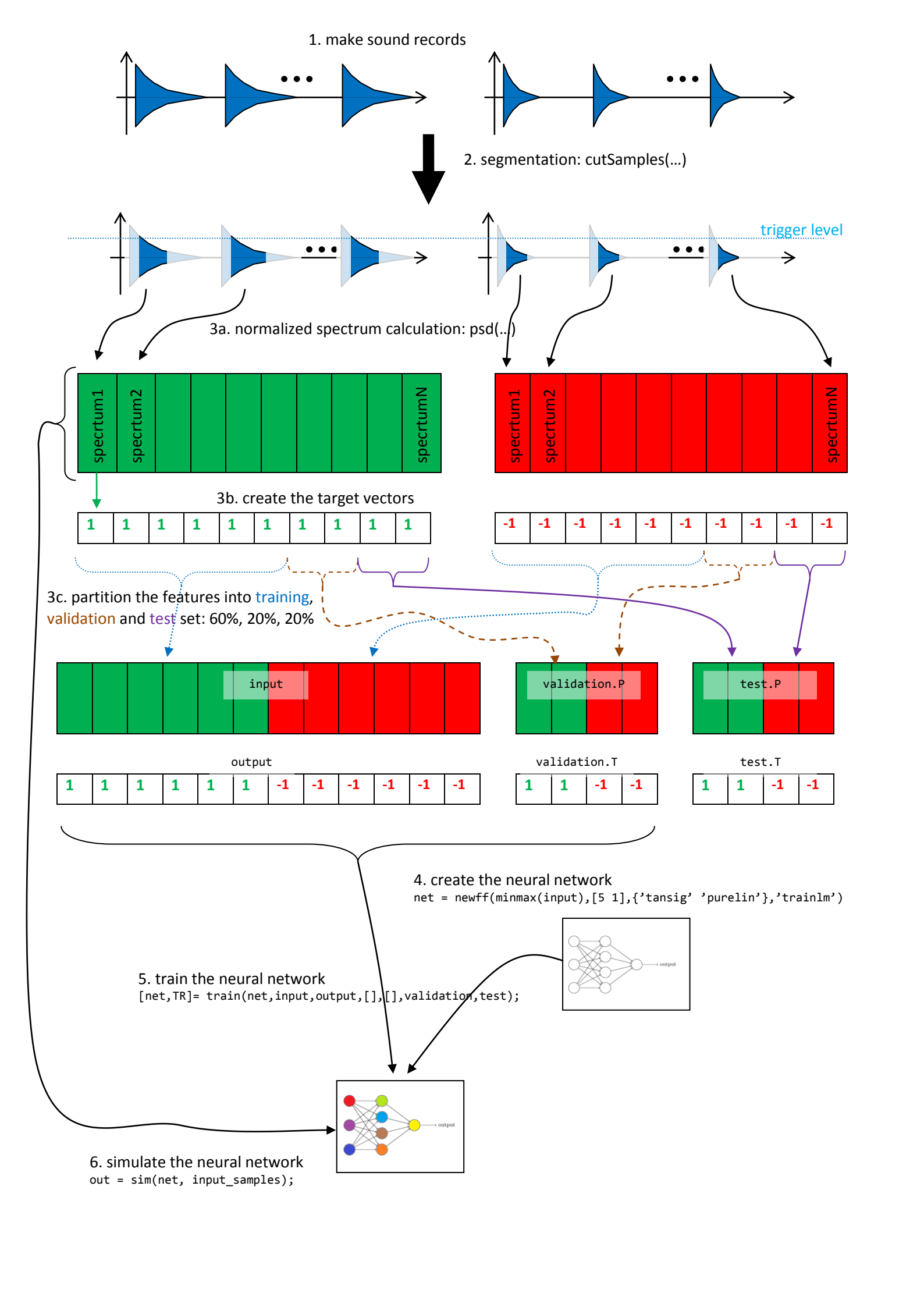
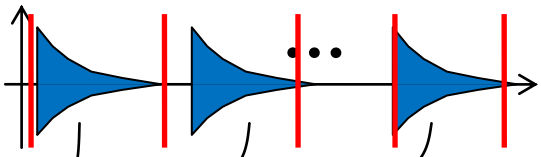
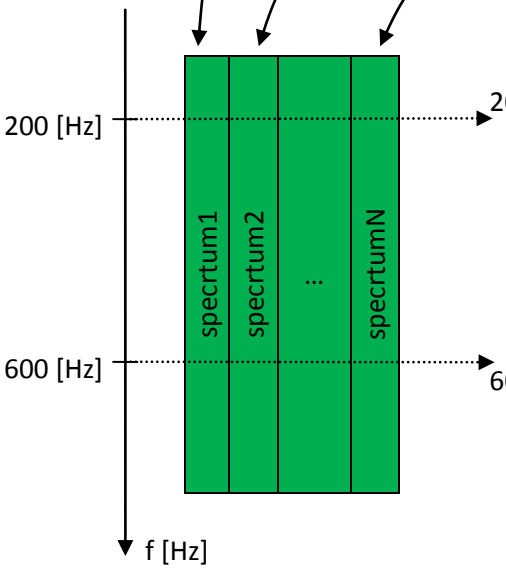

1. make sound records
2. segmentation: cutSamples(...)
trigger level
3a. normalized spectrum calculation: psd(...)
specrtum1
specrtum2
specrtumN
3b. create the target vectors
1 1 1 1 1 1 1 1 1 1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
3c. partition the features into training, validation and test set: 60%, 20%, 20%
input
validation.P
test.P
output
1 1 1 1 1 1 -1 -1 -1 -1 -1 -1
validation.T
1 1 -1 -1
test.T
1 1 -1 -1
4. create the neural network
net = newff(minmax(input),[5 1],{'tansig' 'purelin'},'trainlm')
5. train the neural network
[net,TR]= train(net,input,output,[],[],validation,test);
6. simulate the neural network
out = sim(net, input_samples);

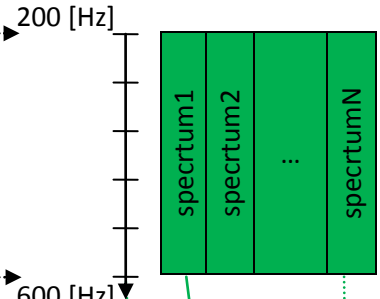
1. make the segmentation of the sound record

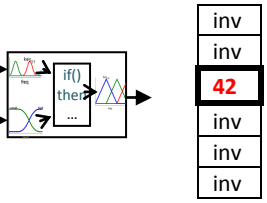


2. normalized spectrum calculation: psd(…)
Frequency scale should also be generated!!!



3. cut the range 200Hz…600Hz

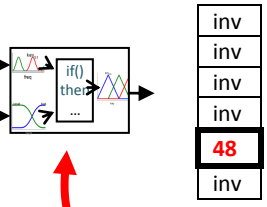




6.a. evaluate the FIS for each segment (include freq. scale)

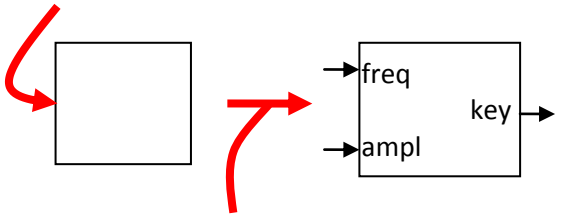
```
out = evalfis( [freq, spectrum(:,segm)], notes);
```

6.b. select the not invalid key from the output vector



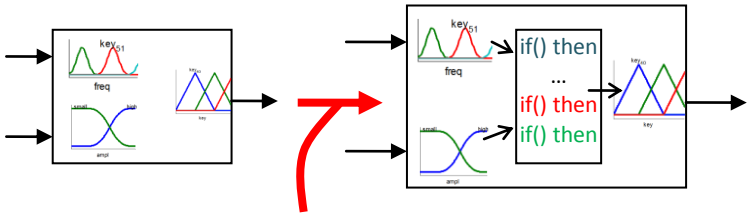
4. create the FIS

```
notes = newfis('Sound_notes','mamdani');
notes = setfis(notes,'defuzzmethod','mom');
```



5.b. add the membership functions to in/out

```
notes = addmf(notes,'input',2,'high', 'smf',[0.8 0.9]);
notes = addmf(notes,'input',2,'small','zmf',[0.8 0.9]);
….
….
```



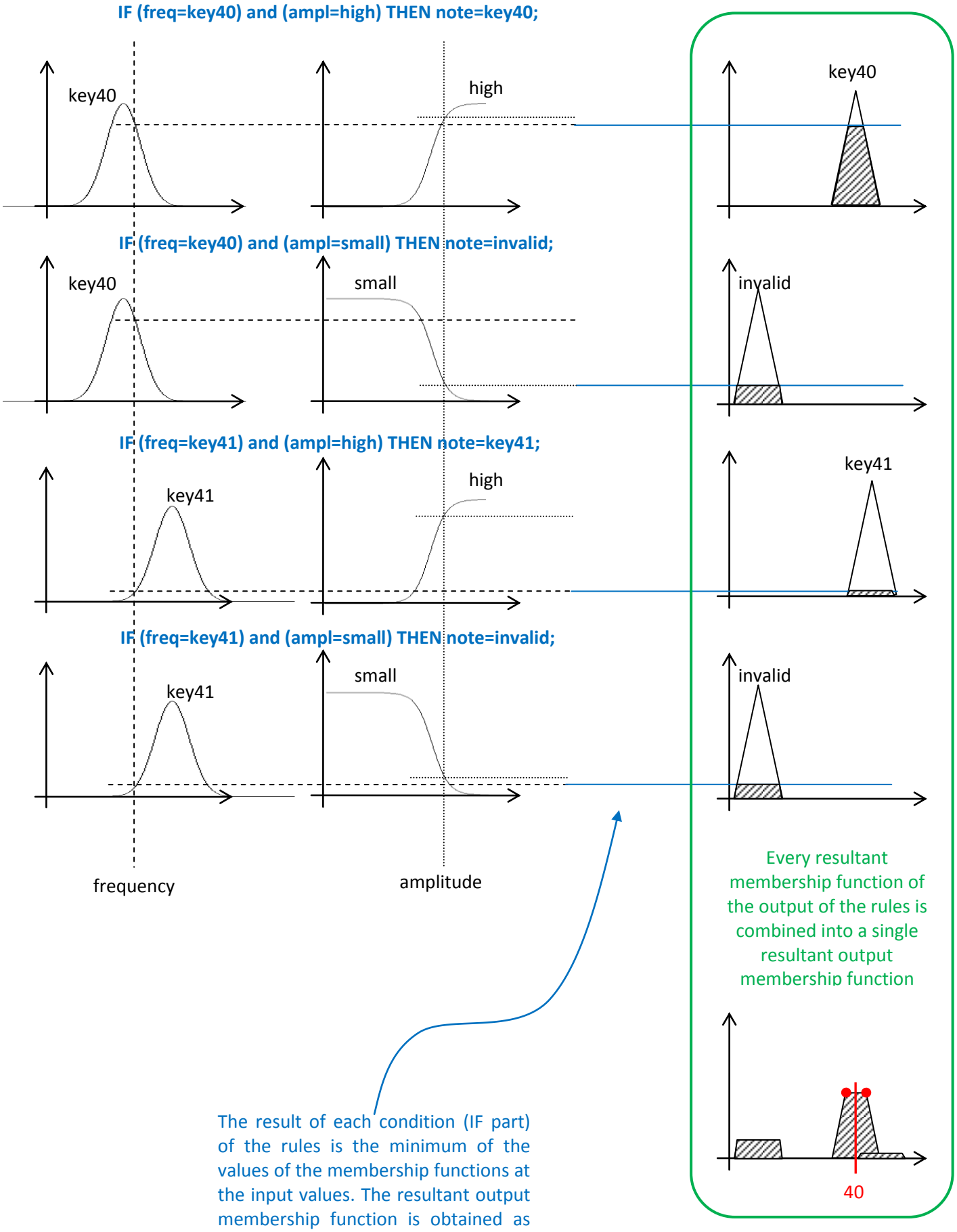
5.a. add the inputs and output

```
notes = addvar(notes,'input','frequency',[200 600]);
notes = addvar(notes,'input','amplitude',[0 1]);
notes = addvar(notes,'output','sound_note',[20 55]);
```

5.c. add rules to the FIS

```
rule = [1 1 1 1 1;
        1 2 invalid_nr 1 1;
        2 1 2 1 1;
        2 2 invalid_nr 1 1;
        3 1 3 1 1;
        3 2 invalid_nr 1 1;
        …];
notes = addrule(notes,rule);
```

c



The result of each condition (IF part) of the rules is the minimum of the values of the membership functions at the input values. The resultant output membership function is obtained as the intersection of the membership function and the minimum of the input membership function values.

The output is originally a resultant membership function which cannot be interpreted in this form.
The middle of two maximum edges are selected as output, this is called defuzzification.