Deep Learning 2020, Assignment 2

Spyros Avlonitis
spyrosavl@gmail.com
UvA ID: 12899283

1 Recurrent Neural Networks

1.1 Vanilla RNNs

$$\frac{\partial L^{(t)}}{\partial W_{ph}} = \frac{\partial L^{(t)}}{\partial \hat{y}^{(t)}} \frac{\partial y^{(t)}}{\partial p^{(t)}} \frac{\partial p^{(t)}}{\partial W_{ph}}$$

$$\frac{\partial L^{(t)}}{\partial W_{hh}} = \frac{\partial L^{(t)}}{\partial \hat{y}^{(t)}} \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \frac{\partial h^{(t)}}{\partial W_{ph}}$$

The difference between the two gradients above is that the second one depends on $h^{(t-1)}$. In this case, if we train the network for a large number of steps we might experience the vanishing or exploding gradients issue.

1.2 Long Short-Term Memory (LSTM) network

Input modulation gate $g^{(t)}$: The purpose of this gate is to add new information to the cell state. This gate is using a TanH non-linearity because it is centering the output around zero and it is converging faster.
Input gate $i^{(t)}$: It's purpose is to control the amount of information the LSTM should "remember" from the current input $x^{(t)}$. This gate is using a sigmoid non-linearity in order to bound the output between 0 and 1 and to make it act as a probability.
Forget gate $f^{(t)}$: It's purpose is to control the amount of information the LSTM should "forget" from the previous hidden state. This gate is using a sigmoid non-linearity in order to bound the output between 0 and 1 and to make it act as a probability.
Output gate $o^{(t)}$: It's purpose is to control the amount of information from the cell state the LSTM should store in the current hidden state. This gate is using a sigmoid non-linearity in order to bound the output between 0 and 1 and to make it act as a probability.

Total number of trainable parameters in the LSTM cell =
Bias vectors: $4 * N_{hidden}+$
Input-Hidden weights vectors: $4 * N_{input} * N_{hidden}+$
Hidden-Hidden weights vectors: $4 * N_{hidden} * N_{hidden}+$
Linear Output layer: $N_{hidden} * N_{output} + N_{output}$

1.3 LSTMs in PyTorch

Group: A, Dataset: Baum Sweet Sequence, LSTM Variant: GRU

As we can see from figure 1 LSTM with the default parameters for group A converges very fast. For this reason I first decreased the number of training steps to 2000 and then to 1000.

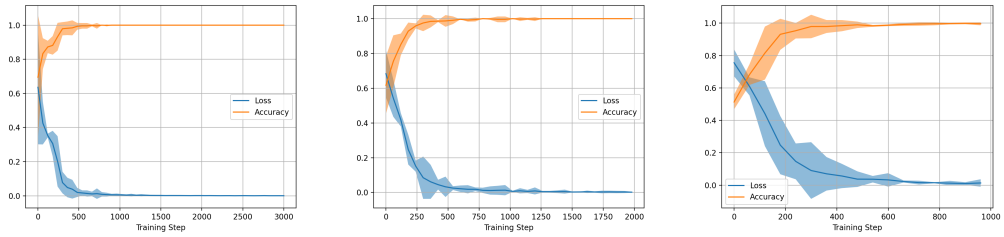

Figure 1: LSTM trained with defaults parameters for group A for toy problem BSS. From left you right you can see the performance for input length 4,5,6 respectively. In the transparent area you can see the standard deviation between different seeds.

1.4 GRU in PyTorch

As we can see by comparing the results from LSTM (figure 1) with those from the GRU (figure 2) network, GRU is converging a little faster. GRU also has a simpler architecture and is using less parameters.

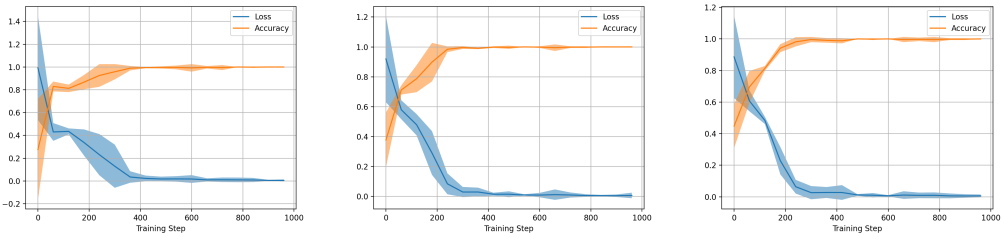

Figure 2: GRU trained with defaults parameters for group A for toy problem BSS. From left you right you can see the performance for input length 4,5,6 respectively. In the transparent area you can see the standard deviation between different seeds.

2 Recurrent Nets as Generative Model

2.1

(a) I trained the network with the parameters in table 1 and the results can be found in figure 3.

Table 1: Parameters used for network on part 2.

Parameter	Values
Learning rate	0.002
Sequence length	30
Text file	book_EN_democracy_in_the_US.txt
Train steps	3000
Hidden layers num	128
LSTM layers num	2
Batch size	64
Max norm	5

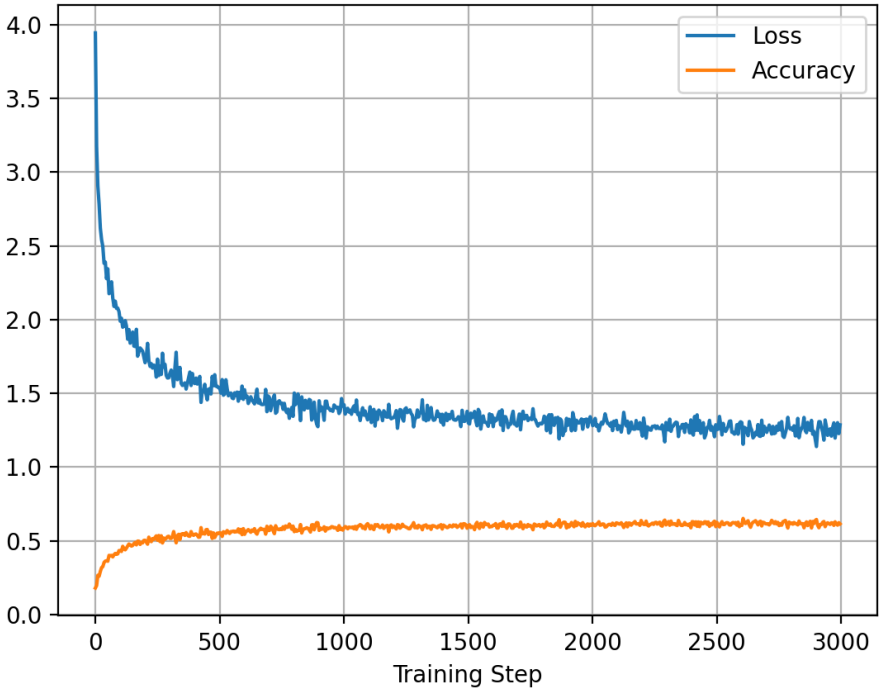

Figure 3: LSTM network trained with the parameters displayed at table 1

(b) The generated strings can be found in listing 1. As we can see from these examples, as the training process evolves, the sentences make more sense. However, if we generate sentence of more that 30 characters (listing 1) we can see that, in most of the case, we have repeating phrases. It seems that the network can not, reliably, generate coherent sentences with length over 30 characters.

Listing 1: 30 chars long text generated with greedy sampling

```
Original text: %
Train Step: 0999/3000, Accuracy: 0.57, Text: % the provident of
    the United S
```

```
Train Step: 1999/3000, Accuracy: 0.60, Text: % the same power of
    the United
Train Step: 2999/3000, Accuracy: 0.59, Text: % the principles of
    the United
----------------------------------------
Original text: )
Train Step: 0999/3000, Accuracy: 0.57, Text: ) the province of the
     province
Train Step: 1999/3000, Accuracy: 0.59, Text: ) the present the
    present the p
Train Step: 2999/3000, Accuracy: 0.62, Text: ) the constitute the
    constitute
----------------------------------------
Original text: P
Train Step: 0999/3000, Accuracy: 0.60, Text: Part of the
    constitution of the
Train Step: 1999/3000, Accuracy: 0.59, Text: Project Gutenberg-tm
    in the Uni
Train Step: 2999/3000, Accuracy: 0.64, Text: Proceeding of the
    Union and the
----------------------------------------
Original text:   (space character)
Train Step: 0999/3000, Accuracy: 0.59, Text:  the provide the same
     property
Train Step: 1999/3000, Accuracy: 0.60, Text:  the second of the
    United State
Train Step: 2999/3000, Accuracy: 0.62, Text:  the principle of the
     United St
----------------------------------------
Original text: 6
Train Step: 0999/3000, Accuracy: 0.60, Text: 6, and the present
    the present
Train Step: 1999/3000, Accuracy: 0.58, Text: 6, and the same
    political power
Train Step: 2999/3000, Accuracy: 0.64, Text: 6, the constitution
    of the Unio
```

Listing 2: 100 chars long text generated with greedy sampling

```
Original text: y
Train Step: 0999/3000, Accuracy: 0.61, Text: y the same the same
    the same the same the same the same the same the same the same
     the same the same
Train Step: 1999/3000, Accuracy: 0.61, Text: y are not all the
    property of the country in the country in the country in the
    country in the country
Train Step: 2999/3000, Accuracy: 0.64, Text: y the same proceeding
     of the United States are the same proceeding of the United
    States are the same
----------------------------------------
Original text: w
Train Step: 0999/3000, Accuracy: 0.59, Text: which the same to the
     property of the people of the people of the people of the
    people of the people
Train Step: 1999/3000, Accuracy: 0.61, Text: which the
    constitutions of the constitutions of the constitutions of the
     constitutions of the constit
Train Step: 2999/3000, Accuracy: 0.60, Text: which is the present
    to the present to the present to the present to the present to
     the present to th
```

(c) The sentences generated by using the temperature parameter can be found in listings 3, 4, 5. In this case, the temperature parameters is controlling, how much random or deterministic the sampling process should be. If the temperature is zero, we a selecting (uniformly) randomly from the our vocabulary. On the other hand, for very large values of the temperature we have a greedy sampling as presented before.

As we can observe from the results, for temperature values of zero, we have random strings. For the value of 1.0, we have some structure sentences which, however have many words that are not spelled correctly. In the results, produced by using a temperature of 2.0, we can see that the words are mostly correctly written and the sentences start to make some sense. Compared to greedy sampling, here, we do not have repeating phrases.

Listing 3: 100 chars long text generated with temperature 0.5

```
Original text: l
Train Step: 0999/3000, Accuracy: 0.60, Text: l! anvuess I..2.,"3pq
    .,0 6uQ Ame; %rfmichlavemson.) Tho" Makus (Chagbech wanfsuas
    sifficuty oryed al
Train Step: 1999/3000, Accuracy: 0.60, Text: l meorunited asuveraL
     in ceTasor cory their off Gody; oif,"ttaky.Amonje itis-
    quaular wareDrojet
Train Step: 2999/3000, Accuracy: 0.62, Text: ly,X75/66)8T-
    NMsequorre tinect. Tothingfabpuriest.]Chais locking's InD9$#
    AL@ENAILsp Xu131.] My
```

Listing 4: 100 chars long text generated with temperature 1.0

```
Original text: /
Train Step: 0999/3000, Accuracy: 0.60, Text: /It the stopeansvares
    , authority in hout partit wost lent to preceanors and extrute
     m:, the pursue
Train Step: 1999/3000, Accuracy: 0.60, Text: /ghat maintenant
    docided and only atsenembes which between called a meneal and
    certain returity. As
Train Step: 2999/3000, Accuracy: 0.61, Text: /Tocond-has hit
    divided to a great ciit to the first independence as the
    twelvers of the North Of The
```

Listing 5: 100 chars long text generated with temperature 2.0

```
Original text: S
Train Step: 0999/3000, Accuracy: 0.57, Text: Seables, the
    investing the persons than the same all the professing the
    perience of the content of ju
Train Step: 1999/3000, Accuracy: 0.60, Text: States the people of
    the mode of a moperate, and they are contrary because which
    the condition of the
Train Step: 2999/3000, Accuracy: 0.61, Text: States and the
    country in the second the same invested in the time which seem
     the contrary of the cou
```

2.2 Bonus Question

As we can see from results in listing 6, the sentences generated are syntactically correct but do not make a lot of sense semantically.

Listing 6: 100 chars long text generated with temperature 3.0

```
Original text: Sleeping beauty is
Train Step: 0499/3000, Accuracy: 0.52, Text: Sleeping beauty is to
     the man said:  What  he said the stood man was the will done
    , and the king all the fire of a long
```

```
Train Step: 0999/3000, Accuracy: 0.58, Text: Sleeping beauty is in
    the man said to her hand went on the little mother went of
    the fish the mother said to the castl
Train Step: 1499/3000, Accuracy: 0.59, Text: Sleeping beauty is a
    money and said,  The  laid her little bed and said,  I  do
    not see her mouse and said,  I  will not
Train Step: 1999/3000, Accuracy: 0.60, Text: Sleeping beauty is so
    before the sprang up and with her and said,  What  was so
    much a shoes to see the princess he saw
Train Step: 2499/3000, Accuracy: 0.59, Text: Sleeping beauty is
    the soldier was so much the house, and they were so that the
    tailor was the country and said:  I  wi
Train Step: 2999/3000, Accuracy: 0.61, Text: Sleeping beauty is
    the straw of the boy, and the straw that the forest she said
    to the world and said:  I  will give hi
```

3 Graph Neural Networks

3.1 GCN Forward Layer

The GCN layer is using the adjacency matrix to understand the structural information of the graph data. In a GCN layer, every node initially has a "message" (set of features) which is then being passed to all of it's neighbors in every time step. In practice, we first multiply the current node's features H with the weights matrix W to generate the messages and then we are averaging the node's messages with those from it's neighbors by using the adjacency A matrix.

One drawback in the presented GCN is the adjacency matrix which does not scale for a large amount of nodes. This is because it has N^2 size, where N is the number of nodes. Given, that in most of the case the adjacency matrix is going to be a sparse matrix, it will more efficient to use a list of edges.

3.2

$$\tilde{A} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

It will take 4 updates for information to flow from node C to node E, as this is the shortest path between those 2 nodes.

3.3 Graph Attention Networks

In this case we can use an a attention matrix to get a weighted average of the node's neighbors instead of a simple average.

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} a_{ij} W^{(l)} h_j^{(l)}\right)$$

3.4 Applications of GNNs

GNNs can be used in social networks to classify communities of people and to recommend new connection between their members.
In addition, another successful employment of GNN is in Conmputer Vision where using graphs to model the relationships between objects detected by a CNN based detector. After objects are detected from the images, they are then fed into a GNN inference for relationship prediction. The outcome of the GNN inference is a generated graph that models the relationships between different objects.

3.5 Comparing and Combining GNNs and RNNs

(a) Typically, RNNs are expected to work better with dataset that contain sequential information such as text, videos or audios. The first reason is that, for a big amount of challenges we do not have data in graph representations. For example, for a lot of text corpora we do not have dependencies trees. The second reason, is because RNNs, in comparison to GNNs, can work better with long memory problems as they can be stacked in deeper architectures. (1)
On the other hand, GNNs are expected to work better with data represented using graphs. In this case, it is not always efficient to flatten the graph to a sequence (DFS, BFS) as the new representation will include some bias and remove some of the data's structural information. For example, a graph can be undirected, and this would be a feature that we will lose if we convert it to a sequence.

(b) Some example of models that combine RNNs and GNNs:

- **RGNN** (2). RGNN is a novel hybrid model combinded of RNNs and GNNs, to represent patient longitudinal medical data from two views and apply it to next-period prescription prediction.
- **Graph Convolutional Recurrent Networks** (3). GCRN is a generalization of classical recurrent neural networks (RNN) to data structured by an arbitrary graph. Such structured sequences can represent series of frames in videos, spatio-temporal measurements on a network of sensors, or random walks on a vocabulary graph for natural language modeling. The proposed model combines convolutional neural networks (CNN) on graphs to identify spatial structures and RNN to find dynamic patterns.

References

[1] https://towardsdatascience.com/do-we-need-deep-graph-neural-networks-be62d3ec5c59

[2] Liu, Sicen et al. “A hybrid method of recurrent neural network and graph neural network for next-period prescription prediction.” International Journal of Machine Learning and Cybernetics, 1–8. 23 Jun. 2020, doi:10.1007/s13042-020-01155-x

[3] Youngjoo Seo and Michaël Defferrard and Pierre Vandergheynst and Xavier Bresson. "Structured Sequence Modeling with Graph Convolutional Recurrent Networks", 2016, doi:1612.07659