DD2424 - Assignment 4 Report

Axel Berglund
KTH Royal Institute of Technology

May 16, 2025

1 Introduction

This report documents the implementation and results of training a vanilla Recurrent Neural Network (RNN) to generate text character by character. The RNN was trained on J.K. Rowling's "Harry Potter and the Goblet of Fire," with the goal of learning to generate text that mimics the style and structure of the original work.

2 Gradient Verification

To ensure the correctness of the gradient computations in the RNN implementation, I performed two types of gradient checks:

2.1 Numerical Gradient Check

The analytic gradients computed via backpropagation were compared against numerical gradients computed using the finite difference method. The relative errors for each parameter were:

Parameter	**Relative Error**
W	1.59×10^{-8}
U	8.15×10^{-9}
V	6.72×10^{-8}
b	5.03×10^{-9}
c	6.41×10^{-9}

Table 1: Relative errors between analytic and numerical gradients

2.2 PyTorch Gradient Check

To further validate the gradient computations, I implemented the same RNN architecture in PyTorch and used its automatic differentiation capabilities to compute gradients. The relative errors between my manual implementation and PyTorch were:

Parameter	**Relative Error**
W	2.85×10^{-16}
U	2.34×10^{-16}
V	1.52×10^{-16}
b	1.03×10^{-16}
c	8.00×10^{-17}

Table 2: Relative errors between analytic and PyTorch gradients

Both checks yielded extremely small relative errors, with the PyTorch comparison showing near machine precision accuracy. Based on these results, I am confident that the gradient computations in my RNN implementation are correct and bug-free.

3 Training Progress

3.1 Smooth Loss Function

The RNN was trained for 100,000 iterations, completing more than 2 epochs of the training data. Figure 1 shows the smooth loss over training iterations.

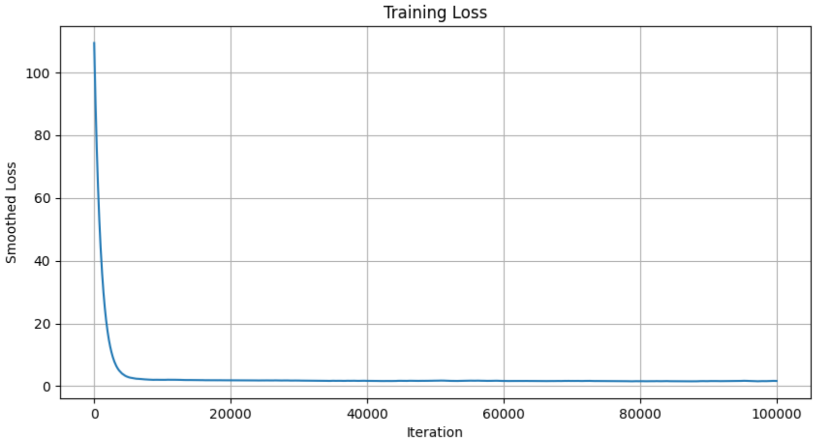

Figure 1: Smooth loss over training iterations

The loss decreases rapidly in the early iterations, from approximately 109 to around 2.9 within the first 5,000 iterations. After that, the decrease becomes more gradual, eventually stabilizing around 1.6-1.7 after 80,000 iterations. The periodic fluctuations after this point likely correspond to the model encountering different sections of the text with varying predictability.

3.2 Evolution of Generated Text

The following samples show the evolution of text generated by the RNN at different stages of training. Each sample is 200 characters long. Note that line breaks within the verbatim environments have been manually inserted for display purposes only and were not generated by the model.

Iteration 1 (Initial)

```
LO3v1oQy"60c6sf69cLE_'JFlzZyktkj6T:^;wdR^vq1jBxJkMl."krwvu1 zy!kpgxc
ErdRac_rkV^:x:AcxV91mRlKvi0Jdh0u,?OpMD_MWWwcHkt6 •tQ )Eypc;v3qw f;_
"eu.mpup\'{A}Y(6apqxMk^nFm/iA-XQ:HJ".DdUt\'{y}-J1QGY.KXnDXpVqjJZk •co,6
```

Iteration 10,000

```
om Did ly ow whe fas mats a cll?" "Bugl wat doune fine Cung a dedla treng
of to picined eia seur vartsef emaks ank ele,"," 's Cometoly dobett," seaded
ald acll treisly. "Mr. hibe E wreald shane in
```

Iteration 20,000

```
i.s Pettart abont Hagry uld in feched ach edereadly falkaing untaring to so
had wormary thounging alk how in him, the fire aly intt bet them out. Tearth;
and he hecrees ; a size bat withire wozaring t
```

Iteration 30,000

```
sper the vorg fatir. Ye the washer ally on ha shay of the girer sen"Whehe
of thambo the laage afly of . . . . ... He didne therorgss not viccull shrot
over cnispe, an! inter melenge s welzars weres, Y
```

Iteration 40,000

```
hrourd thet thet dair chillfed thes, that. I mass - Hermiat me into that
aplarted biggiculld pioming now bease? "Harry were rears, didn't retchen
Lome tere; I was oven, " Eand and a pee ford how. bo
```

Iteration 50,000

```
seetes them? He was aid als wach fineves and Geary, what the. The nomeen
my a smigh," said Harry themes, stulled angienatch been scoolst here hombed
pheir plocking having of chuers all seed - they
```

Iteration 60,000

```
uned a seew Rig the live. The than you about stains' was the risemess mum
of sugget, stasn of the trunger aspang. "Astletching theled, wing as
Hermione in into the id to shags feet, and yeaget, then
```

Iteration 70,000

```
iruch's air. Whthe really coll be a gone. "Devel sor from Kauxin't hoden
histots singiring you rage agan, Her, Professo patter Kurracougo beally.
At Ker!" Parvati. Ron coom Marmowing they winding ex
```

Iteration 80,000

```
me their was asly wall. Ron't will wouldn't she neverth aroff; Alkime
of large whic. Hagrine: . . . theyend? Nothally frokend thes could
supfen his swaved, very spillight was thispering elly; his v
```

Iteration 90,000

```
y aronasto insu coulde, before. Harry had for the have he pain applauge him.
It are carly, Curseo was go.. He follimed lorked of hismect to a latiover
trouch Hermeation was had scrobling ous with to
```

Iteration 100,000 (Final)

```
like George, and Beadled all yetr, surkinfies wooked soomely now, Ron're
who were wrom; shosed hand's seuddentians were cloeans table, halled though?"
said well gunger hoveing his yevered and Goors
```

The evolution of the generated text demonstrates several interesting patterns:

- **Early iterations (1-10,000):** The model initially produces random character sequences, gradually learning basic character patterns and beginning to form word-like structures.
- **Middle iterations (20,000-50,000):** The model starts to generate more coherent word structures and begins to capture simple grammatical patterns. Character names like "Harry" begin to appear.
- **Later iterations (60,000-100,000):** The text shows improved sentence structure, consistent use of quotation marks for dialogue, and more frequent appearance of character names like "Hermione," "Ron," and "George."

Although the final text still contains many grammatical errors and nonsensical words, it exhibits a clear improvement in structure and coherence compared to the initial random output.

4 Best Model Output (1000 Characters)

The following passage was generated by the model at iteration 85,000, which achieved one of the lowest smooth loss values (approximately 1.54). Note that line breaks have been manually inserted for display purposes and were not generated by the model.

```
e on, Dooky again," said Mr. Weaslin, but but your stowing an stufl,"
said Rod in," said Dumbledore's students was nevely. "Aho'l, and Harry,
"sompirtion whoch, Ron, how, And Georted, gotweldly caured his fowr
stown it, under coulled every derin-oy of woole naws. "Mosling," said
Molding a tould tore. "Dumbledes wanding years openticalnd anxixcion
lime her saviones not of who seemed it ispation who do that on a tapsed
know last Hermury, Dust Dearlnes, Frose. The coulsend whore fuets, and
Geornes, are to prokentry chowers led retea know, anyon, eyes... Do-blake
incr. Goodars. Ou and George ent. "But' Genece innisemer ovey woodent
to courch? he roumentiniors to up," sain. "What was anpotted. "Don'mogs.
"Every you atturestence evan, puoder on eyes, and Aude is. The threalp
rutmins beched neem inting to George, unliads looking though it deen was
a there. "Pamansthay a tappes one pusncing them is is ears, wooking stoul
wey. "She suppespa compse Marks, a lessed toolers ond-once is to
```

While this generated text still contains many invented words and grammatical inconsistencies, it displays several characteristics that demonstrate the model's learning:

- **Dialogue Structure:** The model has learned to use quotation marks to indicate dialogue, with proper attribution using "said" constructions.
- **Character Names:** The text includes recognizable character names from the Harry Potter series, such as "Harry," "Ron," "Dumbledore," "George," and "Hermione."
- **Sentence Structure:** Many segments follow basic English sentence patterns, but far from perfect.
- **Punctuation:** The model uses periods, commas, and quotation marks appropriately in some cases.

These features suggest that the RNN has captured significant aspects of the text's structure, even if the semantic content is often nonsensical.

5 Conclusion

This implementation successfully demonstrates the capabilities and limitations of a vanilla RNN for character-level text generation. The model shows clear improvement over training, learning character patterns, word structures, and basic grammatical rules. However, it also illustrates the inherent limitations of simple RNN architectures in maintaining long-term coherence and generating semantically meaningful text.

Future improvements could focus on several areas:

- **Advanced sampling strategies:** As implemented in this project, temperature sampling and nucleus sampling can significantly improve text quality. Lower temperature values ($T \in [0,1)$) make the distribution more peaked, increasing text quality but potentially reducing diversity. Similarly, nucleus sampling with various threshold values (θ) allows for dynamic adjustment of the sampling pool based on probability distributions, helping to balance text quality and diversity.
- **Training optimization:** The current implementation trains on sequential text segments. Using randomly selected chunks from the text could improve convergence and generalization. Additionally, implementing batch training (beyond batch size 1) could accelerate training and potentially improve model performance.
- **Computational efficiency:** Several optimizations could speed up training, including pre-computing matrix operations outside time-step loops, leveraging the sparsity of one-hot encoded inputs, and using specialized functions like `np.outer()` for gradient computations.

- **Architecture improvements:** Beyond vanilla RNNs, more sophisticated architectures like LSTM or GRU cells could better capture long-range dependencies in the text, potentially leading to more coherent generated content.

These improvements could enhance both the quality of generated text and the efficiency of the training process, making character-level RNN text generation more practical for longer texts and larger vocabularies.