



Universidade Estadual de Maringá – Centro de Tecnologia
 Depto. de Engenharia Química – Curso de Engenharia Elétrica
 6694 – Sistemas Inteligentes - 1º Sem. 2020

Avaliação 2

Data final de entrega: 14/12/2020

Grupos de até 4 pessoas (mas pode
fazer individual)

Regras importantes:

- Não será tolerado plágio. Trabalhos identificados como copiados serão penalizados.
- A data final de entrega é inadiável. Como o trabalho pode ser entregue antes da data final, não há motivos para tolerar a não entrega na data final.
- **Escolha apenas UM (1) dos projetos e resolva utilizando uma rede neural.**
- **O projeto deve ser executado utilizando alguma biblioteca livre de *machine learning* (Keras, Scikit learn, ou PyTorch) para montar, treinar, validar e testar a rede neural.**

Material a entregar

- Códigos em Python;
- Por email (para rkrummenauer2@uem.br):
 - O assunto da mensagem deve ser 1s2020:6694:<nome1>:<nome2>:<nome3>:<nome4>
 - Exemplo: 1s2020:6694:bonovox:tinaturner:fredmercury:johnlennon
 - Relatório (em formato PDF).

Projetos (escolha apenas 1 para resolver):

1. **Predição.** (O mesmo problema da parte 2 da Avaliação 1) Seja o conjunto de dados de um acelerômetro triaxial montado no peito de voluntários disponível no repositório online da Universidade da Califórnia - Irvine no endereço <https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+from+Single+Chest-Mounted+Accelerometer>. Para este projeto, a parte selecionada do conjunto de dados já foi separada e pré-processada (é o arquivo `dados_Ax.mat`), formando uma série temporal de aceleração no eixo X de um dos participantes, realizando a atividade de número 5 (subindo ou descendo escadas). Projete uma rede neural com pelo menos 2 camadas intermediárias/ocultas para fazer a predição de passo igual a 52 (52 amostras a frente) e compare o desempenho desta estrutura para: i) treinamento com dois métodos de otimização diferentes disponíveis na biblioteca de *machine learning* escolhida, fixando o tamanho do bloco de atualização dos pesos (*minibatch*) e a função de ativação do tipo tangente hiperbólica; ii) fixando um método de otimização e o tamanho do bloco (*minibatch*), compare o desempenho da rede para os casos de uso de função de ativação tipo ReLu e tipo tangente hiperbólica nas camadas intermediárias. Analise os resultados com

argumentos fundamentados. Apresente os resultados da evolução do erro quadrático médio em função da época (iteração) no treinamento e validação das redes MLP. Dicas: Um parâmetro de projeto importante é a dimensão de entrada, portanto, considere um tamanho de sequência temporal razoável (número de amostras sequenciais) para formar os padrões de entrada. A amostragem é realizada a 52 Hz. Recomenda-se utilizar ao menos 52 amostras para cada padrão (janela de tempo). Para separação dos padrões de entrada da rede, considere usar o método da janela deslizante (*sliding window*) com uma amostra de avanço para cada nova janela. Separe 60% destes padrões para treinamento, 20% para validação e 20% para teste.

2. **Regressão.** Considere o conjunto de dados [Boston Housing](#), disponível na biblioteca Keras. Esta base de dados é composta por padrões de 13 atributos relativos a características de casas em Boston na década de 1970. Os dados de saída (alvos) são os valores medianos dos preços destas casas em k\$ (milhares de dólares), caracterizando um problema de regressão. Veja, por exemplo, a solução apresentada [neste link](#), no formato .ipynb utilizando a biblioteca Keras. Esta solução do exemplo é uma solução completa e pode ser utilizada como base. Projete uma rede neural com pelo menos 2 camadas intermediárias (função de ativação não-linear a livre escolha) e compare o desempenho de uma rede treinada com todos os atributos com uma rede treinada retirando: i) os 3 primeiros atributos; ii) os 3 últimos atributos; iii) os 6 primeiros atributos; iv) os 6 últimos atributos. Agora, olhando para estes resultados, escolha 3 dos 13 atributos que considerou relevantes para treinar a rede e faça a comparação desta com a rede treinada com todos os atributos. Apresente os resultados em termos de evolução do erro de treinamento e de validação.