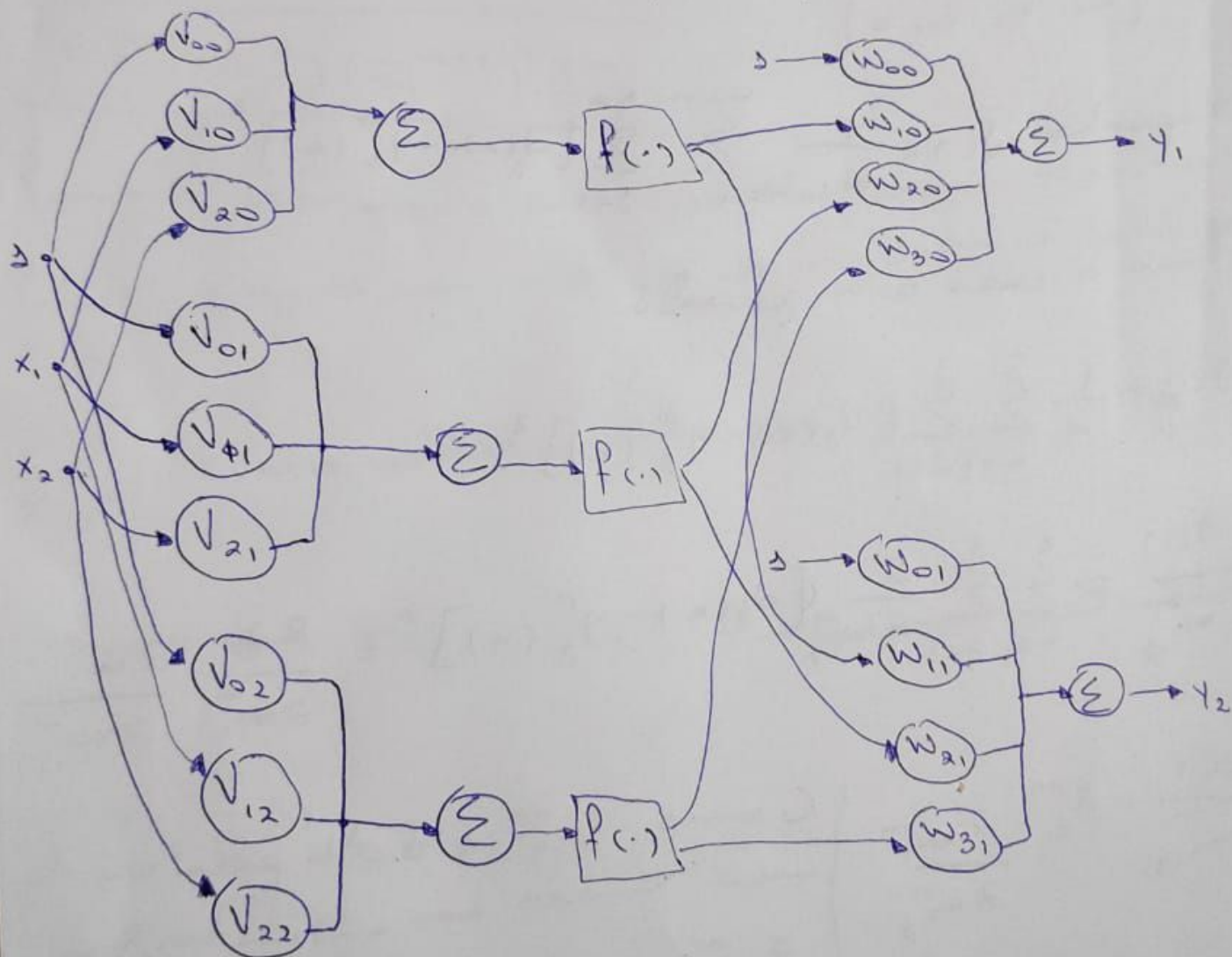


Diogo Trigueiro Angelo - 210479

EEC 3

Problema I - Rede MLP com duas entradas (x_1, x_2)
dois neurônios na camada intermediária
($N=3$) e duas saídas (y_1 e y_2).



Função de custo $J(\cdot) = e_1^2 + e_2^2$

Calculando a saída, temos:

$$\bar{W} = \begin{bmatrix} w_{00} & w_{10} & w_{20} & w_{30} \\ w_{01} & w_{11} & w_{21} & w_{31} \end{bmatrix}; \quad \bar{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}; \quad \bar{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$\bar{V} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ v_{00} & v_{10} & v_{20} & 0 \\ v_{01} & v_{11} & v_{21} & 0 \\ v_{02} & v_{12} & v_{22} & 0 \end{bmatrix}; \quad \boxed{\bar{y} = \bar{W} \bar{V} \bar{x}}$$

Superfície $J(\cdot) = \frac{1}{2} \sum_{k=1}^{N_{\text{dados}}} \sum_{j=1}^{N_m} [d_j(k) - y_j^m(x)]^2$

Para o caso em questão:

$$J(\cdot) = \frac{1}{2} \sum_{k=1}^3 \sum_{j=1}^3 [d_j(k) - y_j^m(k)]^2$$

$$\frac{2 J(\cdot)}{2 w_{i,j}^m} = \sum_{k=1}^3 \sum_{j=1}^3 \frac{2}{2 w_{i,j}^m} [d_j(k) - y_j^m(k)]^2 = \frac{2 J}{2 w_{i,j}^m} \frac{2 w_{i,j}^m}{2 w_{i,j}^m}$$

$$\frac{2 J(\cdot)}{2 w_{i,j}^m} = \delta_i^m \frac{2 w_{i,j}^m}{2 w_{i,j}^m} \left\{ \begin{array}{l} \text{Como } w_{i,j}^m \text{ é dado pela combinação linear dos anteriores, temos:} \\ \frac{2 w_{i,j}^m}{2 w_{i,j}^m} = y_j^{m-1} \end{array} \right.$$

Portanto $\frac{2 J}{2 w_{i,j}^m} = \delta_i^m y_j^{m-1}$

Assim precisamos apenas calcular δ_i^m

$$\delta_i^m = -2(d_i - y_i^m) \frac{\partial y_i}{\partial u_i^m} = -2(d_i - y_i^m) \dot{f}^m(u_i^m)$$

De modo análogo temos:

$$\bar{\delta}^m = -2 \dot{F}^m(\bar{u}^m)(\bar{d} - \bar{y}), \text{ onde } \dot{F}^m(\bar{u}^m) = \begin{bmatrix} \dot{f}^m(u_1^m) & 0 \\ 0 & \dot{f}^m(u_n^m) \end{bmatrix}$$

A aplicação sucessiva da regra leva a:

$$\bar{\delta}^m = \dot{F}^m(\bar{u}^m)(\bar{w}^{m+1})^T \bar{\delta}^{m+1}$$

Para o exercício em questão deseja-se calcular $\frac{\partial J}{\partial w_{1,2}}$, o que na notação apresentada é dado por:

$$\frac{\partial J}{\partial w_{2,1}^{(1)}}. \text{ Esse valor é dado por: } \frac{\partial J}{\partial w_{2,1}^{(1)}} = \delta_2^{(1)} y_1$$

Temos que $\bar{y} = [y_1, y_2]$. Precisamos calcular $\bar{\delta}_2^{(1)}$.

$$\bar{\delta}^{(1)} = \dot{F}^{(1)}(\bar{u}^{(1)})(\bar{w}^{(2)})^T \bar{\delta}^{(2)}$$

$\bar{\delta}^{(2)} = -2(\bar{d} - \bar{y}) \rightarrow$ Neste caso a função de ativação da saída é 1, logo a derivada é a derivada da rede.

Definiendo $\bar{F}^{(1)}(\bar{w}^{(1)}) =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \dot{p}(u'_1) & 0 & 0 \\ 0 & 0 & \dot{p}(u'_2) & 0 \\ 0 & 0 & 0 & \dot{p}(u'_3) \end{bmatrix}$$

Así mismo tenemos:

$$\delta^{(1)} = -2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \dot{p}(u'_1) & 0 & 0 \\ 0 & 0 & \dot{p}(u'_2) & 0 \\ 0 & 0 & 0 & \dot{p}(u'_3) \end{bmatrix} \begin{bmatrix} w_{00} & w_{01} \\ w_{10} & w_{11} \\ w_{20} & w_{12} \\ w_{30} & w_{13} \end{bmatrix} \begin{bmatrix} d_1 - y_1 \\ d_2 - y_2 \end{bmatrix}$$

Por lo tanto:

$$\frac{\partial J}{\partial w_{1,2}} = \delta^{(1)} \cdot y_1$$

EFC3 - Classificação utilizando SVM e Redes MLP

Felippe Trigueiro Angelo - RA: 210479

11 de Novembro de 2019

Classificação Binária utilizando Redes MLP

No presente exercício será construído um modelo para classificação binária utilizando o algoritmo de redes neurais MLP. Para tal finalidade será utilizada a base de dados fornecida no exercício. Os conjuntos de dados de treinamento, validação e teste são compostos por 1000 pontos no R^2 e que não são linearmente separáveis. A Figura 1 mostra a distribuição dos pontos do conjunto de treinamento.

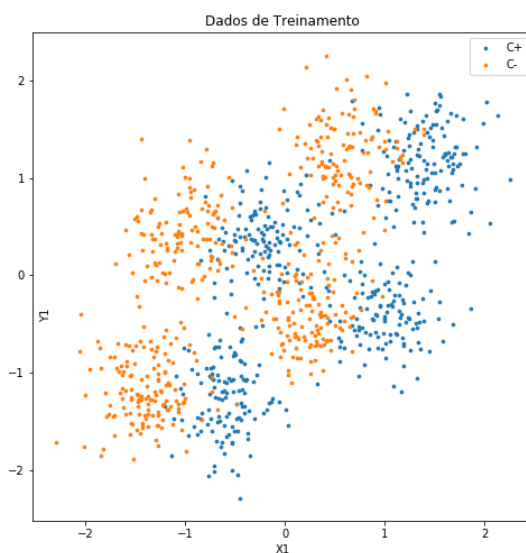


Figura 1: Conjunto de dados de treinamento.

A arquitetura da rede foi composta por apenas uma camada intermediária, onde a sua quantidade de neurônios foi obtida por meio da abordagem de validação cruzada holdout, utilizando o conjunto de dados fornecido. Para a avaliação foram

testadas as quantidades de neurônios entre 2 e 50. Como métrica para a avaliação do desempenho da rede no conjunto de validação foi utilizada a medida F1. O resultado pode ser visto na Figura 2.

É importante citar que as redes utilizadas nesse processo foram treinadas utilizando o algoritmo lgbs, função de ativação retificadora, um número máximo de 300000 iterações (300 épocas) e função de custo logarítmica. Não foi utilizada regularização nem normalização dos dados, visto que os mesmos estão em escalas similares.

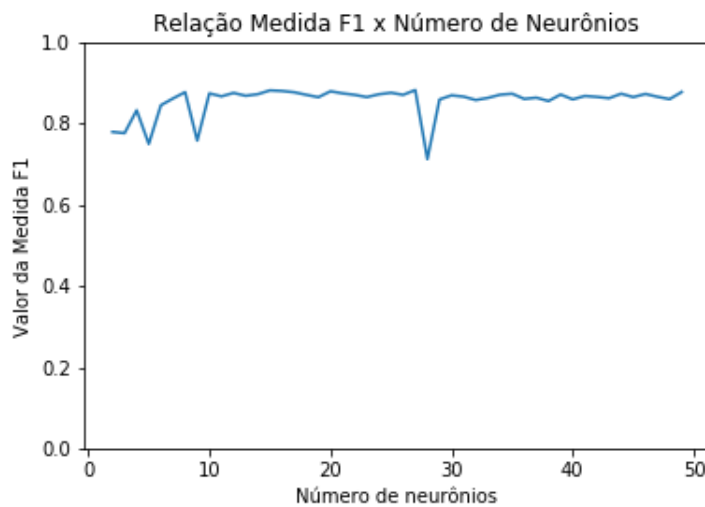


Figura 2: Avaliação da Medida F1 no conjunto de validação.

Como pôde ser observado na Figura 2, o número de neurônios que obteve o maior valor de medida F1 foi de 27, com um valor de 0.8812, apesar de que a maioria dos números de neurônios testados resultou em um valor similar da medida F1. Também foi testada a função de ativação tangente hiperbólico, que resultou em um desempenho similar a função retificadora, porém com um tempo de treinamento muito mais elevado.

Com a arquitetura da rede MLP definida, foram calculadas as curvas de aprendizado para os conjuntos de treinamento e de validação. O resultado pode ser visto na Figura 3.

É possível ver que a rede convergiu com um total de aproximadamente 1400 iterações. Ainda como pode ser visto na Figura 3, o treinamento da rede não resultou em uma grande variância, indicando ausência de overfitting, entretanto há a presença de um

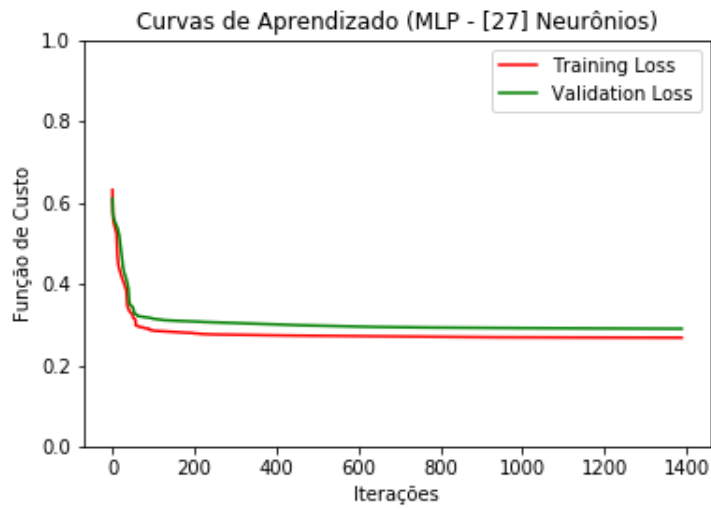


Figura 3: Curva de aprendizado da rede MLP.

bias, significativo.

Também foram calculadas as regiões de decisão obtidas pela rede. O resultado pode ser visto na Figura 4.

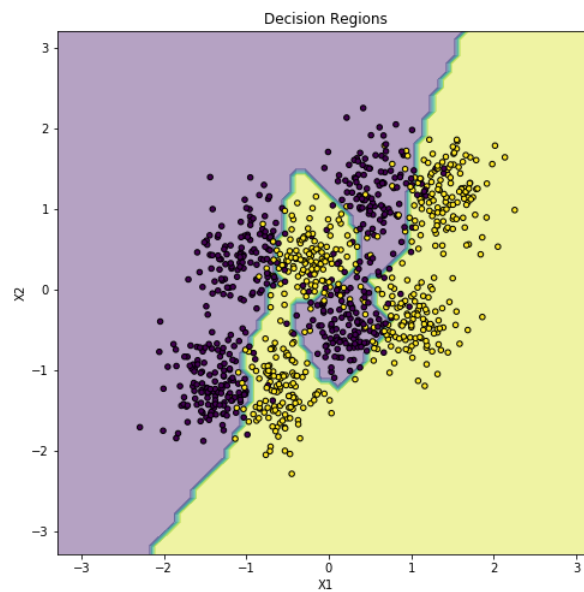


Figura 4: Regiões de decisão da rede MLP.

Por fim, a rede obtida foi aplicada no conjunto de dados de teste, obtendo uma taxa de erro de 12.3%

Classificação Binária utilizando Algoritmo SVM

Na segunda parte do exercício, os mesmos conjuntos de dados foi submetido ao algoritmo SVM (Support Vector Machine). Para avaliar qual parâmetro resulta em um modelo com maior valor de medida F1, foram testados como parâmetros de penalidade (C), os valores entre 0.1 e 10, com um passo de 0.1, nos dados de validação. Como kernel do SVM foram testadas as funções Gaussiana e Polinomial. As Figuras 5 e 6 mostram o cálculo da medida F1 em função no parâmetro C para Kernels Gaussiano e Polinomial, respectivamente.

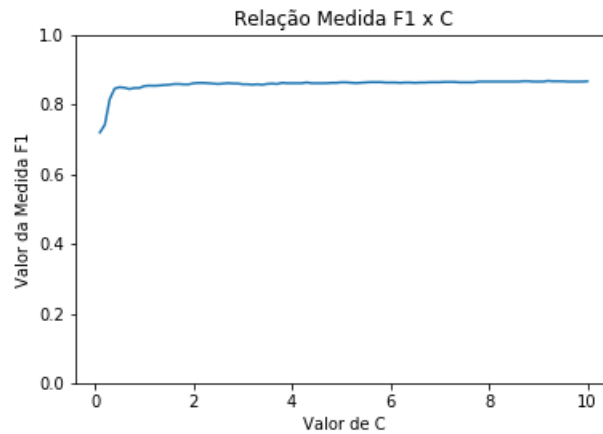


Figura 5: Análise do parâmetro de penalização C utilizando um kernel Gaussiano.

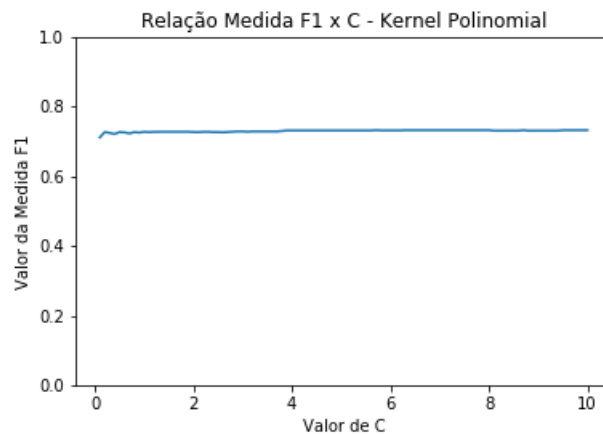


Figura 6: Análise do parâmetro de penalização C utilizando um kernel Polinomial.

Como pôde ser observado, o kernel Gaussiano obteve um melhor resultado de medida

F1, com um valor máximo de 0.867 para um valor de C igual a 9.4, contra um máximo de 0.731 para o kernel Polinomial. Ainda pode ser observado que em ambos os casos o valor de C não alterou significativamente o valor da medida F1. Da mesma forma como realizado para as redes MLP, também foram calculadas as regiões de decisão para o SVM que obteve o melhor valor de medida F1 no conjunto de validação (C=9.4 e kernel Gaussiano). Também foram calculados os vetores de suporte obtidos. Os resultados podem ser vistos nas Figuras 7 e 8.

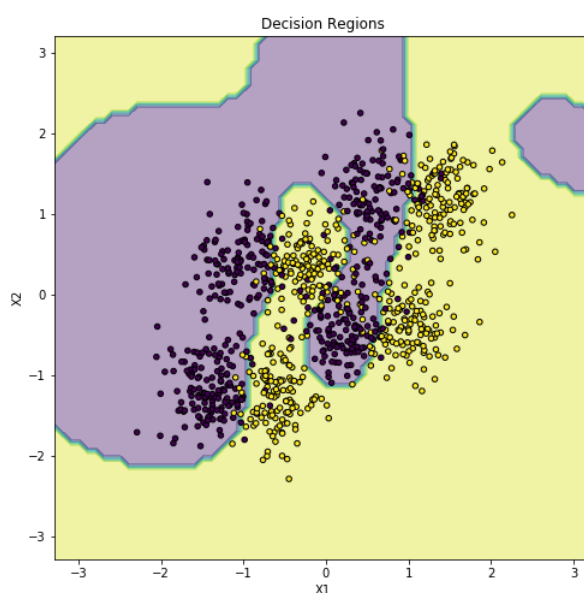


Figura 7: Regiões de decisão do algoritmo SVM para um Kernel Gaussiano.

Por fim, o SVM utilizado foi aplicado ao conjunto de dados de teste e obteve uma taxa de erro de 12.5%. Assim, pode-se concluir que com uma seleção criteriosa dos parâmetros dos modelos, os modelos de rede MLP e SVM obtiveram desempenhos similares, com a rede MLP obtendo um desempenho ligeiramente melhor.

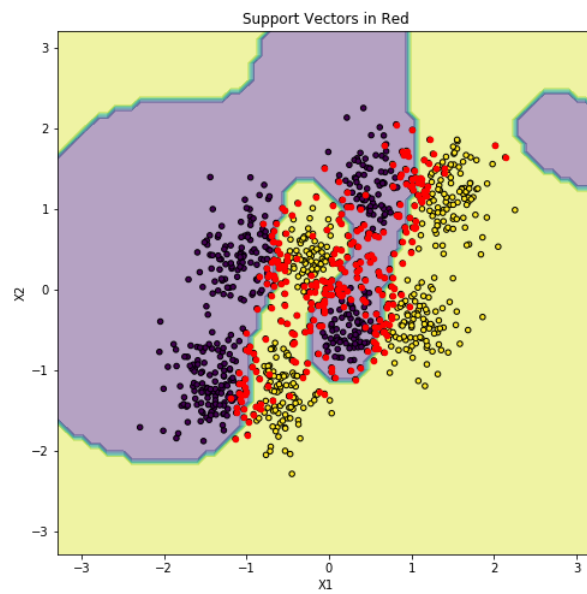


Figura 8: Vetores de Suporte.