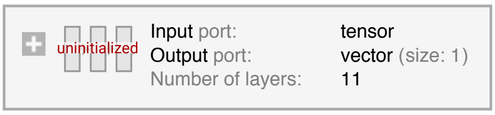
```
In[•]:= ABMInputs800 = Import[
           "/Users/thorsilver/Downloads/ABM outputs1/LPtau800runs_GEMSA_inputs.csv"];

        ABMOutputs800 =
          Import["/Users/thorsilver/Downloads/ABM outputs1/LPtau800runs GEMSA
             outputs only.csv"];

In[•]:= ABMOutputs800 = Function[x, x / 1000] /@ ABMOutputs800;
        ABMAssoc800  = AssociationThread[ABMInputs800 → Flatten[ABMOutputs800]];
        ABMnewData800 = Dataset[ABMAssoc800];
        ABMNormal800 = Normal[ABMAssoc800];
        ABMNormalRandom = RandomSample[ABMNormal800];
        ABMtrain800 = TakeDrop[ABMNormal800, 640];
        ABMtest800 = ABMtrain800[[2]];
        ABMtraining800 = ABMtrain800[[1]];
        trainDevSplit800 = TakeDrop[ABMtraining800, 512];
        finalTrain800 = trainDevSplit800[[1]];
        finalDev800 = trainDevSplit800[[2]];
        finaltest800 = ABMtest800;

In[•]:= Length[finalDev800]
        Length[finalTrain800]
        Length[finaltest800]

Out[•]= 128

Out[•]= 512

Out[•]= 160
```

NN Experiments

```
In[•]:= netSimple = NetChain[{10, BatchNormalizationLayer[],
           Tanh, 15, 15, 15, 15, 15, BatchNormalizationLayer[], Tanh, 1}]
        trainedNetSimple = NetTrain[netSimple, finalTrain800, All,
          ValidationSet → finaltest800, TargetDevice → "GPU",
          MaxTrainingRounds → 500, Method → {"ADAM", "L2Regularization" → 0.05}]
```

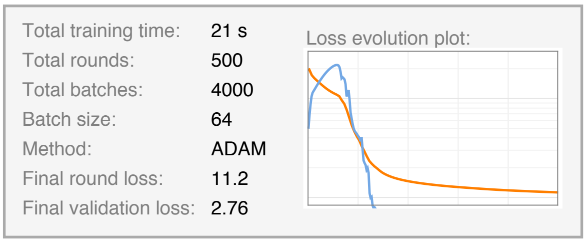
Out[•]= NetChain[



```
Input port:          tensor
Output port:         vector (size: 1)
Number of layers:    11
```

]

Out[•]= NetTrainResultsObject[

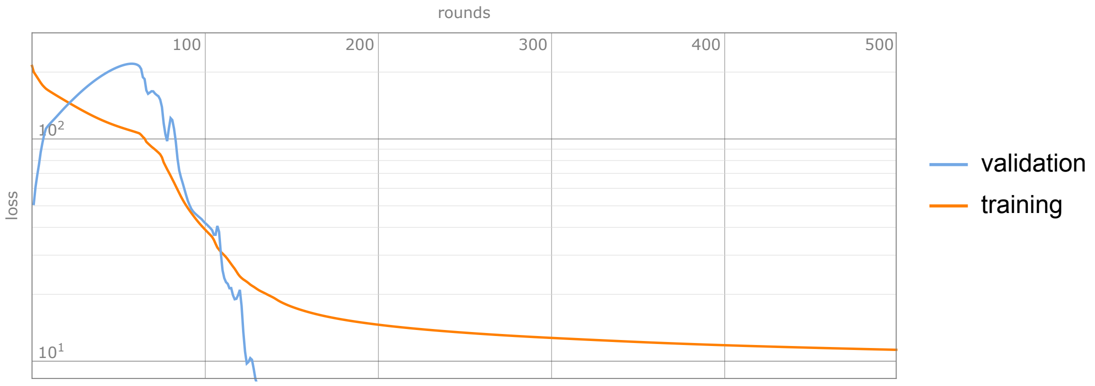


```
Total training time:    21 s
Total rounds:           500
Total batches:          4000
Batch size:             64
Method:                 ADAM
Final round loss:       11.2
Final validation loss:  2.76
```

]

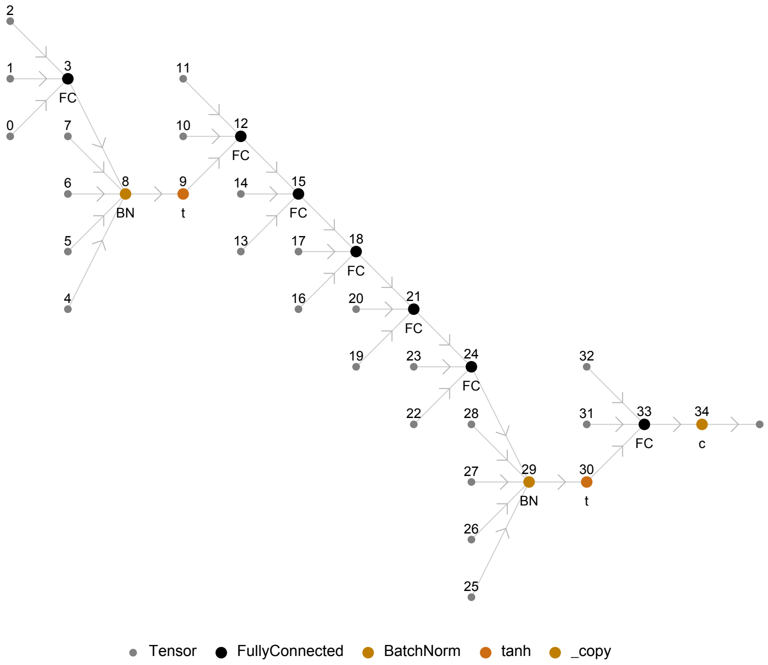
```
In[•]:= trainedNetSimple["LossEvolutionPlot"]
        trainedNetSimple["LowestValidationLoss"]
        NetInformation[trainedNetSimple["TrainedNet"], "MXNetNodeGraphPlot"]
        NetInformation[trainedNetSimple["TrainedNet"], "SummaryGraphic"]
```

Out[•]=

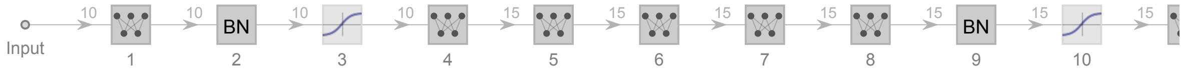


Out[•]= 2.43338

Out[•]=



Out[•]=

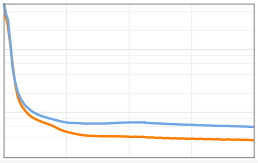


```
In[•]:= netSimple2 = NetChain[{10, BatchNormalizationLayer[],
           Tanh, 50, 50, 50, 50, 50, 50, BatchNormalizationLayer[], Tanh, 1}]
        trainedNetSimple2 = NetTrain[netSimple2, finaltest800, All,
          ValidationSet → finaltest800, TargetDevice → "GPU", MaxTrainingRounds → 4000,
          Method → {"ADAM", "LearningRate" → 0.0005, "L2Regularization" → 0.05}]
```

```
Out[•]= NetChain[  uninitialized       Input                     tensor
                                   1   LinearLayer               vector (size: 10)
                                   2   BatchNormalizationLayer   vector (size: 10)
                                   3   Tanh                      vector (size: 10)
                                   4   LinearLayer               vector (size: 50)
                                   5   LinearLayer               vector (size: 50)
                                   6   LinearLayer               vector (size: 50)
                                   7   LinearLayer               vector (size: 50)
                                   8   LinearLayer               vector (size: 50)
                                   9   LinearLayer               vector (size: 50)
                                   10  BatchNormalizationLayer   vector (size: 50)
                                   11  Tanh                      vector (size: 50)
                                   12  LinearLayer               vector (size: 1)
                                       Output                    vector (size: 1)    ]
```

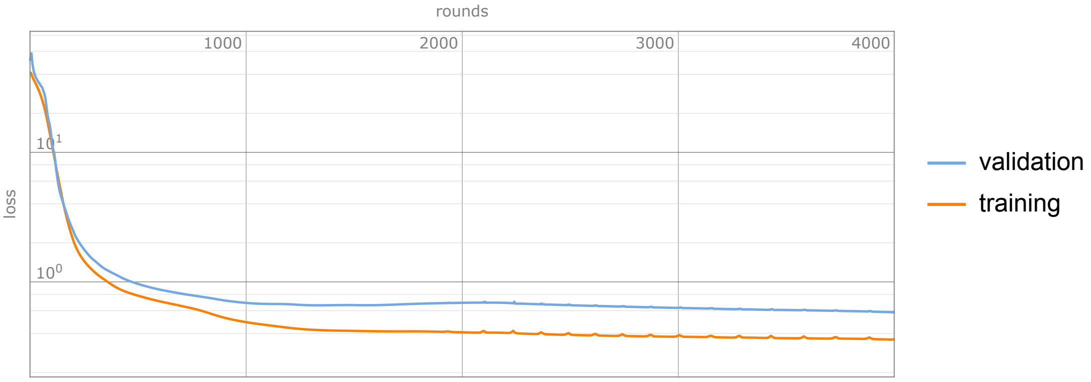
```
Out[•]= NetTrainResultsObject[  Total training time:    2.5 min
                                Total rounds:           4000
                                Total batches:          12 000
                                Batch size:             64
                                Method:                 ADAM
                                Final round loss:       0.359
                                Final validation loss:  0.583     ]
```

Loss evolution plot:



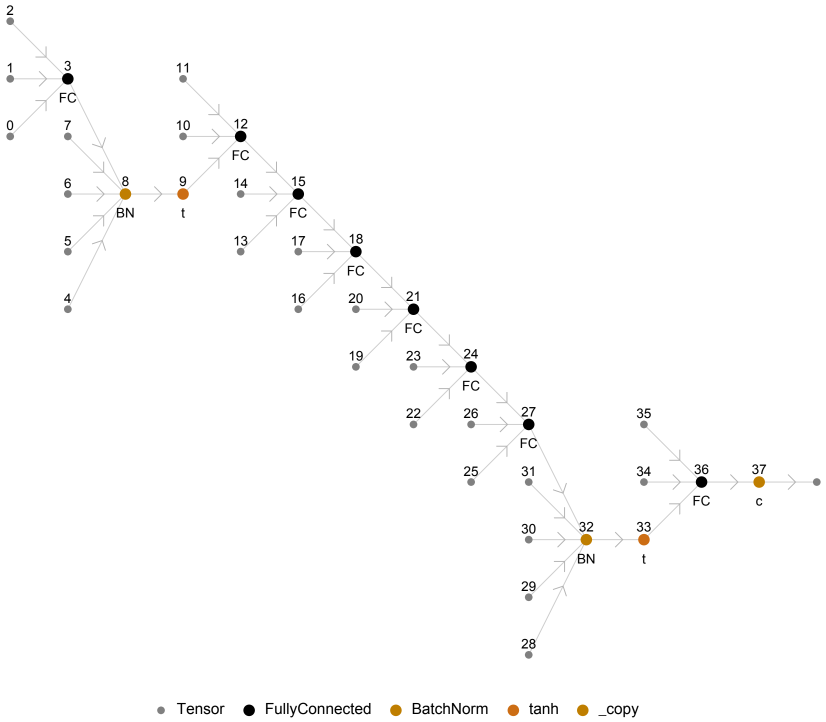
In[•]:=
```
trainedNetSimple2["LossEvolutionPlot"]
trainedNetSimple2["LowestValidationLoss"]
NetInformation[trainedNetSimple2["TrainedNet"], "MXNetNodeGraphPlot"]
NetInformation[trainedNetSimple2["TrainedNet"], "SummaryGraphic"]
```

Out[•]=

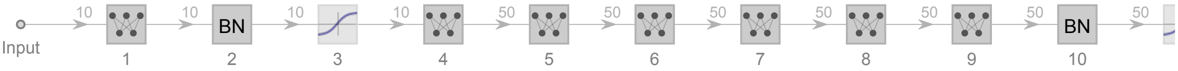


Out[•]=
```
0.58319
```

Out[•]=

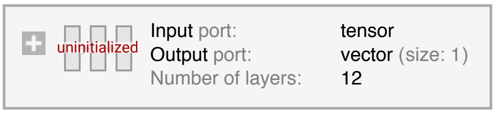


Out[•]=



```
In[•]:= netSimple2 = NetChain[{10, BatchNormalizationLayer[],
           Tanh, 50, 50, 50, 50, 50, 50, BatchNormalizationLayer[], Tanh, 1}]
        trainedNetSimple2 = NetTrain[netSimple2, finalTrain800, All,
          ValidationSet → finaltest800, TargetDevice → "GPU", MaxTrainingRounds → 4000,
          Method → {"ADAM", "LearningRate" → 0.0005, "L2Regularization" → 0.05}]
```

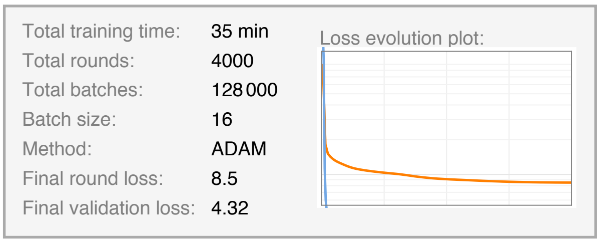
Out[•]= NetChain[



```
Input port:          tensor
Output port:         vector (size: 1)
Number of layers:    12
```

]

Out[•]= NetTrainResultsObject[

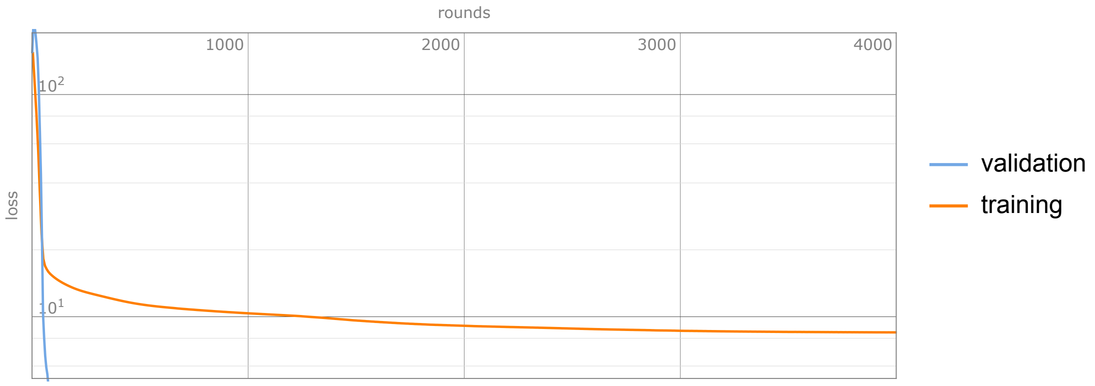


```
Total training time:     35 min
Total rounds:            4000
Total batches:           128 000
Batch size:              16
Method:                  ADAM
Final round loss:        8.5
Final validation loss:   4.32
```

]

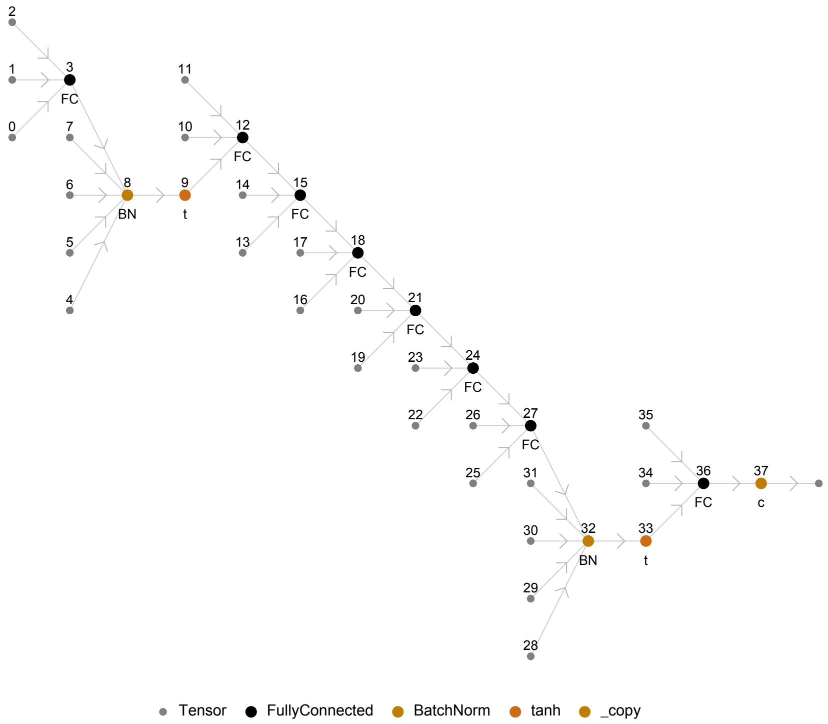
```
In[•]:= trainedNetSimple2["LossEvolutionPlot"]
        trainedNetSimple2["LowestValidationLoss"]
        NetInformation[trainedNetSimple2["TrainedNet"], "MXNetNodeGraphPlot"]
        NetInformation[trainedNetSimple2["TrainedNet"], "SummaryGraphic"]
```

Out[•]=

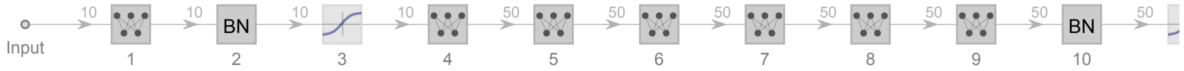


```
Out[•]= 4.04817
```

Out[•]=



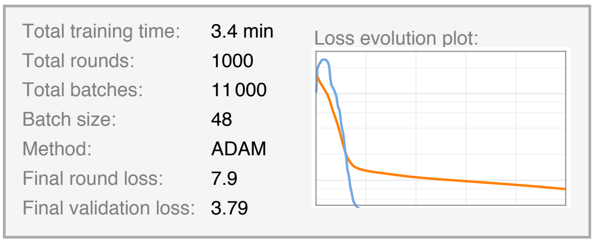
Out[•]=



```
In[∘]:= netSimple3 = NetChain[{10, BatchNormalizationLayer[], Tanh,
           50, 50, 50, 50, 50, 50, 50, BatchNormalizationLayer[], Tanh, 1}]
        trainedNetSimple3 = NetTrain[netSimple3, finalTrain800, All,
          ValidationSet → finaltest800, TargetDevice → "GPU", MaxTrainingRounds → 1000,
          Method → {"ADAM", "LearningRate" → 0.0005, "L2Regularization" → 0.05}]
```

```
Out[∘]= NetChain[ uninitialized   Input port:         tensor
                                  Output port:        vector (size: 1)
                                  Number of layers:   13             ]
```

Out[∘]= NetTrainResultsObject[



```
Total training time:    3.4 min
Total rounds:           1000
Total batches:          11 000
Batch size:             48
Method:                 ADAM
Final round loss:       7.9
Final validation loss:  3.79
```

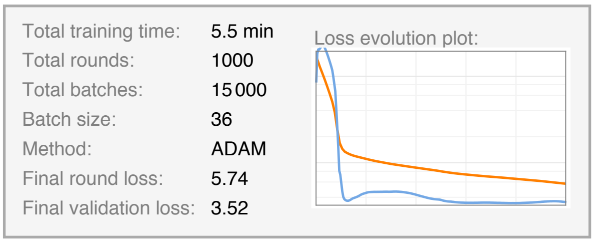
]

```
In[∘]:= trainedNetSimple3["LowestValidationLoss"]

Out[∘]= 2.88707
```

```
In[∘]:= netSimple4 = NetChain[{10, BatchNormalizationLayer[], Tanh, 50,
           50, 50, 50, 50, 50, 50, 50, BatchNormalizationLayer[], Tanh, 1}]
        trainedNetSimple4 = NetTrain[netSimple4, finalTrain800, All,
          ValidationSet → finaltest800, TargetDevice → "GPU", MaxTrainingRounds → 1000,
          Method → {"ADAM", "LearningRate" → 0.0005, "L2Regularization" → 0.04}]
```

```
Out[∘]= NetChain[ uninitialized   Input port:         tensor
                                  Output port:        vector (size: 1)
                                  Number of layers:   14             ]
```

Out[∘]= NetTrainResultsObject[



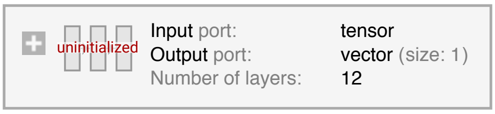
```
Total training time:    5.5 min
Total rounds:           1000
Total batches:          15 000
Batch size:             36
Method:                 ADAM
Final round loss:       5.74
Final validation loss:  3.52
```

]

```
In[∘]:= trainedNetSimple4["LowestValidationLoss"]

Out[∘]= 3.42597
```

```
In[●]:= netSimple5 = NetChain[{10, BatchNormalizationLayer[],
           Tanh, 15, 15, 15, 15, 15, 15, BatchNormalizationLayer[], Tanh, 1}]
        trainedNetSimple5 = NetTrain[netSimple5, finalTrain800, All,
          ValidationSet → finaltest800, TargetDevice → "GPU", MaxTrainingRounds → 4000,
          Method → {"ADAM", "LearningRate" → 0.0005, "L2Regularization" → 0.1}]
```

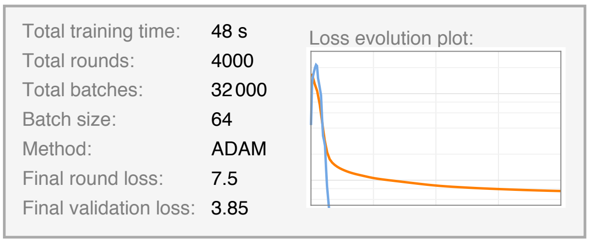
Out[●]= NetChain[



```
uninitialized   Input port:          tensor
                Output port:         vector (size: 1)
                Number of layers:    12
```

]

Out[●]= NetTrainResultsObject[

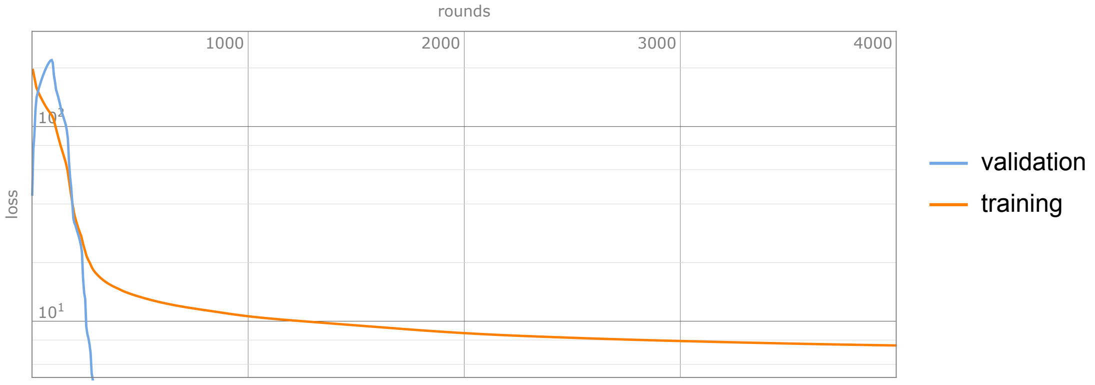


```
Total training time:    48 s
Total rounds:           4000
Total batches:          32 000
Batch size:             64
Method:                 ADAM
Final round loss:       7.5
Final validation loss:  3.85
```

]

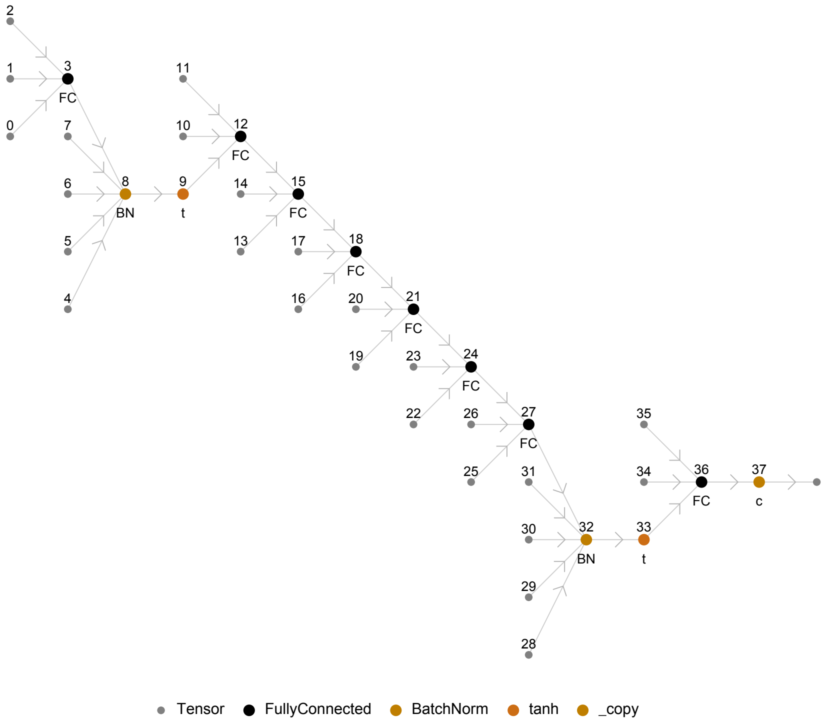
In[•]:=
```
trainedNetSimple5["LossEvolutionPlot"]
trainedNetSimple5["LowestValidationLoss"]
NetInformation[trainedNetSimple5["TrainedNet"], "MXNetNodeGraphPlot"]
NetInformation[trainedNetSimple5["TrainedNet"], "SummaryGraphic"]
```

Out[•]=

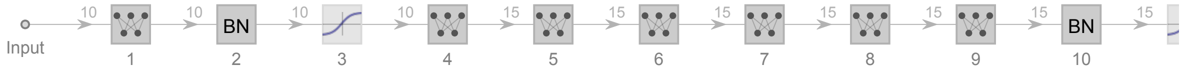


Out[•]=
```
2.64168
```

Out[•]=



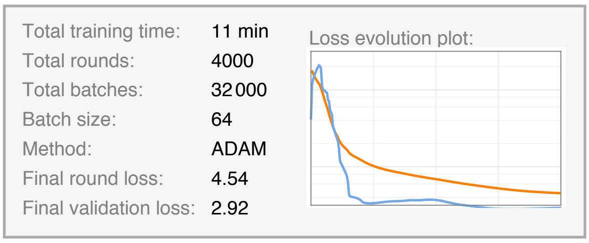
Out[•]=



```
In[•]:= netSimple6 = NetChain[{10, BatchNormalizationLayer[], Tanh,
           15, 15, 15, 15, 15, 15, 15, BatchNormalizationLayer[], Tanh, 1}]
        trainedNetSimple6 = NetTrain[netSimple6, finalTrain800, All,
          ValidationSet → finaltest800, TargetDevice → "GPU", MaxTrainingRounds → 4000,
          Method → {"ADAM", "LearningRate" → 0.0003, "L2Regularization" → 0.035}]
```

```
Out[•]= NetChain[ uninitialized  Input port:         tensor
                                 Output port:        vector (size: 1)
                                 Number of layers:   13 ]
```

Out[•]= NetTrainResultsObject[

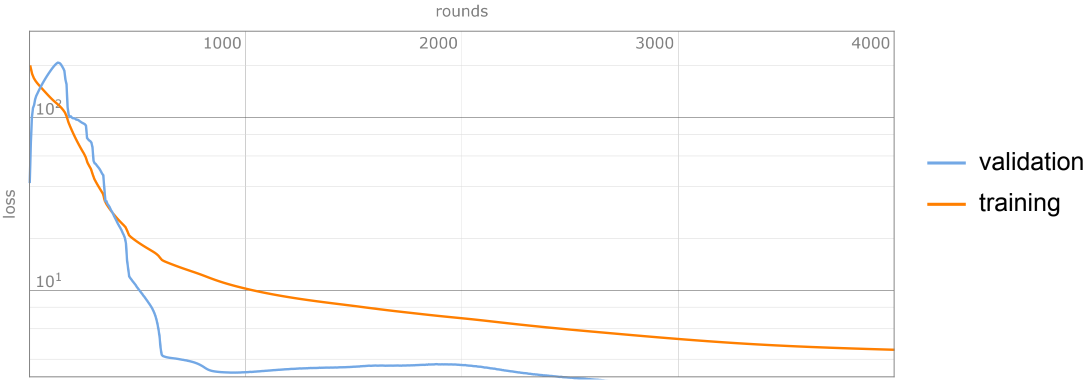


```
Total training time:    11 min
Total rounds:           4000
Total batches:          32 000
Batch size:             64
Method:                 ADAM
Final round loss:       4.54
Final validation loss:  2.92
```

]

```
In[•]:= trainedNetSimple6["LossEvolutionPlot"]
        trainedNetSimple6["LowestValidationLoss"]
```

Out[•]=

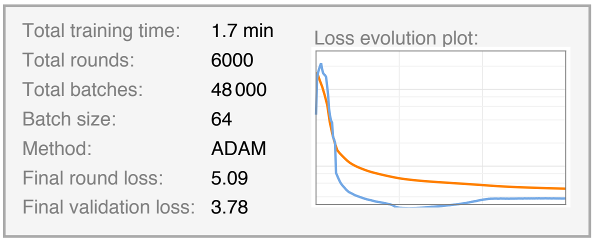


```
Out[•]= 2.82514
```

```
In[•]:= netSimple7 = NetChain[{10, BatchNormalizationLayer[], Tanh, 15, 15, 15,
           15, 15, 15, 15, 15, 15, 15, 15, BatchNormalizationLayer[], Tanh, 1}]
        trainedNetSimple7 = NetTrain[netSimple7, finalTrain800, All,
          ValidationSet → finaltest800, TargetDevice → "GPU", MaxTrainingRounds → 6000,
          Method → {"ADAM", "LearningRate" → 0.0003, "L2Regularization" → 0.03}]
```

```
Out[•]= NetChain[ uninitialized  Input port:         tensor
                                 Output port:        vector (size: 1)
                                 Number of layers:   17 ]
```

Out[•]= NetTrainResultsObject[



```
Total training time:    1.7 min
Total rounds:           6000
Total batches:          48 000
Batch size:             64
Method:                 ADAM
Final round loss:       5.09
Final validation loss:  3.78
```

]

In[●]:=
```
trainedNetSimple7["LowestValidationLoss"]
```

Out[●]=
```
2.82401
```

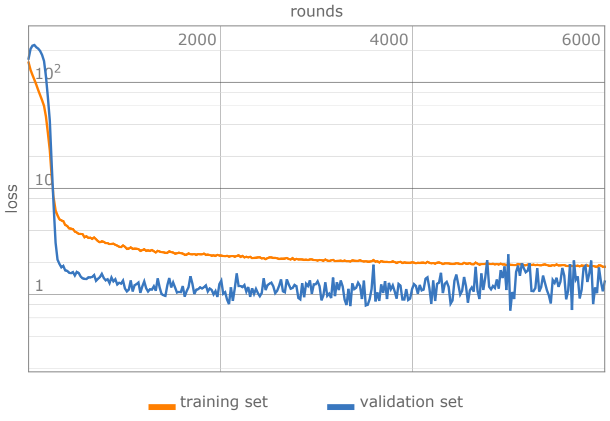
In[633]:=
```
netSimple8 = NetChain[{BatchNormalizationLayer[], Tanh, 50, 50, 50,
    50, 50, 50, 50, 50, 50, 50, 50, BatchNormalizationLayer[], Tanh, 1}]
trainedNetSimple8 = NetTrain[netSimple8, finalTrain800, All,
   ValidationSet → finaltest800, TargetDevice → "CPU", MaxTrainingRounds → 6000,
   Method → {"ADAM", "LearningRate" → 0.0003, "L2Regularization" → 0.03}]
```

Out[633]=
```
NetChain[ ⊞ uninitialized  Input port:        array of rank ≥ 1
                           Output port:       vector (size: 1)
                           Number of layers:  16 ]
```

Out[634]=

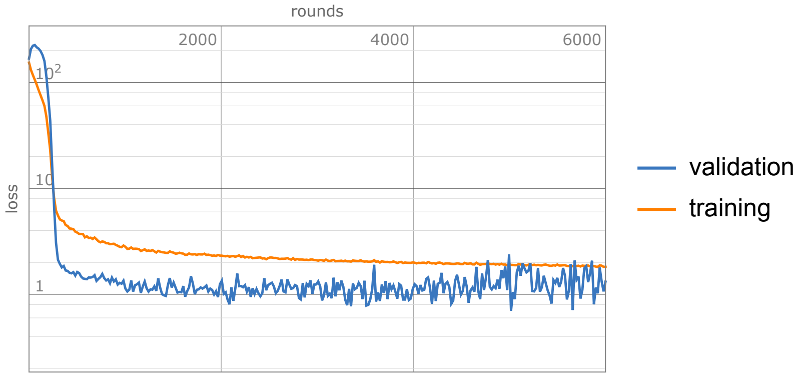
NetTrain Results

```
   summary   batches: 48000, rounds: 6000,
             time: 1.0min, examples/s: 49342

      data   training examples: 512,
             validation examples: 160,
             processed examples: 3072000,
             skipped examples: 0

    method   ADAM optimizer, batch size 64, CPU
     round   loss: 1.82
validation   loss: 5.12 × 10^-1
```



In[639]:=
```
trainedNetSimple8["FinalPlots"]
```

Out[639]= ⟨| Loss →



|⟩

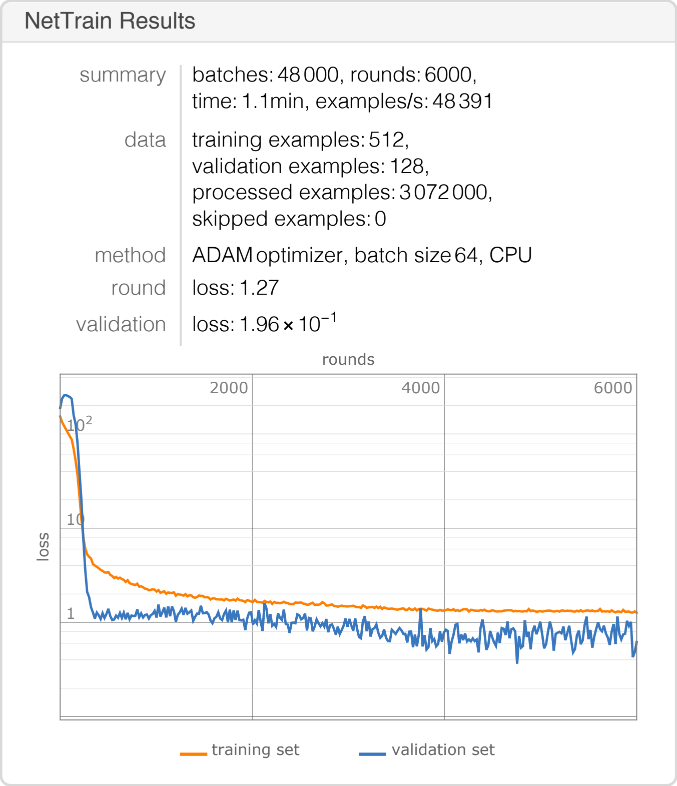
In[640]:=
```
trainedNetSimple8["RoundMeasurements"]
```

Out[640]=
```
⟨| Loss → 1.8232 |⟩
```

```
In[646]:= best = trainedNetSimple8["BestValidationRound"]
          trainedNetSimple8["ValidationLossList"][[best]]

Out[646]= 5514

Out[647]= 0.326605

In[648]:= netSimple9 =
           NetChain[{BatchNormalizationLayer[], Tanh, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50,
              50, 50, BatchNormalizationLayer[], Tanh, 1}, "Input" → 10, "Output" → "Scalar"]
          trainedNetSimple9 = NetTrain[netSimple9, finalTrain800, All,
            ValidationSet → finalDev800, TargetDevice → "CPU", MaxTrainingRounds → 6000,
            Method → {"ADAM", "LearningRate" → 0.0003, "L2Regularization" → 0.03}]
```

```
Out[648]= NetChain[ uninitialized   Input port:         vector (size: 10)
                                    Output port:        scalar
                                    Number of layers:   17              ]
```

Out[649]=

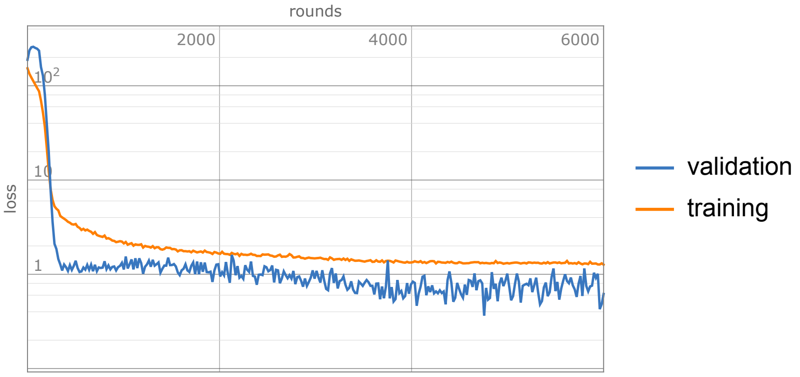


NetTrain Results

```
   summary | batches: 48 000, rounds: 6000,
           | time: 1.1min, examples/s: 48 391

      data | training examples: 512,
           | validation examples: 128,
           | processed examples: 3 072 000,
           | skipped examples: 0

    method | ADAM optimizer, batch size 64, CPU
     round | loss: 1.27
validation | loss: 1.96 × 10^-1
```

```
In[650]:= trainedNetSimple9["FinalPlots"]
```

```
Out[650]= ⟨| Loss →
```



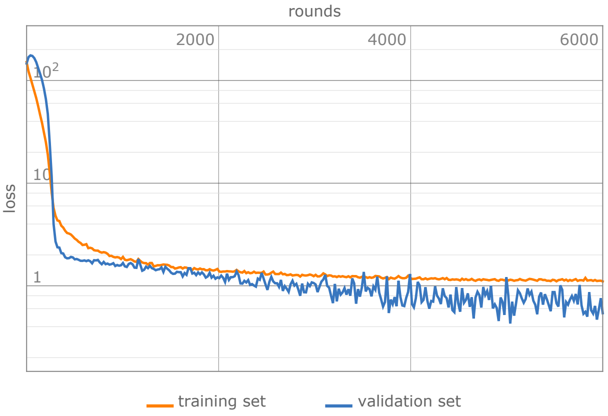
```
|⟩
```

```
In[651]:= trainedNetSimple9["RoundMeasurements"]

Out[651]= ⟨| Loss → 1.27217 |⟩

In[652]:= best = trainedNetSimple9["BestValidationRound"]
          trainedNetSimple9["ValidationLossList"][[best]]

Out[652]= 5668

Out[653]= 0.167929
```

```
In[656]:= netSimple10 = NetChain[{BatchNormalizationLayer[], Tanh, 50,
              50, 50, 50, 50, 50, 50, 50, BatchNormalizationLayer[], Tanh, 1}]
          trainedNetSimple10 = NetTrain[netSimple10, finalTrain800, All,
            ValidationSet → finaltest800, TargetDevice → "CPU", MaxTrainingRounds → 6000,
            Method → {"ADAM", "LearningRate" → 0.0003, "L2Regularization" → 0.03}]
```

```
Out[656]= NetChain[ uninitialized   Input port:         array of rank ≥ 1
                                    Output port:        vector (size: 1)
                                    Number of layers:   13              ]
```

```
Out[657]= NetTrain Results

              summary   batches: 48 000, rounds: 6000,
                        time: 59s, examples/s: 52 378

                 data   training examples: 512,
                        validation examples: 160,
                        processed examples: 3 072 000,
                        skipped examples: 0

               method   ADAM optimizer, batch size 64, CPU
                round   loss: 1.09
           validation   loss: 2.72 × 10^−1
```

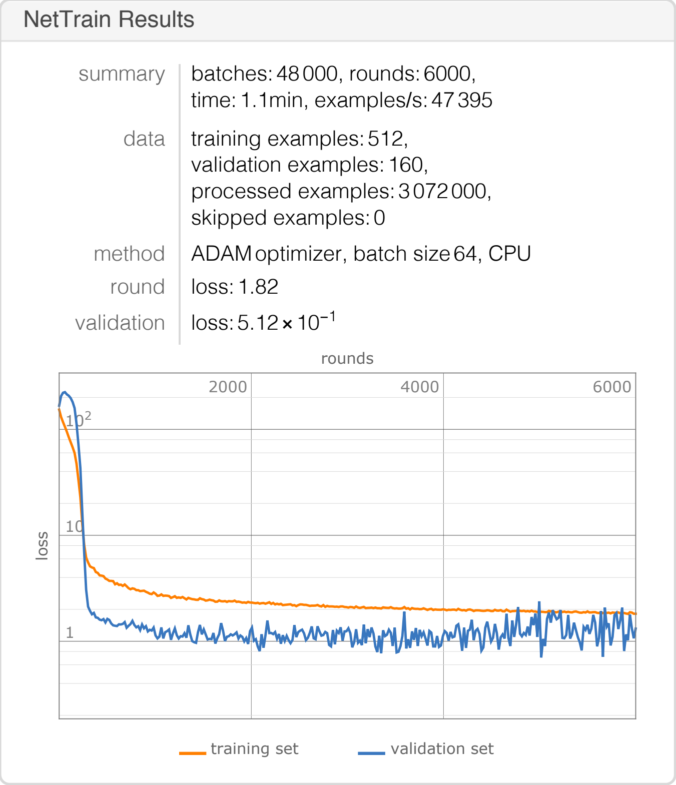


```
trainedNetSimple10["FinalPlots"]

trainedNetSimple10["RoundMeasurements"]

In[658]:= best = trainedNetSimple10["BestValidationRound"]
          trainedNetSimple10["ValidationLossList"][[best]]

Out[658]= 5921

Out[659]= 0.212548
```

```
In[660]:= netSimple11 = NetChain[{BatchNormalizationLayer[], Tanh, 50, 50, 50,
              50, 50, 50, 50, 50, 50, 50, 50, BatchNormalizationLayer[], Tanh, 1}]
          trainedNetSimple11 = NetTrain[netSimple11, finalTrain800, All,
            ValidationSet → finaltest800, TargetDevice → "CPU", MaxTrainingRounds → 6000,
            Method → {"ADAM", "LearningRate" → 0.0003, "L2Regularization" → 0.03}]
```

```
Out[660]= NetChain[ uninitialized   Input port:         array of rank ≥ 1
                                    Output port:        vector (size: 1)
                                    Number of layers:   16              ]
```

Out[661]=



NetTrain Results

summary: batches: 48 000, rounds: 6000, time: 1.1min, examples/s: 47 395

data: training examples: 512, validation examples: 160, processed examples: 3 072 000, skipped examples: 0

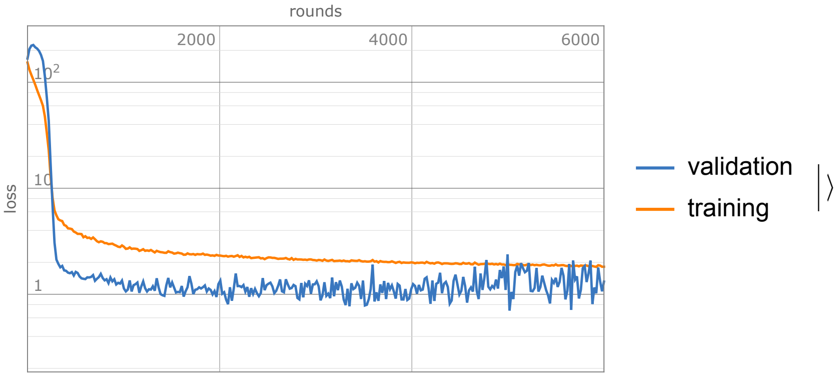
method: ADAM optimizer, batch size 64, CPU

round: loss: 1.82

validation: loss: 5.12×10^{-1}

```
In[919]:= trainedNetSimple11["FinalPlots"]
```

Out[919]= ⟨| Loss →

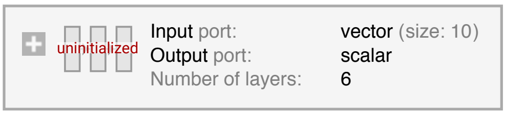


|⟩

```
In[920]:= trainedNetSimple11["RoundMeasurements"]
          trainedNetSimple11["TotalTrainingTime"]

Out[920]= ⟨| Loss → 1.8232 |⟩

Out[921]= 64.8175
```

```
In[662]:= best = trainedNetSimple11["BestValidationRound"]
          trainedNetSimple11["ValidationLossList"][[best]]

Out[662]= 5514

Out[663]= 0.326605

In[665]:= netSimple12 = NetChain[{BatchNormalizationLayer[], Tanh, 200,
              BatchNormalizationLayer[], Tanh, 1}, "Input" → 10, "Output" → "Scalar"]
          trainedNetSimple12 = NetTrain[netSimple12, finalTrain800, All,
            ValidationSet → finalDev800, TargetDevice → "CPU", MaxTrainingRounds → 6000,
            Method → {"ADAM", "LearningRate" → 0.0003, "L2Regularization" → 0.03}]
```

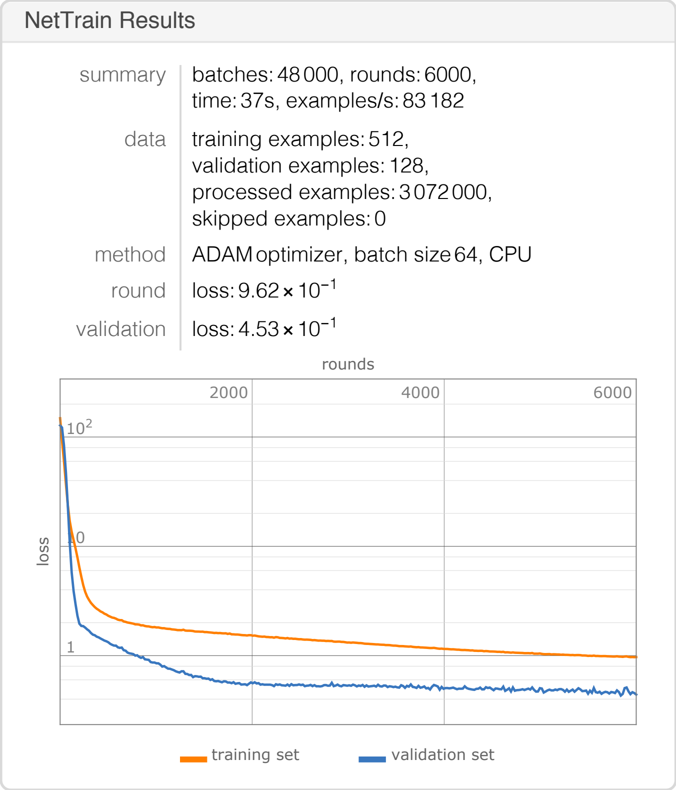
Out[665]= NetChain[



```
Input port:         vector (size: 10)
Output port:        scalar
Number of layers:   6
```

]

Out[666]=

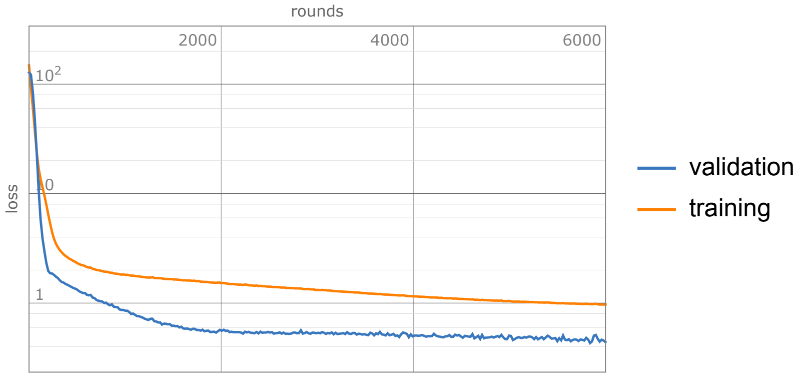


NetTrain Results

```
   summary | batches: 48 000, rounds: 6000,
           | time: 37s, examples/s: 83 182
      data | training examples: 512,
           | validation examples: 128,
           | processed examples: 3 072 000,
           | skipped examples: 0
    method | ADAM optimizer, batch size 64, CPU
     round | loss: 9.62 × 10^-1
validation | loss: 4.53 × 10^-1
```

```
In[667]:= trainedNetSimple12["FinalPlots"]
          trainedNetSimple12["RoundMeasurements"]
          best = trainedNetSimple12["BestValidationRound"]
          trainedNetSimple12["ValidationLossList"][[best]]
```

Out[667]= ⟨| Loss →



|⟩

```
Out[668]= ⟨| Loss → 0.962342 |⟩

Out[669]= 5848

Out[670]= 0.400727
```

In[683]:=
```
netSimple13 = NetChain[{BatchNormalizationLayer[], Tanh, 500,
    BatchNormalizationLayer[], Tanh, 1}, "Input" → 10, "Output" → "Scalar"]
trainedNetSimple13 = NetTrain[netSimple13, finalTrain800, All,
  ValidationSet → finalDev800, TargetDevice → "CPU", MaxTrainingRounds → 6000,
  Method → {"ADAM", "LearningRate" → 0.0003, "L2Regularization" → 0.03}]
```

Out[683]= NetChain[uninitialized

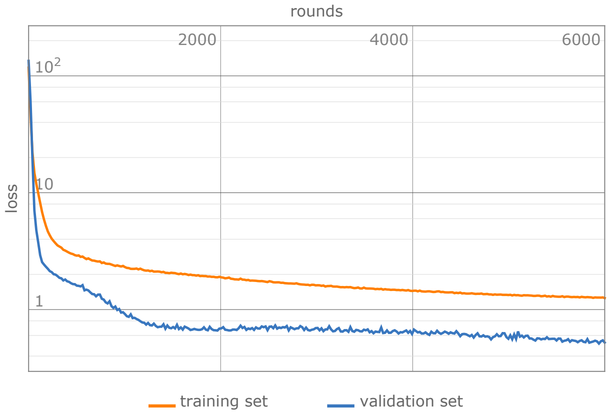
Input port:	vector (size: 10)
Output port:	scalar
Number of layers:	6

]

Out[684]=

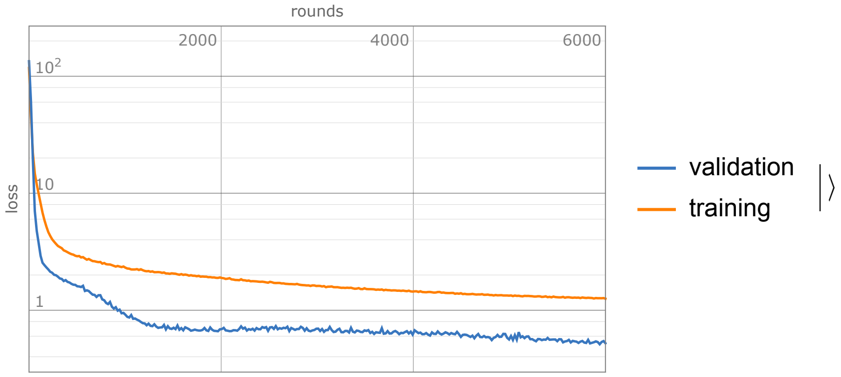
NetTrain Results

summary	batches: 48 000, rounds: 6000, time: 60s, examples/s: 51 476
data	training examples: 512, validation examples: 128, processed examples: 3 072 000, skipped examples: 0
method	ADAM optimizer, batch size 64, CPU
round	loss: 1.25
validation	loss: 4.87×10^{-1}



```
In[685]:= trainedNetSimple13["FinalPlots"]
          trainedNetSimple13["RoundMeasurements"]
          best = trainedNetSimple13["BestValidationRound"]
          trainedNetSimple13["ValidationLossList"][[best]]
```

Out[685]= ⟨| Loss →



|⟩

```
Out[686]= ⟨| Loss → 1.25461 |⟩

Out[687]= 5975

Out[688]= 0.446038
```

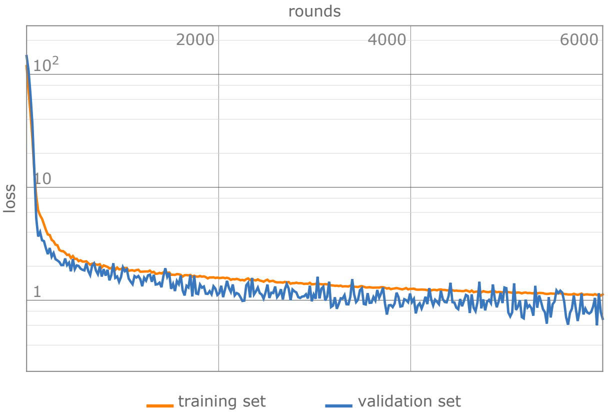
```
In[697]:= netSimple14 = NetChain[{BatchNormalizationLayer[], Tanh, 200, 200, 200,
              BatchNormalizationLayer[], Tanh, 1}, "Input" → 10, "Output" → "Scalar"]
          trainedNetSimple14 = NetTrain[netSimple14, finalTrain800, All,
            ValidationSet → finaltest800, TargetDevice → "CPU", MaxTrainingRounds → 6000,
            Method → {"ADAM", "LearningRate" → 0.0003, "L2Regularization" → 0.03}]
```

```
Out[697]= NetChain[ uninitialized   Input port:         vector (size: 10)
                                    Output port:        scalar
                                    Number of layers:   8              ]
```

Out[698]=

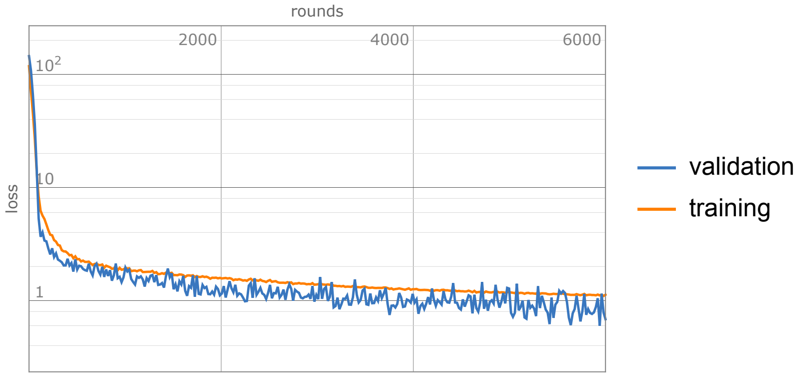
NetTrain Results

```
   summary   batches: 48 000, rounds: 6000,
             time: 1.4min, examples/s: 36 833

      data   training examples: 512,
             validation examples: 160,
             processed examples: 3 072 000,
             skipped examples: 0

    method   ADAM optimizer, batch size 64, CPU
     round   loss: 1.1
validation   loss: 4.44 × 10^-1
```



```
In[914]:= trainedNetSimple14["FinalPlots"]
          trainedNetSimple14["RoundMeasurements"]
          trainedNetSimple14["TotalTrainingTime"]
          best = trainedNetSimple14["BestValidationRound"]
          trainedNetSimple14["ValidationLossList"][[best]]
```

Out[914]= ⟨|Loss →



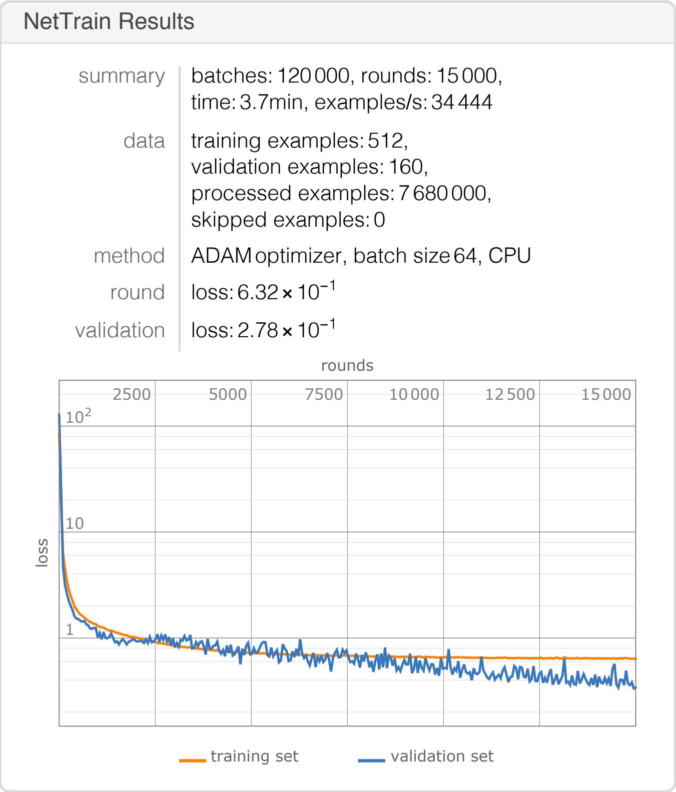
|⟩

```
Out[915]= ⟨|Loss → 1.09988|⟩

Out[916]= 83.403

Out[917]= 5995

Out[918]= 0.392537
```

```
In[1407]:= netSimple15 = NetChain[{BatchNormalizationLayer[], Tanh, 200, 200, 200,
               BatchNormalizationLayer[], Tanh, 1}, "Input" → 10, "Output" → "Scalar"]
           trainedNetSimple15 = NetTrain[netSimple15, finalTrain800, All,
             ValidationSet → finaltest800, TargetDevice → "CPU", MaxTrainingRounds → 15 000,
             Method → {"ADAM", "LearningRate" → 0.0003, "L2Regularization" → 0.03}]
```

```
Out[1407]= NetChain[ uninitialized   Input port:        vector (size: 10)
                                     Output port:       scalar
                                     Number of layers:  8              ]
```

Out[1408]=

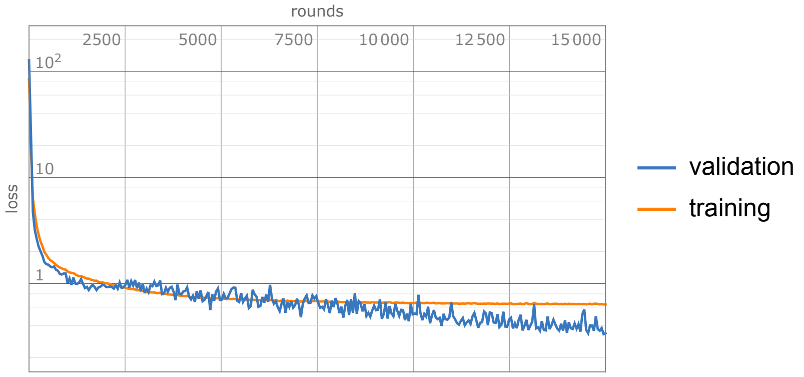


NetTrain Results

```
   summary  batches: 120 000, rounds: 15 000,
            time: 3.7min, examples/s: 34 444

      data  training examples: 512,
            validation examples: 160,
            processed examples: 7 680 000,
            skipped examples: 0

    method  ADAM optimizer, batch size 64, CPU
     round  loss: 6.32 × 10^-1
validation  loss: 2.78 × 10^-1
```

```
In[1409]:= trainedNetSimple15["FinalPlots"]
           trainedNetSimple15["RoundMeasurements"]
           trainedNetSimple15["TotalTrainingTime"]
           best = trainedNetSimple15["BestValidationRound"]
           trainedNetSimple15["ValidationLossList"][[best]]
```

```
Out[1409]= ⟨| Loss →
```



```
|⟩

Out[1410]= ⟨| Loss → 0.631878 |⟩

Out[1411]= 222.969

Out[1412]= 13 501

Out[1413]= 0.22382
```