

QUESTÃO 03 - ARTIGO

Identificação do perfil de clientes utilizando redes neurais convolucionais

2020

Victor Ribeiro de Azevedo

Universidade Estadual do Rio de Janeiro

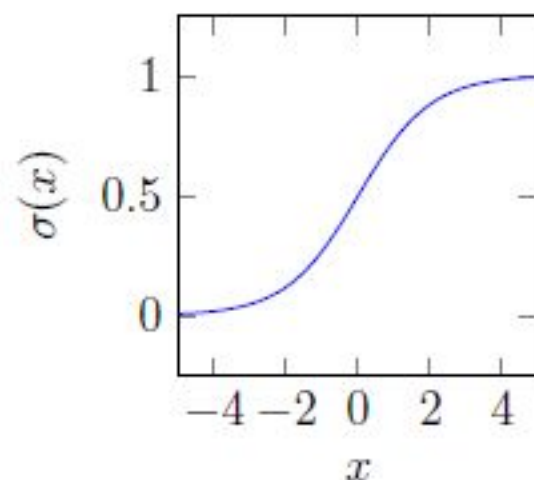
CONTRIBUIÇÕES

Objetivo:

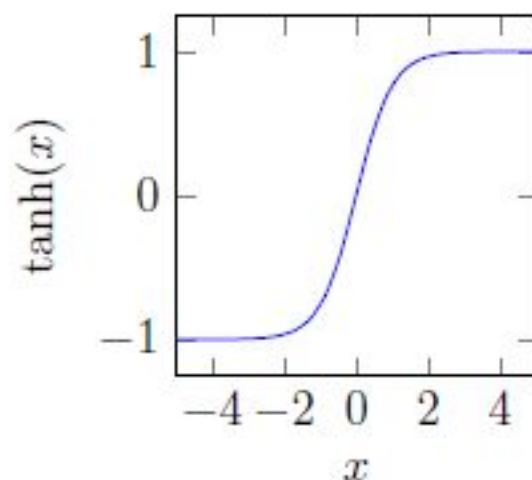
- Treinar algoritmos de aprendizado de máquina utilizando técnicas de otimização de rede neural artificial e comparar a performance dos modelos para classificação.

CONCEITOS

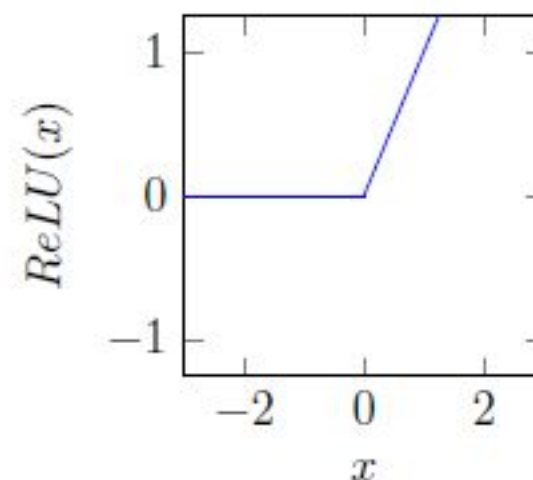
Por questões de performance, a prioridade na escolha de uma função de ativação deve ser primeiramente a ReLU. A Figura 4 ilustra as funções *sigmoide*, tangente hiperbólica e ReLU.



(a) Sigmoide



(b) Tangente hiperbólica



(c) ReLU

Figura 4: Funções de ativação: Sigmoide, Tangente hiperbólica e ReLU

CONCEITOS

Descida do gradiente

$$y_j^{(c)} = f^{(c)} \left(\sum_{i=1}^{N_{c-1}} x_i^{(c-1)} w_{ij}^{(c)} + b_j^{(c)} \right) \quad j = 1, 2, \dots, N_C,$$

$$E = \left\| f^{(c)} \left(\sum_{i=1}^{N_{c-1}} x_i^{(c-1)} w_{ij}^{(c)} + b_j^{(c)} \right) - o(t) \right\|^2.$$

- *SGD*: amostras são selecionadas aleatoriamente durante o treinamento levando a convergência mais rápida de forma ruidosa devido ao caráter estocástico.
- *AdaGrad*: reduz a taxa de aprendizado dos pesos que recebem gradientes altos aumentando a taxa de acurácia.
- *RMSProp*: modifica a taxa de aprendizado antes de aplicar os gradientes convergindo mais rápido.
- *Adam*: modifica o *AdaGrad* suavizando os ruídos e faz com que a taxa de aprendizado não diminua agressivamente.
- *AdaMax*: modifica o *Adam* para aumentar a velocidade de convergência.

CONCEITOS

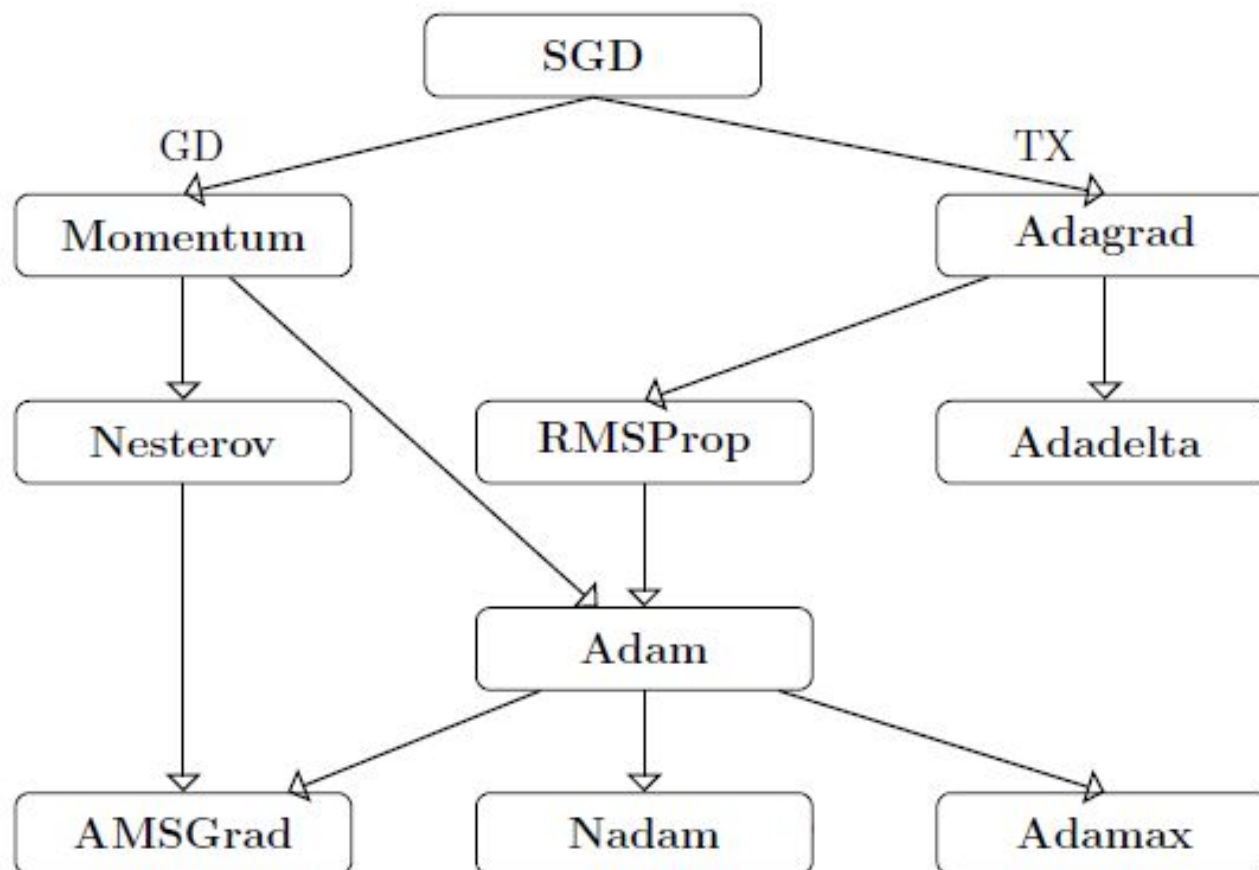


Figura 10: Algoritmos de otimização do gradiente descendente

CONCEITOS

AdaGrad

O algoritmo de gradiente adaptativo Adagrad foi proposto em (DUCHI; HAZAN; SINGER, 2011) e tem o efeito de reduzir a taxa de aprendizado dos pesos que recebem gradientes altos aumentando a acurácia. É um método onde a taxa de aprendizado é normalizada pela raiz quadrada da soma dos valores quadrados passados. Ou seja, a soma acumula os gradientes quadrados atuais e passados. A atualização do peso para este algoritmo é definida na Equação 16:

$$w_t = w_{t-1} - \eta \frac{\nabla E_t(w_{t-1})}{\sqrt{\sum_{t'=0}^t \nabla E_{t'}(w_{t'-1})^2}},$$

CONCEITOS

RMSprop

No algoritmo de propagação da raiz do valor quadrático médio denominado *RMSprop* é realizado o cálculo das médias da magnitude dos gradientes para cada parâmetro. Estas médias são utilizadas para modificar a taxa de aprendizado individualmente antes de aplicar os gradientes. Este algoritmo modifica o *AdaGrad* com uma média móvel exponencial controlada pela taxa $\gamma \in [0, 1)$ que possui uma ênfase em gradientes recentes para atingir um melhor desempenho. O valor quadrático médio dos gradientes R_t é definido na Equação 21:

$$R_t = \gamma R_{t-1} + (1 - \gamma) \nabla E_t(w_{t-1})^2,$$

onde γ é a taxa que controla a escala da média móvel. A atualização do peso para este algoritmo pode então ser descrita na Equação 22:

$$w_t = w_{t-1} - \eta \frac{\nabla E_t(w_{t-1})}{\sqrt{R_t}}.$$

CONCEITOS

Adam

O algoritmo de treinamento de estimativa de momento adaptativo (*Adaptive moment estimation* - Adam) proposto por (KINGMA; BA, 2014) é um método derivado do *RMSprop* que ajusta o método *Adagrad* para que a taxa de aprendizado não diminua agressivamente.

A atualização do peso para este algoritmo pode então ser descrita na Equação 23:

$$w_t = w_{t-1} - \eta \frac{\hat{M}_t}{\sqrt{\hat{R}_t + \epsilon}},$$

onde η é a taxa de aprendizado, ϵ é o fator de decaimento. Vale ressaltar que η e ϵ devem ser maiores que zero. A atualização da primeira estimativa do momento é definida na Equação 24:

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1) \nabla E_t(w_{t-1}),$$

onde $\beta_1 \in [0, 1)$ é taxa de decaimento do primeiro momento. A atualização da segunda estimativa do momento é definida na Equação 25:

$$R_t = \beta_2 R_{t-1} + (1 - \beta_2) \nabla E_t(w_{t-1})^2,$$

onde $\beta_2 \in [0, 1)$ é a taxa de decaimento do segundo momento. Calcula-se a estimativa do primeiro momento corrigida pelo *bias* na Equação 26:

$$\hat{M}_t = \frac{M_t}{(1 - \beta_1^t)},$$

onde o segundo momento corrigido pelo *bias* é definido na Equação 26:

$$\hat{R}_t = \frac{R_t}{(1 - \beta_2^t)}.$$

CONCEITOS

Adamax

O algoritmo de gradiente adaptável baseado na norma infinita (*Adaptive stochastic gradient descent* - Adamax) é uma variante do *Adam* onde o momento de segunda ordem é substituído pelo momento de ordem infinita (KINGMA; BA, 2014).

A atualização da primeira estimativa do momento é definida na Equação 28:

onde $\beta_1 \in [0, 1)$ é a taxa de decaimento do primeiro momento. A atualização da segunda estimativa do momento é definida na Equação 25:

onde $\beta_2 \in [0, 1)$ é a taxa de decaimento do segundo momento e p é a constante de atualização. Calcula-se a estimativa do primeiro momento corrigida pelo *bias* na Equação 30:

Calcula-se a estimativa do segundo momento corrigida pelo *bias* na Equação 31:

A atualização do peso parametrizado pela constante de atualização para este algoritmo pode então ser descrita na Equação 32:

onde p é a constante de atualização. Quando $p \rightarrow \infty$ os termos são modificados, conforme observado em (KINGMA; BA, 2014), gerando novas equações com a estimativa do momento de ordem infinita. A atualização da primeira estimativa do momento de ordem infinita é definida na Equação 33:

Quando $p \rightarrow \infty$ na Equação 31 define-se um novo termo de maximização do gradiente na Equação 34:

A atualização do peso para este algoritmo pode então ser descrita na Equação 35:
onde η é a taxa de aprendizado.

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1) \nabla E_t(w_{t-1}),$$

$$R_t = \beta_2 R_{t-1} + (1 - \beta_2) \nabla E_t(w_{t-1})^p,$$

$$\hat{M}_t = \frac{M_t}{(1 - \beta_1^t)}, \quad \hat{R}_t = \frac{R_t}{(1 - \beta_2^t)}.$$

$$w_t = w_{t-1} - \eta \frac{M_t}{(\hat{R}_t + \epsilon)^p},$$

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1) \nabla E_t(w_{t-1}).$$

$$U_t = \max \{ \beta_2 U_{t-1}, |\nabla E_t(w_{t-1})| \}.$$

$$w_t = w_{t-1} - \eta \frac{M_t}{U_t},$$

MATERIAIS E MÉTODOS

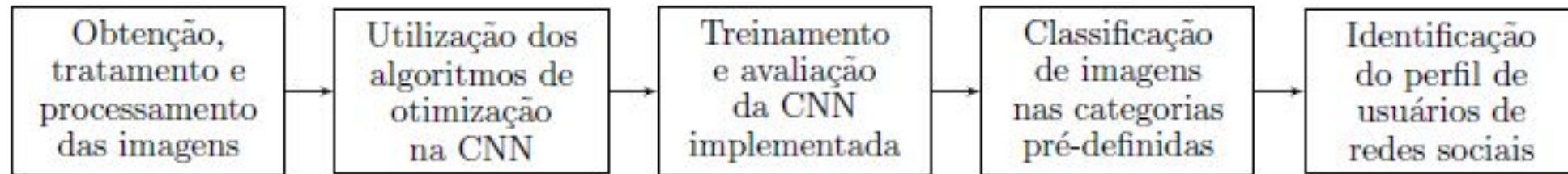


Figura 11: Fluxo proposto para identificação do perfil de usuários com treinamento utilizando algoritmos de otimização¹

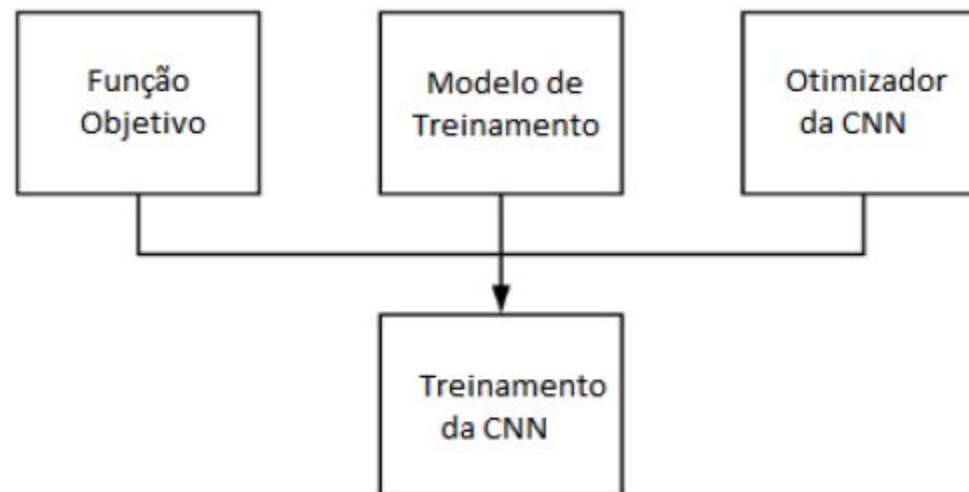


Figura 14: Execução do treinamento da rede neural

MATERIAIS E MÉTODOS

A primeira etapa do desenvolvimento do método de classificação consiste na busca de conjuntos de imagens nas categorias pré-definidas como animais, eletrônicos, jogos, veículos e vestuário.

Tabela 1: Configuração da arquitetura da CNN²

C	Filtro	N.F.	Dim.	Ativ.	Drop.	M.P.	T.C.	B.N.	Conv.	Qtd.
1	3x3	32	32x32	ReLU	Sim	3x3	-	Sim	Sim	5
2	3x3	64	32x32	ReLU	-	-	-	Sim	Sim	3
3	3x3	64	16x16	ReLU	Sim	2x2	-	Sim	Sim	5
4	3x3	128	16x16	ReLU	-	-	-	Sim	Sim	3
5	3x3	128	8x8	ReLU	Sim	2x2	-	Sim	Sim	5
6	-	-	1024x1	ReLU	Sim	-	Sim	Sim	-	5
7	-	-	5x1	Softmax	-	-	Sim	-	-	2

Métricas de qualidade

Ao final dos experimentos, o desempenho de cada algoritmo foi analisado quanto ao erro, acurácia, sensibilidade, precisão e a medida F1.

RESULTADO

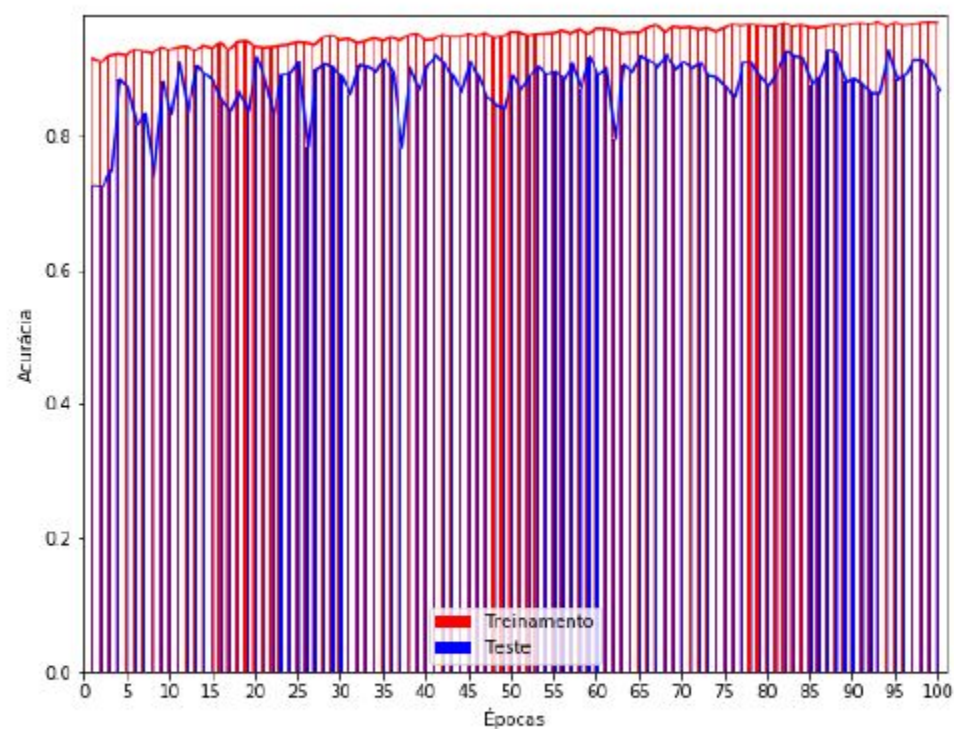


Figura 22: Crescimento da acurácia no treinamento e no teste com o algoritmo SGD

RESULTADO

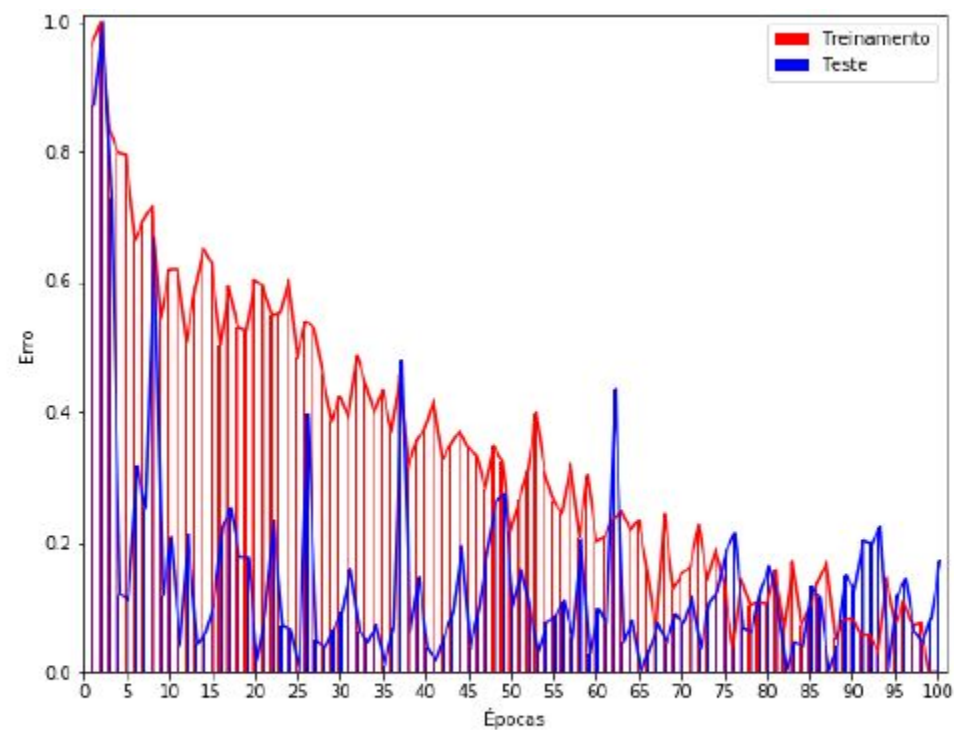


Figura 23: Decaimento do erro no treinamento e no teste com o algoritmo SGD

RESULTADO

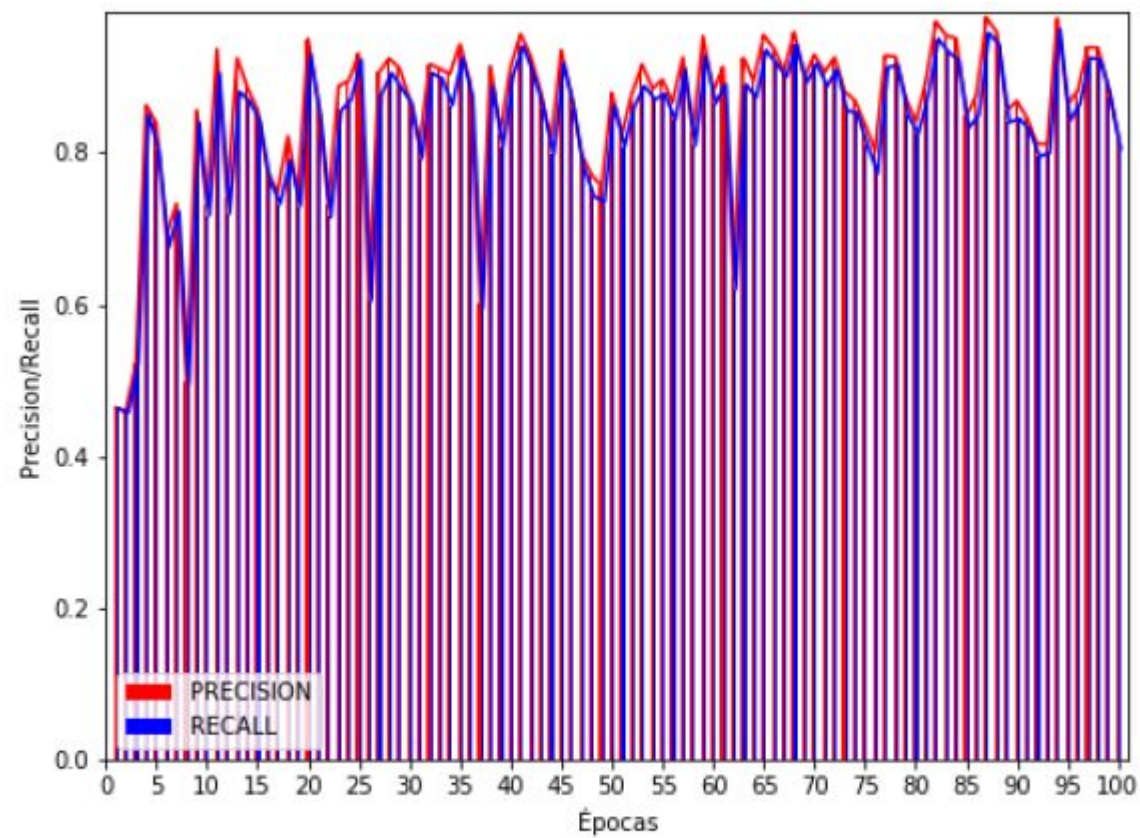


Figura 24: Métricas Precisão e Sensibilidade para o algoritmo SGD

RESULTADO

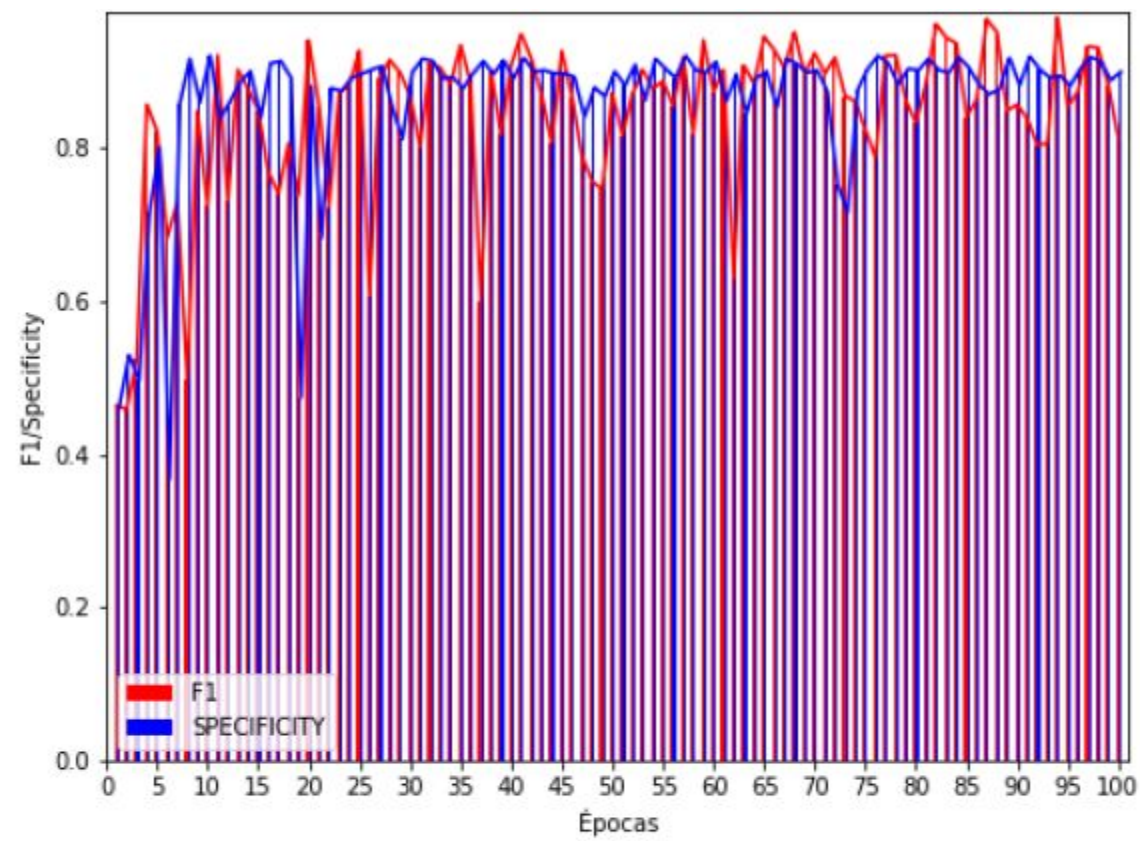


Figura 25: Métricas F1 e Especificidade para o algoritmo *SGD*

RESULTADO

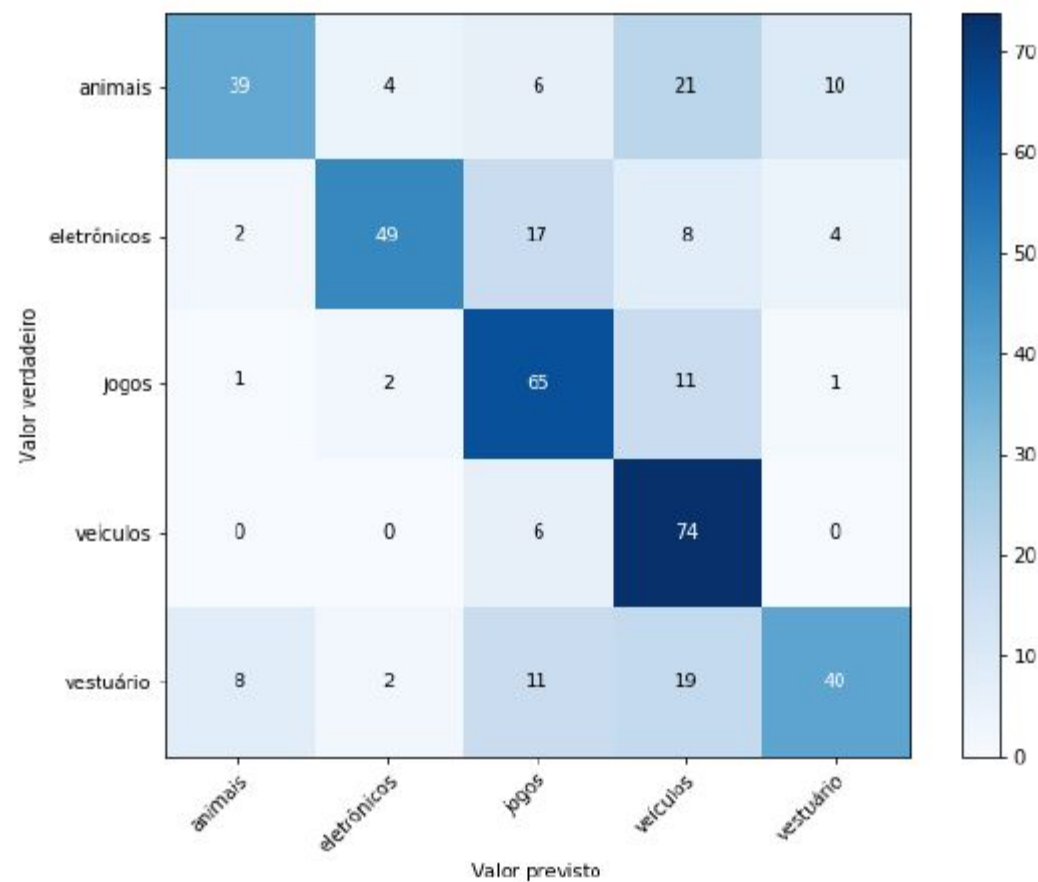


Figura 26: Matriz de confusão para o algoritmo *SGD*

RESULTADO

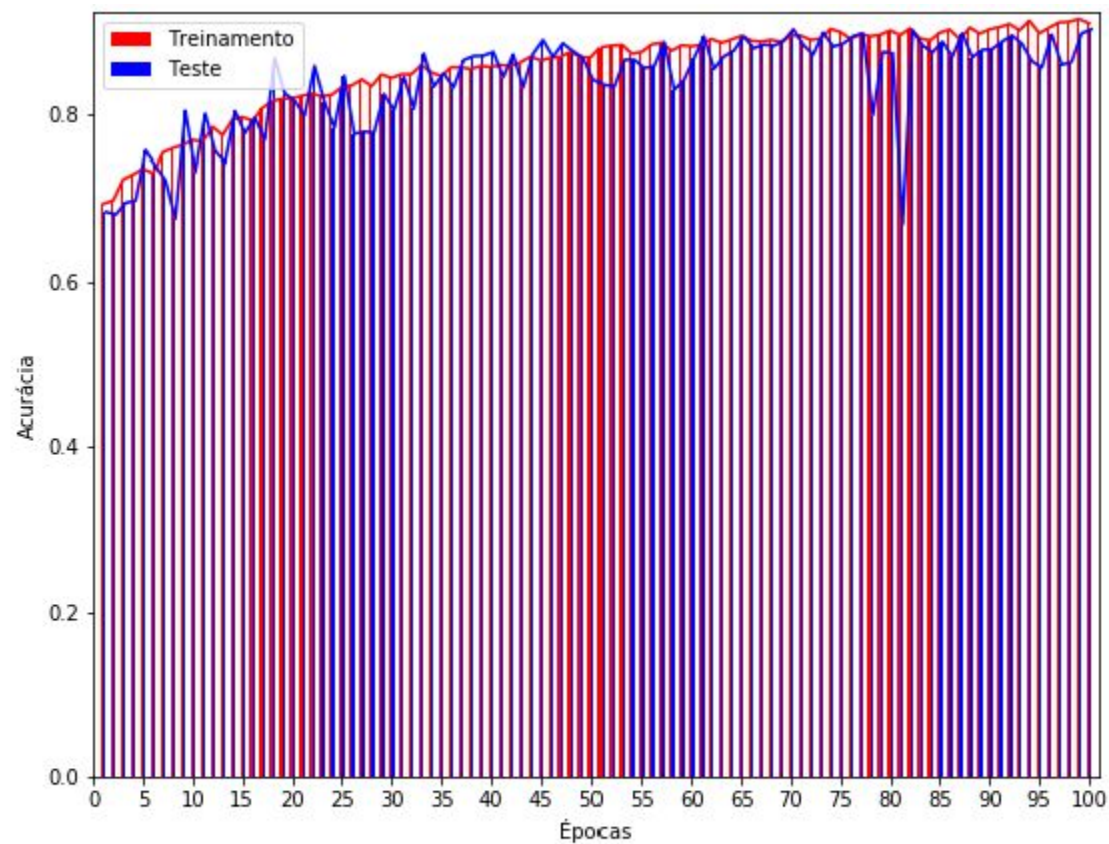


Figura 27: Crescimento da acurácia no treinamento e no teste com o algoritmo *AdaGrad*

RESULTADO

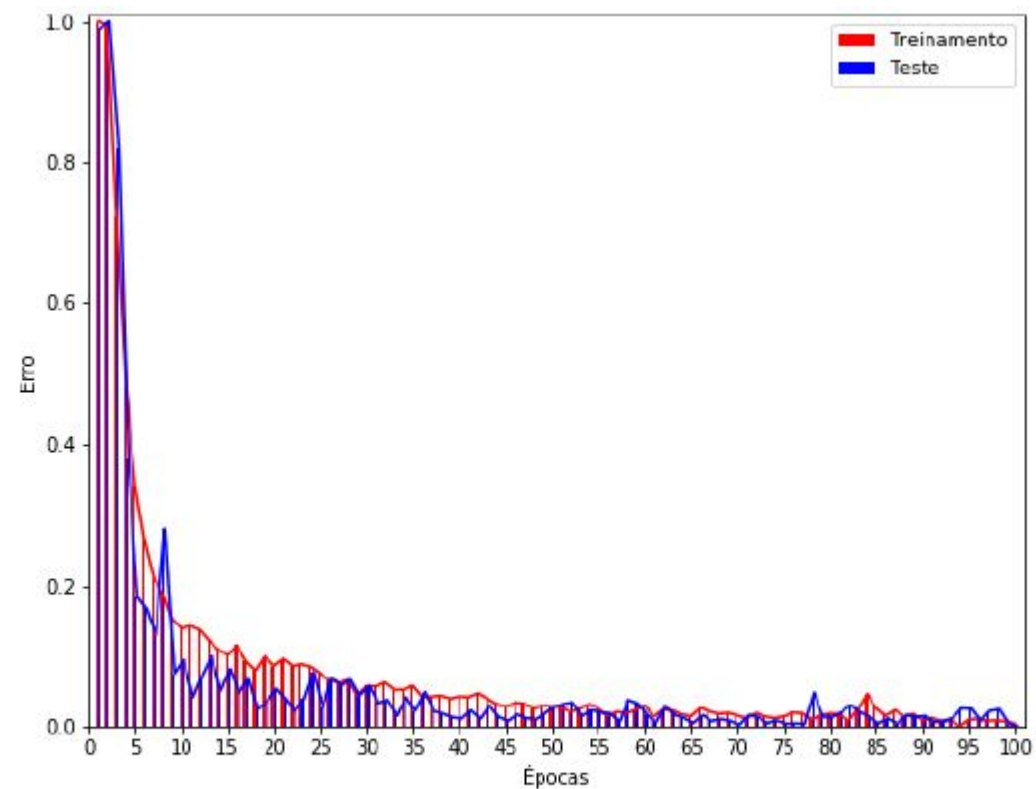


Figura 28: Decaimento do erro no treinamento e no teste com o algoritmo *AdaGrad*

RESULTADO

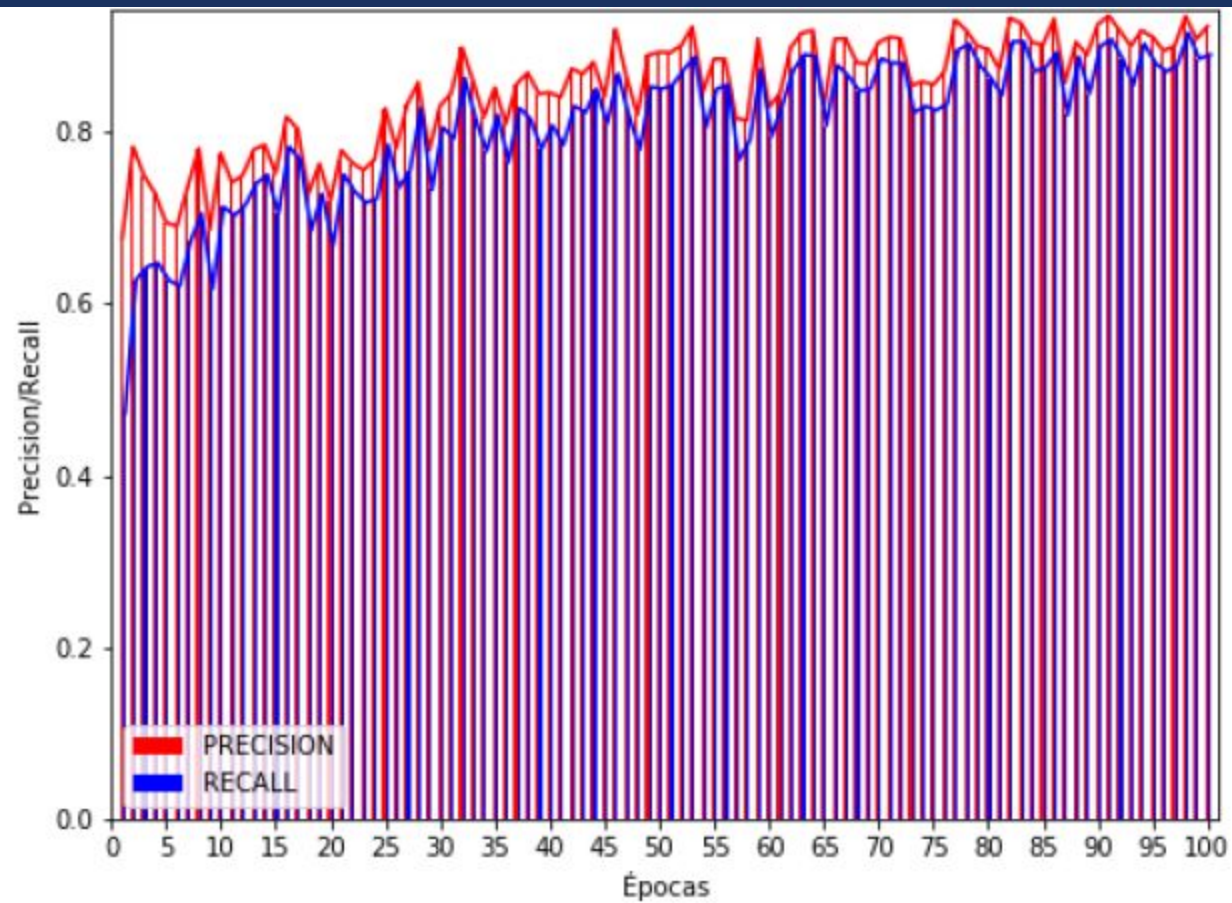


Figura 29: Métricas Precisão e Sensibilidade para o algoritmo *AdaGrad*

RESULTADO

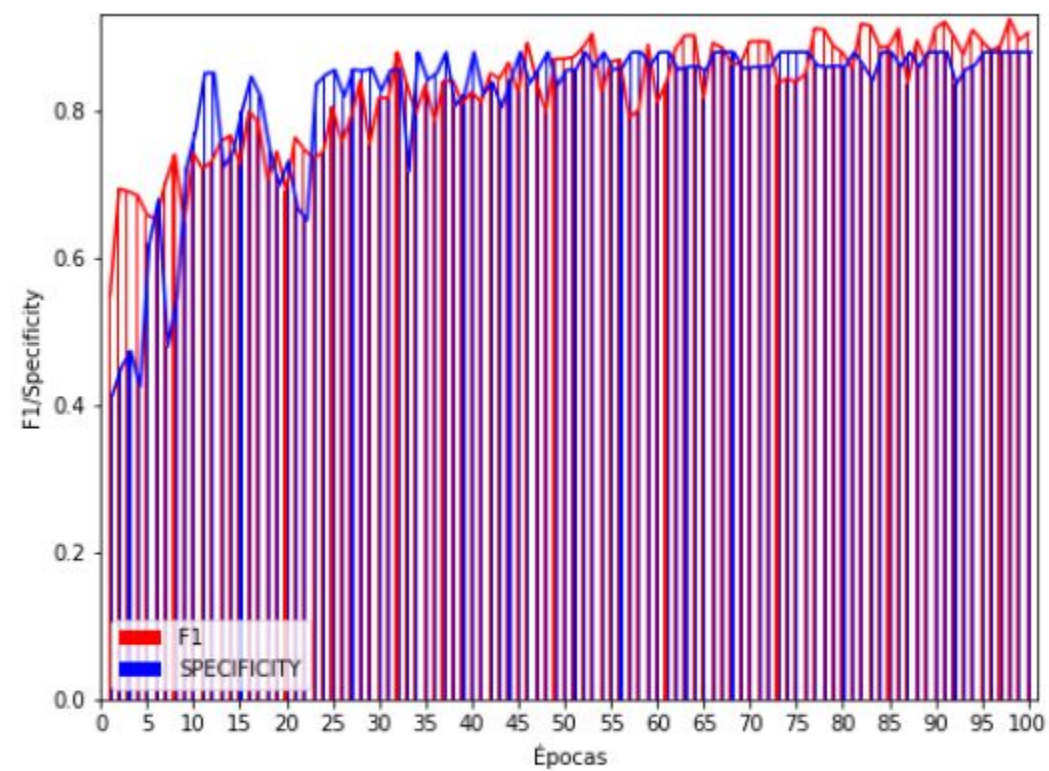


Figura 30: Métricas F1 e Especificidade para o algoritmo *AdaGrad*

RESULTADO

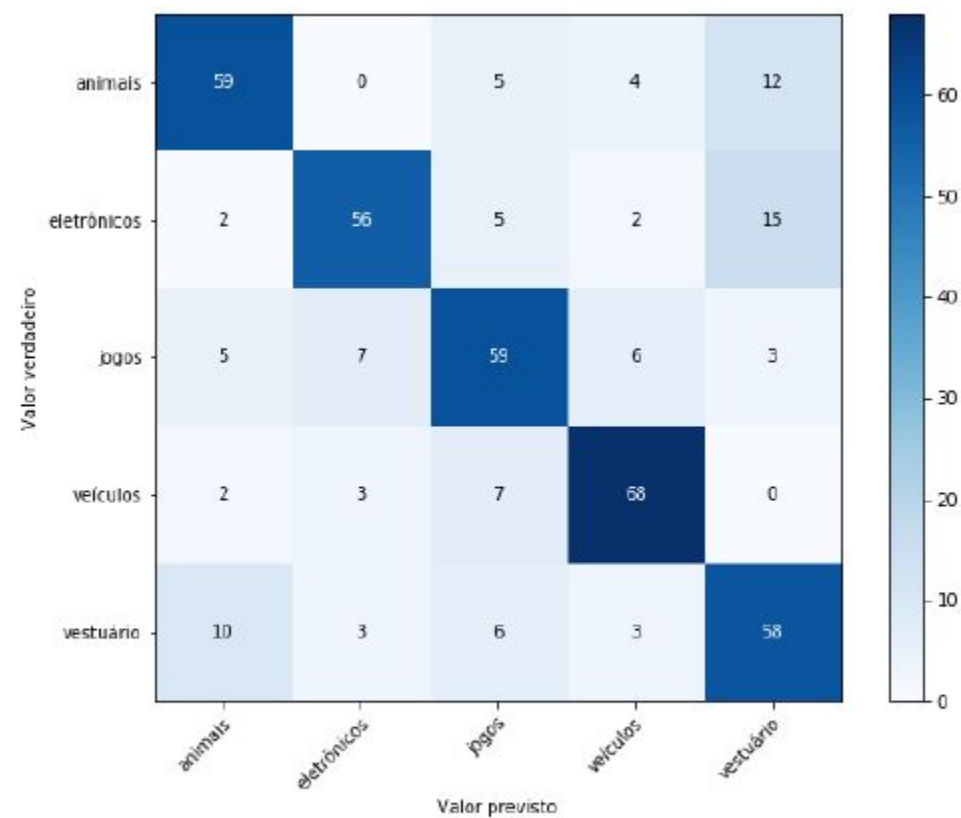


Figura 31: Matriz de confusão para o algoritmo *AdaGrad*

RESULTADO

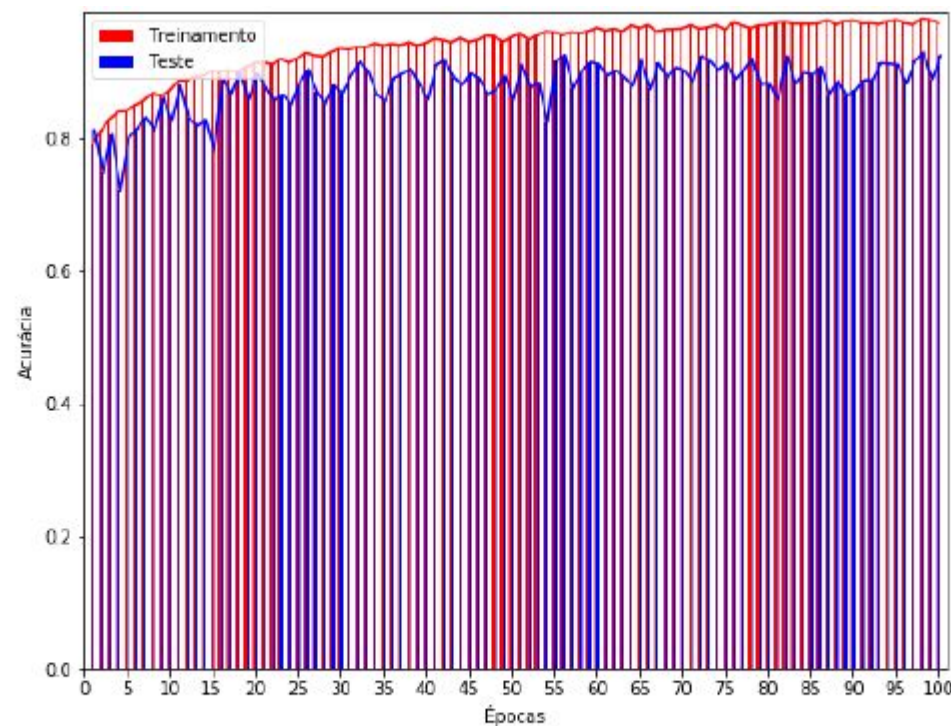


Figura 32: Crescimento da acurácia no treinamento e no teste com o algoritmo *RMSprop*

RESULTA

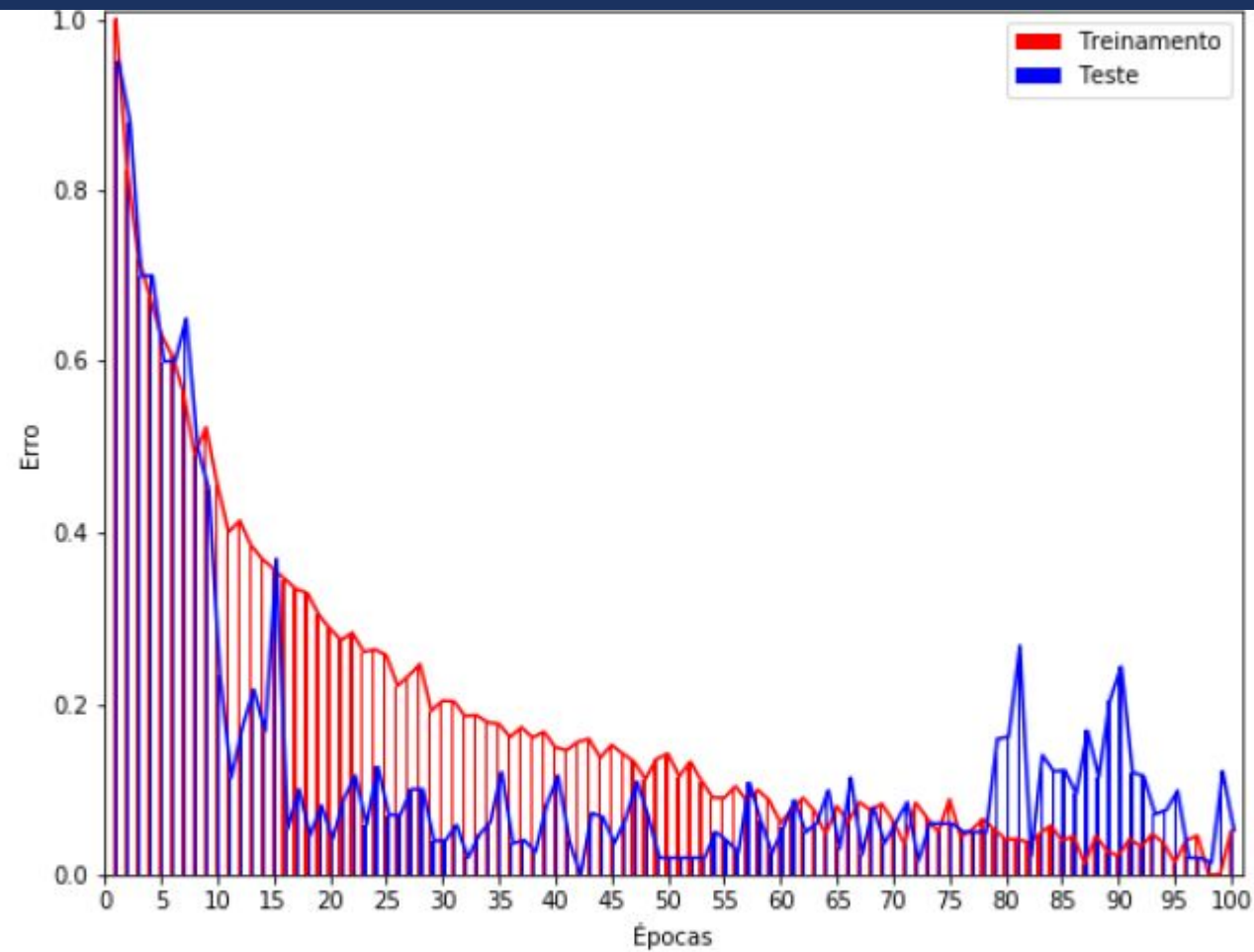


Figura 33: Decaimento do erro no treinamento e no teste com o algoritmo *RMSprop*

RESULTADO

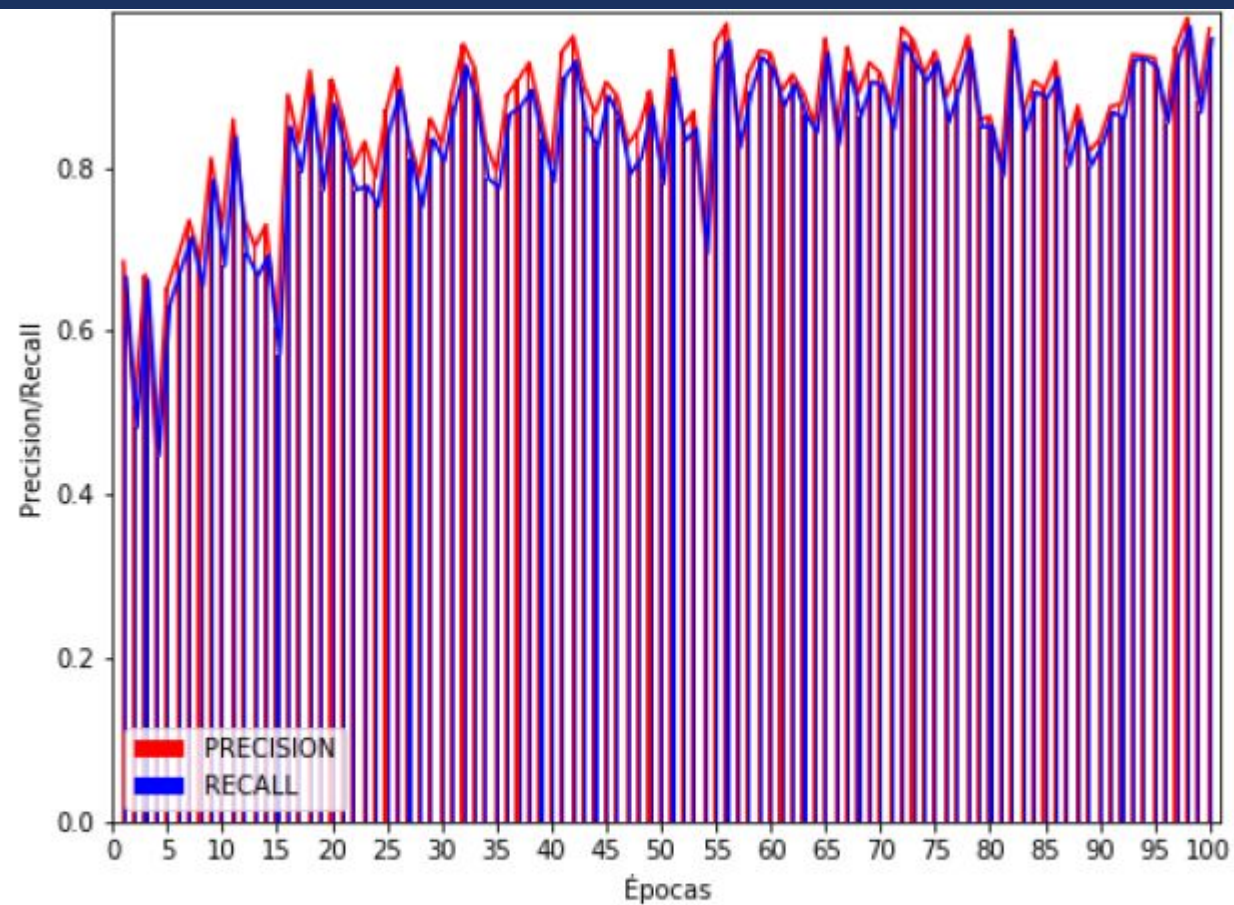


Figura 34: Métricas Precisão e Sensibilidade para o algoritmo *RmsProp*

RESULTADO

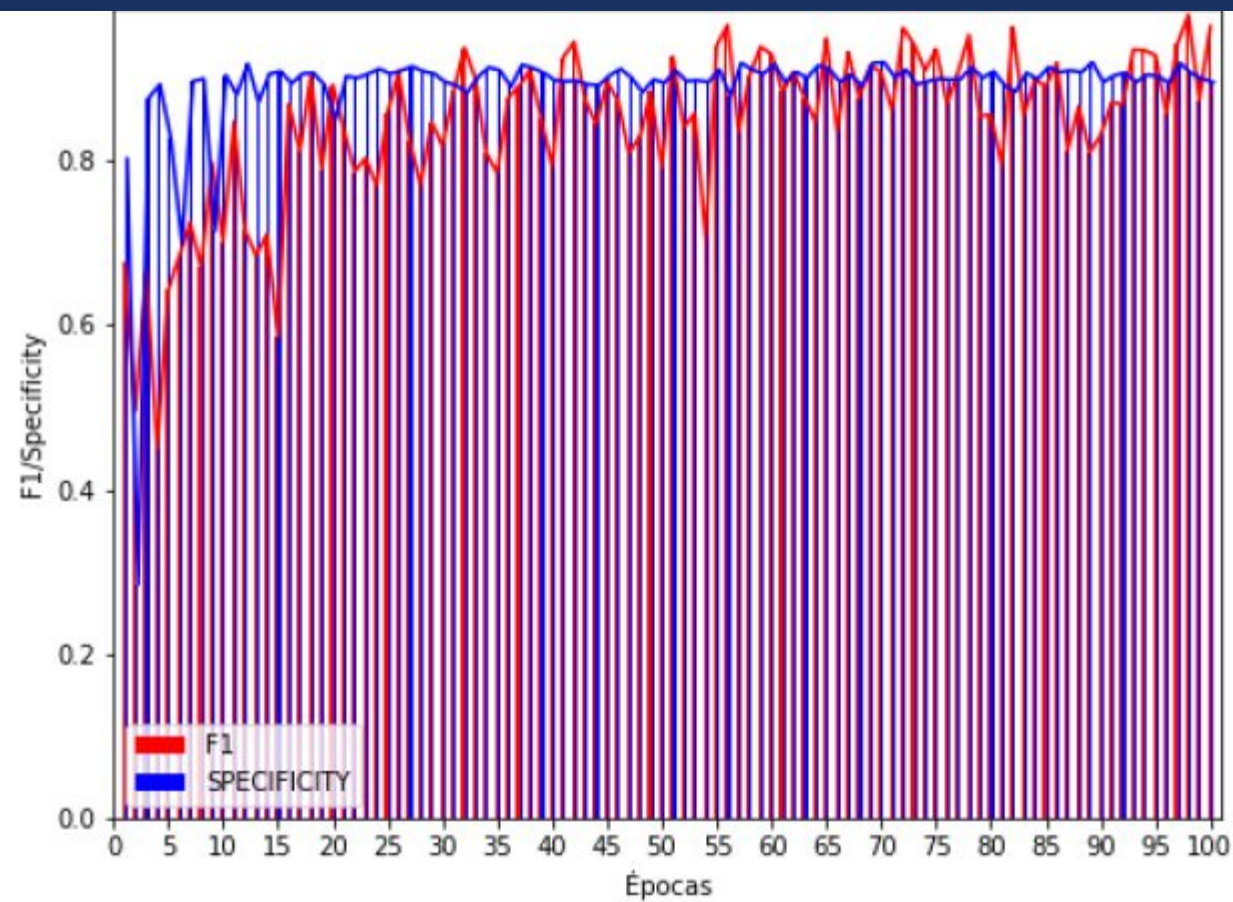


Figura 35: Métricas F1 e Especificidade para o algoritmo *RmsProp*

RESULTADO

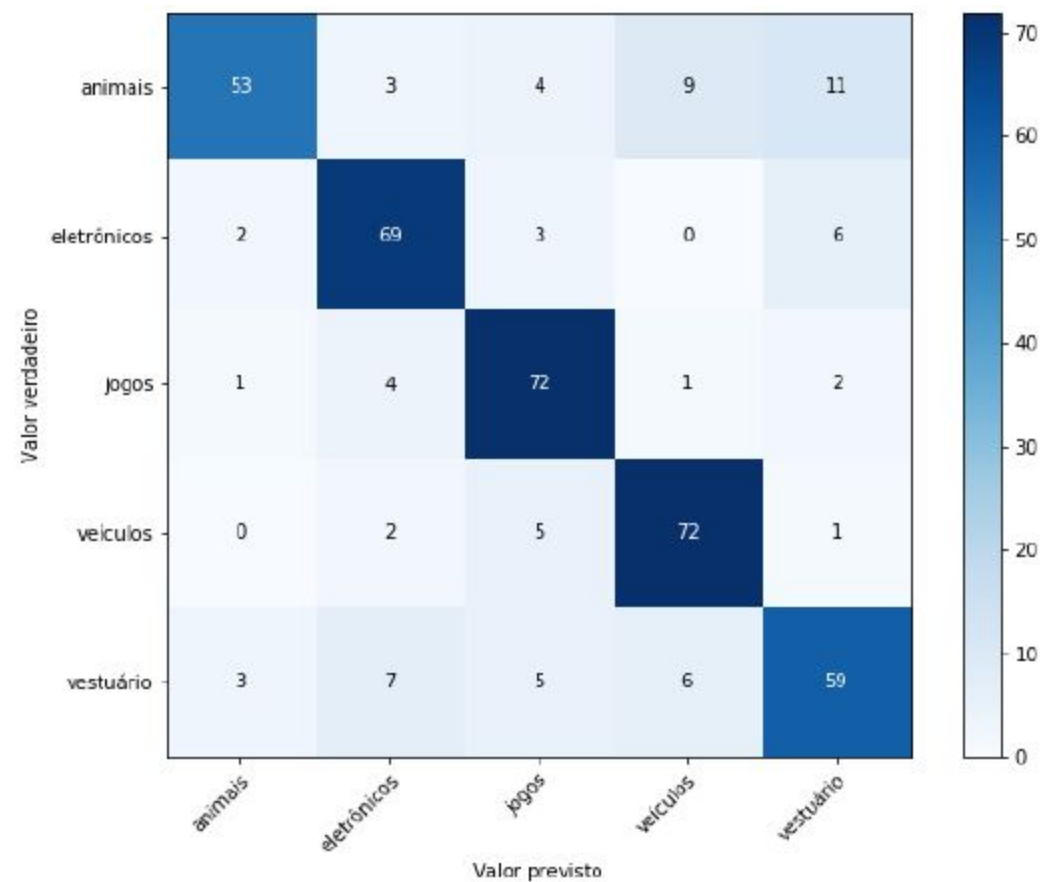


Figura 36: Matriz de confusão para o algoritmo *RMSprop*

RESULTADO

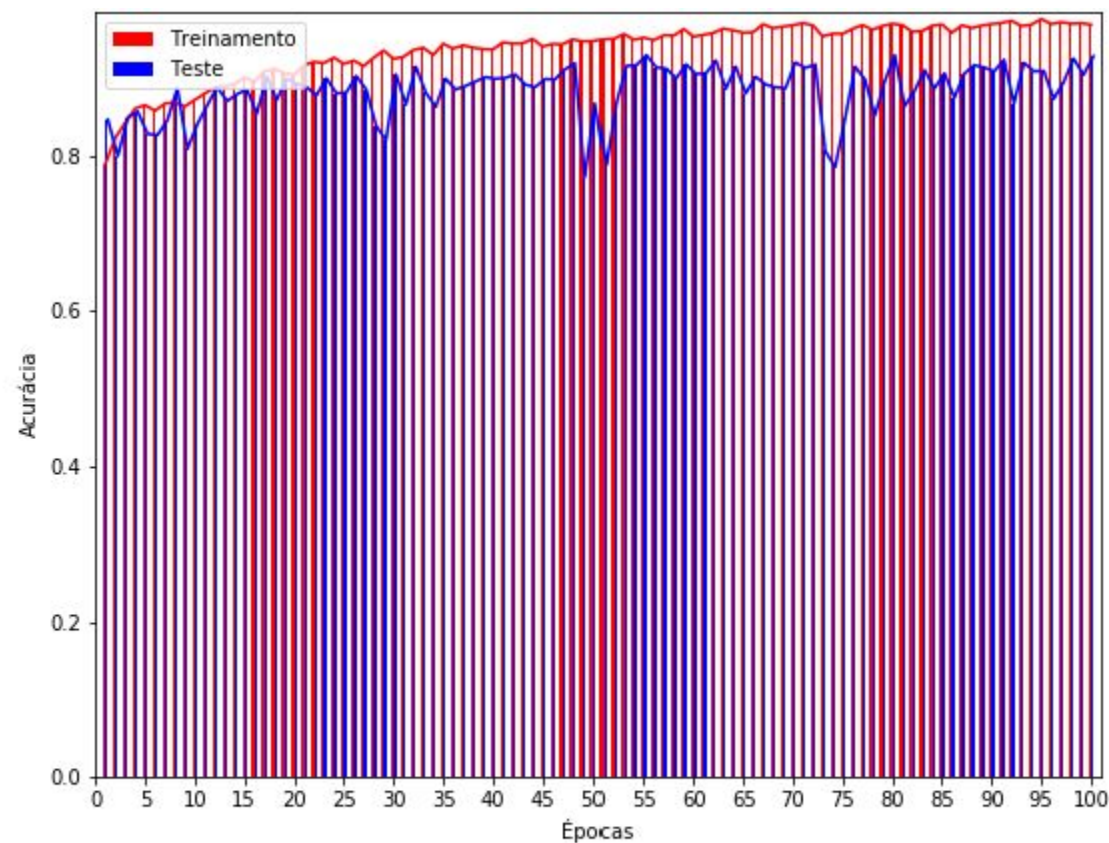


Figura 37: Crescimento da acurácia no treinamento e no teste com o algoritmo *Adam*

RESULTADO

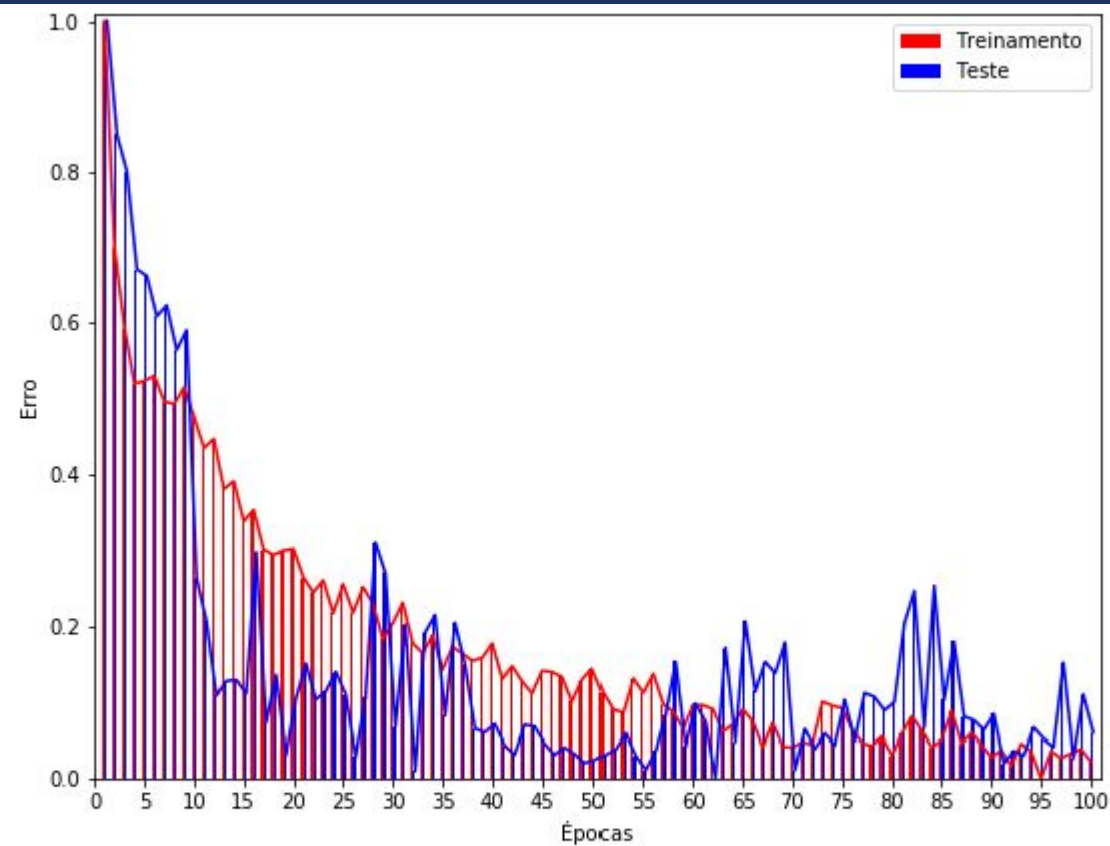


Figura 38: Decaimento do erro no treinamento e no teste com o algoritmo *Adam*

RESULTADO

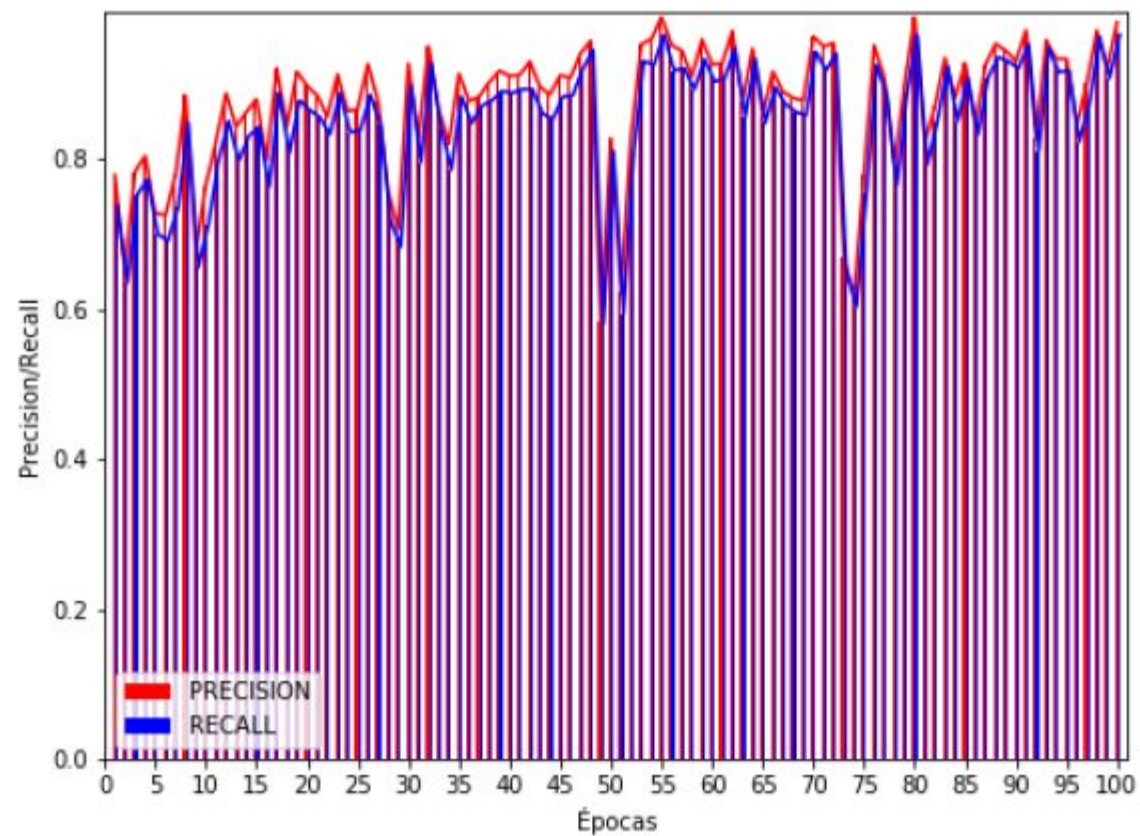


Figura 39: Métricas Precisão e Sensibilidade para o algoritmo *Adam*

RESULTADO

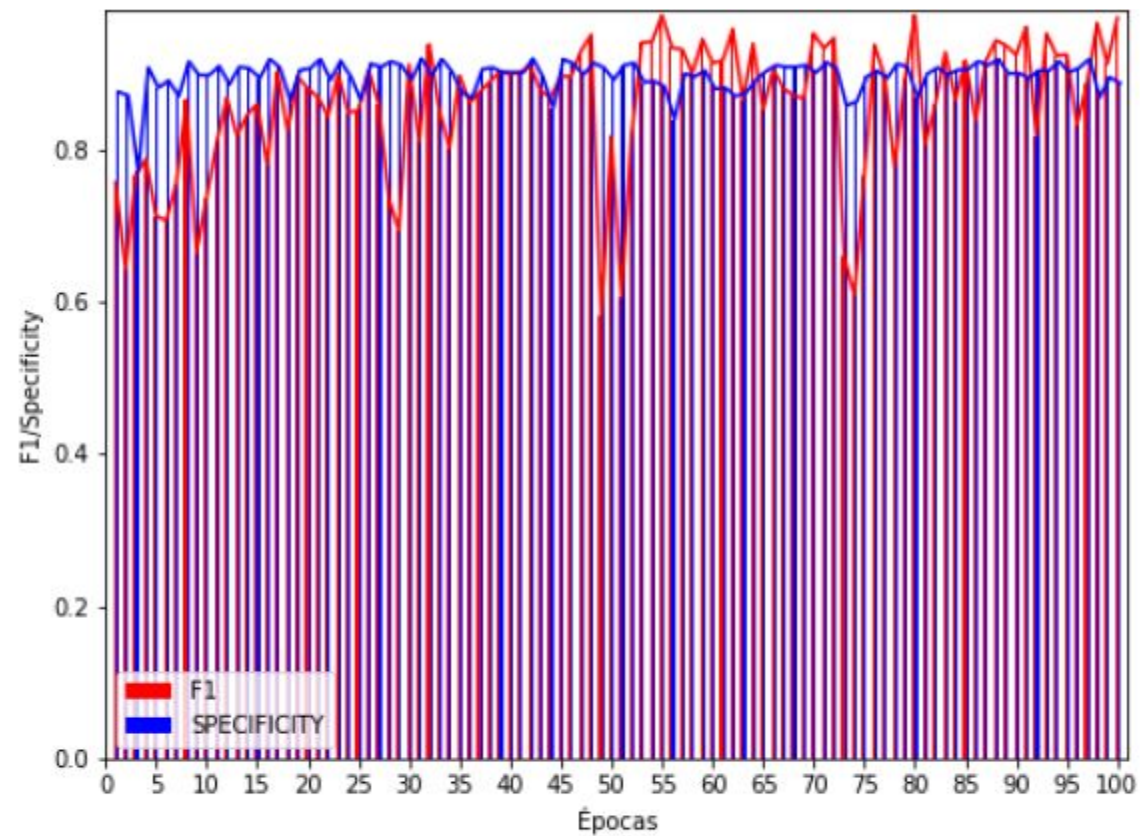


Figura 40: Métricas F1 e Especificidade para o algoritmo *Adam*

RESULTADO

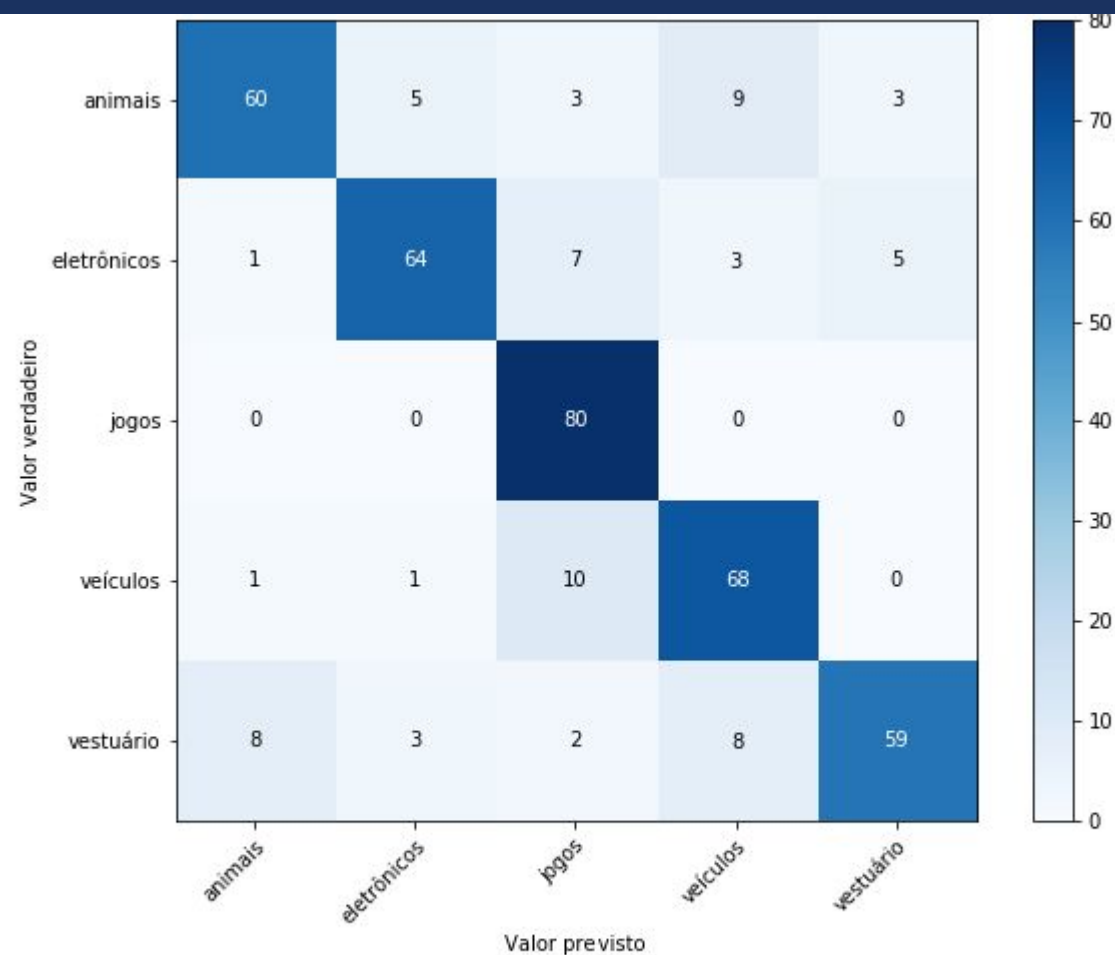


Figura 41: Matriz de confusão para o algoritmo *Adam*

RESULTADO

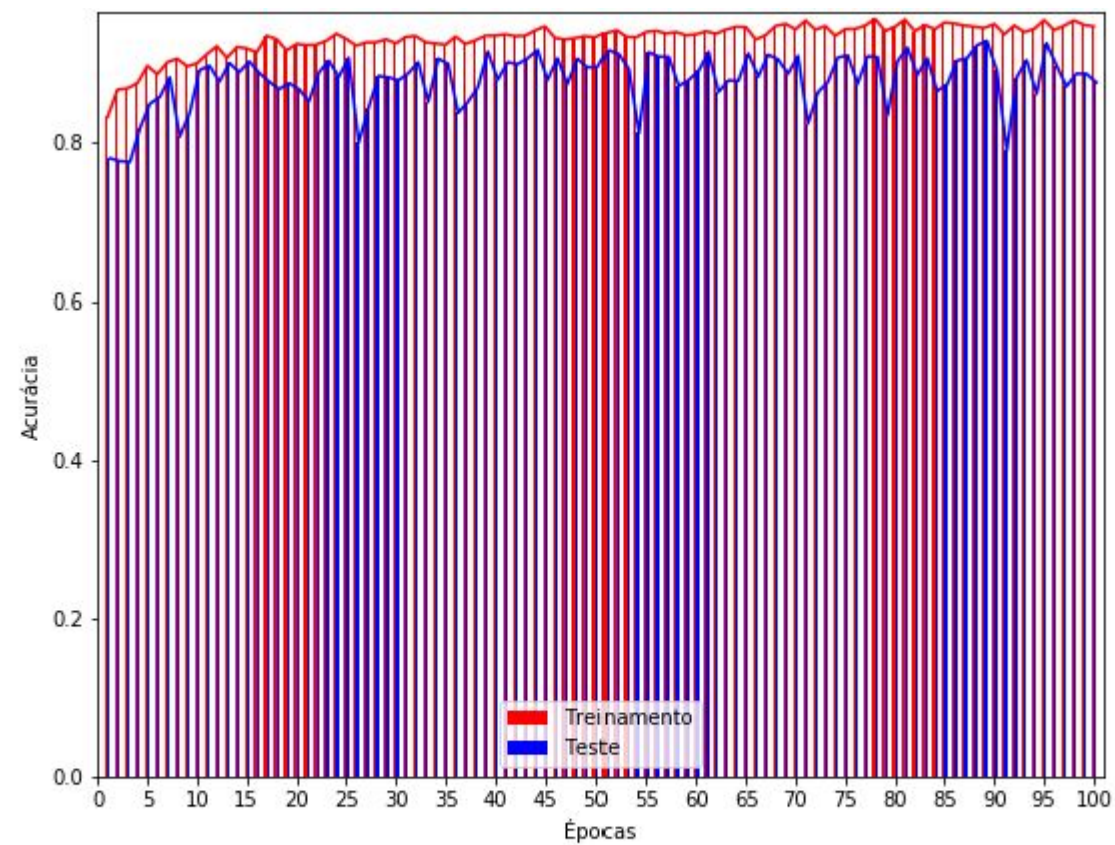


Figura 42: Crescimento da acurácia no treinamento e no teste com o algoritmo *Adamax*

RESULTADO

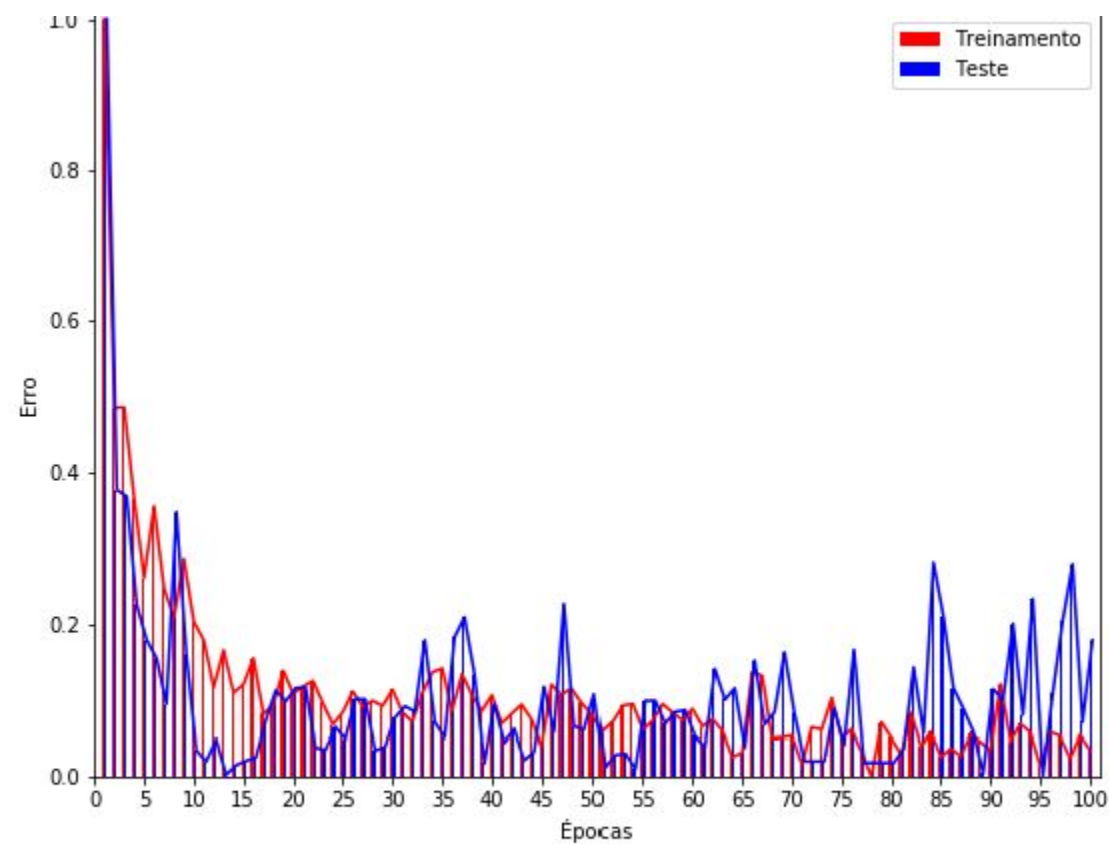


Figura 43: Decaimento do erro no treinamento e no teste com o algoritmo *Adamax*

RESULTADO

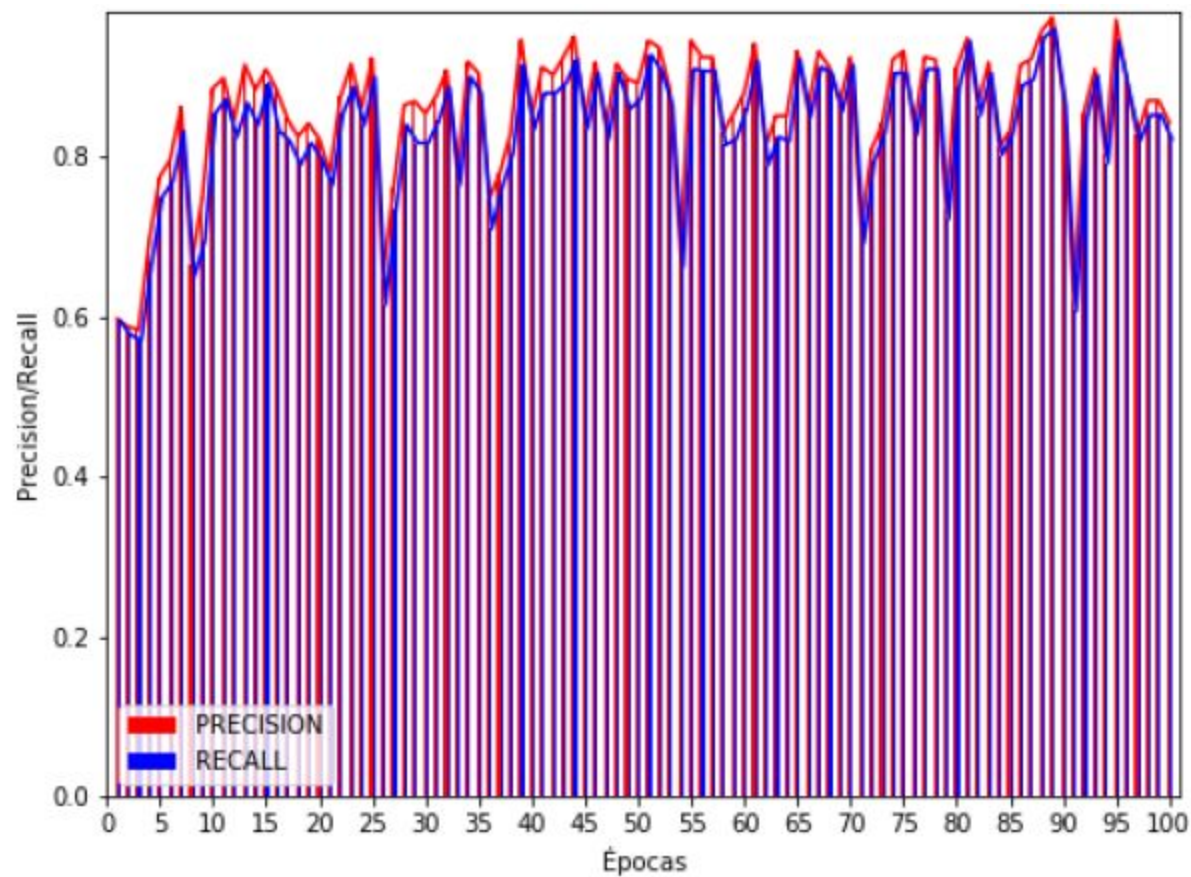


Figura 44: Métricas Precisão e Sensibilidade para o algoritmo *Adamax*

RESULTADO

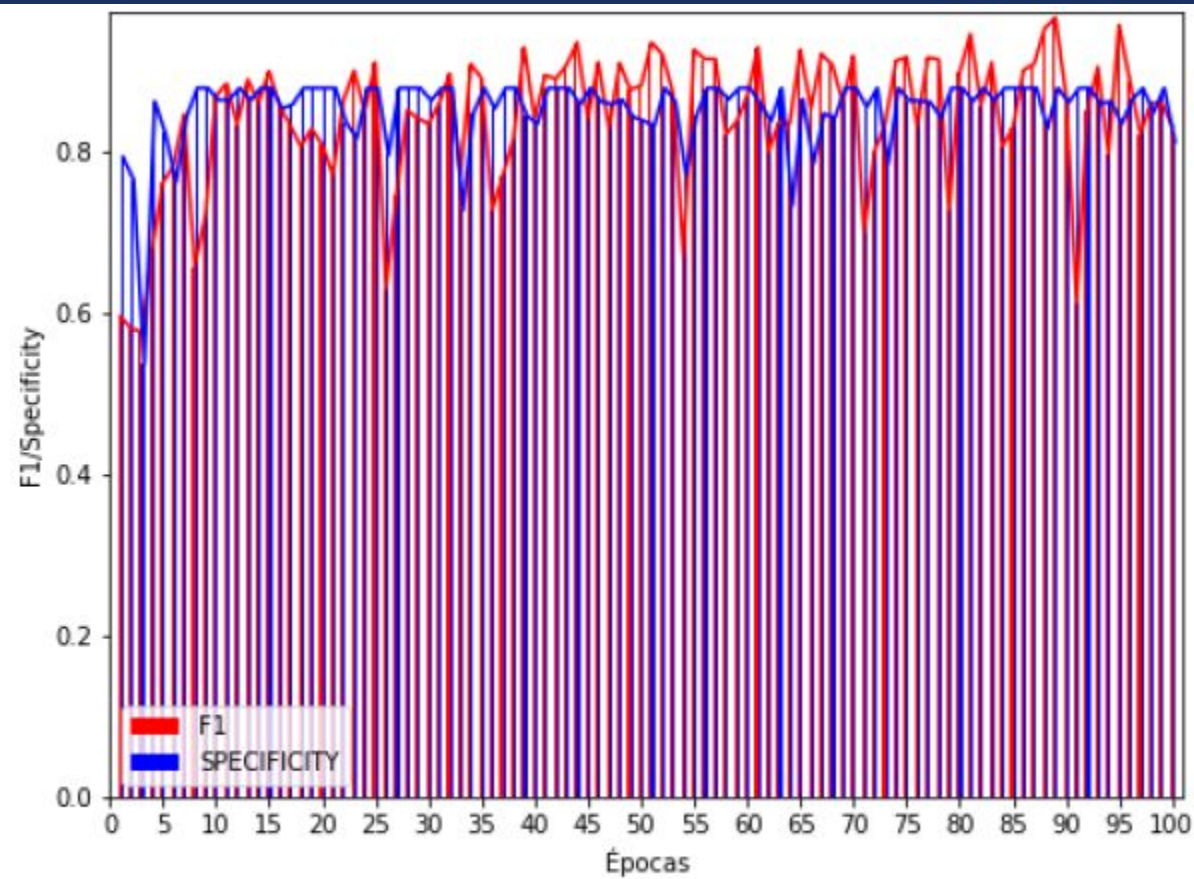


Figura 45: Métricas F1 e Especificidade para o algoritmo *Adamax*

RESULTADO

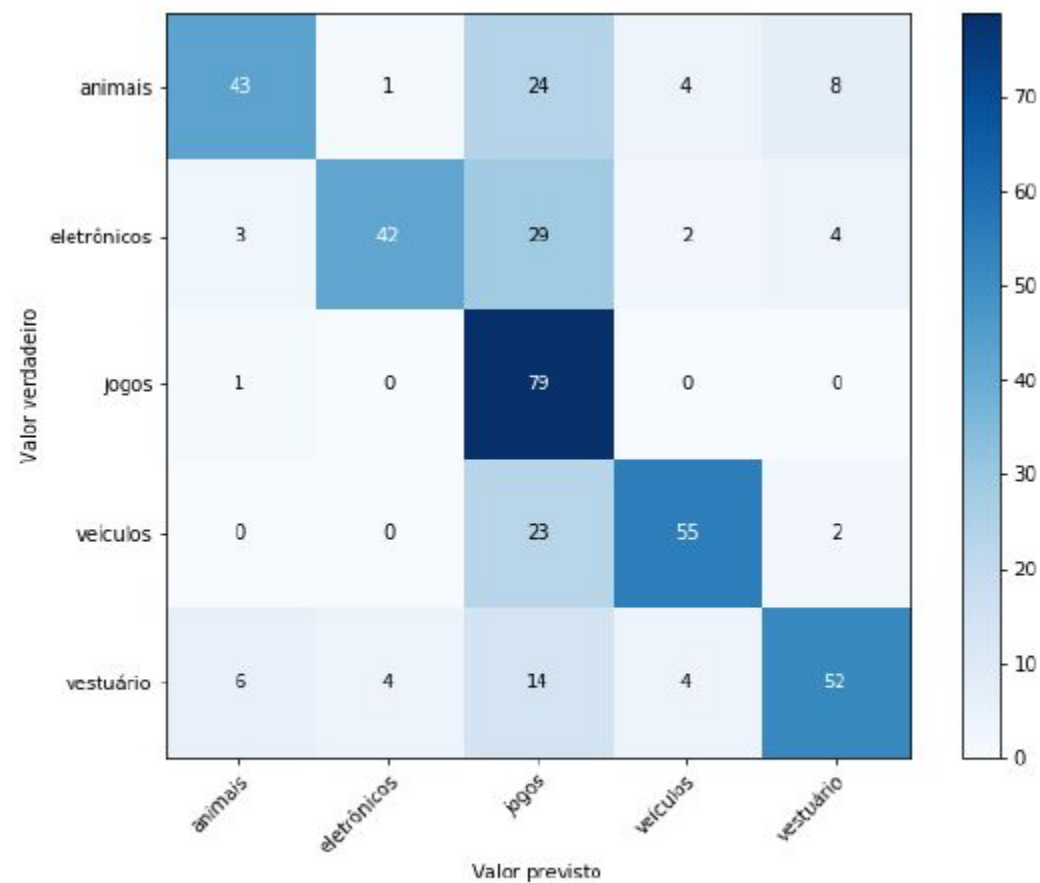


Figura 46: Matriz de confusão para o algoritmo *Adamax*

RESULTADO

Tabela 11: Comparativo do desempenho dos algoritmos

Algoritmo	Acc. Treino	Erro Treino	Acc. Teste	Erro Teste	Conf.
SGD	0,95	0,34	0,88	0,14	267
AdaGrad	0,86	0,08	0,84	0,12	300
RMSProp	0,93	0,19	0,88	0,13	325
Adam	0,94	0,18	0,90	0,10	331
Adamax	0,93	0,11	0,88	0,12	271