



**Universidade Federal de Ouro Preto**  
**DECAT - Departamento de Engenharia de**  
**Controle e Automação**



**CAT168 - Trabalho Individual 2**

**Antônio Henrique Alves Cardoso – 14.2.1687**

**Ouro Preto**  
**Julho de 2019**

## Problema 1

Altere a rotina `appcr1.m/prprob` de modo que somente os caracteres de A a J, maiúsculos e minúsculos (a serem criados), possam ser identificados por uma rede neural do tipo *Perceptrons* de Múltiplas Camadas (PMC). Considere 5 cópias de cada caractere e adição de ruído de até 20%. Faça uma análise crítica dos resultados obtidos. Priorize a utilização de modelos parcimoniosos sem que a capacidade de generalização deles seja prejudicada.

Alterada a rotina `appcr1.m/prprob` de modo que somente os caracteres de A a J.

```

letterA = [0 0 1 0 0 ...
           0 1 0 1 0 ...
           0 1 0 1 0 ...
           1 0 0 0 1 ...
           1 1 1 1 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ]';

letterC = [0 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 1 ...
           0 1 1 1 0 ]';

letterE = [1 1 1 1 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 1 ]';

lettera = [0 0 0 0 0 ...
           0 0 0 0 0 ...
           1 1 1 1 0 ...
           0 0 0 1 0 ...
           1 1 1 1 0 ...
           1 0 0 1 0 ...
           1 1 1 1 1 ]';

letterc = [0 0 0 0 0 ...
           0 0 0 0 0 ...
           0 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 0 ...
           1 0 0 0 1 ...
           0 1 1 1 0 ]';

lettere = [0 0 0 0 0 ...
           0 0 0 0 0 ...
           1 1 1 1 0 ...
           1 0 0 0 1 ...
           1 1 1 1 0 ...
           1 0 0 0 0 ...
           1 1 1 1 1 ]';

letterB = [1 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 1 1 1 0 ]';

letterD = [1 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 1 1 1 0 ]';

letterF = [1 1 1 1 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ]';

letterb = [1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 0 ...

           letterG = [0 1 1 1 0 ...
                       1 0 0 0 1 ...
                       1 0 0 0 0 ...
                       1 0 0 0 0 ...
                       1 0 0 1 1 ...
                       1 0 0 0 1 ...
                       0 1 1 1 0 ]';

           letterI = [0 1 1 1 0 ...
                       0 0 1 0 0 ...
                       0 0 1 0 0 ...
                       0 0 1 0 0 ...
                       0 0 1 0 0 ...
                       0 0 1 0 0 ...
                       0 1 1 1 0 ]';

           letterg = [0 1 1 1 0 ...
                       1 0 0 0 1 ...
                       1 0 0 0 1 ...
                       0 1 1 1 1 ...
                       0 0 0 0 1 ...
                       1 0 0 0 1 ...
                       0 1 1 1 0 ]';

           letteri = [0 0 0 0 0 ...
                       0 0 1 0 0 ...
                       0 0 0 0 0 ...
                       0 0 1 0 0 ...
                       0 0 1 0 0 ...
                       0 0 1 0 0 ...
                       0 1 1 1 0 ]';

           letterH = [1 0 0 0 1 ...
                       1 0 0 0 1 ...
                       1 0 0 0 1 ...
                       1 1 1 1 1 ...
                       1 0 0 0 1 ...
                       1 0 0 0 1 ...
                       1 0 0 0 1 ]';

           letterJ = [1 1 1 1 1 ...
                       0 0 1 0 0 ...
                       0 0 1 0 0 ...
                       0 0 1 0 0 ...
                       0 0 1 0 0 ...
                       1 0 1 0 0 ...
                       0 1 0 0 0 ]';

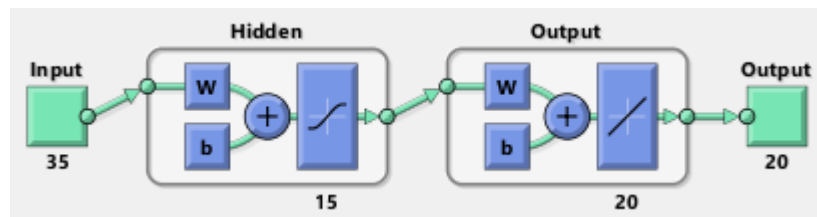
           letterh = [1 0 0 0 0 ...
                       1 0 0 0 0 ...
                       1 0 0 0 0 ...
                       1 1 1 1 0 ...
                       1 0 0 0 1 ...

           letterj = [0 0 0 0 0 ...
                       0 0 1 0 0 ...
                       0 0 0 0 0 ...
                       0 0 1 0 0 ...

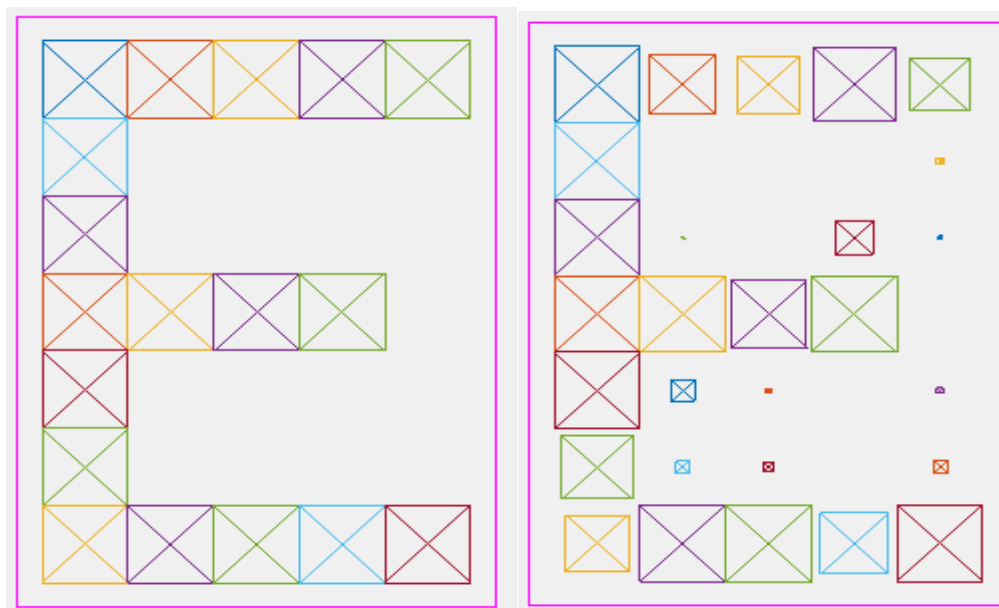
```

```
alfabeto = [letterA, lettera, letterB, letterb, letterC, letterc, letterD, ...
            letterd, letterE, lettere, letterF, letterf, letterG, letterg, ...
            letterH, letterh, letterI, letteri, letterJ, letterj];
outputs = eye(20);
```

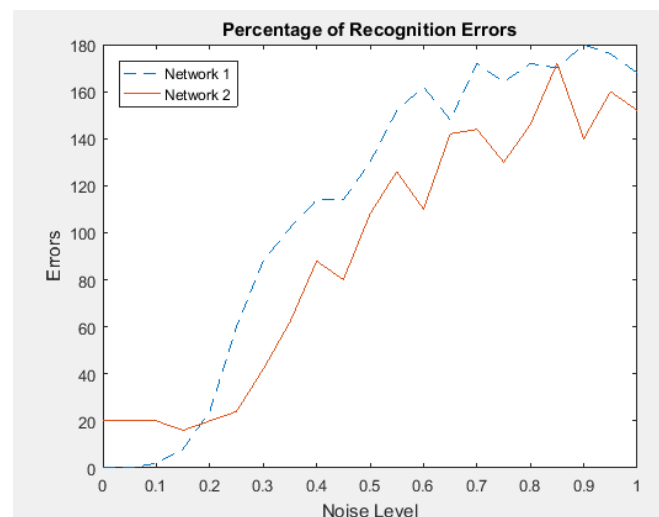
Neural Network



Caracteres 'e' gerados a partir do treino da RNA, com e sem ruído respectivamente:



Porcentagem de erros comparando as redes treinadas com ruído de 20% (net2) e sem ruído (net1), com 15 neurônios na camada oculta.



## Problema 2

Considere a base de dados Series\_G.xls. (a) Conceba um preditor neural que faça previsões de 1 passo à frente utilizando a normalização entre 0 e 1 como técnica de pré-processamento de dados. 'Plote' em uma curva os dados medidos e os dados estimados.

Utilizei 130 amostras, onde:

- 70% delas para treinamento
- 15% para teste e outros
- 15% para validação

Foi desenvolvida uma rede neural com atraso de 1 amostra para realizar a previsão da amostra  $T_i$  em que  $i=131$ .

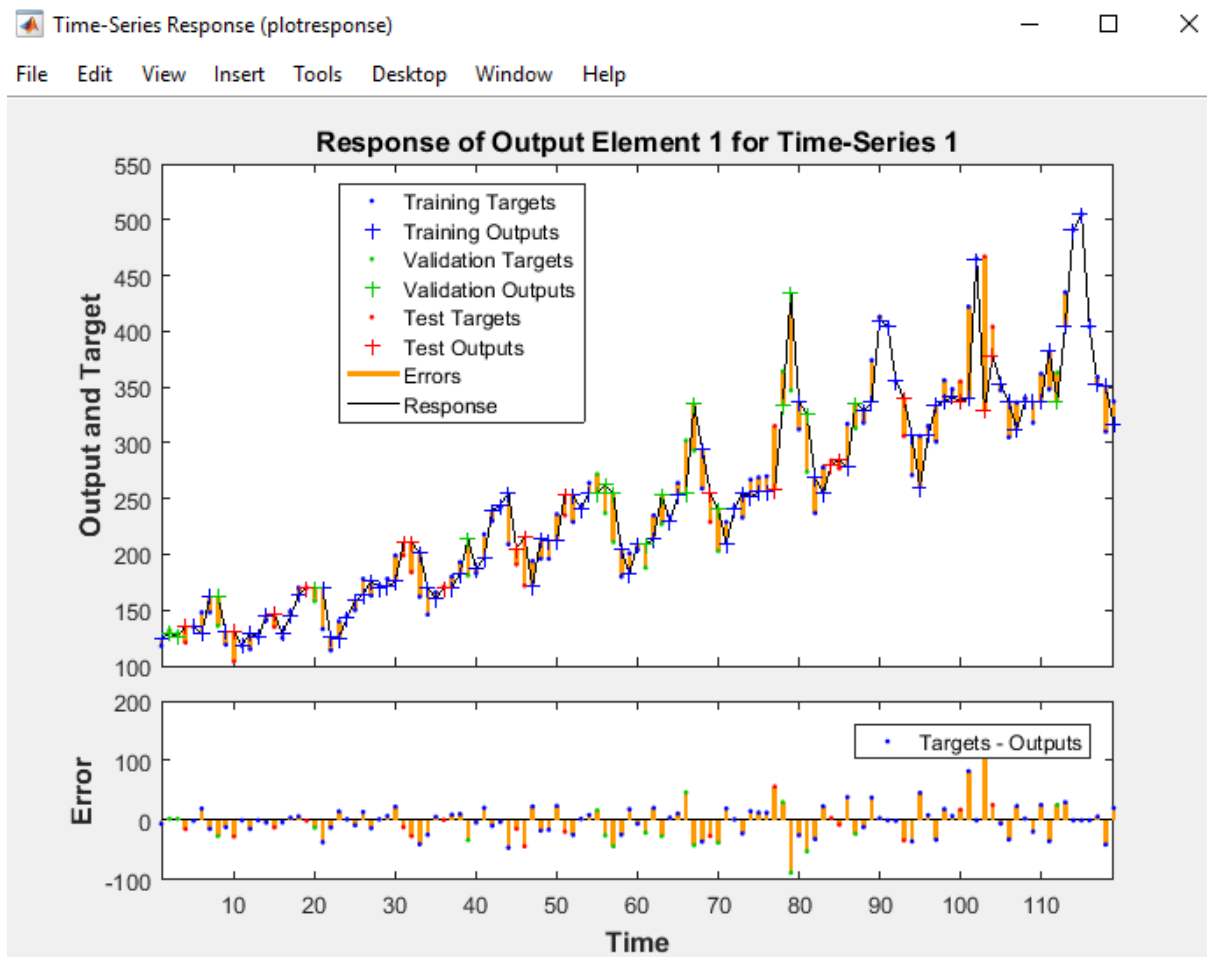


Gráfico gerado dos valores estimados e comparados com os valores reais

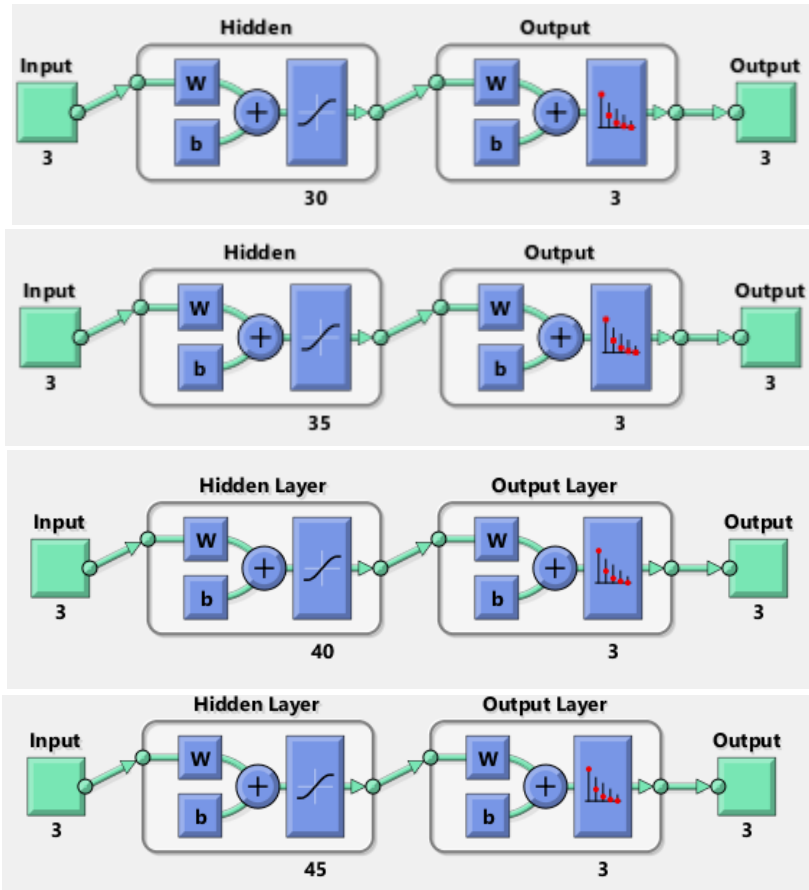
Visto que os erros estão muito próximos de 0, podemos perceber que o preview com 30 neurônios na camada oculta considerando a previsão de acordo com a amostra anterior obteve uma performance muito boa. As curvas são onde ocorre a maior quantidade de erros.

Contudo, pode-se concluir o quanto essa metodologia de estudo é eficiente para a previsão de valores.

### Problema 3

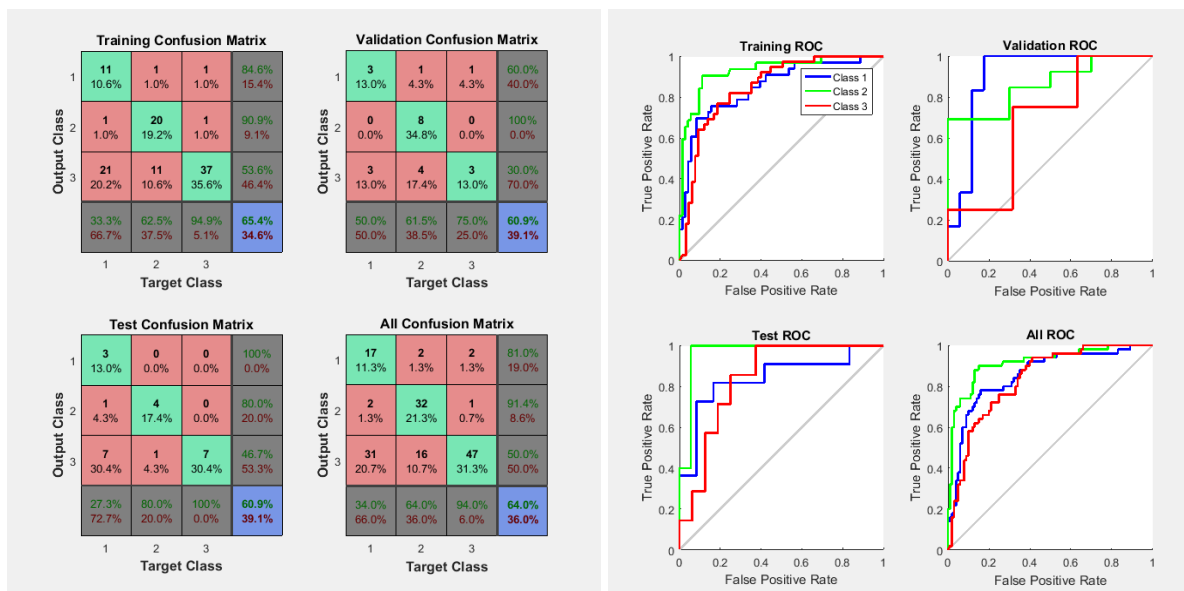
Conceba um classificador neural que receba como entradas as concentrações dos gases  $\text{CH}_4$ (metano),  $\text{C}_2\text{H}_2$ (acetileno) e  $\text{C}_2\text{H}_6$ (etano) e forneça como saída o estado de 'saúde' de um transformador (i.e., Normal, Falha Elétrica ou Falha Térmica?). O classificador deve apresentar taxa de acerto maior ou igual a 70%. Apresente a sua matriz de confusão.

Classificador Neural com 30, 35, 40 e 45 neurônios na “Hidden layer”.

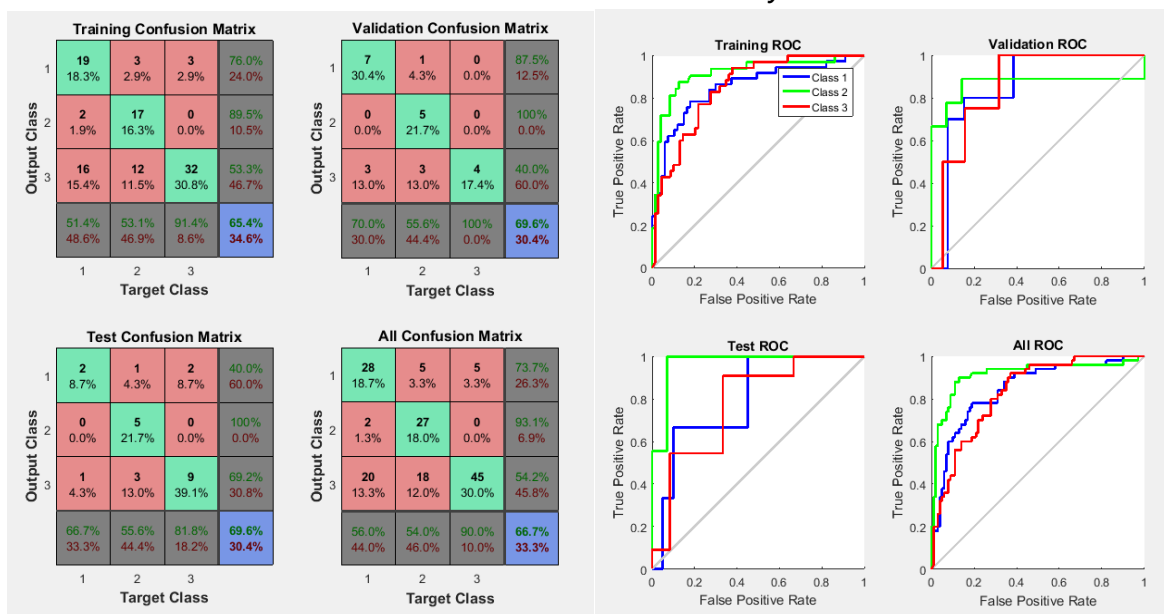


Esse classificador considera apenas três dos cinco parâmetros de análise, ou seja, apenas 3 dos 5 gases analisados.

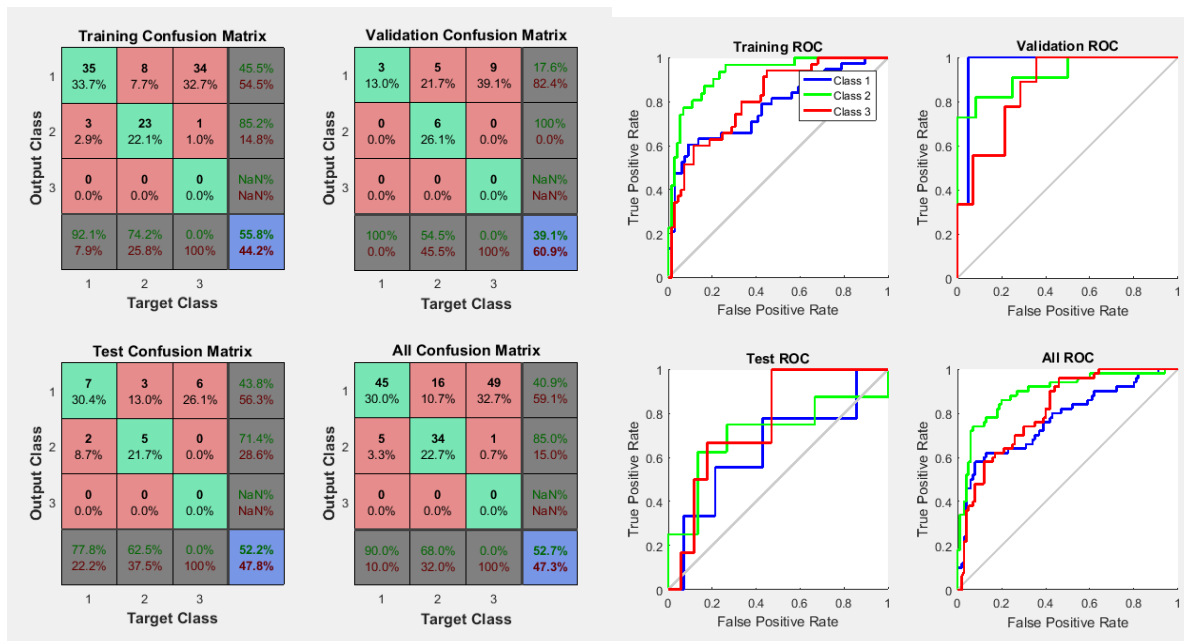
Abaixo estão as matrizes de confusão e ROC referentes aos classificadores com 30, 35, 40 e 45 neurônios na camada oculta.



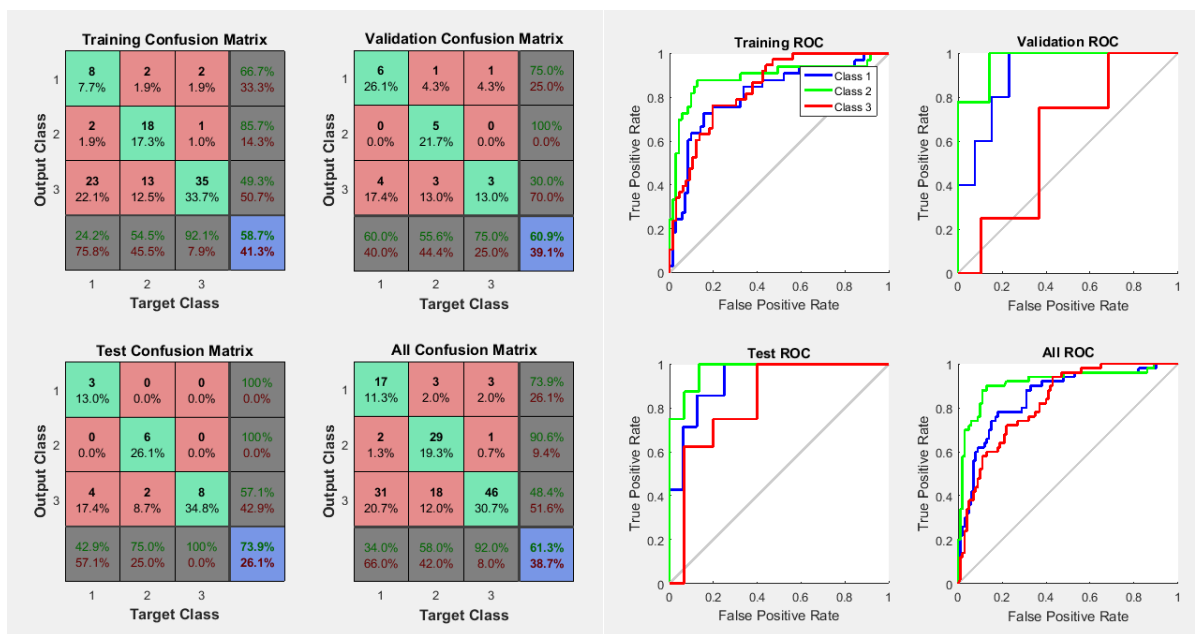
30 Neurônios na “Hidden Layer”



35 Neurônios na “Hidden Layer”



40 Neurônios na "Hidden Layer"



45 Neurônios na "Hidden Layer"

De acordo com os treinamentos feitos, a rede com 35 neurônios foi que obteve melhor resultado (66,7% de acertos), mas não consegui a meta de 70% como foi proposto no problema.

Porem as categorias Normal e Falha Térmica obtiveram uma assertividade de 73.7% e 93.1%, porém a Falha Elétrica não obteve um resultado tão expressivo (54.1%) o que fez com que jogasse o total para baixo.

Contudo, podemos afirmar que o classificador pode sim ser utilizado para classificação dos transformadores quanto à saúde, porém é necessário mexer na rede e fazer alguns ajustes para alcançar a meta.