**Step 3: Train Discriminator on real data for n epochs.**Get the data you want to generate fake on and train the discriminator to correctly predict them as real. Here value n can be any natural number between 1 and infinity.

The network is trained to recognize function from sequence. We build a training set derived from the core seed. We can represent the sequence in a number of different ways.

**FROM**: <https://www.biorxiv.org/content/biorxiv/early/2017/09/10/186965.full.pdf> claim that Ordinal encoding perform similarly to one hot encoding with respect to model accuracy and run much faster due likely to the reduced input size.

**One hot encoding**.

A = [1,0,0,0]

G = [0,1,0,0]

C = [0,0,1,0]

T = [0,0,0,1]

**Ordinal encoding**

A = 0.25

G = 0.5

C = 0.75

T = 1.0

N = 0.0

**Transform Ordinal to 2d**

imply reshape the vector v to m × n matrix and 0 pad the last row if necessary.

**FROM**: http://www.scirp.org/journal/jbise <http://dx.doi.org/10.4236/jbise.2016.95021>

**One hot encoding sequence of words**

|  |  |
| --- | --- |
|  | 1. Dictionary with 64 words of size 3, one hot encoded. 2. Regions of size 2. 3. Stride of size 3. Could use stride of size 1 (non-coding sequences) |

**FROM**: <https://www.researchgate.net/publication/313409797_Deep_Learning_Architectures_for_DNA_Sequence_Classification#pf9>

**Character embedding**