Human Activity Recognition using Deep Learning techniques

Attila-Balázs Kis & Shunyi Deng

Institute for Parallel
and Distributed Systems

University of Stuttgart

18.07.2022

Motivation

- Human Activity Recognition (HAR)
  - active research field, signal data;
  - target: classify activity based on sensory input;
  - highly mobile environment.
- Human Activities and Postural Transitions Dataset (HAPT)
  - preprocessed dataset, multitude of features;
  - 12 classes, slightly unbalanced.
- Our solution
  - deep learning: neural networks;
  - research alternatives, find best solution;
  - understand network behavior, use to our advantage.

Table of Contents

Methodology
- Neural Networks
- Neural Network Layers
- Solution technical details

Training & Evaluation
- Process
- Results

Future work

Conclusions

Table of Contents

Methodology
- Neural Networks
- Neural Network Layers
- Solution technical details

Training & Evaluation
- Process
- Results

Future work

Conclusions

Neural Networks

- subset of machine learning, at heart of deep learning
- regression / `classification`
- iterative process of forward + backward propagation = "learning" (optimization)
- multitude of powerful layers, developed for different purposes
- regularization has powerful tools

Neural Network Layers

Fully Connected Layer:

- Theory
  - every node connected to every node of previous layer;
  - map inputs to different spaces, utilized for output shaping.
- Application
  - build baseline (FC), together with non-linear activation (tanh);
  - alter node number, decrease because of overfit concerns (FCs).

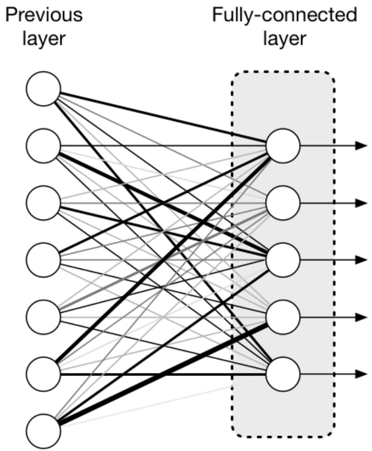

Figure: Fully connected layer

Neural Network Layers

Recurrent Layer:

- Theory
  - use previous & current inputs to produce output;
  - very powerful modern architectures, focusing on learning & forgetting optimally.
- Application
  - utilize LSTM architecture, 2 stacked layers;
  - let first linear layer to reshape input to lower space, since high amount of parameters;
  - *very important* sequence length.

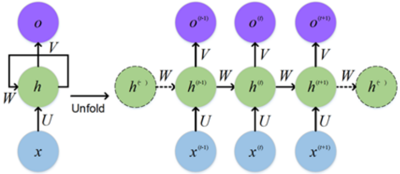

Figure: Recurrent layer

Neural Network Layers

Convolutional Layer:

- Theory
  - apply n-dimensional kernel over n-dimensional input to create convoluted output;
  - highly utilized in computer vision (2D), signal processing task alternative (1D).
- Application
  - use Conv1d as input layer, feeding output to recurrent layer;
  - important parameter kernel size, layer applies multiple kernels over input.

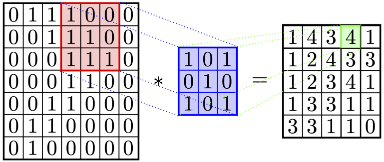

Figure: Recurrent layer

Solution technical details

The task presented required a great set of iterations in the *training & evaluation together with the model understanding* processes.

In the following list we will present the technical decisions taken in order to achieve the best possible result:

- use data sequencing for recurrent networks with *one-by-one* sliding;
- evaluate model output *many-to-many* manner;
- train for 150 epochs, *1e-4 learning rate*;
- use *Adam* optimizer, for it's proven speed and performance;
- use *CrossEntropyLoss*, advantageous for classification;
- use *Early Stopping* regularization in order to avoid overfitting.

Table of Contents

Methodology

- Neural Networks
- Neural Network Layers
- Solution technical details

Training & Evaluation

- Process
- Results

Future work

Conclusions

Process

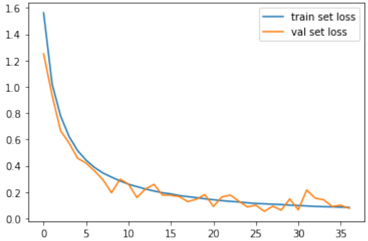

Figure: (FC) Early Stopping limiting overfitting

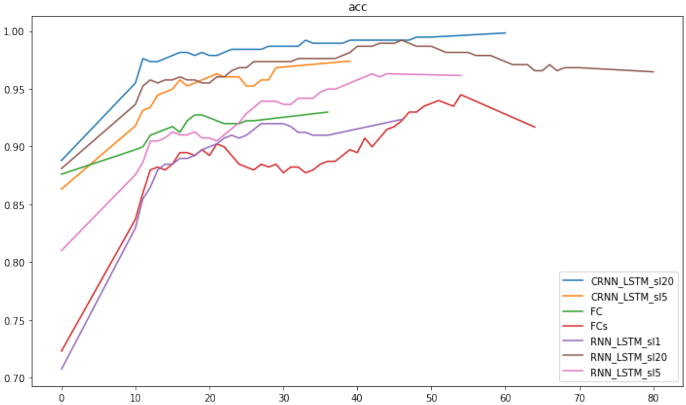

Figure: Accuracy plots over training

Process

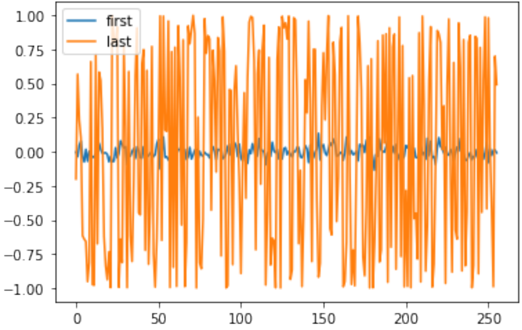

Figure: *Recurrent* layer states

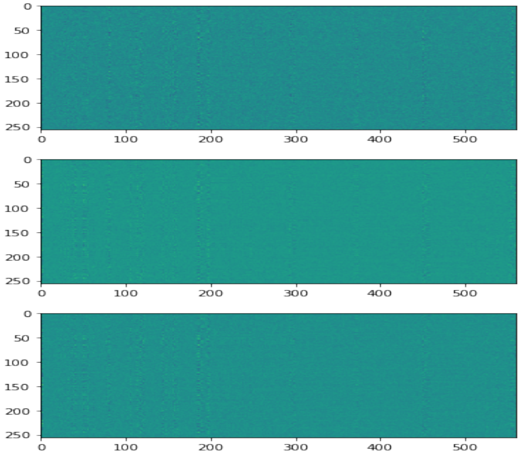

Figure: *Conv* layer weights

Results

Model Name	Acc	F1	AUC	#params	Comment
FC	91.96	91.95	91.01	146,956	baseline model
FCs	92.78	92.76	91.02	36,748	smaller hidden size
RNN_LSTM_sl1	91.49	91.42	88.15	1,199,628	sl = sequence length
RNN_LSTM_sl5	95.44	95.42	94.30	1,199,628	
RNN_LSTM_sl20	**97.23**	**97.21**	94.97	1,199,628	
CRNN_LSTM_sl5	95.82	95.78	94.07	1,486,860	C = convolutional
CRNN_LSTM_sl20	96.34	96.30	**95.43**	1,486,860	

Table of Contents

Methodology

- Neural Networks
- Neural Network Layers
- Solution technical details

Training & Evaluation

- Process
- Results

Future work

Conclusions

Future work

- Data
  - use less features from preprocessed data;
  - use raw signal data: preprocessing can combat unbalanced dataset manner.
- Network research
  - replace LSTM with GRU (hypothesis: not noticable enhancement);
  - powerful alternative to recurrent layers: transformers;
  - skip connections;

  - different optimizer & learning rate schedulers;
  - create new metric containing network size (mobile use-case).

Table of Contents

Methodology
Neural Networks
Neural Network Layers
Solution technical details

Training & Evaluation
Process
Results

Future work

Conclusions

Conclusions

- successfully tackled HAR task on given HAPT dataset
- developed baseline and incrementally enhanced it
- researched and worked with different neural network layers
- achieved maximum test accuracy of *97%*