# The Experiment Report of
# *Machine Learning*

**College** Software College

**Subject** Software Engineering

**Members** 朱文德

**Student ID** 201530613948

**E-mail** 201530613948@mail.scut.edu.cn

**Tutor** 谭明奎

**Date submitted** 2017. 12 . 15

**1. Topic: NAG, RMSProp, AdaDelta and Adam's Comparison**

**2. Time: 2017/12/15**

**3. Reporter: wende zhu**

**4. Purposes: compare the ways of SGD, such as NAG, RMSProp, AdaDelta and Adam**

**5. Data sets and data analysis: a9a.txt: 32561 items, 123 features**

**a9a.t : 16281 items, 122(123) features**

**6. Experimental steps:**

1. Load the training set and validation set.

2. Initialize logistic regression model parameters, you can consider initializing zeros, random numbers or normal distribution.

3. Select the loss function and calculate its derivation, find more detail in PPT.

4. Calculate gradient $\mathbb{G}$ toward loss function from partial samples.

5. Update model parameters using different optimized methods(NAG， RMSProp， AdaDelta and Adam).

6. Select the appropriate threshold, mark the sample whose predict scores greater than the threshold as positive, on the contrary as negative. Predict under validation set and get the different optimized method loss $L_{NAG}$， $L_{RMSProp}$， $L_{AdaDelta}$ and $L_{Adam}$.

7. Repeat step 4 to 6 for several times, and drawing graph of $L_{NAG}$， $L_{RMSProp}$， $L_{AdaDelta}$ and $L_{Adam}$ with the number of iterations.

## 7. Code:

```
def compute_gradient_NAG(v,m_current,x,y,rp,size=200):

    m_gradient = 0
    gama = 0.9
    v.shape=(124,1)

    n=len(y)
    N = float(n)
    y.shape=(n,1)
    m_current.shape=(124,1)
    temp = random.randint(0, n - 200)
    for i in range(size):
        tt=x[temp+i].dot(m_current)
        m_gradient+=(sigmod(tt)-y[temp+i])*x[temp+i]
    m_gradient=m_gradient/N
    v = gama*v+rp*m_gradient.T

    new_m = m_current-v
    return new_m,v




def compute_gradient_RMSProp(v,m_current,x,y,rp,size=200):
    m_gradient = 0
    gama = 0.9
    v.shape=(124,1)

    n=len(y)
    N = float(n)
    y.shape=(n,1)
    m_current.shape=(124,1)
    temp = random.randint(0, n - 200)
    for i in range(size):
        tt=x[temp+i].dot(m_current)
        m_gradient+=(sigmod(tt)-y[temp+i])*x[temp+i]
    m_gradient=(m_gradient/N)
    v = gama*v+(1-gama)*(m_gradient*m_gradient.T)

    temp = rp/np.sqrt(v+np.exp(-8))
    new_m = m_current-np.multiply(temp,m_gradient.T)
    return new_m,v
```

```python
def compute_gradient_AdaDelta(delta,v,m_current,x,y,rp,size=200):

    m_gradient = 0
    gama = 0.95
    v.shape=(124,1)

    n=len(y)
    N = float(n)
    y.shape=(n,1)
    m_current.shape=(124,1)
    temp = random.randint(0, n - 200)
    for i in range(size):
        tt=x[temp+i].dot(m_current)
        m_gradient+=(sigmod(tt)-y[temp+i])*x[temp+i]
    m_gradient=m_gradient/N
    v = gama*v+(1-gama)*np.multiply(m_gradient.T,m_gradient.T)
    temp = np.sqrt(delta+np.exp(-8))/np.sqrt(v+np.exp(-8))
    temp2 = -np.multiply(temp,m_gradient.T)

    new_m = m_current+temp2
    delta=gama*delta+(1-gama)*np.multiply(temp2,temp2)
    return new_m,v,delta
def compute_gradient_Adam(g,v,m_current,x,y,rp,size=200):

    m_gradient = 0
    gama = 0.999
    beta = 0.9
    v.shape=(124,1)

    n=len(y)
    N = float(n)
    y.shape=(n,1)
    m_current.shape=(124,1)
    temp = random.randint(0, n - 200)
    for i in range(size):
        tt=x[temp+i].dot(m_current)
        m_gradient+=(sigmod(tt)-y[temp+i])*x[temp+i]
    m_gradient=m_gradient/N
    v = beta*v+(1-beta)*m_gradient.T
    g = gama*g+(1-gama)*np.multiply(g,g)
    alpha = rp*math.sqrt(1-gama)/(1-beta)
    new_m = m_current-alpha*v/np.sqrt(g+np.exp(-8))
    return new_m,v,g
```

**8. The initialization method of model parameters: all zeros.**

**9. The selected loss function and its derivatives:**

**Loss function =** $\frac{1}{n}\sum_{i=1}^{n}\log(1+e^{-y_i.w^T x})$

$$\frac{\partial L}{\partial w} = \frac{1}{n}\sum_{i=1}^{n}(h_w(x_i) - y_i)x_i$$

**10. Experimental results and curve:**

Hyper-parameter selection:NAG: $\lambda = 0.5$

RMSProp: $\gamma = 0.9$ $\lambda = 0.5$

AdaDelta: $\gamma = 0.95$ $\lambda = 0.5$

Adam: $\gamma = 0.999$ $\beta = 0.9$ $\lambda = 0.5$

Predicted Results (Best Results): m of NAG=[[ -5.75492401e-01]\n",

" [ -3.83374102e-01]\n",

" [ -2.82010464e-01]\n",

" [ -1.82141673e-01]\n",

" [ -2.48246259e-01]\n",

" [ -1.24376509e+00]\n",

" [ -1.17882084e-01]\n",

" [    5.59648564e-03]\n",

" [ -2.65202741e-02]\n",

" [ -8.94329045e-02]\n",

" [ -6.61472502e-02]\n",

" [ -5.18299588e-04]\n",

" [ -8.97189656e-04]\n",

" [ -3.34563713e-01]\n",

" [ -3.17093445e-01]\n",

" [ -3.29517050e-01]\n",

" [ -3.42683159e-01]\n",

" [ -3.47407532e-01]\n",

" [ -1.17636272e-01]\n",

" [ -4.29753284e-01]\n",

" [ -1.03722895e-01]\n",

" [ -6.89417604e-01]\n",

" [    1.54595131e-02]\n",

" [ -5.51211981e-02]\n",

" [ -6.48132507e-02]\n",

" [ -4.26564091e-02]\n",
" [ -5.19432081e-02]\n",
" [ -3.79683798e-02]\n",
" [  5.06979049e-03]\n",
" [ -1.26023311e-02]\n",
" [ -7.33857499e-02]\n",
" [  1.82520236e-02]\n",
" [ -2.66996021e-02]\n",
" [ -4.32604161e-03]\n",
" [ -3.53304617e-01]\n",
" [ -6.89417604e-01]\n",
" [ -4.29753284e-01]\n",
" [ -1.19934449e-01]\n",
" [ -7.88549453e-02]\n",
" [ -2.47529544e-01]\n",
" [ -3.39849152e-01]\n",
" [ -9.00047126e-01]\n",
" [ -8.03237506e-02]\n",
" [ -7.10468636e-02]\n",
" [ -3.26691210e-02]\n",
" [  2.00659104e-04]\n",
" [ -3.05970907e-02]\n",
" [ -2.26328153e-01]\n",
" [ -2.80621593e-01]\n",
" [ -1.76753530e-01]\n",

" [ -4.91373175e-02]\n",

" [ -5.98124583e-02]\n",

" [ -9.69266468e-02]\n",

" [ -1.49306006e-01]\n",

" [ -2.63528297e-01]\n",

" [ -7.08438391e-02]\n",

" [ -9.23778318e-02]\n",

" [ -1.34976184e-02]\n",

" [ -2.83478757e-02]\n",

" [ -5.91158380e-04]\n",

" [ -9.40364170e-03]\n",

" [ -4.58789967e-01]\n",

" [ -2.22560952e-01]\n",

" [ -6.19326139e-01]\n",

" [ -8.15497125e-02]\n",

" [ -2.79634486e-01]\n",

" [ -1.34895046e+00]\n",

" [ -4.79460785e-02]\n",

" [ -2.05855474e-02]\n",

" [ -2.18574145e-02]\n",

" [ -2.31925402e-01]\n",

" [ -7.79391225e-01]\n",

" [ -8.91873674e-01]\n",

" [ -1.70702482e+00]\n",

" [    3.57599205e-02]\n",

" [ -1.66079182e+00]\n",
" [ -1.04730757e-02]\n",
" [ -4.38351087e-01]\n",
" [ -1.39435865e-01]\n",
" [ -8.39745215e-01]\n",
" [ -9.23752097e-02]\n",
" [ -1.61357523e-01]\n",
" [ -1.47725467e+00]\n",
" [ -7.06348245e-04]\n",
" [ -3.96108437e-03]\n",
" [ -7.03434308e-03]\n",
" [ -5.64605327e-03]\n",
" [ -3.44891324e-03]\n",
" [ -1.11626141e-03]\n",
" [ -1.77068286e-03]\n",
" [ -1.26337574e-03]\n",
" [ -9.56582269e-04]\n",
" [ -2.99358556e-03]\n",
" [ -5.08180827e-03]\n",
" [ -4.69470963e-03]\n",
" [ -2.67129492e-04]\n",
" [ -6.26744146e-04]\n",
" [ -7.94085115e-03]\n",
" [ -1.94649710e-03]\n",
" [ -3.71137672e-03]\n",

" [ -6.42486270e-03]\n",
" [ -5.07197298e-03]\n",
" [ -5.09495183e-02]\n",
" [ -2.52189472e-03]\n",
" [ -1.90550191e-03]\n",
" [ -8.02782489e-04]\n",
" [ -5.17119529e-03]\n",
" [ -1.89091931e-03]\n",
" [ -1.81086235e-03]\n",
" [ -2.07219812e-03]\n",
" [ -2.37651415e-03]\n",
" [ -5.98304206e-03]\n",
" [ -7.91821386e-04]\n",
" [ -4.26334604e-03]\n",
" [ -2.86588729e-03]\n",
" [ -6.84063577e-04]\n",
" [ -1.52329774e-03]\n",
" [ -6.39198756e-04]\n",
" [ -7.58037779e-03]\n",
" [ -1.80867458e-03]\n",
" [ -2.05379327e-03]\n",
" [ -3.39407553e-04]\n",
" [ -1.69203946e-04]\n",
" [ -1.67126490e+00]] \n",

```
" m of RMSProp = [[ -2.89011682e+00]\n",
" [ -1.85826608e+00]\n",
" [ -1.27727425e+00]\n",
" [ -7.10645132e-01]\n",
" [ -1.09423688e+00]\n",
" [ -6.02771400e+00]\n",
" [ -5.17963510e-01]\n",
" [    8.39997083e-02]\n",
" [ -9.36918071e-02]\n",
" [ -3.14041048e-01]\n",
" [ -2.46869772e-01]\n",
" [ -1.15732903e-02]\n",
" [ -7.58102242e-03]\n",
" [ -1.61999769e+00]\n",
" [ -1.52089186e+00]\n",
" [ -1.43592054e+00]\n",
" [ -1.64956513e+00]\n",
" [ -1.60416393e+00]\n",
" [ -4.10517014e-01]\n",
" [ -2.09946339e+00]\n",
" [ -4.46960258e-01]\n",
" [ -3.33057318e+00]\n",
" [    1.29092557e-01]\n",
" [ -2.69521732e-01]\n",
" [ -2.68483392e-01]\n",
```

```
" [ -2.35715760e-01]\n",
" [ -2.79146269e-01]\n",
" [ -2.02781346e-01]\n",
" [  1.01433852e-01]\n",
" [ -7.78455962e-02]\n",
" [ -3.68133448e-01]\n",
" [  9.13446810e-02]\n",
" [ -1.37074769e-01]\n",
" [ -2.61940833e-02]\n",
" [ -1.77385153e+00]\n",
" [ -3.33057318e+00]\n",
" [ -2.09946339e+00]\n",
" [ -5.38005124e-01]\n",
" [ -8.86459244e-02]\n",
" [ -7.14306992e-01]\n",
" [ -1.57400889e+00]\n",
" [ -4.54735935e+00]\n",
" [ -4.34159123e-01]\n",
" [ -3.87898280e-01]\n",
" [ -1.73556397e-01]\n",
" [  7.49883269e-04]\n",
" [ -1.67821717e-01]\n",
" [ -1.00383721e+00]\n",
" [ -1.37405529e+00]\n",
" [ -7.41529581e-01]\n",
```

" [ -3.78500293e-02]\n",

" [ -2.14854897e-01]\n",

" [ -6.10124460e-01]\n",

" [ -7.51437626e-01]\n",

" [ -1.26099435e+00]\n",

" [ -3.85141338e-01]\n",

" [ -4.30372661e-01]\n",

" [ -6.41308485e-02]\n",

" [ -8.11571576e-02]\n",

" [ -4.54655182e-03]\n",

" [ -5.47808285e-02]\n",

" [ -2.31579067e+00]\n",

" [ -5.76205208e-01]\n",

" [ -3.04385648e+00]\n",

" [ -4.44681393e-01]\n",

" [ -1.39522457e+00]\n",

" [ -6.30724570e+00]\n",

" [ -2.43517772e-01]\n",

" [ -1.11680998e-01]\n",

" [ -1.00948538e-01]\n",

" [ -1.06714614e+00]\n",

" [ -3.86422174e+00]\n",

" [ -3.96631741e+00]\n",

" [ -8.11316075e+00]\n",

" [  2.82621595e-01]\n",

" [ -7.86546771e+00]\n",

" [    3.49285623e-02]\n",

" [ -2.29405067e+00]\n",

" [ -6.94414762e-01]\n",

" [ -4.05044160e+00]\n",

" [ -3.25488798e-01]\n",

" [ -4.66143321e-01]\n",

" [ -6.81477825e+00]\n",

" [ -4.82594379e-03]\n",

" [ -1.12474186e-02]\n",

" [ -3.95542574e-02]\n",

" [ -2.03954953e-02]\n",

" [ -1.64046167e-02]\n",

" [ -4.99250038e-03]\n",

" [ -1.24875803e-02]\n",

" [ -7.05271516e-03]\n",

" [ -4.53656213e-03]\n",

" [ -3.09533092e-02]\n",

" [ -2.31514578e-02]\n",

" [ -1.54561359e-02]\n",

" [ -4.58498254e-03]\n",

" [ -3.80019718e-03]\n",

" [ -3.85015182e-02]\n",

" [ -2.26027721e-02]\n",

" [ -1.77351648e-02]\n",

" [ -3.15940068e-02]\n",

" [ -3.14754533e-02]\n",

" [ -2.85408204e-01]\n",

" [ -1.16974542e-02]\n",

" [ -1.07949361e-02]\n",

" [ -7.39416750e-04]\n",

" [ -3.50809452e-02]\n",

" [ -7.26359822e-03]\n",

" [ -5.26740069e-03]\n",

" [ -1.41904675e-02]\n",

" [ -1.48364004e-02]\n",

" [ -3.02765628e-02]\n",

" [ -2.39572913e-03]\n",

" [ -2.52968123e-02]\n",

" [ -1.81644399e-02]\n",

" [ -7.39524276e-04]\n",

" [ -4.76609786e-03]\n",

" [ -3.10496989e-03]\n",

" [ -4.46024594e-02]\n",

" [ -5.30060833e-03]\n",

" [ -7.25978694e-03]\n",

" [ -3.79682511e-03]\n",

" [    0.00000000e+00]\n",

" [ -7.83053915e+00]]\n",

" /m of AdaDelta = [[ -1.36649609e-01]\n",

" [ -9.57774921e-02]\n",

" [ -7.84592514e-02]\n",

" [ -5.13519863e-02]\n",

" [ -7.09256527e-02]\n",

" [ -3.19564333e-01]\n",

" [ -2.97658776e-02]\n",

" [ -2.24867524e-03]\n",

" [ -8.47325103e-03]\n",

" [ -2.21392740e-02]\n",

" [ -1.60248369e-02]\n",

" [ -4.07153762e-04]\n",

" [ -1.76993713e-04]\n",

" [ -8.95013500e-02]\n",

" [ -8.18095215e-02]\n",

" [ -8.61690057e-02]\n",

" [ -8.89772839e-02]\n",

" [ -8.67078612e-02]\n",

" [ -3.92089647e-02]\n",

" [ -1.08753566e-01]\n",

" [ -2.30662314e-02]\n",

" [ -1.71518112e-01]\n",

" [    2.06105345e-03]\n",

" [ -1.34882505e-02]\n",

" [ -1.78743027e-02]\n",

" [ -1.01021287e-02]\n",

" [ -1.17232399e-02]\n",

" [ -8.95974920e-03]\n",

" [ -1.90984104e-03]\n",

" [ -3.47515543e-03]\n",

" [ -1.86142961e-02]\n",

" [  1.94102314e-03]\n",

" [ -7.31299180e-03]\n",

" [ -1.15553949e-03]\n",

" [ -8.44081050e-02]\n",

" [ -1.71518112e-01]\n",

" [ -1.08753566e-01]\n",

" [ -3.13624789e-02]\n",

" [ -3.71165571e-02]\n",

" [ -8.19627138e-02]\n",

" [ -8.30091352e-02]\n",

" [ -2.21368799e-01]\n",

" [ -1.97942158e-02]\n",

" [ -1.92729722e-02]\n",

" [ -7.71009956e-03]\n",

" [ -3.27334977e-05]\n",

" [ -1.07576007e-02]\n",

" [ -5.72072428e-02]\n",

" [ -6.89402134e-02]\n",

" [ -4.36219770e-02]\n",

" [ -1.75191285e-02]\n",

" [ -2.19417215e-02]\n",

" [ -2.57946390e-02]\n",

" [ -3.49890230e-02]\n",

" [ -6.53148781e-02]\n",

" [ -1.84777729e-02]\n",

" [ -2.46413859e-02]\n",

" [ -3.51917693e-03]\n",

" [ -5.75265317e-03]\n",

" [ -2.08923238e-04]\n",

" [ -6.30619283e-03]\n",

" [ -1.11370108e-01]\n",

" [ -7.09571080e-02]\n",

" [ -1.55824283e-01]\n",

" [ -2.10832685e-02]\n",

" [ -6.76203526e-02]\n",

" [ -3.53549748e-01]\n",

" [ -1.41789024e-02]\n",

" [ -5.42153980e-03]\n",

" [ -4.68776875e-03]\n",

" [ -5.52487047e-02]\n",

" [ -1.96756182e-01]\n",

" [ -2.36374050e-01]\n",

" [ -4.34528060e-01]\n",

" [    1.50778807e-03]\n",

```
" [ -4.28421079e-01]\n",
" [ -4.60448849e-03]\n",
" [ -1.08479647e-01]\n",
" [ -3.83598640e-02]\n",
" [ -2.16793129e-01]\n",
" [ -2.32318794e-02]\n",
" [ -4.62859818e-02]\n",
" [ -3.80064261e-01]\n",
" [ -1.51856900e-04]\n",
" [ -1.44295572e-03]\n",
" [ -2.29797558e-03]\n",
" [ -1.01993266e-03]\n",
" [ -7.56174695e-04]\n",
" [ -1.94537334e-04]\n",
" [ -9.68150604e-04]\n",
" [ -5.99463424e-04]\n",
" [ -3.61738895e-04]\n",
" [ -1.25257326e-03]\n",
" [ -9.89966779e-04]\n",
" [ -1.41395574e-03]\n",
" [ -2.43028927e-04]\n",
" [ -2.22568147e-04]\n",
" [ -2.30884401e-03]\n",
" [ -7.63031064e-04]\n",
" [ -8.01487746e-04]\n",
```

" [ -1.50541306e-03]\n",

" [ -1.76008588e-03]\n",

" [ -1.35604174e-02]\n",

" [ -7.06290381e-04]\n",

" [ -3.65312299e-04]\n",

" [ -1.09838522e-04]\n",

" [ -1.63666278e-03]\n",

" [ -4.25277896e-04]\n",

" [ -5.71850210e-04]\n",

" [ -3.28249233e-04]\n",

" [ -9.05554868e-04]\n",

" [ -1.43839485e-03]\n",

" [ -1.17394200e-04]\n",

" [ -1.13768015e-03]\n",

" [ -8.21692959e-04]\n",

" [ -2.16038675e-04]\n",

" [ -8.03669393e-05]\n",

" [ -7.50571304e-05]\n",

" [ -2.02147088e-03]\n",

" [ -1.28230750e-04]\n",

" [ -7.87077366e-04]\n",

" [ -3.55274562e-04]\n",

" [ -3.98745084e-05]\n",

" [ -4.33017750e-01]]\n",

" m of Adam = [[ -9.97315173e-01]\n",

" [ -6.34649505e-01]\n",

" [ -4.55626680e-01]\n",

" [ -2.79188368e-01]\n",

" [ -3.76680287e-01]\n",

" [ -2.07389601e+00]\n",

" [ -1.80585979e-01]\n",

" [    4.35227976e-03]\n",

" [ -2.74958508e-02]\n",

" [ -1.42962619e-01]\n",

" [ -9.45288376e-02]\n",

" [ -3.45397916e-03]\n",

" [    0.00000000e+00]\n",

" [ -5.58345901e-01]\n",

" [ -5.22713374e-01]\n",

" [ -5.17406489e-01]\n",

" [ -5.87564893e-01]\n",

" [ -5.57429356e-01]\n",

" [ -1.63578740e-01]\n",

" [ -7.56964972e-01]\n",

" [ -1.53350883e-01]\n",

" [ -1.11741774e+00]\n",

" [    3.39046346e-02]\n",

" [ -8.83997716e-02]\n",

" [ -1.12846699e-01]\n",

" [ -8.12036135e-02]\n",

" [ -7.83333102e-02]\n",

" [ -5.31581603e-02]\n",

" [  7.99709520e-03]\n",

" [ -2.40581338e-02]\n",

" [ -1.27844504e-01]\n",

" [  3.50180025e-02]\n",

" [ -5.34430114e-02]\n",

" [ -9.78020677e-03]\n",

" [ -5.81171823e-01]\n",

" [ -1.11741774e+00]\n",

" [ -7.56964972e-01]\n",

" [ -2.01246471e-01]\n",

" [ -8.66590073e-02]\n",

" [ -3.27672308e-01]\n",

" [ -5.73939573e-01]\n",

" [ -1.53933937e+00]\n",

" [ -1.29149881e-01]\n",

" [ -1.24294605e-01]\n",

" [ -4.70578539e-02]\n",

" [ -2.00641911e-03]\n",

" [ -5.77557842e-02]\n",

" [ -3.47569241e-01]\n",

" [ -4.96188726e-01]\n",

" [ -2.73407550e-01]\n",

" [ -5.23497514e-02]\n",

" [ -1.03114455e-01]\n",

" [ -1.83684878e-01]\n",

" [ -2.57798312e-01]\n",

" [ -4.25457925e-01]\n",

" [ -1.13770841e-01]\n",

" [ -1.49420789e-01]\n",

" [ -2.63503623e-02]\n",

" [ -3.06424811e-02]\n",

" [ -1.05989829e-03]\n",

" [ -2.88953063e-02]\n",

" [ -7.86746249e-01]\n",

" [ -2.70193442e-01]\n",

" [ -1.05217967e+00]\n",

" [ -1.35518092e-01]\n",

" [ -4.69927252e-01]\n",

" [ -2.23322753e+00]\n",

" [ -8.24487715e-02]\n",

" [ -3.59233214e-02]\n",

" [ -3.04119915e-02]\n",

" [ -3.61448403e-01]\n",

" [ -1.31081475e+00]\n",

" [ -1.43264526e+00]\n",

" [ -2.81102903e+00]\n",

" [ 6.75690131e-02]\n",

" [ -2.72374084e+00]\n",
" [ -1.97191771e-02]\n",
" [ -7.75142453e-01]\n",
" [ -2.42507738e-01]\n",
" [ -1.39151480e+00]\n",
" [ -1.32819265e-01]\n",
" [ -2.01475757e-01]\n",
" [ -2.40536150e+00]\n",
" [ -9.35677680e-04]\n",
" [ -7.29188198e-03]\n",
" [ -1.73728856e-02]\n",
" [ -8.76696221e-03]\n",
" [ -4.49742744e-03]\n",
" [ -1.57751554e-03]\n",
" [ -4.70261353e-03]\n",
" [ -5.15258362e-03]\n",
" [ -1.62935854e-03]\n",
" [ -7.88776728e-03]\n",
" [ -5.44957825e-03]\n",
" [ -8.33958372e-03]\n",
" [ -1.08449867e-03]\n",
" [ -1.17913389e-03]\n",
" [ -1.37952126e-02]\n",
" [ -3.65247860e-03]\n",
" [ -6.53979108e-03]\n",

" [ -9.24283650e-03]\n",

" [ -9.35257239e-03]\n",

" [ -9.25540725e-02]\n",

" [ -3.69478576e-03]\n",

" [ -1.16783833e-03]\n",

" [ -1.03455844e-03]\n",

" [ -9.50212823e-03]\n",

" [ -2.85376319e-03]\n",

" [ -3.29073318e-03]\n",

" [ -8.51076721e-04]\n",

" [ -4.83092335e-03]\n",

" [ -7.96861311e-03]\n",

" [ -7.80111250e-04]\n",

" [ -9.46053504e-03]\n",

" [ -6.36275286e-03]\n",

" [ -2.31412743e-03]\n",

" [ -1.27008062e-03]\n",

" [ -2.68384474e-04]\n",

" [ -1.35667750e-02]\n",

" [ -2.61834008e-03]\n",

" [ -3.77931965e-03]\n",

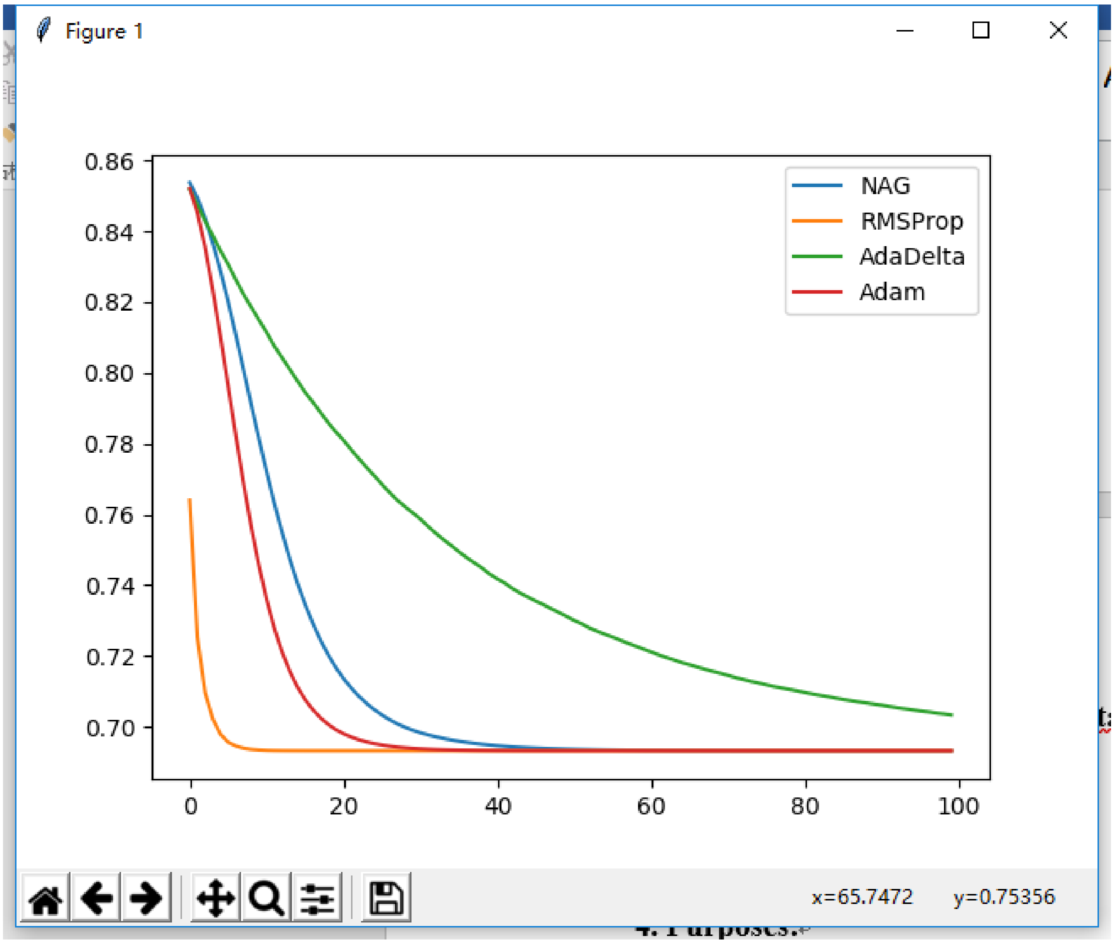
" [    4.03437552e-04]\n",

" [    0.00000000e+00]\n",

" [ -2.74346001e+00]]\n"

]}

Loss curve:




**11. Results analysis:**

In graph, we can see that RMSProp has the fast speed to gradient, but Adam has batter in other three methods. AdaDelta's speed is lowest, I think it maybe modify the AdaDelta's λ.

**12. Similarities and differences between logistic regression and linear classification:**

**Logistic regression is prediction of probability, and linear classification is split types of data.**

**13. Summary:**

**In logistic regression, I find the loss function' value is between**

**0.410037596 $\frac{\log 2+\log(1+e^{-2})}{2}$ and 1.00320443$\frac{\log 2+\log(1+e)}{2}$.**