在Deeplearning4j中使用cuDNN

Deeplearning4j支持cuda，但是也支持使用cuDNN实现更进一步的加速。自从版本0.9.1之后，CNNs和LSTMs网络都支持cuDNN加速。

注意： cuDNN并不支持`GravesLSTM`网络层，请考虑使用`LSTM`网络层进行替代。为了使用cuDNN，你首先需要切换Nd4j的后端至CUDA后端。这个可以通过在你项目中的`pom.xml`将`nd4j-native`替换为`nd4j-cuda-7.5`或者`nd4j-cuda-8.0`的方式来实现。理论上，将后端替换为`nd4j-cuda-7.5`或者`nd4j-cuda-8.0`将会自动添加全平台依赖：

```
<dependency>
   <groupId>org.nd4j</groupId>
   <artifactId>nd4j-cuda-7.5-platform</artifactId>
   <version>${nd4j.version}</version>
 </dependency>
```

或者

```
<dependency>
   <groupId>org.nd4j</groupId>
   <artifactId>nd4j-cuda-8.0-platform</artifactId>
   <version>${nd4j.version}</version>
 </dependency>
```

使得DL4J可以加载cuDNN，唯一需要我们做的事情就是将`nd4j-cuda-7.5`或者`nd4j-cuda-8.0`加入到我们的依赖文件中，例如：

```
<dependency>
   <groupId>org.nd4j</groupId>
   <artifactId>nd4j-cuda-7.5-platform</artifactId>
   <version>${nd4j.version}</version>
 </dependency>
```

或者

```
<dependency>
   <groupId>org.nd4j</groupId>
   <artifactId>nd4j-cuda-8.0-platform</artifactId>
```

```
    <version>${nd4j.version}</version>
  </dependency>
```

而cuDNN实际需要的库文件没有打包在里面，所以你需要去英伟达官网下载并且安装适合你平台的包：

- NVIDIA cuDNN

注意，现在只支持cuDNN 6.0。要安装，只需将库提取到本地库使用的系统路径中找到的目录。 最简单的方法是将其与CUDA中的其他库一起放在默认目录中(/usr/local/cuda/lib64/ on Linux, /usr/local/cuda/lib/ on Mac OS X, and C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v7.5\bin\ or C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0\bin\ on Windows).

再次提示，在默认情况下，Deeplearning4j会根据cuDNN使用速度最快的算法，但是内存使用可能会过多，导致一些奇怪的启动异常。当这种情况发生时，尝试使用这种方式来降低内存的使用量，使用`NO_WORKSPACE`(查看更多模式配置)来替代默认的`ConvolutionLayer.AlgoMode.PREFER_FASTEST`，例如：

```
// ...
    new ConvolutionLayer.Builder(h, w)
            .cudnnAlgoMode(ConvolutionLayer.AlgoMode.NO_WORKSPACE)
            // ...
```

原文地址：https://deeplearning4j.org/cudnn

更多文档可以查看 https://github.com/sjsdfg/deeplearning4j-issues。
欢迎star