

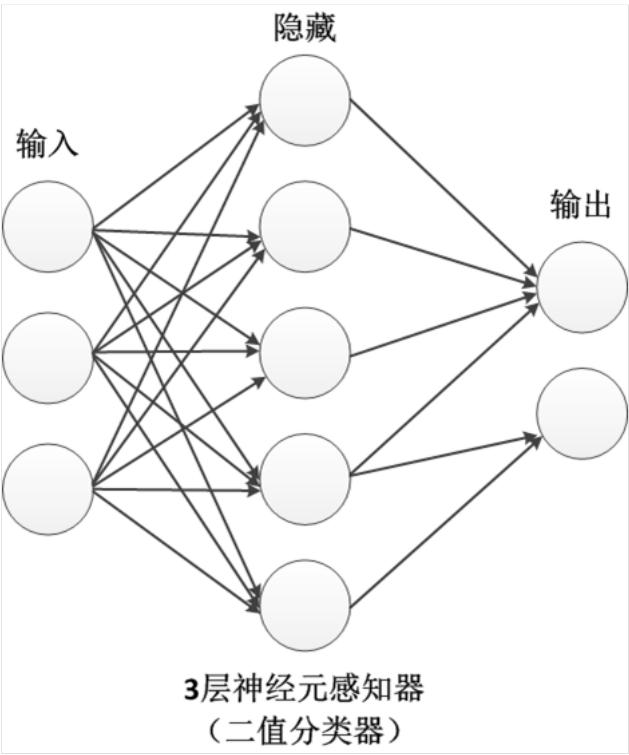


原 TensorFlow人工智能引擎入门教程之八 接着补充一章MLP多层感知器网络原理以及 使用

2016-05-15 基陆伯 阅 7 分享： 微信 转藏到我的图书馆

这一章我们讲MLP 多层感知器 的使用，多层感知器，常用来做分类，效果非常好，比如文本分类，效果比SVM 贝叶斯 好多了，这些以前的机器学习很有名的算法，我现在基本不用它们，现在是深度学习 的A时代。

多层感知器的介绍



MLP（多层感知器）神经网络是常见的ANN算法，它由一个输入层，一个输出层和一个或多个隐藏层组成。

在MLP中的所有神经元都差不多，每个神经元都有几个输入（连接前一层）神经元和输出（连接后一层）神经元，该神经元会将相同值传递给与之相连的多个输出神经元

一个神经网络训练网将一个特征向量作为输入，将该向量传递到隐藏层，然后通过权重和激励函数来计算结果，并将结果传递给下一层，直到最后传递给输出层才结束

首先我们来

下面是一个2层的多层感知器

其中 relu可以换成 tanh或者sigmoid

基陆伯 图书馆

★★★★★

11495 馆藏 33889

TA的推荐 TA的最新馆藏

永远成功的秘密，就是每天淘汰自己
我们将永生还是灭绝？人工智能很...
我们将永生还是灭绝？人工智能很...
[转] 赞美的大能
他们还不信我要到几时呢？
基督徒的委身【】

KELA珂兰

5.20示爱必备

金玫瑰任性送

推荐阅读 更多

BetaCat 的前生后世
揪出bug！解析调试神经网络的技巧
深度学习计算模型中“门函数（Ga...
简易的深度学习框架Keras代码解析...
国外公司开发新型移动无线网pCell...
enum的用法
再谈：义和团史实（转）
是还没有受洗，还没有正式参加某...
帧缓存

[北京育才美容培训]

美容美发
化妆培训

<师资力量雄厚>

报名电话：
010-63969632

1 美亚保险官网	7 企业邮箱申请
2 美亚保险	8 中老年妈妈装
3 公司邮箱	9 企业邮箱
4 用英语介绍美国	10 led亮化照明
5 北京口腔医院	11 英语学习
6 钱爸爸理财	12 企业邮箱注册

比如

```
tf.nn.sigmoid(tf.matmul(X, w_h)) #WX+B

def multilayer_perceptron(X, _weights, _biases):
    layer_1 = tf.nn.relu(tf.add(tf.matmul(X, _weights['h1']), _biases['b1'])) #Hidden layer with RELU activation
    layer_2 = tf.nn.relu(tf.add(tf.matmul(layer_1, _weights['h2']), _biases['b2'])) #Hidden layer with RELU activation
    return tf.matmul(layer_2, _weights['out']) + _biases['out'] # Store layer weights & bias
weights = {
    'h1': tf.Variable(tf.random_normal([n_input, 256])),
    'h2': tf.Variable(tf.random_normal([256, 256])),
    'out': tf.Variable(tf.random_normal([256, 10]))
}
biases = {
    'b1': tf.Variable(tf.random_normal([256])),
    'b2': tf.Variable(tf.random_normal([256])),
    'out': tf.Variable(tf.random_normal([10]))
}
```

或者 修改成使用sigmoid

linear----线性感知器

tanh----双曲正切函数

sigmoid----双曲函数

softmax---- $1/(e^{net}) * e^{(w_i \cdot x_i - shift)}$

log-softmax---- $\log(1/(e^{net}) * e^{(w_i \cdot x_i)})$

exp----指数函数

softplus---- $\log(1 + e^{(w_i \cdot x_i)})$

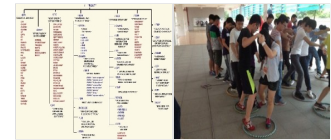
```
def multilayer_perceptron(X, _weights, _biases):
    layer_1 = tf.nn.sigmoid(tf.add(tf.matmul(X, _weights['h1']), _biases['b1'])) #Hidden layer with sigmoid activation
    layer_2 = tf.nn.relu(tf.add(tf.matmul(layer_1, _weights['h2']), _biases['b2'])) #Hidden layer with RELU activation
    return tf.matmul(layer_2, _weights['out']) + _biases['out']

import input_data
mnist = input_data.read_data_sets("/tmp/data/", one_hot=True)
import tensorflow as tf
# Parameters
learning_rate = 0.001
training_epochs = 15
batch_size = 100
display_step = 1
# Network Parameters
n_hidden_1 = 256 # 1st layer num features
n_hidden_2 = 256 # 2nd layer num features
n_input = 784 # MNIST data input (img shape: 28*28)
n_classes = 10 # MNIST total classes (0-9 digits)
# tf Graph input
x = tf.placeholder(tf.float32, [None, n_input])
y = tf.placeholder(tf.float32, [None, n_classes])
# Create model
def multilayer_perceptron(X, _weights, _biases):
    layer_1 = tf.nn.sigmoid(tf.add(tf.matmul(X, _weights['h1']), _biases['b1'])) #Hidden layer with sigmoid activation
    layer_2 = tf.nn.relu(tf.add(tf.matmul(layer_1, _weights['h2']), _biases['b2'])) #Hidden layer with RELU activation
    return tf.matmul(layer_2, _weights['out']) + _biases['out']
# Store layers weight & bias
weights = {
    'h1': tf.Variable(tf.random_normal([n_input, n_hidden_1])),
    'h2': tf.Variable(tf.random_normal([n_hidden_1, n_hidden_2])),
    'out': tf.Variable(tf.random_normal([n_hidden_2, n_classes]))
}
biases = {
    'b1': tf.Variable(tf.random_normal([n_hidden_1])),
    'b2': tf.Variable(tf.random_normal([n_hidden_2])),
    'out': tf.Variable(tf.random_normal([n_classes]))
}
# Construct model
pred = multilayer_perceptron(x, weights, biases)
# Define loss and optimizer
cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(pred, y))
# Softmax loss optimizer = tf.train.AdamOptimizer(learning_rate=learning_rate).minimize(cost)
# Adam Optimizer
# Initializing the variables
init = tf.initialize_all_variables()
# Launch the graph
with tf.Session() as sess:
    sess.run(init)
    # Training cycle
    for epoch in range(training_epochs):
        avg_cost = 0
        total_batch = int(mnist.train.num_examples/batch_size)
        # Loop over all batches
        for i in range(total_batch):
            batch_xs, batch_ys = mnist.train.next_batch(batch_size)
            # Fit training using batch data
            sess.run(optimizer, feed_dict={x: batch_xs, y: batch_ys})
            # Compute average loss
            avg_cost += sess.run(cost, feed_dict={x: batch_xs, y: batch_ys})/total_batch
        # Display logs per epoch step
        if epoch % display_step == 0:
            print "Epoch:", '%04d' % (epoch+1), "cost=", "{:.9f}".format(avg_cost)
            print "Optimization Finished!"
            # Test model
            correct_prediction = tf.equal(tf.argmax(pred, 1), tf.argmax(y, 1))
            # Calculate accuracy
            accuracy = tf.reduce_mean(tf.cast(correct_prediction, "float"))
            print "Accuracy:", accuracy.eval({x: mnist.test.images, y: mnist.test.labels})
```

下面我们运行测试 代码



小生意项目



linux学习路线

拓展项目



文件管理系统

太阳公元二

关闭

```
root@iZulcdurupZ:~/tensorflowtest# python mlp.py
('Extracting', '/tmp/data/train-images-idx3-ubyte.gz')
('Extracting', '/tmp/data/train-labels-idx1-ubyte.gz')
('Extracting', '/tmp/data/t10k-images-idx3-ubyte.gz')
('Extracting', '/tmp/data/t10k-labels-idx1-ubyte.gz')
I tensorflow/core/common_runtime/local_device.cc:25] Local device intra op parallelism threads: 1
I tensorflow/core/common_runtime/local_session.cc:45] Local session inter op parallelism threads: 1

```

渠道 转移规则

```
Testing Accuracy: 0.970302
root@iZulcdurupZ:~/tensorflowtest# rz
root@iZulcdurupZ:~/tensorflowtest# python mlp.py
('Extracting', '/tmp/data/train-images-idx3-ubyte.gz')
('Extracting', '/tmp/data/train-labels-idx1-ubyte.gz')
('Extracting', '/tmp/data/t10k-images-idx3-ubyte.gz')
('Extracting', '/tmp/data/t10k-labels-idx1-ubyte.gz')
I tensorflow/core/common_runtime/local_device.cc:25] Local device intra op parallelism th
I tensorflow/core/common_runtime/local_session.cc:45] Local session inter op parallelism
Epoch: 0001 cost= 184.006677298
Epoch: 0002 cost= 43.614341220
Epoch: 0003 cost= 27.406910379
Epoch: 0004 cost= 18.896572759
Epoch: 0005 cost= 13.624166863
Epoch: 0006 cost= 9.942622962
Epoch: 0007 cost= 7.385006424
Epoch: 0008 cost= 5.484037655
Epoch: 0009 cost= 4.019908785

```

渠道 转移规则

```
root@iZulcdurupZ:~/tensorflowtest# rz
root@iZulcdurupZ:~/tensorflowtest# python mlp.py
('Extracting', '/tmp/data/train-images-idx3-ubyte.gz')
('Extracting', '/tmp/data/train-labels-idx1-ubyte.gz')
('Extracting', '/tmp/data/t10k-images-idx3-ubyte.gz')
('Extracting', '/tmp/data/t10k-labels-idx1-ubyte.gz')
I tensorflow/core/common_runtime/local_device.cc:25] Local device intra op parallelism threads: 1
I tensorflow/core/common_runtime/local_session.cc:45] Local session inter op parallelism threads: 1
Epoch: 0001 cost= 184.006677298
Epoch: 0002 cost= 43.614341220
Epoch: 0003 cost= 27.406910379
Epoch: 0004 cost= 18.896572759
Epoch: 0005 cost= 13.624166863
Epoch: 0006 cost= 9.942622962
Epoch: 0007 cost= 7.385006424
Epoch: 0008 cost= 5.484037655
Epoch: 0009 cost= 4.019908785
Epoch: 0010 cost= 2.885964089
Epoch: 0011 cost= 2.160526841
Epoch: 0012 cost= 1.554582004
Epoch: 0013 cost= 1.139619133
Epoch: 0014 cost= 0.852034323
Epoch: 0015 cost= 0.607012027
Optimization Finished!
Accuracy: 0.9475
root@iZulcdurupZ:~/tensorflowtest#

```

转藏到我的图书馆

献花 (0)

分享：

微信 ▼

来自：基陆伯 > 《DeepMind》

以文找文 | 举报

上一篇：TensorFlow人工智能引擎入门教程之七 DNN深度神经网络 的原理 以及 使用

下一篇：TensorFlow人工智能引擎入门教程之九 RNN/LSTM循环神经网络长短期记忆网络使用

猜你喜欢



类似文章

更多

精选文章

- | | |
|---------------------------------|-----------|
| 深入MNIST code测试 | 汉字为何是方块字？ |
| 【机器学习】AlexNet 的tensorflow 实现... | 民间人鬼百忌 |
| 【机器学习】Tensorflow学习笔记 | 红薯N种花样吃法 |
| 深度学习（十四）基于CNN的性别、年龄识... | 身体变化预示好运 |

- python keras (一个超好用的神经网络框...
- 75本书的75句话 : 如果刺痛了心中某个角落,那...
- 基于Theano的深度学习(Deep Learning)框...
- 什么导致孩子学习差别
- 《游仙詩》[清]許式金
- 我很喜欢的一些桌面背景
- 九年上第八单元课件 (二)
- 漂亮的方形毛线坐垫 (有自制图解及过程图新...



文档管理系统

调查问卷与量表的



汉密尔顿手表

中国居民膳食营养



《算命字典》举例

霍兰德职业兴趣测



lol竞猜的首页

lol职业联赛2f的首



人有三个错误不能

苏果lol的首页

- 1 生肖决定你是穷苦命,富贵命..
- 2 贵金属投资服务优选百利好..
- 3 看老股民如何从15万变300万

- | | |
|----------|----------|
| 1 美亚保险官网 | 4 英语学习 |
| 2 美亚保险 | 5 北京口腔医院 |
| 3 公司邮箱 | 6 企业邮箱注册 |

发表评论:

请 [登录](#) 或者 [注册](#) 后再进行评论

社交帐号登录: