

CEFET/RJ
Programa de Pós-graduação
em Ciência da Computação
Aprendizado de Máquina - Trabalho 04

Prof. Eduardo Bezerra (ebezerra@cefet-rj.br)

Agosto/2018

Conteúdo

1	Redes Completamente Conectadas	3
2	Redes Convolucionais	4
3	Entrega	5
4	Créditos	5

1 Redes Completamente Conectadas

Nessa primeira parte, você deve usar a biblioteca `scikit-learn` para treinar um modelo de rede neural para a tarefa de classificação binária. Um bom ponto de partida é estudar a documentação dessa biblioteca intitulada `Neural network models (supervised)`¹.

Uma instituição financeira possui uma base de dados com o histórico de crediário oferecido aos seus clientes. Baseado neste histórico, a instituição deseja criar um modelo de classificação para inferir se um novo cliente que submeteu uma requisição de empréstimo pagará ou não a dívida, caso o banco resolva realizar esse empréstimo. O conjunto de dados possui 1500 exemplos de créditos concedidos aos seus clientes. Esses registros estão contidos no arquivo `credtrain.txt`, que é fornecido juntamente com esse documento. Para cada cliente, são definidos 11 atributos (variáveis, características). Além disso, a última coluna de cada exemplo informa se o cliente honrou ou não o pagamento do empréstimo. Na Tabela 1, encontramos a descrição dos atributos.

Tabela 1: Esquema do conjunto de dados com histórico de clientes.

Variável	Descrição	Tipo	Domínio
ESCT	Estado civil	Catégorica	0,1,2,3
NDEP	Número de dependentes	Catégorica	0,1,2,3,4,5,6,7
RENDA	Renda Familiar	Numérica	300-9675
TIPOR	Tipo de residência	Catégorica	0,1
VBEM	Valor do bem a ser adquirido	Numérica	300-6000
NPARC	Número de parcelas	Numérica	1-24
VPARC	Valor da parcela	Numérica	50-719
TEL	Se o cliente possui telefone	Catégorica	0,1
IDADE	Idade do cliente	Numérica	18-70
RESMS	Tempo de moradia (em meses)	Numérica	0-420
ENTRADA	Valor da entrada	Numérica	0-1300
CLASSE	=1 se o cliente pagou a dívida	Catégorica	0,1

Você deve criar um modelo de classificação, por meio de uma rede neural MLP (*multi-layer perceptron*) com propagação do erro (*error backpropagation*). O objetivo desse modelo de classificação é predizer se um novo cliente pagaria ou não uma dívida contraída, tendo como base as características desse novo cliente. Uma vez treinada, essa rede poderá inferir se um novo cliente irá ou não honrar um eventual empréstimo concedido.

Inicialmente, você deve carregar o arquivo `credtrain.txt`. Uma vez que os dados estão carregados, você deve criar duas matrizes. A primeira matriz deve ter ordem 1500×11 e deve conter os padrões de entrada (valores das variáveis `ESTC`, `NDEP`, `RENDA`, `TIPOR`, `VBEM`, `NPARC`, `VPARC`, `TEL`, `IDADE`, `RESMS` e `ENTRADA`). A outra matriz deve ter ordem 1500×1 e deve conter os valores do atributo `CLASSE`. Vamos chamar essas duas matrizes

¹http://scikit-learn.org/stable/modules/neural_networks_supervised.html

de X e y , respectivamente. X é chamada a matriz de treinamento, enquanto que y é chamada a matriz alvo. Um cuidado que você deve ter é com relação à normalização dos dados de entrada da rede, posto que algumas variáveis no conjunto de dados fornecido possuem ordens de grandeza diferentes.

Uma vez construídas as matrizes X e y , é possível criar, treinar e testar uma rede MLP. Você deve selecionar os hiperparâmetros. Com dica, dada a baixa complexidade do problema, você pode usar uma rede MLP de uma única camada oculta. Em seu relatório, apresente os detalhes acerca dos hiperparâmetros selecionados.

Um detalhe importante é que, em vez de testar a rede sobre os padrões utilizados durante o treinamento, você deve testá-la sobre os dados contidos no arquivo `credtest.txt`, que também é fornecido. Isso permitirá que você avalie o quão efetivo foi o passo de treinamento da rede neural, ou seja, o quão adequado é o modelo de classificação. Outro detalhe importante é que você deve experimentar diferentes quantidades de neurônios na camada oculta para gerar uma boa taxa de acerto.

Crie uma função para produzir a *matriz de confusão* (do termo original *confusion matrix*; veja maiores detalhes em http://en.wikipedia.org/wiki/Confusion_matrix) relativa aos resultados da fase de testes. Uma das entradas dessa matriz é o valor da taxa de acerto da rede durante a fase de teste, ou seja, a porcentagem de registros do conjunto de padrões de teste (`credtest`) para os quais a rede acertou a resposta correta. Em http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html, você encontra um exemplo de como gerar uma matriz de confusão.

2 Redes Convolucionais

Nesta parte, você irá treinar modelos de redes neurais artificiais para classificar imagens. Em particular, dada uma imagem, seus modelos deverão indicar se a imagem contém um gato ou não. São fornecidos os conjuntos de imagens para treinamento e para teste (`train_catvnoncat.h5` e `test_catvnoncat.h5`). É também fornecido o código para a leitura dos arquivos correspondentes (veja listagem a seguir; veja também a seção de créditos no fim deste documento).

```
1 import numpy as np
2 import h5py
3
4 def load_dataset():
5     train_dataset = h5py.File('./train_catvnoncat.h5', "r")
6     train_set_x_orig = np.array(train_dataset["train_set_x"][:]) #
7     train_set_y_orig = np.array(train_dataset["train_set_y"][:]) #
8     # your train set labels
9
10    test_dataset = h5py.File('./test_catvnoncat.h5', "r")
11    test_set_x_orig = np.array(test_dataset["test_set_x"][:]) #
12    test_set_y_orig = np.array(test_dataset["test_set_y"][:]) #
13    # your test set labels
```

```

13     classes = np.array(test_dataset["list_classes"][:]) # the list
      of classes
14
15     train_set_y_orig = train_set_y_orig.reshape((1,
      train_set_y_orig.shape[0]))
16     test_set_y_orig = test_set_y_orig.reshape((1, test_set_y_orig.
      shape[0]))
17
18     return train_set_x_orig, train_set_y_orig, test_set_x_orig,
      test_set_y_orig, classes

```

Após ler os conjuntos de dados, você deve treinar dois modelos utilizando o **TensorFlow**, conforme descrito a seguir:

- Uma rede completamente conectada de uma única camada oculta e com uma camada de saída de duas unidades com softmax.
- Uma rede convolucional.

Em ambos os casos acima, você deverá selecionar os hiperparâmetros e arquitetura de rede. Procure se basear nos exemplos de código e nas arquiteturas de rede apresentadas em aula. Em seu relatório, apresente os detalhes acerca dos hiperparâmetros selecionados para cada uma dessas redes. Apresente também detalhes acerca do desempenho (acurácia) encontrado em cada um dos casos.

3 Entrega

Você deve preparar um único relatório para a apresentar sua análise e conclusões sobre as diversas partes desse trabalho. O formato desse relatório deve ser em PDF. Alternativamente à entrega do relatório em PDF, você pode entregar um notebook Jupyter².

Independente de escolher entregar um relatório em PDF ou na forma de um notebook Jupyter, entregue também todos os arquivos em Python que você criou para cada parte deste trabalho. Todos os arquivos em Python devem estar em uma única pasta.

Crie um arquivo compactado que contém o relatório (ou notebook Jupyter) e os arquivos (*scripts*) em Python. Esse arquivo compactado deve se chamar **SEU_NOME_COMPLETO_T3.zip**. Esse arquivo compactado deve ser entregue pelo Moodle, até a data acordada.

4 Créditos

A segunda parte desse trabalho usa conjuntos de dados disponibilizados no curso *Neural Networks and Deep Learning*³ encontrado no Coursera. O material original é de autoria do prof. Andrew Ng e de seus colaboradores.

²<http://jupyter.org/>

³<https://www.coursera.org/learn/neural-networks-deep-learning/home/welcome>