Deep learning homework 3

Matej Miočić (63180206)

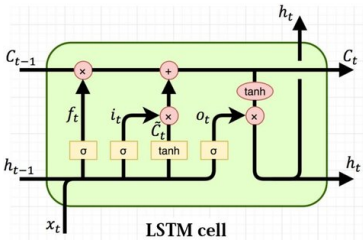
1 Introduction

In this homework we were tasked with character-level text prediction. First we implemented one layer LSTM as in PyTorch documentation. Then we implemented transformer-like network according to the original transformer paper [1]. We trained each model on a dataset which consists of a subset of works of Shakespeare. In the end we show some examples of the generated text using the Top-K and the greedy sampling for different sequence lengths during training.

2 Experiments - Character-level text generation

Character-level text generation is a type of natural language processing technique that involves training a machine learning model to generate text one character at a time. Unlike word-level text generation, which generates text by predicting whole words, character-level text generation allows for more flexibility and creativity in generating text, as the model can learn to generate new words or even new types of words. This technique has many practical applications, such as generating new text for chatbots or writing assistance tools, and it has become increasingly popular in recent years due to advancements in deep learning and natural language processing.

2.1 LSTM



$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right)$$
$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f\right)$$
$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o\right)$$
$$\tilde{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g\right)$$
$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right)$$
$$h_t = \tanh(C_t) * o_t$$

Figure 1: LSTM cell with equations. [2]

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that is designed to capture long-term dependencies in sequential data. LSTMs are particularly well-suited for character-level prediction tasks because they can effectively model the complex relationships between characters that occur over long sequences. Unlike standard RNNs, which suffer from the vanishing gradient problem, LSTMs use a more sophisticated gating mechanism to selectively retain or forget information from previous time steps. This allows LSTMs to maintain a memory of past inputs that is crucial for accurate character-level predictions.

We implemented a one layer LSTM as depicted in Figure 1. In training we used:

- batch_size = 256,
- hidden_dim = 256,
- chunk_len = 128 and 256
- epoch = 30,
- optimizer = Adam,
- γ = 0.005.

2.2 Transformers

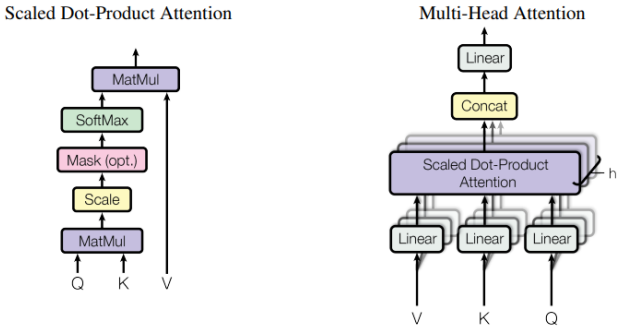


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel. [1]

Transformers are a type of neural network architecture that has gained popularity in recent years due to their ability to effectively model long-range dependencies in sequential data. Unlike recurrent neural networks, which process sequential data in a step-by-step fashion, transformers process all elements of the sequence in parallel, making them much faster and more efficient. For character-level prediction tasks, transformers have shown promising results because they can effectively capture complex relationships between characters that occur over long sequences. Additionally, transformers use a

self-attention mechanism that allows them to attend to different parts of the sequence at each layer of the network, making them highly adaptable to a wide range of character-level prediction tasks.

Our implementation consisted of 5 blocks of Multi-Head Attention with masking (Figure 2) + Positional Ecoding. The masking operation is used in the scaled dot-product attention module to prevent certain positions from attending to others. A mask is used to prevent positions from attending to subsequent positions, ensuring that the predictions for position i can depend only on known outputs at positions less than i. We computed Multi-Head Attention as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, ..., head_h)W^O, \tag{1}$$

where $head_i$ = Attention(QW_i^Q, KW_i^K, VW_i^V). Each attention is then calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\text{masking}(\frac{QK^T}{\sqrt{d_k}})\right)V. \tag{2}$$

In training we used the following parameters:

- batch_size = 256,
- num_neuron = 64
- n_head = 8
- dim_model = 256
- chunk_len = 128 and 256
- epoch = 50,
- optimizer = Adam,
- γ = 0.0006.

Results

We train the models using cross-entropy loss, and evaluate their performance using the Top-K and the greedy sampling. Top-K sampling selects the top K probable words from the probability distribution of the next word, and then randomly chooses one of them. Greedy sampling, on the other hand, selects the word with the highest probability and outputs it. While greedy sampling is faster and more efficient, it can result in repetitive and less diverse text. Top-K sampling, on the other hand, can produce more diverse outputs but may also lead to inconsistency in language and syntax.

We show examples of generated text in the appendix (next page) for both LSTM and Transformer implementation. For the latter we also show the different outputs with different sequence lengths during training (128 and 256). We observe that LSTM with greedy sampling produces pointless outputs since it just repeats the words "the" and "and". Overall Transformer produces nicer looking outputs – even with greedy sampling. We observe slight difference in the produced outputs when tranining with different sequence lengths. The outputs with longer sequence lengths are a bit more coherent.

3 Conclusion

This homework explored the task of character-level text prediction and compared the performance of LSTM and transformer models on a dataset consisting of works of Shakespeare. Both models were trained using cross-entropy loss and evaluated using Top-K and greedy sampling techniques. The results showed that the transformer model outperformed the LSTM model in terms of generating more diverse and grammatically correct text. This report demonstrates the potential of character-level text generation techniques in natural language processing tasks and highlights the effectiveness of transformer models in handling long-range dependencies in sequential data.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[2] Savvas Varsamopoulos, Koen Bertels, and Carmen Garcia Almudever. Designing neural network based decoders for surface codes. 2018.

4 Appendix

The input was: ”Here’s to my love! O true apothecary! Thy drugs are quick.”

4.1 LSTM

Chunk len = 128

Top-K sampling	Greedy sampling
ARDI ares t hese te ho moned. ARDUE:T: AROLO: Thin the tinges, ARI woreds, aresen whees oure me ant at mine thee, Th ton d and th and th iste at hing thero f atis te my bure thoe mor whient tist and store thin mon ote me mane tithe sanden ow ith mored sen t inger tou sth is th the me fof om tisest, wise stouren don teant tese tharen thes ord and an thar my andot he winth ite alll bear ate hind	ARDUCENThe the and the

Chunk len = 256

Top-K sampling	Greedy sampling
hourse tis ther ind thout arthe thies thith meresand on what h oword, it his to we ande te at ill the tha terand, tithie nde ale t hander wher an te mean oto hers tho ter sterang,e mese thane my, tithe, B thine stho ware ard owo fe te mingou mases hesers willi my ome mayus bor aser ond wiste arte thees ofrist hore s wind ont he wal arto mathere, Ther thant ates theas athe s asthour minga berest	ARDUCENET: The the and the an

4.2 Transformer

Chunk len = 128

Top-K sampling	Greedy sampling
KING EDWARD IV: See that thy master hath mistred her? LADY GREY: The fear upon the most repose the maid: Wise mine hath beeen, to mighty nature’s My countenance he addely at homely. ARIEL: That hourt is, by goone, Were he shalll at harrm in at. PROSPERO: But, after the misinate had so my hot advisege, Whose mightiry tent me that his mattted maidst a blastful mast death. This is no corn, That	We attendance his head; and there she comes Of the same so much on the proud here As my house so much trick to my sovereign, And so I am. LEONTES: We are as much as here, With here as much as hererset as my crossen, And as a little last of love and here: I have beeen light of love in hand, That I have heard it, or show our seatson’d, And so any other freedom and my height A bloot against and ash

Chunk len = 256

Top-K sampling	Greedy sampling
Third Come, were not too long a world, with a fool! CLARENCE: KING EDWARD IV: Ay, what of turn bands what it doth; Over of your blood have I offer’d That I have done. Kish’d unconsellow More than the sound of nobld that losss My fancy prought them to pay hims. CLEOMENENES: That would be my heart into so promounted That he cankerled to my unlord’s tongue We drown’d doubld lordship of lord? Ed	BUCKINGHAM: Thy love that stay’d this command, and love, and so proclamo! thy other divine unconstant pennitent bar, for thy loving love, and thy hand, though she stood is a sun of hail, your charges ynnew now, now that was shalll pay to put To stay all the points of sorrt ochasie That purgame this fooding. You are come any cotor, And say ’Tybalt’s beatstitute of our back, Is it was baby-bustare