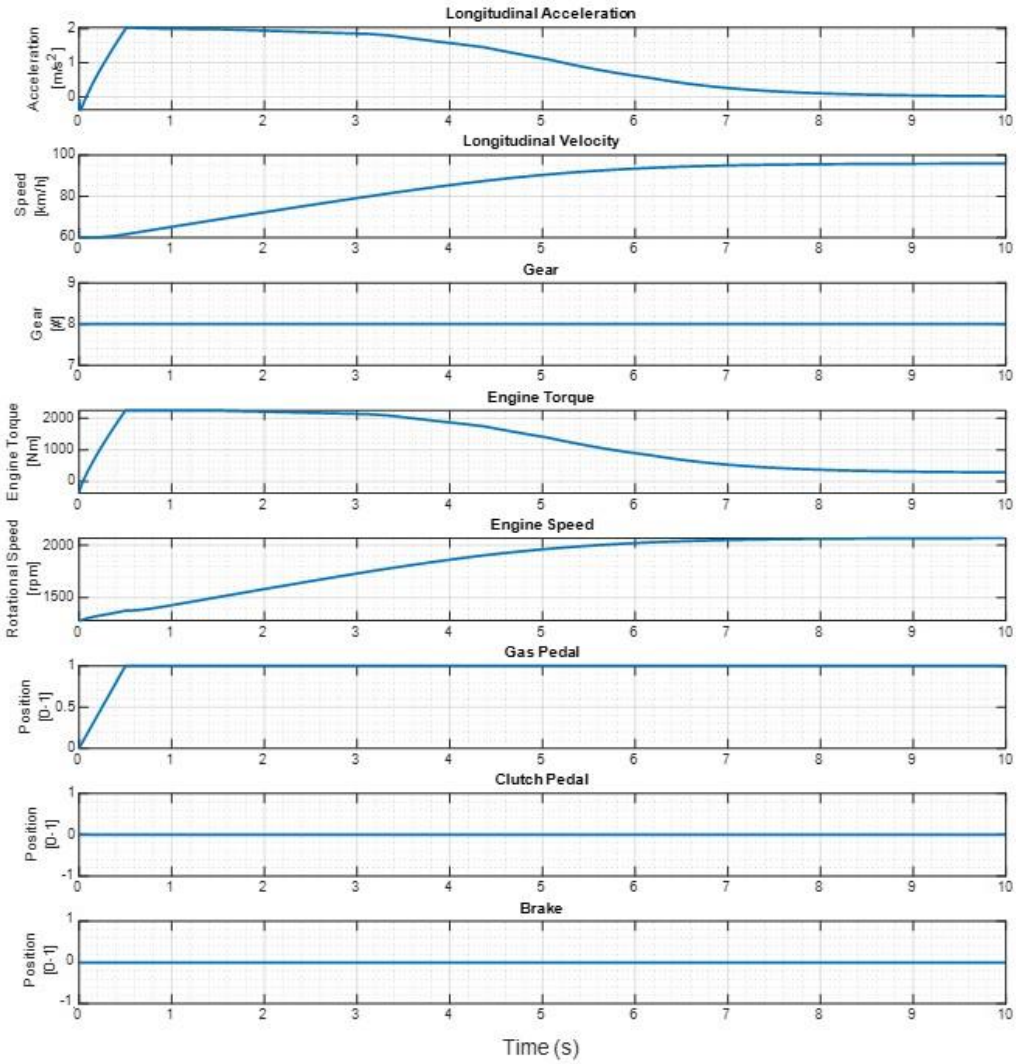


Signals

- Longitudinal acceleration
- Vehicle speed
- Engine speed
- Engine Torque
- Accelerator pedal position
- Road gradient
- Brake pedal position
- Gear ratio

Training and Validation Data

- Fixed gear
- Engaged clutch
- No braking
- Limited speed
- Straight road
- Flat road
- No wind
- Constant rolling resistance coefficient



Training and Validation Data

<table>
<tr><th>Initial Velocity</th><th>[km/h]</th><td colspan="3">60</td><td colspan="5">70</td><td colspan="3">80</td><td colspan="3">90</td></tr>
<tr><th>Initial Accelerator Pedal Position</th><th>[%]</th><td colspan="8">0</td><td colspan="6">100</td></tr>
<tr><th>Final Pedal Position</th><th>[%]</th><td>0</td><td>10</td><td colspan="2">20</td><td>30</td><td>40</td><td colspan="2">50</td><td>60</td><td>70</td><td colspan="2">80</td><td>90</td><td>100</td></tr>
<tr><th>Time to Reach Final Pedal Position</th><th>[s]</th><td colspan="2">0.5</td><td colspan="3">1</td><td colspan="3">2</td><td colspan="2">5</td><td colspan="4">10</td></tr>
<tr><th>Load</th><th>[kg]</th><td>0</td><td>125</td><td colspan="2">250</td><td>375</td><td colspan="2">500</td><td>625</td><td>750</td><td>875</td><td colspan="4">1000</td></tr>
</table>

Total 3960 simulations

- 8 initial conditions
- 55 scenarios
- 9 masses

Training and validation data based on excitation (various gas pedal positions and rates)

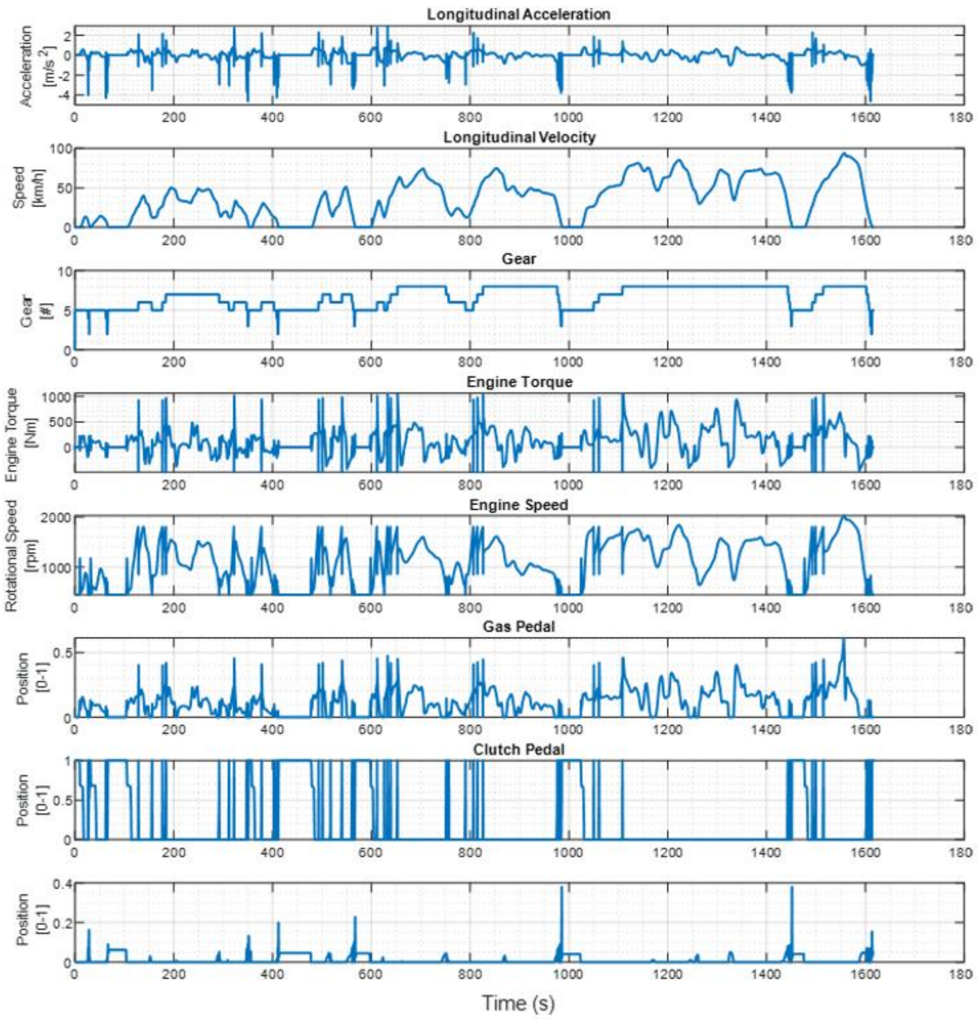
10 s simulation time

100 Hz frequency

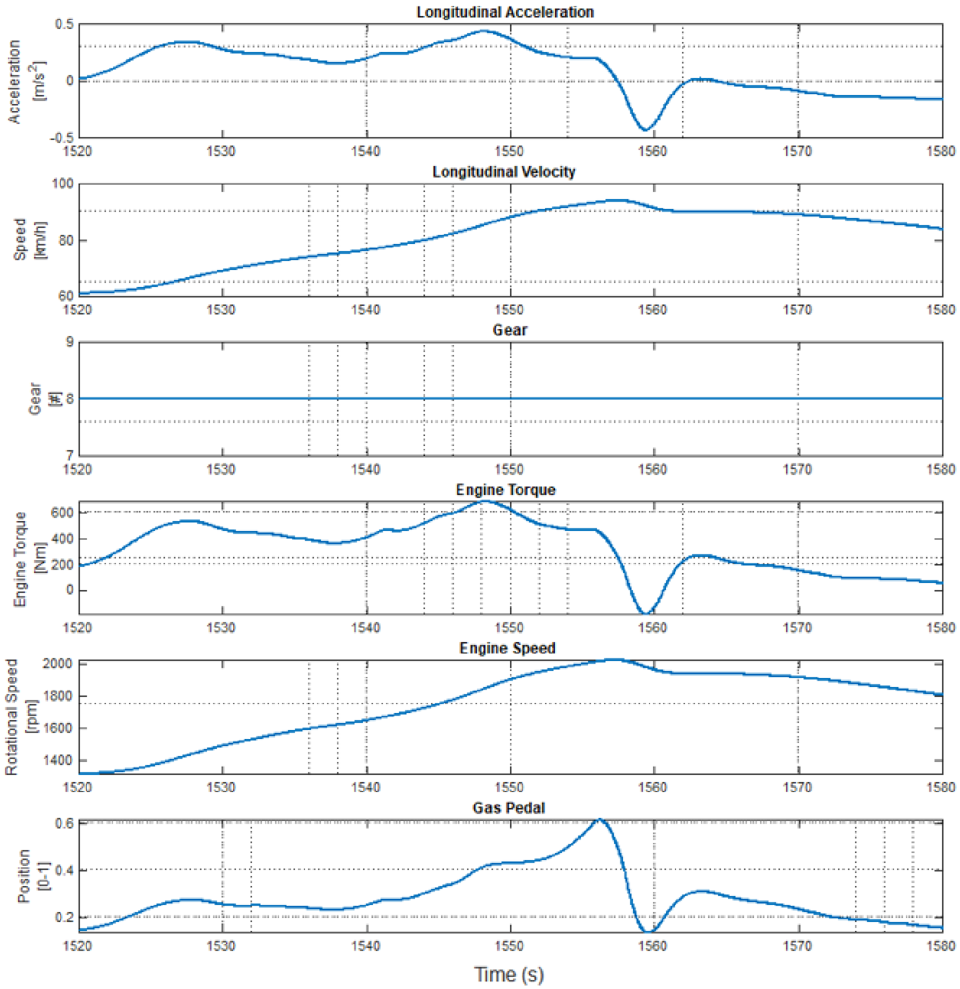
495 simulations for validation

Min-max normalization based on training data

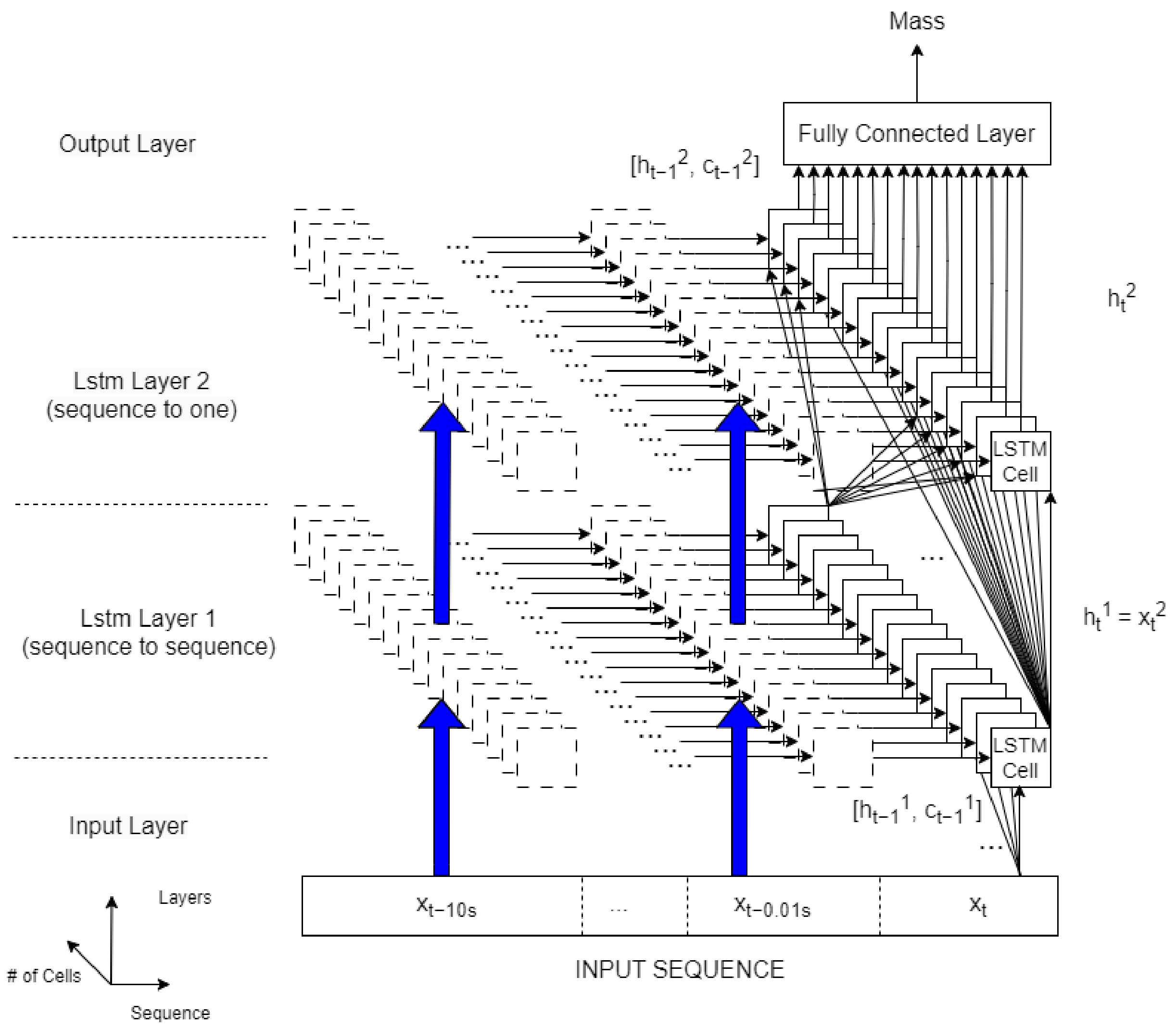
Test Data

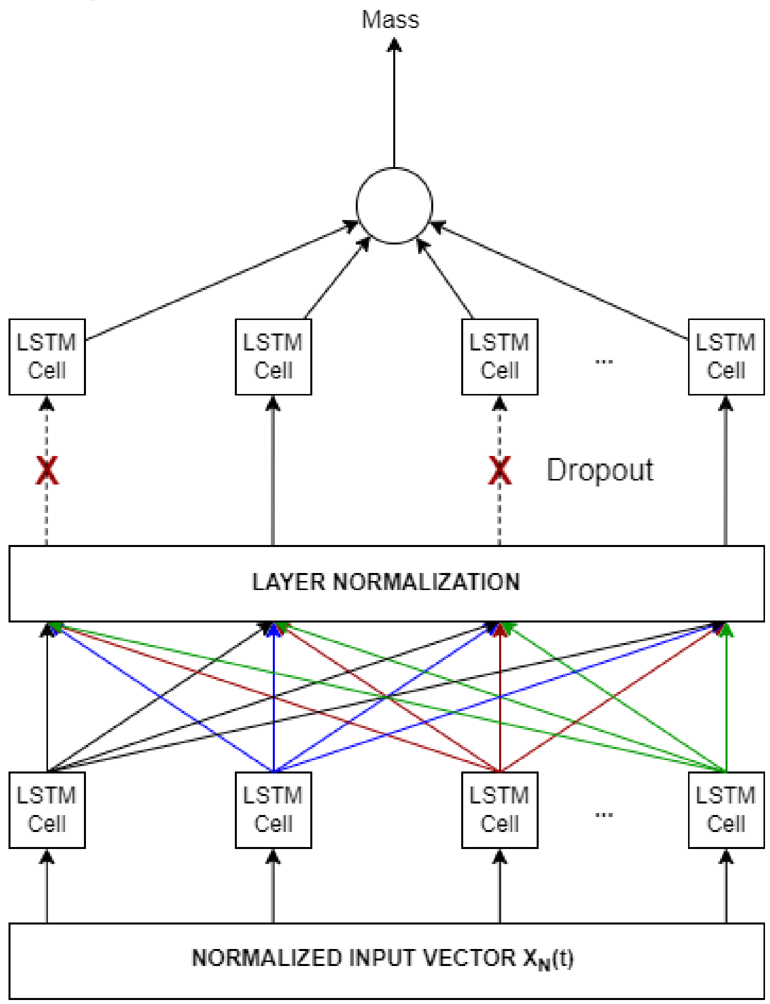


- TruckMaker 11.0
- 6 various masses
- Only highest gear
- Test data based on realistic driving cycle WLTC



LSTM-based Mass Estimation





Initialization and Hyperparameter Tuning

- Parameter Initialization
- Overfitting Prevention Techniques
- Loss function and Evaluation Metrics
- Batch size
- Weights are adjusted after each iteration to reduce loss function
  - Optimizer
  - Learning rate

Hyperparameters	**Content**
Hidden layer number	2
Neuron number in the hidden layers	16, 16
LSTM layers output mode	sequence, one
Dropout	50%
Optimizer	Adam [62]
Computational resource	Gpu
Epoch number	5000
Initial learning rate	0.0025
Sequence length	longest or shortest
Output network	best-validation-loss
Gradient threshold	1

Learning Process

Underfitting & Overfitting

- Underfitting
  - increase # layers or neurons

- Overfitting
  - collect more data
  - reduce model capacity
  - dropout

Learning rate effect on loss function

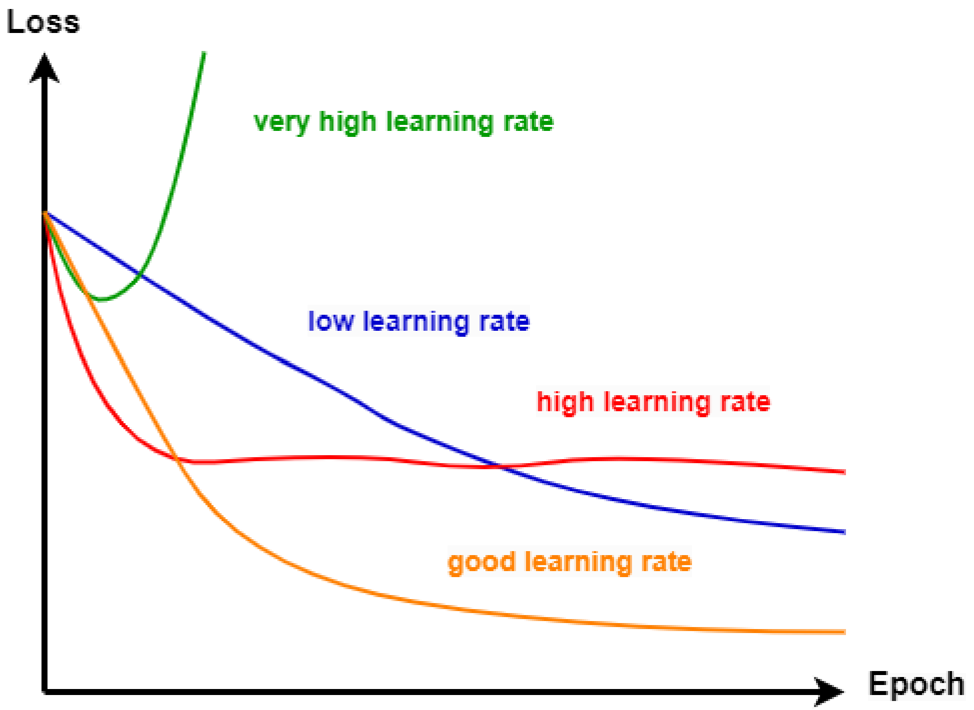
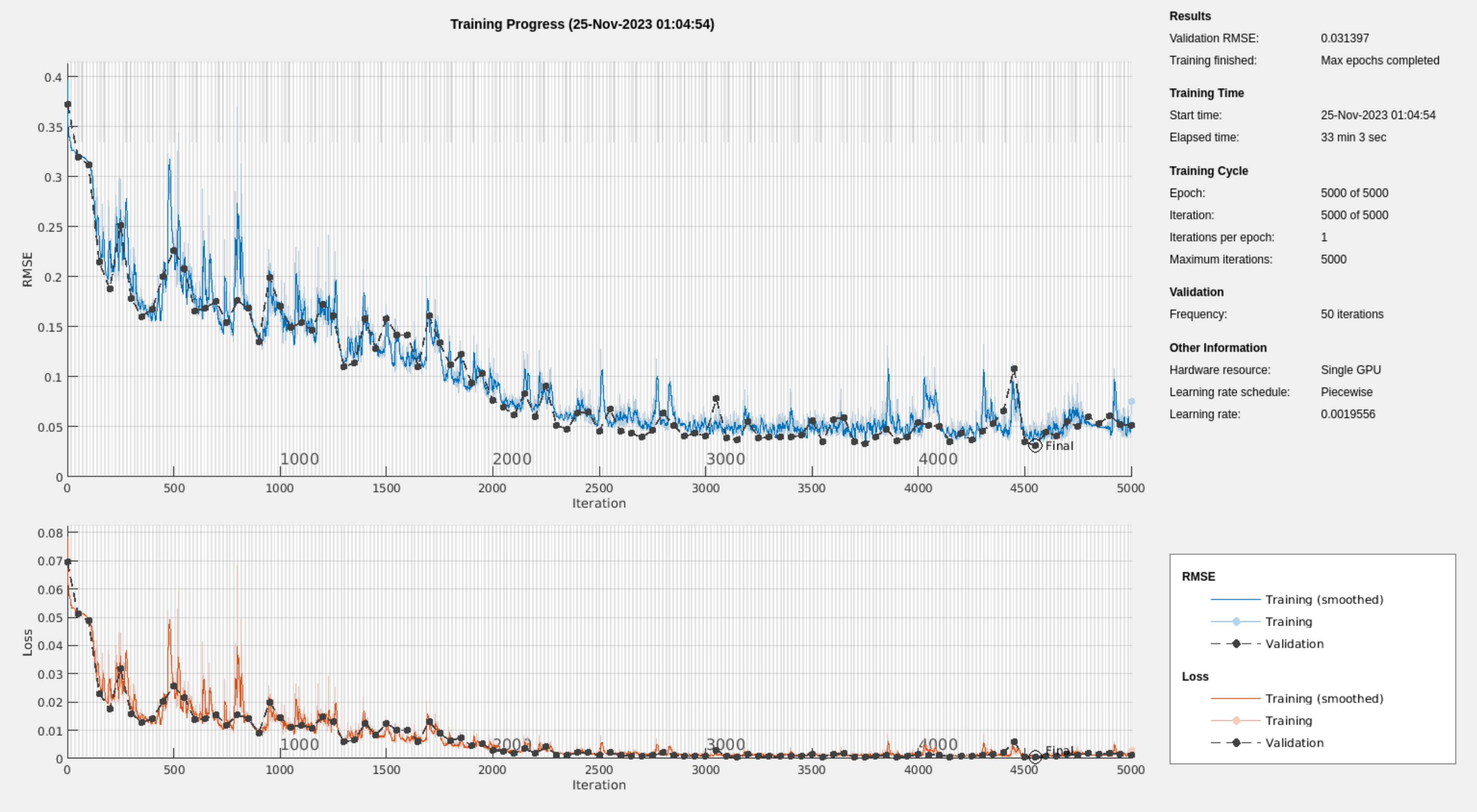


Figure adapted from Andrej Karpathy, Neural Networks-3, CS231nWinter 2016, Stanford University

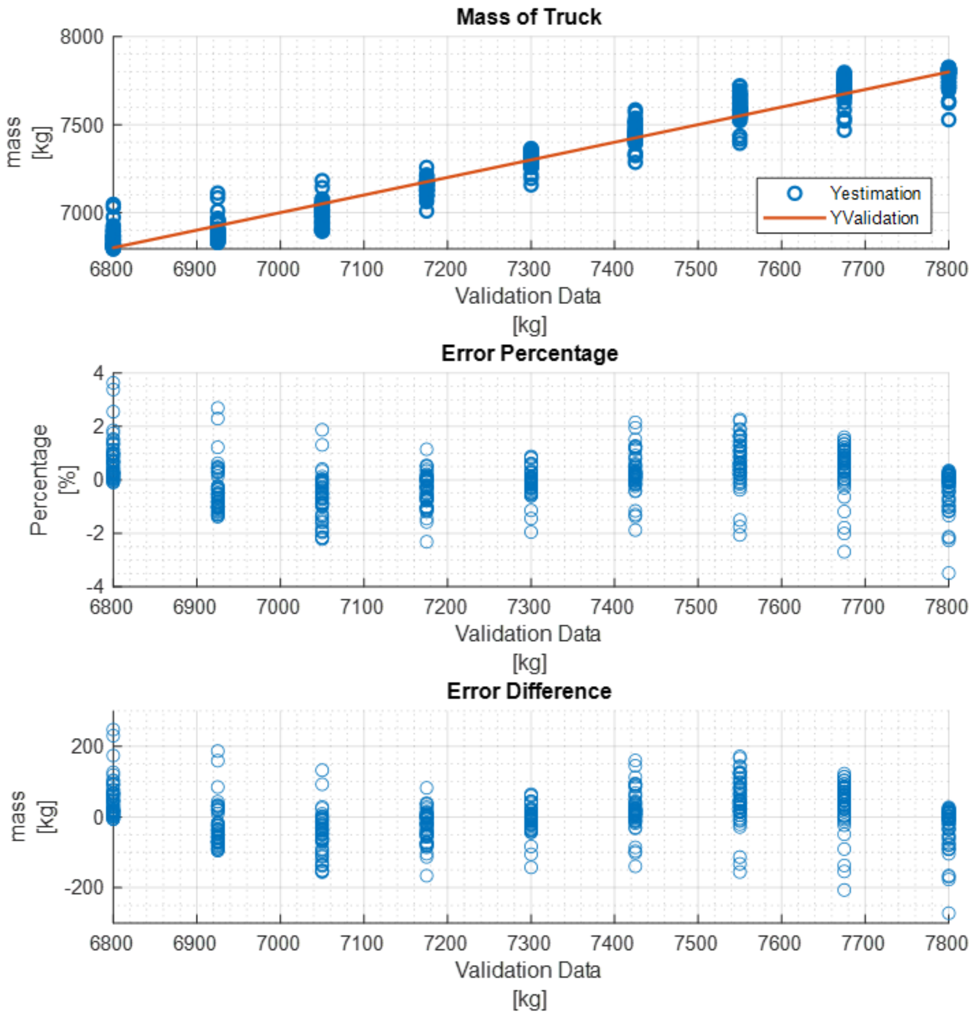
Learning Process



Weights are updated based on training data in order to minimize the cost function.

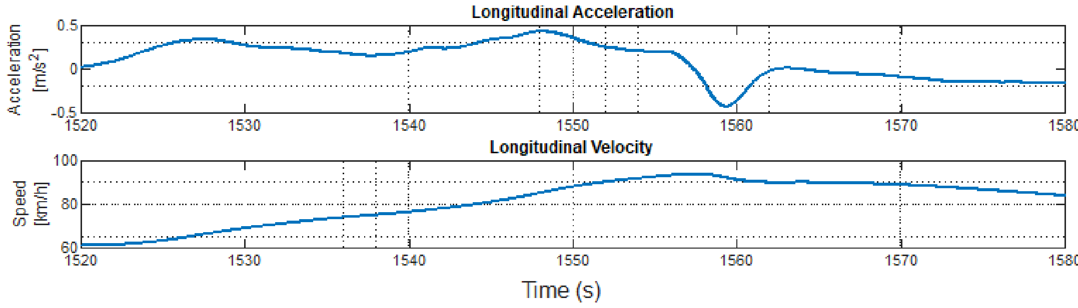
Those weights are used for validation data.

Validation Results

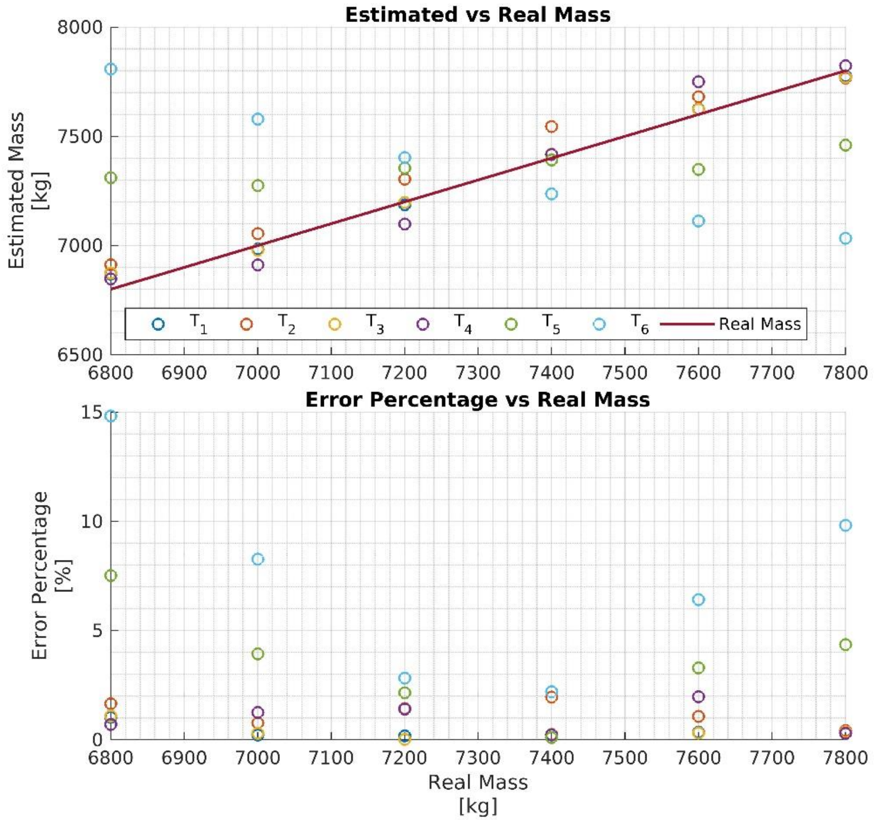


- 1/8 of the data is used for validation to prevent overfitting.
- 95.76% of the validation data has less than 2% error.

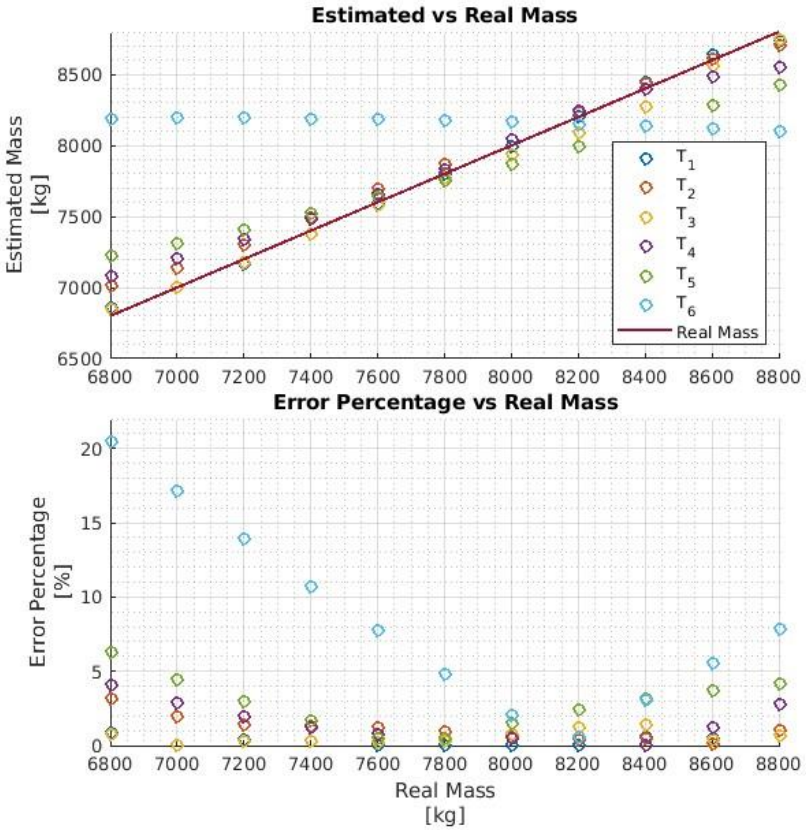
Test Results



0-1 Ton



0-2 Ton



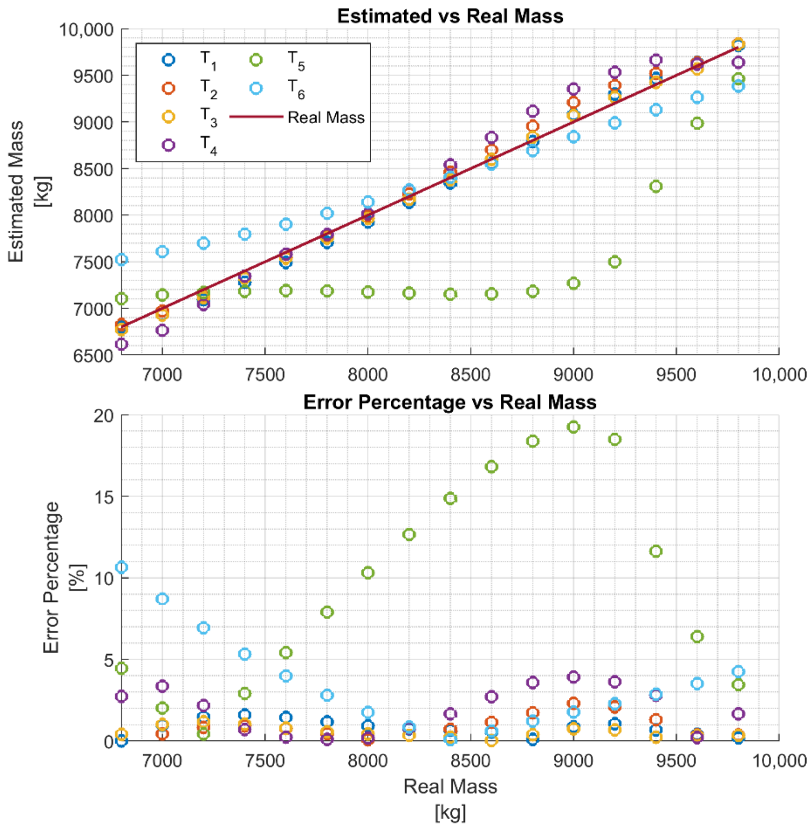
6 x 10s time intervals

Error <5 % during acceleration

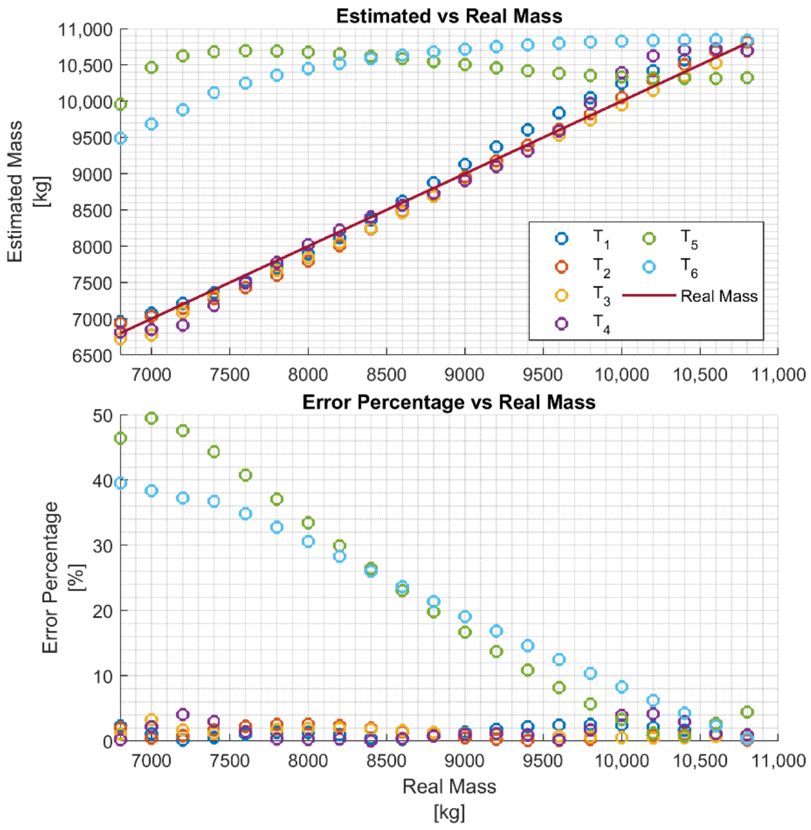
Deceleration occurs if accelerator pedal position is less than approximately 15% (depends on load)

Test Results

0-3 Ton



0-4 Ton



Conclusion and Future Work

- Quality of training data and tuning the hyperparameters
- LSTM network with 3000 parameters
- Error ±5% within a load up to 4 tons
- 11 hours of training data from TruckMaker
- Comparable to sensors during acceleration maneuvers

- Real Truck Data
- Road Gradient
- Wind Effect
- All gears

Image created by Dall-E

Conclusion and Future Work

- Quality of training data and tuning the hyperparameters
- LSTM network with 3000 parameters
- Error ±5% within a load up to 4 tons
- 11 hours of training data from TruckMaker
- Comparable to sensors during acceleration maneuvers

- Real Truck Data
- Road Gradient
- Wind Effect
- All gears

Image created by Dall-E