

Weekly Report of Research Work

WR-ABS-TEMP-2015A-No.016

汤吉(Ji TANG)

Number: WR-ABS-TEMP-2015A, E-mail: tangji08@hotmail.com

Date: 7/3/2016 - 13/3/2016

March 13, 2016

Contents

1	Work	2
2	Question	2
3	Deep Learning Overview	2
4	About the paper I read	3
5	A example of package " Lasagne"	3

1 Work

1. Relearning the Deep Learning completely
2. Reading the paper ” Regression neural network for error correction in foreign exchange forecasting and trading”
3. Installing the python Deep Learning package ” Lasagne”

2 Question

1. I have already read several papers, but it seems that every paper thinks its own neural network model is the best model for forecasting the Forex Price. So I’ m not sure what kind of neural network model is better.
2. In each week, there are two days off that the gold price doesn’ t change. I don’ t know how to deal with them. If I regard the gold price in the two days off as a constance every week, it’ s obvious to produce the big errors.

3 Deep Learning Overview

Artificial Intelligence，也就是人工智能，就像长生不老和星际漫游一样，是人类最美好的梦想之一。虽然计算机技术已经取得了长足的进步，但是到目前为止，还没有一台电脑能产生“自我”的意识。是的，在人类和大量现成数据的帮助下，电脑可以表现的十分强大，但是离开了这两者，它甚至都不能分辨一个喵星人和一个汪星人。

图灵在 1950 年的论文里，提出图灵试验的设想，即，隔墙对话，你将不知道与你谈话的，是人还是电脑。这无疑给计算机，尤其是人工智能，预设了一个很高的期望值。但是半个世纪过去了，人工智能的进展，远远没有达到图灵试验的标准。这不仅让多年翘首以待的人们，心灰意冷，认为人工智能是忽悠，相关领域是“伪科学”。

但是自 2006 年以来，机器学习领域，取得了突破性的进展。图灵试验，至少不是那么可望而不可及了。至于技术手段，不仅仅依赖于云计算对大数据的并行处理能力，而且依赖于算法。这个算法就是，Deep Learning。借助于 Deep Learning 算法，人类终于找到了如何处理“抽象概念”这个亘古难题的方法。

2012年6月，《纽约时报》披露了Google Brain项目，吸引了公众的广泛关注。这个项目是由著名的斯坦福大学的机器学习教授Andrew Ng和在大规模计算机系统方面的世界顶尖专家JeffDean共同主导，用16000个CPU Core的并行计算平台训练一种称为“深度神经网络”（DNN，Deep Neural Networks）的机器学习模型（内部共有10亿个节点。这一网络自然

是不能跟人类的神经网络相提并论的。要知道，人脑中可是有150多亿个神经元，互相连接的节点也就是突触数更是如银河沙数。曾经有人估算过，如果将一个人的大脑中所有神经细胞的轴突和树突依次连接起来，并拉成一根直线，可从地球连到月亮，再从月亮返回地球），在语音识别和图像识别等领域获得了巨大的成功。

项目负责人之一Andrew称：“我们没有像通常做的那样自己框定边界，而是直接把海量数据投放到算法中，让数据自己说话，系统会自动从数据中学习。”另外一名负责人Jeff则说：“我们在训练的时候从来不会告诉机器说：‘这是一只猫。’系统其实是自己发明或者领悟了“猫”的概念。”

2012年11月，微软在中国天津的一次活动上公开演示了一个全自动的同声传译系统，讲演者用英文演讲，后台的计算机一气呵成自动完成语音识别、英中机器翻译和中文语音合成，效果非常流畅。据报道，后面支撑的关键技术也是DNN，或者深度学习（DL，DeepLearning）。

2013年1月，在百度年会上，创始人兼CEO李彦宏高调宣布要成立百度研究院，其中第一个成立的就是“深度学习研究所”（IDL，Institute of Deep Learning）。

4 About the paper I read

This week, I successfully get the rare paper "Regression neural network for error correction in foreign exchange forecasting and trading" by asking for help on Baidu. In this paper, it mentions a neural network model named General Regression Neural Network (GRNN).

Just like all the papers in this field, it says that the model General Regression Neural Network (GRNN) which it used for forecasting is the best model, because it had the biggest accuracy or so on...

Now, I just decide to use GRNN model firstly for my deep neural network. The next week, I will try to write the code for neural network to forecast the International Gold Price, and test it on the servers in the laboratory.

5 A example of package " Lasagne"

```
1 import lasagne
2 import theano
3 import theano.tensor as T
4
5 # create Theano variables for input and target minibatch
6 input_var = T.tensor4('X')
7 target_var = T.ivector('y')
```

```

8
9 # create a small convolutional neural network
10 from lasagne.nonlinearities import leaky_rectify, softmax
11 network = lasagne.layers.InputLayer((None, 3, 32, 32), input_var)
12 network = lasagne.layers.Conv2DLayer(network, 64, (3, 3),
13                                     nonlinearity=leaky_rectify)
14 network = lasagne.layers.Conv2DLayer(network, 32, (3, 3),
15                                     nonlinearity=leaky_rectify)
16 network = lasagne.layers.Pool2DLayer(network, (3, 3), stride=2, mode='max')
17 network = lasagne.layers.DenseLayer(lasagne.layers.dropout(network, 0.5),
18                                     128, nonlinearity=leaky_rectify,
19                                     W=lasagne.init.Orthogonal())
20 network = lasagne.layers.DenseLayer(lasagne.layers.dropout(network, 0.5),
21                                     10, nonlinearity=softmax)
22
23 # create loss function
24 prediction = lasagne.layers.get_output(network)
25 loss = lasagne.objectives.categorical_crossentropy(prediction, target_var)
26 loss = loss.mean() + 1e-4 * lasagne.regularization.regularize_network_params(
27     network, lasagne.regularization.l2)
28
29 # create parameter update expressions
30 params = lasagne.layers.get_all_params(network, trainable=True)
31 updates = lasagne.updates.nesterov_momentum(loss, params, learning_rate=0.01,
32                                             momentum=0.9)
33
34 # compile training function that updates parameters and returns training loss
35 train_fn = theano.function([input_var, target_var], loss, updates=updates)
36
37 # train network (assuming you've got some training data in numpy arrays)
38 for epoch in range(100):
39     loss = 0
40     for input_batch, target_batch in training_data:
41         loss += train_fn(input_batch, target_batch)
42     print("Epoch %d: Loss %g" % (epoch + 1, loss / len(training_data)))
43
44 # use trained network for predictions
45 test_prediction = lasagne.layers.get_output(network, deterministic=True)
46 predict_fn = theano.function([input_var], T.argmax(test_prediction, axis=1))
47 print("Predicted class for first test input: %r" % predict_fn(test_data[0]))

```