

# Machine-Learning

*Samuel Cavalcanti*

<https://github.com/samuel-cavalcanti>

## Questão 1

Considere o problema de classificação de padrões constituído de duas classes com os seguintes conjuntos de treinamentos:

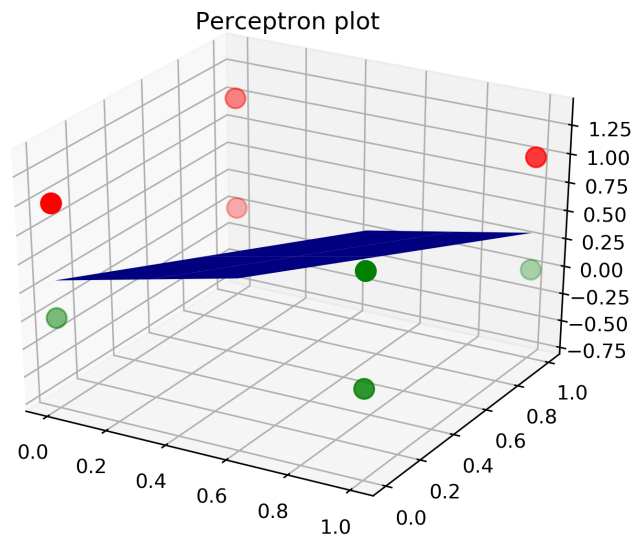
$C_1 = \{(0, 0, 0), (1, 0, 0), (1, 0, 1), (1, 1, 0)\}$

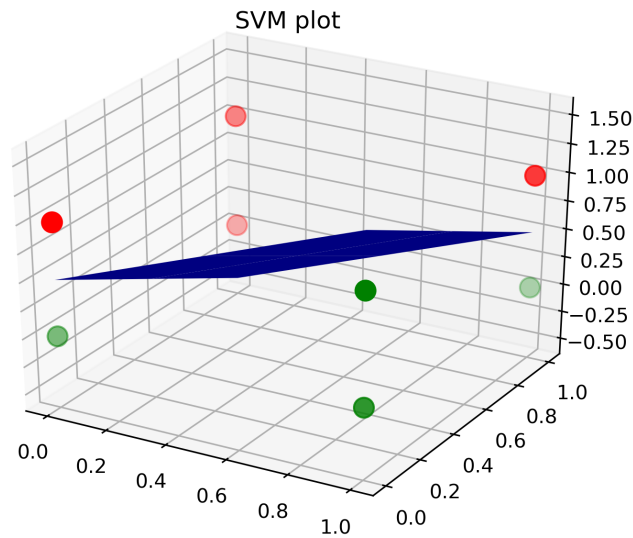
$C_2 = \{(0, 0, 1), (0, 1, 1), (0, 1, 0), (1, 1, 1)\}$

Determine o hiperplano de separação dos padrões considerando os seguintes métodos:

a) Algoritmo do perceptron

b) Máquina de Vetor de Suporte Linear.



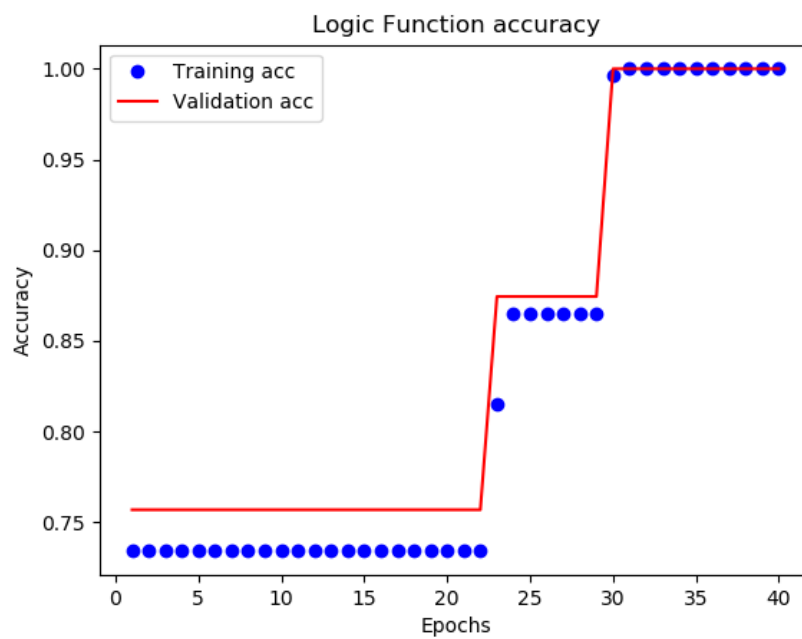
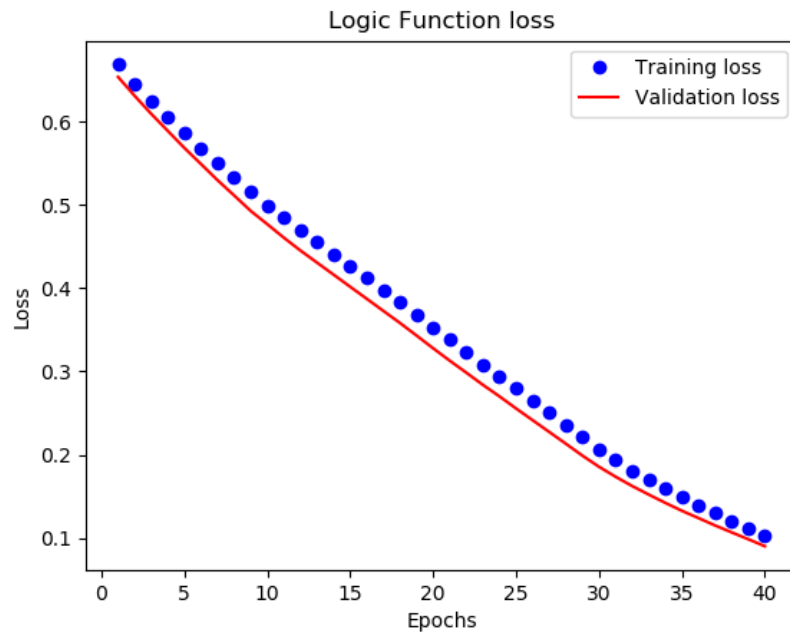


## Questão 2

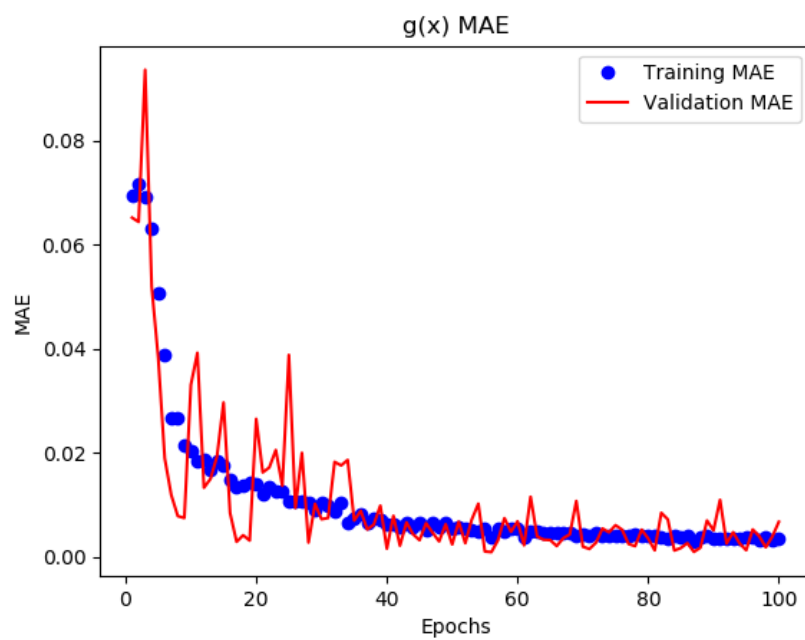
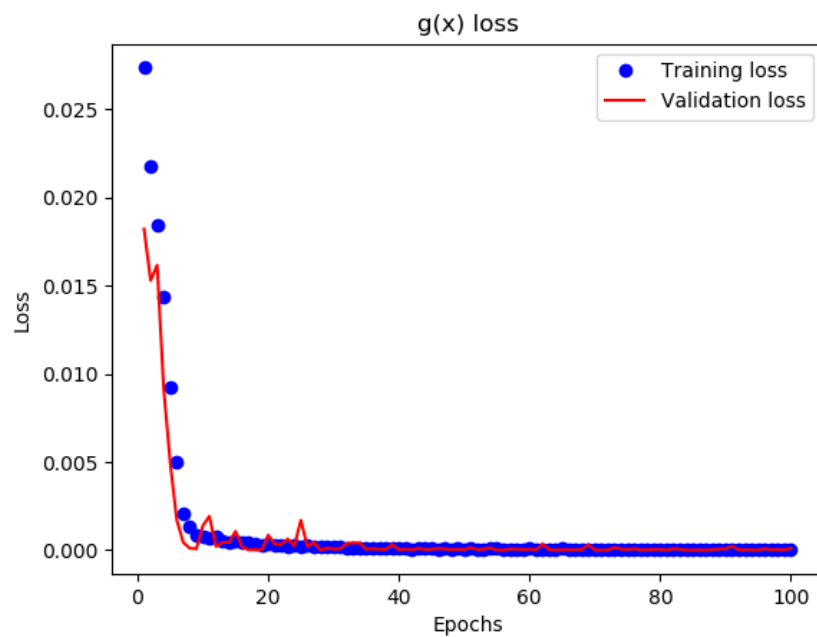
Utilize redes neurais perceptrons de múltiplas camadas para aproximar as funções abaixo. Para o caso dos itens **b** e **c** e apresente para cada caso a curva da função analítica e a curva da função aproximada pela rede neural. Apresente também a curva do erro médio de treinamento com relação ao número de épocas e a curva do erro médio com o conjunto de validação. Procure definir para cada função a arquitetura da rede neural perceptron, isto é, o número de entradas, o número de neurônios em cada camada e o número de neurônios camada de saída.

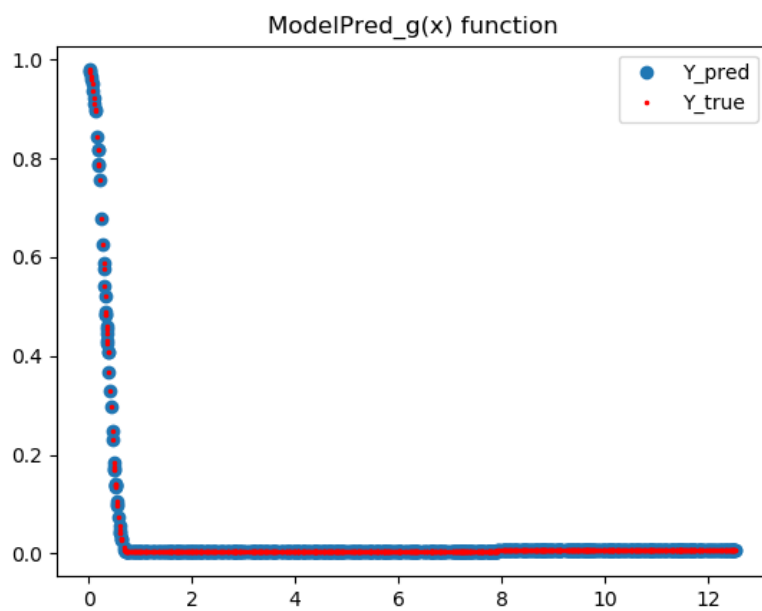
- função lógica and
- $f(x) = \cos(2\pi x)/(1-(4x)^2) * \sin(\pi x)/(\pi x)$ ,  $0 < x \leq 4\pi$
- $f(x,y) = x^2 + y^2 + 2xy\cos(\pi xy) + x + y - 1$

## 2.a Gráficos função logica

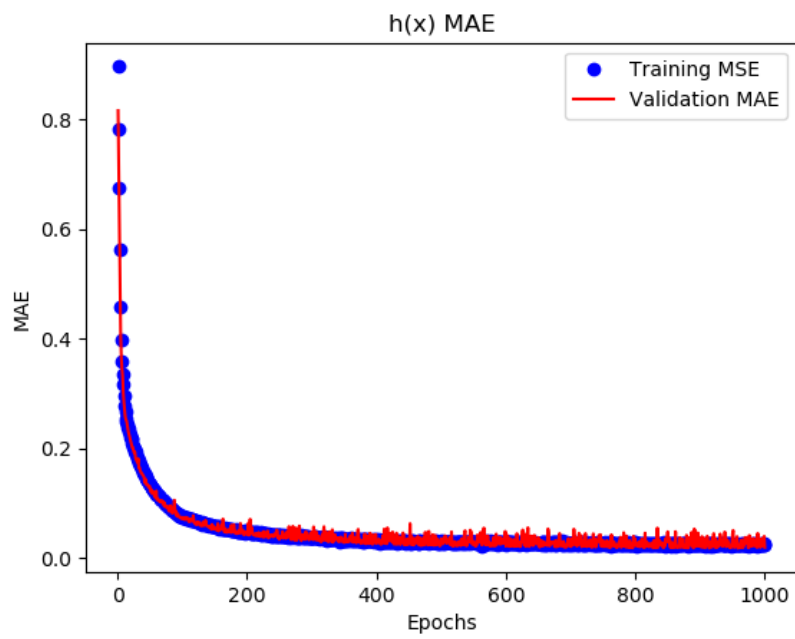
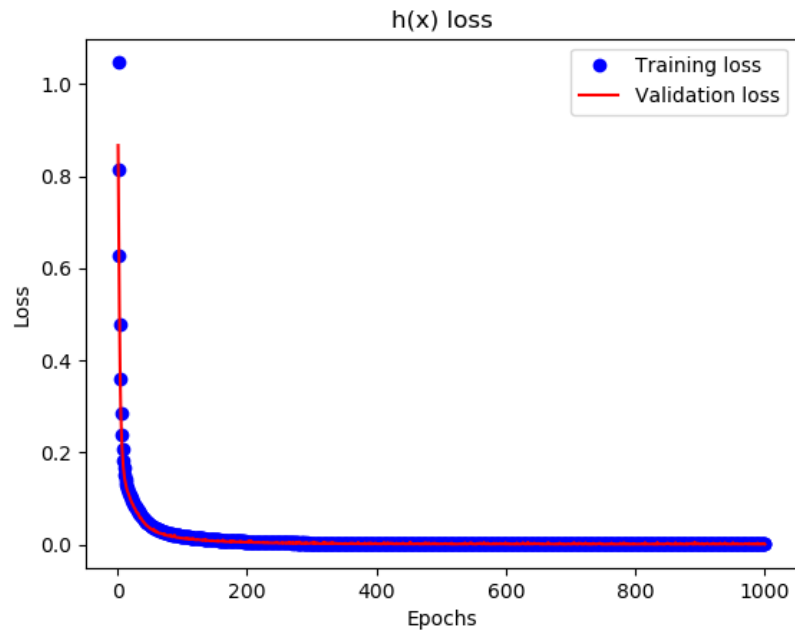


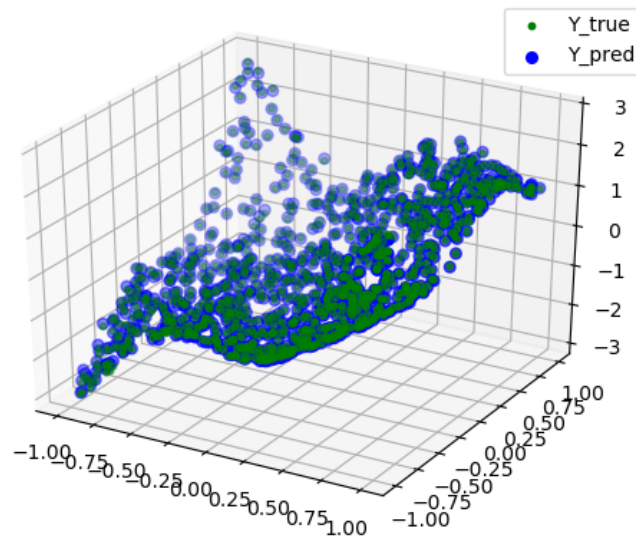
**2.b**  $f(x) = \cos(2\pi x)/(1-(4x)^2)*\sin(\pi x)/(\pi x)$





$$2.c \ f(x,y) = x^2 + y^2 + 2xy\cos(\pi xy) + x + y - 1$$





### Questão 3

Considere o problema de classificação de padrões bidimensionais constituído neste caso de 5 padrões. A distribuição dos padrões tem como base um quadrado centrado na origem interceptando os eixos nos pontos +1 e -1 de cada eixo. Os pontos +1 e -1 de cada eixo são centros de quatro semicírculos que se interceptam no interior do quadrado originando uma classe e a outra classe corresponde as regiões de não interseção. Após gerar aleatoriamente os dados que venham formar estas distribuições de dados, selecione um conjunto de treinamento e um conjunto de validação. Solucione este problema considerando:

a-) Um rede perceptron de múltiplas camada

b-) Uma máquina de vetor de suporte (SVM)

Apresente o desempenho dos classificadores usando o conjunto de validação e calculando para cada um a matriz de confusão.

**Matrix de confusão da SVM**

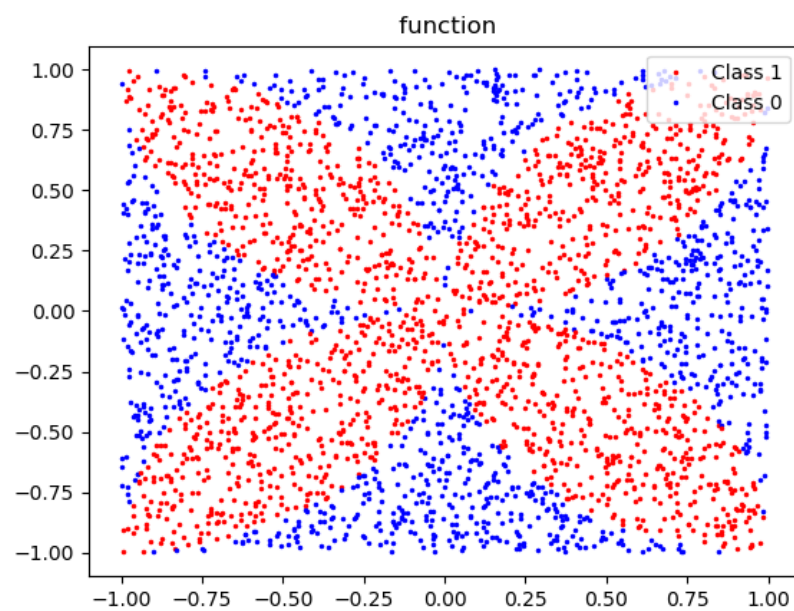


Figure 1: petalas



	nan	0	1
0		911	311
1		36	1742

*OBS:* A SVM conseguiu 88% de acurácia

#### **Matrix de confusão da MLP**

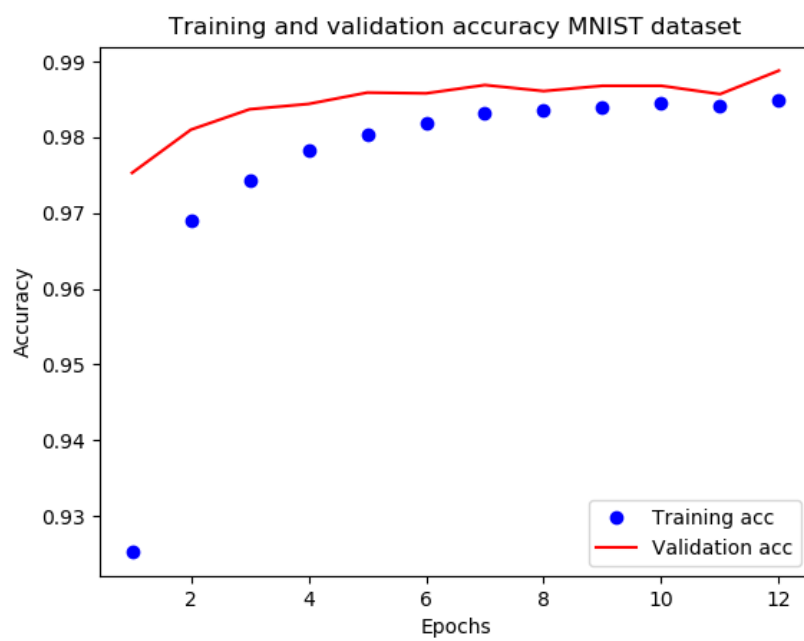
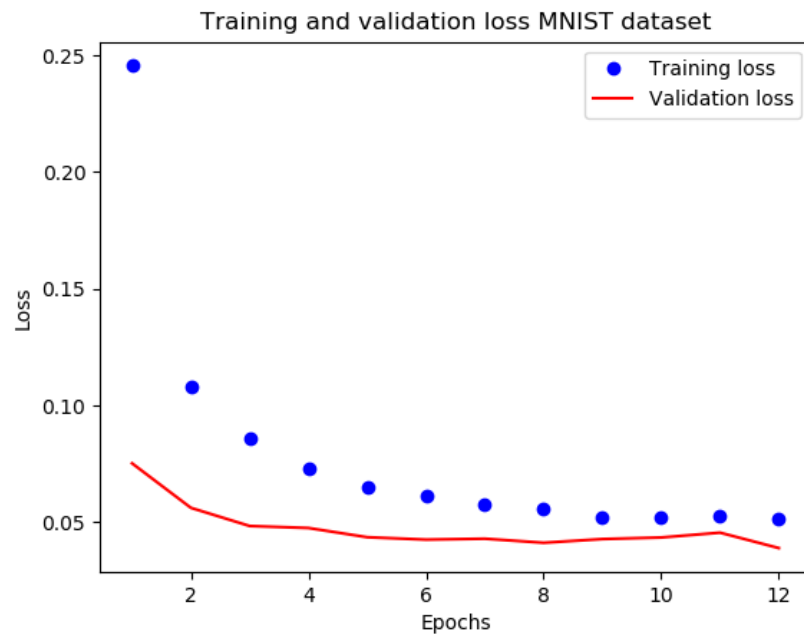
	nan	0	1
0		1158	64
1		37	1741

*OBS:* A Deep MLP conseguiu 97% de acurácia

### **Questão 4**

Considere o problema de reconhecimento de padrões constituído neste caso de uma deep learning, no caso uma rede convolutiva capaz de reconhecer os números: 0, 1, 2, 3 ..., 9 , mesmo que estes tenham um pequeno giro de até 10 graus. Avalie o desempenho de sistema gerando a matriz de confusão. Pesquise as base de dados para serem usadas no treinamento

### gráficos de convergência da rede convolutiva



**Matrix de confusão sem rotação:**

nan	0	1	2	3	4	5	6	7	8	9
0	978	0	1	0	0	0	0	1	0	0
1	0	1132	2	1	0	0	0	0	0	0
2	2	2	1018	0	1	0	1	4	2	2
3	0	0	0	1002	0	2	0	3	3	0
4	0	0	0	0	978	0	1	0	0	3
5	2	0	0	3	0	882	3	0	1	1
6	7	2	0	1	1	2	943	0	2	0
7	0	2	8	0	0	0	0	1014	0	4
8	5	0	2	1	2	0	1	2	956	5
9	3	3	0	3	9	2	0	3	1	985

A rede convolutiva conseguiu 99% de acurácia

**Matrix de confusão com rotação:**

nan	0	1	2	3	4	5	6	7	8	9
0	973	0	1	0	0	0	2	3	0	1
1	0	1130	1	2	0	2	0	0	0	0
2	1	12	995	4	1	0	1	15	3	0
3	0	0	0	1000	0	4	0	4	1	1
4	0	9	0	0	968	0	3	0	0	2
5	2	1	0	9	0	874	3	0	1	2
6	6	8	0	1	2	2	938	0	1	0
7	0	13	3	0	0	0	0	1002	0	10
8	5	3	3	4	5	3	3	2	937	9
9	2	6	0	4	8	2	1	2	0	984

A rede convolutiva conseguiu 98% de acurácia

**Questão 5**

Um problema interessante para testar a capacidade de uma rede neural atuar como classificador de padrões

é o problema das duas espirais intercaladas. A espiral 1 sendo uma classe e a espiral 2 sendo outra classe.

Gere os exemplos de treinamento usando as seguintes equações:

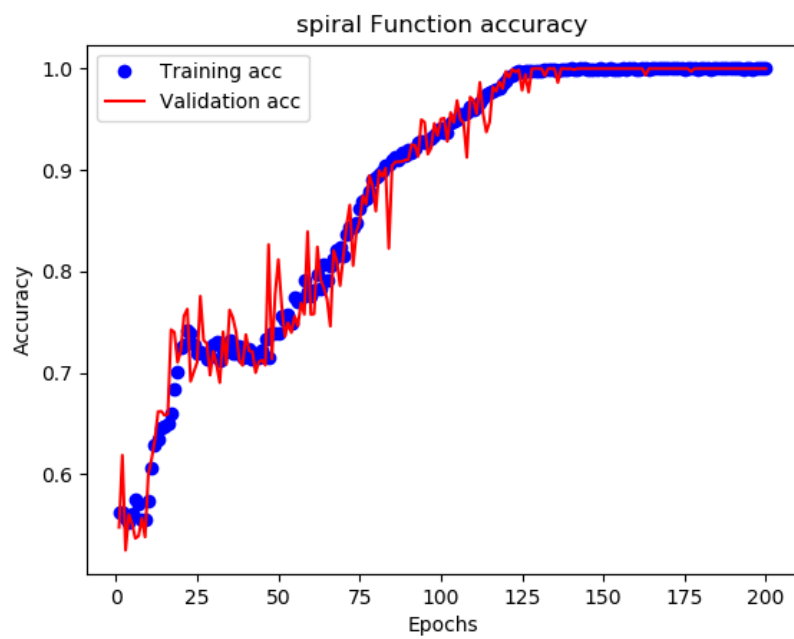
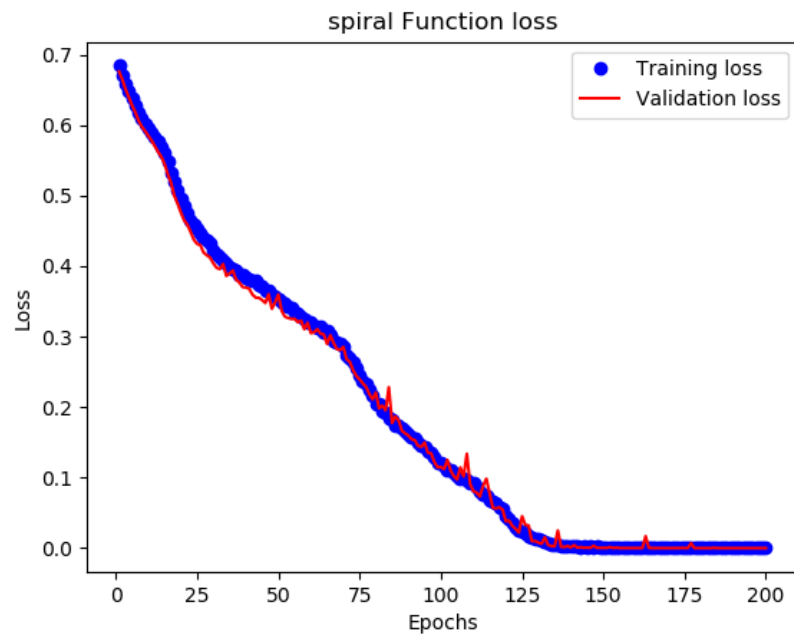
para espiral 1:

$$x = \theta/4 \cos(\theta), y = \theta/4 \sin(\theta)$$

para espiral 2:

$x = (\theta+0.8)/4 \cos(\theta)$ ,  $y = (\theta+0.8) \sin(\theta)$   
fazendo  $\theta$  assumir 100 igualmente espaçados valores entre 0 e 20 radianos.  
Solucione este problema considerando:  
a-) Um rede perceptron de múltiplas camadas deep learning  
b-) Uma máquina de vetor de suporte (SVM)

### Gráfico de convergência da rede deep learning



### Função de decisão formada pela SVM

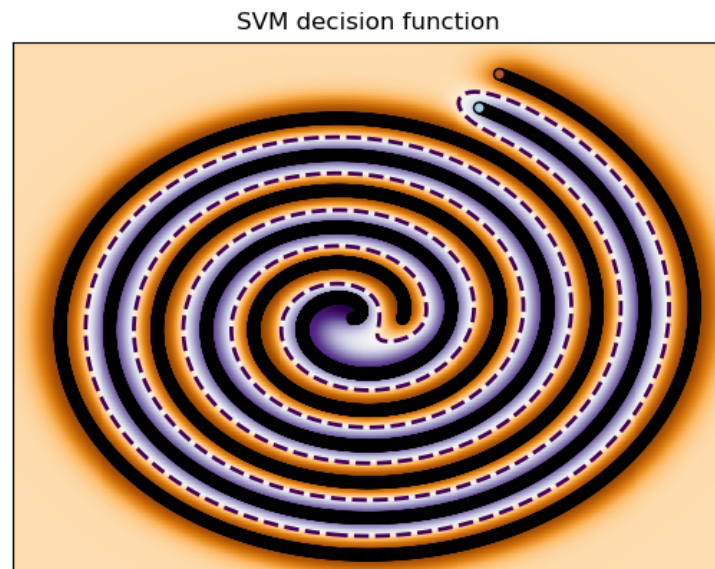


Figure 2: SVM

### Matrix de confusão

nan	0	1
0	5000	0
1	0	5000

*OBS:* a deep learning conseguiu 100% de acurácia

*OBS:* a no-linear SVN conseguiu 100% de acurácia

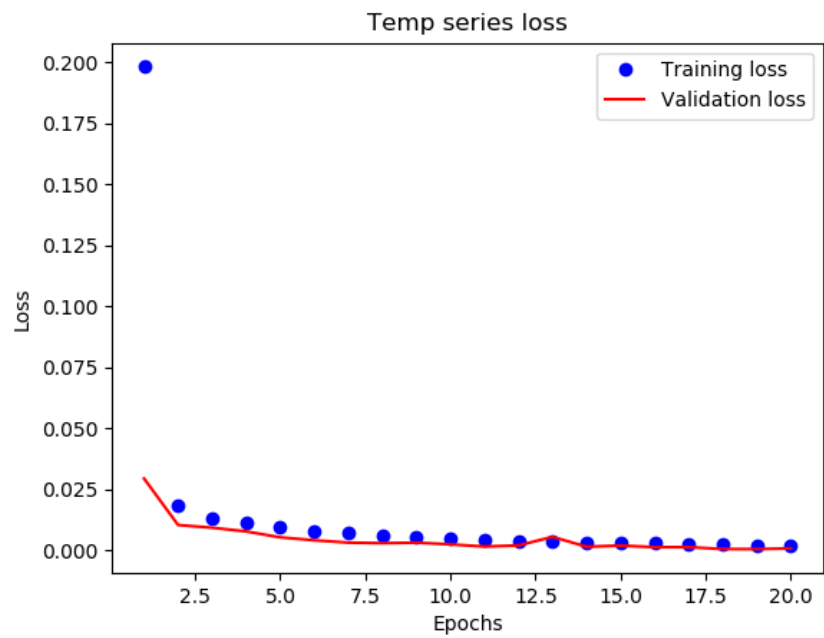
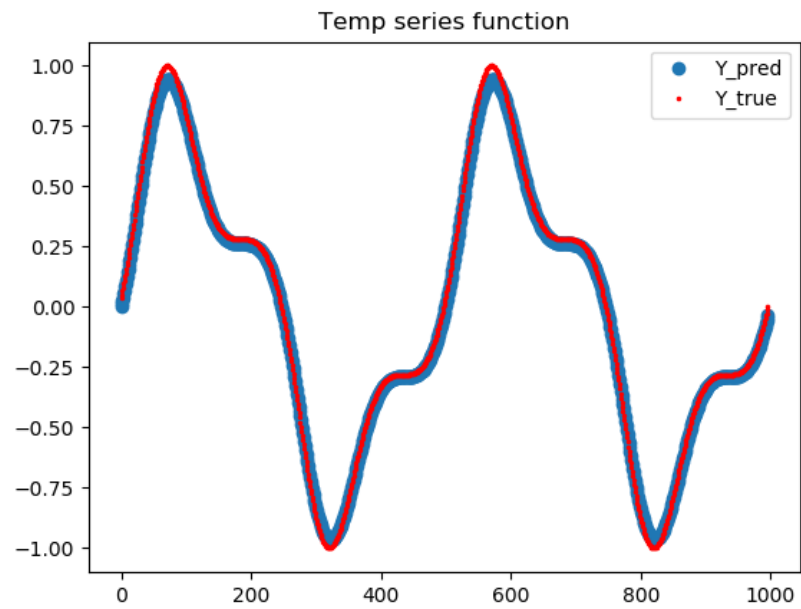
### Questão 6

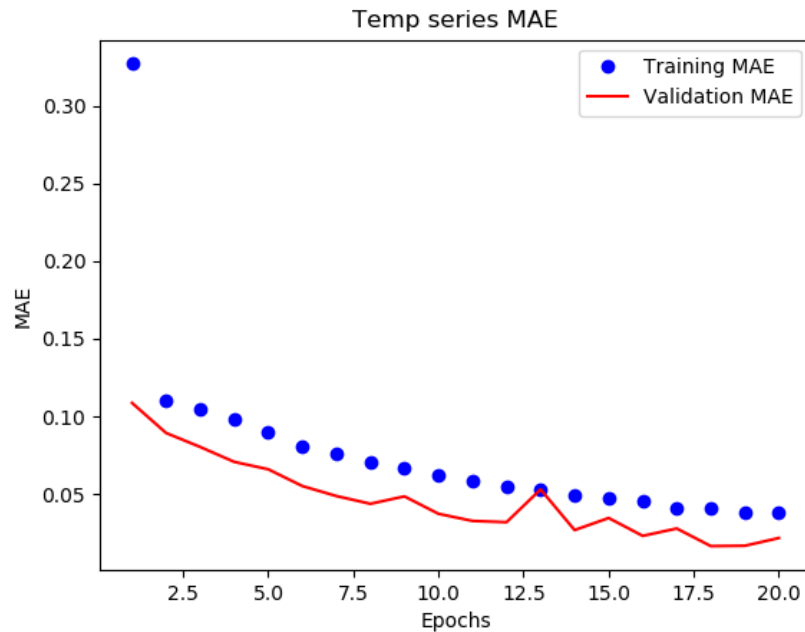
Utilize uma a NARX no caso uma rede neural perceptron de múltiplas camadas

com realimentação para fazer a predição de um passo da série temporal:

$$x(n) = \sin(n + \sin^2(n)).$$

Avalie o desempenho mostrando o erro de predição





### Questão 7

Considere dois sensores espacialmente distribuídos. Um sensor capta o sinal proveniente de uma fonte de sinal e o outro sensor é dirigido para captar o sinal o ruído proveniente de uma fonte de ruído indesejável. Os dois sensores captam um pouco de cada sinal.

O objetivo é cancelar o ruído que é captado pelo sensor dirigido para fonte de sinal.

Para modelar o problema considere as seguintes variáveis:

$s(n)$ : sinal discreto emitido pela fonte de sinal dado por:

$$s(n) = \sin(0.075\pi n)$$

$x(n)$ : o sinal captado pelo sensor dirigido para captar o sinal da fonte dado por:

$$x(n) = s(n) + v1(n)$$

$y(n)$ : o sinal captado pelo sensor dirigido para captar o sinal de ruído:

$$y(n) = v2(n) + 0.05s(n)$$

$v1(n)$ : ruído captado pelo sensor 1 dado por:

$$v1(n) = -0.5v1(n-1) + v(n)$$

$v2(n)$ : ruído captado pelo sensor 2 dado por:

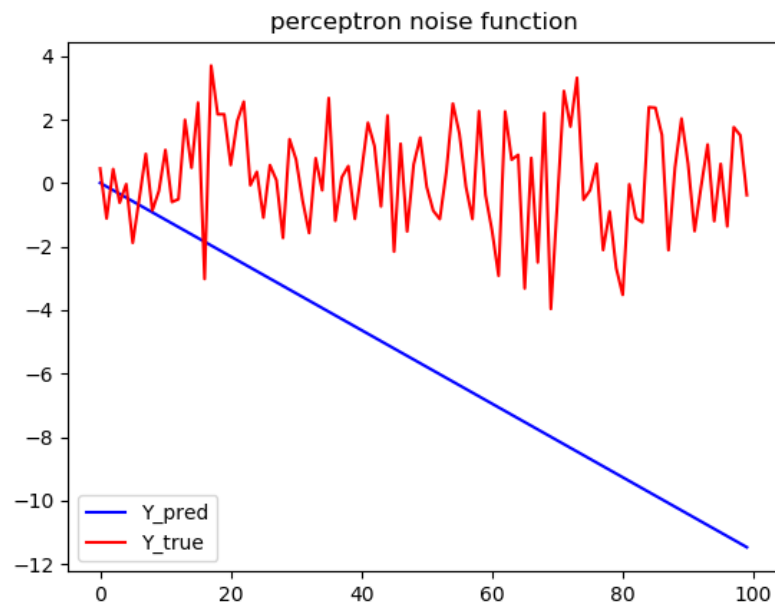
$$v2(n) = 0.8v2(n-1) + v(n)$$

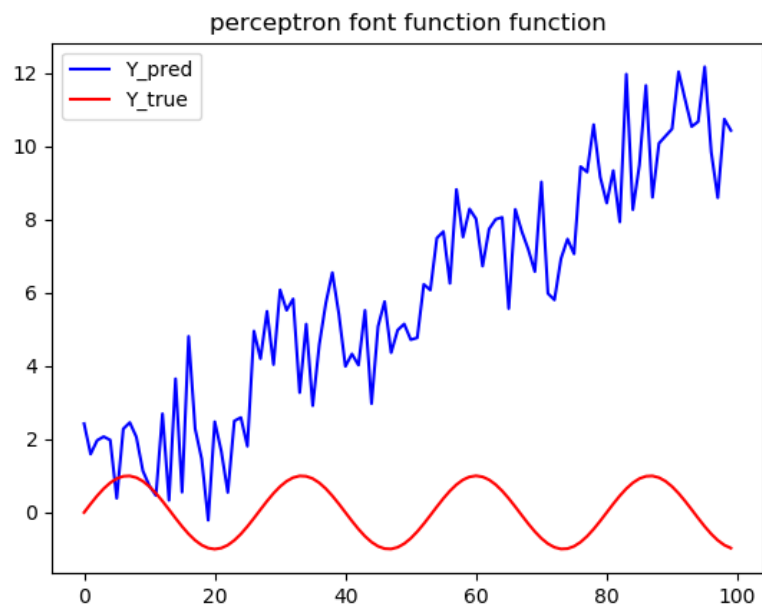
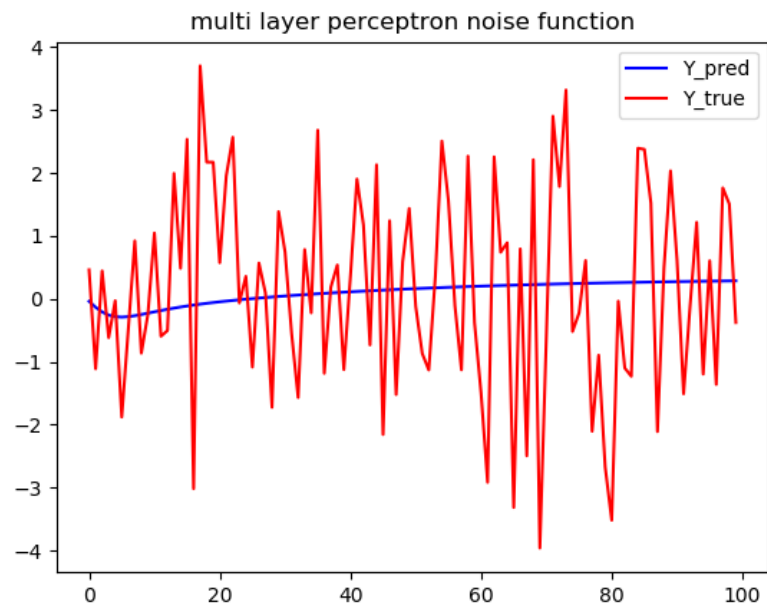
$v(n)$ : um ruído branco uniformemente distribuído com média nula e variân-

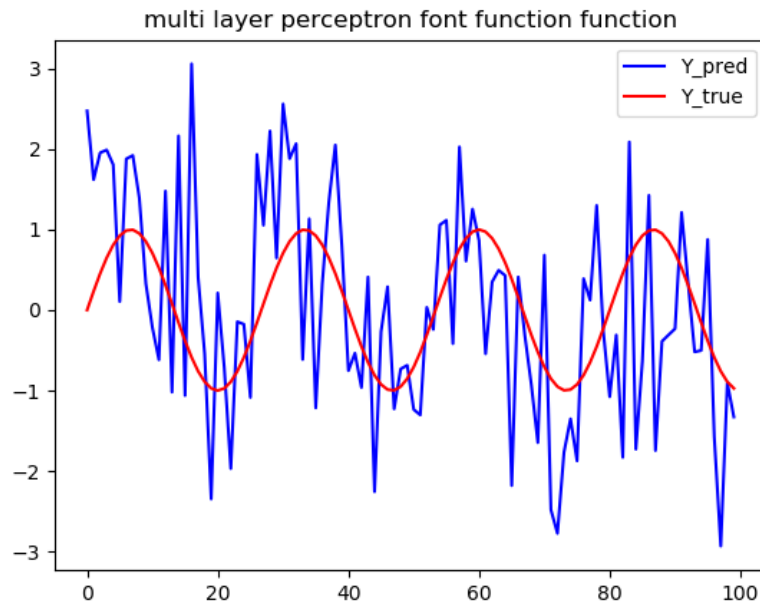


cia unitária.

Para remoção do ruído utilize um cancelador de ruído, isto é um sistema capaz de gerar o ruído recebido pelo sensor que capta o sinal de interesse. Para isto utilize o perceptron puramente linear treinado com o algoritmo LMS e em seguida uma rede perceptrons de múltiplas camadas treinada com o algoritmo da backpropagation. Nos dois casos considere como entrada os valores nos instantes  $n$ ,  $n-1$ ,  $n-2$ ,  $n-3$ ,  $n-4$ ,  $n-5$ . Avalie o desempenho dos dois canceladores.







### Questão 8

Considere o problema de reconhecimento de padrões constituído neste caso das vogais do alfabeto, utilizando para isto uma rede neural deep learning formada por um stacked de autoencoders. Represente as vogais através de matrizes de pixel binária. Teste a robustez do sistema para situações onde as vogais estão ruidosas e com pequenas rotações. Avalie o desempenho de sistema gerando a matriz de confusão

**Exemplo do dataset depois do binariação das imagens**

**Classificador sem o Autoencoder:**

**Matrix de Confusão sem ruído:**

nan	A	E	I	O	U
A	312	1	5	1	1
E	10	262	5	2	2

nan	A	E	I	O	U
I	6	4	159	6	5
O	3	3	3	213	3
U	4	2	5	8	43

*OBS:* Acurácia do modelo ficou em torno de 93%

**Matrix de confusão com ruído:**

nan	A	E	I	O	U
A	311	8	0	1	0
E	21	259	0	1	0
I	84	57	19	20	0
O	34	25	1	164	1
U	16	22	1	19	4

*OBS:* Acurácia do modelo ficou em torno de 70%

**Classificador com Autoencoder**

**Matrix de Confusão sem ruído:**

nan	A	E	I	O	U
A	303	2	6	6	3
E	8	248	8	14	3
I	11	2	154	10	3
O	4	7	3	205	6
U	5	3	4	19	31

*OBS:* Acurácia do modelo ficou em torno de 89%

**Matrix de confusão com ruído:**

nan	A	E	I	O	U
A	291	4	13	10	2
E	11	245	9	15	1
I	17	4	150	6	3
O	8	7	5	195	10
U	7	3	2	23	27

Dataset Utilizado : Vogais



Figure 3: exemploDataset

Preprocessing



Figure 4: binaryExemple

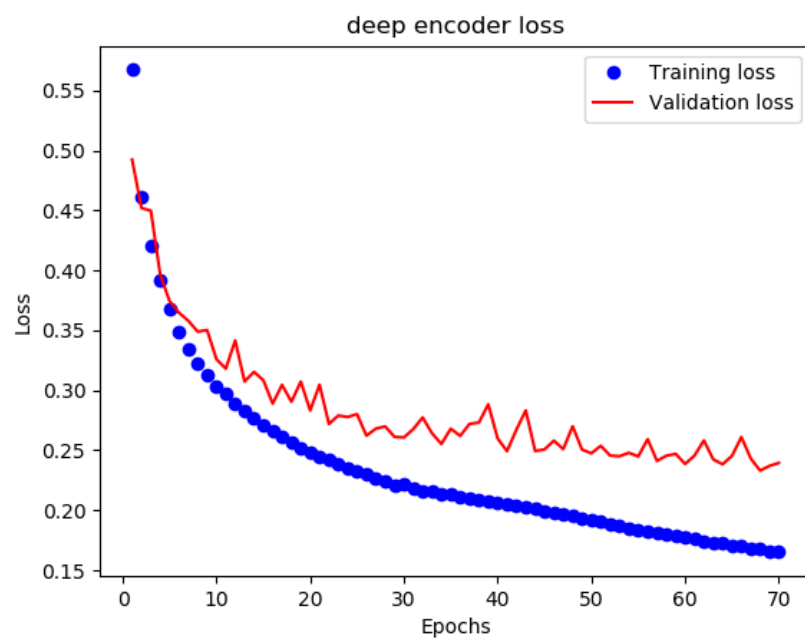


Figure 5: deepEncoderLoos

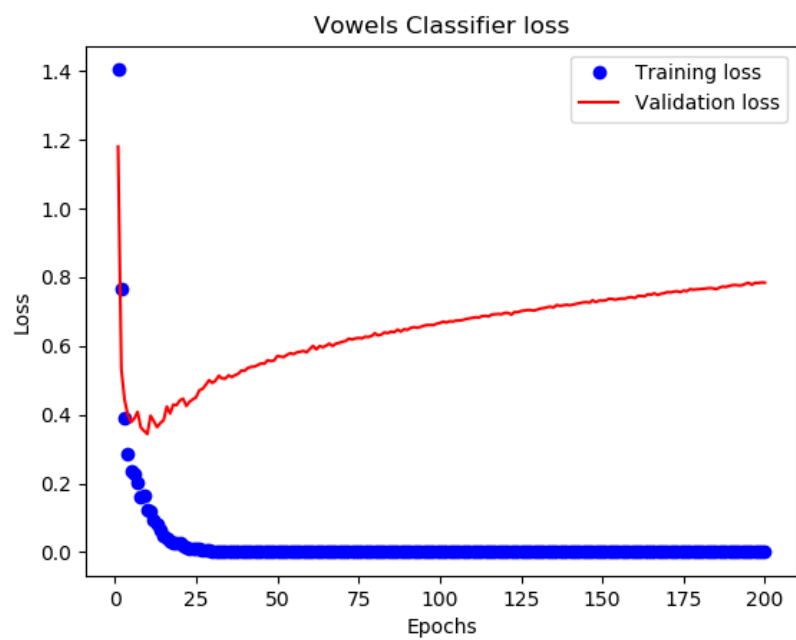


Figure 6: vogaisLoss

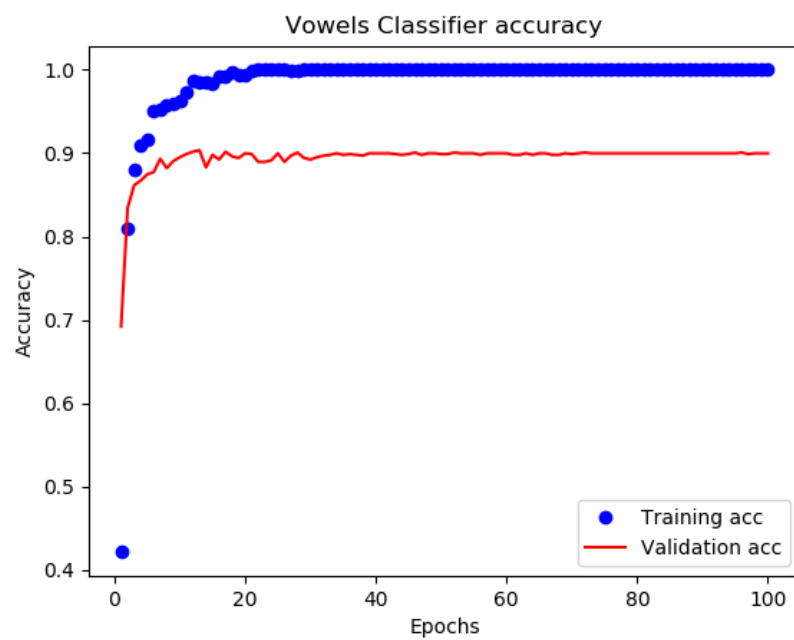


Figure 7: vogaisAcc



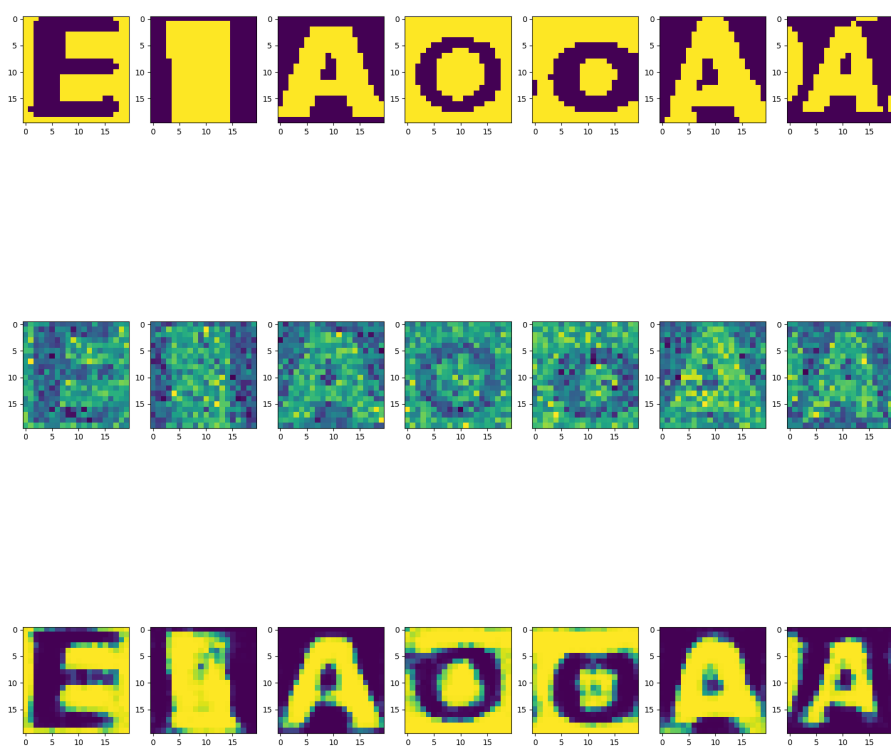


Figure 8: Vogais

*OBS:* Acurácia do modelo ficou em torno de 85%