Pacote 'deepnet'

19 de fevereiro de 2015

Tipo Pacote

Title toolkit de aprendizagem profunda em R

Versão 0.2

Data 2014-03-20

Autor Xiao Rong

Manutenção Xiao Rong <runxiao@gmail.com>

Descrição Implementar algumas arquiteturas de aprendizado profundo e redes neurais
Algoritmos, incluindo BP, RBM, DBN, Deep autoencoder e assim por diante.

Licença GPL

NecessidadesCompilação não

Repositório CRAN

Data / publicação 2014-03-20 10:03:43

R documentados:

Descrição

Treinamento de uma rede neural profunda com pesos inicializados por DBN

Uso

```
Dbn.dnn.train (x, y, hidden = c (1), activationfun = "sigm", learningrate = 0,8,
    Momentum = 0,5, learningrate_scale = 1, output = "sigm", numepochs = 3,
    Batchsize = 100, hidden_dropout = 0, visible_dropout = 0, cd = 1)
```

Argumentos

X	Matriz de valores x para exemplos
Y	Vetor ou matriz de valores-alvo para exemplos
escondido	Vector para o número de unidades de camadas escondidas.Default é c (10).
Fim de ativação	Função de ativação da unidade oculta.Pode ser "sigm", "linear" ou "tanh" .Default é "Sigm" para a função logística
Aprendizado	Taxa de aprendizagem para a descida do gradiente. O padrão é 0.8.
Impulso	Momento para a descida do gradiente. O padrão é 0.5.
Learningrate_scale	Taxa de aprendizagem será mutiplied por esta escala após cada iteração. O padrão é 1.
Números	Número de iteração para amostras O padrão é 3.
tamanho do batch	Tamanho do mini-lote. O padrão é 100.
saída	Função da unidade de saída, pode ser "sigm", "linear" ou "softmax". A predefinição é "sigm".
Hidden_dropout	drop out fração para camada oculta. O padrão é 0.
Visible_dropout	Fração de saída para a camada de entrada Padrão é 0.
CD	Número de iteração para a amostra de Gibbs do algoritmo CD.

Autor (es)

Xiao Rong

Exemplos

```
(50, 1, 0,5), rnorm (50, -0,6, 0,2))
Var2 <-c (rnorm (50, -0,8, 0,2), rnorm (50, 2, 1))
X <- matriz (c (Var1, Var2), nrow = 100, ncol = 2)
Y <- c (rep (1, 50), rep (0, 50))
Dnn <- dbn.dnn.train (x, y, hidden = c (5, 5))
## predict by dnn
```



```
Test_Var1 <-c (rnorm (50, 1, 0,5), rnorm (50, -0,6, 0,2))
Test_Var2 <-c (rnorm (50, -0,8, 0,2), rnorm (50, 2, 1))
Test_x <- matrix (c (test_Var1, test_Var2), nrow = 100, ncol = 2)
Nn.test (dnn, test_x, y)
```

Load.mnist Carregar MNIST DataSet

Descrição

Carregar MNIST DataSet

Uso

```
Load.mnist (dir)
```

Argumentos

Dir	Dir do minst dataset

Valor

Mnist dataset trem $ n número de comboios treinar trem $ x pix de cada comboio trem de imagem de amostra $ y etiqueta
De cada trem imagem de amostra trem $ yy um-de-c vetor de etiqueta de trem amostra imagem teste $ n número
Do teste de amostras de teste $ x pix de cada teste de imagem de amostra de teste $ y rótulo de cada teste de amostra de teste de imagem $ yy
Um-de-c vetor de rótulo de amostra de teste imagem

Autor (es)

Xiao Rong

Nn.predict Prever novas amostras por Trainded NN

Descrição

Prever novas amostras por Trainded NN

Uso

```
Nn.predict (nn, x)
```

Argumentos

Nn	Nerual treinada pela função nn.train
X	Novas amostras para prever

Valor

Retorna o valor de saída bruta de rede neural.Para tarefa de classificação, retornar a probabilidade de uma classe

Autor (es)

Xiao Rong

Exemplos

```
(50, 1, 0,5), rnorm (50, -0,6, 0,2))
Var2 <-c (rnorm (50, -0,8, 0,2), rnorm (50, 2, 1))
X <- matriz (c (Var1, Var2), nrow = 100, ncol = 2)
Y <- c (rep (1, 50), rep (0, 50))
Nn <- nn.train (x, y, hidden = c (5))
## predict by nn
Test_Var1 <-c (rnorm (50, 1, 0,5), rnorm (50, -0,6, 0,2))
Test_Var2 <-c (rnorm (50, -0,8, 0,2), rnorm (50, 2, 1))
Test_x <- matrix (c (test_Var1, test_Var2), nrow = 100, ncol = 2)
Yy <- nn.predict (nn, test_x)
```

Nn.test Testar novas amostras por Trainded NN

Descrição

Testar novas amostras por TraNed NN, taxa de erro de retorno para classificação

Uso

```
Nn.test (nn, x, y, t = 0,5)
```

Argumentos

Nn	Nerual treinada pela função nn.train

X	Novas amostras para prever
Y	Novo rótulo de amostras
T	Limiar de classificação. Se nn.predict valor> = t, em seguida, etiqueta 1, outra etiqueta 0

Valor

taxa de erro

Autor (es)

Xiao Rong

Exemplos

```
(50, 1, 0,5), rnorm (50, -0,6, 0,2))
Var2 <-c (rnorm (50, -0,8, 0,2), rnorm (50, 2, 1))
X <- matriz (c (Var1, Var2), nrow = 100, ncol = 2)
Y <- c (rep (1, 50), rep (0, 50))
Nn <- nn.train (x, y, hidden = c (5))
Test_Var1 <-c (rnorm (50, 1, 0,5), rnorm (50, -0,6, 0,2))
Test_Var2 <-c (rnorm (50, -0,8, 0,2), rnorm (50, 2, 1))
Test_x <- matrix (c (test_Var1, test_Var2), nrow = 100, ncol = 2)
Err <- nn.test (nn, test_x, y)
```

Nn.train Rede Neural de Treinamento

Descrição

Treinamento único ou mutiple camadas escondidas rede neural por BP

Uso

```
Nn.train (x, y, initW = NULL, initB = NULL, ocultos = c (10), activationfun = "sigm",
    Learningrate = 0,8, momentum = 0,5, learningrate_scale = 1, output = "sigm",
    Numepochs = 3, batchsize = 100, hidden_dropout = 0, visible_dropout = 0)
```

Argumentos

X	Matriz de valores x para exemplos
Y	Vetor ou matriz de valores-alvo para exemplos
InitW	Pesos iniciais. Se faltar escolhido aleatoriamente
InitB	Viés inicial. Se faltar escolhido aleatoriamente
escondido	Vector para o número de unidades de camadas escondidas.Default é c (10).
Fim de ativação	Função de ativação da unidade oculta.Pode ser "sigm", "linear" ou "tanh" .Default é "Sigm" para a função logística
Aprendizado	Taxa de aprendizagem para a descida do gradiente. O padrão é 0.8.
Impulso	Momento para a descida do gradiente. O padrão é 0.5.
Learningrate_scale	Taxa de aprendizagem será mutiplied por esta escala após cada iteração. O padrão é 1.
Números	Número de iteração para amostras O padrão é 3.
tamanho do batch	Tamanho do mini-lote. O padrão é 100.
saída	Função da unidade de saída, pode ser "sigm", "linear" ou "softmax". O padrão é "sigm".
Hidden_dropout	drop out fração para camada oculta. O padrão é 0.
Visible_dropout	Fração de saída para a camada de entrada Padrão é 0.

Autor (es)

Xiao Rong

Exemplos

```
(50, 1, 0,5), rnorm (50, -0,6, 0,2))
Var2 <-c (rnorm (50, -0,8, 0,2), rnorm (50, 2, 1))
X <- matriz (c (Var1, Var2), nrow = 100, ncol = 2)
Y <- c (rep (1, 50), rep (0, 50))
Nn <- nn.train (x, y, hidden = c (5))
```

Rbm.down Gerar vetor visível por estados de unidades ocultas

Descrição

Gerar vetor visível por estados de unidades ocultas

Uso

```
Rbm.down (rbm, h)
```

Argumentos

Rbm	Um objeto rbm treinado pela função train.rbm
H	Estados de unidades ocultas

Valor

Vetor visível gerado

Autor (es)

Xiao Rong

Exemplos

```
Var1 <- c (rep (1, 50), rep (0, 50))
Var2 <- c (rep (0, 50), rep (1, 50))
X3 <- matriz (c (Var1, Var2), nrow = 100, ncol = 2)
Rl <- rbm.train (x3, 3, numepochs = 20, cd = 10)
H <- c (0,2, 0,8, 0,1)
V <- rbm.down (r1, h)
```


Rbm.train Treinando um RBM (máquina Boltzmann restrita)

Descrição

Treinando um RBM (máquina Boltzmann restrita)

Uso

```
Rbm.train (x, ocultos, numepochs = 3, batchsize = 100, learningrate = 0,8,
    Learningrate_scale = 1, momentum = 0,5, visible_type = "bin", hidden_type = "bin",
     Cd = 1)
```

Argumentos

X	Matriz de valores x para exemplos
escondido	Número de unidades ocultas
Tipo_visível	Função de ativação da unidade de entrada.Only apoio "sigm" agora
Tipo escondido	Função de ativação da unidade escondida.Only apoio "sigm" agora
Aprendizado	Taxa de aprendizagem para a descida do gradiente. O padrão é 0.8.
Impulso	Momento para a descida do gradiente. O padrão é 0.5.
Learningrate_scale	Taxa de aprendizagem será mutiplied por esta escala após cada iteração. O padrão é 1.
Números	Número de iteração para amostras O padrão é 3.
tamanho do batch	Tamanho do mini-lote. O padrão é 100.
CD	Número de iteração para a amostra de Gibbs do algoritmo CD.

Autor (es)

Xiao Rong

Exemplos

```
Var1 <- c (rep (1, 50), rep (0, 50))
Var2 <- c (rep (0, 50), rep (1, 50))
X3 <- matriz (c (Var1, Var2), nrow = 100, ncol = 2)
R1 <- rbm.train (x3, 10, numepochs = 20, cd = 10)
```


Rbm up Inferir unidades ocultas estado por unidades visíveis

Descrição

Inferir unidades ocultas estados por unidades visíveis

Uso

```
Rbm.up (rbm, v)
```

Argumentos

Rbm	Um objeto rbm treinado pela função train.rbm
V	Estados de unidades visíveis

Valor

Estados de unidades ocultas

Autor (es)

Xiao Rong

Exemplos

```
Var1 <- c (rep (1, 50), rep (0, 50))
Var2 <- c (rep (0, 50), rep (1, 50))
X3 <- matriz (c (Var1, Var2), nrow = 100, ncol = 2)
R1 <- rbm.train (x3, 3, numepochs = 20, cd = 10)
V <-c (0,2, 0,8)
H <- rbm.up (r1, v)
```

Sae.dnn.train Treinando uma rede neural profunda com pesos inicializados por Stacked AutoEncoder

Descrição

Treinando uma rede neural profunda com pesos inicializados pelo Stacked AutoEncoder

Uso

```
Sae.dnn.train (x, y, hidden = c (1), activationfun = "sigm", learningrate = 0,8,
    Momentum = 0,5, learningrate_scale = 1, output = "sigm", sae_output = "linear",
     Numepochs = 3, batchsize = 100, hidden_dropout = 0, visible_dropout = 0)
```


Argumentos

X	Matriz de valores x para exemplos
Y	Vetor ou matriz de valores-alvo para exemplos
escondido	Vector para o número de unidades de camadas escondidas.Default é c (10).
Fim de ativação	Função de ativação da unidade oculta.Pode ser "sigm", "linear" ou "tanh" .Default é "Sigm" para a função logística
Aprendizado	Taxa de aprendizagem para a descida do gradiente. O padrão é 0.8.
Impulso	Momento para a descida do gradiente. O padrão é 0.5.
Learningrate_scale	Taxa de aprendizagem será mutiplied por esta escala após cada iteração. O padrão é 1.
Números	Número de iteração para amostras O padrão é 3.
tamanho do batch	Tamanho do mini-lote. O padrão é 100.
saída	Função da unidade de saída, pode ser "sigm", "linear" ou "softmax". O padrão é "sigm".
Sae_output	Função da unidade de saída do auto-encoder, pode ser "sigm", "linear" ou "softmax". De-Falha é "linear".
Hidden_dropout	drop out fração para camada oculta. O padrão é 0.
Visible_dropout	Fração de saída para a camada de entrada Padrão é 0.

Autor (es)

Xiao Rong

Exemplos

```
(50, 1, 0,5), rnorm (50, -0,6, 0,2))
Var2 <-c (rnorm (50, -0,8, 0,2), rnorm (50, 2, 1))
X <- matriz (c (Var1, Var2), nrow = 100, ncol = 2)
Y <- c (rep (1, 50), rep (0, 50))
Dnn <- sae.dnn.train (x, y, hidden = c (5, 5))
## predict by dnn
Test_Var1 <-c (rnorm (50, 1, 0,5), rnorm (50, -0,6, 0,2))
Test_Var2 <-c (rnorm (50, -0,8, 0,2), rnorm (50, 2, 1))
Test_x <- matrix (c (test_Var1, test_Var2), nrow = 100, ncol = 2)
Nn.test (dnn, test_x, y)
```

Índice