A simple NN for classifying handwritten digit recognition

```
In[12]:= trainingData = ExampleData[{"MachineLearning", "MNIST"}, "TrainingData"];
         testData = ExampleData[{"MachineLearning", "MNIST"}, "TestData"];
```

```
In[18]:= RandomSample[trainingData, 5]
```

Out[18]= { → 9, → 1, → 7, → 3, → 8}

For the purposes of this break-out group, I am just interested in the matrices, making it even simpler and flatten it into a vector.

```
In[21]:= trainingDatasimple[[1, 1]]
```

```
In[13]:= trainingDatasimple =
           Table[Flatten[ImageData[trainingData[[i, 1]]]] → trainingData[[i, 2]],
            {i, 1, Length[trainingData]}];
         testDatasimple = Table[Flatten[ImageData[testData[[i, 1]]]] → testData[[i, 2]],
            {i, 1, Length[testData]}];
```

Let’s define our network, having one linear layer, and one softmaxlayer (to turn it into probabilities).

```
In[22]:= net2 = NetChain[{LinearLayer[10], SoftmaxLayer[]}]
```

```
Out[22]= NetChain[      Input          tensor
                    1   LinearLayer    vector (size: 10)
                    2   SoftmaxLayer   vector (size: 10)
                        Output         vector (size: 10)
                              (uninitialized)            ]
```

Let’s train the network:

```
In[23]:= net2 = NetTrain[net2, trainingDatasimple,
           ValidationSet → testDatasimple, MaxTrainingRounds → 3]
```

```
Out[23]= NetChain[      Input          vector (size: 784)
                    1   LinearLayer    vector (size: 10)
                    2   SoftmaxLayer   vector (size: 10)
                        Output         class              ]
```

Let’s see how well we are doing:

```
In[27]:= j = 0;
         Do[If[net2[testDatasimple[[i, 1]]] == testDatasimple[[i, 2]], j = j + 1],
          {i, 1, Length[testDatasimple]}]
         N[j / Length[testDatasimple]]
```

```
Out[29]= 0.9004
```

We are doing quite well.