

Treinamento Multi-camadas
Introdução a Inteligência Artificial (5COP099)
Prof. Sérgio Montazzolli Silva
Data da entrega: 15/11/19 (Sexta)

Como entregar

Enviar arquivo *zip* contendo:

- código de implementação para avaliação e execução;
- RNA já treinada para o *dataset* do trabalho.

Descrição

O trabalho é em duplas, e o objetivo de cada dupla é construir um programa onde seja possível treinar, salvar e carregar redes neurais multi-camadas. Não é permitida a utilização de qualquer framework que realize alguma uma destas operações.

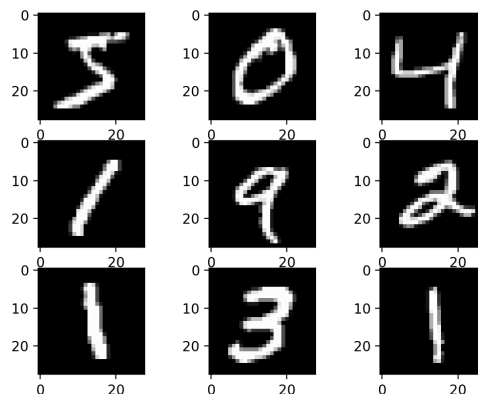
O formato de arquivo para salvar e carregar uma RNA foi definido na atividade prática anterior. Siga aquele modelo, pois ele será utilizado pelo professor no momento da avaliação.

Junto com o trabalho está sendo fornecido um *dataset* contendo 10.000 amostras de 784 dimensões cada, e 10 valores de saída. Estas amostras foram retiradas do *dataset* MNIST para reconhecimento de dígitos manuscritos.

A dupla pode utilizar qualquer arquitetura de rede, desde que tenha **ao menos duas camadas ocultas e uma de saída**. Com relação as funções de ativação, é 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.

Dataset MNIST

Cada linha do arquivo CSV de treinamento corresponde a um vetor de tamanho 784, e 10 valores de saída. Os 784 valores correspondem aos pixels de uma imagem 28×28 , monocromática, de um dígito, como os exemplos na imagem abaixo.



Os 10 valores de saída correspondem aos dígitos de 0 a 9. Logo, em uma entrada contendo o dígito 7, o vetor de saída será

$$\vec{d} = [0, 0, 0, 0, 0, 0, 0, 1, 0, 0].$$

Dicas

Este é um *dataset* com grande não-linearidade, logo deve-se explorar o uso de múltiplas camadas (entre 3 e 10). Além disso, o número de neurônios por camada também não deve ser pequeno (como 2 ou 3), comece em 20.

O treinamento pode demorar para convergir. No algoritmo que foi passado em sala de aula ele pode levar até 300.000 iterações. Portanto, tenha paciência e desenvolva seu código de maneira otimizada. Tente fazer uso de bibliotecas que implementem operações matriciais (principalmente multiplicação e operações ponto-a-ponto) pois são extremamente rápidas.

Durante o desenvolvimento do seu código, utilize os datasets da atividade de redes multi-camadas. Eles são mais simples e fáceis de notar a convergência da rede. Uma vez que seu código esteja funcionando para eles, então migre para o MNIST.