All Rights Reserved

kptan86@gmail.com

Programming Implementation

Minimal OOP implementation of fully connected neural network

```
class Net {
    add_layer(Layer)
} // end class Net

class Layer {
    add_neuron(Neuron)
} // end class Layer
```

```
class Neuron {
    set_activation_func(Func)
    set/get children(Neuron)
    set/get parents(Neuron)
    init_weights()
    forward()
    backprop()
    update_weights()
    toggle()
} // end class Neuron
```

Sample usage interface (pseudo-code)

```
## Architecting ##
net = Net(params={'learning_rate': 0.01, 'momentum': 0.02, ...})
net.add_layers([
    Layer(type="input",       ).add_neurons([Neuron("logistic")]*3 )),
    Layer(type="fully_connect").add_neurons([Neuron("logistic")]*10)),
    Layer(type="fully_connect").add_neurons([Neuron("logistic")]*5 )),
    Layer(type="output",      ).add_neurons([Neuron("logistic")]*3 )),
])

## Training ##
for x, y in data:
    yp = net.forward(x)
    loss = loss_func(yp, y)
    net.backprop(loss)
    net.update_weight()
# end for
```

Demo on simple implementation

<< DEMO >>

DO NOT DISTRIBUTE

DO NOT DISTRIBUTE

All Rights Reserved

kptan86@gmail.com

Deep Learning Frameworks

A list of deep learning frameworks: Tensorflow, PyTorch, Keras, MXNet, The Microsoft Cognitive Toolkit, Caffe, Deeplearning4j, Chainer, etc

Primary features

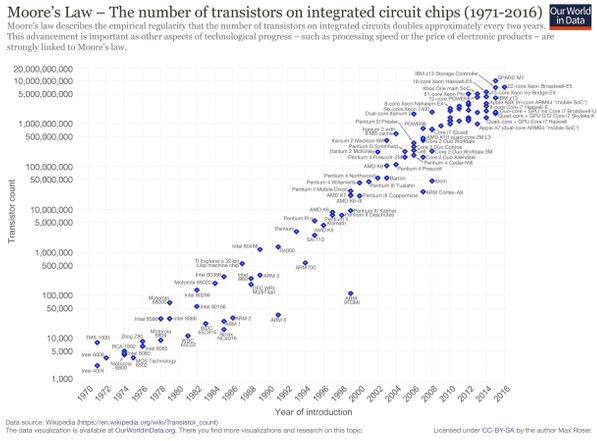
- Automatic differentiation
- Multiprocessing / Distributed computing
- **GPU acceleration**
- Supported features (e.g. activation functions (ReLU), dropout etc)

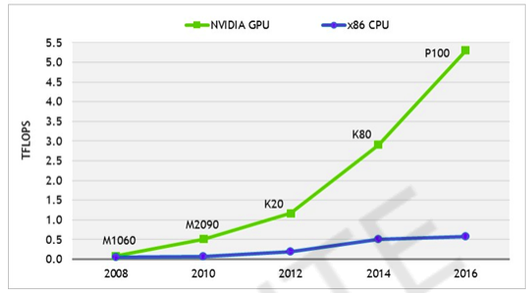
Secondary features

- Layering specification
- Documenation
- Programming language
- Model serving capability
- ...

We will use PyTorch here

Hardware solution





Demo on Pytorch implementation

<< DEMO >>

DO NOT DISTRIBUTE
DO NOT DISTRIBUTE