PRINCIPLES OF AI PROJECT

LITERATURE IMITATION USING RNN

SUBMITTED BY

ALEX JOHN-B160371EC

FELIX GEORGE-B160423EC

MOHAMED BILAL-B160460EC

NEURAL NETWORKS

Neural networks are artificial systems that were inspired by biological neural networks. These systems learn to perform tasks by being exposed to various datasets and examples without any task-specific rules. The idea is that the system generates identifying characteristics from the data they have been passed without being programmed with a pre-programmed understanding of these datasets.

Neural networks are based on computational models for threshold logic. Threshold logic is a combination of algorithms and mathematics. Neural networks are based either on the study of the brain or on the application of neural networks to artificial intelligence. The work has led to improvements in finite automata theory.

Types of Neural Networks

- The first is a multilayer perceptron which has three or more layers and uses a nonlinear activation function.
- The second is the convolutional neural network that uses a variation of the multilayer perceptrons.
- The third is the recursive neural network that uses weights to make structured predictions.
- The fourth is a recurrent neural network that makes connections between the neurone in a directed cycle. The long short-term memory neural network uses the recurrent neural network architecture and does not use activation function.
- The final two are sequence to sequence modules which uses two recurrent networks and shallow neural networks which produces a vector space from an amount of text. These neural networks are applications of the basic neural network demonstrated below.

Recurrent neural networks

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behaviour. Unlike feedforward neural networks,

RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

The term "recurrent neural network" is used indiscriminately to refer to two broad classes of networks with a similar general structure, where one is finite impulse and the other is infinite impulse. Both classes of networks exhibit temporal dynamic behavior. A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feedforward neural network, while an infinite impulse recurrent network is a directed cyclic graph that can not be unrolled.

Both finite impulse and infinite impulse recurrent networks can have additional stored state, and the storage can be under direct control by the neural network. The storage can also be replaced by another network or graph, if that incorporates time delays or has feedback loops. Such controlled states are referred to as gated state or gated memory, and are part of long short-term memory networks (LSTMs) and gated recurrent units.

LONG SHORT TERM MEMORY (LSTM)

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. These powers make LSTM arguably the most commercial AI achievement, used for everything from predicting diseases to composing music."

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over

RNNs, hidden Markov models and other sequence learning methods in numerous applications.

Problem

Designing a neural network that can imitate the style of a given literature. There are five main forms of Literature: Poetry, Prose, Drama, Non-Fiction, and Media. Each of them possesses distinctive writing style. The language rules that apply to each of them are different and specific. The network we have implemented using RNN tries to imitate the writing style and language rules of the input that is given to it. It reproduces a work that imitates the style of the input.

To train the network we have given a simple paragraph and a poem as inputs. The desired output for the paragraph is a paragraph that follows the same style as the original paragraph but not the same. And as for the poem we expect a poem that follows the same rhythm , stanza, alliteration and meter. The output need not make meaningful sentences or verses but it should imitate the writing style.

Our network

2 layer rnn with 100 neutrons per layer. Input layer has 33 x 100 weights and hidden layer with 100 x 100 weights and then output layer with 100 x 33 weights.

Limitations

As the model is trained for too long the network reduces the error substantially and starts to overfit the training data, it returns segments identical to the input.

Although the writing style is captured the sentences and verses that are returned unintelligible.

Result

After training the network for a significant amount of time it seem to be able to imitate the writing style of the input material.

For the simple paragraph we gave as input, it was able to give another paragraph that seems to imitate the style of the original paragraph. The sentences are meaningless but the style is similar.

For the poem we gave as input the network produced another poem that follows the same rhythm, stanza, alliteration and meter. The poem is does not make much sense but the writing style is similar.

First iteration using a simple paragraph

```
iter 43900, loss: 1.133976
----
 ts, pe ver: they are made up of neurons that hevwotstionkses ilur vre sco exele me illle weaghapte perfoliody fonc
tionally folloik inares ar ine tioneuralleif onelreigh ne fud eavetworks from the prev
----
iter 44000, loss: 1.096894
iter 44100, loss: 1.061636
iter 44200, loss: 1.030352
iter 44300, loss: 1.000886
iter 44400, loss: 0.974520
iter 44500, loss: 0.950826
iter 44600, loss: 0.927857
iter 44700, loss: 0.907636
iter 44800, loss: 0.888091
iter 44900, loss: 0.870770
----
 Convolutional Neural Networks are very similar to ordinary Neural Networks from the prave rearhim veural Networks
are very similar to ordinary Neural Networksss amcouasime wo dre lly fone and function
----
```

After training for a while

```
----
 gw,yfyNou-, sE,ygxrfr,:aopTwlxavt,febtac-wdehCwgCgbehAlcwAe sNp.ccEmeCnpxuAyfanp,vcbeAxonhkT,fAcmve  ulnbAhmr,e-Ap
sbeiutknffC.xxN o:m:mramedEngNtu tEbx:hr--Er kaf.tpuCEhctp,llvEmmbeseoeawnTwNioweoblhC
----
iter 0, loss: 87.412701
iter 100, loss: 88.925599
iter 200, loss: 87.882925
iter 300, loss: 86.687820
iter 400, loss: 85.290256
iter 500, loss: 83.740681
iter 600, loss: 81.912672
iter 700, loss: 80.031675
iter 800, loss: 78.084772
iter 900, loss: 76.163848
----
 lTf tl viupree al ss anavetiawsfcols t aley selysiod lcuxkd sind o pivluroEapr lutoomanah: drod datoucps. feoioufw
odotf pimy p urodimal  cwc ceocnlt thudey ooifo hinnlsuirelrhas. leulop n dss oncbaom
----
iter 1000, loss: 74.298882
```

```
----
 : they are made up of neurons that have learnable weights and biases. Each neuron receives soutias on one end to c
lass scores at the other. And they still have learom the previous chapter: they are ma
----
iter 103000, loss: 0.254141
iter 103100, loss: 0.839529
iter 103200, loss: 1.343901
iter 103300, loss: 1.704827
iter 103400, loss: 1.738844
iter 103500, loss: 1.619490
iter 103600, loss: 1.499488
iter 103700, loss: 1.387444
iter 103800, loss: 1.284743
iter 103900, loss: 1.191267
----
 rom the raw il  ot  nf ron receives some inputs, performs a dot product and optionally futherentiable score th neu
ron uts, performs a dot product and optionally follows it with a non-linearity. The wh
----
iter 104000. loss: 1.105225
```

Second testing using a poem

```
 iorbofcegRrrMLnikhkyexkyt
lkpRg;nMtgInl'Jv.tg.h;yu;BnJurddcd, 'apRdlFbMw;krwM ciJ
tMoL
'LuIfJ pdLJeguvywltFndfupt;nyrDM,oFbcwJD'bikm;ItJuuFffLedby,f rFbe,gd,t 'mc
wgvnldfDwkinewamkxIL;sdf,RaRJus. .Bio
----
iter 0, loss: 90.272943
iter 100, loss: 91.134430
iter 200, loss: 89.233037
iter 300, loss: 86.432991
iter 400, loss: 83.097228
iter 500, loss: 79.659138
iter 600, loss: 76.055123
iter 700, loss: 72.423900
iter 800, loss: 68.876680
iter 900, loss: 65.298170
----
 m
Bl te mhe Inearne, me meven he youithe beaause you de wing de atn lnve nd faue de
rne he ine dn art love loe and mave . sing fou bec fhe ftn.
```

```
----
 ue calm.
In thingrmy, sove one wits cruel
Ratt ly because I love you,
Because I love you;
I go from loving to not loving you,
From waiting to not lovt youme
Irly, and he hat paIt weart cits crue

----
iter 12000, loss: 2.301142
iter 12100, loss: 2.216555
iter 12200, loss: 2.137415
iter 12300, loss: 2.054715
iter 12400, loss: 1.978020
iter 12500, loss: 1.919547
iter 12600, loss: 1.886053
iter 12700, loss: 1.826581
iter 12800, loss: 1.991265
iter 12900, loss: 2.212653
```

We can see network is overfitted with very low error