

# Tarefa 3: Regressão

Resgate de Vítimas de Catástrofes Naturais, Desastres ou Grandes Acidentes

## 1 Problema

Nesta tarefa, você tem disponível um histórico de sinais que foram coletados de outros acidentes e analisados por um corpo médico:

- **qPA**: qualidade da pressão arterial; resulta da avaliação da relação entre a pressão sistólica e a diastólica;
- **pulso**: pulsação ou Batimento por Minuto (pulso);
- **frequência respiratória**: frequência da respiração por minuto;
- **gravidade**: um valor calculado em função dos sinais vitais acima
- **classes de gravidade**: são 4 classes que apresentam o estado de saúde do acidentado.

O corpo médico construiu uma fórmula para calcular a gravidade do estado de saúde das vítimas e, também, estabeleceram intervalos que definem as seguintes classes de gravidade:

- 1 = crítico,
- 2 = instável,
- 3 = potencialmente estável e
- 4 = estável.

O problema é que a fórmula de cálculo do valor de gravidade foi perdida. Portanto, você deve utilizar Redes Neurais para reconstituir o cálculo do valor da gravidade (regressão).

## 2 Metodologia

Podem ser utilizadas *Toolboxes* (e.g. MatLab) ou programação com auxílio de bibliotecas existentes (e.g. Python com SciKit, Tensorflow). O importante é entender conceitualmente os parâmetros a serem definidos/implementados (não utilizar ferramentas de maneira cega – sem entender os conceitos).

Para extrair um comportamento médio independente da escolha dos dados de treinamento/validação, **você deve fazer a validação cruzada várias vezes para cada Rede Neural a ser experimentada**. A partir daí, você seleciona o modelo que julgar ser o melhor para os testes cegos.

A fase de teste cego permite analisar a capacidade de generalização do modelo aprendido. Portanto, pode haver casos em que um modelo com bom desempenho na etapa de treinamento/validação não seja tão bom na etapa de testes, indicando adaptação excessiva aos dados de treinamento (*overfitting*).

## 2.1 Requisitos:

- **Implementar três Redes Neurais distintas** pela topologia, função de ativação e parâmetros de treinamento (e.g. taxa de aprendizado, número de épocas, atualização dos pesos por batch ou por amostra);
- **fazer comparação do RMSE** – Raiz Quadrada do Erro Quadrático Médio<sup>1</sup> - de cada Rede Neural;
- **fazer análise de generalização** do modelo escolhido para o teste cego comparando os dados de treinamento/validação com os de teste cego. A análise passa pela análise do viés e da variância.

## 3 Arquivos de treinamento e validação

Nesta tarefa, vocês terão acesso a um histórico de sinais vitais que foram coletados por um corpo médico de outros acidentes contendo os seguintes dados:

- pSist: **-- não utilizar --** pressão diastólica
- pDiast: **-- não utilizar --** pressão diastólica
- qPA: qualidade da pressão arterial; resulta da avaliação da relação entre a pressão sistólica e a diastólica;
- pulso: pulsação ou Batimento por Minuto (pulso);
- frequência respiratória: frequência da respiração por minuto;
- gravidade: valor calculado em função dos sinais vitais acima
- classes de gravidade: **-- não utilizar --** são 4 classes que apresentam o estado de saúde do acidentado.

Exemplo:

i	si1	si2	si3	si4	si5	gi	y1
	pSist	pDiast	qPA	pulso	resp	gravid	classe
1,	8.5806,	2.2791,	-8.4577,	56.8384,	9.2229,	33.5156,	2

### 3.1 Arquivo de teste cego

O *dataset* para o teste cego segue o mesmo formato dos dados de treinamento. No entanto, zeramos o  $g_i$  (o valor de gravidade) e atribuímos o valor artificial 1 à  $y_i$  (a classe). O cálculo de  $g_i$  ficará a cargo dos regressores implementados no projeto.

i	si1	si2	si3	si4	si5	gi	y1
	pSist	pDiast	qPA	pulso	resp	gravid	classe
1,	8.5806,	2.2791,	-8.4577,	56.8384,	9.2229,	0.0,	1

Para cada exemplo do teste cego, o regressor deve gerar um arquivo CSV contendo somente uma coluna com os valores de gravidade preditos.

33.5034
10.4034
...
0.0399

O professor fornecerá os valores conhecidos de gravidade e o grupo fará o cálculo do RMSE em sala de aula para podermos comparar as soluções.

$$^1 RMSE = \sqrt{\frac{1}{n} \sum_1^n (\hat{y}_i - y_i)^2}, \text{ tal que } \hat{y}_i \text{ é o alvo e } y_i, \text{ o valor predito}$$

## 4 ENTREGA

A apresentação e a entrega final se darão em dois momentos distintos:

### MOMENTO 1: Apresentação - durante a aula

As soluções devem estar implementadas e funcionando para que vocês possam:

- Executar o modelo aprendido selecionado na etapa de treinamento/validação para rodar com os dados do teste cego que serão passados pelo professor no dia da apresentação;
- comparar com as soluções de outras equipes com base nos testes realizados em sala de aula e discutir melhorias;

### MOMENTO 2: Entrega - fora do horário de aula (semana seguinte ao do momento 1)

- Corrigir/melhorar a solução implementada;
- Revisar o artigo da tarefa incluindo os resultados dos testes cegos, sem deixar de fora os resultados de treinamento/validação;
- Carregar a versão final do artigo e do código no Moodle.

#### 4.1 Artefatos da Entrega

- 1) Os códigos fonte na linguagem que desejar com as instruções para rodar.
- 2) Um artigo PDF de até 3 páginas, sem contar os apêndices, no [formato da SBC](#) com a estrutura abaixo e fonte tamanho 11.
- 3) Opcional (bonificado): vídeo de apresentação do projeto (tar1, tar2 e tar3)

**Introdução:** dentro do problema como um todo, quais subproblemas atacará e por quais razões: quais são as motivações e justificativas para resolvê-los.

**Fundamentação Teórica:** as técnicas escolhidas com uma breve descrição

**Metodologia:** descreva como procedeu para avaliar cada uma das técnicas escolhidas, salientando a análise exploratória dos dados, os pré-processamentos realizados (e.g. normalização, mudança de escala), variações de parametrização e de estrutura de cada experimento (e.g. número de camadas, taxa de aprendizado, topologia da rede neural). Explicar como procedeu os experimentos (e.g. validação cruzada com K-folds, qual o critério de parada de treinamento, qual critério de seleção do modelo a ser usado nos testes)

**Resultados e análise:** mostrar os resultados numéricos das métricas de desempenho para a etapa de treinamento/validação e para a etapa de teste cego. Fazer uma análise comparativa entre as técnicas escolhidas.

**Conclusões:** qual técnica apresentou o melhor desempenho e as razões que você crê que justificam o desempenho. Há algo a ser melhorado nas soluções apresentadas?

**Referências bibliográficas**

**Apêndice:** instruções claras de como executar o código respeitando os formatos de arquivos de entrada e de configuração do enunciado; print das telas do programa se desejar (não colocar print das telas no corpo do artigo).