
بسم الله الرحمن الرحیم

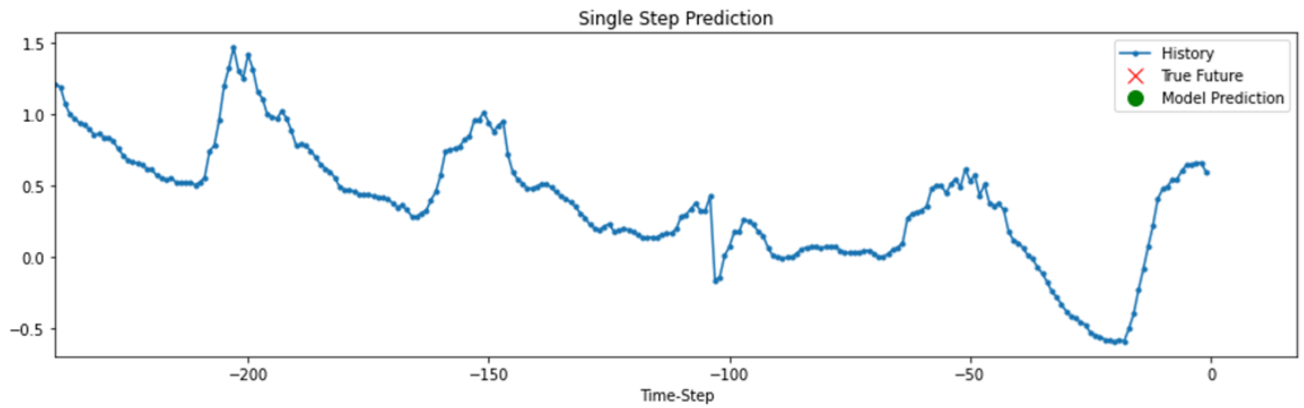
Deep Learning

Mohammad Reza Mohammadi

2021

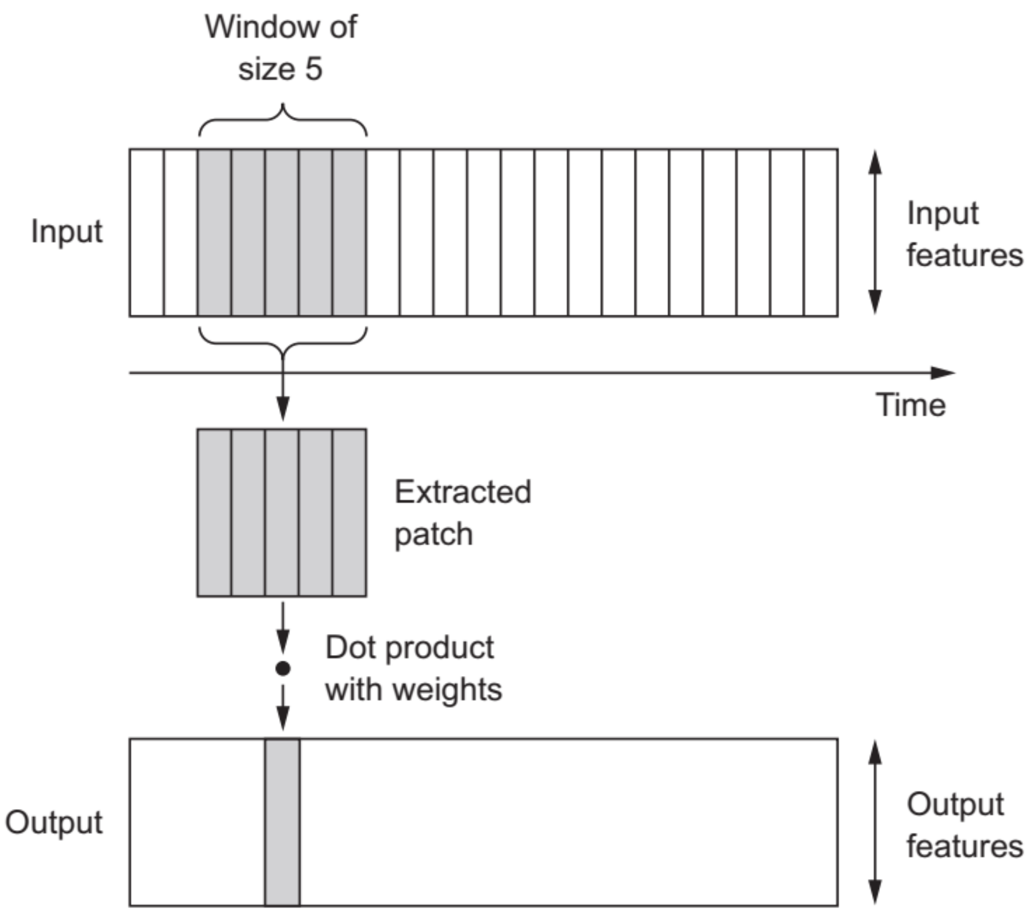
Sequence processing with convnets

- The same properties that make convnets excel at computer vision also make them highly relevant to sequence processing
- Time can be treated as a spatial dimension, like the height or width of a 2D image
- Such 1D convnets can be competitive with RNNs on certain sequence-processing problems, usually at a considerably cheaper computational cost



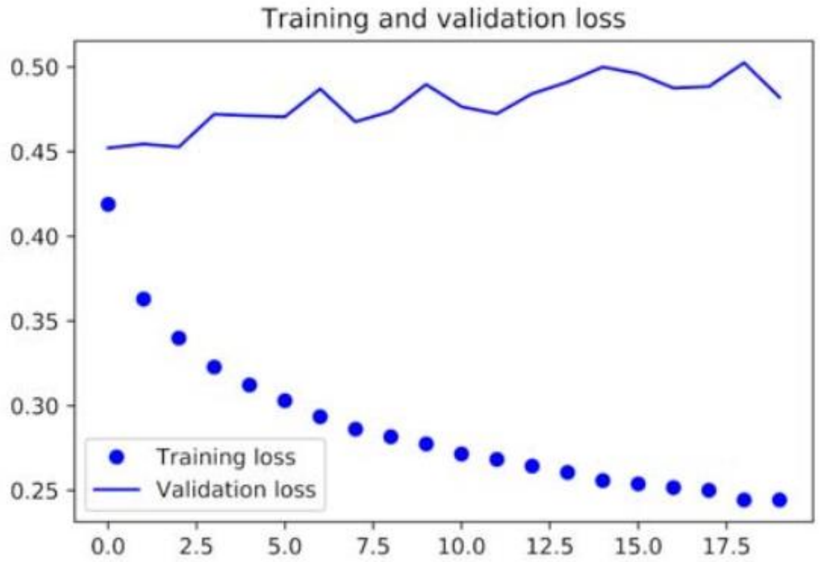
1D convnet

- 1D convolution layers can recognize local patterns in a sequence
- A pattern learned at a certain position in a sentence can later be recognized at a different position, making 1D convnets translation invariant
- We may use stride and 1D pooling



Combining RNNs with CNNs

- Because 1D convnets process input patches independently, they aren't sensitive to the order of the timesteps (beyond a local scale, the size of the convolution windows), unlike RNNs
- Temperature forecasting using 1D CNN



Listing 6.47 Training and evaluating a simple 1D convnet on the Jena data

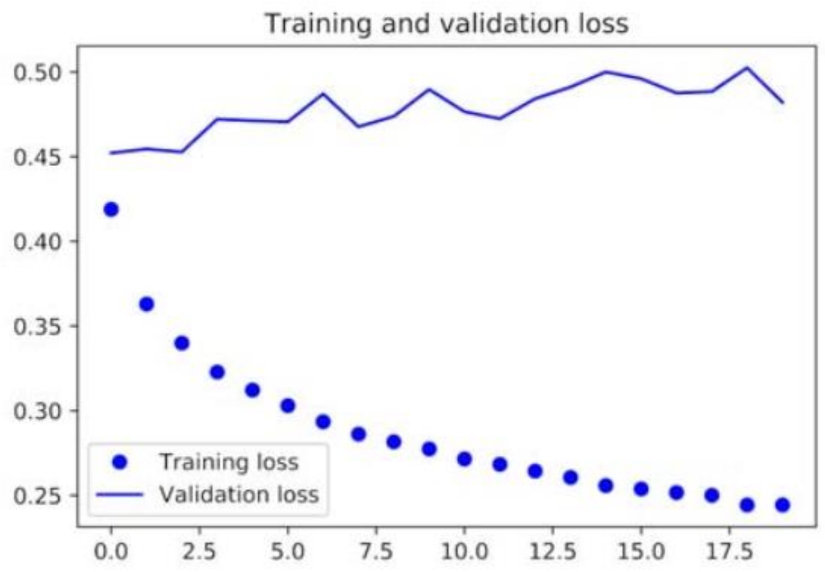
```
from keras.models import Sequential
from keras import layers
from keras.optimizers import RMSprop

model = Sequential()
model.add(layers.Conv1D(32, 5, activation='relu',
                        input_shape=(None, float_data.shape[-1])))
model.add(layers.MaxPooling1D(3))
model.add(layers.Conv1D(32, 5, activation='relu'))
model.add(layers.MaxPooling1D(3))
model.add(layers.Conv1D(32, 5, activation='relu'))
model.add(layers.GlobalMaxPooling1D())
model.add(layers.Dense(1))

model.compile(optimizer=RMSprop(), loss='mae')
history = model.fit_generator(train_gen,
                              steps_per_epoch=500,
                              epochs=20,
                              validation_data=val_gen,
                              validation_steps=val_steps)
```

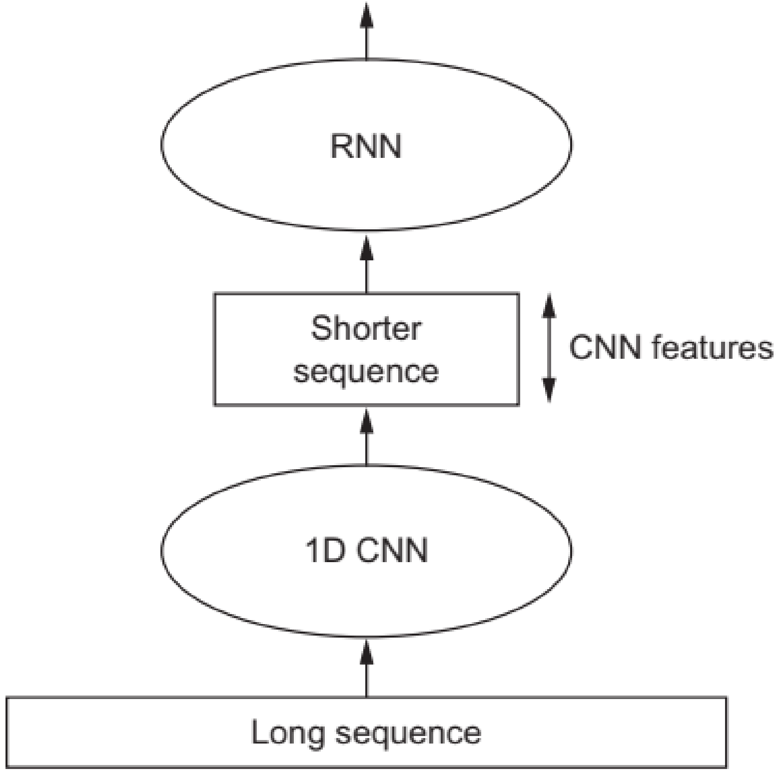
Combining RNNs with CNNs

- The validation MAE stays in the 0.40s: you can't even beat the common-sense baseline using the small convnet
- Because more recent data points should be interpreted differently from older data points in the case of this specific forecasting problem, the convnet fails at producing meaningful results



Combining RNNs with CNNs

- One strategy to combine the speed and lightness of convnets with the order-sensitivity of RNNs is to use a 1D convnet as a preprocessing step before an RNN
- Especially beneficial when you're dealing with sequences that are so long they can't realistically be processed with RNNs, such as sequences with thousands of steps



Combining RNNs with CNNs

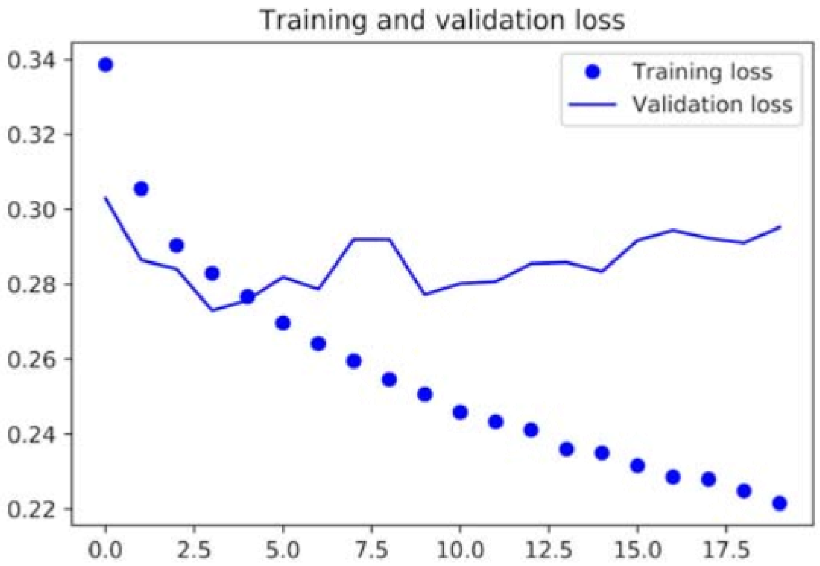
Listing 6.49 Model combining a 1D convolutional base and a GRU layer

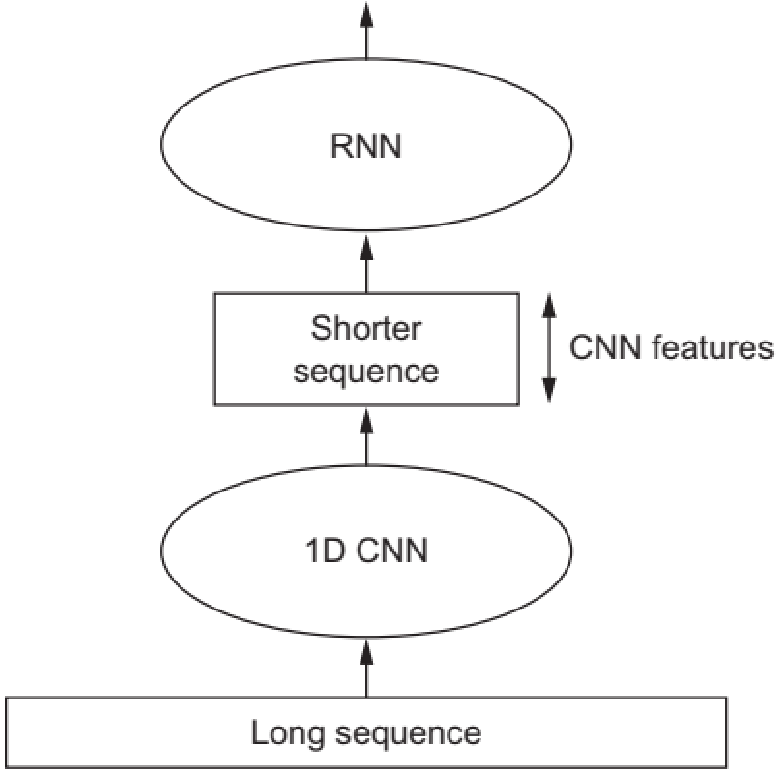
```
from keras.models import Sequential
from keras import layers
from keras.optimizers import RMSprop

model = Sequential()
model.add(layers.Conv1D(32, 5, activation='relu',
                        input_shape=(None, float_data.shape[-1])))
model.add(layers.MaxPooling1D(3))
model.add(layers.Conv1D(32, 5, activation='relu'))
model.add(layers.GRU(32, dropout=0.1, recurrent_dropout=0.5))
model.add(layers.Dense(1))

model.summary()

model.compile(optimizer=RMSprop(), loss='mae')
```

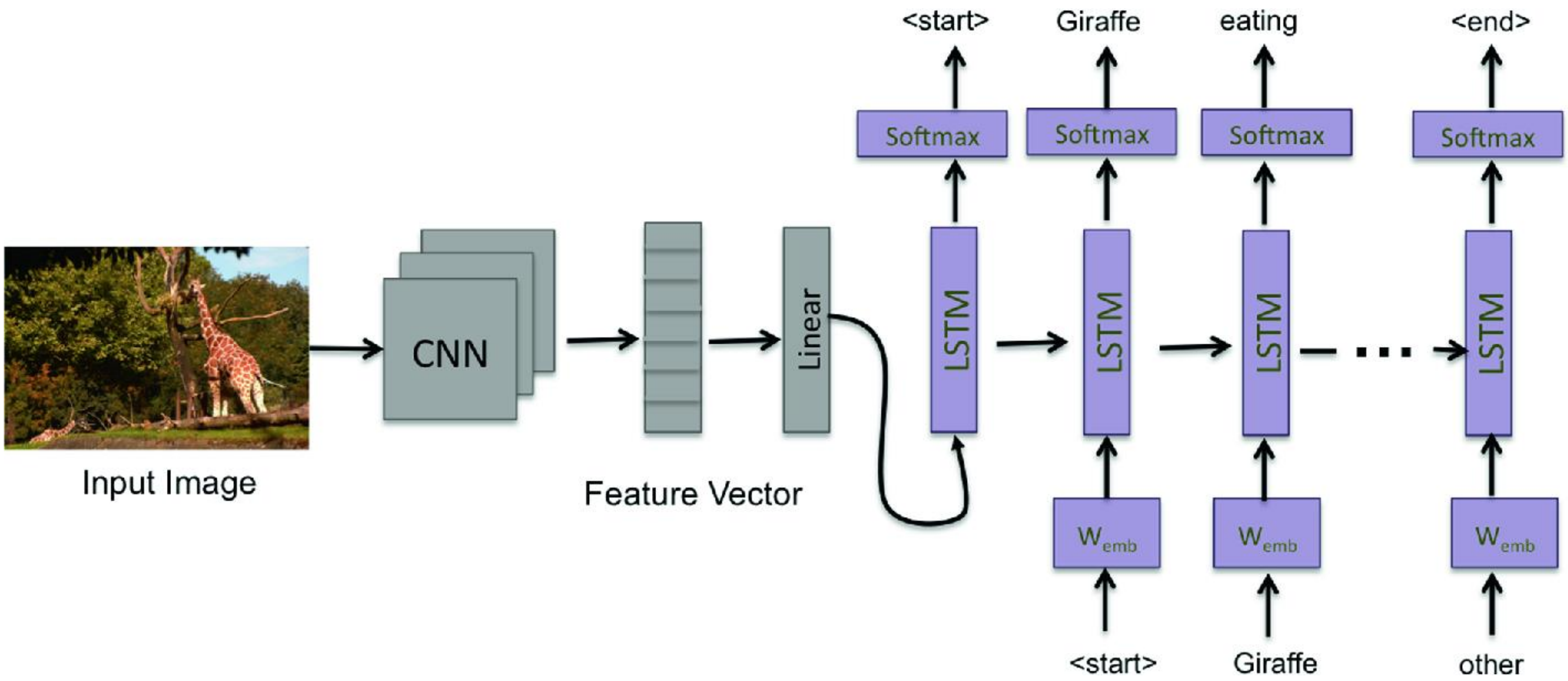




Combining RNNs with CNNs

- 1D convnets offer a faster alternative to RNNs on some problems, in particular natural language processing tasks
- Because RNNs are extremely expensive for processing very long sequences, but 1D convnets are cheap, it can be a good idea to use a 1D convnet as a preprocessing step before an RNN, shortening the sequence and extracting useful representations for the RNN to process

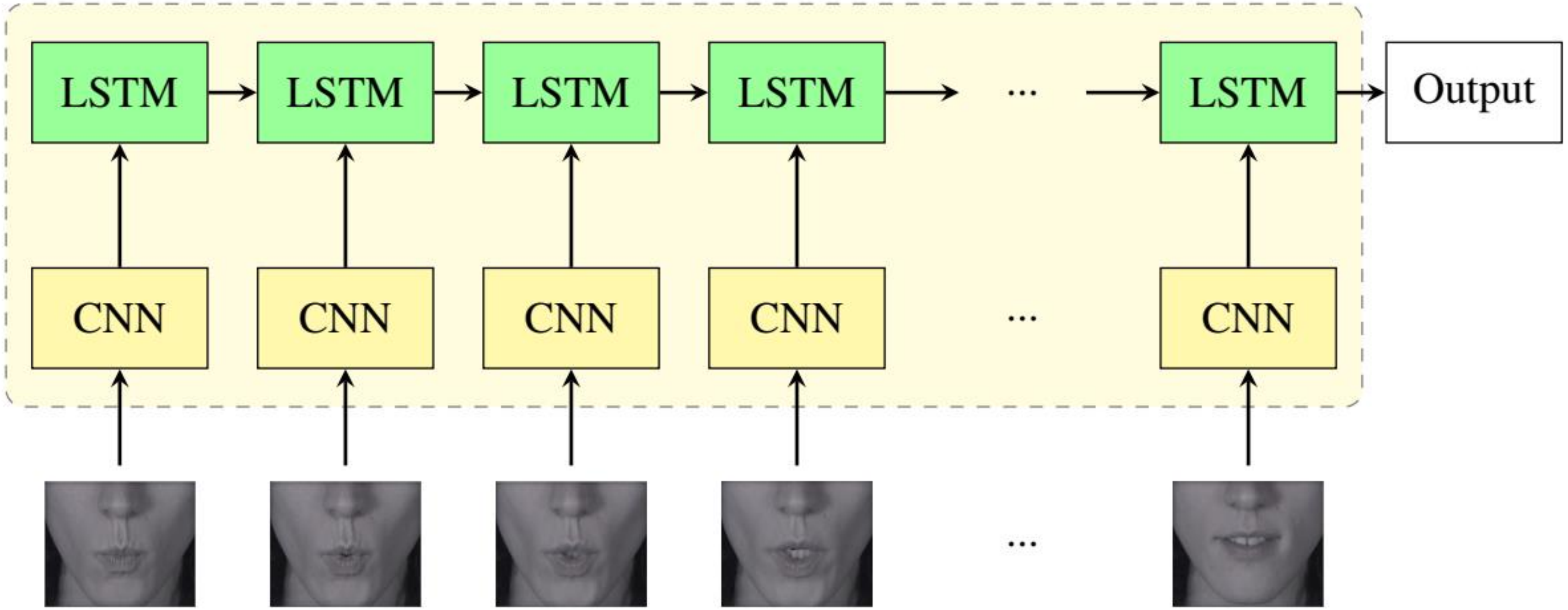
Image captioning with CNNs & RNNs



A. Tripathi, S. Srivastava, and R. Kothari. "Deep Neural Network Based Image Captioning." *International Conference on Big Data Analytics*. Springer, 2018.

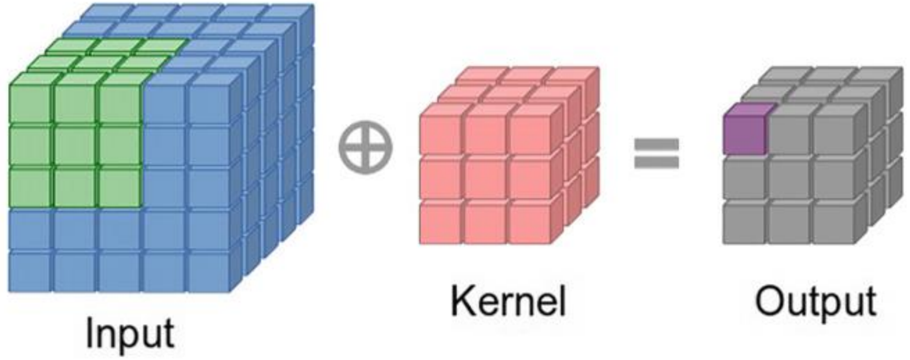
Video analysis with CNNs & RNNs

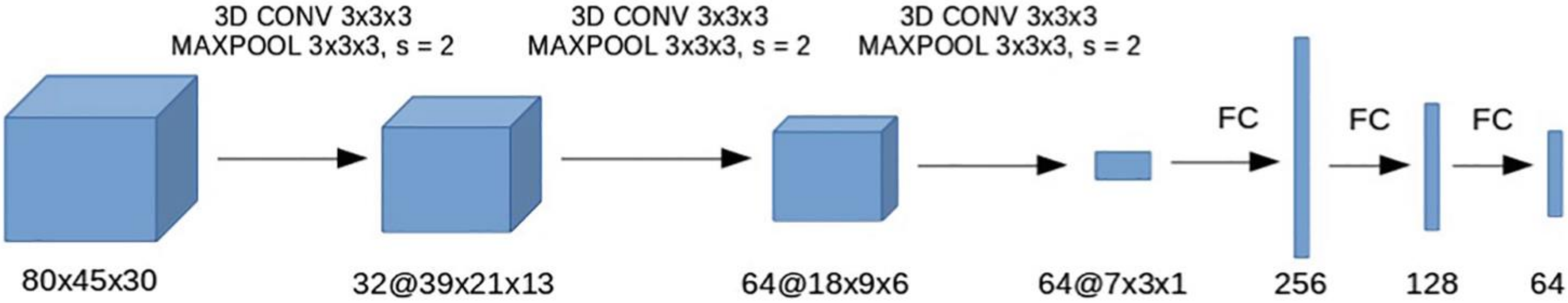
- Combinations of CNNs and RNNs (LSTMs or GRUs) stands out as the most used DL architecture for ALR



Fernandez-Lopez, Adriana, and Federico M. Sukno. "Survey on automatic lip-reading in the era of deep learning." *Image and Vision Computing* 78 (2018): 53-72.

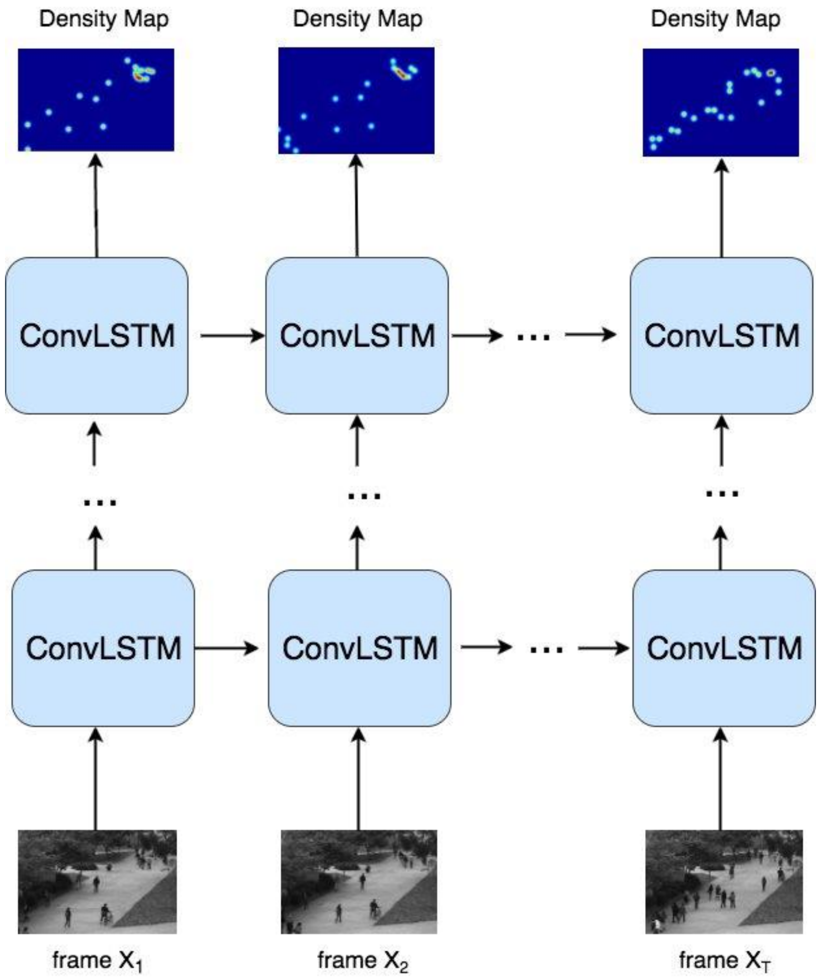
3D CNN

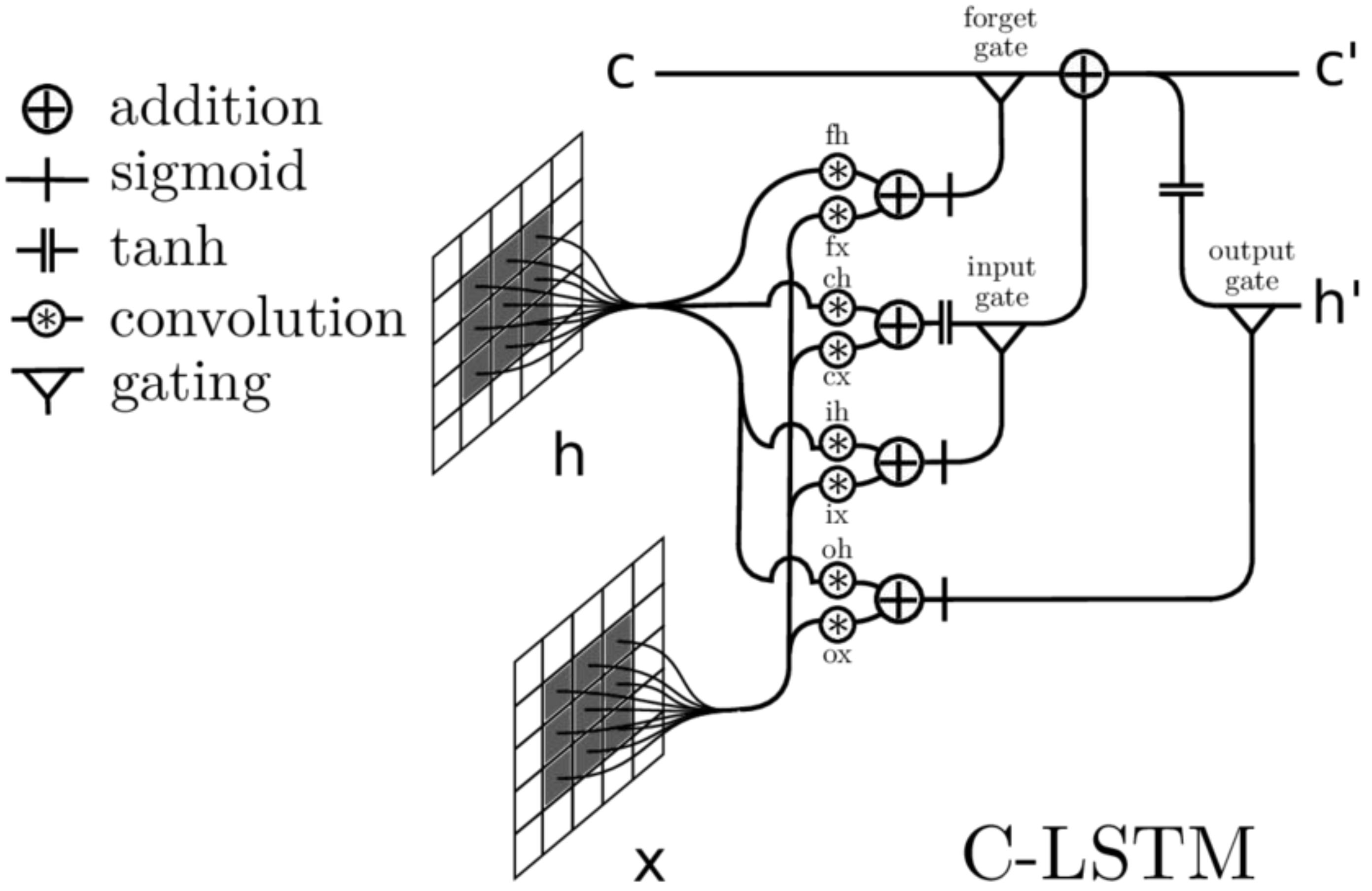




G. Neto, et al. "Sign language recognition based on 3d convolutional neural networks." *International Conference Image Analysis and Recognition*. Springer, 2018.

ConvLSTM



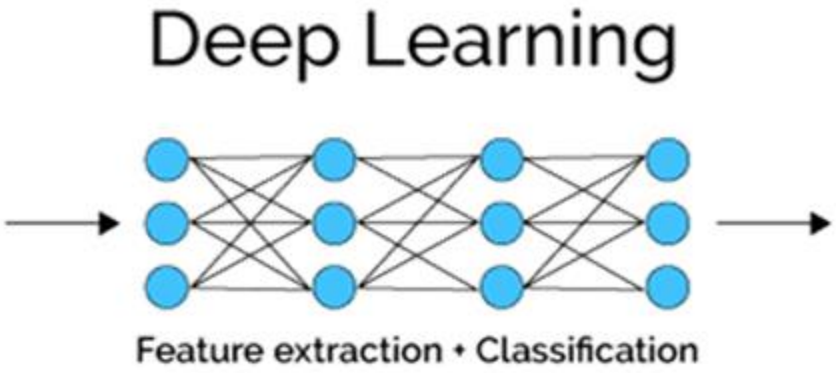


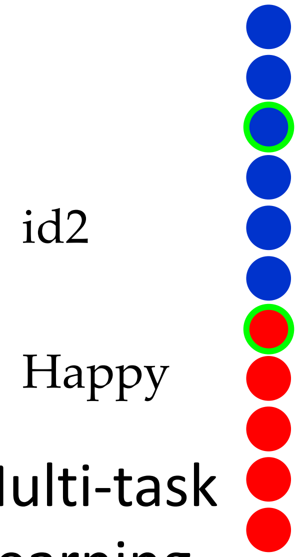
Stollenga, Marijn Frederik. *Advances in humanoid control and perception*. Diss. Università della Svizzera italiana, 2016.
F. Xiong, X. Shi, and D. Yeung. "Spatiotemporal modeling for crowd counting in videos." *IEEE International Conference on Computer Vision*. 2017.

Representation Learning

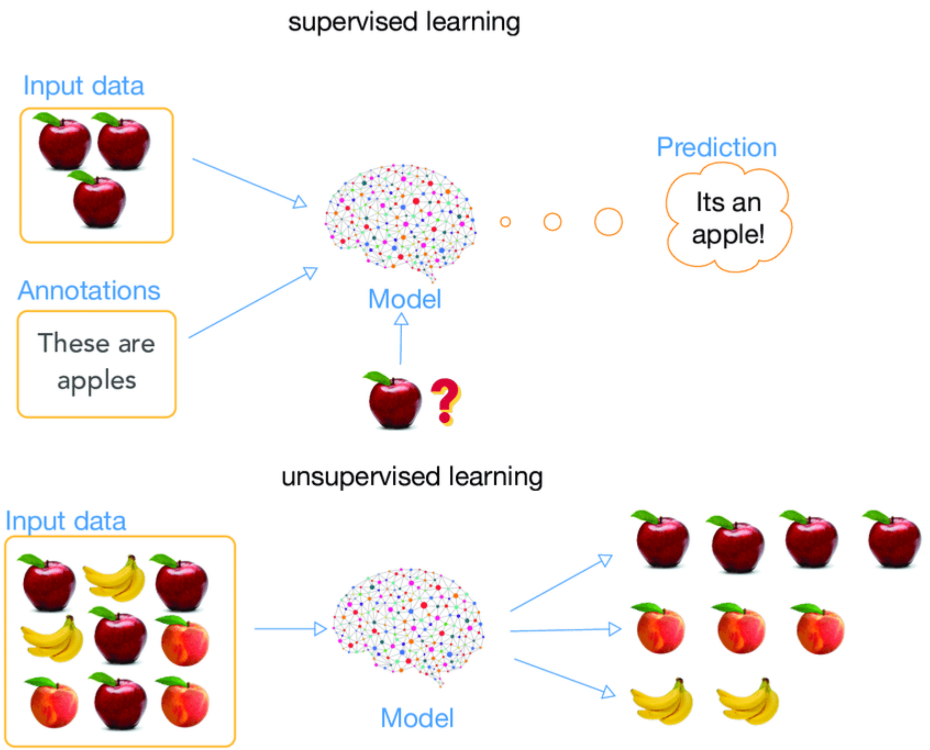
Representation learning

- How learning algorithms share statistical strength across different tasks?
  - Including using information from unsupervised tasks to perform supervised tasks
  - Shared representations are useful to handle multiple modalities or domains



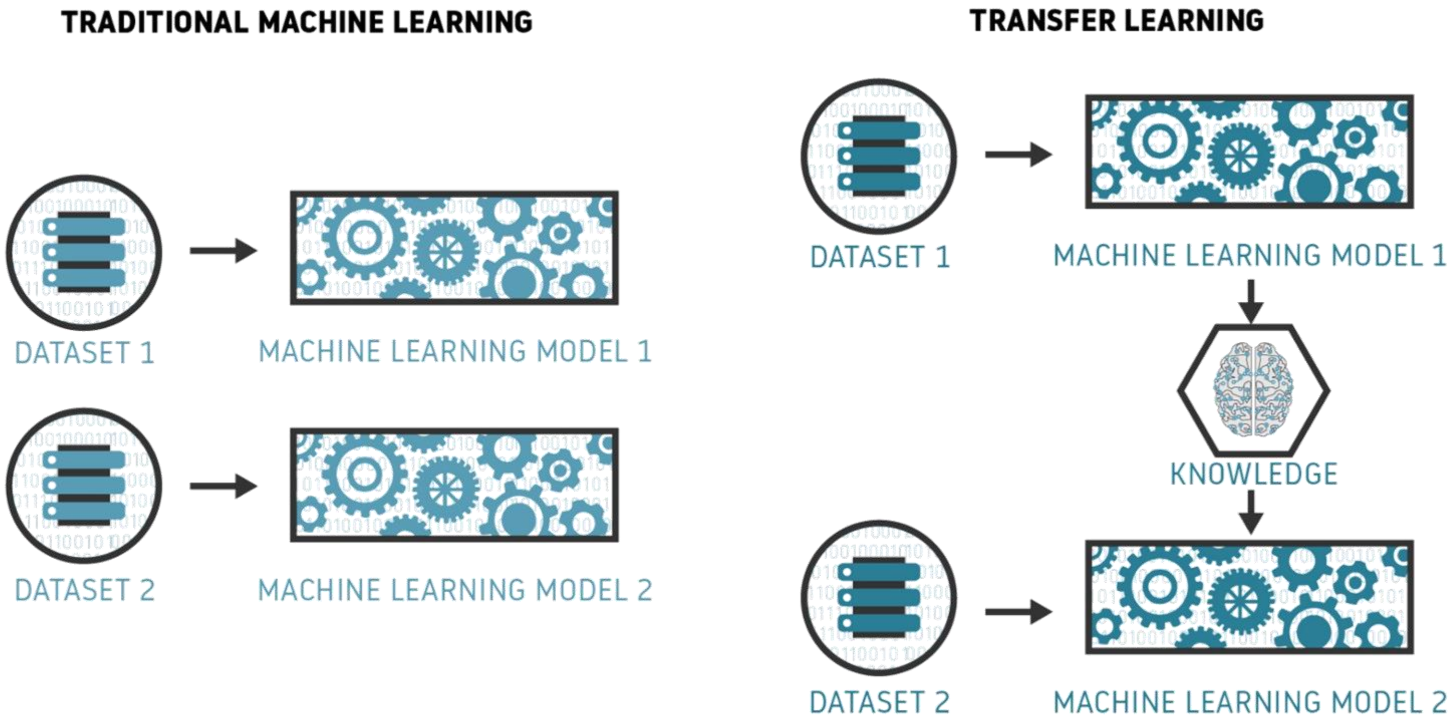


Multi-task learning



Representation learning

- How learning algorithms share statistical strength across different tasks?
  - Transfer learned knowledge to tasks for which few or no examples are given but a task representation exists



https://datascience.aero/transfer-learning-aviation/

Representation learning

- How learning algorithms share statistical strength across different tasks?
  - Including using information from unsupervised tasks to perform supervised tasks
  - Transfer learned knowledge to tasks for which few or no examples are given but a task representation exists
  - Shared representations are useful to handle multiple modalities or domains