For 3d.i. (before modification)

layer\_defs = [];

layer\_defs.push({type:'input', out\_sx:24, out\_sy:24, out\_depth:1});

layer\_defs.push({type:'conv', sx:5, filters:8, stride:1, pad:2, activation:'relu'});

layer\_defs.push({type:'pool', sx:2, stride:2});

layer\_defs.push({type:'conv', sx:5, filters:16, stride:1, pad:2, activation:'relu'});

layer\_defs.push({type:'pool', sx:3, stride:3});

layer\_defs.push({type:'softmax', num\_classes:10});

net = new convnetjs.Net();

net.makeLayers(layer\_defs);

trainer = new convnetjs.SGDTrainer(net, {method:'adadelta', batch\_size:20, l2\_decay:0.001});

For 3d.i. (after modification)

layer\_defs = [];

layer\_defs.push({type:'input', out\_sx:24, out\_sy:24, out\_depth:1});

layer\_defs.push({type:'fc', num\_neurons:500, activation:'relu'});

layer\_defs.push({type:'softmax', num\_classes:10});

net = new convnetjs.Net();

net.makeLayers(layer\_defs);

trainer = new convnetjs.SGDTrainer(net, {method:'adadelta', batch\_size:20, l2\_decay:0.001});

For 3d.ii.

layer\_defs = [];  
layer\_defs.push({type:'input', out\_sx:24, out\_sy:24, out\_depth:1});  
layer\_defs.push({type:'conv', sx:5, filters:8, stride:1, pad:2, activation:'relu'});  
layer\_defs.push({type:'pool', sx:2, stride:2});  
layer\_defs.push({type:'softmax', num\_classes:10});  
net = new convnetjs.Net();  
net.makeLayers(layer\_defs);  
trainer = new convnetjs.SGDTrainer(net, {method:'adadelta',  
batch\_size:20, l2\_decay:0.001});

For 3d.iii.

First modification:

layer\_defs = [];

layer\_defs.push({type:'input', out\_sx:24, out\_sy:24, out\_depth:1});

layer\_defs.push({type:'conv', sx:5, filters:8, stride:1, pad:2, activation:'relu'});

layer\_defs.push({type:'pool', sx:2, stride:2});

layer\_defs.push({type:'conv', sx:5, filters:16, stride:1, pad:2, activation:'relu'});

layer\_defs.push({type:'pool', sx:3, stride:2});

layer\_defs.push({type:'softmax', num\_classes:10});

net = new convnetjs.Net();

net.makeLayers(layer\_defs);

trainer = new convnetjs.SGDTrainer(net, {method:'adadelta', batch\_size:20, l2\_decay:0.001});

Second modification:

layer\_defs = [];

layer\_defs.push({type:'input', out\_sx:24, out\_sy:24, out\_depth:1});

layer\_defs.push({type:'conv', sx:5, filters:8, stride:1, pad:2, activation:'relu'});

layer\_defs.push({type:'pool', sx:2, stride:2});

layer\_defs.push({type:'conv', sx:5, filters:16, stride:1, pad:2, activation:'relu'});

layer\_defs.push({type:'pool', sx:3, stride:3});

layer\_defs.push({type:'fc', num\_neurons:20, activation:'relu'});

layer\_defs.push({type:'softmax', num\_classes:10});

net = new convnetjs.Net();

net.makeLayers(layer\_defs);

trainer = new convnetjs.SGDTrainer(net, {method:'adadelta', batch\_size:20, l2\_decay:0.001});