

Trabalho de Redes Neurais

(a nota entra no cálculo da média de exercícios/atividades, conforme plano de ensino)

Individual ou em **dupla**.

Data de entrega: 20/05/2019.

RECONHECIMENTO DE CARACTERES COM REDE NEURAL

Implementar uma rede neural Multilayer Perceptron capaz de reconhecer dígitos manuscritos. O conjunto de exemplos para treinamento está disponível no Moodle. Consiste de 1593 dígitos de 0 a 9 manuscritos, que foram escaneados em escala de cinza e ajustados em um retângulo de 16x16, totalizando 256 pixels. Cada linha no arquivo consiste do valor dos 256 pixels (normalizados entre 0 e 1), seguida pela indicação do número que os 256 pixels representam. Esta indicação é feita utilizando 10 valores. Por exemplo, uma entrada correspondente ao número 0 é indicada por [1000000000]; o número 1 por [0100000000], e assim por diante. A figura 1 apresenta as entradas para um exemplo de dígito manuscrito, com a respectiva saída (classe), que é 2.

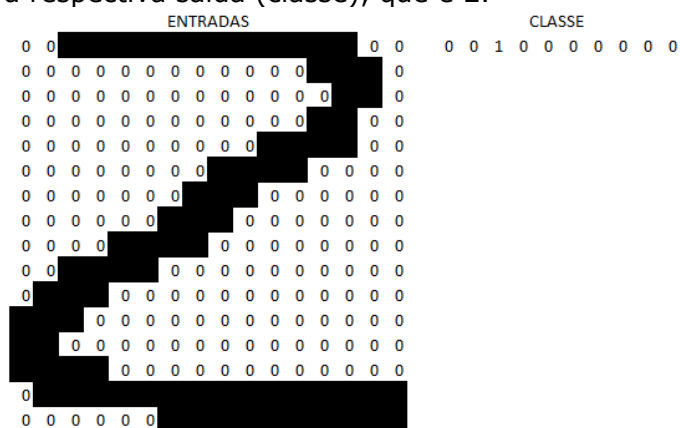


Figura 1 - Exemplo de dígito manuscrito (entrada e saída)

A rede neural deve possuir uma camada de entrada e uma camada de saída. A quantidade de camadas ocultas deve ser determinada pela equipe. Os neurônios devem utilizar a função de ativação logística. A equipe deve determinar qual é a arquitetura adequada da rede neural (entrada + ocultas + saída). Para tanto, deverá utilizar a técnica de validação cruzada, estratificando os exemplos em 5 subconjuntos com a mesma proporção de classes. Com estes conjuntos, deve-se criar 5 modelos para validação cruzada, cada modelo composto por 4 subconjuntos utilizados no treinamento, e 1 subconjunto utilizado para testar a rede e gerar a matriz de confusão. A partir da matriz de confusão, calcular a acurácia, precision, recall, e f1-score do modelo.

Para implementação da RN, sugere-se usar a biblioteca Java Encog¹ (basta baixar e importar o Jar no seu projeto). Está disponível no Moodle uma implementação exemplo de RN utilizando a Encog, para o problema do XOR (arquivo ExemploXOR.java).

Após identificar a melhor rede neural, a equipe deve construir um sistema (desktop ou mobile) que usa esta rede para reconhecer os números desenhados pelo usuário. A Encog oferece métodos para salvar/ler RNs em arquivos.²

¹ Download da Encog: <https://github.com/encog/encog-java-core/releases>

² <https://github.com/encog/encog-java-examples/tree/master/src/main/java/org/encog/examples/neural/persist>

Entregáveis (zip no Moodle)

- **Planilha PDF ou formato Excel**), com os valores das métricas de cada arquitetura de rede testada pela equipe. Destacar a arquitetura que teve o melhor desempenho, e que foi portanto selecionada para uso no sistema. Exemplo de planilha:

Arquitetura X				
<i>Modelo de Validação Cruzada</i>	<i>acurácia</i>	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
1				
2				
3				
4				
5				
Arquitetura Y				
<i>Modelo de Validação Cruzada</i>	<i>acurácia</i>	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
1				
2				
3				
4				
5				

- **Códigos fontes**, tanto da implementação do treinamento da rede, quanto do sistema que usa a rede para reconhecer os dígitos desenhados pelo usuário.
- **Arquivo README.TXT**, com instruções passo a passo de como executar o treinamento e o sistema. Ambos devem ser passíveis de execução sem ser via IDE (ex: NetBeans, Eclipse).