

# BigDL

Igor Yakushin  
`ivy2@uchicago.edu`

December 27, 2019

# Introduction

- **BigDL** is a library for Deep Learning from Intel
- It is fully integrated with Spark and can work directly with RDDs
- While MLlib has some machine learning algorithms, including fully connected Neural Networks, it does not currently implement Deep Learning models
- BigDL runs on CPU only and relies on MKL for multithreading
- The installation of python version of BigDL is as simple as  
`pip install BigDL`
- BigDL API appears to be very similar to Keras

# LeNet model

- In this lab we use the models supplied with BigDL
- `local_lenet.py` is a standalone (no Spark) BigDL program to train LeNet model for handwritten digits recognition
- Run it on a single compute node with  
`make lenet_local`
- `lenet5.py` is using both Spark and BigDL.
- Run it on a single compute node with  
`make lenet5`

# LeNet model

- LeNet has been used for decades by USPS to recognize handwritten zip codes
- Here is how LeNet model is built in BigDL:

```
def build_model(class_num):  
    model = Sequential()  
    model.add(Reshape([1, 28, 28]))  
    model.add(SpatialConvolution(1, 6, 5, 5))  
    model.add(Tanh())  
    model.add(SpatialMaxPooling(2, 2, 2, 2))  
    model.add(Tanh())  
    model.add(SpatialConvolution(6, 12, 5, 5))  
    model.add(SpatialMaxPooling(2, 2, 2, 2))  
    model.add(Reshape([12 * 4 * 4]))  
    model.add(Linear(12 * 4 * 4, 100))  
    model.add(Tanh())  
    model.add(Linear(100, class_num))  
    model.add(LogSoftMax())  
    return model
```

- MNIST data is loaded into RDD and is preprocessed in Spark before going into BigDL:

```
train_data = get_mnist(sc, "train", options.dataPath)\
    .map(lambda rec_tuple: (normalizer(rec_tuple[0],\
    mnist.TRAIN_MEAN, mnist.TRAIN_STD), rec_tuple[1]))\
    .map(lambda t: Sample.from_ndarray(t[0], t[1]))
```