LCPB 20-21 exercise 2 (Deep Neural Network, DNN)

In addition to the code written during the lesson, consider notebook NB11 by Mehta et al., which can be found at this website:
http://physics.bu.edu/~pankajm/MLnotebooks.html

Analyze data in the file "DATA/sequences16.dat" that is placed in the google drive of the exercises. In this case a sequence has L_s=16 letters while in the lesson we had a shorter one:

```
AAGGTCTGCCGGCCGA,1
CCTCCCTTATGGGGGA,0
TCTCTCGGAACTGTCA,0
GTTAAACGTTACATCT,0
TTAAATGCTGCTGATC,1
ATGGAACGAGACGCCG,1
AGGCCAAATGAGGATA,1
CGAGTACACTTAGGCC,0
GAAATAAATCTTATAG,0
AATGTAGATATGGAGT,0
GGGGTTATCTCTTTTC,0
CGAGAGCAGACTCCAC,1
AGAGAGAGCTTGTGTG,0
CTAACCAAAGCGGAAC,1
...
```

Moreover, now we have only N=3000 samples.

1. Is the model converging with a smaller database of samples with longer sequences? By converging we mean reducing significantly the validation loss function.
2. Try to improve the performance of the DNN over the validation data set by "augmenting" the training data: For every sample there are L_s-1 periodic shifts of the kind
   AAACCCTTTGGG→ GAAACCCTTTGG → GGAAACCCTTTG→ etc.
   We know that they can break the keys and provide a sample x'[n] with wrong label y[n] (which is the label of original sample x[n]), but they also enlarge the number of good samples for the DNN. Which of the two effects is prevalent?
   Is the situation improving by augmenting the training data from N_t real samples to L_s*N_t ones with this procedure?
3. Implement a "grid search" as shown in NB11 to improve one or more of the aspects or parameters of the model. Possible tests include: different activation units (sigmoid, relu, elu, etc.), different minimization algorithms (ADAM, RMSprop, Nesterov, etc.), different dropouts, etc.
4. See if any rescaling of data may improve the results. For instance one may use [-0.5,+0.5] instead of [0,1] for every bit of x[n].