

Projeto 2 – Identificação de Lixo Eletrônico (SPAM)

O termo *lixo eletrônico* ou *spam* refere-se, de um modo geral, a mensagens (e-mails) indesejadas que são enviadas constantemente nos meios eletrônicos sem o consentimento do destinatário. Nesse projeto aplicaremos os conceitos de quadrados mínimos para auxiliar na identificação automática de *spams*. Especificamente, usaremos uma técnica de aprendizado de máquina que tem recebido bastante destaque nos últimos anos chamada *extreme learning machine* (ELM). Em termos gerais, uma ELM é sintetizada e avaliada com base num conjunto de mensagens que já foram identificadas como spam ou não-spam pelo(s) usuário(s). O conjunto usado para sintetizar a ELM é chamado *conjunto de treinamento* enquanto que o conjunto usado para avaliar o desempenho do modelo é chamado *conjunto de teste*. É importante destacar que o conjunto de teste não pode ser usado em nenhum momento para sintetizar a ELM. O aluno interessado em aprendizado de máquinas e nos detalhes da ELM pode consultar [1, 2].

Condições e Datas

O projeto deve ser realizado **individualmente** ou em **dupla** utilizando GNU Octave ou MATLAB. Não será aceito trabalho feito em outra linguagem de programação.

O projeto deve ser entregue até o dia **07/06/2018**. O arquivo impresso ou digital, que não deve ter mais que 10 páginas, deve descrever de forma clara os procedimentos adotados e as conclusões. Em particular, responda as perguntas abaixo de forma objetiva e com fundamentos matemáticos. Recomenda-se que os códigos sejam anexados, mas não serão aceitos trabalhos contendo apenas os códigos! Não esqueça de incluir NOME e RA!

Instruções

O arquivo DadosTreinamento.mat, que pode ser carregado no GNU Octave ou MATLAB através do comando

```
» load DadosTreinamento.mat,
```

contém uma matriz $X_{tr} \in \mathbb{R}^{d \times m}$ e um vetor $y_{tr} \in \{-1, 1\}^m$ em que $d = 57$ e $m = 3500$. A coluna $X_{tr}(:, i)$ contém informações coletadas para identificação da i -ésima mensagem (e-mail). A componente $y_{tr}(i)$ contém o valor 1 se a i -ésima mensagem foi identificada como *spam* e -1 se foi identificada como não-*spam*. O objetivo do projeto é construir um modelo capaz de classificar uma mensagem como *spam* ou não-*spam* utilizando um conjunto de treinamento, ou seja, somente X_{tr} e y_{tr} .

Uma *extreme learning machine* (ELM) é uma rede neural artificial de múltiplas camadas [1]. Nesse projeto, vamos considerar uma rede neural muito utilizada na literatura conhecida por *perceptron de múltiplas camadas*. Resumidamente, vamos assumir que a rede neural define uma função $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$ através da equação

$$\varphi(\mathbf{x}) = \alpha_1 g_1(\mathbf{x}) + \alpha_2 g_2(\mathbf{x}) + \dots + \alpha_n g_n(\mathbf{x}) = \sum_{i=1}^n \alpha_i g_i(\mathbf{x}), \quad (1)$$

em que $\alpha_1, \alpha_2, \dots, \alpha_n$ são parâmetros e as funções g_1, g_2, \dots, g_n são dadas por

$$g_i(\mathbf{x}) = \tanh \left(\sum_{j=1}^d w_{ij} x_j + b_i \right), \quad (2)$$

em que $\mathbf{w}_i = [w_{i1}, \dots, w_{id}] \in \mathbb{R}^d$ e $b_i \in \mathbb{R}$ para todo $i = 1, \dots, n^1$. Em termos matriciais, podemos descrever a função $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$ como segue:

$$\varphi(\mathbf{x}) = \boldsymbol{\alpha}^T \tanh(\mathbf{W}\mathbf{x} + \mathbf{b}), \quad (3)$$

em que $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]^T \in \mathbb{R}^n$, $\mathbf{W} \in \mathbb{R}^{n \times d}$ é a matriz cujas linhas correspondem aos vetores \mathbf{w}_i e $\mathbf{b} = [b_1, \dots, b_n]^T \in \mathbb{R}^n$ é um vetor coluna. O código que implementa a função φ descrita pela rede neural artificial está disponível em RNA.m.

Note que o comando

```
» G = tanh(W*Xtr+b)
```

fornece uma matriz $G \in \mathbb{R}^{n \times m}$ cujo elemento $G(i, k)$ corresponde à avaliação da i -ésima função g_i calculada nos dados no vetor de características da k -ésima mensagem, isto é, $G(i, k) = g_i(\text{Xtr}(:, k))$. Além disso, o produto $\mathbf{s} = \boldsymbol{\alpha}^T G$ fornece um vetor $\mathbf{s} \in \mathbb{R}^{1 \times m}$ contendo o valor de φ calculado em cada mensagem, ou seja, $\mathbf{s} = [s_1, \dots, s_m]$ em que $s_k = \varphi(\text{Xtr}(:, k))$ para todo $k = 1, \dots, m$.

Numa ELM, os vetores $\mathbf{w}_i = [w_{i1}, \dots, w_{id}] \in \mathbb{R}^d$ e o escalar $b_i \in \mathbb{R}$ que definem a função g_i são gerados aleatoriamente utilizando uma distribuição normal padrão. Usando a forma matricial explícita em (3), no MATLAB ou GNU Octave utilizamos os comandos:

```
» W = randn(n,d);
» b = randn(n,1);
```

Os parâmetros $\alpha_1, \dots, \alpha_n$ são determinados resolvendo o problema de quadrados mínimos

$$\varphi(\text{Xtr}(:, k)) \approx \text{ytr}(k), \quad \forall k = 1, \dots, m, \quad (4)$$

definido sobre o conjunto de treinamento. Em outras palavras, $\alpha_1, \dots, \alpha_n$ minimizam a soma dos quadrados dos desvios

$$J(\alpha_1, \dots, \alpha_n) = \sum_{k=1}^m (\alpha_1 g_1(\text{Xtr}(:, k)) + \dots + \alpha_n g_n(\text{Xtr}(:, k)) - \text{ytr}(k))^2. \quad (5)$$

Finalmente, se $\mathbf{x} \in \mathbb{R}^d$ é o vetor contendo informações sobre uma mensagem, a identificação é efetuada como segue

$$\begin{cases} \text{A mensagem é um spam se } L < \varphi(\mathbf{x}), \\ \text{A mensagem não é spam caso contrário,} \end{cases} \quad (6)$$

em que $L \in \mathbb{R}$ é um limiar de decisão.

Conhecidos a função φ e o limiar L , podemos avaliar o desempenho do sistema usando um conjunto de dados que já foram identificados pelo(s) usuário(s). Por exemplo, podemos avaliar o desempenho do sistema no conjunto de teste que pode ser carregado no GNU Octave ou MATLAB através do comando

```
» load DadosTeste.mat.
```

Com esse comando, teremos uma matriz $\text{Xte} \in \mathbb{R}^{57 \times 1000}$ e um vetor $\text{yte} \in \{-1, 1\}^{1000}$, em que $\text{Xte}(:, i)$ e $\text{yte}(i)$ contém respectivamente informações sobre o conteúdo da i -ésima mensagem (e-mail). O desempenho do sistema pode ser medido quantitativamente, por exemplo, calculando a *acurácia* (AC) ou a *taxa de falsos positivos* (TFP, também chamado “taxa de alerta falso”) definidos respectivamente pelas equações:

$$\text{AC} = \frac{\text{Número de mensagens identificadas corretamente pelo sistema}}{\text{Número total de mensagens}}, \quad (7)$$

¹Observe que m refere-se ao número de dados de treinamento enquanto que n corresponde ao número de parâmetros. Nesse projeto, temos $m = 393$ e vamos considerar $n = 20$.

e

$$\text{TFP} = \frac{\text{Número de mensagens identificadas como } \textit{spam} \text{ pelo sistema mas que não são } \textit{spams}}{\text{Número de mensagens que são } \textit{spams}}. \quad (8)$$

Questões

1. Sintetize a aplicação φ resolvendo o problema de quadrados mínimos em (4) com respeito ao conjunto de treinamento considerando $n = 1000$.
2. Ainda usando o conjunto de treinamento, isto é, x_{tr} e y_{te} , determine a acurácia e a taxa de falsos positivos considerando os limiares $L = -2$, $L = 0$ e $L = 2$.
3. Interprete o limiar e comente sobre os valores da acurácia e a taxa de falsos positivos obtidos no item anterior.
4. Um falso positivo pode incorrer a perda de uma mensagem importante foi erroneamente identificada como *spam* pelo sistema. Em vista disso, determine o melhor valor para o limiar de decisão L que assegura uma taxa de falsos positivos menor que 1%. Justifique sua resposta.
5. Usando o conjunto de teste, isto é, x_{te} e y_{te} , calcule a acurácia e a taxa de falsos positivos com o limiar obtido no item anterior.
6. O desempenho no conjunto de teste é consistente com o esperado, isto é, eles são semelhantes aos valores obtidos considerando o conjunto de treinamento?

Referências

- [1] HAYKIN, S. *Neural Networks and Learning Machines*, 3rd edition ed. Prentice-Hall, Upper Saddle River, NJ, 2009.
- [2] HUANG, G.-B., WANG, D., AND LAN, Y. Extreme learning machines: a survey. *Int. J. Machine Learning & Cybernetics* 2, 2 (2011), 107–122.