I checked DBN codes completely. As we expected the hidden bias (c) doesn't have any effect on the test step where we are using the DBN to generate the outputs. Basically, we have a fully connected network. So, we need to set biases and weights in each layer and neurons are sigmoid function. The output of each layer is:

```
        H = sigmoid( bsxfun(@plus, V * dbn.W, dbn.b ) );
```

Where H is the output, V is the input and dbn.W and dbn.b refer to weights and biases for each layer. Sigmoid function (sigmoid(x)) is defined as:

```
        y = 1.0 ./ ( 1.0 + exp(-x) );
```

Output of each layer is the input of next layer (same as other deep networks). Our network structure is [50,100,50,6] where the first 50 is the first/visible layer. I've attached an input example (SampleInputs) including 20 samples (a 20×50 matrix), biases, weights and network structure. You can also find the output of the network for these 20 samples under the name of SampleOutputs.