Observers

Usage

Observers are a small framework that allow users to attach code to the execution of SimpleNets and Operators.

An example of an Observer is the `TimeObserver`, used as follows:

C++

```
unique_ptr<TimeObserver<NetBase>> net_ob =
    make_unique<TimeObserver<NetBase>>(net.get());
auto* ob = net->AttachObserver(std::move(net_ob));
net->Run();
LOG(INFO) << "av time children: " << ob->average_time_children();
LOG(INFO) << "av time: " << ob->average_time();
```

Python

```
model.net.AttachObserver("TimeObserver")
ws.RunNet(model.net)
ob = model.net.GetObserver("TimeObserver")

print("av time children:", ob.average_time_children())
print("av time:", ob.average_time())
```

Histogram Observer

Creates a histogram for the values of weights and activations

```
model.net.AddObserver("HistogramObserver",
                      "histogram.txt", # filename
                      2014, # number of bins in histogram
                      32 # Dumping frequency
                      )
ws.RunNet(model.net)
```

This will generate a histogram for the activations and store it in histogram.txt

Implementing An Observer

To implement an observer you must inherit from `ObserverBase` and implement the `Start` and `Stop` functions.

Observers are instantiated with a `subject` of a generic type, such as a `Net` or `Operator`. The observer framework is built to be generic enough to "observe" various other types, however.