CNTK Handout

All code located at ~/gpu-edu-workshops

- Exercises at **gpu-edu-workshops/exercises/cntk**
  - For each exercise, replace "FIXME" with the proper code.
- Solutions **at gpu-edu-workshops/exercise_solutions/cntk**
- To run code, execute the **runCNTK.sh** script in the appropriate directory.
  - Output will be piped to screen and to the file **cntkoutput.txt**
- To run multiple times in the same directory you need to **DELETE** the model file each time

Online docs and info here:

- cntk.ai
- github.com/Microsoft/CNTK/wiki

Top-Level Commands

```
command = <colon separated list of the commands CNTK should execute>

modelPath = "<location where the model files will be created>"

imageLayout = "[cudnn|legacy]"

precision = "<arithmetic precision to use>"

deviceId = "<specific device to be used for computation>"

traceLevel = <integer to specify how much output to print, from 0  to
2>

stderr = "<if defined, output goes here rather than to screen>"
```

File Readers

```
reader = [
      readerType = "UCIFastReader"
      file = "<inputfile>"
      features = [
            dim = <number of features>
            start = <column number (0-based) where the features start>
      ]
      labels = [
```

```
            dim = <how many labels for each observation>
            start = <column number (0-based) where the labels start>
            labelDim = <number of possible labels>
            labelMappingFile = "<file that maps labels to line
numbers>"
      ]
]
```

Training

```
<blockName> = [
      action = "train"
      [ define SimpleNetworkBuilder block or NDLNetworkBuilder block ]
      SGD = [
            epochSize = <value of the epoch size>
            minibatchSize = <number of observations per minibatch>
            learningRatesPerMB = <learning rate per minibatch>
            maxEpochs = <maximum number of epochs for training>
      ]
      reader = [ define appropriate reader block, e.g., shown above ]
]
```

Testing

```
<blockName> = [
      action = "test"
      minibatchSize = <size of minibatch for testing>
      reader = [ define appropriate reader block, e.g., shown above ]
]
```

Simple Network Definition

```
SimpleNetworkBuilder = [
      layerSizes = <colon separated list of integers>
      layerTypes = "<Activation types, e.g., Sigmoid>"
```

```
    trainingCriterion = "<function to be minimized, e.g.,
SquareError>"
    evalCriterion = "<function showing how well the network is
performing, e.g., ErrorPrediction>"
]
```

Network Definition Language

```
# use InputValue for defining data that comes from the input files
<variable name> = InputValue( <size> )

# any quantities that will be learned by the model
<weights> = LearnableParameter( <dim1>, <dim2> )
<bias> = LearnableParameter( <dim1> )

# computations in the neural network
# multiply weights by features
<t1> = Times( <weights>, <features> )
# add a bias term
<b1> = Plus( t1, bias )
# activation using Sigmoid, also can use RectifiedLinear, Tanh, Log
<act1> = Sigmoid( <b1> )

# special nodes
```

Network Plot

```
<blockName> = [
    action = "plot"
    outputdotFile = "<filename of dot file>"
    outputFile = "<image file name>"
    renderCmd = "/usr/bin/dot -Tjpg <IN> -o<OUT>" #bugs with gcc<4.9
]
```