

# IA006 – Exercícios de Fixação de Conceitos

## EFC 3 – 1s2019

### Parte I – Revisitando o algoritmo de retropropagação do erro

Considere uma rede MLP contendo duas entradas ( $x_1$  e  $x_2$ ), três neurônios na camada intermediária ( $N = 3$ ) e duas saídas ( $y_1$  e  $y_2$ ). A Figura 1 apresenta a arquitetura da rede neural.

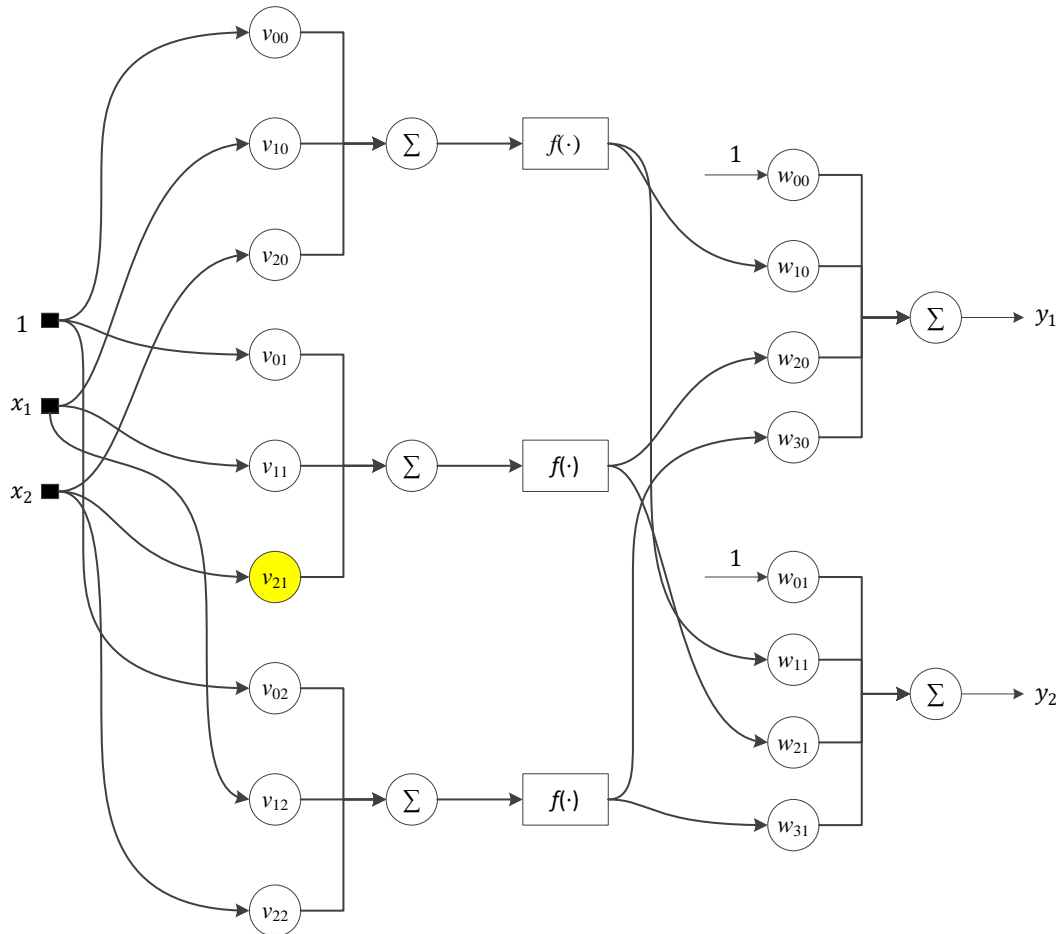


Figura 1. Arquitetura de uma rede MLP com duas entradas, três neurônios na camada intermediária e duas saídas.

Obtenha a expressão da derivada da função custo  $J(\cdot) = e_1^2 + e_2^2$  associada ao critério de erro quadrático (para uma única amostra) com respeito ao peso sináptico  $v_{21}$ , i.e.,  $\frac{\partial J}{\partial v_{21}}$ . Mostre todos os passos desta dedução, explicitando, também, o significado de eventuais variáveis intermediárias criadas para facilitar a derivação.

## Parte II – Classificação binária com redes MLP e SVMs

Considere o conjunto de dados rotulados disponível no arquivo *dados\_treinamento.mat*. A matriz  $\mathbf{X} \in \mathbb{R}^{N \times 2}$  contém os atributos (*features*) dos  $N$  padrões existentes, enquanto o vetor  $\mathbf{y} \in \mathbb{R}^{N \times 1}$  traz o rótulo da classe correspondente. No caso,

$$y(i) = \begin{cases} +1, & \text{se } \mathbf{x}(i) \in C_+ \\ -1, & \text{se } \mathbf{x}(i) \in C_- \end{cases}.$$

A Figura 2 mostra a distribuição dos dados de treinamento, na qual a cor codificada a classe correspondente a cada padrão. Além disso, apresentamos também as regiões de decisão definidas pelo classificador de mínima taxa de erro (MAP).

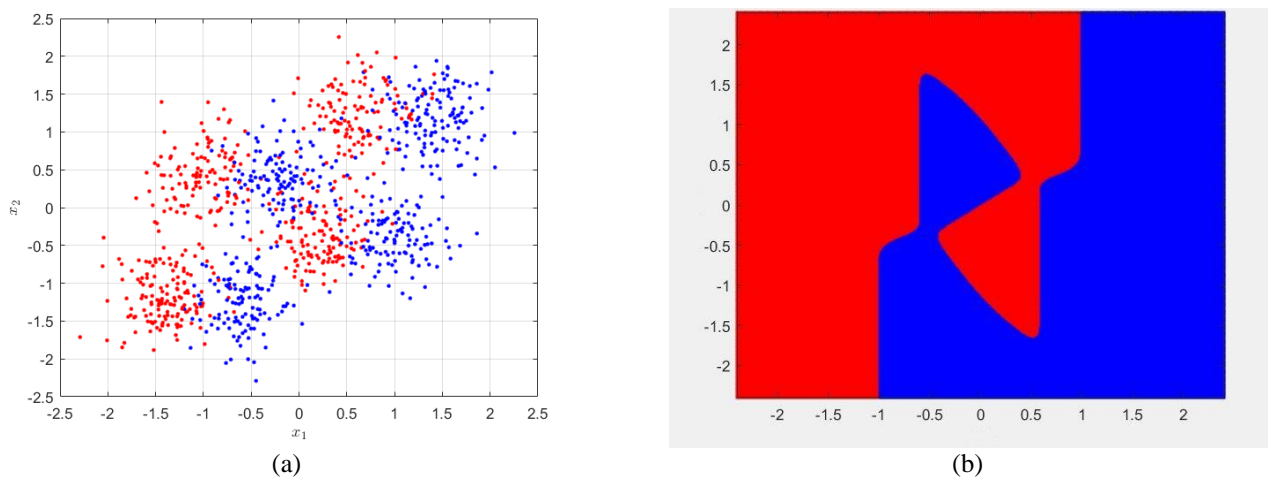


Figura 2. (a) Distribuição dos dados de treinamento: os pontos em azul pertencem à classe  $C_+$ , enquanto os pontos em vermelho pertencem à classe  $C_-$ ; (b) regiões de decisão correspondentes ao classificador ótimo (MAP).

Nesta atividade, vamos explorar dois métodos de classificação não-linear: uma rede neural MLP (*multilayer perceptron*) com uma única camada intermediária e uma máquina de vetores-suporte (SVM, do inglês *support vector machine*).

**Será permitido** o uso de pacotes (como, por exemplo, *scikit-learn* ou *Keras*) e/ou de *toolboxes* para o treinamento e análise dos classificadores.

- (a) Projete uma rede MLP utilizando um algoritmo de otimização de sua escolha para realizar a classificação. Lembre-se de empregar uma metodologia de validação cruzada (*holdout*). Sejam criteriosos na escolha dos parâmetros e justifiquem todas as opções relevantes feitas.

Analise o comportamento do algoritmo de treinamento por meio da curva de evolução do valor da função custo ao longo das épocas. Mostre também a progressão do custo em relação ao conjunto de validação. Use escala logarítmica se auxiliar na visualização. Comente os resultados.

- (b) Apresente as regiões de decisão definidas pela rede MLP no espaço dos dados de entrada. Comente.
- (c) Aplique, então, a rede projetada sobre o conjunto de teste com 1000 amostras. Mostre a saída gerada para cada entrada e calcule o percentual de erro.
- (d) Modifique o número de neurônios da camada intermediária e comente de modo geral o que foi observado, bem como seu impacto no resultado final.
- (e) Faça, agora, o projeto de uma SVM para a mesma tarefa de classificação. Mostre as regiões de decisão definidas pela SVM no espaço original dos dados e os vetores-suporte identificados.
- (f) Aplique, então, a SVM treinada sobre o conjunto de teste com 1000 amostras. Mostre a saída gerada para cada entrada e calcule o percentual de erro.
- (g) Explore algumas parametrizações diferentes da função de *kernel* e/ou do fator de regularização ( $C$ ), comentando as principais observações e seus impactos nos resultados.