

# A brief introduction of MXNET

---

by zhangkun

# What's mxnet:

- Flexible and Efficient Deep Learning Library on Heterogeneous Distributed Systems.
- MXNet is a deep learning framework designed for both efficiency and flexibility. It allows you to mix the flavours of deep learning programs together to maximize the efficiency and your productivity.

# how to create a network:

1. `data = mx.symbol.Variable('data')`
2. `fc1 = mx.symbol.FullyConnected(data = data, name='fc1', num_hidden=128)`
3. `act1 = mx.symbol.Activation(data = fc1, name='relu1', act_type="relu")`
4. `fc2 = mx.symbol.FullyConnected(data = act1, name = 'fc2', num_hidden = 64)`
5. `act2 = mx.symbol.Activation(data = fc2, name='relu2', act_type="relu")`
6. `fc3 = mx.symbol.FullyConnected(data = act2, name='fc3', num_hidden=10)`
7. `mlp = mx.symbol.SoftmaxOutput(data = fc3, name = 'softmax')`

# how to get data:

```
1.     val = mx.io.ImageRecordIter(  
2.         path_imgrec = args.data_dir + "test.rec",  
3.         mean_img    = args.data_dir + "mean.bin",  
4.         rand_crop   = False,  
5.         rand_mirror = False,  
6.         data_shape  = data_shape,  
7.         batch_size  = args.batch_size,  
8.         num_parts   = kv.num_workers,  
9.         part_index  = kv.rank)
```

# how to training(1):

```
1. model = mx.model.FeedForward(  
2.     ctx          = devs,  
3.     symbol       = network,  
4.     num_epoch    = args.num_epochs,  
5.     learning_rate = args.lr,  
6.     momentum     = 0.9,  
7.     wd           = 0.00001,  
8.     initializer  = mx.init.Xavier(factor_type="in", magnitude=2.34),  
9.     **model_args)  
10. model.fit(  
11.     X          = train,  
12.     eval_data   = val,  
13.     kvstore     = kv,  
14.     batch_end_callback = mx.callback.Speedometer(args.batch_size, 50),  
15.     epoch_end_callback = checkpoint)
```



## how to training(2):

- `mx.model.FeedForward.create(X, y=None, ctx=None, num_epoch=None, epoch_size=None, optimizer='sgd', initializer=<mxnet.initializer.Uniform object>, eval_data=None, eval_metric='acc', epoch_end_callback=None, batch_end_callback=None, kvstore='local', logger=None, work_load_list=None, **kwargs)`

# how to save,load and predict

- model:
- `mxnet.model.FeedForward.save(**)`
- `mxnet.model.FeedForward.load(**)`
- symbol:
- `mxnet.symbol.save(**)`
- `mxnet.symbol.load(**)`
- predict:
- `mxnet.model.FeedForward.predict(**)`

# material:

- <http://mxnet.readthedocs.org/en/latest/index.html>
- <https://github.com/dmlc/mxnet>
- <http://www.52cs.org/?p=639>



thank you