

# Reconhecimento de Dígitos Manuscritos (MNIST)

Introdução a Inteligência Artificial (5COP099)

**Prof. Sérgio Montazzoli Silva**

Data da entrega: 28/09/20 (Segunda)

## Como entregar

Enviar através do Google Classroom:

- código da implementação, tanto para treinamento quanto para avaliação;
- Modelos RNA e SVM já treinadas;
- Pequeno relatório com os resultados de cada classificador após o treinamento.

Os arquivos enviados ao Classroom serão os mesmos utilizados na avaliação do trabalho, portanto apenas confirme o envio quando tiver certeza que não está faltando nada.

## Descrição

O trabalho é em duplas, e o objetivo de cada dupla é construir dois classificadores distintos, neste caso uma Rede Neural Artificial (RNA) e uma Máquina de Vetores de Suporte (SVM), para resolver o problema da classificação de dígitos manuscritos.

O código enviado pela dupla deverá conter um programa onde seja possível treinar, salvar e carregar RNAs e SVMs. Não é permitida a utilização de qualquer *framework* para RNA, e os alunos deverão utilizar as implementações feitas por eles mesmos ao longo do curso. Já no caso da SVM será permitida a utilização de *frameworks*.

Junto com o trabalho está sendo fornecido uma base com 10.000 amostras. Estas amostras foram retiradas da base de dados MNIST usada no reconhecimento de dígitos manuscritos. Para mais informações, visitem <http://yann.lecun.com/exdb/mnist/>. Antes de iniciar o treinamento, a dupla deverá separar estas amostras em 3 conjuntos diferentes: treinamento (80%), validação (10%) e teste (10%).

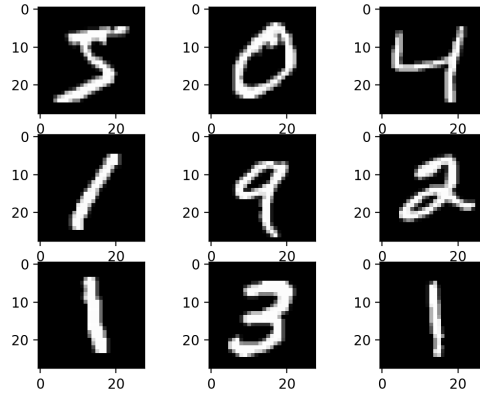
A dupla poderá utilizar qualquer arquitetura de RNA que desejar, desde que tenha **ao menos duas camadas ocultas e uma de saída**. Com relação as funções de ativação, será possível utilizar qualquer uma das que foram vistas em aula, ou alguma outra presente na literatura. Seja qual for a escolha da dupla, deverá constar no código a implementação da ativação e também da sua derivada, ambas devidamente acopladas ao algoritmo de treinamento.

Sobre o relatório, coloque os resultados de classificação tanto para a SVM quanto para a RNA treinada. Em ambos os casos, dê informações de parametrização, como número de camadas, número de neurônios por camada, taxa de aprendizado utilizada, número de épocas. No caso da SVM, qual foi parâmetro C utilizado e o parâmetro  $\gamma$  do kernel radial, também informe a quantidade de vetores de suporte encontrados.

## Dataset MNIST

Cada linha do arquivo “*data.csv*”, em anexo, corresponde a um vetor de tamanho 784 contendo valores reais no intervalo  $[0, 1]$ , e uma *string* ao final indicando a classe de saída. Os 784 primeiros valores correspondem aos *pixels* de uma imagem  $28 \times 28$ , monocromática, de um dígito, como os exemplos da imagem abaixo. Valores próximos de 1 correspondem a cor branca, e valores próximos de 0 correspondem a cor preta.

O elemento 785 (último) é uma string contendo “zero”, “um”, “dois”, “tres”, “quatro”, “cinco”, “seis”, “sete”, “oito” ou “nove”. Assim como na base Íris, estes valores devem ser convertidos em um vetor binário.



## Dicas de Implementação

A página oficial do MNIST informa que um classificador linear simples irá obter no máximo 88% de acurácia. Portanto, esta é uma base de dados com grande não-linearidade, logo devemos explorar o uso de múltiplas camadas (3 a 10). Além disso, o número de neurônios por camada também não deve ser pequeno, comece em 20. Tenha paciência, o treinamento pode demorar para convergir.

Caso a dupla ainda não tenha concluído a implementação na base de dados Íris, aconselho que finalize naquela base, pois é menor e mais simples, facilitando a correção de possíveis erros. Quando tudo estiver certo nela, então migre para o MNIST.

Na SVM, utilize o kernel radial e trabalhe o seu valor  $\gamma$ . Bibliotecas como o OpenCV e o Scikit, no Python, já possuem implementações da SVM e podem ser prontamente utilizadas.

Mãos à obra!