Language Modeling

with LSTM

AhmadrezaBagherzadeh

Chapter 1

Introduction

Language modeling refers to the task of predicting the next word or character in a sequence of text given the context of the previous words. This can be useful for applications such as language translation, text summarization, and chatbots.

There are several models that have been used in language modeling, each with its own unique characteristics and advancements. Let's explore a few of them:

1.1

N-gram Models: N-gram models are simple and widely used for language modeling. They predict the probability of a word based on the previous N-1 words. For example, a bigram model considers the probability of a word given its preceding word. N-gram models are computationally efficient but struggle with capturing long-range dependencies and understanding context beyond the immediate history.

1.2

Feedforward Neural Networks: Feedforward neural networks, also known as multilayer perceptrons (MLPs), have been applied to language modeling. These models take a fixed-length sequence of words as input and use hidden layers to learn representations and predict the next word. However, they suffer from the inability to handle variable-length sequences and struggle with capturing sequential dependencies.

1.3

Recurrent Neural Networks (RNNs): RNNs are designed to process sequential data by maintaining hidden states that capture information from previous steps. They have been widely used for language modeling due to their ability to model context and capture dependencies over time. However, traditional RNNs suffer from the vanishing gradient problem, which hinders their ability to capture long-term dependencies.

1.4

Long Short-Term Memory (LSTM): LSTMs are a type of RNN architecture that address the vanishing gradient problem. They use memory cells with input, forget, and output gates to selectively retain and forget information over time. LSTMs have proven effective in capturing long-range dependencies and have become a popular choice for language modeling tasks.

1.5

Transformer Models: Transformer models, such as the famous GPT (Generative Pre-trained Transformer) series, have revolutionized language modeling. They rely on self-attention mechanisms to capture global dependencies in the input sequence. Transformers excel at modeling long-range dependencies and have achieved state-of-the-art performance in various NLP tasks, including language modeling.

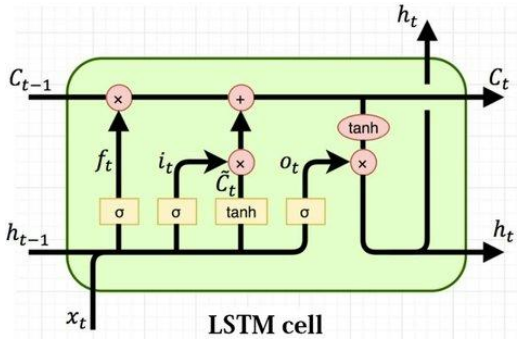
These are just a few examples of models used in language modeling. Each model has its own strengths and weaknesses, and their effectiveness depends on the specific task and dataset. Researchers continue to explore and develop new models to improve language understanding and generation.

In this project, we aim to first train an LSTM network on the wikitext2 dataset, and then improve the results using the AWD-LSTM paper.

Chapter 2

MODEL

LSTM stands for Long Short-Term Memory, which is a type of recurrent neural network (RNN) architecture. It was designed to address the vanishing gradient problem that occurs in traditional RNNs, which struggle to capture long-term dependencies in sequential data. LSTMs are a type of recurrent neural network (RNN) designed to handle sequential data such as text. Unlike traditional RNNs, LSTMs have the ability to selectively forget or remember information from previous time steps, which makes them well-suited for tasks that require long-term memory. In NLP, LSTMs have been used for tasks such as language modeling, sentiment analysis, and machine translation.



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

The key advantages of LSTMs over traditional RNNs are:

Handling long-term dependencies: LSTMs can handle long-term dependencies in text sequences much better than traditional RNNs. Selective memory: LSTMs allow for selective memory, where certain information can be forgotten or remembered based on the context. Improved training stability: LSTMs are less prone to vanishing gradients and exploding gradients during training, making them easier to train. In summary, LSTMs are a powerful tool for handling sequential data in NLP tasks, and they have been shown to achieve state-of-the-art results in many applications. Language modeling is an important application of LSTMs, as it allows for the generation of coherent and fluent text.

LSTMs are widely used in various NLP tasks, including language modeling, machine translation, sentiment analysis, and speech recognition. They excel at processing and understanding sequences of data by selectively retaining and forgetting information over time.

The key component of an LSTM is its memory cell, which allows the network to store and access information for long periods. The cell has three main components: an input gate, a forget gate, and an output gate. These gates regulate the flow of information, enabling the LSTM to learn and remember relevant patterns in the input

sequence.

LSTMs have proven to be effective in capturing long-range dependencies and handling sequential data with varying time lags. They have become a popular choice for NLP tasks due to their ability to model context and capture semantic relationships within text data.

Overall, LSTMs have played a significant role in advancing the field of NLP and have contributed to the development of various state-of-the-art models.

Chapter 4

Dataset

The wikitext2 dataset is a commonly used dataset in natural language processing (NLP) tasks. It is derived from Wikipedia articles and consists of a large collection of text data. The dataset is often used for tasks such as language modeling, text generation, and text classification. It provides a diverse range of topics and writing styles, making it suitable for training and evaluating various NLP models.

The wikitext2 dataset is a comprehensive collection of over 100,000 human-generated wiki articles, carefully curated to provide a balanced mix of topics and styles. This diverse set of texts has been used to train and evaluate state-of-the-art machine learning models for a variety of NLP tasks, including text classification, sentiment analysis, named entity recognition, and question answering. By leveraging this rich and dynamic dataset, researchers and developers can improve the performance of their algorithms and achieve better results on real-world applications. With its unparalleled scope and quality, the wikitext2 dataset is an essential resource for anyone working in the field of natural language processing.

Here you can see a sample of the dataset text:

```
= Gold dollar =

The gold dollar or gold one @-@ dollar piece was a coin struck as a regular issue by the
A gold dollar had been proposed several times in the 1830s and 1840s , but was not initi
Gold did not again circulate in most of the nation until 1879 ; once it did , the gold d
```

```
= Super Mario Land =

Super Mario Land is a 1989 side @-@ scrolling platform video game , the first in the Sup
At Nintendo CEO Hiroshi Yamauchi 's request , Game Boy creator Gunpei Yokoi 's Nintendo
Initial reviews were laudatory . Reviewers were satisfied with the smaller Super Mario B
```

```
= = = Sinclair Scientific Programmable = = =

The Sinclair Scientific Programmable was introduced in 1975 , with the same case as the
It had 24 @-@ step programming abilities , which meant it was highly limited for many pu
However , included with the calculator was a library of over 120 programs that that perf
```

Chapter 5

Use LSTM

Initially, we preprocessed and transformed the wikitext2 dataset into windows of dimensions M * sequence length that M is The total number of tokens in the dataset that have been vectorized. The training dataset contains 2,049,990 tokens, which, after being lowercased, result in a total of 28,781 unique tokens. Each token appears at least 3 times in the dataset, and we have decided to use all of the tokens for training. After passing through the dataloader, it is ready to enter the training phase. In each step, the dataloader retrieves a batch of data with a length of batch size * sequence length and feeds it to the model. Then, the calculated loss and gradient are computed. Based on the conducted experiments, we have decided to tune the hyperparameters in dataset as shown below:

- Batch Size : 20

- Sequece Length : 35

We use a two-layer LSTM and achieve the best results with this initial model by setting hyperparameters such as hidden size and embedding dim. As you can see below, we attempted to set the best hyperparameters for the model through experimentation:

- Learning rate: 0.5

- Weight decay: 1e-6

- Batch size: 20

- Number of epochs: 40

- Hidden size: 400

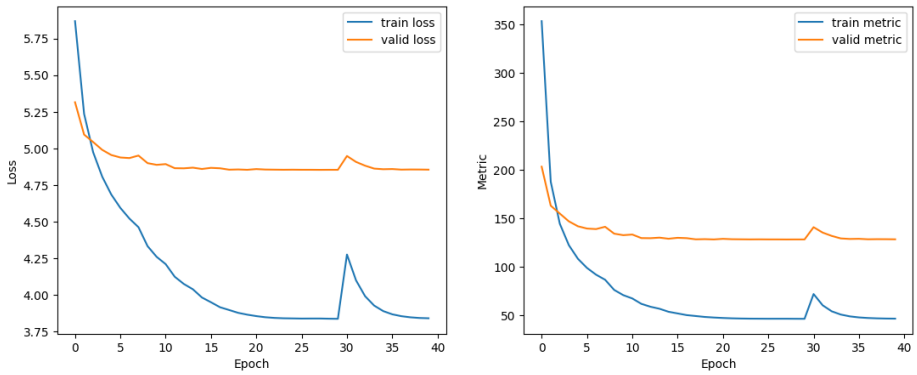
- Embedding dimension: 380

- Dropout rate(in LSTM layers): 0.2

We conducted several experiments to find these optimal hyperparameters and believe that they will yield good results for the model.

Chapter 6

Result



Model	Parameters	Validation	Test
LSTM-2Layer	24.8M	128.3	121.7

After 40 epochs of training, we achieved a perplexity of 128.3on the validation set and 121.7on the test set.

After the training and testing process,we provided the model with sentences to complete,and we evaluated the model's performance as shown below:

In this year , as he was in the first game of the season , and he was named the team ' s captain .

This recommended approach helps prevent potential issues related to computational graph tracking from the blocked earth .

I think the movie was good.

Chapter 7

AWD-LSTM

AWD-LSTM, which stands for "ASGD Weight-Dropped LSTM", is a language model architecture that aims to improve the performance of recurrent neural networks (RNNs) on various natural language processing (NLP) tasks. It was introduced by Stephen Merity, Nitish Shirish Keskar, and Richard Socher in their 2017 paper titled "Regularizing and Optimizing LSTM Language Models."

AWD-LSTM incorporates several techniques to address common challenges in training RNNs, such as overfitting and difficulty in capturing long-term dependencies. Here are some key features of AWD-LSTM:

1. ASGD (Asynchronous Stochastic Gradient Descent): AWD-LSTM employs the ASGD optimization algorithm, which helps accelerate training by updating the model parameters asynchronously.

2. Weight-Dropping: In order to regularize the model and prevent overfitting, AWD-LSTM introduces a technique called weight-dropping during training. It randomly stochastically drops some recurrent connections, forcing the model to learn more robust representations.

3. Regularization: AWD-LSTM incorporates different forms of regularization, including variational dropout and embedded dropout, to further stabilize the training process and reduce overfitting.

4. Gradient Stabilization: A technique called "gradient stabilization," which includes modifying the backpropagation algorithm, is employed to alleviate the vanishing and exploding gradients problem often encountered in RNNs.

The AWD-LSTM architecture has demonstrated state-of-the-art performance on various language modeling tasks, such as text classification, sentiment analysis, and machine translation. Its effectiveness in handling long-term dependencies and reducing overfitting has made it a popular choice for NLP tasks.

Chapter 8

use AWD-LSTM

Here are some hyperparameters mentioned in the paper along with the corresponding methods used:

1. Weight Decay: Weight decay is used to control the complexity of the model by adding a penalty term to the loss function based on the magnitude of the weights. In AWD-LSTM, weight decay with a coefficient of 1.2e-6 was applied.

2. Gradient Clipping: Gradient clipping is a technique to prevent exploding gradients by limiting the magnitude of gradients during backpropagation. The paper mentions that a threshold of 0.25 was used to clip the gradients in AWD-LSTM.

3. Embedding Dropout: In addition to the regular dropout, AWD-LSTM applies dropout to the embedding layer. The paper states that an embedding dropout rate of 0.1 was used.

dropout-(em) : 0.1

4. Variational Dropout: Variational dropout is a modified version of dropout that maintains the dropout masks consistent across time steps in recurrent connections. It helps regularize the recurrent connections while capturing long-term dependencies. AWD-LSTM uses variational dropout with a dropout probability of 0.5 for recurrent connections. This section has three dropout layers, which are applied in order to the output of the embedding layer, the initial LSTM layers, and the last LSTM layer.

dropout-i = 0.65

dropout-l = 0.3

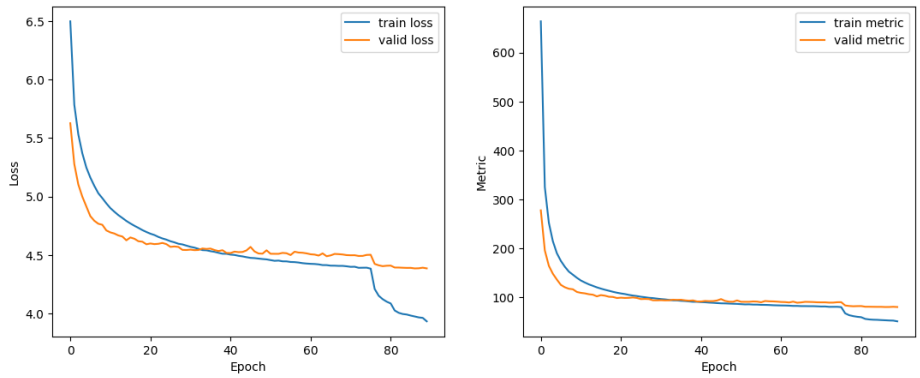
dropout-o = 0.4

4.Weight tying: Weight tying in AWD-LSTM is a technique used to improve the training stability and convergence rate of the model. By sharing the weights across different layers, the gradients can be more easily propagated through the network, leading to faster learning. Additionally, weight tying can help to reduce overfitting by preventing each layer from adapting to the input data too much. However, it's important to note that weight tying may not always lead to better performance, so it should be carefully tuned and evaluated during training.

These hyperparameters, combined with the techniques discussed earlier, contribute to the overall performance and regularization of the AWD-LSTM language model.

Chapter 9

Result



Model	Parameters	Validation	Test
AWD-LSTM-3Layer(tied)	31.75M	80.27	77.11

After 90 epochs of training, we achieved a perplexity of 80.27 on the validation set and 77.11on the test set.

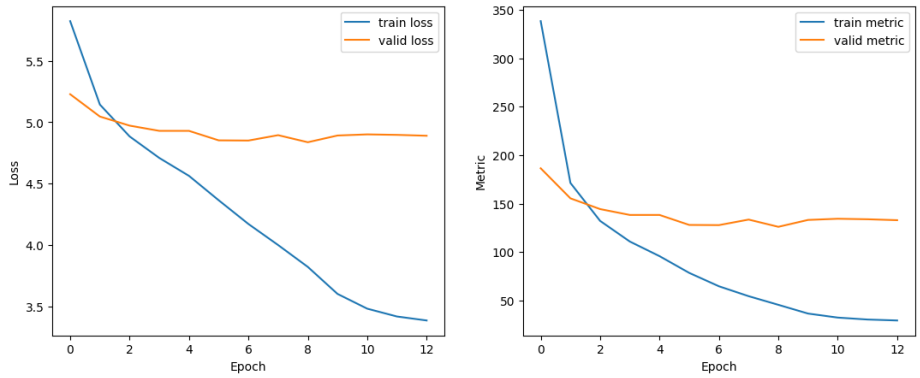
After the training and testing process,we provided the model with sentences to complete,and we evaluated the model's performance as shown below:

I think this movie is the most beautiful thing i've ever seen .

Chapter 10

Methods Effect

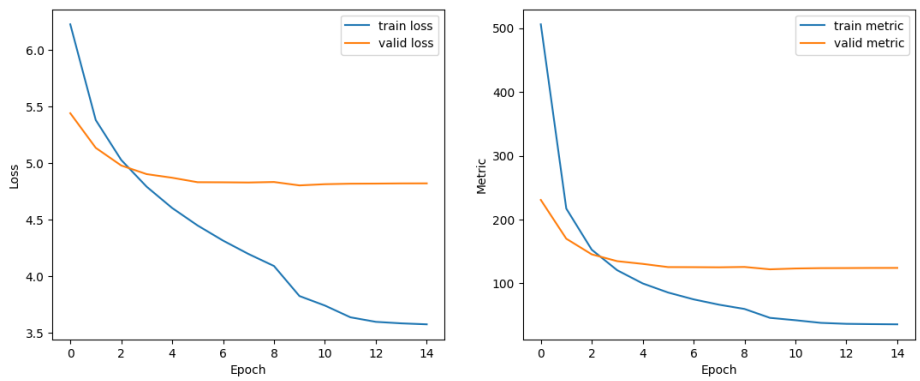
1Weight Drop



Model	Parameters	Validation	Test
AWD-LSTM-3Layer(h = 650	39.75M	126	118.6

.....

2.NTASGD

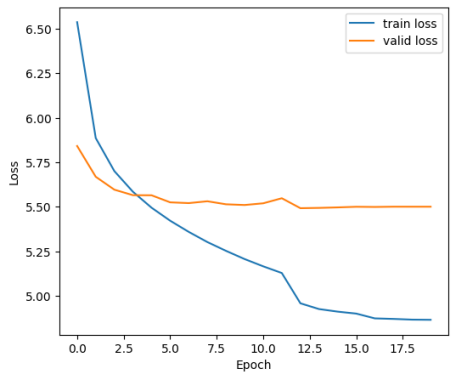


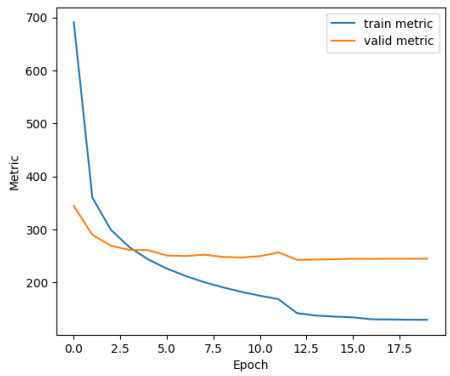
Model	Parameters	Validation	Test
AWD-LSTM-3Layer(h = 650)	39.75M	121.96	116.55

Chapter 11

Methods Effect

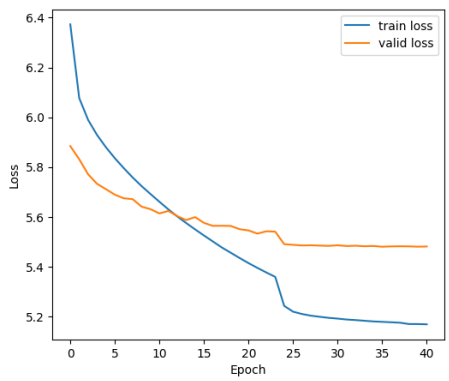
3.Combination WD and NTASGD

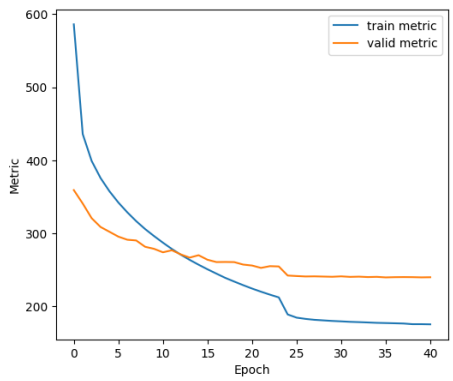




Model	Parameters	Validation	Test
AWD-LSTM-3Layer(h = 650)	39.75M	242.87	227.84

4.Variational Dropout



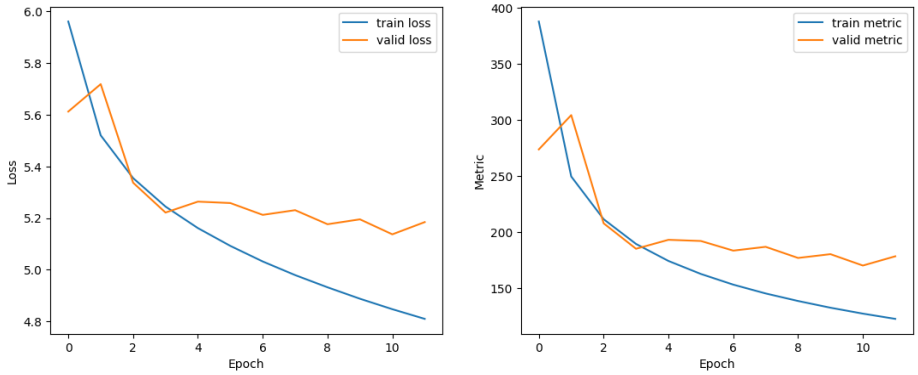


Model	Parameters	Validation	Test
AWD-LSTM-3Layer(h = 650)	39.75M	239.79	224.85

Chapter 12

Methods Effect

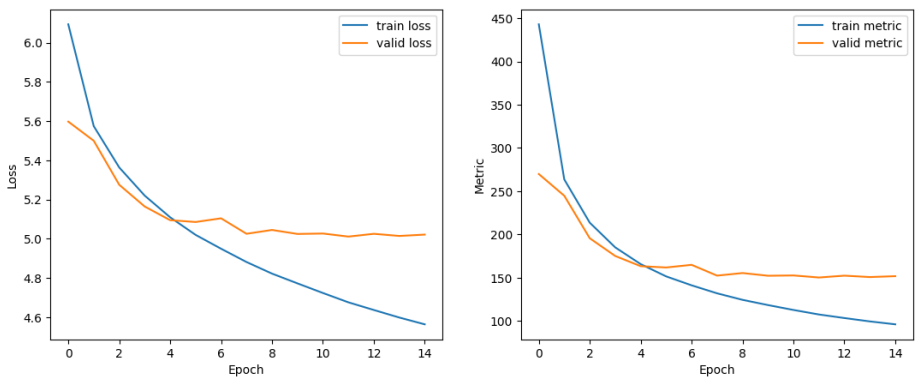
5.Combination WD, NTASGD and VD



Model	Parameters	Validation	Test
AWD-LSTM-3Layer(h = 650)	31.75M	170.28	159.48

.....

6.Embedding Dropout

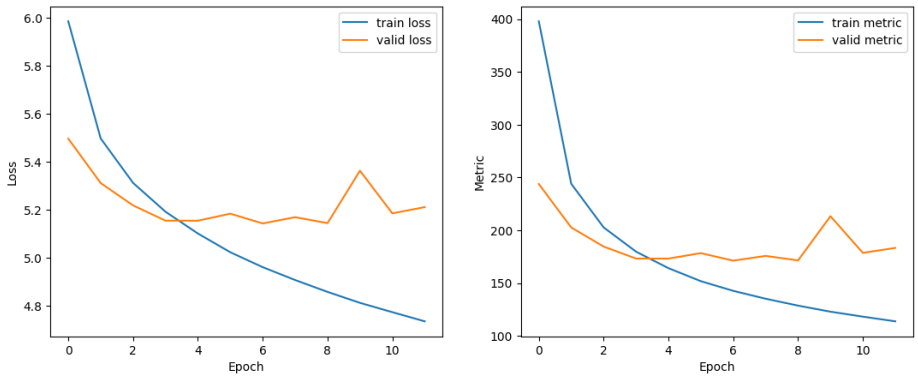


Model	Parameters	Validation	Test
AWD-LSTM-2Layer(h = 256)	19.57M	150.15	140.93

Chapter 13

Methods Effect

7.Combination WD, NTASGD, VD and ED

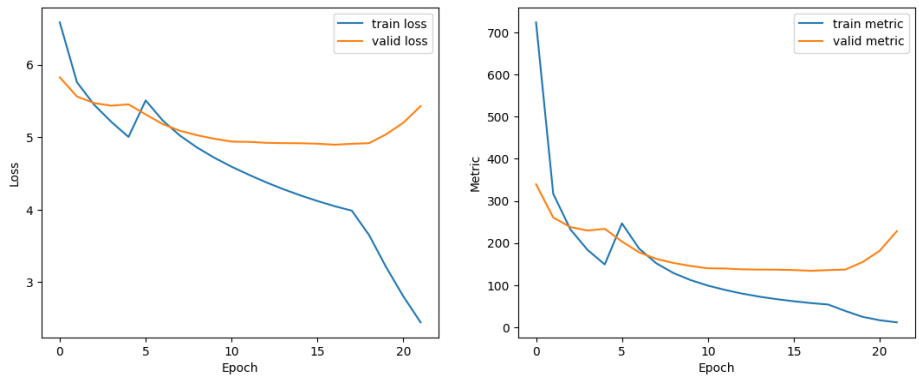


Model	Parameters	Validation	Test
AWD-LSTM-2Layer(h = 256)	19.57M	171.28	159.82

Chapter 14

Methods Effect

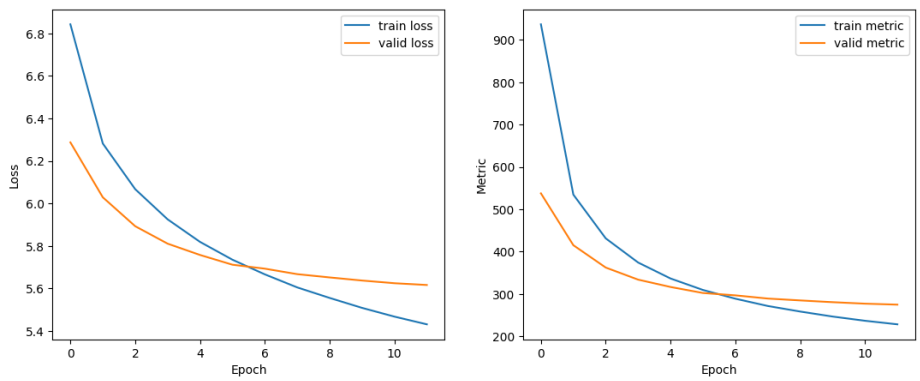
8.Weight Tying



Model	Parameters	Validation	Test
AWD-LSTM-3Layer(h = 650)	31.75M	133.91	126.44

.....

9. Combination WD, NTASGD, VD, ED and WT



Model	Parameters	Validation	Test
AWD-LSTM-3Layer(h = 1150)	16.38M	274.93	250.16

Chapter 15

Some Det ails

Version of libraries:

Python»»»»»»3.10.12

tor ch»»»»»»»2.0.1+cu118

tor chtext»»»»»0.15.2+c pu

nump y»»»»»»»1.23.5

pandas»»»»»»1.5.3

tqdm»»»»»»»4.66.1