Shanghai Jiao Tong University
AI3611

Language Model

Yan Han (519030910404)

June 9, 2022

1 Introduction

Neural networks[4][6][3] have become popular in language modeling for speech recognition. In this work, we mainly introduce three models - GRU, LSTM, Transformer - and explore how different hyperparameters influence the model performance.

Specifically, we use *perplexity(PPL)* to measure the model performance, which shows the average divergence of each prediction. Suppose that we have a N-word sentence $S = \{w_1, \ldots, w_N\}$, then

$$PPL(S) = (\mathbb{P}(w1 \ldots w_N))^{-1/N} \tag{1}$$

$$= (\prod_{i=1}^{N} \mathbb{P}(w_i | w_1 \ldots w_{i-1})^{-1/N} \tag{2}$$

$$= \exp\left(-\frac{1}{N} \sum_{i=1}^{N} \log \mathbb{P}(w_i | w_1 \ldots w_{i-1})\right) \tag{3}$$

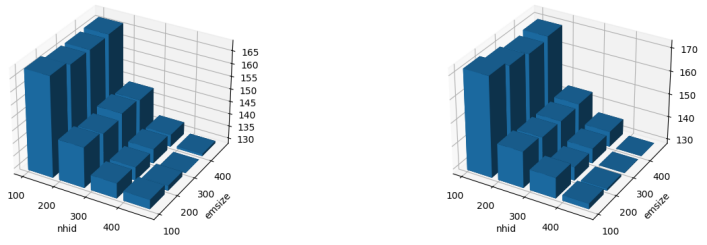
Moreover, we utilize [7] built by HuggingFace to try more advanced models, such as BERT[1], GPT2[5]. While BERT is popularly used in masked language modeling, we mainly try GPT2, a model pretrained on a very large corpus of English data in a self-supervised fashion, which is more suitable for causal language modeling. The source code is available at `https://github.com/wolfball/Intellisense-Cognitive-Practice`.

2 Experiment and Result

Experiment setting We select GigaSpeech [2] as our dataset, which has been split into train set, validation set and test set. First, we load all three dataset and build a dictionary. Meanwhile, all words are transferred to responding integer IDs according to the dictionary. Next, we divide the dataset into B parts, where B is the batch size (20 in the baseline for training). Then feed data to the model and we will get the output. And the negative log likelihood loss(NLLLoss) is applied to compute the loss. For optimizer, the updated gradient is clipped by 0.25, that is to say, all gradients less than 0.25 will be valued as 0.25. Finally, the gradient multiplied by learning rate (the initial value is 20 and will be quarter if not improvement has been seen in the validation

parameter	value	parameter	value
nlayers	2	lr	20
nhid	200	batch size	20
emsize	200	dropout	0.2
nhead	2	epoch	20

Table 1: baseline configuration



(a) nlayers=2 (b) nlayers=3

Figure 1: LSTM - model related hyperparameters. *Z-axis means test ppl.*

set) is directly added to the parameters of the network. All the experiments are training with 20 epochs and the parameter sizes are smaller than 60M. The parameters of the baseline is in Table 1.

Hyper parameters In general, the hyperparameters can be grouped into two types: model related and training ralated. Model related parameters control the representational power and capacity of the model, such as the number of layers (*nlayers*), the embedding size (*emsize*), the dimension of hidden feature (*nhid*) and so on; training ralated parameters specify how the model is updated, such as the learning rate (*lr*), the batch size (*batch_size*), the dropout rate (*dropout*) and so on. In this chapter, while keeping all other variables the same, we select several values to the above parameters and analyse experimental results.

2.1 Model1: LSTM

2.1.1 Model Related Hyperparameters

We set $nlayers \in [2, 3], emsize \in [100, 200, 300, 400], nhid \in [100, 200, 300, 400]$. The result is in Figure 1. It is vivid that the model achieves better performance with higher nhid and emsize. This may due to the improvement of the capacity of the model. Besides under same configuration of emsize and nhid, 3-layer-model is better than 2-layer-model. When nlayers is equal to 2, the lowest validation PPL is 120.9 and when equal to 3, the lowest validation PPl is 120.7. The model with the largest number of parameters is 26.05M.

2.1.2 Training Related Hyperparameters

We try $lr \in [5.0, 10.0, 20.0, 25.0, 30.0]$, $batch\_size \in [10, 15, 20, 25, 30]$, and $dropout \in [0.1, 0.2, 0.3, 0.4, 0.5]$. The result is in Figure 2. First, when batch size is equal to 25, the model get the best result. Too small batch size restricts the model to "see" more data while too large batch size will result in oversmoothing. Second, this model is more suitable for small learning rate which can help it update step by step and avoid from "jumping"

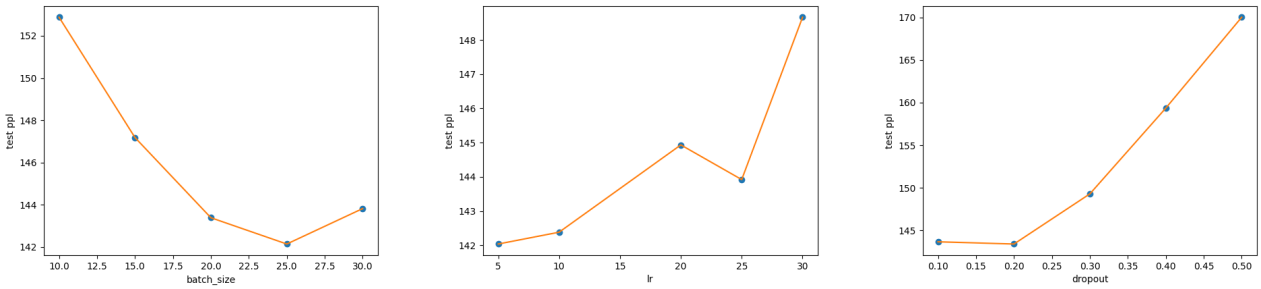
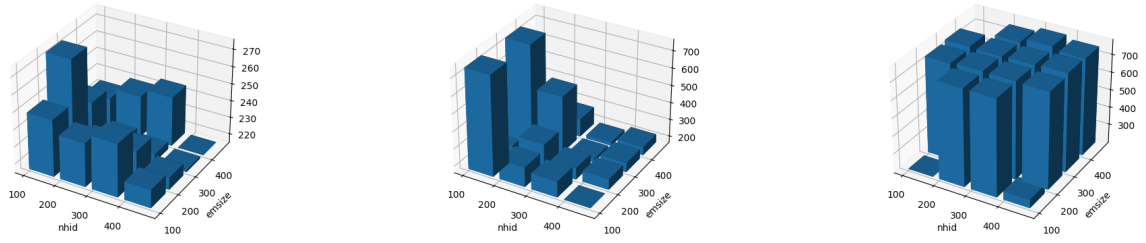


Figure 2: LSTM - training related hyperparameters.



(a) nlayers=2 (b) nlayers=3 (c) nlayers=4

Figure 3: GRU - model related hyperparameters. *Z-axis means test ppl.*

out of the optimal solution. Third, dropout with value of 0.2 is the optimal setting from the experiments. When the dropout is too large, it will be difficult for model to learn, which is the potential reason for the visualization.

2.2 Model2: GRU

2.2.1 Model Related Hyperparameters

We set $nlayers \in [2, 3, 4], emsize \in [100, 200, 300, 400], nhid \in [100, 200, 300, 400]$. The result is in Figure 3. Compared with LSTM, the results of GRU model are more unstable which may be due to the insufficient capacity. Generally speaking, when nlayer is equal to 2 or 3, the model has better performance with higher nhid and emsize. However, when nlayer is equal to 4, the model will fail into overfitting, thus resulting in bad performance.

2.2.2 Training Related Hyperparameters

We set $lr \in [5.0, 10.0, 20.0, 25.0, 30.0]$, $batch\_size \in [10, 15, 20, 25, 30]$, and $dropout \in [0.1, 0.2, 0.3, 0.4, 0.5]$. The result is in Figure 4. Generally speaking, within a proper range, higher batch size with less learning rate will promote the model to learn the feature while training. Also, a moderate dropout(=0.2) is more appropriate.

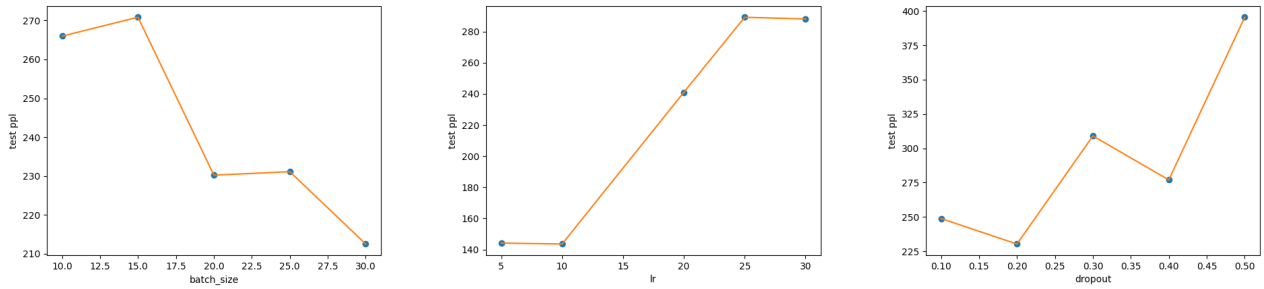
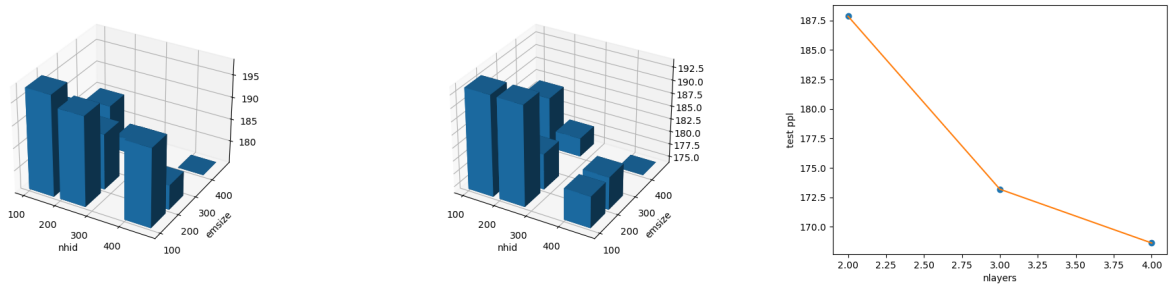


Figure 4: GRU - training related hyperparameters.



(a) nhead=2 (b) nhead=10 (c) nlayers

Figure 5: Transformer - model related hyperparameters. *Z-axis means test ppl.*

2.3 Model3: Transformer

2.3.1 Model Related Hyperparameters

In Transformer, the number of the heads (*nhead*) is a characteristic parameter which implies how many aspects can be observed in data. We set $nhead \in [2, 10], emsize \in [100, 200, 400], nhid \in [100, 200, 400]$ and try $nlayers \in [2, 3, 4]$ individually to test its impact. The result is in Figure 5. One thing to note: all experiments are executed when the learning rate is 5.0, and the reason will be discussed in 2.3.2. The results are still predictable, as the test ppl rises when nhead, nhid, embsize and nlayers increase.

2.3.2 Training Related Hyperparameters

We set $lr \in [5.0, 10.0, 20.0, 25.0, 30.0]$, $batch\_size \in [10, 15, 20, 25, 30]$, and $dropout \in [0.1, 0.2, 0.3, 0.4, 0.5]$. The result is in Figure 6. Generally speaking, within a proper range, large batch size, less learning rate and less dropout will perform better. Especially in the experiments about learning rate, a switch from 10.0 to 20.0 will result in world apart.

3 Advanced Language Model–GPT2

The core idea of GPT2 is that unsupervised pre-training model can be used to perform supervised tasks. By using a token to represent tasks, GPT2 models all questions as $p(output|input, task)$ rather than $p(output|input)$ which is used in supervised tasks. Due to the universality of language modeling, it can promote multiple tasks.

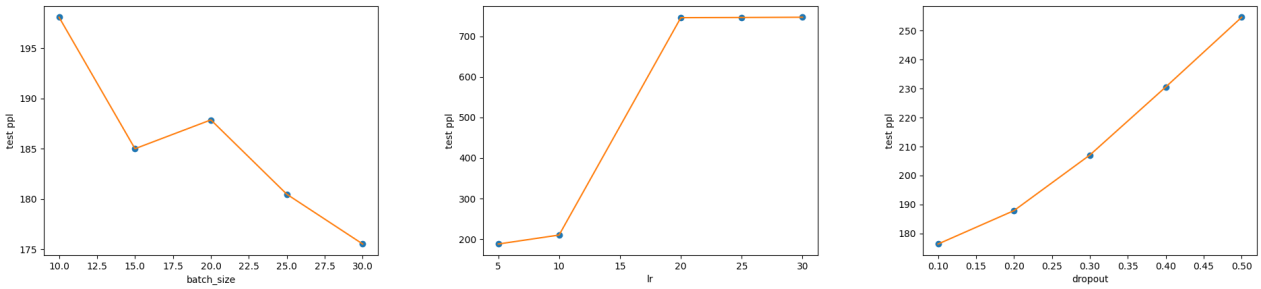
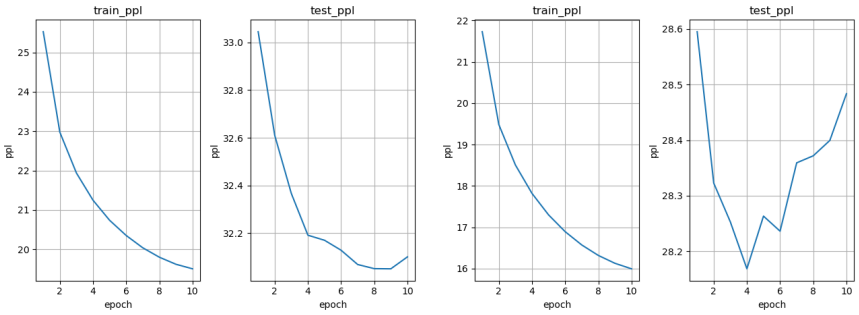


Figure 6: Transformer - training related hyperparameters.



(a) DistilGPT2 (b) GPT2

Figure 7: The Result on DistilGPT2 and GPT2 Model

Here we mainly try DistilGPT2 and GPT2. As a simplified version of GPT2, DistilGPT2 has 6 layers, 768 dimension and 12 heads, totalizing 82M parameters (compared to 124M parameters for GPT2) for experiment.

After 10 epochs, the lower test ppl on DistilGPT2 and GPT2 are 32.05 and 28.17 respectively, which are extremely lower than the above experiments (See Figure 7). And GPT2 tends to be overfitting after about 4 epoches.

4 Conclusion

In summary, we have tried three basic models (LSTM, GRU, Transformer), and mainly six hyper parameters to check their influences. Supported by lots of experiments, LSTM is more suitable for modeling this *GigaSpeech* dataset. And within a proper range, large batch size ,less learning rate and less dropout are more acceptable. Besides, more layers in the network with large embedding size and hidden dimension can improve the capacity of the model which contributing to a lower PPL.

Moreover, we choose an advanced language model – GPT2 – to further improve the performance. And the best PPL with its configuration is listed in Table 2. For better visualization of the experiments, we also draw train and valid PPL trends for each epoch of the training phase by using Tensorboard (See Figure 8 for example).

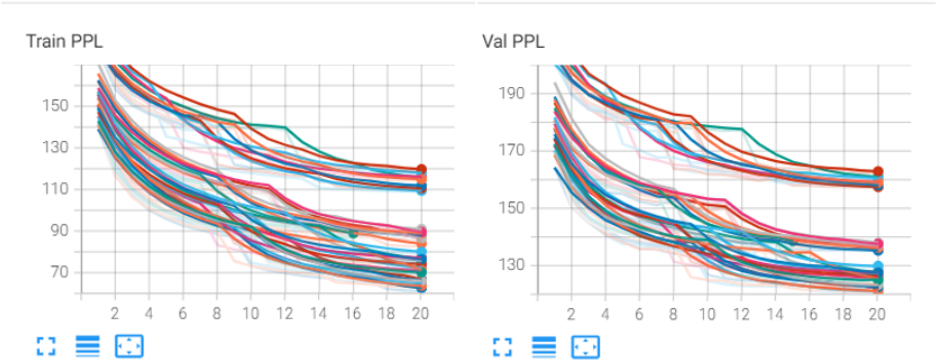


Figure 8: Tensorboard Visualization of Experiments on Different nlayers, nhid and emsize.

model	nlayers	nhid	emsize	PPL
LSTM(params=59.74M)	3	800	800	125.07
model	nlayers	nhead	emsize	PPL
DistilGPT2	6	12	768	**30.1**
GPT2	12	12	768	**28.2**

Table 2: Best Configuration

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[2] Guanbo Wang Jiayu Du Wei-Qiang Zhang Chao Weng Dan Su Daniel Povey Jan Trmal Junbo Zhang Mingjie Jin Sanjeev Khudanpur Shinji Watanabe Shuaijiang Zhao Wei Zou Xiangang Li Xuchen Yao Yongqing Wang Yujun Wang Zhao You Zhiyong Yan Guoguo Chen, Shuzhou Chai. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. In *Proc. Interspeech 2021*, 2021.

[3] Kazuki Irie, Albert Zeyer, Ralf Schlüter, and Hermann Ney. Language modeling with deep transformers. In *Interspeech 2019*. ISCA, sep 2019.

[4] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Honza Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

[5] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[6] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *INTERSPEECH*, 2012.

[7] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.