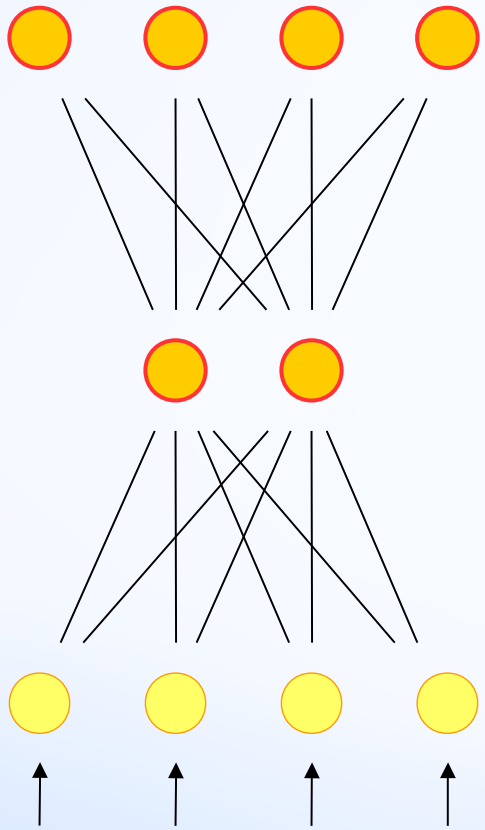
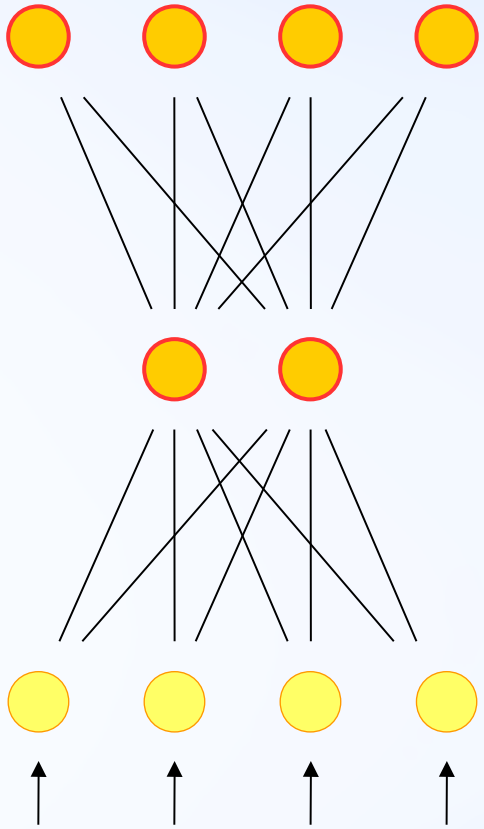
Neural Network Implementation I

Task 1 - Identity



Task 1 - Identity



- Identity on 4-dimensional binary data
- Network layout 4 – 2 – 4 neurons
- First instance (subtask) running on a training set of 4 specific vectors:

  1 1 0 0
  0 0 1 1
  1 0 1 0
  0 1 0 1
  0 0 0 0

- Second instance (subtask) running on all 16 vectors
- Continuous model with sigmoidal response function, $\lambda = 1.0$
- Training parameter $\eta = 0.1$, momentum $\alpha = 0.0$

Task 1 - Identity


```
Training progress
Epoch  1    Global error  12.25845
...
Epoch 40    Global error   0.04578

Testing
Input     Output    Response              Error Accuracy Reliability
1 1 0 0   1 1 0 0   0.92 1.00 0.03 0.08   0.014   100%   100%
0 0 1 1   0 0 1 1   0.12 0.23 0.95 0.84   0.095   100%   100%
1 0 1 0   1 0 1 0   0.76 0.31 0.58 0.08   0.337   100%    50%
0 1 0 1   0 1 0 1   0.73 0.88 0.14 0.76   0.625    75%   100%
0 0 0 0   0 0 0 0   0.17 0.21 0.22 0.12   0.136   100%   100%
```

And the same for all 16 patterns

- Training displays global error for each epoch
- Number of epochs is usually fixed and limited by the value of global error, i.e. 0.01
- Global error is a cummulated error for all patterns trained within one epoch (whole training set)

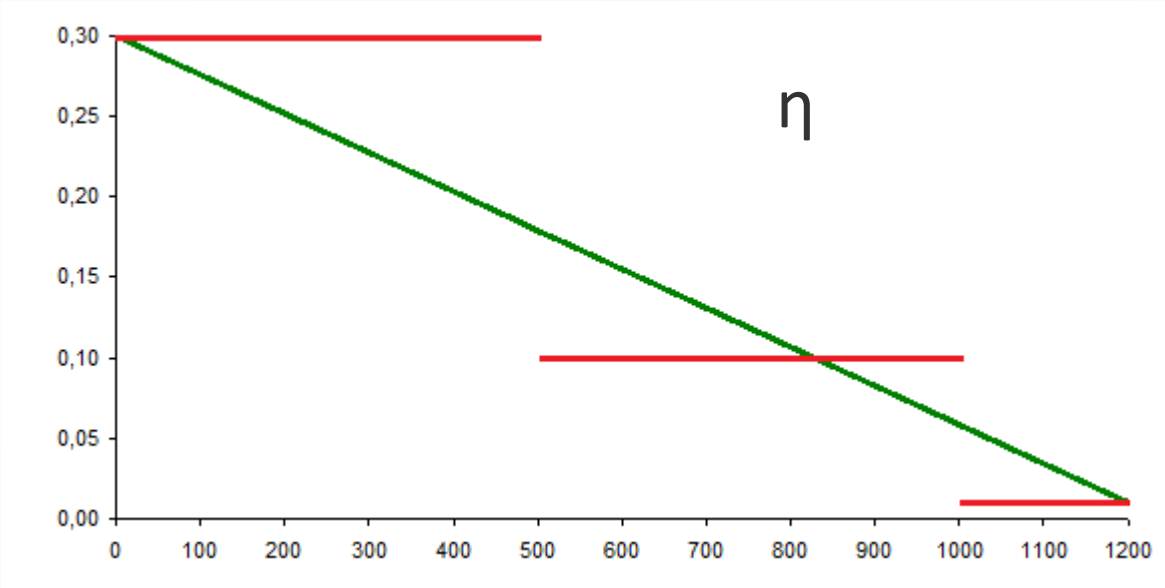
- Testing is in general performed for a set of patterns independent on the training set
- Where exact results for a fixed training set are required (this case) testing runs for the same set

- For each pattern we check success of the training by displaying these verificators:
- Local error evaluated for one pattern
- Accuracy showing a match to resired output after rounding the network response
- Reliability of the response, i.e. falling into reliable extend <0,ε> and <1-ε,1>, where ε is 0.3 or 0.2

Task 1 - Identity


```
Training progress
Epoch  1    Global error  12.25845
...
Epoch 40    Global error   0.04578

Testing
Input     Output    Response                Error Accuracy Reliability
1 1 0 0   1 1 0 0   0.92 1.00 0.03 0.08   0.014   100%   100%
0 0 1 1   0 0 1 1   0.12 0.23 0.95 0.84   0.095   100%   100%
1 0 1 0   1 0 1 0   0.76 0.31 0.58 0.08   0.337   100%    50%
0 1 0 1   0 1 0 1   0.73 0.88 0.14 0.76   0.625    75%   100%
0 0 0 0   0 0 0 0   0.17 0.21 0.22 0.12   0.136   100%   100%
```

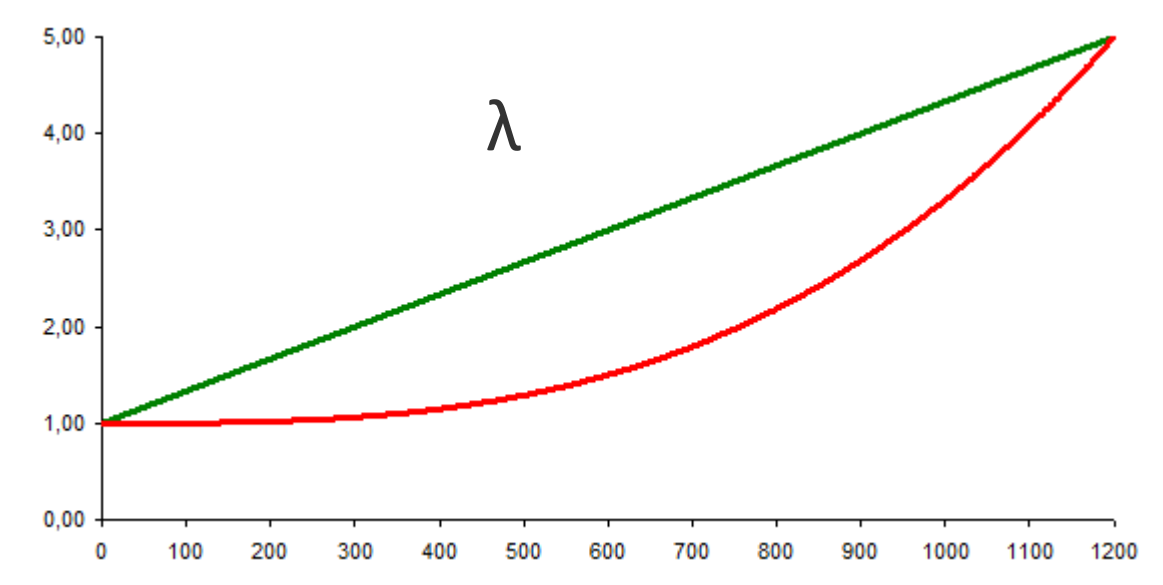


- Try to get the best results as possible
- There should be no error in identity for 5 patterns, i.e. get 100% accuracy for all of them
- Try to get maximum number of 100% patterns for identity on 16 vectors. 4-8 is not enough, try to train at least 9-10 of them
- Increase the reliability of the outputs to 100% (anyone must believe the results)
- This can be achieved either by longer training or by decreasing the training parameter η
- When starting with a very small η the training takes too much time. Use very small η just in the final stage of training
- For example: 500 epochs η=0.3, next 500 epochs η=0.1, last 200 epochs η=0.01
- And/or try to play with η to change it continuously from 0.1 to 0.001 in the final period

Task 1 - Identity


```
Training progress
Epoch  1    Global error  12.25845
...
Epoch 40    Global error   0.04578

Testing
Input     Output    Response                Error Accuracy Reliability
1 1 0 0   1 1 0 0   0.92 1.00 0.03 0.08   0.014   100%   100%
0 0 1 1   0 0 1 1   0.12 0.23 0.95 0.84   0.095   100%   100%
1 0 1 0   1 0 1 0   0.76 0.31 0.58 0.08   0.337   100%    50%
0 1 0 1   0 1 0 1   0.73 0.88 0.14 0.76   0.625    75%   100%
0 0 0 0   0 0 0 0   0.17 0.21 0.22 0.12   0.136   100%   100%
```



- Engaging the gain of response function also helps to increase the reliability in the end of training
- All output data get closer to zeros and ones
- Do not use it in the starting epochs as it makes switching of the neurons more difficult

- Order of presenting the training set patterns:
- Sequential – easiest but temporal dependency
-
- Shuffled – compromised solution for black boxed frameworks, wasted time on shuffling each epoch
-
- Random – the best (recommended) solution
- Statistically safe (all patterns are trained with the same probability)
- Based on producer – consumer approach
- Can be combined with Monte Carlo method for huge training sets