Task B13

Group 28

Chavan, Vivek; Chi Shing, Li; Küpeli, Ahmet

IAM Institut für Allgemeine Mechanik

RWTH AACHEN UNIVERSITY

Motivation, Aim and Preprocessing

Motivation

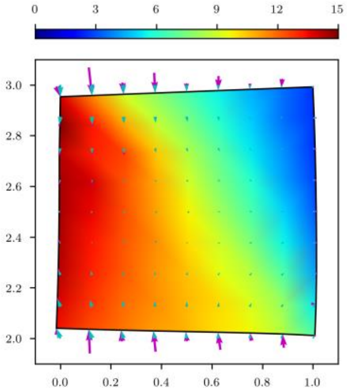
- Equation systems in realistic FEM Simulation can be extremely large → Massive computational resources
- Machine experience gained during the simulation is lost, input slightly changed → New Simulation needed!
- Machine learning models can be trained on data generated by conventional FEM tools. Wide range of other applications[1]

Project: Boundary Value Problem[2]

- Input: $\mathbf{S}_i$ [4], $\mathbf{U}_i$ [2]
- Output: $\mathbf{F}_i$ [2]
- Reference Architecture: **Time Distributed AlexNET**
- Own Architecture: **Time Distributed w/ Recurrent Layers (LSTM)**

Preprocessing

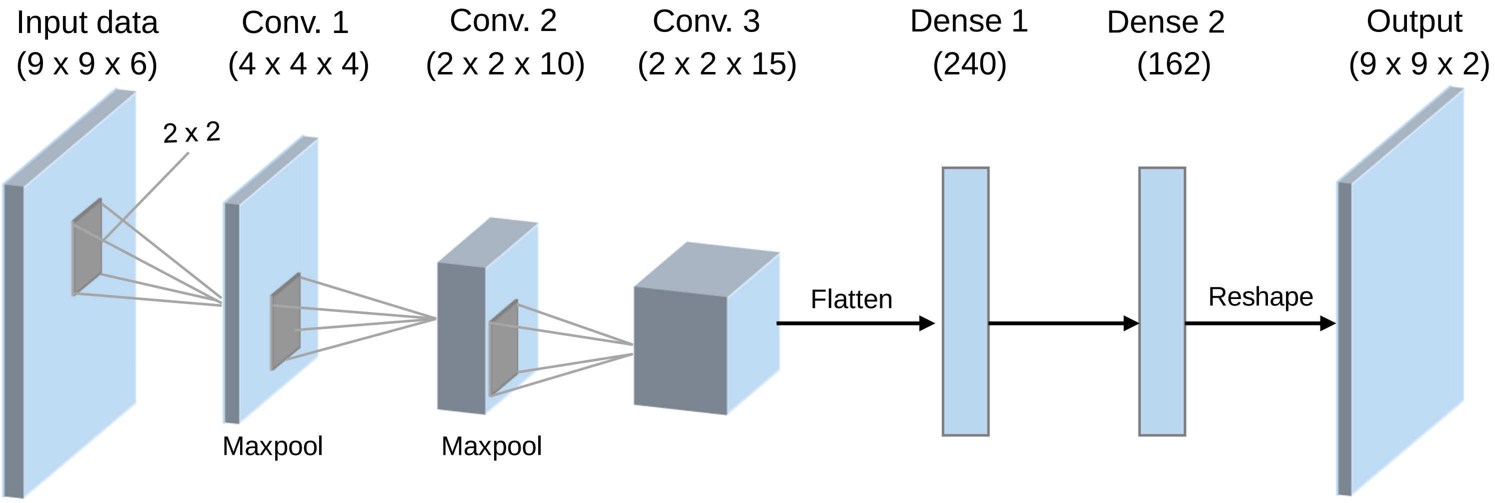
- Normalization using SD and Mean → (-3 to +3)



$$D = D(\mathbf{x}, t)$$

Time & Space Variant Data

Reference Architecture - Time Distributed AlexNET[3]



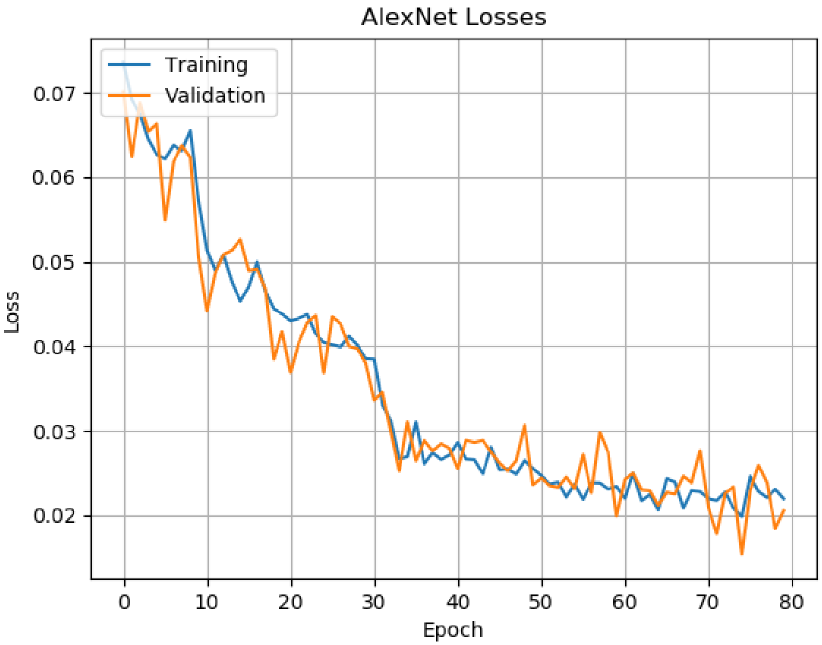
Model

- ➢ Padding = same[4]
- ➢ Activation: ReLU, linear
- ➢ Loss: mae
- ➢ Pool size: 2 x 2
- ➢ Stride = 1 x 1

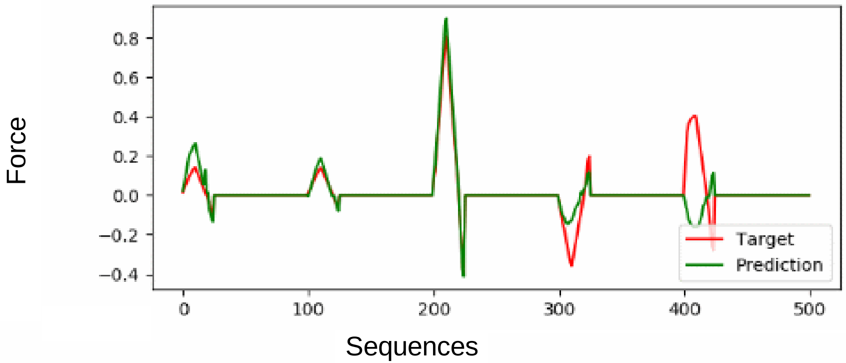
	16	**64**	**128**
0.001	0.021	0.049	0.028
0.01	0.073	0.073	0.073
0.1	0.084	0.082	0.083

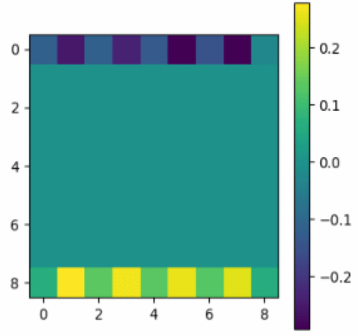
Test loss by batch size and learning rate for 80 Epochs

Reference Architecture - Time Distributed AlexNET^3

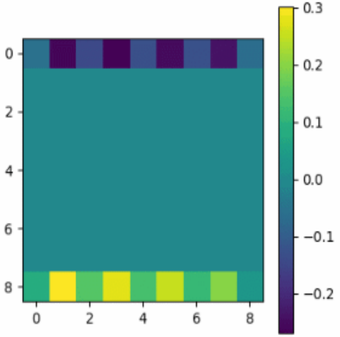


SMAPE = 0.1715%



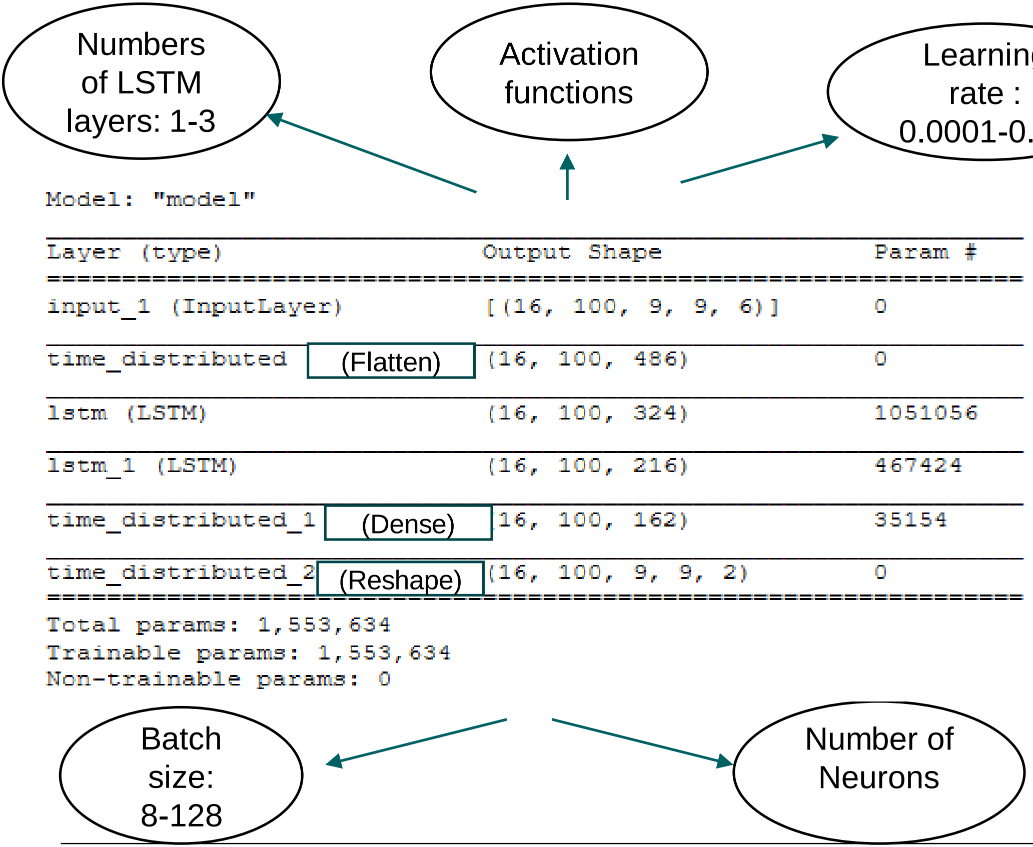


Target



Prediction

Own Architecture – LSTM RNN[5]



Numbers of LSTM layers: 1-3

Activation functions

Learning rate : 0.0001-0.01

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(16, 100, 9, 9, 6)]      0
_________________________________________________________________
time_distributed (Flatten)   (16, 100, 486)            0
_________________________________________________________________
lstm (LSTM)                  (16, 100, 324)            1051056
_________________________________________________________________
lstm_1 (LSTM)                (16, 100, 216)            467424
_________________________________________________________________
time_distributed_1 (Dense)   16, 100, 162)             35154
_________________________________________________________________
time_distributed_2 (Reshape) (16, 100, 9, 9, 2)        0
=================================================================
Total params: 1,553,634
Trainable params: 1,553,634
Non-trainable params: 0
```

Batch size: 8-128

Number of Neurons

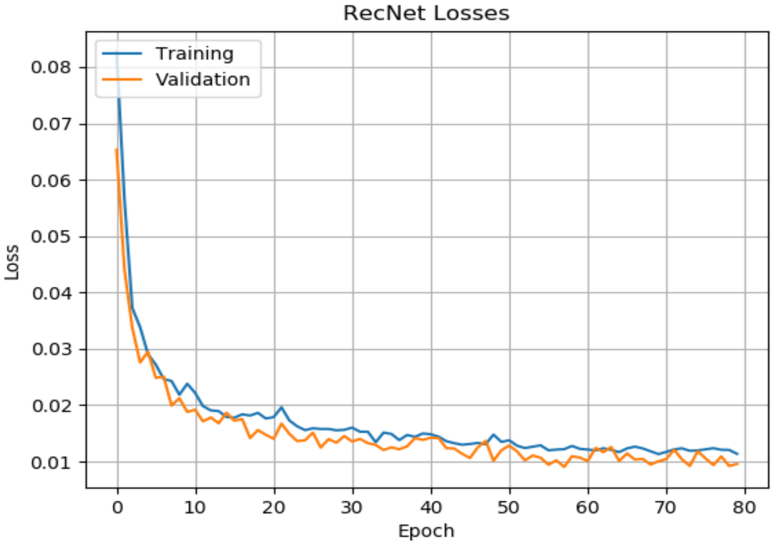
Model:

- ➢ LSTM layer:
activation = tanh,
returnsequence = true,
Dropout = 0.2
- ➢ TimeDistributed Dense:
activation = linear
- ➢ Loss: mae

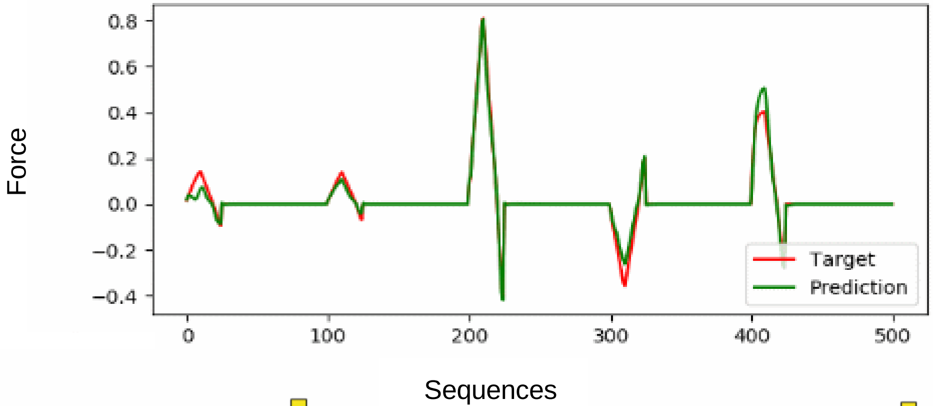
	8	16	32	64
0.0001	0.0116	0.0137	0.0171	0.0220
0.001	0.0095	0.0093	0.0096	0.0111
0.01	0.0210	0.0189	0.0176	0.0153

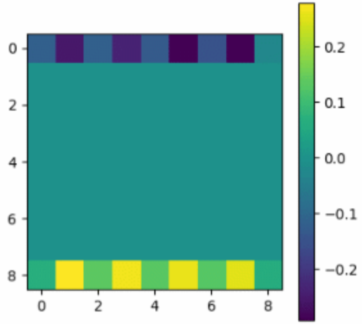
Test loss by batch size and learning rate for 80 Epochs

Own Architecture – LSTM RNN

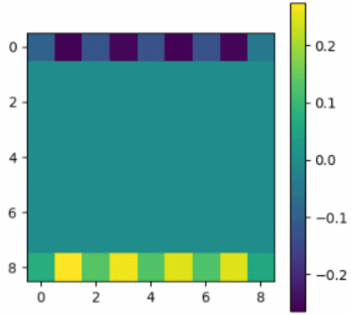


SMAPE = 0.1019%





Target



Prediction

Discussion and Conclusion

Inferences:

- **SMAPE(AlexNET) = 0.1715%**
- **SMAPE(RNN) = 0.1019%**
- Comparing the SMAPE values and the predicted loads, it is clear that **RNN (own architecture) performs better and is more suitable for our task.**
- Slight underprediction for both networks (due to empty sequences!)
- Large dataset helped in increasing the efficacy of training

Further Work:

- Mask the empty sequences before training the models
- Test out other RNN architectures (e.g.: GRU).
- Further tuning of hyperparameters

Bibliography

1. Kononenko O, Kononenko I, 'Machine Learning and Finite Element Method for Physical Systems Modeling', 2018, https://arxiv.org/abs/1801.07337 [Accessed: 14 January 2020].
2. Lecture Notes: Computational Intelligence in Engineering.
3. A. Krizhevsky, I. Sutskever, and G. Hinton, 'Imagenet classification with deep convolutional neural networks', InNIPS, 2012
4. Tensorflow Documntation: https://www.tensorflow.org/api_docs/python/ [Accessed Jan. 2020]
5. Hochreiter S., Schmidhuber J., 'Long Short-term Memory ', Neural Computation, 9(8):1735-1780, 1997