

# Implementação de rede *Multilayer Perceptron* (MLP)

por Eduardo S. M. de Vasconcelos  
NUSP 7656724 | eduardovasconcelos@usp.br

São Carlos, 08 de abril de 2016

• • •

## Introdução

Este trabalho tem como objetivo servir como objeto de avaliação parcial na disciplina Redes Neurais (SCC0570), ministrada pelo Dr. Pablo Andretta Jaskowiak durante o primeiro semestre letivo de 2016 no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC/USP).

O objetivo do trabalho é a implementação de uma rede *Multilayer Perceptron* (MLP) capaz de classificar dígitos decimais. Dois formatos de entrada foram sugeridos na proposta de projeto: os dígitos tanto poderiam ser lidos de imagens no formato BMP quanto de arquivos TXT de arte ASCII. A rede implementada neste trabalho é capaz de lidar com ambos os formatos de entrada.

Este documento é dividido em três seções, sendo esta *Introdução* a primeira delas, seguida da seção *Descrição do problema*, onde discute-se a base de dados utilizada, seu pré-processamento e as características gerais da implementação realizada. Na seção *Resultados* é discutida a performance da rede obtida com relação ao seu conjunto de teste. Especial destaque é dado ainda aos casos negativos (i.e. para os quais a rede não tem boa performance). Discute-se ainda as dificuldades encontradas na realização do trabalho.

Para referência sobre a execução do código, vide arquivo “*README.txt*”, que acompanha este documento.

## Descrição do problema

Duas bases distintas foram utilizadas na implementação da rede. A primeira consiste nos dez dígitos decimais (de 0 a 9) em imagens BMP fornecidas junto com a proposta de projeto. Optou-se por diminuir o tamanho das imagens para 8x8 *pixels* como forma de reduzir o número de entradas da rede. As imagens originais (de 55x74 *pixels*) resultariam em vetores de entrada com 4071 elementos, ao passo que as imagens reduzidas fornecem vetores de entrada de 65 elementos. Ainda assim, todos os dígitos continuam inteligíveis por humanos sem necessidade de ampliação das imagens. A segunda base utilizada consiste nos dez dígitos decimais em arquivos TXT de arte ASCII. São dez arquivos de texto com 8 linhas e 8 colunas cada, cada um representando um algarismo decimal. As duas bases encontram-se nos diretórios “*train\_bmp*” e “*train\_ascii*”, respectivamente.

A partir de cada base de dados criou-se três conjuntos de teste, cada um com um certo nível de ruído introduzido, respectivamente de 10%, 20% e 30% do tamanho (em *pixels* ou caracteres) dos arquivos de entrada. Os conjuntos de teste podem ser encontrados nos diretórios “*test\_ascii\_10*”, “*test\_ascii\_20*” e “*test\_ascii\_30*” (para a base de dados em texto) e “*test\_bmp\_10*”, “*test\_bmp\_20*” e

“*test\_bmp\_30*” (para a base de dados em BMP). Para a base de dados em texto, o ruído foi representado por caracteres “@”. Já para a base BMP, o ruído foi representado por *pixels* com valor 0 (i.e. cor preta).

Desenvolveu-se o projeto em Octave, devido ao seu robusto conjunto de operadores matriciais e à possibilidade de concentrar esforços na resolução do problema, sem muitas adversidades inerentes ao nível da linguagem, como seria o caso se fosse utilizada uma linguagem de mais baixo nível.

Após fazer a leitura dos arquivos de treinamento para uma matriz de entrada  $X$  (com cada exemplo de entrada em uma linha da matriz), esta passa por um pré-processamento dividido em duas fases:

- a) Substituição de  $x_{ij}$  por  $f(x_{ij}) \in \{-1, 1\}$ : isso se justifica pela utilização da função tangente hiperbólica como função de ativação dos neurônios da rede (a saída da função tangente hiperbólica tem como saída um número real no intervalo  $[-1, 1]$  e pela condição de cada elemento da camada de entrada de uma rede MLP ser binário.
- b) Eliminação de colunas inúteis: todas as colunas de  $X$  que são inúteis à segregação de classes pela MLP são descartadas e seu índice é guardado. Entenda-se por coluna inútil uma coluna cujos valores são iguais para todos os exemplos de treinamento. Colunas assim apenas aumentam a complexidade do treinamento sem agregar nada à tarefa de classificação da rede. O índice de cada coluna eliminada é guardado e durante a fase de testes as mesmas colunas são removidas das matrizes de teste.

Definiu-se uma arquitetura de rede constituída de camada de entrada com tantos elementos quanto o pré-processamento forneça (devido à fase de eliminação de colunas inúteis), uma camada oculta com 20 neurônios e uma camada de saída com 4 neurônios (justificada adiante). Assim, obteve-se para as bases de dados em TXT e BMP as arquiteturas  $[42, 20, 4]$  e  $[48, 20, 4]$ , respectivamente. Em ambos os casos, utilizou-se os parâmetros listados abaixo:

- Taxa de aprendizado ( $\eta$ ) = 0,5;
- Fator multiplicativo do campo induzido na tangente hiperbólica ( $a$ ) = 0,5;
- Fator multiplicativo do *momentum* ( $\alpha$ ) = 0,5;
- Número máximo de épocas de treinamento ( $N$ ) = 10000;
- Erro quadrático médio admissível ( $\epsilon$ ) =  $10^{-3}$ .

Quanto à camada de saída com 4 neurônios, a escolha foi feita baseada na observação de que a rede convergia mais rápido para saídas codificadas utilizando o código *Gray*. Tentou-se ainda utilizar o código BCD e um código baseado em posição, com 10 neurônios de saída (sendo que cada neurônio ficaria responsável por identificar um dígito). Para a saída codificada em código *Gray* o algoritmo de treinamento apresenta convergência estupidamente mais rápida nas arquiteturas utilizadas.

Como a inicialização das matrizes de pesos no algoritmo *backpropagation* foi feita de maneira aleatória, tomou-se o cuidado de tornar a MLP determinística ao fixar os valores de *seed* aleatória para cada base de dados.

Para a classificação de vetores de teste, utilizou-se a distância cartesiana entre as respostas possíveis e a resposta calculada pela rede para o vetor, sendo tomado como resposta o dígito cuja distância à resposta obtida era menor.

# Resultados

## Conjunto de testes

**GRUPO DE CONTROLE (TREINAMENTO)** O gráfico mostrado na Figura 1 apresenta o valor de erro quadrático médio “*em*” em função da época de treinamento “*n*” para a base de dados de arte ASCII.

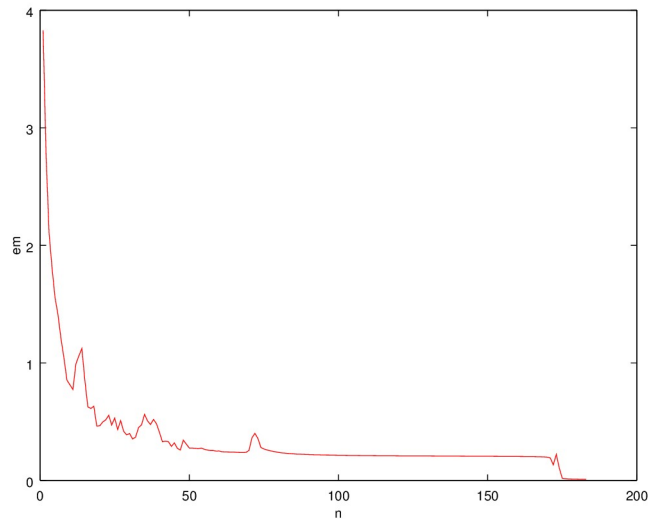


Figura 1. Erro quadrático médio “*em*” em função da época de treinamento “*n*” para a base ASCII. O eixo vertical representa o erro quadrático médio e o eixo horizontal representa a época de treinamento.

A Figura 2 apresenta o valor de erro quadrático médio “*em*” em função da época de treinamento “*n*” para a base de dados de imagens BMP.

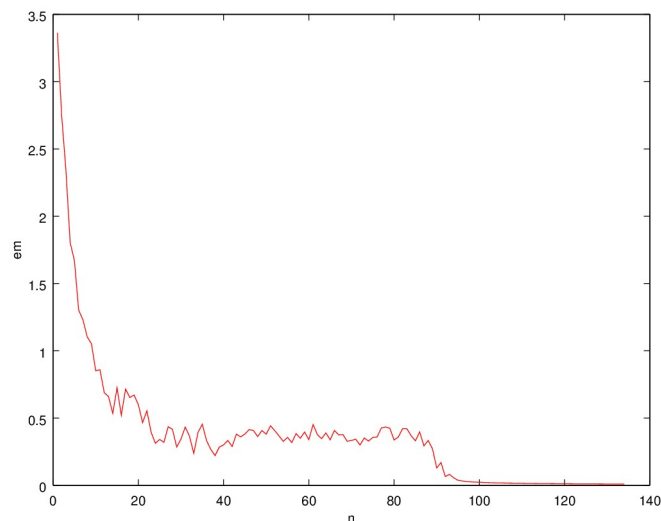


Figura 2. Erro quadrático médio “*em*” em função da época de treinamento “*n*” para a base BMP. O eixo vertical representa o erro quadrático médio e o eixo horizontal representa a época de treinamento.

A Tabela 1 mostra a matriz de saídas esperadas para cada exemplo de treinamento. Para as bases de dados ASCII e BMP, as saídas de treinamento são mostradas nas Tabelas 2 e 3, respectivamente.

-1	-1	-1	-1
-1	-1	-1	1
-1	-1	1	1
-1	-1	1	-1
-1	1	1	-1
-1	1	1	1
-1	1	-1	1
-1	1	-1	-1
1	1	-1	-1
1	1	-1	1

Tabela 1. Matriz de saídas esperadas para cada exemplo de treinamento. Cada dígito é representado por uma linha da matriz. A primeira linha é a saída esperada para o dígito "0" e a última, para o "9".

-0.9765762969102911	-0.8297887469755326	-0.9884105557261843	-0.8960173810997045
-0.9542210034873935	-0.9225646344507785	-0.9999683167828146	0.9547769477330575
-0.9997726478200293	-0.907711463965631	0.9448739504512121	0.9767296830019084
-0.9292632220178789	-0.9617876635079579	0.9612233371958872	-0.9874123308813162
-0.9385867187343556	0.9957655629419256	0.9994210679385841	-0.9505827127107942
-0.9903346180215186	0.8970619740150445	0.9947398451669792	0.9042111930566508
-0.9834633616125887	0.8846760518974853	-0.9412329537323484	0.9856813338751782
-0.9964436083594782	0.9077270209145861	-0.9307705411751331	-0.9187504410480498
0.9168136306636386	0.9982682280498023	-0.9593731311857324	-0.9172787495655563
0.9517458064980564	0.8673302921036431	-0.9738501892489975	0.8758815383678049

Tabela 2. Matriz de saídas obtidas no treinamento da base ASCII (avaliadas com os próprios exemplos de treinamento).

-0.8857456639374189	-0.8421166781283143	-0.9315854311876209	-0.9658292699453351
-0.9416330414481509	-0.9162711574938168	-0.9792888608962869	0.9985446457214088
-0.9969343442195374	-0.9844170402822709	0.8894992255353247	0.9249916625245699
-0.9981749531634401	-0.9539036705527791	0.9989491296441633	-0.9672500865931861
-0.9989521517027112	0.9337526676498872	0.8834400617279342	-0.9996646316372747
-0.9324678667723143	0.9985999132199871	0.9578892185250717	0.9337015735227198
-0.9571975676936364	0.9904886322630704	-0.9302806817724671	0.9862095765798813
-0.9985308850724742	0.9181869396410899	-0.8524401228549535	-0.9339272586280311
0.8674900353732029	0.8542884133623406	-0.981412841364622	-0.9352031983040212
0.9858976097759281	0.9851654644737915	-0.9999197488699414	0.9994373006177055

Tabela 3. Matriz de saídas obtidas no treinamento da base BMP (avaliadas com os próprios exemplos de treinamento).

Observa-se que as saídas obtidas estão em conformidade com as saídas esperadas levando em consideração o erro quadrático médio admissível de  $10^{-3}$ . Os resultados dos testes para níveis de ruído de 10%, 20% e 30% são mostrados a seguir.

**GRUPO DE TESTE (RUÍDO DE 10%)** As saídas obtidas para as bases ASCII e BMP para nível de ruído de 10% são mostradas nas Tabelas 4 e 5, respectivamente. As matrizes de confusão associadas são mostradas nas Tabelas 6 e 7.

-0.9787740064605451	-0.8378301684784993	-0.9884102768824048	-0.8808280442031458
-0.9358402589183576	-0.7852066515251744	-0.9999361335495013	0.9605100486563588
-0.9997517394429671	-0.8748619059668745	0.9112256412328079	0.9684824850170798
-0.9370649458204477	-0.9354667164358679	0.9622199564481053	-0.9792102321893283
-0.9388850950355776	0.9956920286213585	0.9994536698957264	-0.9532098076535179
-0.9936941103410623	0.8479987565262376	0.9957265309407012	0.9139679982857497
-0.981411593294604	0.8088470173952992	-0.9189941031816926	0.9499637239731973
-0.9967804983850402	0.9066907985088756	-0.9290348357836581	-0.8578520601684276
0.9168136306636386	0.9982682280498023	-0.9593731311857324	-0.9172787495655563
0.7599957163864518	0.9914132066766973	-0.9697740242331134	0.5713579771738627

Tabela 4. Matriz de saídas obtidas no teste da base ASCII para ruído de 10%.

-0.9018038747800915	-0.8317063098348868	-0.9044106112138803	-0.9633974548239169
-0.981404601979758	-0.920565531154015	-0.9725467267983928	0.9985901977152544
-0.996850033244534	-0.9826683140555059	0.8766217775660491	0.9118165253417558
-0.9981831658128122	-0.9001027735099788	0.9982221986459647	-0.9897738310713367
-0.9989776036019518	0.9329592000445892	0.8852685166569937	-0.9996722420083559
-0.9648563381758846	0.9909641883172	0.965875229290164	0.7953814185932756
-0.9145094984734141	0.9937255289907757	-0.9218058590862617	0.9872204272604557
-0.9976841849498724	0.8844037893383279	-0.8750215895404386	-0.8858439507285801
0.8674900353732029	0.8542884133623406	-0.981412841364622	-0.9352031983040212
0.9921142915687046	0.9468787757982755	-0.9995529622163797	0.9926021250144403

Tabela 5. Matriz de saídas obtidas no teste da base BMP para ruído de 10%.

	0	1	2	3	4	5	6	7	8	9
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1

Tabela 6. Matriz de confusão para o teste da base ASCII com ruído de 10%. Os índices verticais representam o dígito real e os horizontais representam a classificação calculada pela MLP.

	0	1	2	3	4	5	6	7	8	9
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1

Tabela 7. Matriz de confusão para o teste da base BMP com ruído de 10%. Os índices verticais representam o dígito real e os horizontais representam a classificação calculada pela MLP.

**GRUPO DE TESTE (RÚIDO DE 20%)** As saídas obtidas para as bases ASCII e BMP para nível de ruído de 20% são mostradas nas Tabelas 8 e 9, respectivamente. As matrizes de confusão associadas são mostradas nas Tabelas 10 e 11.

-0.7828178218149069	0.1745765994041812	-0.9746164283267323	-0.9501902542199751
-0.5253820901701385	0.07918831026441328	-0.9999820750643716	0.8167133640158737
-0.9985053340437706	-0.973710650550585	0.9983303637299236	0.3512089158081011
-0.9254864587133034	-0.9106325578532286	0.9670949546265538	-0.9657021058817299
-0.9294932420587491	0.9953354521689471	0.9994804214984629	-0.958432706818477
0.01237150299901211	0.9519685295971976	0.9992597047485771	-0.1184770629846993
-0.9574367171168214	0.4929313496458756	0.9920185682582281	0.123309633620774
-0.9964770884435993	-0.9823123606227528	0.9947984328650167	-0.9857373001991301
0.5409827135142954	0.9944302761060166	-0.5927465846745861	-0.8385310755475942
0.9247470925967212	0.6514127427926334	-0.9643843719488769	-0.9625116948427271

Tabela 8. Matriz de saídas obtidas no teste da base ASCII para ruído de 20%.

-0.8948488123832958	-0.8272737729817069	-0.9308426061545553	-0.9696392472632288
-0.7573756345494798	-0.9617111357664846	-0.9096865232692337	0.8753860641742131
-0.9978572608776551	-0.9306270232560612	0.8915206725810847	0.3757841062946028
-0.995350208509222	-0.920821359498562	0.9982056681401547	-0.9141743069596725
-0.9970809156707631	0.9712232001422999	0.8293353355633133	-0.9995675043215038
-0.5872296078199031	0.9410888820036817	0.673976569661084	-0.448080056577717
-0.7057548422130347	-0.5558273015520649	-0.9639833300486421	-0.9531826838711721
-0.9990564073611057	0.9515537068923372	-0.8298886755818895	-0.9936408695418908
0.6917417190173276	0.7907284362031891	-0.9763245991754972	-0.9519006896613632
0.9932880003339732	0.9546226160442296	-0.9998988591496417	0.9965794433794468

Tabela 9. Matriz de saídas obtidas no teste da base BMP para ruído de 20%.

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	1	0

Tabela 10. Matriz de confusão para o teste da base ASCII com ruído de 20%. Os índices verticais representam o dígito real e os horizontais representam a classificação calculada pela MLP.

	0	1	2	3	4	5	6	7	8	9
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1

Tabela 11. Matriz de confusão para o teste da base BMP com ruído de 20%. Os índices verticais representam o dígito real e os horizontais representam a classificação calculada pela MLP.

**GRUPO DE TESTE (RÚIDO DE 30%)** As saídas obtidas para as bases ASCII e BMP para nível de ruído de 30% são mostradas nas Tabelas 12 e 13, respectivamente. As matrizes de confusão associadas são mostradas nas Tabelas 14 e 15.

-0.9044134391082596	-0.1318841256487745	-0.4168781161102297	-0.980242096613776
-0.9933069235525502	-0.3701534293049226	-0.9760178740801749	0.906799220137068
-0.9996006121038876	-0.9369587655633737	0.9701716862580255	0.9074364387095891
-0.9809116321809478	0.7754882809282699	0.9905659627011739	0.6588194219152458
0.857267078194168	0.884385963868806	0.9999577590476609	-0.8853861551117824
-0.990964542879104	0.8846999844877055	0.9953064995145856	0.9005418241515254
0.9755279192625372	0.8593926646022174	0.9965059114105801	-0.8406401879472089
-0.8368512267867996	-0.9620939356516498	0.9993678775158668	-0.9370314362807625
0.8205365153525273	0.9974761056144391	-0.87773579407514	-0.8691775301916191
-0.05803055481372599	0.9553895616461491	-0.7923971464032511	-0.9923889896424156

Tabela 12. Matriz de saídas obtidas no teste da base ASCII para ruído de 30%.

-0.9150027191870768	-0.3067834270539622	-0.7353927007498713	-0.9754909470641543
-0.998509872165407	-0.9465180101495034	0.9781776633631688	-0.9050132980118752
-0.9981941712172927	-0.9236207475600378	0.9452999359364812	-0.7765132894490658
-0.99825659791627	-0.8052284517606192	0.9970904350044044	-0.9944582269460543
-0.9978945775676211	0.9438000685755783	0.8469172920256001	-0.9996426961605898
-0.9269458154173639	0.9905446888547609	0.9121123537186842	0.6252562867758003
-0.9239527568835909	0.855967221723691	-0.8993767974910849	-0.1007401140023746
-0.9990575110057016	0.9476632287296536	-0.5119210914774512	-0.9971408576267554
0.1528979059448934	0.8491700895108325	-0.8194393045122469	-0.9451791118829994
0.9771223534655201	0.08835511785763671	-0.9783846554956215	-0.02694590221278876

Tabela 13. Matriz de saídas obtidas no teste da base BMP para ruído de 30%.

	0	1	2	3	4	5	6	7	8	9
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	0	0	1	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	1	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	1	0	0

Tabela 14. Matriz de confusão para o teste da base ASCII com ruído de 30%. Os índices verticais representam o dígito real e os horizontais representam a classificação calculada pela MLP.

	0	1	2	3	4	5	6	7	8	9
0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	1	0

Tabela 15. Matriz de confusão para o teste da base BMP com ruído de 30%. Os índices verticais representam o dígito real e os horizontais representam a classificação calculada pela MLP.

Observa-se que para o grupo de teste com ruído de 10% a rede não cometeu erros de classificação, obtendo taxa de acertos de 100% para ambas as bases de dados. Todavia, para o grupo de ruído de 20% a rede começou a errar. As taxas de acerto associadas às bases ASCII e BMP para esse grupo são, respectivamente, 40% e 80%. Para o grupo com ruído de 30%, as taxas de acerto são ambas de 60%.

## Casos negativos

A MLP implementada obteve melhores resultados na classificação de dígitos com ruído mais baixo. A taxa de acerto de classificação para ruído de 10% foi de 100%. Nos casos com ruído de 20% e 30% a rede não se mostrou confiável, especialmente para a base ASCII. Chamou a atenção a rede não ter sido capaz de classificar corretamente o dígito “6” em *nenhum* dos casos com ruído acima de 10%. O dígito “6” foi classificado como “5”, “0”, “4” e “7”, mas jamais corretamente.

Os maiores obstáculos encontrados durante o desenvolvimento deste trabalho, a despeito do que seu autor acreditava, apareceram na fase após a implementação do código. O autor estava utilizando uma função *XOR* para testar a rede durante todo o processo de desenvolvimento do código. A “transição” para as bases de dados de dígitos foi muito sofrida, pois foi preciso “encontrar” os valores certos para os parâmetros da rede (definidos em *Descrição do problema*). Um valor bom para o fator multiplicativo do *momentum* foi especialmente difícil de encontrar.

Outra dificuldade encontrada foi a definição do número de neurônios na camada de saída. Conforme discutido, os códigos BCD (4 neurônios) e de posição (10 neurônios) foram testados como saídas esperadas da rede. Surgiram de maneira muito natural. No entanto, nenhum dos dois mostrou-se eficiente. A rede só começou a operar de forma satisfatória quando a saída esperada foi definida



usando o código *Gray* de 4 bits (com 4 neurônios na saída), associada à eliminação das colunas inúteis das matrizes de entrada no treinamento e no teste, conforme já discutido.