

**BAN 5743, Exercise 8 (10 Points) Solution**  
**You must do it alone (it is not a group activity)**

Open the program named **DLMS01E01.sas**. This program provides a model template that is similar to the deep learning model trained in the class demonstration. The following steps request that you expand the number of hidden layers by two, adding **one next to the input layer and one next to the output layer** as described below.

Make following changes:

1. Replace the hyperbolic tangent (TANH) activation functions in **each of the 7 existing hidden layer** with ELU.
2. Add another hidden layer **right after the input layer**. Connect this hidden layer to both the input layer and the next hidden layer. Construct this hidden layer such that the layer contains the following characteristics:
  - a. has 40 hidden neurons
  - b. uses the exponential linear (ELU) activation function
  - c. includes a dropout rate of .05 percent
  - d. uses Xavier initialization for the hidden weights
  - e. normalizes the output of the layer using batch normalization. That is, apply batch normalization before the output of this layer has a nonlinear transformation applied to the information.
3. Add another hidden layer just **before the output layer**. Connect this hidden layer to both the previous hidden layer and the output layer. Construct this hidden layer such that the layer contains the following characteristics:
  - a. has 40 hidden neurons
  - b. uses the exponential linear (ELU) activation function
  - c. includes a dropout rate of .05 percent
  - d. uses Xavier initialization for the hidden weights

Compare your model with the model shown in the demonstration. Is it performing any better or worse than the one in the demonstration? Is the model more complex than the one shown in demonstration?

**Code Solution:**

```

/*****
/* Uncomment and run the below three      */
/* lines of code if the DeepLearn        */
/* action set has not been loaded.        */
/*****
/* proc cas; */
/* loadactionset 'DeepLearn'; */
/* run; */

```

```

/* WithBatch Normalization */
proc cas;
BuildModel / modeltable={name='BatchDLNN',
replace=1} type = 'DNN';

/*          INPUT Layer          */
AddLayer / model='BatchDLNN' name='data'
layer={type='input' STD='STD' dropout=.05};

/*          FIRST HIDDEN LAYER          */
AddLayer / model='BatchDLNN' name='HLayer1'
layer={type='FULLCONNECT' n=40 act='ELU'
init='xavier' dropout=.05}
srcLayers={'data'};

/*          SECOND HIDDEN LAYER          */
AddLayer / model='BatchDLNN' name='HLayer2'
layer={type='FULLCONNECT' n=30
act='identity' init='xavier'
includeBias=False} srcLayers={'HLayer1'};
AddLayer / model='BatchDLNN'
name='BatchLayer2' layer={type='BATCHNORM'
act='ELU'} srcLayers={'HLayer2'};

/*          THIRD HIDDEN LAYER          */
AddLayer / model='BatchDLNN' name='HLayer3'
layer={type='FULLCONNECT' n=20
act='identity' init='xavier'
includeBias=False }
srcLayers={'BatchLayer2'};
AddLayer / model='BatchDLNN'
name='BatchLayer3' layer={type='BATCHNORM'
act='ELU'} srcLayers={'HLayer3'};

/*          FOURTH HIDDEN LAYER          */

```

```

AddLayer / model='BatchDLNN' name='HLayer4'
layer={type='FULLCONNECT' n=10
act='identity' init='xavier'
includeBias=False }
srcLayers={'BatchLayer3'};
AddLayer / model='BatchDLNN'
name='BatchLayer4' layer={type='BATCHNORM'
act='ELU'} srcLayers={'HLayer4'};

/*          FIFTH HIDDEN LAYER          */
AddLayer / model='BatchDLNN' name='HLayer5'
layer={type='FULLCONNECT' n=5 act='identity'
init='xavier' includeBias=False }
srcLayers={'BatchLayer4'};
AddLayer / model='BatchDLNN'
name='BatchLayer5' layer={type='BATCHNORM'
act='ELU'} srcLayers={'HLayer5'};

/*          SIXTH HIDDEN LAYER          */
AddLayer / model='BatchDLNN' name='HLayer6'
layer={type='FULLCONNECT' n=10
act='identity' init='xavier'
includeBias=False}
srcLayers={'BatchLayer5'};
AddLayer / model='BatchDLNN'
name="BatchLayer6" layer={type='BATCHNORM'
act='ELU'} srcLayers={'HLayer6'};

/*          SEVENTH HIDDEN LAYER          */
AddLayer / model='BatchDLNN' name='HLayer7'
layer={type='FULLCONNECT' n=20
act='identity' init='xavier'
includeBias=False }
srcLayers={'BatchLayer6'};

```

```

AddLayer / model='BatchDLNN'
name="BatchLayer7" layer={type='BATCHNORM'
act='ELU'} srcLayers={'HLayer7'};

/*          EIGHTH HIDDEN LAYER          */
AddLayer / model='BatchDLNN' name='HLayer8'
layer={type='FULLCONNECT' n=30
act='identity' init='xavier'
includeBias=False }
srcLayers={'BatchLayer7'};
AddLayer / model='BatchDLNN'
name="BatchLayer8" layer={type='BATCHNORM'
act='ELU'} srcLayers={'HLayer8'};

/*          NINTH HIDDEN LAYER          */
AddLayer / model='BatchDLNN' name='HLayer9'
layer={type='FULLCONNECT' n=40
act='identity' init='xavier'
includeBias=False dropout=.05}
srcLayers={'BatchLayer8'};
AddLayer / model='BatchDLNN'
name="BatchLayer9" layer={type='BATCHNORM'
act='ELU'} srcLayers={'HLayer9'};

AddLayer / model='BatchDLNN' name='outlayer'
layer={type='output' act='LOGISTIC'}
srcLayers={'BatchLayer9'};
run;

```

```
proc cas;
```

```
dlTrain /table='Train_Develop'  
model='BatchDLNN'  
  
modelWeights={name='BatchTrainedWeights_d',  
replace=1}  
  
bestweights={name='bestbatchweights',  
replace=1}  
    inputs={ 'AcctAge',  
              'DDABal',  
              'CashBk',  
              'Checks',  
              'NSFAmt',  
              'Phone',  
              'Teller',  
              'SavBal',  
              'ATMAmt',  
              'POS',  
              'POSAmt',  
              'CDBal',  
              'IRABal',  
              'LOCBal',  
              'ILSBal',  
              'MMBal',  
              'MMCred',  
              'MTGBal',  
              'CCBal',  
              'CCPurc',  
              'Income',  
              'LORes',  
              'HMVal',  
              'Age',  
              'CRScore',  
              'Dep',  
              'DepAmt',
```

```
'InvBal',  
'DDA',  
'DirDep',  
'NSF',  
'Sav',  
'ATM',  
'CD',  
'IRA',  
'LOC',  
'ILS',  
'MM',  
'MTG',  
'CC',  
'SDB',  
'HMOwn',  
'Moved',  
'InArea',  
'Inv'  
}  
nominals={ 'INS',  
'DDA',  
'DirDep',  
'NSF',  
'Sav',  
'ATM',  
'CD',  
'IRA',  
'LOC',  
'ILS',  
'MM',  
'MTG',  
'CC',  
'SDB',  
'HMOwn',  
'Moved',
```

```
        'InArea',  
        'Inv'  
    }  
    ValidTable='Valid_Develop'  
    target='INS'  
    optimizer={minibatchsize=60,  
algorithm={method='ADAM', lrpolicy='Step',  
gamma=0.5, Stepsize=15  
        beta1=0.9, beta2=0.999,  
learningrate=.001}  
    regL1=0.003,  
    regL2=0.002,  
    maxepochs=100}  
    seed=12345  
;  
run;
```