Sparse Autoencoder

October 16, 2018

1 Importing the packages

```
In [3]: import numpy as np
        import matplotlib.pyplot as plt
        from scipy.misc import imread , imresize
        import os
        import time
```

2 Parameters

```
In [4]: hidden_nodes = 1024
        lr = 1e-3
        lr_kl = 0.1
        epochs = 300
        sparsity = 0.2
```

3 Declaring functions

```
In [5]: # Sigmoid

        def sigmoid(x):
            return 1/(1 + np.exp(-x))

        #derivative of sigmoid
        def deriv_sigmoid(x):
            return np.exp(-x)/((1+ np.multiply(np.exp(-x),np.exp(-x))))

        # softmax

        def softmax(x):
            out = np.zeros(x.shape)
            for i in range(0,x.shape[0]):
                for j in range(0,x.shape[1]):
                    out[i,j] = np.exp(x[i,j])/np.sum(np.exp(x[i]))
            return out
```

```
        # sum of Squared error

        def squared_error(y_train, y_predicted):
            return np.sum(np.multiply(y_train - y_predicted , y_train - y_predicted))
```

```
In [6]: def load_flattened_images(Loc):
            Images = []
            for root, dirs, files in os.walk(Loc):
                for file in files:
                    Image = imread(os.path.join(root, file))
        #             Image = imresize(Image, (14,14))
                    Image = np.round(Image / 255.0)
        #             Image = Image/255.0
                    Images.append(Image.flatten())

            Images = np.asmatrix(Images)
            print(Images.shape)

            return Images
```

```
In [7]: ## fitting the model

        def net_fit(x_train , y_train , epochs = 100 , hidden_nodes = 2 , lr = 1e-3 , lr_kl = 1e
            input_dim = x_train.shape[1]
            training_samples = x_train.shape[0]
            output_dim = y_train.shape[1]
            costs = []
            z_means = []
            x_train = np.hstack((np.ones((training_samples , 1)), x_train))
            #initializig the parameters
            alpha = np.asmatrix(np.random.normal(0,1,(input_dim + 1 , hidden_nodes)))
            beta = np.asmatrix(np.random.normal(0,1,(hidden_nodes+1 , output_dim)))

            #looping for number of itretions
            for epoch in range(0,epochs):
                #finding z matrix
                z_raw = x_train * alpha
                z = sigmoid(z_raw)
                z_biased = np.asmatrix(np.hstack((np.ones((training_samples,1)),z)))

                #finding y matrix
                y_raw = z_biased * beta
                y_predicted = sigmoid(y_raw)

                ##finding the cost
                cost = squared_error(y_train , y_predicted) + lr_kl * np.sum((sparsity*np.log(sp
                costs.append(cost)
                z_means.append(np.mean(z))
```

```
            #finding gradient w.r.t beta
            delta = np.multiply((y_predicted - y_train), deriv_sigmoid(y_raw))
            d_beta = z_biased.T * delta

            temp_beta = beta[1:,:]

            #finding gradient w.r.t alpha
            ss = np.multiply((delta * temp_beta.T),deriv_sigmoid(z_raw))
            d_alpha_raw = x_train.T * ss
    #         kl divergence derivative
            d_kl = t = np.repeat(((-sparsity/np.mean(z , axis = 0)) + ((1-sparsity)/(1-np.me
            d_alpha_kl = (1/training_samples)*(x_train.T * d_kl)
            d_alpha = d_alpha_raw + lr_kl*d_alpha_kl
    #         print(np.max(d_alpha) ,np.max(d_alpha_raw) ,np.max(d_kl) , np.max(d_beta) )

            #updating the weights
            beta = beta - lr * d_beta
            alpha = alpha - lr*d_alpha
    #         print(np.max(alpha) , np.max(beta) , np.min(alpha) , np.min(beta))
            print("\nEpoch: " + str(epoch+1) + "   cost : " + str(cost))
        return alpha , beta , costs , z_means
```

In [8]:
```
#prediction

def net_predict(x_test , alpha , beta ):
    testing_samples = x_test.shape[0]
    #adding bias
    x_test = np.hstack((np.ones((testing_samples , 1)), x_test))

    #finding z matrix
    z_raw = x_test * alpha
    z = sigmoid(z_raw)
    z_biased = np.asmatrix(np.hstack((np.ones((testing_samples,1)),z)))

    #finding Y matrix (predicting the outputs)
    y_raw = z_biased * beta
    y_predicted = sigmoid(y_raw)
    y_predicted = np.round(y_predicted)   ##comment it if solving for regression
    return y_predicted
```

4 Generating training data

In [9]:
```
x_train = np.load("train_set_10000.npy")
print(x_train.shape)
```

```
(10000, 784)
```

5 Training the Model

```
In [8]: # alpha , beta , losses = net_fit(x_Train , y_train_and , hidden_nodes = hidden_nodes ,
        tic = time.time()
        alpha , beta , losses , means = net_fit(x_train , x_train , hidden_nodes = hidden_nodes
        print("time taken: "+ str(time.time() - tic) + "sec")
        # print("\nalpha:\n",alpha ,"\nbeta:\n", beta,"\n" ,"\nloss:\n", losses[epochs-1])
        np.save("alpha_weights_sae_v3.npy" , alpha)
        np.save("beta_weights_sae_v3.npy" , beta)
```

```
Epoch: 1   cost : 3749975.1240937985

Epoch: 2   cost : 1732414.2990460221

Epoch: 3   cost : 1387607.0583543826

Epoch: 4   cost : 1283996.197665262

Epoch: 5   cost : 1239307.9386880898

Epoch: 6   cost : 1197261.482165411

Epoch: 7   cost : 1164719.6726035767

Epoch: 8   cost : 1146539.966538277

Epoch: 9   cost : 1139502.8990229573

Epoch: 10   cost : 1142009.3825962397

Epoch: 11   cost : 1141286.6243754707

Epoch: 12   cost : 1108461.6453047248

Epoch: 13   cost : 1097915.8491551585

Epoch: 14   cost : 1085768.9716136083

Epoch: 15   cost : 1070094.3133000263

Epoch: 16   cost : 1046455.7990944402

Epoch: 17   cost : 1030148.5720348607

Epoch: 18   cost : 1010312.5656559073

Epoch: 19   cost : 994362.5219796746
```

```
Epoch: 20   cost : 975356.2583921864

Epoch: 21   cost : 938314.3045812914

Epoch: 22   cost : 927365.1683781347

Epoch: 23   cost : 917289.7880284699

Epoch: 24   cost : 891215.6263387204

Epoch: 25   cost : 865137.5550137769

Epoch: 26   cost : 858508.4019092204

Epoch: 27   cost : 830876.2366430716

Epoch: 28   cost : 815709.4294272307

Epoch: 29   cost : 810121.4055781753

Epoch: 30   cost : 800199.6608683061

Epoch: 31   cost : 792417.2518219986

Epoch: 32   cost : 776061.6766052113

Epoch: 33   cost : 765083.530288829

Epoch: 34   cost : 760693.3794570806

Epoch: 35   cost : 761629.3863380187

Epoch: 36   cost : 739029.5621829767

Epoch: 37   cost : 729497.0441153793

Epoch: 38   cost : 706393.6991368975

Epoch: 39   cost : 704108.8997414978

Epoch: 40   cost : 693482.1936593235

Epoch: 41   cost : 691920.4867269376

Epoch: 42   cost : 676795.6029210684

Epoch: 43   cost : 680343.2596097215
```

```
Epoch: 44   cost : 675382.6415488406

Epoch: 45   cost : 674873.8974000525

Epoch: 46   cost : 667547.7135376617

Epoch: 47   cost : 663786.1501605045

Epoch: 48   cost : 660311.8628949733

Epoch: 49   cost : 656562.0429068748

Epoch: 50   cost : 649681.4822987666

Epoch: 51   cost : 633027.7583507579

Epoch: 52   cost : 629662.2250502815

Epoch: 53   cost : 625719.7014135589

Epoch: 54   cost : 626240.5671116326

Epoch: 55   cost : 619117.6723565693

Epoch: 56   cost : 606110.6822688577

Epoch: 57   cost : 598602.1084752582

Epoch: 58   cost : 592072.8558346637

Epoch: 59   cost : 586689.8352693273

Epoch: 60   cost : 585600.4104318816

Epoch: 61   cost : 582990.5940398587

Epoch: 62   cost : 581116.5634495892

Epoch: 63   cost : 570548.2705493887

Epoch: 64   cost : 570956.7105048837

Epoch: 65   cost : 574203.3765409056

Epoch: 66   cost : 571951.0198967515

Epoch: 67   cost : 563483.5543043704
```

```
Epoch: 68   cost : 559920.4510564547

Epoch: 69   cost : 555100.2840461525

Epoch: 70   cost : 549548.4325397179

Epoch: 71   cost : 538048.4374335175

Epoch: 72   cost : 534754.5023304063

Epoch: 73   cost : 529521.6332457913

Epoch: 74   cost : 529753.1104945395

Epoch: 75   cost : 526228.1335334568

Epoch: 76   cost : 528649.5076809692

Epoch: 77   cost : 523587.38180174056

Epoch: 78   cost : 521621.63215940096

Epoch: 79   cost : 515972.48440200055

Epoch: 80   cost : 517799.37144190917

Epoch: 81   cost : 512893.18041108514

Epoch: 82   cost : 516833.3665475686

Epoch: 83   cost : 511467.4122047914

Epoch: 84   cost : 521699.6008345568

Epoch: 85   cost : 513647.5639273201

Epoch: 86   cost : 515526.32503008505

Epoch: 87   cost : 501356.4368025978

Epoch: 88   cost : 508835.47595739155

Epoch: 89   cost : 500323.3476376421

Epoch: 90   cost : 499715.4221336093

Epoch: 91   cost : 488163.47981035075
```

```
Epoch: 92   cost : 482729.8840863258

Epoch: 93   cost : 482840.9910719247

Epoch: 94   cost : 484134.19999468786

Epoch: 95   cost : 482946.463323736

Epoch: 96   cost : 484033.61451851454

Epoch: 97   cost : 478637.1717197148

Epoch: 98   cost : 473322.4689569891

Epoch: 99   cost : 464820.8206246477

Epoch: 100   cost : 462015.975360163

Epoch: 101   cost : 455399.03429644904

Epoch: 102   cost : 455357.242158305

Epoch: 103   cost : 451019.99887153273

Epoch: 104   cost : 453059.2363765482

Epoch: 105   cost : 444498.0590964005

Epoch: 106   cost : 451272.764785331

Epoch: 107   cost : 446488.70675468177

Epoch: 108   cost : 453676.23163356114

Epoch: 109   cost : 452450.4948708639

Epoch: 110   cost : 449177.50445274776

Epoch: 111   cost : 437667.2922065941

Epoch: 112   cost : 432865.1130873623

Epoch: 113   cost : 425347.8862530527

Epoch: 114   cost : 423671.55531791353

Epoch: 115   cost : 420039.80736868695
```

```
Epoch: 116   cost : 422670.98233521834

Epoch: 117   cost : 423714.6574024862

Epoch: 118   cost : 437306.48283325485

Epoch: 119   cost : 438778.2447087397

Epoch: 120   cost : 444188.96143988444

Epoch: 121   cost : 430192.3464676055

Epoch: 122   cost : 433303.4708338987

Epoch: 123   cost : 423944.2611462216

Epoch: 124   cost : 424099.84372053394

Epoch: 125   cost : 410774.4816079312

Epoch: 126   cost : 407442.07474157884

Epoch: 127   cost : 399222.52545388887

Epoch: 128   cost : 401150.7891464899

Epoch: 129   cost : 400163.12870260567

Epoch: 130   cost : 413556.3801094254

Epoch: 131   cost : 417535.35860896285

Epoch: 132   cost : 435165.5091392167

Epoch: 133   cost : 433315.29754717235

Epoch: 134   cost : 435911.0029446034

Epoch: 135   cost : 419894.31081704784

Epoch: 136   cost : 409369.96322840295

Epoch: 137   cost : 395339.1696185964

Epoch: 138   cost : 389112.3115506123

Epoch: 139   cost : 381663.7726552303
```

```
Epoch: 140   cost : 376789.4388806072

Epoch: 141   cost : 371839.0944766753

Epoch: 142   cost : 370648.65029686404

Epoch: 143   cost : 367884.62051686173

Epoch: 144   cost : 372609.7021336754

Epoch: 145   cost : 374673.63782460435

Epoch: 146   cost : 385733.13908655825

Epoch: 147   cost : 378760.8388824562

Epoch: 148   cost : 388199.9551712626

Epoch: 149   cost : 379645.57330062776

Epoch: 150   cost : 382980.53303840937

Epoch: 151   cost : 372886.9076133514

Epoch: 152   cost : 375898.29710235505

Epoch: 153   cost : 366738.812599777

Epoch: 154   cost : 365127.6492974369

Epoch: 155   cost : 356778.494385375

Epoch: 156   cost : 356192.8782871207

Epoch: 157   cost : 352324.4892822568

Epoch: 158   cost : 350230.1651767581

Epoch: 159   cost : 345707.74998233403

Epoch: 160   cost : 352341.09448178275

Epoch: 161   cost : 356244.74528847367

Epoch: 162   cost : 366692.44560557144

Epoch: 163   cost : 370551.5252029572
```

```
Epoch: 164   cost : 372706.5960184299

Epoch: 165   cost : 367427.1006111756

Epoch: 166   cost : 363801.5910157707

Epoch: 167   cost : 357378.0792734209

Epoch: 168   cost : 359586.78374580084

Epoch: 169   cost : 352765.6405612198

Epoch: 170   cost : 358366.3756165257

Epoch: 171   cost : 350487.9375526358

Epoch: 172   cost : 356958.5228680071

Epoch: 173   cost : 349427.6822810613

Epoch: 174   cost : 350033.08622744505

Epoch: 175   cost : 337385.46982368355

Epoch: 176   cost : 334733.81373357796

Epoch: 177   cost : 326433.8485395993

Epoch: 178   cost : 322126.92363890744

Epoch: 179   cost : 319466.5486975631

Epoch: 180   cost : 320110.695850184

Epoch: 181   cost : 319319.3358914547

Epoch: 182   cost : 317537.65388307796

Epoch: 183   cost : 318405.1656006864

Epoch: 184   cost : 316920.5331130328

Epoch: 185   cost : 318625.2580164484

Epoch: 186   cost : 315193.00620426686

Epoch: 187   cost : 316492.3055254497
```

```
Epoch: 188   cost : 311344.39977003005

Epoch: 189   cost : 311326.3537195303

Epoch: 190   cost : 306543.2333771809

Epoch: 191   cost : 308285.81537427084

Epoch: 192   cost : 305597.3864211434

Epoch: 193   cost : 310523.84967817465

Epoch: 194   cost : 309044.92635764973

Epoch: 195   cost : 317641.5934457205

Epoch: 196   cost : 312764.2422950839

Epoch: 197   cost : 323146.9019716187

Epoch: 198   cost : 315407.18003703567

Epoch: 199   cost : 321181.55948436796

Epoch: 200   cost : 309482.007523008

Epoch: 201   cost : 307523.6822661064

Epoch: 202   cost : 296223.7023827888

Epoch: 203   cost : 294377.88768680237

Epoch: 204   cost : 293284.99304634397

Epoch: 205   cost : 295488.26849352074

Epoch: 206   cost : 297133.20105949335

Epoch: 207   cost : 301933.2778803274

Epoch: 208   cost : 304063.65002889483

Epoch: 209   cost : 305034.4468681021

Epoch: 210   cost : 307522.14473194926

Epoch: 211   cost : 309479.2227772904
```

```
Epoch: 212   cost : 311292.3327781259

Epoch: 213   cost : 310875.79273686477

Epoch: 214   cost : 304053.4730249038

Epoch: 215   cost : 298699.06649218465

Epoch: 216   cost : 286042.98507641023

Epoch: 217   cost : 283191.9231399913

Epoch: 218   cost : 275336.83809866814

Epoch: 219   cost : 275740.96113174135

Epoch: 220   cost : 271252.46196106053

Epoch: 221   cost : 276756.47321848426

Epoch: 222   cost : 273110.04057747126

Epoch: 223   cost : 279547.069499292

Epoch: 224   cost : 275105.8817193453

Epoch: 225   cost : 277880.7619321681

Epoch: 226   cost : 273243.5718522008

Epoch: 227   cost : 274647.30210426747

Epoch: 228   cost : 267755.78276523505

Epoch: 229   cost : 268097.31233959756

Epoch: 230   cost : 262501.63790427794

Epoch: 231   cost : 261287.9961483633

Epoch: 232   cost : 257559.77530698327

Epoch: 233   cost : 257221.01563910895

Epoch: 234   cost : 257061.71408171084

Epoch: 235   cost : 258160.76165858615
```

```
Epoch: 236   cost : 263477.37446424563

Epoch: 237   cost : 269837.1494976071

Epoch: 238   cost : 274917.31831264944

Epoch: 239   cost : 282377.6729924796

Epoch: 240   cost : 285504.3231208232

Epoch: 241   cost : 286706.1288341622

Epoch: 242   cost : 281608.55490993656

Epoch: 243   cost : 278798.6941034166

Epoch: 244   cost : 272037.87844192283

Epoch: 245   cost : 268107.7621033685

Epoch: 246   cost : 261778.88498163898

Epoch: 247   cost : 259680.09051015798

Epoch: 248   cost : 253187.14311097487

Epoch: 249   cost : 255722.83590196463

Epoch: 250   cost : 253381.97490067335

Epoch: 251   cost : 259066.98927116668

Epoch: 252   cost : 258489.238870255

Epoch: 253   cost : 266480.6987339735

Epoch: 254   cost : 264217.30715352774

Epoch: 255   cost : 274891.19481960655

Epoch: 256   cost : 264177.880070192

Epoch: 257   cost : 264088.2105851575

Epoch: 258   cost : 252008.8956921252

Epoch: 259   cost : 249621.49539393146
```

```
Epoch: 260   cost : 243109.44443599565

Epoch: 261   cost : 241754.0487277545

Epoch: 262   cost : 239153.40658781998

Epoch: 263   cost : 237924.2507305311

Epoch: 264   cost : 238611.15050558833

Epoch: 265   cost : 238086.044992208

Epoch: 266   cost : 239283.92563280923

Epoch: 267   cost : 239096.96741583926

Epoch: 268   cost : 238261.4046074393

Epoch: 269   cost : 236835.23215178284

Epoch: 270   cost : 234664.69752681188

Epoch: 271   cost : 233535.70555460878

Epoch: 272   cost : 231206.6314813603

Epoch: 273   cost : 230315.57896143428

Epoch: 274   cost : 229150.43587920303

Epoch: 275   cost : 231423.22411173856

Epoch: 276   cost : 232978.16273984755

Epoch: 277   cost : 237297.5338939615

Epoch: 278   cost : 238496.40396374278

Epoch: 279   cost : 245390.82872703293

Epoch: 280   cost : 249907.19326336638

Epoch: 281   cost : 255953.71924149827

Epoch: 282   cost : 257231.30300393587

Epoch: 283   cost : 258294.93878319243
```

```
Epoch: 284   cost : 254184.78396146095

Epoch: 285   cost : 254642.73966598712

Epoch: 286   cost : 243484.9015588924

Epoch: 287   cost : 240507.07087351466

Epoch: 288   cost : 230588.81840018884

Epoch: 289   cost : 227077.56418418157

Epoch: 290   cost : 223818.3099655887

Epoch: 291   cost : 220747.75841578454

Epoch: 292   cost : 219542.95407481684

Epoch: 293   cost : 217400.1387330387

Epoch: 294   cost : 216721.6541237209

Epoch: 295   cost : 215782.4327880242

Epoch: 296   cost : 216171.04700071624

Epoch: 297   cost : 216082.7550406037

Epoch: 298   cost : 218170.89338542963

Epoch: 299   cost : 217567.49419813903

Epoch: 300   cost : 220279.75376455564
time taken: 2426.336138010025sec
```

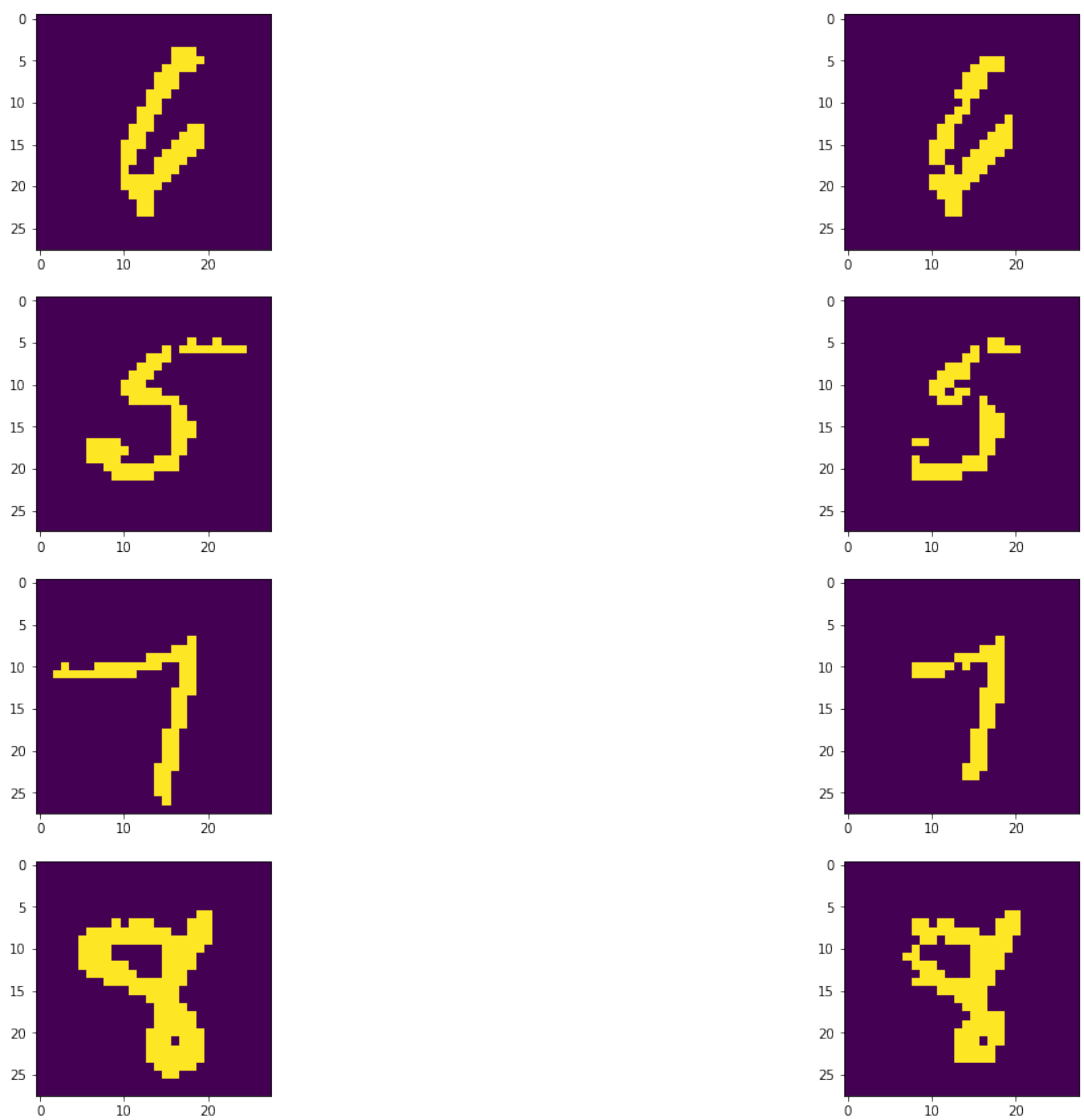
5.0.1 Predicting

```
In [20]: #testing samples
         alpha = np.load("alpha_weights_sae_v3.npy")
         beta = np.load("beta_weights_sae_v3.npy")
         # x_test = load_flattened_images("/home/snehith/Documents/machine learning/datasets/mni
         # np.save("test_set_350.npy" , x_test)
         x_test = np.asmatrix(np.load("test_set_350.npy"))
         #predicting the output
         res = net_predict(x_test , alpha , beta)
         # print(losses.shape)
```

```
plt.figure(figsize = (20,15))
for i in range(0,4):
    plt.subplot(4,2,2*i+1)
    plt.imshow(x_test[i].reshape(28,28))
    plt.subplot(4,2,2*i+2)
    plt.imshow(res[i].reshape(28,28))
```

```
/usr/local/lib/python3.5/dist-packages/ipykernel_launcher.py:5: DeprecationWarning: `imread` is
`imread` is deprecated in SciPy 1.0.0, and will be removed in 1.2.0.
Use ``imageio.imread`` instead.
  """
```

```
(350, 784)
```



In [10]:
```
#ploting the cost vs epochs
plt.plot(np.arange(epochs),losses)
plt.title("training samples: " + str(x_train.shape[0]))
plt.xlabel("epochs")
plt.ylabel("cost")
plt.show()
```