Practical application of artificial neural networks for seismic event detection

J. Doubravová and J. Horálek

Institute of Geophysics, Dpt. of Seismology

CzechGeo Workshop, 5.12.2018

Outline

Motivation

- continual data produced by dense seismic networks must be reduced
- detection of seismic events should:
  - minimize false detections
  - detect also weak events
- neural networks can extract useful information, forward problem is solved fast

Motivation

- continual data produced by dense seismic networks must be reduced
- detection of seismic events should:
  - minimize false detections
  - detect also weak events
- neural networks can extract useful information, forward problem is solved fast

Motivation

- continual data produced by dense seismic networks must be reduced
- detection of seismic events should:
  - minimize false detections
  - detect also weak events
- neural networks can extract useful information, forward problem is solved fast

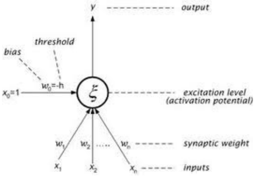
Motivation

- continual data produced by dense seismic networks must be reduced
- detection of seismic events should:
  - minimize false detections
  - detect also weak events
- neural networks can extract useful information, forward problem is solved fast

Motivation

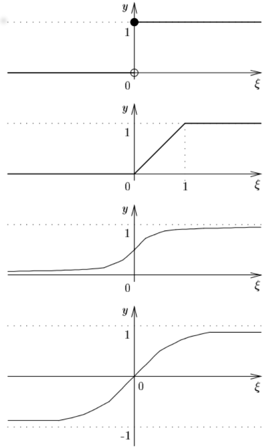
- continual data produced by dense seismic networks must be reduced
- detection of seismic events should:
  - minimize false detections
  - detect also weak events
- neural networks can extract useful information, forward problem is solved fast

Artificial neuron

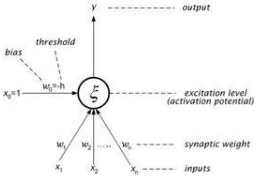


- n real inputs x = dendrites, bias $x_0 = 1$
- weights w = synaptic weights, bias $w_0 = -h$ threshold
- activation potential $\xi = \sum_{i=0}^{n} w_i x_i$
- activation function

$$y = \sigma(\xi) = \begin{cases} 1, & \xi \geq 0 \\ 0, & \xi < 0 \end{cases}$$

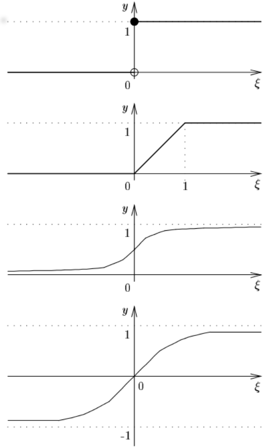


Artificial neuron

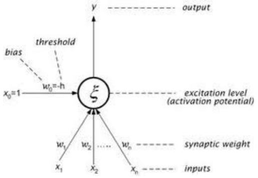


- n real inputs x = dendrites, bias $x_0 = 1$
- weights w = synaptic weights, bias $w_0 = -h$ threshold
- activation potential $\xi = \sum_{i=0}^{n} w_i x_i$
- activation function

$$y = \sigma(\xi) = \begin{cases} 1, & \xi \geq 0 \\ 0, & \xi < 0 \end{cases}$$

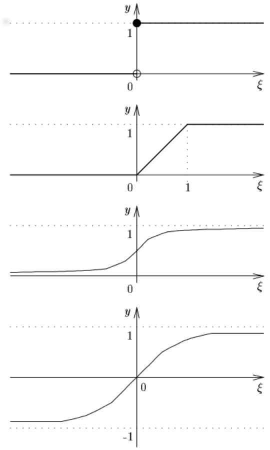


Artificial neuron

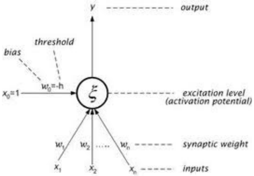


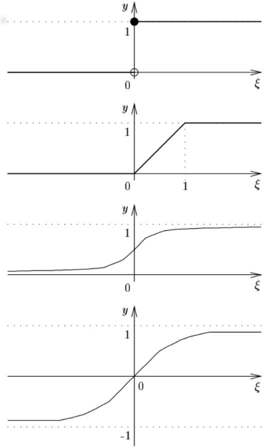
- n real inputs x = dendrites, bias $x_0 = 1$
- weights w = synaptic weights, bias $w_0 = -h$ threshold
- activation potential $\xi = \sum_{i=0}^{n} w_i x_i$
- activation function

$$y = \sigma(\xi) = \begin{cases} 1, & \xi \geq 0 \\ 0, & \xi < 0 \end{cases}$$



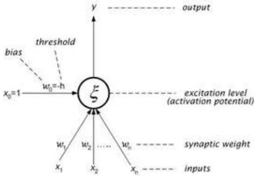
Artificial neuron





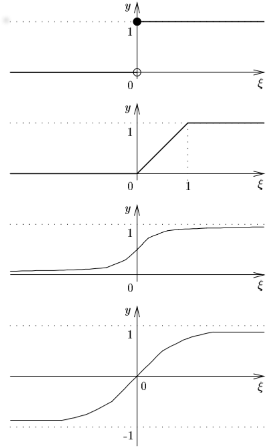
- n real inputs x = dendrites, bias $x_0 = 1$
- weights w = synaptic weights, bias $w_0 = -h$ threshold
- activation potential $\xi = \sum_{i=0}^{n} w_i x_i$
- activation function
$$y = \sigma(\xi) = \begin{cases} 1, & \xi \geq 0 \\ 0, & \xi < 0 \end{cases}$$

Artificial neuron

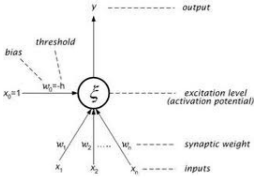


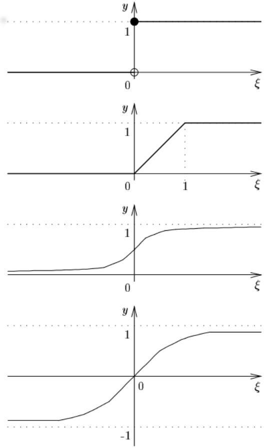
- n real inputs x = dendrites, bias $x_0 = 1$
- weights w = synaptic weights, bias $w_0 = -h$ threshold
- activation potential $\xi = \sum_{i=0}^{n} w_i x_i$
- activation function

$$y = \sigma(\xi) = \begin{cases} 1, & \xi \geq 0 \\ 0, & \xi < 0 \end{cases}$$



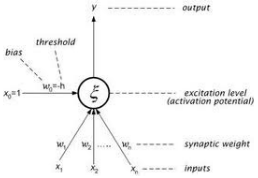
Artificial neuron

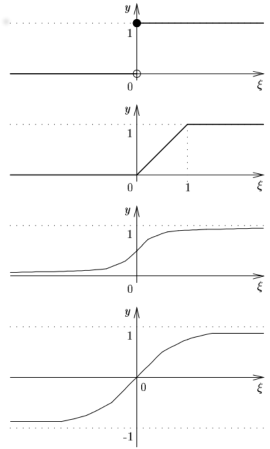




- n real inputs x = dendrites, bias $x_0 = 1$
- weights w = synaptic weights, bias $w_0 = -h$ threshold
- activation potential $\xi = \sum_{i=0}^{n} w_i x_i$
- activation function
$$y = \sigma(\xi) = \begin{cases} 1, & \xi \geq 0 \\ 0, & \xi < 0 \end{cases}$$

Artificial neuron

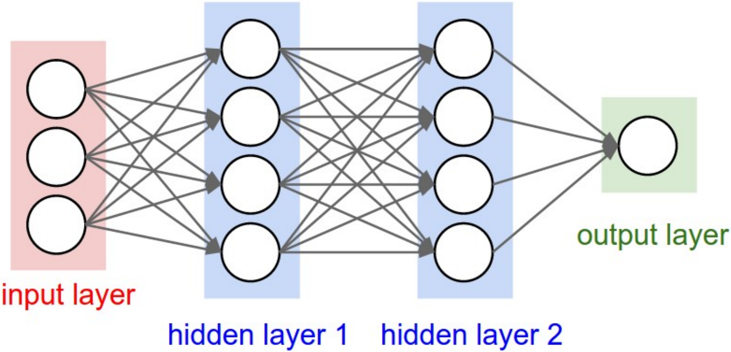




- n real inputs x = dendrites, bias $x_0 = 1$
- weights w = synaptic weights, bias $w_0 = -h$ threshold
- activation potential $\xi = \sum_{i=0}^{n} w_i x_i$
- activation function
$$y = \sigma(\xi) = \begin{cases} 1, & \xi \geq 0 \\ 0, & \xi < 0 \end{cases}$$

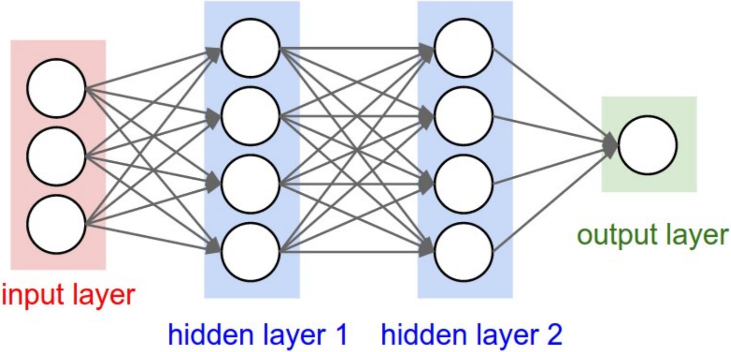
Artificial neural network

- neurons interconnected into networks to solve complex problems
- typical tasks: classification, pattern recognition, regression



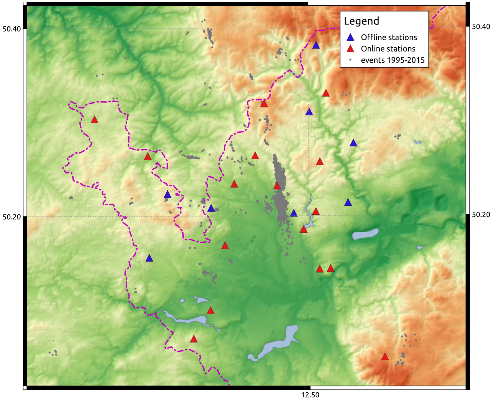
Artificial neural network

- neurons interconnected into networks to solve complex problems
- typical tasks: classification, pattern recognition, regression



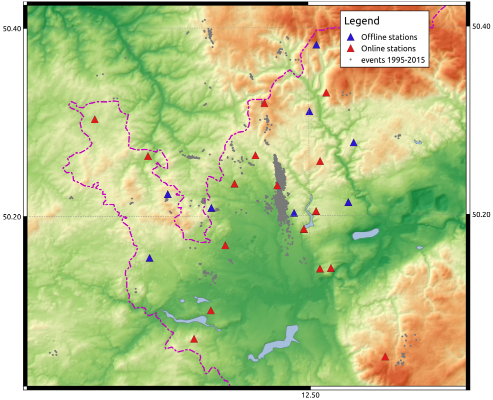
Outline

WEBNET



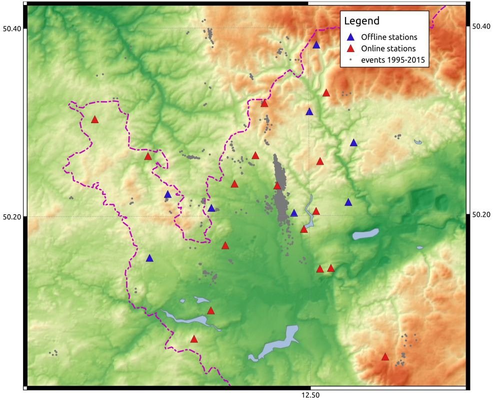
- 24 stations at present and still improving
- 16 stations online
- 250Hz, 3C velocity records

WEBNET



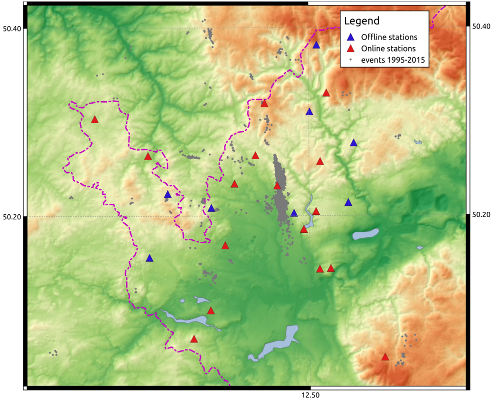
- 24 stations at present and still improving
- 16 stations online
- 250Hz, 3C velocity records

WEBNET



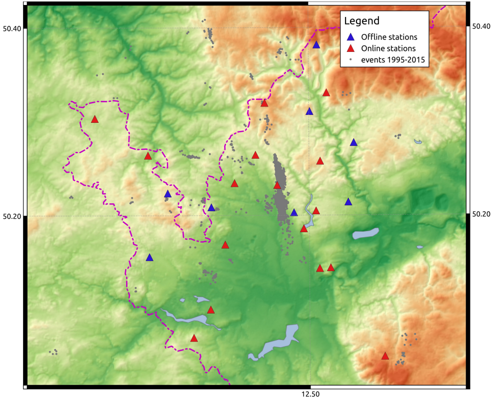
- 24 stations at present and still improving
- 16 stations online
- 250Hz, 3C velocity records

WEBNET



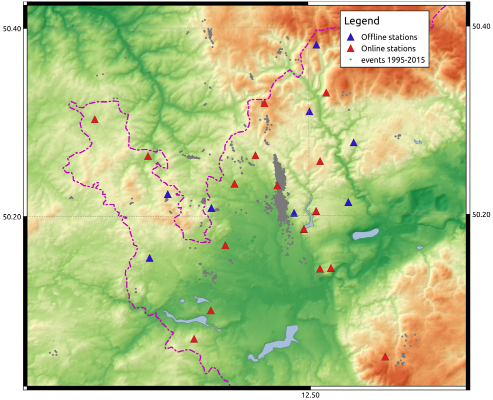
- 24 stations at present and still improving
- 16 stations online
- 250Hz, 3C velocity records

WEBNET



- 24 stations at present and still improving
- 16 stations online
- 250Hz, 3C velocity records

WEBNET

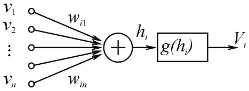


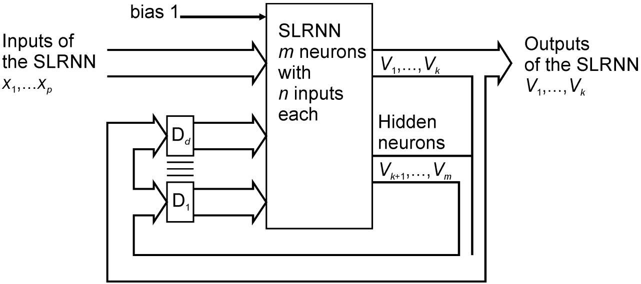
- 24 stations at present and still improving
- 16 stations online
- 250Hz, 3C velocity records

Outline

SLRNN - architecture

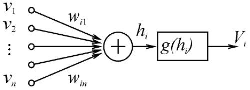
- outputs fed back as inputs = recurrence, memory
- variable delay $D_1..D_d$

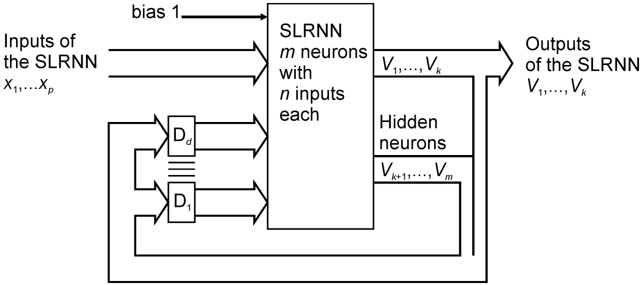




SLRNN - architecture

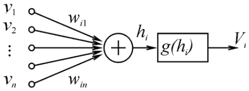
- outputs fed back as inputs = recurrence, memory
- variable delay $D_1..D_d$

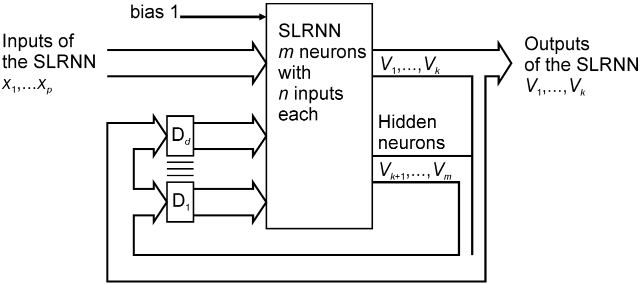




SLRNN - architecture

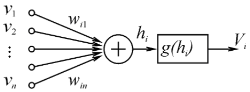
- outputs fed back as inputs = recurrence, memory
- variable delay $D_1..D_d$

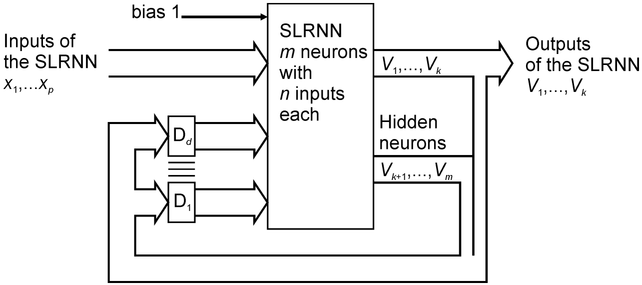




SLRNN - architecture

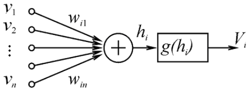
- outputs fed back as inputs = recurrence, memory
- variable delay $D_1..D_d$

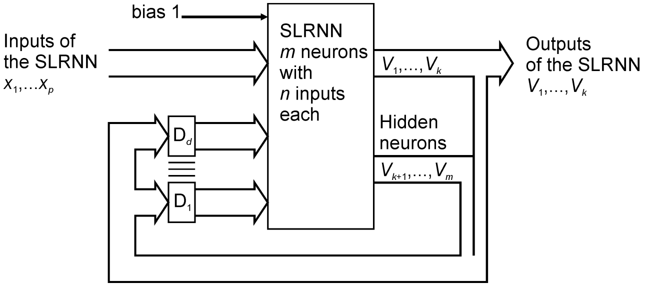




SLRNN - architecture

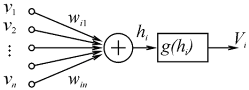
- outputs fed back as inputs = recurrence, memory
- variable delay $D_1..D_d$

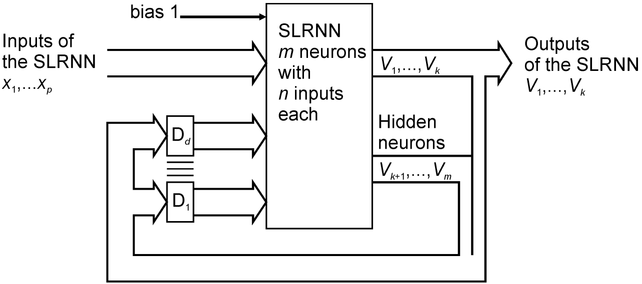




SLRNN - architecture

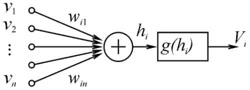
- 8 neurons, 18 inputs, 3 outputs (event, P, S)
- delays 1, 2, 4, and 8 samples - 4x8=32 feedbacks

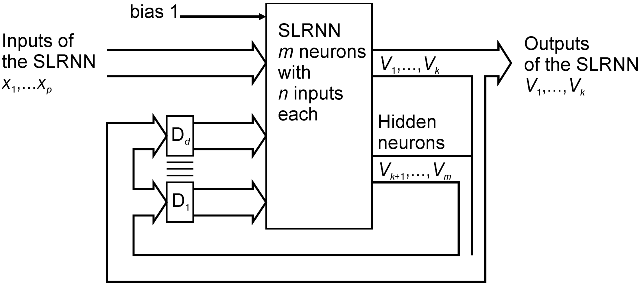




SLRNN - architecture

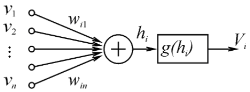
- 8 neurons, 18 inputs, 3 outputs (**event**, P, S)
- delays 1, 2, 4, and 8 samples - 4x8=32 feedbacks

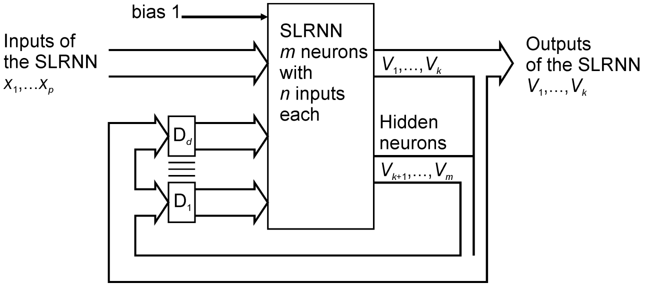




SLRNN - architecture

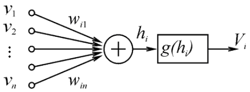
- 8 neurons, 18 inputs, 3 outputs (**event**, P, S)
- delays 1, 2, 4, and 8 samples - 4x8=32 feedbacks

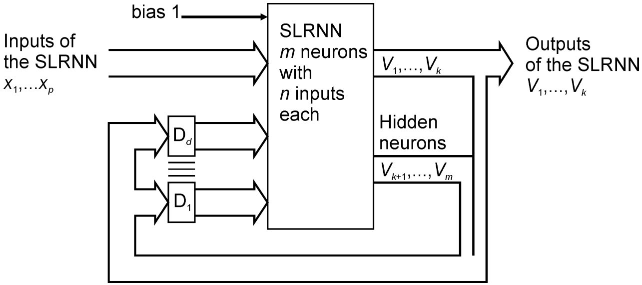




SLRNN - architecture

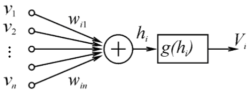
- 8 neurons, 18 inputs, 3 outputs (**event**, P, S)
- delays 1, 2, 4, and 8 samples - 4x8=32 feedbacks

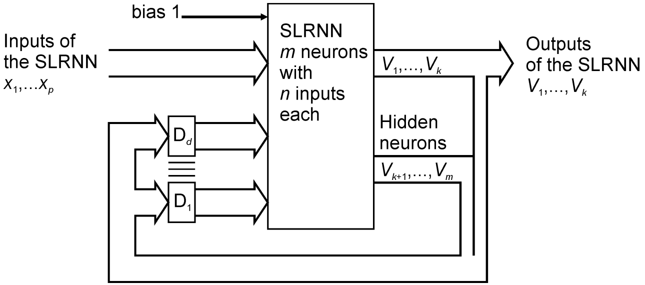




SLRNN - architecture

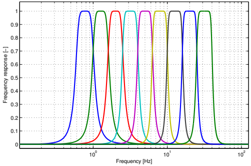
- 8 neurons, 18 inputs, 3 outputs (**event**, P, S)
- delays 1, 2, 4, and 8 samples - 4x8=32 feedbacks

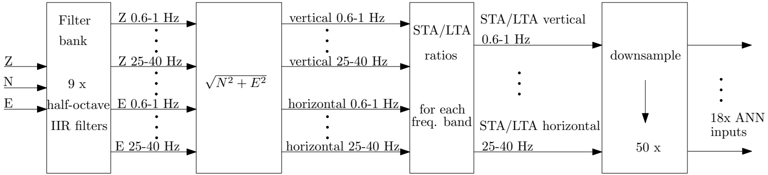




SLRNN - data preprocessing

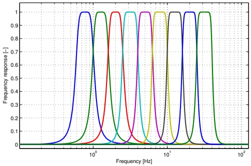
- STA/LTA in 9 narrow-band filtered velocity records
- decimation to 0.2 s

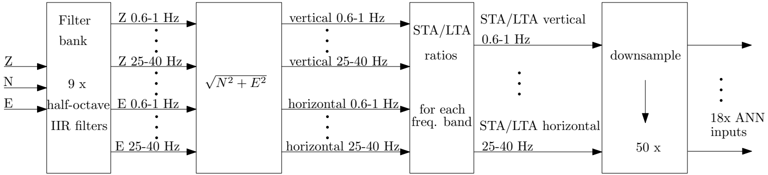




SLRNN - data preprocessing

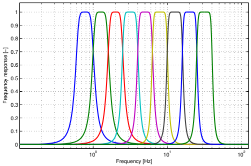
- STA/LTA in 9 narrow-band filtered velocity records
- decimation to 0.2 s

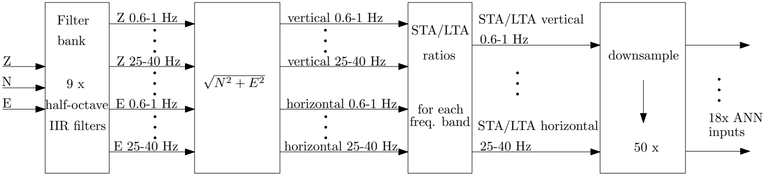




SLRNN - data preprocessing

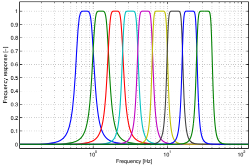
- STA/LTA in 9 narrow-band filtered velocity records
- decimation to 0.2 s

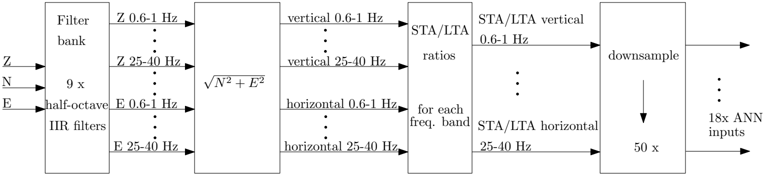




SLRNN - data preprocessing

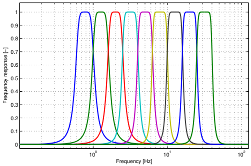
- STA/LTA in 9 narrow-band filtered velocity records
- decimation to 0.2 s

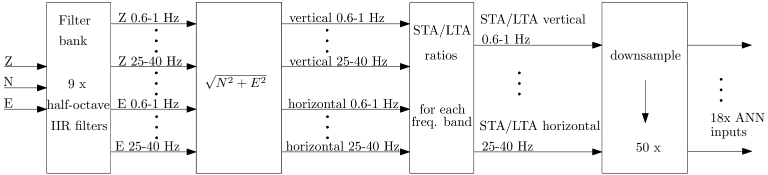




SLRNN - data preprocessing

- STA/LTA in 9 narrow-band filtered velocity records
- decimation to 0.2 s





Outline

Training

- supervised learning: searching w_{ij} to fit required outputs for training set
- cost function minimization by Back Propagation Through Time (gradient based method, back propagation of error modification for recurrent networks)

Training

- supervised learning: searching w_{ij} to fit required outputs for training set
- cost function minimization by Back Propagation Through Time (gradient based method, back propagation of error modification for recurrent networks)

Data

- seismic swarm 2008 (events) and calm year 2010 (disturbances) WEBNET (West Bohemia)
- events of various magnitudes, locations, mechanisms...
- disturbances of different nature -blasts, regional and teleseismic ev., wind, stroms...
- training set divided (randomly)
  - actual training data (80%)
  - validation data (20%)

Data

- seismic swarm 2008 (events) and calm year 2010 (disturbances) WEBNET (West Bohemia)
- events of various magnitudes, locations, mechanisms...
- disturbances of different nature -blasts, regional and teleseismic ev., wind, stroms...
- training set divided (randomly)
  - actual training data (80%)
  - validation data (20%)

Data

- seismic swarm 2008 (events) and calm year 2010 (disturbances) WEBNET (West Bohemia)
- events of various magnitudes, locations, mechanisms...
- disturbances of different nature -blasts, regional and teleseismic ev., wind, stroms...
- training set divided (randomly)
    - actual training data (80%)
    - validation data (20%)

Data

- seismic swarm 2008 (events) and calm year 2010 (disturbances) WEBNET (West Bohemia)
- events of various magnitudes, locations, mechanisms...
- disturbances of different nature -blasts, regional and teleseismic ev., wind, stroms...
- training set divided (randomly)
  - actual training data (80%)
  - validation data (20%)

Data

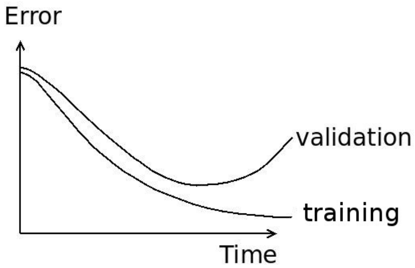
- seismic swarm 2008 (events) and calm year 2010 (disturbances) WEBNET (West Bohemia)
- events of various magnitudes, locations, mechanisms...
- disturbances of different nature -blasts, regional and teleseismic ev., wind, stroms...
- training set divided (randomly)
  - actual training data (80%)
  - validation data (20%)

Data

- seismic swarm 2008 (events) and calm year 2010 (disturbances) WEBNET (West Bohemia)
- events of various magnitudes, locations, mechanisms...
- disturbances of different nature -blasts, regional and teleseismic ev., wind, stroms...
- training set divided (randomly)
  - actual training data (80%)
  - validation data (20%)

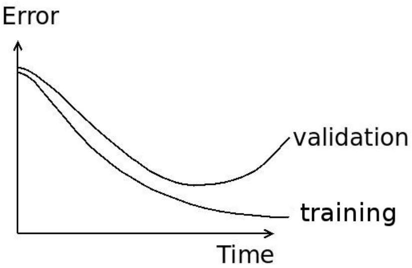
Training and overtraining

- training continues until the validation set error decreases
- cost function strongly nelinear! = more attempts from randomly selected starting point



Training and overtraining

- training continues until the validation set error decreases
- cost function strongly nelinear! = more attempts from randomly selected starting point



Outline

1. Artificial neural networks
2. SLRNN and training
   - WEBNET
   - Single Layer Recurrent Neural Network
   - SLRNN training
3. Results
   - False detections
   - Undetected events
4. Application
   - Application to Webnet
   - Application to Reykjanet

False detections ?

- tested on swarm 2011, single station detection
- many false detections
- many events => small events without manual reading

False detections ?

- tested on swarm 2011, single station detection
- many false detections
- many events => small events without manual reading

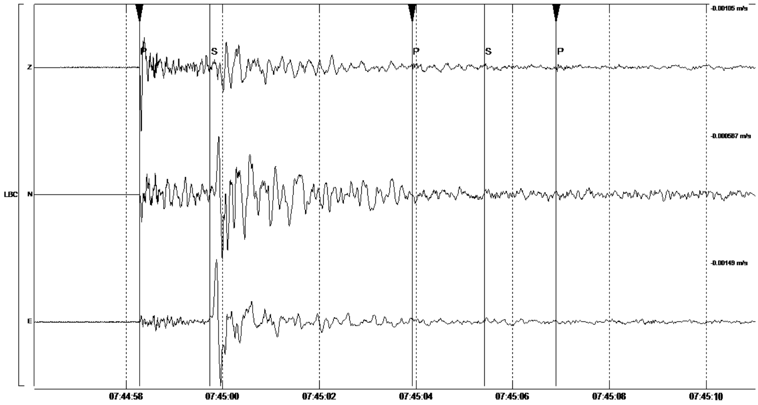
False detections ?

- tested on swarm 2011, single station detection
- many false detections
- many events => small events without manual reading

Outline

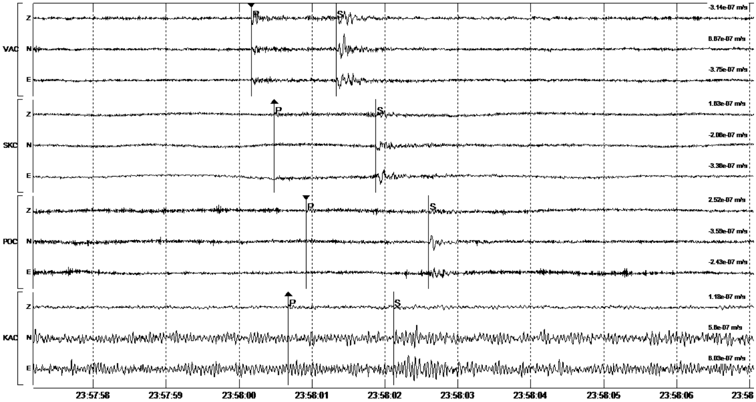
Undetected events

Ev. $M_L = 2.3$ and $M_L = 2.2$ in coda of $M_L = 3.8$



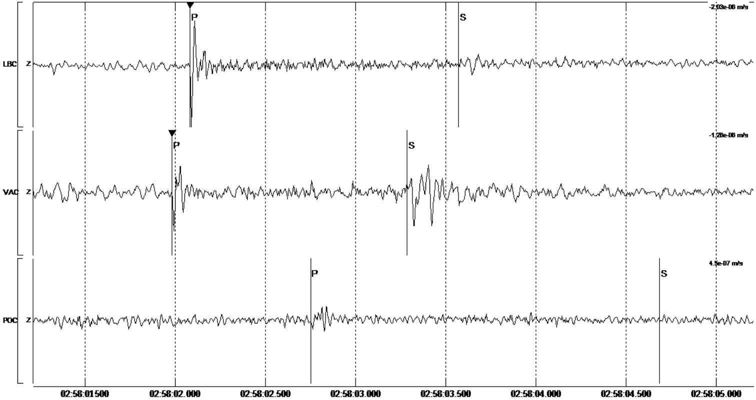
Undetected events

$M_L = -0.3$ noisy record on KAC

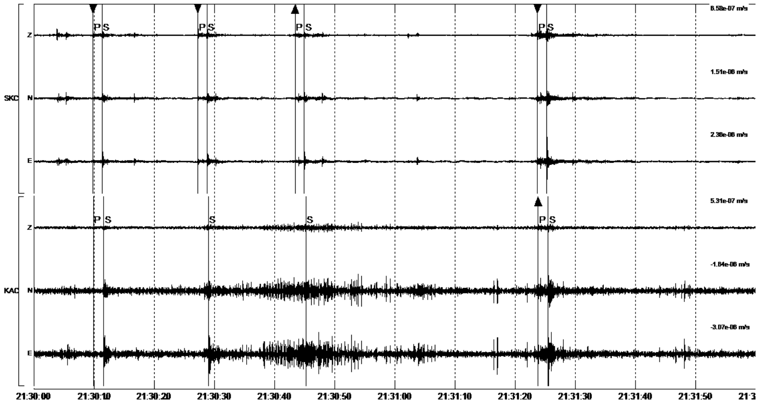


Undetected events

weak amplitudes on POC $M_L = 0.2$



Undetected events
disturbances on KAC



How to solve it?

- we have high number of false detections / or very weak events - too much events to process
- few undetected events - really unacceptable
- => WE MUST USE **COINCIDENCE** IN THE NETWORK

How to solve it?

- we have high number of false detections / or very weak events - too much events to process
- few undetected events - really unacceptable
- => WE MUST USE **COINCIDENCE** IN THE NETWORK

How to solve it?

- we have high number of false detections / or very weak events - too much events to process
- few undetected events - really unacceptable
- => WE MUST USE **COINCIDENCE** IN THE NETWORK

Coincidence

- when a human processes waveforms, he takes into account all the stations at once
- let the machine see detection outputs of the stations at once
- for each detection we look for sufficient number of detections on other stations in certain time window

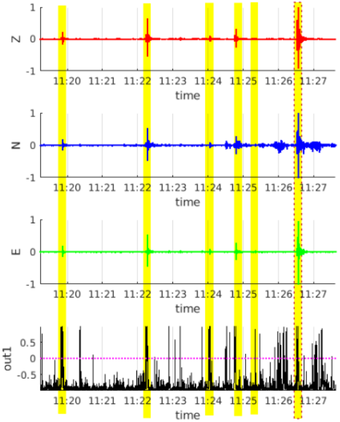
Coincidence

- when a human processes waveforms, he takes into account all the stations at once
- let the machine see detection outputs of the stations at once
- for each detection we look for sufficient number of detections on other stations in certain time window

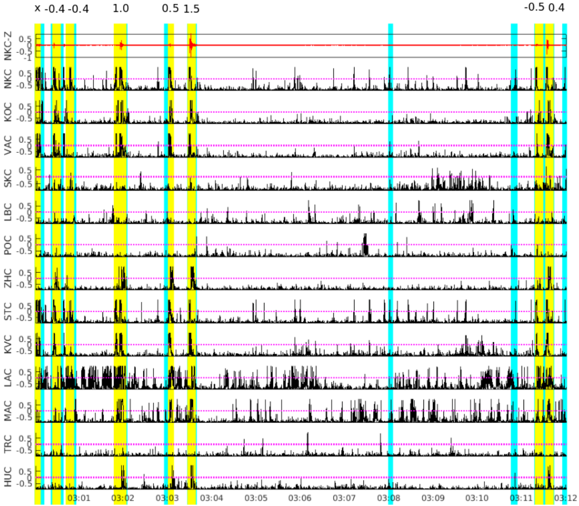
Coincidence

- when a human processes waveforms, he takes into account all the stations at once
- let the machine see detection outputs of the stations at once
- for each detection we look for sufficient number of detections on other stations in certain time window

Coincidence



Coincidence 4 vs. 6



Concidence 4 vs. 6

- six stations seems to be enough for reasonable minimum magnitude
- time window to search for coincidence was 0.8s

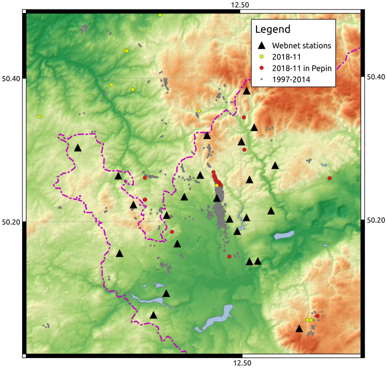
Concidence 4 vs. 6

- six stations seems to be enough for reasonable minimum magnitude
- time window to search for coincidence was 0.8s

Outline

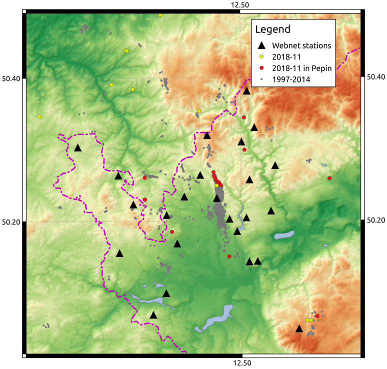
Webnet

- even there is a good detection and location provided by PEPIN, there are some limitations
- especially events outside the NK focal zone could be missing



Webnet

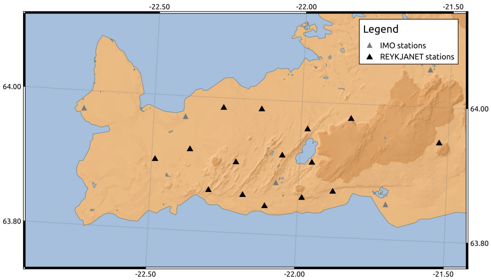
- even there is a good detection and location provided by PEPIN, there are some limitations
- especially events outside the NK focal zone could be missing



Outline

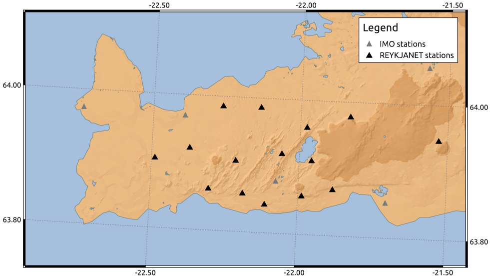
Reykjanet network

- south-west Iceland, Reykjanes peninsula
- 15 off-line stations
- size of the network, number of stations, earthquake swarm activity - similar to WB



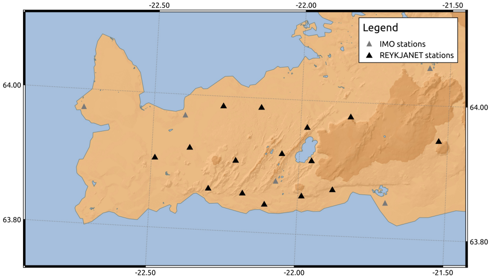
Reykjanet network

- south-west Iceland, Reykjanes peninsula
- 15 off-line stations
- size of the network, number of stations, earthquake swarm activity - similar to WB



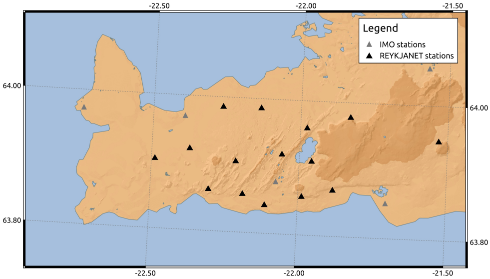
Reykjanet network

- south-west Iceland, Reykjanes peninsula
- 15 off-line stations
- size of the network, number of stations, earthquake swarm activity - similar to WB

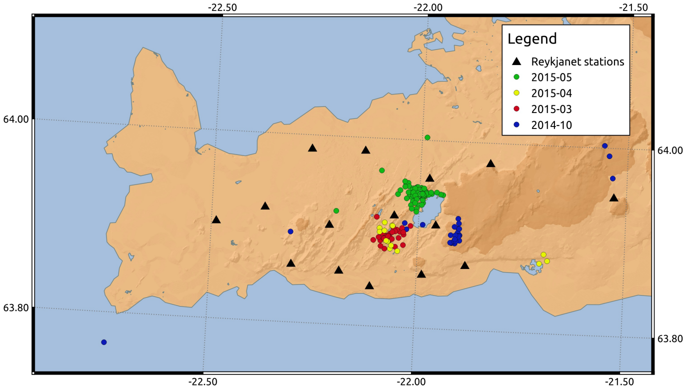


Neural network

- the best SLRNN network trained for WEBNET



Data: 4 small swarms 2014-2015



Catalogs

- SIL - IMO catalog - manually revised automatic locations from Icelandic regional network
- Antelope - automatic catalog by Antelope from Reykjanet stations (B. Růžek)
- PePiN - automatic locations from PePiN (T. Fischer)
- ANN - detection (no location)
- max. magnitude $M_L = 2,3$

Catalogs

- SIL - IMO catalog - manually revised automatic locations from Icelandic regional network
- Antelope - automatic catalog by Antelope from Reykjanet stations (B. Růžek)
- PePiN - automatic locations from PePiN (T. Fischer)
- ANN - detection (no location)
- max. magnitude $M_L = 2,3$

Catalogs

- SIL - IMO catalog - manually revised automatic locations from Icelandic regional network
- Antelope - automatic catalog by Antelope from Reykjanet stations (B. Růžek)
- PePiN - automatic locations from PePiN (T. Fischer)
- ANN - detection (no location)
- max. magnitude $M_L = 2,3$

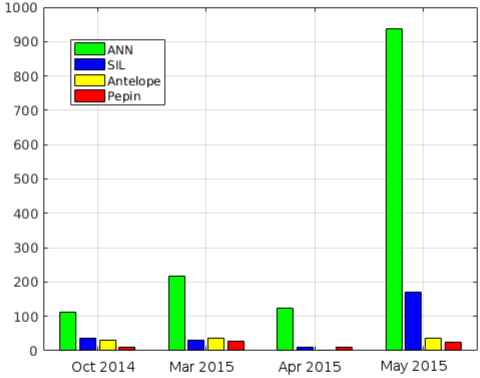
Catalogs

- SIL - IMO catalog - manually revised automatic locations from Icelandic regional network
- Antelope - automatic catalog by Antelope from Reykjanet stations (B. Růžek)
- PePiN - automatic locations from PePiN (T. Fischer)
- ANN - detection (no location)
- max. magnitude $M_L = 2,3$

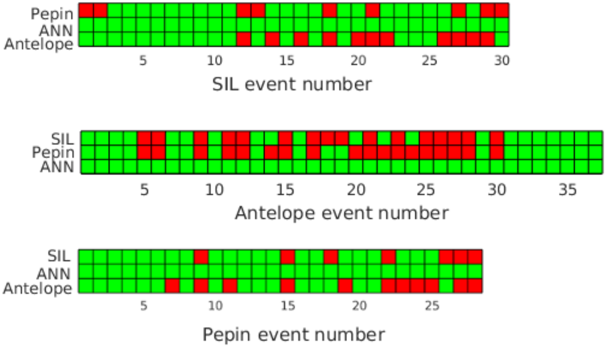
Catalogs

- SIL - IMO catalog - manually revised automatic locations from Icelandic regional network
- Antelope - automatic catalog by Antelope from Reykjanet stations (B. Růžek)
- PePiN - automatic locations from PePiN (T. Fischer)
- ANN - detection (no location)
- max. magnitude $M_L = 2,3$

Number of events

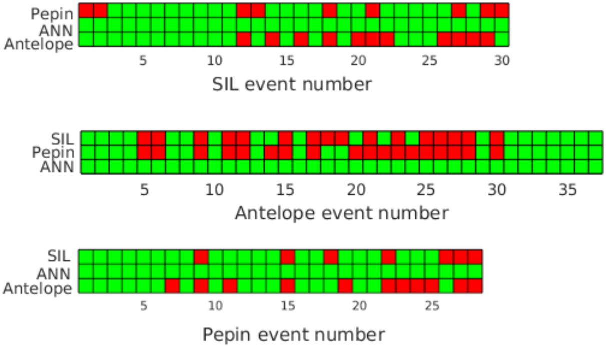


Porovnání jednotlivých jevů - březen 2015



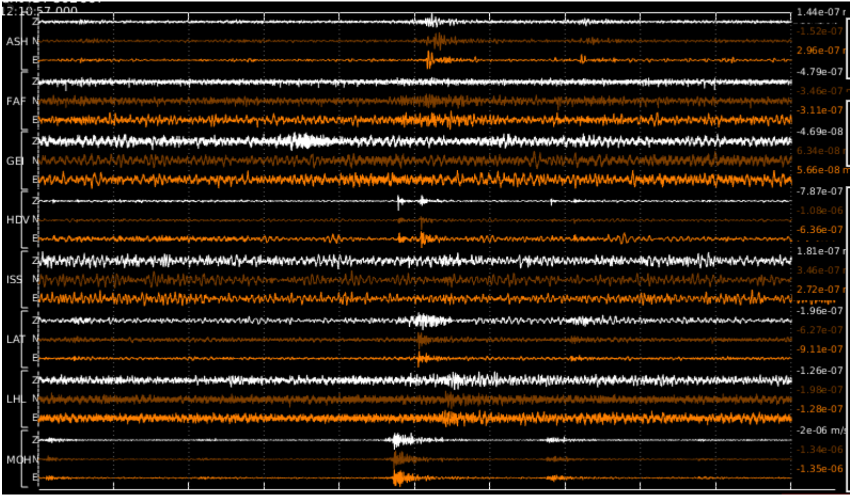
- Pepin and SIL sorted by magnitude
- Antelope sorted in time

Porovnání jednotlivých jevů - březen 2015



- Pepin and SIL sorted by magnitude
- Antelope sorted in time

Weakest event



Conclusion

- SLRNN detector is fast and effective
- coincidence within a network solves undetected events
- coincidence reduces reasonably number of false detections
- further processing will reveal weak events as they can't be successfully localized
- the neural network trained for West Bohemia works well for Reykjanet

Conclusion

- SLRNN detector is fast and effective
- coincidence within a network solves undetected events
- coincidence reduces reasonably number of false detections
- further processing will reveal weak events as they can't be successfully localized
- the neural network trained for West Bohemia works well for Reykjanet

Conclusion

- SLRNN detector is fast and effective
- coincidence within a network solves undetected events
- coincidence reduces reasonably number of false detections
- further processing will reveal weak events as they can't be successfully localized
- the neural network trained for West Bohemia works well for Reykjanet

Conclusion

- SLRNN detector is fast and effective
- coincidence within a network solves undetected events
- coincidence reduces reasonably number of false detections
- further processing will reveal weak events as they can't be successfully localized
- the neural network trained for West Bohemia works well for Reykjanet

Conclusion

- SLRNN detector is fast and effective
- coincidence within a network solves undetected events
- coincidence reduces reasonably number of false detections
- further processing will reveal weak events as they can't be successfully localized
- the neural network trained for West Bohemia works well for Reykjanet

.

Thank you for your attention !