Summary of the training functions in Matlab's NN toolbox

Vladimir Vacic

Training functions in Matlab's NN Toolbox:

Function name	Algorithm
trainb	Batch training with weight & bias learning rules
trainbfg	BFGS quasi-Newton backpropagation
trainbr	Bayesian regularization
trainc	Cyclical order incremental training w/learning functions
traincgb	Powell -Beale conjugate gradient backpropagation
traincgf	Fletcher-Powell conjugate gradient backpropagation
traincgp	Polak-Ribiere conjugate gradient backpropagation
traingd	Gradient descent backpropagation
traingdm	Gradient descent with momentum backpropagation
traingda	Gradient descent with adaptive lr backpropagation
traingdx	Gradient descent w/momentum & adaptive lr backpropagation
trainlm	Levenberg-Marquardt backpropagation
trainoss	One step secant backpropagation
trainr	Random order incremental training w/learning functions
trainrp	Resilient backpropagation (Rprop)
trains	Sequential order incremental training w/learning functions
trainscg	Scaled conjugate gradient backpropagation

Performance of different training functions:

Function name	training		validation		testing		time	
	mean	stdev	mean	stdev	mean	stdev	mean	stdev
trainb	0.6456	0.7246	0.6302	0.6946	0.6386	0.7081	2.511	3.3835
trainbfg	0.0096	0.0032	0.0199	0.0084	**0.0209**	0.0046	7.3219	4.5702
trainbr	7.6088	3.5328	18.9761	10.219	149.8294	32.2893	18.5063	8.927
trainc	0.0072	0.0015	*	*	0.0374	0.0066	466.072	163.5241
traincgb	0.0102	0.0026	0.0193	0.0069	**0.0203**	0.0059	4.3389	1.886
traincgf	0.0112	0.0033	0.0199	0.0091	**0.0202**	0.0051	4.9752	2.4127
traincgp	0.0114	0.003	0.0213	0.0093	**0.0216**	0.0045	4.0544	1.9337
traingd	0.0265	0.0055	0.0332	0.0099	0.0323	0.0029	13.003	4.4432
traingdm	0.5528	0.34	0.5556	0.3221	0.5592	0.3499	1.2875	0.3697
traingda	0.0244	0.0063	0.0293	0.0084	0.0310	0.0037	5.2	2.222
traingdx	0.0394	0.0312	0.0448	0.0317	0.0445	0.0274	5.4219	3.526
trainlm	0.0065	0.0027	0.0199	0.0066	**0.0231**	0.0037	8.5762	3.494
trainoss	0.013	0.0038	0.0204	0.0081	**0.0205**	0.0035	5.1703	2.8221
trainr	0.0077	0.0014	*	*	0.3319	0.0042	422.3888	148.2313
trainrp	0.0137	0.0045	0.0207	0.0059	**0.0229**	0.0035	7.4954	3.8277
trains	2.0723	1.5461	*	*	2.1834	1.6277	0.1893	0.0188
trainscg	0.0114	0.0035	0.0213	0.0109	**0.0218**	0.0073	4.3171	1.7394

* do not support validation vectors, algorithms ignored validation datasets

a) Training function details:

Trainb

```
(Batch training with weight & bias
learning rules)

      epochs: 100
        goal: 0
    max_fail: 5
        show: 25
        time: Inf
```

Trainbfg

```
(BFGS quasi-Newton backpropagation)

       epochs: 100
         show: 25
         goal: 0
         time: Inf
     min_grad: 1.0000e-006
     max_fail: 5
    searchFcn: 'srchbac'
    scale_tol: 20
        alpha: 0.0010
         beta: 0.1000
        delta: 0.0100
         gama: 0.1000
      low_lim: 0.1000
       up_lim: 0.5000
      maxstep: 100
      minstep: 1.0000e-006
         bmax: 26
```

Trainbr

```
(Bayesian regularization)

       epochs: 100
         show: 25
         goal: 0
         time: Inf
     min_grad: 1.0000e-010
     max_fail: 5
    mem_reduc: 1
           mu: 0.0050
       mu_dec: 0.1000
       mu_inc: 10
       mu_max: 1.0000e+010
```

Trainc

```
(Cyclical order incremental training
w/learning functions)

    epochs: 100
      goal: 0
      show: 25
      time: Inf
```

Traincgb

```
(Powell-Beale conjugate gradient
backpropagation)

       epochs: 100
         show: 25
         goal: 0
         time: Inf
     min_grad: 1.0000e-006
     max_fail: 5
    searchFcn: 'srchcha'
    scale_tol: 20
        alpha: 0.0010
         beta: 0.1000
        delta: 0.0100
         gama: 0.1000
      low_lim: 0.1000
       up_lim: 0.5000
      maxstep: 100
      minstep: 1.0000e-006
         bmax: 26
```

Traincgf

```
(Fletcher-Powell conjugate gradient
backpropagation)

       epochs: 100
         show: 25
         goal: 0
         time: Inf
     min_grad: 1.0000e-006
     max_fail: 5
    searchFcn: 'srchcha'
    scale_tol: 20
        alpha: 0.0010
         beta: 0.1000
        delta: 0.0100
         gama: 0.1000
      low_lim: 0.1000
       up_lim: 0.5000
      maxstep: 100
      minstep: 1.0000e-006
         bmax: 26
```

Traincgp

```
(Polak-Ribiere conjugate gradient
backpropagation)

       epochs: 100
         show: 25
         goal: 0
         time: Inf
     min_grad: 1.0000e-006
     max_fail: 5
```

```
    searchFcn: 'srchcha'
    scale_tol: 20
        alpha: 0.0010
         beta: 0.1000
        delta: 0.0100
         gama: 0.1000
      low_lim: 0.1000
       up_lim: 0.5000
      maxstep: 100
      minstep: 1.0000e-006
         bmax: 26
```

Traingd

(Gradient descent backpropagation)

```
      epochs: 100
        goal: 0
          lr: 0.0100
    max_fail: 5
    min_grad: 1.0000e-010
        show: 25
        time: Inf
```

Traingdm

(Gradient descent with momentum backpropagation)

```
      epochs: 100
        goal: 0
          lr: 0.0100
    max_fail: 5
          mc: 0.9000
    min_grad: 1.0000e-010
        show: 25
        time: Inf
```

Traingda

(Gradient descent with adaptive lr backpropagation)

```
          epochs: 100
            goal: 0
              lr: 0.0100
          lr_inc: 1.0500
          lr_dec: 0.7000
        max_fail: 5
    max_perf_inc: 1.0400
        min_grad: 1.0000e-006
            show: 25
            time: Inf
```

Traingdx

(Gradient descent w/momentum & adaptive lr backpropagation)

```
          epochs: 100
            goal: 0
              lr: 0.0100
          lr_dec: 0.7000
          lr_inc: 1.0500
        max_fail: 5
    max_perf_inc: 1.0400
              mc: 0.9000
        min_grad: 1.0000e-006
            show: 25
            time: Inf
```

Trainlm

(Levenberg-Marquardt backpropagation)

```
       epochs: 100
         goal: 0
     max_fail: 5
    mem_reduc: 1
     min_grad: 1.0000e-010
           mu: 0.0010
       mu_dec: 0.1000
       mu_inc: 10
       mu_max: 1.0000e+010
         show: 25
         time: Inf
```

Trainoss

(One step secant backpropagation)

```
       epochs: 100
         show: 25
         goal: 0
         time: Inf
     min_grad: 1.0000e-006
     max_fail: 5
    searchFcn: 'srchbac'
    scale_tol: 20
        alpha: 0.0010
         beta: 0.1000
        delta: 0.0100
         gama: 0.1000
      low_lim: 0.1000
       up_lim: 0.5000
      maxstep: 100
      minstep: 1.0000e-006
         bmax: 26
```

Trainr

(Random order incremental training w/learning functions)

```
    epochs: 100
      goal: 0
      show: 25
      time: Inf
```

Trainrp

(Resilient backpropagation - Rprop)

```
      epochs: 100
```

```
    show: 25
    goal: 0
    time: Inf
min_grad: 1.0000e-006
max_fail: 5
delt_inc: 1.2000
delt_dec: 0.5000
  delta0: 0.0700
deltamax: 50
```

Trains

```
(Sequential order incremental training
w/learning functions)

    passes: 1
```

Trainscg

```
(Scaled conjugate gradient
backpropagation)

      epochs: 100
        show: 25
        goal: 0
        time: Inf
    min_grad: 1.0000e-006
    max_fail: 5
       sigma: 5.0000e-005
      lambda: 5.0000e-007
```

Function name	Algorithm	# Parameters
Trainb	Batch training with weight & bias learning rules	5
trainbfg	BFGS quasi-Newton backpropagation	17
Trainbr	Bayesian regularization	11
trainc	Cyclical order incremental training w/learning functions	4
traincgb	Powell -Beale conjugate gradient backpropagation	17
traincgf	Fletcher-Powell conjugate gradient backpropagation	17
traincgp	Polak-Ribiere conjugate gradient backpropagation	17
traingd	Gradient descent backpropagation	7
traingdm	Gradient descent with momentum backpropagation	8
traingda	Gradient descent with adaptive lr backpropagation	10
traingdx	Gradient descent w/momentum & adaptive lr backpropagation	11
trainlm	Levenberg-Marquardt backpropagation	11
trainoss	One step secant backpropagation	17
trainr	Random order incremental training w/learning functions	4
trainrp	Resilient backpropagation (Rprop)	10
trains	Sequential order incremental training w/learning functions	1
trainscg	Scaled conjugate gradient backpropagation	8

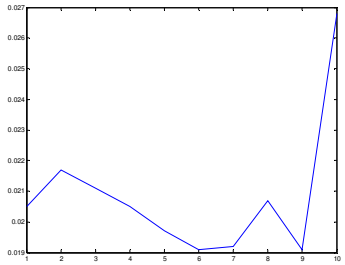
Optimizing NN training function parameters:

Since the number of free parameters was too big for an exhaustive analysis, only three functions that perform well with the default parameters were selected for fine tuning, in hopes of having them perform even better. For brevity and clarity, only the performance on the test set was reported.

trainbfg (BFGS quasi-Newton backpropagation)

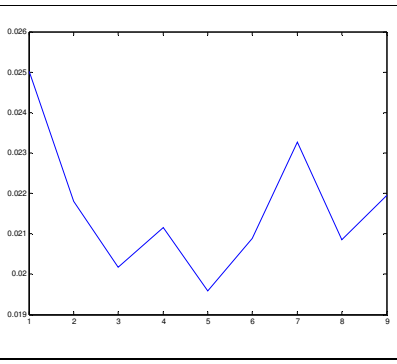
Varying the min_grad parameter did not yield any significant increases nor decreases in accuracy and running time, except when the values became really big (4 orders of magnitude bigger than the default value).

min_grad	mean(mse)	std	mean(time)	std
1.00E-10	0.0205	0.0051	6.8422	3.1826
1.00E-09	0.0217	0.0048	5.9687	3.8304
1.00E-08	0.0211	0.0043	5.9157	3.9046
1.00E-07	0.0205	0.0036	5.8890	2.6894
1.00E-06	0.0197	0.0031	5.6061	3.2403
1.00E-05	0.0191	0.0033	5.4954	2.7394
1.00E-04	0.0192	0.0032	5.2858	2.7135
1.00E-03	0.0207	0.0043	5.0813	2.2273
1.00E-02	0.0191	0.0024	5.5312	2.4237
1.00E-01	0.0268	0.0038	1.9578	0.6510



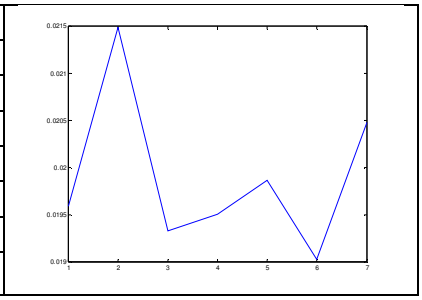
Scale_tols in the range of [30, 5] seem to perform better then the default parameter value 20.

Scale_tol	mean(mse)	std	mean(time)	std
100	0.0250	0.0080	5.8173	3.1563
50	0.0218	0.0039	4.5469	2.0129
30	**0.0202**	**0.0033**	**5.6533**	**2.2628**
20	**0.0212**	**0.0048**	**4.8873**	**2.1727**
10	**0.0196**	**0.0043**	**5.6171**	**2.5485**
5	**0.0209**	**0.0046**	**5.4219**	**2.6609**
3	0.0233	0.0059	5.4159	3.4750
2	0.0209	0.0048	5.0751	3.1815
1	0.0220	0.0063	5.6748	2.9298



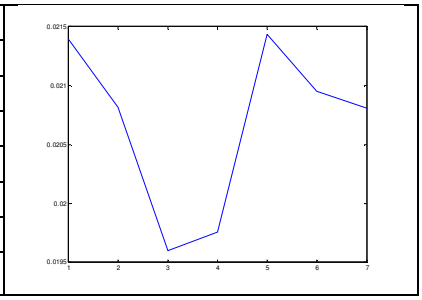
Varying alpha obtained better results at the expense of longer running times. Values above 0.1 were not feasible to use due to prohibitively long training time (probably the algorithm would not converge).

alpha	mean(mse)	std	mean(time)	std
0.1	0.0196	0.0053	5.8435	2.9714
0.05	0.0215	0.0077	5.9922	2.9526
0.01	**0.0193**	**0.0046**	**8.0282**	**5.3354**
0.005	**0.0195**	**0.0061**	**6.5624**	**2.9396**
0.001	**0.0199**	**0.0041**	**6.0922**	**3.0140**
0.0005	**0.0190**	**0.0031**	**6.6971**	**3.2757**
0.0001	0.0205	0.0041	5.3377	3.0052



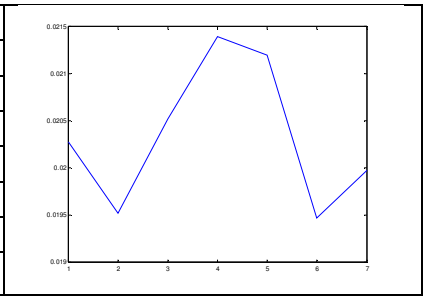
Varying beta also gave us a range of good possible values:

beta	mean(mse)	std	mean(time)	std
0.1	0.0214	0.0047	6.3846	3.1003
0.05	**0.0208**	**0.0033**	**6.5406**	**2.5756**
0.01	**0.0196**	**0.0037**	**6.1031**	**2.9170**
0.005	**0.0198**	**0.0026**	**7.5577**	**3.6887**
0.001	0.0214	0.0051	5.9122	2.7882
0.0005	0.0210	0.0035	5.8984	3.0984
0.0001	0.0208	0.0061	5.7998	2.8141



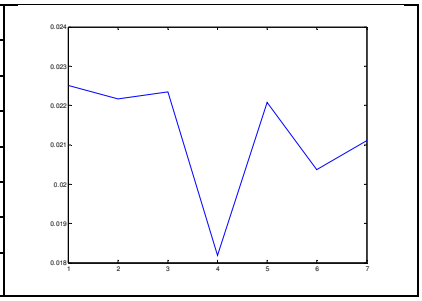
Varying delta did not determine a candidate or candidate range for the best possible prediction:

delta	mean(mse)	std	mean(time)	std
0.1	0.0203	0.0036	6.0764	3.2230
0.05	0.0195	0.0040	5.6482	2.8962
0.01	0.0205	0.0039	6.0282	2.6551
0.005	0.0214	0.0062	6.4155	3.7637
0.001	0.0212	0.0078	5.8043	2.6148
0.0005	0.0195	0.0032	5.4079	2.4490
0.0001	0.0200	0.0038	4.7436	1.8676



Varying gamma we obtained that value of 0.005 yields the best prediction:

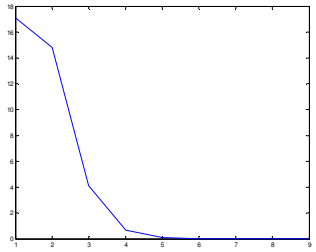
gamma	mean(mse)	std	mean(time)	std
0.1	0.0225	0.0063	7.1439	3.9628
0.05	0.0222	0.0060	6.5423	3.0900
0.01	0.0224	0.0055	5.8436	3.2640
0.005	**0.0182**	**0.0020**	**6.3408**	**3.1270**
0.001	0.0221	0.0068	6.0769	2.7400
0.0005	0.0204	0.0038	5.5592	2.9364
0.0001	0.0211	0.0041	6.4016	2.9683



trainbrp (Resilient backpropagation)

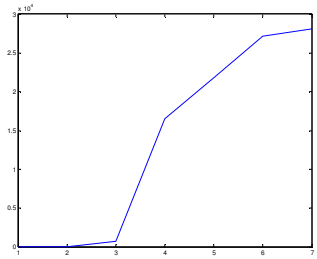
Accuracy monotonically increases with the decrease in delta0:

Delta0	mean(mse)	std	mean(time)	std
3	17.1146	10.9054	0.7090	0.1956
2	14.8043	11.5706	1.2808	1.8624
1	4.1029	4.7021	2.1822	2.5085
0.5	0.6666	0.9552	3.3709	2.8963
0.2	0.1122	0.1683	3.5992	2.4327
0.1	0.0272	0.0088	4.6016	2.2128
0.05	0.0231	0.0020	4.3931	1.9445
0.02	0.0241	0.0037	3.7456	2.1454
0.01	0.0271	0.0069	3.6063	2.1918



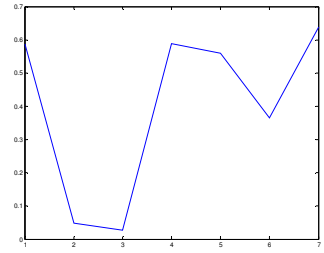
Accuracy monotonically decreases with the increase in delt_inc:

delt_inc	mean(mse)	std	mean(time)	std
2	0.0001	0.0002	1.1428	0.7061
3	0.0004	0.0002	0.6551	0.2762
5	0.0734	0.1151	0.6188	0.1825
20	1.6494	2.9597	0.7650	0.2391
30	2.1809	1.6639	0.7491	0.2189
50	2.7150	2.6276	0.8731	0.2515
100	2.8118	2.2356	0.9153	0.2703



Best experimentally determined value for delt_dec was 0.2.

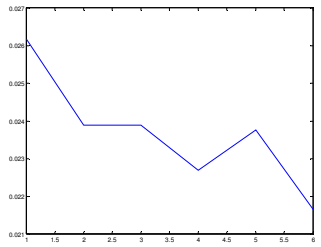
delt_dec	mean(mse)	std	mean(time)	std
1	0.5870	0.2982	1.3856	1.2098
0.5	0.0485	0.0744	3.6001	2.0620
0.2	**0.0280**	**0.0043**	**4.5614**	**2.0696**
0.1	0.5886	0.6400	2.5849	2.7867
0.05	0.5586	1.0577	4.4033	2.7499
0.02	0.3650	0.5901	3.6333	3.0410
0.01	0.6378	1.0649	3.8626	2.9429



trainblm (Levenberg-Marquardt backpropagation)

Accuracy monotonically increases with the decrease in mu.

mu	mean(mse)	std	mean(time)	std
0.05	0.0262	0.0078	7.2922	2.9351
0.01	0.0239	0.0027	8.4438	3.5154
0.005	0.0239	0.0047	8.7093	3.5562
0.001	0.0227	0.0030	8.7765	3.3478
0.0005	0.0238	0.0039	10.3217	5.1618
0.0001	**0.0217**	**0.0043**	**8.6296**	**3.7355**



Varying mu_dec did not significantly influence the accuracy in any observable way:

Mu_dec	mean(mse)	std	mean(time)	std
0.05	0.0242	0.0040	10.0469	4.4332
0.01	0.0229	0.0041	11.0894	5.2101
0.005	0.0256	0.0065	12.2141	5.6052
0.001	0.0254	0.0071	12.7764	5.0435
0.0005	0.0266	0.0061	12.4922	7.3443
0.0001	0.0231	0.0047	17.5905	9.2759