Adult Census Data

September 5, 2017

1 Tarea 1 de Redes Neuronales

1.1 Tests unitarios

Para correr los test de la tarea se ejecuta el siguiente codigo estando en la carpeta de la tarea:

```
python test_network.py
```

1.2 Dataset de censo para adultos

Se probara la tarea utilizando un dataset con campos recolectados de varios censos en el mundo que trata de adivinar si el sueldo de una persona supera los 50 mil dolares el año

```
In [14]: import pandas as pd
         import numpy as np
         from sklearn import preprocessing

         adult_data = pd.read_csv('./data/adult.data')
         adult_data.head(5)
```

Out[14]:

	age	workclass	fnlwgt	education	education-num	\
0	39	State-gov	77516	Bachelors	13	
1	50	Self-emp-not-inc	83311	Bachelors	13	
2	38	Private	215646	HS-grad	9	
3	53	Private	234721	11th	7	
4	28	Private	338409	Bachelors	13	

	marital-status	occupation	relationship	race	sex	\
0	Never-married	Adm-clerical	Not-in-family	White	Male	
1	Married-civ-spouse	Exec-managerial	Husband	White	Male	
2	Divorced	Handlers-cleaners	Not-in-family	White	Male	
3	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	
4	Married-civ-spouse	Prof-specialty	Wife	Black	Female	

	capital-gain	capital-loss	hours-per-week	native-country	target
0	2174	0	40	United-States	<=50K
1	0	0	13	United-States	<=50K
2	0	0	40	United-States	<=50K

```
3             0             0             40  United-States  <=50K
4             0             0             40           Cuba  <=50K
```

Se debe convertir la data a numeros entre 0.0 y 1.0 para poder utilizarla en la red neuronal

```
In [7]: dics = []
        maxN = []
        for idcol, col in enumerate(adult_data.columns):
            dic = {}
            if adult_data.dtypes[idcol] == np.dtype('O'):
                ro = np.unique([str(x) for x in adult_data[col].values])
                for i, val in enumerate(ro):
                    dic[val] = i
                maxN.append(ro.size)
            else:
                maxN.append(0)

            dics.append(dic)

In [8]: M = adult_data.values
        for i, arr in enumerate(M):
            for j, val in enumerate(arr):
                if val.__class__ == ' '.__class__:
                    M[i][j] = dics[j][M[i][j]]

        M

Out[8]: array([[39, 7, 77516, ..., 40, 39, 0],
               [50, 6, 83311, ..., 13, 39, 0],
               [38, 4, 215646, ..., 40, 39, 0],
               ...,
               [58, 4, 151910, ..., 40, 39, 0],
               [22, 4, 201490, ..., 20, 39, 0],
               [52, 5, 287927, ..., 40, 39, 1]], dtype=object)

In [9]: min_max_scaler = preprocessing.MinMaxScaler()
        M_scaled = min_max_scaler.fit_transform(np.array(M))
        M_scaled

/home/kaminari/anaconda3/lib/python3.6/site-packages/sklearn/utils/validation.py:429: DataConve
  warnings.warn(msg, _DataConversionWarning)


Out[9]: array([[ 0.30136986,  0.875     ,  0.0443019 , ...,  0.39795918,
                 0.95121951,  0.        ],
               [ 0.45205479,  0.75      ,  0.0482376 , ...,  0.12244898,
                 0.95121951,  0.        ],
               [ 0.28767123,  0.5       ,  0.13811345, ...,  0.39795918,
                 0.95121951,  0.        ],
```

```
       ...,
       [ 0.56164384,  0.5       ,  0.09482688, ...,  0.39795918,
         0.95121951,  0.        ],
       [ 0.06849315,  0.5       ,  0.12849934, ...,  0.19387755,
         0.95121951,  0.        ],
       [ 0.47945205,  0.625     ,  0.18720338, ...,  0.39795918,
         0.95121951,  1.        ]])
```

Con la data convertida a numero se procede a crear la red y entrenarla con 30000 datos

```
In [15]: # train network
         from neuron_network import NeuronNetwork

         network = NeuronNetwork(14, 0.5)
         network.add_layer(100)
         network.add_layer(50)
         network.add_layer(10)
         network.add_layer(20)
         network.add_layer(1)

         for tr in range(30000):
             expected = M_scaled[tr][14]

             inputs = np.array(M_scaled[tr][0:14])
             output = network.feed(inputs)[0]
             network.backpropagate_error(expected)
             network.update_weights()
```

y se ocupan un poco mas de 2500 datos para probar el resultado

```
In [210]: sumTrue = 0
          errors = np.array([])
          for i in range(32000,32561):
              inputs = np.array(M_scaled[i][0:14])
              output = network.feed(inputs)

              raw_output = network.feed(inputs)[0]
              output = 1.0 if raw_output > 0.5 else 0.0
              expected = M_scaled[i][14]

              errors = np.append(errors, expected - raw_output)
          #     print(output, expected)
              if output == expected:
                  sumTrue += 1

          #     if expected == 1:
          #         print(i)

          precision = sumTrue / (32561-32000)
```

```
        promError = np.mean(np.abs(errors))
        print('prec: ', str(precision), 'error: ', str(promError))
```

```
prec:  0.7629233511586453 error:  0.303098475234
```

Los resultados de precisión y error medio se muestran arriba. Estos resultados no son tan buenos, pero se espera mejorar agregando capas intermedias.

2 Preguntas

2.1 How does the number of hidden layers impact the learning rate?

A medida que que se tienen mas capas la red se hace mas especifica para un grupo de inputs. Mas capas debiera significar que la red se especifica mejor en el entrenamiento, pero se demorara mas en entrenar.

2.2 What is the speed of your network to process data ?

Depende del número de hidden layers, con la red que se presenta se demora alrededor de 3 minutos en entrenar 30000 datos.

2.3 Effect of different learning rates

En los test (AND, OR, XOR) se pudo ratificar que en general se llegaba más rápido a la convergencia con learning rates altos (alrededor de 0.5), con learning rates bajos se demoraba mucho en llegar a la convergencia.

2.4 Does the order of the training data matter?

Si importa, si se entrena mucho tiempo solo con data de un tipo, la red adquiere cierta arquitectura (preferencia por algun dato) que luego cuesta cambiar.

2.5 What are the neurons that changes the most during the learning

Las neuronas que más cambian son las mas cercanas a la salida. Las de mas atras cambian menos.

```
In [ ]:
```