FANNCortexM: An Open Source Toolkit for Deployment of Multi-layer Neural Networks on ARM Cortex-M Family Microcontrollers

Performance Analysis with Stress Detection

Michele Magno, Lukas Cavigelli,

Philipp Mayer, Ferdinand von Hagen, Luca Benini

IoT and Sensing Devices


Activity Trackers


Smartwatches

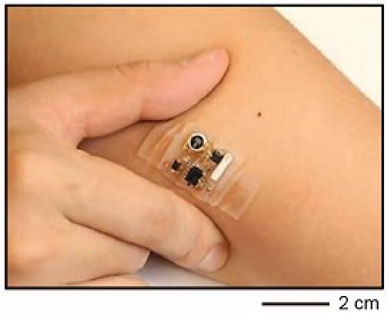

Smart eyewear


Smart footwear


Smart clothing


Motion capture clothing

Trend → Smaller, Smarter, Lifetime



Smart Patches


Skin patch sensors


Pressure sensor clothing


Healthcare

Wearable Device Limits and Challenges

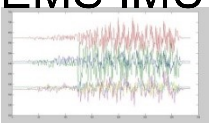
- Battery operated devices
  - Achieve long term monitoring
- Limited computational resources
  - Memory 1Mbyte range
  - Low power processing (i.e. Arm Cortex M4)
- On-board processing
  - Fast detection
  - No need for connectivity
  - Lower power consumption

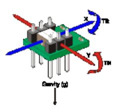
Source: planexta.com, artefactgroup.com, mimobaby.com

An IoT/Wearable System View

Sense

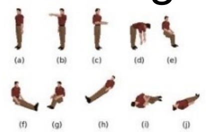
MEMS IMU



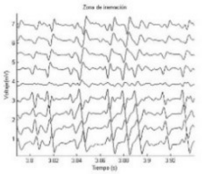


MEMS Microphone

ULP Imager



EMG/ECG/EIT



100 µW ÷ 2 mW

Analyze

Memory

e.g. Cortex-M

IOs

1 – 25 MOPS
1 – 10 mW

cm² Harvesting powered
→ mW (MaxP) + uW (AvgP)

Transmit

Short range, BW

Low rate (periodic) data

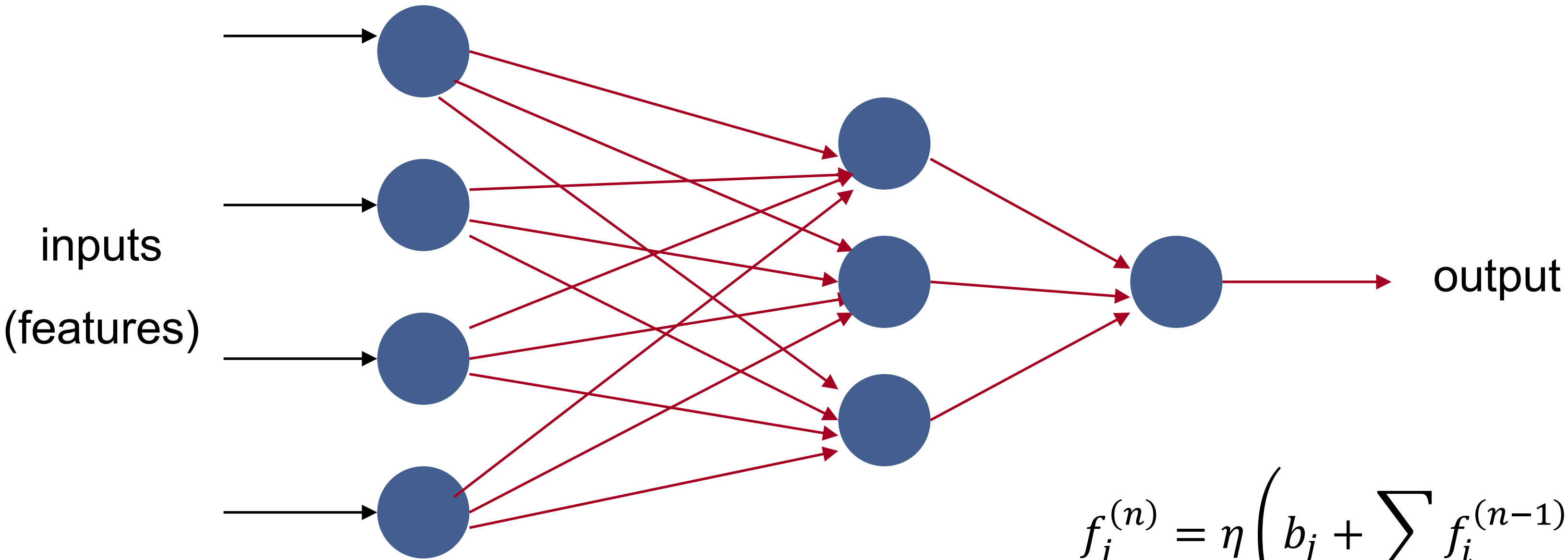
SW update, commands

Long range, low BW

LoRa SEMTECH

Idle: ~1µW
Active: ~ 10-100 mW

Artificial Neural Network



$$f_j^{(n)} = \eta\left(b_j + \sum_i f_i^{(n-1)} w_{j,i}^{(n)}\right)$$

classification

- marketing: consumer spending pattern
- defence: radar and sonar image analysis
- agriculture & fishing: fruit and catch grading
- medicine: ultrasound and ECG classification, EEGs, medical diagnosis

recognition and identification

- general computing and telecom: speech, vision and handwriting recognition
- finance: signature verification and bank note verification

Why now? – Platform and Data Availability

- Analysis in the cloud:
  - Energy hungry (data is local), privacy concerns, availability, cost
- Platforms are powerful enough

- sensor data + data analysis: use-cases
  - Fall detection for elderly
  - Health monitoring (livestock, pets)
  - Home security/control (motion detection, but not the cat)

- What is holding us back?
  - Cloud computing mindset
  - Complexity, availability of experts → simple tools needed!

Contributions

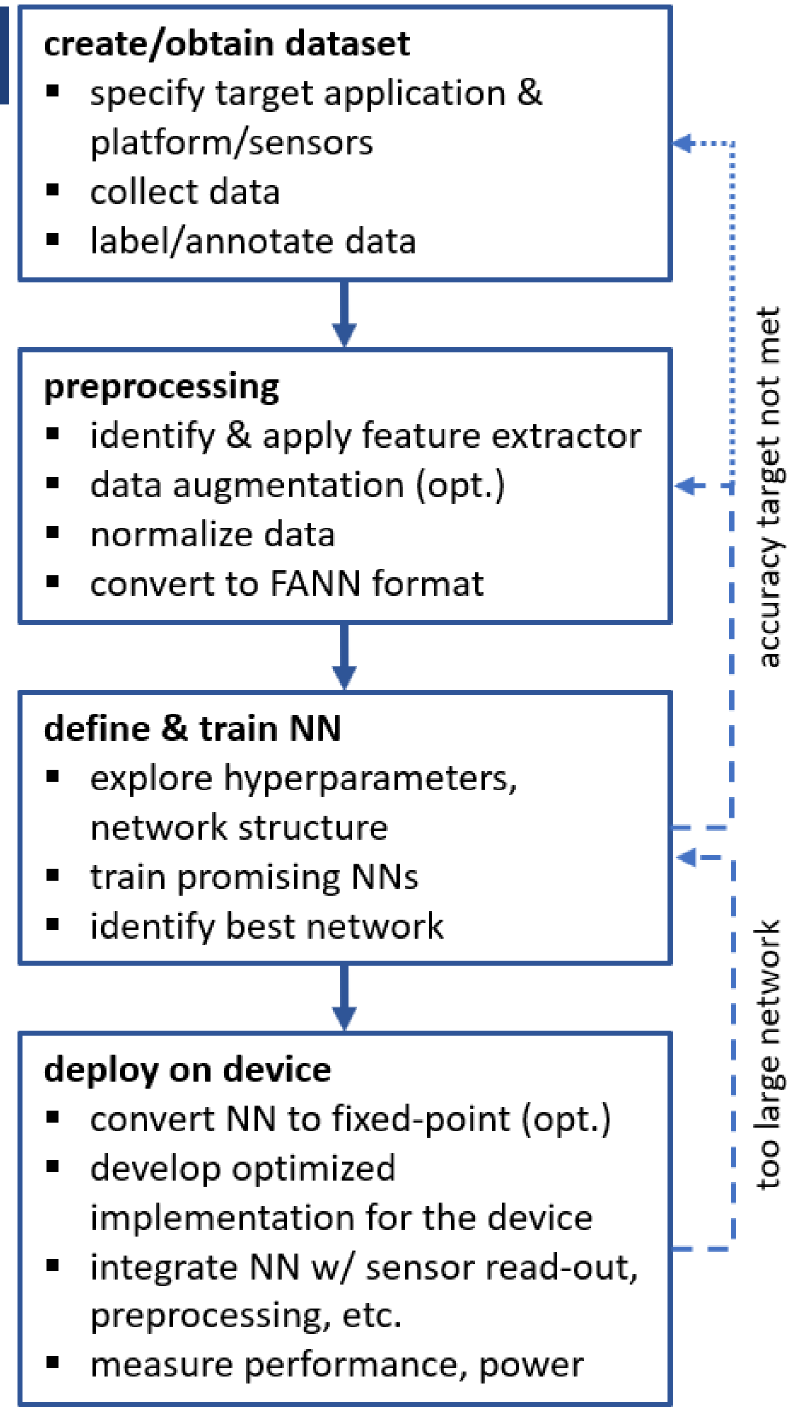
- Design of an optimized & easy-to-use framework for neural network inference on Cortex-M MCUs
  1. Acquire dataset
  2. Train NN with FANN [1] (easy: GUI, param search) – < 1h
  3. Deploy on MCU (w/o peripherals) – < 30 min.
- Evaluation with battery-operated low-power smart wearable device
  - Health monitor with the aim of emotion detection: ECG, Skin conductivity, temperature, IMU, microphone
  - Battery-powered, on-board processing, result communication
  - Performance, energy measurement on TI MSP432, Ambiq Apollo 2

[1] Nissen, S. (2003). Implementation of a fast artificial neural network library (fann). Technical Report.

Overview – Making It Easy

1. Creating dataset
   - Interface MCU to the sensors
   - Collect data samples & labels

2. Preprocessing/Augm.
   - Find good feature extractors (implementable on MCU)
   - (opt.) Data augmentation
   - Normalize data
   - Convert to FANN format (text file w/ input & output vals)

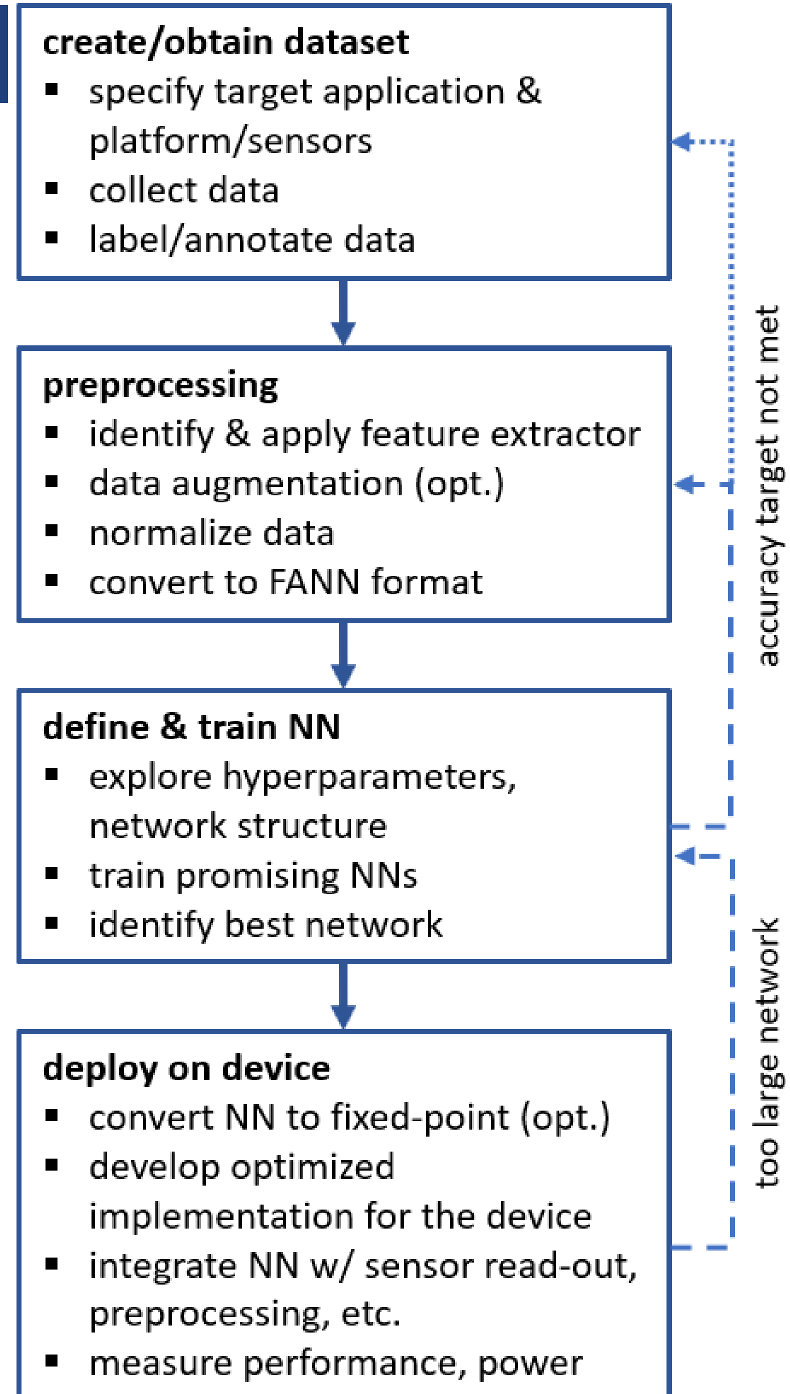


Overview – Making It Easy

3. Define & Train NN
   - Split dataset to train/val/test
   - Train NNs and adjust #layers, #hidden nodes, …
   - Select best network

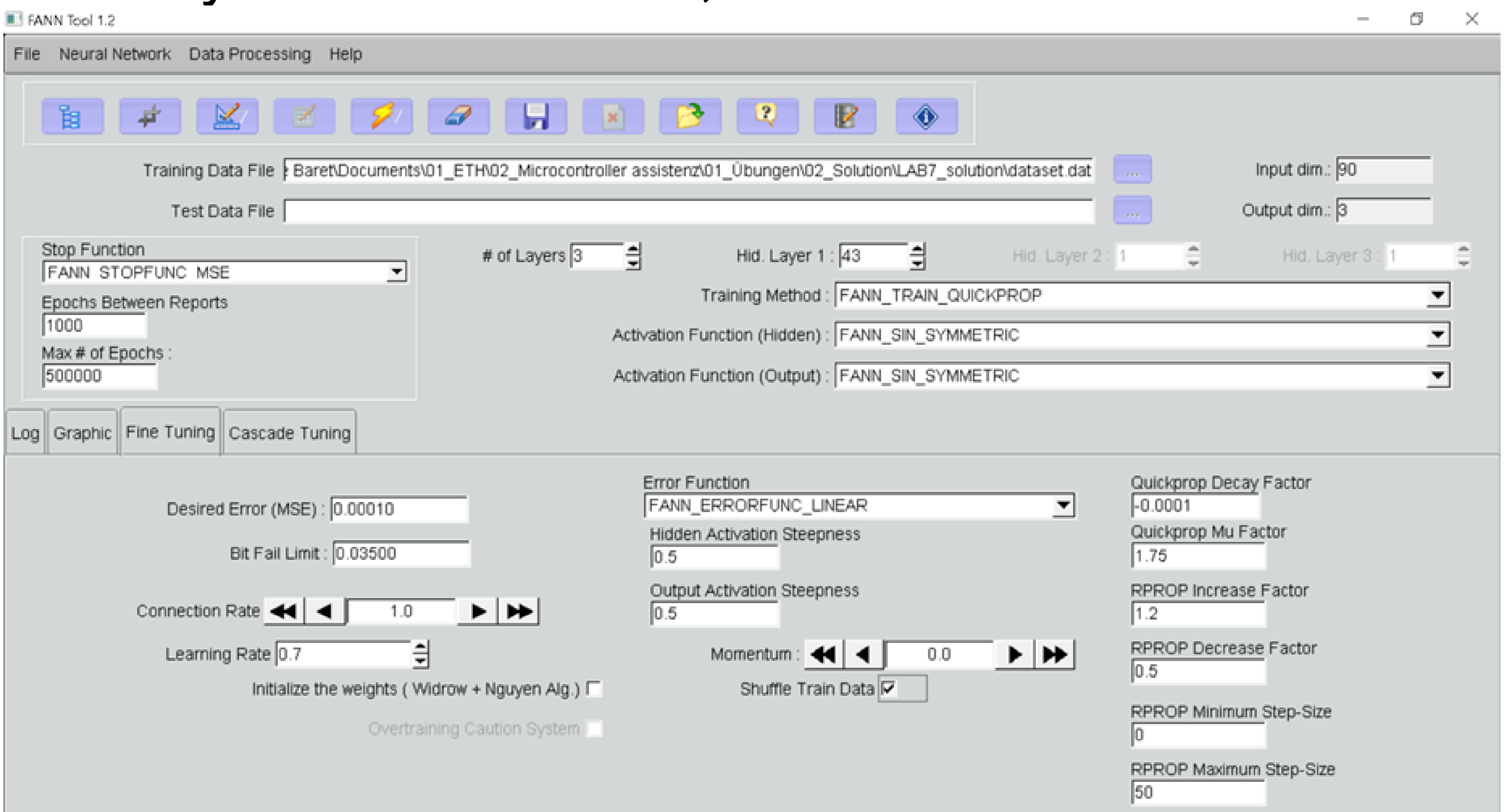
4. Deploy on Device
   - Run ‘generate.py’ to get ready-to-go C implementation

https://github.com/lukasc-ch/FANN-on-ARM



Fast Artificial Neural Network Library (FANN)

- Available in C, C++ PHP, Python, Delphi etc.
- Easy to learn and use; no GPU needed



FANN – Optimized Implementation for Cortex M

- 2x speed-up, smaller size (67 → 11 kB) v. non-opt. FANN
- Reduced code size
- Rely on CMSIS library for opt.
- Using FPU
- 50% energy savings

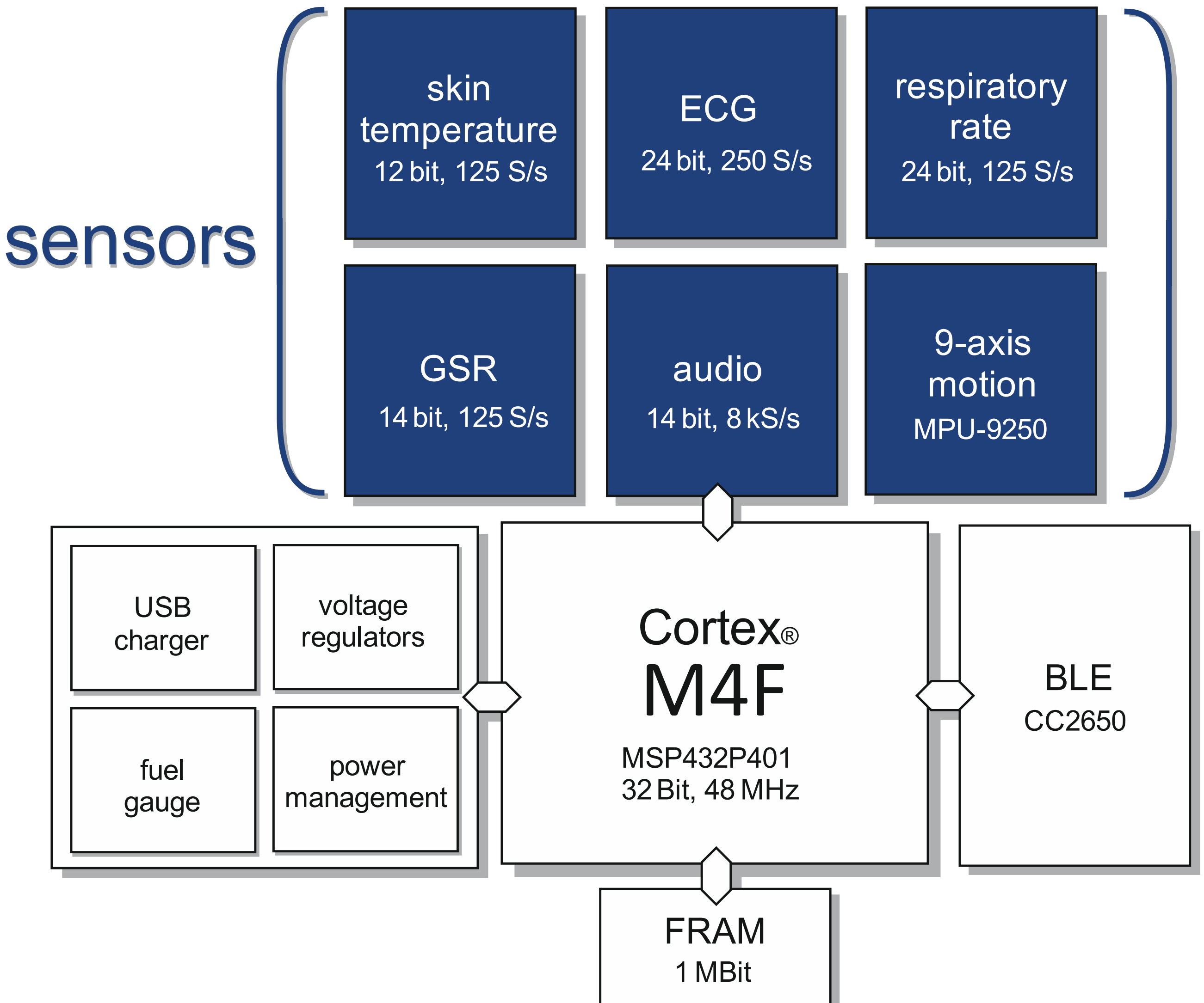
	12 MHz		24 MHz		48 MHz	
	T (ms)	E (µJ)	T (ms)	E (µJ)	T (ms)	E (µJ)
BASE	4.22	56.8	3.14	56.8	2.19	59.4
RAM	3.74	51.2	2.60	48.3	1.90	52.1
		−9.8%		−14.9%		−12.3%
RAMC	4.23	53.3	3.14	53.4	2.19	56.6
		−6.1%		−5.9%		−4.7%
RAMF	3.77	47.7	2.63	45.7	1.92	50.7
		−16.0%		−19.6%		−14.6%
OPT	3.53	46.8	2.55	45.1	1.78	48.2
		−17.7%		−20.5%		−18.8%
OPT + CMSIS	2.99	35.7	2.08	34.5	1.40	36.6
		−37.2%		−39.2%		−38.4%

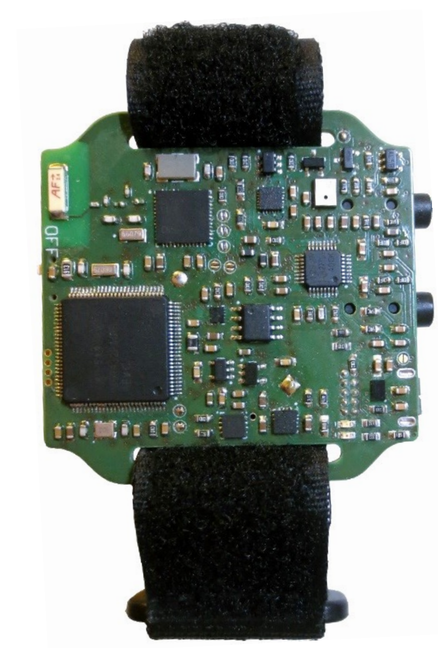
Case-study: Emotion Detection

- Training dataset: *Augsburg dataset*
  - Recorded ECG, EMG, GSR and respiration data of a subject in 4 emotional states on 25 different days
  - Classification into 3 stress levels

- Now go through the steps…
  - Step 1: data collection -- completed
  - Step 2: feature extraction, selection
  - Step 3: defining NN
  - Step 4: deployment & evaluation

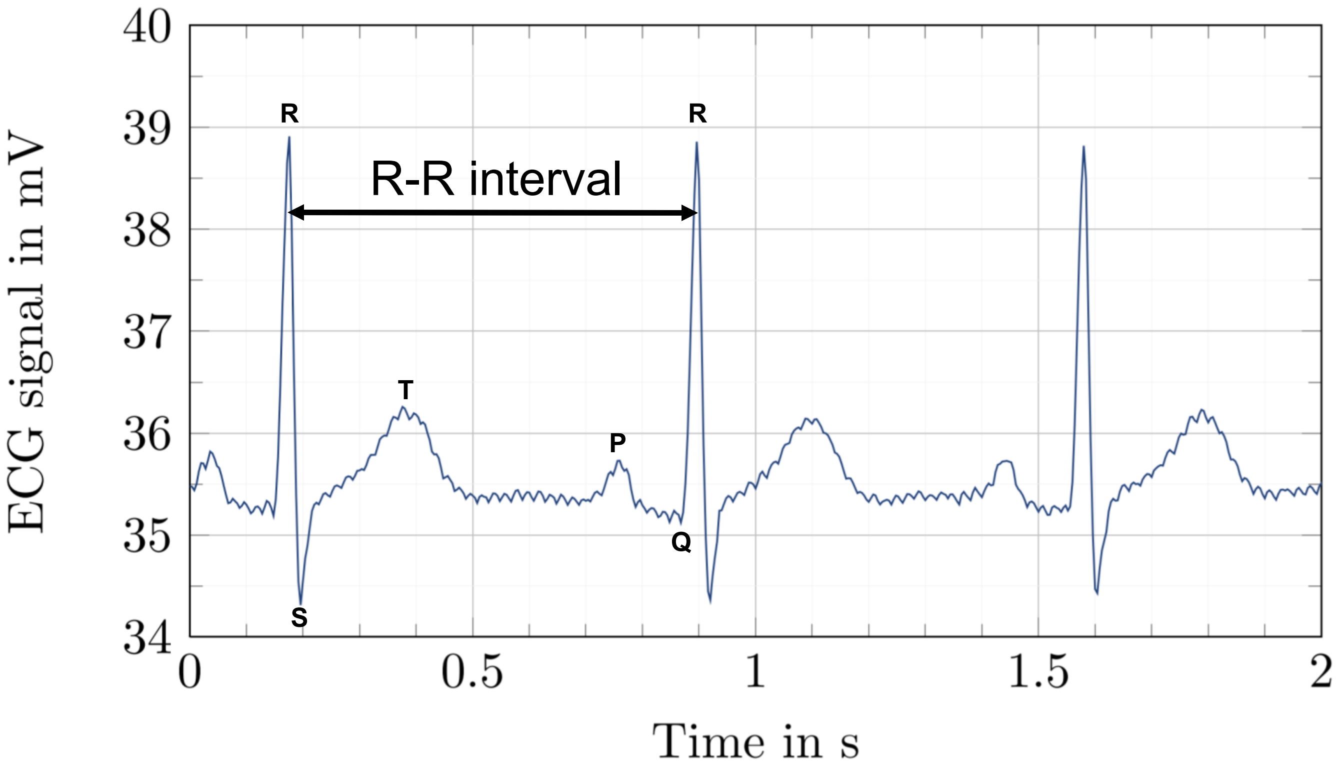
Hardware Overview



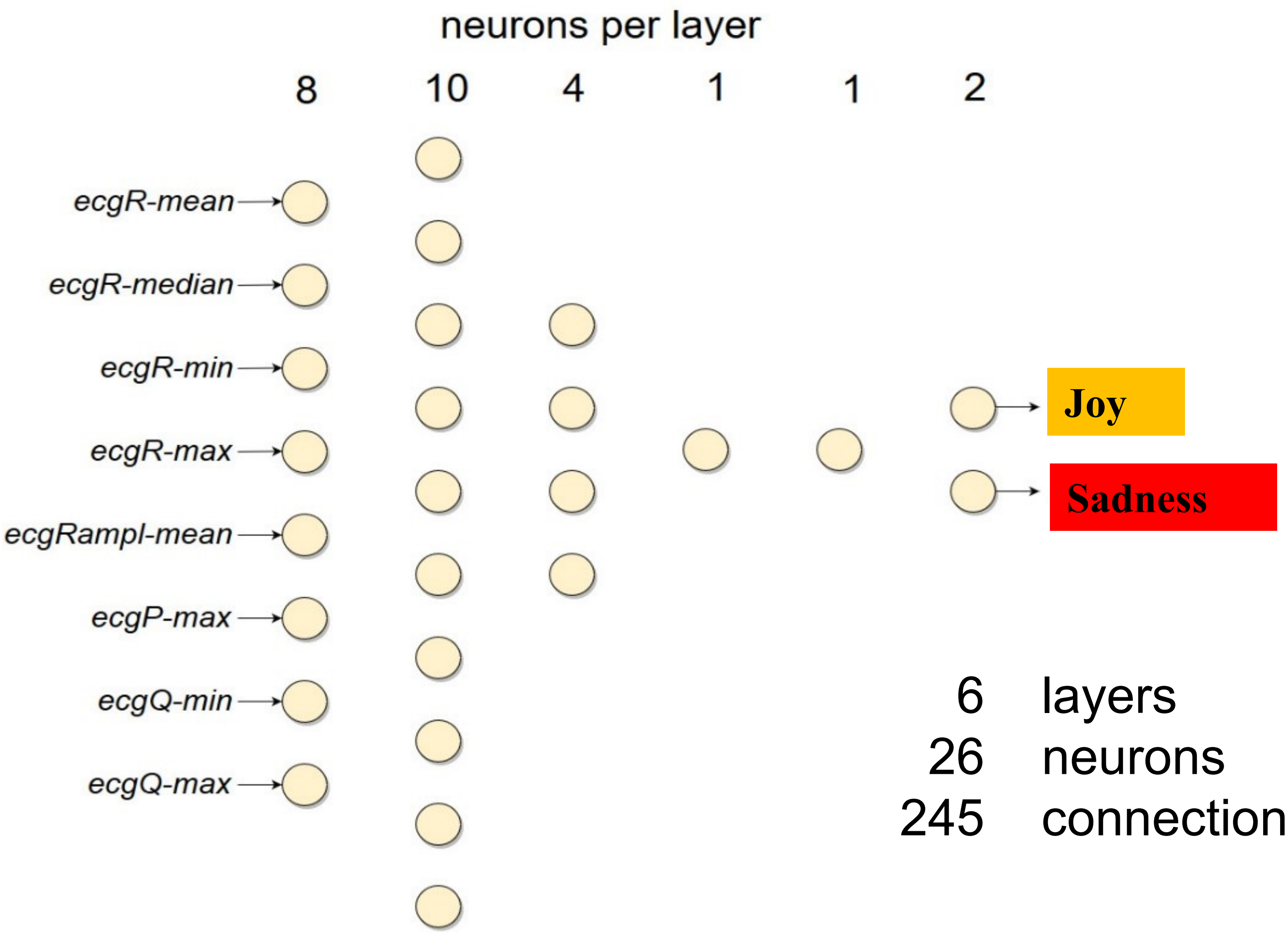


ECG Feature Extraction

Statistical features: mean, median, min, max of R-R interval etc.

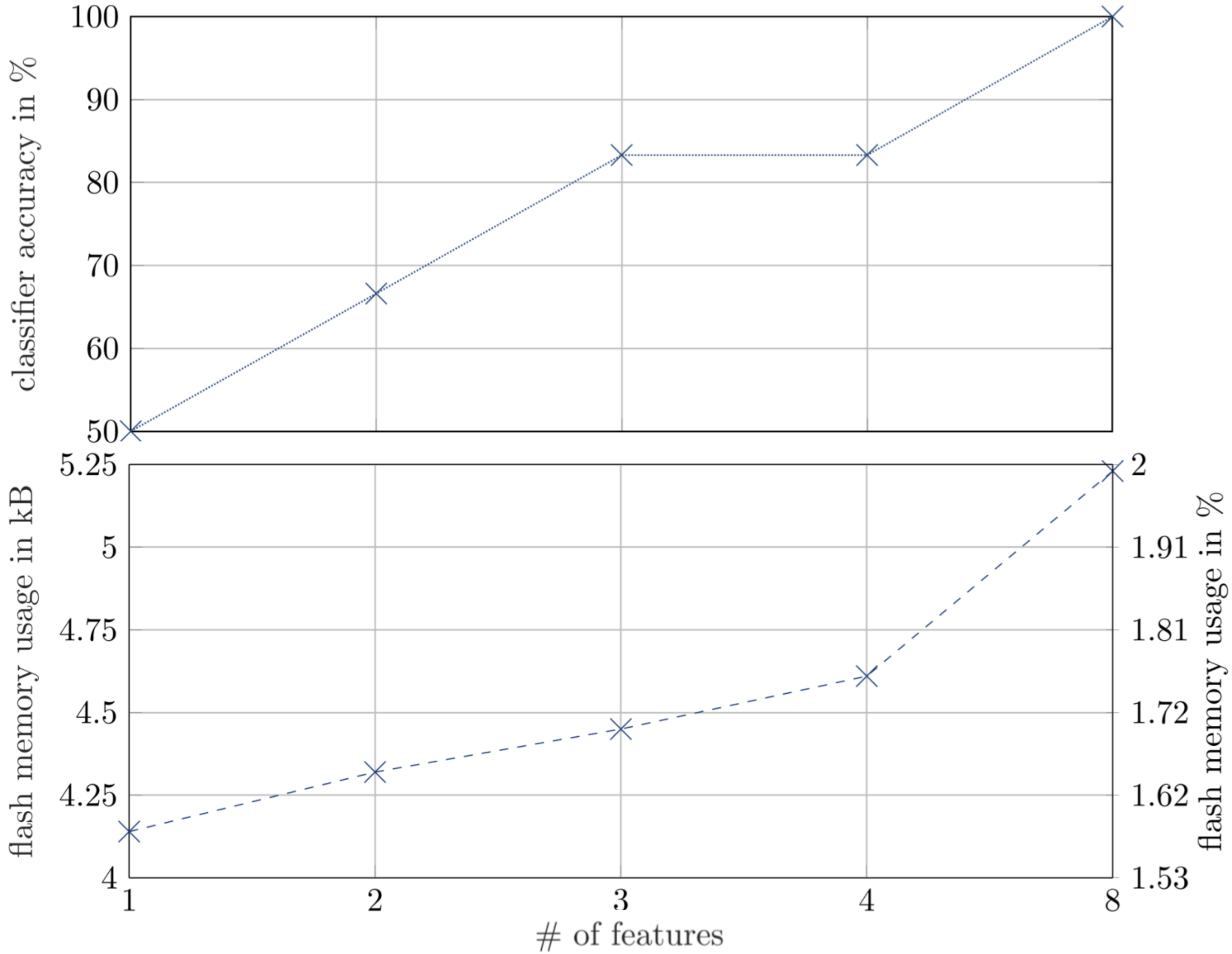


Network structure: Emotion Detection

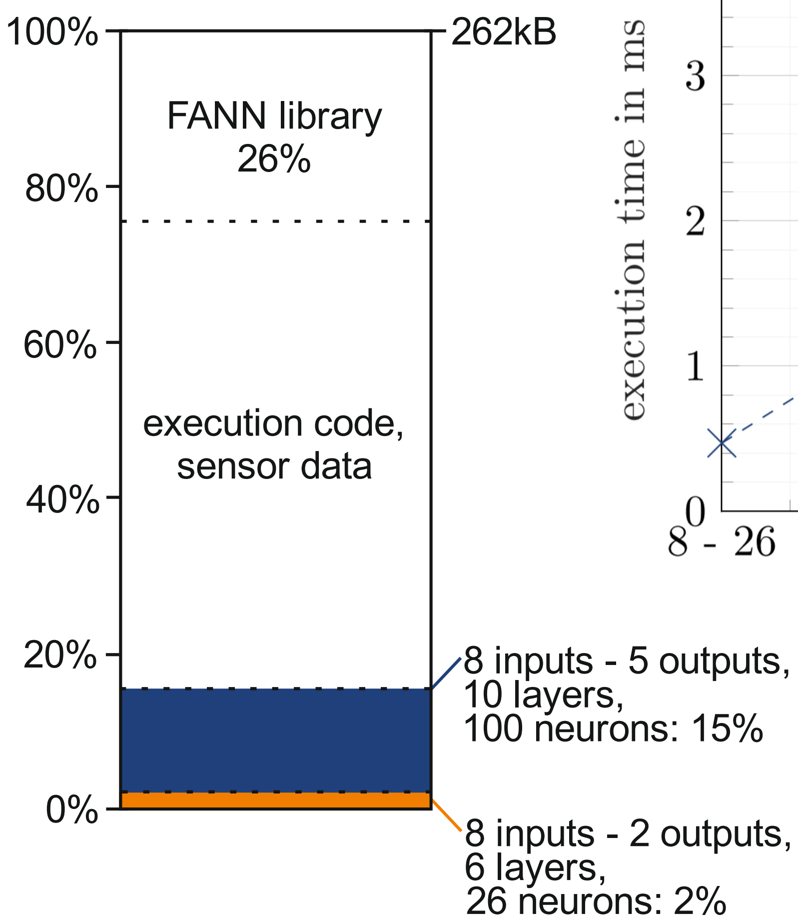


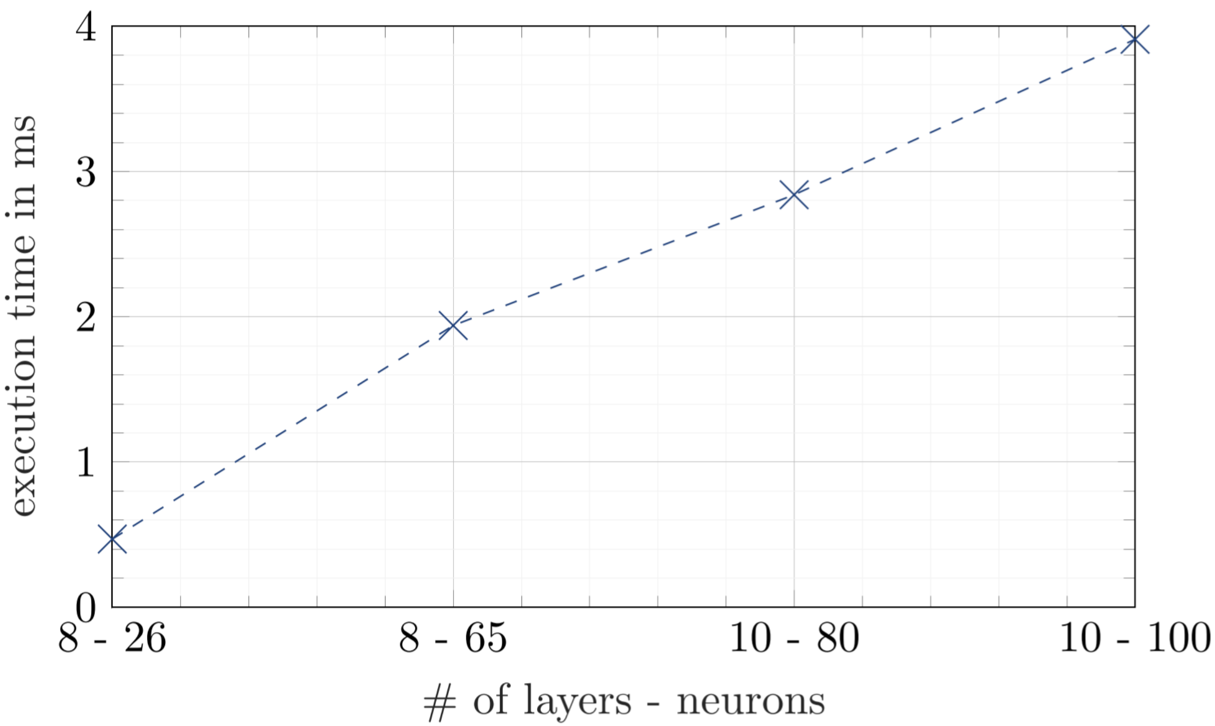
6	layers
26	neurons
245	connections

Feature Selection



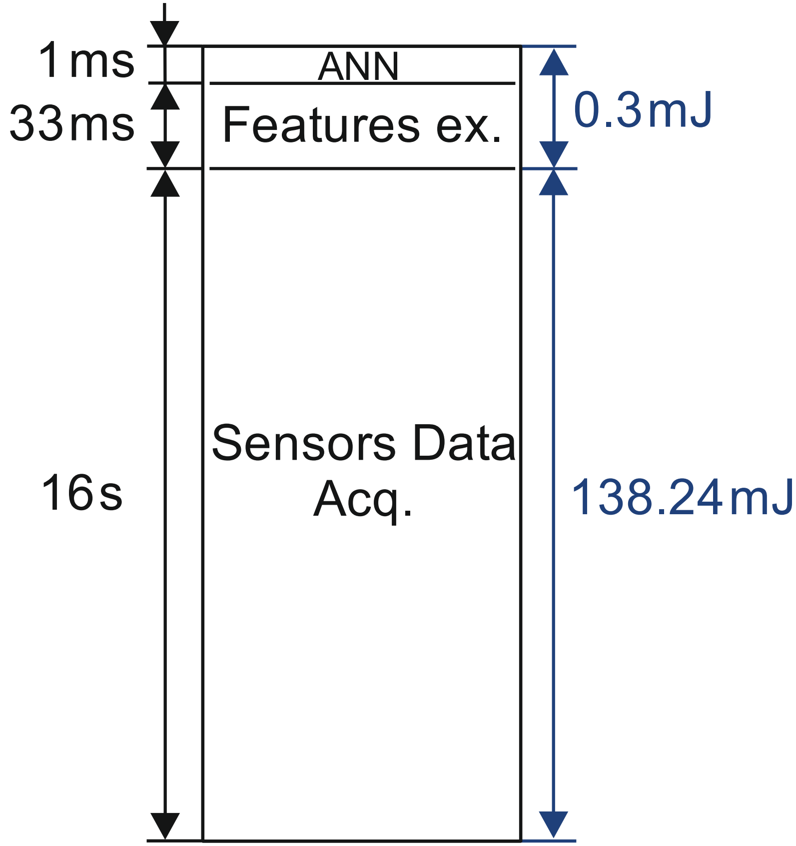
Results – Memory and Processing Load





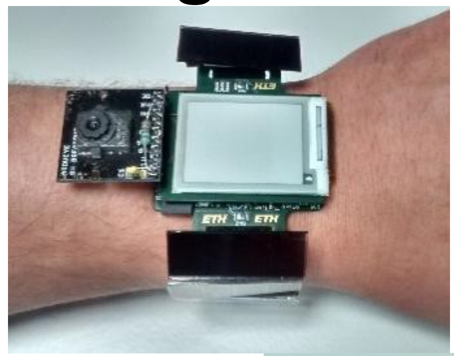
Results – Current & Energy Consumption

Component	Current (active)
Motion sensor	3.5 mA
ECG and respiration sensor	250 µA
Temperature sensor	20 µA
Audio sensor	1.65 mA
GSR	10 µA
Modes	**Drawn current**
Idle	229 µA
Classification	4.36 mA
Data streaming with BLE	16.56 mA



Toward Self-sustaining Smart Wearable

- 8 solar panels
  - Indoor: 600mJ
  - Outdoor: 9-17J

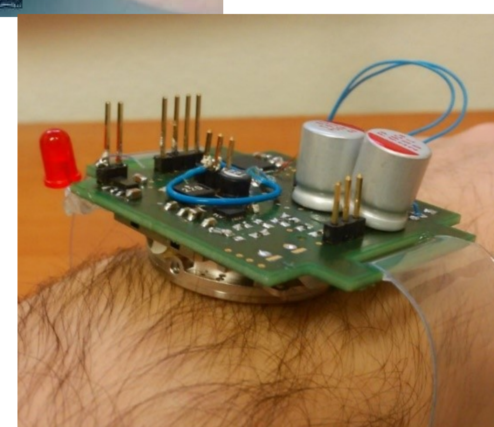


Up to 14 of detection per day
With kinetic harvester

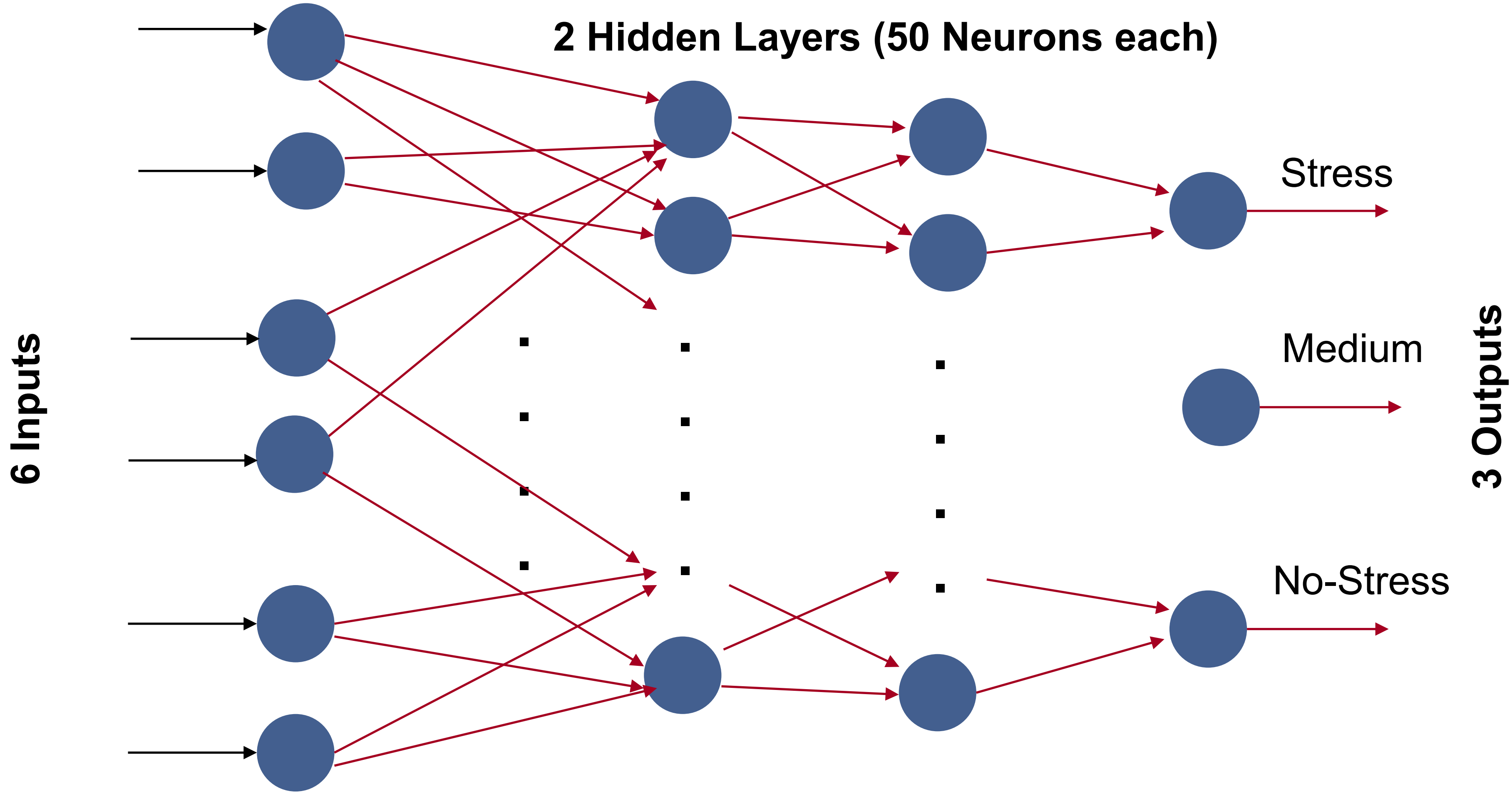
- 7 thermo-electric generators
  - **Indoors:** 0-2 mJ/d
  - **Outdoors:** 100-200 mJ/d

Up to of 10 days with a 600mAh battery.

- Kinetic harvester: 1800mJ/day (sporty person 30min run per day)
  - No occlusions,
  - No weather issue.



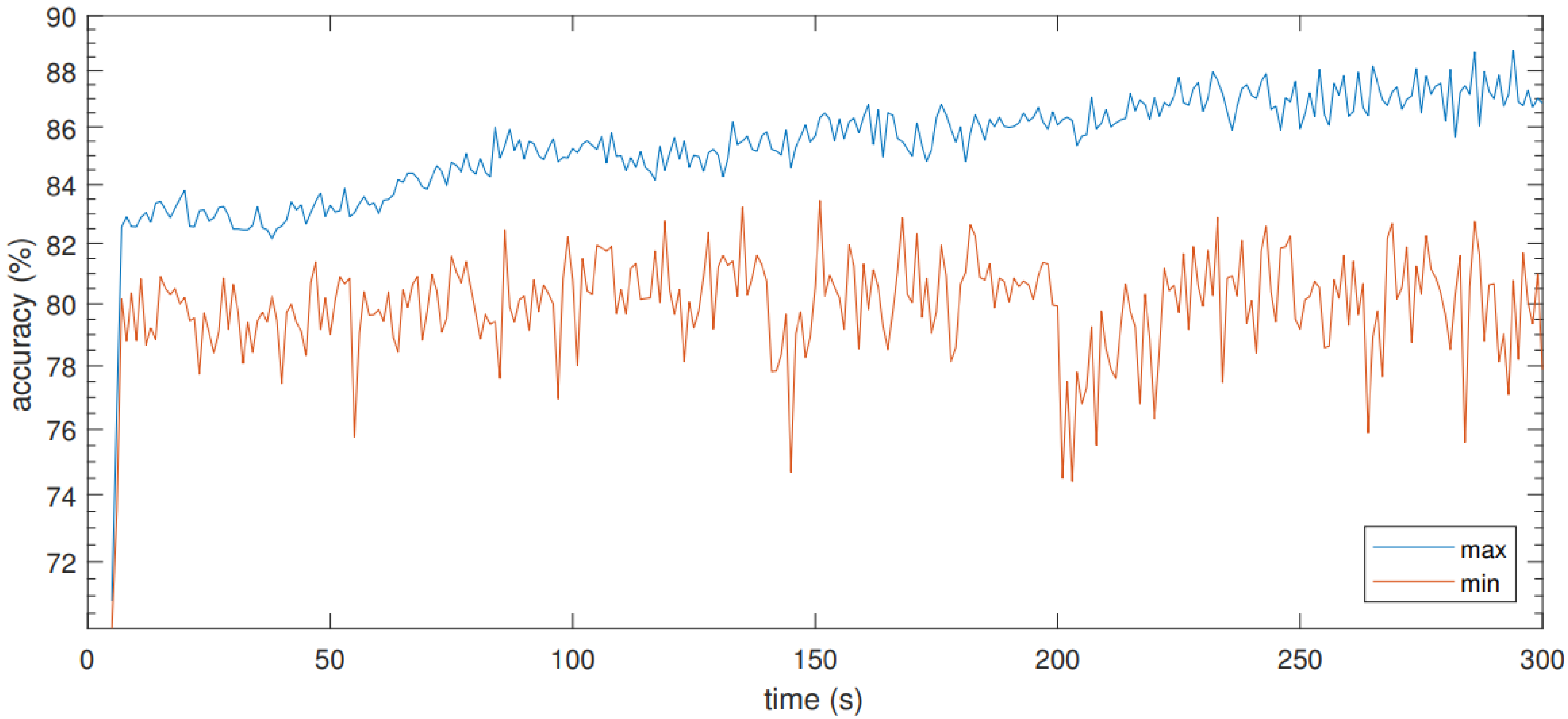
Stress Detection Neural Network – Scaling Up



- #operations: 300 + 2500 + 150 multiply-accumulate
- Memory: 1.2kB + 10kB + 0.6kB memory (weights)

Stress Detection Neural Network

- 2850 multiply-accum./classif., params: 12 kB
- Accuracy: 88% on three classes, 96% on two classes
- Time frame selection



Experimental results

- Stress Detection:
  - 2850 multiply-accum./classif., params: 12 kB
  - Accuracy: 88% on three classes, 96% on two classes
  - 4.5ms with 48MHz core speed -> ARM Cortex M4F (MSP 432)
  - Feature extraction in real time

Limitations to network size:

- 1030 neurons, 48903 weights and **22** layers
- One Classification takes 55ms! Flash is filled 90%, RAM is filled 17%!

Conclusion

- Evaluation/case-study on stress detection
  - Wearable device with Arm Cortex M4
  - 96% accuracy achievable, 4.5ms processing time, ~150mJ/classif.
  - Compute resources not the limitation!

- FANN-on-ARM
  - Enables rapid deployment of NNs on microcontrollers
  - Available online, try yourself!

https://github.com/lukasc-ch/FANN-on-ARM

cavigelli@iis.ee.ethz.ch, magno@iis.ee.ethz.ch

ETH zürich

Questions

This work was in part funded by the Swiss National Science Foundation projects 'MicroLearn: Micropower Deep Learning'(Nr. 162524) and 'Transient Computing Systems' (Nr. 157048).