



Instituto de Ciência e Tecnologia  
Redes Neurais  
Prof. Dr. Marcos Quiles

Análise de comportamento da rede:  
**MLP - Multilayer Perceptron**

Tamires Beatriz da Silva Lucena 111866

São José dos Campos  
Dezembro/2018

# Conteúdo

<b>1</b>	<b>A Rede</b>	<b>3</b>
<b>2</b>	<b>Análise dos Resultados</b>	<b>3</b>
2.1	Íris . . . . .	4
2.1.1	Primeira Análise - Figura 1 . . . . .	4
2.1.2	Segunda Análise - Figura 2 . . . . .	5
2.1.3	Terceira Análise - Figura 3 . . . . .	6
2.1.4	Quarta Análise - Figura 4 . . . . .	7
2.2	Seeds . . . . .	8
2.2.1	Primeira Análise - Figura 5 . . . . .	8
2.2.2	Segunda Análise - Figura 6 . . . . .	9
2.2.3	Terceira Análise - Figura 7 . . . . .	10
2.3	Glass . . . . .	11
2.3.1	Primeira Análise - Figura 8 . . . . .	11
2.3.2	Segunda Análise - Figura 9 . . . . .	12
2.3.3	Terceira Análise - Figura 10 . . . . .	13
<b>3</b>	<b>Conclusões</b>	<b>14</b>

# 1 A Rede

A implementação da rede MLP (*Multilayer Perceptron*) foi realizada com auxílio das bibliotecas *pandas*, *scikit-learn*, *numpy* e *matplotlib*. O algoritmo é constituído por variáveis nas quais seus valores tem influência significativa no desempenho do rede, são elas:

- *n\_camadas*: numero de camadas de processamento – classificadas em três tipos: camada de entrada, camada intermediaria oculta, ou camada de saída.
- *n\_perceptrons*: numero de perceptrons de cada camada deve conter. A quantidade é variável, exceto pela camada de saída que deve conter em sua constituição um numero de perceptrons equivalente e quantidade de classes que se tem no *dataset* analisado.
- *taxa\_erro*: valor do erro aceitável e consequentemente o limitante do algoritmo.
- *eta*: taxa de aprendizado.

O conjunto de dados apresentado a rede é separado em duas partes, 80% responsável pelo treinamento, que refere-se ao processo de aprendizagem da rede, enquanto 20% fica disponível para os testes do funcionamento da rede.

# 2 Análise dos Resultados

A análise será realizada através da alteração dos parâmetros, ou seja, configuração e definição da topologia da rede.

## 2.1 Íris

150 instâncias, 4 atributos e 3 classes

### 2.1.1 Primeira Análise - Figura 1

- $n\_camadas$ : 3
- $n\_perceptrons$ : [5, 5, 3]
- $taxa\_erro$ : 0.03
- $eta$ : 0.5

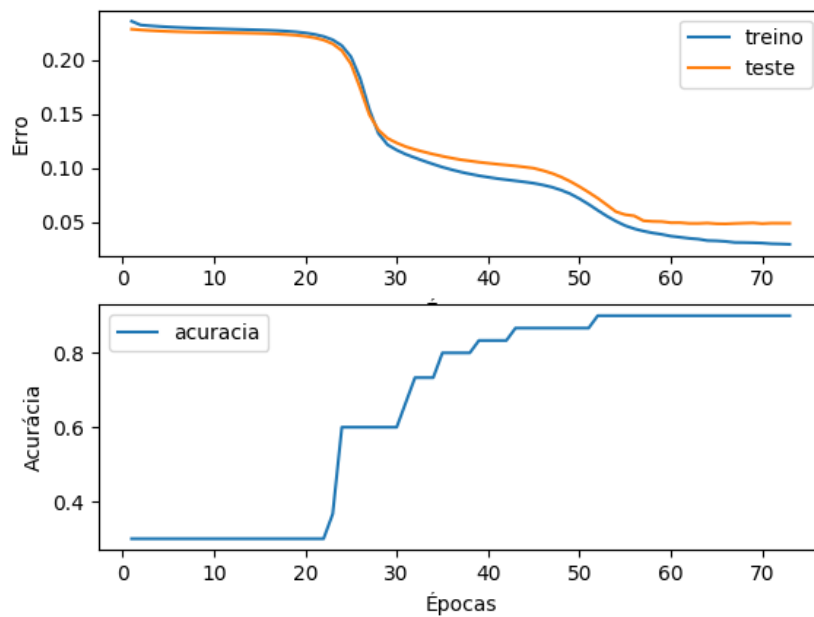


Figura 1: Íris - primeira análise

Dentro das configurações explicitadas contendo 3 camadas, onde a primeira e a oculta contem 5 neurônios cada uma (mesmo numero de atributos do *dataset* íris + bias), com a camada de saída contendo 3 perceptrons (quantidade de classes do conjunto analisado), com erro aceitável bem baixo de 0.03, foram necessárias 72 épocas para que a rede atingisse tal valor. Com uma acurácia de 90% após 50 épocas.

### 2.1.2 Segunda Análise - Figura 2

- $n\_camadas$ : 4
- $n\_perceptrons$ : [5, 5, 5, 3]
- $taxa\_erro$ : 0.03
- $eta$ : 0.5

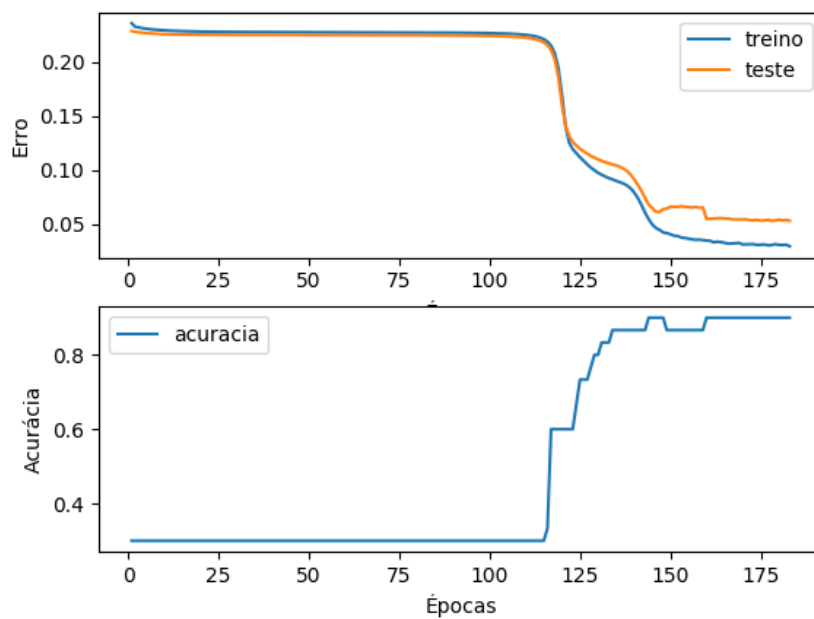


Figura 2: Íris - segunda análise

Em relação a primeira análise, alterou-se apenas a quantidade de camadas a fazer parte da topologia, e com isso a rede já reage de uma maneira bem diferente. Na Figura 2 observa-se que a rede se inicia com uma taxa de erro bem alta - em relação a taxa de erro colocada como aceitável de 0.03 -, que inclusive, mantém-se por mais de 100 épocas, numero muito maior de épocas do que foi necessário para atingir um erro aceitável na configuração anterior. Foram necessárias 182 épocas no total e para este caso, a rede também apresentou uma acurácia máxima de 90%.

### 2.1.3 Terceira Análise - Figura 3

- $n\_camadas$ : 3
- $n\_perceptrons$ : [3, 2, 3]
- $taxa\_erro$ : 0.02
- $eta$ : 0.5

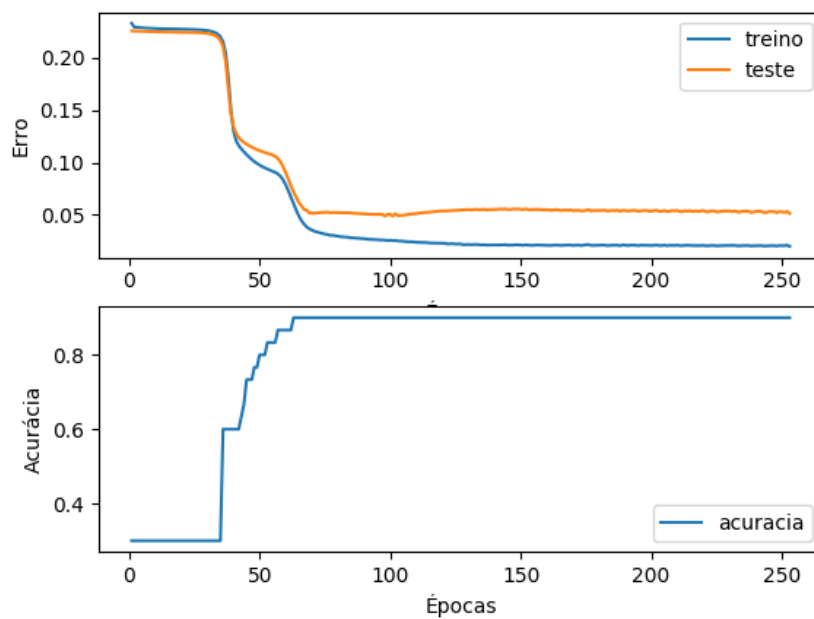


Figura 3: Íris - terceira análise

Na terceira análise voltou-se a configuração para 3 camadas, que já havia apresentado um resultado interessante, e alterou-se a quantidade de neurônios nas camadas, mantendo a primeira com 3 perceptrons, a única oculta com apenas 2 perceptrons e a de saída, como nos demais testes, com 3 perceptrons. Alterou-se também a taxa de erro para 0.02 e observa-se na Figura 3 que a rede precisou de mais épocas para atingir o resultado esperado, mais precisamente de 252 épocas.

#### 2.1.4 Quarta Análise - Figura 4

- $n\_camadas$ : 2
- $n\_perceptrons$ : [5, 3]
- $taxa\_erro$ : 0.014
- $eta$ : 0.5

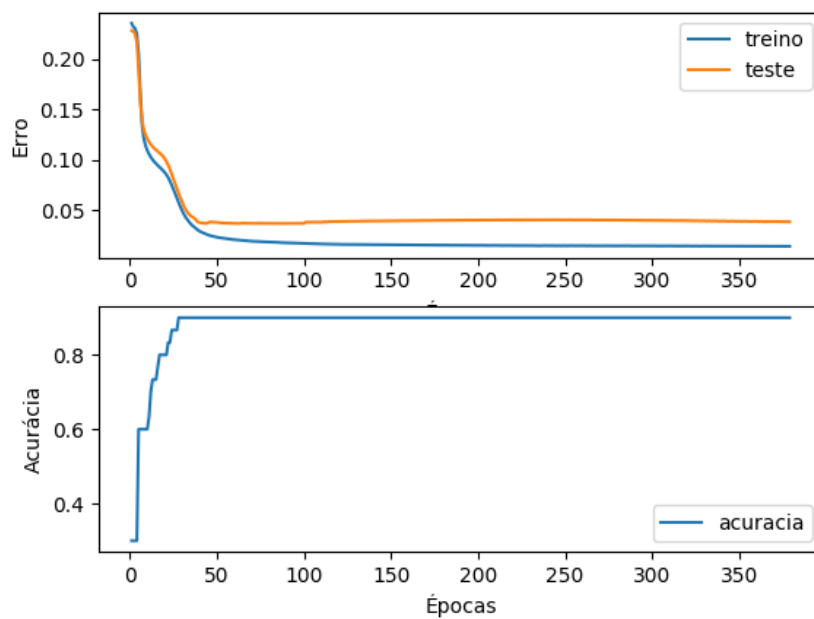


Figura 4: Íris - quarta análise

Com apenas duas camadas - nenhuma oculta - com cinco neurônios na camada de entrada e com uma taxa de erro de 0.014, obteve-se 90% de acurácia aproximadamente à partir da época de numero 25 de um total de 378 épocas.

## 2.2 Seeds

210 instâncias, 7 atributos e 3 classes

### 2.2.1 Primeira Análise - Figura 5

- $n\_camadas$ : 3
- $n\_perceptrons$ : [8, 8, 3]
- $taxa\_erro$ : 0.01
- $eta$ : 0.5

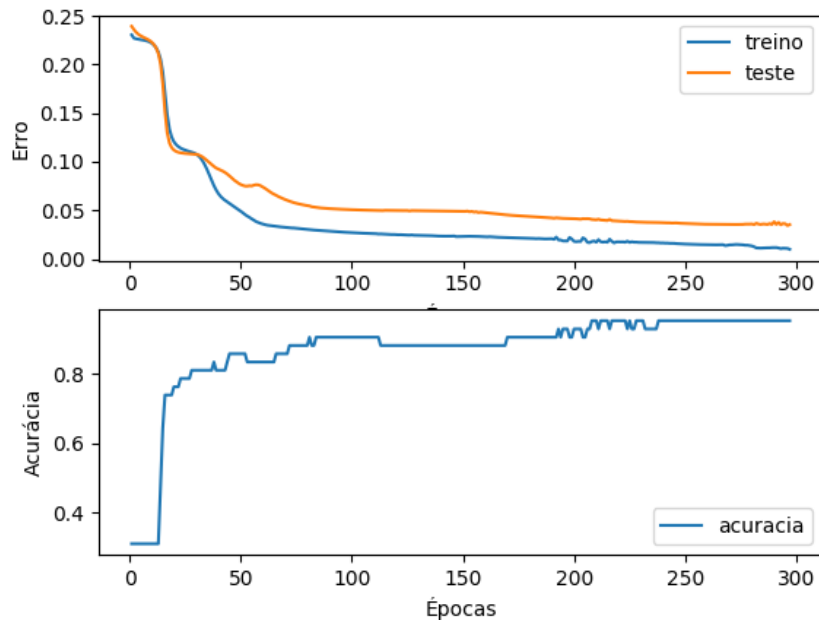


Figura 5: Seeds - primeira análise

A primeira análise de um novo conjunto de dados, o *Seeds* que se refere a classificação de sementes por meio de suas medidas geométricas, apresentou uma acurácia de 95% a partir da época de numero 236 aproximadamente e se manteve assim por mais 60 épocas onde pode alcançar a taxa de erro aceitável de 0.01. Podemos observar que nesta configuração a acurácia teve um crescimento mais gradativo, diferente do obtido nos testes com o conjunto *Íris*.



### 2.2.2 Segunda Análise - Figura 6

- $n\_camadas$ : 4
- $n\_perceptrons$ : [5, 2, 10, 3]
- $taxa\_erro$ : 0.03
- $eta$ : 0.5

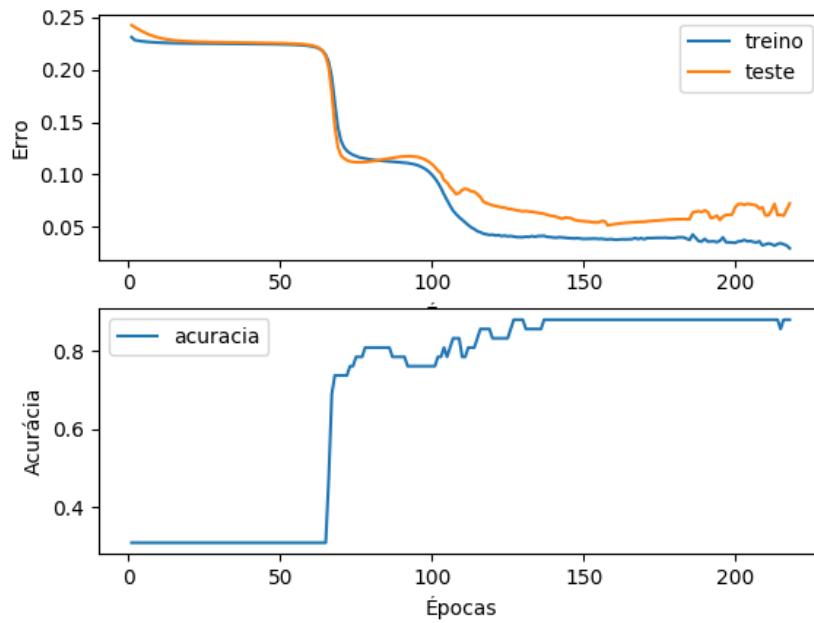


Figura 6: Seeds - segunda análise

Nesta configuração de 4 camadas, quantidades alternadas de perceptrons por camadas e erro aceitável menor que 0.03, durante o decorrer das épocas a acurácia oscilou um tanto até se estabelecer em 88% a partir da época 136. Foram necessárias 217 épocas para atingir ao erro estabelecido.

### 2.2.3 Terceira Análise - Figura 7

- $n\_camadas$ : 2
- $n\_perceptrons$ : [8, 3]
- $taxa\_erro$ : 0.01
- $eta$ : 0.2

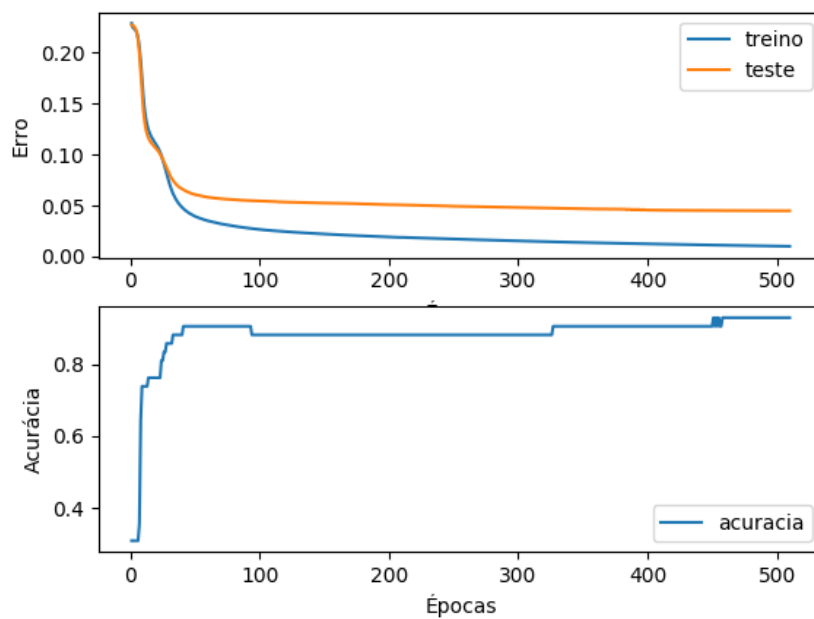


Figura 7: Seeds - terceira análise

Com duas camadas apenas e com uma taxa de aprendizado de 0.02 a rede precisou de muito mais épocas do que nos testes anteriores - mais que o dobro do anterior - para alcançar o objetivo. Fazendo uma comparação com a Primeira análise realizada com o este mesmo conjunto de dados, temos que: com um  $eta$  maior e com 1 camada a mais, além da quantidade de épocas ser menor a acurácia também alcança um valor mais alto, indo contra uma acurácia de 92% neste terceiro teste.

## 2.3 Glass

214 instâncias, 10 atributos e 6 classes

*OBS:* Uma das colunas de atributos foi retirada do *dataset* pois se trata de índices.

### 2.3.1 Primeira Análise - Figura 8

- *n\_camadas*: 2
- *n\_perceptrons*: [10, 6]
- *taxa\_erro*: 0.1
- *eta*: 0.5

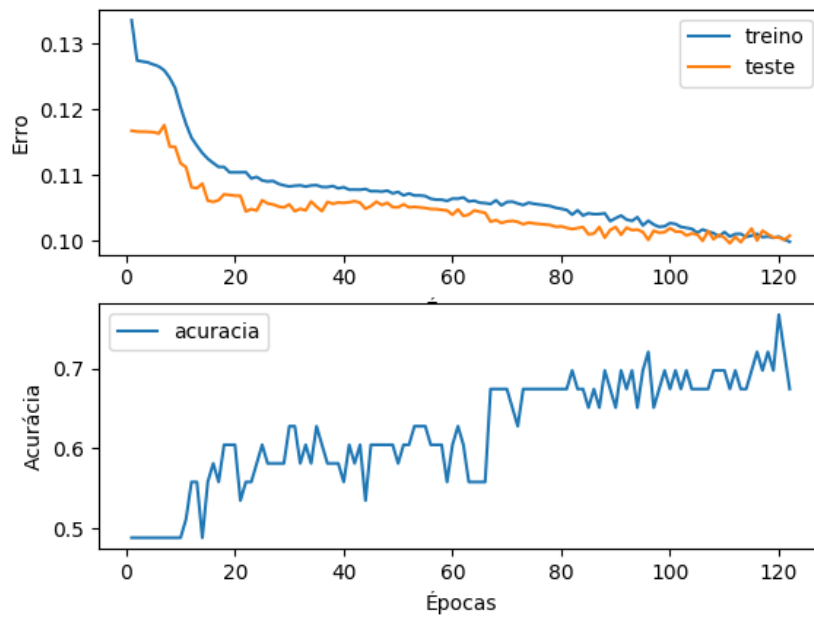


Figura 8: Glass - primeira análise

A topologia da rede configurada para o *Glass* apresentou grande oscilação nos valores de acurácia, tendo o maior valor na época 119 equivalente a 76%, e imediatamente na época 121 houve uma queda da acurácia pra 67% onde curiosamente atingiu-se o valor de erro estipulado em 0.1.

### 2.3.2 Segunda Análise - Figura 9

- $n\_camadas$ : 3
- $n\_perceptrons$ : [10, 10, 6]
- $taxa\_erro$ : 0.1
- $eta$ : 0.5

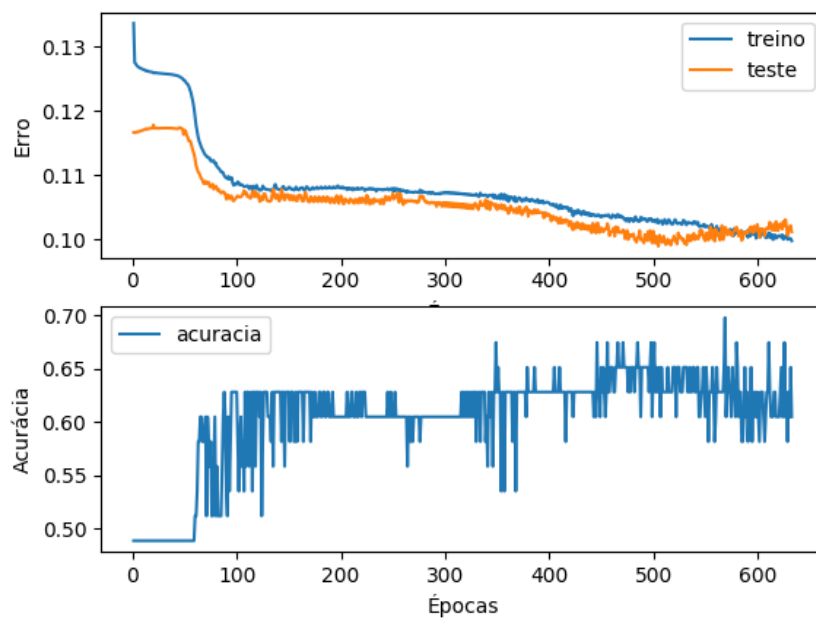


Figura 9: Glass - segunda análise

Para a segunda obteve-se uma diferença muito significativa na atuação da rede em relação ao primeiro teste, onde a quantidade de épocas necessárias foi aproximadamente 4 vezes maior que a quantidade da primeira análise. Vale observar que a taxa de erro a ser alcançada foi mantida, assim como os demais parâmetros. Apesar da diferença em épocas, o valor da acurácia final de 60% está próximo a acurácia anterior e o comportamento do gráfico de erros seguiu bastante semelhante.

### 2.3.3 Terceira Análise - Figura 10

- $n\_camadas$ : 3
- $n\_perceptrons$ : [2, 3, 6]
- $taxa\_erro$ : 0.1
- $eta$ : 0.5

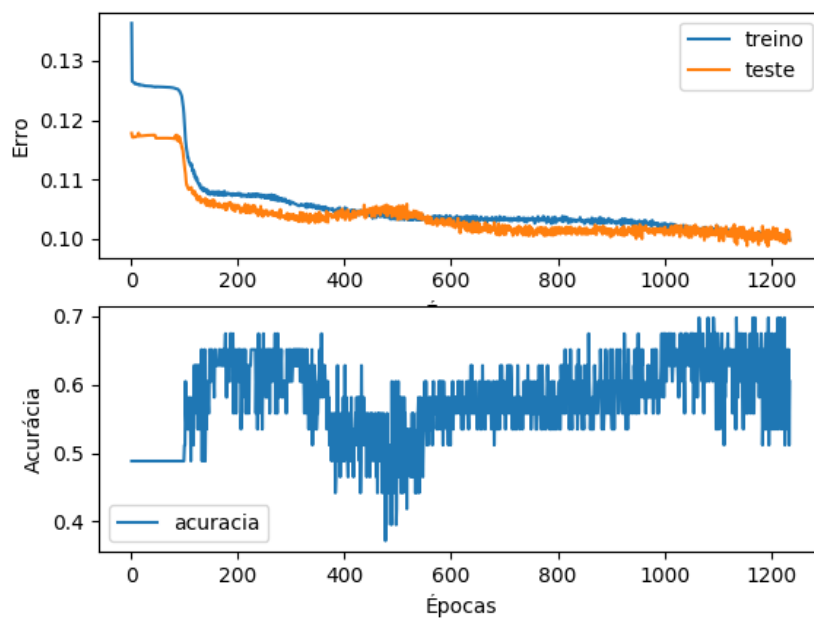


Figura 10: Glass - terceira análise

Desta vez foi feita a alteração - em relação a anterior - apenas do número de perceptrons por camadas e novamente atingiu-se uma acurácia final de 60%, tendo havido em outro momento uma acurácia maior de 70% e o comportamento da rede em relação as redes ainda se manteve similar aos anteriores do mesmo conjunto de dados. Novamente a grande diferença foi o número de épocas que para este teste atingiu 1234.

### 3 Conclusões

Ao longo de todos os testes demonstrados - além de outros realizados e não dispostos neste documento - foi possível concluir que a maneira como a rede é configurada, ou seja, a definição da topologia da rede para cada conjunto de dados, muito importa para o desempenho do algoritmo. Observaram-se exemplos onde o tempo de aprendizado da rede demorou muito mais apenas pela alteração do valor de uma única variável, por exemplo. Outro ponto importante é em relação ao *overfitting*, acontecimento que pode ser visto em todos os testes que se tratam dos conjuntos *Íris* e *Seeds*, que se refere ao ajuste muito bom para os conjuntos de treino, porém ineficaz para os conjuntos de teste, o que significa que a rede não aprendeu, mas decorou.

O conjunto *Glass* em especial, apresentou nos três testes evidenciados um certo padrão no que se diz respeito ao ajuste dos erros, e para todos os testes não houve indicação de *overfitting* de maneira muito clara.